AMERICAN UNIVERSITY OF BEIRUT

PREDICTING BIRTH DEFECTS USING COST SENSITIVE MACHINE LEARNING

by
AHMAD MOHAMAD-ALI HAMANDI

A thesis
submitted in partial fulfillment of the requirements
for the degree of Master of Science
to the Department of Computer Science
of the Faculty of Arts and Sciences
at the American University of Beirut

Beirut, Lebanon
January 2021

AMERICAN UNIVERSITY OF BEIRUT

PREDICTING BIRTH DEFECTS USING COST SENSITIVE MACHINE LEARNING

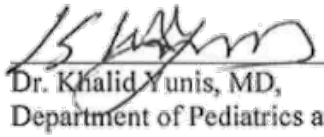
by

AHMAD MOHAMAD-ALI HAMANDI

Approved by:

Dr. Fatima Abu Salem, PhD, Advisor
Department of Computer Science

Dr. Hassan Dhaini, PhD, Member of Committee
Department of Environmental Health



Dr. Khalid Yunis, MD, Member of Committee
Department of Pediatrics and Adolescent Medicine

Dr. Mohamed El Baker Nassar, PhD, Member of Committee
Department of Computer Science

Date of thesis/dissertation defense: 14th of January 2021

AMERICAN UNIVERSITY OF BEIRUT

THESIS, DISSERTATION, PROJECT RELEASE FORM

Student Name: _____Hamandi_____Ahmad_____Mohamad-Ali_
Last First Middle

● Master's Thesis ○ Master's Project ○ Doctoral Dissertation

☐ I authorize the American University of Beirut to: (a) reproduce hard or electronic copies of my thesis, dissertation, or project; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes.

☑ I authorize the American University of Beirut, to: (a) reproduce hard or electronic copies of it; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes

after: **One ---- year from the date of submission of my thesis, dissertation, or project.**
Two ---- years from the date of submission of my thesis, dissertation, or project.
Three ✓- years from the date of submission of my thesis, dissertation, or project

_____February 7, 2021_____

Signature Date

Acknowledgements

First of all, I want to thank the Almighty Allah for giving me this opportunity and the power to finish my master's degree. I want to thank HIM for HIS companionship throughout this long journey.

I am deeply thankful to my family for their continuous support and prayers; my mother, my father, my two brothers and my lovely fiance. Thank you for your patience, thank you for your support, thank you for being there when I mostly needed you.

I want to thank my advisor for her continuous help and support for making this project a reality, and for teaching me the skills when needed to accomplish this thesis.

An Abstract of the Thesis of

Ahmad Mohamad-Ali Hamandi for Master of Science
Major: Computer Science

Title: Predicting Birth Defects Using Cost Sensitive Machine Learning

Many studies were made to tackle the issue of birth defects. Most of them focus on medical causes only like consanguinity degree, folic acid intake, diabetes, etc. A set of studies were made on the effect of ambient air pollution on the health of newborns. Studies that involve the use of artificial intelligence to detect birth defects from ambient air pollution are rare. In my thesis, I use data science and machine learning to build a tool that predicts birth defects from ambient air pollution and medical data. In our study, we used several techniques to build trustable and interpretable predictions for imbalanced data. To tackle the issue of imbalanced data, we used several techniques. One technique was to perform data sampling; in this technique, we balance the data before performing any learning process, this technique was beneficial for many models; for instance the performance of logistic regression was improved when using oversampling techniques and F2 score recorded a 5% improvement. Another technique is called cost-sensitive learning, in this technique, we use specific algorithms that can perform modeling for imbalanced data. Also, we performed feature selection to identify which features are the most important features for our study, and we were able to identify several features in the feature selection process that were confirmed through a process called SHAP. Feature selection is a process that reduces the number of features in a machine learning process to enhance the modeling performance. SHAP is a technique used to highlight the contribution of each feature in a specific prediction. Our main focus was to predict the probability of having a birth defect and to export trustable and explainable results to the end-user. After comparing several models using several configurations, we found that cost sensitive logistic regression and support vector machines were the best performing ones. Cost sensitive logistic regression was the best for performing both classification and probability prediction. Support vector machines was the most similar model in terms of performance. Cost sensitive logistic regression

recorded an F2 score of 93.46% on the training data when performing classification. For probability prediction, cost sensitive logistic regression recorded a Brier Skill Score of 74.23%, and Support Vector Machines recorded a Brier Skill Score of 69.21%. Baseline Brier Skill Score is 5%, F2 score recorded 0% as a baseline performance by uniformly classifying all instances to the majority class. We were able to identify that cost sensitive logistic regression is the best in terms of training time and ease-of-use; it is faster and less biased compared to other models, cost sensitive logistic regression tends to make less mistakes over frequent patterns of the data. Ease-of-use is defined here as a model that can predict birth defects with a smaller number of features and that can perform early prediction for birth defects during the first few weeks of pregnancy, some of these features are consanguinity degree, mother age, folic acid consumption before pregnancy and chronic disease, BMI, exposure to AIR pollutants prior window of risk and during window of risk. Also, all models in our study revealed same or better performance when running them on selected features, therefore, in terms of ease-of-use they are all aligned. Also, SHAP revealed the same trustworthiness for both selected models. SHAP highlights the contribution of each feature in the prediction process, some of these contributions were aligned with the literature, which gives more trust to the model that we are using, we found that features like consanguinity degree, mother age, folic acid consumption before pregnancy and chronic disease are top contributors to the decision of the models, in addition to other air pollutants, also SHAP showed the contribution of each feature with a certain direction; this means that it allowed us to detect if a specific feature will give a higher or lower probability of having a specific birth defect. For example, one of the findings of SHAP is that it showed that consumption of folic acid intake before pregnancy will lead to lower risk of getting a birth defect, higher mother age and lower mother education will lead to higher probability of getting a birth defect, which is aligned with the literature.

Contents

Acronyms

AAP	Ambient Air Pollution
BD	Birth Defect
GU	Genitourinary
CHD	Coronary Heart Disease
CNS	Central Nervous System
GI	Gastrointestinal
MUSC	Musculoskeletal
SMOTE	Synthetic Minority Over-sampling TEchnique
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negaitive
AUC	Area Under the Curve
ROC	Receiver Operating Characteristic
BSS	Brier Skill Score
ADASYN	Adaptive synthetic Sampling
SVM	Support Vector Machine
LDA	Linear Discriminant Analysis
QDA	Quadratic Discriminant Analysis
GNB	Gaussian Naive Bayes
MNB	Multinomial Naive Bayes
GPC	Gaussian Process
ENN	Edited Nearest Neighbours

Chapter 1

Introduction

A study published by World Health Organization (WHO) shows that over 300,000 deaths occur per year in infants under four weeks due to birth defects [1], these statistics do not include any terminated pregnancies, stillbirths, life-long disabilities, or health conditions accompanying newborn throughout his/her life. This fact suggests that the number of affected children and people with birth defects is much higher. When the study was published approximately 50% of the congenital anomalies were not linked to any specific cause. Among the causes listed in the other 50%, the study shows that advanced maternal age, consanguinity between parents, and exposure to environmental pollution are direct causes of birth defects. Ambient Air Pollution (AAP) is causing a considerable environmental disease burden (Tobollik et al. 2015) [2]. AAP includes the emissions of several ranges particulate matter (PM) like $PM_{2.5}$ for example which over time increases the risk of morbidity and mortality (Hoek et al. 2013; Liang et al. 2014) [3], [4]. Nitric Oxides NO_x is also a product of fuel combustion that was found to be associated with a higher BD risk (Xiong et al., 2019) [5]. Other air pollutants include SO_x, more specifically SO_2, one of the air pollutants that is produced by engines. SO_2 can easily react with other pollutants in the air forming more harmful secondary pollutants, part of which contributes to particulate matter (EPA, 2019).

Congenital malformations can range from minor to severe or fatal malformations (Ren et al., 2018) [6]. Several studies have reported a strong association between AAP and BD while other studies were negative (Schembari et al., 2014; Gianicolo et al., 2014) [7], [8]. A meta-analysis reported an association between exposure to air pollution and congenital malformations, particularly congenital heart defects, neural tube defects, respiratory system defects, orofacial clefts, and digestive system defects (Chen et al. 2014) [9].

Another study conducted on 8,865 Chinese cases and examining the association between birth defects and exposure to NO_2, SO_2, and PM_{10} during the first trimester showed an incremental risk of 10.3% per 10 $\mu g/m3$ of NO_2 for neural

tube and cardiovascular defects (Wang et al., 2018) [10].

The issue of ambient air pollution in Lebanon is increasing. The emission of air pollutants from various sources including industrial activities, diesel power plants, domestic generators compensating for daily long power outages, fuel combustion and evaporation from heavy traffic, and the recent solid waste crisis (Waked & Afif 2012; Massoud & Merhebi 2016; Saliba et al. 2006) [11], [12].

Many studies were made to tackle the issue of BD, BD with medical data and BD with AAP, but very few of them involve the use of artificial intelligence or data science.

Data science is an interdisciplinary field aiming to turn data into real value. Data may be structured or unstructured, big or small, static or streaming. The value may be provided in the form of predictions, automated decisions, models learned from data, or any type of data visualization delivering insights. Data science includes data extraction, data preparation, data exploration, data transformation, storage and retrieval, computing infrastructures, various types of mining and learning, presentation of explanations and predictions, and the exploitation of results taking into account ethical, social, legal, and business aspects. [13]

Data science and machine learning are two strongly connected fields that allow us to build predictive tools. They are used today to solve the world's biggest problems in a wide range of fields; this can be used in banking, economy, medicine, etc. Almost every set of data can be used by a data science tool to translate it into meaningful information.

In this thesis, we use data science and machine learning to build a predictive data-driven assessment tool that can learn from imbalanced AAP and medical data to predict birth defects. The tool predicts birth defects, through both binary classification and probabilistic prediction. The study involved the use of several data visualization and exploration techniques such as clustering, TSNE tools, and Google Facets. Data Cleaning and data engineering are performed as data preparation for the data to be used during the machine learning process. Also, we used feature selection to get reveal sets of highly important features for the decision of the models. We performed machine learning on a very wide set of models to get the best performing ones. Cost sensitive logistic regression is the best performing model for both classification and probability prediction with a 93.46% and 74.23% scores for both F2 score and Brier Skill Score respectively. Support vector machines revealed a similar performance in the probability prediction process and recorded a Brier Skill Score of 69.21%. The performance of the used models was assessed using several other techniques to get the best configurations, these techniques included the use of several metrics like accuracy, G-mean, F2 score,

AUC ROC, AUC Precision-Recall, Brier Skill Score, ROC and Precision-Recall. Also, we performed learning over clustered data, learning over features selected from the feature selection process. Modeling over features selected in the feature selection process is an important finding, models showed same or in some cases better performance when using these selected features, this is important because it allows the user to use less data to get the same predictions, also, it allows the user to get early prediction to the birth defects during the first few weeks of pregnancy. In addition, we performed some advanced evaluation procedures to find the best performing model, the first technique is top k precision and recall comparison, this technique allows us to compare the precision and recall values at top k, which reveals how well would the model perform in case of limited access to the clinics or in case of emergency. The second technique that we performed is the comparison of classifier predictions, this technique allowed us to compare the best performing models and to find their pros and cons. The third technique that we focused on was the study of bias, we tackled the issue of finding if any model is biased towards any sector of the data or not. Also, we used several techniques to build trust for the end-user; the most important one was SHAP, which is a technique that reveals the contribution of each feature during the prediction process, SHAP revealed several important findings, the first one is that features that recorded a high contribution during the prediction process were aligned with selected features in the feature selection process, this finding allows the user to trust the used models, second finding is that some of these features are aligned with the literature, for instance, consanguinity degree was one of the features that was classified as a top contributor, SHAP showed that higher consanguinity degree leads to higher chance of having a birth defect, other top contributor are mother age, folic acid consumption before pregnancy and chronic diseases, SHAP showed that higher consumption of folic acid before pregnancy leads to lower risk of having a birth defect. Other techniques were used to tackle the issue of imbalanced data, these were several ones; we used a technique called probability threshold moving, we used cost-sensitive learning and sampling techniques. Cost-sensitive learning technique is a modification applied to the machine learning model so that it performs less mistakes on a specific class, sampling technique is a technique used to project syntactic instances to the training data so that it would be exposed to a bigger and unbiased dataset. As a comparison, logistic regression recorded a 3-4% better F2 score on the training dataset when using cost sensitive and data sampling techniques, sampling techniques also allowed this same model to have a better F2 score that is improved by 5%. Finally, we provided a clinical prototype that allows clinicians to perform the prediction easily in the clinic.

In a nutshell, my thesis' main objective is to answer the following question: "Can AI predict birth defects?".

Chapter 2

Literature Review

A statistical study by World Health Organization (WHO) was published in 2010 that shows that birth defects is causing the death of 330,000 newborn yearly for babies who didn't live more than 4 weeks. The study shows these statistics do not include any terminated pregnancies, stillbirths, life-long disabilities or health conditions accompanying newborn throughout his/her life. When the study was published approximately 50% of the congenital anomalies were not linked to any specific cause. Among the causes listed in the other 50%, the study shows that advanced maternal age, consanguinity between parents and exposure to environmental pollution are direct causes to birth defects. Also it is important to mention that the study shows that "The vast majority (94%) of congenital anomalies occur in low- and middle-income countries".

Many researches were made to detect causes of birth defects, most of them followed a statistical approach rather than being machine learning oriented. Few studies were made using machine learning. For instance, some papers discuss the detection of CHD (congenital heart defects). An article was published on the classification for CHD using 3 different algorithms: Logistic Regression, Support Vector Machines and Random Forest [14], the authors of the paper were able to achieve accuracies near 90% for the 3 models. Another paper addressed the same topic with a variety of birth defects [15], the authors of the paper performed several modeling techniques to detect pregnancy outcomes using medical data, and were able to detect it with a high accuracy (above 85%) however they focused on medical data only, without going into socio-economical and ambient air pollution causes.

Another study was made in 2018 that use machine learning to study the effect of PM_{10} on the effect of the maternal exposure to PM10 on the risk of congenital heart defect [16]. The issue with the paper is that it only studies the effect of 1 pollutant, and 1 birth defect.

The topic of classification from imbalanced data was discussed in [17]. The

author presents several frameworks and methodologies that can be followed to learn from imbalanced data. The author presents how we can choose the correct metrics for evaluating the performance of any used machine learning tool. Learning from imbalanced data was present in several forms; for example, how data sampling techniques can be used to achieve better classification performance, how tuning can be used to achieve the best performance, how scaling can be used or cost sensitive machine learning algorithms can be used to achieve good performance for learning from imbalanced data.

Also, the authors of [18] discuss the topic of learning from Imbalanced data. Similar to [17] this book discusses how to choose a set of metrics for evaluation, and how accuracy alone is not an indicative metric. The authors present different approaches to deal with minimal data, this includes algorithmic intervention through cost sensitive learning or the use of other algorithms, and data level modification through the use of several data augmentation techniques.

Cost sensitive learning techniques are popular in medical problems. In [19] the authors used a cost sensitive SVM model to predict post-operative life expectancy of lung cancer patients.

Seizure prediction using Electroencephalogram (EEG) and Support Vector Machines (SVM) classification was studied in [20]. The authors trained Support Vector Machines (SVM) and Cost Sensitive Support Vector Machines (CSVM) over data extracted from EEG. The authors used Feature Extraction and double cross-validation techniques as preprocessing to the data before the learning process. For assessment they used F-beta score with beta=2, which is known as F2 score in addition as well as sensitivity. Their final findings was an improvement of sensitivity from 93.7% to 97.5%.

Cost Sensitive classification for lung cancer nodules was studied in [21]. Cost sensitive learning was used to build a system that detects lung cancer from CT images. The authors used CT images of potential lung cancer patients to build the classifier. The major issue that the authors were trying to solve is the reduction of False Positives (FP) for their machine learning tools. The cause of data imbalance in the dataset is that the healthy population is highly represented compared to the ill population. The authors used different techniques to optimize their cost sensitive classifiers; they used oversampling, undersampling, AdaCost, Random Forest, Cost Sensitive Support Vector Machines and Linear discriminant analysis. For assessment they used two metrics: G-mean and AUC, then they compared the models that outperformed using ROC curve. Best classifiers include Cost sensitive support vector machines, reported G-mean values were around 83.7% to 98% and AUC scores ranged from 65.5% to 98.6%.

Similarly, in [22] dealt with classification of malign and benign classes for thyroid nodules. The issue with the given data is that much more patients have benign nodules than those with malicious ones. Before the application of machine learning algorithms, variants of SMOTE oversampling technique was applied to the data in order to create a balance between both classes.

Another imbalanced learning problem was described in [23], where the authors want to differentiate between P2P botnet network traffic and normal traffic. The data was imbalanced because more normal network traffic was present compared to abnormal ones. They used cost-sensitive random forest with random undersampling to solve this issue.

Another example where imbalanced learning techniques were used. The authors of [24] built a tool that detects early school dropouts. The aim of this study was to detect if students will leave school during the course or not. Given the fact that most students don't dropout from school, the data was imbalanced. The authors used SMOTE oversampling to balance the data in order to perform the needed learning.

The author of [17] presents a framework for probability prediction from imbalanced data using Haberman Breast Cancer Survival dataset. The dataset describes the survival of breast cancer patients. It contains 306 cases only, 225 are negative and 81 are positive. So overall the minority class represents around 26% of the data and the majority represents 84%. A set of models was used to predict the probability of survival of the patients: logistic regression, linear discriminant analysis, quadratic discriminant analysis, Gaussian Naive Bayes, Multinominal Naive Bayes and Gaussian Process. Also, the author tested the models using different scaling options; he used standardization and compare it to power transform. To make the prediction, he used repeated cross validation and focused on Brier Skill Score as a major metric to assess the performance of his models. The highest Brier Skill Score he was able to report was 0.110 using Logistic Regression with power transform.

An example of imbalanced classification problem was presented in [17] through the classification of oil spills. The target of this study is to identify oil spills from Satellite Radar Images. The dataset was introduced in the 1998 paper by Miroslav Kubat, et al. titled Machine Learning for the Detection of Oil Spills in Satellite Radar Images. The overall number of cases in this dataset is 937 cases; 896 of them are negative cases and 41 are positive. The overall distribution of the data is around 96% to 4%. Both authors used G-mean as a major metric to assess the performance of the used models. The intuition behind using G-mean was to balance precision and recall because both of them are important for them. A baseline G-mean value was derived by uniformly classifying all the data to the

majority class; in this case an average G-mean value of 47.8% was reported. Then three models were tested using a repeated cross validation process; these models are logistic regression, linear discriminant analysis and Gaussian Naive Bayes. Since Logistic Regression performed better in the first test set, the author tries then to boost its the performance by balancing the class weights and testing this model with different scaling options. Scaling options that were tested are: normalization, standardization and power transform. Another set of tests was done by using SMOTEENN, a mixed oversampling and undersampling technique. The best reported G-mean was 87.3% with Logistic regression using power transform and SMOTEENN.

Chapter 3

Overall Aims

1. Build a predictive classifier that predicts birth defects from a panel of medical, environmental and socio-economical data.
2. Build a probabilistic classifier that predicts the probability of having a birth defect from a panel of medical, environmental and socio-economical data.
3. Study the performance of cost sensitive machine learning algorithms over imbalanced data.
4. Compare the performance of machine learning algorithms after using cost sensitive techniques.
5. Build interpretability tools to interpret the results of machine learning outcomes and to build trust for the user.
6. Use Feature Selection techniques and Unsupervised learning to detect some meaningful patterns.

Chapter 4

Data Collection

Data used in this study was collected in Lebanon between June 2014 and June 2017. The study consists of two types of participants: cases and controls. Cases represent mothers of infants with birth defects. Controls represent mothers of infants without birth defects. Cases are identified from the Lebanese National Birth Defect Registry (NBDR) which is maintained by the Ministry of Public Health (MoPH), while controls are identified from the records of the National Collaborative Perinatal Neonatal Network (NCPNN), a non-profit Lebanese network reaching 34 Lebanese hospitals, with its coordinating center located at the American University of Beirut Medical Center, and which collects and maintains perinatal and neonatal information. Air quality data was acquired from the national air quality network (ERML) managed by the Lebanese Ministry of Environment. An Institutional Review Board (IRB) approval was obtained from the American University of Beirut before beginning the study.

Chapter 5

Code Book

Below, we display the used code book used in this thesis.

Common Features	
Educational level of the mother	Numeric: 0=Unknown, 1=Illiterate, 2= Read/Write, 3= Elementary, 4=Intermediate, 5=Secondary, 6=Technical, 7=University (undergraduate), 8=University (Graduate)
Degree on consanguinity	1= First Degree, 2=Second Degree, 3= Third Degree or more
Sex of the baby	0=Male, 1=Female
Status of parental consanguinity	0=No, 1=yes
Tobacco exposure including cigarettes and arguileh	1= cigarettes 2=arguileh 3=cigarettes and arguileh
Status of alcohol intake during pregnancy	0=No, 1=yes
Status of folic acid intake before pregnancy	0=No, 1=yes
Medical History - presence of chronic condition	0= no chronic conditions 1= one or more chronic condition
mothers age	Numeric
number of viable gestations out of the total parity	Numeric
Total number of abortions	Numeric
Number of Spontanous abortions	Numeric
Number of Induced abortions	Numeric
Number of living children	Numeric
Body mass index (Measure of obesity)	Numeric
Number of cigarettes per week	Numeric

Table 5.1: Code Book - Common Features

Air Pollution for All Defects	
PM2.5_week_number - Average exposure to PM 2.5 on the week specified	Numeric – used for all defects data only for each week of the 44 weeks (total of 132 features)
NO2_week_number - Average exposure to NO2 on the week specified	Numeric – used for all defects data only for each week of the 44 weeks (total of 132 features)
SO2_week_number - Average exposure to SO2 on the week specified	Numeric – used for all defects data only for each week of the 44 weeks (total of 132 features)

Table 5.2: Code Book - AAP for All Defects

Air Pollution for Per Birth Defect	
Average exposure of PM2.5 of all weeks prior to window of risk	Numeric – used for per defect data only
Average exposure of PM2.5 for the weeks that represent the window of risk	Numeric – used for per defect data only
Average exposure of NO2 of all weeks prior to window of risk	Numeric – used for per defect data only
Average exposure of NO2 for the weeks that represent the window of risk	Numeric – used for per defect data only
Average exposure of SO2 of all weeks prior to window of risk	Numeric – used for per defect data only
Average exposure of SO2 for the weeks that represent the window of risk	Numeric – used for per defect data only

Table 5.3: Code Book - AAP for Per Birth Defect

Chapter 6

Data Cleaning and Preprocessing

6.1. Missing Data. We had to solve the issue of missing data. First, we looked at the reason for missing data. The data was integrated from several sources; then inserted in one sheet, therefore; parts of the data were not available for controls or not available for cases. For these columns, the only solution was to drop them.

6.1.1. Missing Data for Controls and Cases. After removing the columns mentioned above, we had to deal with columns that have missing data but where we can do some data replacement. We did this using several techniques:

- Delete data if no logical replacement available
- Perform data imputation by replacement empty cells by the average of the column's data when the data is numerical and normally distributed
- Perform data imputation by majority when the data is categorical
- Perform data imputation by injecting new category for binary categories

The performed imputation is summarized in the table below:

Column name	% of missing		% out of total	% out of control or case	
Mom education	2.40%	case	0	0	-
		control	2.41	2.52	Removed
Consanguinity degree	76.90%	case	0	0	-
		control	1.52	1.59	Removed
Sex	0.70%	case	0.08	1.83	created "other" category
		control	0.67	0.7	created "other" category
Parental consanguinity	1.20%	case	0	0	-
		control	1.17	1.23	Removed
Tobacco Exposure	0.20%	case	0.18	3.98	Imputed by majority
		control	0	0	-
# of cigarettes/week	91.20%	case	0.19	4.28	Impute by majority frequency
		control	5.27	5.51	Removed
Alcohol during	1.50%	case	0.14	3.06	Imputed by majority
		control	1.39	1.45	Imputed by majority
FA intake	1.90%	case	0	0	-
		control	1.92	2.01	Removed
Medical History	0.90%	case	0	0	-
		control	0.9	0.94	Removed
Mom's age	1.10%	case	0.07	1.53	Impute by average
		control	1.08	1.13	Impute by average
Parity	0.70%	case	0	0	-
		control	0.71	0.74	Removed
Total abortions	0.80%	case	0	0	-
		control	0.82	0.86	Removed
Spontaneous abortion	1.10%	case	0	0	-
		control	1.1	1.15	Removed
Induced abortion	1.50%	case	0	0	-
		control	1.5	1.57	Removed

Chapter 7

Methodology

7.1. Approach. Our machine learning approach is split into four parts:

1. Clustering: Clustering was done to evaluate the splits of the data and to get some insight into the distribution of the data.
2. Feature Selection: We perform feature selection using several techniques to classify features that are highly correlated with birth defects.
3. True/False classification, i.e. we predict that this person will have a birth defect, or this other person will not have a birth defect.
4. Predicting probabilities, we predict how probable will this person have a birth defect. i.e. we predict that this person has a 52% chance to have a birth defect, or this other person has a 90% chance to have a birth defect. A higher probability means a higher chance of having a birth defect.

7.2. Used Data. The data that we are using to train our models is an imbalanced case/control data; it contains 6124 rows. 327 are cases (positive) and 5797 are controls (negative). Therefore, the distribution is around 5.3% to 94.7% for positive and negative classes respectively.

7.3. Clustering. We applied K-means Clustering using Gower's distance. This technique is specialized in clustering mixed data types, i.e. categorical and numerical. Gower distance is computed as the average of partial dissimilarities across individuals. Each partial dissimilarity (and thus Gower distance) ranges in [0 1].

For a numerical feature, partial dissimilarity is the ratio between 1) absolute difference of observations and 2) maximum range observed from all individuals. For a qualitative feature partial dissimilarity equals 1 only if observations have a

different value. Zero otherwise.[25]
The code was implemented using R.

7.4. Feature Selection.

7.4.1. What is Feature Selection. A machine learning dataset for classification or regression is comprised of rows and columns, like a spread-sheet. Rows are often referred to as instances and columns are referred to as features. Feature selection is a technique used to reduce the number of input variables to those that are believed to be most useful to a model to predict the target variable. Feature selection also refers to a set of techniques that select a subset of the most relevant features (columns) for a dataset. Fewer features can allow machine learning algorithms to run more efficiently (less space or time complexity) and be more effective.
Briefly, feature selection will allow us to perform the three following tasks:

1. Data interpretation.
2. Model interpretation.
3. Data interpretation.

7.4.2. Types of Feature Selection. We have 2 types of Feature Selection:

1. Supervised – uses target variable (outcome)
2. Unsupervised – does not use target variable

Supervised methods intend to remove irrelevant variables (not important to the target variable). Unsupervised methods intend to remove redundant variables (Correlated variables and features with low variance).
Supervised methods are divided into three categories:

1. Filter: Select subsets of features based on their relationship with the target - Uses statistical measures to compute correlation or dependance score between input variables that can be filtered and chooses the most relevant features.
2. Wrapper: Search subsets of features that perform according to a predictive model.
3. Intrinsic: Algorithms that perform automatic feature selection during training.

7.4.3. Supervised Technique - Wrapper. Wrapper feature selection methods create many models with different subsets of input features and select those features that result in the best performing model according to a performance metric – this technique is computationally expensive. Wrapper methods evaluate multiple models using procedures that add and/or remove predictors to find the optimal combination that maximizes model performance, example of these methods is RFE.

7.4.4. Supervised Technique - Filter. Filter feature selection methods use statistical techniques to evaluate the relationship between each input variable and the target variable, and these scores are used as the basis to rank and choose those input variables that will be used in the models.

7.4.5. Supervised Technique - Automated Algorithms. Intrinsic feature selection can be referred to machine learning algorithms that perform feature selection automatically. This includes algorithms such as penalized regression models like Lasso and decision trees, including ensembles of decision trees like random forest.

7.4.6. Statistics for Feature Selection. There is a specific statistical method that should be used for each combination of input/output types. In our case since we are dealing with mixed data types, different algorithms can perform feature selection; however, the most relevant ones are RFE and Tree-based models.

7.4.7. What is the Best Feature Selection Method. This is unknowable. Just like there is no best machine learning algorithm, there is no best feature selection technique. At least not universally. Instead, you must discover what works best for your specific problem using careful systematic experimentation. The performance of feature selection methods can be assessed using the traditional metrics like accuracy or f-measure etc. For instance, RFE is produced by several machine learning models. We can explore the performance of these base models and compare the produced results. The best performing model will produce the most accurate feature selection results. It is actually more likely than not for two FS approaches to reveal different sets of important features. The final verdict is incumbent on the actual modeling performance which would confirm whether or not those features played a highly decisive role. This can be seen as reporting the union of the results produced by several feature selection techniques, not the intersection.

7.4.8. How can we filter input variables. There are two main techniques for filtering input variables. The first is to rank all input variables by their score and select the k-top input variables with the largest score. The second approach is

to convert the scores into a percentage of the largest score and select all features above a minimum percentile.

7.4.9. Using RFE for Feature Selection. Recursive Feature Elimination or RFE for short is an efficient approach for eliminating features from a training dataset for feature selection. RFE is a popular feature selection algorithm because it is easy to configure and use, and because it is effective at selecting those features (columns) in a training dataset that are more or most relevant in predicting the target variable. There are two important configuration options when using RFE:

1. The choice in the number of features to select.
2. The choice of the algorithm used to help choose features.

The performance of the method is not strongly dependent on these hyperparameters being configured well.

7.4.10. RFE – How it works?. RFE is a wrapper-type feature selection algorithm. This means that a different machine learning algorithm is given and used in the core of the method, is wrapped by RFE, and used to help select features. This contrasts with filter-based feature selections that score each feature and select those features with the largest (or smallest) score. RFE is a wrapper-style feature selection algorithm that also uses filter-based feature selection internally. RFE works by searching for a subset of features by starting with all features in the training dataset and successfully removing features until the desired number remains. This is achieved by fitting the given machine learning algorithm used in the core of the model, ranking features by importance, discarding the least important features, and re-fitting the model. This process is repeated until a specified number of features remain. When the full model is created, a measure of variable importance is computed that ranks the predictors from most important to least. At each stage of the search, the least important predictors are iteratively eliminated prior to rebuilding the model. Features are scored either using the provided machine learning model (e.g. some algorithms like decision trees offer importance scores) or by using a statistical method.

7.4.11. Feature Importance. Feature importance refers to techniques that assign a score to input features based on how useful they are at predicting a target variable. There are many types and sources of feature importance scores, although popular examples include statistical correlation scores, coefficients calculated as part of linear models, decision trees, and permutation importance scores. Feature importance scores play an important role in a predictive modeling project, including providing insight into the data, insight into the model, and the basis

for dimensionality reduction and feature selection that can improve the efficiency and effectiveness of a predictive model on the problem. Feature importance refers to a class of techniques for assigning scores to input features to a predictive model that indicates the relative importance of each feature when making a prediction. The relative scores can highlight which features may be most relevant to the target, and the converse, which features are the least relevant. This may be interpreted by a domain expert and could be used as the basis for gathering more or different data. The scores produced by feature importance can be used in a range of situations in a predictive modeling problem, such as:

1. Better understanding the data.
2. Better understanding a model.
3. Reducing the number of input features.

Feature importance can be used to improve a predictive model. This can be achieved by using the importance scores to select those features to delete (lowest scores) or those features to keep (highest scores). This is a type of feature selection and can simplify the problem that is being modeled, speed up the modeling process (deleting features is called dimensionality reduction), and in some cases, improve the performance of the model. Ranking predictors in this manner can be very useful when sifting through large amounts of data.

7.4.12. Feature importance – Application. Feature importance scores can be fed to a wrapper model, such as the SelectFromModel class, to perform feature selection. Each feature importance technique has the potential to rank the importance of input features differently, creating a different view of the data. As such, there is no best feature importance technique. If the goal is to find the subset of input features that result in the best model performance, then a suite of different feature selection techniques should be tried including different feature importance methods. There are many ways to calculate feature importance scores and many models that can be used for this purpose. Nevertheless, the scores between feature importance methods cannot be compared directly. Instead, the scores for each input variable are relative to each other for a given method. In this tutorial, we will look at three main types of more advanced feature importance; they are:

1. Feature importance from model coefficients.
2. Feature importance from decision trees.
3. Feature importance from permutation testing.

7.4.13. Coefficients as Feature Importance. Linear machine learning algorithms fit a model where the prediction is the weighted sum of the input values. Examples include linear regression, logistic regression, and extensions that add regularization, such as ridge regression, LASSO, and the elastic net. All of these algorithms find a set of coefficients to use in the weighted sum in order to make a prediction. These coefficients can be used directly as a crude type of feature importance score.

7.4.14. Logistic Regression Feature Importance.

1. Logistic regression produces coefficients per feature.
2. Decision Tree can also produce feature importance coefficients.
3. CART.
4. Random Forest.
5. Permutation Feature Importance.
6. Feature Selection with Importance.

7.4.15. How to Interpret Feature Selection Scores. You can interpret the scores as a specific technique relative importance ranking of the input variables. The importance scores are relative, not absolute. This means you can only compare the input variable scores to each other as calculated by a single method.

7.4.16. How Do You Use The Importance Scores?. Scores can be used for:

- Data interpretation.
- Model interpretation.
- Feature selection.

7.5. Choosing the Best Machine Learning Tool and Options. Since there is more than one approach for learning from imbalanced data, one of our aims is to build a framework, mainly software, that will help us to do a comparative analysis of different techniques.

To build the software we had to answer several questions, listed below:

- Which machine learning algorithm works best on the given data?

  - Literature Review
  - Start with Baseline Performance and improve
- Is data labeling or encoding required?
  - Dealing with mixed data types (categorical/numerical/binary) vs. dealing with numerical/binary
    - Use One hot encoding to have numerical/binary.
  - Why is it important for us to transform categorical data into non-categorical?
    - Data sampling options very limited for categorical data.
- Is scaling required?
  - Look at data distribution to get some insight
  - Look at algorithms that are sensitive to unscaled data
  - Test models without scaling
  - Standardization (z-score) for Normally distributed data and Normalization (min-max) for other columns
  - Use Min-max or Z-scale for all the data
  - Use Min-max followed by Power Transform for all the data
- How to maintain model stability?
  - Repeated Stratified K-fold, Stratified Train / Test splits
- How to evaluate our results?
  - Basic evaluation of TP, TN, FP, FN
    - Why TP, TN, FP, FN are not indicative and how to replace them with more indicative metrics?
  - What is the best metric in our case?
    - Classification
      - G-mean
      - F2-score
      - AUC
      - Accuracy
    - Probabilities Prediction
      - BSS
      - AUC ROC
      - AUC PR

7.6. Choosing the Best Machine Learning algorithm. Choosing the best machine learning algorithm is done by testing different machine learning algorithms over the data and comparing the outcomes. In our study, we do the comparison over the model itself while changing the scaling options and while introducing cost-sensitive learning techniques. We always compare to a baseline performance, which is the best performance a model can achieve without introducing any data sampling or cost-sensitive techniques. We conclude by performing a comparative assessment of all the models.

7.7. Cost Sensitive Learning and Cost Matrix. Cost-sensitive learning is an algorithm modification that addresses class imbalance. The main concept is based on strongly penalizing mistakes for some classes defined in what is called a cost matrix. For example, if we specify that the minority class has a higher cost than the majority class, then the used model will tend to do fewer mistakes for the minority during the learning process. Penalization is a technique used to reduce overfitting over the majority class, which is needed when the percentage of the majority class is very high.

7.7.1. Cost Matrix. The process of cost-sensitive learning is dependent on cost matrices; a cost matrix is a matrix that quantifies the penalty given to each class. There are several ways to build a cost matrix. For example, suppose we are learning from financial data, where the importance of one class can be quantified, i.e. one can say that: not giving a loan to a customer who will fail to pay back to the bank is 100 times more costly than losing this customer, so it is easy to build a cost matrix in this case because the loss and gain are explicit. However, in the medical field, quantifying these numbers can be related to some ethical questions. Other ways to build the cost matrix state that we can build the cost matrix using the inverse of the data distribution. In our case, this means setting the importance of the class that has 95% of the population to 5 and setting the importance of the class that has 5% of the population to 95. We generalize this approach with the help of tuning as follows: instead of choosing the cost matrix explicitly, we pass multiple combinations of cost matrices; we include one that has the 95/5 approach mentioned above and other combinations that cover both extremes of the data, i.e. 1/100, 10/90, 20/80 .... 90/10, 100/1.
We try to fit all these cost matrices to the given data, and we choose the cost matrix that gives us the best score for which we are doing the tuning process; for example, best accuracy or best F2 Score, etc.

7.7.2. Data Sampling. A second approach that we use to improve our results is data sampling techniques. This approach is split into two parts:

- Minority Over Sampling
- Minority Over Sampling with Majority Under sampling

Minority Over Sampling

For minority over-sampling we use SMOTE-NC. This technique oversamples the minority class by injecting synthetic samples.

SMOTE NC: Synthetic Minority Over-sampling Technique for Nominal and Continuous. SMOTE first selects a minority class instance at random and its k nearest minority class neighbors. The synthetic instance is then created by choosing one of the k nearest neighbors b at random and connecting a and b to form a line segment in the feature space. The synthetic instances are generated as a convex combination of the two chosen instances a and b. (Page 47, Imbalanced Learning: Foundations, Algorithms, and Applications, 2013)

Minority Over Sampling with Majority Under Sampling

This is a very similar approach as above, we are using SMOTE for sampling but instead of having the same number of instances in the majority class, these are under sampled. Options here include:

- SMOTE with Tomek Links: SMOTE is used with Tomek Links, which refers to a method for identifying pairs of nearest neighbors in a dataset that have different classes. Removing one or both of the examples in these pairs (such as the examples in the majority class) has the effect of making the decision boundary in the training dataset less noisy or ambiguous.(imbalanced classification with python)

7.8. Calibrated Probabilities. Not all machine learning algorithms are designed to predict probabilities. For example, Logistic Regression can predict probabilities, but SVM cannot. SVM can predict what is called uncalibrated probabilities instead. A model to predict calibrated probabilities must explicitly be trained under a probabilistic framework, such as maximum likelihood estimation. [17]. However, algorithms such as SVM, Decision Trees, and KNN can produce class labels and probability-like scores. Therefore, to use them for probability prediction we must calibrate their probabilities.
Probabilities are calibrated by rescaling their values so they better match the distribution observed in the training data. [17].

7.8.1. How Calibrated Probabilities is Performed. The training data is split into two parts; the first for training and the second for calibration. After training

scores are produced, the calibration process is performed using Platt scaling, which is a technique that uses logistic regression to readjust the output that fits the real distribution of the data.

7.9. Data Contamination. Data contamination can be defined as sharing too much information between training and testing. This can occur in different places in the modeling process; data undersampling, oversampling and data scaling. Scaling the data before splitting it into training and testing can contaminate the training data because it will be exposed to the distribution in the testing set. The same mistake can happen when doing oversampling and undersampling, if we do it on all the data, before splitting it, training data will be exposed to samples from the testing data.
We solve this issue by applying data sampling to the training only. And for scaling, we apply to scale on training data, then we use the same ranges to scale the testing data.

7.10. List of Machine Learning Models.

1. Classification Models
    - Shallow Modeling
        - Logistic Regression
        - Support Vector Machines
        - Decision Trees
        - XGBOOST
        - KNN
        - AdaBoost
    - Cost Sensitive Shallow Modeling
        - Cost Sensitive Logistic Regression
        - Cost Sensitive Support Vector Machines
        - Cost Sensitive Decision Trees
        - Cost Sensitive XGBOOST
        - Cost Sensitive Random Forest
        - AdaCost
        - CatBoost
    - Deep Learning (both normal and cost sensitive)
2. Probability Prediction Models
    - Linear Models

- Logistic Regression
- Cost Sensitive Logistic Regression
- Linear Discriminant Analysis (LDA)
- Quadratic Discriminant Analysis (QDA)
- Gaussian Naive Bayes (GNB)
- Multinomial Naive Bayes (MNB)
- Gaussian Process (GPC)
- Calibrated Support Vector Machines
- Calibrated Decision Tree
- Calibrated KNN

7.11. Is data encoding required?. To answer this question, it is necessary to look at the columns inside our data. In the dataset there are 148 features, only two of them are categorical -Tobacco exposure and gender of the baby- all other columns are numerical.

In our case, the categorical columns were transformed into numerical values using a technique called One Hot encoding. In this technique, we create a new column for each category in a categorical column. And then, the column is filled by either 0 or 1, where 1 indicates the presence of this category and 0 indicates the absence of it. For example, the column indicating the gender of the newborn is replaced by two columns, the first indicates if this is a female and the other column indicates if this is a male. Using this technique we get rid of the categorical data, and we end up with numerical/binary data only.

Overall, the cost for using one-hot encoding is an addition of three extra columns or features to our datasheet. Two extra columns for Tobacco exposure and one extra column for gender.

7.12. Is scaling required for our data?. Two factors indicate if we have to use scaling or not:

1. Data distribution
2. Nature of the model that we are using

7.12.1. Data Distribution. One should check if data is normally distributed or not. In fact, a few columns in the dataset have an almost normal distribution. For instance, BMI, mother's age, and mother's education are all normally distributed, but the rest of the data is not normally distributed. This indicates that scaling is required.

Also, we should mention that the range of the data is different; for example, a mother's education is a number between 1 and 8, a smoking index is a number between 0 and 2800. This affects models such as logistic regression, support vector machines, and all other linear models.

7.12.2. Nature of the model. Linear and distance-based models require or sometimes perform better with scaling; examples of these models: SVM and Logistic Regression.
Rule-based models are scaling insensitive; examples of these models: Decision Trees and XGBOOST

7.12.3. What types of scaling is done in our case. Given the mentioned distribution of the data, scaling is essential before running most of the machine learning algorithms, at least for linear and non-rule based or tree-based models. We use several options for scaling:

- No scaling
- Standardized normally distributed columns and normalized non-normally distributed columns
- Standardize everything
- Normalize everything
- Normalization followed by power transform
- Customized, specifying which scaling applied on which data

7.13. How to maintain model stability. Any chosen model needs to be stable. Having a stable model means that it performs well on training, testing, and new data. To achieve this, we mix several techniques, each one solves a potential instability in the model.

First, all data splitting are stratified, meaning that the data distribution is the same among any subset that is used for training or testing; for example, when we split the data into 80% training and 20% testing, 80% of birth defects will go to the training subset and 20% will go to the testing subset.

Second, we use a technique called k-fold cross-validation. The algorithm used is the following: first, the data is split into k subsets, then we perform training and testing k times. Each time we keep one subset for testing and we use all

other subsets for training followed by testing on the k^{th} subset. The advantage of using k-fold cross-validation is that testing is done over all the data, and this is how underfitting is reduced. The second benefit of this technique is that the testing process is done using the best hyperparameters.

The third technique used is repeated cross-validation. In this technique, after splitting the data into multiple training sets, each training set is replicated ten times, and the training process is performed over the data ten times, this allows the model to have a better performance because it is exposed to the same data more than once.

Fourth, each machine learning algorithm has a wide set of hyperparameters, and we want to test the stability of the machine learning algorithm with a specified set of hyperparameters. To perform this, we use nested k-fold cross validation; in this process, the data is split into k' folds, this process is similar to k-fold cross-validation, the data is split into k' subsets, training is done k' times, each time over all the data except the k'^{th} subset, which is used for testing. The difference now, is that in the training process we perform hyperparameter tuning, and a set of best hyperparameters is selected, then this same set is used in the testing process.

As a summary, this process is referred to as repeated nested k-fold cross-validation.

Because we get the results for each fold of the nested cross-validation process, we used two values for each reported evaluation metric: average and standard deviation reported over the validation and testing results. The average will show us how good is this model and the standard deviation will show us how stable it is. For the evaluation process, both numbers are always taken into account. Training results are reported for the best performing hyperparameter combination over the training dataset, and these same hyperparameters are used for testing.

7.14. How to evaluate the performance of a machine learning algorithm. First, we have to know what is produced each time we run our machine learning algorithm. In our case, the output of the classification process is True/False, or so-called binary. So, whenever we pass the data to our algorithm, it will predict an output, this output can be classified based on the following confusion matrix:

	Real True	Real False
Predicted True	True Positive (TP)	False Positive (FP)
Predicted False	False Negative (FN)	True Negative (TN)

However, counting the number of TP and TN is not indicative in any machine learning process. That's why we use different sets of metrics to evaluate the performance of our model.

Traditionally, for classification problems, accuracy is used to assess the performance of a model. For imbalanced learning, accuracy can be very misleading. As an example of using accuracy only, in our dataset, the data distribution is 5% for cases and 95% for control. In this example, if we classify the whole population as a control we will get an accuracy of 95% because we get 95% of the classification to be correct, yet this may look like a good performance, the model will have zero skill for classifying the cases. That's why we focus on the following metrics:

- For Classification
  - Recall
  - Precision
  - G-mean
  - F2 Score
  - AUC ROC
  - Accuracy
- For Probabilities
  - BSS (Brier Skill Score)
  - AUC ROC
  - AUC PR

7.14.1. Accuracy. Accuracy is defined by the following formula:

$$Accuracy \ = \ \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions} \tag{7.1}$$

Maximizing the accuracy is essential for any classification process; however, for imbalanced classification problems, looking only at this measure can be misleading. Having a higher accuracy is not an indicator that the model can differentiate between any two different categories. As an example, we look at the data that we have, where 95% are controls and 5% are cases, if we uniformly classify the data as controls, the number of correct predictions would be 95%, meaning that the accuracy would be 95% which is a very high accuracy; however, this accuracy in this specific context means that the model cannot detect any birth defect and it is useless.

7.14.2. Recall. Recall's formula below:

$$Recall \ = \ \frac{TP}{TP \ + \ FN} \tag{7.2}$$

The formula of recall has two components TP and FN. Recall is inversely proportional to FN, meaning that whenever FN is low recall is higher. Having too many FNs is not a good index for our model, because each FN can be translated into "someone who has a birth defect that was diagnosed [from our model] as someone normal". The best case for recall is when FN is zero and recall will be equal to 1 or 100%.

7.14.3. Precision.

$$Precision \ = \ \frac{TP}{TP \ + \ FP} \tag{7.3}$$

The formula of precision also has two components: TP and FP. Compared to Recall, it is slightly less important in our case, because precision is inversely proportional to FP, where FP can be translated into "someone who is in a good health and being diagnosed by our model to someone who has a birth defect". The best case for precision is when FP is zero and precision will be equal to 1 or 100%.

7.14.4. G-mean. G-mean or geometric mean, is defined as the square root of the multiplication of the precision and recall

$$G - mean \ = \ \sqrt{Precision \ \times \ Recall} \tag{7.4}$$

Since Precision and Recall are important as mentioned before, sometimes they are not indicative enough; that's why we need a metric that combines both. To get the intuition of using G-mean we look at the formula which is defined above, if we derive it in terms of TP, FN and FP, we get the following:
$G{-}mean \ = \ \sqrt{Precision \ \times \ Recall} = \sqrt{\frac{TP}{TP \ + \ FP} \ \times \ \frac{TP}{TP \ + \ FN}} = \frac{TP}{\sqrt{(TP \ + \ FP){\times}(TP \ + \ FN)}}$.
The optimal solution of this formula is when FP and FN are equal to zero, in this case G-mean will be equal to 1 or 100%. This metric is used to get a balance between precision and recall.

7.14.5. F2 Score. F2 score is a good metric when one class is more important than the other. In our case, it is more important to predict the existence of a birth defect instead of predicting the absence of it. In other words, an unpredicted birth defect has a higher cost than a falsely predicted one.
F2 score is mainly derived from the generalized formula of F1 score, which is called fbeta score.

F1 score assumes that both classes are equally important (similar to G-mean). F2 score puts a higher weight on the Recall (the positive class). The formula of F1 score is the following:

$$F_1\ Score = \frac{2.precision\ .\ recall}{precision\ +\ recall} \tag{7.5}$$

The formula of fbeta score is the following:

$$F_\beta = (1\ +\ \beta^2).\frac{precision\ .\ recall}{(\beta^2.precision)\ +\ recall} \tag{7.6}$$

And F2 score is derived from setting $\beta = 2$

$$F_2\ Score = \frac{5\ .\ precision\ .\ recall}{4\ .\ precision\ +\ recall} \tag{7.7}$$

Numerically, the F1 Score balances between precision and recall, whereas the F2 Score gives a higher weight to recall because of the multiplication of denominator's precision by 4. This means that whenever the number of FN drops F2 gets affected more than the F1 score. F1 score gets affected by both FN and FP equally. The range of both can go from zero to one (0 to 100%), 100 means that FN and FP are equal to zero.

7.14.6. AUC ROC. We can get the AUC ROC by computing the Area Under the Curve of the Receiver Operating Characteristics, which is the curve that illustrates the trade-off between TPR and FPR.
TPR is the Recall metric defined previously, and FPR is the false positive rate, which can be derived by the following formula:

$$FPR\ =\ \frac{FP}{TN\ +\ FP} \tag{7.8}$$

In normal cases AUC ROC ranges between 0.5 and 1 (50% to 100%), 0.5 means that the model cannot differentiate between case and control, 1 means it can differentiate between them without any mistake.

7.14.7. Brier Skill Score. This metric is used for determining probabilities, it is derived from the Brier Score and the Brier Score reference, we can get Brier Score directly during the assessment stage of the model (i.e. training and testing), however, to get an indicative metric, we divide Brier Score by the reference Brier Score, which is the probability of getting positives from the dataset. We then normalize as follows:

$$Brier\ Skill\ Score\ =\ 1\ -\ \frac{Brier\ Score}{Brier\ Score_{ref}} \tag{7.9}$$

In the formula above $Brier\ Score_{ref} = the\ percentage\ of\ birth\ defects\ in\ the\ dataset$

A good Brier Skill Score is a number above the probability of the minority class; for example, if the minority class forms 5% of the population, then any BSS above 0.05 (5%) means that the BSS is well estimated. Any number below 0.05 means that the model failed to predict the probabilities correctly.

7.14.8. AUC PR. We can get the AUC PR by computing the Area Under the Curve of the Precision-Recall curve, which is the curve that illustrates the trade-off between Recall and Precision.
Precision and Recall are already defined.

In normal cases AUC PR ranges between 0.5 and 1 (50% to 100%), 0.5 means that the model has no skill do detect positive cases. A score near 1 means that the model can detect the positive class without any mistake.

7.15. Results.

1. Baseline Results
2. Cost Sensitive Learning Results
3. Data Sampling Results
4. Advanced Algorithms

7.15.1. Baseline Performance. The first part of our tests consists of reporting the results of our set of machine learning algorithms without the introduction of any cost-sensitive techniques.

7.15.2. Cost Sensitive Learning. Cost-sensitive learning is an algorithm modification that addresses class imbalance. The main concept is based on strongly penalizing mistakes for some classes defined in what is called a cost matrix. For example, if we specify that the minority class has a higher cost than the majority class, then the used model will tend to do fewer mistakes for the minority during the learning process.

7.16. Framework. I have created a systematic framework that addresses all the points mentioned in the methodology.

Outline of the framework below:

1. Data processing
   (a) Reading data from CSV/Excel files
   (b) Performing One Hot Encoding for specific features

2. Pipeline preparation (Steps performed during the learning process)
    (a) Scaling (We choose 1 of the list)
        i. Standardization for normally distributed / Normalization for other features.
        ii. Normalization only
        iii. Standardization only
        iv. Normalization followed by Power Transform
    (b) Smote (optional) - (We choose one from the list)
        i. SMOTE-NC
        ii. SMOTE with Tomek
    (c) Modeling [choosing machine learning tool with a set of hyperparameters] – (We choose 1 of the list)
        i. Logistic Regression
        ii. SVM
        iii. Decision Tree
        iv. XGBOOST
        v. Linear Discriminant Analysis (LDA)
        vi. Quadratic Discriminant Analysis (QDA)
        vii. Gaussian Naive Bayes (GNB)
        viii. Multinomial Naive Bayes (MNB)
        ix. Gaussian Process (GPC)
        x. Choosing a set of parameters [passed along with the model]
3. Repeated Nested Cross Validation – with hyperparameter tuning
    (a) Train / Test stratified split
    (b) Training set also split into k folds during repeated cross validation process
    (c) Random or Grid Search is used to indicate the best hyper parameters
4. Saving results to output files
    (a) Training and Testing Results
        i. For each model with the chosen configurations
        ii. For each scorer and each metric
        iii. For each testing split in the cross-validation process
        iv. Adding the best parameters of each round of the testing process
    (b) Plots

   i. Learning curve, scalability and performance (training time)
   ii. ROC Curve
   iii. Precision Recall curve

5. Reporting the summary of the test splits using the mean values over all the metrics from all the splits

Chapter 8

Clustering

We performed K-means clustering using Gower's Distance without scaling. Gower's distance does not require any scaling because it evaluates each feature independently by creating what is called a dissimilarity matrix. [26] And then the comparison process is done over this matrix. We report some observations below:

8.1. Observations.

Observation for All Defects Sheet

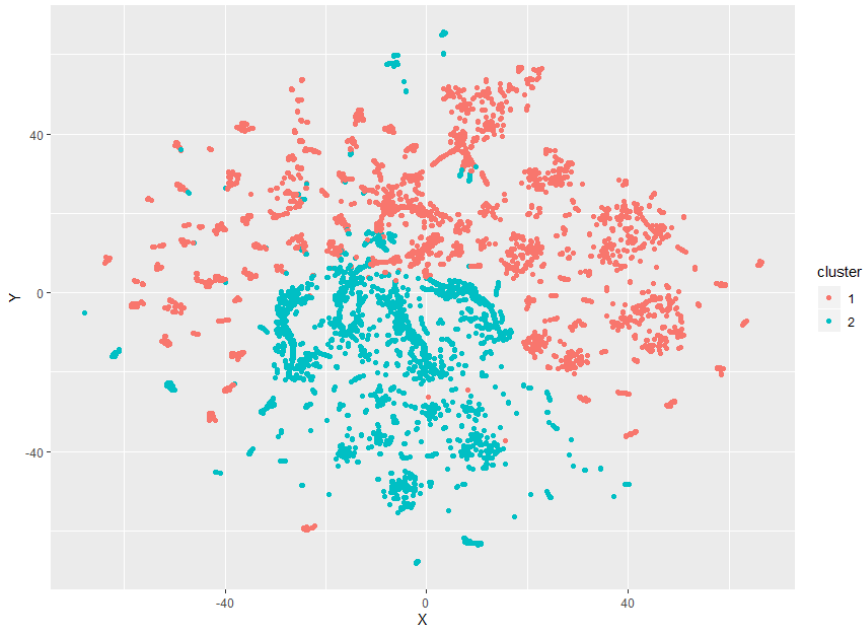


Figure 8.1: Cluster Plot for All Defects

Column Name	Cluster 1 – Mean Values	Cluster 2 – Mean Values
Mom Education (level 0 to 8)	5.006	4.184
Parental Consanguinity	25.90%	44.50%
Folic Acid Before	43.40%	12.30%
Medical History	24%	57%
Total Abortions	0.43	0.77
Number of Cigarettes / week	30.09	27.9
SO2 for all weeks	Range: 10-2 to 10-1	Range: 7 to 13
Percentage of Females	60.94%	39.06%
Percentage of Males	54.66%	45.34%

Table 8.1: Highlighted Columns for All Defects Sheet

Interpretation for All Defects Sheet

Women in the first cluster have a higher educational level than women in the second cluster (on average). The percentage of consanguinity degree in the first cluster is less (almost 25% to 44%), a higher percentage used to take folic acid before pregnancy (43% to 12.3%). 24% of cluster 1 have medical history compared to a 57% in the second cluster. The abortion rate is higher in the second cluster, but the rate is not that indicative (almost from 0.4 to 0.7 abortions on average). On average, women in cluster 1 are exposed to 2 cigarettes less than women in cluster 2 (weekly). The SO2 levels are much less in cluster 1. Overall, more females are exposed to birth defects than males.
So, we think that the first cluster is composed of controls and the second cluster is composed of cases.

8.1.1. Observations for CNS Sheet. First we display the plot for CNS.

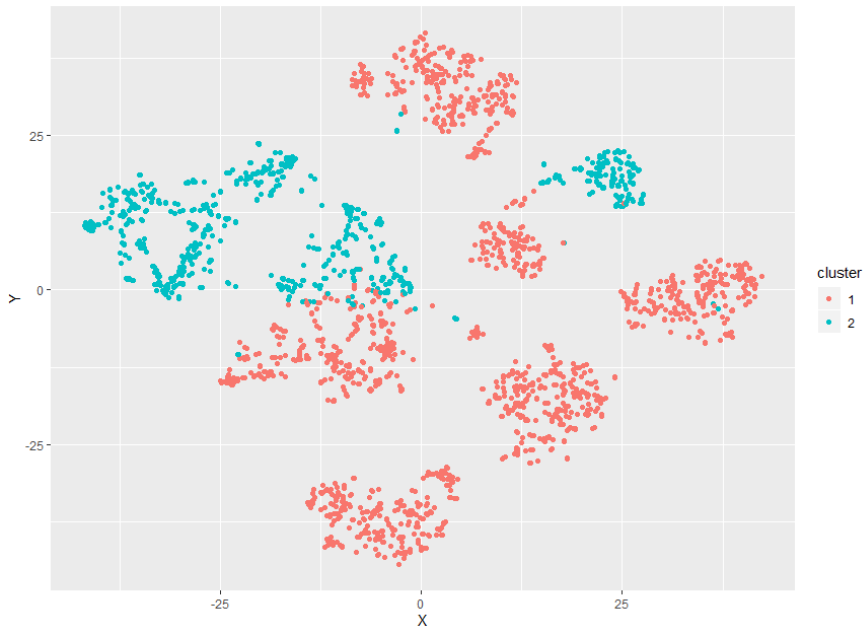


Figure 8.2: Cluster Plot for CNS

Column Name	Cluster 1 – Mean Values	Cluster 2 – Mean Values
Mom Education (level 0 to 8)	4.907	3.472
Parental Consanguinity	11.50%	77.05%
Tobacco Exposure (level 1 to 3)	0.147	0.476
Folic Acid Before	38.70%	5.20%
Medical History	21.60%	82.10%
Total Abortions	0.4727	0.8223
Number of Cigarettes / week	16.67	13.07
Air Pollution Data	Similar	Similar
Percentage of Females	57.90%	42.10%

Table 8.2: Highlighted Columns for CNS Sheet

Interpretation for CNS Sheet

Women in the first cluster have a higher educational level than women in the second cluster (on average). The percentage of consanguinity degree in the first cluster is less (almost 11.5% to 77.0%), a higher percentage used to take folic acid before pregnancy (38.7% to 5.2%). 21.6% of cluster 1 have medical history

compared to 82.1% in the second cluster. The abortion rate is higher in the second cluster, but the rate is not that indicative (almost from 0.47 to 0.8 abortions on average). On average, women in cluster 1 are exposed to 3.67 cigarettes more than women in cluster 2 (on a weekly basis).
So, we think that the first cluster is composed of controls and the second cluster is composed of cases.

Figure 8.3: Cluster Plot for GU

Column Name	Cluster 1 – Mean Values	Cluster 2 – Mean Values
Mom Education (level 0 to 8)	5.045	4.18
Parental Consanguinity	10.10%	70.70%
Tobacco Exposure (level 1 to 3)	0.1484	0.3114
Folic Acid Before	53.40%	1.50%
Medical History	5%	80.30%
Total Abortions	0.43	0.27
Number of Cigarettes / week	20	37
WoR SO2	2.19	11.57
Percentage of Females	76.40%	23.60%

Table 8.3: Highlighted Columns for GU Sheet

8.1.2. Observation for GU Sheet.

Interpretation for GU Sheet

Women in the first cluster have a higher educational level than women in the second cluster (on average). The percentage of consanguinity degree in the first cluster is less (almost 10.1% to 70.7%), a higher percentage used to take folic acid before pregnancy (53.4% to 1.5%). 5% of cluster 1 have medical history compared to 80.3% in the second cluster. The abortion rate is higher in the first cluster, but the rate is not that indicative (almost from 0.43 to 0.27 abortions on average). On average, women in cluster 1 are exposed to 13 cigarettes less than women in cluster 2 (on a weekly basis). WoR SO2 levels are higher for the second cluster.
So, we think that the first cluster is composed of controls and the second cluster is composed of cases.

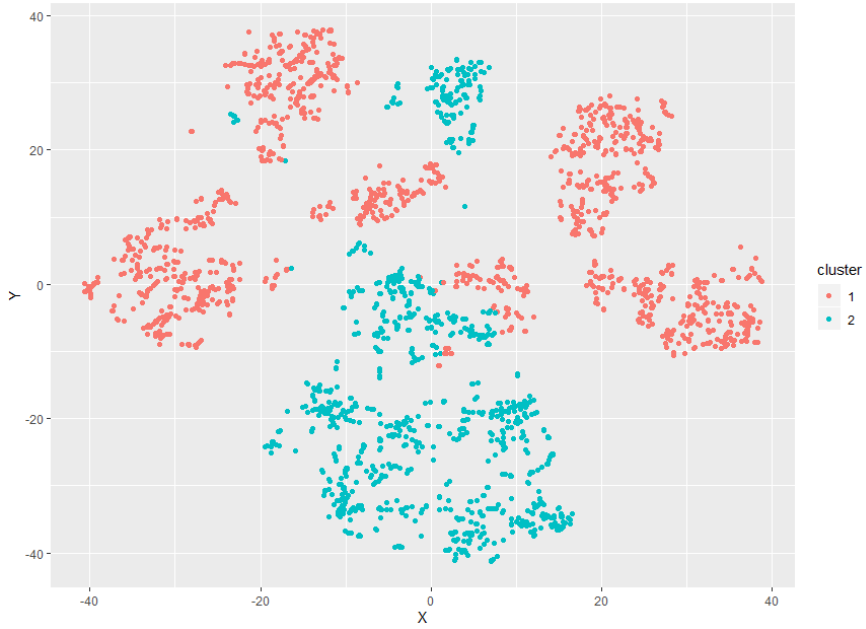


Figure 8.4: Cluster Plot for MUSC

Column Name	Cluster 1 – Mean Values	Cluster 2 – Mean Values
Mom Education (level 0 to 8)	4.981	4.021
Parental Consanguinity	10.20%	75.30%
Tobacco Exposure (level 1 to 3)	0.16	0.38
Folic Acid Before	43.30%	5.80%
Medical History	9%	86.00%
Total Abortions	0.45	0.54
Number of Cigarettes / week	23.96	19.94
Percentage of Females	39.80%	60.20%
Percentage of Males	79.40%	20.60%

Table 8.4: Highlighted Columns for MUSC Sheet

8.1.3. Observations for MUSC Sheet.

8.1.4. Interpretation for MUSC Sheet. Women in the first cluster have a higher educational level than women in the second cluster (on average). The percentage of consanguinity degree in the first cluster is less (almost 10.2% to 75.3%), a

higher percentage used to take folic acid before pregnancy (43.3% to 5.8%). 8.9% of cluster 1 have medical history compared to 86% in the second cluster. The abortion rate is higher in the first cluster, but the rate is not that indicative (almost from 0.45 to 0.54 abortions on average). On average, women in cluster 1 are exposed to 4 cigarettes more than women in cluster 2 (on a weekly basis). So, we think that the first cluster is composed of controls and the second cluster is composed of cases.

Chapter 9

Data Visualization

We used several tools to visualize the data.

9.1. TSNE for Clustered Data. TSNE is a tool to visualize high-dimensional data. It converts similarities between data points to joint probabilities and tries to minimize the Kullback-Leibler divergence between the joint probabilities of the low-dimensional embedding and the high-dimensional data. t-SNE has a cost function that is not convex, i.e. with different initializations we can get different results. **??**

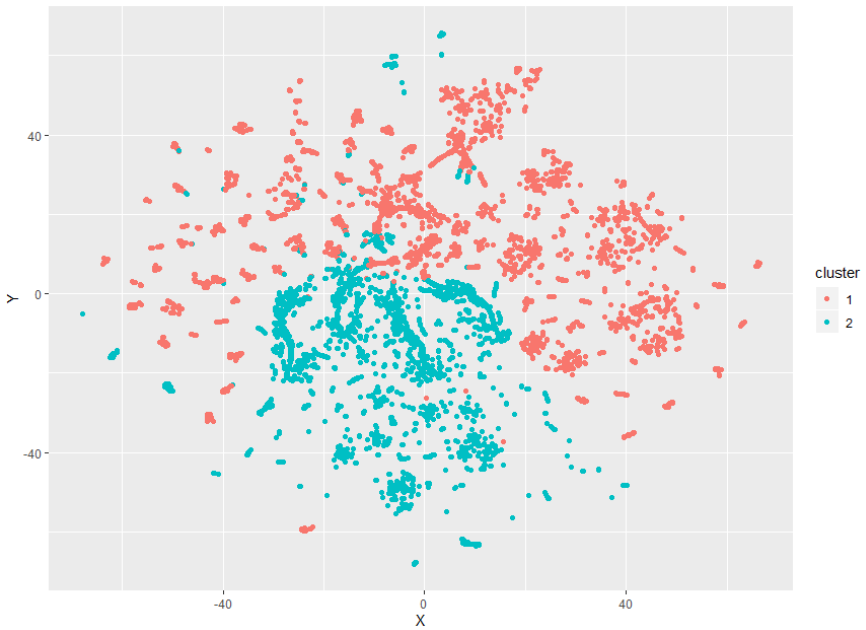
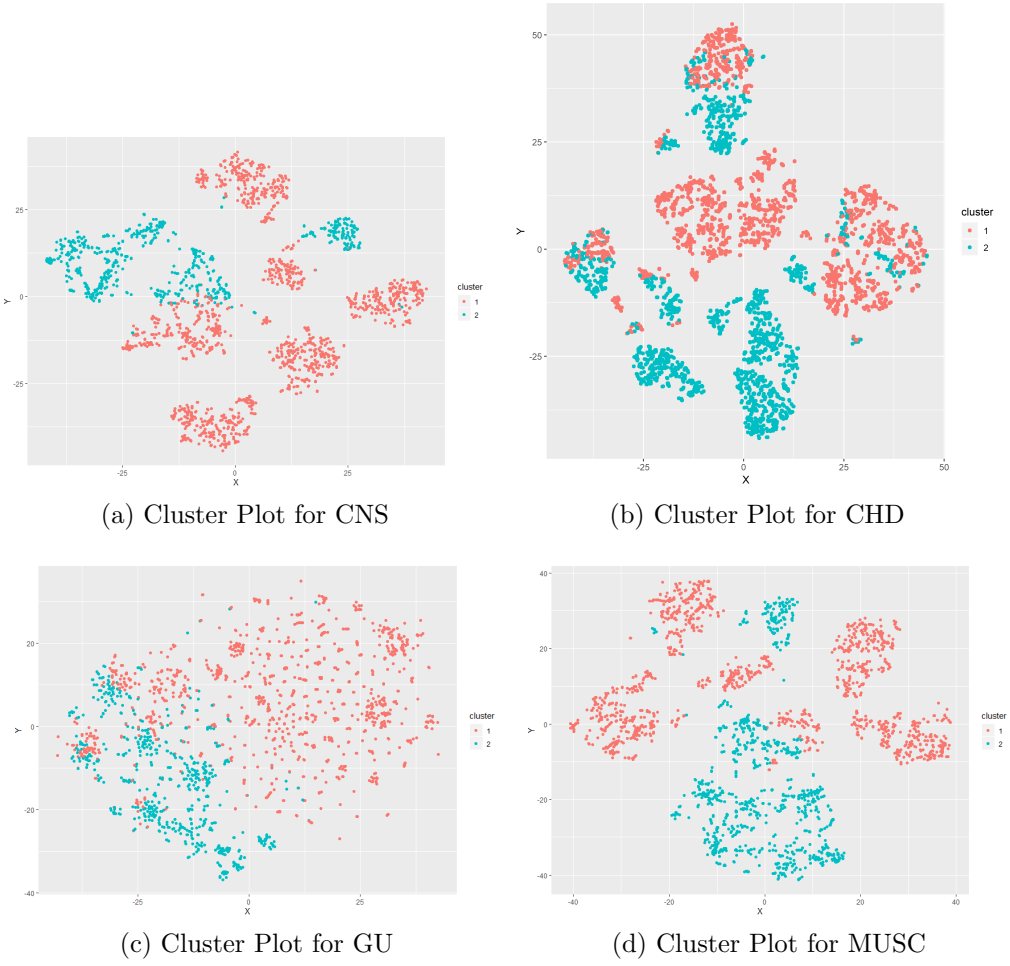


Figure 9.1: Cluster Plot for All Defects

Figure 9.1 shows that the data can be split into two or more clusters. This is

also the case for figures 9.2 for per defect clusters.



(a) Cluster Plot for CNS

(b) Cluster Plot for CHD

(c) Cluster Plot for GU

(d) Cluster Plot for MUSC

Figure 9.2: Clustering Plots

Figures in 9.2 show that the data can be split into multiple clusters. (a) shows that the data can be split into up to 9 different groups. (b) shows that the data can be split into up to 6 groups of more. (c) shows that the data can be split into 2 groups. (d) shows that the data can be split into 7 or more groups.
Plots (a), (b) and (d) suggest that the data can be split into more than two clusters, which may negatively affect the results of the machine learning process and binary classification, because it would be difficult for the models to split the data into two classes during the prediction process. However, figure (c) shows good separability between 2 clusters, this can be seen from the colors of each cluster, and this should enhance the performance of the machine learning models. For instance, we displayed F2 score results for cost sensitive logistic regression on all four datasets in table 9.1. It is clear that this model is performing better on GU sheet by 14% to 19%.

	CHD	CNS	GU	MUSC
F2 Score	0.409(0.0965)	0.4594(0.0376)	**0.592(0.113)**	0.4447(0.1182)

Table 9.1: Cost Sensitive Logistic Regression - F2 Score Comparison

9.2. Google Facets Data Visualization. We used Google Facets to display the input data and to get some insight into the distribution of the data within the input columns. Facets is an open-source visualization tool released by Google under the PAIR(People + AI Research) initiative. This tool helps us to understand and analyze the Machine Learning datasets. Facets consist of two visualizations, both of which help to drill down the data and provide great insights without much work at the user's end. **??**

In figures 9.3 and 9.3 We can see that maternal age and BMI are normally distributed, thus the sample can be seen a normal and not biased in terms of maternal age and BMI.

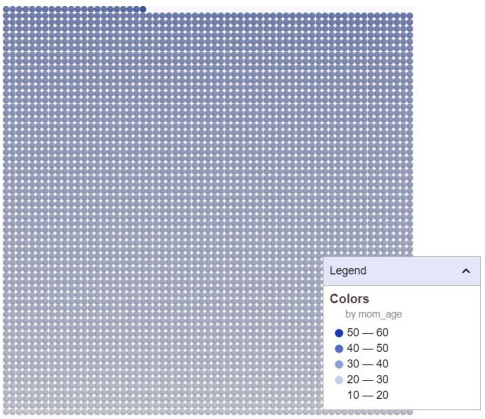


Figure 9.3: Mother Age

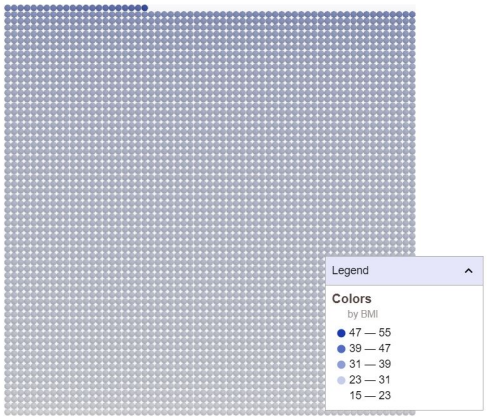


Figure 9.4: BMI

Then we display mother education in figure 9.5, we can see that categories 4 and 5 are the prominent categories, also we can notice that category 0 is very minimal. Category 0 is the category of illiterate women.

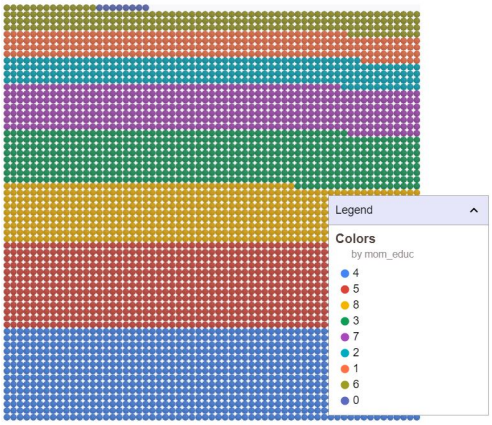


Figure 9.5: Mother Education

In figure 9.6 we represent the distribution of folic acid intake before pregnancy, we can see that almost half of the population consumes folic acid before pregnancy.

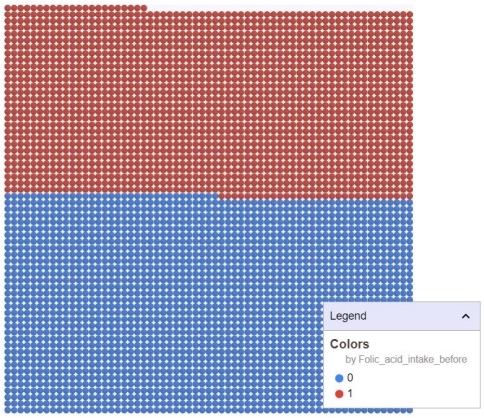


Figure 9.6: Folic Acid Intake Before Pregnancy

Now, we display medical history in figure 9.7. We can see that most of the population does not have a medical history.

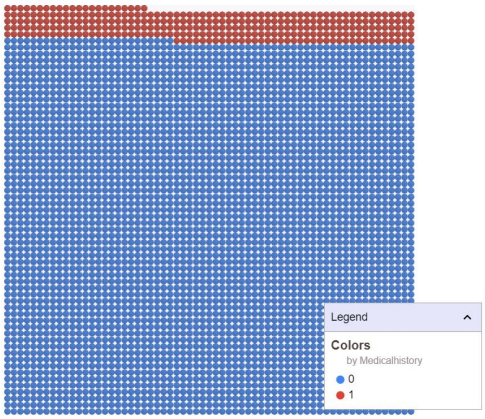


Figure 9.7: Medical History of Pregnant Woman

Then, we display the presence of birth defect in figure 9.8. Here we can see that the presence of birth defect represent the minority class.

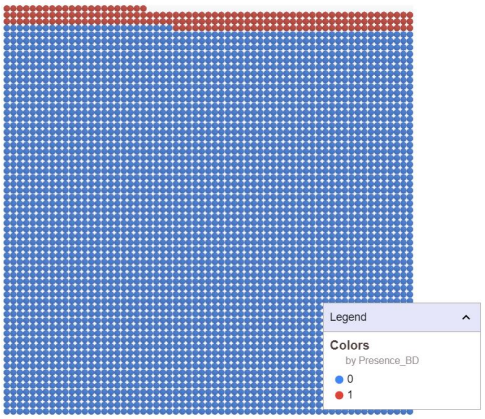


Figure 9.8: Presence of Birth Defect

Next, we display parity feature in 9.9. We can see that parity is well distributed over the controls and cases.

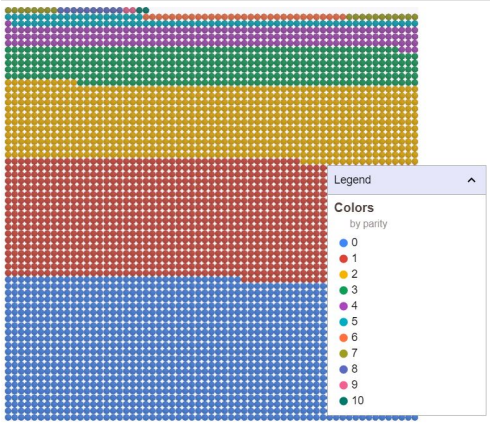


Figure 9.9: Parity

Next, we display the consanguinity degree in 9.11. We can see that most of the population have a consanguinity degree of 4 or no consanguinity.

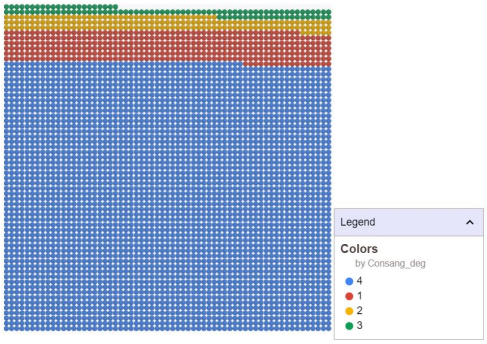


Figure 9.10: Consanguinity Degree

Next, we display the consanguinity degree in 9.11. We can see that most of the population have a consanguinity degree of 4 or no consanguinity.

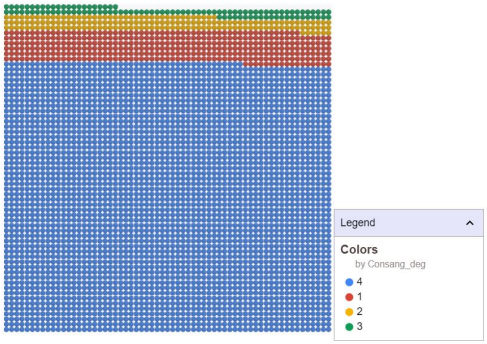


Figure 9.11: Consanguinity Degree

Chapter 10

Feature Selection

10.1. Results. Below we present the results per birth defect and for all defects.

	CNS			
	Feature Importance	PCA	RFE	univariate selection
mom education	100	82	84	99
Consanguinity degree	100	13	100	100
sex	48	98	95	92
parental consanguinity	35	63	78	100
Tobacco exposure	89	66	96	46
Alcohol during pregnancy	0	0	34	0
Folic acid intake before pregnancy	100	2	100	100
Medical History	100	100	100	100
mother age	83	6	74	14
parity	22	19	70	83
total abortions	7	21	84	53
spontaneous abortions	11	70	75	6
induced abortions	0	3	92	0
number living children	28	22	83	85
BMI	89	100	64	5
pre lag NO2	35	2	71	2
pre lag SO2	8	70	53	34
pre lag PM2.5	26	35	47	1
Window of Risk NO2	34	30	76	4
Window of Risk SO2	23	4	54	14
Window of Risk PM2.5	62	0	96	60
Number of cigarettes per week	0	24	52	2

Table 10.1: Feature Selection Scores for CNS

	CHD			
	Feature Importance	PCA	RFE	Univariate Selection
mother education	100	81	58	44
Consanguinity degree	100	22	100	100
sex	47	98	74	43
parental consanguinity	62	46	97	100
Tobacco exposure	87	3	99	70
Alcohol during pregnancy	0	0	36	0
Folic acid intake before pregnancy	68	0	100	100
Medical History	100	100	100	100
mother age	38	5	68	2
parity	48	42	100	83
total abortions	28	67	100	12
spontaneous abortions	24	20	100	5
induced abortions	0	0	31	0
living children	18	18	94	57
BMI	48	80	48	46
pre lag NO2	26	52	34	13
pre lag SO2	29	11	58	98
pre lag PM2.5	62	12	79	0
Window of Risk NO2	9	88	64	9
Window of Risk SO2	34	3	72	100
Window of Risk PM2.5	65	0	70	0
Number of cigarettes per week	7	99	95	18

Table 10.2: Feature Selection Scores for CHD

	GU			
	Feature Importance	PCA	RFE	univariate selection
mother education	100	46	26	49
Consanguinity degree	100	53	100	100
sex	100	3	100	100
parental consanguinity	33	28	73	100
Tobacco exposure	95	20	94	38
Alcohol during pregnancy	0	0	0	0
Folic acid intake before pregnancy	89	0	97	100
Medical History	100	100	99	100
mother age	87	32	54	53
parity	13	69	35	0
total abortions	12	96	23	8
spontaneous abortions	14	0	27	32
induced abortions	0	0	19	0
living children	28	31	36	0
BMI	71	99	17	1
pre lag NO2	41	1	52	86
pre lag SO2	26	3	19	100
pre lag PM2.5	3	68	52	0
Window of Risk NO2	47	0	40	25
Window of Risk SO2	33	0	44	100
Window of Risk PM2.5	8	97	30	8
Number of cigarettes per week	0	73	9	0

Table 10.3: Feature Selection Scores for GU

	MUSC			
	Feature Importance	PCA	RFE	univariate selection
mother education	100	22	96	98
Consanguinity degree	100	44	100	100
sex	62	96	54	46
parental consanguinity	24	12	98	100
Tobacco exposure	94	6	97	92
Alcohol during pregnancy	0	0	20	0
Folic acid intake before pregnancy	97	0	100	100
Medical History	100	100	100	100
mother age	64	20	44	6
parity	67	92	73	3
total abortions	34	11	99	17
spontaneous abortions	54	19	100	13
induced abortions	0	0	40	0
living children	37	53	77	3
BMI	76	91	68	96
pre lag NO2	26	43	83	35
pre lag SO2	5	29	55	71
pre lag PM2.5	2	60	26	0
Window of Risk NO2	47	26	96	93
Window of Risk SO2	4	0	59	25
Window of Risk PM2.5	6	3	63	2
Number of cigarettes per week	1	92	73	0

Table 10.4: Feature Selection Scores for MUSC

	Feature Importance	PCA	RFE	univariate selection
mother education	100	0	36	23
consanguinity degree	100	0	100	100
sex	55	0	8	3
parental consanguinity	100	0	97	100
Tobacco exposure	36	0	31	74
Alcohol during pregnancy	0	0	6	0
Folic acid intake before pregnancy	100	0	100	100
Medical History	100	100	100	100
mother age	94	0	50	0
parity	15	91	29	0
total abortions	4	0	24	0
spontaneous abortions	5	0	24	0
induced abortions	63	0	42	0
living children	24	11	13	0
BMI	93	0	14	7
Number of cigarettes per week	0	0	15	0
NO2_w1	4	0	21	2
SO2_w1	0	3	11	3
PM25_w1	0	0	13	0
NO2_w2	2	0	7	0
SO2_w2	0	20	8	0
PM25_w2	0	0	22	0
NO2_w3	0	0	14	0
SO2_w3	0	0	7	0
PM25_w3	1	1	5	0
NO2_w4	0	1	7	0
SO2_w4	0	1	13	0
PM25_w4	0	0	6	0
NO2_w5	1	0	7	0
SO2_w5	0	0	7	0
PM25_w5	53 1	0	2	0
NO2_w6	4	0	4	0
SO2_w6	0	0	7	0
PM25_w6	0	0	15	0
NO2_w7	12	0	17	92

	Feature Importance	PCA	RFE	univariate selection
PM25_w7	9	0	6	0
NO2_w8	17	0	14	58
SO2_w8	0	0	12	0
PM25_w8	17	0	31	15
NO2_w9	3	0	7	1
SO2_w9	0	0	3	0
PM25_w9	2	0	5	5
NO2_w10	2	0	8	60
SO2_w10	0	0	23	0
PM25_w10	0	0	14	0
NO2_w11	0	0	3	49
SO2_w11	0	0	9	0
PM25_w11	0	0	2	0
NO2_w12	0	0	16	71
SO2_w12	0	0	5	0
PM25_w12	0	1	3	0
NO2_w13	0	2	16	19
SO2_w13	0	0	4	0
PM25_w13	1	0	9	0
NO2_w14	0	0	14	0
SO2_w14	0	4	4	1
PM25_w14	0	0	3	0
NO2_w15	0	23	4	5
SO2_w15	0	0	16	0
PM25_w15	0	0	8	0
NO2_w16	0	1	12	0
SO2_w16	0	0	10	0
PM25_w16	0	0	17	0
NO2_w17	1	0	21	0
SO2_w17	0	0	9	0
PM25_w17	0	54	54	3	0
NO2_w18	0	0	10	0
SO2_w18	0	0	11	1
PM25_w18	0	0	3	0
NO2_w19	2	0	8	2

	Feature Importance	PCA	RFE	univariate selection
PM25_w19	1	1	2	0
NO2_w20	0	0	4	0
SO2_w20	2	0	8	0
PM25_w20	0	0	6	0
NO2_w21	0	0	6	38
SO2_w21	0	0	16	0
PM25_w21	1	0	4	0
NO2_w22	1	0	5	0
SO2_w22	0	0	5	0
PM25_w22	1	8	2	0
NO2_w23	1	0	30	0
SO2_w23	0	0	4	0
PM25_w23	4	3	8	0
NO2_w24	0	0	3	0
SO2_w24	0	0	8	0
PM25_w24	2	0	9	0
NO2_w25	0	0	2	0
SO2_w25	0	0	25	0
PM25_w25	0	0	4	0
NO2_w26	0	32	3	0
SO2_w26	0	0	2	0
PM25_w26	0	11	3	0
NO2_w27	0	1	10	0
SO2_w27	0	0	2	0
PM25_w27	0	9	10	0
NO2_w28	0	0	5	0
SO2_w28	0	0	5	0
PM25_w28	0	0	13	0
NO2_w29	2	2	10	0
SO2_w29	0	0	10	0
PM25_w29	0	0	1	0
NO2_w30	0	5	5	0
SO2_w30	0	0	8	0
PM25_w30	0	0	5	0
NO2_w31	1	0	4	0

	Feature Importance	PCA	RFE	univariate selection
PM25_w31	0	0	9	0
NO2_w32	0	0	7	0
SO2_w32	0	0	19	0
PM25_w32	1	0	7	0
NO2_w33	1	3	5	0
SO2_w33	0	0	4	0
PM25_w33	1	0	3	0
NO2_w34	0	1	8	0
SO2_w34	0	0	2	0
PM25_w34	0	0	8	0
NO2_w35	0	0	5	3
SO2_w35	0	0	4	0
PM25_w35	2	0	4	0
NO2_w36	0	57	5	0
SO2_w36	0	0	4	0
PM25_w36	0	0	2	0
NO2_w37	0	1	10	0
SO2_w37	0	0	6	0
PM25_w37	4	61	2	0
NO2_w38	1	12	10	0
SO2_w38	0	0	4	2
PM25_w38	3	31	2	0
NO2_w39	0	18	7	0
SO2_w39	0	7	2	8
PM25_w39	0	0	2	0
NO2_w40	0	4	16	0
SO2_w40	0	16	4	38
PM25_w40	0	0	15	0
NO2_w41	1	0	13	0
SO2_w41	0	0	3	0
PM25_w41	0	56	14	10	0
NO2_w42	1	5	23	0
SO2_w42	0	0	9	0
PM25_w42	0	91	4	9
NO2_w43	0	12	4	0

	Feature Importance	PCA	RFE	univariate selection
PM25_w43	0	0	23	0
NO2_w44	1	0	4	0
SO2_w44	0	0	14	10
PM25_w44	0	0	6	0

Table 10.9: Feature Selection Scores for All Defects - Part 5

10.2. How to Evaluate Feature Selection Results. To evaluate feature selection results, this was done by picking all scores above 60/100. Also, we looked at features picked by more than one technique.

10.3. Selected Features. ALL Defects Sheet:

- Mother Education
- Consanguinity Degree
- Parental Consanguinity
- Folic Acid Intake Before Pregnancy
- Medical History
- Mother Age
- Induced Abortions
- BMI
- Tobacco exposure
- Parity
- NO2 week 7
- PM2.5 week 37
- PM2.5 week 42

CHD:

- mother education

- Consanguinity degree
- sex
- parental consanguinity
- Tobacco exposure
- Medical History
- mother age
- parity
- total abortions
- spontaneous abortions
- living children
- BMI
- pre lag SO2
- pre lag PM2.5
- Window of Risk NO2
- Window of Risk SO2
- Window of Risk PM2.5
- Number of cigarettes per week

CNS:

- Mother Education
- Consanguinity Degree
- Sex
- Parental Consanguinity
- Tobacco Exposure
- Folic Acid Intake Before Pregnancy
- Medical History

- Mother Age
- Parity
- Total Abortions
- Spontaneous Abortions
- Induced Abortions
- Living Children
- BMI
- Pre lag NO2
- Pre lag SO2
- Window of Risk NO2
- Window of Risk PM2.5

GU:

- Mother Education
- Consanguinity Degree
- Sex
- Parental Consanguinity
- Tobacco Exposure
- Folic Acid Intake Before Pregnancy
- Medical History
- Mother Age
- Parity
- Total Abortions
- BMI
- Pre lag NO2
- Pre lag SO2

- Pre lag PM2.5
- Window of Risk SO2
- Window of Risk PM2.5
- Number of cigarettes per week

MUSC:

- mother education
- Consanguinity degree
- sex
- parental consanguinity
- Tobacco exposure
- Folic Acid Intake Before Pregnancy
- Medical History
- Mother Age
- Parity
- Total Abortions
- Spontaneous Abortions
- Living Children
- BMI
- Pre lag NO2
- Pre lag SO2
- Pre lag PM2.5
- Window of Risk NO2
- Window of Risk PM2.5
- Number of cigarettes per week

Chapter 11

Classification

11.1. Classification Experiments. In the classification section, we used a process called repeated nested cross-validation described in section 6.13. to assess the performance of several machine learning models with different configurations.

In this process, we test our models using several scaling techniques and we optimize the results for several metrics.

For the scaling techniques, models are tested first without scaling, then for linear models like logistic regression, SVM, KNN and for Neural Network, we add the following scaling techniques: Normalization, Standardization, Normalization followed by Power Transform and a mixed approach where we perform standardization for normally distributed columns and normalization for the other numerical columns.

During the optimization process, the model chooses the best configuration which optimizes a specific metric. In our case, we optimized for three metrics; F2 score, Gmean, and Pr-Recall, then once done, we choose the configuration that performed the best.

The list of models that we used is the following:

- Adaboost
- AdaCost
- Catboost
- Decision Tree
- Cost Sensitive Decision Tree

- Decision Tree using Hellinger Distance
- Decision Tree with Oversampled training data using SMOTENC
- Decision Tree with mixed oversampling and undersampling using SMOTE-TOMEK
- KNN
- Logistic Regression
- Cost Sensitive Logistic Regression
- Logistic Regression with oversampled training data using SMOTENC
- Logistic Regression with mixed oversampling and undersampling using SMOTE-TOMEK
- SVM
- Cost Sensitive SVM
- SVM with oversampled training data using SMOTENC
- SVM with mixed oversampling and undersampling using SMOTETOMEK
- Random Forest
- Cost Sensitive Random Forest
- XGBOOST
- Cost Sensitive XGBOOST
- XGBOOST with oversampled training data using SMOTENC
- XGBOOST with mixed oversampling and undersampling using SMOTE-TOMEK
- Neural Networks

11.1.1. Used Metrics And Desirable Bounds. We are reporting the results of the following metrics:

- F2 Score - desirable bounds: should go above 50% up to 100%
- Gmean - desirable bounds: should go above 50% up to 100%

- AUC ROC - desirable bounds: should go above 50% up to 100%
- Accuracy - desirable bounds: should go above 50% up to 100%. It is normal to have an accuracy that is as high 95% or above, because if all control class (majority) was classified correctly, the accuracy will each 95%.

For each metric, we report validation results (training) and testing results. For validation results we report the average across all training splits and the standard deviation across them, more specifically; we report the average and standard deviation for the best performing combination of hyperparameters, for example for the first split we get the best performing combination of hyperparameters, the same concept for the second split, etc. Then testing is done using this same configuration. And finally, we report average and standard deviation over all testing results.

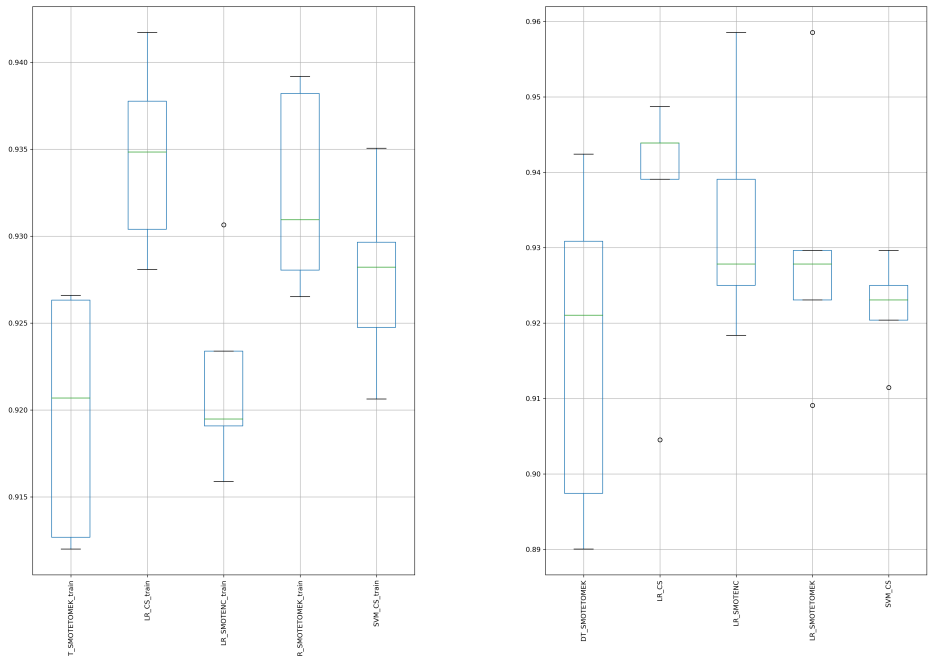
In the tables of results, we always start with the average of (validation or testing) and following by standard deviation of validation or testing (between parentheses).

11.2. Results. Detailed results tables are present in the appendix, please refer to tables A.421 A.422 A.423 A.424 A.425 A.426 A.427 A.428 A.429 A.430 A.431 A.432 A.433 A.434 A.

11.3. Observations. In the observation section, we display box plots for the produced results and we analyze them.

11.3.1. Observations For All Defects. Below we display the box plots for the best set of models. All other sets of box plots are displayed in the appendix.
To assess the performance of classification models, we used several metrics: F2 score, G-mean, AUC ROC, and Accuracy.
Below we will display the box plots for all the four metrics for both training and testing.
First, we will display comparative box plots for all the picked models.
First, we will display the box plots of the F2 score on both training and testing sets. We can see that the best performing model on the training set is the cost-sensitive logistic regression, followed by the logistic regression with SMOTE TOMEK, then cost-sensitive support vector machines, then logistic regression with SMOTE-NC, and finally decision tree with SMOTE TOMEK. Also, if we assess the plots of the produced on the testing set, we can see slightly different results, the only difference is that logistic regression with SMOTE-NC has a better performance for the testing set. Also, we assessed the spread of the results produced by all the models; the first observation is that the spread is smaller for the training sets compared to the testing sets. The second observation is that the

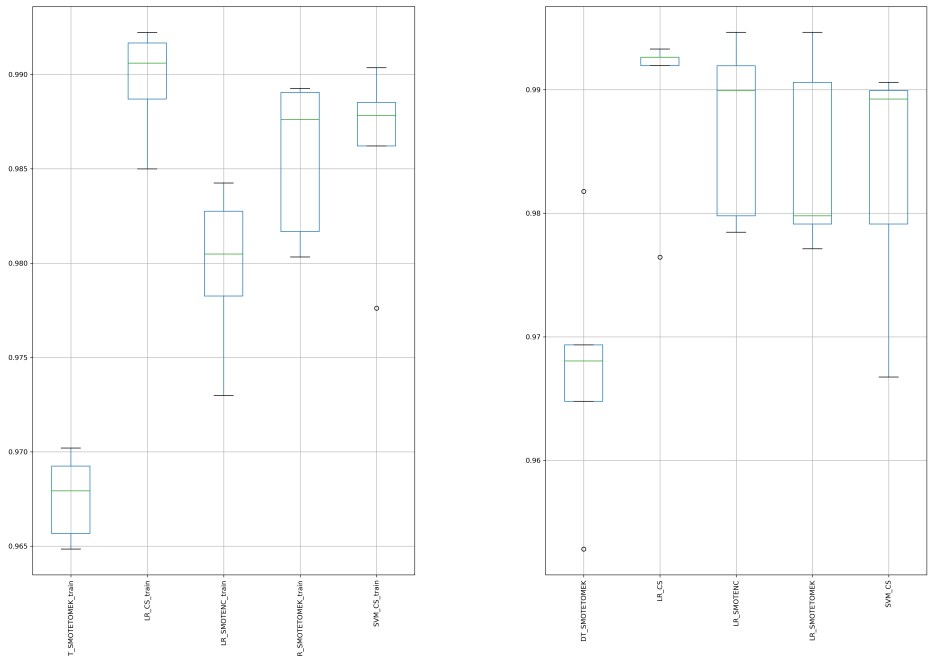
decision tree has the highest standard deviation for both training and testing. Other models are producing smaller deviations - 1% or less for training and 2% for testing for all models.



(a) Training (b) Testing

Figure 11.1: Box Plots for All Defects - F2 Score

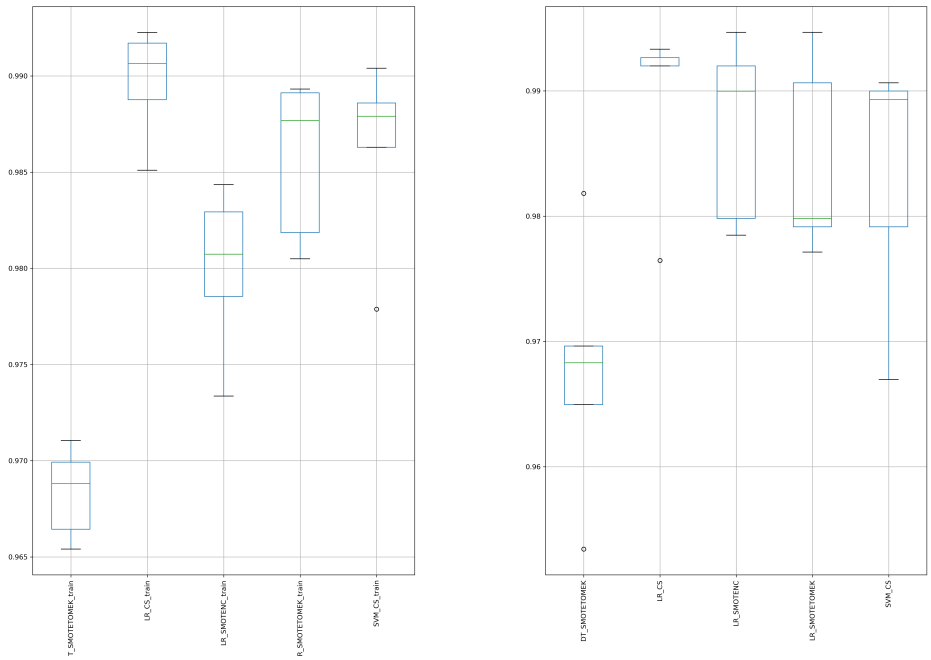
Second, we will evaluate the plots of Gmean, we can see that the results are very similar to the F2 score results, again, the best one is cost-sensitive logistic regression, cost-sensitive support vector machines, then logistic regression with SMOTE TOMEK, followed by logistic regression with SMOTE-NC and finally decision tree with SMOTE TOMEK. Again, we can see that logistic regression with SMOTE TOMEK is performing better on the testing set compared to the other models. Also, we assessed the spread of the results, we can see logistic regression with SMOTE TOMEK has the highest spread on the training set which is only 1% other models have spreads of less than 0.5% on the training set. If we look at the testing set, we can see that the spread is almost 1% for all the models except cost-sensitive logistic regression, which has a very minimal spread of almost 0.2%.



(a) Training (b) Testing

Figure 11.2: Box Plots for All Defects - Gmean Score

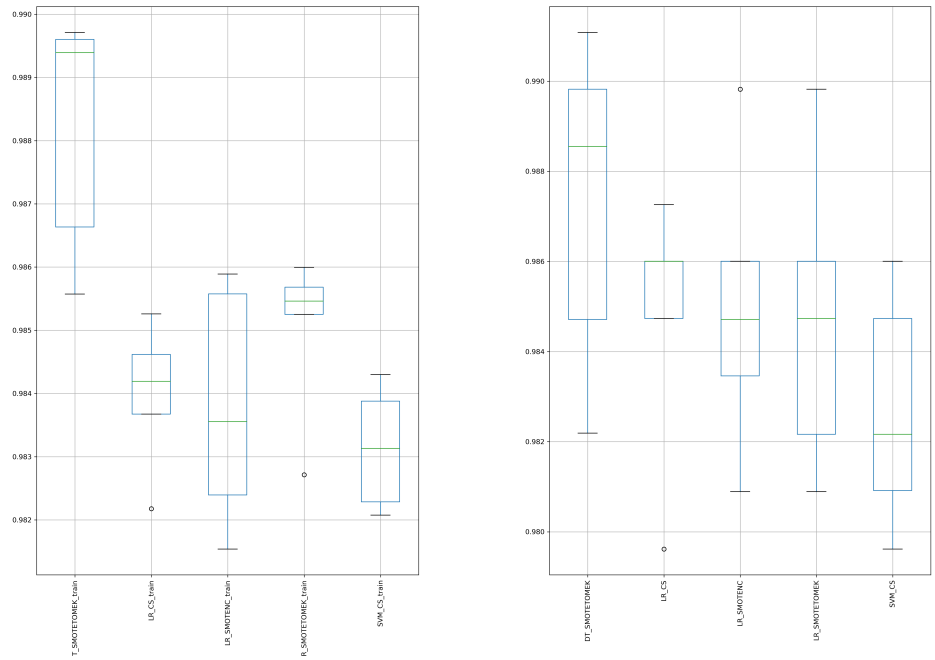
Below, we display the box plots of the AUC ROC results for both training and testing. We can see that they follow the exact shape of the Gmean scores, but we can see that the average is slightly higher, however, the spread is the same, it is almost 1% for all models on both training and testing sets.



(a) Training (b) Testing

Figure 11.3: Box Plots for All Defects - AUC ROC

Finally, we display the box plots of accuracy for all the models, we can see that the accuracy follows a different trend, specifically for the decision tree. This can be caused by the ability of the decision tree to predict the negative class at a more accurate rate compared to the positive class. That is why we can see that the F2 score for the decision tree is lower than the other models. Overall, all accuracy scores are high on average and the spread is very low; it is almost less than 0.3% for all models. This can be normal, because of the class imbalance, and that is why we are focusing more on other metrics such as F2 score, Gmean, and AUC ROC.



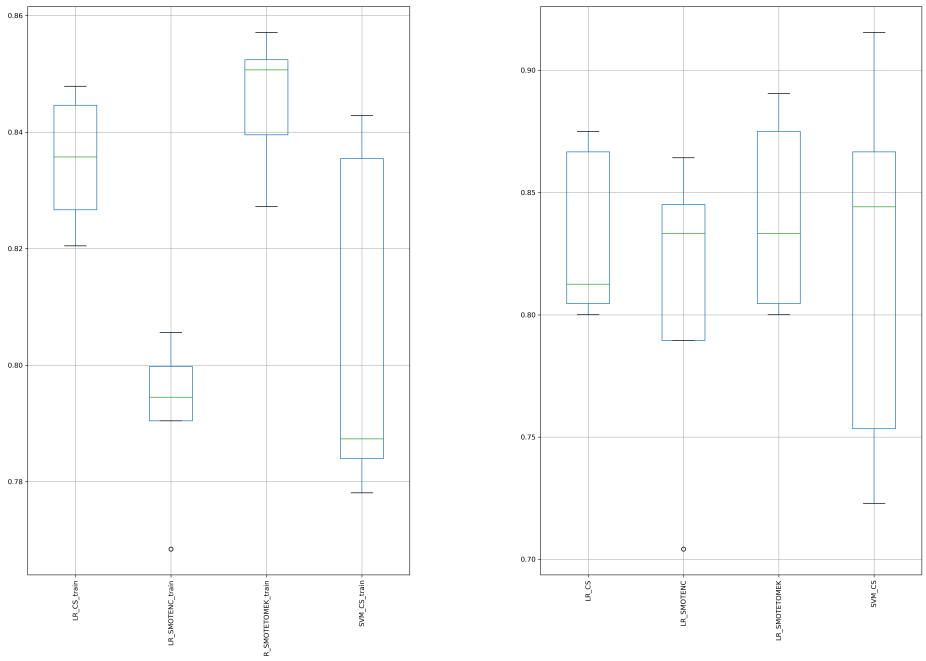
(a) Training (b) Testing

Figure 11.4: Box Plots for All Defects - Accuracy

11.3.2. Observations For Per Defect Results. For CHD, CNS, GU and MUSC datasets, all models performed poorly, and could not predict the defects accurately.
We proposed a solution for this issue by merging the datasets two by two and running the models on the merged datasets. So, we ended up with six new datasets:

- CHD - CNS
- CHD - GU
- CHD - MUSC
- CNS - GU
- CNS - MUSC
- GU - MUSC

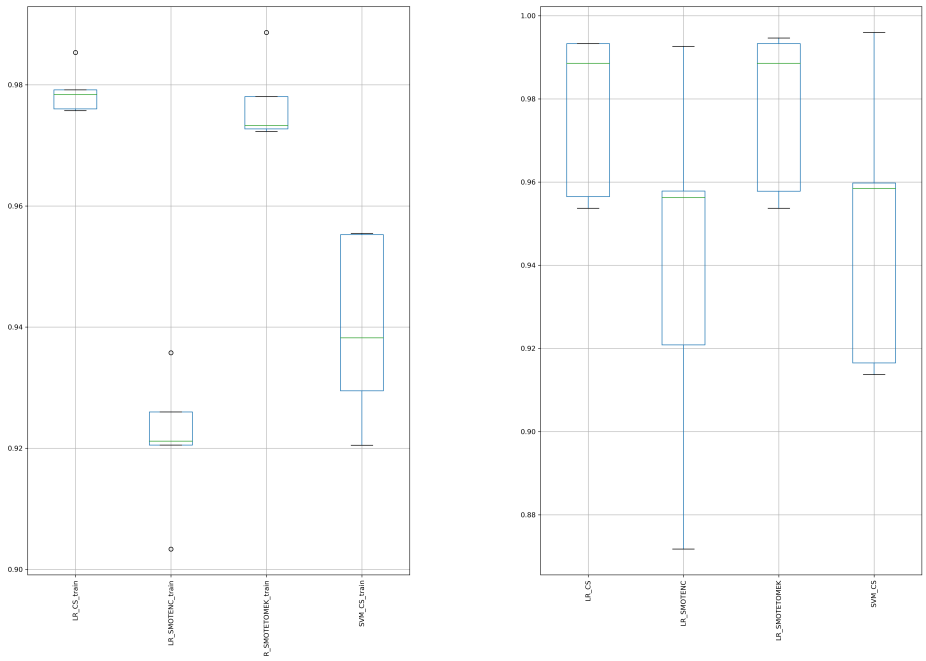
11.3.3. Observations For CHD - CNS. In the figures below, we assess the training and testing scores for the F2 score for predicting CHD and CNS. We can see that on average the best performing models are logistic regression with SMOTE TOMEK and cost-sensitive logistic regression. Also, we can see that they have an acceptable spread of around 2%. Cost-sensitive support vector machines have a lower F2 score on average and a very wide spread.



(a) Training (b) Testing

Figure 11.5: Box Plots for CHD CNS Defects - F2 Score

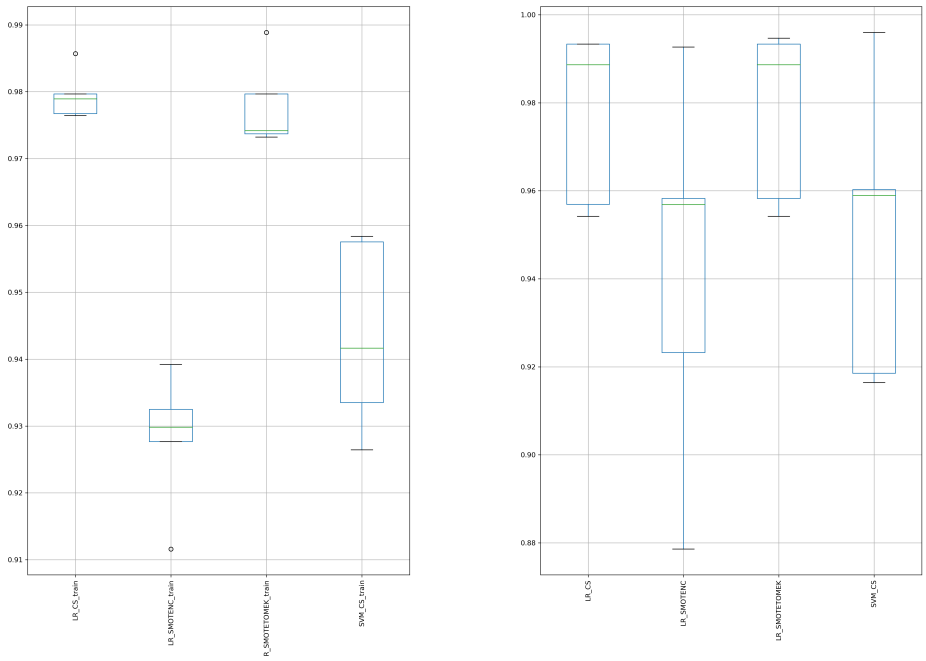
In the figures below, we observe the performance of the models using the Gmean metric. We can see that all the results are above 90% on average for both training and testing. The spread is less than 2% for all models on the training set except for cost-sensitive support vector machines, it has a spread of around 4%. The spread is higher for the testing set, it is around 4% for all the models.



(a) Training (b) Testing

Figure 11.6: Box Plots for CHD CNS Defects - Gmean Score

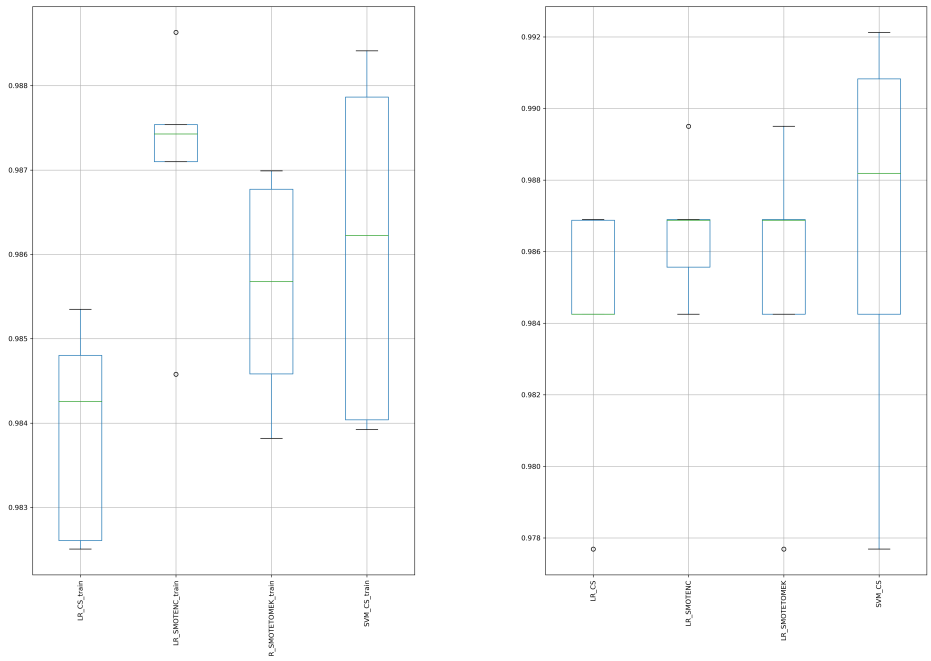
Results for AUC ROC follow the same trend as Gmean, the best performing models in terms of average result and spread are cost-sensitive logistic regression and logistic regression with SMOTE TOMEK. The rest of the models have similar spreads but the average performance is lower.



(a) Training (b) Testing

Figure 11.7: Box Plots for CHD CNS Defects - AUC ROC

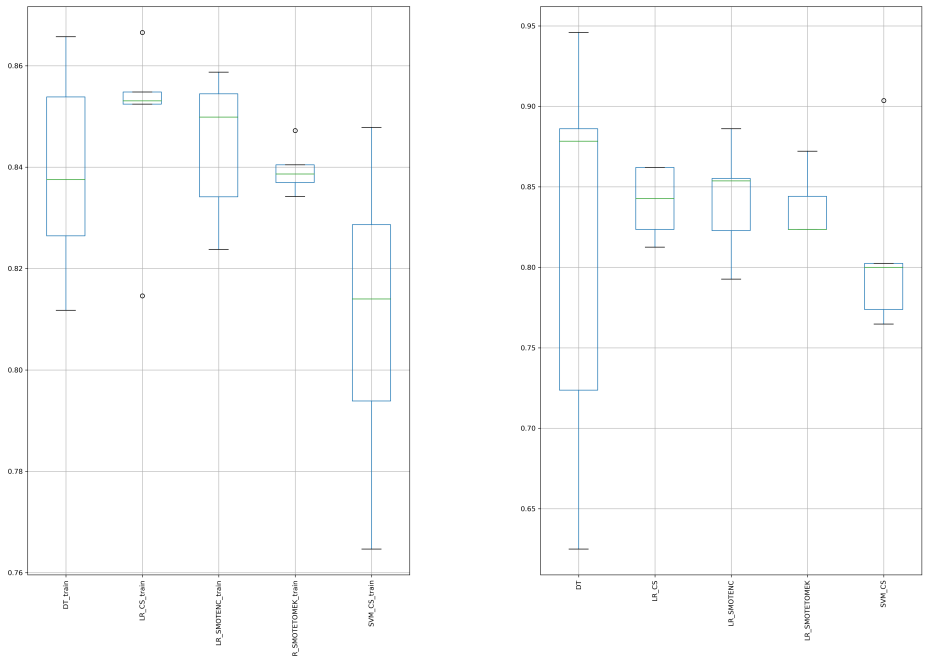
Now, for the accuracy metric, the lowest score is almost 98% for all the models. The spread is around 0.2% for all the models for both training and testing sets.



(a) Training (b) Testing

Figure 11.8: Box Plots for CHD CNS Defects - Accuracy

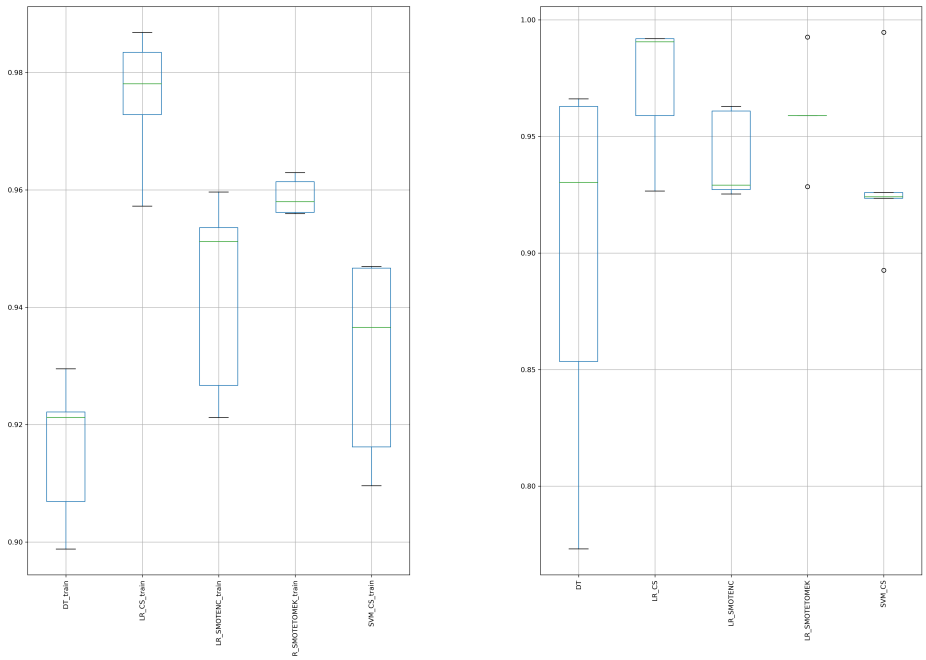
11.3.4. Observations For CHD - GU. In the figures below, we assess the training and testing scores for the F2 score for predicting CHD and GU. We can see that on average the best performing models are cost-sensitive logistic regression and logistic regression with SMOTE-NC. The spread is minimal (less than 1% for both cost-sensitive logistic regression and logistic regression with SMOTE TOMEK). However, the other models have a very widespread, decision tree that has the widest spread, which means that the results are not that consistent, same for cost-sensitive support vector machines.



(a) Training (b) Testing

Figure 11.9: Box Plots for CHD GU Defects - F2 Score

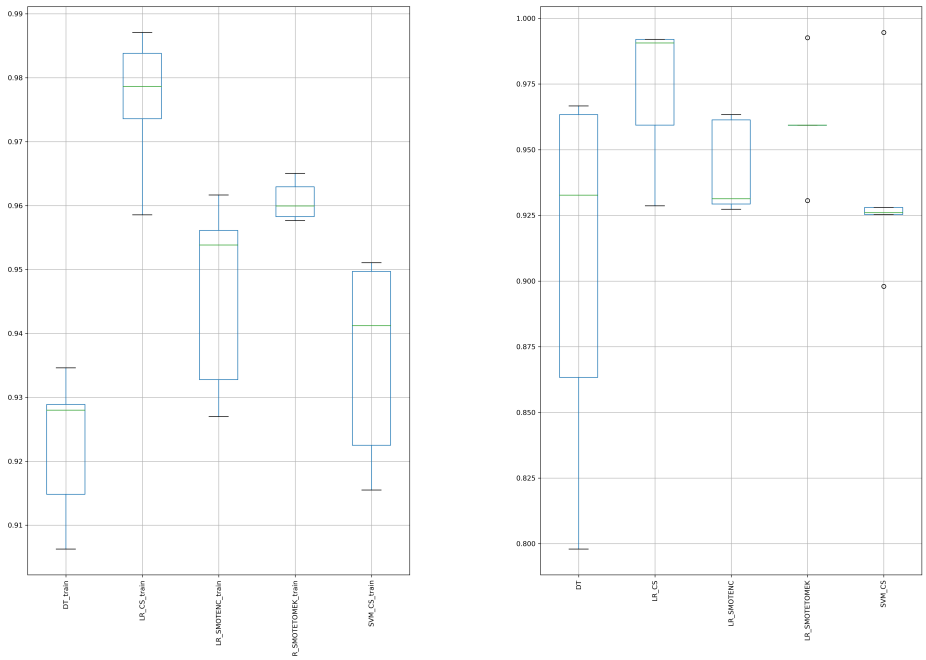
The best reported average Gmean is reported for cost-sensitive logistic regression for both training and testing. The spread is less than 4% for all the models. However, we can see that the decision tree has a very wide spread on the testing set.



(a) Training (b) Testing

Figure 11.10: Box Plots for CHD GU Defects - Gmean Score

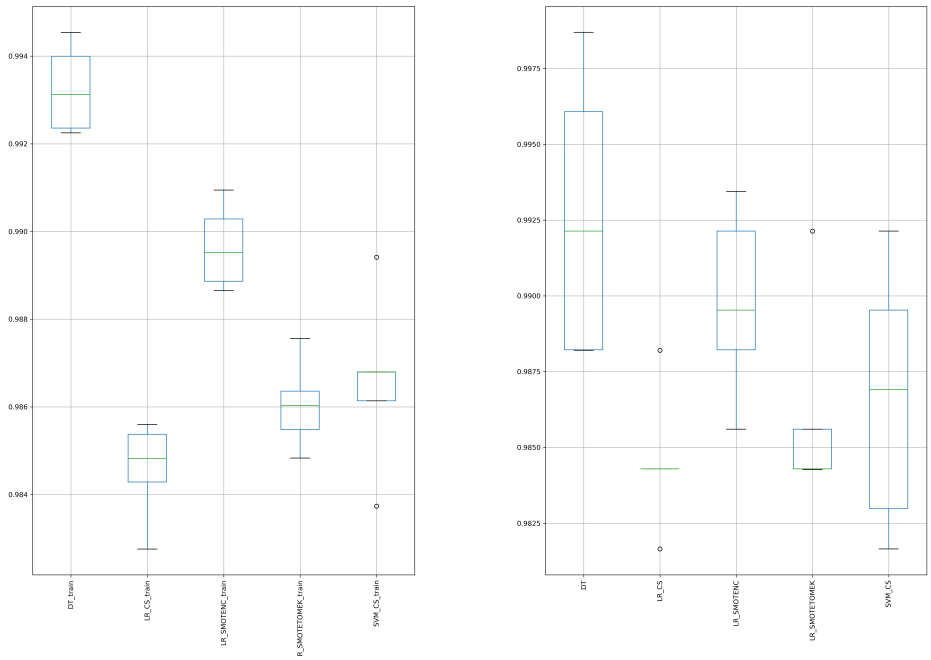
Again, cost-sensitive logistic regression has the highest AUC ROC score on average. The spread is less than 2% for both training and testing. Decision tree and cost-sensitive support vector machines have the worst performance on average and has the biggest spread.



(a) Training (b) Testing

Figure 11.11: Box Plots for CHD GU Defects - AUC ROC

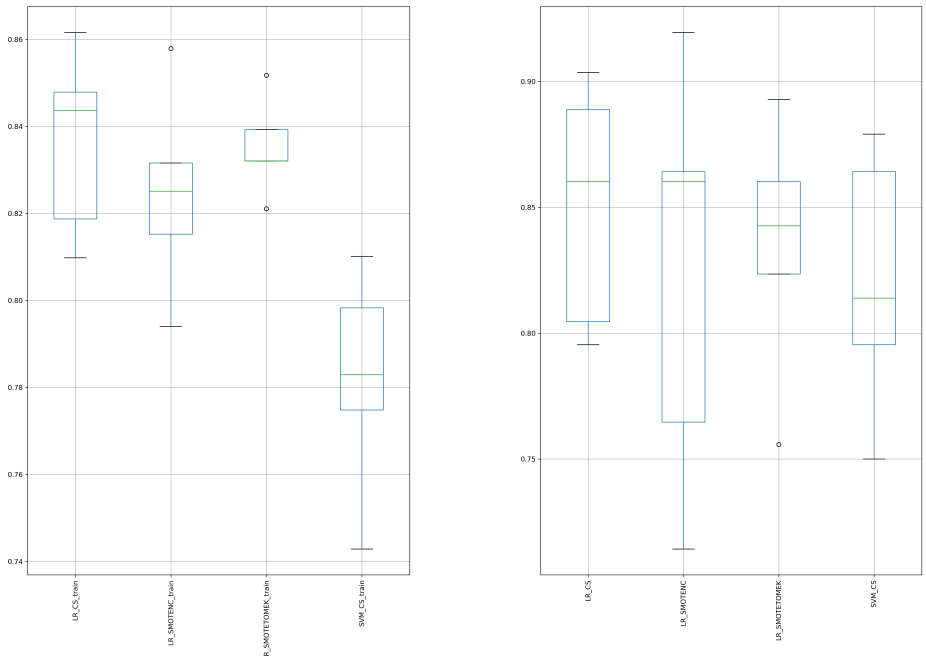
We can see that decision tree has the highest average accuracy with a big spread on the testing set. The spread is not going beyond 0.2% for any model on the training set or beyond 1% on the testing set.



(a) Training (b) Testing

Figure 11.12: Box Plots for CHD GU Defects - Accuracy

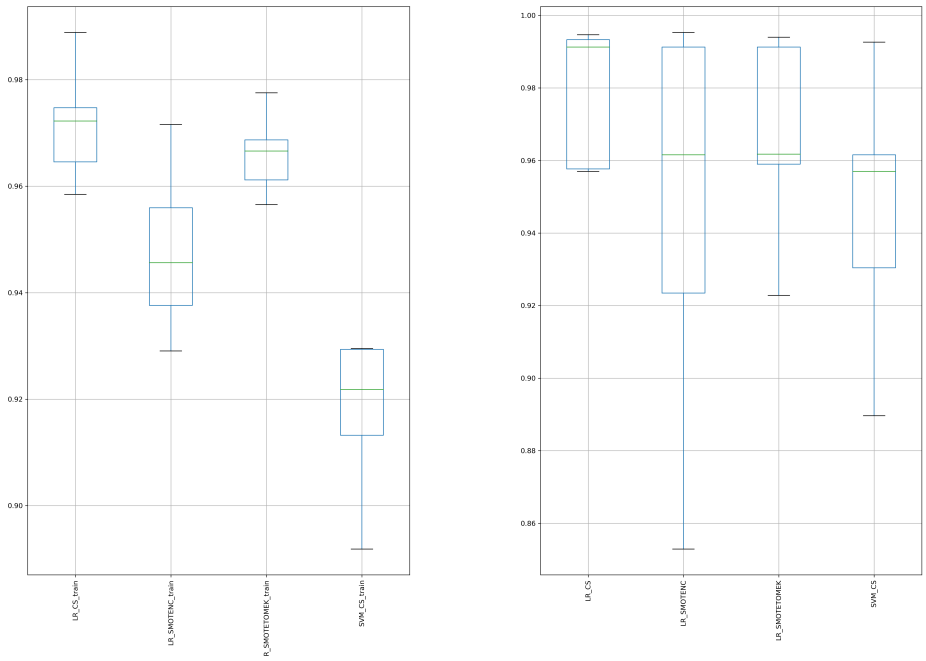
11.3.5. Observations For CHD - MUSC. We can see that the best performing model in terms of average F2 score is cost-sensitive logistic regression, however, the spread is around 4% on the training set and around 8% for the testing set. All other models have an acceptable performance on average, but it is noticeable that the spread is very high for the testing set.



(a) Training (b) Testing

Figure 11.13: Box Plots for CHD - MUSC Defects - F2 Score

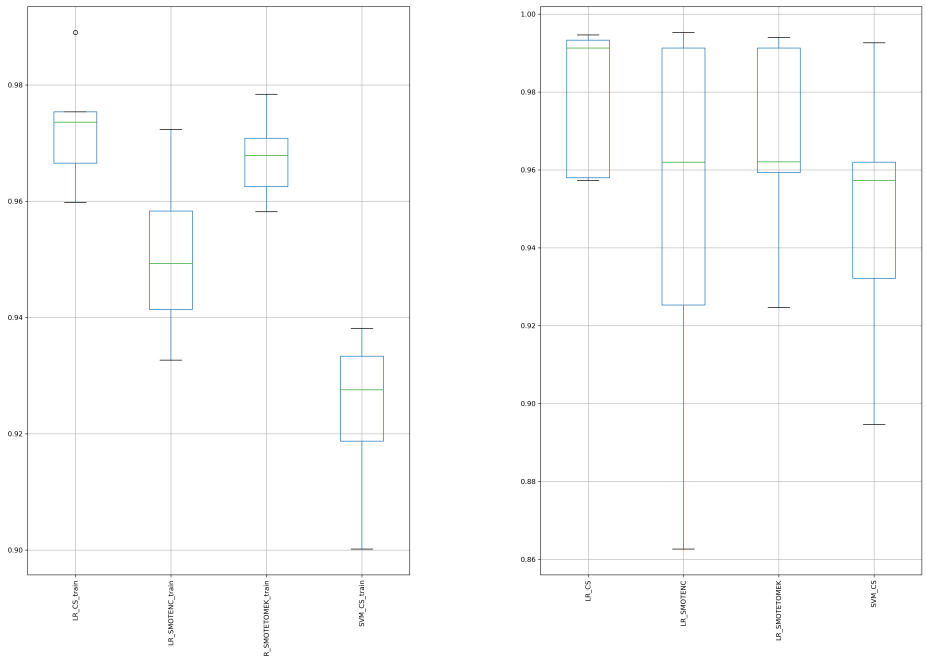
The best two models on average are cost-sensitive logistic regression and logistic regression with SMOTE TOMEK. The spread of both models is around 1% for the training set and 3% for the testing set. The other two models achieved lower scores on average.



(a) Training (b) Testing

Figure 11.14: Box Plots for CHD - MUSC Defects - Gmean Score

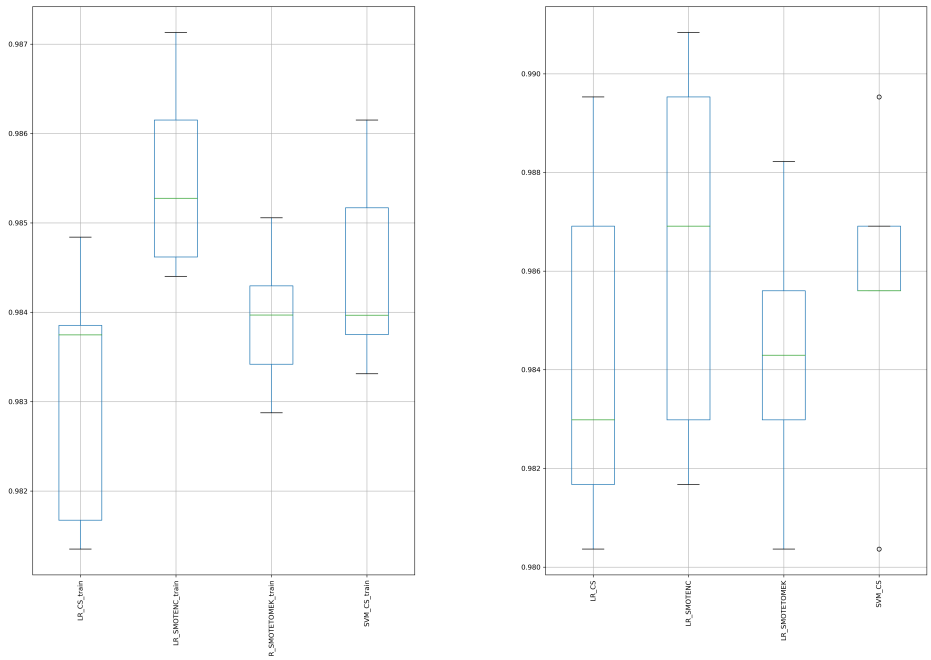
The best average AUC ROC is reported for cost-sensitive logistic regression same for the spread, the average is around 99% and the spread is around 3% on the training set. The other models have a lower average AUC ROC for both training and testing but the spread is the same.



(a) Training (b) Testing

Figure 11.15: Box Plots for CHD - MUSC Defects - AUC ROC

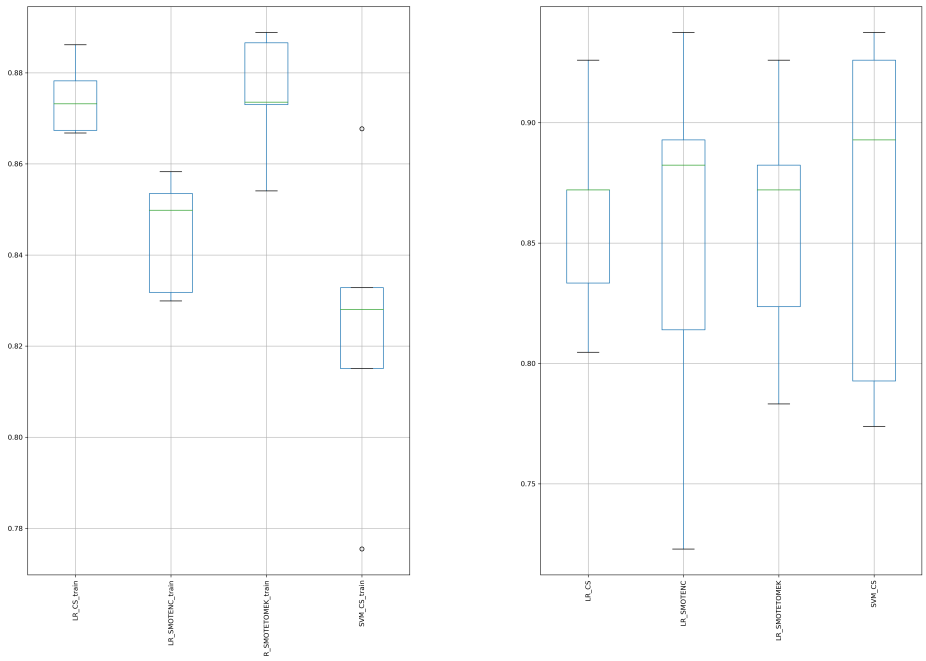
We can see that cost-sensitive logistic regression has the lowest average accuracy on both training and testing sets, also, it has the highest spread. But we can see that the accuracy is not very different for all the models, all the averages and spreads are reported between 98.2% and 98.7% on the training set and between 98% and 99% on the testing set.



(a) Training (b) Testing

Figure 11.16: Box Plots for CHD - MUSC Defects - Accuracy

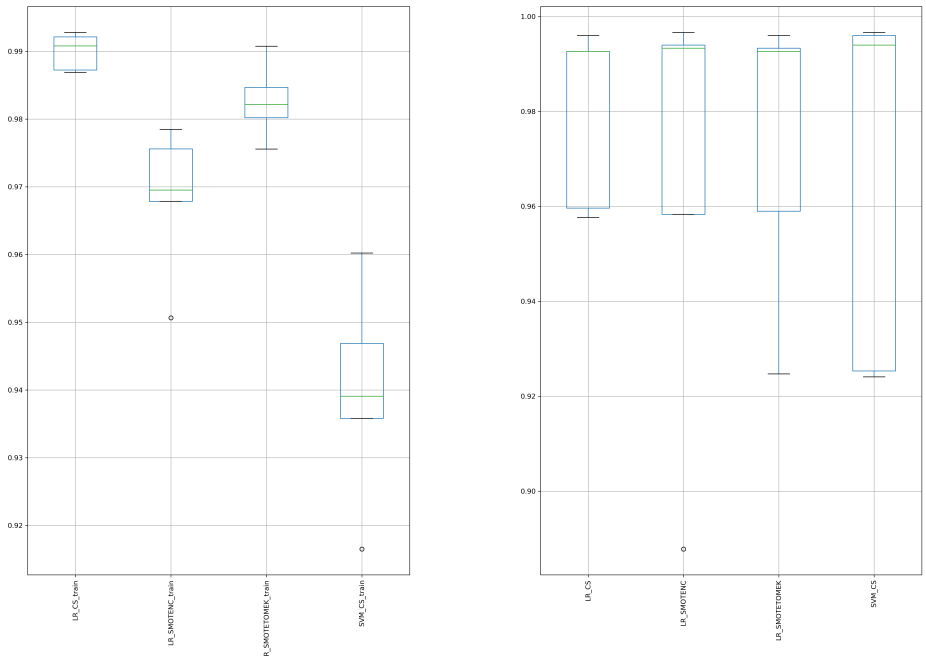
11.3.6. Observations For CNS - GU. Below we display the box plots for the F2 score of both training and testing sets of CNS and GU. First, the spread is wide on both training and testing sets. The best performing models in terms of average performance and spread are cost-sensitive logistic regression and logistic regression with SMOTE TOMEK. Their spread is less than 2% on the training set, cost-sensitive logistic regression has a lower spread on the testing set compared to the other models.



(a) Training (b) Testing

Figure 11.17: Box Plots for CNS - GU Defects - F2 Score

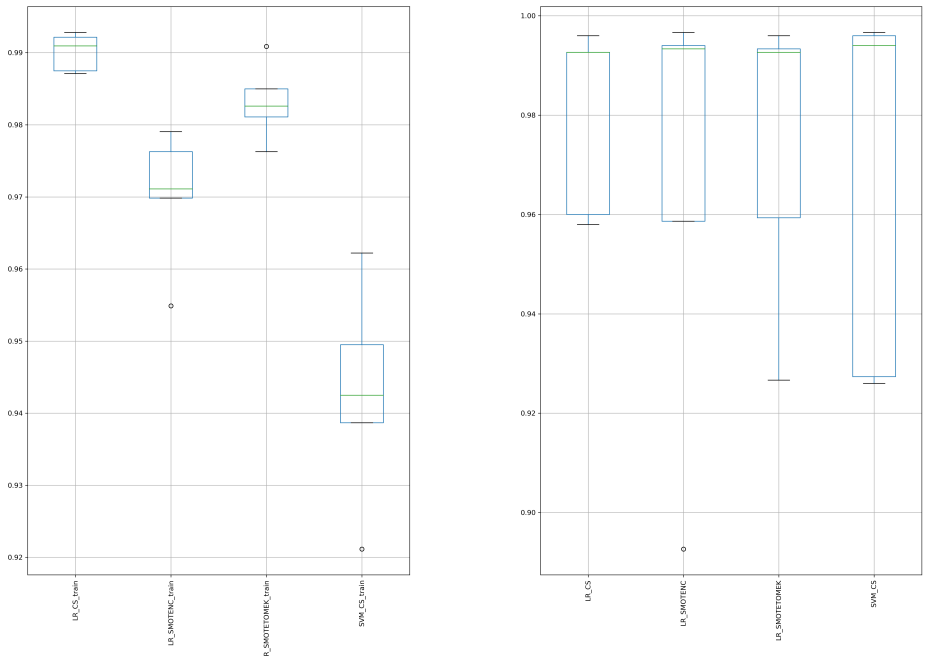
Below we display the Gmean scores for all models, we can see that the best performing model is cost-sensitive logistic regression, it has the highest score on average and its spread is around 0.5% on the training set. We can see that all models have similar performance on average on the testing set, but logistic regression with SMOTE TOMEK and cost-sensitive support vector machines have a wider spread.



(a) Training (b) Testing

Figure 11.18: Box Plots for CNS - GU Defects - Gmean Score

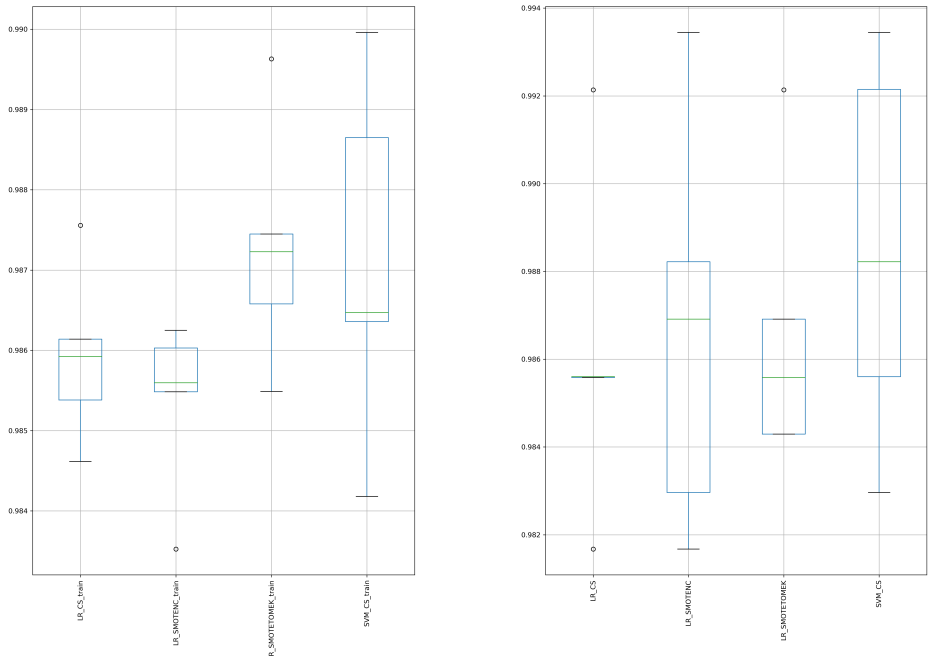
Similar to Gmean, cost-sensitive logistic regression has the highest average AUC ROC score on the training set. The spread is almost the same for the testing set.



(a) Training (b) Testing

Figure 11.19: Box Plots for CNS - GU Defects - AUC ROC

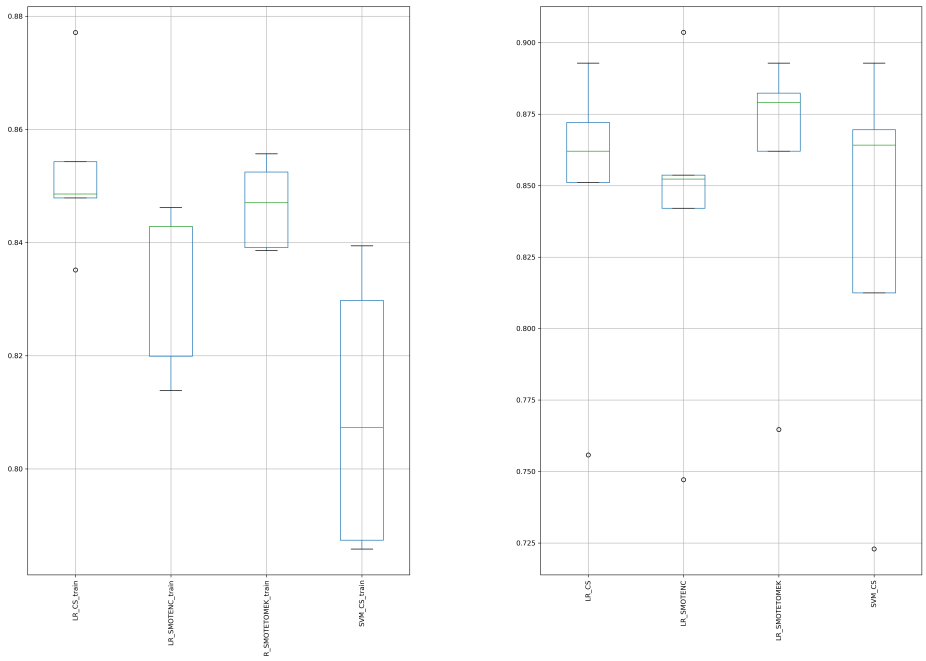
For accuracy, we can see that all the models have almost similar accuracies on the training set but with different spreads, all the values are between 98.5% and 99%. On the testing set, the minimal spread was recorded for cost-sensitive logistic regression, but again, all the average performance is similar.



(a) Training (b) Testing

Figure 11.20: Box Plots for CNS - GU Defects - Accuracy

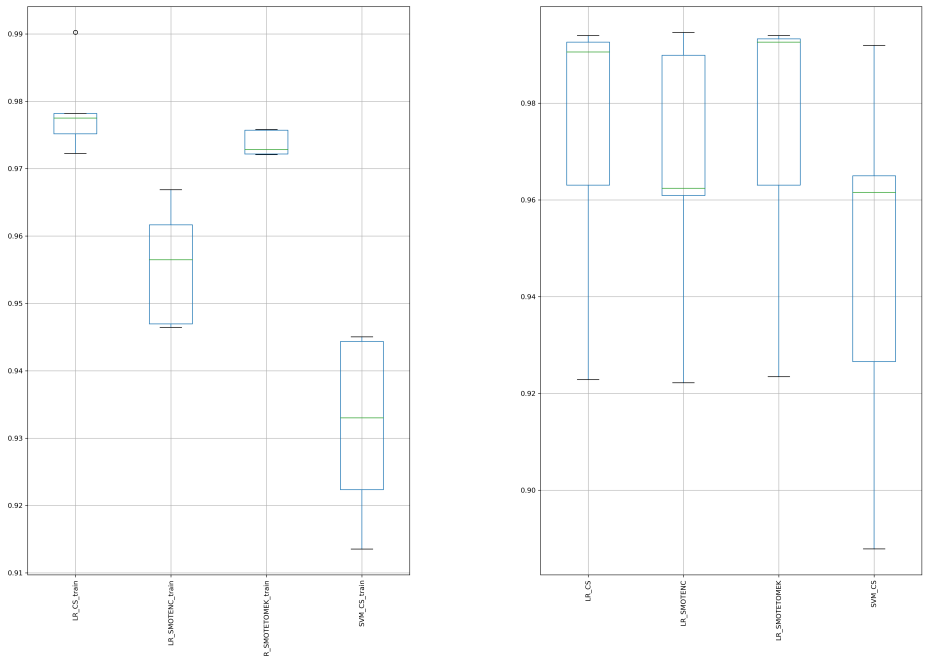
11.3.7. Observations For CNS - MUSC. Below, we display the box plots for the F2 score of CNS - MUSC. We can see that cost-sensitive logistic regression has the best average performance and the lowest spread on the training set. However, logistic regression with SMOTE TOMEK has slightly better performance on the testing set.



(a) Training (b) Testing

Figure 11.21: Box Plots for CNS - MUSC Defects - F2 Score

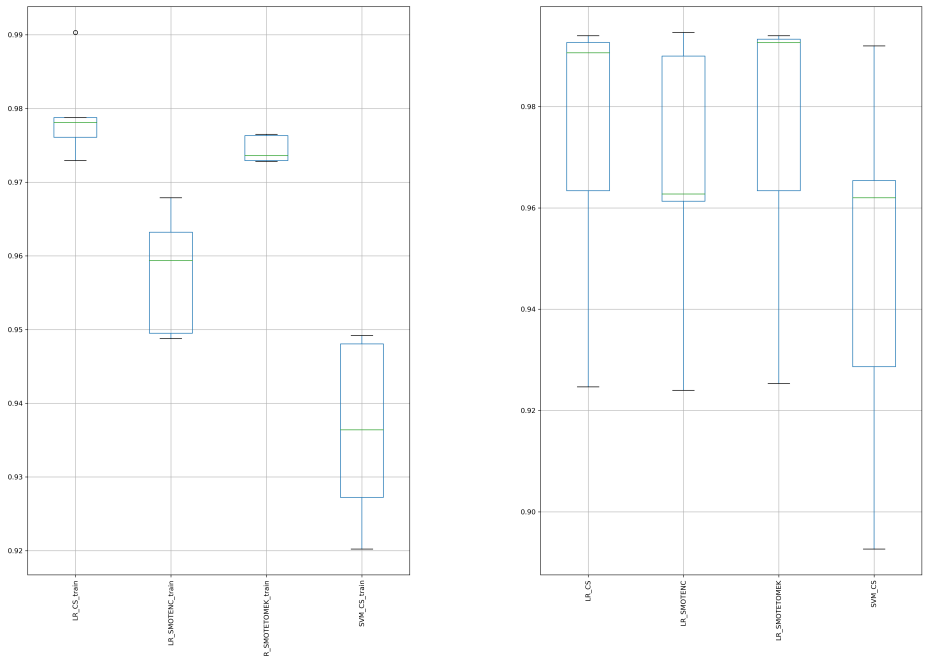
From the figures below, we can see that cost-sensitive logistic regression has the best average performance and lowest spread on the training set. However, we can see that the performance is similar for all models on the testing set except for cost sensitive support vector machines.



(a) Training (b) Testing

Figure 11.22: Box Plots for CNS - MUSC Defects - Gmean Score

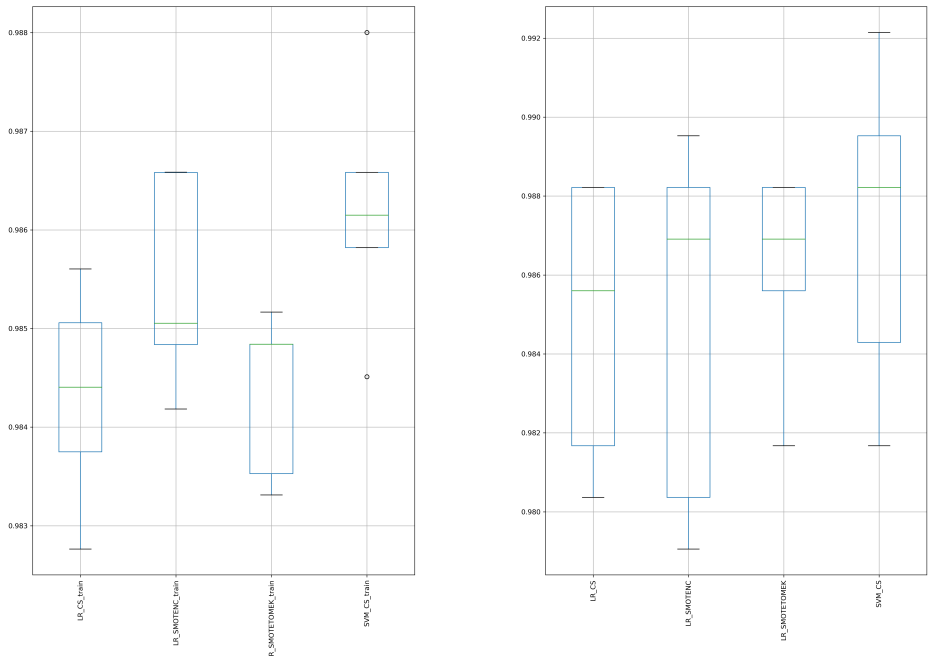
From the figures below, we can see that cost-sensitive logistic regression has the best average performance and lowest spread on the training set. However, we can see that the performance is similar for all models on the testing set except for cost sensitive support vector machines.



(a) Training (b) Testing

Figure 11.23: Box Plots for CNS - MUSC Defects - AUC ROC

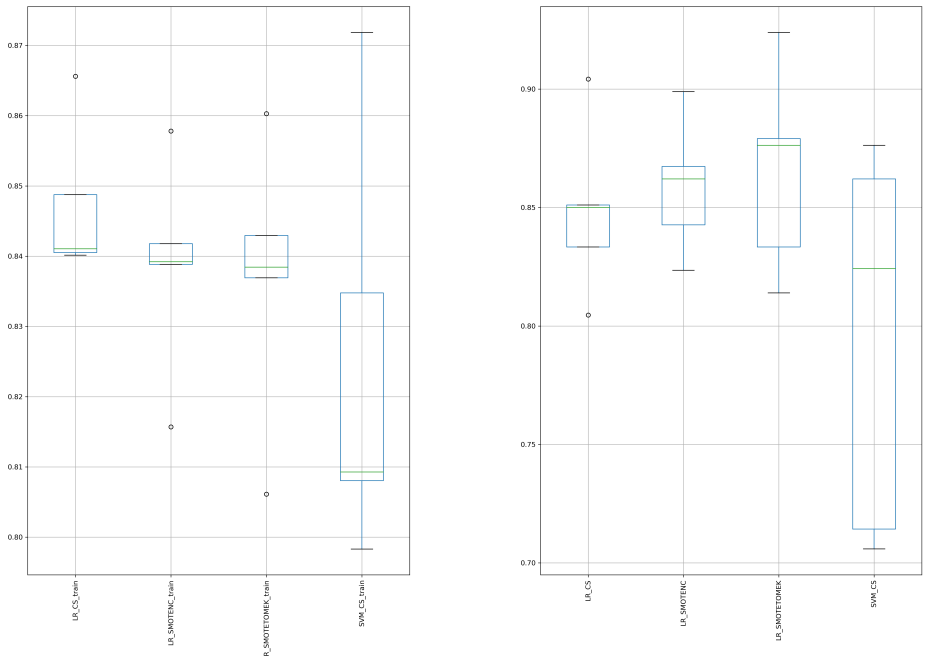
We can see that the performance of all models is similar, all reported accuracies are between 98.3% and 98.6% on the training set. The spread is minimal, almost 0.3%.



(a) Training (b) Testing

Figure 11.24: Box Plots for CNS - MUSC Defects - Accuracy

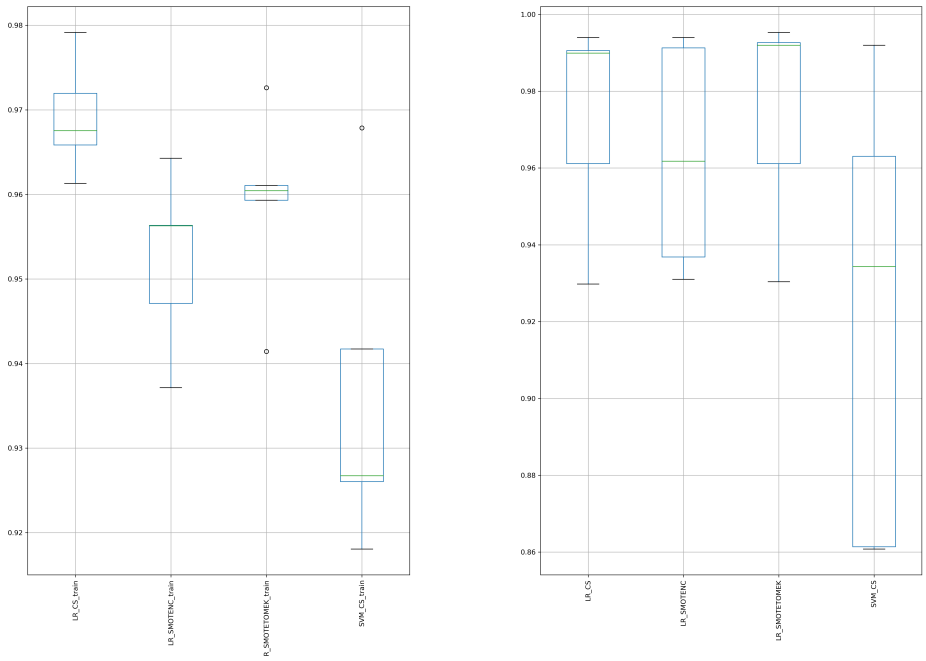
11.3.8. Observations For GU - MUSC. The best performing model on average that achieved the highest F2 score is cost-sensitive logistic regression. We can see that the average performance is around 85% and the spread is less than 1% on the training set. Also, we can see that logistic regression with SMOTE TOMEK has a good performance on the testing set, however, its spread is wider than the one of cost-sensitive logistic regression.



(a) Training (b) Testing

Figure 11.25: Box Plots for GU - MUSC Defects - F2 Score

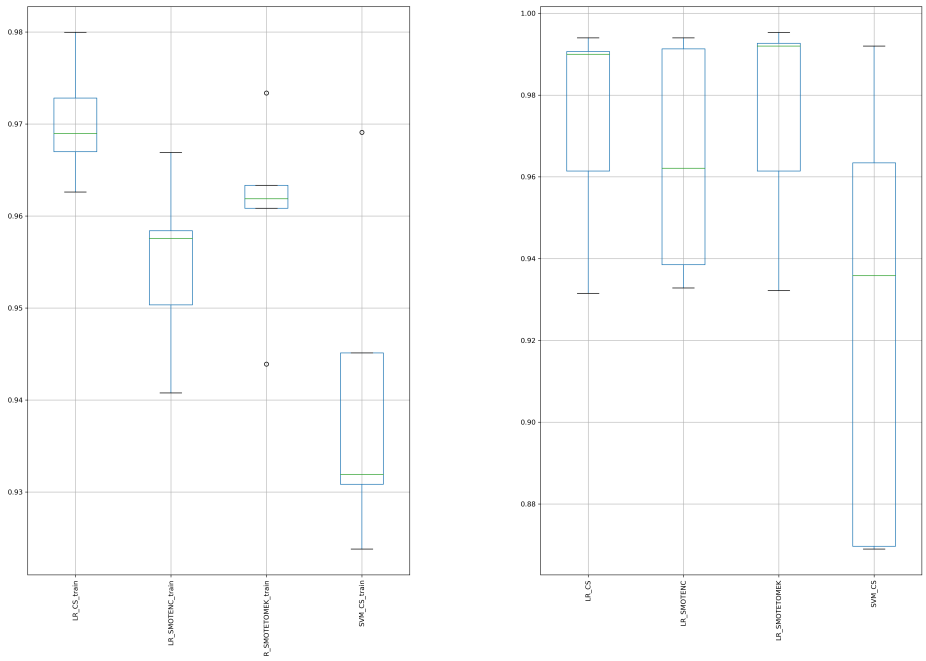
The best achieved average Gmean is for cost sensitive logistic regression. We can see that the spread is also minimal on the training set, almost less than 0.5%, and around 2% on the testing set.



(a) Training (b) Testing

Figure 11.26: Box Plots for GU - MUSC Defects - Gmean Score

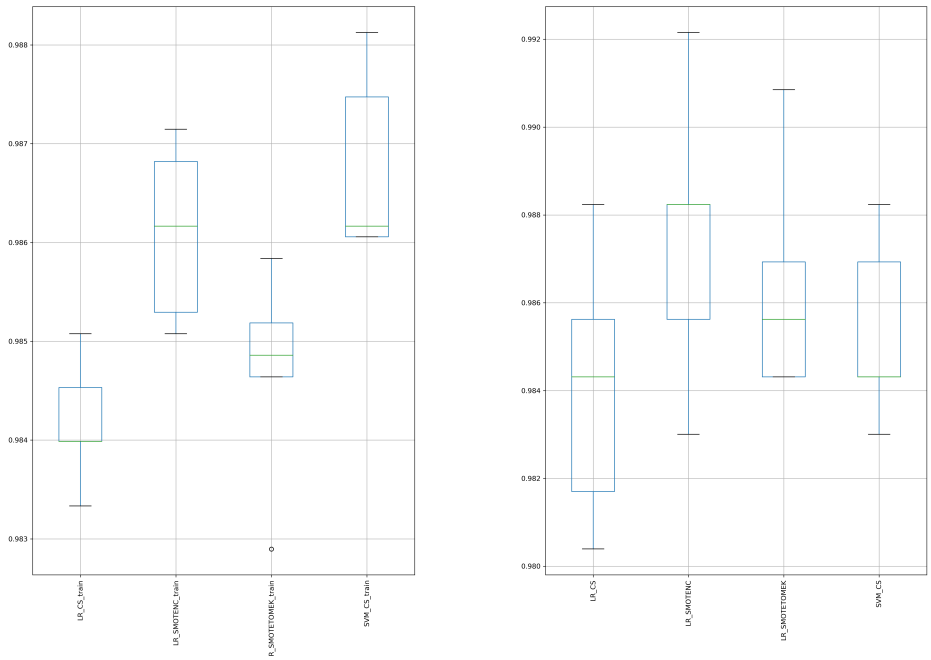
The best-achieved average AUC ROC is for cost-sensitive logistic regression. We can see that the spread is also minimal on the training set, almost less than 0.5%, and around 2% on the testing set.



(a) Training (b) Testing

Figure 11.27: Box Plots for GU - MUSC Defects - AUC ROC

All accuracies are similar for all models, the reported scores on the training set is between 98.3% and 98.8%. And the reported results on the testing set is between 98.2% and 99%. So the spread is acceptable.



(a) Training (b) Testing

Figure 11.28: Box Plots for GU - MUSC Defects - Accuracy

11.4. Observations on Advanced Metrics. In this section, we will assess the performance of the models using the following advanced metrics:

- Specificity
- Adjusted Geometric mean (AG-Mean)
- Cohen Kappa
- Macro-averaged accuracy (MAA)
- Mean class weighted accuracy (MCWA)
- Optimized precision

	Training	Testing
Specificity	0.9834(0.0013)	0.9842(0.0026)
AG-Mean	0.9865(0.0016)	0.9868(0.0049)
Cohen Kappa	0.8492(0.0087)	0.8522(0.0261)
MAA	0.9897(0.0029)	0.9894(0.0073)
MCWA	0.9897(0.0029)	0.9894(0.0073)
Optimized precision	1.9698(0.0074)	1.9722(0.0129)

Table 11.1: Advanced Metrics for All Defects - Cost Sensitive Logistic Regression

	Training	Testing
Specificity	0.9841(0.0013)	0.9842(0.004)
AG-Mean	0.9815(0.0022)	0.9807(0.01)
Cohen Kappa	0.6866(0.0169)	0.679(0.0471)
MAA	0.9795(0.0037)	0.9773(0.0199)
MCWA	0.9795(0.0037)	0.9773(0.0199)
Optimized precision	1.9394(0.0108)	1.9371(0.0526)

Table 11.2: Advanced Metrics for CHD - CNS - Cost Sensitive Logistic Regression

	Training	Testing
Specificity	0.9849(0.0009)	0.985(0.0035)
AG-Mean	0.9803(0.006)	0.9785(0.0129)
Cohen Kappa	0.7156(0.0185)	0.7033(0.0233)
MAA	0.9763(0.0112)	0.9725(0.0282)
MCWA	0.9763(0.0112)	0.9725(0.0282)
Optimized precision	1.9301(0.0327)	1.9212(0.0814)

Table 11.3: Advanced Metrics for CHD - GU - Cost Sensitive Logistic Regression

	Training	Testing
Specificity	0.9835(0.0013)	0.9845(0.0034)
AG-Mean	0.9777(0.0061)	0.9816(0.0111)
Cohen Kappa	0.6978(0.0219)	0.7086(0.0612)
MAA	0.9729(0.0109)	0.9789(0.0194)
MCWA	0.9729(0.0109)	0.9789(0.0194)
Optimized precision	1.9194(0.0321)	1.9435(0.0491)

Table 11.4: Advanced Metrics for CHD - MUSC - Cost Sensitive Logistic Regression

	Training	Testing
Specificity	0.9858(0.0011)	0.9864(0.0035)
AG-Mean	0.9879(0.0014)	0.983(0.0107)
Cohen Kappa	0.7397(0.0152)	0.7296(0.0622)
MAA	0.9901(0.0027)	0.9799(0.0191)
MCWA	0.9901(0.0027)	0.9799(0.0191)
Optimized precision	1.9705(0.0077)	1.9452(0.0495)

Table 11.5: Advanced Metrics for CNS - GU - Cost Sensitive Logistic Regression

	Training	Testing
Specificity	0.9845(0.0009)	0.9853(0.0034)
AG-Mean	0.9816(0.0038)	0.979(0.0157)
Cohen Kappa	0.7167(0.0191)	0.7127(0.0585)
MAA	0.9792(0.0066)	0.9731(0.0299)
MCWA	0.9792(0.0066)	0.9731(0.0299)
Optimized precision	1.9386(0.0193)	1.9236(0.0837)

Table 11.6: Advanced Metrics for CNS - MUSC - Cost Sensitive Logistic Regression

	Training	Testing
Specificity	0.9848(0.0006)	0.9845(0.0037)
AG-Mean	0.977(0.0034)	0.9788(0.0129)
Cohen Kappa	0.7228(0.0107)	0.7149(0.0408)
MAA	0.9703(0.0065)	0.9735(0.0269)
MCWA	0.9703(0.0065)	0.9735(0.0269)
Optimized precision	1.9119(0.0189)	1.9248(0.0762)

Table 11.7: Advanced Metrics for GU - MUSC - Cost Sensitive Logistic Regression

11.5. Interpretation. From the box plots in the observation section, we can filter the models by best F2 score, Gmean, AUC ROC, and Accuracy for both testing and training.

For All defects sheet, we get the following models with their respective configurations:

- Decision Tree with SMOTE TOMEK
- Cost Sensitive Logistic Regression
- Logistic Regression using SMOTENC
- Logistic Regression using SMOTE TOMEK
- Cost Sensitive Support Vector Machine

However, after performing statistical hypotheses testing between all configurations of logistic regression and decision tree, it was clear that cost-sensitive logistic regression is performing better according to F2 scores.
Hypothesis testing was performed using the 5 x 2 cv paired t test proposed in [27] with a significance level $\alpha = 0.05$. 5x2 cv paired t test is a paired null hypotheses testing method that performs 5 iterations of two-fold cross-validation to evaluate the null hypothesis of dependent variables. The p-value of all tests is greater than α, this means that the null hypothesis of the models having different performance is rejected; therefore, the models have the same mean performance.

Now, for the CHD, CNS, GU, and MUSC datasets. All models have a poor performance. The results are below the acceptable rate or the baseline ones.

However, we can mention that there was some improvement in the results for most of the cost-sensitive versions of the algorithms.

When mixing the birth defects, we were able to achieve a much better performance compared to the performance reported on the single defect datasets:

For CHD-CNS, cost-sensitive logistic regression has the best average performance for F2 score, Gmean, AUC ROC, and Accuracy for both testing and training.

For CHD-GU, cost-sensitive logistic regression has the best average performance for F2 score, Gmean, AUC ROC, and Accuracy for both testing and training.

For CHD-MUSC, cost-sensitive logistic regression has the best average performance for F2 score, Gmean, AUC ROC, and Accuracy for both testing and training.

For CNS-GU, cost-sensitive logistic regression has the best average performance for F2 score, Gmean, AUC ROC, and Accuracy for both testing and training.

For CNS-MUSC, cost-sensitive logistic regression has the best average performance for F2 score, Gmean, AUC ROC, and Accuracy for both testing and training.

For GU-MUSC, cost-sensitive logistic regression has the best average performance for F2 score, Gmean, AUC ROC, and Accuracy for both testing and training.

Also, for all defects sheets and per pair of defects we can see that cost-sensitive logistic regression is achieving the best Specificity, AG-Mean, Cohen Kappa, MAA, MCWA, and Optimized precision.

So as a conclusion, we can see that cost-sensitive logistic regression has always better average performance and minimal spread compared to other models on all pairs of defects.

Chapter 12

Probability Prediction

12.1. Probability Prediction Experiments. In the probability prediction section, we used nested cross-validation described in section 6.13. to assess the performance of several machine learning models with different configurations.

In this process, we test our models using several scaling techniques and we optimize the results for several metrics.

For the scaling techniques, models are tested first without scaling, then for linear models, we add the following scaling techniques: Normalization, Standardization, Normalization followed by Power Transform and a mixed approach where we perform standardization for normally distributed columns and normalization for the other numerical columns.

During the optimization process, the model chooses the best configuration which optimizes a specific metric. In our case, we optimized for only one metric in the probability prediction process, which is the Brier Skill Score (BSS).

The list of models that we used is the following:

- GNB
- GNB with oversampled training using SMOTENC
- GPC
- GPC with oversampled training using SMOTENC
- LDA
- LDA with oversampled training using SMOTENC

- MNB
- MNB with oversampled training using SMOTENC
- QDA
- QDA with oversampled training using SMOTENC
- Logistic Regression
- Cost sensitive logistic regression
- Calibrated SVM
- Calibrated KNN
- Calibrated Decision Trees

12.1.1. Used Metrics And Desirable Bounds. We are reporting the results of the following metrics:

- Brier Skill Score (BSS) - Desirable bounds:
  - All Defects: desirable numbers are above 0.05
  - For Per Defect Datasets: desirable numbers are above 0.01
- AUC ROC - desirable bounds: should go above 50% up to 100%
- AUC PR - desirable bounds: should go above 50% up to 100%

For each metric, we report validation results (training) and testing results. For validation results, we report the average across all training splits and the standard deviation across them. The same thing is done for testing results.

In the tables of results, we always start with the average of (validation or testing) and following by the standard deviation of validation or testing (between parentheses).
A good BSS score for the sheet containing all defects is a number above 0.05 because the minority represent 5% of the overall data.

A good BSS score for all other sheets is a number above 0.01 because the minority represents almost 1% of the overall data.

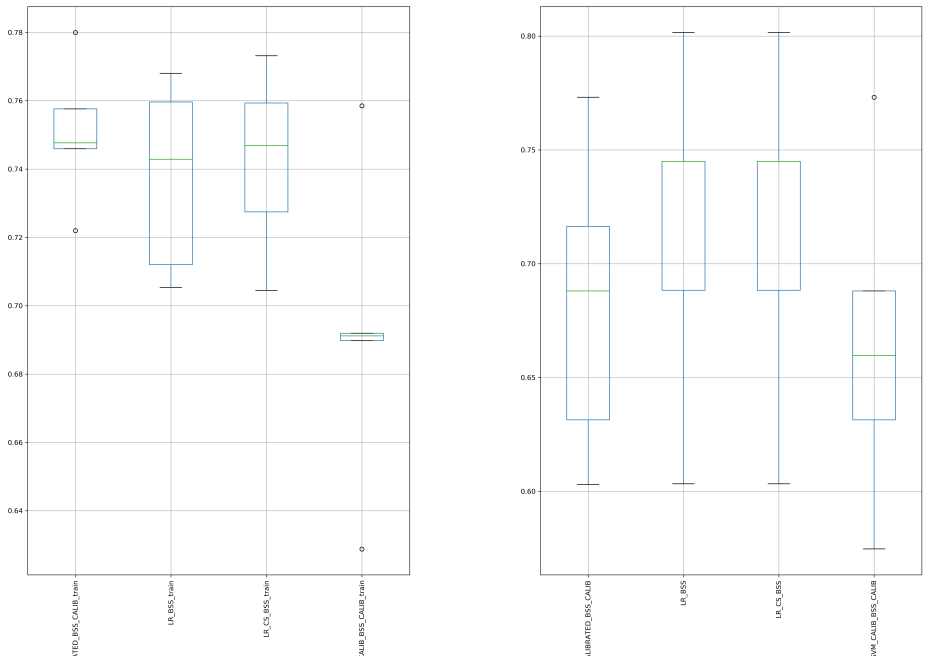
To choose a reference Brier Skill Score (BSS) we refer to the percentage of the minority class in the data. If the model recorded a BSS lower or equal to the

percentage of the minority class, this means that the model has no skill, and the process will be worse than having a random decision based on the percentage of the data. If the model is achieving a higher score, this means that the model has an advantage over the random decision.

12.2. Results. Detailed results tables are present in the appendix, please refer to tables B.141 B.142 B.143 B.144 B.145 B.146 B.147 B.148 B.149 B.150 B.151.

12.3. Observations.

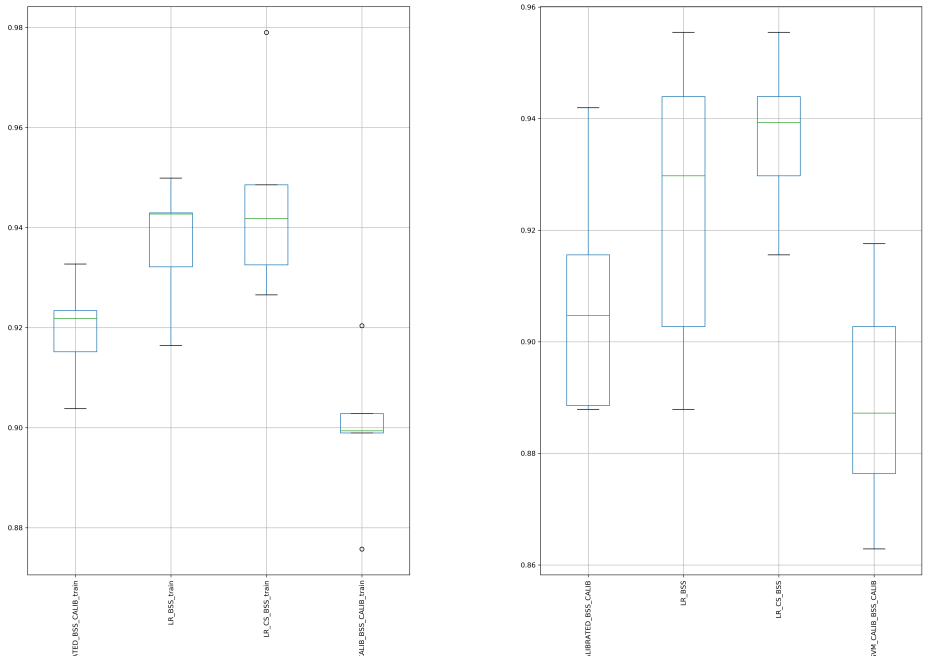
12.3.1. Observations for All Defects. Below we display the box plots for the best set of models. All other sets of box plots are displayed in the appendix.
First, we will display comparative box plots for the Brier Skill Score of all the picked models. Results produced by testing are more sparse compared to the results produced on the validation. If we compare the averages, we can see that the calibrated decision tree has the highest score on the training set and is less spread than the plots produced by logistic regression and cost sensitive logistic regression. Also, we can see that cost-sensitive logistic regression has a slightly better performance on average compared to the normal logistic regression. Calibrated support vector machines are performing well, but the other models are performing better for both training and testing.



(a) Training (b) Testing

Figure 12.1: Box Plots for All Defects Brier Skill Score

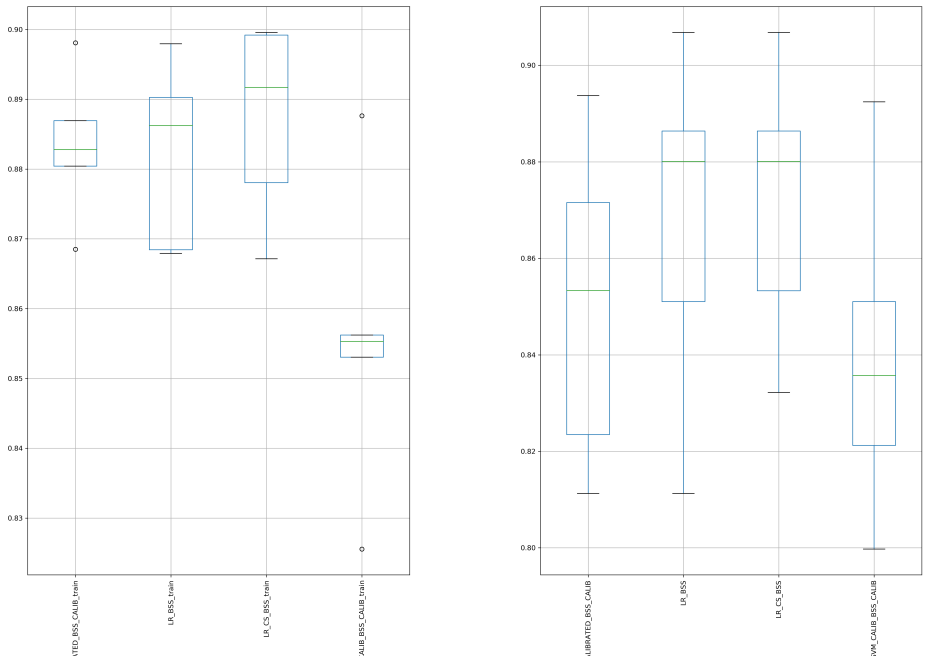
Second, we will compare box plots for AUC ROC training and testing. We can see that all four models are performing well in terms of AUC ROC. The average performance is around 90%. Again, for AUC ROC curves, we can see that cost-sensitive logistic regression has the best performance, normal logistic regression, then calibrated decision tree, and finally calibrated support vector machines.



(a) Training (b) Testing

Figure 12.2: Box Plots for All Defects AUC ROC

Now, we will compare the performance of the models based on the AUC Precision-Recall metric. AUC Precision-Recall is important for us because we want to evaluate the performance of all the models in predicting the positive class and to predict risk scores. Again, cost-sensitive logistic regression has the best performance for both training and testing, then normal logistic regression, then calibrated decision tree, and finally calibrated support vector machines.



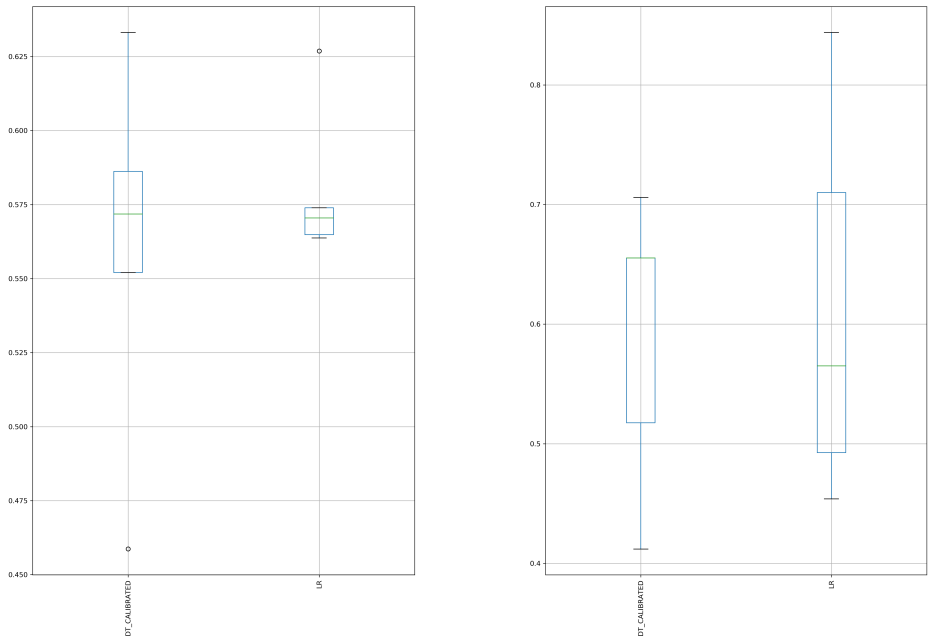
(a) Training (b) Testing

Figure 12.3: Box Plots for All Defects AUC PR

12.3.2. Observations For Per Defect Results. For CHD, CNS, GU, and MUSC datasets, all models performed poorly, and could not predict the defects accurately.
We proposed a solution for this issue by merging the datasets two by two and running the models on the merged datasets. So, we ended up with six new datasets:

- CHD - CNS
- CHD - GU
- CHD - MUSC
- CNS - GU
- CNS - MUSC
- GU - MUSC

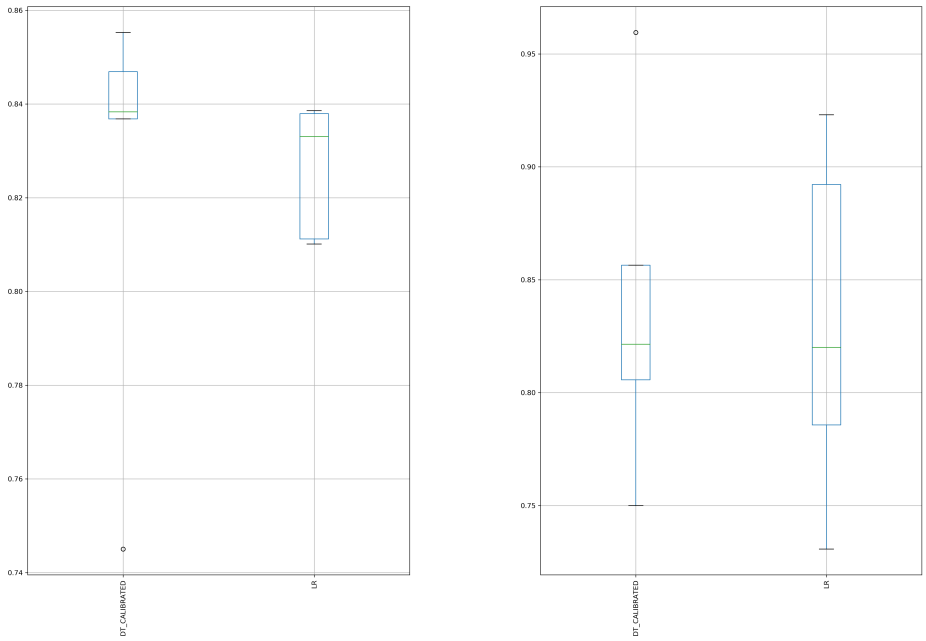
12.3.3. Observations For CHD - CNS. We can see that BSS is similar for both models on average for both training and testing sets. But the spread is higher for calibrated decision tree on the training set and wider for logistic regression on the testing set.



(a) Training (b) Testing

Figure 12.4: Box Plots for CHD - CNS Brier Skill Score

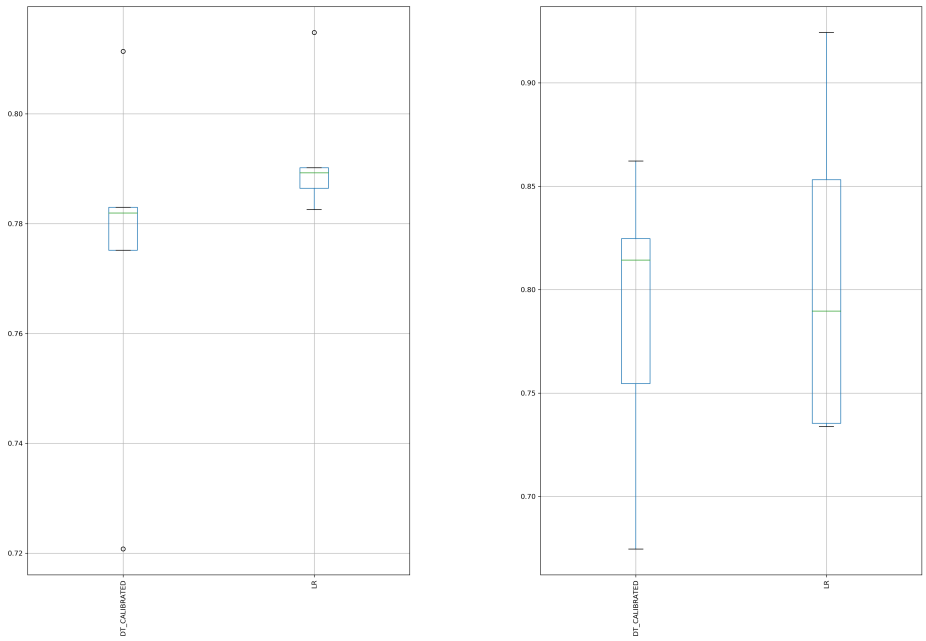
We can see that calibrated decision tree has better performance than logistic regression because the average is higher on the training set and the spread is smaller for both training and testing sets.



(a) Training (b) Testing

Figure 12.5: Box Plots for CHD - CNS AUC ROC

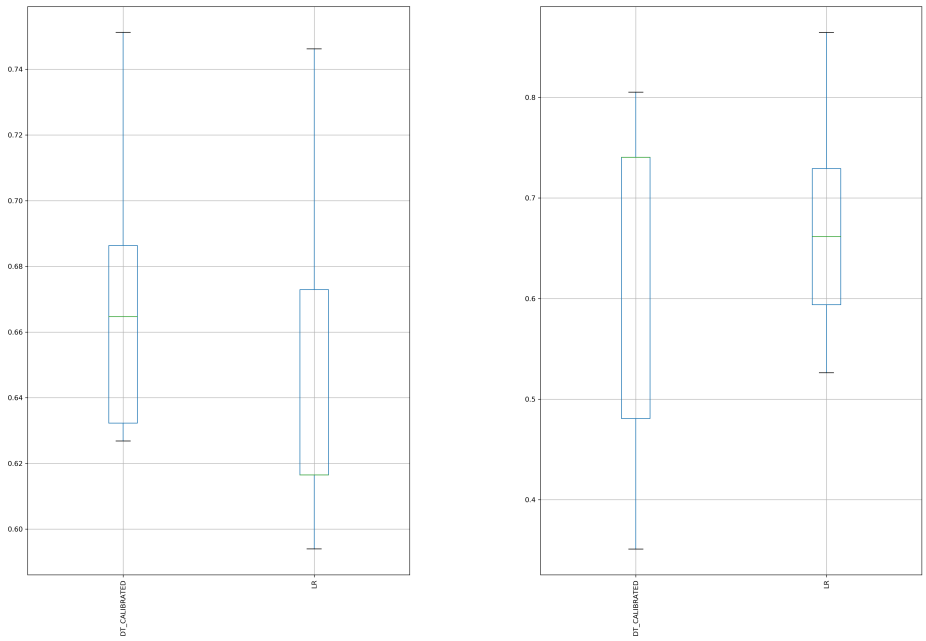
Logistic regression has a better average AUC PR on the training set and a smaller spread. However, we can see that both models are producing similar results on the testing sets.



(a) Training (b) Testing

Figure 12.6: Box Plots for CHD - CNS AUC PR

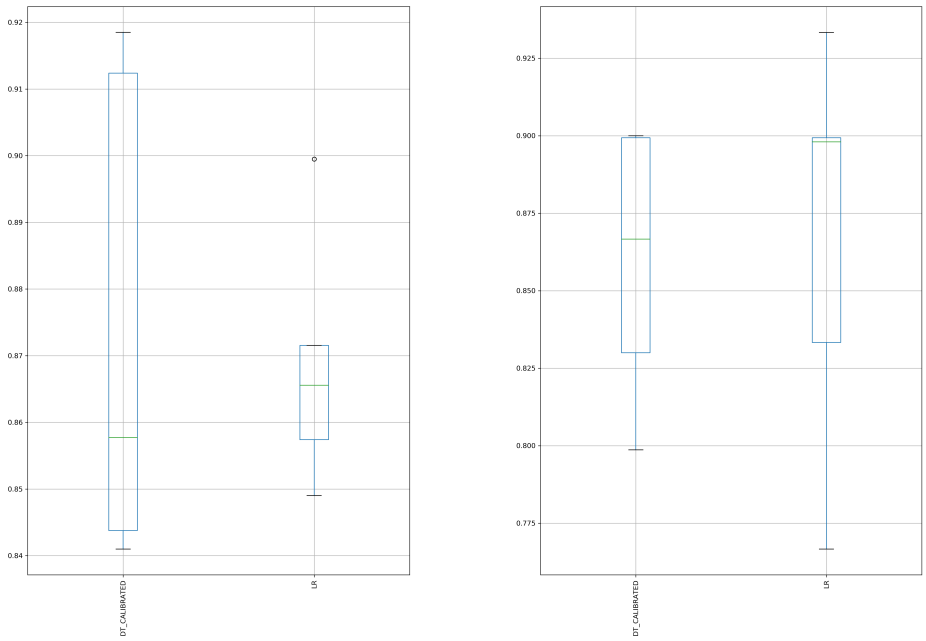
12.3.4. Observations For CHD - GU. We can see that the BSS score is better for logistic regression because the average is higher and the spread is smaller on the testing set.



(a) Training (b) Testing

Figure 12.7: Box Plots for CHD - GU Brier Skill Score

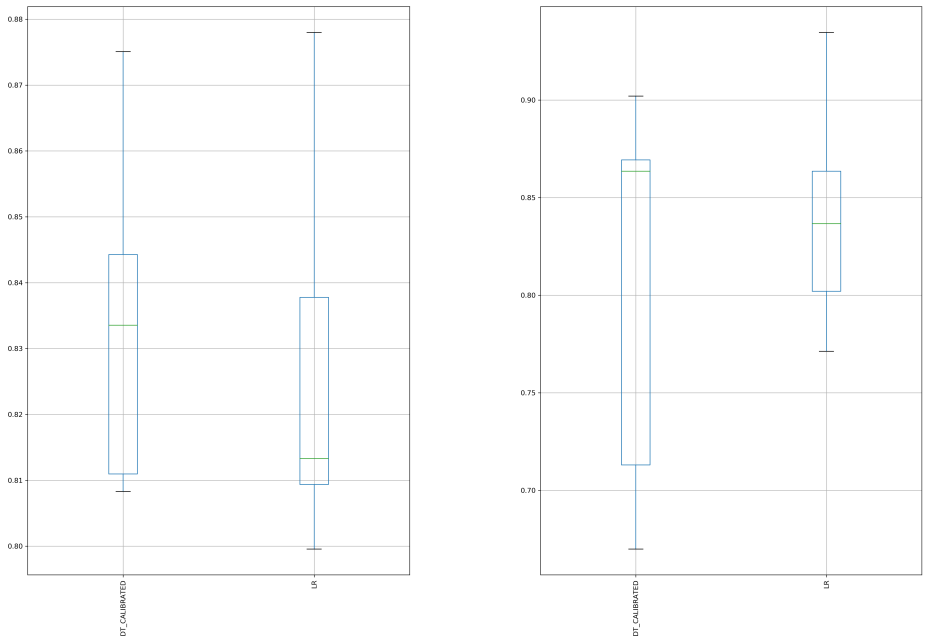
Also, AUC ROC scores are better for logistic regression, both average and spread are better than calibrated decision tree for both training and testing sets.



(a) Training (b) Testing

Figure 12.8: Box Plots for CHD - GU AUC ROC

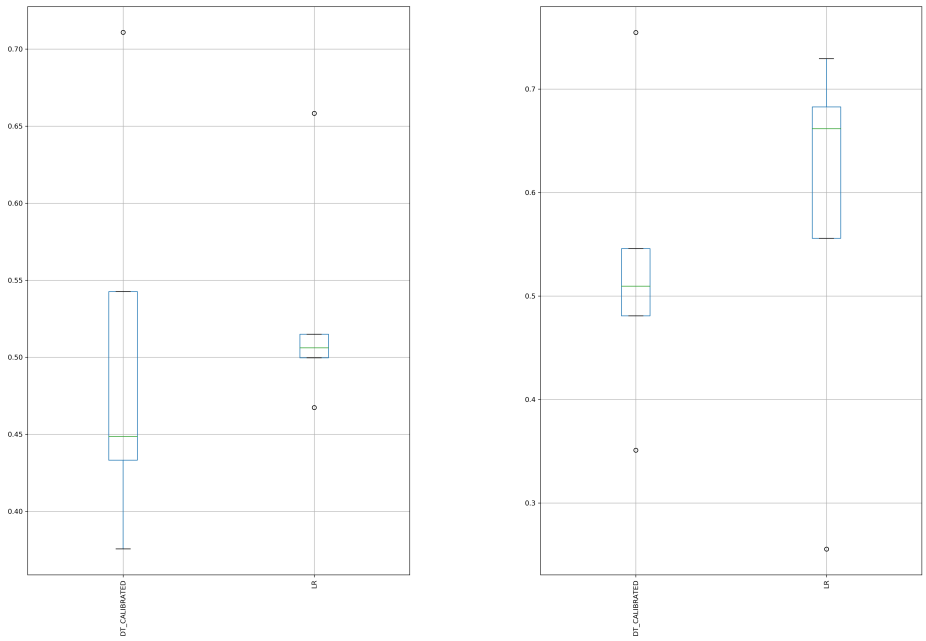
Also, AUC PR scores are better for logistic regression, both average and spread are better than calibrated decision tree for both training and testing sets.



(a) Training (b) Testing

Figure 12.9: Box Plots for CHD - GU AUC PR

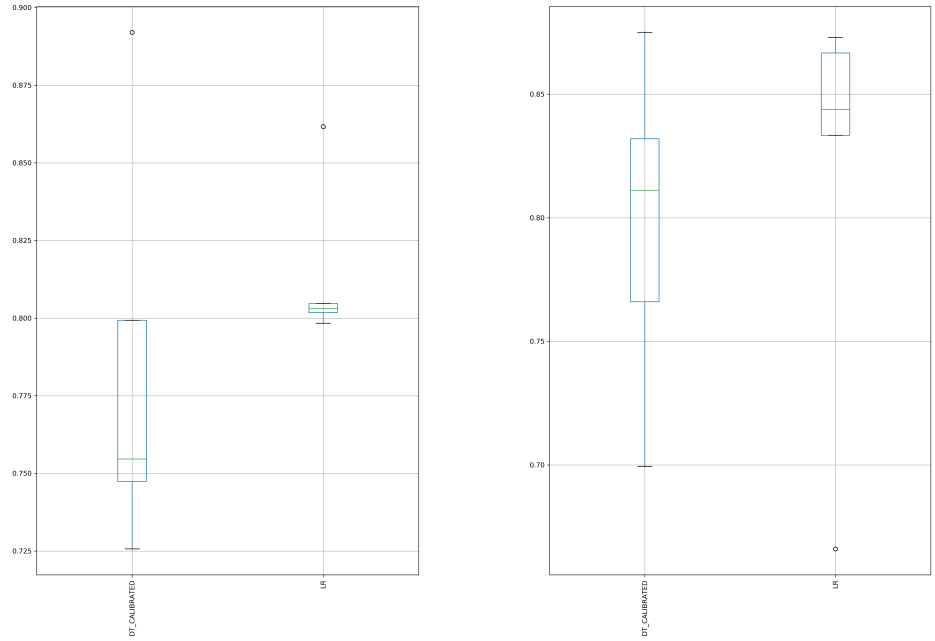
12.3.5. Observations For CHD - MUSC. We can see that the BSS score is better for logistic regression because the average is higher and the spread is smaller on the training set and a higher average score on the testing set.



(a) Training (b) Testing

Figure 12.10: Box Plots for CHD - MUSC Brier Skill Score

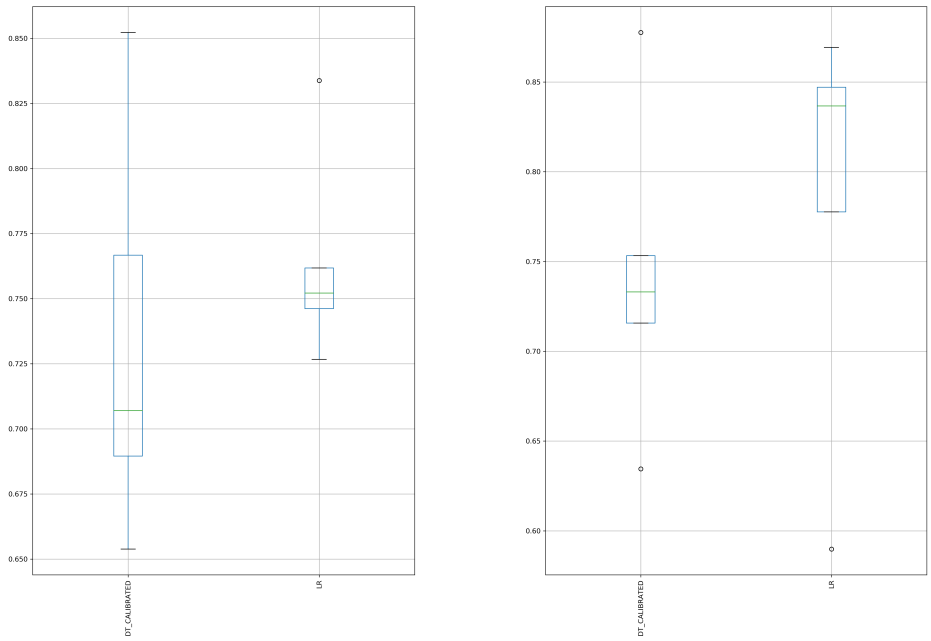
Also, AUC ROC scores are better for logistic regression, both average and spread are better than calibrated decision tree for both training and testing sets.



(a) Training (b) Testing

Figure 12.11: Box Plots for CHD - MUSC AUC ROC

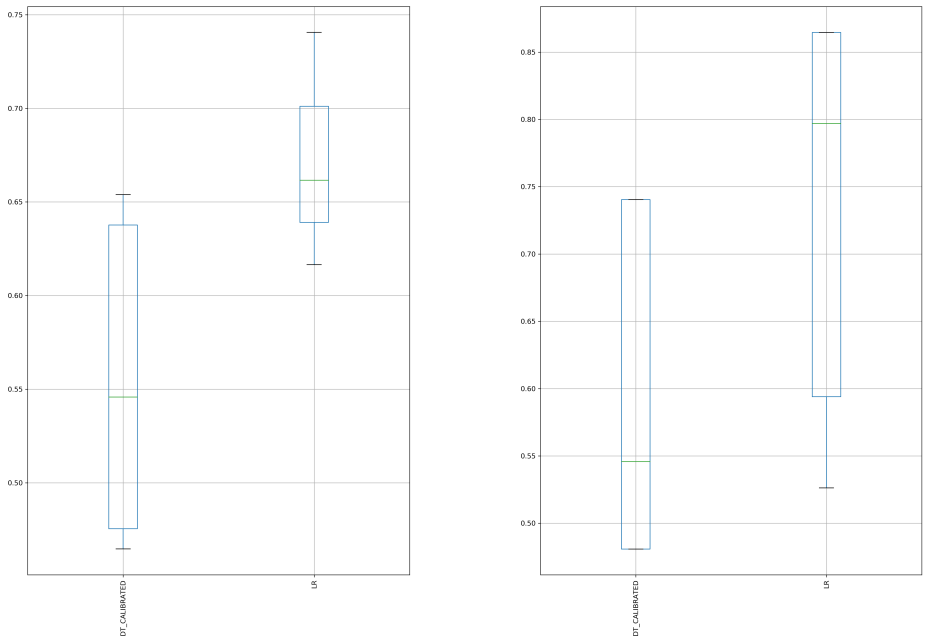
AUC PR scores are better for logistic regression, both average and spread are better than calibrated decision tree for both training and testing sets.



(a) Training (b) Testing

Figure 12.12: Box Plots for CHD - MUSC AUC PR

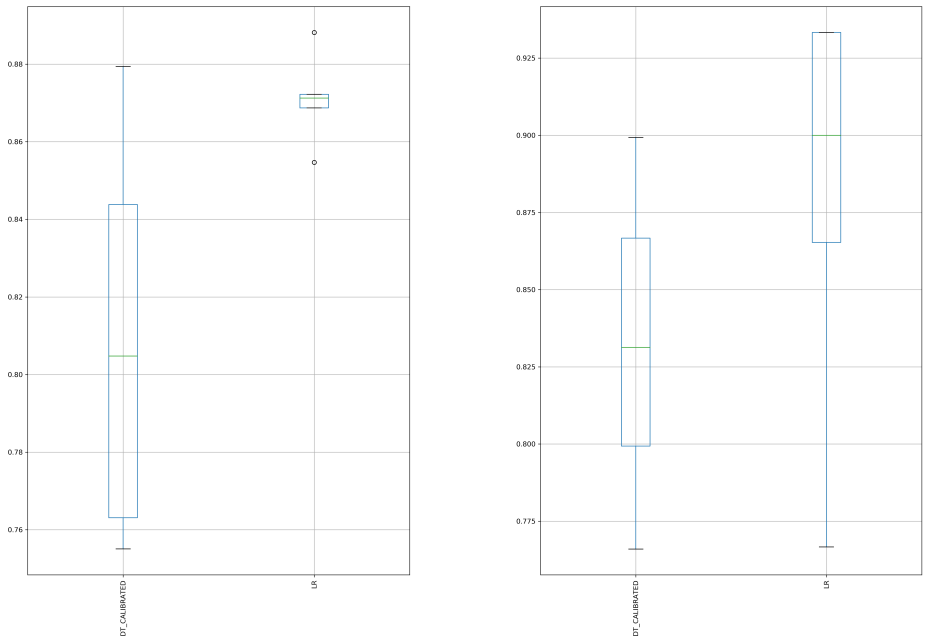
12.3.6. Observations For CNS - GU. BSS scores are better for logistic regression, both average and spread are better than calibrated decision tree for both training and testing sets.



(a) Training (b) Testing

Figure 12.13: Box Plots for CNS - GU Brier Skill Score

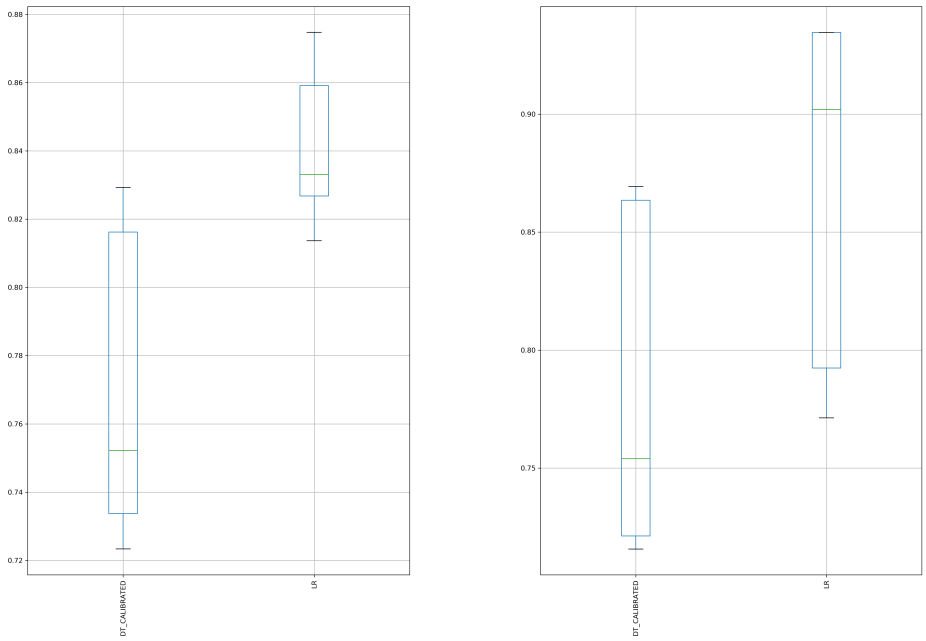
Also, AUC ROC scores are better for logistic regression, both average and spread are better than calibrated decision tree for both training and testing sets.



(a) Training (b) Testing

Figure 12.14: Box Plots for CNS - GU AUC ROC

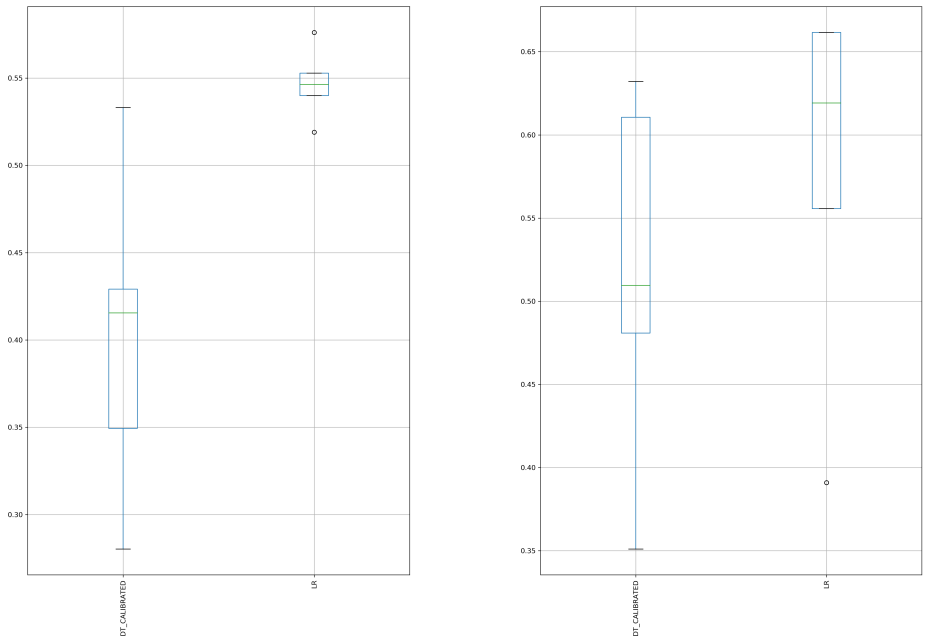
Also, AUC PR scores are better for logistic regression, both average and spread are better than calibrated decision tree for both training and testing sets.



(a) Training (b) Testing

Figure 12.15: Box Plots for CNS - GU AUC PR

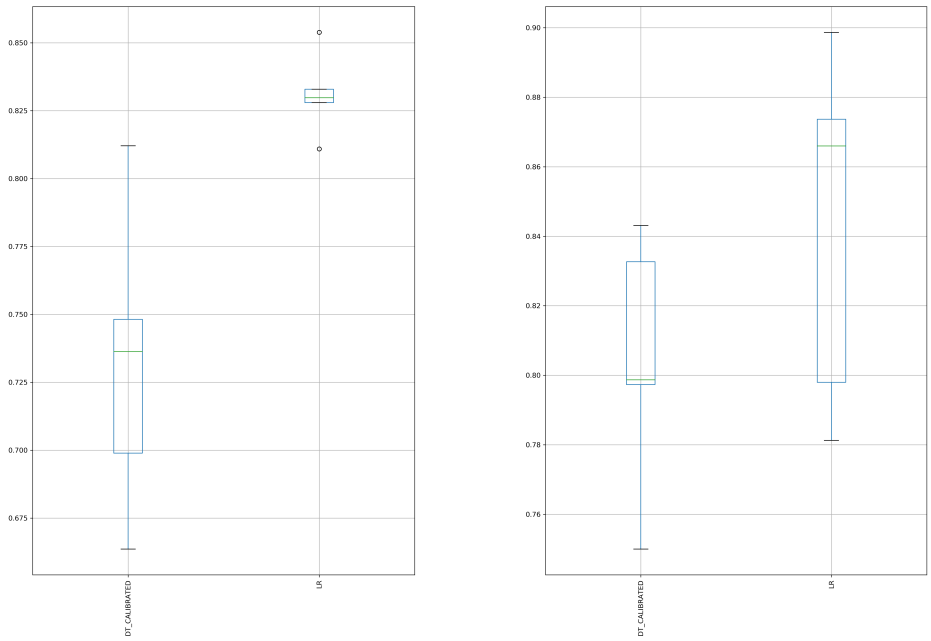
12.3.7. Observations For CNS - MUSC. BSS scores are better for logistic regression, both average and spread are better than calibrated decision tree for both training and testing sets.



(a) Training (b) Testing

Figure 12.16: Box Plots for CNS - MUSC Brier Skill Score

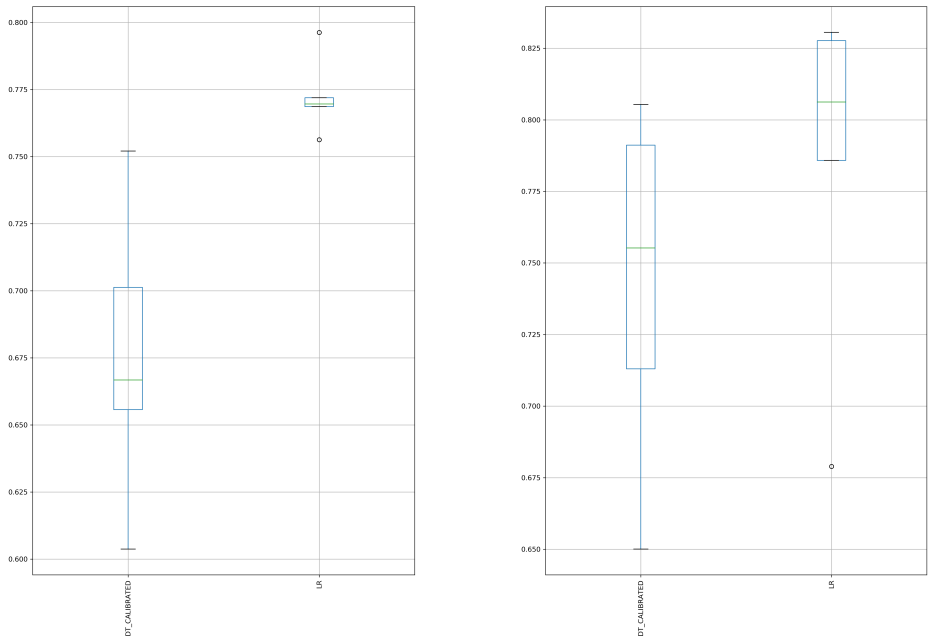
Also, AUC ROC scores are better for logistic regression, both average and spread are better than calibrated decision tree for both training and testing sets.



(a) Training (b) Testing

Figure 12.17: Box Plots for CNS - MUSC AUC ROC

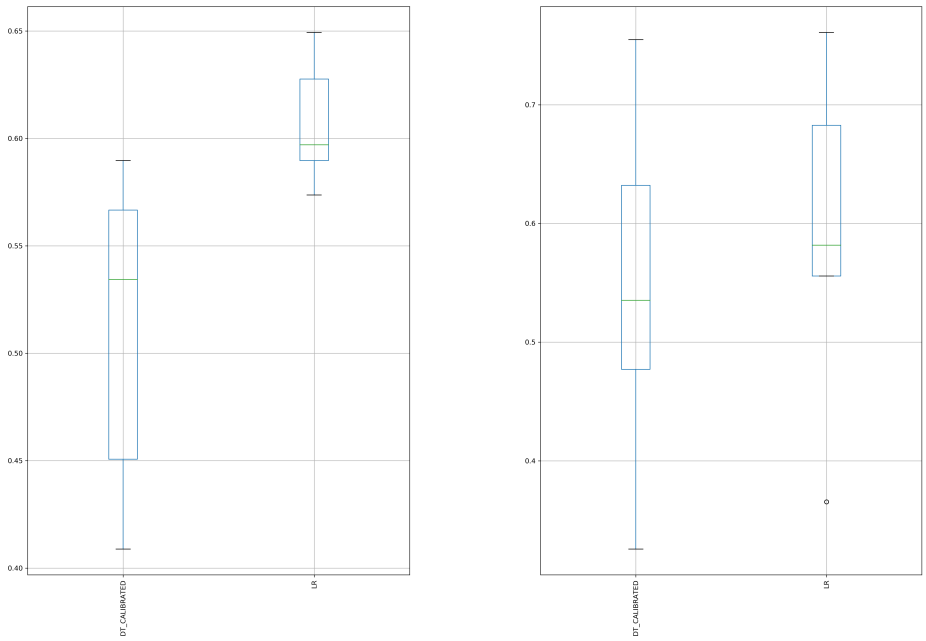
Also, AUC PR scores are better for logistic regression, both average and spread are better than calibrated decision tree for both training and testing sets.



(a) Training (b) Testing

Figure 12.18: Box Plots for CNS - MUSC AUC PR

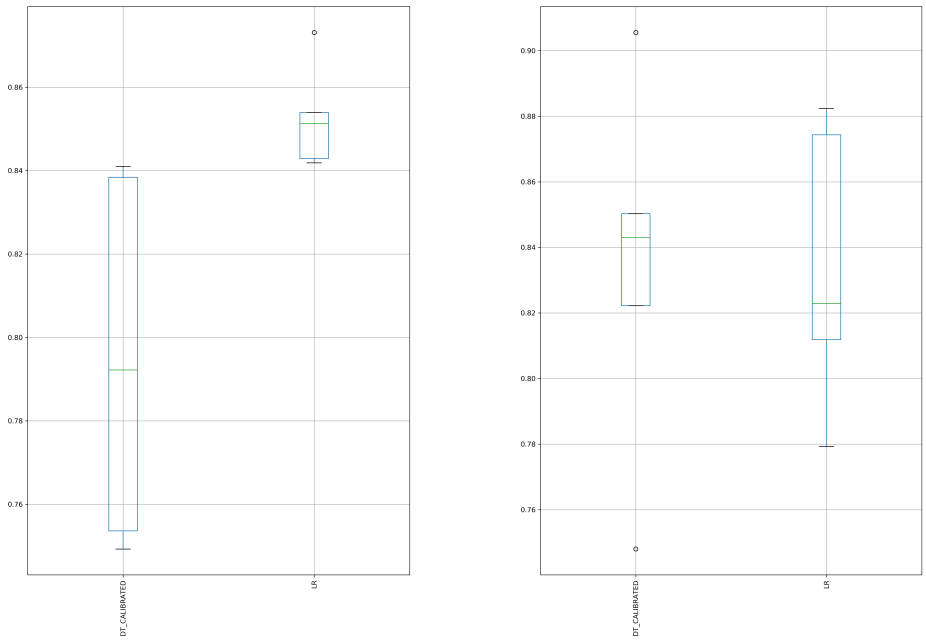
12.3.8. Observations For GU - MUSC. BSS scores are better for logistic regression, both average and spread are better than calibrated decision tree for both training and testing sets.



(a) Training (b) Testing

Figure 12.19: Box Plots for GU - MUSC Brier Skill Score

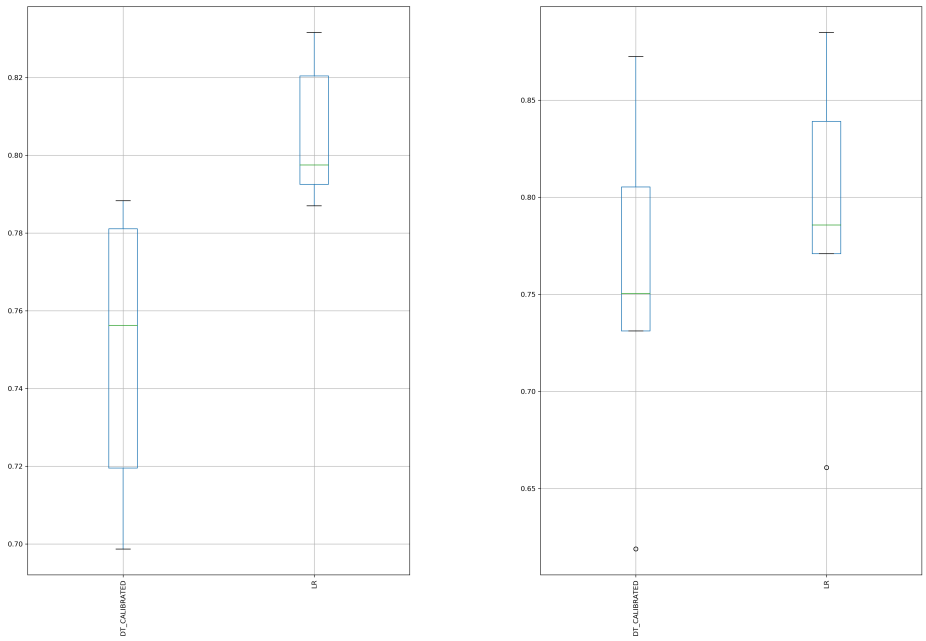
Also, AUC ROC scores are better for logistic regression, both average and spread are better than calibrated decision tree for both training and testing sets.



(a) Training (b) Testing

Figure 12.20: Box Plots for GU - MUSC AUC ROC

Also, AUC PR scores are better for logistic regression, both average and spread are better than calibrated decision tree for both training and testing sets.



(a) Training (b) Testing

Figure 12.21: Box Plots for GU - MUSC AUC PR

12.4. Interpretation. For All defects sheet, we get the following models with their respective configurations:

- Calibrated Decision Tree
- Calibrated Support Vector Machine
- Logistic Regression
- Cost Sensitive Logistic Regression

For CHD, CNS, GU, and MUSC sheets the models have a poor performance, this can be identified by the negative Brier Skill Score on all testing sets. Yet, we can identify a few important observations on the validation sets. Calibrated KNN and calibrated decision tree are performing better than all the other models for all these datasets. However, since the testing brier skill scores are all negative, we cannot pick them to perform predictions.
When mixing the birth defects, we were able to achieve a much better performance

compared to the performance reported on the single defect datasets. Selected models have almost similar performance, however, we can clearly see that normalized cost-sensitive logistic regression has always better average performance and minimal spread compared to the other models.

Chapter 13

Probability Threshold Moving

13.1. Problem Definition. Classification models derive their results from probability prediction techniques. Initially, the probability is computed and a number between 0 and 1 is produced, this number represents a specific percentage, for example in our case, a prediction of 0.5 means that this person has a 50% chance of getting a birth defect. Then, after deriving the probability score, if the number is greater than 0.5 the instance is classified as 1 or positive class, if the number is less than 0.5 then the instance is classified as 0 or negative. In imbalanced classification problems, where skewed data is classified, the 0.5 threshold is not necessarily the best threshold and can produce non-optimal results or poor performance. A technique called probability threshold moving can be applied to boost the produced results. There are several techniques to perform probability threshold moving, for instance, it can be derived from ROC and Precision-Recall curves.

13.2. Converting Probabilities to Class Labels. Machine learning algorithms derive class labels from given probabilities. First, a probability score is predicted, then a class label is derived from that probability score. The default threshold for the conversion process from probability score to a class label is performed by the following formula:

$$Prediction < 0.5 = Class\ 0 \tag{13.1}$$

$$Prediction \geq 0.5 = Class\ 1 \tag{13.2}$$

Although this formula works well for traditional classification problems, however, it can generate non-optimal results when the machine learning models are trained on imbalanced data. Therefore, the threshold moving technique can be used to optimize the classification process by tuning the traditional 0.5 thresholds

and by choosing a more reasonable number.

13.3. Threshold-Moving for Imbalanced Classification. Many techniques can be used to deal with imbalanced data classification. We already used several ones in previous sections, such as algorithm modification using cost-sensitive machine learning and sampling techniques such as oversampling and undersampling. In this section, we display a simpler technique to the ones used before and that can be combined with them to boost their performance. This technique is called a probability threshold moving.
Threshold moving can be applied in several ways; the first approach is by analyzing the ROC curve of the predicted probabilities and by tuning the threshold using AUC ROC results, although this is an effective technique; it is solely used when both classes are equally important. Since the positive class is more important for us, we use another technique that focuses more on the positive class. The approach is similar to the AUC ROC approach, however, it uses the precision-recall curve, and tuning is done over the AUC Precision-Recall metric. AUC Precision-Recall is the area under the precision-recall curve. Precision is defined as the number of true positives over the number of true positives plus the number of false positives. Recall is defined as the number of true positives over the number of true positives plus the number of false negatives. The positive class in our data is the minority, that is why AUC Precision-Recall is more suitable for our study.
The algorithm used is the following:

1. Fit Model on the Training Dataset
2. Predict Probabilities on the Test Dataset
3. For each threshold in Thresholds:
   (a) Convert probabilities to Class Labels using the threshold
   (b) Evaluate Class Labels
   (c) If Score is Better than Best Score. Adopt Threshold.
4. Use Adopted Threshold When Making Class Predictions on New Data

13.4. Optimal Threshold for Precision - Recall Curve. In this section, we illustrate the results generated by performing threshold moving on the best performing models.
We applied this algorithm over all the selected models to boost their performances and summarized the results in the tables for each dataset, for each table, we first

display the name of the model with the corresponding configuration, then we display the initial F2 score before performing threshold moving, then we display F2 score after performing threshold moving and we report the optimal threshold reported by the used algorithm:

13.4.1. Classification for All Defects. Below we show the results for all defects:

Model with Configuration	Initial F2 Score	F2 Score After	Thresho
Decision Tree - SMOTE TOMEK	91.97%	92.11%	0.0
Cost Sensitive Logistic Regression - Normalized	93.46%	96.35%	0.7
Logistic Regression - SMOTENC - Normalized	92.17%	95.86%	0.7
Logistic Regression - SMOTE TOMEK - Normalized	93.26%	95.86%	0.9

Table 13.1: Probability Threshold Moving Training Results - All Defects

As we can see, the F2 score was boosted for all the models
There is a 0.14% improvement for Decision Tree, 2.89% improvement for cost-sensitive logistic regression, 3.69% improvement for logistic regression with SMOTENC, and 2.6% improvement for logistic regression with SMOTE TOMEK.

13.4.2. Classification for CHD - CNS. Below we show the results for CHD - CNS:

Used Model	Initial F2 Score	F2 Score After	Threshold
Cost Sensitive Logistic Regression - Normalized	83%	89.74%	0.414

Table 13.2: Probability Threshold Moving Training Results - CHD - CNS

As we can see, F2 score was boosted for cost sensitive logistic regression by 6%.

13.4.3. Classification for CHD - GU. Below we show the results for CHD - GU
:

Used Model	Initial F2 Score	F2 Score After	Threshold
Cost Sensitive Logistic Regression - Normalized	84.83%	89.74%	0.745

Table 13.3: Probability Threshold Moving Training Results - CHD - GU

As we can see, the F2 score was boosted for cost-sensitive logistic regression by almost 5%.

13.4.4. Classification for CHD - MUSC. Below we show the results for CHD - MUSC:

Used Model	Initial F2 Score	F2 Score After	Threshold
Cost Sensitive Logistic Regression - Normalized	83.6%	81.52%	0.259

Table 13.4: Probability Threshold Moving Training Results - CHD - MUSC

We can see, F2 score was not boosted for cost-sensitive logistic regression in the case of CHD - MUSC.

13.4.5. Classification for CNS - GU. Below we show the results for CNS - GU:

Used Model	Initial F2 Score	F2 Score After	Threshold
Cost Sensitive Logistic Regression - Normalized	87%	85.24%	0.292

Table 13.5: Probability Threshold Moving Training Results - CNS - GU

We can see, F2 score was not boosted for cost-sensitive logistic regression in the case of CNS - GU.

13.4.6. Classification for CNS - MUSC. Below we show the results for CNS - MUSC:

Used Model	Initial F2 Score	F2 Score After	Threshold
Cost Sensitive Logistic Regression - Normalized	85.26%	86.42%	0.666

Table 13.6: Probability Threshold Moving Training Results - CNS - MUSC

We can see, F2 score was not boosted for cost-sensitive logistic regression in the case of CNS - MUSC.

13.4.7. Classification for GU - MUSC. Below we show the results for GU - MUSC:

Used Model	Initial F2 Score	F2 Score After	Threshold
Cost Sensitive Logistic Regression - Normalized	84.7%	88.61%	0.871

Table 13.7: Probability Threshold Moving Training Results - GU - MUSC

We can see, F2 score was boosted for cost-sensitive logistic regression by almost 4%.

13.5. Summary Probability Threshold Moving Chapter. In this chapter, we used a technique called probability threshold moving. This technique is used to boost the performance of classification results when performing classification over imbalanced datasets. We summarize our findings below:

- Using a Threshold of 0.746 boosted the performance of cost-sensitive logistic regression over the all defects sheet by 2.89%.
- Using a Threshold of 0.772 boosted the performance of logistic regression with SMOTENC over the all defects sheet by 3.69%.
- Using a Threshold of 0.955 boosted the performance of logistic regression with SMOTE TOMEK over the all defects sheet by 2.6%.
- Using a Threshold of 0.414 boosted the performance of cost-sensitive logistic regression over the CHD-CNS sheet by 6.74%.
- Using a Threshold of 0.745 boosted the performance of cost-sensitive logistic regression over the CHD-GU sheet by 4.91%.
- Using a Threshold of 0.871 boosted the performance of cost-sensitive logistic regression over the GU-MUSC sheet by 3.91%.
- Threshold moving on CHD-MUSC, CNS-GU, and CNS-MUSC sheets did not achieve any better performance.

From the list of configurations that we concluded in this chapter, we end up with the following used configurations for classification:
For All Defects: cost sensitive logistic regression with a Threshold of 0.746, logistic regression with SMOTENC with a Threshold of 0.772, and logistic regression with SMOTE TOMEK with a Threshold of 0.955.

For CHD-CNS: we use cost-sensitive logistic regression with a Threshold of 0.414.
For CHD-GU: we use cost-sensitive logistic regression with a Threshold of 0.745.
For GU-MUSC: we use cost-sensitive logistic regression with a Threshold of 0.871.
All remaining models use the default configurations because no improvement in their performance was recorded during the process of Threshold moving. These configurations are going to be used in chapter 14 and chapter 15 for all the classification results.

Chapter 14

Analysis of predictive models

14.1. Evaluation using traditional metrics. In this section, we evaluate the used models using traditional metrics. Binary classification can be evaluated using a set of metrics such as accuracy, precision, recall, and AUC ROC. Moreover, since we are dealing with an imbalanced classification problem, we need another set of important metrics which are Gmean and F2 score. The role of these metrics is to evaluate the performance of the model in predicting the positive class, which is more important than the negative class in our study. F2 score and Gmean represent a combination of precision and recall metrics, these two metrics are directly proportional to the number of true positives. Finally, we compare the ROC and Precision-Recall curves to display the goodness of the selected models.
In addition to the binary classification problem, we performed probability prediction to predict risk estimates rather than limiting the study to 0/1 classification. To evaluate the probability predictors, we used a metric called Brier Skill Score, described in the methodology.

To understand what is happening is the ROC curve and Precision-Recall curve, first, let's evaluate the ROC curve. ROC curves show that these models are performing well since all of them are away from the diagonal line. Whenever the curve is away from the diagonal, this means that the models have a better confidence level. Although the ROC curve shows that the models are performing well, but in our study, we are more interested in studying how well our models are performing with the positive class, that is why we have to look into the Precision-Recall curve. To evaluate the Precision-Recall curve, we follow the same strategy, whenever the curve is away from the diagonal line, this means that the model is performing well.

The x-axis of the ROC curve displays the false positive rate (fpr) which is the division of false positives over the sum of false-positive and true negatives. The y-axis displays the true positive rate (tpr) which is the division of the number of

true positives over the sum of true positive and false negatives.

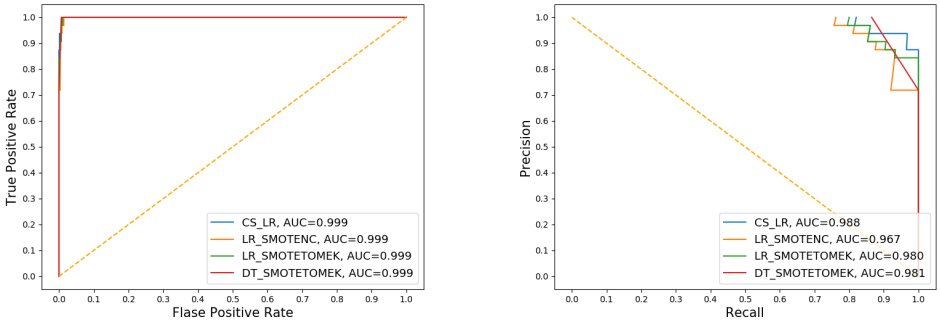
The x-axis of the Precision-Recall curve displays the recall rate and the y-axis displays the y-axis.

Here we provide a table with the best configuration for each pair of model / datasheet.

Data Sheet	Model	Used Scaling	Hyper Parameters	Treshold Moving [for classification]	Feature Selection
All Defects	Cost Sensitive Logistic Regression	Normalization	solver:liblinear, penalty:l1, class weight: 0: 1, 1: 10	0.746	No
	SMOTENC - Logistic Regression	Normalization	solver:liblinear, penalty:l1	0.772	No
	SMOTE Tomek - Logistic Regression	Normalization	solver:liblinear, penalty:l1	0.955	Better
	SMOTE Tomek - Decision Tree	None	criterion='gini', max_depth=5, min_samples_leaf=8, min_samples_split=5	None	Better
CHD - CNS	Cost Sensitive Logistic Regression	Normalization	solver:liblinear, penalty:l1, class weight: 0: 1, 1: 100	0.414	No
CHS - GU	Cost Sensitive Logistic Regression	Normalization	solver:liblinear, penalty:l1, class weight: 0: 1, 1: 100	0.745	Better
CHD - MUSC	Cost Sensitive Logistic Regression	Normalization	solver:liblinear, penalty:l1, class weight: 0: 1, 1: 100	None	Better
CNS - GU	Cost Sensitive Logistic Regression	Normalization	solver:liblinear, penalty:l1, class weight: 0: 1, 1: 100	None	No
CNS - MUSC	Cost Sensitive Logistic Regression	Normalization	solver:liblinear, penalty:l1, class weight: 0: 1, 1: 100	None	No
GU - MUSC	Cost Sensitive Logistic Regression	Normalization	solver:liblinear, penalty:l1, class weight: 0: 1, 1: 100	None	Better

Table 14.1: Best Configuration For Model / Data Sheet Pairs

14.1.1. All Defects. Below we display the ROC and Precision-Recall curves of the top-performing models for all defects.

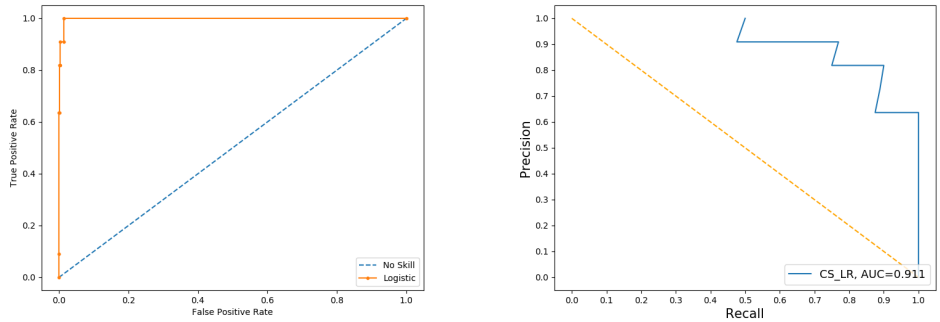


(a) Comparison Plot for ROC Curves (b) Comparison Plot for Precision Recall Curves

Figure 14.1: All Defects

We can see that the decision tree with SMOTE TOMEK is better than logistic regression with SMOTENC and logistic regression with SMOTE TOMEK, but cost-sensitive logistic regression is outperforming.

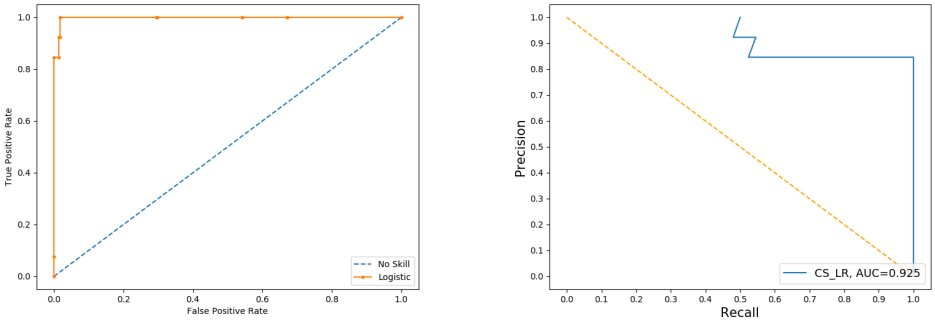
14.1.2. CHD - CNS. Below we display the ROC and Precision-Recall curves of the top performing models for CHD - CNS.



(a) Comparison Plot for ROC Curves (b) Comparison Plot for Precision Recall Curves

Figure 14.2: CHD - CNS

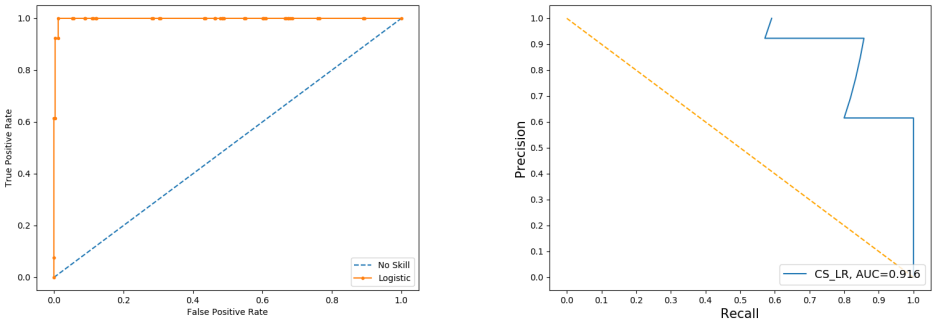
14.1.3. CHD - GU. Below we display the ROC and Precision-Recall curves of the top performing models for CHD - GU.



(a) Comparison Plot for ROC Curves (b) Comparison Plot for Precision Recall Curves

Figure 14.3: CHD - GU

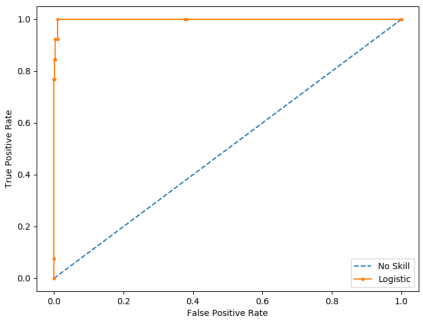
14.1.4. CHD - MUSC. Below we display the ROC and Precision-Recall curves of the top performing models for CHD - MUSC.



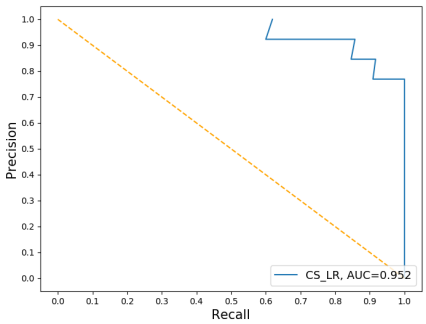
(a) Comparison Plot for ROC Curves (b) Comparison Plot for Precision Recall Curves

Figure 14.4: CHD - MUSC

14.1.5. CNS - GU. Below we display the ROC and Precision-Recall curves of the top performing models for CNS - GU.



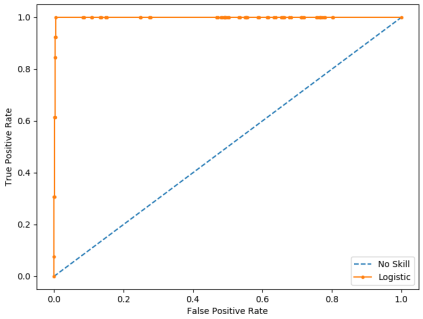
(a) Comparison Plot for ROC Curves



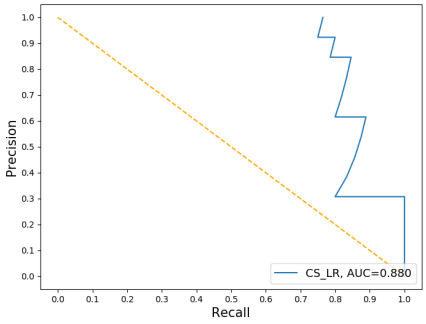
(b) Comparison Plot for Precision Recall Curves

Figure 14.5: CNS - GU

14.1.6. CNS - MUSC. Below we display the ROC and Precision-Recall curves of the top performing models for CNS - MUSC.



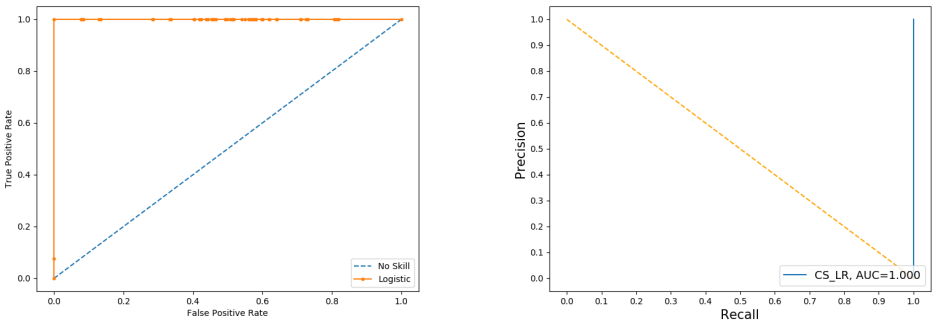
(a) Comparison Plot for ROC Curves



(b) Comparison Plot for Precision Recall Curves

Figure 14.6: CNS - MUSC

14.1.7. GU - MUSC. Below we display the ROC and Precision-Recall curves of the top performing models for GU - MUSC.



(a) Comparison Plot for ROC Curves (b) Comparison Plot for Precision Recall Curves

Figure 14.7: GU - MUSC

14.2. Ensuring the quality of risk estimates. In this section, we study the quality of risk estimates. Although some models can perform 0/1 binary classification, practitioners sometimes are more interested in seeing a risk estimate or a probability score that can guide them during the evaluation process. That's why we used several machine learning tools that can predict these risk estimates and we evaluated them using an empirical risk estimate plot.
To produce the empirical risk estimate plots, we first rank the estimated scores in descending order. We then group them into 10 bins based on their risk scores. The bottom 10% of the instances that have the least risk are grouped into a single bin. The second bin includes women who have a risk score between 10% and 20% and so on. For each bin, we compute the mean empirical risk, which is the fraction of the pregnant women from that bin who actually have a birth defect. We then plot a curve where values on the X-axis denote the upper percentile limit of a bin and values on the Y-axis correspond to the mean empirical risk of the corresponding bins. Monotonically increasing empirical risk curves denote that the algorithm is producing accurate risk estimates. Non-monotonic curves mean that the used algorithm is producing non-accurate results and that women with high-risk scores according to that specific algorithm do not have birth defects.

14.2.1. All Defects. In our case, we display the empirical risk estimate curves for all four selected models: logistic regression, cost-sensitive logistic regression, calibrated support vector machines, and calibrated decision tree.
The graphs show an almost straight line parallel to the x-axis for the first nine bins, then we have an ascending line. This behavior shows that the predictions are accurate and that for low predicted probabilities i.e. 0% 10% a very minimal number of mistakes was done by the models. This can be observed by the

monotonically non-decreasing graph. Also, we can observe that all four models are overlapped, this means that all of them have good performance.

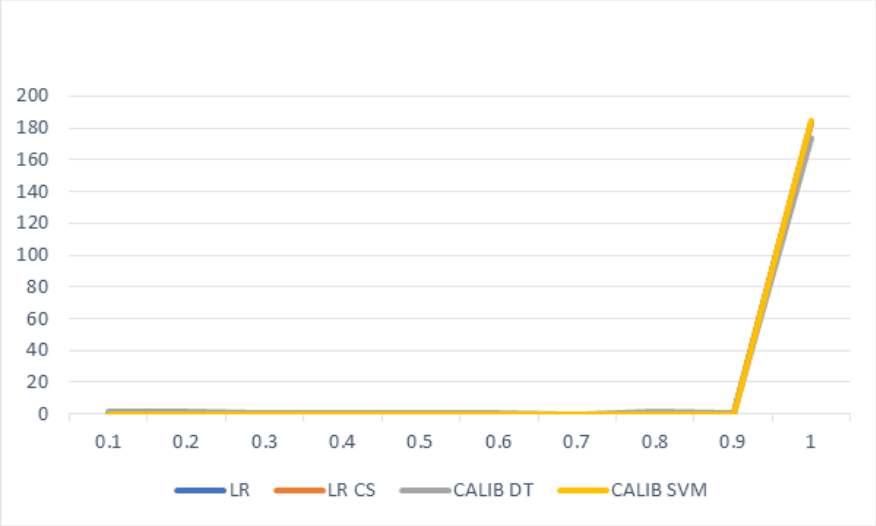


Figure 14.8: Comparison between Emperical Risk Estimates for All Defects over all the models

14.2.2. CHD - CNS. In our case, we display the empirical risk estimate curve for the best-selected model which is cost-sensitive logistic regression.
The graphs show an almost straight line parallel to the x-axis for the first nine bins, then we have an ascending line. This behavior shows that the predictions are accurate and that for low predicted probabilities i.e. 0% 10% a very minimal number of mistakes was done by the models. This can be observed by the monotonically non-decreasing graph.

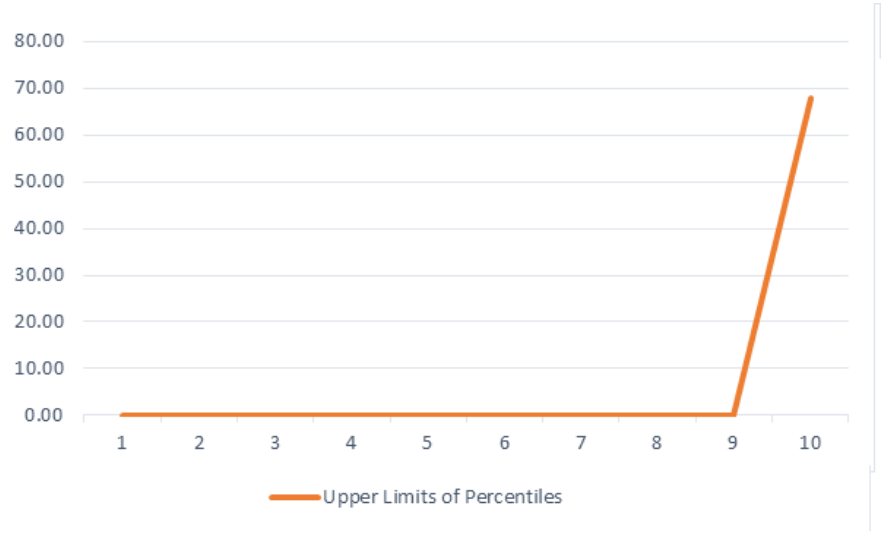


Figure 14.9: Emperical Risk Estimates for CHD - CNS

14.2.3. CHD - GU. In our case, we display the empirical risk estimate curve for the best-selected model which is cost-sensitive logistic regression.
The graphs show an almost straight line parallel to the x-axis for the first nine bins, then we have an ascending line. This behavior shows that the predictions are accurate and that for low predicted probabilities i.e. 0% 10% a very minimal number of mistakes was done by the models. This can be observed by the monotonically non-decreasing graph.

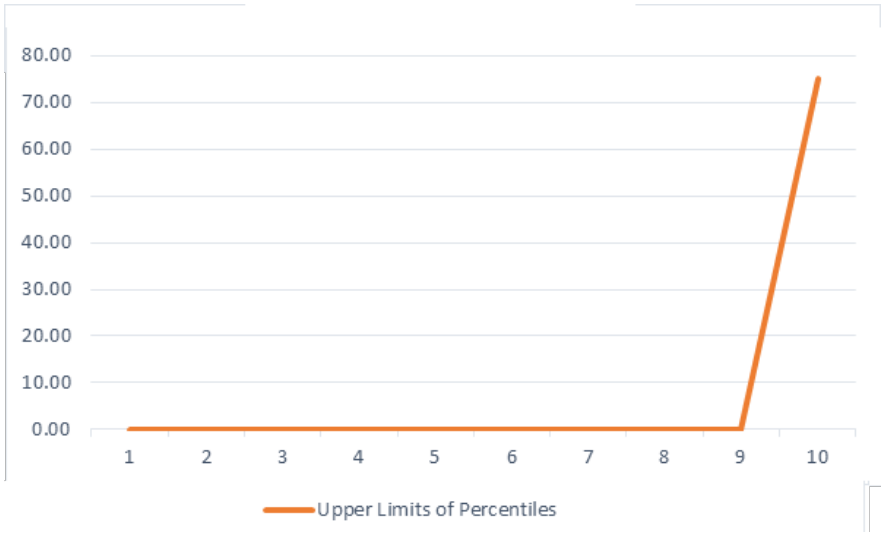


Figure 14.10: Emperical Risk Estimates for CHD - GU

14.2.4. CHD - MUSC. In our case, we display the empirical risk estimate curve for the best selected model which is cost-sensitive logistic regression.
The graphs show an almost straight line parallel to the x-axis for the first nine bins, then we have an ascending line. This behavior shows that the predictions are accurate and that for low predicted probabilities i.e. 0% 10% a very minimal number of mistakes was done by the models. This can be observed by the monotonically non-decreasing graph.

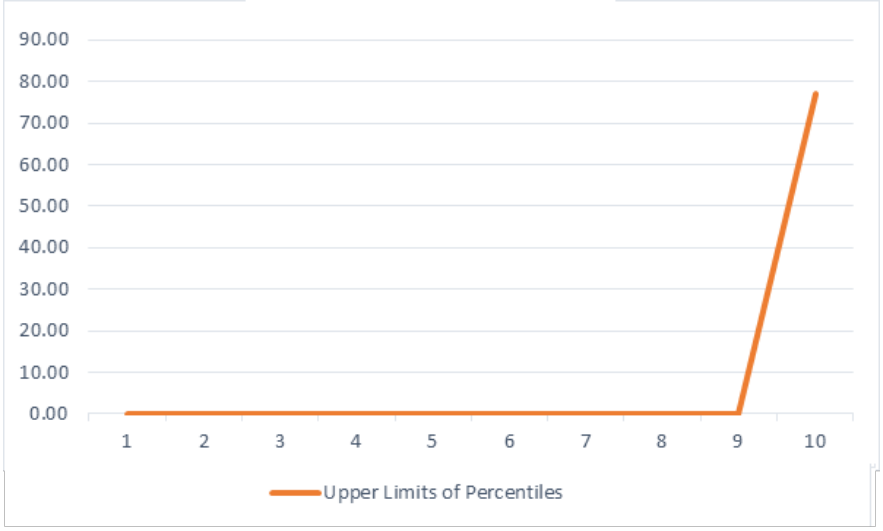


Figure 14.11: Emperical Risk Estimates for CHD - MUSC

14.2.5. CNS - GU. In our case, we display the empirical risk estimate curve for the best selected model which is cost sensitive logistic regression.
The graphs show an almost straight line parallel to the x-axis for the first nine bins, then we have an ascending line. This behavior shows that the predictions are accurate and that for low predicted probabilities i.e. 0% 10% a very minimal number of mistakes was done by the models. This can be observed by the monotonically non-decreasing graph.

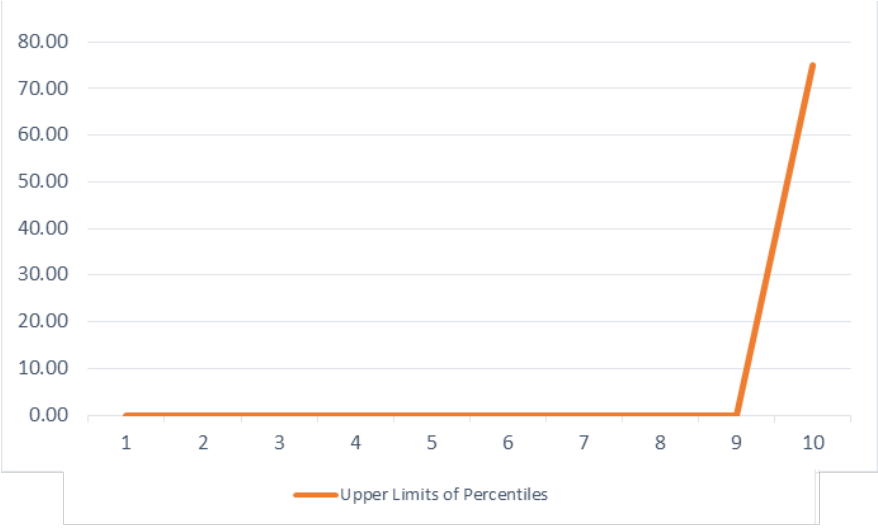


Figure 14.12: Emperical Risk Estimates for CNS - GU

14.2.6. CNS - MUSC. In our case, we display the empirical risk estimate curve for the best selected model which is cost sensitive logistic regression.
The graphs show an almost straight line parallel to the x-axis for the first nine bins, then we have an ascending line. This behavior shows that the predictions are accurate and that for low predicted probabilities i.e. 0% 10% a very minimal number of mistakes was done by the models. This can be observed by the monotonically non-decreasing graph.

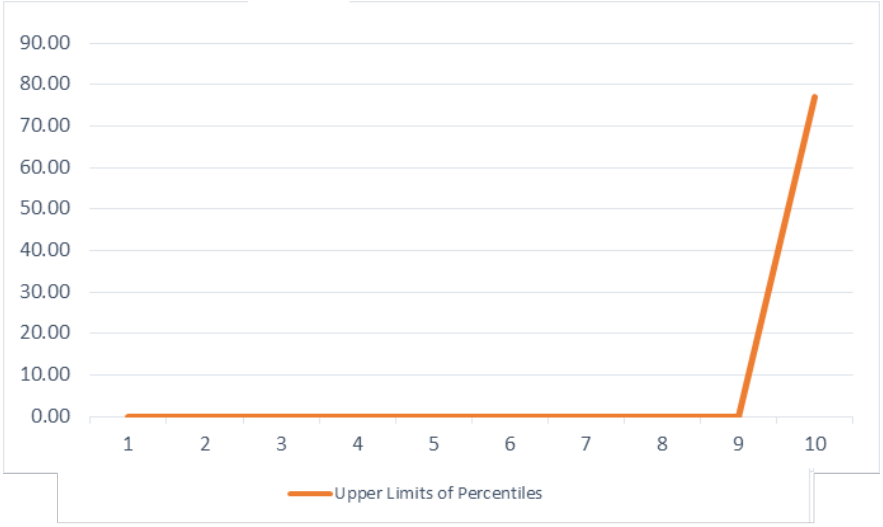


Figure 14.13: Emperical Risk Estimates for CNS - MUSC

14.2.7. GU - MUSC. In our case, we display the empirical risk estimate curve for the best selected model which is cost sensitive logistic regression.
The graphs show an almost straight line parallel to the x-axis for the first nine bins, then we have an ascending line. This behavior shows that the predictions are accurate and that for low predicted probabilities i.e. 0% 10% a very minimal number of mistakes was done by the models. This can be observed by the monotonically non-decreasing graph.

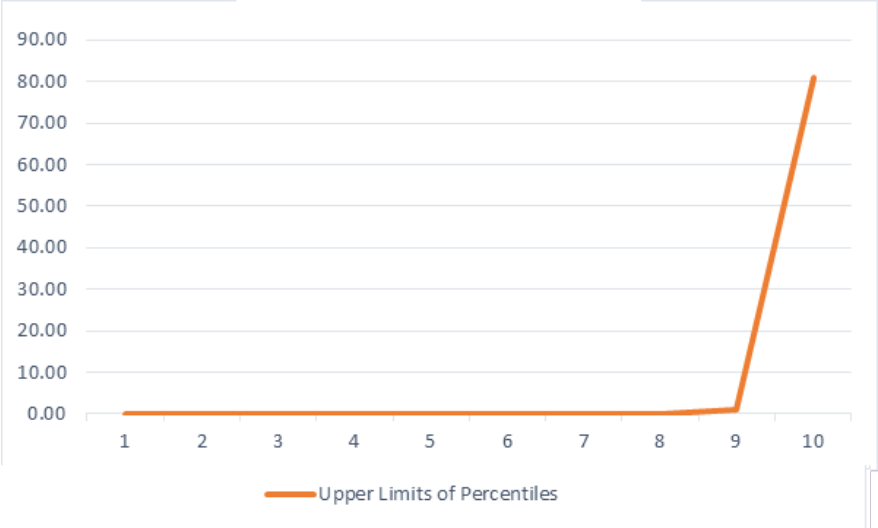
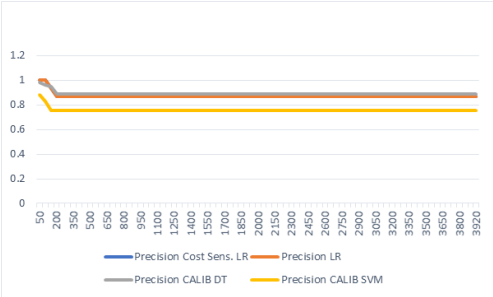


Figure 14.14: Emperical Risk Estimates for GU - MUSC

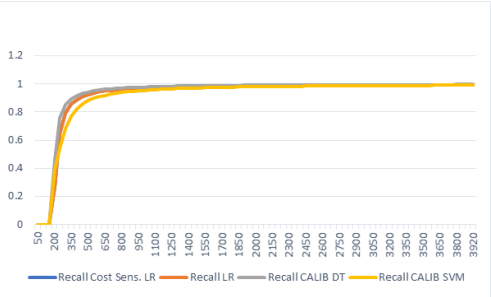
14.3. Comparative evaluation of risk estimates. In this section, we perform a comparative evaluation of risk estimates. To perform the evaluation, we focus on two very important metrics: precision and recall. These two metrics are essential to our study because they help us evaluate the performance of the models in predicting the positive. Although it is important to display precision and recall values for the overall performance of the model, sometimes, clinicians are more interested in the precision and recall at the top k. Precision and recall at top k is a concept used to rank the estimations and to evaluate the risk estimates based on their ranges, where k is a fixed threshold specified by the clinicians. This procedure, helps to identify pregnant women with higher risk of having birth defects, and thus, this process will allow to identify which women to diagnose in case of limited places in the emergency room. We can plot recall at top k and precision at top k curves to have a better understanding of the risk estimates produced at top k. The x-axis of both Top k precision and recall curves displays the number of instances. The y-axis of Top k Precision displays the precision value. The y-axis of Top k Recall displays the recall value. For All defects sheet k is set to 200 and it is set to 800 for pairs of defect comparison.

Top k values represent women with highest risk of developing birth defects, and that's why it is important to give them a higher priority.

The precision and recall vary with the value of k as is observed in the figures below.



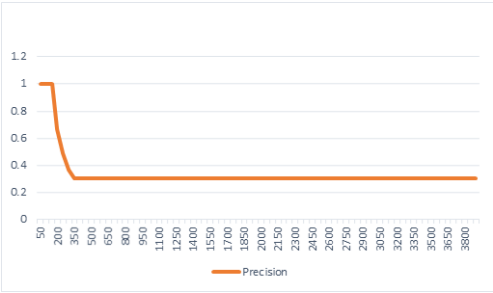
(a) Precision at top k



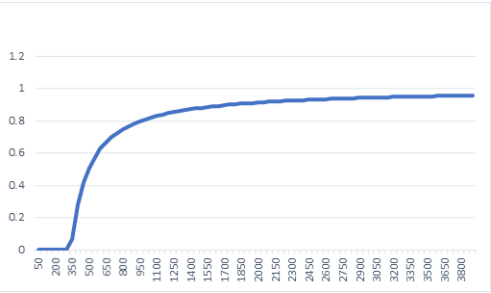
(b) Recall at top k

We can observe that precision for logistic regression and cost-sensitive logistic regression are aligned and are the best for k less than 180; however, the calibrated decision tree has a better performance for k less than 200. Calibrated support vector machines performed well, but the other two models performed better. Also, logistic regression and cost-sensitive logistic regression's recall curves are aligned. Calibrated decision tree performs slightly better than both of them. Calibrated support vector machines are the worst.

14.3.1. CHD - CNS. In the precision at top k and recall at top k curves for CHD - CNS, we can observe that the best precision for cost sensitive logistic regression is the best for k less than 100, and the best-recorded recall is recorded for k greater than 800.



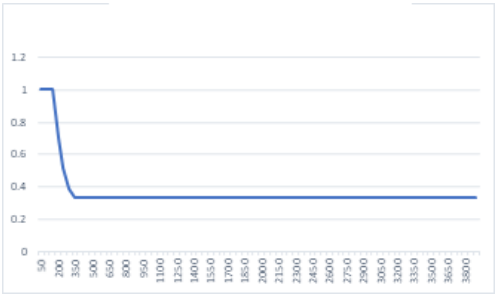
(c) Precision at top k



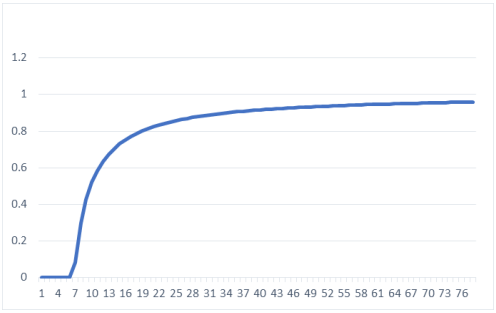
(d) Recall at top k

14.3.2. CHD - GU. In the precision at top k and recall at top k curves for CHD - GU, we can observe that the best precision for cost sensitive logistic regression

is the best for k less than 15, and the best-recorded recall is recorded for k greater than 200.

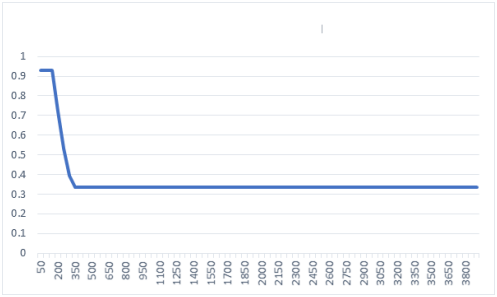


(e) Precision at top k

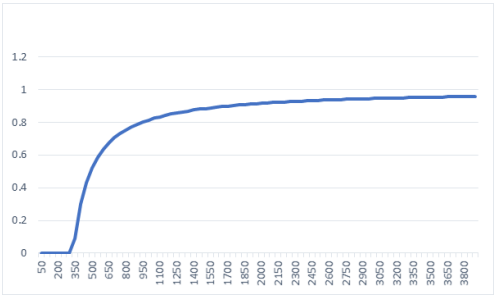


(f) Recall at top k

14.3.3. CHD - MUSC. In the precision at top k and recall at top k curves for CHD - MUSC, we can observe that the best precision for cost sensitive logistic regression is the best for k less than 200 and the best recorded recall is recorded for k greater than 800.

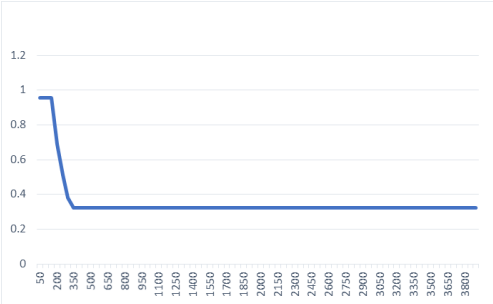


(g) Precision at top k

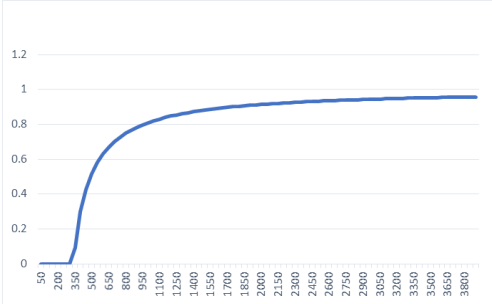


(h) Recall at top k

14.3.4. CNS - GU. In the precision at top k and recall at top k curves for CNS - GU, we can observe that the best precision for cost sensitive logistic regression is the best for k less than 200 and the best recorded recall is recorded for k greater than 800.

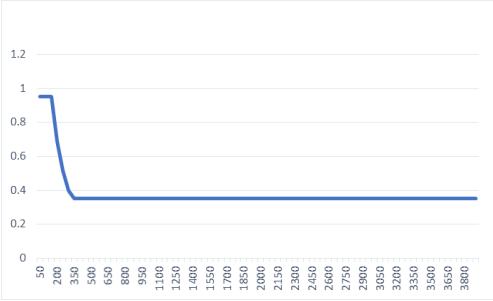


(i) Precision at top k

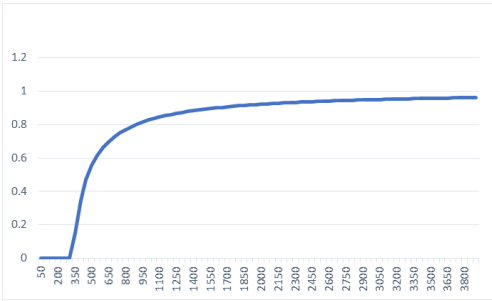


(j) Recall at top k

14.3.5. CNS - MUSC. In the precision at top k and recall at top k curves for CNS - MUSC, we can observe that the best precision for cost sensitive logistic regression is the best for k less than 200 and the best recorded recall is recorded for k greater than 800.

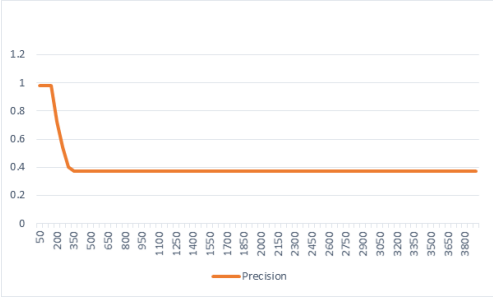


(k) Precision at top k

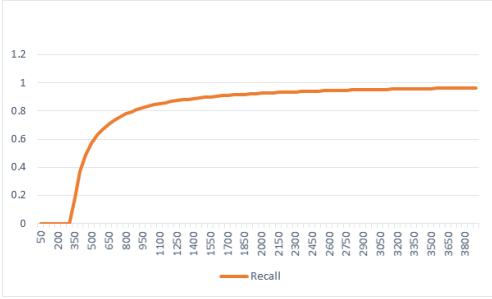


(l) Recall at top k

14.3.6. GU - MUSC. In the precision at top k and recall at top k curves for GU - MUSC, we can observe that the best precision for cost sensitive logistic regression is the best for k less than 200 and the best recorded recall is recorded for k greater than 800.



(m) Precision at top k



(n) Recall at top k

14.4. Running Best Models on Clustered Data. In this section, we run the best-selected models with their respective configurations on clustered data. Running the best models on the clustered data may boost their performance because the used data will have some common patterns, which will allow the model to focus on the most important features and to neglect these patterns.
First, the data used is generated by Gower's distance clustering technique discussed in the clustering chapter. Two clusters were produced for each dataset.

Below we will display a comparison between the recorded results. For each model, we displayed the training results in the first table, and the testing results in the second table. Also, for each table, we represent the results on all the data then we display the results for each cluster.

14.4.1. Results for All Defects. Below we display the results for All Defects:

	F2	**Gmean**	**AUC ROC**	**Accurac**
DT SMOTETOMEK	0.9197(0.0071)	0.9676(0.0023)	0.9683(0.0024)	0.9882(0.0
cluster 1 DT SMOTETOMEK	0.877(0.0134)	0.9449(0.0065)	0.9474(0.0061)	0.9859(0.0
cluster 2 DT SMOTETOMEK	0.8342(0.039)	0.919(0.0228)	0.9242(0.0213)	0.9702(0.0
LR CS	0.9346(0.0055)	0.9896(0.0029)	0.9897(0.0029)	0.984(0.00
cluster 1 LR CS	0.9293(0.0107)	0.989(0.0031)	0.9892(0.0031)	0.9853(0.0
cluster 2 LR CS	0.9318(0.0095)	0.986(0.0035)	0.9862(0.0035)	0.977(0.00
LR SMOTETOMEK	0.9326(0.0058)	0.9856(0.0043)	0.9857(0.0042)	0.985(0.00
cluster 1 LR SMOTENC	0.918(0.0137)	0.9811(0.0052)	0.9816(0.005)	0.9848(0.0
cluster 1 LR SMOTETOMEK	0.9232(0.0163)	0.9864(0.0071)	0.9866(0.007)	0.9844(0.0
LR SMOTENC	0.9217(0.0057)	0.9797(0.0044)	0.98(0.0043)	0.9838(0.0
cluster 2 LR SMOTENC	0.9128(0.023)	0.97(0.013)	0.971(0.0123)	0.9774(0.0
cluster 2 LR SMOTETOMEK	0.9258(0.014)	0.9805(0.0073)	0.9809(0.007)	0.9776(0.0

Table 14.2: Comparative Results on Training Set For Best Modeling Tools

	F2	Gmean	AUC ROC	Accurac
DT SMOTETOMEK	0.9164(0.0221)	0.9674(0.0104)	0.9676(0.0102)	0.9873(0.0
cluster 1 DT SMOTETOMEK	0.8867(0.0831)	0.9493(0.0496)	0.9511(0.0473)	0.9872(0.0
cluster 2 DT SMOTETOMEK	0.8662(0.1103)	0.9368(0.0781)	0.9409(0.0698)	0.9757(0.0
LR CS	0.936(0.0179)	0.936(0.0179)	0.9894(0.0073)	0.9847(0.0
cluster 1 LR CS	0.9327(0.0278)	0.9888(0.0108)	0.9888(0.0108)	0.9865(0.0
cluster 2 LR CS	0.9271(0.0462)	0.9812(0.0163)	0.9813(0.0161)	0.9775(0.0
LR SMOTENC	0.9338(0.0157)	0.987(0.0074)	0.987(0.0074)	0.985(0.003
cluster 1 LR SMOTENC	0.9059(0.029)	0.9746(0.0179)	0.9748(0.0176)	0.984(0.004
cluster 2 LR SMOTENC	0.921(0.0472)	0.9749(0.0188)	0.9752(0.0186)	0.9784(0.0
LR SMOTETOMEK	0.9296(0.0181)	0.9843(0.0078)	0.9843(0.0078)	0.9847(0.0
cluster 1 LR SMOTETOMEK	0.9242(0.0338)	0.984(0.0136)	0.9841(0.0135)	0.9858(0.0
cluster 2 LR SMOTETOMEK	0.9318(0.0433)	0.9821(0.0163)	0.9823(0.0161)	0.9793(0.0

Table 14.3: Comparative Results on Testing Set For Best Modeling Tools

From the tables 14.2 and 14.3. We can see that the results for both training and testing did not improve when we performing modeling on clustered data, instead there is a drop in the performance of all algorithms.

14.4.2. Results for CHD - CNS. Below we display the results for CHD - CNS:

CHD CNS	F2	Gmean	AUC ROC	Accuracy
LR CS	0.8483(0.0197)	0.9757(0.0116)	0.9763(0.0112)	0.9846(0.0011)
LR CS - Cluster 1	0.8082(0.0323)	0.9642(0.0196)	0.9675(0.0173)	0.985(0.0017)
LR CS - Cluster 2	0.7765(0.037)	0.9271(0.021)	0.9378(0.0143)	0.9759(0.0034)

Table 14.4: Comparing Modeling Results on Clustered Data - Training

CHD CNS	F2	Gmean	AUC ROC	Accuracy
LR CS	0.8318(0.0361)	0.9771(0.0202)	0.9773(0.0199)	0.984(0.0038)
LR CS - Cluster 1	0.7915(0.0549)	0.9564(0.052)	0.9581(0.0494)	0.9847(0.0016)
LR CS - Cluster 2	0.7709(0.107)	0.9426(0.0961)	0.9475(0.0853)	0.9738(0.0085)

Table 14.5: Comparing Modeling Results on Clustered Data - Testing

From tables 14.4 and 14.5 we can see that training and testing results didn't improve for all recorded metrics on the clustered data.

14.4.3. Results for CHD - GU. Below we display the results for CHD - GU:

CHD GU	F2	Gmean	AUC ROC	Accuracy
LR CS	0.8483(0.0197	0.9757(0.0116)	0.9763(0.0112)	0.9846(0.0011)
LR CS - Cluster 1	0.8093(0.022)	0.9694(0.0102)	0.9711(0.0093)	0.984(0.0016)
LR CS - Cluster 2	0.9112(0.0178)	0.9879(0.0086)	0.9886(0.0074)	0.9863(0.0025)

Table 14.6: Comparing Modeling Results on Clustered Data - Training

CHD GU	F2	Gmean	AUC ROC	Accuracy
LR CS	0.8704(0.0198)	0.9925(0.0013)	0.9925(0.0013)	0.9853(0.0025)
LR CS - Cluster 1	0.8079(0.0616)	0.968(0.0318)	0.9686(0.031)	0.984(0.0074)
LR CS - Cluster 2	0.9266(0.0274)	0.9937(0.0028)	0.9938(0.0027)	0.9879(0.0053)

Table 14.7: Comparing Modeling Results on Clustered Data - Testing

From tables 14.6 and 14.7 we can see that the results didn't improve for first cluster, but there is an improvement in the performance on the second cluster for F2 score by 7% on the training set and 5% on the testing set.

14.4.4. Results for CHD - MUSC. Below we display the results for CHD - MUSC:

CHD MUSC	F2	Gmean	AUC ROC	Accuracy
LR CS	0.8363(0.0214)	0.9718(0.0115)	0.9729(0.0109)	0.9831(0.0015)
LR CS - Cluster 1	0.7929(0.0304)	0.9527(0.0186)	0.9563(0.0165)	0.9828(0.0016)
LR CS - Cluster 2	0.848(0.0276)	0.9668(0.0146)	0.9695(0.0124)	0.9784(0.0033)

Table 14.8: Comparing Modeling Results on Clustered Data - Training

CHD MUSC	F2	Gmean	AUC ROC	Accuracy
LR CS	0.8506(0.0488)	0.9788(0.0196)	0.9789(0.0194)	0.9843(0.0038)
LR CS - Cluster 1	0.7944(0.0572)	0.9495(0.0254)	0.9504(0.0249)	0.984(0.0039)
LR CS - Cluster 2	0.8553(0.0877)	0.9727(0.041)	0.9733(0.0398)	0.9795(0.0096)

Table 14.9: Comparing Modeling Results on Clustered Data - Testing

From tables 14.8 and 14.9 we can see that training and testing results didn't improve for all recorded metrics on the clustered data.

14.4.5. Results for CNS - GU. Below we display the results for CNS - GU:

CNS GU	F2	Gmean	AUC ROC	Accuracy
LR CS	0.8743(0.0081)	0.99(0.0028)	0.9901(0.0027)	0.9859(0.0011)
LR CS - Cluster 1	0.826(0.0176)	0.9838(0.0107)	0.9845(0.0099)	0.9851(0.0007)
LR CS - Cluster 2	0.8516(0.0258)	0.9681(0.014)	0.97(0.0127)	0.9761(0.0027)

Table 14.10: Comparing Modeling Results on Clustered Data - Training

CNS GU	F2	Gmean	AUC ROC	Accuracy
LR CS	0.8616(0.0458)	0.9797(0.0193)	0.9799(0.0191)	0.9861(0.0038)
LR CS - Cluster 1	0.8255(0.0606)	0.9801(0.0276)	0.9805(0.0268)	0.9858(0.0064)
LR CS - Cluster 2	0.8509(0.0665)	0.9605(0.04)	0.9614(0.0389)	0.9787(0.0078)

Table 14.11: Comparing Modeling Results on Clustered Data - Testing

From tables 14.10 and 14.11 we can see that training and testing results didn't improve for all recorded metrics on the clustered data.

14.4.6. Results for CNS - MUSC. Below we display the results for CNS - MUSC:

CNS MUSC	F2	Gmean	AUC ROC	Accuracy
LR CS	0.8526(0.0154)	0.9787(0.0069)	0.9792(0.0066)	0.9843(0.0011)
LR CS - Cluster 1	0.8324(0.0098)	0.9792(0.0083)	0.9803(0.0076)	0.985(0.0011)
LR CS - Cluster 2	0.8386(0.0321)	0.9665(0.0201)	0.9691(0.0177)	0.9755(0.0029)

Table 14.12: Comparing Modeling Results on Clustered Data - Training

CNS MUSC	F2	Gmean	AUC ROC	Accuracy
LR CS	0.8468(0.0531)	0.9726(0.0306)	0.9731(0.0299)	0.9848(0.0037)
LR CS - Cluster 1	0.831(0.0339)	0.981(0.0239)	0.9813(0.0233)	0.9847(0.0046)
LR CS - Cluster 2	0.8346(0.0447)	0.9723(0.0324)	0.9728(0.0315)	0.974(0.0053)

Table 14.13: Comparing Modeling Results on Clustered Data - Testing

From tables 14.12 and 14.13 we can see that training and testing results didn't improve for all recorded metrics on the clustered data.

14.4.7. Results for GU - MUSC. Below we display the results for GU - MUSC:

GU MUSC	F2	Gmean	AUC ROC	Accuracy
LR CS	0.8472(0.0109)	0.9692(0.0068)	0.9703(0.0065)	0.9842(0.0007)
LR CS - Cluster 1	0.831(0.0208)	0.9792(0.0087)	0.9802(0.0081)	0.9847(0.0017)
LR CS - Cluster 2	0.8649(0.0271)	0.9669(0.0193)	0.9692(0.0172)	0.9778(0.0019)

Table 14.14: Comparing Modeling Results on Clustered Data - Training

GU MUSC	F2	Gmean	AUC ROC	Accuracy
LR CS	0.8487(0.0363)	0.9731(0.0276)	0.9735(0.0269)	0.9841(0.0031)
LR CS - Cluster 1	0.8122(0.0429)	0.9695(0.0296)	0.97(0.0289)	0.984(0.0051)
LR CS - Cluster 2	0.8599(0.0299)	0.9622(0.0313)	0.9631(0.0301)	0.9787(0.0137)

Table 14.15: Comparing Modeling Results on Clustered Data - Testing

From tables 14.14 and 14.15 we can see that training and testing results didn't improve for all recorded metrics on the clustered data.

14.5. How to Incorporate Clustering into the Prediction Process. To implement the findings of this chapter, we can do the following during the prediction process:

1. Get Patient data.
2. Map the data to a cluster
3. If the patient belongs to a cluster that is known to have improved, then we use the model trained on this cluster, otherwise, we use the model trained on all data.

Chapter 15

Interpreting Classifier Output

After analyzing the performance of the models and their performance in the previous chapter, now we focus on interpreting the produced results. The goal of this chapter is to identify patterns of mistakes, study the performance of the best performing models after integrating features that are identified in the feature selection process, identify features that have a high impact on the prediction process using an interpretability tool called SHAP.

15.1. Integrating Feature Selection. In this section, we use selected features from the Feature Selection chapter and we run our best performing models on these features only. This process may improve the performance of the models because only important features are tested.
Below we will display a comparative table, showing the improvements in the performance of the selected models. From the produced results we can see that the performance of all models is almost stable after integrating selected features in the feature selection process and some of them have slight improvement. This is a very important finding because this may allow early prediction for the defects or prediction with fewer features, also we can see that feature selection allows early prediction for the birth defects because we can predict birth defects without the integration of post window of risk air pollution features. Using fewer features in the prediction process allows the user to predict birth defects without collecting too much data.
First, we will display the training results and then the testing results.

15.1.1. Results for All Defects. Below we will show the results for All Defects

	F2	Gmean	AUC ROC	Accuracy
DT_SMOTETOMEK	0.9197(0.0071)	0.9676(0.0023)	0.9683(0.0024)	0.9882(0.0019)
DT_SMOTETOMEK - FS	0.9217(0.0094)	0.9718(0.0033)	0.9725(0.0031)	0.9871(0.0018)
LR_CS	0.9346(0.0055)	0.9896(0.0029)	0.9897(0.0029)	0.984(0.0012)
LR_CS - FS	0.9335(0.0058)	0.9882(0.0027)	0.9883(0.0027)	0.9843(0.0008)
LR_SMOTENC	0.9217(0.0057)	0.9797(0.0044)	0.98(0.0043)	0.9838(0.0019)
LR_SMOTENC - FS	0.9275(0.0009)	0.9825(0.0005)	0.9827(0.0005)	0.9845(0.0004)
LR_SMOTETOMEK	0.9326(0.0058)	0.9856(0.0043)	0.9857(0.0042)	0.985(0.0013)
LR_SMOTETOMEK - FS	0.9351(0.0046)	0.9895(0.0022)	0.9896(0.0022)	0.9842(0.0008)

Table 15.1: Comparing Modeling Results Before and After Integrating Feature Selection - All Defects - Training

	F2	Gmean	AUC ROC	Accuracy
DT_SMOTETOMEK	0.9164(0.0221)	0.9674(0.0104)	0.9676(0.0102)	0.9873(0.0037)
DT_SMOTETOMEK - FS	0.9375(0.0157)	0.9811(0.0093)	0.9811(0.0092)	0.9885(0.0029)
LR_CS	0.936(0.0179)	0.936(0.0179)	0.9894(0.0073)	0.9847(0.003)
LR_CS - FS	0.9319(0.0212)	0.9867(0.0088)	0.9867(0.0088)	0.9845(0.0034)
LR_SMOTENC	0.9338(0.0157)	0.987(0.0074)	0.987(0.0074)	0.985(0.0033)
LR_SMOTENC - FS	0.9305(0.0111)	0.9844(0.007)	0.9844(0.007)	0.985(0.0021)
LR_SMOTETOMEK	0.9296(0.0181)	0.9843(0.0078)	0.9843(0.0078)	0.9847(0.0035)
LR_SMOTETOMEK - FS	0.9351(0.0182)	0.9893(0.0072)	0.9893(0.0072)	0.9845(0.0032)

Table 15.2: Comparing Modeling Results Before and After Integrating Feature Selection - All Defects - Testing

We can see that the performance of Decision Tree was improved by 2% for both testing and training. For other models, the performance did not have any notable improvement. However, we have to mention that the performance did not deteriorate, this means that the dropped features were not too important for the modeling process.

15.1.2. Results for CHD - CNS. Below we will show the results for CHD - CNS

CHD_CNS	F2	Gmean	AUC ROC	Accuracy
LR_CS	0.8483(0.0197)	0.9757(0.0116)	0.9763(0.0112)	0.9846(0.0011)
LR_CS - FS	0.839(0.011)	0.984(0.0045)	0.9843(0.0043)	0.9836(0.0011)

Table 15.3: Modeling Results Before and After Integrating Feature Selection - CHD - CNS - Training

CHD_CNS	F2	Gmean	AUC ROC	Accuracy
LR_CS	0.8318(0.0361)	0.9771(0.0202)	0.9773(0.0199)	0.984(0.0038)
LR_CS - FS	0.8384(0.043)	0.984(0.0175)	0.9842(0.0173)	0.9837(0.0046)

Table 15.4: Modeling Results Before and After Integrating Feature Selection - CHD - CNS - Testing

From the recorded results of F2 score, Gmean, AUC ROC, and Accuracy on both training and testing sets, we can see that the performance of the model is better when running cost-sensitive logistic regression without introducing feature selection.

15.1.3. Results for CHD - GU. Below we will show the results for CHD - GU

CHD_GU	F2	Gmean	AUC ROC	Accuracy
LR_CS	0.8483(0.0197	0.9757(0.0116)	0.9763(0.0112)	0.9846(0.0011)
LR_CS - FS	0.8557(0.0067)	0.9871(0.0041)	0.9873(0.0039)	0.9837(0.001)

Table 15.5: Modeling Results Before and After Integrating Feature Selection - CHD - GU - Training

CHD_GU	F2	Gmean	AUC ROC	Accuracy
LR_CS	0.8406(0.0224)	0.972(0.029)	0.9725(0.0282)	0.9845(0.0023)
LR_CS - FS	0.8515(0.0409)	0.9851(0.0155)	0.9852(0.0153)	0.9838(0.0042)

Table 15.6: Modeling Results Before and After Integrating Feature Selection - CHD - GU - Testing

From the recorded results of F2 score, Gmean, AUC ROC, and Accuracy on both training and testing sets, we can see that the performance of the model is better when introducing feature selection.

15.1.4. Results for CHD - MUSC. Below we will show the results for CHD - MUSC

CHD_MUSC	F2	Gmean	AUC ROC	Accuracy
LR_CS	0.8363(0.0214)	0.9718(0.0115)	0.9729(0.0109)	0.9831(0.0015)
LR_CS - FS	0.8546(0.0111)	0.9844(0.0046)	0.9847(0.0044)	0.9837(0.0011)

Table 15.7: Modeling Results Before and After Integrating Feature Selection - CHD - MUSC - Training

CHD_MUSC	F2	Gmean	AUC ROC	Accuracy
LR_CS	0.8506(0.0488)	0.9788(0.0196)	0.9789(0.0194)	0.9843(0.0038)
LR_CS - FS	0.8545(0.0423)	0.9851(0.0161)	0.9852(0.016)	0.9838(0.0039)

Table 15.8: Modeling Results Before and After Integrating Feature Selection - CHD - MUSC - Testing

From the recorded results of F2 score, Gmean, AUC ROC, and Accuracy on both training and testing sets, we can see that the performance of the model is better when introducing feature selection.

15.1.5. Results for CNS - GU. Below we will show the results for CNS - GU

CNS_GU	F2	Gmean	AUC ROC	Accuracy
LR_CS	0.8743(0.0081)	0.99(0.0028)	0.9901(0.0027)	0.9859(0.0011)
LR_CS - FS	0.8519(0.0084)	0.9838(0.0015)	0.9841(0.0015)	0.9837(0.0012)

Table 15.9: Modeling Results Before and After Integrating Feature Selection - CNS - GU - Training

CNS_GU	F2	Gmean	AUC ROC	Accuracy
LR_CS	0.8616(0.0458)	0.9797(0.0193)	0.9799(0.0191)	0.9861(0.0038)
LR_CS - FS	0.8534(0.0407)	0.9852(0.0155)	0.9853(0.0154)	0.984(0.0041)

Table 15.10: Modeling Results Before and After Integrating Feature Selection - CNS - GU - Testing

From the recorded results of F2 score, Gmean, AUC ROC and Accuracy on both training and testing sets, we can see that the performance of the model is better when running cost sensitive logistic regression without introducing feature selection.

15.1.6. Results for CNS - MUSC. Below we will show the results for CNS - MUSC

CNS_MUSC	F2	Gmean	AUC ROC	Accuracy
LR_CS	0.8526(0.0154)	0.9787(0.0069)	0.9792(0.0066)	0.9843(0.0011)
LR_CS - FS	0.8555(0.0122)	0.9838(0.0055)	0.9842(0.0053)	0.9839(0.001)

Table 15.11: Modeling Results Before and After Integrating Feature Selection - CNS - MUSC - Training

CNS_MUSC	F2	Gmean	AUC ROC	Accuracy
LR_CS	0.8468(0.0531)	0.9726(0.0306)	0.9731(0.0299)	0.9848(0.0037)
LR_CS - FS	0.8563(0.0418)	0.9852(0.0162)	0.9853(0.0161)	0.984(0.0036)

Table 15.12: Modeling Results Before and After Integrating Feature Selection - CNS - MUSC - Testing

From the recorded results of F2 score, Gmean, AUC ROC and Accuracy on both training and testing sets, we can see that the performance of the model is stable after limiting the features to the ones selected in the feature selection process.

15.1.7. Results for GU - MUSC. Below we will show the results for GU - MUSC

GU_MUSC	F2	Gmean	AUC ROC	Accuracy
LR_CS	0.8472(0.0109)	0.9692(0.0068)	0.9703(0.0065)	0.9842(0.0007)
LR_CS - FS	0.8626(0.0047)	0.984(0.0005)	0.9843(0.0005)	0.9839(0.0005)

Table 15.13: Modeling Results Before and After Integrating Feature Selection - GU - MUSC - Training

GU_MUSC	F2	Gmean	AUC ROC	Accuracy
LR_CS	0.8487(0.0363)	0.9731(0.0276)	0.9735(0.0269)	0.9841(0.0031)
LR_CS - FS	0.8632(0.0219)	0.9856(0.0137)	0.9857(0.0136)	0.9841(0.0017)

Table 15.14: Modeling Results Before and After Integrating Feature Selection - GU - MUSC - Testing

From the recorded results of F2 score, Gmean, AUC ROC and Accuracy on both training and testing sets, we can see that the performance of the model is better when introducing feature selection.

15.2. Interpretability Using SHAP.

15.2.1. How it works. The goal of SHAP is to explain the prediction of an instance x by computing the contribution of each feature to the prediction. The SHAP explanation method computes Shapley values from coalitional game theory. The feature values of a data instance act as players in a coalition. Shapley values tell us how to fairly distribute the "payout" (= the prediction) among the features. A player can be an individual feature value, e.g. for tabular data. A player can also be a group of feature values.[28]

Shap can produce several types of plots that can show the contribution of the features in several ways, these are summary plots, force plots and dependence plots. These plots are useful to show the contribution of each feature in the prediction process.

15.2.2. Takeaway Message from SHAP. This chapter allows us to interpret the probabilities produced by the models. SHAP allowed us to interpret the results produced by the models. We were able to identify the impact of each feature, whether it causes a higher probability of getting a birth defect or a lower probability. Also, after comparing these results with the results of features selected in the feature selection process, it was clear that observations made in the feature selection chapter are aligned with the findings done in this chapter, which shows that both results are consistent. The results were mostly aligned with respect to AAP features, medical history, consanguinity degree and folic acid intake before pregnancy. SHAP allows the user to get more information than just prediction scores, it reveals the contribution of each feature in the data during the prediction process, which builds trust from the user perspective. Compared to Feature Selection, SHAP highlights the positive and negative contribution of each feature, whereas; feature selection only reveals features that allow the

models to perform with better performance. For example, medical history and consanguinity degree were classified by SHAP as top contributors for the decision of the models, which is aligned with the literature, because many research studies show that consanguinity degree is one of the most important factors that causes birth defects.

15.2.3. Shap Results - All Defects. We will display plots for the top selected models.

First, we will display the Mean Absolute Shapley Value plots, which shows the importance of each feature in the decision of the model, we can see that for all models, consanguinity degree, medical history, and folic acid intake, mother education and mother age are major contributors to the decision of the models:

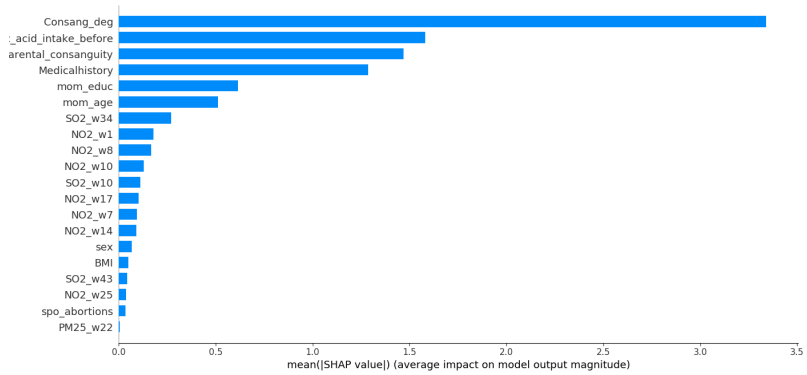


Figure 15.1: Cost Sensitive Logistic Regression Mean Absolute Shapley Value Plot

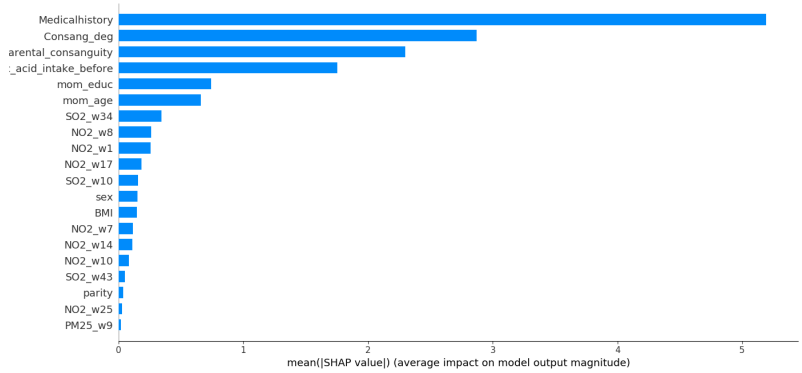


Figure 15.2: Logistic Regression - SMOTE TOMEK - Mean Absolute Shapley Value Plot

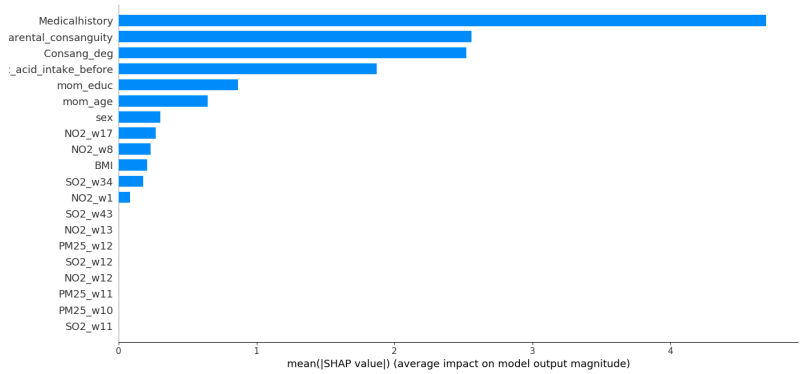


Figure 15.3: Logistic Regression - SMOTENC Mean Absolute Shapley Value Plot

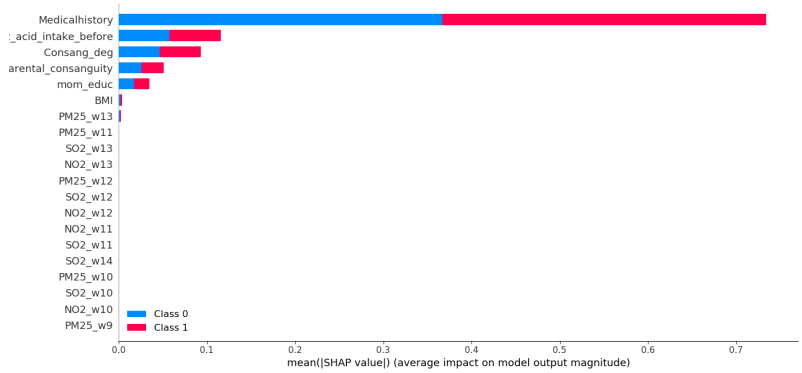


Figure 15.4: Decision Tree - SMOTE TOMEK Mean Absolute Shapley Value Plot

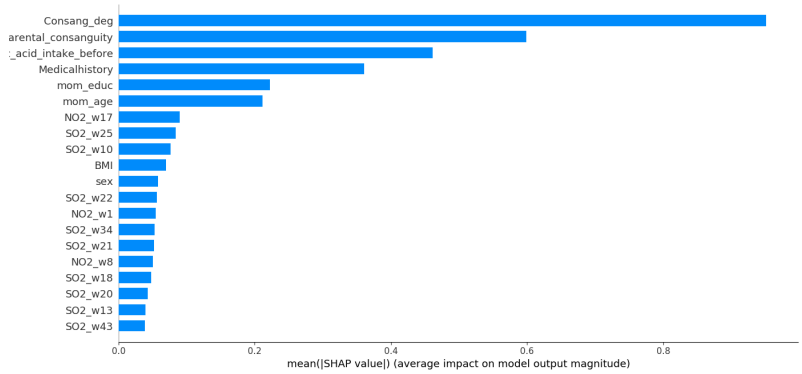


Figure 15.5: Support Vector Machines - Mean Absolute Shapley Value Plot

As a comparison to the four plots, we can see that consanguinity degree, medical history, folic acid intake before pregnancy, mother education, and maternal age are always selected as the top five contributors to the decision of the models, which is aligned with the findings of feature selection chapter because all these features were selected in the feature selection process.
Now, to get an idea of the positive or negative impact caused by these features on the prediction process, we display the importance plots and summary plots, which shows the positive or negative impact of each feature on the prediction itself:
Displaying the importance plot for the decision tree is not supported by the SHAP tool, so we will display it for other models only.

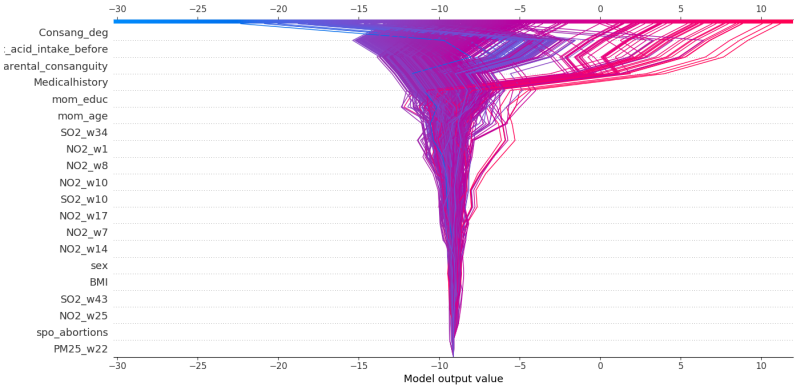


Figure 15.6: Cost Sensitive Logistic Regression Decision Plot

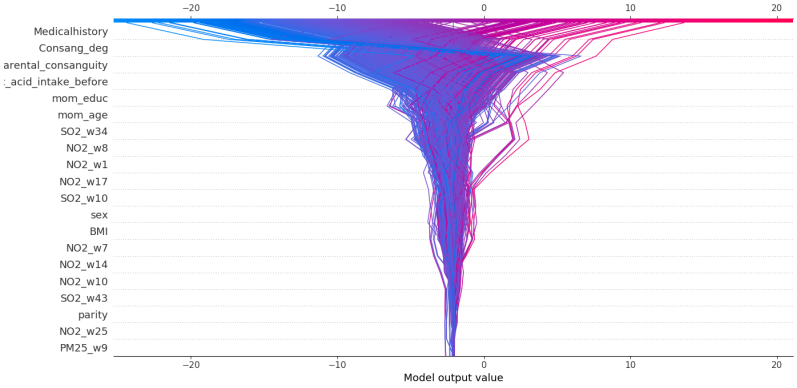


Figure 15.7: Logistic Regression - SMOTE TOMEK - Decision Plot

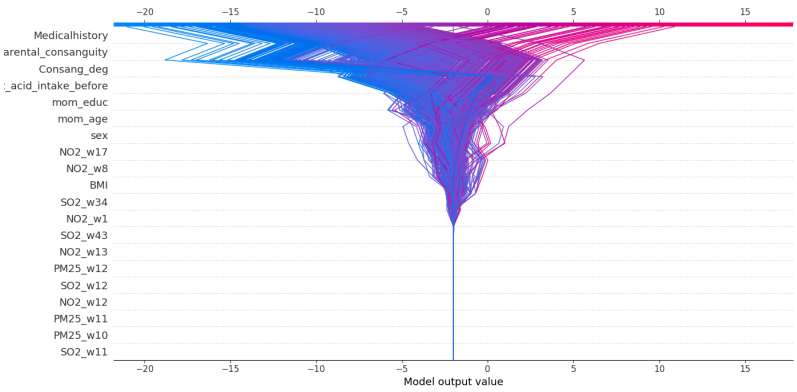


Figure 15.8: Logistic Regression - SMOTENC Decision Plot

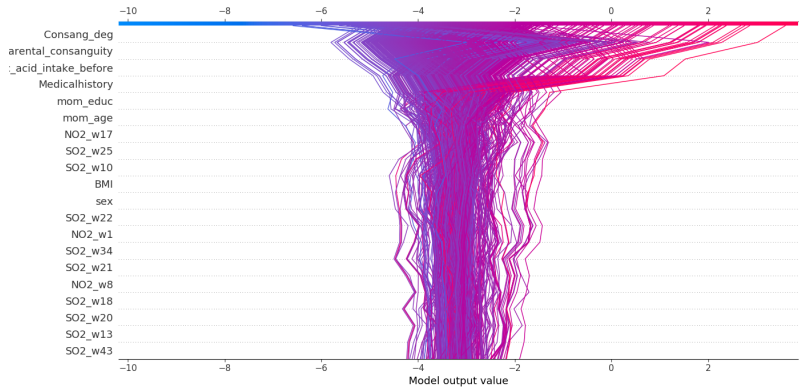


Figure 15.9: Support Vector Machines - Decision Plot

Figures 15.10, 15.11 and 15.12 we can see that for the three models, medical history and consanguinity degree play a positive role in the decision of the models, meaning that whenever these values are higher, these models tend to predict a higher probability. It is the inverse for mother education and folic acid intake.

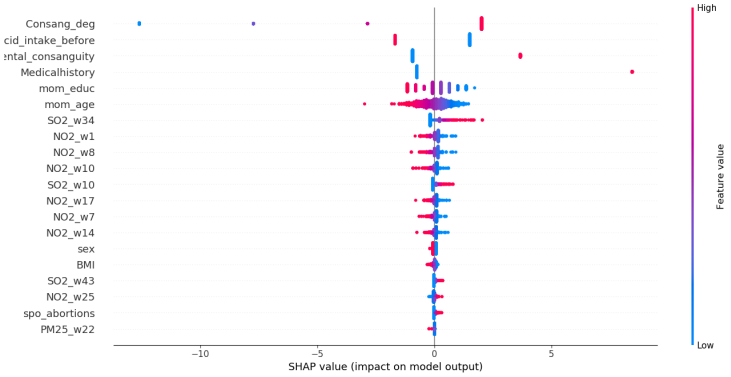


Figure 15.10: Cost Sensitive Logistic Regression Summary Plot

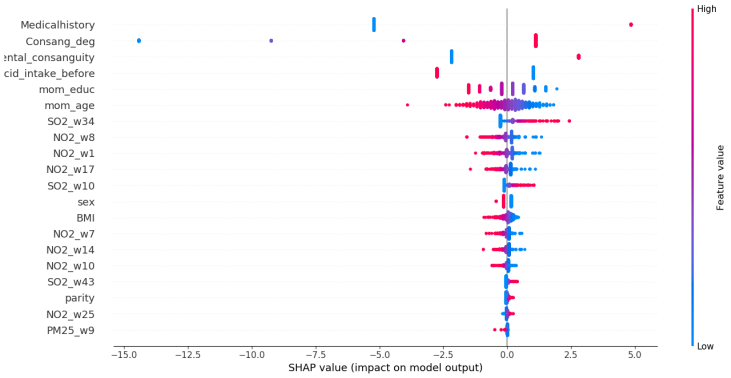


Figure 15.11: Logistic Regression - SMOTE TOMEK - Summary Plot

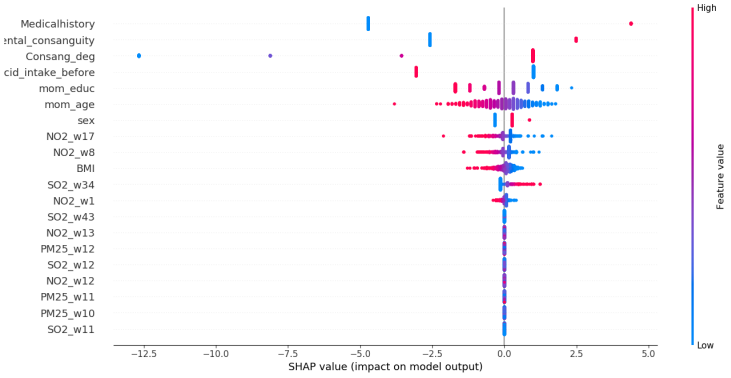


Figure 15.12: Logistic Regression - SMOTENC Summary Plot

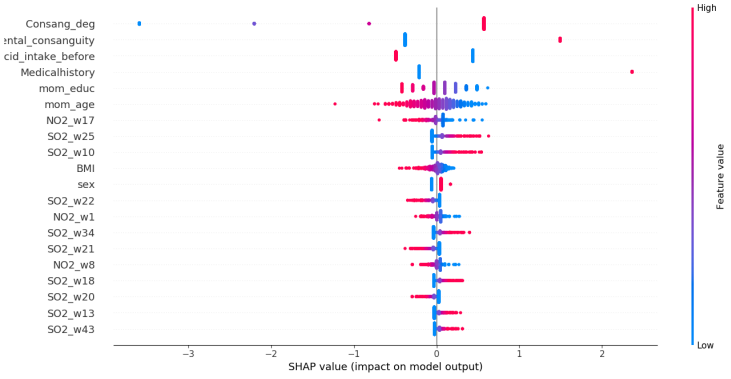


Figure 15.13: Support Vector Machines - Summary Plot

Decision plots represent the path of the decision process and how it is affected by each feature. The red color indicates the positive impact and the blue color indicates the negative impact. For example, if we observe any of these plots, we can see blue and red lines going from top to bottom of the figure, the blue ones are controls or negative cases and the red ones are positive cases. In both figures of Logistic Regression with SMOTE TOMEK and Logistic Regression with SMOTENC we can see that the absence or presence of medical history is highly affecting the decision of the model, i.e. classifying this specific instance as positive or negative. Also, the consanguinity degree follows the same trend, whereas, folic acid intake before pregnancy is negatively affecting the decision, which means that if a woman is consuming more folic acid before pregnancy she has less chance to give birth to a child with birth defects.
To identify more trends and insights, we will look closer at the summary plots, which summarize the observations of the decision plots. These plots have a vertical line centered at zero, and the red color indicates the positive impact of the feature, blue color indicates the negative impact on the decision of the model. Since all the plots look similar to some extent, we will look at them in parallel, we can see that the consanguinity degree, which has a very high impact on the decision of the model, follows an ascending trend, meaning that whenever the consanguinity degree is higher, the chance of getting a birth defect is higher. Looking into the folic acid intake before pregnancy, we can see that it is the inverse, this can be identified by the blue color in the positive area of the figure and the red color in the negative area, meaning that consumption of folic acid before pregnancy will prevent birth defects. Medical history also is a major contributor, we can see that it is purely blue in the negative area, and purely red in the positive area, this means that medical history has a high positive impact on the decision of the model, this can be explained as the following: women with medical history have a higher probability that they give birth to babies with birth defects. Mother education is proportional to the decision of the model, meaning that more educated mothers are less exposed to birth defects.

Now, after exploring summary plots and decision plots we will look at dependency plots. Dependency plots will allow us to explore the features one by one and to look into patterns for each feature separately.

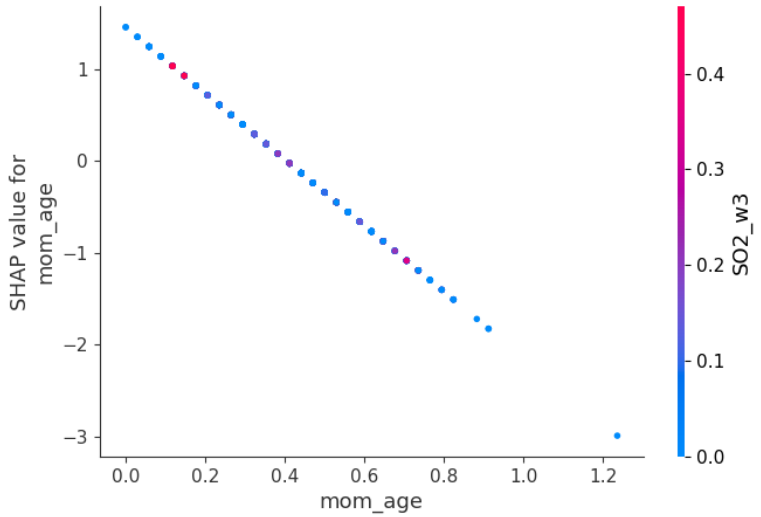


Figure 15.14: Cost Sensitive Logistic Regression Dependency Plot - Mother Age

We can see that maternal age at some ranges combined with exposure to SO2 prior window of risk is directly linked to birth defects. This can be seen as red and purple colors in the data points in the figure above.

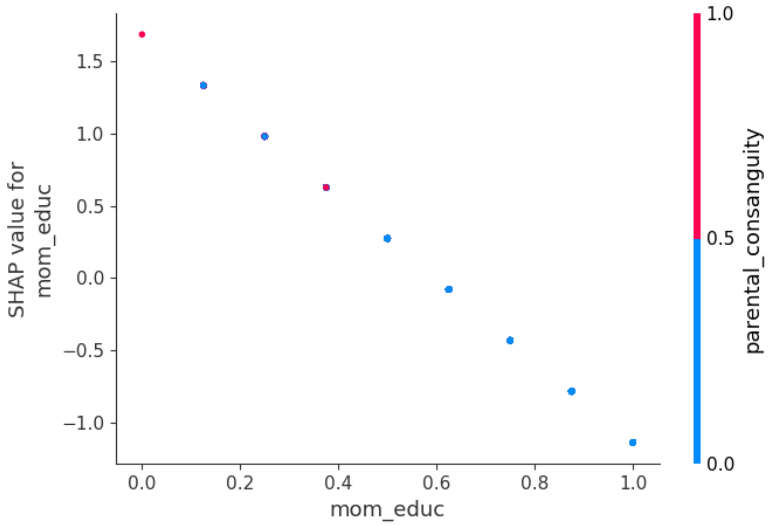


Figure 15.15: Cost Sensitive Logistic Regression Dependency Plot - Mother Education

From the figure above, we can see that low values in mother education feature combined with parental consanguinity is associated with a higher probability of

birth defects, this can be interpreted as a combination of illiteracy and parental consanguinity can be associated with a birth defect. Also, we can see in the figures below that this trend is observed for all selected models and configurations:

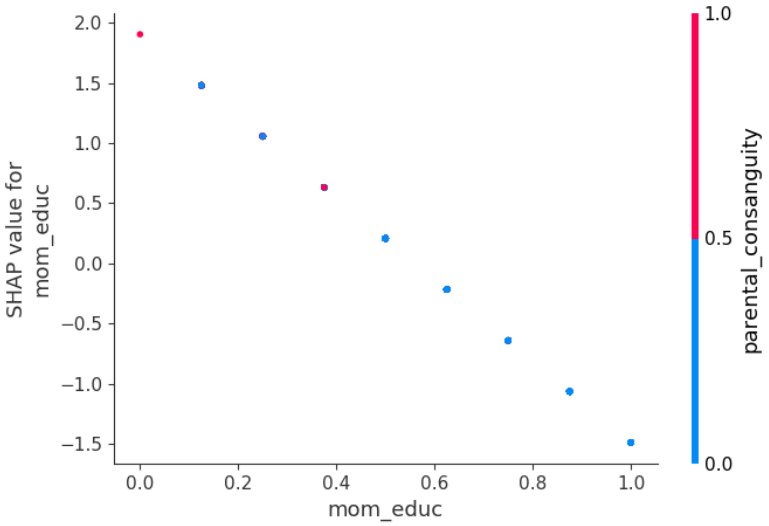


Figure 15.16: Logistic Regression - SMOTE TOMEK Dependency Plot - Mother Education

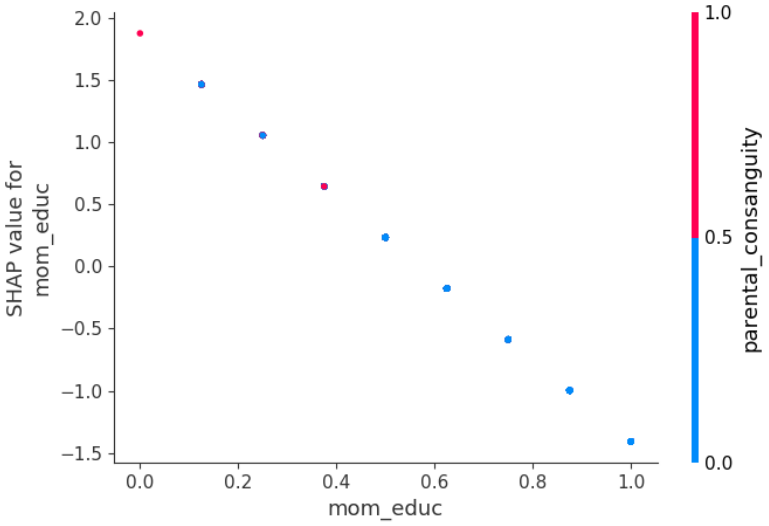


Figure 15.17: Logistic Regression - SMOTENC Dependency Plot - Mother Education

Finally, we will summarize the air pollution fields that showed high impact on the decision of the models:

- Exposure to SO2 post window of risk
- Exposure to NO2 pre window of risk
- Exposure to NO2 during window of risk
- Exposure to NO2 post window of risk

These air pollutants are common between almost all models, and these appear in the window of risk of many birth defects, we can see the danger of the exposure to NO2 and SO2, because these two are main contributors to birth defects according to Shap.

15.2.4. Shap Results - CHD - CNS. In figure 15.18 we can see that the following features have the highest impact on the decision of the model. We will list them from highest impact to lowest impact.

1. Consanguinity degree
2. Folic Acid Intake before pregnancy
3. Medical History
4. Mother Age
5. Number of Living Children

Then we list the air pollution features that have the highest impact on the model decision.

1. Exposure to NO2 pre window of risk
2. Exposure to NO2 during window of risk
3. Exposure to SO2 post window of risk
4. Exposure to PM2.5 post window of risk

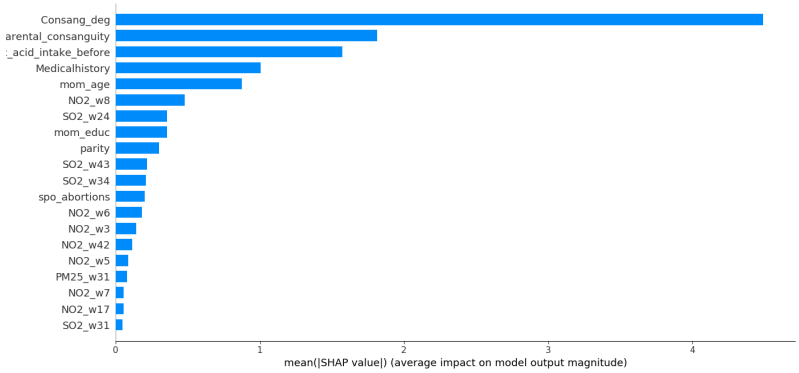


Figure 15.18: Mean Absolute Shapley Value Plot

From figure 15.19 we can see the positive and negative impact of the top features, like consanguinity degree, folic acid intake before pregnancy, medical history. Then we can see that the values are lower for the remaining features.

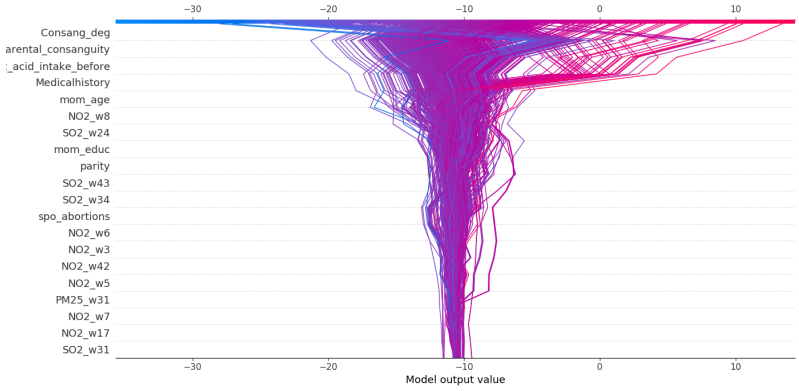


Figure 15.19: Decision Plot

From 15.20 we can see that high values of consanguinity degree have a high impact on the decision of the model. High values of folic acid intake have a negative impact on the decision of the model, meaning that high values in the folic acid intake may lead to a prediction of zero (negative class). Medical history follows the same trend as the consanguinity degree. Younger pregnant women are less prone to get a birth defect, same for mother education.

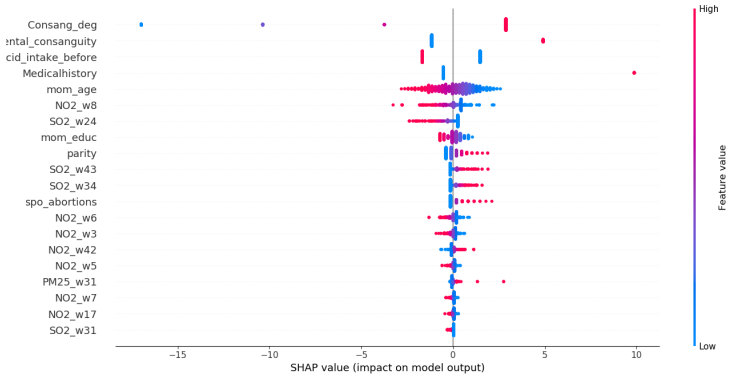


Figure 15.20: Summary Plot

As a summary for CHD - CNS Shaply plots, we can report the following:

- All of the consanguinity degree, folic acid intake, medical history, mother age, and maternal education have a high impact on the decision of the model.
- We can see that exposure to NO2 in weeks 2, 3, 5, 6, 7, and 8 have a high impact on the decision of the model. This is very noticeable because it covers the whole period of the window of risk for both CHD and CNS which starts in week 2 and ends in week 7 and 8.

Also, we can see that these results are aligned with feature selection, because all of medical history, consanguinity degree, mother education, exposure to NO2 during the window of risk are part of the selected features in the feature selection process.

15.2.5. Shap Results - CHD - GU. From 15.21 we can see that the following features have the highest impact on the decision of the model. We will list them from highest impact to lowest impact.

1. Consanguinity degree
2. Folic Acid Intake before pregnancy
3. Medical History
4. Mother Age
5. Sex of the baby - specifically females

6. BMI

Then we list the air pollution features that have the highest impact on the model decision.

1. Exposure to NO2 pre window of risk
2. Exposure to NO2 during window of risk
3. Exposure to SO2 during window of risk
4. Exposure to PM2.5 post window of risk

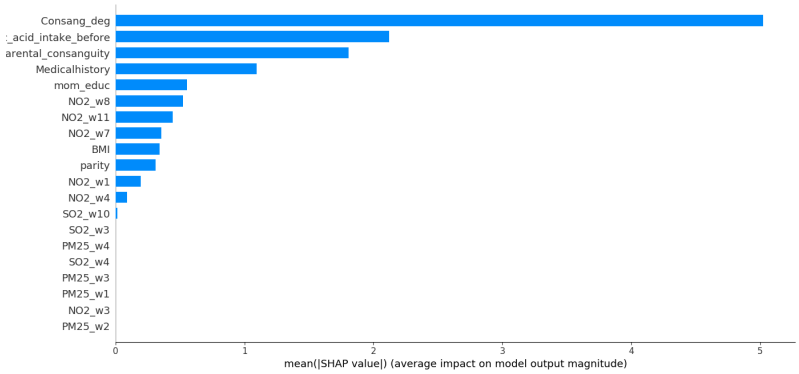


Figure 15.21: Mean Absolute Shapley Value Plot

From figure 15.22 we can see the positive and negative impact of the top features, like consanguinity degree, folic acid intake before pregnancy, medical history. Then we can see that the values are lower for the remaining features.

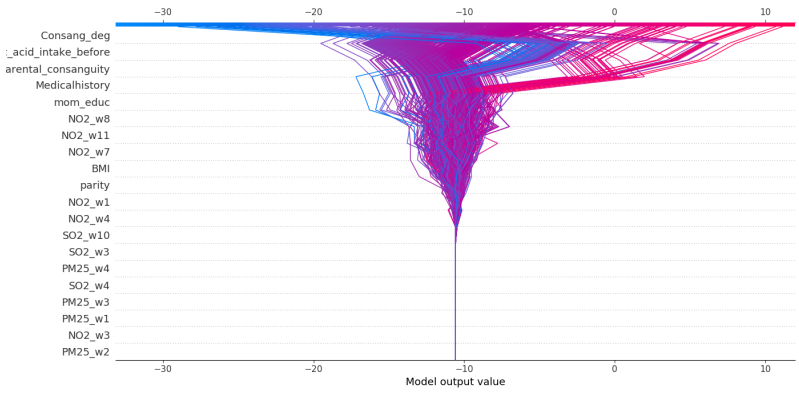


Figure 15.22: Decision Plot

From 15.23 we can see that high values of consanguinity degree have a high impact on the decision of the model. High values of folic acid intake have a negative impact on the decision of the model, meaning that high values in the folic acid intake may will lead to prediction of lower probabilities. Medical history follows the same trend as the consanguinity degree. Younger pregnant women are less prone to get a birth defect, same for mother education.

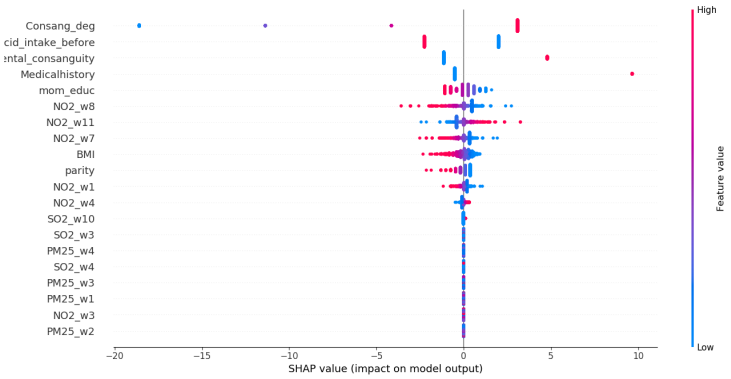


Figure 15.23: Summary Plot

As a summary for CHD - GU Shaply plots, we can report the following:

- All of consanguinity degree, folic acid intake, medical history, mother age and mother education, BMI, Gender of the newborn have high impact on the decision of the model.
- We can see that exposure to NO2 prior and during window of risk of both CHD and GU have high impact on the decision of the model. Exposure to SO2 during window of risk of both CHD and GU, we can conclude that exposure to NO2 during the window of risk has a high impact on the decision of the model. Also, exposure to SO2 during window of risk of both CHD and GU have high impact on the decision of the model.

Also, we can see that these results are aligned with feature selection, because all of medical history, consanguinity degree, mother education, exposure to NO2 and exposure to SO2 during window of risk are part of the selected features in the feature selection process.

15.2.6. Shap Results - CHD - MUSC. From 15.24 we can see that the following features have the highest impact on the decision of the model. We will list them from highest impact to lowest impact.

1. Consanguinity degree
2. Medical History
3. Folic Acid Intake before pregnancy
4. Mother Education
5. Sex of the baby - specifically males
6. Mother Age
7. High exposure to tobacco
8. Number of living children
9. BMI

Then we list the air pollution features that have the highest impact on the model decision.

1. Exposure to NO2 pre window of risk
2. Exposure to NO2 during window of risk
3. Exposure to NO2 post window of risk
4. Exposure to SO2 during window of risk
5. Exposure to PM2.5 post window of risk

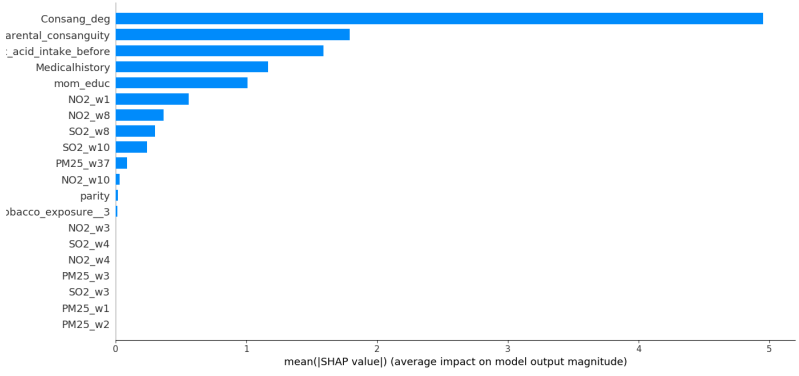


Figure 15.24: Mean Absolute Shapley Value Plot

From figure 15.25 we can see the positive and negative impact of the top features, like consanguinity degree, folic acid intake before pregnancy, medical history. Then we can see that the values are lower for the remaining features.

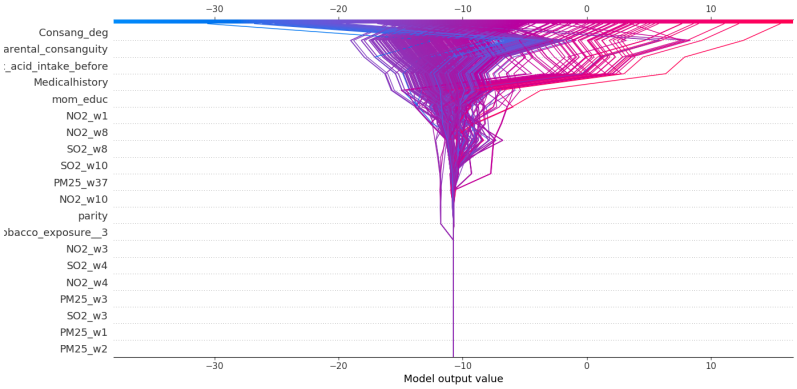


Figure 15.25: Decision Plot

From **??** we can see that high values of consanguinity degree have a high impact on the decision of the model. High values of folic acid intake have a negative impact on the decision of the model, meaning that high values in the folic acid intake may will lead to prediction of lower probabilities. Medical history follows the same trend as the consanguinity degree. Younger pregnant women are less prone to get a birth defect, same for mother education.

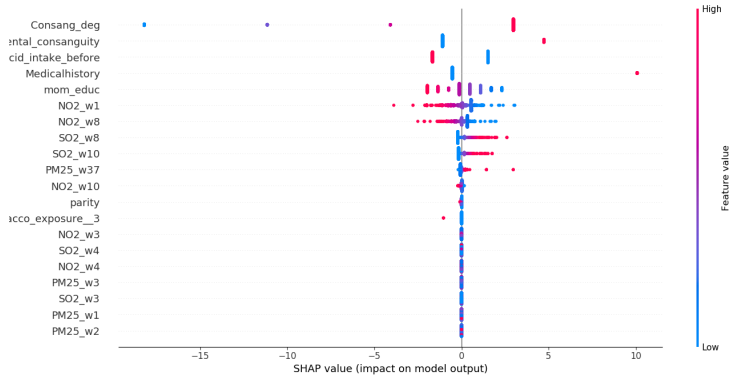


Figure 15.26: Summary Plot

As a summary for CHD - MUSC Shaply plots, we can report the following:

- All of consanguinity degree, folic acid intake, medical history, mother age

and mother education, BMI, Gender of the newborn and exposure to tobacco have high impact on the decision of the model.

- We can see that exposure to NO2 prior, during and post window of risk have high impact on the decision of the model. Same for the exposure of SO2 during window of risk and for the exposure of PM2.5 post window of risk of both CHD and MUSC.

Also, we can see that these results are aligned with feature selection, because all of medical history, consanguinity degree, mother education, exposure to NO2 during window of risk are part of the selected features in the feature selection process.

15.2.7. Shap Results - CNS - GU. From 15.27 we can see that the following features have the highest impact on the decision of the model. We will list them from highest impact to lowest impact.

1. Consanguinity degree
2. Folic Acid Intake before pregnancy
3. Medical History
4. Mother Age
5. Mother Education
6. BMI
7. Parity
8. Sex of the baby - specifically females

Then we list the air pollution features that have the highest impact on the model decision.

1. Exposure to NO2 during window of risk
2. Exposure to SO2 during window of risk
3. Exposure to SO2 post window of risk
4. Exposure to PM2.5 post window of risk

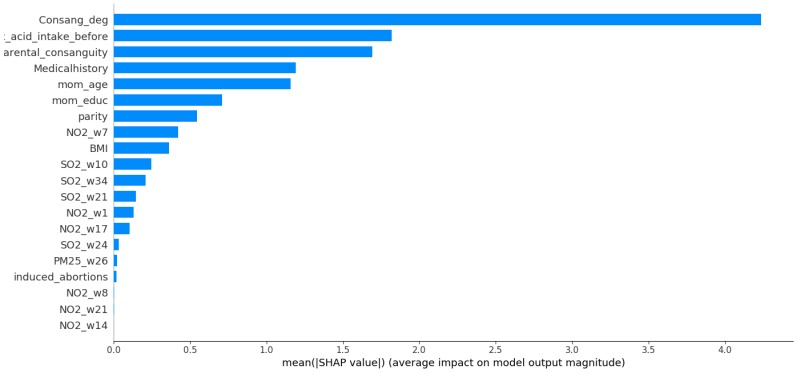


Figure 15.27: Mean Absolute Shapley Value Plot

From 15.28 we can identify that the following three feature have the highest impact on the decision of the model: Consanguinity degree, folic acid intake and medical history.

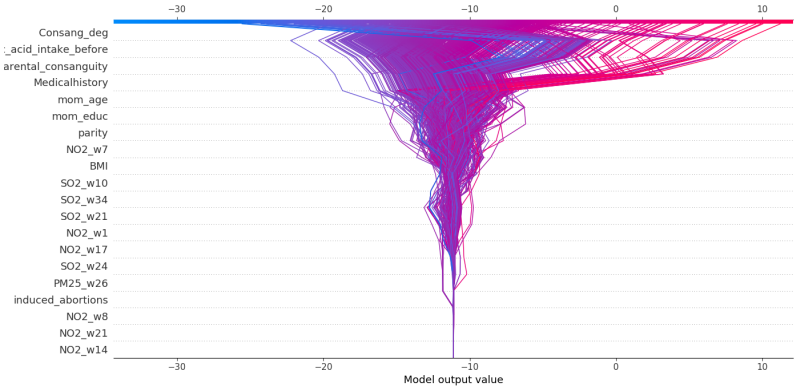


Figure 15.28: Decision Plot

From 15.29 we can see that high values of consanguinity degree have a high impact on the decision of the model. High values of folic acid intake have a negative impact on the decision of the model, meaning that high values in the folic acid intake may will lead to prediction of lower probabilities. Medical history follows the same trend as the consanguinity degree. Younger pregnant women are less prone to get a birth defect, same for mother education.

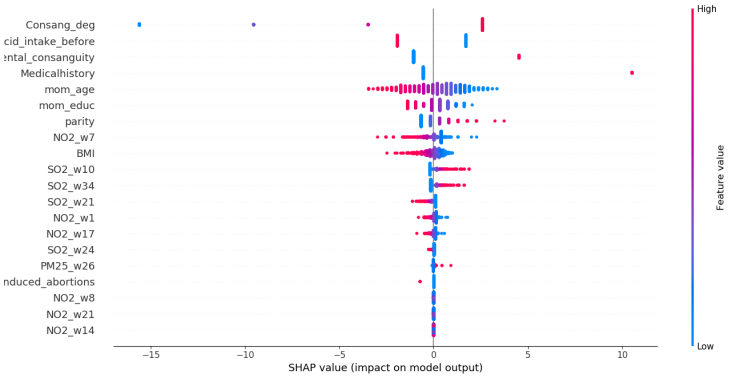


Figure 15.29: Summary Plot

Also, we can see that these results are aligned with feature selection, because all of medical history, consanguinity degree, mother education, exposure to NO2 and SO2 during window of risk are part of the selected features in the feature selection process.

15.2.8. Shap Results - CNS - MUSC. From 15.30 we can see that the following features have the highest impact on the decision of the model. We will list them from highest impact to lowest impact.

1. Consanguinity degree
2. Folic Acid Intake before pregnancy
3. Medical History
4. Mother Education
5. Mother Age
6. Sex of the baby - specifically females
7. Spontaneous abortions
8. Exposure to Tobacco

Then we list the air pollution features that have the highest impact on the model decision.

1. Exposure to SO2 during window of risk

2. Exposure to SO2 post window of risk
3. Exposure to NO2 pre window of risk
4. Exposure to NO2 during window of risk
5. Exposure to NO2 post window of risk
6. Exposure to PM2.5 post window of risk

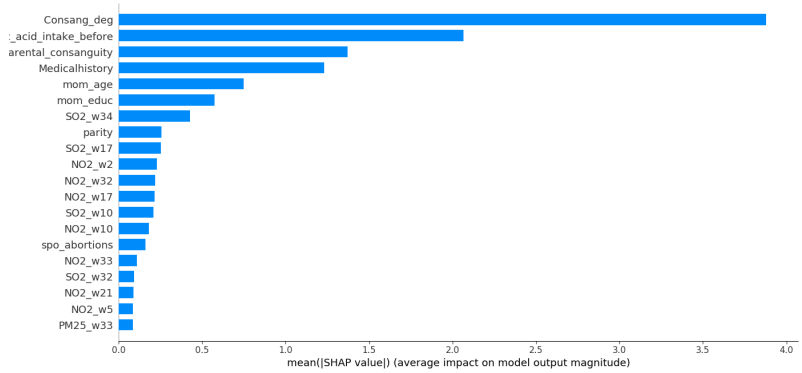


Figure 15.30: Mean Absolute Shapley Value Plot

From 15.31 we can identify that the following three feature have the highest impact on the decision of the model: Consanguinity degree, folic acid intake and medical history.

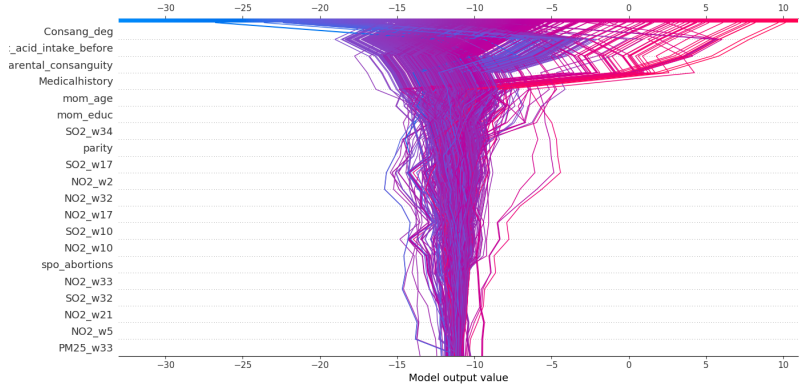


Figure 15.31: Decision Plot

From 15.32 we can see that high values of consanguinity degree have a high impact on the decision of the model. High values of folic acid intake have a

negative impact on the decision of the model, meaning that high values in the folic acid intake may will lead to prediction of zero (negative class). Medical history follows the same trend as the consanguinity degree. Younger pregnant women are less prone to get a birth defect, same for mother education.

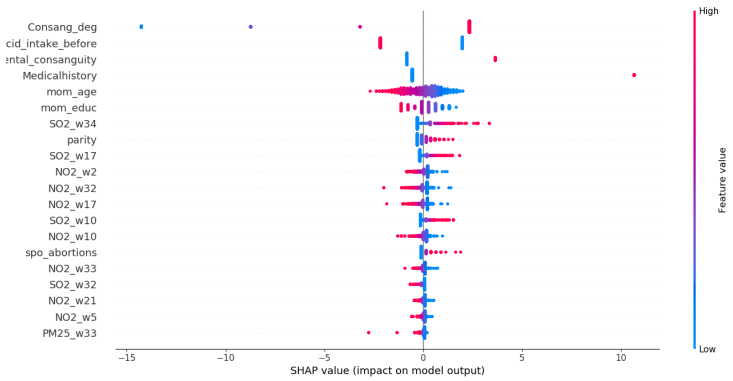


Figure 15.32: Summary Plot

As a summary for CNS - MUSC Shaply plots, we can report the following:

- All of consanguinity degree, folic acid intake, medical history, mother age and mother education, BMI, Gender of the newborn have high impact on the decision of the model.
- We can see that exposure to NO2 during window of risk have high impact on the decision of the model. Exposure to SO2 during window of risk has a high impact of the decision of the model.

Also, we can see that these results are aligned with feature selection, because all of medical history, consanguinity degree, mother education, exposure to NO2 and SO2 during window of risk are part of the selected features in the feature selection process.

15.2.9. Shap Results - GU - MUSC. From 15.33 we can see that the following features have the highest impact on the decision of the model. We will list them from highest impact to lowest impact.

1. Consanguinity degree
2. Folic Acid Intake before pregnancy

3. Medical History
4. Mother Age
5. Mother Education
6. Sex of the baby - specifically females
7. Spontaneous abortions
8. Exposure to Tobacco

Then we list the air pollution features that have the highest impact on the model decision.

1. Exposure to SO2 during window of risk
2. Exposure to NO2 pre window of risk
3. Exposure to NO2 during window of risk
4. Exposure to NO2 post window of risk

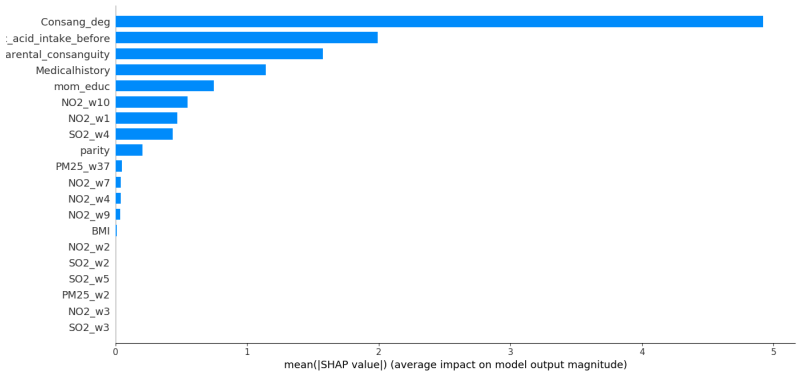


Figure 15.33: Mean Absolute Shapley Value Plot

From 15.34 we can identify that the following three feature have the highest impact on the decision of the model: Consanguinity degree, folic acid intake and medical history.

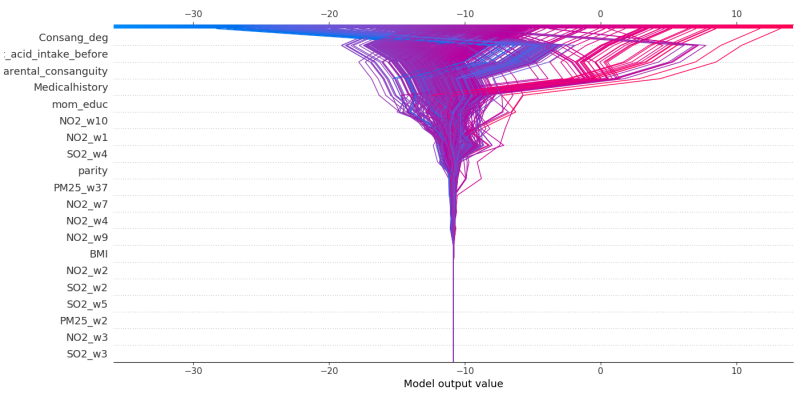


Figure 15.34: Decision Plot

From 15.35 we can see that high values of consanguinity degree have a high impact on the decision of the model. High values of folic acid intake have a negative impact on the decision of the model, meaning that high values in the folic acid intake may will lead to prediction of zero (negative class). Medical history follows the same trend as the consanguinity degree. Younger pregnant women are less prone to get a birth defect, same for mother education.

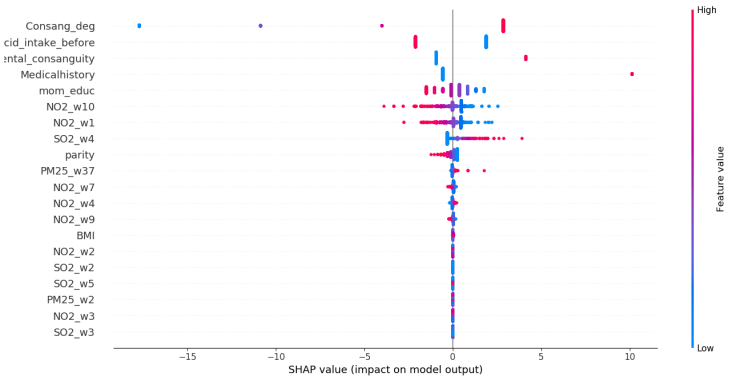


Figure 15.35: Summary Plot

As a summary for GU - MUSC Shaply plots, we can report the following:

- All of consanguinity degree, folic acid intake, medical history, mother age and mother education, BMI, Gender of the newborn have high impact on the decision of the model.
- We can see that exposure to NO2 has a very high impact. Specifically, exposure to NO2 during, pre and post window of risk of both birth defects

(GU and MUSC). Exposure to SO2 during window of risk of GU has high impact on the decision of the model.

Also, we can see that these results are aligned with feature selection, because all of medical history, consanguinity degree, mother education, exposure to NO2 and SO2 during window of risk are part of the selected features in the feature selection process.

15.3. Characterizing Prediction Mistakes. Our goal in this section is to identify patterns of mistakes done by algorithms. Clinicians are interested in such patterns of mistakes to know whether a model can be used or not even if it has recorded good performance using listed metrics. In this section, we study if the models are biased towards a sector of the data and predominantly making mistakes on it. We were able to identify two frequent patterns in the given datasets by performing FP-growth technique. The first pattern is related to mother education, we were able to identify that our models tend to misclassify part of the instances having intermediate or secondary education. The second pattern is related to air pollution, we were able to identify that our models tend to missclassify instances exposed to PM2.5 post window of risk. The reason of both misclassifications can be directly related to the over-representation of this section of the data. From the performed tests, we can see that the percentage of missclassified instances is always between the following two values: the percentage of the first frequent pattern and the percentage of the second frequent pattern, which are 31% and 43% respectively. The following steps were used to perform the mentioned tasks:

1. Identify frequent patterns, we did this part using FP-growth technique and summarized the results in the next section. The output of FP-growth technique is a set of patterns identified over the input dataset, for example, if feature A has some specific trend or majority for almost 80% of the data, in this case it will be classified as a frequent pattern.

2. Rank predicted instances based on risk score estimates. Then set the value to 1 for the top ranked predictions and 0 for others.

3. Add a field to the datasheet that indicates whether this instance was predicted correctly, set it to 1 for wrong predictions and 0 for others.

4. For each frequent pattern identified by FP-growth technique, compute the probability of mistake on each instance. This is done by iterating over all the predictions where the pattern is true and computing the fraction of these predictions where mistake field is set to 1.

5. Sort the patterns of mistakes in descending order to identify the top mistake patterns.

Using FP-growth technique we were able to identify three frequent patterns in the datasets:

- Mom Education - 43% intermediate and secondary
- PM 2.5 post window of risk - 31% between 21 and 22

Then, we applied the procedure above, we ended up with the following mistake patterns.

15.3.1. Prediction Mistakes For All Defects. For All defects we were able to identify the following patterns of mistakes:

- 38.5% of the mistakes recorded for cost sensitive logistic regression have either patterns 1 or 2 or both together. Mother education between 4 and 5, and PM 2.5 during post window of risk is between 21 and 22.
- 43.82% of the mistakes recorded for decision tree have either patterns 1 or 2 or both together. Mother education between 4 and 5, and PM 2.5 during post window of risk is between 21 and 22.
- 43.84% of the mistakes recorded for cost sensitive support vector machines have either patterns 1 or 2 or both together. Mother education between 4 and 5, and PM 2.5 during post window of risk is between 21 and 22.
- 43.70% of the mistakes recorded for logistic regression have either patterns 1 or 2 or both together. Mother education between 4 and 5, and PM 2.5 during post window of risk is between 21 and 22.

If Mother education between is between 4 and 5, and PM 2.5 during post window of risk is between 21 and 22, then probability of mistake is 38.5%.

15.3.2. Prediction Mistakes For CHD - CNS. For All defects we were able to identify the following patterns of mistakes:

- 40.6% of the mistakes recorded for cost sensitive logistic regression have either patterns 1 or 2 or both together. Mother education between 4 and 5, and PM 2.5 during post window of risk is between 21 and 22.

If Mother education between is between 4 and 5, and PM 2.5 during post window of risk is between 21 and 22, then probability of mistake is 40.6%.

15.3.3. Prediction Mistakes For CHD - GU. For All defects we were able to identify the following patterns of mistakes:

- 46.4% of the mistakes recorded for cost sensitive logistic regression have either patterns 1 or 2 or both together. Mother education between 4 and 5, and PM 2.5 during post window of risk is between 21 and 22.

If Mother education between is between 4 and 5, and PM 2.5 during post window of risk is between 21 and 22, then probability of mistake is 46.4%.

15.3.4. Prediction Mistakes For CHD - MUSC. For All defects we were able to identify the following patterns of mistakes:

- 35.5% of the mistakes recorded for cost sensitive logistic regression have either patterns 1 or 2 or both together. Mother education between 4 and 5, and PM 2.5 during post window of risk is between 21 and 22.

If Mother education between is between 4 and 5, and PM 2.5 during post window of risk is between 21 and 22, then probability of mistake is 35.5%.

15.3.5. Prediction Mistakes For CNS - GU. For All defects we were able to identify the following patterns of mistakes:

- 44% of the mistakes recorded for cost sensitive logistic regression have either patterns 1 or 2 or both together. Mother education between 4 and 5, and PM 2.5 during post window of risk is between 21 and 22.

If Mother education between is between 4 and 5, and PM 2.5 during post window of risk is between 21 and 22, then probability of mistake is 44%.

15.3.6. Prediction Mistakes For CNS - MUSC. For All defects we were able to identify the following patterns of mistakes:

- 42.4% of the mistakes recorded for cost sensitive logistic regression have either patterns 1 or 2 or both together. Mother education between 4 and 5, and PM 2.5 during post window of risk is between 21 and 22.

If Mother education between is between 4 and 5, and PM 2.5 during post window of risk is between 21 and 22, then probability of mistake is 42.4%.

15.3.7. Prediction Mistakes For GU - MUSC. For All defects we were able to identify the following patterns of mistakes:

- 48.1% of the mistakes recorded for cost sensitive logistic regression have either patterns 1 or 2 or both together. Mother education between 4 and 5, and PM 2.5 during post window of risk is between 21 and 22.

If Mother education between is between 4 and 5, and PM 2.5 during post window of risk is between 21 and 22, then probability of mistake is 48.1%.

Data Sheet / Model	Mistake Pattern	Percentage of Mistakes
All Defects / cost sensitive logistic regression	if mother education = intermediate or secondary or PM 2.5 post window of risk ∈ [21, 22], then mistake	38.5%
All Defects / logistic regression	if mother education = intermediate or secondary or PM 2.5 post window of risk ∈ [21, 22], then mistake	43.70%
All Defects / cost sensitive support vector machines	if mother education = intermediate or secondary or PM 2.5 post window of risk ∈ [21, 22], then mistake	43.84%
All Defects / decision tree	if mother education = intermediate or secondary or PM 2.5 post window of risk ∈ [21, 22], then mistake	43.82%
CHD - CNS / cost sensitive logistic regression	if mother education = intermediate or secondary or PM 2.5 post window of risk ∈ [21, 22], then mistake	40.6%
CHD - GU / cost sensitive logistic regression	if mother education = intermediate or secondary or PM 2.5 post window of risk ∈ [21, 22], then mistake	46.4%
CHD - MUSC / cost sensitive logistic regression	if mother education = intermediate or secondary or PM 2.5 post window of risk ∈ [21, 22], then mistake	35.5%
CNS - GU / cost sensitive logistic regression	if mother education = intermediate or secondary or PM 2.5 post window of risk ∈ [21, 22], then mistake	44%
CNS - MUSC / cost sensitive logistic regression	if mother education = intermediate or secondary or PM 2.5 post window of risk ∈ [21, 22], then mistake	42.4%
GU - MUSC / cost sensitive logistic regression	if mother education = intermediate or secondary or PM 2.5 post window of risk ∈ [21, 22], then mistake	48.1%

Table 15.15: Summary of the Results

15.3.8. Summary of The Results. Table 15.15 shows the mistake patterns and percentage of mistakes for each pattern, percentage of mistakes is the ratio of mistakes recorded over the frequent patterns. The percentage is derived from the following formula:

$\frac{Number\ of\ Mistakes\ over\ frequent\ Patterns}{Total\ Number\ of\ Mistakes}$

From table 15.15 we can see that models are biased towards a sector of the data. This sector of data is an over-represented one. Models are making mistakes in two frequent patterns. These patterns are secondary / intermediate mother education and PM 2.5 post window of risk. PM 2.5 post window or risk is not a pattern that the clinician or the patient should worry about because this feature was not selected by feature selection and it didn't show any high contribution to the decision of the models in SHAP. In the other hand, mother education is an important feature because it is selected by feature selection and was classified as an important contributor to the decision of the model in SHAP, however; the percentage of mistakes made over this sector of the data looks normal for all sets of models, because this pattern represents 43% of the overall data, which is almost the same percentage of mistakes done over this sector of the data. So, our overall findings are the following:

- 43% of the individuals in our dataset have the following pattern: mother education = intermediate or secondary or PM 2.5 post window of risk $\in$ [21, 22]
- Out of all the mistakes done by the models, around 43% of them have the same pattern mentioned above.
- These two numbers are almost aligned, which means that models are not biased towards this sector of the data.

15.4. Comparing Classifier Predictions. After analyzing each model separately and creating a list of best models with their respective configurations, it is important to compare the performance of these models to know which one is the best. This process can be described as producing comparative analysis for classifier predictions. Although, several statistical algorithms can produce these similarity scores where the correlation can be computed, we will use Jaccard similarity score to produce the rank correlation of the output of these sets of algorithms.

Also, we should note that similarity is important for us at top k, not only on the overall dataset, especially that we are dealing with risk scores and clinicians might be interested in the similarity at some top k level. In this comparison process we used top k as a percentage of the total data.

We used Jaccard similarity score computed between sets of predictions produced in each model. The way Jaccard similarity score is computed is the following: given two sets A and B, first the intersection between the two sets is computed, then this number is divided by the union. In our case, we looked at the similarity between predictions made for each case. We performed Jaccard similarity test

between sets of produced predictions at top k.

We performed similarity test between all pairs of algorithms and we plotted them below:

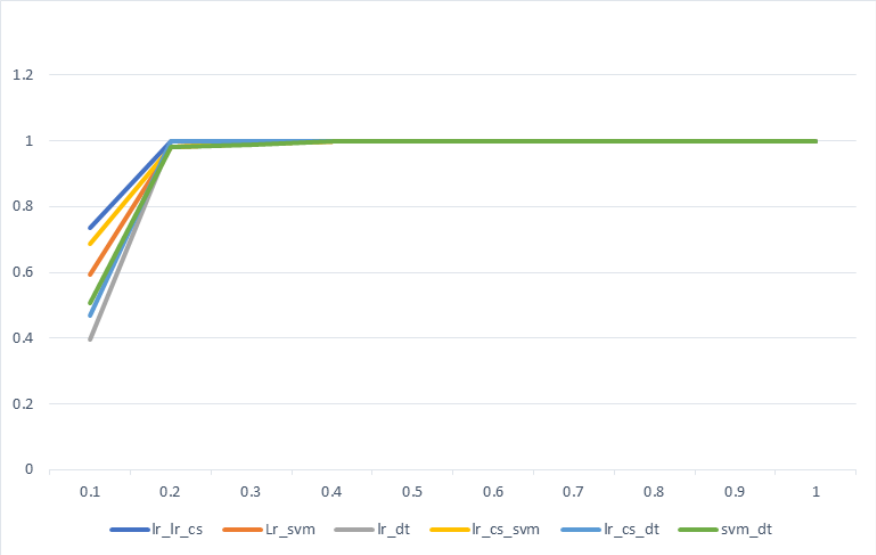


Figure 15.36: Similarity between models at Top K

As we can see, the most similar set of models is logistic regression and cost sensitive logistic regression, the next set of similar models is logistic regression and support vector machines. Also, we can see that the dissimilarity is minimal for large values of k greater than 20%.
So, what we did in this section is comparing the performance of all the models, to get the best ones at top k. We can see that all the models are disposable at k greater than 20%, therefore, we can look closer at the models for k less than 20%.
The best model in terms of efficiency (training time), ease-of-use and top k precision and top k recall is cost sensitive logistic regression.

Here, we display a table of pros and cons evaluating similar models: logistic regression, cost sensitive logistic regression and support vector machines, first we display a table showing the alignment between SHAP results and Feature Selection results for selected models.

	logistic regression	cost sensitive logistic regression	SVM
Consanguinity Degree	aligned	aligned	aligned
Folic Acid intake	aligned	aligned	aligned
Medical History	aligned	aligned	aligned
Mother education	aligned	aligned	aligned
Mother age	aligned	aligned	aligned
Gender	aligned	not aligned	aligned
BMI	aligned	not aligned	aligned
Parity	not aligned	not aligned	not aligned
Air Pollution	not fully aligned	not fully aligned	not fully aligned

Table 15.16: SHAP - Feature Selection Alignment

	logistic regression	cost sensitive logistic regression	SVM
Ease-of-use	same ease of use	same ease of use	same ease of use
Training Time	171 seconds	184 seconds	376 seconds
trustworthy shap	Partial Alignment	Partial Alignment	Partial Alignment
top k precision	best	best	worst
top k recall	best	best	worst
bias with respect to sectors of the data	more biased for mistakes done on frequent patterns (43.7%)	minimal recorded bias on frequent patterns (38.5%)	more biased for mistakes done on frequent patterns (43.84%)

Table 15.17: Models Pros and Cons

Size of training dataset here is 3928. Ease of use is defined for models when they can be used with selected features from feature selection process, since all the models have the same performance or better performance for feature selection, therefore, ease-of-use was the same for all of them. SHAP alignment is defined as: how aligned with Feature selection.

15.5. Takeaway message from this Chapter. In this chapter, we interpreted the output and performance of the used models. The interpretation process covered several approaches: Integrating feature selection, interpreting models' output using SHAP, characterizing prediction mistakes and comparing classifier predictions. All these approaches helped us to identify which models produce the most trustable predictions. Also, we were able to identify features that have the highest contribution in the prediction process in the SHAP section.

Chapter 16

Impact of Machine Learning and Artificial Intelligence on Promoting Precision Public Health

Precision Health and Precision Public Health can be defined as considering all the variations in gene, environment, and lifestyle while providing preventing measures and to design efficient interventions to the individual and population, respectively, on time.[29]

Much public health is addressed by precision public health, such as health promotions and health disparities in a population and environmental data. [30] Data science and artificial intelligence can provide better insights, can deliver service in a more efficient, accurate, and productive way from traditional health care and medicine.[31]

Data analytics approaches based on machine learning to automate the identification of patterns in data sets and improve decisions making have shown promising results in biomedicine and healthcare. [32]

Meanwhile, the Rocke Feller foundation is one of the foundations that work on the topic of precision public health, their targeted locations are India and Uganda. Topics they are dealing with are Environmental and social variables affecting food, water, transport, and human support systems. They aim to make community health more proactive and responsive to population needs with the goal of improving the health and wellbeing of people around the world. Their initiative focuses on using predictive analysis for big data to prevent health threats. Examples of these solutions: In Bangladesh, they used social mapping, a local census, and mobile data technology to estimate the placing of birthing and other

health facilities inaccessible locations, this project was associated with improved maternal and neonatal health outcomes. Another initiative was done in Senegal to slow the spread of malaria by tracking population movement, researchers combined cellular phone data from calls and texts with data on malaria incidence to track the relationship between travelers and the spread of the disease, and finally, strict policies were assigned following this study.[33]

VastBiome, a drug discovery company, uses deep learning in a distributed supercomputing environment to discover new drugs. More specifically, they use machine learning to assess how the microbiome impacts diseases. [34]

GNS Inc. published a study that uses ensemble-based and Bayesian modeling to predict risk factors causing hypoglycemia among patients with type 2 diabetes. The input data was a wide range of demographics, comorbidities, and medications, many predictive machine learning tools were trained on the given data and assessed using AUC ROC score, the best score was recorded for a logistic regression-based model which was 0.73. [35]

Clevy.io a French startup has created a chatbot for COVID-19 with the help of AWS infrastructure. The chatbot uses AI to make it easier for people to find official government communications about the COVID-19 virus from real-time information from the French government and the World Health Organization. Moreover, the chatbot helps people with known symptoms to do self check-up. [36]

Another study on the COVID-19 virus was done using machine learning to forecast the spread of the virus. Researches from Chan Zuckerberg Biohub in California built a machine learning model that estimates the number of undetected infections and their consequences for public health. This was done by processing the data from 12 regions around the globe. These researchers partnered with AWS Diagnostic Development Initiative to run their machine learning modeling tasks. [36]

In our study, we created a new approach for detecting some of the confounders of a set of birth defects using the Feature Selection technique. Feature selection is a technique used to identify which features' variations affect the outcome of machine learning tools. Our algorithm is the following: first, we run the feature selection tool to identify which features are important for the decision making of the machine learning tool, the process is done through k-fold cross-validation, using 10 folds, then we select the top 10 features that were selected in this process. Finally, we repeat the process 10 times to get a number between 0 and 100, where 100 means that the feature is selected everywhere, and 0 means that the feature was never selected by the feature selection process.

Feature Selection results can be impactful to the study of Birth Defects exploration and prediction in a multitude of ways. Some of these impacts affect the birth defect study at the data level, and others affect it at the modeling level, should practitioners opt to approach the birth defect prediction problem using machine learning models, for example:

- Data interpretation: the highest-ranking features can guide medical practitioners whereby they can interpret the data in such a way that helps them prioritize areas of care such that they can exercise utmost intervention, and rank patients in order of highest risk for experiencing birth defects.

- Feature Selection: for any prospective Machine Learning based, predictive modeling exercise to be undertaken with the present datasets, feature selection helps reduce the input space in such a way that only the highly relevant features are retained. If the overall performance of such models with the selected features is at par or better than their performance on all of the original input set, this is considered to be highly impactful for the following reasons. First off, the training time itself would be dramatically reduced, leading to faster decisions with the same or better accuracy. Secondly, collecting data is an expensive endeavor. When only a limited number of features are shown to be capable of attaining comparable or better predictive accuracy than with all the given features, the material and administrative costs of data collection is substantially reduced.

- Model Interpretation: when Machine learning models are built then tested, they can be subjected to machine learning interpretability techniques that extract the rules that led to a given black-box model to its predictions. If those rules turn out to comply with FS results, this signals that the model in question is to be trusted by the end-user, since its actions would be validated by what the FS exploration says is most important about the input space about the response variable.

- Dataset quality assurance: Our FS findings are consistent with the vast literature that asserts that degree of parental consanguinity, folic acid intake initiation before pregnancy, presence of chronic conditions, and tobacco exposure, are some of the most important features relevant to all four birth defects outcomes. This gives ample assurance that our sample dataset is of a quality that ensures sufficient representativeness and generalization capabilities. This will be of utmost importance once we attempt to build machine learning-based predictive models around the sample dataset.

- Actionable Insight: Our FS findings reveal an important role for average PM2.5 exposure during the window of risk for three birth defects (GUD,

NTD, and CHD) with CHD also additionally being affected by the average exposure before the window of risk. These findings can help public health practitioners consolidate their case around the epidemiological impact of air pollution on birth defects and hopefully propel the argument that air pollution impact in Lebanon is validated by national data.

This approach allowed us to identify new causes of birth defects. And this study may lead to new regulations in the long term to prevent environmental confounders causing birth defects.

Moreover, through the use of out interpretable algorithms, we will help the clinicians to identify direct causes of symptoms causing the birth defects.

Chapter 17

Impact of Machine Learning and Artificial Intelligence on Promoting Precision Medicine and Significance of the Study from Medical Point of View

The ambition of precision medicine is to generate early diagnosis for individuals. based on biological data, genes, and environmental data. It takes advantage of artificial intelligence (AI) algorithms to predict risk in certain diseases such as cancers and cardiovascular disease.[37]. Supervised learning algorithms have been successfully applied to problems in the prediction, diagnosis, and treatment of CVD as well as image analysis.[38]. Precision medicine has the potential to revolutionize healthcare by treating patients as their own unique entity. This process consists of looking into the unique genetic and environmental components of each individual. [39] [40] [41]

The topic of precision medicine is important nowadays. By performing a simple search on Google Trends, we can see that as of 2014, this topic is rising and more searches related to it are done daily.[42]

Several startups around the world are investing in the domain of precision medicine, for instance; InsightRX is one of these startups. They built a cloud-based clinical decision support platform to help clinicians individualize treatment at the point of care, the platform provides personalized medicine by generating predictive analysis. Their system is built on top of demographics, biomarkers, genetics, and drug levels. More specifically, they combine the following four indicators: clinical characteristics, dosing history, labs and biomarkers, and pharmacogenetics. Then, they use a machine learning process known as Model-based

Bayesian forecasting to generate insights for clinicians. [43]

Foundation for Precision Medicine, a non-profit US-based company uses Google Cloud to detect Alzheimer's disease using big data. Their target is to perform an early assessment of the disease to get better treatment for it. A team of data scientists from Harvard, MIT, and John Hopkins are leading studies to develop specific machine learning algorithms that will be used in this domain. The team claims that their tool can help detect Alzheimer's months or years in advance. This tool was built using machine learning trained using Google Cloud Compute Engine and ran on its GPUs.[44]

Tempus is another company involved in precision medicine solutions. They use machine learning models combined with data extracted from clinical reports to produce individualized actionable insights for the physicians. In one of their projects, they used supervised learning to classify Tumor mutation burden (TMB), AUC ROC was used for assessment in a cross-validation process. [45]

Benevolent AI, a company that applies AI and machine learning to discover new ways of medicine delivery for patients. One of their solutions is a platform that provides patient-specific treatments. They use machine learning models to enable scientists to determine the right mechanism to modulate and to identify patient endotypes to respond to treatment. Another service they provide is that through AI models, they predict potential disease targets that can be overlooked by scientists. Their source of knowledge is a wide set of structured and unstructured biomedical data that covers relationships between diseases, genes, and drugs. The working team has engineers, biologists, chemists, machine learning experts. One of their work consist of the use of supervised learning to diagnosis and treatment of neurodegenerative diseases. They used MRI scan images, labeled by a neuropathologist. The end product will reduce the time and cost involved in performing clinical trials.[46]

A team at Emory University developed an algorithm for the early prediction of sepsis using cloud-based technology. They created a tool that predicts the onset of sepsis in intensive care-unit patients four to 12 hours sooner than typical clinical recognition. They trained and validated the models on 30,000 and 50,000 patient cohorts. Samples were taken from patients living 1000 miles apart. They were able to achieve an AUC score of 0.85. Their tool not only predicts sepsis in advance of time, but it also reveals the top causes per prediction. [47]

The topic of precision medicine was part of our study. To create our data-driven classification and probability prediction models, we used supervised learning, our models were trained over a wide range of data. Similar to all the research papers and the projects and success stories mentioned before, we aim to create

a tool that provides individualized and personalized assessment for the pregnant woman. So, any assessment given to the clinicians is based on the individual data of the pregnant woman. The assessment given by our tool is an early assessment because we are predicting the existence of a birth defect before the birth of the baby. Therefore, we are providing the clinicians with additional information that could help him/her to save the life of the new born, with less cost and time.

Chapter 18

Notebook Functionality

In this chapter we describe the produced notebook functionalities and how to use them. The main purpose of the notebook is to provide clinicians with a prototype tool that allows them to perform live predictions for birth defects. The notebook is a web application built using a programming tool called Flask that runs on top of a programming language called python. The notebook can be accessed from *http* : //178.79.189.11/

18.1. Notebook Pages and Functionalities. The notebook is made up of three pages:

- Home page: in this page the user chooses the birth defect combination that he/she wants to predict.

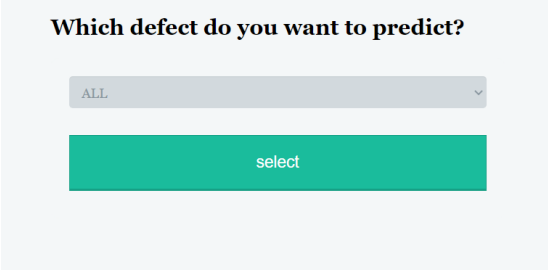


Figure 18.1: Step 1

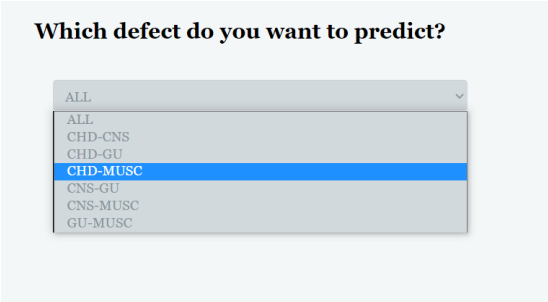


Figure 18.2: Step 2

- Prediction Form: in this page, the user specifies anonymous values of some recorded data for a specific patient. The form is pre-filled with default values for the ease of use, however, all values can be changed. Once the user clicks on predict button, the prediction page will display a prediction percentage score.

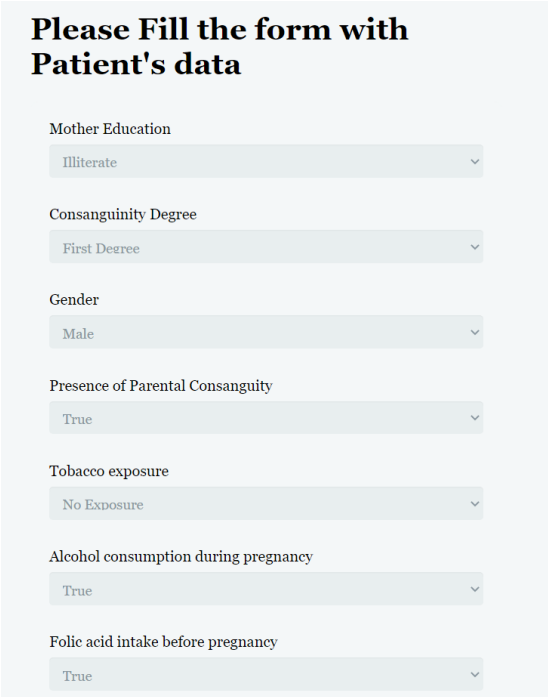


Figure 18.3: Step 3

- Results: the user gets redirected to this page after filling the prediction form, this page displays the predicted score in percentage for the newborn to have a specific birth defect, also; the associated SHAP values are displayed for the user.

Prediction Score:

The Patient has **94%** chance of having a birth defect.

SHAP representation

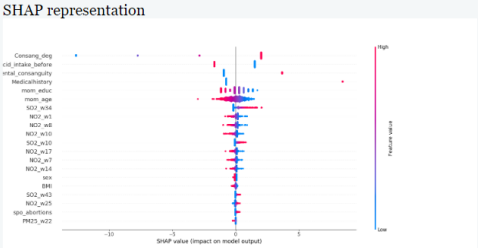


Figure 18.4: Step 4

Chapter 19

Running Best Models on New Datasheets

Below we display the results of best performing models when running them on the new datasheets. We can see that the performance is extremely bad for pairs of defects compared to the initial datasets, this can be observed in both tables for all the recorded metrics.

TRAINING	F2	Gmean	AUC ROC	Accuracy
All - OLD	0.9346(0.0055)	0.9896(0.0029)	0.9897(0.0029)	0.984(0.0012)
All - NEW	0.9083(0.0103)	0.9808(0.0024)	0.9809(0.0024)	0.9753(0.003)

Table 19.1: Good Performance - Training Results

TESTING	F2	Gmean	AUC ROC	Accuracy
All - OLD	0.936(0.0179)	0.936(0.0179)	0.9894(0.0073)	0.9847(0.003)
All - NEW	0.9019(0.0125)	0.9789(0.0084)	0.979(0.0085)	0.9739(0.003)

Table 19.2: Good Performance - Testing Results

TRAINING	F2	Gmean	AUC ROC	Accuracy
CHD_CNS - OLD	0.8351(0.0116)	0.9789(0.0039)	0.9795(0.0037)	0.9839(0.0013)
CHD_CNS - NEW	0.2254(0.0069)	0.633(0.0109)	0.6421(0.0112)	0.5545(0.0085)
CHD_GU - OLD	0.8483(0.0197)	0.9757(0.0116)	0.9763(0.0112)	0.9846(0.0011)
CHD_GU - NEW	0.2304(0.0191)	0.6647(0.0239)	0.6709(0.0222)	0.6121(0.0431)
CHD_MUSC - OLD	0.8363(0.0214)	0.9718(0.0115)	0.9729(0.0109)	0.9831(0.0015)
CHD_MUSC - NEW	0.236(0.0224)	0.6451(0.0326)	0.6504(0.0303)	0.5971(0.0492)
CNS_GU - OLD	0.8743(0.0081)	0.99(0.0028)	0.9901(0.0027)	0.9859(0.0011)
CNS_GU - NEW	0.2395(0.005)	0.6133(0.0088)	0.6318(0.008)	0.4991(0.0108)
CNS_MUSC - OLD	0.8526(0.0154)	0.9787(0.0069)	0.9792(0.0066)	0.9843(0.0011)
CNS_MUSC - NEW	0.2463(0.0053)	0.6068(0.0101)	0.6298(0.0083)	0.4817(0.0142)
GU_MUSC - OLD	0.8472(0.0109)	0.9692(0.0068)	0.9703(0.0065)	0.9842(0.0007)
GU_MUSC - NEW	0.2452(0.0181)	0.6313(0.0307)	0.6453(0.0268)	0.5345(0.0451)

Table 19.3: Bad Performance - Training Results

TESTING	F2	Gmean	AUC ROC	Accuracy
CHD_CNS - OLD	0.8318(0.0361)	0.9771(0.0202)	0.9773(0.0199)	0.984(0.0038)
CHD_CNS - NEW	0.2293(0.0094)	0.6391(0.0138)	0.65(0.0169)	0.5435(0.0227)
CHD_GU - OLD	0.8406(0.0224)	0.972(0.029)	0.9725(0.0282)	0.9845(0.0023)
CHD_GU - NEW	0.2112(0.0156)	0.6389(0.0302)	0.6442(0.0312)	0.5968(0.0407)
CHD_MUSC - OLD	0.8506(0.0488)	0.9788(0.0196)	0.9789(0.0194)	0.9843(0.0038)
CHD_MUSC - NEW	0.2412(0.0176)	0.6554(0.0241)	0.6632(0.0295)	0.5839(0.0455)
CNS_GU - OLD	0.8616(0.0458)	0.9797(0.0193)	0.9799(0.0191)	0.9861(0.0038)
CNS_GU - NEW	0.2407(0.0108)	0.6121(0.0125)	0.6351(0.0188)	0.4824(0.0203)
CNS_MUSC - OLD	0.8468(0.0531)	0.9726(0.0306)	0.9731(0.0299)	0.9848(0.0037)
CNS_MUSC - NEW	0.2361(0.0155)	0.5886(0.0204)	0.613(0.0254)	0.4588(0.0164)
GU_MUSC - OLD	0.8487(0.0363)	0.9731(0.0276)	0.9735(0.0269)	0.9841(0.0031)
GU_MUSC - NEW	0.2379(0.018)	0.6195(0.0286)	0.6349(0.0274)	0.515(0.0512)

Table 19.4: Bad Performance - Testing Results

Chapter 20

Performing Ensemble Learning

In this chapter we perform ensemble learning on All defects sheet, pairs of defects sheets and per birth defect sheets. We performed ensemble learning using five different techniques: Bagging, Balanced Bagging, Bootstrap Random Forest, Easy Ensemble and Weighted Random Forest. First, we display the table of results, then, we compare and interpret them.

20.1. Results. Below we display the results:

All	Bagging	BalancedBagging	BootstrapRandomForest	EasyEnsemble	WeightedRandomForest
F2	0.8459(0.0184)	0.8977(0.0061)	0.5665(0.0303)	0.8841(0.004)	0.5651(0.0305)
Gmean	0.9111(0.0113)	0.9846(0.0019)	0.7147(0.0215)	0.9835(0.0007)	0.7138(0.0215)
AUC ROC	0.9162(0.0099)	0.9847(0.0018)	0.759(0.0157)	0.9836(0.0007)	0.7586(0.0157)
Accuracy	0.9875(0.0015)	0.9734(0.0017)	0.9754(0.0015)	0.9688(0.0012)	0.9752(0.0018)

Table 20.1: All Defects - Training

All	Bagging	BalancedBagging	BootstrapRandomForest	EasyEnsemble	WeightedRandomForest
F2	0.8466(0.0442)	0.8988(0.0212)	0.5855(0.071)	0.882(0.0165)	0.5676(0.062)
Gmean	0.9125(0.0295)	0.9839(0.0055)	0.7326(0.0497)	0.9833(0.0026)	0.7185(0.043)
AUC ROC	0.9162(0.0268)	0.9839(0.0055)	0.7685(0.036)	0.9834(0.0026)	0.7583(0.0312)
Accuracy	0.9873(0.0024)	0.9743(0.0064)	0.9751(0.0048)	0.9684(0.005)	0.9751(0.004)

Table 20.2: All Defects - Testing

CHD	Bagging	BalancedBagging	BootstrapRandomForest	EasyEnsemble	WeightedRandomForest
F2	0.1415(0.0416)	0.4011(0.0113)	0.1176(0.0473)	0.3997(0.0074)	0.0953(0.0296)
Gmean	0.2061(0.0549)	0.9593(0.0062)	0.1692(0.0691)	0.9641(0.0011)	0.139(0.0401)
AUC ROC	0.5655(0.0197)	0.9604(0.0058)	0.5516(0.0206)	0.9647(0.0011)	0.5416(0.013)
Accuracy	0.9897(0.0008)	0.9313(0.0022)	0.9916(0.0005)	0.9301(0.0021)	0.9914(0.0002)

Table 20.3: CHD - Training

CHD	Bagging	BalancedBagging	BootstrapRandomForest	EasyEnsemble	WeightedRandomForest
F2	0.0312(0.0699)	0.3911(0.0312)	0.0(0.0)	0.3971(0.032)	0.0(0.0)
Gmean	0.0754(0.1687)	0.9505(0.0303)	0.0(0.0)	0.964(0.0047)	0.0(0.0)
AUC ROC	0.5131(0.0315)	0.9512(0.0301)	0.4999(0.0003)	0.9647(0.0045)	0.4999(0.0003)
Accuracy	0.9888(0.0011)	0.9313(0.0083)	0.9906(0.0007)	0.93(0.0089)	0.9906(0.0007)

Table 20.4: CHD - Testing

CNS	Bagging	BalancedBagging	BootstrapRandomForest	EasyEnsemble	WeightedRandomForest
F2	0.1089(0.018)	0.3944(0.0057)	0.0643(0.0489)	0.3917(0.0038)	0.0508(0.0329)
Gmean	0.1631(0.0242)	0.9622(0.0032)	0.0928(0.0699)	0.9639(0.0007)	0.0749(0.0486)
AUC ROC	0.5499(0.0092)	0.963(0.003)	0.5282(0.0216)	0.9646(0.0006)	0.5221(0.0143)
Accuracy	0.9897(0.0003)	0.9309(0.0011)	0.9913(0.0004)	0.9299(0.0013)	0.9912(0.0004)

Table 20.5: CNS - Training

CNS	Bagging	BalancedBagging	BootstrapRandomForest	EasyEnsemble	WeightedRandomForest
F2	0.0323(0.0721)	0.3941(0.0212)	0.0345(0.0771)	0.3905(0.0191)	0.0345(0.0771)
Gmean	0.0755(0.1688)	0.9647(0.0031)	0.0756(0.169)	0.9642(0.0029)	0.0756(0.169)
AUC ROC	0.5133(0.0318)	0.9653(0.003)	0.5142(0.032)	0.9648(0.0028)	0.5142(0.032)
Accuracy	0.9893(0.0015)	0.9313(0.0059)	0.9911(0.0009)	0.9302(0.0056)	0.9911(0.0009)

Table 20.6: CNS - Testing

GU	Bagging	BalancedBagging	BootstrapRandomForest	EasyEnsemble	WeightedRandomForest
F2	0.1204(0.0761)	0.4363(0.0077)	0.0413(0.0379)	0.4276(0.0069)	0.0381(0.0379)
Gmean	0.1858(0.1138)	0.9617(0.0043)	0.0644(0.0584)	0.9638(0.0032)	0.0613(0.0609)
AUC ROC	0.5549(0.0359)	0.9626(0.004)	0.5178(0.0168)	0.9645(0.003)	0.516(0.0166)
Accuracy	0.988(0.0006)	0.9346(0.0024)	0.9892(0.0005)	0.9319(0.0014)	0.989(0.0005)

Table 20.7: GU - Training

GU	Bagging	BalancedBagging	BootstrapRandomForest	EasyEnsemble	WeightedRandomForest
F2	0.141(0.1044)	0.4306(0.0278)	0.0(0.0)	0.4247(0.023)	0.0303(0.0678)
Gmean	0.3116(0.1855)	0.9657(0.004)	0.0(0.0)	0.9649(0.0034)	0.0707(0.1581)
AUC ROC	0.561(0.0445)	0.9663(0.0039)	0.4994(0.0009)	0.9655(0.0033)	0.5122(0.0281)
Accuracy	0.9882(0.0036)	0.9332(0.0077)	0.9887(0.0018)	0.9317(0.0065)	0.9897(0.0014)

Table 20.8: GU - Testing

MUSC	Bagging	BalancedBagging	BootstrapRandomForest	EasyEnsemble	WeightedRandomForest
F2	0.0644(0.0177)	0.4636(0.0099)	0.0045(0.0063)	0.461(0.0069)	0.013(0.0179)
Gmean	0.1115(0.0343)	0.9616(0.005)	0.0072(0.0099)	0.9655(0.0009)	0.021(0.0289)
AUC ROC	0.5291(0.008)	0.9624(0.0048)	0.5019(0.0027)	0.9661(0.0009)	0.5055(0.0077)
Accuracy	0.9864(0.0004)	0.9344(0.0019)	0.9886(0.0002)	0.933(0.0018)	0.9887(0.0002)

Table 20.9: MUSC - Training

MUSC	Bagging	BalancedBagging	BootstrapRandomForest	EasyEnsemble	WeightedRandomForest
F2	0.0513(0.0702)	0.4553(0.0299)	0.0(0.0)	0.4593(0.0243)	0.027(0.0604)
Gmean	0.1332(0.1823)	0.9554(0.0246)	0.0(0.0)	0.9656(0.0034)	0.0667(0.1491)
AUC ROC	0.5211(0.0303)	0.9559(0.0247)	0.4999(0.0003)	0.9662(0.0033)	0.511(0.0249)
Accuracy	0.9869(0.0014)	0.9345(0.006)	0.9885(0.0006)	0.9332(0.0065)	0.9887(0.0009)

Table 20.10: MUSC - Testing

CHD CNS	Bagging	BalancedBagging	BootstrapRandomForest	EasyEnsemble	WeightedRandomForest
F2	0.7455(0.0492)	0.7345(0.005)	0.1823(0.033)	0.7327(0.0051)	0.1809(0.0423)
Gmean	0.8466(0.0329)	0.9826(0.0014)	0.3012(0.052)	0.983(0.0004)	0.3014(0.0588)
AUC ROC	0.8648(0.0252)	0.9827(0.0014)	0.5785(0.0144)	0.9832(0.0004)	0.5778(0.0187)
Accuracy	0.9926(0.0015)	0.9674(0.0008)	0.9849(0.0006)	0.967(0.0007)	0.9847(0.0006)

Table 20.11: CHD CNS - Training

CHD CNS	Bagging	BalancedBagging	BootstrapRandomForest	EasyEnsemble	WeightedRandomForest
F2	0.7701(0.086)	0.7314(0.0177)	0.1787(0.0926)	0.7314(0.0177)	0.1759(0.1512)
Gmean	0.872(0.046)	0.9832(0.0015)	0.3747(0.1066)	0.9832(0.0015)	0.362(0.1517)
AUC ROC	0.8805(0.0408)	0.9833(0.0015)	0.5747(0.0399)	0.9833(0.0015)	0.5746(0.0675)
Accuracy	0.9921(0.004)	0.9672(0.0029)	0.9848(0.002)	0.9672(0.0029)	0.9845(0.003)

Table 20.12: CHD CNS - Testing

CHD GU	Bagging	BalancedBagging	BootstrapRandomForest	EasyEnsemble	WeightedRandomForest
F2	0.813(0.0352)	0.7531(0.0095)	0.2345(0.0553)	0.7526(0.0091)	0.2449(0.0554)
Gmean	0.8903(0.0193)	0.9821(0.002)	0.3722(0.0701)	0.983(0.0008)	0.3845(0.0682)
AUC ROC	0.9(0.0175)	0.9822(0.0019)	0.6021(0.0251)	0.9832(0.0008)	0.6065(0.025)
Accuracy	0.994(0.0015)	0.9673(0.0016)	0.9841(0.001)	0.9671(0.0015)	0.9843(0.0011)

Table 20.13: CHD GU - Training

CHD GU	Bagging	BalancedBagging	BootstrapRandomForest	EasyEnsemble	WeightedRandomForest
F2	0.8222(0.1196)	0.747(0.0327)	0.265(0.1147)	0.7498(0.0347)	0.2643(0.1354)
Gmean	0.8982(0.0769)	0.9769(0.014)	0.463(0.1234)	0.983(0.0031)	0.4602(0.136)
AUC ROC	0.9055(0.0685)	0.977(0.0139)	0.6132(0.0505)	0.9832(0.0031)	0.6132(0.0606)
Accuracy	0.994(0.0035)	0.9678(0.0059)	0.9845(0.0019)	0.967(0.006)	0.9845(0.0025)

Table 20.14: CHD GU - Testing

CHD MUSC	Bagging	BalancedBagging	BootstrapRandomForest	EasyEnsemble	WeightedRandomForest
F2	0.7105(0.0498)	0.7589(0.0097)	0.2297(0.0449)	0.7579(0.0091)	0.2236(0.0644)
Gmean	0.8257(0.0313)	0.9827(0.0018)	0.3791(0.0539)	0.9831(0.0008)	0.3683(0.0769)
AUC ROC	0.8463(0.0264)	0.9829(0.0018)	0.5994(0.0204)	0.9832(0.0008)	0.5966(0.0293)
Accuracy	0.9904(0.0017)	0.9675(0.0015)	0.9834(0.001)	0.9672(0.0016)	0.9836(0.0011)

Table 20.15: CHD MUSC - Training

CHD MUSC	Bagging	BalancedBagging	BootstrapRandomForest	EasyEnsemble	WeightedRandomForest
F2	0.7381(0.0941)	0.7716(0.0439)	0.2856(0.1132)	0.755(0.0385)	0.1685(0.1159)
Gmean	0.8476(0.0695)	0.9845(0.0039)	0.4869(0.1039)	0.983(0.0036)	0.3595(0.1247)
AUC ROC	0.8608(0.0564)	0.9846(0.0038)	0.6226(0.0516)	0.9832(0.0035)	0.5707(0.0509)
Accuracy	0.9911(0.0019)	0.9699(0.0075)	0.9843(0.0021)	0.967(0.007)	0.9825(0.0024)

Table 20.16: CHD MUSC - Testing

CNS GU	Bagging	BalancedBagging	BootstrapRandomForest	EasyEnsemble	WeightedRandomForest
F2	0.7277(0.0512)	0.7563(0.0088)	0.2317(0.0365)	0.7522(0.0084)	0.2261(0.0466)
Gmean	0.8375(0.0322)	0.9819(0.002)	0.3845(0.0438)	0.9831(0.0008)	0.3732(0.065)
AUC ROC	0.8572(0.0275)	0.982(0.0019)	0.5998(0.0165)	0.9832(0.0007)	0.5976(0.021)
Accuracy	0.9911(0.0017)	0.968(0.0019)	0.984(0.0008)	0.9671(0.0015)	0.9839(0.0009)

Table 20.17: CNS GU - Training

CNS GU	Bagging	BalancedBagging	BootstrapRandomForest	EasyEnsemble	WeightedRandomForest
F2	0.8034(0.1566)	0.7584(0.0278)	0.2351(0.1353)	0.7498(0.0347)	0.2202(0.1286)
Gmean	0.8879(0.107)	0.9838(0.0024)	0.3993(0.2252)	0.983(0.0031)	0.3854(0.2182)
AUC ROC	0.8985(0.0939)	0.984(0.0024)	0.6(0.0577)	0.9832(0.0031)	0.5933(0.0548)
Accuracy	0.9932(0.0037)	0.9686(0.0047)	0.9843(0.0023)	0.967(0.006)	0.984(0.0021)

Table 20.18: CNS GU - Testing

CNS MUSC	Bagging	BalancedBagging	BootstrapRandomForest	EasyEnsemble	WeightedRandomForest
F2	0.6491(0.0566)	0.7612(0.0114)	0.1736(0.057)	0.7566(0.0102)	0.1799(0.053)
Gmean	0.7842(0.0389)	0.9833(0.0009)	0.3053(0.0893)	0.9825(0.0019)	0.3075(0.0762)
AUC ROC	0.815(0.03)	0.9835(0.0009)	0.5741(0.0248)	0.9827(0.0019)	0.5775(0.0233)
Accuracy	0.9886(0.0018)	0.9676(0.0017)	0.9824(0.0012)	0.9671(0.0017)	0.9826(0.0011)

Table 20.19: CNS MUSC - Training

CNS MUSC	Bagging	BalancedBagging	BootstrapRandomForest	EasyEnsemble	WeightedRandomForest
F2	0.7628(0.1107)	0.7606(0.0321)	0.1247(0.0658)	0.755(0.0385)	0.1716(0.0629)
Gmean	0.8627(0.0789)	0.9836(0.003)	0.3128(0.0824)	0.983(0.0036)	0.372(0.078)
AUC ROC	0.8742(0.065)	0.9837(0.003)	0.5514(0.028)	0.9832(0.0035)	0.5715(0.0271)
Accuracy	0.9919(0.0022)	0.9681(0.0058)	0.9814(0.0017)	0.967(0.007)	0.9825(0.0012)

Table 20.20: CNS MUSC - Testing

GU MUSC	Bagging	BalancedBagging	BootstrapRandomForest	EasyEnsemble	WeightedRandomForest
F2	0.7223(0.0346)	0.7738(0.0072)	0.251(0.0649)	0.7684(0.0046)	0.2741(0.0624)
Gmean	0.8315(0.0229)	0.9822(0.0023)	0.4116(0.0817)	0.9826(0.001)	0.4375(0.0705)
AUC ROC	0.8512(0.0187)	0.9823(0.0022)	0.6086(0.0295)	0.9828(0.001)	0.62(0.0289)
Accuracy	0.9906(0.0011)	0.9683(0.0011)	0.9829(0.001)	0.9673(0.0009)	0.983(0.001)

Table 20.21: GU MUSC - Training

GU MUSC	Bagging	BalancedBagging	BootstrapRandomForest	EasyEnsemble	WeightedRandomForest
F2	0.7823(0.0708)	0.778(0.0256)	0.2529(0.1805)	0.765(0.0222)	0.2407(0.1637)
Gmean	0.8747(0.0437)	0.9842(0.0018)	0.4401(0.1757)	0.983(0.0017)	0.4306(0.1628)
AUC ROC	0.883(0.0386)	0.9844(0.0018)	0.6091(0.0811)	0.9832(0.0017)	0.6032(0.073)
Accuracy	0.9919(0.0037)	0.9694(0.0035)	0.983(0.0037)	0.9671(0.0033)	0.9827(0.0034)

Table 20.22: GU MUSC - Testing

20.2. Interpretation. We can see that for per birth defect sheet have extremely poor performance, F2 score around 0 to 20%. For pairs of defects sheets and all defects sheets the performance is better, but it did not exceed the performance of our top performing models, and have an F2 score of at least 5% to 10% less than the top performing models for both pairs of defects sheet and all defects sheet.

Chapter 21

One Class Classification

In this chapter we perform one-class classification using several methods. One-class classification is a used machine learning method to identify outliers and anomalies. First, we will use One-class Support Vector Machines, then we will use Isolation Forest, then Local Outlier Factor and Elliptical Envelope algorithms. We will use F-measure to assess the performance of our models.

21.1. Results. Below we display the results of one class classification:

SHEET	**Recorded score in %**
ALL	7.8
CHD_CNS	1.12
CHD_GU	6.74
CHD_MUSC	7.72
CNS_GU	3.1
CNS_MUSC	4.92
GU_MUSC	9.24
CHD	0
CNS	0
GU	0
MUSC	0

Table 21.1: Recorded F-measure for One-class SVM

SHEET	Recorded score in %
All	0
CHD_CNS	1.9
CHD_GU	3.08
CHD_MUSC	6.12
CNS_GU	0
CNS_MUSC	1.54
GU_MUSC	4.04
CHD	4.7
GU	8.9
MUSC	0
CNS	0

Table 21.2: Recorded F-measure for Elliptical Envelope Algorithm

SHEET	**Recorded score in %**
All	1.86
CHD_CNS	2.86
CHD_GU	4.86
CHD_MUSC	1.82
CNS_GU	2
CNS_MUSC	0
GU_MUSC	1.82
CHD	11.56
GU	3.08
MUSC	2.1
CNS	4.9

Table 21.3: Recorded F-measure for Isolation Forest Algorithm

SHEET	Recorded score in %
All	14.26
CHD_CNS	13.96
CHD_GU	11.32
CHD_MUSC	16.32
CNS_GU	3.5
CNS_MUSC	9.26
GU_MUSC	9.3
CHD	1.9
GU	0
MUSC	2
CNS	0

Table 21.4: Recorded F-measure for Local Outlier Factor Algorithm

21.2. Interpretation. We can see that none of the used algorithms did not have any acceptable performance, all results were recorded around 0% to a maximum of 16%.

Chapter 22

Limitations

The data that we used to train our models is formed of several set of features: The first set is formed of medical features. The second set is formed of socio-economical features. The third set is formed of air pollution data. All data sets are required for the our models to run. One issue that we faced in our study is under-reporting of the data. Under-reporting affects how precise is the model, better reported data will allow the model to be exposed to more cases and to be ready to do the prediction more precisely. It is a Lebanese case study, where all data was collected on the Lebanese territories.

Chapter 23

Conclusion

In this study, we built tools that perform early prediction for birth defects. We created tools that can be easily used by clinicians to predict birth defects before it happens. We were able to detect several contributors to the decision of our predictive models. Our findings were aligned with the literature, as the top contributors were: consanguinity degree, medical history, folic acid intake before pregnancy. Mother age appeared as one of the top contributors for cardiovascular disease. BMI appeared as a top contributor for gastro-urinary disease. Top ambient air pollution contributors were exposure to PM 2.5. Our tool, requires ambient air pollution features in addition to medical and socio-economical features to predict any instance. Data imbalance is biasing the performance of the models, if we compare the F2 score of the per defect sheet and all defects sheet, which have 1% cases and 5% cases respectively, we can see that F2 score is improved from 59.2% to 93.4%. The techniques that we used improved the performance of the models, for instance; F2 score was improved from 87.3% to 93.4% using the cost sensitive version of logistic regression. One advantage of the use of artificial intelligence, machine learning specifically, is that it allows us to build tools that perform prediction, which is not the case in statistical approach, which finds relationships between variables (for example correlation). Behind the scenes, artificial intelligence uses mathematical and statistical concepts to build these tools, for instance, SHAP, the tool that we used to get the contribution of each feature, uses a mathematical concept called Game Theory to find the contribution of each feature. Results revealed using SHAP were aligned with those derived from the statistical analyses that was done.

Appendix A

Classification Results

We are reporting both cross validation and testing results.

Reported results are mainly the averages and standard deviations of the validation and testing results, where we report the average of the repeated cross validation results [Reported as cross validation results] and the average of testing results [Reported as testing results].

Optimized for best F2 score, best G-mean and best Precision-recall AUC. NOTE: in the tables below "F2" is used for F2 score, "G" for G-mean and "PR – AUC" for Precision/Recall AUC score.

A.1. How To Read The Results. Each table below has two top rows that represent over which metrics we are doing the tuning and the scaling options. For example, for the first table there's "No Scaling" that covers the first three columns (F2, G-mean and PR-Recall). So the first three columns are dedicated for tuning for the best F2, G-mean and PR-Recall, all of them in the without scaling. The second three columns are dedicated for tuning for the same metric but for Normalized scaling, third three columns are used for Power Transform scaling, then Part 2 of the table represents the STD scaling: standardized scaling, and finally the TRD scaling, which represents the mixed standardization and normalization.

Rows represent the score for each metric, for example, the row that starts with "Accuracy" has the scores for accuracy, where the first score is reported for accuracy score for best F2, second cell to the right is accuracy score when tuning for best G-mean etc.

For both testing and training results we are reporting the average and the standard deviation of the results in the following form: average (standard deviation).

ADABOOST	**No Scaling**		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.8671(0.0107)	0.8589(0.0113)	0.8608(0.0055)
Gmean	0.9282(0.0067)	0.9225(0.0072)	0.9229(0.0025)
AUC ROC	0.9314(0.0063)	0.9262(0.0067)	0.9266(0.0025)
Accuracy	0.9873(0.0008)	0.987(0.001)	0.9874(0.001)

Table A.1: ADABOOST - Training

ADABOOST	**No Scaling**		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.8585(0.0169)	0.8595(0.0272)	0.8528(0.0429)
Gmean	0.9236(0.0116)	0.9237(0.016)	0.9203(0.0282)
AUC ROC	0.9261(0.0107)	0.9263(0.015)	0.9233(0.0261)
Accuracy	0.9865(0.0026)	0.9868(0.0032)	0.986(0.0031)

Table A.2: ADABOOST - Testing

ADACOST	**No Scaling**		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.8848(0.0074)	0.8811(0.0035)	0.852(0.0372)
Gmean	0.9687(0.0197)	0.983(0.0005)	0.9197(0.0257)
AUC ROC	0.9698(0.0184)	0.9832(0.0005)	0.9256(0.0226)
Accuracy	0.9755(0.0107)	0.9679(0.001)	0.986(0.0015)

Table A.3: ADACOST - Training

ADACOST	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.8584(0.081)	0.8803(0.0134)	0.8343(0.0902)
Gmean	0.9481(0.0711)	0.983(0.0022)	0.9048(0.0645)
AUC ROC	0.9513(0.0642)	0.9832(0.0021)	0.9107(0.0578)
Accuracy	0.9758(0.0105)	0.9679(0.0041)	0.9865(0.0026)

Table A.4: ADACOST - Testing

CATBOOST	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.8759(0.019)	0.8755(0.0169)	0.8803(0.0202)
Gmean	0.9317(0.0123)	0.9315(0.0103)	0.9357(0.0122)
AUC ROC	0.9351(0.0114)	0.9345(0.0098)	0.9384(0.0115)
Accuracy	0.9886(0.0012)	0.9887(0.0014)	0.9885(0.0016)

Table A.5: CATBOOST - Training

CATBOOST	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.88(0.0363)	0.88(0.0363)	0.88(0.0363)
Gmean	0.9356(0.022)	0.9356(0.022)	0.9356(0.022)
AUC ROC	0.9375(0.0207)	0.9375(0.0207)	0.9375(0.0207)
Accuracy	0.9885(0.0032)	0.9885(0.0032)	0.9885(0.0032)

Table A.6: CATBOOST - Testing

DT	**No Scaling**		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.8619(0.0091)	0.8617(0.0152)	0.8682(0.0106)
Gmean	0.925(0.0051)	0.9244(0.0087)	0.9283(0.0066)
AUC ROC	0.9286(0.0046)	0.928(0.0081)	0.9319(0.0062)
Accuracy	0.9869(0.0009)	0.9871(0.0015)	0.9876(0.001)

Table A.7: Decision Tree - Training

DT	**No Scaling**		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.8559(0.0335)	0.8505(0.0197)	0.8524(0.0299)
Gmean	0.9209(0.0198)	0.9179(0.0128)	0.9181(0.017)
AUC ROC	0.9237(0.0184)	0.9209(0.0117)	0.9211(0.0159)
Accuracy	0.9868(0.0037)	0.9863(0.0028)	0.9868(0.0034)

Table A.8: Decision Tree - Testing

DT CS	**No Scaling**		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.8784(0.0058)	0.8775(0.0095)	0.8753(0.0051)
Gmean	0.9378(0.0049)	0.9381(0.0057)	0.9329(0.0032)
AUC ROC	0.9403(0.0046)	0.9405(0.0054)	0.9359(0.0029)
Accuracy	0.9869(0.0011)	0.9865(0.0009)	0.9879(0.001)

Table A.9: Cost Sensitive Decision Tree - Training

DT CS	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.8522(0.0369)	0.8651(0.0589)	0.8638(0.0382)
Gmean	0.9227(0.0199)	0.931(0.0352)	0.9265(0.0232)
AUC ROC	0.9252(0.0186)	0.9333(0.0334)	0.929(0.0217)
Accuracy	0.9847(0.0044)	0.9855(0.0048)	0.987(0.0033)

Table A.10: Cost Sensitive Decision Tree - Testing

DT HELLINGER	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.885(0.0076)	0.8811(0.0034)	0.8713(0.0092)
Gmean	0.9758(0.0164)	0.983(0.0005)	0.9313(0.0057)
AUC ROC	0.9762(0.0157)	0.9832(0.0005)	0.9342(0.0053)
Accuracy	0.9724(0.0097)	0.9679(0.001)	0.9873(0.001)

Table A.11: Decision Tree HELLINGER Distance - Training

DT HELLINGER	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.8754(0.0137)	0.8803(0.0134)	0.8678(0.0245)
Gmean	0.9715(0.025)	0.983(0.0022)	0.9248(0.0159)
AUC ROC	0.9721(0.0241)	0.9832(0.0021)	0.9275(0.0146)
Accuracy	0.9712(0.0097)	0.9679(0.0041)	0.9891(0.0021)

Table A.12: Decision Tree HELLINGER Distance - Testing

DT SMOTENC	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.8766(0.0082)	0.8667(0.0123)	0.8817(0.015)
Gmean	0.9354(0.0061)	0.9291(0.0082)	0.9376(0.0092)
AUC ROC	0.9382(0.0059)	0.9325(0.0074)	0.9402(0.0085)
Accuracy	0.9874(0.0003)	0.9867(0.0008)	0.9881(0.0012)

Table A.13: Decision Tree - SMOTENC - Training

DT SMOTENC	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.8466(0.0393)	0.8743(0.0501)	0.8465(0.0412)
Gmean	0.9173(0.0245)	0.9346(0.0307)	0.9172(0.0265)
AUC ROC	0.9203(0.0228)	0.9367(0.0284)	0.9203(0.0246)
Accuracy	0.9852(0.0032)	0.987(0.0045)	0.9852(0.0031)

Table A.14: Decision Tree - SMOTENC - Testing

DT SMOTETOMEK	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.9085(0.0052)	0.9197(0.0071)	0.9108(0.0182)
Gmean	0.9609(0.0017)	0.9676(0.0023)	0.9589(0.013)
AUC ROC	0.9619(0.0017)	0.9683(0.0024)	0.9601(0.012)
Accuracy	0.9872(0.0015)	0.9882(0.0019)	0.9888(0.0012)

Table A.15: Decision Tree - SMOTETOMEK - Training

DT SMOTETOMEK	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.9205(0.0195)	0.9164(0.0221)	0.908(0.0476)
Gmean	0.9701(0.0113)	0.9674(0.0104)	0.9594(0.0283)
AUC ROC	0.9703(0.011)	0.9676(0.0102)	0.9602(0.027)
Accuracy	0.9875(0.0033)	0.9873(0.0037)	0.9878(0.0041)

Table A.16: Decision Tree - SMOTETOMEK - Testing

KNN	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.1746(0.0253)	0.1762(0.0198)	0.1164(0.0401)	0.7101(0.0135)	0.7223(0.0186)	0.7111(0.014)
Gmean	0.381(0.0379)	0.3846(0.0278)	0.2641(0.0808)	0.8271(0.0089)	0.8366(0.0125)	0.8262(0.0074)
AUC ROC	0.5686(0.0132)	0.5693(0.01)	0.5467(0.016)	0.8436(0.0069)	0.851(0.0106)	0.8426(0.0061)
Accuracy	0.9275(0.0026)	0.9286(0.0031)	0.9473(0.011)	0.9768(0.0016)	0.977(0.0015)	0.9777(0.0024)

Table A.17: KNN - Training - Part 1

KNN	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.5636(0.0218)	0.5651(0.0232)	0.5694(0.012)	0.5704(0.0142)	0.572(0.0217)	0.5668(0.0071)
Gmean	0.7288(0.0154)	0.7292(0.0154)	0.7325(0.0088)	0.7331(0.009)	0.7344(0.0152)	0.7298(0.005)
AUC ROC	0.7663(0.0109)	0.7666(0.0119)	0.7686(0.0061)	0.7692(0.0064)	0.7703(0.0109)	0.7679(0.0031)
Accuracy	0.9658(0.002)	0.9662(0.0024)	0.9664(0.0011)	0.9667(0.0018)	0.9667(0.002)	0.9665(0.0012)

Table A.18: KNN - Training - Part 2

KNN	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.5377(0.0498)	0.5338(0.061)	0.5349(0.0512)
Gmean	0.7032(0.036)	0.6997(0.0457)	0.7007(0.0382)
AUC ROC	0.7491(0.0259)	0.7473(0.0318)	0.7477(0.027)
Accuracy	0.9683(0.0029)	0.9681(0.0033)	0.9681(0.0028)

Table A.19: KNN - Training - Part 3

KNN	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.1663(0.038)	0.1663(0.038)	0.1044(0.0751)	0.7193(0.1102)	0.7391(0.1135)	0.7065(0.1231)
Gmean	0.3932(0.0485)	0.3932(0.0485)	0.2816(0.119)	0.8322(0.0728)	0.8474(0.0769)	0.8218(0.0863)
AUC ROC	0.5636(0.0192)	0.5636(0.0192)	0.5414(0.0315)	0.8471(0.0591)	0.8601(0.0623)	0.8395(0.0675)
Accuracy	0.9272(0.0069)	0.9272(0.0069)	0.9486(0.0073)	0.9779(0.0078)	0.9781(0.0077)	0.9781(0.0066)

Table A.20: KNN - Testing - Part 1

KNN	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.5736(0.0455)	0.5736(0.0455)	0.5736(0.0455)	0.5853(0.053)	0.5853(0.053)	0.5853(0.053)
Gmean	0.7374(0.031)	0.7374(0.031)	0.7374(0.031)	0.7444(0.0418)	0.7444(0.0418)	0.7444(0.0418)
AUC ROC	0.7694(0.0233)	0.7694(0.0233)	0.7694(0.0233)	0.7753(0.031)	0.7753(0.031)	0.7753(0.031)
Accuracy	0.9669(0.0043)	0.9669(0.0043)	0.9669(0.0043)	0.9684(0.0021)	0.9684(0.0021)	0.9684(0.0021)

Table A.21: KNN - Testing - Part 2

KNN	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.2806(0.2084)	0.2806(0.2084)	0.2806(0.2084)
Gmean	0.5029(0.3331)	0.5029(0.3331)	0.5029(0.3331)
AUC ROC	0.6606(0.1329)	0.6606(0.1329)	0.6606(0.1329)
Accuracy	0.6077(0.3277)	0.6077(0.3277)	0.6077(0.3277)

Table A.22: KNN - Testing - Part 3

LR	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.8007(0.0261)	0.8002(0.0201)	0.7982(0.0249)	0.8805(0.0137)	0.879(0.0097)	0.8791(0.0132)
Gmean	0.8881(0.0168)	0.8881(0.0139)	0.8872(0.0158)	0.9376(0.0082)	0.9364(0.0054)	0.9369(0.0082)
AUC ROC	0.8953(0.015)	0.8954(0.0123)	0.8942(0.0141)	0.9403(0.0074)	0.939(0.0051)	0.9396(0.0076)
Accuracy	0.9819(0.0019)	0.9817(0.0011)	0.9816(0.0018)	0.9878(0.0011)	0.9878(0.001)	0.9875(0.0009)

Table A.23: Logistic Regression - Training - Part 1

LR	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.8818(0.0161)	0.8798(0.0167)	0.8735(0.021)	0.8811(0.0206)	0.8791(0.0224)	0.88(0.0185)
Gmean	0.937(0.01)	0.9356(0.0101)	0.9308(0.0146)	0.938(0.0124)	0.9368(0.0136)	0.9379(0.0115)
AUC ROC	0.9394(0.0093)	0.9383(0.0093)	0.934(0.0132)	0.9403(0.0116)	0.9394(0.0129)	0.9404(0.0109)
Accuracy	0.9885(0.0011)	0.9883(0.0013)	0.9883(0.0009)	0.9878(0.0017)	0.9877(0.0018)	0.9875(0.0014)

Table A.24: Logistic Regression - Training - Part 2

LR	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.8851(0.0085)	0.8846(0.0101)	0.883(0.014)
Gmean	0.9405(0.0049)	0.94(0.0062)	0.9385(0.009)
AUC ROC	0.9428(0.0047)	0.9423(0.0058)	0.941(0.0084)
Accuracy	0.988(0.0009)	0.9881(0.0009)	0.9882(0.0009)

Table A.25: Logistic Regression - Training - Part 3

LR	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.8151(0.0196)	0.8105(0.0183)	0.8105(0.0183)	0.8728(0.0272)	0.8683(0.0296)	0.8763(0.0285)
Gmean	0.8967(0.0142)	0.8937(0.0153)	0.8937(0.0153)	0.9324(0.0164)	0.9295(0.0186)	0.9352(0.0166)
AUC ROC	0.9014(0.0128)	0.8987(0.0136)	0.8987(0.0136)	0.9344(0.0154)	0.9317(0.0175)	0.9369(0.0156)
Accuracy	0.9835(0.0027)	0.9832(0.0023)	0.9832(0.0023)	0.9875(0.0033)	0.9873(0.0032)	0.9875(0.0033)

Table A.26: Logistic Regression - Testing - Part 1

LR	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.8513(0.0386)	0.8523(0.0388)	0.8574(0.048)	0.8709(0.0397)	0.8709(0.0397)	0.8709(0.0397)
Gmean	0.9179(0.0229)	0.9181(0.0229)	0.9209(0.0305)	0.9321(0.0223)	0.9321(0.0223)	0.9321(0.0223)
AUC ROC	0.921(0.0211)	0.9211(0.0211)	0.924(0.0282)	0.9341(0.0213)	0.9341(0.0213)	0.9341(0.0213)
Accuracy	0.9865(0.0034)	0.9868(0.0033)	0.9873(0.0034)	0.987(0.0038)	0.987(0.0038)	0.987(0.0038)

Table A.27: Logistic Regression - Testing - Part 2

LR	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.7199(0.1436)	0.78(0.143)	0.6829(0.0903)
Gmean	0.8313(0.1132)	0.8796(0.1114)	0.8041(0.0647)
AUC ROC	0.8504(0.0967)	0.8913(0.095)	0.8241(0.054)
Accuracy	0.9794(0.0023)	0.9789(0.0019)	0.9781(0.005)

Table A.28: Logistic Regression - Testing - Part 3

LR CS	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.8806(0.0164)	0.8756(0.0191)	0.8733(0.0248)	0.9346(0.0055)	0.9336(0.0061)	0.8898(0.0301)
Gmean	0.9553(0.0109)	0.9511(0.0114)	0.946(0.015)	0.9896(0.0029)	0.99(0.0017)	0.9464(0.025)
AUC ROC	0.9563(0.0103)	0.9525(0.0107)	0.9477(0.0141)	0.9897(0.0029)	0.9901(0.0017)	0.9485(0.0238)
Accuracy	0.9803(0.0014)	0.9803(0.0019)	0.9817(0.0023)	0.984(0.0012)	0.9835(0.0016)	0.9871(0.0012)

Table A.29: Cost Sensitive Logistic Regression - Training - Part 1

LR CS	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.9306(0.0057)	0.9164(0.0104)	0.9058(0.0194)	0.9194(0.0174)	0.9128(0.0163)	0.8954(0.0327)
Gmean	0.9797(0.0036)	0.9831(0.0034)	0.9559(0.0164)	0.9736(0.0076)	0.976(0.0054)	0.95(0.0242)
AUC ROC	0.9799(0.0035)	0.9832(0.0034)	0.9573(0.0156)	0.9741(0.0074)	0.9763(0.0052)	0.9519(0.0228)
Accuracy	0.9868(0.0022)	0.9806(0.0022)	0.9884(0.0012)	0.9856(0.0028)	0.9823(0.0041)	0.9875(0.0013)

Table A.30: Cost Sensitive Logistic Regression - Training - Part 2

LR CS	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.9339(0.005)	0.9327(0.007)	0.8824(0.0144)
Gmean	0.9881(0.0029)	0.9894(0.0022)	0.9382(0.0091)
AUC ROC	0.9882(0.0028)	0.9894(0.0022)	0.9409(0.0082)
Accuracy	0.9844(0.0011)	0.9835(0.0018)	0.9881(0.001)

Table A.31: Cost Sensitive Logistic Regression - Training - Part 3

LR CS	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.8742(0.0581)	0.8776(0.0549)	0.871(0.0681)	0.936(0.0179)	0.9322(0.0165)	0.8957(0.0505)
Gmean	0.9501(0.0361)	0.9528(0.0338)	0.9451(0.0414)	0.9894(0.0072)	0.9889(0.007)	0.9486(0.0366)
AUC ROC	0.9512(0.0348)	0.9537(0.0327)	0.9466(0.0398)	0.9894(0.0073)	0.9889(0.007)	0.9502(0.0351)
Accuracy	0.9804(0.0054)	0.9804(0.0054)	0.9814(0.006)	0.9847(0.003)	0.9837(0.0028)	0.9883(0.0037)

Table A.32: Cost Sensitive Logistic Regression - Testing - Part 1

LR CS	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.906(0.0342)	0.9022(0.0214)	0.8936(0.0209)	0.897(0.0463)	0.8678(0.0522)	0.8804(0.0374)
Gmean	0.9636(0.0225)	0.974(0.0119)	0.9487(0.015)	0.9601(0.0291)	0.945(0.0299)	0.9401(0.0226)
AUC ROC	0.9641(0.022)	0.974(0.0118)	0.9498(0.0143)	0.9608(0.0283)	0.946(0.0291)	0.9417(0.0214)
Accuracy	0.9855(0.0026)	0.9799(0.0026)	0.9875(0.0039)	0.984(0.0038)	0.9804(0.0057)	0.9868(0.004)

Table A.33: Cost Sensitive Logistic Regression - Testing - Part 2

LR CS	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.6973(0.1358)	0.7005(0.1362)	0.6772(0.095)
Gmean	0.8135(0.1051)	0.8165(0.1055)	0.7975(0.0726)
AUC ROC	0.835(0.0909)	0.8375(0.091)	0.8196(0.059)
Accuracy	0.9794(0.0017)	0.9791(0.0019)	0.9794(0.0017)

Table A.34: Cost Sensitive Logistic Regression - Testing - Part 3

LR SMOTENC	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.8506(0.019)	0.861(0.0128)	0.8557(0.0129)	0.9217(0.0057)	0.9199(0.0109)	0.9179(0.0093)
Gmean	0.9294(0.0146)	0.937(0.011)	0.9338(0.0103)	0.9797(0.0044)	0.9792(0.0051)	0.9758(0.0076)
AUC ROC	0.9323(0.0134)	0.9392(0.0102)	0.9361(0.0096)	0.98(0.0043)	0.9795(0.0049)	0.9762(0.0073)
Accuracy	0.9811(0.0013)	0.9813(0.0012)	0.9809(0.0011)	0.9838(0.0019)	0.9834(0.0019)	0.9843(0.0004)

Table A.35: Logistic Regression - SMOTENC - Training - Part 1

LR SMOTENC	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.9173(0.014)	0.9158(0.0065)	0.9128(0.0192)	0.9062(0.0146)	0.9045(0.0075)	0.9035(0.0132)
Gmean	0.9724(0.0034)	0.9766(0.0052)	0.9642(0.0127)	0.9663(0.0062)	0.9694(0.0064)	0.96(0.0091)
AUC ROC	0.9729(0.0034)	0.9768(0.0051)	0.9651(0.0121)	0.967(0.0058)	0.9699(0.0062)	0.9611(0.0087)
Accuracy	0.9853(0.0033)	0.9832(0.0012)	0.9873(0.0014)	0.9843(0.0025)	0.9824(0.0005)	0.986(0.0011)

Table A.36: Logistic Regression - SMOTENC - Training - Part 2

LR SMOTENC	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.9259(0.0078)	0.9249(0.0078)	0.9248(0.0081)
Gmean	0.9842(0.003)	0.9862(0.0035)	0.9807(0.0053)
AUC ROC	0.9843(0.003)	0.9863(0.0035)	0.9809(0.0052)
Accuracy	0.9833(0.002)	0.9821(0.0013)	0.9844(0.0008)

Table A.37: Logistic Regression - SMOTENC - Training - Part 3

LR SMOTENC	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.8369(0.075)	0.8369(0.075)	0.8414(0.0688)	0.9338(0.0157)	0.9211(0.0231)	0.925(0.028)
Gmean	0.9199(0.0482)	0.9199(0.0482)	0.9229(0.044)	0.987(0.0074)	0.9787(0.0162)	0.9793(0.0167)
AUC ROC	0.9231(0.0455)	0.9231(0.0455)	0.9258(0.0419)	0.987(0.0074)	0.9789(0.0159)	0.9794(0.0165)
Accuracy	0.9807(0.0053)	0.9807(0.0053)	0.9809(0.0049)	0.985(0.0033)	0.9842(0.0026)	0.9852(0.0037)

Table A.38: Logistic Regression - SMOTENC - Testing - Part 1

LR SMOTENC	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.9349(0.0104)	0.9306(0.0158)	0.9116(0.035)	0.8983(0.0513)	0.911(0.0285)	0.8935(0.0505)
Gmean	0.9828(0.0096)	0.9843(0.0103)	0.9643(0.0244)	0.9623(0.0334)	0.973(0.0159)	0.9529(0.0316)
AUC ROC	0.9829(0.0095)	0.9844(0.0102)	0.9649(0.0235)	0.9631(0.0321)	0.9732(0.0157)	0.954(0.0304)
Accuracy	0.987(0.0036)	0.985(0.0051)	0.987(0.0029)	0.9835(0.0043)	0.9832(0.004)	0.9857(0.0045)

Table A.39: Logistic Regression - SMOTENC - Testing - Part 2

LR SMOTENC	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.8665(0.1111)	0.8923(0.0183)	0.7878(0.1561)
Gmean	0.9484(0.0898)	0.9849(0.0029)	0.8872(0.1187)
AUC ROC	0.9529(0.0798)	0.985(0.0028)	0.8986(0.1024)
Accuracy	0.9789(0.0032)	0.9715(0.0054)	0.9781(0.0024)

Table A.40: Logistic Regression - SMOTENC - Testing - Part 3

LR SMOTETOMEK	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.8571(0.0137)	0.8609(0.0101)	0.8604(0.0119)	0.9326(0.0058)	0.9279(0.0064)	0.9323(0.0068)
Gmean	0.9345(0.0077)	0.9368(0.0058)	0.9368(0.0072)	0.9856(0.0043)	0.9872(0.0034)	0.9848(0.0042)
AUC ROC	0.9368(0.0073)	0.939(0.0056)	0.939(0.0069)	0.9857(0.0042)	0.9873(0.0034)	0.985(0.0041)
Accuracy	0.9811(0.0015)	0.9814(0.0011)	0.9813(0.0012)	0.985(0.0013)	0.9827(0.0007)	0.9852(0.0007)

Table A.41: Logistic Regression - SMOTETOMEK - Training - Part 1

LR SMOTETOMEK	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.9238(0.0061)	0.9228(0.0049)	0.914(0.0148)	0.9122(0.0136)	0.9114(0.01)	0.9091(0.0165)
Gmean	0.9752(0.0026)	0.9787(0.0036)	0.9638(0.0096)	0.9691(0.0053)	0.9714(0.0054)	0.9626(0.0099)
AUC ROC	0.9757(0.0024)	0.979(0.0035)	0.9647(0.0091)	0.9697(0.0052)	0.9719(0.0052)	0.9636(0.0095)
Accuracy	0.9864(0.0025)	0.9846(0.0006)	0.9879(0.001)	0.9851(0.0029)	0.9838(0.0021)	0.9868(0.0014)

Table A.42: Logistic Regression - SMOTETOMEK - Training - Part 2

LR SMOTETOMEK	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.9319(0.0074)	0.9306(0.0061)	0.9329(0.0063)
Gmean	0.9843(0.0045)	0.9852(0.0043)	0.9838(0.003)
AUC ROC	0.9845(0.0044)	0.9854(0.0042)	0.9839(0.003)
Accuracy	0.9853(0.0012)	0.9845(0.0013)	0.9858(0.0009)

Table A.43: Logistic Regression - SMOTETOMEK - Training - Part 3

LR SMOTETOMEK	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.8521(0.0555)	0.8512(0.0558)	0.8521(0.0555)	0.9296(0.0181)	0.9321(0.0024)	0.9325(0.0204)
Gmean	0.9292(0.0361)	0.9291(0.0361)	0.9292(0.0361)	0.9843(0.0078)	0.9888(0.0047)	0.9847(0.0085)
AUC ROC	0.9314(0.034)	0.9313(0.034)	0.9314(0.034)	0.9843(0.0078)	0.9889(0.0047)	0.9847(0.0086)
Accuracy	0.9819(0.0035)	0.9817(0.0038)	0.9819(0.0035)	0.9847(0.0035)	0.9837(0.0021)	0.9855(0.0033)

Table A.44: Logistic Regression - SMOTETOMEK - Testing - Part 1

LR SMOTETOMEK	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.9252(0.0199)	0.927(0.0194)	0.9028(0.0412)	0.8867(0.0539)	0.9013(0.0422)	0.8849(0.0487)
Gmean	0.9751(0.012)	0.9796(0.0098)	0.9566(0.0237)	0.9519(0.0355)	0.9629(0.0269)	0.9474(0.0298)
AUC ROC	0.9752(0.0119)	0.9797(0.0097)	0.9574(0.0229)	0.953(0.0342)	0.9635(0.0261)	0.9486(0.0285)
Accuracy	0.987(0.0029)	0.9857(0.0034)	0.9873(0.0042)	0.984(0.0038)	0.9842(0.0033)	0.9852(0.0039)

Table A.45: Logistic Regression - SMOTETOMEK - Testing - Part 2

LR SMOTETOMEK	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.8393(0.1553)	0.8216(0.1477)	0.6936(0.1374)
Gmean	0.931(0.1205)	0.9282(0.1189)	0.813(0.1053)
AUC ROC	0.939(0.1028)	0.9362(0.1012)	0.8344(0.0912)
Accuracy	0.9768(0.003)	0.9715(0.0092)	0.9781(0.0033)

Table A.46: Logistic Regression - SMOTETOMEK - Testing - Part 3

SVM	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.8066(0.0425)	0.8065(0.0372)	0.8098(0.0404)	0.8568(0.0267)	0.8581(0.0249)	0.8607(0.0233)
Gmean	0.901(0.0277)	0.8993(0.0235)	0.9016(0.0259)	0.9234(0.0189)	0.9248(0.0181)	0.9266(0.0177)
AUC ROC	0.9066(0.0249)	0.9052(0.021)	0.9075(0.0232)	0.9274(0.0173)	0.9285(0.0165)	0.93(0.0159)
Accuracy	0.9781(0.0029)	0.9789(0.0028)	0.9788(0.0031)	0.9858(0.0014)	0.9858(0.0011)	0.9858(0.0013)

Table A.47: SVM - Training - Part 1

SVM	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.8546(0.0243)	0.8566(0.0215)	0.8492(0.0263)	0.8544(0.0334)	0.8501(0.0214)	0.855(0.025)
Gmean	0.9226(0.0153)	0.9237(0.014)	0.9196(0.0167)	0.9255(0.0198)	0.9232(0.0132)	0.9262(0.0155)
AUC ROC	0.9262(0.014)	0.9274(0.0128)	0.9238(0.015)	0.9289(0.0182)	0.9268(0.0124)	0.9297(0.0141)
Accuracy	0.9855(0.0017)	0.9857(0.0013)	0.9849(0.0019)	0.9841(0.003)	0.9837(0.0017)	0.984(0.002)

Table A.48: SVM - Training - Part 2

SVM	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.8651(0.013)	0.8698(0.0136)	0.8677(0.0207)
Gmean	0.9289(0.0089)	0.9316(0.01)	0.9297(0.0146)
AUC ROC	0.9323(0.0083)	0.9349(0.0091)	0.9328(0.0134)
Accuracy	0.9863(0.0009)	0.9867(0.001)	0.9869(0.0008)

Table A.49: SVM - Training - Part 3

SVM	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.8133(0.0871)	0.8133(0.0871)	0.8133(0.0871)	0.8654(0.0296)	0.8645(0.031)	0.8556(0.0421)
Gmean	0.9025(0.054)	0.9025(0.054)	0.9025(0.054)	0.9314(0.0169)	0.9313(0.0171)	0.9254(0.0238)
AUC ROC	0.9072(0.0485)	0.9072(0.0485)	0.9072(0.0485)	0.9333(0.016)	0.9332(0.0162)	0.9278(0.0223)
Accuracy	0.9799(0.0073)	0.9799(0.0073)	0.9799(0.0073)	0.9855(0.0031)	0.9852(0.0034)	0.9847(0.0043)

Table A.50: SVM - Testing - Part 1

SVM	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.859(0.042)	0.859(0.042)	0.859(0.042)	0.84(0.0445)	0.84(0.0445)	0.84(0.0445)
Gmean	0.9258(0.027)	0.9258(0.027)	0.9258(0.027)	0.9163(0.03)	0.9163(0.03)	0.9163(0.03)
AUC ROC	0.9283(0.0253)	0.9283(0.0253)	0.9283(0.0253)	0.9194(0.0278)	0.9194(0.0278)	0.9194(0.0278)
Accuracy	0.9857(0.003)	0.9857(0.003)	0.9857(0.003)	0.9835(0.0024)	0.9835(0.0024)	0.9835(0.0024)

Table A.51: SVM - Testing - Part 2

SVM	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.4983(0.3408)	0.6833(0.2383)	0.3819(0.2781)
Gmean	0.6745(0.3919)	0.8725(0.1255)	0.6005(0.3527)
AUC ROC	0.7879(0.1838)	0.8859(0.1063)	0.729(0.1508)
Accuracy	0.8552(0.2007)	0.8608(0.2045)	0.743(0.2701)

Table A.52: SVM - Testing - Part 3

SVM CS	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.8428(0.0269)	0.8406(0.0252)	0.8321(0.0259)	0.9277(0.0054)	0.9257(0.0053)	0.9003(0.0352)
Gmean	0.9255(0.0158)	0.924(0.0151)	0.917(0.0155)	0.9861(0.005)	0.9862(0.0027)	0.9601(0.0316)
AUC ROC	0.9287(0.0145)	0.9269(0.0137)	0.9211(0.014)	0.9862(0.0049)	0.9863(0.0026)	0.9616(0.0294)
Accuracy	0.9801(0.0027)	0.9799(0.0025)	0.9801(0.0028)	0.9831(0.001)	0.9824(0.0007)	0.9849(0.0025)

Table A.53: Cost Sensitive SVM - Training - Part 1

SVM CS	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.8933(0.0213)	0.867(0.0391)	0.8881(0.0164)	0.8723(0.0205)	0.8549(0.0291)	0.8758(0.0138)
Gmean	0.9516(0.0134)	0.9445(0.0111)	0.9475(0.0104)	0.9388(0.0101)	0.9481(0.0091)	0.9416(0.0069)
AUC ROC	0.9531(0.0127)	0.9462(0.011)	0.9492(0.0098)	0.9411(0.0094)	0.9494(0.0089)	0.9436(0.0068)
Accuracy	0.9861(0.0017)	0.9798(0.0093)	0.9861(0.0014)	0.9844(0.0028)	0.974(0.0079)	0.9845(0.0019)

Table A.54: Cost Sensitive SVM - Training - Part 2

SVM CS	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.9212(0.0062)	0.9197(0.0089)	0.8818(0.0357)
Gmean	0.9777(0.0036)	0.9785(0.0042)	0.9415(0.0273)
AUC ROC	0.978(0.0035)	0.9788(0.0041)	0.944(0.0254)
Accuracy	0.9845(0.002)	0.9837(0.0014)	0.9866(0.0009)

Table A.55: Cost Sensitive SVM - Training - Part 3

SVM CS	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.8172(0.0702)	0.8172(0.0702)	0.8166(0.0713)	0.9219(0.0068)	0.9284(0.0093)	0.8774(0.0499)
Gmean	0.9078(0.0487)	0.9078(0.0487)	0.9056(0.0467)	0.9831(0.0103)	0.9883(0.0053)	0.9483(0.0378)
AUC ROC	0.912(0.0441)	0.912(0.0441)	0.9098(0.0422)	0.9832(0.0102)	0.9883(0.0053)	0.9497(0.036)
Accuracy	0.9791(0.0045)	0.9791(0.0045)	0.9799(0.005)	0.9827(0.0026)	0.9827(0.0026)	0.9824(0.0023)

Table A.56: Cost Sensitive SVM - Testing - Part 1

SVM CS	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.8758(0.0256)	0.8742(0.0248)	0.8758(0.0256)	0.8654(0.0711)	0.8378(0.0189)	0.8591(0.0745)
Gmean	0.9396(0.0147)	0.9482(0.0136)	0.9396(0.0147)	0.9353(0.0438)	0.9383(0.0257)	0.9298(0.0471)
AUC ROC	0.941(0.014)	0.949(0.0129)	0.941(0.014)	0.9375(0.0417)	0.9395(0.0239)	0.9325(0.0447)
Accuracy	0.9855(0.0025)	0.9812(0.0073)	0.9855(0.0025)	0.9837(0.0056)	0.9728(0.0056)	0.984(0.0056)

Table A.57: Cost Sensitive SVM - Testing - Part 2

SVM CS	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0397(0.0887)	0.0(0.0)	0.259(0.202)
Gmean	0.0(0.0)	0.0(0.0)	0.3688(0.3438)
AUC ROC	0.5(0.0)	0.5(0.0)	0.6152(0.1136)
Accuracy	0.7718(0.4051)	0.9529(0.0)	0.5359(0.4115)

Table A.58: Cost Sensitive SVM - Testing - Part 3

RF	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.7069(0.0172)	0.7041(0.0081)	0.7093(0.021)
Gmean	0.8156(0.0114)	0.8139(0.0055)	0.8172(0.0145)
AUC ROC	0.8352(0.0104)	0.8338(0.0047)	0.8365(0.0114)
Accuracy	0.9813(0.0009)	0.981(0.0007)	0.9814(0.001)

Table A.59: Random Forest - Training

RF	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.7274(0.0645)	0.6788(0.042)	0.6635(0.0255)
Gmean	0.8297(0.0428)	0.7997(0.0301)	0.7898(0.018)
AUC ROC	0.8445(0.0352)	0.8194(0.0236)	0.8111(0.0143)
Accuracy	0.9827(0.0044)	0.9789(0.0028)	0.9779(0.0037)

Table A.60: Random Forest - Testing

RF CS	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.865(0.0121)	0.8612(0.0107)	0.8441(0.0067)
Gmean	0.9465(0.0075)	0.9437(0.0058)	0.921(0.0043)
AUC ROC	0.9481(0.007)	0.9453(0.0055)	0.9247(0.0036)
Accuracy	0.9786(0.0018)	0.9785(0.0014)	0.9825(0.001)

Table A.61: Cost Sensitive Random Forest - Training

RF CS	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.8602(0.0415)	0.8304(0.0623)	0.8456(0.0528)
Gmean	0.9415(0.0322)	0.9216(0.0331)	0.9216(0.0318)
AUC ROC	0.9429(0.0309)	0.9239(0.0312)	0.9243(0.0299)
Accuracy	0.9794(0.0035)	0.9773(0.008)	0.9829(0.005)

Table A.62: Cost Sensitive Random Forest - Testing

SVM SMOTENC	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.158(0.02)	0.1498(0.0118)	0.1555(0.0181)	0.8842(0.0049)	0.8847(0.0058)	0.8825(0.0057)
Gmean	0.3473(0.0258)	0.3386(0.0221)	0.3558(0.0272)	0.9559(0.0033)	0.956(0.0034)	0.9545(0.0039)
AUC ROC	0.5626(0.0099)	0.5586(0.0062)	0.5608(0.0092)	0.9569(0.0032)	0.9569(0.0033)	0.9557(0.0037)
Accuracy	0.9381(0.0049)	0.9386(0.0047)	0.9377(0.0055)	0.9812(0.0005)	0.9813(0.0008)	0.9812(0.0006)

Table A.63: SVM SMOTENC - Training - Part 1

SVM SMOTENC	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.8008(0.0058)	0.807(0.0082)	0.8016(0.0106)	0.7534(0.018)	0.749(0.0179)	0.7524(0.0178)
Gmean	0.9088(0.0035)	0.9133(0.0061)	0.9095(0.0083)	0.8749(0.0127)	0.8723(0.0122)	0.8746(0.0116)
AUC ROC	0.9127(0.0028)	0.917(0.0057)	0.9133(0.0073)	0.8826(0.0109)	0.8799(0.0107)	0.882(0.0106)
Accuracy	0.9723(0.0011)	0.9724(0.0011)	0.9722(0.0006)	0.9699(0.0013)	0.9694(0.0011)	0.9698(0.0018)

Table A.64: SVM SMOTENC - Training - Part 2

SVM SMOTENC	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.901(0.0127)	0.8966(0.0071)	0.8972(0.0108)
Gmean	0.9654(0.0074)	0.9629(0.0036)	0.9624(0.0057)
AUC ROC	0.9661(0.0071)	0.9635(0.0034)	0.9632(0.0055)
Accuracy	0.9829(0.0012)	0.9825(0.001)	0.9828(0.0014)

Table A.65: SVM SMOTENC - Training - Part 3

SVM SMOTENC	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.1628(0.0639)	0.1628(0.0639)	0.1628(0.0639)	0.8919(0.0379)	0.8919(0.0379)	0.8919(0.0379)
Gmean	0.3779(0.0787)	0.3779(0.0787)	0.3779(0.0787)	0.9616(0.0262)	0.9616(0.0262)	0.9616(0.0262)
AUC ROC	0.5639(0.0308)	0.5639(0.0308)	0.5639(0.0308)	0.9621(0.0255)	0.9621(0.0255)	0.9621(0.0255)
Accuracy	0.9376(0.008)	0.9376(0.008)	0.9376(0.008)	0.9817(0.0026)	0.9817(0.0026)	0.9817(0.0026)

Table A.66: SVM SMOTENC - Testing - Part 1

SVM SMOTENC	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.794(0.0597)	0.794(0.0597)	0.794(0.0597)	0.7598(0.0722)	0.7598(0.0722)	0.7598(0.0722)
Gmean	0.9026(0.0349)	0.9026(0.0349)	0.9026(0.0349)	0.8763(0.044)	0.8763(0.044)	0.8763(0.044)
AUC ROC	0.9061(0.0323)	0.9061(0.0323)	0.9061(0.0323)	0.8827(0.0397)	0.8827(0.0397)	0.8827(0.0397)
Accuracy	0.9728(0.0071)	0.9728(0.0071)	0.9728(0.0071)	0.9723(0.0068)	0.9723(0.0068)	0.9723(0.0068)

Table A.67: SVM SMOTENC - Testing - Part 2

SVM SMOTENC	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0(0.0)	0.0(0.0)	0.0(0.0)
Gmean	0.0(0.0)	0.0(0.0)	0.0(0.0)
AUC ROC	0.5(0.0)	0.5(0.0)	0.5(0.0)
Accuracy	0.9529(0.0)	0.9529(0.0)	0.9529(0.0)

Table A.68: SVM SMOTENC - Testing - Part 3

SVM SMOTETOMEK	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.1595(0.0178)	0.1541(0.0184)	0.152(0.0205)
Gmean	0.3503(0.0266)	0.3445(0.0251)	0.3397(0.0252)
AUC ROC	0.5623(0.0086)	0.5601(0.0095)	0.5586(0.0108)
Accuracy	0.9361(0.0047)	0.9368(0.0048)	0.9365(0.0057)

Table A.69: SVM - SMOTETOMEK - Training

SVM SMOTETOMEK	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.1711(0.0535)	0.1711(0.0535)	0.1711(0.0535)
Gmean	0.3917(0.0717)	0.3917(0.0717)	0.3917(0.0717)
AUC ROC	0.5679(0.0256)	0.5679(0.0256)	0.5679(0.0256)
Accuracy	0.9353(0.0106)	0.9353(0.0106)	0.9353(0.0106)

Table A.70: SVM - SMOTETOMEK - Testing

XGBoost	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.8874(0.0222)	0.8911(0.0146)	0.8869(0.0115)
Gmean	0.9394(0.013)	0.9421(0.008)	0.9383(0.0058)
AUC ROC	0.942(0.0117)	0.9444(0.0074)	0.9411(0.0055)
Accuracy	0.9892(0.0019)	0.9894(0.0015)	0.9895(0.0014)

Table A.71: XGBoost - Training

XGBoost	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.8789(0.0408)	0.8639(0.0244)	0.889(0.0417)
Gmean	0.9354(0.026)	0.9267(0.0147)	0.9389(0.0275)
AUC ROC	0.9373(0.0246)	0.929(0.0137)	0.9408(0.0258)
Accuracy	0.9883(0.0028)	0.987(0.0025)	0.9901(0.0025)

Table A.72: XGBoost - Testing

XGBoost CS	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.8871(0.019)	0.8913(0.0183)	0.8826(0.0207)
Gmean	0.9398(0.0114)	0.9423(0.0102)	0.9354(0.0124)
AUC ROC	0.9422(0.0105)	0.9448(0.0094)	0.9384(0.0112)
Accuracy	0.989(0.0015)	0.9893(0.0018)	0.9894(0.0017)

Table A.73: Cost Sensitive XGBOOST - Training

XGBoost CS	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.8924(0.0299)	0.8878(0.0432)	0.8713(0.031)
Gmean	0.9441(0.0186)	0.9411(0.0269)	0.9299(0.0165)
AUC ROC	0.9454(0.0177)	0.9427(0.0254)	0.9321(0.0157)
Accuracy	0.9891(0.0023)	0.9888(0.0035)	0.988(0.0033)

Table A.74: Cost Sensitive XGBOOST - Testing

XGBoost SMOTENC	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.9132(0.0178)	0.9153(0.0132)	0.9172(0.0186)
Gmean	0.9615(0.0123)	0.9624(0.0092)	0.9637(0.0122)
AUC ROC	0.9626(0.0117)	0.9634(0.0089)	0.9646(0.0117)
Accuracy	0.9885(0.0011)	0.9889(0.001)	0.9889(0.0013)

Table A.75: XGBoost - SMOTENC - Training

XGBoost SMOTENC	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.9158(0.0557)	0.9158(0.0557)	0.9158(0.0557)
Gmean	0.9648(0.0314)	0.9648(0.0314)	0.9648(0.0314)
AUC ROC	0.9655(0.0305)	0.9655(0.0305)	0.9655(0.0305)
Accuracy	0.988(0.0057)	0.988(0.0057)	0.988(0.0057)

Table A.76: XGBoost - SMOTENC - Testing

XGBoost SMOTETOMEK	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.8986(0.016)	0.897(0.0118)	0.8952(0.014)
Gmean	0.9493(0.0098)	0.9485(0.0074)	0.9463(0.0076)
AUC ROC	0.951(0.0092)	0.9504(0.0068)	0.9482(0.0073)
Accuracy	0.9888(0.0013)	0.9886(0.001)	0.989(0.0014)

Table A.77: XGBoost - SMOTETOMEK - Training

XGBoost SMOTETOMEK	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.8971(0.0571)	0.9015(0.0548)	0.9016(0.0557)
Gmean	0.9489(0.0344)	0.9518(0.0326)	0.9518(0.0328)
AUC ROC	0.9503(0.0331)	0.953(0.0314)	0.953(0.0316)
Accuracy	0.9885(0.0045)	0.9888(0.0044)	0.9888(0.0046)

Table A.78: XGBoost SMOTETOMEK - Testing

NN	No Scaling			NORM		
Optimized For	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.3434(0.0431)	0.3003(0.0606)	0.2857(0.043)	0.8282(0.0244)	0.8234(0.0088)	0.8338(0.0137)
Gmean	0.5066(0.0564)	0.4521(0.0809)	0.4363(0.0567)	0.9046(0.0158)	0.9013(0.0064)	0.9085(0.009)
AUC ROC	0.6642(0.0217)	0.6419(0.0286)	0.6332(0.0222)	0.9115(0.013)	0.908(0.0057)	0.9143(0.0078)
Accuracy	0.9536(0.0067)	0.9549(0.0038)	0.9533(0.0089)	0.9842(0.0017)	0.9841(0.0005)	0.9846(0.0011)

Table A.79: Neural Network - Training - Part 1

NN	POWER			STD		
Optimized For	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.8428(0.0246)	0.8561(0.0125)	0.8456(0.0209)	0.8099(0.0279)	0.8003(0.0272)	0.8095(0.0188)
Gmean	0.912(0.0153)	0.9216(0.0081)	0.9159(0.0132)	0.8949(0.0172)	0.8883(0.0166)	0.895(0.0105)
AUC ROC	0.9169(0.0135)	0.9253(0.0075)	0.92(0.012)	0.9011(0.0155)	0.8954(0.0147)	0.901(0.0094)
Accuracy	0.986(0.0019)	0.9864(0.0009)	0.9853(0.0014)	0.9821(0.0022)	0.9817(0.0022)	0.9819(0.0021)

Table A.80: Neural Network - Training - Part 2

NN	TRD		
Optimized For	**F2**	**G-mean**	**PR-Recall**
F2	0.8305(0.0202)	0.8444(0.0169)	0.8401(0.0191)
Gmean	0.906(0.0137)	0.9153(0.0104)	0.9129(0.0141)
AUC ROC	0.9123(0.0119)	0.92(0.0095)	0.9182(0.0125)
Accuracy	0.9844(0.0011)	0.9852(0.0013)	0.9848(0.0007)

Table A.81: Neural Network - Training - Part 3

NN	No Scaling			NORM		
Optimized For	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.3759(0.1782)	0.4934(0.2277)	0.2786(0.1636)	0.8346(0.047)	0.8707(0.046)	0.8642(0.0522)
Gmean	0.565(0.1709)	0.664(0.2023)	0.4677(0.1874)	0.9108(0.0324)	0.9386(0.0314)	0.9307(0.0402)
AUC ROC	0.669(0.0852)	0.7336(0.1176)	0.6217(0.0741)	0.9145(0.0297)	0.9403(0.0298)	0.9333(0.0371)
Accuracy	0.9616(0.01)	0.9623(0.0097)	0.9595(0.0063)	0.984(0.0026)	0.9842(0.0026)	0.9855(0.0017)

Table A.82: Neural Network - Testing - Part 1

NN	POWER			STD		
Optimized For	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.8496(0.0468)	0.82(0.052)	0.8398(0.028)	0.8125(0.0326)	0.8306(0.0512)	0.8008(0.0289)
Gmean	0.9177(0.0258)	0.897(0.0346)	0.9118(0.0149)	0.8963(0.0209)	0.9103(0.03)	0.89(0.0175)
AUC ROC	0.9207(0.0239)	0.9022(0.0314)	0.9152(0.0137)	0.901(0.019)	0.9138(0.028)	0.8952(0.0158)
Accuracy	0.986(0.0048)	0.985(0.0035)	0.9852(0.0048)	0.9827(0.0038)	0.9827(0.0047)	0.9814(0.0026)

Table A.83: Neural Network - Testing - Part 2

NN	TRD		
Optimized For	**F2**	**G-mean**	**PR-Recall**
F2	0.092(0.2058)	0.3037(0.4295)	0.1978(0.2262)
Gmean	0.1273(0.2847)	0.3479(0.4836)	0.3499(0.3729)
AUC ROC	0.5405(0.0907)	0.6538(0.223)	0.6167(0.1637)
Accuracy	0.9567(0.0085)	0.9621(0.0126)	0.9256(0.0704)

Table A.84: Neural Network - Testing - Part 3

A.3. Classification For CHD. Classification For CHD

ADABOOST	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.2291(0.0625)	0.2142(0.0561)	0.2304(0.0815)
Gmean	0.3685(0.0809)	0.3422(0.0636)	0.3703(0.1132)
AUC ROC	0.6205(0.0363)	0.6116(0.0276)	0.6227(0.0446)
Accuracy	0.9842(0.002)	0.9841(0.0018)	0.9842(0.0018)

Table A.85: ADABOOST - Training

ADABOOST	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.1435(0.2079)	0.0762(0.1096)	0.1458(0.1858)
Gmean	0.2371(0.3275)	0.1814(0.2543)	0.2765(0.2789)
AUC ROC	0.5669(0.1025)	0.5369(0.0635)	0.5657(0.0894)
Accuracy	0.9832(0.0056)	0.9799(0.0028)	0.9842(0.0059)

Table A.86: ADABOOST - Testing

ADACOST	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.401(0.0107)	0.399(0.0093)	0.4007(0.0092)
Gmean	0.964(0.0012)	0.964(0.0012)	0.964(0.0012)
AUC ROC	0.9647(0.0011)	0.9647(0.0011)	0.9647(0.0011)
Accuracy	0.93(0.0022)	0.93(0.0022)	0.93(0.0022)

Table A.87: ADACOST - Training

ADACOST	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.3971(0.032)	0.3971(0.032)	0.3971(0.032)
Gmean	0.964(0.0047)	0.964(0.0047)	0.964(0.0047)
AUC ROC	0.9647(0.0045)	0.9647(0.0045)	0.9647(0.0045)
Accuracy	0.93(0.0089)	0.93(0.0089)	0.93(0.0089)

Table A.88: ADACOST - Testing

CATBOOST	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.1661(0.0627)	0.1801(0.0683)	0.1777(0.0807)
Gmean	0.2382(0.0891)	0.2544(0.093)	0.2504(0.1108)
AUC ROC	0.5757(0.0297)	0.5807(0.0303)	0.5801(0.0368)
Accuracy	0.9913(0.0005)	0.9915(0.0007)	0.9914(0.0009)

Table A.89: CATBOOST - Training

CATBOOST	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0333(0.0745)	0.0333(0.0745)	0.0333(0.0745)
Gmean	0.0755(0.1689)	0.0755(0.1689)	0.0755(0.1689)
AUC ROC	0.5142(0.0317)	0.5142(0.0317)	0.5142(0.0317)
Accuracy	0.9908(0.0006)	0.9908(0.0006)	0.9908(0.0006)

Table A.90: CATBOOST - Testing

DT	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.2248(0.0694)	0.235(0.0725)	0.2059(0.0344)
Gmean	0.3614(0.098)	0.3882(0.1079)	0.3285(0.0444)
AUC ROC	0.6151(0.0355)	0.6242(0.0416)	0.6062(0.0181)
Accuracy	0.9845(0.0023)	0.9838(0.0018)	0.9842(0.0012)

Table A.91: Decision Tree - Training

DT	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.1689(0.2594)	0.0263(0.0588)	0.0527(0.0722)
Gmean	0.2573(0.361)	0.0752(0.1681)	0.1503(0.2058)
AUC ROC	0.581(0.1294)	0.5091(0.0316)	0.5242(0.0379)
Accuracy	0.983(0.0068)	0.9809(0.0033)	0.9827(0.0032)

Table A.92: Decision Tree - Testing

DT CS	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.25(0.0454)	0.264(0.0566)	0.2349(0.049)
Gmean	0.4028(0.0601)	0.4187(0.0687)	0.3583(0.0756)
AUC ROC	0.6323(0.0276)	0.6409(0.033)	0.6188(0.0259)
Accuracy	0.9848(0.0005)	0.9842(0.0013)	0.9863(0.0005)

Table A.93: Cost Sensitive Decision Tree - Training

DT CS	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0526(0.1177)	0.1382(0.1656)	0.1332(0.1576)
Gmean	0.1064(0.2378)	0.281(0.2801)	0.2808(0.2798)
AUC ROC	0.5241(0.0635)	0.5677(0.0873)	0.5676(0.0865)
Accuracy	0.9824(0.003)	0.9847(0.0016)	0.9845(0.0038)

Table A.94: Cost Sensitive Decision Tree - Testing

DT HELLINGER	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.1893(0.0586)	0.1903(0.0648)	0.2033(0.0508)
Gmean	0.3027(0.087)	0.3146(0.105)	0.3213(0.0724)
AUC ROC	0.5979(0.032)	0.598(0.0324)	0.6032(0.025)
Accuracy	0.9842(0.0012)	0.9843(0.0018)	0.9848(0.0017)

Table A.95: Decision Tree HELLINGER Distance - Training

DT HELLINGER	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.1758(0.1618)	0.1748(0.1615)	0.2035(0.1688)
Gmean	0.352(0.2324)	0.352(0.2324)	0.3832(0.2466)
AUC ROC	0.5808(0.0798)	0.5814(0.079)	0.5956(0.0826)
Accuracy	0.9863(0.0045)	0.9873(0.0022)	0.9875(0.0023)

Table A.96: Decision Tree HELLINGER Distance - Testing

DT SMOTENC	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.2779(0.0455)	0.2805(0.0341)	0.274(0.0418)
Gmean	0.4342(0.0537)	0.4559(0.047)	0.4366(0.041)
AUC ROC	0.6491(0.0233)	0.6573(0.022)	0.6495(0.0235)
Accuracy	0.9831(0.002)	0.9829(0.0014)	0.9827(0.0012)

Table A.97: Decision Tree SMOTENC - Training

DT SMOTENC	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.2061(0.1901)	0.1853(0.0834)	0.1809(0.0647)
Gmean	0.4024(0.2758)	0.4334(0.0908)	0.4314(0.0769)
AUC ROC	0.6074(0.1096)	0.5928(0.0421)	0.5912(0.0349)
Accuracy	0.9827(0.0029)	0.9819(0.004)	0.9822(0.0027)

Table A.98: Decision Tree SMOTENC - Testing

DT SMOTETOMEK	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.4216(0.0833)	0.432(0.0929)	0.3879(0.115)
Gmean	0.7319(0.0817)	0.7711(0.0656)	0.6006(0.1922)
AUC ROC	0.7971(0.0564)	0.8147(0.0514)	0.7392(0.1031)
Accuracy	0.9698(0.0103)	0.9676(0.0102)	0.9805(0.0051)

Table A.99: Decision Tree SMOTETOMEK - Training

DT SMOTETOMEK	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.3797(0.1503)	0.4648(0.1225)	0.2709(0.1452)
Gmean	0.7383(0.1512)	0.8173(0.1108)	0.5619(0.1948)
AUC ROC	0.7762(0.1101)	0.835(0.0894)	0.6683(0.1119)
Accuracy	0.9667(0.0202)	0.971(0.01)	0.9773(0.0084)

Table A.100: Decision Tree SMOTETOMEK - Testing

KNN	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.1535(0.0703)	0.1665(0.0682)	0.1488(0.0533)	0.1859(0.0385)	0.1892(0.054)	0.1715(0.0551)
Gmean	0.2434(0.1118)	0.2668(0.1082)	0.2346(0.084)	0.2927(0.0582)	0.2867(0.0776)	0.26(0.0786)
AUC ROC	0.5747(0.0356)	0.5802(0.0355)	0.5719(0.0281)	0.5932(0.0202)	0.5919(0.0261)	0.5823(0.0296)
Accuracy	0.9872(0.0023)	0.9873(0.0026)	0.9873(0.0024)	0.9867(0.0011)	0.9865(0.0013)	0.9881(0.0031)

Table A.101: KNN - Training - Part 1

KNN	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.1791(0.0644)	0.185(0.063)	0.1618(0.0836)	0.2184(0.0726)	0.208(0.0583)	0.1974(0.0837)
Gmean	0.2858(0.0911)	0.2836(0.0821)	0.2474(0.1261)	0.3273(0.1052)	0.3324(0.098)	0.3001(0.1277)
AUC ROC	0.5862(0.0313)	0.5878(0.0307)	0.5781(0.0425)	0.6084(0.0345)	0.6037(0.0278)	0.597(0.0426)
Accuracy	0.9861(0.0022)	0.986(0.0023)	0.9884(0.0023)	0.9865(0.0018)	0.9868(0.0012)	0.9878(0.0018)

Table A.102: KNN - Training - Part 2

KNN	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.2417(0.038)	0.2431(0.0418)	0.1849(0.0165)
Gmean	0.3736(0.0421)	0.3557(0.044)	0.2713(0.0296)
AUC ROC	0.6217(0.0221)	0.6202(0.0209)	0.5857(0.0107)
Accuracy	0.9867(0.0016)	0.987(0.0016)	0.9894(0.0037)

Table A.103: KNN - Training - Part 3

KNN	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.1181(0.1187)	0.1181(0.1187)	0.1181(0.1187)
Gmean	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.2574(0.2434)	0.2574(0.2434)	0.2574(0.2434)
AUC ROC	0.4969(0.0017)	0.4969(0.0017)	0.4969(0.0017)	0.5533(0.0609)	0.5533(0.0609)	0.5551(0.0588)
Accuracy	0.9847(0.0036)	0.9847(0.0036)	0.9847(0.0036)	0.9842(0.005)	0.9842(0.005)	0.9878(0.0019)

Table A.104: KNN - Testing - Part 1

KNN	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.1538(0.2051)	0.1538(0.2051)	0.0538(0.0742)	0.1678(0.189)	0.1678(0.189)	0.1435(0.1761)
Gmean	0.2961(0.3134)	0.2961(0.3134)	0.1457(0.1997)	0.3075(0.3005)	0.3075(0.3005)	0.2764(0.2786)
AUC ROC	0.5802(0.1167)	0.5802(0.1167)	0.5242(0.0354)	0.5802(0.0942)	0.5802(0.0942)	0.5664(0.0881)
Accuracy	0.985(0.0024)	0.985(0.0024)	0.9863(0.0039)	0.985(0.0047)	0.985(0.0047)	0.9857(0.0056)

Table A.105: KNN - Testing - Part 2

KNN	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0(0.0)	0.0(0.0)	0.0(0.0)
Gmean	0.0(0.0)	0.0(0.0)	0.0(0.0)
AUC ROC	0.5(0.0)	0.5(0.0)	0.5(0.0)
Accuracy	0.9908(0.0006)	0.9908(0.0006)	0.9908(0.0006)

Table A.106: KNN - Testing - Part 3

LR	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.0048(0.0066)	0.0(0.0)	0.0132(0.0189)	0.0(0.0)	0.0(0.0)	0.0(0.0)
Gmean	0.0077(0.0105)	0.0(0.0)	0.0201(0.0287)	0.0(0.0)	0.0(0.0)	0.0(0.0)
AUC ROC	0.5018(0.003)	0.4996(0.0001)	0.5057(0.009)	0.5(0.0)	0.5(0.0)	0.5(0.0)
Accuracy	0.9901(0.0005)	0.99(0.0003)	0.99(0.0004)	0.9908(0.0001)	0.9908(0.0002)	0.9908(0.0001)

Table A.107: Logistic Regression - Training - Part 1

LR	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.0189(0.0163)	0.02(0.0188)	0.0176(0.019)	0.0136(0.0195)	0.0075(0.0114)	0.0101(0.0167)
Gmean	0.0278(0.0235)	0.0308(0.0292)	0.0269(0.0292)	0.0201(0.0288)	0.0115(0.0172)	0.0154(0.0251)
AUC ROC	0.508(0.0072)	0.5087(0.0083)	0.5075(0.0082)	0.5058(0.0088)	0.5031(0.0049)	0.504(0.0072)
Accuracy	0.9904(0.0002)	0.9906(0.0003)	0.9905(0.0003)	0.9903(0.0002)	0.9904(0.0001)	0.9901(0.0003)

Table A.108: Logistic Regression - Training - Part 2

LR	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0(0.0)	0.0(0.0)	0.0(0.0)
Gmean	0.0(0.0)	0.0(0.0)	0.0(0.0)
AUC ROC	0.5(0.0)	0.5(0.0001)	0.5(0.0)
Accuracy	0.9908(0.0001)	0.9908(0.0)	0.9908(0.0001)

Table A.109: Logistic Regression - Training - Part 3

LR	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)
Gmean	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)
AUC ROC	0.4996(0.0004)	0.4994(0.0005)	0.4995(0.0005)	0.5(0.0)	0.5(0.0)	0.5(0.0)
Accuracy	0.9901(0.0011)	0.9896(0.0011)	0.9898(0.0013)	0.9908(0.0006)	0.9908(0.0006)	0.9908(0.0006)

Table A.110: Logistic Regression - Testing - Part 1

LR	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)
Gmean	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)
AUC ROC	0.4996(0.0009)	0.4995(0.0008)	0.4996(0.0009)	0.4995(0.0003)	0.4996(0.0004)	0.4996(0.0004)
Accuracy	0.9901(0.0017)	0.9898(0.0016)	0.9901(0.0017)	0.9898(0.0009)	0.9901(0.0011)	0.9901(0.0011)

Table A.111: Logistic Regression - Testing - Part 2

LR	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0(0.0)	0.0(0.0)	0.0(0.0)
Gmean	0.0(0.0)	0.0(0.0)	0.0(0.0)
AUC ROC	0.5(0.0)	0.5(0.0)	0.5(0.0)
Accuracy	0.9908(0.0006)	0.9908(0.0006)	0.9908(0.0006)

Table A.112: Logistic Regression - Testing - Part 3

LR CS	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.4258(0.02)	0.4167(0.0296)	0.4055(0.0357)	0.4471(0.0427)	0.4208(0.033)	0.4411(0.0657)
Gmean	0.8383(0.0794)	0.9033(0.0159)	0.736(0.098)	0.8189(0.0761)	0.938(0.0179)	0.7725(0.0761)
AUC ROC	0.8626(0.0592)	0.9098(0.0143)	0.806(0.0591)	0.8526(0.0554)	0.9418(0.0164)	0.8227(0.0435)
Accuracy	0.959(0.0105)	0.9467(0.0062)	0.9679(0.0066)	0.9642(0.0159)	0.9402(0.0084)	0.9705(0.0027)

Table A.113: Cost Sensitive Logistic Regression - Training - Part 1

LR CS	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.445(0.0372)	0.4286(0.038)	0.4262(0.031)	0.4412(0.0263)	0.4173(0.0266)	0.433(0.0499)
Gmean	0.876(0.073)	0.9145(0.0205)	0.7876(0.0827)	0.8579(0.076)	0.904(0.0156)	0.8383(0.0929)
AUC ROC	0.8912(0.0597)	0.9213(0.0187)	0.835(0.0542)	0.8804(0.058)	0.9135(0.0125)	0.8758(0.0573)
Accuracy	0.9567(0.0137)	0.9466(0.0072)	0.9654(0.0108)	0.9582(0.0132)	0.9464(0.0067)	0.9578(0.0139)

Table A.114: Cost Sensitive Logistic Regression - Training - Part 2

LR CS	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.4389(0.049)	0.4133(0.0327)	0.4387(0.0479)
Gmean	0.8378(0.0974)	0.9408(0.0068)	0.7626(0.053)
AUC ROC	0.8674(0.0674)	0.9436(0.0063)	0.8167(0.0291)
Accuracy	0.9608(0.0147)	0.9373(0.0109)	0.9707(0.0026)

Table A.115: Cost Sensitive Logistic Regression - Training - Part 3

LR CS	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.4268(0.1185)	0.3799(0.0818)	0.4242(0.1187)	0.409(0.0965)	0.4077(0.0512)	0.4167(0.1008)
Gmean	0.8591(0.0794)	0.8682(0.0919)	0.806(0.1455)	0.7952(0.1403)	0.9416(0.0388)	0.7827(0.1224)
AUC ROC	0.8663(0.0701)	0.8749(0.0829)	0.8291(0.116)	0.8199(0.1172)	0.9421(0.0387)	0.8077(0.0992)
Accuracy	0.956(0.0138)	0.945(0.0067)	0.9664(0.0098)	0.9656(0.0133)	0.9379(0.0096)	0.9695(0.0056)

Table A.116: Cost Sensitive Logistic Regression - Testing - Part 1

LR CS	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.3687(0.1711)	0.3908(0.0805)	0.38(0.192)	0.4052(0.1016)	0.3694(0.0772)	0.4206(0.1044)
Gmean	0.7919(0.2532)	0.8872(0.0927)	0.7586(0.2621)	0.839(0.1145)	0.867(0.0917)	0.8538(0.1283)
AUC ROC	0.8346(0.1665)	0.8923(0.0864)	0.8106(0.1792)	0.8524(0.0917)	0.8737(0.0827)	0.8671(0.1061)
Accuracy	0.956(0.014)	0.9443(0.0067)	0.9646(0.0082)	0.9565(0.0123)	0.9425(0.0048)	0.9575(0.0112)

Table A.117: Cost Sensitive Logistic Regression - Testing - Part 2

LR CS	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.4361(0.1941)	0.3651(0.1708)	0.2943(0.2862)
Gmean	0.8376(0.2723)	0.8407(0.2734)	0.5707(0.5217)
AUC ROC	0.8799(0.1802)	0.8831(0.1811)	0.7712(0.249)
Accuracy	0.9616(0.028)	0.9399(0.0482)	0.9672(0.0306)

Table A.118: Cost Sensitive Logistic Regression - Testing - Part 3

LR SMOTENC	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.3985(0.0395)	0.3826(0.0254)	0.391(0.0459)	0.366(0.0469)	0.3607(0.0389)	0.3639(0.0415)
Gmean	0.7827(0.0402)	0.7383(0.0495)	0.7638(0.044)	0.7321(0.037)	0.72(0.0446)	0.729(0.0266)
AUC ROC	0.824(0.0299)	0.81(0.0187)	0.8126(0.0269)	0.7925(0.027)	0.7862(0.0195)	0.7875(0.0208)
Accuracy	0.962(0.0062)	0.9617(0.0069)	0.9624(0.0079)	0.9618(0.0084)	0.9604(0.0075)	0.9628(0.0087)

Table A.119: Logistic Regression SMOTENC - Training - Part 1

LR SMOTENC	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.3887(0.0394)	0.3766(0.0484)	0.38(0.0347)	0.3808(0.048)	0.3859(0.0443)	0.3712(0.048)
Gmean	0.7277(0.0191)	0.7094(0.0398)	0.7155(0.0373)	0.7314(0.0216)	0.7328(0.0265)	0.7055(0.0305)
AUC ROC	0.7881(0.0195)	0.7802(0.0183)	0.7776(0.0144)	0.7875(0.0085)	0.791(0.015)	0.7752(0.0128)
Accuracy	0.9673(0.0069)	0.9668(0.0073)	0.9672(0.007)	0.9661(0.008)	0.9664(0.0065)	0.9667(0.008)

Table A.120: Logistic Regression SMOTENC - Training - Part 2

LR SMOTENC	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.4071(0.0359)	0.3997(0.0273)	0.4118(0.0288)
Gmean	0.7918(0.0412)	0.7929(0.0461)	0.7972(0.0177)
AUC ROC	0.8316(0.0295)	0.8324(0.0315)	0.8381(0.0135)
Accuracy	0.9618(0.0068)	0.9601(0.0063)	0.9616(0.0063)

Table A.121: Logistic Regression SMOTENC - Training - Part 3

LR SMOTENC	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.3632(0.0586)	0.3733(0.0681)	0.3562(0.1009)	0.3846(0.0564)	0.398(0.0517)	0.387(0.0563)
Gmean	0.7669(0.0982)	0.7823(0.1149)	0.7602(0.1596)	0.7976(0.1255)	0.8159(0.0931)	0.7978(0.1256)
AUC ROC	0.7916(0.072)	0.8055(0.088)	0.7931(0.1119)	0.8195(0.0967)	0.8314(0.0765)	0.8198(0.0968)
Accuracy	0.9621(0.0045)	0.9616(0.0052)	0.9616(0.0052)	0.9613(0.0112)	0.9603(0.0113)	0.9618(0.0109)

Table A.122: Logistic Regression SMOTENC - Testing - Part 1

LR SMOTENC	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.4085(0.107)	0.4025(0.0979)	0.4028(0.0992)	0.3851(0.0459)	0.3949(0.0512)	0.3782(0.0557)
Gmean	0.7684(0.1144)	0.7681(0.1141)	0.7681(0.1142)	0.7672(0.0871)	0.768(0.0876)	0.7493(0.1118)
AUC ROC	0.7956(0.0886)	0.7952(0.0882)	0.7952(0.0884)	0.792(0.0652)	0.7929(0.0658)	0.7806(0.0814)
Accuracy	0.97(0.0083)	0.9692(0.0077)	0.9692(0.0075)	0.9664(0.0087)	0.9682(0.0078)	0.9684(0.0083)

Table A.123: Logistic Regression SMOTENC - Testing - Part 2

LR SMOTENC	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.3065(0.1979)	0.2967(0.181)	0.2959(0.1843)
Gmean	0.6737(0.4179)	0.6718(0.4159)	0.6711(0.4153)
AUC ROC	0.7953(0.2142)	0.7937(0.2115)	0.7931(0.211)
Accuracy	0.9588(0.0303)	0.9557(0.0361)	0.9544(0.0397)

Table A.124: Logistic Regression SMOTENC - Testing - Part 3

LR SMOTETOMEK	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.4156(0.0348)	0.4104(0.0382)	0.4155(0.04)	0.4179(0.0433)	0.4302(0.0382)	0.4251(0.0324)
Gmean	0.8645(0.0135)	0.862(0.0222)	0.8642(0.033)	0.8761(0.0241)	0.908(0.0186)	0.8947(0.0266)
AUC ROC	0.8804(0.0076)	0.879(0.0194)	0.8819(0.0183)	0.8952(0.0171)	0.9159(0.0175)	0.9071(0.0182)
Accuracy	0.9527(0.0082)	0.9522(0.0071)	0.9525(0.0079)	0.9515(0.0075)	0.949(0.0086)	0.95(0.0074)

Table A.125: Logistic Regression SMOTETOMEK - Training - Part 1

LR SMOTETOMEK	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.4169(0.0509)	0.4267(0.0466)	0.4289(0.0455)	0.4154(0.0455)	0.4064(0.0346)	0.4228(0.0425)
Gmean	0.8475(0.0291)	0.8661(0.0245)	0.865(0.0224)	0.8507(0.0359)	0.8364(0.0306)	0.8535(0.0355)
AUC ROC	0.8675(0.0259)	0.8779(0.0205)	0.8803(0.0184)	0.8734(0.0276)	0.8701(0.0232)	0.8749(0.0211)
Accuracy	0.9557(0.0072)	0.9555(0.0071)	0.9559(0.0075)	0.9542(0.0084)	0.9531(0.0072)	0.956(0.0075)

Table A.126: Logistic Regression SMOTETOMEK - Training - Part 2

LR SMOTETOMEK	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.4223(0.0322)	0.4223(0.032)	0.417(0.029)
Gmean	0.8956(0.0175)	0.8877(0.0243)	0.8805(0.0386)
AUC ROC	0.9067(0.0172)	0.9045(0.0161)	0.8952(0.0276)
Accuracy	0.9493(0.006)	0.9494(0.0063)	0.9505(0.0079)

Table A.127: Logistic Regression SMOTETOMEK - Training - Part 3

LR SMOTETOMEK	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.3452(0.0891)	0.3581(0.072)	0.3481(0.0847)	0.4037(0.0653)	0.419(0.0815)	0.4145(0.0686)
Gmean	0.8048(0.1321)	0.8245(0.1064)	0.8052(0.1317)	0.8731(0.0877)	0.901(0.1071)	0.8874(0.0984)
AUC ROC	0.8239(0.1122)	0.838(0.0949)	0.8245(0.1117)	0.8798(0.0783)	0.9072(0.0974)	0.894(0.0888)
Accuracy	0.9491(0.0064)	0.9488(0.0062)	0.9501(0.0056)	0.9511(0.0037)	0.9494(0.0042)	0.9511(0.0043)

Table A.128: Logistic Regression SMOTETOMEK - Testing - Part 1

LR SMOTETOMEK	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.4126(0.0898)	0.4141(0.0857)	0.4153(0.086)	0.3531(0.0885)	0.3617(0.095)	0.3584(0.0857)
Gmean	0.8748(0.12)	0.875(0.1196)	0.8751(0.1196)	0.8057(0.132)	0.82(0.1318)	0.8063(0.1316)
AUC ROC	0.8843(0.1054)	0.8845(0.1048)	0.8847(0.1049)	0.825(0.112)	0.837(0.111)	0.8257(0.1115)
Accuracy	0.9529(0.0024)	0.9534(0.0021)	0.9537(0.0021)	0.9511(0.0065)	0.9504(0.0057)	0.9527(0.0064)

Table A.129: Logistic Regression SMOTETOMEK - Testing - Part 2

LR SMOTETOMEK	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.3304(0.1984)	0.3768(0.1631)	0.3731(0.1596)
Gmean	0.7591(0.425)	0.8446(0.2749)	0.8442(0.2747)
AUC ROC	0.8599(0.2024)	0.8868(0.182)	0.8864(0.1818)
Accuracy	0.9465(0.0333)	0.9473(0.0332)	0.9465(0.0333)

Table A.130: Logistic Regression SMOTETOMEK - Testing - Part 3

RF	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.122(0.0476)	0.1267(0.0605)	0.1283(0.0563)
Gmean	0.1783(0.067)	0.1774(0.082)	0.1827(0.0792)
AUC ROC	0.5533(0.0211)	0.5561(0.027)	0.5566(0.0252)
Accuracy	0.9917(0.0003)	0.9917(0.0006)	0.9917(0.0005)

Table A.131: Cost Sensitive Random Forest - Training

RF	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0(0.0)	0.0(0.0)	0.0(0.0)
Gmean	0.0(0.0)	0.0(0.0)	0.0(0.0)
AUC ROC	0.5(0.0)	0.4999(0.0003)	0.5(0.0)
Accuracy	0.9908(0.0006)	0.9906(0.0007)	0.9908(0.0006)

Table A.132: Cost Sensitive Random Forest - Testing

RF CS	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.3977(0.0495)	0.3808(0.0425)	0.3353(0.0246)
Gmean	0.6445(0.0562)	0.6206(0.0461)	0.5088(0.0564)
AUC ROC	0.7411(0.0269)	0.7331(0.024)	0.6789(0.0264)
Accuracy	0.98(0.0023)	0.9793(0.0017)	0.9854(0.0063)

Table A.133: Cost Sensitive Random Forest - Training

RF CS	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.4196(0.1701)	0.4001(0.15)	0.2891(0.2632)
Gmean	0.7254(0.1849)	0.7093(0.167)	0.5017(0.3766)
AUC ROC	0.7732(0.1371)	0.7587(0.1211)	0.6807(0.1889)
Accuracy	0.9781(0.0025)	0.9776(0.0014)	0.9842(0.0049)

Table A.134: Cost Sensitive Random Forest - Testing

SVM	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0112(0.0251)	0.0049(0.0068)	0.0101(0.0167)
Gmean	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0163(0.0363)	0.0077(0.0105)	0.0154(0.0251)
AUC ROC	0.4999(0.0001)	0.4999(0.0001)	0.4999(0.0001)	0.5049(0.011)	0.502(0.0028)	0.5043(0.007)
Accuracy	0.9907(0.0002)	0.9907(0.0002)	0.9906(0.0002)	0.9908(0.0003)	0.9905(0.0006)	0.9906(0.0004)

Table A.135: SVM - Training - Part 1

SVM	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.1288(0.0602)	0.1097(0.054)	0.1269(0.0698)	0.0617(0.0616)	0.0692(0.0749)	0.0783(0.0601)
Gmean	0.1941(0.0858)	0.1636(0.0779)	0.1884(0.0997)	0.0957(0.0939)	0.1037(0.1073)	0.1166(0.0835)
AUC ROC	0.5589(0.0288)	0.55(0.0269)	0.5578(0.0327)	0.5278(0.0286)	0.5306(0.0345)	0.5345(0.027)
Accuracy	0.9897(0.001)	0.9895(0.0006)	0.9897(0.0011)	0.9892(0.0007)	0.9894(0.0011)	0.9894(0.0012)

Table A.136: SVM - Training - Part 2

SVM	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0083(0.0185)	0.0077(0.0172)	0.007(0.0102)
Gmean	0.0124(0.0277)	0.0115(0.0258)	0.0115(0.0172)
AUC ROC	0.5038(0.0085)	0.5032(0.0072)	0.5031(0.0046)
Accuracy	0.9907(0.0003)	0.9907(0.0003)	0.9905(0.0004)

Table A.137: SVM - Training - Part 3

SVM	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)
Gmean	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)
AUC ROC	0.5(0.0)	0.5(0.0)	0.5(0.0)	0.4996(0.0009)	0.4996(0.0009)	0.4996(0.0009)
Accuracy	0.9908(0.0006)	0.9908(0.0006)	0.9908(0.0006)	0.9901(0.0017)	0.9901(0.0017)	0.9901(0.0017)

Table A.138: SVM - Testing - Part 1

SVM	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.0667(0.0915)	0.0667(0.0915)	0.0667(0.0915)	0.0312(0.0699)	0.0312(0.0699)	0.0312(0.0699)
Gmean	0.1511(0.2069)	0.1511(0.2069)	0.1511(0.2069)	0.0754(0.1687)	0.0754(0.1687)	0.0754(0.1687)
AUC ROC	0.5273(0.0397)	0.5273(0.0397)	0.5273(0.0397)	0.5127(0.0317)	0.5127(0.0317)	0.5127(0.0317)
Accuracy	0.9888(0.0035)	0.9888(0.0035)	0.9888(0.0035)	0.988(0.0011)	0.988(0.0011)	0.988(0.0011)

Table A.139: SVM - Testing - Part 2

SVM	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0(0.0)	0.0(0.0)	0.0(0.0)
Gmean	0.0(0.0)	0.0(0.0)	0.0(0.0)
AUC ROC	0.5(0.0)	0.5(0.0)	0.5(0.0)
Accuracy	0.9908(0.0006)	0.9908(0.0006)	0.9908(0.0006)

Table A.140: SVM - Testing - Part 3

SVM CS	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.4468(0.0661)	0.4097(0.0455)	0.4349(0.0703)	0.4802(0.0473)	0.4181(0.0387)	0.4756(0.0605)
Gmean	0.8278(0.0604)	0.8615(0.0311)	0.7708(0.0767)	0.8113(0.0497)	0.9117(0.0412)	0.805(0.0531)
AUC ROC	0.8593(0.0392)	0.8787(0.0203)	0.8264(0.0445)	0.8424(0.0356)	0.9241(0.0328)	0.8385(0.0402)
Accuracy	0.9634(0.015)	0.9517(0.0089)	0.9678(0.0117)	0.9725(0.0012)	0.9444(0.009)	0.9722(0.0016)

Table A.141: Cost Sensitive SVM - Training - Part 1

SVM CS	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.4137(0.048)	0.417(0.0619)	0.4176(0.047)	0.4387(0.0623)	0.4097(0.042)	0.4156(0.0632)
Gmean	0.7822(0.0485)	0.8011(0.0081)	0.7815(0.0398)	0.7947(0.0623)	0.8378(0.0358)	0.7095(0.091)
AUC ROC	0.8211(0.0301)	0.8461(0.003)	0.827(0.0232)	0.8409(0.0389)	0.8684(0.0189)	0.788(0.0524)
Accuracy	0.9641(0.0116)	0.9589(0.0136)	0.9638(0.0121)	0.9662(0.0097)	0.954(0.0087)	0.9737(0.0037)

Table A.142: Cost Sensitive SVM - Training - Part 2

SVM CS	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.462(0.0439)	0.4114(0.0281)	0.466(0.0531)
Gmean	0.826(0.0757)	0.9073(0.0282)	0.8367(0.0775)
AUC ROC	0.8637(0.0558)	0.9172(0.0217)	0.8648(0.0569)
Accuracy	0.9655(0.0153)	0.9439(0.0087)	0.9655(0.0153)

Table A.143: Cost Sensitive SVM - Training - Part 3

SVM CS	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.4072(0.1095)	0.3876(0.0696)	0.3968(0.1182)	0.4813(0.1276)	0.421(0.0635)	0.4813(0.1276)
Gmean	0.798(0.1213)	0.8569(0.0872)	0.7624(0.1367)	0.8336(0.1203)	0.9292(0.0664)	0.8336(0.1203)
AUC ROC	0.8193(0.0973)	0.865(0.0784)	0.7931(0.1095)	0.8495(0.0977)	0.9312(0.0633)	0.8495(0.0977)
Accuracy	0.9644(0.0116)	0.9499(0.0079)	0.9687(0.0079)	0.9718(0.0046)	0.9443(0.0066)	0.9718(0.0046)

Table A.144: Cost Sensitive SVM - Testing - Part 1

SVM CS	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.3549(0.1364)	0.3347(0.1025)	0.3549(0.1364)	0.3635(0.1435)	0.3654(0.049)	0.3871(0.1556)
Gmean	0.7412(0.1693)	0.753(0.1856)	0.7412(0.1693)	0.7381(0.1533)	0.8268(0.0875)	0.722(0.1799)
AUC ROC	0.7799(0.1212)	0.7913(0.1387)	0.7799(0.1212)	0.7753(0.1133)	0.8389(0.0789)	0.7689(0.1359)
Accuracy	0.9633(0.0095)	0.958(0.01)	0.9633(0.0095)	0.9649(0.0077)	0.9506(0.0081)	0.9733(0.0058)

Table A.145: Cost Sensitive SVM - Testing - Part 2

SVM CS	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.3587(0.2362)	0.3439(0.1314)	0.3587(0.2362)
Gmean	0.688(0.4051)	0.8422(0.2743)	0.688(0.4051)
AUC ROC	0.7995(0.1977)	0.8839(0.1826)	0.7995(0.1977)
Accuracy	0.9672(0.0186)	0.9415(0.0258)	0.9672(0.0186)

Table A.146: Cost Sensitive SVM - Testing - Part 3

SVM SMOTENC	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.0017(0.0037)	0.0(0.0)	0.0(0.0)	0.3208(0.0421)	0.3339(0.0444)	0.3317(0.0423)
Gmean	0.0038(0.0085)	0.0(0.0)	0.0(0.0)	0.6323(0.0308)	0.6401(0.0415)	0.6439(0.0487)
AUC ROC	0.4952(0.0031)	0.4947(0.0021)	0.4946(0.0024)	0.7324(0.0201)	0.7365(0.018)	0.7403(0.0181)
Accuracy	0.9792(0.0051)	0.9804(0.004)	0.98(0.0047)	0.967(0.0069)	0.9677(0.007)	0.9675(0.0068)

Table A.147: SVM SMOTENC - Training - Part 1

SVM SMOTENC	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.2298(0.0787)	0.2511(0.0674)	0.2439(0.0741)	0.3045(0.0884)	0.3005(0.072)	0.3141(0.0771)
Gmean	0.4088(0.109)	0.4383(0.1026)	0.4153(0.0957)	0.5072(0.0997)	0.5129(0.088)	0.5451(0.0806)
AUC ROC	0.6356(0.0446)	0.645(0.0389)	0.6411(0.0431)	0.6815(0.0462)	0.6785(0.0412)	0.6864(0.0429)
Accuracy	0.977(0.0055)	0.9781(0.0046)	0.9782(0.005)	0.9786(0.0049)	0.9782(0.0044)	0.9784(0.0044)

Table A.148: SVM SMOTENC - Training - Part 2

SVM SMOTENC	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.2694(0.0432)	0.2653(0.0471)	0.2674(0.0562)
Gmean	0.4725(0.0563)	0.4799(0.0424)	0.4657(0.0673)
AUC ROC	0.6619(0.024)	0.6618(0.0245)	0.6599(0.0313)
Accuracy	0.9768(0.0033)	0.9767(0.0039)	0.9772(0.0037)

Table A.149: SVM SMOTENC - Training - Part 3

SVM SMOTENC	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.3433(0.127)	0.3433(0.127)	0.3433(0.127)
Gmean	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.7172(0.2077)	0.7172(0.2077)	0.7172(0.2077)
AUC ROC	0.4968(0.0036)	0.4968(0.0036)	0.4968(0.0036)	0.7692(0.1222)	0.7692(0.1222)	0.7692(0.1222)
Accuracy	0.9845(0.0076)	0.9845(0.0076)	0.9845(0.0076)	0.9667(0.0081)	0.9667(0.0081)	0.9667(0.0081)

Table A.150: SVM SMOTENC - Testing - Part 1

SVM SMOTENC	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.1859(0.0934)	0.1859(0.0934)	0.1859(0.0934)	0.269(0.1094)	0.269(0.1094)	0.269(0.1094)
Gmean	0.4561(0.129)	0.4561(0.129)	0.4561(0.129)	0.5599(0.1433)	0.5599(0.1433)	0.5599(0.1433)
AUC ROC	0.6048(0.0644)	0.6048(0.0644)	0.6048(0.0644)	0.6599(0.0818)	0.6599(0.0818)	0.6599(0.0818)
Accuracy	0.9776(0.0039)	0.9776(0.0039)	0.9776(0.0039)	0.9781(0.0023)	0.9781(0.0023)	0.9781(0.0023)

Table A.151: SVM SMOTENC - Testing - Part 2

SVM SMOTENC	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0(0.0)	0.0(0.0)	0.0(0.0)
Gmean	0.0(0.0)	0.0(0.0)	0.0(0.0)
AUC ROC	0.5(0.0)	0.5(0.0)	0.5(0.0)
Accuracy	0.9908(0.0006)	0.9908(0.0006)	0.9908(0.0006)

Table A.152: SVM SMOTENC - Testing - Part 3

SVM SMOTETOMEK	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0(0.0)	0.001(0.0022)	0.0(0.0)
Gmean	0.0(0.0)	0.0037(0.0083)	0.0(0.0)
AUC ROC	0.4945(0.0022)	0.4954(0.0028)	0.494(0.0025)
Accuracy	0.98(0.0042)	0.9795(0.0037)	0.9789(0.0048)

Table A.153: SVM SMOTETOMEK - Training

SVM SMOTETOMEK	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0(0.0)	0.0(0.0)	0.0(0.0)
Gmean	0.0(0.0)	0.0(0.0)	0.0(0.0)
AUC ROC	0.4963(0.0034)	0.4963(0.0034)	0.4963(0.0034)
Accuracy	0.9835(0.0071)	0.9835(0.0071)	0.9835(0.0071)

Table A.154: SVM SMOTETOMEK - Testing

XGBoost	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.2259(0.0802)	0.2144(0.088)	0.2156(0.0799)
Gmean	0.3173(0.0949)	0.3023(0.1147)	0.2971(0.1042)
AUC ROC	0.6046(0.0389)	0.598(0.0423)	0.5976(0.0375)
Accuracy	0.9909(0.0007)	0.9909(0.0011)	0.9912(0.0008)

Table A.155: XGBoost - Training

XGBoost	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0606(0.1355)	0.0929(0.1367)	0.0625(0.1398)
Gmean	0.1067(0.2386)	0.1822(0.2555)	0.1068(0.2387)
AUC ROC	0.5279(0.0632)	0.542(0.0632)	0.5278(0.0636)
Accuracy	0.9901(0.0011)	0.9898(0.0009)	0.9898(0.0013)

Table A.156: XGBoost - Testing

XGBoost CS	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.1688(0.0718)	0.1604(0.0665)	0.1802(0.0868)
Gmean	0.2465(0.1009)	0.2272(0.0904)	0.2598(0.1189)
AUC ROC	0.5755(0.0333)	0.5733(0.0322)	0.5817(0.04)
Accuracy	0.9911(0.0008)	0.9911(0.0004)	0.9913(0.0008)

Table A.157: Cost Sensitive XGBoost - Training

XGBoost CS	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0678(0.0929)	0.0678(0.0929)	0.0678(0.0929)
Gmean	0.1511(0.207)	0.1511(0.207)	0.1511(0.207)
AUC ROC	0.5283(0.0391)	0.5283(0.0391)	0.5283(0.0391)
Accuracy	0.9908(0.0011)	0.9908(0.0011)	0.9908(0.0011)

Table A.158: Cost Sensitive XGBoost - Testing

XGBoost SMOTENC	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.3195(0.0857)	0.3106(0.0831)	0.293(0.0802)
Gmean	0.505(0.1047)	0.4813(0.1045)	0.4626(0.1077)
AUC ROC	0.6737(0.0459)	0.6708(0.0465)	0.6618(0.0433)
Accuracy	0.9838(0.0019)	0.9834(0.0021)	0.9831(0.0018)

Table A.159: XGBoost SMOTENC - Training

XGBoost SMOTENC	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.3007(0.1175)	0.3007(0.1175)	0.3007(0.1175)
Gmean	0.5635(0.1489)	0.5635(0.1489)	0.5635(0.1489)
AUC ROC	0.6646(0.0821)	0.6646(0.0821)	0.6646(0.0821)
Accuracy	0.984(0.0029)	0.984(0.0029)	0.984(0.0029)

Table A.160: XGBoost SMOTENC - Testing

XGBoost SMOTETOMEK	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.2975(0.0889)	0.3012(0.0602)	0.3052(0.0855)
Gmean	0.4539(0.1084)	0.4525(0.0868)	0.4528(0.1178)
AUC ROC	0.6552(0.0476)	0.6581(0.0358)	0.6556(0.0458)
Accuracy	0.9863(0.001)	0.9867(0.0009)	0.9873(0.0007)

Table A.161: XGBoost SMOTETOMEK - Training

XGBoost SMOTETOMEK	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.2998(0.1708)	0.2745(0.1837)	0.278(0.1883)
Gmean	0.5335(0.1729)	0.5024(0.1866)	0.5025(0.1868)
AUC ROC	0.652(0.0948)	0.638(0.1021)	0.6381(0.1024)
Accuracy	0.987(0.0017)	0.9873(0.0016)	0.9875(0.0017)

Table A.162: XGBoost SMOTETOMEK - Testing

NN	No Scaling			NORM		
Optimized For	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.01(0.0161)	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)
Gmean	0.0143(0.0228)	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)
AUC ROC	0.5054(0.0096)	0.4998(0.0002)	0.4999(0.0001)	0.4999(0.0001)	0.4999(0.0001)	0.5(0.0001)
Accuracy	0.9905(0.0003)	0.9904(0.0003)	0.9905(0.0001)	0.9906(0.0002)	0.9907(0.0002)	0.9908(0.0001)

Table A.163: Neural Network - Training - Part 1

NN	POWER			STD		
Optimized For	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.107(0.0565)	0.1009(0.0555)	0.1041(0.0638)	0.079(0.0176)	0.0807(0.0554)	0.0723(0.0242)
Gmean	0.1606(0.0752)	0.1532(0.0764)	0.1558(0.0902)	0.1279(0.023)	0.1257(0.0852)	0.1071(0.0257)
AUC ROC	0.5493(0.0276)	0.5474(0.0274)	0.549(0.0304)	0.537(0.0071)	0.5374(0.0276)	0.5339(0.0109)
Accuracy	0.9892(0.0007)	0.9887(0.0005)	0.9887(0.0011)	0.9878(0.001)	0.9885(0.0012)	0.9882(0.0007)

Table A.164: Neural Network - Training - Part 2

NN	TRD		
Optimized For	**F2**	**G-mean**	**PR-Recall**
F2	0.0037(0.0083)	0.0021(0.0047)	0.0051(0.0115)
Gmean	0.0047(0.0105)	0.0038(0.0086)	0.0077(0.0172)
AUC ROC	0.5014(0.0036)	0.501(0.0024)	0.5019(0.0049)
Accuracy	0.9904(0.0003)	0.9906(0.0001)	0.9903(0.0004)

Table A.165: Neural Network - Training - Part 3

NN	No Scaling			NORM		
Optimized For	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)
Gmean	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)
AUC ROC	0.5(0.0)	0.5(0.0)	0.5(0.0)	0.5(0.0)	0.5(0.0)	0.5(0.0)
Accuracy	0.9908(0.0006)	0.9908(0.0006)	0.9908(0.0006)	0.9908(0.0006)	0.9908(0.0006)	0.9908(0.0006)

Table A.166: Neural Network - Testing - Part 1

NN	POWER			STD		
Optimized For	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.0948(0.1403)	0.0866(0.1283)	0.0(0.0)	0.0919(0.084)	0.0978(0.1449)	0.1168(0.0664)
Gmean	0.1823(0.2556)	0.1819(0.2551)	0.0(0.0)	0.2262(0.2065)	0.1824(0.2558)	0.2966(0.1662)
AUC ROC	0.5412(0.0642)	0.5409(0.0628)	0.4991(0.0011)	0.5403(0.0394)	0.5414(0.0646)	0.553(0.03)
Accuracy	0.9883(0.0034)	0.9878(0.0021)	0.9891(0.0026)	0.9865(0.0026)	0.9888(0.0046)	0.9873(0.0032)

Table A.167: Neural Network - Testing - Part 2

NN	TRD		
Optimized For	**F2**	**G-mean**	**PR-Recall**
F2	0.0(0.0)	0.0(0.0)	0.0(0.0)
Gmean	0.0(0.0)	0.0(0.0)	0.0(0.0)
AUC ROC	0.5(0.0)	0.5(0.0)	0.5(0.0)
Accuracy	0.9908(0.0006)	0.9908(0.0006)	0.9908(0.0006)

Table A.168: Neural Network - Testing - Part 3

A.4. Classification For CNS. Classification For CNS

ADABOOST	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.2342(0.0516)	0.2418(0.0441)	0.2274(0.0385)
Gmean	0.3671(0.0652)	0.3874(0.0617)	0.3547(0.0483)
AUC ROC	0.6245(0.0265)	0.6284(0.0249)	0.6187(0.0229)
Accuracy	0.9847(0.0017)	0.9848(0.0017)	0.9851(0.0014)

Table A.169: ADABOOST - Training

ADABOOST	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.2431(0.1397)	0.1635(0.112)	0.1878(0.1749)
Gmean	0.4497(0.2565)	0.3633(0.2175)	0.3432(0.317)
AUC ROC	0.6241(0.0783)	0.5808(0.0603)	0.5956(0.0956)
Accuracy	0.9845(0.0042)	0.9829(0.0034)	0.9842(0.0034)

Table A.170: ADABOOST - Testing

ADACOST	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.392(0.0052)	0.3915(0.0046)	0.3918(0.0054)
Gmean	0.9639(0.0007)	0.9639(0.0007)	0.9639(0.0007)
AUC ROC	0.9646(0.0007)	0.9646(0.0007)	0.9646(0.0007)
Accuracy	0.9297(0.0013)	0.9297(0.0013)	0.9297(0.0013)

Table A.171: ADACOST - Training

ADACOST	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.3887(0.0175)	0.3887(0.0175)	0.3887(0.0175)
Gmean	0.9639(0.0027)	0.9639(0.0027)	0.9639(0.0027)
AUC ROC	0.9646(0.0026)	0.9646(0.0026)	0.9646(0.0026)
Accuracy	0.9297(0.0052)	0.9297(0.0052)	0.9297(0.0052)

Table A.172: ADACOST - Testing

CATBOOST	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.1351(0.0392)	0.14(0.0453)	0.1307(0.0425)
Gmean	0.1962(0.0582)	0.2063(0.0647)	0.1862(0.0629)
AUC ROC	0.5606(0.0187)	0.5628(0.0206)	0.56(0.0204)
Accuracy	0.9912(0.0002)	0.9912(0.0003)	0.991(0.0001)

Table A.173: CATBOOST - Training

CATBOOST	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.1312(0.1356)	0.1312(0.1356)	0.1312(0.1356)
Gmean	0.2579(0.244)	0.2579(0.244)	0.2579(0.244)
AUC ROC	0.5564(0.0599)	0.5564(0.0599)	0.5564(0.0599)
Accuracy	0.9906(0.0014)	0.9906(0.0014)	0.9906(0.0014)

Table A.174: CATBOOST - Testing

DT	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.2359(0.0683)	0.206(0.0462)	0.2324(0.0466)
Gmean	0.3766(0.098)	0.3432(0.0691)	0.3678(0.0598)
AUC ROC	0.6231(0.038)	0.6066(0.0226)	0.6211(0.0228)
Accuracy	0.9852(0.0009)	0.9842(0.0013)	0.9846(0.0017)

Table A.175: Decision Tree - Training

DT	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.1796(0.1361)	0.2323(0.0863)	0.2169(0.1466)
Gmean	0.3869(0.245)	0.4931(0.1165)	0.4185(0.2534)
AUC ROC	0.5952(0.0799)	0.6233(0.0577)	0.6106(0.0802)
Accuracy	0.9835(0.0018)	0.9829(0.0059)	0.9857(0.0028)

Table A.176: Decision Tree - Testing

DT CS	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.264(0.0351)	0.2609(0.0129)	0.2802(0.0296)
Gmean	0.4075(0.0528)	0.4036(0.0183)	0.421(0.0411)
AUC ROC	0.639(0.0213)	0.6353(0.0084)	0.6455(0.0181)
Accuracy	0.9861(0.0009)	0.9854(0.0012)	0.9859(0.0018)

Table A.177: Cost Sensitive Decision Tree - Training

DT CS	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.1724(0.191)	0.2136(0.1752)	0.1894(0.1854)
Gmean	0.3124(0.3017)	0.4112(0.268)	0.3765(0.2659)
AUC ROC	0.5816(0.0945)	0.6097(0.096)	0.5963(0.1078)
Accuracy	0.9845(0.0046)	0.984(0.0023)	0.9855(0.0029)

Table A.178: Cost Sensitive Decision Tree - Testing

DT HELLINGER	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.2138(0.0288)	0.2071(0.0296)	0.223(0.0295)
Gmean	0.3377(0.0478)	0.3296(0.0421)	0.3599(0.0377)
AUC ROC	0.6119(0.0182)	0.6079(0.0162)	0.616(0.0196)
Accuracy	0.9849(0.0006)	0.9846(0.0008)	0.9853(0.0005)

Table A.179: Decision Tree Hellinger Distance - Training

DT HELLINGER	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.262(0.2475)	0.2714(0.2559)	0.2133(0.2037)
Gmean	0.4112(0.3777)	0.4116(0.378)	0.3671(0.3387)
AUC ROC	0.6385(0.1336)	0.6394(0.1335)	0.6103(0.1078)
Accuracy	0.985(0.0032)	0.9868(0.0031)	0.9852(0.0028)

Table A.180: Decision Tree Hellinger Distance - Testing

DT SMOTENC	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.2534(0.0379)	0.2559(0.0505)	0.2311(0.0281)
Gmean	0.4286(0.0625)	0.4195(0.0547)	0.3823(0.0442)
AUC ROC	0.642(0.0216)	0.6407(0.0278)	0.6282(0.0148)
Accuracy	0.9812(0.0013)	0.982(0.0013)	0.9822(0.0016)

Table A.181: Decision Tree SMOTENC - Training

DT SMOTENC	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.2567(0.1851)	0.1833(0.15)	0.3165(0.134)
Gmean	0.5229(0.2124)	0.406(0.2733)	0.5929(0.1374)
AUC ROC	0.6502(0.1268)	0.6076(0.1056)	0.6795(0.095)
Accuracy	0.9801(0.0026)	0.9799(0.0038)	0.9822(0.0018)

Table A.182: Decision Tree SMOTENC - Testing

DT SMOTETOMEK	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.3559(0.0292)	0.3593(0.0308)	0.2763(0.0312)
Gmean	0.7123(0.0322)	0.7261(0.0488)	0.4323(0.0486)
AUC ROC	0.7826(0.0231)	0.788(0.0272)	0.6588(0.0234)
Accuracy	0.9599(0.0091)	0.9617(0.0061)	0.9818(0.0016)

Table A.183: Decision Tree SMOTETOMEK - Training

DT SMOTETOMEK	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.3476(0.1998)	0.4443(0.1128)	0.2038(0.1829)
Gmean	0.7152(0.2249)	0.8257(0.1136)	0.407(0.2747)
AUC ROC	0.7635(0.1641)	0.8419(0.1)	0.6093(0.1075)
Accuracy	0.9521(0.0341)	0.9672(0.0111)	0.9832(0.004)

Table A.184: Decision Tree SMOTETOMEK - Testing

KNN	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.0926(0.0604)	0.087(0.0564)	0.0935(0.059)	0.1616(0.0204)	0.1512(0.0189)	0.1524(0.0444)
Gmean	0.1501(0.0949)	0.1411(0.0885)	0.1525(0.094)	0.2617(0.0348)	0.253(0.0436)	0.2478(0.0699)
AUC ROC	0.5452(0.03)	0.5419(0.0276)	0.5468(0.03)	0.5813(0.0104)	0.5766(0.0126)	0.5796(0.0232)
Accuracy	0.9857(0.0031)	0.9858(0.003)	0.9856(0.003)	0.9841(0.0006)	0.9837(0.0006)	0.984(0.0006)

Table A.185: KNN - Training - Part 1

KNN	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.2211(0.0786)	0.199(0.0857)	0.1718(0.0987)	0.204(0.0326)	0.1881(0.0406)	0.163(0.0603)
Gmean	0.3565(0.1136)	0.3169(0.1232)	0.2728(0.1598)	0.3328(0.0568)	0.312(0.0616)	0.2521(0.1089)
AUC ROC	0.6185(0.0437)	0.6051(0.0505)	0.5877(0.0575)	0.6081(0.0199)	0.597(0.0224)	0.5794(0.0359)
Accuracy	0.9847(0.0011)	0.9844(0.0013)	0.9871(0.0033)	0.9838(0.0011)	0.9837(0.0011)	0.9881(0.003)

Table A.186: KNN - Training - Part 2

KNN	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.2508(0.0297)	0.2459(0.0358)	0.2599(0.0502)
Gmean	0.3648(0.0326)	0.3661(0.0547)	0.3842(0.075)
AUC ROC	0.6262(0.0187)	0.6235(0.0209)	0.6312(0.0251)
Accuracy	0.9881(0.0004)	0.9882(0.0006)	0.9882(0.0005)

Table A.187: KNN - Training - Part 3

KNN	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.0294(0.0658)	0.0294(0.0658)	0.0294(0.0658)	0.1654(0.0479)	0.1654(0.0479)	0.1654(0.0479)
Gmean	0.0753(0.1685)	0.0753(0.1685)	0.0753(0.1685)	0.4075(0.0691)	0.4075(0.0691)	0.4075(0.0691)
AUC ROC	0.5113(0.0318)	0.5113(0.0318)	0.5113(0.0318)	0.5815(0.0307)	0.5815(0.0307)	0.5815(0.0307)
Accuracy	0.9855(0.0034)	0.9855(0.0034)	0.9855(0.0034)	0.9842(0.0029)	0.9842(0.0029)	0.9842(0.0029)

Table A.188: KNN - Testing - Part 1

KNN	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.2128(0.1406)	0.2128(0.1406)	0.1044(0.161)	0.2002(0.1699)	0.2002(0.1699)	0.1508(0.1817)
Gmean	0.4184(0.2532)	0.4184(0.2532)	0.2055(0.2976)	0.4069(0.2742)	0.4069(0.2742)	0.3008(0.3139)
AUC ROC	0.6098(0.0808)	0.6098(0.0808)	0.5541(0.0914)	0.6095(0.1063)	0.6095(0.1063)	0.5829(0.1142)
Accuracy	0.9842(0.0034)	0.9842(0.0034)	0.986(0.0041)	0.9837(0.0039)	0.9837(0.0039)	0.987(0.0049)

Table A.189: KNN - Testing - Part 2

KNN	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0(0.0)	0.0(0.0)	0.0(0.0)
Gmean	0.0(0.0)	0.0(0.0)	0.0(0.0)
AUC ROC	0.5(0.0)	0.5(0.0)	0.5(0.0)
Accuracy	0.9911(0.0)	0.9911(0.0)	0.9911(0.0)

Table A.190: KNN - Testing - Part 3

LR	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.0606(0.0748)	0.0545(0.0675)	0.0612(0.0637)	0.0(0.0)	0.0(0.0)	0.0(0.0)
Gmean	0.0888(0.1058)	0.0783(0.0974)	0.0897(0.091)	0.0(0.0)	0.0(0.0)	0.0(0.0)
AUC ROC	0.5268(0.0338)	0.524(0.0306)	0.5272(0.0288)	0.5(0.0)	0.5(0.0)	0.5(0.0)
Accuracy	0.9907(0.0003)	0.9906(0.0008)	0.9904(0.0008)	0.9911(0.0)	0.9911(0.0)	0.9911(0.0)

Table A.191: Logistic Regression - Training - Part 1

LR	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.034(0.062)	0.0364(0.068)	0.0277(0.044)	0.0369(0.0478)	0.0331(0.054)	0.0289(0.0577)
Gmean	0.0488(0.0881)	0.0506(0.0931)	0.0401(0.0626)	0.0547(0.0705)	0.046(0.0724)	0.0411(0.0815)
AUC ROC	0.5148(0.0273)	0.5164(0.0313)	0.512(0.0194)	0.516(0.021)	0.5148(0.0246)	0.5127(0.0256)
Accuracy	0.9909(0.0004)	0.9908(0.0004)	0.9908(0.0005)	0.9911(0.0003)	0.991(0.0004)	0.9912(0.0004)

Table A.192: Logistic Regression - Training - Part 2

LR	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0063(0.014)	0.0063(0.014)	0.0(0.0)
Gmean	0.0086(0.0191)	0.0086(0.0191)	0.0(0.0)
AUC ROC	0.5028(0.0062)	0.5028(0.0062)	0.5(0.0)
Accuracy	0.9911(0.0001)	0.9911(0.0001)	0.9911(0.0)

Table A.193: Logistic Regression - Training - Part 3

LR	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.0333(0.0745)	0.0333(0.0745)	0.0333(0.0745)	0.0(0.0)	0.0(0.0)	0.0(0.0)
Gmean	0.0755(0.1689)	0.0755(0.1689)	0.0755(0.1689)	0.0(0.0)	0.0(0.0)	0.0(0.0)
AUC ROC	0.5138(0.0319)	0.5138(0.0319)	0.5138(0.0319)	0.5(0.0)	0.5(0.0)	0.5(0.0)
Accuracy	0.9903(0.0007)	0.9903(0.0007)	0.9903(0.0007)	0.9911(0.0)	0.9911(0.0)	0.9911(0.0)

Table A.194: Logistic Regression - Testing - Part 1

LR	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.0345(0.0771)	0.0345(0.0771)	0.0345(0.0771)	0.0345(0.0771)	0.0345(0.0771)	0.0345(0.0771)
Gmean	0.0756(0.169)	0.0756(0.169)	0.0756(0.169)	0.0756(0.169)	0.0756(0.169)	0.0756(0.169)
AUC ROC	0.5138(0.0323)	0.5138(0.0323)	0.514(0.0321)	0.5139(0.0322)	0.5139(0.0322)	0.5139(0.0322)
Accuracy	0.9903(0.0025)	0.9903(0.0025)	0.9908(0.0014)	0.9906(0.0019)	0.9906(0.0019)	0.9906(0.0019)

Table A.195: Logistic Regression - Testing - Part 2

LR	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.1198(0.1871)	0.1198(0.1871)	0.1198(0.1871)
Gmean	0.2695(0.4241)	0.2695(0.4241)	0.2695(0.4241)
AUC ROC	0.608(0.2047)	0.608(0.2047)	0.6081(0.2046)
Accuracy	0.9807(0.0227)	0.9807(0.0227)	0.9809(0.0228)

Table A.196: Logistic Regression - Testing - Part 3

LR CS	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.4243(0.0079)	0.4176(0.0152)	0.408(0.0191)	0.4568(0.0097)	0.4521(0.0108)	0.455(0.0148)
Gmean	0.8126(0.0715)	0.8665(0.0186)	0.775(0.0826)	0.9127(0.054)	0.9413(0.0099)	0.8977(0.0676)
AUC ROC	0.8468(0.047)	0.8805(0.0148)	0.8259(0.0534)	0.921(0.0446)	0.9446(0.0087)	0.913(0.0512)
Accuracy	0.9627(0.0099)	0.9552(0.0019)	0.9648(0.0101)	0.9559(0.0085)	0.9502(0.002)	0.9564(0.0079)

Table A.197: Cost Sensitive Logistic Regression - Training - Part 1

LR CS	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.4417(0.0132)	0.4368(0.0114)	0.4301(0.0206)	0.439(0.0134)	0.4442(0.0219)	0.4369(0.007)
Gmean	0.8917(0.0089)	0.8871(0.0119)	0.7886(0.0618)	0.8798(0.0672)	0.9107(0.0218)	0.8777(0.0494)
AUC ROC	0.9008(0.0084)	0.9001(0.0085)	0.832(0.0462)	0.8952(0.0491)	0.9206(0.0159)	0.8939(0.039)
Accuracy	0.9561(0.0029)	0.9558(0.0025)	0.9672(0.0067)	0.9569(0.0081)	0.9529(0.0019)	0.9567(0.0083)

Table A.198: Cost Sensitive Logistic Regression - Training - Part 2

LR CS	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.4524(0.0109)	0.4274(0.0341)	0.4522(0.0222)
Gmean	0.8472(0.0759)	0.9444(0.0096)	0.8685(0.0974)
AUC ROC	0.8748(0.0547)	0.947(0.0088)	0.895(0.0662)
Accuracy	0.9635(0.0096)	0.9419(0.0107)	0.9599(0.0087)

Table A.199: Cost Sensitive Logistic Regression - Training - Part 3

LR CS	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.4243(0.0237)	0.4102(0.0711)	0.4408(0.0612)	0.4594(0.0376)	0.4471(0.044)	0.437(0.0324)
Gmean	0.8419(0.0863)	0.8706(0.0903)	0.8432(0.0877)	0.9335(0.0386)	0.9454(0.0395)	0.9006(0.0928)
AUC ROC	0.8536(0.0722)	0.8779(0.0806)	0.8549(0.0738)	0.9344(0.038)	0.9461(0.039)	0.9066(0.0819)
Accuracy	0.9623(0.0109)	0.9544(0.0031)	0.9649(0.0111)	0.9542(0.006)	0.9493(0.0061)	0.9552(0.0103)

Table A.200: Cost Sensitive Logistic Regression - Testing - Part 1

LR CS	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.4248(0.0372)	0.4257(0.0364)	0.4695(0.039)	0.4392(0.0441)	0.4454(0.0383)	0.4362(0.044)
Gmean	0.8879(0.061)	0.8881(0.0613)	0.8634(0.0406)	0.8862(0.0985)	0.9178(0.061)	0.9021(0.0721)
AUC ROC	0.8922(0.0569)	0.8923(0.0571)	0.87(0.036)	0.8939(0.0879)	0.9203(0.0577)	0.9062(0.0677)
Accuracy	0.9547(0.0073)	0.9549(0.0064)	0.9669(0.0082)	0.958(0.0104)	0.9542(0.0042)	0.9544(0.0093)

Table A.201: Cost Sensitive Logistic Regression - Testing - Part 2

LR CS	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.1416(0.2026)	0.2603(0.1784)	0.0875(0.1957)
Gmean	0.3006(0.4398)	0.5511(0.3683)	0.1941(0.4341)
AUC ROC	0.6216(0.2044)	0.703(0.1708)	0.5942(0.2107)
Accuracy	0.9796(0.0208)	0.9725(0.0202)	0.9814(0.0216)

Table A.202: Cost Sensitive Logistic Regression - Testing - Part 3

LR SMOTENC	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.4017(0.0487)	0.4031(0.0556)	0.409(0.0437)	0.4335(0.0317)	0.4019(0.0173)	0.4264(0.0243)
Gmean	0.7712(0.0419)	0.7471(0.0694)	0.7715(0.0579)	0.8151(0.0403)	0.7973(0.0436)	0.817(0.0189)
AUC ROC	0.8093(0.0308)	0.8131(0.0388)	0.8159(0.0354)	0.8514(0.0247)	0.8361(0.0244)	0.8576(0.0104)
Accuracy	0.9669(0.0052)	0.9667(0.0044)	0.9669(0.0043)	0.9641(0.0054)	0.961(0.0041)	0.9621(0.0034)

Table A.203: Logistic Regression SMOTENC - Training - Part 1

LR SMOTENC	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.4247(0.0194)	0.4171(0.0364)	0.4438(0.0453)	0.3996(0.0328)	0.3801(0.0177)	0.3959(0.0345)
Gmean	0.7767(0.0172)	0.769(0.0322)	0.8034(0.0558)	0.7235(0.0496)	0.7284(0.035)	0.7459(0.0467)
AUC ROC	0.8293(0.0088)	0.8238(0.0133)	0.8456(0.0305)	0.7989(0.0307)	0.7906(0.0194)	0.8074(0.028)
Accuracy	0.9673(0.0042)	0.9672(0.0045)	0.9679(0.0042)	0.9692(0.0031)	0.9668(0.0028)	0.9664(0.0035)

Table A.204: Logistic Regression SMOTENC - Training - Part 2

LR SMOTENC	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.4397(0.0467)	0.4271(0.0394)	0.4205(0.0575)
Gmean	0.8509(0.0396)	0.8369(0.0423)	0.8151(0.0603)
AUC ROC	0.875(0.0298)	0.8628(0.0258)	0.8514(0.0341)
Accuracy	0.9618(0.0039)	0.9605(0.0039)	0.9619(0.0045)

Table A.205: Logistic Regression SMOTENC - Training - Part 3

LR SMOTENC	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.4528(0.0763)	0.4528(0.0763)	0.4528(0.0763)	0.4518(0.0767)	0.4553(0.0306)	0.4507(0.0353)
Gmean	0.846(0.0693)	0.846(0.0693)	0.846(0.0693)	0.8742(0.0903)	0.8919(0.0339)	0.8769(0.0418)
AUC ROC	0.8558(0.0595)	0.8558(0.0595)	0.8558(0.0595)	0.8819(0.0802)	0.895(0.0303)	0.8816(0.0374)
Accuracy	0.9666(0.0061)	0.9666(0.0061)	0.9666(0.0061)	0.9623(0.0055)	0.9603(0.0058)	0.9618(0.0059)

Table A.206: Logistic Regression SMOTENC - Testing - Part 1

LR SMOTENC	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.5004(0.052)	0.4845(0.0715)	0.4956(0.0512)	0.4548(0.0697)	0.4782(0.0394)	0.4909(0.0672)
Gmean	0.8793(0.0668)	0.8618(0.0893)	0.879(0.0664)	0.8313(0.0593)	0.8642(0.0418)	0.8798(0.044)
AUC ROC	0.8857(0.062)	0.8716(0.0795)	0.8854(0.0617)	0.8429(0.0506)	0.8708(0.0373)	0.8847(0.0397)
Accuracy	0.97(0.0043)	0.97(0.0043)	0.9692(0.0055)	0.9692(0.0059)	0.9684(0.0061)	0.9679(0.0065)

Table A.207: Logistic Regression SMOTENC - Testing - Part 2

LR SMOTENC	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.3155(0.185)	0.2739(0.1925)	0.2219(0.2184)
Gmean	0.6035(0.365)	0.6021(0.4017)	0.437(0.4207)
AUC ROC	0.7326(0.163)	0.7426(0.194)	0.6637(0.1779)
Accuracy	0.9751(0.0156)	0.9669(0.0203)	0.9789(0.0136)

Table A.208: Logistic Regression SMOTENC - Testing - Part 3

LR SMOTETOMEK	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.4246(0.0163)	0.4302(0.0174)	0.4305(0.0131)	0.4607(0.0099)	0.4558(0.0173)	0.4638(0.0222)
Gmean	0.8326(0.0259)	0.8384(0.0317)	0.835(0.0317)	0.9132(0.0196)	0.9281(0.0204)	0.9153(0.0332)
AUC ROC	0.8568(0.0172)	0.8639(0.0222)	0.8611(0.0209)	0.9204(0.0163)	0.933(0.018)	0.9239(0.0237)
Accuracy	0.9616(0.0023)	0.9616(0.0024)	0.9615(0.002)	0.9567(0.0024)	0.9534(0.0027)	0.9562(0.0025)

Table A.209: Logistic Regression SMOTETOMEK - Training - Part 1

LR SMOTETOMEK	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.4511(0.0148)	0.4547(0.0143)	0.455(0.0128)	0.4332(0.0248)	0.4321(0.0218)	0.4305(0.0241)
Gmean	0.8651(0.0302)	0.8714(0.0287)	0.8716(0.0168)	0.8509(0.0225)	0.854(0.0239)	0.8478(0.027)
AUC ROC	0.8795(0.0256)	0.8874(0.0208)	0.885(0.014)	0.8781(0.021)	0.8742(0.0151)	0.8696(0.0185)
Accuracy	0.9619(0.0045)	0.9613(0.0021)	0.9619(0.0023)	0.9591(0.0025)	0.959(0.0023)	0.9599(0.0038)

Table A.210: Logistic Regression SMOTETOMEK - Training - Part 2

LR SMOTETOMEK	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.4547(0.0093)	0.4496(0.0103)	0.465(0.0168)
Gmean	0.9012(0.0293)	0.9112(0.0184)	0.9158(0.0239)
AUC ROC	0.913(0.0213)	0.9176(0.0161)	0.9222(0.022)
Accuracy	0.9563(0.0022)	0.9545(0.0014)	0.9571(0.0036)

Table A.211: Logistic Regression SMOTETOMEK - Training - Part 3

LR SMOTETOMEK	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.4476(0.0725)	0.4461(0.0718)	0.4461(0.0718)	0.4381(0.0416)	0.4508(0.034)	0.4358(0.0433)
Gmean	0.8751(0.0748)	0.875(0.0748)	0.875(0.0748)	0.9038(0.0527)	0.9327(0.0392)	0.9036(0.0527)
AUC ROC	0.8814(0.0643)	0.8812(0.0642)	0.8812(0.0642)	0.9065(0.0496)	0.9336(0.0386)	0.9062(0.0496)
Accuracy	0.9613(0.0037)	0.961(0.0037)	0.961(0.0037)	0.9549(0.0033)	0.9526(0.0036)	0.9544(0.0043)

Table A.212: Logistic Regression SMOTETOMEK - Testing - Part 1

LR SMOTETOMEK	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.4441(0.028)	0.4397(0.025)	0.4443(0.0233)	0.4638(0.0688)	0.4638(0.0688)	0.4543(0.0549)
Gmean	0.8765(0.0418)	0.8761(0.0418)	0.8764(0.0413)	0.9049(0.0755)	0.9049(0.0755)	0.8907(0.063)
AUC ROC	0.8811(0.0375)	0.8807(0.0374)	0.8811(0.0369)	0.9089(0.071)	0.9089(0.071)	0.895(0.0588)
Accuracy	0.9608(0.0042)	0.96(0.0038)	0.9608(0.005)	0.9598(0.0037)	0.9598(0.0037)	0.9603(0.0045)

Table A.213: Logistic Regression SMOTETOMEK - Testing - Part 2

LR SMOTETOMEK	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.3318(0.1967)	0.2967(0.1693)	0.2925(0.1674)
Gmean	0.6904(0.3957)	0.7136(0.4092)	0.6075(0.3513)
AUC ROC	0.7976(0.1838)	0.8191(0.1959)	0.7292(0.1471)
Accuracy	0.9636(0.0165)	0.9501(0.0262)	0.9684(0.0181)

Table A.214: Logistic Regression SMOTETOMEK - Testing - Part 3

RF	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0698(0.0417)	0.066(0.0341)	0.0771(0.042)
Gmean	0.1006(0.0609)	0.0979(0.0504)	0.1098(0.059)
AUC ROC	0.5304(0.0182)	0.5288(0.015)	0.5338(0.0186)
Accuracy	0.9914(0.0003)	0.9914(0.0003)	0.9915(0.0003)

Table A.215: Random Forest - Training

RF	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0345(0.0771)	0.0333(0.0745)	0.069(0.0944)
Gmean	0.0756(0.169)	0.0755(0.1689)	0.1512(0.207)
AUC ROC	0.5142(0.032)	0.5139(0.0318)	0.5283(0.0394)
Accuracy	0.9911(0.0009)	0.9906(0.0007)	0.9911(0.0013)

Table A.216: Random Forest - Testing

RF CS	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.3223(0.0498)	0.3225(0.037)	0.2152(0.1083)
Gmean	0.5424(0.0577)	0.547(0.0482)	0.313(0.1766)
AUC ROC	0.6964(0.0323)	0.698(0.0221)	0.6111(0.0751)
Accuracy	0.9797(0.0014)	0.9797(0.0023)	0.9887(0.0039)

Table A.217: Cost Sensitive Random Forest - Training

RF CS	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.4754(0.0935)	0.4747(0.0549)	0.2256(0.1818)
Gmean	0.7463(0.0939)	0.7845(0.0482)	0.4116(0.2679)
AUC ROC	0.7794(0.0703)	0.8053(0.0389)	0.6117(0.0941)
Accuracy	0.9837(0.0036)	0.9789(0.0037)	0.988(0.0059)

Table A.218: Cost Sensitive Random Forest - Testing

SVM	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)
Gmean	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)
AUC ROC	0.5(0.0001)	0.5(0.0)	0.5(0.0)	0.5(0.0)	0.5(0.0)	0.5(0.0)
Accuracy	0.991(0.0001)	0.991(0.0001)	0.991(0.0001)	0.9911(0.0)	0.9911(0.0)	0.9911(0.0)

Table A.219: SVM - Training - Part 1

SVM	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.0826(0.0471)	0.081(0.0496)	0.0812(0.0553)	0.0786(0.0572)	0.0735(0.0532)	0.0745(0.0541)
Gmean	0.1251(0.0714)	0.1221(0.0734)	0.1199(0.0799)	0.1181(0.0846)	0.109(0.0791)	0.1083(0.0801)
AUC ROC	0.5376(0.0216)	0.537(0.0231)	0.5369(0.0255)	0.5357(0.0264)	0.5341(0.0256)	0.5335(0.0242)
Accuracy	0.9902(0.0005)	0.9902(0.0006)	0.99(0.0007)	0.9898(0.0009)	0.9899(0.0007)	0.9897(0.0007)

Table A.220: SVM - Training - Part 2

SVM	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0724(0.0512)	0.0715(0.0488)	0.0736(0.0519)
Gmean	0.1052(0.0771)	0.1065(0.0725)	0.1081(0.0752)
AUC ROC	0.532(0.0226)	0.5314(0.0216)	0.5325(0.023)
Accuracy	0.9912(0.0004)	0.9911(0.0004)	0.9912(0.0003)

Table A.221: SVM - Training - Part 3

SVM	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)
Gmean	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)
AUC ROC	0.5(0.0)	0.5(0.0)	0.5(0.0)	0.5(0.0)	0.5(0.0)	0.5(0.0)
Accuracy	0.9911(0.0)	0.9911(0.0)	0.9911(0.0)	0.9911(0.0)	0.9911(0.0)	0.9911(0.0)

Table A.222: SVM - Testing - Part 1

SVM	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.0333(0.0745)	0.0333(0.0745)	0.0333(0.0745)	0.0333(0.0745)	0.0333(0.0745)	0.0333(0.0745)
Gmean	0.0755(0.1689)	0.0755(0.1689)	0.0755(0.1689)	0.0755(0.1689)	0.0755(0.1689)	0.0755(0.1689)
AUC ROC	0.5135(0.032)	0.5135(0.032)	0.5135(0.032)	0.5131(0.0323)	0.5131(0.0323)	0.5131(0.0323)
Accuracy	0.9898(0.0018)	0.9898(0.0018)	0.9898(0.0018)	0.9891(0.0033)	0.9891(0.0033)	0.9891(0.0033)

Table A.223: SVM - Testing - Part 2

SVM	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0232(0.0349)	0.0232(0.0349)	0.0232(0.0349)
Gmean	0.1415(0.2529)	0.1415(0.2529)	0.1415(0.2529)
AUC ROC	0.5098(0.063)	0.5098(0.063)	0.5098(0.063)
Accuracy	0.758(0.4186)	0.758(0.4186)	0.758(0.4186)

Table A.224: SVM - Testing - Part 3

SVM CS	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.4051(0.0069)	0.3898(0.0103)	0.3837(0.0238)	0.4641(0.0186)	0.4488(0.0297)	0.4614(0.0176)
Gmean	0.799(0.0381)	0.8022(0.0224)	0.7442(0.0531)	0.8813(0.0662)	0.9146(0.031)	0.8715(0.0693)
AUC ROC	0.8363(0.0188)	0.8301(0.011)	0.8077(0.0288)	0.896(0.0534)	0.9236(0.0269)	0.8924(0.0524)
Accuracy	0.9626(0.0052)	0.9601(0.0015)	0.9636(0.0061)	0.962(0.0067)	0.9534(0.0029)	0.9623(0.0087)

Table A.225: Cost Sensitive SVM - Training - Part 1

SVM CS	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.424(0.0282)	0.4217(0.0334)	0.4214(0.0525)	0.4032(0.0205)	0.3895(0.0223)	0.3646(0.0542)
Gmean	0.7642(0.0909)	0.8056(0.0476)	0.7293(0.126)	0.774(0.0448)	0.7644(0.0503)	0.6631(0.1514)
AUC ROC	0.823(0.0533)	0.838(0.0364)	0.8026(0.0789)	0.8159(0.0255)	0.8169(0.0266)	0.7681(0.0777)
Accuracy	0.969(0.0074)	0.9649(0.0029)	0.9731(0.008)	0.9657(0.0042)	0.9623(0.0021)	0.9704(0.0088)

Table A.226: Cost Sensitive SVM - Training - Part 2

SVM CS	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.4538(0.0226)	0.4448(0.0086)	0.4362(0.0146)
Gmean	0.8312(0.1051)	0.906(0.0165)	0.7987(0.115)
AUC ROC	0.8614(0.0737)	0.9158(0.0128)	0.8437(0.0765)
Accuracy	0.9665(0.012)	0.9542(0.0023)	0.9663(0.0125)

Table A.227: Cost Sensitive SVM - Training - Part 3

SVM CS	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.4201(0.0355)	0.4102(0.053)	0.4271(0.0348)	0.4094(0.0475)	0.4544(0.0446)	0.4172(0.0533)
Gmean	0.8436(0.0669)	0.8428(0.0684)	0.8441(0.0668)	0.8415(0.0837)	0.9319(0.0658)	0.8433(0.0679)
AUC ROC	0.8532(0.057)	0.8522(0.0588)	0.8538(0.0568)	0.8523(0.0753)	0.9342(0.0622)	0.8528(0.0581)
Accuracy	0.9616(0.0051)	0.9595(0.0025)	0.9628(0.0055)	0.9598(0.0047)	0.9537(0.0033)	0.9608(0.0054)

Table A.228: Cost Sensitive SVM - Testing - Part 1

SVM CS	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.4383(0.0696)	0.4366(0.0672)	0.3867(0.0748)	0.4372(0.0379)	0.4195(0.0551)	0.3893(0.0547)
Gmean	0.8285(0.0821)	0.8448(0.0689)	0.7525(0.1402)	0.8449(0.0671)	0.8436(0.0686)	0.7753(0.098)
AUC ROC	0.8417(0.0696)	0.8545(0.0591)	0.7866(0.1)	0.8547(0.0571)	0.8531(0.0589)	0.7993(0.0777)
Accuracy	0.9669(0.0072)	0.9641(0.0054)	0.97(0.0113)	0.9646(0.005)	0.9613(0.0021)	0.9669(0.0061)

Table A.229: Cost Sensitive SVM - Testing - Part 2

SVM CS	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.093(0.208)	0.0862(0.1928)	0.093(0.208)
Gmean	0.1501(0.3357)	0.1663(0.3719)	0.1501(0.3357)
AUC ROC	0.5557(0.1246)	0.5682(0.1525)	0.5557(0.1246)
Accuracy	0.9893(0.004)	0.986(0.0114)	0.9893(0.004)

Table A.230: Cost Sensitive SVM - Testing - Part 3

SVM SMOTENC	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.0279(0.046)	0.0282(0.0595)	0.0313(0.0631)	0.3275(0.0431)	0.3409(0.057)	0.3398(0.0555)
Gmean	0.0515(0.0866)	0.056(0.1149)	0.0554(0.1133)	0.6301(0.0662)	0.6628(0.0866)	0.6562(0.0919)
AUC ROC	0.5118(0.0243)	0.5129(0.0333)	0.5132(0.032)	0.7411(0.0343)	0.7518(0.0501)	0.7485(0.0439)
Accuracy	0.984(0.0056)	0.9828(0.0067)	0.9846(0.0054)	0.9672(0.0053)	0.9674(0.0047)	0.9676(0.0046)

Table A.231: SVM SMOTENC - Training - Part 1

SVM SMOTENC	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.3198(0.0362)	0.2965(0.0476)	0.3026(0.0276)	0.3017(0.0318)	0.3171(0.0277)	0.286(0.037)
Gmean	0.5601(0.052)	0.503(0.0853)	0.5254(0.0464)	0.5225(0.0515)	0.554(0.0481)	0.4999(0.0629)
AUC ROC	0.6948(0.0242)	0.682(0.0334)	0.6862(0.0181)	0.6917(0.0265)	0.6986(0.0263)	0.6795(0.0261)
Accuracy	0.9789(0.0016)	0.9787(0.0015)	0.9781(0.0016)	0.9772(0.0021)	0.9777(0.0027)	0.978(0.0022)

Table A.232: SVM SMOTENC - Training - Part 2

SVM SMOTENC	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.3614(0.0637)	0.3443(0.0554)	0.3454(0.0295)
Gmean	0.6314(0.0605)	0.6025(0.064)	0.5964(0.0447)
AUC ROC	0.7311(0.0382)	0.7218(0.0337)	0.7227(0.0219)
Accuracy	0.9767(0.0029)	0.9767(0.0025)	0.9763(0.0027)

Table A.233: SVM SMOTENC - Training - Part 3

SVM SMOTENC	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.3522(0.0942)	0.3522(0.0942)	0.3522(0.0942)
Gmean	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.7388(0.0933)	0.7388(0.0933)	0.7388(0.0933)
AUC ROC	0.4965(0.0058)	0.4965(0.0058)	0.4965(0.0058)	0.7695(0.0709)	0.7695(0.0709)	0.7695(0.0709)
Accuracy	0.9842(0.0116)	0.9842(0.0116)	0.9842(0.0116)	0.9641(0.0089)	0.9641(0.0089)	0.9641(0.0089)

Table A.234: SVM SMOTENC - Testing - Part 1

SVM SMOTENC	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.35(0.098)	0.35(0.098)	0.35(0.098)	0.3533(0.0443)	0.3533(0.0443)	0.3533(0.0443)
Gmean	0.6421(0.109)	0.6421(0.109)	0.6421(0.109)	0.6687(0.0444)	0.6687(0.0444)	0.6687(0.0444)
AUC ROC	0.7068(0.07)	0.7068(0.07)	0.7068(0.07)	0.7195(0.0314)	0.7195(0.0314)	0.7195(0.0314)
Accuracy	0.9801(0.0057)	0.9801(0.0057)	0.9801(0.0057)	0.9771(0.0049)	0.9771(0.0049)	0.9771(0.0049)

Table A.235: SVM SMOTENC - Testing - Part 2

SVM SMOTENC	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0(0.0)	0.0(0.0)	0.0(0.0)
Gmean	0.0(0.0)	0.0(0.0)	0.0(0.0)
AUC ROC	0.5(0.0)	0.5(0.0)	0.5(0.0)
Accuracy	0.9911(0.0)	0.9911(0.0)	0.9911(0.0)

Table A.236: SVM SMOTENC - Testing - Part 3

SVM SMOTETOMEK	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0228(0.0423)	0.0231(0.0411)	0.0242(0.0495)
Gmean	0.0467(0.0847)	0.0475(0.0845)	0.0482(0.0975)
AUC ROC	0.5091(0.0222)	0.5096(0.0222)	0.5098(0.028)
Accuracy	0.9818(0.0075)	0.9818(0.0081)	0.981(0.008)

Table A.237: SVM SMOTETOMEK - Training

SVM SMOTETOMEK	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0(0.0)	0.0(0.0)	0.0(0.0)
Gmean	0.0(0.0)	0.0(0.0)	0.0(0.0)
AUC ROC	0.4945(0.0089)	0.4945(0.0089)	0.4945(0.0089)
Accuracy	0.9801(0.0177)	0.9801(0.0177)	0.9801(0.0177)

Table A.238: SVM SMOTETOMEK - Testing

XGBoost	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.2096(0.0514)	0.2105(0.0514)	0.2241(0.04)
Gmean	0.2933(0.0624)	0.3056(0.0689)	0.3247(0.0566)
AUC ROC	0.5969(0.0242)	0.5953(0.0224)	0.601(0.0189)
Accuracy	0.9912(0.0008)	0.9912(0.0006)	0.9915(0.0004)

Table A.239: XGBoost - Training

XGBoost	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.196(0.1377)	0.1617(0.1144)	0.2583(0.1442)
Gmean	0.3647(0.2185)	0.3333(0.1983)	0.4643(0.1262)
AUC ROC	0.5848(0.0602)	0.5701(0.051)	0.6134(0.0645)
Accuracy	0.9908(0.0028)	0.9898(0.003)	0.9913(0.0029)

Table A.240: XGBoost - Testing

XGBoost CS	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.207(0.0497)	0.2021(0.0373)	0.1968(0.0378)
Gmean	0.2934(0.0705)	0.295(0.0509)	0.2754(0.0578)
AUC ROC	0.5969(0.0234)	0.5934(0.0195)	0.5905(0.0174)
Accuracy	0.9911(0.0003)	0.9908(0.0004)	0.9914(0.0006)

Table A.241: Cost Sensitive XGBoost - Training

XGBoost CS	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.2302(0.0895)	0.2252(0.0942)	0.1969(0.1373)
Gmean	0.4403(0.0858)	0.4401(0.0861)	0.3648(0.2185)
AUC ROC	0.5994(0.0394)	0.5987(0.04)	0.5849(0.0602)
Accuracy	0.9916(0.0019)	0.9903(0.0031)	0.9911(0.0024)

Table A.242: Cost Sensitive XGBoost - Testing

XGBoost SMOTENC	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.2746(0.0817)	0.2658(0.0577)	0.2588(0.0475)
Gmean	0.4517(0.1116)	0.4393(0.0689)	0.4374(0.0583)
AUC ROC	0.6585(0.0452)	0.6503(0.0335)	0.6455(0.024)
Accuracy	0.9812(0.0025)	0.9813(0.0018)	0.9815(0.0026)

Table A.243: XGBoost SMOTENC - Training

XGBoost SMOTENC	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.3087(0.0678)	0.3087(0.0678)	0.3087(0.0678)
Gmean	0.6021(0.0656)	0.6021(0.0656)	0.6021(0.0656)
AUC ROC	0.6783(0.0397)	0.6783(0.0397)	0.6783(0.0397)
Accuracy	0.9796(0.0024)	0.9796(0.0024)	0.9796(0.0024)

Table A.244: XGBoost SMOTENC - Testing

XGBoost SMOTETOMEK	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.2536(0.034)	0.2522(0.0524)	0.2541(0.0346)
Gmean	0.3882(0.0615)	0.3916(0.0832)	0.3881(0.05)
AUC ROC	0.6307(0.0202)	0.6308(0.0256)	0.6284(0.0144)
Accuracy	0.9861(0.0007)	0.9863(0.0014)	0.9869(0.0015)

Table A.245: XGBoost SMOTETOMEK - Training

XGBoost SMOTETOMEK	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.3546(0.1252)	0.3722(0.1396)	0.3723(0.1298)
Gmean	0.6005(0.1002)	0.6011(0.1005)	0.6012(0.1002)
AUC ROC	0.6817(0.0646)	0.6828(0.0648)	0.6829(0.0643)
Accuracy	0.9865(0.0039)	0.9885(0.0043)	0.9888(0.0041)

Table A.246: XGBoost SMOTETOMEK - Testing

NN	No Scaling			NORM		
Optimized For	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.0(0.0)	0.0045(0.0061)	0.0042(0.006)	0.0111(0.0166)	0.0051(0.0115)	0.0049(0.0068)
Gmean	0.0(0.0)	0.0077(0.0105)	0.0077(0.0105)	0.0141(0.0211)	0.0077(0.0172)	0.0077(0.0105)
AUC ROC	0.5(0.0)	0.5021(0.003)	0.5021(0.0029)	0.5049(0.0074)	0.5021(0.0049)	0.5021(0.003)
Accuracy	0.9911(0.0)	0.991(0.0001)	0.9909(0.0004)	0.9909(0.0002)	0.9908(0.0001)	0.9908(0.0002)

Table A.247: Neural Network - Training - Part 1

NN	POWER			STD		
Optimized For	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.0604(0.0298)	0.0464(0.0332)	0.0552(0.0291)	0.057(0.0358)	0.0527(0.0347)	0.0534(0.0246)
Gmean	0.0951(0.0441)	0.0711(0.0495)	0.0872(0.0452)	0.0881(0.0579)	0.0819(0.0542)	0.0877(0.0387)
AUC ROC	0.5283(0.0144)	0.5212(0.0163)	0.5261(0.0147)	0.5268(0.0176)	0.524(0.0166)	0.5252(0.0123)
Accuracy	0.9883(0.0007)	0.9884(0.0004)	0.9882(0.0004)	0.9886(0.0007)	0.9884(0.0007)	0.9887(0.0008)

Table A.248: Neural Network - Training - Part 2

NN	TRD		
Optimized For	**F2**	**G-mean**	**PR-Recall**
F2	0.0163(0.0058)	0.0259(0.0175)	0.021(0.0211)
Gmean	0.024(0.0083)	0.0393(0.0273)	0.0295(0.029)
AUC ROC	0.5069(0.0025)	0.5114(0.0079)	0.5091(0.0094)
Accuracy	0.9906(0.0003)	0.9907(0.0001)	0.9905(0.0002)

Table A.249: Neural Network - Training - Part 3

NN	No Scaling			NORM		
Optimized For	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.0(0.0)	0.0323(0.0721)	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)
Gmean	0.0(0.0)	0.0755(0.1688)	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)
AUC ROC	0.5(0.0)	0.514(0.0314)	0.5(0.0)	0.4996(0.0009)	0.4997(0.0006)	0.4996(0.0009)
Accuracy	0.9911(0.0)	0.9908(0.0006)	0.9911(0.0)	0.9903(0.0017)	0.9906(0.0011)	0.9903(0.0017)

Table A.250: Neural Network - Testing - Part 1

NN	POWER			STD		
Optimized For	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.0303(0.0678)	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0323(0.0721)	0.0(0.0)
Gmean	0.0754(0.1686)	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0755(0.1688)	0.0(0.0)
AUC ROC	0.5125(0.0315)	0.4985(0.0018)	0.4991(0.0014)	0.4994(0.0011)	0.5134(0.0318)	0.4995(0.0005)
Accuracy	0.9878(0.0019)	0.988(0.0037)	0.9893(0.0028)	0.9898(0.0022)	0.9896(0.0028)	0.9901(0.0011)

Table A.251: Neural Network - Testing - Part 2

NN	TRD		
Optimized For	**F2**	**G-mean**	**PR-Recall**
F2	0.0(0.0)	0.0(0.0)	0.0(0.0)
Gmean	0.0(0.0)	0.0(0.0)	0.0(0.0)
AUC ROC	0.5(0.0)	0.5(0.0)	0.5(0.0)
Accuracy	0.9911(0.0)	0.9911(0.0)	0.9911(0.0)

Table A.252: Neural Network - Testing - Part 3

A.5. Classification For GU. Classification For GU

ADABOOST	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.2619(0.0714)	0.2655(0.0683)	0.2566(0.0398)
Gmean	0.4145(0.0862)	0.4263(0.0918)	0.3992(0.0384)
AUC ROC	0.6399(0.0388)	0.6376(0.0348)	0.6335(0.023)
Accuracy	0.9832(0.0025)	0.9836(0.0016)	0.9836(0.0021)

Table A.253: ADABOOST - Training

ADABOOST	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.3385(0.0939)	0.3211(0.0839)	0.2906(0.1315)
Gmean	0.5839(0.0874)	0.5652(0.0623)	0.5323(0.1328)
AUC ROC	0.6708(0.0522)	0.6584(0.036)	0.6456(0.0718)
Accuracy	0.9852(0.0046)	0.9852(0.0046)	0.9842(0.0045)

Table A.254: ADABOOST - Testing

ADACOST	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.4274(0.0058)	0.4273(0.0058)	0.4275(0.0065)
Gmean	0.9649(0.0009)	0.9649(0.0009)	0.9649(0.0008)
AUC ROC	0.9655(0.0008)	0.9655(0.0008)	0.9655(0.0008)
Accuracy	0.9317(0.0016)	0.9317(0.0016)	0.9317(0.0016)

Table A.255: ADACOST - Training

ADACOST	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.4247(0.023)	0.4247(0.023)	0.4247(0.023)
Gmean	0.9649(0.0034)	0.9649(0.0034)	0.9649(0.0034)
AUC ROC	0.9655(0.0033)	0.9655(0.0033)	0.9655(0.0033)
Accuracy	0.9317(0.0065)	0.9317(0.0065)	0.9317(0.0065)

Table A.256: ADACOST - Testing

CATBOOST	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.1071(0.0325)	0.1231(0.0635)	0.1088(0.0341)
Gmean	0.165(0.0461)	0.1887(0.0894)	0.1684(0.0521)
AUC ROC	0.5487(0.0163)	0.556(0.0306)	0.5494(0.017)
Accuracy	0.989(0.0004)	0.9892(0.0002)	0.9887(0.0005)

Table A.257: CATBOOST - Training

CATBOOST	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0573(0.0787)	0.0573(0.0787)	0.0573(0.0787)
Gmean	0.1412(0.1934)	0.1412(0.1934)	0.1412(0.1934)
AUC ROC	0.5239(0.0341)	0.5239(0.0341)	0.5239(0.0341)
Accuracy	0.9882(0.0019)	0.9882(0.0019)	0.9882(0.0019)

Table A.258: CATBOOST - Testing

DT	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.2626(0.0809)	0.2814(0.0758)	0.2482(0.0676)
Gmean	0.4135(0.1121)	0.4442(0.0999)	0.3993(0.0784)
AUC ROC	0.6359(0.0468)	0.6461(0.0392)	0.629(0.0331)
Accuracy	0.9836(0.0013)	0.9842(0.0017)	0.984(0.0019)

Table A.259: Decision Tree - Training

DT	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.2912(0.1178)	0.3662(0.0874)	0.2715(0.196)
Gmean	0.5357(0.1144)	0.6065(0.0732)	0.4555(0.2863)
AUC ROC	0.6452(0.0579)	0.6837(0.0442)	0.6336(0.1034)
Accuracy	0.9834(0.0055)	0.9862(0.004)	0.9849(0.0049)

Table A.260: Decision Tree - Testing

DT CS	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.2948(0.053)	0.3056(0.0432)	0.2896(0.0545)
Gmean	0.4684(0.0658)	0.489(0.0616)	0.4534(0.0676)
AUC ROC	0.6561(0.0294)	0.6614(0.0241)	0.6508(0.0306)
Accuracy	0.9839(0.0009)	0.9839(0.0013)	0.9847(0.0023)

Table A.261: Cost Sensitive Decision Tree - Training

DT CS	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.3774(0.1176)	0.2991(0.2066)	0.2281(0.0994)
Gmean	0.6194(0.1263)	0.4813(0.2883)	0.4623(0.1103)
AUC ROC	0.6961(0.0797)	0.6462(0.1059)	0.609(0.0524)
Accuracy	0.9862(0.0027)	0.9854(0.0051)	0.9852(0.0038)

Table A.262: Cost Sensitive Decision Tree - Testing

DT HELLINGER	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.297(0.0804)	0.3055(0.0771)	0.271(0.0876)
Gmean	0.4637(0.1014)	0.4704(0.1176)	0.4317(0.1141)
AUC ROC	0.6509(0.0431)	0.6575(0.0426)	0.6369(0.0457)
Accuracy	0.985(0.0019)	0.985(0.001)	0.985(0.0015)

Table A.263: Decision Tree HELLINGER Distance - Training

DT HELLINGER	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.312(0.2031)	0.3144(0.2058)	0.3142(0.2053)
Gmean	0.5032(0.294)	0.5033(0.2941)	0.5033(0.2941)
AUC ROC	0.6586(0.1045)	0.6586(0.1048)	0.6588(0.1044)
Accuracy	0.9854(0.0041)	0.9854(0.0045)	0.9859(0.0042)

Table A.264: Decision Tree HELLINGER Distance - Testing

DT SMOTENC	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.3292(0.0573)	0.3458(0.053)	0.3278(0.0726)
Gmean	0.4996(0.0601)	0.5293(0.0544)	0.5082(0.0838)
AUC ROC	0.6742(0.03)	0.6825(0.0261)	0.6772(0.0423)
Accuracy	0.9842(0.0027)	0.9844(0.0019)	0.9831(0.0028)

Table A.265: Decision Tree SMOTENC - Training

DT SMOTENC	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.4379(0.2175)	0.3472(0.2681)	0.4125(0.2115)
Gmean	0.6871(0.1661)	0.57(0.242)	0.6505(0.168)
AUC ROC	0.7435(0.1186)	0.6822(0.1482)	0.7197(0.115)
Accuracy	0.9822(0.0082)	0.9832(0.0069)	0.9839(0.0066)

Table A.266: Decision Tree SMOTENC - Testing

DT SMOTETOMEK	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.5544(0.0505)	0.5251(0.0998)	0.5565(0.0519)
Gmean	0.7945(0.0461)	0.8203(0.029)	0.7944(0.0336)
AUC ROC	0.8388(0.0252)	0.8502(0.028)	0.8282(0.025)
Accuracy	0.9818(0.0026)	0.9713(0.0161)	0.9831(0.0029)

Table A.267: Decision Tree SMOTETOMEK - Training

DT SMOTETOMEK	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.5247(0.1888)	0.3833(0.0645)	0.5369(0.1641)
Gmean	0.7786(0.1107)	0.6818(0.0418)	0.7797(0.1094)
AUC ROC	0.8046(0.0917)	0.7284(0.0278)	0.8062(0.0899)
Accuracy	0.9807(0.0132)	0.9767(0.0087)	0.9837(0.0067)

Table A.268: Decision Tree SMOTETOMEK - Testing

KNN	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.0064(0.0104)	0.0093(0.0154)	0.0127(0.0098)	0.2931(0.081)	0.2856(0.0688)	0.2882(0.0756)
Gmean	0.011(0.0169)	0.0154(0.025)	0.0225(0.0199)	0.4372(0.0925)	0.4264(0.0718)	0.4316(0.082)
AUC ROC	0.5017(0.0034)	0.5031(0.0058)	0.5036(0.0046)	0.6451(0.04)	0.6434(0.037)	0.6418(0.034)
Accuracy	0.9873(0.0033)	0.9873(0.0033)	0.9846(0.0029)	0.9866(0.0025)	0.9865(0.0023)	0.9871(0.003)

Table A.269: KNN - Training - Part 1

KNN	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.3614(0.1006)	0.3463(0.0983)	0.3257(0.1115)	0.3027(0.0645)	0.295(0.0682)	0.2941(0.0764)
Gmean	0.5473(0.121)	0.5166(0.1065)	0.4808(0.1311)	0.4612(0.0695)	0.4505(0.1026)	0.4574(0.1038)
AUC ROC	0.6849(0.054)	0.6795(0.0538)	0.6687(0.061)	0.6541(0.033)	0.6496(0.0355)	0.6511(0.0409)
Accuracy	0.9864(0.0011)	0.9861(0.0018)	0.9858(0.0017)	0.9854(0.0012)	0.9857(0.0011)	0.9853(0.0012)

Table A.270: KNN - Training - Part 2

KNN	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.197(0.0688)	0.2065(0.0655)	0.1796(0.0553)
Gmean	0.2977(0.0958)	0.3167(0.0905)	0.2735(0.0734)
AUC ROC	0.5931(0.033)	0.5976(0.0319)	0.5819(0.0256)
Accuracy	0.9877(0.0022)	0.9878(0.0019)	0.9889(0.0021)

Table A.271: KNN - Training - Part 3

KNN	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.0(0.0)	0.0256(0.0573)	0.0256(0.0573)	0.2431(0.1252)	0.2431(0.1252)	0.2945(0.1562)
Gmean	0.0(0.0)	0.0704(0.1575)	0.0704(0.1575)	0.4841(0.1291)	0.4841(0.1291)	0.5258(0.1629)
AUC ROC	0.4986(0.0022)	0.5107(0.0269)	0.5098(0.0274)	0.6204(0.0626)	0.6204(0.0626)	0.6463(0.0836)
Accuracy	0.9872(0.0044)	0.9867(0.0047)	0.9849(0.0037)	0.9834(0.0062)	0.9834(0.0062)	0.9857(0.004)

Table A.272: KNN - Testing - Part 1

KNN	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.378(0.1658)	0.378(0.1658)	0.378(0.1658)	0.3215(0.2283)	0.3215(0.2283)	0.3215(0.2283)
Gmean	0.5966(0.1454)	0.5966(0.1454)	0.5966(0.1454)	0.539(0.2012)	0.539(0.2012)	0.539(0.2012)
AUC ROC	0.684(0.0789)	0.684(0.0789)	0.684(0.0789)	0.6584(0.1152)	0.6584(0.1152)	0.6584(0.1152)
Accuracy	0.9867(0.0065)	0.9867(0.0065)	0.9867(0.0065)	0.9852(0.0073)	0.9852(0.0073)	0.9852(0.0073)

Table A.273: KNN - Testing - Part 2

KNN	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0(0.0)	0.0(0.0)	0.0(0.0)
Gmean	0.0(0.0)	0.0(0.0)	0.0(0.0)
AUC ROC	0.4997(0.0003)	0.4997(0.0003)	0.4999(0.0003)
Accuracy	0.9895(0.0007)	0.9895(0.0007)	0.9897(0.0006)

Table A.274: KNN - Testing - Part 3

LR	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.189(0.0431)	0.1887(0.0327)	0.1787(0.0425)	0.0628(0.0305)	0.049(0.0326)	0.0558(0.0356)
Gmean	0.2813(0.0672)	0.2895(0.0407)	0.2767(0.0593)	0.0957(0.0497)	0.074(0.0453)	0.0873(0.0551)
AUC ROC	0.5873(0.0204)	0.5867(0.0165)	0.5828(0.0205)	0.5276(0.0134)	0.5217(0.0147)	0.5247(0.0159)
Accuracy	0.9891(0.0008)	0.9891(0.0005)	0.9891(0.0007)	0.9901(0.0002)	0.99(0.0006)	0.9898(0.0007)

Table A.275: Logistic Regression - Training - Part 1

LR	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.1552(0.062)	0.1577(0.0589)	0.166(0.0536)	0.1502(0.0327)	0.1457(0.046)	0.1595(0.0477)
Gmean	0.2347(0.0803)	0.2375(0.0726)	0.2522(0.0715)	0.2289(0.0537)	0.2254(0.0709)	0.2403(0.0591)
AUC ROC	0.5718(0.0291)	0.5727(0.028)	0.5751(0.0264)	0.5684(0.0158)	0.5661(0.0215)	0.5715(0.022)
Accuracy	0.9894(0.0006)	0.9891(0.0009)	0.9894(0.0006)	0.9893(0.0006)	0.9891(0.0006)	0.9895(0.0004)

Table A.276: Logistic Regression - Training - Part 2

LR	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0968(0.046)	0.0819(0.0431)	0.0835(0.0327)
Gmean	0.1514(0.068)	0.1258(0.0669)	0.128(0.0473)
AUC ROC	0.5431(0.0208)	0.5362(0.0193)	0.5372(0.0151)
Accuracy	0.9897(0.0007)	0.9898(0.0007)	0.9896(0.0006)

Table A.277: Logistic Regression - Training - Part 3

LR	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.2496(0.1151)	0.2496(0.1151)	0.2496(0.1151)	0.0303(0.0678)	0.0294(0.0658)	0.0588(0.1315)
Gmean	0.4633(0.1108)	0.4633(0.1108)	0.4633(0.1108)	0.0707(0.1581)	0.0707(0.158)	0.1(0.2236)
AUC ROC	0.611(0.053)	0.611(0.053)	0.611(0.053)	0.5121(0.0282)	0.512(0.0279)	0.5246(0.0561)
Accuracy	0.9892(0.0033)	0.9892(0.0033)	0.9892(0.0033)	0.9895(0.0014)	0.9892(0.0011)	0.9897(0.0019)

Table A.278: Logistic Regression - Testing - Part 1

LR	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.1944(0.1295)	0.2162(0.1553)	0.2162(0.1553)	0.1987(0.187)	0.1693(0.1797)	0.1693(0.1797)
Gmean	0.3701(0.2166)	0.3924(0.2379)	0.3924(0.2379)	0.3223(0.2977)	0.293(0.2827)	0.293(0.2827)
AUC ROC	0.5859(0.0563)	0.5981(0.0714)	0.598(0.0716)	0.5865(0.0845)	0.5737(0.0821)	0.5737(0.0821)
Accuracy	0.9885(0.0044)	0.9882(0.0042)	0.988(0.0042)	0.9897(0.0042)	0.989(0.0039)	0.989(0.0039)

Table A.279: Logistic Regression - Testing - Part 2

LR	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.2792(0.3825)	0.2857(0.3912)	0.2792(0.3825)
Gmean	0.3453(0.4728)	0.3455(0.4731)	0.3453(0.4728)
AUC ROC	0.6487(0.2037)	0.649(0.204)	0.6487(0.2037)
Accuracy	0.9905(0.0011)	0.991(0.0014)	0.9905(0.0011)

Table A.280: Logistic Regression - Testing - Part 3

LR CS	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.5445(0.023)	0.5278(0.0286)	0.5433(0.0239)	0.592(0.0366)	0.4744(0.0534)	0.5941(0.0406)
Gmean	0.8362(0.0501)	0.9032(0.0148)	0.8054(0.0256)	0.8735(0.0318)	0.9298(0.0112)	0.8741(0.0334)
AUC ROC	0.8601(0.0378)	0.9138(0.0146)	0.8393(0.0174)	0.8936(0.0221)	0.9347(0.0096)	0.8935(0.0243)
Accuracy	0.9766(0.0079)	0.9655(0.0031)	0.9802(0.0011)	0.9784(0.0018)	0.9481(0.0178)	0.9779(0.002)

Table A.281: Cost Sensitive Logistic Regression - Training - Part 1

LR CS	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.5872(0.0515)	0.538(0.0338)	0.5864(0.0419)	0.5476(0.0415)	0.5296(0.0316)	0.5295(0.0365)
Gmean	0.8529(0.0307)	0.9138(0.0217)	0.8466(0.0383)	0.8874(0.0323)	0.918(0.0173)	0.8048(0.0301)
AUC ROC	0.8768(0.0238)	0.9215(0.0182)	0.8747(0.0213)	0.901(0.0244)	0.9242(0.0159)	0.8397(0.0179)
Accuracy	0.9799(0.0027)	0.9654(0.0034)	0.98(0.0019)	0.9701(0.01)	0.9637(0.0031)	0.9788(0.0014)

Table A.282: Cost Sensitive Logistic Regression - Training - Part 2

LR CS	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.5757(0.0473)	0.4959(0.0372)	0.5881(0.0509)
Gmean	0.8594(0.0342)	0.9402(0.011)	0.8622(0.0371)
AUC ROC	0.8767(0.0274)	0.9436(0.0095)	0.882(0.0296)
Accuracy	0.9785(0.0023)	0.9509(0.012)	0.9787(0.0027)

Table A.283: Cost Sensitive Logistic Regression - Training - Part 3

LR CS	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.526(0.0817)	0.5162(0.1039)	0.5202(0.0873)	0.592(0.113)	0.4803(0.0664)	0.592(0.113)
Gmean	0.8272(0.0414)	0.8916(0.0619)	0.7974(0.0655)	0.8834(0.0655)	0.9489(0.0353)	0.8834(0.0655)
AUC ROC	0.8397(0.0349)	0.8959(0.0588)	0.8168(0.0522)	0.8897(0.0578)	0.9494(0.0351)	0.8897(0.0578)
Accuracy	0.9767(0.0079)	0.9654(0.01)	0.9802(0.0063)	0.9777(0.0074)	0.9488(0.012)	0.9777(0.0074)

Table A.284: Cost Sensitive Logistic Regression - Testing - Part 1

LR CS	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.5907(0.1066)	0.5332(0.1206)	0.594(0.1028)	0.5477(0.059)	0.5542(0.0737)	0.5293(0.0759)
Gmean	0.8705(0.0609)	0.9031(0.089)	0.8707(0.0608)	0.893(0.0767)	0.9441(0.0364)	0.8135(0.042)
AUC ROC	0.8781(0.0537)	0.9085(0.0815)	0.8784(0.0536)	0.8989(0.069)	0.9448(0.0359)	0.8286(0.0355)
Accuracy	0.9792(0.0075)	0.9659(0.0138)	0.9797(0.0065)	0.9714(0.0051)	0.9641(0.0092)	0.9792(0.0062)

Table A.285: Cost Sensitive Logistic Regression - Testing - Part 2

LR CS	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.4907(0.0803)	0.3849(0.1851)	0.5328(0.0616)
Gmean	0.8793(0.1525)	0.8843(0.1242)	0.929(0.0474)
AUC ROC	0.8948(0.1202)	0.8963(0.0999)	0.9312(0.0446)
Accuracy	0.9631(0.0187)	0.8437(0.2225)	0.9619(0.0163)

Table A.286: Cost Sensitive Logistic Regression - Testing - Part 3

LR SMOTENC	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.5577(0.0302)	0.5695(0.0286)	0.5544(0.0376)	0.5727(0.0166)	0.576(0.0269)	0.5777(0.014)
Gmean	0.8049(0.0303)	0.8142(0.0251)	0.799(0.0329)	0.8645(0.0242)	0.8686(0.0247)	0.8538(0.0224)
AUC ROC	0.8394(0.0173)	0.8437(0.0144)	0.8348(0.0221)	0.8841(0.014)	0.8851(0.0151)	0.8781(0.0136)
Accuracy	0.982(0.0014)	0.9824(0.0014)	0.9822(0.001)	0.9769(0.0013)	0.9773(0.0013)	0.9787(0.0022)

Table A.287: Logistic Regression SMOTENC - Training - Part 1

LR SMOTENC	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.5866(0.0529)	0.5986(0.0433)	0.5774(0.0576)	0.5492(0.044)	0.5391(0.0339)	0.5468(0.0306)
Gmean	0.8332(0.0461)	0.8437(0.0356)	0.8305(0.0402)	0.8109(0.0388)	0.7977(0.0192)	0.808(0.0212)
AUC ROC	0.8625(0.0292)	0.873(0.0211)	0.8543(0.0301)	0.8454(0.0219)	0.8422(0.0151)	0.8435(0.0105)
Accuracy	0.9819(0.002)	0.9821(0.0017)	0.9821(0.0025)	0.9803(0.0027)	0.9793(0.0023)	0.9798(0.0024)

Table A.288: Logistic Regression SMOTENC - Training - Part 2

LR SMOTENC	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.5771(0.0265)	0.575(0.0221)	0.5849(0.022)
Gmean	0.866(0.021)	0.8671(0.0144)	0.8698(0.0182)
AUC ROC	0.8861(0.0099)	0.8842(0.0087)	0.8879(0.0155)
Accuracy	0.9769(0.0019)	0.9771(0.0019)	0.9778(0.0021)

Table A.289: Logistic Regression SMOTENC - Training - Part 3

LR SMOTENC	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.5668(0.1732)	0.5703(0.1786)	0.5726(0.1781)	0.5646(0.1017)	0.5646(0.1017)	0.5591(0.0868)
Gmean	0.8252(0.1018)	0.8253(0.102)	0.8254(0.1019)	0.8691(0.0609)	0.8691(0.0609)	0.856(0.0516)
AUC ROC	0.8419(0.0863)	0.842(0.0865)	0.8421(0.0864)	0.8766(0.0537)	0.8766(0.0537)	0.8649(0.0452)
Accuracy	0.9809(0.0097)	0.9812(0.01)	0.9814(0.0101)	0.9762(0.0073)	0.9762(0.0073)	0.9774(0.0062)

Table A.290: Logistic Regression SMOTENC - Testing - Part 1

LR SMOTENC	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.616(0.1216)	0.5876(0.1417)	0.5788(0.1447)	0.5645(0.0953)	0.5598(0.0964)	0.5598(0.0964)
Gmean	0.8709(0.0762)	0.8527(0.1145)	0.8393(0.1078)	0.8421(0.0606)	0.8419(0.0608)	0.8419(0.0608)
AUC ROC	0.8796(0.0696)	0.8668(0.0986)	0.8546(0.0932)	0.8536(0.0524)	0.8534(0.0527)	0.8534(0.0527)
Accuracy	0.9822(0.0056)	0.9812(0.0058)	0.9817(0.0057)	0.9797(0.0076)	0.9792(0.0071)	0.9792(0.0071)

Table A.291: Logistic Regression SMOTENC - Testing - Part 2

LR SMOTENC	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.5493(0.0819)	0.5559(0.0838)	0.5774(0.0867)
Gmean	0.8813(0.0593)	0.8816(0.0591)	0.8691(0.0747)
AUC ROC	0.8872(0.0561)	0.8876(0.0559)	0.8777(0.0681)
Accuracy	0.9726(0.0075)	0.9734(0.0083)	0.9784(0.0035)

Table A.292: Logistic Regression SMOTENC - Testing - Part 3

LR SMOTETOMEK	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.5164(0.0422)	0.4969(0.0335)	0.5144(0.0224)	0.5522(0.0359)	0.5424(0.032)	0.55(0.0367)
Gmean	0.8446(0.0334)	0.8349(0.0215)	0.8447(0.0235)	0.9161(0.0148)	0.9076(0.0146)	0.9011(0.0157)
AUC ROC	0.8694(0.021)	0.856(0.0181)	0.8712(0.0149)	0.9221(0.0144)	0.9171(0.0151)	0.9131(0.01)
Accuracy	0.9711(0.0029)	0.9702(0.0027)	0.9709(0.0018)	0.9677(0.0033)	0.9671(0.0033)	0.9689(0.0051)

Table A.293: Logistic Regression SMOTETOMEK - Training - Part 1

LR SMOTETOMEK	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.5671(0.0572)	0.5519(0.0483)	0.5611(0.0561)	0.5226(0.0393)	0.5234(0.0496)	0.5267(0.0407)
Gmean	0.8821(0.0354)	0.8814(0.0213)	0.8793(0.0345)	0.8608(0.0274)	0.8654(0.0316)	0.8543(0.0304)
AUC ROC	0.8972(0.027)	0.895(0.0188)	0.8925(0.0289)	0.8793(0.0213)	0.8816(0.0265)	0.8793(0.0229)
Accuracy	0.9739(0.0041)	0.9718(0.0044)	0.9738(0.0034)	0.9707(0.0037)	0.9703(0.0033)	0.9712(0.0027)

Table A.294: Logistic Regression SMOTETOMEK - Training - Part 2

LR SMOTETOMEK	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.5646(0.0339)	0.5516(0.0382)	0.5557(0.0357)
Gmean	0.9112(0.0224)	0.9051(0.0228)	0.9097(0.0154)
AUC ROC	0.9188(0.0197)	0.913(0.0192)	0.9181(0.0142)
Accuracy	0.97(0.0035)	0.9693(0.003)	0.9691(0.0032)

Table A.295: Logistic Regression SMOTETOMEK - Training - Part 3

LR SMOTETOMEK	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.4991(0.0888)	0.515(0.1048)	0.4972(0.0891)	0.5338(0.073)	0.5396(0.0743)	0.5229(0.0879)
Gmean	0.8383(0.0601)	0.852(0.072)	0.8382(0.0602)	0.9056(0.0545)	0.9187(0.047)	0.8911(0.0777)
AUC ROC	0.8492(0.0522)	0.8617(0.063)	0.8491(0.0523)	0.9089(0.0513)	0.9208(0.0447)	0.8968(0.0703)
Accuracy	0.9709(0.0104)	0.9711(0.0106)	0.9706(0.0102)	0.9666(0.0099)	0.9656(0.0091)	0.9671(0.0089)

Table A.296: Logistic Regression SMOTETOMEK - Testing - Part 1

LR SMOTETOMEK	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.5513(0.1239)	0.5601(0.1127)	0.5576(0.1305)	0.5139(0.0928)	0.5148(0.0953)	0.5189(0.0974)
Gmean	0.8801(0.08)	0.8933(0.0788)	0.8805(0.0804)	0.8531(0.0517)	0.8531(0.0518)	0.8534(0.0518)
AUC ROC	0.8871(0.0732)	0.8992(0.0713)	0.8875(0.0735)	0.8616(0.0456)	0.8616(0.0456)	0.8618(0.0456)
Accuracy	0.9724(0.0095)	0.9719(0.0105)	0.9731(0.0095)	0.9709(0.0102)	0.9709(0.0109)	0.9714(0.0113)

Table A.297: Logistic Regression SMOTETOMEK - Testing - Part 2

LR SMOTETOMEK	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.4771(0.0623)	0.4642(0.051)	0.4718(0.0625)
Gmean	0.9364(0.0249)	0.934(0.0488)	0.9472(0.0254)
AUC ROC	0.9373(0.0247)	0.936(0.046)	0.948(0.0251)
Accuracy	0.9493(0.0189)	0.9468(0.0198)	0.946(0.0186)

Table A.298: Logistic Regression SMOTETOMEK - Testing - Part 3

RF	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0473(0.044)	0.0432(0.0385)	0.0324(0.0287)
Gmean	0.0724(0.0657)	0.0655(0.0595)	0.0473(0.0401)
AUC ROC	0.5204(0.0191)	0.5188(0.0171)	0.5138(0.0127)
Accuracy	0.9895(0.0007)	0.9896(0.0004)	0.9896(0.0002)

Table A.299: Random Forest - Training

RF	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0606(0.083)	0.0303(0.0678)	0.0606(0.083)
Gmean	0.1414(0.1936)	0.0707(0.1581)	0.1414(0.1936)
AUC ROC	0.5247(0.0345)	0.5122(0.0281)	0.5247(0.0345)
Accuracy	0.99(0.0013)	0.9897(0.001)	0.99(0.0013)

Table A.300: Random Forest - Testing

RF CS	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.5563(0.0365)	0.5393(0.0371)	0.5419(0.0507)
Gmean	0.8102(0.0266)	0.8024(0.0206)	0.7848(0.0425)
AUC ROC	0.8396(0.0167)	0.834(0.0147)	0.8329(0.0266)
Accuracy	0.9814(0.0023)	0.9806(0.0026)	0.9812(0.0024)

Table A.301: Cost Sensitive Random Forest - Training

RF CS	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.5011(0.1172)	0.56(0.1411)	0.5447(0.0783)
Gmean	0.7666(0.0682)	0.8263(0.0881)	0.8273(0.0643)
AUC ROC	0.7925(0.0545)	0.8418(0.0734)	0.8413(0.0557)
Accuracy	0.9812(0.0077)	0.9807(0.007)	0.9797(0.0039)

Table A.302: Cost Sensitive Random Forest - Testing

SVM	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.087(0.082)	0.0733(0.0809)	0.0816(0.0751)	0.0508(0.059)	0.0475(0.0477)	0.05(0.0586)
Gmean	0.135(0.1224)	0.1133(0.1185)	0.1222(0.108)	0.082(0.0961)	0.0754(0.0724)	0.0787(0.0936)
AUC ROC	0.5402(0.0389)	0.5332(0.037)	0.5374(0.0354)	0.5225(0.0267)	0.5212(0.0216)	0.5217(0.0262)
Accuracy	0.9889(0.0009)	0.9889(0.001)	0.9889(0.001)	0.9893(0.0008)	0.9894(0.0005)	0.9893(0.0006)

Table A.303: SVM - Training - Part 1

SVM	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.1412(0.1114)	0.1366(0.1122)	0.1389(0.0932)	0.1807(0.0729)	0.1757(0.067)	0.172(0.0881)
Gmean	0.2175(0.1571)	0.2121(0.1669)	0.2155(0.1332)	0.2665(0.0984)	0.2651(0.0904)	0.2636(0.1276)
AUC ROC	0.5676(0.0537)	0.5655(0.0558)	0.5667(0.0478)	0.5838(0.0354)	0.5825(0.0327)	0.5797(0.0422)
Accuracy	0.9882(0.0011)	0.9883(0.0012)	0.9886(0.0016)	0.9893(0.0004)	0.9889(0.0009)	0.9889(0.0005)

Table A.304: SVM - Training - Part 2

SVM	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0(0.0)	0.0(0.0)	0.0(0.0)
Gmean	0.0(0.0)	0.0(0.0)	0.0(0.0)
AUC ROC	0.5(0.0)	0.5(0.0)	0.5(0.0001)
Accuracy	0.9899(0.0)	0.9899(0.0)	0.9899(0.0001)

Table A.305: SVM - Training - Part 3

SVM	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.0303(0.0678)	0.0303(0.0678)	0.0303(0.0678)	0.0(0.0)	0.0(0.0)	0.0303(0.0678)
Gmean	0.0707(0.1581)	0.0707(0.1581)	0.0707(0.1581)	0.0(0.0)	0.0(0.0)	0.0707(0.1581)
AUC ROC	0.5119(0.0283)	0.5119(0.0283)	0.5119(0.0283)	0.4996(0.0009)	0.4996(0.0009)	0.5121(0.0282)
Accuracy	0.989(0.003)	0.989(0.003)	0.989(0.003)	0.9892(0.0017)	0.9892(0.0017)	0.9895(0.0019)

Table A.306: SVM - Testing - Part 1

SVM	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.1073(0.1112)	0.1073(0.1112)	0.1073(0.1112)	0.2766(0.2127)	0.1955(0.1531)	0.2195(0.2442)
Gmean	0.2408(0.2278)	0.2408(0.2278)	0.2408(0.2278)	0.4343(0.2762)	0.3635(0.2304)	0.3344(0.3314)
AUC ROC	0.5485(0.0511)	0.5485(0.0511)	0.5484(0.0513)	0.6239(0.0994)	0.5862(0.0715)	0.599(0.1138)
Accuracy	0.988(0.004)	0.988(0.004)	0.9877(0.0039)	0.9902(0.0035)	0.9892(0.0026)	0.99(0.0033)

Table A.307: SVM - Testing - Part 2

SVM	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0(0.0)	0.0(0.0)	0.0(0.0)
Gmean	0.0(0.0)	0.0(0.0)	0.0(0.0)
AUC ROC	0.5(0.0)	0.5(0.0)	0.5(0.0)
Accuracy	0.99(0.0)	0.99(0.0)	0.99(0.0)

Table A.308: SVM - Testing - Part 3

SVM CS	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.5148(0.0249)	0.5045(0.0204)	0.5081(0.0269)	0.5937(0.0416)	0.5197(0.0644)	0.5695(0.0443)
Gmean	0.8025(0.013)	0.8348(0.0261)	0.7861(0.0165)	0.8733(0.0291)	0.8811(0.0163)	0.8345(0.0366)
AUC ROC	0.8363(0.0069)	0.8604(0.0177)	0.8305(0.009)	0.8866(0.0235)	0.8958(0.0111)	0.8631(0.0238)
Accuracy	0.9769(0.0032)	0.9709(0.0022)	0.9769(0.0034)	0.979(0.0018)	0.9662(0.0083)	0.9798(0.0022)

Table A.309: Cost Sensitive SVM - Training - Part 1

SVM CS	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.5335(0.0568)	0.5271(0.041)	0.5411(0.0414)	0.5271(0.0317)	0.4995(0.0288)	0.5051(0.0562)
Gmean	0.7825(0.0454)	0.8242(0.0164)	0.7919(0.0566)	0.8181(0.0056)	0.8104(0.0255)	0.7172(0.0551)
AUC ROC	0.826(0.032)	0.8544(0.0152)	0.827(0.0377)	0.8509(0.0044)	0.8498(0.0183)	0.784(0.0336)
Accuracy	0.9809(0.0077)	0.9757(0.0042)	0.9819(0.0076)	0.976(0.0044)	0.9726(0.0035)	0.985(0.0015)

Table A.310: Cost Sensitive SVM - Training - Part 2

SVM CS	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.5788(0.0516)	0.5503(0.0725)	0.5731(0.0397)
Gmean	0.8559(0.0363)	0.8759(0.0333)	0.8486(0.024)
AUC ROC	0.8771(0.0293)	0.8897(0.0284)	0.8715(0.0184)
Accuracy	0.9788(0.0025)	0.9722(0.0087)	0.979(0.0028)

Table A.311: Cost Sensitive SVM - Training - Part 3

SVM CS	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.4934(0.1094)	0.4901(0.0942)	0.4934(0.1094)	0.5744(0.1011)	0.5369(0.0636)	0.5722(0.1067)
Gmean	0.7933(0.0986)	0.8379(0.0619)	0.7933(0.0986)	0.8567(0.052)	0.9061(0.0554)	0.8565(0.0521)
AUC ROC	0.8152(0.0809)	0.8486(0.0543)	0.8152(0.0809)	0.8656(0.0457)	0.9094(0.0522)	0.8654(0.0457)
Accuracy	0.9772(0.0059)	0.9696(0.0082)	0.9772(0.0059)	0.9789(0.0076)	0.9676(0.0044)	0.9784(0.0092)

Table A.312: Cost Sensitive SVM - Testing - Part 1

SVM CS	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.4687(0.1461)	0.5563(0.099)	0.4696(0.11)	0.4573(0.064)	0.494(0.054)	0.4948(0.0862)
Gmean	0.7424(0.1356)	0.8547(0.0717)	0.7447(0.1185)	0.7645(0.0654)	0.8256(0.041)	0.7502(0.0786)
AUC ROC	0.7798(0.1038)	0.8647(0.0624)	0.78(0.0928)	0.7901(0.0511)	0.8378(0.0345)	0.7813(0.055)
Accuracy	0.9804(0.0059)	0.9772(0.0069)	0.9809(0.0052)	0.9764(0.0049)	0.9729(0.004)	0.9834(0.0046)

Table A.313: Cost Sensitive SVM - Testing - Part 2

SVM CS	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.3087(0.2941)	0.1298(0.1757)	0.1069(0.213)
Gmean	0.573(0.5237)	0.3882(0.4909)	0.1831(0.4095)
AUC ROC	0.7733(0.2508)	0.6717(0.2346)	0.5833(0.1863)
Accuracy	0.9676(0.0258)	0.742(0.413)	0.7875(0.4348)

Table A.314: Cost Sensitive SVM - Testing - Part 3

SVM SMOTENC	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.106(0.0273)	0.1073(0.0284)	0.1084(0.0195)	0.5089(0.038)	0.5082(0.0367)	0.5043(0.0299)
Gmean	0.3732(0.0967)	0.383(0.1072)	0.3806(0.0803)	0.7686(0.0379)	0.7663(0.03)	0.7646(0.0267)
AUC ROC	0.5918(0.0349)	0.5928(0.0361)	0.5901(0.0281)	0.8179(0.0178)	0.8155(0.0209)	0.8148(0.0152)
Accuracy	0.9157(0.0076)	0.9145(0.009)	0.9151(0.009)	0.9797(0.0017)	0.9797(0.0016)	0.9796(0.0014)

Table A.315: SVM SMOTENC - Training - Part 1

SVM SMOTENC	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.5194(0.0259)	0.5111(0.04)	0.5148(0.0459)	0.4961(0.0365)	0.5074(0.0391)	0.4791(0.0351)
Gmean	0.7354(0.0288)	0.7461(0.0217)	0.7386(0.0407)	0.7282(0.0337)	0.7296(0.0408)	0.7006(0.0311)
AUC ROC	0.797(0.0173)	0.7925(0.0213)	0.7971(0.0245)	0.7856(0.023)	0.7923(0.0228)	0.7751(0.0213)
Accuracy	0.9847(0.0013)	0.9844(0.0018)	0.9843(0.0017)	0.9838(0.001)	0.984(0.0011)	0.9837(0.0011)

Table A.316: SVM SMOTENC - Training - Part 2

SVM SMOTENC	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.5432(0.0255)	0.5485(0.0214)	0.5352(0.0221)
Gmean	0.8123(0.0255)	0.8221(0.0274)	0.806(0.0225)
AUC ROC	0.8411(0.0179)	0.8465(0.0221)	0.8396(0.0181)
Accuracy	0.9794(0.0023)	0.9797(0.002)	0.9791(0.0021)

Table A.317: SVM SMOTENC - Training - Part 3

SVM SMOTENC	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.1139(0.0444)	0.1139(0.0444)	0.1139(0.0444)	0.5295(0.1252)	0.5295(0.1252)	0.5295(0.1252)
Gmean	0.4947(0.103)	0.4947(0.103)	0.4947(0.103)	0.8107(0.0861)	0.8107(0.0861)	0.8107(0.0861)
AUC ROC	0.5973(0.0543)	0.5973(0.0543)	0.5973(0.0543)	0.8286(0.0718)	0.8286(0.0718)	0.8286(0.0718)
Accuracy	0.9132(0.0104)	0.9132(0.0104)	0.9132(0.0104)	0.9792(0.0075)	0.9792(0.0075)	0.9792(0.0075)

Table A.318: SVM SMOTENC - Testing - Part 1

SVM SMOTENC	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.4998(0.135)	0.4998(0.135)	0.4998(0.135)	0.5287(0.0921)	0.5287(0.0921)	0.5287(0.0921)
Gmean	0.7487(0.0971)	0.7487(0.0971)	0.7487(0.0971)	0.7837(0.0566)	0.7837(0.0566)	0.7837(0.0566)
AUC ROC	0.7813(0.0725)	0.7813(0.0725)	0.7813(0.0725)	0.8058(0.0452)	0.8058(0.0452)	0.8058(0.0452)
Accuracy	0.9834(0.006)	0.9834(0.006)	0.9834(0.006)	0.9829(0.0054)	0.9829(0.0054)	0.9829(0.0054)

Table A.319: SVM SMOTENC - Testing - Part 2

SVM SMOTENC	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0(0.0)	0.0(0.0)	0.0(0.0)
Gmean	0.0(0.0)	0.0(0.0)	0.0(0.0)
AUC ROC	0.5(0.0)	0.5(0.0)	0.5(0.0)
Accuracy	0.99(0.0)	0.99(0.0)	0.99(0.0)

Table A.320: SVM SMOTENC - Testing - Part 3

SVM SMOTETOMEK	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.1105(0.0233)	0.1066(0.0169)	0.1056(0.0221)
Gmean	0.3971(0.1022)	0.4032(0.082)	0.3948(0.1021)
AUC ROC	0.5958(0.0334)	0.5923(0.0277)	0.59(0.0329)
Accuracy	0.9118(0.0074)	0.9114(0.0095)	0.9123(0.0096)

Table A.321: SVM SMOTETOMEK - Training

SVM SMOTETOMEK	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.1199(0.0606)	0.1199(0.0606)	0.1199(0.0606)
Gmean	0.5116(0.1286)	0.5116(0.1286)	0.5116(0.1286)
AUC ROC	0.6074(0.0736)	0.6074(0.0736)	0.6074(0.0736)
Accuracy	0.9087(0.0126)	0.9087(0.0126)	0.9087(0.0126)

Table A.322: SVM SMOTETOMEK - Testing

XGBoost	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.1459(0.0752)	0.1439(0.102)	0.1333(0.0953)
Gmean	0.2244(0.1142)	0.2196(0.1524)	0.1986(0.141)
AUC ROC	0.5674(0.0357)	0.567(0.0499)	0.5618(0.0457)
Accuracy	0.9878(0.0008)	0.9877(0.0007)	0.9876(0.0007)

Table A.323: XGBoost - Training

XGBoost	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.2535(0.1157)	0.2535(0.1157)	0.2535(0.1157)
Gmean	0.4634(0.1108)	0.4634(0.1108)	0.4634(0.1108)
AUC ROC	0.6114(0.0529)	0.6114(0.0529)	0.6114(0.0529)
Accuracy	0.99(0.0029)	0.99(0.0029)	0.99(0.0029)

Table A.324: XGBoost - Testing

XGBoost CS	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.1465(0.0801)	0.1653(0.0841)	0.1562(0.077)
Gmean	0.2266(0.1176)	0.2535(0.1171)	0.2287(0.1071)
AUC ROC	0.5668(0.0378)	0.5759(0.039)	0.5724(0.0373)
Accuracy	0.9889(0.001)	0.9889(0.0013)	0.9889(0.0008)

Table A.325: Cost Sensitive XGBoost - Training

XGBoost CS	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.1423(0.1451)	0.1423(0.1451)	0.1423(0.1451)
Gmean	0.2704(0.2541)	0.2704(0.2541)	0.2704(0.2541)
AUC ROC	0.5612(0.063)	0.5612(0.063)	0.5612(0.063)
Accuracy	0.9887(0.0031)	0.9887(0.0031)	0.9887(0.0031)

Table A.326: Cost Sensitive XGBoost - Testing

XGBoost SMOTENC	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.4366(0.0589)	0.4663(0.0772)	0.4631(0.0678)
Gmean	0.6367(0.0654)	0.6722(0.0729)	0.6633(0.0648)
AUC ROC	0.74(0.0285)	0.756(0.041)	0.7545(0.0352)
Accuracy	0.9849(0.002)	0.9851(0.0022)	0.9854(0.0024)

Table A.327: XGBoost SMOTENC - Training

XGBoost SMOTENC	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.477(0.1525)	0.477(0.1525)	0.477(0.1525)
Gmean	0.714(0.1116)	0.714(0.1116)	0.714(0.1116)
AUC ROC	0.7573(0.0827)	0.7573(0.0827)	0.7573(0.0827)
Accuracy	0.9849(0.0068)	0.9849(0.0068)	0.9849(0.0068)

Table A.328: XGBoost SMOTENC - Testing

XGBoost SMOTETOMEK	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.4(0.0666)	0.379(0.068)	0.3999(0.1028)
Gmean	0.5931(0.0619)	0.5509(0.0943)	0.5827(0.117)
AUC ROC	0.7151(0.0379)	0.7022(0.0377)	0.7121(0.056)
Accuracy	0.9853(0.0017)	0.9852(0.0016)	0.9857(0.002)

Table A.329: XGBoost SMOTETOMEK - Training

XGBoost SMOTETOMEK	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.4091(0.1196)	0.4017(0.1151)	0.4017(0.1151)
Gmean	0.6599(0.1102)	0.6438(0.0915)	0.6438(0.0915)
AUC ROC	0.7198(0.0708)	0.7079(0.0562)	0.7079(0.0562)
Accuracy	0.9842(0.0046)	0.9852(0.0058)	0.9852(0.0058)

Table A.330: XGBoost SMOTETOMEK - Testing

NN	No Scaling			NORM		
Optimized For	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.0239(0.0194)	0.0114(0.0139)	0.0193(0.0247)	0.1701(0.1174)	0.1967(0.1273)	0.1777(0.1079)
Gmean	0.0339(0.0252)	0.017(0.019)	0.0285(0.0396)	0.2609(0.1698)	0.2888(0.1748)	0.2679(0.1511)
AUC ROC	0.5113(0.0095)	0.5054(0.0068)	0.5088(0.0115)	0.5809(0.0554)	0.5963(0.065)	0.5858(0.0533)
Accuracy	0.9899(0.0003)	0.9897(0.0002)	0.9899(0.0002)	0.989(0.0006)	0.9889(0.0008)	0.9884(0.0005)

Table A.331: Neural Network - Training - Part 1

NN	POWER			STD		
Optimized For	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.2248(0.0519)	0.2297(0.0814)	0.2026(0.0702)	0.2113(0.0383)	0.2255(0.0499)	0.2152(0.0117)
Gmean	0.3551(0.0787)	0.3474(0.1127)	0.3054(0.0828)	0.3298(0.0555)	0.3433(0.0769)	0.3313(0.011)
AUC ROC	0.6103(0.0255)	0.6131(0.0417)	0.5978(0.0352)	0.603(0.0193)	0.6116(0.0214)	0.6062(0.0057)
Accuracy	0.9873(0.0009)	0.9874(0.0006)	0.9876(0.0009)	0.9875(0.0002)	0.9871(0.0009)	0.9875(0.0007)

Table A.332: Neural Network - Training - Part 2

NN	TRD		
Optimized For	**F2**	**G-mean**	**PR-Recall**
F2	0.1564(0.0616)	0.1793(0.1222)	0.1691(0.0887)
Gmean	0.2317(0.0855)	0.2722(0.1644)	0.2564(0.1239)
AUC ROC	0.5732(0.029)	0.5842(0.0592)	0.5792(0.043)
Accuracy	0.9885(0.001)	0.9885(0.0011)	0.9884(0.0006)

Table A.333: Neural Network - Training - Part 3

NN	No Scaling			NORM		
Optimized For	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.2042(0.1669)	0.1589(0.1656)	0.2153(0.0759)
Gmean	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.3852(0.251)	0.2926(0.2822)	0.4404(0.0802)
AUC ROC	0.4999(0.0003)	0.4999(0.0003)	0.5(0.0)	0.5976(0.0829)	0.5732(0.0809)	0.5976(0.0347)
Accuracy	0.9897(0.0006)	0.9897(0.0006)	0.99(0.0)	0.9872(0.004)	0.988(0.0021)	0.9872(0.0045)

Table A.334: Neural Network - Testing - Part 1

NN	POWER			STD		
Optimized For	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.2058(0.1393)	0.2057(0.0973)	0.1558(0.1747)	0.2737(0.1625)	0.2474(0.1517)	0.1896(0.1511)
Gmean	0.392(0.2374)	0.4333(0.1169)	0.2923(0.2822)	0.4658(0.2648)	0.4433(0.254)	0.3632(0.2303)
AUC ROC	0.5977(0.0699)	0.5972(0.0547)	0.5728(0.0807)	0.6345(0.0805)	0.6218(0.0753)	0.5861(0.0706)
Accuracy	0.9875(0.0032)	0.9864(0.0032)	0.9872(0.0047)	0.9867(0.0031)	0.9862(0.0035)	0.989(0.0021)

Table A.335: Neural Network - Testing - Part 2

NN	TRD		
Optimized For	**F2**	**G-mean**	**PR-Recall**
F2	0.0(0.0)	0.0(0.0)	0.0(0.0)
Gmean	0.0(0.0)	0.0(0.0)	0.0(0.0)
AUC ROC	0.5(0.0)	0.5(0.0)	0.5(0.0)
Accuracy	0.99(0.0)	0.99(0.0)	0.99(0.0)

Table A.336: Neural Network - Testing - Part 3

A.6. Classification For MUSC. Classification For MUSC

ADABOOST	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.1881(0.0268)	0.2011(0.0367)	0.1823(0.0332)
Gmean	0.3316(0.0317)	0.3647(0.0512)	0.3159(0.066)
AUC ROC	0.5963(0.0145)	0.6041(0.0186)	0.59(0.0179)
Accuracy	0.9799(0.0016)	0.9795(0.0024)	0.981(0.0008)

Table A.337: ADABOOST - Training

ADABOOST	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0654(0.0599)	0.1062(0.1046)	0.1507(0.0935)
Gmean	0.1989(0.1815)	0.2537(0.2382)	0.3476(0.2031)
AUC ROC	0.5274(0.0308)	0.5497(0.0554)	0.5721(0.0501)
Accuracy	0.9777(0.0035)	0.9784(0.0034)	0.9792(0.004)

Table A.338: ADABOOST - Testing

ADACOST	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.4608(0.0069)	0.4603(0.0074)	0.4603(0.007)
Gmean	0.9655(0.0009)	0.9655(0.0009)	0.9655(0.0009)
AUC ROC	0.9661(0.0009)	0.9661(0.0009)	0.9661(0.0009)
Accuracy	0.933(0.0017)	0.933(0.0017)	0.933(0.0017)

Table A.339: ADACOST - Training

ADACOST	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.4585(0.0256)	0.4585(0.0256)	0.4585(0.0256)
Gmean	0.9655(0.0036)	0.9655(0.0036)	0.9655(0.0036)
AUC ROC	0.9661(0.0035)	0.9661(0.0035)	0.9661(0.0035)
Accuracy	0.933(0.0069)	0.933(0.0069)	0.933(0.0069)

Table A.340: ADACOST - Testing

CATBOOST	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0681(0.028)	0.0737(0.0218)	0.0825(0.0424)
Gmean	0.1086(0.0402)	0.1189(0.0311)	0.1318(0.0644)
AUC ROC	0.5294(0.0125)	0.5318(0.0096)	0.5359(0.0187)
Accuracy	0.9881(0.0007)	0.9881(0.0006)	0.988(0.0006)

Table A.341: CATBOOST - Training

CATBOOST	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0263(0.0588)	0.0263(0.0588)	0.0263(0.0588)
Gmean	0.0666(0.149)	0.0666(0.149)	0.0666(0.149)
AUC ROC	0.5102(0.025)	0.5102(0.025)	0.5102(0.025)
Accuracy	0.9872(0.001)	0.9872(0.001)	0.9872(0.001)

Table A.342: CATBOOST - Testing

DT	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.2138(0.0437)	0.1875(0.0327)	0.1872(0.0599)
Gmean	0.358(0.0576)	0.3314(0.0558)	0.324(0.0909)
AUC ROC	0.609(0.0228)	0.5947(0.0173)	0.5961(0.0296)
Accuracy	0.9804(0.0015)	0.9798(0.001)	0.9794(0.0022)

Table A.343: Decision Tree - Training

DT	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.1341(0.0496)	0.1267(0.0844)	0.1761(0.0574)
Gmean	0.3592(0.0615)	0.3199(0.1914)	0.4141(0.0751)
AUC ROC	0.5618(0.0251)	0.5611(0.0452)	0.5841(0.0301)
Accuracy	0.9807(0.003)	0.9792(0.0038)	0.9812(0.002)

Table A.344: Decision Tree - Testing

DT CS	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.2259(0.0228)	0.246(0.0302)	0.2363(0.0371)
Gmean	0.3933(0.0316)	0.4164(0.0467)	0.3775(0.0476)
AUC ROC	0.6164(0.0119)	0.6279(0.0203)	0.619(0.0199)
Accuracy	0.9806(0.0021)	0.9805(0.0013)	0.9823(0.0024)

Table A.345: Cost Sensitive Decision Tree - Training

DT CS	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.1267(0.111)	0.2154(0.0584)	0.2598(0.1449)
Gmean	0.3136(0.2043)	0.4626(0.0853)	0.512(0.1688)
AUC ROC	0.5611(0.0603)	0.6063(0.0371)	0.6378(0.0837)
Accuracy	0.9792(0.0031)	0.9817(0.0041)	0.9789(0.0029)

Table A.346: Cost Sensitive Decision Tree - Testing

DT HELLINGER	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.1889(0.0288)	0.1891(0.0529)	0.1913(0.0274)
Gmean	0.333(0.0391)	0.3226(0.0703)	0.3284(0.0378)
AUC ROC	0.5944(0.0138)	0.5952(0.0277)	0.596(0.0146)
Accuracy	0.981(0.0019)	0.9816(0.002)	0.9814(0.001)

Table A.347: Decision Tree HELLINGER Distance - Training

DT HELLINGER	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.1803(0.0621)	0.1986(0.0933)	0.2085(0.111)
Gmean	0.4144(0.0755)	0.4353(0.1041)	0.4357(0.105)
AUC ROC	0.5846(0.0308)	0.5952(0.0471)	0.5958(0.0488)
Accuracy	0.9822(0.0016)	0.9814(0.0027)	0.9827(0.0058)

Table A.348: Decision Tree HELLINGER Distance - Testing

DT SMOTENC	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.2723(0.0304)	0.2734(0.0199)	0.2909(0.047)
Gmean	0.4629(0.0376)	0.4657(0.035)	0.491(0.0688)
AUC ROC	0.6459(0.0155)	0.6478(0.0114)	0.6572(0.0219)
Accuracy	0.9786(0.0012)	0.9785(0.0007)	0.9791(0.0018)

Table A.349: Decision Tree SMOTENC - Training

DT SMOTENC	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.3085(0.1051)	0.2671(0.0915)	0.2811(0.1275)
Gmean	0.5819(0.1188)	0.5429(0.1172)	0.5375(0.1501)
AUC ROC	0.6703(0.0709)	0.6478(0.0699)	0.6496(0.0805)
Accuracy	0.9779(0.004)	0.9769(0.0044)	0.9804(0.0027)

Table A.350: Decision Tree SMOTENC - Testing

DT SMOTETOMEK	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.4396(0.0394)	0.4209(0.0306)	0.3617(0.1005)
Gmean	0.7898(0.0533)	0.778(0.0392)	0.6006(0.1869)
AUC ROC	0.8224(0.0382)	0.818(0.0257)	0.723(0.1068)
Accuracy	0.9626(0.0047)	0.9589(0.0033)	0.9752(0.0091)

Table A.351: Decision Tree SMOTETOMEK - Training

DT SMOTETOMEK	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.3984(0.1908)	0.4364(0.1522)	0.3293(0.1485)
Gmean	0.7599(0.229)	0.8132(0.166)	0.6276(0.2165)
AUC ROC	0.8042(0.1699)	0.8363(0.1332)	0.7121(0.1438)
Accuracy	0.9603(0.0102)	0.9586(0.0079)	0.9736(0.0157)

Table A.352: Decision Tree SMOTETOMEK - Testing

KNN	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.1204(0.0247)	0.1248(0.0363)	0.1216(0.0264)	0.1397(0.0294)	0.1454(0.0141)	0.1241(0.0402)
Gmean	0.2159(0.0492)	0.2174(0.0567)	0.2199(0.0396)	0.2442(0.0556)	0.2723(0.029)	0.2232(0.0764)
AUC ROC	0.5568(0.0136)	0.5591(0.0194)	0.5571(0.0142)	0.5683(0.0145)	0.5708(0.0088)	0.561(0.0213)
Accuracy	0.9816(0.0005)	0.9818(0.0008)	0.9816(0.001)	0.9799(0.0011)	0.98(0.0008)	0.9812(0.004)

Table A.353: KNN - Training - Part 1

KNN	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.2175(0.087)	0.2118(0.084)	0.1791(0.1111)	0.1597(0.0264)	0.1652(0.0312)	0.1745(0.0405)
Gmean	0.353(0.1097)	0.3512(0.1161)	0.2791(0.1576)	0.2792(0.0583)	0.2846(0.0494)	0.3127(0.0635)
AUC ROC	0.6088(0.0461)	0.6044(0.042)	0.5855(0.0591)	0.5778(0.0139)	0.581(0.0177)	0.5845(0.0218)
Accuracy	0.9835(0.0015)	0.9833(0.0018)	0.9867(0.0034)	0.9824(0.0007)	0.9822(0.001)	0.9821(0.0008)

Table A.354: KNN - Training - Part 2

KNN	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.1459(0.0582)	0.1583(0.0742)	0.1608(0.0692)
Gmean	0.2451(0.0901)	0.2598(0.1152)	0.2748(0.1062)
AUC ROC	0.5696(0.0292)	0.574(0.0368)	0.5768(0.0339)
Accuracy	0.9836(0.0014)	0.984(0.0016)	0.9837(0.0015)

Table A.355: KNN - Training - Part 3

KNN	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.0435(0.0595)	0.0435(0.0595)	0.0435(0.0595)	0.0871(0.0489)	0.0871(0.0489)	0.0871(0.0489)
Gmean	0.1326(0.1815)	0.1326(0.1815)	0.1326(0.1815)	0.2651(0.1482)	0.2651(0.1482)	0.2651(0.1482)
AUC ROC	0.5175(0.0295)	0.5175(0.0295)	0.5175(0.0295)	0.5394(0.0235)	0.5394(0.0235)	0.5399(0.0223)
Accuracy	0.9799(0.0023)	0.9799(0.0023)	0.9799(0.0023)	0.9797(0.0032)	0.9797(0.0032)	0.9807(0.0051)

Table A.356: KNN - Testing - Part 1

KNN	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.1756(0.2473)	0.1756(0.2473)	0.1497(0.159)	0.1163(0.0822)	0.1163(0.0822)	0.1163(0.0822)
Gmean	0.2637(0.3658)	0.2637(0.3658)	0.2761(0.2664)	0.2932(0.1744)	0.2932(0.1744)	0.2932(0.1744)
AUC ROC	0.5858(0.1275)	0.5858(0.1275)	0.5653(0.0727)	0.5519(0.0395)	0.5519(0.0395)	0.5519(0.0395)
Accuracy	0.9847(0.003)	0.9847(0.003)	0.9874(0.0032)	0.9827(0.0016)	0.9827(0.0016)	0.9827(0.0016)

Table A.357: KNN - Testing - Part 2

KNN	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0(0.0)	0.0(0.0)	0.0(0.0)
Gmean	0.0(0.0)	0.0(0.0)	0.0(0.0)
AUC ROC	0.5(0.0)	0.5(0.0)	0.5(0.0)
Accuracy	0.9887(0.0)	0.9887(0.0)	0.9887(0.0)

Table A.358: KNN - Testing - Part 3

LR	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.1422(0.0297)	0.1289(0.0391)	0.1475(0.0367)	0.007(0.0156)	0.0(0.0)	0.0039(0.0054)
Gmean	0.2221(0.0463)	0.2114(0.0611)	0.2383(0.0609)	0.011(0.0247)	0.0(0.0)	0.0067(0.0091)
AUC ROC	0.5634(0.0138)	0.5568(0.0184)	0.5658(0.0167)	0.5029(0.0068)	0.4999(0.0001)	0.5016(0.0022)
Accuracy	0.9881(0.0005)	0.9881(0.0006)	0.9884(0.0004)	0.9885(0.0002)	0.9885(0.0002)	0.9886(0.0002)

Table A.359: Logistic Regression - Training - Part 1

LR	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.1457(0.0632)	0.1446(0.061)	0.1443(0.0571)	0.1289(0.0385)	0.1183(0.0488)	0.1032(0.0548)
Gmean	0.2261(0.0947)	0.2279(0.0962)	0.2257(0.0841)	0.1972(0.0648)	0.1911(0.0746)	0.1637(0.0725)
AUC ROC	0.5651(0.0287)	0.5648(0.0284)	0.5642(0.0261)	0.558(0.0183)	0.5531(0.0233)	0.546(0.0267)
Accuracy	0.9886(0.0006)	0.9888(0.0004)	0.9886(0.0004)	0.9883(0.0006)	0.9883(0.0005)	0.988(0.0004)

Table A.360: Logistic Regression - Training - Part 2

LR	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.002(0.0044)	0.002(0.0044)	0.0026(0.0057)
Gmean	0.0033(0.0075)	0.0033(0.0075)	0.0038(0.0086)
AUC ROC	0.5008(0.0019)	0.5008(0.0018)	0.501(0.0025)
Accuracy	0.9886(0.0003)	0.9886(0.0002)	0.9886(0.0003)

Table A.361: Logistic Regression - Training - Part 3

LR	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.0513(0.0702)	0.0513(0.0702)	0.0988(0.1593)	0.0(0.0)	0.0(0.0)	0.0(0.0)
Gmean	0.1332(0.1823)	0.1332(0.1823)	0.1819(0.2636)	0.0(0.0)	0.0(0.0)	0.0(0.0)
AUC ROC	0.5212(0.0302)	0.5212(0.0302)	0.5434(0.0722)	0.5(0.0)	0.5(0.0)	0.5(0.0)
Accuracy	0.9872(0.0006)	0.9872(0.0006)	0.9877(0.0014)	0.9887(0.0)	0.9887(0.0)	0.9887(0.0)

Table A.362: Logistic Regression - Testing - Part 1

LR	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.1316(0.0025)	0.1289(0.0907)	0.1316(0.0025)	0.0751(0.1102)	0.0751(0.1102)	0.0751(0.1102)
Gmean	0.3331(0.0001)	0.294(0.1749)	0.3331(0.0001)	0.1607(0.2254)	0.1607(0.2254)	0.1607(0.2254)
AUC ROC	0.5549(0.0004)	0.5548(0.0391)	0.5549(0.0004)	0.5324(0.0492)	0.5324(0.0492)	0.5324(0.0492)
Accuracy	0.9887(0.0009)	0.9885(0.001)	0.9887(0.0009)	0.9877(0.0011)	0.9877(0.0011)	0.9877(0.0011)

Table A.363: Logistic Regression - Testing - Part 2

LR	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0(0.0)	0.0(0.0)	0.0(0.0)
Gmean	0.0(0.0)	0.0(0.0)	0.0(0.0)
AUC ROC	0.5(0.0)	0.5(0.0)	0.5(0.0)
Accuracy	0.9887(0.0)	0.9887(0.0)	0.9887(0.0)

Table A.364: Logistic Regression - Testing - Part 3

LR CS	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.4827(0.0298)	0.4734(0.0278)	0.4614(0.0384)	0.4989(0.0222)	0.4729(0.0431)	0.4915(0.0231)
Gmean	0.9039(0.0145)	0.8971(0.0133)	0.831(0.0916)	0.9304(0.0499)	0.9592(0.0093)	0.9071(0.0711)
AUC ROC	0.9116(0.0114)	0.9048(0.0117)	0.8585(0.0633)	0.936(0.0425)	0.9602(0.0089)	0.9199(0.0558)
Accuracy	0.9512(0.0045)	0.9507(0.005)	0.9592(0.0097)	0.9494(0.0135)	0.9364(0.0125)	0.9513(0.015)

Table A.365: Cost Sensitive Logistic Regression - Training - Part 1

LR CS	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.4815(0.0348)	0.4792(0.0344)	0.4873(0.0331)	0.4676(0.0242)	0.4675(0.0315)	0.477(0.0264)
Gmean	0.8818(0.0536)	0.9086(0.0202)	0.8877(0.0486)	0.8893(0.0217)	0.9075(0.0156)	0.8747(0.0541)
AUC ROC	0.8943(0.0433)	0.9155(0.0175)	0.8979(0.039)	0.8986(0.0196)	0.9132(0.014)	0.8865(0.0436)
Accuracy	0.955(0.0096)	0.9497(0.0044)	0.9555(0.0095)	0.9504(0.0048)	0.9468(0.0062)	0.9548(0.0113)

Table A.366: Cost Sensitive Logistic Regression - Training - Part 2

LR CS	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.4959(0.0272)	0.49(0.026)	0.4878(0.0208)
Gmean	0.9191(0.049)	0.9494(0.0163)	0.8853(0.0751)
AUC ROC	0.9256(0.0418)	0.9513(0.0151)	0.9024(0.0565)
Accuracy	0.9512(0.0127)	0.9443(0.0048)	0.9542(0.0138)

Table A.367: Cost Sensitive Logistic Regression - Training - Part 3

LR CS	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.4476(0.0647)	0.4431(0.0591)	0.4109(0.1011)	0.4447(0.1182)	0.4871(0.0271)	0.4386(0.1125)
Gmean	0.8817(0.0672)	0.8811(0.0664)	0.8024(0.1364)	0.8899(0.1795)	0.9593(0.022)	0.8795(0.1752)
AUC ROC	0.8861(0.0628)	0.8856(0.0619)	0.8239(0.1017)	0.9069(0.1425)	0.9598(0.022)	0.8965(0.1386)
Accuracy	0.9485(0.0082)	0.9475(0.0087)	0.9558(0.0146)	0.9463(0.0123)	0.9423(0.0086)	0.9473(0.01)

Table A.368: Cost Sensitive Logistic Regression - Testing - Part 1

LR CS	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.4682(0.0764)	0.4691(0.0779)	0.4634(0.0767)	0.4592(0.075)	0.4576(0.0269)	0.4198(0.1031)
Gmean	0.8877(0.1362)	0.9158(0.0585)	0.8772(0.1305)	0.9016(0.0995)	0.905(0.0452)	0.835(0.1609)
AUC ROC	0.8992(0.1146)	0.9177(0.0567)	0.8888(0.1089)	0.9074(0.0879)	0.9071(0.0433)	0.8548(0.1259)
Accuracy	0.9528(0.0134)	0.9458(0.0129)	0.9538(0.012)	0.9473(0.0106)	0.9465(0.0073)	0.9518(0.0137)

Table A.369: Cost Sensitive Logistic Regression - Testing - Part 2

LR CS	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.1084(0.2425)	0.2982(0.274)	0.18(0.2465)
Gmean	0.1951(0.4363)	0.5724(0.5231)	0.3858(0.5283)
AUC ROC	0.595(0.2129)	0.7726(0.25)	0.686(0.2547)
Accuracy	0.9812(0.0162)	0.9631(0.0237)	0.9656(0.0316)

Table A.370: Cost Sensitive Logistic Regression - Testing - Part 3

LR SMOTENC	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.4592(0.0432)	0.4558(0.0204)	0.4597(0.03)	0.4762(0.0294)	0.4905(0.0243)	0.4784(0.0373)
Gmean	0.7964(0.0495)	0.7994(0.0404)	0.8003(0.0499)	0.8352(0.0286)	0.858(0.0138)	0.8349(0.0218)
AUC ROC	0.8302(0.0369)	0.83(0.0283)	0.8358(0.0304)	0.8618(0.0164)	0.8748(0.0111)	0.8575(0.0166)
Accuracy	0.9645(0.0047)	0.9647(0.0046)	0.9641(0.0052)	0.9613(0.0039)	0.961(0.0038)	0.962(0.0048)

Table A.371: Logistic Regression SMOTENC - Training - Part 1

LR SMOTENC	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.4386(0.0414)	0.4454(0.0452)	0.4459(0.0507)	0.4377(0.0433)	0.4429(0.0459)	0.4503(0.0371)
Gmean	0.7616(0.0447)	0.7669(0.0542)	0.7663(0.0544)	0.7844(0.0345)	0.7749(0.0509)	0.7846(0.037)
AUC ROC	0.8121(0.0269)	0.8132(0.0284)	0.8128(0.0321)	0.8144(0.0245)	0.8198(0.0329)	0.8165(0.0241)
Accuracy	0.9651(0.0032)	0.9657(0.0039)	0.9665(0.0034)	0.9638(0.0046)	0.9641(0.0045)	0.9662(0.0064)

Table A.372: Logistic Regression SMOTENC - Training - Part 2

LR SMOTENC	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.4734(0.0485)	0.481(0.0275)	0.4864(0.0433)
Gmean	0.8497(0.0323)	0.8677(0.0151)	0.8566(0.0301)
AUC ROC	0.8662(0.0258)	0.8809(0.0113)	0.8749(0.0243)
Accuracy	0.9592(0.0049)	0.9578(0.0037)	0.9604(0.0045)

Table A.373: Logistic Regression SMOTENC - Training - Part 3

LR SMOTENC	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.4264(0.1588)	0.4264(0.1588)	0.4264(0.1588)	0.4733(0.0917)	0.4797(0.073)	0.4461(0.1549)
Gmean	0.784(0.1888)	0.784(0.1888)	0.784(0.1888)	0.8473(0.1102)	0.8621(0.0781)	0.8092(0.1941)
AUC ROC	0.8163(0.1342)	0.8163(0.1342)	0.8163(0.1342)	0.8598(0.0889)	0.87(0.0668)	0.8382(0.1362)
Accuracy	0.9626(0.0041)	0.9626(0.0041)	0.9626(0.0041)	0.9616(0.0056)	0.9601(0.0053)	0.9624(0.0058)

Table A.374: Logistic Regression SMOTENC - Testing - Part 1

LR SMOTENC	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.4419(0.1341)	0.4444(0.1349)	0.4398(0.1345)	0.4532(0.2125)	0.4501(0.2094)	0.4135(0.2006)
Gmean	0.7927(0.1437)	0.793(0.1438)	0.7807(0.1386)	0.7837(0.2563)	0.7835(0.256)	0.745(0.2432)
AUC ROC	0.8173(0.11)	0.8176(0.1102)	0.8072(0.1063)	0.8287(0.1631)	0.8286(0.1626)	0.7958(0.1536)
Accuracy	0.9646(0.005)	0.9651(0.0041)	0.9664(0.0045)	0.9654(0.005)	0.9651(0.0043)	0.9654(0.0047)

Table A.375: Logistic Regression SMOTENC - Testing - Part 2

LR SMOTENC	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.1303(0.1787)	0.1303(0.1787)	0.1303(0.1787)
Gmean	0.2465(0.3389)	0.2465(0.3389)	0.2465(0.3389)
AUC ROC	0.5742(0.1031)	0.5742(0.1031)	0.5742(0.1031)
Accuracy	0.9834(0.0082)	0.9834(0.0082)	0.9834(0.0082)

Table A.376: Logistic Regression SMOTENC - Testing - Part 3

LR SMOTETOMEK	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.4607(0.0334)	0.4681(0.0203)	0.4635(0.0166)	0.5034(0.0203)	0.5025(0.0277)	0.5012(0.0239)
Gmean	0.839(0.0327)	0.8533(0.0136)	0.8449(0.0213)	0.9166(0.0158)	0.9205(0.0205)	0.9057(0.0213)
AUC ROC	0.8595(0.0233)	0.8708(0.013)	0.8643(0.0159)	0.9225(0.0141)	0.9252(0.0182)	0.9146(0.0169)
Accuracy	0.9584(0.0055)	0.9574(0.0039)	0.9575(0.0052)	0.9538(0.0054)	0.9526(0.0039)	0.9549(0.0052)

Table A.377: Logistic Regression SMOTETOMEK - Training - Part 1

LR SMOTETOMEK	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.4801(0.0236)	0.4809(0.0194)	0.4871(0.0309)	0.4767(0.0208)	0.4706(0.0187)	0.4695(0.0208)
Gmean	0.8684(0.0246)	0.8701(0.0143)	0.8741(0.0202)	0.8631(0.0353)	0.8559(0.0256)	0.8541(0.0235)
AUC ROC	0.8796(0.0211)	0.8836(0.01)	0.8878(0.0176)	0.8812(0.0226)	0.8752(0.0147)	0.8714(0.0166)
Accuracy	0.9579(0.0042)	0.9571(0.0036)	0.9578(0.0039)	0.9572(0.0055)	0.9568(0.0063)	0.9574(0.0053)

Table A.378: Logistic Regression SMOTETOMEK - Training - Part 2

LR SMOTETOMEK	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.5108(0.017)	0.501(0.0259)	0.5112(0.0256)
Gmean	0.9239(0.0191)	0.9194(0.0173)	0.9247(0.0175)
AUC ROC	0.9289(0.0169)	0.9254(0.0151)	0.9286(0.0162)
Accuracy	0.9539(0.0042)	0.9525(0.0039)	0.9539(0.0043)

Table A.379: Logistic Regression SMOTETOMEK - Training - Part 3

LR SMOTETOMEK	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.4244(0.1114)	0.436(0.1057)	0.4244(0.1114)	0.4587(0.0599)	0.4898(0.0492)	0.4494(0.0704)
Gmean	0.8167(0.1326)	0.832(0.1131)	0.8167(0.1326)	0.8811(0.092)	0.9185(0.0565)	0.8685(0.0997)
AUC ROC	0.8351(0.1128)	0.8458(0.1001)	0.8351(0.1128)	0.8879(0.0807)	0.9206(0.0543)	0.877(0.0881)
Accuracy	0.9563(0.0086)	0.9558(0.0074)	0.9563(0.0086)	0.9521(0.006)	0.9516(0.0051)	0.9523(0.0059)

Table A.380: Logistic Regression SMOTETOMEK - Testing - Part 1

LR SMOTETOMEK	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.4887(0.0941)	0.4832(0.0938)	0.4801(0.09)	0.4565(0.0932)	0.455(0.0914)	0.4444(0.101)
Gmean	0.8823(0.1104)	0.8819(0.1105)	0.8816(0.1103)	0.855(0.1278)	0.8549(0.1277)	0.8411(0.1382)
AUC ROC	0.8909(0.0984)	0.8904(0.0986)	0.8902(0.0985)	0.8682(0.107)	0.8681(0.1069)	0.8572(0.1161)
Accuracy	0.9581(0.0078)	0.9571(0.0065)	0.9566(0.0056)	0.9566(0.0082)	0.9563(0.0079)	0.9566(0.0084)

Table A.381: Logistic Regression SMOTETOMEK - Testing - Part 2

LR SMOTETOMEK	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.32(0.2378)	0.4131(0.2367)	0.3158(0.2373)
Gmean	0.6285(0.4383)	0.7477(0.4185)	0.6072(0.4199)
AUC ROC	0.7732(0.2261)	0.8475(0.1972)	0.7527(0.2078)
Accuracy	0.9641(0.0232)	0.9591(0.0172)	0.9671(0.021)

Table A.382: Logistic Regression SMOTETOMEK - Testing - Part 3

RF	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0152(0.0049)	0.0121(0.0122)	0.0071(0.0103)
Gmean	0.0249(0.009)	0.0205(0.0216)	0.011(0.0162)
AUC ROC	0.5063(0.0024)	0.505(0.0055)	0.5028(0.0045)
Accuracy	0.9882(0.0003)	0.9883(0.0004)	0.9882(0.0004)

Table A.383: Random Forest - Training

RF	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0(0.0)	0.0263(0.0588)	0.0263(0.0588)
Gmean	0.0(0.0)	0.0666(0.149)	0.0666(0.149)
AUC ROC	0.4997(0.0003)	0.5107(0.0247)	0.5107(0.0247)
Accuracy	0.9882(0.0007)	0.9882(0.0007)	0.9882(0.0007)

Table A.384: Random Forest - Testing

RF CS	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.433(0.0544)	0.4144(0.0575)	0.3514(0.1209)
Gmean	0.7473(0.0554)	0.7318(0.0542)	0.575(0.2279)
AUC ROC	0.7929(0.0321)	0.78(0.0358)	0.7171(0.1023)
Accuracy	0.9684(0.0034)	0.9678(0.0026)	0.9764(0.0107)

Table A.385: Cost Sensitive Random Forest - Training

RF CS	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.3991(0.172)	0.3746(0.1478)	0.3117(0.1988)
Gmean	0.744(0.1832)	0.7057(0.1568)	0.5947(0.2144)
AUC ROC	0.7838(0.1446)	0.7523(0.1195)	0.6895(0.1427)
Accuracy	0.9634(0.0069)	0.9664(0.0071)	0.9724(0.0159)

Table A.386: Cost Sensitive Random Forest - Testing

SVM	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.0017(0.0037)	0.0022(0.005)	0.0(0.0)	0.0039(0.0054)	0.002(0.0044)	0.0038(0.0052)
Gmean	0.0033(0.0074)	0.0038(0.0086)	0.0(0.0)	0.0067(0.0091)	0.0033(0.0075)	0.0067(0.0091)
AUC ROC	0.5006(0.0017)	0.5009(0.0024)	0.4997(0.0003)	0.5016(0.0021)	0.5008(0.0017)	0.5015(0.002)
Accuracy	0.9882(0.0004)	0.9883(0.0003)	0.9882(0.0007)	0.9885(0.0003)	0.9886(0.0002)	0.9884(0.0006)

Table A.387: SVM - Training - Part 1

SVM	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.1877(0.0604)	0.1696(0.0326)	0.1913(0.0549)	0.1069(0.0448)	0.1049(0.034)	0.1109(0.0374)
Gmean	0.2919(0.0826)	0.2619(0.052)	0.3113(0.0813)	0.1782(0.072)	0.1691(0.058)	0.1834(0.0581)
AUC ROC	0.5861(0.0291)	0.5793(0.0164)	0.588(0.0264)	0.5476(0.0202)	0.5473(0.0157)	0.5499(0.0171)
Accuracy	0.9868(0.0008)	0.9869(0.0005)	0.9868(0.0007)	0.9862(0.0007)	0.9862(0.0004)	0.9863(0.0006)

Table A.388: SVM - Training - Part 2

SVM	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0(0.0)	0.0(0.0)	0.0(0.0)
Gmean	0.0(0.0)	0.0(0.0)	0.0(0.0)
AUC ROC	0.5(0.0)	0.5(0.0)	0.5(0.0)
Accuracy	0.9887(0.0)	0.9887(0.0)	0.9887(0.0)

Table A.389: SVM - Training - Part 3

SVM	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)
Gmean	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)
AUC ROC	0.5(0.0)	0.5(0.0)	0.5(0.0)	0.5(0.0)	0.5(0.0)	0.5(0.0)
Accuracy	0.9887(0.0)	0.9887(0.0)	0.9887(0.0)	0.9887(0.0)	0.9887(0.0)	0.9887(0.0)

Table A.390: SVM - Testing - Part 1

SVM	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.2226(0.0546)	0.2226(0.0546)	0.222(0.056)	0.1032(0.1075)	0.1032(0.1075)	0.1032(0.1075)
Gmean	0.4431(0.0617)	0.4431(0.0617)	0.443(0.0618)	0.2274(0.2151)	0.2274(0.2151)	0.2274(0.2151)
AUC ROC	0.5983(0.025)	0.5983(0.025)	0.5982(0.0253)	0.543(0.0472)	0.543(0.0472)	0.543(0.0472)
Accuracy	0.9877(0.001)	0.9877(0.001)	0.9875(0.0015)	0.9869(0.0026)	0.9869(0.0026)	0.9869(0.0026)

Table A.391: SVM - Testing - Part 2

SVM	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0(0.0)	0.0(0.0)	0.0(0.0)
Gmean	0.0(0.0)	0.0(0.0)	0.0(0.0)
AUC ROC	0.5(0.0)	0.5(0.0)	0.5(0.0)
Accuracy	0.9887(0.0)	0.9887(0.0)	0.9887(0.0)

Table A.392: SVM - Testing - Part 3

SVM CS	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.4615(0.0296)	0.4585(0.0409)	0.4452(0.0451)	0.4917(0.035)	0.4889(0.04)	0.4942(0.0336)
Gmean	0.8526(0.0265)	0.8492(0.0235)	0.8386(0.0327)	0.9118(0.0378)	0.9322(0.021)	0.8981(0.0454)
AUC ROC	0.8691(0.0199)	0.8678(0.0184)	0.86(0.0252)	0.9185(0.0316)	0.9358(0.0188)	0.9068(0.0387)
Accuracy	0.9558(0.0067)	0.9552(0.0073)	0.9556(0.0071)	0.9522(0.0066)	0.9474(0.0066)	0.9548(0.0095)

Table A.393: Cost Sensitive SVM - Training - Part 1

SVM CS	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.4494(0.0389)	0.4528(0.0276)	0.4587(0.0395)	0.4427(0.0349)	0.455(0.0374)	0.4445(0.0438)
Gmean	0.8075(0.0362)	0.8333(0.0112)	0.8195(0.0087)	0.8173(0.012)	0.8335(0.0227)	0.8201(0.0261)
AUC ROC	0.8373(0.0249)	0.8526(0.0136)	0.8425(0.0047)	0.8449(0.0124)	0.8537(0.0178)	0.8473(0.0186)
Accuracy	0.9606(0.0092)	0.9578(0.005)	0.9607(0.0093)	0.9578(0.0068)	0.9583(0.006)	0.9577(0.0068)

Table A.394: Cost Sensitive SVM - Training - Part 2

SVM CS	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.5053(0.0248)	0.4887(0.0273)	0.4984(0.0177)
Gmean	0.9064(0.0196)	0.931(0.0076)	0.8993(0.0286)
AUC ROC	0.9123(0.0172)	0.9343(0.0065)	0.9075(0.0242)
Accuracy	0.9559(0.0064)	0.9477(0.0072)	0.9555(0.0052)

Table A.395: Cost Sensitive SVM - Training - Part 3

SVM CS	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.3966(0.0949)	0.3966(0.0949)	0.3966(0.0949)	0.4612(0.094)	0.4455(0.0615)	0.4562(0.1022)
Gmean	0.8121(0.1448)	0.8121(0.1448)	0.8121(0.1448)	0.8812(0.095)	0.9(0.0978)	0.8666(0.125)
AUC ROC	0.8323(0.1105)	0.8323(0.1105)	0.8323(0.1105)	0.8878(0.0841)	0.9059(0.0862)	0.8778(0.104)
Accuracy	0.9508(0.0074)	0.9508(0.0074)	0.9508(0.0074)	0.9518(0.0068)	0.9443(0.0111)	0.9538(0.009)

Table A.396: Cost Sensitive SVM - Testing - Part 1

SVM CS	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.4488(0.0877)	0.4511(0.0846)	0.4488(0.0877)	0.4083(0.1396)	0.4083(0.1396)	0.4083(0.1396)
Gmean	0.8416(0.1375)	0.8561(0.1156)	0.8416(0.1375)	0.8048(0.1969)	0.8048(0.1969)	0.8048(0.1969)
AUC ROC	0.8578(0.1151)	0.8676(0.1008)	0.8578(0.1151)	0.8344(0.1402)	0.8344(0.1402)	0.8344(0.1402)
Accuracy	0.9578(0.0089)	0.9553(0.0044)	0.9578(0.0089)	0.9548(0.0073)	0.9548(0.0073)	0.9548(0.0073)

Table A.397: Cost Sensitive SVM - Testing - Part 2

SVM CS	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0(0.0)	0.0(0.0)	0.092(0.2056)
Gmean	0.0(0.0)	0.0(0.0)	0.1833(0.41)
AUC ROC	0.5(0.0)	0.5(0.0)	0.5834(0.1866)
Accuracy	0.9887(0.0)	0.9887(0.0)	0.9799(0.0196)

Table A.398: Cost Sensitive SVM - Testing - Part 3

SVM SMOTENC	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.1003(0.0459)	0.0926(0.0532)	0.0889(0.0538)	0.3746(0.06)	0.3794(0.0302)	0.3817(0.0399)
Gmean	0.2247(0.11)	0.215(0.1317)	0.2112(0.12)	0.6712(0.078)	0.683(0.0424)	0.6869(0.064)
AUC ROC	0.5564(0.0288)	0.5508(0.0325)	0.5497(0.0308)	0.7502(0.0423)	0.7553(0.0231)	0.755(0.0363)
Accuracy	0.9655(0.0153)	0.9648(0.0137)	0.9649(0.0149)	0.9669(0.0069)	0.9669(0.0063)	0.9674(0.0061)

Table A.399: SVM SMOTENC - Training - Part 1

SVM SMOTENC	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.3049(0.0859)	0.3(0.0765)	0.3144(0.0632)	0.2861(0.0619)	0.2847(0.0709)	0.2934(0.0599)
Gmean	0.5155(0.1174)	0.523(0.1081)	0.5333(0.0664)	0.5131(0.0764)	0.4908(0.086)	0.5158(0.0741)
AUC ROC	0.6742(0.0491)	0.6694(0.0418)	0.6793(0.0377)	0.6672(0.0336)	0.6654(0.0414)	0.6712(0.0328)
Accuracy	0.9764(0.0024)	0.9762(0.0028)	0.9763(0.0031)	0.9734(0.0042)	0.9738(0.004)	0.9737(0.004)

Table A.400: SVM SMOTENC - Training - Part 2

SVM SMOTENC	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.4564(0.0102)	0.4484(0.0087)	0.4409(0.0186)
Gmean	0.8307(0.0232)	0.8241(0.025)	0.8191(0.0244)
AUC ROC	0.858(0.015)	0.8492(0.0176)	0.8429(0.0175)
Accuracy	0.958(0.0023)	0.958(0.0029)	0.9581(0.0031)

Table A.401: SVM SMOTENC - Training - Part 3

SVM SMOTENC	No Scaling			NORM		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.1126(0.1016)	0.1126(0.1016)	0.1126(0.1016)	0.4161(0.1573)	0.4161(0.1573)	0.4161(0.1573)
Gmean	0.3119(0.2035)	0.3119(0.2035)	0.3119(0.2035)	0.7395(0.1389)	0.7395(0.1389)	0.7395(0.1389)
AUC ROC	0.556(0.061)	0.556(0.061)	0.556(0.061)	0.7753(0.1077)	0.7753(0.1077)	0.7753(0.1077)
Accuracy	0.9691(0.0146)	0.9691(0.0146)	0.9691(0.0146)	0.9684(0.0083)	0.9684(0.0083)	0.9684(0.0083)

Table A.402: SVM SMOTENC - Testing - Part 1

SVM SMOTENC	POWER			STD		
Optimized for	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.3022(0.1521)	0.3022(0.1521)	0.3022(0.1521)	0.3076(0.1905)	0.3076(0.1905)	0.3076(0.1905)
Gmean	0.5718(0.1678)	0.5718(0.1678)	0.5718(0.1678)	0.5839(0.1928)	0.5839(0.1928)	0.5839(0.1928)
AUC ROC	0.6697(0.0903)	0.6697(0.0903)	0.6697(0.0903)	0.6796(0.1214)	0.6796(0.1214)	0.6796(0.1214)
Accuracy	0.9767(0.0056)	0.9767(0.0056)	0.9767(0.0056)	0.9747(0.0052)	0.9747(0.0052)	0.9747(0.0052)

Table A.403: SVM SMOTENC - Testing - Part 2

SVM SMOTENC	TRD		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0(0.0)	0.0(0.0)	0.0(0.0)
Gmean	0.0(0.0)	0.0(0.0)	0.0(0.0)
AUC ROC	0.5(0.0)	0.5(0.0)	0.5(0.0)
Accuracy	0.9887(0.0)	0.9887(0.0)	0.9887(0.0)

Table A.404: SVM SMOTENC - Testing - Part 3

SVM SMOTETOMEK	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0899(0.0503)	0.0907(0.0407)	0.0869(0.0605)
Gmean	0.2096(0.1206)	0.2145(0.108)	0.1859(0.1288)
AUC ROC	0.5496(0.0317)	0.5494(0.0254)	0.5463(0.0366)
Accuracy	0.9668(0.0126)	0.9659(0.0132)	0.9677(0.0123)

Table A.405: SVM SMOTETOMEK - Training

SVM SMOTETOMEK	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0951(0.0696)	0.0951(0.0696)	0.0951(0.0696)
Gmean	0.291(0.1732)	0.291(0.1732)	0.291(0.1732)
AUC ROC	0.5443(0.0404)	0.5443(0.0404)	0.5443(0.0404)
Accuracy	0.9676(0.0143)	0.9676(0.0143)	0.9676(0.0143)

Table A.406: SVM SMOTETOMEK - Testing

XGBoost	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.1643(0.0369)	0.1534(0.0433)	0.1482(0.0466)
Gmean	0.2528(0.0493)	0.2367(0.0551)	0.2372(0.0716)
AUC ROC	0.574(0.0176)	0.5702(0.0218)	0.5662(0.0229)
Accuracy	0.9884(0.0005)	0.9874(0.0003)	0.9877(0.0004)

Table A.407: XGBoost - Training

XGBoost	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0777(0.071)	0.0996(0.1011)	0.0772(0.0707)
Gmean	0.1998(0.1824)	0.2272(0.2149)	0.1997(0.1823)
AUC ROC	0.5319(0.0309)	0.5425(0.0467)	0.5317(0.031)
Accuracy	0.9867(0.0023)	0.9859(0.0022)	0.9862(0.0029)

Table A.408: XGBoost - Testing

XGBoost CS	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.1581(0.047)	0.1606(0.0449)	0.1462(0.0504)
Gmean	0.2432(0.0639)	0.2494(0.0683)	0.2287(0.0775)
AUC ROC	0.5712(0.0223)	0.5729(0.0222)	0.5648(0.0232)
Accuracy	0.9878(0.0003)	0.988(0.0004)	0.9876(0.0003)

Table A.409: Cost Sensitive XGBoost - Training

XGBoost CS	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.0783(0.0715)	0.0777(0.071)	0.0777(0.071)
Gmean	0.1998(0.1824)	0.1998(0.1824)	0.1998(0.1824)
AUC ROC	0.5321(0.031)	0.5319(0.0309)	0.5318(0.0311)
Accuracy	0.9869(0.0019)	0.9867(0.0021)	0.9864(0.0024)

Table A.410: Cost Sensitive XGBoost - Testing

XGBoost SMOTENC	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.3161(0.0585)	0.3335(0.0654)	0.3125(0.0708)
Gmean	0.5401(0.0629)	0.5674(0.0874)	0.5295(0.0932)
AUC ROC	0.681(0.0335)	0.693(0.0386)	0.6779(0.0395)
Accuracy	0.9764(0.003)	0.9767(0.0026)	0.9766(0.0032)

Table A.411: XGBoost SMOTENC - Training

XGBoost SMOTENC	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.2606(0.1196)	0.2606(0.1196)	0.2606(0.1196)
Gmean	0.538(0.1408)	0.538(0.1408)	0.538(0.1408)
AUC ROC	0.6464(0.0722)	0.6464(0.0722)	0.6464(0.0722)
Accuracy	0.9741(0.0045)	0.9741(0.0045)	0.9741(0.0045)

Table A.412: XGBoost SMOTENC - Testing

XGBoost SMOTETOMEK	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.2648(0.0492)	0.243(0.0259)	0.2445(0.0458)
Gmean	0.4199(0.0694)	0.4119(0.0245)	0.3824(0.0679)
AUC ROC	0.632(0.0274)	0.6227(0.0153)	0.6207(0.0248)
Accuracy	0.9837(0.0006)	0.9826(0.0008)	0.984(0.0007)

Table A.413: XGBoost SMOTETOMEK - Training

XGBoost SMOTETOMEK	No Scaling		
Optimized for	**F2**	**G-mean**	**PR-Recall**
F2	0.2226(0.0829)	0.2226(0.0829)	0.1806(0.0879)
Gmean	0.4629(0.0866)	0.4629(0.0866)	0.4082(0.1101)
AUC ROC	0.6067(0.0398)	0.6067(0.0398)	0.5853(0.0484)
Accuracy	0.9824(0.0032)	0.9824(0.0032)	0.9837(0.0025)

Table A.414: XGBoost SMOTETOMEK - Testing

NN	No Scaling			NORM		
Optimized For	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.0019(0.0041)	0.0085(0.0098)	0.0058(0.0053)	0.0039(0.0054)	0.0045(0.0063)	0.0065(0.006)
Gmean	0.0033(0.0074)	0.0121(0.0119)	0.01(0.0091)	0.0067(0.0091)	0.0072(0.0099)	0.0105(0.0097)
AUC ROC	0.5007(0.0019)	0.5038(0.0045)	0.5024(0.0022)	0.5014(0.0022)	0.5017(0.0027)	0.5026(0.0026)
Accuracy	0.9885(0.0002)	0.9885(0.0001)	0.9886(0.0001)	0.9883(0.0004)	0.9883(0.0005)	0.9884(0.0002)

Table A.415: Neural Network - Training - Part 1

NN	POWER			STD		
Optimized For	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.1619(0.0358)	0.1671(0.0588)	0.1798(0.0574)	0.1509(0.0345)	0.1567(0.0497)	0.1706(0.0305)
Gmean	0.2602(0.0485)	0.2583(0.087)	0.275(0.0778)	0.2363(0.0543)	0.2501(0.071)	0.2574(0.0335)
AUC ROC	0.576(0.0175)	0.5782(0.0282)	0.5854(0.0291)	0.5695(0.0167)	0.5732(0.0224)	0.579(0.0139)
Accuracy	0.9869(0.001)	0.9868(0.0007)	0.987(0.0005)	0.9871(0.0008)	0.9873(0.0009)	0.9875(0.0008)

Table A.416: Neural Network - Training - Part 2

NN	TRD		
Optimized For	**F2**	**G-mean**	**PR-Recall**
F2	0.0065(0.006)	0.0069(0.0108)	0.0161(0.0157)
Gmean	0.0105(0.0097)	0.011(0.0169)	0.0264(0.0234)
AUC ROC	0.5024(0.0026)	0.5026(0.0047)	0.5071(0.0078)
Accuracy	0.988(0.0002)	0.9879(0.0004)	0.9875(0.0008)

Table A.417: Neural Network - Training - Part 3

NN	No Scaling			NORM		
Optimized For	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.0256(0.0573)	0.0(0.0)	0.0(0.0)	0.0244(0.0545)	0.0(0.0)	0.0263(0.0588)
Gmean	0.0666(0.1489)	0.0(0.0)	0.0(0.0)	0.0665(0.1487)	0.0(0.0)	0.0666(0.149)
AUC ROC	0.5109(0.0243)	0.5(0.0)	0.4999(0.0003)	0.5106(0.0237)	0.5(0.0)	0.511(0.0246)
Accuracy	0.9885(0.0006)	0.9887(0.0)	0.9885(0.0006)	0.988(0.0017)	0.9887(0.0)	0.9887(0.0)

Table A.418: Neural Network - Testing - Part 1

NN	POWER			STD		
Optimized For	**F2**	**G-mean**	**PR-Recall**	**F2**	**G-mean**	**PR-Recall**
F2	0.184(0.1636)	0.1204(0.1178)	0.1877(0.1245)	0.0698(0.156)	0.1714(0.1658)	0.2733(0.1429)
Gmean	0.3599(0.2428)	0.2546(0.2391)	0.3697(0.2237)	0.1152(0.2575)	0.3035(0.2805)	0.4961(0.1536)
AUC ROC	0.5866(0.0832)	0.5542(0.0543)	0.5866(0.0622)	0.5328(0.0734)	0.5768(0.0739)	0.6304(0.0739)
Accuracy	0.9862(0.0015)	0.9872(0.0016)	0.9862(0.0025)	0.9885(0.0006)	0.9885(0.0019)	0.9859(0.0027)

Table A.419: Neural Network - Testing - Part 2

NN	TRD		
Optimized For	**F2**	**G-mean**	**PR-Recall**
F2	0.0(0.0)	0.0(0.0)	0.0(0.0)
Gmean	0.0(0.0)	0.0(0.0)	0.0(0.0)
AUC ROC	0.5(0.0)	0.5(0.0)	0.5(0.0)
Accuracy	0.9887(0.0)	0.9887(0.0)	0.9887(0.0)

Table A.420: Neural Network - Testing - Part 3

A.7. Summary of Classification Results.

			Training - Testing			
Model	Best Scaling	Best Opt. For	F2	Gmean	AUC ROC	Accuracy
ADABOOST	No Scaling	F2	0.8671(0.0107)	0.9282(0.0067)	0.9314(0.0063)	0.9873(0.0008)
			0.8585(0.0169)	0.9236(0.0116)	0.9261(0.0107)	0.9865(0.0026)
ADACOST	No Scaling	F2	0.8848(0.0074)	0.9687(0.0197)	0.9698(0.0184)	0.9755(0.0107)
			0.8584(0.081)	0.9481(0.0711)	0.9513(0.0642)	0.9758(0.0105)
CATBOOST	No Scaling	PR-Recall	0.8803(0.0202)	0.9357(0.0122)	0.9384(0.0115)	0.9885(0.0016)
			0.88(0.0363)	0.9356(0.022)	0.9375(0.0207)	0.9885(0.0032)
Decision Tree	No Scaling	F2	0.8619(0.0091)	0.925(0.0051)	0.9286(0.0046)	0.9869(0.0009)
			0.8559(0.0335)	0.9209(0.0198)	0.9237(0.0184)	0.9868(0.0037)
Cost Sensitive Decision Tree	No Scaling	F2	0.8784(0.0058)	0.9378(0.0049)	0.9403(0.0046)	0.9869(0.0011)
			0.8522(0.0369)	0.9227(0.0199)	0.9252(0.0186)	0.9847(0.0044)
Decision Tree - HELLINGER Distance	No Scaling	Gmean	0.8811(0.0034)	0.983(0.0005)	0.9832(0.0005)	0.9679(0.001)
			0.8803(0.0134)	0.983(0.0022)	0.9832(0.0021)	0.9679(0.0041)
Decision Tree - SMOTENC	No Scaling	PR-Recall	0.8817(0.015)	0.9376(0.0092)	0.9402(0.0085)	0.9881(0.0012)
			0.8465(0.0412)	0.9172(0.0265)	0.9203(0.0246)	0.9852(0.0031)
Decision Tree - SMOTETOMEK	No Scaling	Gmean	0.9197(0.0071)	0.9676(0.0023)	0.9683(0.0024)	0.9882(0.0019)
			0.9164(0.0221)	0.9674(0.0104)	0.9676(0.0102)	0.9873(0.0037)
KNN	NORM	Gmean	0.7223(0.0186)	0.8366(0.0125)	0.851(0.0106)	0.977(0.0015)
			0.7391(0.1135)	0.8474(0.0769)	0.8601(0.0623)	0.9781(0.0077)
Logistic Regression	NORM	F2	0.8805(0.0137)	0.9376(0.0082)	0.9403(0.0074)	0.9878(0.0011)
			0.8728(0.0272)	0.9324(0.0164)	0.9344(0.0154)	0.9875(0.0033)
Cost Sensitive Logistic Regression	NORM	F2	0.9346(0.0055)	0.9896(0.0029)	0.9897(0.0029)	0.984(0.0012)
			0.936(0.0179)	0.936(0.0179)	0.9894(0.0073)	0.9847(0.003)
Logistic Regression - SMOTENC	NORM	F2	0.9217(0.0057)	0.9797(0.0044)	0.98(0.0043)	0.9838(0.0019)
			0.9338(0.0157)	0.987(0.0074)	0.987(0.0074)	0.985(0.0033)

Table A.421: Classification Summary Results For All Defects - Part 1

			Training - Testing			
Model	Best Scaling	Best Opt. For	F2	Gmean	AUC ROC	Accuracy
Logistic Regression - SMOTETOMEK	NORM	F2	0.9326(0.0058)	0.9856(0.0043)	0.9857(0.0042)	0.985(0.0013)
			0.9296(0.0181)	0.9843(0.0078)	0.9843(0.0078)	0.9847(0.0035)
SVM	NORM	F2	0.8568(0.0267)	0.9234(0.0189)	0.9274(0.0173)	0.9858(0.0014)
			0.8654(0.0296)	0.9314(0.0169)	0.9333(0.016)	0.9855(0.0031)
Cost Sensitive SVM	NORM	F2	0.9277(0.0054)	0.9861(0.005)	0.9862(0.0049)	0.9831(0.001)
			0.9219(0.0068)	0.9831(0.0103)	0.9832(0.0102)	0.9827(0.0026)
Random Forest	No Scaling	F2	0.7069(0.0172)	0.8156(0.0114)	0.8352(0.0104)	0.9813(0.0009)
			0.7274(0.0645)	0.8297(0.0428)	0.8297(0.0428)	0.9827(0.0044)
Cost Sensitive Random Forest	No Scaling	F2	0.865(0.0121)	0.9465(0.0075)	0.9481(0.007)	0.9786(0.0018)
			0.8602(0.0415)	0.9415(0.0322)	0.9429(0.0309)	0.9794(0.0035)
SVM SMOTENC	NORM	F2	0.8842(0.0049)	0.9559(0.0033)	0.9569(0.0032)	0.9812(0.0005)
			0.8919(0.0379)	0.9616(0.0262)	0.9621(0.0255)	0.9817(0.0026)
SVM SMOTETOMEK	No Scaling	F2	0.1595(0.0178)	0.3503(0.0266)	0.5623(0.0086)	0.9361(0.0047)
			0.1711(0.0535)	0.3917(0.0717)	0.5679(0.0256)	0.9353(0.0106)
XGBOOST	No Scaling	Gmean	0.8911(0.0146)	0.9421(0.008)	0.9444(0.0074)	0.9894(0.0015)
			0.8639(0.0244)	0.9267(0.0147)	0.929(0.0137)	0.987(0.0025)
Cost Sensitive XGBOOST	No Scaling	Gmean	0.8913(0.0183)	0.9423(0.0102)	0.9448(0.0094)	0.9893(0.0018)
			0.8878(0.0432)	0.9411(0.0269)	0.9427(0.0254)	0.9888(0.0035)
XGBOOST - SMOTENC	No Scaling	Gmean	0.9153(0.0132)	0.9624(0.0092)	0.9634(0.0089)	0.9889(0.001)
			0.9158(0.0557)	0.9648(0.0314)	0.9655(0.0305)	0.988(0.0057)
XGBOOST - SMOTETOMEK	No Scaling	F2	0.8986(0.016)	0.9493(0.0098)	0.951(0.0092)	0.9888(0.0013)
			0.8971(0.0571)	0.9489(0.0344)	0.9503(0.0331)	0.9885(0.0045)
Neural Network	POWER	Gmean	0.8561(0.0125)	0.9216(0.0081)	0.9253(0.0075)	0.9864(0.0009)
			0.82(0.052)	0.897(0.0346)	0.9022(0.0314)	0.985(0.0035)

Table A.422: Summary Results For All Defects - Part 2

			Training - Testing			
Model	Best Scaling	Best Opt. For	F2	Gmean	AUC ROC	Accuracy
ADABOOST	No Scaling	F2	0.2291(0.0625)	0.3685(0.0809)	0.6205(0.0363)	0.9842(0.002)
			0.1435(0.2079)	0.2371(0.3275)	0.5669(0.1025)	0.9832(0.0056)
ADACOST	No Scaling	F2	0.401(0.0107)	0.964(0.0012)	0.9647(0.0011)	0.93(0.0022)
			0.3971(0.032)	0.964(0.0047)	0.9647(0.0045)	0.93(0.0089)
CATBOOST	No Scaling	Gmean	0.1801(0.0683)	0.2544(0.093)	0.5807(0.0303)	0.9915(0.0007)
			0.0333(0.0745)	0.0755(0.1689)	0.5142(0.0317)	0.9908(0.0006)
Decision Tree	No Scaling	Gmean	0.235(0.0725)	0.3882(0.1079)	0.6242(0.0416)	0.9838(0.0018)
			0.0263(0.0588)	0.0752(0.1681)	0.5091(0.0316)	0.9809(0.0033)
Cost Sensitive Decision Tree	No Scaling	Gmean	0.264(0.0566)	0.4187(0.0687)	0.6409(0.033)	0.6409(0.033)
			0.1382(0.1656)	0.281(0.2801)	0.5677(0.0873)	0.9847(0.0016)
Decision Tree - HELLINGER Distance	No Scaling	PR-Recall	0.2033(0.0508)	0.3213(0.0724)	0.6032(0.025)	0.9848(0.0017)
			0.2035(0.1688)	0.3832(0.2466)	0.5956(0.0826)	0.9875(0.0023)
Decision Tree - SMOTENC	No Scaling	Gmean	0.2805(0.0341)	0.4559(0.047)	0.6573(0.022)	0.9829(0.0014)
			0.1853(0.0834)	0.4334(0.0908)	0.5928(0.0421)	0.9819(0.004)
Decision Tree - SMOTETOMEK	No Scaling	Gmean	0.432(0.0929)	0.7711(0.0656)	0.8147(0.0514)	0.9676(0.0102)
			0.4648(0.1225)	0.8173(0.1108)	0.835(0.0894)	0.971(0.01)
KNN	STD	F2	0.2184(0.0726)	0.3273(0.1052)	0.6084(0.0345)	0.9865(0.0018)
			0.1678(0.189)	0.3075(0.3005)	0.5802(0.0942)	0.985(0.0047)
Logistic Regression	STD	F2	0.0136(0.0195)	0.0201(0.0288)	0.5058(0.0088)	0.9903(0.0002)
			0.0(0.0)	0.0(0.0)	0.4995(0.0003)	0.9898(0.0009)
Cost Sensitive Logistic Regression	NORM	F2	0.4471(0.0427)	0.8189(0.0761)	0.8526(0.0554)	0.9642(0.0159)
			0.409(0.0965)	0.7952(0.1403)	0.8199(0.1172)	0.9656(0.0133)
Logistic Regression - SMOTENC	No Scaling	F2	0.3985(0.0395)	0.7827(0.0402)	0.824(0.0299)	0.962(0.0062)
			0.3632(0.0586)	0.7669(0.0982)	0.7916(0.072)	0.9621(0.0045)

Table A.423: Summary Results For CHD - Part 1

			Training - Testing			
Model	Best Scaling	Best Opt. For	F2	Gmean	AUC ROC	Accuracy
Logistic Regression - SMOTETOMEK	NORM	Gmean	0.4302(0.0382)	0.908(0.0186)	0.9159(0.0175)	0.949(0.0086)
			0.419(0.0815)	0.901(0.1071)	0.9072(0.0974)	0.9494(0.0042)
SVM	POWER	F2	0.1288(0.0602)	0.1941(0.0858)	0.5589(0.0288)	0.9897(0.001)
			0.0667(0.0915)	0.1511(0.2069)	0.5273(0.0397)	0.9888(0.0035)
Cost Sensitive SVM	NORM	F2	0.4802(0.0473)	0.8113(0.0497)	0.8424(0.0356)	0.9725(0.0012)
			0.4813(0.1276)	0.8336(0.1203)	0.8495(0.0977)	0.9718(0.0046)
Random Forest	No Scaling	F2	0.122(0.0476)	0.122(0.0476)	0.5533(0.0211)	0.9917(0.0003)
			0.0(0.0)	0.0(0.0)	0.5(0.0)	0.9908(0.0006)
Cost Sensitive Random Forest	No Scaling	F2	0.3977(0.0495)	0.6445(0.0562)	0.7411(0.0269)	0.98(0.0023)
			0.4196(0.1701)	0.7254(0.1849)	0.7732(0.1371)	0.9781(0.0025)
SVM SMOTENC	NORM	F2	0.3208(0.0421)	0.6323(0.0308)	0.7324(0.0201)	0.967(0.0069)
			0.3433(0.127)	0.7172(0.2077)	0.7692(0.1222)	0.9667(0.0081)
SVM SMOTETOMEK	No Scaling	Gmean	0.001(0.0022)	0.0037(0.0083)	0.4954(0.0028)	0.9795(0.0037)
			0.0(0.0)	0.0(0.0)	0.4963(0.0034)	0.9835(0.0071)
XGBOOST	No Scaling	F2	0.2259(0.0802)	0.3173(0.0949)	0.6046(0.0389)	0.9909(0.0007)
			0.0606(0.1355)	0.1067(0.2386)	0.5279(0.0632)	0.9901(0.0011)
Cost Sensitive XGBOOST	No Scaling	PR-Recall	0.1802(0.0868)	0.2598(0.1189)	0.5817(0.04)	0.9913(0.0008)
			0.0678(0.0929)	0.1511(0.207)	0.5283(0.0391)	0.9908(0.0011)
XGBOOST - SMOTENC	No Scaling	F2	0.3195(0.0857)	0.505(0.1047)	0.6737(0.0459)	0.9838(0.0019)
			0.3007(0.1175)	0.5635(0.1489)	0.6646(0.0821)	0.984(0.0029)
XGBOOST - SMOTETOMEK	No Scaling	Gmean	0.3012(0.0602)	0.4525(0.0868)	0.6581(0.0358)	0.9867(0.0009)
			0.2745(0.1837)	0.5024(0.1866)	0.638(0.1021)	0.9873(0.0016)
Neural Network	POWER	F2	0.107(0.0565)	0.1606(0.0752)	0.5493(0.0276)	0.9892(0.0007)
			0.0948(0.1403)	0.1823(0.2556)	0.5412(0.0642)	0.9883(0.0034)

Table A.424: Summary Results For CHD - Part 2

			Training - Testing			
Model	Best Scaling	Best Opt. For	F2	Gmean	AUC ROC	Accuracy
ADABOOST	No Scaling	Gmean	0.2418(0.0441)	0.3874(0.0617)	0.6284(0.0249)	0.9848(0.0017)
			0.1635(0.112)	0.1635(0.112)	0.5808(0.0603)	0.9829(0.0034)
ADACOST	No Scaling	F2	0.392(0.0052)	0.9639(0.0007)	0.9646(0.0007)	0.9297(0.0013)
			0.3887(0.0175)	0.9639(0.0027)	0.9646(0.0026)	0.9297(0.0052)
CATBOOST	No Scaling	Gmean	0.14(0.0453)	0.2063(0.0647)	0.5628(0.0206)	0.9912(0.0003)
			0.1312(0.1356)	0.2579(0.244)	0.5564(0.0599)	0.9906(0.0014)
Decision Tree	No Scaling	F2	0.2359(0.0683)	0.3766(0.098)	0.6231(0.038)	0.9852(0.0009)
			0.1796(0.1361)	0.3869(0.245)	0.5952(0.0799)	0.5952(0.0799)
Cost Sensitive Decision Tree	No Scaling	PR-Recall	0.2802(0.0296)	0.421(0.0411)	0.6455(0.0181)	0.9859(0.0018)
			0.1894(0.1854)	0.3765(0.2659)	0.5963(0.1078)	0.9855(0.0029)
Decision Tree - HELLINGER Distance	No Scaling	PR-Recall	0.223(0.0295)	0.3599(0.0377)	0.616(0.0196)	0.9853(0.0005)
			0.2133(0.2037)	0.3671(0.3387)	0.6103(0.1078)	0.6103(0.1078)
Decision Tree - SMOTENC	No Scaling	Gmean	0.2559(0.0505)	0.4195(0.0547)	0.6407(0.0278)	0.6407(0.0278)
			0.1833(0.15)	0.406(0.2733)	0.6076(0.1056)	0.9799(0.0038)
Decision Tree - SMOTETOMEK	No Scaling	Gmean	0.3593(0.0308)	0.7261(0.0488)	0.788(0.0272)	0.9617(0.0061)
			0.4443(0.1128)	0.8257(0.1136)	0.8419(0.1)	0.9672(0.0111)
KNN	POWER	F2	0.2211(0.0786)	0.3565(0.1136)	0.6185(0.0437)	0.9847(0.0011)
			0.2128(0.1406)	0.4184(0.2532)	0.6098(0.0808)	0.9842(0.0034)
Logistic Regression	No Scaling	PR-Recall	0.0612(0.0637)	0.0897(0.091)	0.5272(0.0288)	0.9904(0.0008)
			0.0333(0.0745)	0.0755(0.1689)	0.5138(0.0319)	0.9903(0.0007)
Cost Sensitive Logistic Regression	NORM	F2	0.4568(0.0097)	0.9127(0.054)	0.921(0.0446)	0.9559(0.0085)
			0.4594(0.0376)	0.9335(0.0386)	0.9344(0.038)	0.9542(0.006)
Logistic Regression - SMOTENC	NORM	F2	0.4335(0.0317)	0.8151(0.0403)	0.8514(0.0247)	0.9641(0.0054)
			0.4518(0.0767)	0.8742(0.0903)	0.8819(0.0802)	0.9623(0.0055)

Table A.425: Summary Results For CNS - Part 1

			Training - Testing			
Model	Best Scaling	Best Opt. For	F2	Gmean	AUC ROC	Accuracy
Logistic Regression - SMOTETOMEK	NORM	F2	0.4607(0.0099)	0.9132(0.0196)	0.9204(0.0163)	0.9567(0.0024)
			0.4381(0.0416)	0.9038(0.0527)	0.9065(0.0496)	0.9549(0.0033)
SVM	POWER	F2	0.0826(0.0471)	0.1251(0.0714)	0.5376(0.0216)	0.9902(0.0005)
			0.0333(0.0745)	0.0755(0.1689)	0.5135(0.032)	0.9898(0.0018)
Cost Sensitive SVM	NORM	F2	0.4641(0.0186)	0.8813(0.0662)	0.896(0.0534)	0.962(0.0067)
			0.4094(0.0475)	0.8415(0.0837)	0.8523(0.0753)	0.9598(0.0047)
Random Forest	No Scaling	PR-Recall	0.0771(0.042)	0.1098(0.059)	0.5338(0.0186)	0.9915(0.0003)
			0.069(0.0944)	0.1512(0.207)	0.5283(0.0394)	0.9911(0.0013)
Cost Sensitive Random Forest	No Scaling	Gmean	0.3225(0.037)	0.547(0.0482)	0.698(0.0221)	0.9797(0.0023)
			0.4747(0.0549)	0.7845(0.0482)	0.8053(0.0389)	0.8053(0.0389)
SVM SMOTENC	NORM	F2	0.3275(0.0431)	0.6301(0.0662)	0.7411(0.0343)	0.9672(0.0053)
			0.3522(0.0942)	0.7388(0.0933)	0.7695(0.0709)	0.9641(0.0089)
SVM SMOTETOMEK	No Scaling	PR-Recall	0.0242(0.0495)	0.0482(0.0975)	0.5098(0.028)	0.981(0.008)
			0.0(0.0)	0.0(0.0)	0.4945(0.0089)	0.9801(0.0177)
XGBOOST	No Scaling	PR-Recall	0.2241(0.04)	0.3247(0.0566)	0.601(0.0189)	0.9915(0.0004)
			0.2583(0.1442)	0.4643(0.1262)	0.6134(0.0645)	0.9913(0.0029)
Cost Sensitive XGBOOST	No Scaling	F2	0.207(0.0497)	0.2934(0.0705)	0.5969(0.0234)	0.9911(0.0003)
			0.2302(0.0895)	0.4403(0.0858)	0.5994(0.0394)	0.9916(0.0019)
XGBOOST - SMOTENC	No Scaling	F2	0.2746(0.0817)	0.4517(0.1116)	0.6585(0.0452)	0.9812(0.0025)
			0.3087(0.0678)	0.6021(0.0656)	0.6783(0.0397)	0.9796(0.0024)
XGBOOST - SMOTETOMEK	No Scaling	PR-Recall	0.2541(0.0346)	0.3881(0.05)	0.6284(0.0144)	0.6284(0.0144
			0.3723(0.1298)	0.6012(0.1002)	0.6829(0.0643)	0.9888(0.0041)
Neural Network	POWER	F2	0.0604(0.0298)	0.0951(0.0441)	0.5283(0.0144)	0.9883(0.0007)
			0.0303(0.0678)	0.0754(0.1686)	0.5125(0.0315)	0.9878(0.0019)

Table A.426: Summary Results For CNS - Part 2

			Training - Testing			
Model	Best Scaling	Best Opt. For	F2	Gmean	AUC ROC	Accuracy
ADABOOST	No Scaling	Gmean	0.2655(0.0683)	0.2655(0.0683)	0.6376(0.0348)	0.9836(0.0016)
			0.3211(0.0839)	0.5652(0.0623)	0.6584(0.036)	0.9852(0.0046)
ADACOST	No Scaling	PR-Recall	0.4275(0.0065)	0.9649(0.0008)	0.9655(0.0008)	0.9317(0.0016)
			0.4247(0.023)	0.9649(0.0034)	0.9655(0.0033)	0.9317(0.0065)
CATBOOST	No Scaling	Gmean	0.1231(0.0635)	0.1887(0.0894)	0.556(0.0306)	0.9892(0.0002)
			0.0573(0.0787)	0.1412(0.1934)	0.5239(0.0341)	0.9882(0.0019)
Decision Tree	No Scaling	Gmean	0.2814(0.0758)	0.4442(0.0999)	0.6461(0.0392)	0.9842(0.0017)
			0.3662(0.0874)	0.6065(0.0732)	0.6837(0.0442)	0.9862(0.004)
Cost Sensitive Decision Tree	No Scaling	Gmean	0.3056(0.0432)	0.489(0.0616)	0.6614(0.0241)	0.9839(0.0013)
			0.2991(0.2066)	0.4813(0.2883)	0.6462(0.1059)	0.9854(0.0051)
Decision Tree - HELLINGER Distance	No Scaling	Gmean	0.3055(0.0771)	0.4704(0.1176)	0.6575(0.0426)	0.985(0.001)
			0.3144(0.2058)	0.5033(0.2941)	0.6586(0.1048)	0.9854(0.0045)
Decision Tree - SMOTENC	No Scaling	Gmean	0.3458(0.053)	0.5293(0.0544)	0.6825(0.0261)	0.9844(0.0019)
			0.3472(0.2681)	0.57(0.242)	0.6822(0.1482)	0.9832(0.0069)
Decision Tree - SMOTETOMEK	No Scaling	PR-Recall	0.5565(0.0519)	0.7944(0.0336)	0.8282(0.025)	0.9831(0.0029)
			0.5369(0.1641)	0.7797(0.1094)	0.8062(0.0899)	0.9837(0.0067)
KNN	POWER	F2	0.3614(0.1006)	0.5473(0.121)	0.6849(0.054)	0.9864(0.0011)
			0.378(0.1658)	0.5966(0.1454)	0.684(0.0789)	0.9867(0.0065)
Logistic Regression	No Scaling	F2	0.189(0.0431)	0.2813(0.0672)	0.5873(0.0204)	0.9891(0.0008)
			0.2496(0.1151)	0.4633(0.1108)	0.611(0.053)	0.9892(0.0033)
Cost Sensitive Logistic Regression	NORM	F2	0.592(0.0366)	0.8735(0.0318)	0.8936(0.0221)	0.9784(0.0018)
			0.592(0.113)	0.8834(0.0655)	0.8897(0.0578)	0.9777(0.0074)
Logistic Regression - SMOTENC	POWER	F2	0.5866(0.0529)	0.8332(0.0461)	0.8625(0.0292)	0.9819(0.002)
			0.616(0.1216)	0.8709(0.0762)	0.8796(0.0696)	0.9822(0.0056)

Table A.427: Summary Results For GU - Part 1

			Training - Testing			
Model	Best Scaling	Best Opt. For	F2	Gmean	AUC ROC	Accuracy
Logistic Regression - SMOTETOMEK	POWER	F2	0.5671(0.0572)	0.8821(0.0354)	0.8972(0.027)	0.9739(0.0041)
			0.5513(0.1239)	0.8801(0.08)	0.8871(0.0732)	0.9724(0.0095)
SVM	STD	F2	0.1807(0.0729)	0.2665(0.0984)	0.5838(0.0354)	0.9893(0.0004)
			0.2766(0.2127)	0.4343(0.2762)	0.6239(0.0994)	0.9902(0.0035)
Cost Sensitive SVM	NORM	F2	0.5937(0.0416)	0.8733(0.0291)	0.8866(0.0235)	0.979(0.0018)
			0.5744(0.1011)	0.8567(0.052)	0.8656(0.0457)	0.9789(0.0076)
Random Forest	No Scaling	F2	0.0473(0.044)	0.0724(0.0657)	0.5204(0.0191)	0.9895(0.0007)
			0.0606(0.083)	0.1414(0.1936)	0.5247(0.0345)	0.99(0.0013)
Cost Sensitive Random Forest	No Scaling	F2	0.5563(0.0365)	0.8102(0.0266)	0.8396(0.0167)	0.9814(0.0023)
			0.5011(0.1172)	0.7666(0.0682)	0.7925(0.0545)	0.9812(0.0077)
SVM SMOTENC	POWER	F2	0.5194(0.0259)	0.7354(0.0288)	0.797(0.0173)	0.9847(0.0013)
			0.4998(0.135)	0.7487(0.0971)	0.7813(0.0725)	0.9834(0.006)
SVM SMOTETOMEK	No Scaling	F2	0.1105(0.0233)	0.3971(0.1022)	0.5958(0.0334)	0.9118(0.0074)
			0.1199(0.0606)	0.5116(0.1286)	0.6074(0.0736)	0.9087(0.0126)
XGBOOST	No Scaling	F2	0.1459(0.0752)	0.2244(0.1142)	0.5674(0.0357)	0.9878(0.0008)
			0.2535(0.1157)	0.4634(0.1108)	0.6114(0.0529)	0.99(0.0029)
Cost Sensitive XGBOOST	No Scaling	Gmean	0.1653(0.0841)	0.2535(0.1171)	0.5759(0.039)	0.9889(0.0013)
			0.1423(0.1451)	0.2704(0.2541)	0.5612(0.063)	0.9887(0.0031)
XGBOOST - SMOTENC	No Scaling	Gmean	0.4663(0.0772)	0.6722(0.0729)	0.756(0.041)	0.9851(0.0022)
			0.477(0.1525)	0.714(0.1116)	0.7573(0.0827)	0.9849(0.0068)
XGBOOST - SMOTETOMEK	No Scaling	F2	0.4(0.0666)	0.5931(0.0619)	0.7151(0.0379)	0.9853(0.0017)
			0.4091(0.1196)	0.6599(0.1102)	0.6599(0.1102)	0.9842(0.0046)
Neural Network	POWER	Gmean	0.2297(0.0814)	0.3474(0.1127)	0.6131(0.0417)	0.9874(0.0006)
			0.2057(0.0973)	0.4333(0.1169)	0.5972(0.0547)	0.9864(0.0032)

Table A.428: Summary Results For GU - Part 2

			Training - Testing			
Model	Best Scaling	Best Opt. For	F2	Gmean	AUC ROC	Accuracy
ADABOOST	No Scaling	Gmean	0.2011(0.0367)	0.3647(0.0512)	0.6041(0.0186)	0.9795(0.0024)
			0.1062(0.1046)	0.2537(0.2382)	0.5497(0.0554)	0.9784(0.0034)
ADACOST	No Scaling	F2	0.4608(0.0069)	0.9655(0.0009)	0.9661(0.0009)	0.933(0.0017)
			0.4585(0.0256)	0.9655(0.0036)	0.9661(0.0035)	0.933(0.0069)
CATBOOST	No Scaling	PR-Recall	0.0825(0.0424)	0.1318(0.0644)	0.5359(0.0187)	0.988(0.0006)
			0.0263(0.0588)	0.0666(0.149)	0.5102(0.025)	0.9872(0.001)
Decision Tree	No Scaling	F2	0.2138(0.0437)	0.358(0.0576)	0.609(0.0228)	0.9804(0.0015)
			0.1341(0.0496)	0.3592(0.0615)	0.5618(0.0251)	0.9807(0.003)
Cost Sensitive Decision Tree	No Scaling	Gmean	0.246(0.0302)	0.4164(0.0467)	0.6279(0.0203)	0.9805(0.0013)
			0.2154(0.0584)	0.4626(0.0853)	0.6063(0.0371)	0.9817(0.0041)
Decision Tree - HELLINGER Distance	No Scaling	PR-Recall	0.1913(0.0274)	0.3284(0.0378)	0.596(0.0146)	0.9814(0.001)
			0.2085(0.111)	0.4357(0.105)	0.5958(0.0488)	0.9827(0.0058)
Decision Tree - SMOTENC	No Scaling	Gmean	0.2734(0.0199)	0.4657(0.035)	0.6478(0.0114)	0.9785(0.0007)
			0.2671(0.0915)	0.5429(0.1172)	0.6478(0.0699)	0.9769(0.0044)
Decision Tree - SMOTETOMEK	No Scaling	F2	0.4396(0.0394)	0.7898(0.0533)	0.8224(0.0382)	0.9626(0.0047)
			0.3984(0.1908)	0.7599(0.229)	0.8042(0.1699)	0.9603(0.0102)
KNN	POWER	F2	0.2175(0.087)	0.353(0.1097)	0.6088(0.0461)	0.9835(0.0015)
			0.1756(0.2473)	0.2637(0.3658)	0.5858(0.1275)	0.9847(0.003)
Logistic Regression	POWER	F2	0.1457(0.0632)	0.2261(0.0947)	0.5651(0.0287)	0.9886(0.0006)
			0.1316(0.0025)	0.3331(0.0001)	0.5549(0.0004)	0.9887(0.0009)
Cost Sensitive Logistic Regression	NORM	F2	0.4989(0.0222)	0.9304(0.0499)	0.936(0.0425)	0.9494(0.0135)
			0.4447(0.1182)	0.8899(0.1795)	0.9069(0.1425)	0.9463(0.0123)
Logistic Regression - SMOTENC	NORM	F2	0.4762(0.0294)	0.8352(0.0286)	0.8618(0.0164)	0.9613(0.0039)
			0.4733(0.0917)	0.8473(0.1102)	0.8598(0.0889)	0.9616(0.0056)

Table A.429: Summary Results For MUSC - Part 1

			Training - Testing			
Model	Best Scaling	Best Opt. For	F2	Gmean	AUC ROC	Accuracy
Logistic Regression - SMOTETOMEK	NORM	F2	0.5034(0.0203)	0.9166(0.0158)	0.9225(0.0141)	0.9538(0.0054)
			0.4587(0.0599)	0.8811(0.092)	0.8879(0.0807)	0.9521(0.006)
SVM	POWER	F2	0.1877(0.0604)	0.1877(0.0604)	0.5861(0.0291)	0.9868(0.0008)
			0.2226(0.0546)	0.4431(0.0617)	0.5983(0.025)	0.9877(0.001)
Cost Sensitive SVM	NORM	F2	0.4917(0.035)	0.9118(0.0378)	0.9185(0.0316)	0.9522(0.0066)
			0.4612(0.094)	0.8812(0.095)	0.8878(0.0841)	0.9518(0.0068)
Random Forest	No Scaling	Gmean	0.0121(0.0122)	0.0205(0.0216)	0.505(0.0055)	0.9883(0.0004)
			0.0263(0.0588)	0.0666(0.149)	0.5107(0.0247)	0.9882(0.0007)
Cost Sensitive Random Forest	No Scaling	F2	0.433(0.0544)	0.7473(0.0554)	0.7929(0.0321)	0.9684(0.0034)
			0.3991(0.172)	0.744(0.1832)	0.7838(0.1446)	0.9634(0.0069)
SVM SMOTENC	NORM	F2	0.3746(0.06)	0.6712(0.078)	0.7502(0.0423)	0.9669(0.0069)
			0.4161(0.1573)	0.7395(0.1389)	0.7753(0.1077)	0.9684(0.0083)
SVM SMOTETOMEK	No Scaling	Gmean	0.0907(0.0407)	0.2145(0.108)	0.5494(0.0254)	0.9659(0.0132)
			0.0951(0.0696)	0.291(0.1732)	0.5443(0.0404)	0.9676(0.0143)
XGBOOST	No Scaling	F2	0.1643(0.0369)	0.2528(0.0493)	0.574(0.0176)	0.9884(0.0005)
			0.0777(0.071)	0.1998(0.1824)	0.5319(0.0309)	0.9867(0.0023)
Cost Sensitive XGBOOST	No Scaling	Gmean	0.1606(0.0449)	0.1606(0.0449)	0.5729(0.0222)	0.988(0.0004)
			0.0777(0.071)	0.1998(0.1824)	0.5319(0.0309)	0.9867(0.0021)
XGBOOST - SMOTENC	No Scaling	Gmean	0.3335(0.0654)	0.5674(0.0874)	0.693(0.0386)	0.9767(0.0026)
			0.2606(0.1196)	0.538(0.1408)	0.6464(0.0722)	0.9741(0.0045)
XGBOOST - SMOTETOMEK	No Scaling	F2	0.2648(0.0492)	0.4199(0.0694)	0.632(0.0274)	0.9837(0.0006)
			0.2226(0.0829)	0.4629(0.0866)	0.6067(0.0398)	0.9824(0.0032)
Neural Network	POWER	PR-Recall	0.1798(0.0574)	0.275(0.0778)	0.5854(0.0291)	0.987(0.0005)
			0.1877(0.1245)	0.3697(0.2237)	0.5866(0.0622)	0.9862(0.0025)

Table A.430: Summary Results For MUSC - Part 2

		Training / Testing		
CHD_CNS	**F2**	**Gmean**	**AUC ROC**	**Accuracy**
DT	0.7699(0.046)	0.8755(0.0233)	0.8877(0.0205)	0.9914(0.0022)
	0.7887(0.0936)	0.8957(0.0493)	0.9015(0.0446)	0.9908(0.004)
LR	0.6857(0.0344)	0.7954(0.0252)	0.8277(0.0185)	0.9926(0.0006)
	0.6991(0.1428)	0.8083(0.0973)	0.8304(0.0782)	0.9932(0.0028)
LR_CS	0.8351(0.0116)	0.9789(0.0039)	0.9795(0.0037)	0.9839(0.0013)
	0.8318(0.0361)	0.9771(0.0202)	0.9773(0.0199)	0.984(0.0038)
LR_SMOTENC	0.7917(0.0142)	0.9214(0.0118)	0.9282(0.0102)	0.9871(0.0015)
	0.8073(0.0638)	0.9399(0.0458)	0.9419(0.0431)	0.9866(0.0019)
LR_SMOTETOMEK	0.8454(0.012)	0.977(0.0069)	0.9779(0.0067)	0.9856(0.0014)
	0.8407(0.0408)	0.9776(0.0201)	0.9778(0.0199)	0.985(0.0045)
SVM_CS	0.8055(0.031)	0.9398(0.0155)	0.9435(0.0143)	0.9861(0.0021)
	0.8205(0.0802)	0.9489(0.0343)	0.95(0.0332)	0.9866(0.0058)

Table A.431: Summary Results for CHD - CNS

		Training / Testing		
CHD_GU	**F2**	**Gmean**	**AUC ROC**	**Accuracy**
DT	0.8391(0.0214)	0.9157(0.0125)	0.9225(0.0116)	0.9933(0.001)
	0.8118(0.1329)	0.8972(0.0829)	0.9048(0.0727)	0.9927(0.0047)
LR	0.7608(0.0348)	0.8524(0.0228)	0.8686(0.0195)	0.9932(0.0009)
	0.763(0.1162)	0.8528(0.0802)	0.8661(0.0663)	0.9937(0.0025)
LR_CS	0.8483(0.0197)	0.9757(0.0116)	0.9763(0.0112)	0.9846(0.0011)
	0.8406(0.0224)	0.972(0.029)	0.9725(0.0282)	0.9845(0.0023)
LR_SMOTENC	0.8442(0.0147)	0.9425(0.0173)	0.9463(0.0154)	0.9897(0.001)
	0.8421(0.0356)	0.9411(0.0191)	0.9425(0.0181)	0.9898(0.0031)
LR_SMOTETOMEK	0.8395(0.0049)	0.9589(0.0031)	0.9608(0.0031)	0.9861(0.001)
	0.8374(0.0214)	0.9596(0.0227)	0.9603(0.022)	0.9861(0.0034)
SVM_CS	0.8098(0.0321)	0.9312(0.0174)	0.936(0.0162)	0.9866(0.002)
	0.8089(0.0554)	0.9322(0.0376)	0.9344(0.0359)	0.9866(0.0044)

Table A.432: Summary Results for CHD - GU

		Training / Testing		
CHD_MUSC	**F2**	**Gmean**	**AUC ROC**	**Accuracy**
DT	0.7587(0.0393)	0.8671(0.0222)	0.8792(0.0194)	0.9901(0.0018)
	0.7548(0.1503)	0.8597(0.1042)	0.8734(0.0841)	0.9911(0.0041)
LR	0.6627(0.0427)	0.7843(0.0297)	0.8162(0.0221)	0.9905(0.0012)
	0.6721(0.1656)	0.7887(0.1199)	0.8165(0.0857)	0.9916(0.0037)
LR_CS	0.8363(0.0214)	0.9718(0.0115)	0.9729(0.0109)	0.9831(0.0015)
	0.8506(0.0488)	0.9788(0.0196)	0.9789(0.0194)	0.9843(0.0038)
LR_SMOTENC	0.8248(0.0233)	0.948(0.0165)	0.9508(0.0153)	0.9855(0.0011)
	0.8246(0.0831)	0.9449(0.0589)	0.9473(0.055)	0.9864(0.004)
LR_SMOTETOMEK	0.8353(0.0113)	0.9661(0.008)	0.9675(0.0078)	0.9839(0.0008)
	0.835(0.0511)	0.9658(0.0289)	0.9663(0.0283)	0.9843(0.0029)
SVM_CS	0.7818(0.0257)	0.9172(0.0156)	0.9236(0.015)	0.9845(0.0012)
	0.8205(0.0524)	0.9462(0.0386)	0.9478(0.0366)	0.9856(0.0033)

Table A.433: Summary Results for CHD - MUSC

		Training / Testing		
CNS_GU	**F2**	**Gmean**	**AUC ROC**	**Accuracy**
DT	0.7586(0.0413)	0.8641(0.0293)	0.8813(0.0216)	0.9903(0.0014)
	0.8091(0.0794)	0.9056(0.0494)	0.9105(0.0449)	0.9911(0.0036)
LR	0.7638(0.0183)	0.8523(0.0108)	0.8693(0.0093)	0.9935(0.0009)
	0.7912(0.1266)	0.8684(0.0833)	0.8797(0.0692)	0.9948(0.0031)
LR_CS	0.8743(0.0081)	0.99(0.0028)	0.9901(0.0027)	0.9859(0.0011)
	0.8616(0.0458)	0.9797(0.0193)	0.9799(0.0191)	0.9861(0.0038)
LR_SMOTENC	0.8447(0.013)	0.9684(0.0108)	0.9702(0.0094)	0.9854(0.0011)
	0.8499(0.0837)	0.966(0.0465)	0.9671(0.0444)	0.9866(0.0047)
LR_SMOTETOMEK	0.8752(0.0139)	0.9827(0.0056)	0.9832(0.0054)	0.9873(0.0015)
	0.8574(0.0552)	0.9731(0.031)	0.9736(0.0303)	0.9866(0.0033)
SVM_CS	0.8238(0.0333)	0.9397(0.016)	0.9428(0.0151)	0.9871(0.0022)
	0.8646(0.0763)	0.9672(0.0388)	0.968(0.0378)	0.9885(0.0044)

Table A.434: Summary Results for CNS - GU

		Training / Testing		
CNS_MUSC	**F2**	**Gmean**	**AUC ROC**	**Accuracy**
DT	0.6956(0.0266)	0.8242(0.0164)	0.846(0.0127)	0.9884(0.001)
	0.6902(0.0978)	0.8173(0.0721)	0.8355(0.0598)	0.9895(0.0019)
LR	0.7037(0.0165)	0.8145(0.0118)	0.8385(0.0098)	0.9914(0.0002)
	0.7171(0.0899)	0.8274(0.0622)	0.8435(0.051)	0.9916(0.0022)
LR_CS	0.8526(0.0154)	0.9787(0.0069)	0.9792(0.0066)	0.9843(0.0011)
	0.8468(0.0531)	0.9726(0.0306)	0.9731(0.0299)	0.9848(0.0037)
LR_SMOTENC	0.8331(0.0151)	0.9557(0.009)	0.9577(0.0084)	0.9854(0.0011)
	0.8398(0.057)	0.966(0.0289)	0.9665(0.0282)	0.9848(0.0048)
LR_SMOTETOMEK	0.8466(0.0077)	0.9737(0.0019)	0.9744(0.0018)	0.9843(0.0009)
	0.8562(0.0523)	0.9733(0.0308)	0.9737(0.03)	0.9861(0.0027)
SVM_CS	0.81(0.0243)	0.9317(0.0138)	0.9362(0.0127)	0.9862(0.0013)
	0.8324(0.0679)	0.9466(0.0402)	0.9481(0.0383)	0.9872(0.0042)

Table A.435: Summary Results for CNS - MUSC

		Training / Testing		
GU_MUSC	**F2**	**Gmean**	**AUC ROC**	**Accuracy**
DT	0.7458(0.0305)	0.8577(0.0188)	0.873(0.0157)	0.9892(0.0014)
	0.7712(0.0406)	0.8746(0.0256)	0.8822(0.0224)	0.9903(0.0033)
LR	0.7321(0.0236)	0.8335(0.0167)	0.8531(0.0136)	0.9918(0.0006)
	0.7056(0.0859)	0.8165(0.053)	0.8341(0.0435)	0.9914(0.003)
LR_CS	0.8472(0.0109)	0.9692(0.0068)	0.9703(0.0065)	0.9842(0.0007)
	0.8487(0.0363)	0.9731(0.0276)	0.9735(0.0269)	0.9841(0.0031)
LR_SMOTENC	0.8387(0.015)	0.9522(0.0104)	0.9548(0.0098)	0.9861(0.0009)
	0.8589(0.0283)	0.963(0.0294)	0.9637(0.0286)	0.9875(0.0034)
LR_SMOTETOMEK	0.8369(0.0196)	0.959(0.0112)	0.9607(0.0106)	0.9847(0.0011)
	0.8653(0.043)	0.9743(0.0282)	0.9747(0.0275)	0.9864(0.0027)
SVM_CS	0.8244(0.0297)	0.9361(0.0197)	0.9401(0.0179)	0.9868(0.001)
	0.7965(0.0812)	0.9223(0.0595)	0.926(0.0554)	0.9854(0.0021)

Table A.436: Summary Results for GU - MUSC

Appendix B

Probability Prediction Results

In this chapter, we report results for both testing and validation for the probability predictions. We report Brier Skill Score (BSS), AUC ROC and AUC PR results.
We are reporting two values for each score; average and standard deviation in the following form: average(standard deviation), where the average and standard deviation are reported across validation and testing results.

B.1. Probability Prediction For All Defects. Probability Prediction For All Defects

GNB	None	NORM	POWER	STD	TRD
BSS	0.4903(0.0478)	0.4467(0.0269)	0.4882(0.0429)	0.4555(0.025)	0.4514(0.0303)
AUC ROC	0.9089(0.0185)	0.9378(0.001)	0.9862(0.0019)	0.9416(0.0052)	0.9392(0.003)
AUC PR	0.7879(0.019)	0.7946(0.0073)	0.8386(0.0097)	0.8004(0.0067)	0.7957(0.009)

Table B.1: GNB - Training

GNB	None	NORM	POWER	STD	TRD
BSS	0.4673(0.1544)	0.4276(0.1029)	0.4786(0.1754)	0.4332(0.0918)	-0.0485(0.0)
AUC ROC	0.9052(0.0147)	0.9326(0.019)	0.9826(0.0143)	0.9353(0.0173)	0.5(0.0)
AUC PR	0.7765(0.0503)	0.784(0.0271)	0.8329(0.0452)	0.7873(0.0201)	0.5235(0.0)

Table B.2: GNB - Testing

GNB - SMOTENC	None	NORM	POWER	STD	TRD
BSS	0.4025(0.0338)	0.4068(0.0338)	0.4758(0.044)	0.4071(0.0384)	0.4184(0.0343)
AUC ROC	0.8834(0.0046)	0.9018(0.0046)	0.9808(0.0045)	0.9041(0.0068)	0.9111(0.0091)
AUC PR	0.7457(0.0128)	0.759(0.0085)	0.8318(0.0107)	0.7611(0.0129)	0.7692(0.0133)

Table B.3: GNB - SMOTENC - Training

GNB - SMOTENC	None	NORM	POWER	STD	TRD
BSS	0.3709(0.1209)	0.4106(0.1105)	0.4729(0.166)	0.3936(0.1308)	-0.0485(0.0)
AUC ROC	0.8824(0.0307)	0.909(0.025)	0.9799(0.013)	0.9061(0.0296)	0.5(0.0)
AUC PR	0.7338(0.0503)	0.7629(0.0398)	0.8299(0.0403)	0.7567(0.0491)	0.5235(0.0)

Table B.4: GNB - SMOTENC - Testing

GPC	None	NORM	POWER	STD	TRD
BSS	-0.5348(0.0528)	0.4821(0.0411)	0.2539(0.0348)	0.2699(0.0477)	0.1259(0.0471)
AUC ROC	0.5566(0.0134)	0.7768(0.0195)	0.7642(0.0089)	0.7685(0.0102)	0.5859(0.0215)
AUC PR	0.1899(0.0288)	0.7493(0.0219)	0.6268(0.0155)	0.6371(0.0242)	0.5928(0.031)

Table B.5: GPC - Training

GPC	None	NORM	POWER	STD	TRD
BSS	-0.5529(0.1583)	0.4673(0.1273)	0.2576(0.1192)	0.2916(0.0633)	-6.8324(8.4781)
AUC ROC	0.5472(0.0141)	0.7716(0.0552)	0.7616(0.0341)	0.7726(0.0302)	0.6819(0.1765)
AUC PR	0.1697(0.0293)	0.7395(0.0719)	0.6241(0.0647)	0.6435(0.0371)	0.5287(0.0215)

Table B.6: GPC - Testing

GPC - SMOTENC	None	NORM	POWER	STD	TRD
BSS	-1.0061(0.0575)	0.5491(0.017)	0.1634(0.0399)	0.2421(0.0744)	0.4936(0.0449)
AUC ROC	0.5896(0.0082)	0.9705(0.0047)	0.7977(0.0115)	0.8148(0.0133)	0.9411(0.0095)
AUC PR	0.222(0.0129)	0.8408(0.0079)	0.6213(0.0194)	0.6555(0.0295)	0.8084(0.017)

Table B.7: GPC - SMOTENC - Training

GPC - SMOTENC	None	NORM	POWER	STD	TRD
BSS	-1.0006(0.2866)	0.5353(0.0254)	0.1725(0.0927)	0.2009(0.1068)	-10.6412(7.1953)
AUC ROC	0.5983(0.0149)	0.9711(0.0211)	0.8007(0.0181)	0.8142(0.0191)	0.6254(0.1156)
AUC PR	0.2368(0.0245)	0.8345(0.0112)	0.6229(0.028)	0.6405(0.0384)	0.5281(0.011)

Table B.8: GPC - SMOTENC - Testing

KNN - CALIBRATED	None	NORM	POWER	STD	TRD
BSS	-0.0202(0.018)	0.5201(0.0176)	0.3283(0.033)	0.2897(0.0383)	0.5302(0.0412)
AUC ROC	0.5305(0.0092)	0.8443(0.0112)	0.731(0.0177)	0.7102(0.0157)	0.8265(0.0148)
AUC PR	0.4257(0.0398)	0.7684(0.0094)	0.6558(0.0231)	0.635(0.0214)	0.7715(0.0214)

Table B.9: KNN - CALIBRATED - Training

KNN - CALIBRATED	None	NORM	POWER	STD	TRD
BSS	-0.0265(0.0734)	0.4952(0.1123)	0.3081(0.0712)	0.2457(0.0974)	-0.9113(2.216)
AUC ROC	0.5314(0.0265)	0.8365(0.063)	0.7217(0.0113)	0.6971(0.0356)	0.6416(0.1486)
AUC PR	0.3928(0.1567)	0.7525(0.0645)	0.6474(0.0527)	0.6083(0.0743)	0.5529(0.0292)

Table B.10: KNN - CALIBRATED - Testing

LDA	None	NORM	POWER	STD	TRD
BSS	0.3989(0.0246)	0.4109(0.0229)	0.6403(0.0239)	0.4089(0.0345)	0.4074(0.0289)
AUC ROC	0.9679(0.0068)	0.9704(0.0068)	0.8369(0.0075)	0.971(0.0054)	0.9711(0.0062)
AUC PR	0.8099(0.0059)	0.813(0.004)	0.8338(0.0122)	0.8142(0.006)	0.8148(0.0065)

Table B.11: LDA - Training

LDA	None	NORM	POWER	STD	TRD
BSS	0.4162(0.0841)	0.4162(0.0841)	0.6373(0.101)	0.4162(0.0841)	0.5296(0.0791)
AUC ROC	0.976(0.011)	0.976(0.011)	0.8322(0.0455)	0.976(0.011)	0.8296(0.0443)
AUC PR	0.8137(0.0192)	0.8137(0.0192)	0.8322(0.0479)	0.8137(0.0192)	0.7715(0.0454)

Table B.12: LDA - Testing

LDA - SMOTENC	None	NORM	POWER	STD	TRD
BSS	0.2829(0.0263)	0.2852(0.0235)	0.2852(0.0225)	0.2833(0.0244)	0.2826(0.0246)
AUC ROC	0.9829(0.0011)	0.9832(0.0006)	0.9832(0.0005)	0.9829(0.0007)	0.9831(0.0006)
AUC PR	0.8008(0.0045)	0.8015(0.0048)	0.8017(0.0031)	0.8019(0.0047)	0.8017(0.004)

Table B.13: LDA - SMOTENC - Training

LDA - SMOTENC	None	NORM	POWER	STD	TRD
BSS	0.2859(0.0905)	0.2859(0.0905)	0.2859(0.0905)	0.2859(0.0905)	0.2859(0.0905)
AUC ROC	0.9832(0.0021)	0.9832(0.0021)	0.9832(0.0021)	0.9832(0.0021)	0.9832(0.0021)
AUC PR	0.7981(0.0155)	0.7981(0.0155)	0.7981(0.0155)	0.7981(0.0155)	0.7981(0.0155)

Table B.14: LDA - SMOTENC - Testing

LR	None	NORM	POWER	STD	TRD
BSS	0.5942(0.0374)	0.7353(0.0171)	0.7376(0.028)	0.729(0.0328)	0.7384(0.0277)
AUC ROC	0.8937(0.0177)	0.9404(0.0072)	0.9368(0.013)	0.9401(0.0097)	0.9416(0.01)
AUC PR	0.8154(0.0192)	0.8826(0.0085)	0.8822(0.0134)	0.8796(0.0144)	0.8844(0.0129)

Table B.15: Logistic Regression - Training

LR	None	NORM	POWER	STD	TRD
BSS	0.6259(0.0507)	0.711(0.0733)	0.7166(0.075)	0.711(0.0836)	0.5353(0.043)
AUC ROC	0.8987(0.0136)	0.9341(0.0183)	0.924(0.0282)	0.9341(0.0213)	0.858(0.091)
AUC PR	0.8254(0.0203)	0.8679(0.0315)	0.8671(0.0371)	0.8673(0.038)	0.7949(0.0328)

Table B.16: Logistic Regression - Testing

LR - CS	None	NORM	POWER	STD	TRD
BSS	0.6017(0.0349)	0.7281(0.0157)	0.7423(0.027)	0.7126(0.0327)	0.7337(0.0234)
AUC ROC	0.9114(0.0545)	0.94(0.007)	0.9457(0.0204)	0.949(0.0242)	0.9416(0.0074)
AUC PR	0.8296(0.0229)	0.8786(0.0081)	0.8871(0.0142)	0.8764(0.0176)	0.8811(0.0118)

Table B.17: Cost Sensitive Logistic Regression - Training

LR - CS	None	NORM	POWER	STD	TRD
BSS	0.5693(0.1225)	0.7166(0.0722)	0.7166(0.075)	0.7053(0.0887)	0.5466(0.0448)
AUC ROC	0.9128(0.0522)	0.9342(0.0182)	0.9368(0.015)	0.9417(0.0214)	0.8198(0.0592)
AUC PR	0.8142(0.0495)	0.8704(0.0306)	0.8718(0.0292)	0.8687(0.0371)	0.7885(0.0248)

Table B.18: Cost Sensitive Logistic Regression - Testing

MNB	MNB	MNB - SMOTENC
BSS	-5.6117(0.2959)	-5.5392(0.284)
AUC ROC	0.5498(0.0117)	0.5521(0.0162)
AUC PR	0.2363(0.012)	0.2375(0.0188)

Table B.19: MNB - Training

MNB	MNB	MNB - SMOTENC
BSS	-5.7604(0.4973)	-5.6414(0.6043)
AUC ROC	0.5427(0.0428)	0.5481(0.0518)
AUC PR	0.2316(0.0425)	0.2352(0.052)

Table B.20: MNB - Testing

QDA	None	NORM	POWER	STD	TRD
BSS	0.3083(0.0477)	0.3227(0.0501)	0.3267(0.0571)	0.3203(0.0481)	-8.2742(0.1449)
AUC ROC	0.808(0.0081)	0.8148(0.0089)	0.8157(0.0122)	0.8131(0.0084)	0.6131(0.0099)
AUC PR	0.6713(0.0199)	0.6799(0.0219)	0.6825(0.0255)	0.6782(0.0194)	0.3665(0.0084)

Table B.21: QDA - Training

QDA	None	NORM	POWER	STD	TRD
BSS	0.3086(0.1308)	0.3086(0.1308)	0.3029(0.1353)	0.3086(0.1308)	-0.0485(0.0)
AUC ROC	0.809(0.0295)	0.809(0.0295)	0.8063(0.0335)	0.809(0.0295)	0.5(0.0)
AUC PR	0.6678(0.0604)	0.6678(0.0604)	0.6641(0.0643)	0.6678(0.0604)	0.5235(0.0)

Table B.22: QDA - Testing

QDA - SMOTENC	None	NORM	POWER	STD	TRD
BSS	0.2981(0.0402)	0.2867(0.0521)	0.3112(0.0563)	0.2695(0.039)	0.3116(0.0543)
AUC ROC	0.7963(0.0061)	0.7792(0.0114)	0.8069(0.0165)	0.7668(0.014)	0.8073(0.0116)
AUC PR	0.6618(0.015)	0.6507(0.027)	0.6735(0.0292)	0.6409(0.021)	0.6731(0.0261)

Table B.23: QDA - SMOTENC - Training

QDA - SMOTENC	None	NORM	POWER	STD	TRD
BSS	0.2462(0.1229)	0.2632(0.1539)	0.2916(0.1472)	0.2405(0.0927)	-0.0485(0.0)
AUC ROC	0.7716(0.0481)	0.7797(0.0454)	0.8009(0.0431)	0.7612(0.0289)	0.5(0.0)
AUC PR	0.6238(0.0693)	0.6341(0.0796)	0.6565(0.0747)	0.6172(0.0474)	0.5235(0.0)

Table B.24: QDA - SMOTENC - Testing

SVM - CALIBRATED	None	NORM	POWER	STD	TRD
BSS	0.5106(0.0706)	0.6796(0.0118)	0.6921(0.0459)	0.6374(0.0384)	0.6825(0.0184)
AUC ROC	0.8102(0.0335)	0.9009(0.013)	0.8995(0.0159)	0.8818(0.0099)	0.9124(0.0056)
AUC PR	0.7628(0.0363)	0.8504(0.0057)	0.8556(0.022)	0.8289(0.0184)	0.8544(0.0081)

Table B.25: SVM - CALIBRATED - Training

SVM - CALIBRATED	None	NORM	POWER	STD	TRD
BSS	0.5123(0.0812)	0.6881(0.1186)	0.6654(0.0734)	0.603(0.0851)	0.1833(0.3198)
AUC ROC	0.8087(0.0265)	0.9027(0.0352)	0.8894(0.0215)	0.8673(0.0248)	0.6648(0.2323)
AUC PR	0.7636(0.0462)	0.8516(0.0569)	0.84(0.0349)	0.8086(0.0436)	0.635(0.1573)

Table B.26: SVM - CALIBRATED - Testing

	DT - CALIBRATED
BSS	0.7507(0.021)
AUC ROC	0.9194(0.0107)
AUC PR	0.8833(0.0107)

Table B.27: CALIBRATED Decision Tree - Training

	DT - CALIBRATED
BSS	0.6824(0.0677)
AUC ROC	0.9077(0.0224)
AUC PR	0.8507(0.0339)

Table B.28: CALIBRATED Decision Tree - Testing

B.2. Probability Prediction For CHD. Probability Prediction For CHD

GNB	None	NORM	POWER	STD	TRD
BSS	-5.463(0.2828)	-5.5622(0.1828)	-5.6787(0.2021)	-5.5904(0.2143)	-5.5623(0.1969)
AUC ROC	0.9515(0.0097)	0.9683(0.0043)	0.9699(0.0009)	0.9626(0.0071)	0.9704(0.0008)
AUC PR	0.5501(0.0118)	0.5676(0.0041)	0.5694(0.0018)	0.5617(0.009)	0.57(0.0022)

Table B.29: GNB - Training

GNB	None	NORM	POWER	STD	TRD
BSS	-5.4801(0.7148)	-5.4801(0.7148)	-5.5665(0.7552)	-5.4801(0.7148)	-5.5665(0.7552)
AUC ROC	0.9562(0.0299)	0.9703(0.0034)	0.9699(0.0035)	0.9703(0.0034)	0.9699(0.0035)
AUC PR	0.5521(0.0327)	0.5678(0.0061)	0.5671(0.0062)	0.5678(0.0061)	0.5671(0.0062)

Table B.30: GNB - Testing

GNB - SMOTENC	None	NORM	POWER	STD	TRD
BSS	-4.0987(0.5805)	-3.6423(0.7766)	-4.2076(0.5313)	-3.457(1.0072)	-4.5165(0.5259)
AUC ROC	0.8421(0.0249)	0.8019(0.0116)	0.8212(0.0258)	0.8091(0.0251)	0.8954(0.0275)
AUC PR	0.434(0.0284)	0.3955(0.0073)	0.4096(0.028)	0.406(0.0168)	0.4925(0.0336)

Table B.31: GNB - SMOTENC - Training

GNB - SMOTENC	None	NORM	POWER	STD	TRD
BSS	-4.6329(1.1934)	-3.9188(1.6289)	-4.4023(0.9306)	-3.1406(1.3114)	-0.0091(0.0006)
AUC ROC	0.8664(0.0506)	0.7903(0.0971)	0.8391(0.0396)	0.7654(0.1017)	0.5(0.0)
AUC PR	0.4568(0.0518)	0.3751(0.1009)	0.4271(0.0408)	0.3514(0.1075)	0.5046(0.0003)

Table B.32: GNB - SMOTENC - Testing

GPC	None	NORM	POWER	STD	TRD
BSS	-0.4808(0.1056)	-0.009(0.0002)	-0.2948(0.1794)	-0.1669(0.1307)	-0.009(0.0002)
AUC ROC	0.581(0.0334)	0.5(0.0)	0.5714(0.0337)	0.5967(0.0269)	0.5(0.0)
AUC PR	0.2333(0.0749)	0.5046(0.0001)	0.2912(0.0959)	0.3412(0.0621)	0.5046(0.0001)

Table B.33: GPC - Training

GPC	None	NORM	POWER	STD	TRD
BSS	-0.6831(0.2583)	-0.0091(0.0006)	-0.3913(0.3421)	-0.1785(0.4003)	-0.0091(0.0006)
AUC ROC	0.5111(0.0323)	0.5(0.0)	0.5814(0.1174)	0.57(0.1017)	0.5(0.0)
AUC PR	0.0387(0.0763)	0.5046(0.0003)	0.1947(0.2197)	0.217(0.2935)	0.5046(0.0003)

Table B.34: GPC - Testing

GPC - SMOTENC	None	NORM	POWER	STD	TRD
BSS	-2.58(0.268)	-2.9846(0.7939)	-2.134(0.6896)	-2.0656(0.5478)	-1.4426(0.3694)
AUC ROC	0.6062(0.0382)	0.7547(0.0196)	0.7007(0.0479)	0.7335(0.0453)	0.6685(0.0369)
AUC PR	0.1633(0.0528)	0.3471(0.0312)	0.3001(0.0751)	0.3335(0.0701)	0.2773(0.0648)

Table B.35: GPC - SMOTENC - Training

GPC - SMOTENC	None	NORM	POWER	STD	TRD
BSS	-2.6256(0.4063)	-2.809(1.1783)	-2.0634(0.3352)	-1.93(0.7267)	-0.0091(0.0006)
AUC ROC	0.5712(0.0935)	0.7794(0.0827)	0.6853(0.0954)	0.7266(0.1122)	0.5(0.0)
AUC PR	0.1149(0.1212)	0.3767(0.084)	0.267(0.116)	0.3259(0.1496)	0.5046(0.0003)

Table B.36: GPC - SMOTENC - Testing

LDA	None	NORM	POWER	STD	TRD
BSS	-2.2133(0.1932)	-2.1732(0.184)	-1.941(0.1383)	-2.2299(0.2289)	-2.2121(0.2097)
AUC ROC	0.8595(0.0269)	0.8568(0.0282)	0.8646(0.0282)	0.8567(0.0275)	0.8578(0.028)
AUC PR	0.4843(0.0394)	0.4801(0.0376)	0.4952(0.0374)	0.4796(0.037)	0.4806(0.037)

Table B.37: LDA - Training

LDA	None	NORM	POWER	STD	TRD
BSS	-2.1605(0.7335)	-2.1605(0.7335)	-1.8435(0.5434)	-2.1605(0.7335)	-2.5822(0.9976)
AUC ROC	0.8636(0.1132)	0.8636(0.1132)	0.865(0.1136)	0.8636(0.1132)	0.8475(0.0992)
AUC PR	0.4835(0.1354)	0.4835(0.1354)	0.4919(0.1396)	0.4835(0.1354)	0.4581(0.1227)

Table B.38: LDA - Testing

LDA - SMOTENC	None	NORM	POWER	STD	TRD
BSS	-5.4738(1.3505)	-5.0203(1.38)	-3.681(1.8581)	-4.7646(1.648)	-6.2038(0.5616)
AUC ROC	0.9257(0.0315)	0.9171(0.0343)	0.8857(0.0314)	0.9128(0.0362)	0.9553(0.0117)
AUC PR	0.523(0.0282)	0.5163(0.0282)	0.4987(0.0373)	0.515(0.0312)	0.5506(0.0119)

Table B.39: LDA - SMOTENC - Training

LDA - SMOTENC	None	NORM	POWER	STD	TRD
BSS	-5.6343(1.7292)	-5.1871(2.2447)	-3.1947(1.7138)	-4.38(2.0095)	-5.5187(2.5263)
AUC ROC	0.9328(0.0774)	0.9209(0.072)	0.8589(0.108)	0.882(0.1261)	0.9335(0.0751)
AUC PR	0.527(0.0767)	0.521(0.0568)	0.4645(0.1187)	0.4797(0.1312)	0.5323(0.0669)

Table B.40: LDA - SMOTENC - Testing

LR	None	NORM	POWER	STD	TRD
BSS	-0.0135(0.0098)	-0.009(0.0002)	-0.0381(0.0498)	-0.0504(0.0253)	-0.0235(0.0205)
AUC ROC	0.5(0.0)	0.5(0.0)	0.5059(0.0082)	0.5042(0.0072)	0.4999(0.0001)
AUC PR	0.4979(0.015)	0.5046(0.0001)	0.464(0.0631)	0.4456(0.0284)	0.4879(0.0205)

Table B.41: Logistic Regression - Training

LR	None	NORM	POWER	STD	TRD
BSS	-0.0091(0.0006)	-0.0091(0.0006)	-0.0956(0.1932)	-0.0668(0.0787)	-0.0091(0.0006)
AUC ROC	0.5(0.0)	0.5(0.0)	0.4996(0.0009)	0.4997(0.0004)	0.5(0.0)
AUC PR	0.5046(0.0003)	0.5046(0.0003)	0.4046(0.2237)	0.3046(0.274)	0.5046(0.0003)

Table B.42: Logistic Regression - Testing

LR - CS	None	NORM	POWER	STD	TRD
BSS	-0.009(0.0002)	-0.009(0.0002)	-0.0123(0.0244)	-0.0235(0.0167)	-0.009(0.0002)
AUC ROC	0.5(0.0)	0.5(0.0)	0.5077(0.0053)	0.4999(0.0001)	0.5(0.0)
AUC PR	0.5046(0.0001)	0.5046(0.0001)	0.4923(0.0266)	0.4879(0.0204)	0.5046(0.0001)

Table B.43: Cost Sensitive Logistic Regression - Training

LR - CS	None	NORM	POWER	STD	TRD
BSS	-0.0091(0.0006)	-0.0091(0.0006)	-0.0956(0.1932)	-0.0091(0.0006)	-0.0091(0.0006)
AUC ROC	0.5(0.0)	0.5(0.0)	0.4996(0.0009)	0.5(0.0)	0.5(0.0)
AUC PR	0.5046(0.0003)	0.5046(0.0003)	0.4046(0.2237)	0.5046(0.0003)	0.5046(0.0003)

Table B.44: Cost Sensitive Logistic Regression - Testing

MNB	MNB	MNG - SMOTENC
BSS	-5.2993(0.5993)	-9.8787(3.0044)
AUC ROC	0.5246(0.0074)	0.533(0.0139)
AUC PR	0.0602(0.0082)	0.0909(0.0189)

Table B.45: MNB - Training

MNB	MNB	MNG - SMOTENC
BSS	-4.8094(1.3577)	-6.7587(2.6378)
AUC ROC	0.5348(0.0564)	0.5544(0.0819)
AUC PR	0.0716(0.0688)	0.0991(0.1009)

Table B.46: MNB - Testing

QDA	None	NORM	POWER	STD	TRD
BSS	-4.399(1.4047)	-4.4976(1.3039)	-4.5215(1.0269)	-4.5208(0.995)	-7.3098(3.0148)
AUC ROC	0.8654(0.1397)	0.8781(0.1341)	0.8702(0.0993)	0.8778(0.0944)	0.7961(0.1174)
AUC PR	0.4932(0.0756)	0.4987(0.0879)	0.474(0.087)	0.4706(0.1033)	0.4444(0.0448)

Table B.47: QDA - Training

QDA	None	NORM	POWER	STD	TRD
BSS	-5.1666(0.8965)	-5.253(0.7736)	-5.2819(0.8319)	-5.1089(0.7664)	-0.9532(1.9727)
AUC ROC	0.9434(0.0387)	0.943(0.0392)	0.9571(0.029)	0.9437(0.0386)	0.5949(0.2128)
AUC PR	0.5399(0.0435)	0.5388(0.0447)	0.5541(0.0304)	0.5402(0.0436)	0.4192(0.2337)

Table B.48: QDA - Testing

QDA - SMOTENC	None	NORM	POWER	STD	TRD
BSS	-4.5962(0.4232)	-4.3731(0.5312)	-4.0636(0.5216)	-3.6887(0.595)	-4.26(0.5696)
AUC ROC	0.8818(0.0129)	0.8288(0.0144)	0.8193(0.0274)	0.8158(0.0177)	0.891(0.0194)
AUC PR	0.4749(0.0142)	0.4164(0.0152)	0.4083(0.0331)	0.4084(0.0212)	0.4884(0.0244)

Table B.49: QDA - SMOTENC - Training

QDA - SMOTENC	None	NORM	POWER	STD	TRD
BSS	-4.6293(1.0355)	-4.7624(1.102)	-4.2365(0.9956)	-3.0003(1.2078)	-0.1533(0.3221)
AUC ROC	0.8664(0.0509)	0.8376(0.0939)	0.8541(0.047)	0.8084(0.1223)	0.4994(0.0014)
AUC PR	0.4565(0.0522)	0.423(0.1038)	0.445(0.0467)	0.4035(0.1365)	0.4046(0.2237)

Table B.50: QDA - SMOTENC - Testing

SVM - CALIBRATED	None	NORM	POWER	STD	TRD
BSS	-0.0392(0.0134)	-0.037(0.0125)	0.0257(0.0473)	-0.0157(0.0254)	-0.0348(0.0062)
AUC ROC	0.501(0.0024)	0.501(0.0025)	0.5332(0.0196)	0.511(0.0157)	0.5092(0.0116)
AUC PR	0.4635(0.0187)	0.4673(0.0138)	0.5043(0.0289)	0.4789(0.0108)	0.4556(0.0196)

Table B.51: SVM - CALIBRATED - Training

SVM - CALIBRATED	None	NORM	POWER	STD	TRD
BSS	-0.0091(0.0006)	-0.0091(0.0006)	-0.0668(0.1287)	-0.038(0.0643)	-0.0091(0.0006)
AUC ROC	0.5(0.0)	0.5(0.0)	0.4997(0.0006)	0.4999(0.0003)	0.5(0.0)
AUC PR	0.5046(0.0003)	0.5046(0.0003)	0.4046(0.2237)	0.4046(0.2237)	0.5046(0.0003)

Table B.52: SVM - CALIBRATED - Testing

	DT - CALIBRATED
BSS	0.009(0.0101)
AUC ROC	0.5089(0.005)
AUC PR	0.5134(0.0049)

Table B.53: CALIBRATED Decision Tree - Training

	DT - CALIBRATED
BSS	-0.0091(0.0006)
AUC ROC	0.5(0.0)
AUC PR	0.5046(0.0003)

Table B.54: CALIBRATED Decision Tree - Testing

KNN - CALIBRATED	None	NORM	POWER	STD	TRD
BSS	0.0337(0.0257)	0.0448(0.0341)	0.0448(0.0253)	0.0594(0.0313)	0.1043(0.0715)
AUC ROC	0.5222(0.0136)	0.5289(0.0168)	0.5289(0.0137)	0.5421(0.0236)	0.5572(0.0341)
AUC PR	0.5233(0.012)	0.5299(0.0172)	0.5299(0.012)	0.5281(0.0126)	0.5579(0.0379)

Table B.55: KNN - CALIBRATED - Training

KNN - CALIBRATED	None	NORM	POWER	STD	TRD
BSS	-0.0091(0.0006)	-0.0091(0.0006)	-0.0956(0.0786)	-0.0668(0.0787)	-0.0091(0.0006)
AUC ROC	0.5(0.0)	0.5(0.0)	0.4996(0.0004)	0.4997(0.0004)	0.5(0.0)
AUC PR	0.5046(0.0003)	0.5046(0.0003)	0.2046(0.274)	0.3046(0.274)	0.5046(0.0003)

Table B.56: KNN - CALIBRATED - Testing

B.3. Probability Prediction For CNS. Probability Prediction For CNS

GNB	None	NORM	POWER	STD	TRD
BSS	-5.3871(0.5222)	-5.4425(0.5682)	-5.5751(0.5645)	-5.4434(0.5667)	-5.4517(0.5462)
AUC ROC	0.9288(0.0149)	0.9301(0.0169)	0.9303(0.0165)	0.9268(0.0185)	0.9298(0.0158)
AUC PR	0.5243(0.0165)	0.5253(0.0203)	0.5247(0.0197)	0.5224(0.0218)	0.5258(0.019)

Table B.57: GNB - Training

GNB	None	NORM	POWER	STD	TRD
BSS	-5.1973(0.9614)	-5.399(1.1004)	-5.5143(1.1275)	-5.399(1.1004)	-0.0089(0.0)
AUC ROC	0.9299(0.0603)	0.929(0.0596)	0.9285(0.0593)	0.929(0.0596)	0.5(0.0)
AUC PR	0.5236(0.0643)	0.5219(0.063)	0.5209(0.0626)	0.5219(0.063)	0.5045(0.0)

Table B.58: GNB - Testing

GNB - SMOTENC	None	NORM	POWER	STD	TRD
BSS	-3.7386(0.3115)	-3.8953(0.3187)	-3.8808(0.3682)	-3.7888(0.3158)	-4.1517(0.0797)
AUC ROC	0.8687(0.0304)	0.8823(0.0196)	0.8791(0.0198)	0.8658(0.0342)	0.9309(0.0174)
AUC PR	0.4676(0.035)	0.482(0.0208)	0.478(0.0232)	0.4621(0.0405)	0.5353(0.0208)

Table B.59: GNB - SMOTENC - Training

GNB - SMOTENC	None	NORM	POWER	STD	TRD
BSS	-3.8137(0.6957)	-3.929(0.6478)	-3.9002(0.6683)	-3.929(0.6478)	-0.0089(0.0)
AUC ROC	0.9078(0.0728)	0.9073(0.0726)	0.8932(0.0932)	0.9073(0.0726)	0.5(0.0)
AUC PR	0.5083(0.0863)	0.5065(0.0855)	0.4906(0.1079)	0.5065(0.0855)	0.5045(0.0)

Table B.60: GNB - SMOTENC - Testing

GPC	None	NORM	POWER	STD	TRD
BSS	-0.6627(0.1526)	-0.0086(0.0)	-0.6021(0.1129)	-0.4498(0.0996)	-0.0086(0.0)
AUC ROC	0.544(0.0303)	0.5(0.0)	0.5591(0.0402)	0.5669(0.0182)	0.5(0.0)
AUC PR	0.1577(0.0818)	0.5045(0.0)	0.1812(0.0621)	0.2317(0.0349)	0.5045(0.0)

Table B.61: GPC - Training

GPC	None	NORM	POWER	STD	TRD
BSS	-0.6718(0.3318)	-0.0089(0.0)	-0.6718(0.3894)	-0.6142(0.3441)	-0.0089(0.0)
AUC ROC	0.5254(0.0397)	0.5(0.0)	0.5537(0.0587)	0.5822(0.0763)	0.5(0.0)
AUC PR	0.0804(0.1093)	0.5045(0.0)	0.1244(0.1198)	0.1616(0.1168)	0.5045(0.0)

Table B.62: GPC - Testing

GPC - SMOTENC	None	NORM	POWER	STD	TRD
BSS	-3.5391(0.4422)	-3.103(0.5047)	-2.354(0.2113)	-2.2814(0.4333)	-1.4513(0.1654)
AUC ROC	0.5569(0.0344)	0.7841(0.0331)	0.7603(0.0297)	0.7599(0.0501)	0.7107(0.0188)
AUC PR	0.0957(0.0442)	0.3762(0.0416)	0.36(0.037)	0.3622(0.0674)	0.3296(0.0201)

Table B.63: GPC - SMOTENC - Training

GPC - SMOTENC	None	NORM	POWER	STD	TRD
BSS	-3.4678(0.3813)	-3.0931(0.8441)	-1.9689(0.9206)	-2.3148(0.8465)	-0.0089(0.0)
AUC ROC	0.5412(0.0596)	0.826(0.0325)	0.8169(0.0636)	0.7446(0.13)	0.5(0.0)
AUC PR	0.0763(0.0741)	0.4239(0.0428)	0.4381(0.09)	0.3404(0.1682)	0.5045(0.0)

Table B.64: GPC - SMOTENC - Testing

LDA	None	NORM	POWER	STD	TRD
BSS	-2.5022(0.5679)	-2.5836(0.5724)	-2.4921(0.1687)	-2.511(0.5223)	-2.5984(0.5441)
AUC ROC	0.7485(0.0332)	0.7526(0.04)	0.7793(0.0189)	0.7567(0.0351)	0.7508(0.034)
AUC PR	0.3412(0.0326)	0.3472(0.0346)	0.3788(0.0258)	0.3516(0.0371)	0.3436(0.0296)

Table B.65: LDA - Training

LDA	None	NORM	POWER	STD	TRD
BSS	-2.6031(0.467)	-2.6031(0.467)	-2.3725(0.5544)	-2.6031(0.467)	-0.0953(0.1289)
AUC ROC	0.7716(0.0501)	0.7716(0.0501)	0.7868(0.0769)	0.7716(0.0501)	0.5138(0.0319)
AUC PR	0.3651(0.0603)	0.3651(0.0603)	0.387(0.0884)	0.3651(0.0603)	0.2686(0.252)

Table B.66: LDA - Testing

LDA - SMOTENC	None	NORM	POWER	STD	TRD
BSS	-6.8805(0.1579)	-6.5557(0.5588)	-6.3761(0.9235)	-6.3865(0.3559)	-7.1539(0.1857)
AUC ROC	0.9388(0.0188)	0.9396(0.034)	0.918(0.0371)	0.9134(0.0375)	0.9607(0.0092)
AUC PR	0.5296(0.0206)	0.5322(0.0371)	0.5082(0.0406)	0.502(0.0422)	0.5536(0.0102)

Table B.67: LDA - SMOTENC - Training

LDA - SMOTENC	None	NORM	POWER	STD	TRD
BSS	-6.8979(0.6867)	-6.2061(1.6464)	-6.2061(2.0458)	-6.4079(1.1192)	-6.4944(1.0881)
AUC ROC	0.9648(0.0031)	0.9537(0.0254)	0.9113(0.1177)	0.967(0.005)	0.9666(0.0049)
AUC PR	0.5569(0.0045)	0.548(0.0217)	0.5002(0.1235)	0.5608(0.0082)	0.5602(0.0082)

Table B.68: LDA - SMOTENC - Testing

LR	None	NORM	POWER	STD	TRD
BSS	-0.0063(0.0049)	-0.0086(0.0)	-0.0085(0.0198)	0.0128(0.0456)	-0.0018(0.015)
AUC ROC	0.5149(0.0333)	0.5(0.0)	0.5154(0.0252)	0.5161(0.0231)	0.5033(0.0075)
AUC PR	0.491(0.0301)	0.5045(0.0)	0.4893(0.0155)	0.5071(0.0256)	0.5078(0.0074)

Table B.69: Logistic Regression - Training

LR	None	NORM	POWER	STD	TRD
BSS	-0.0377(0.0644)	-0.0089(0.0)	-0.0377(0.1579)	-0.0089(0.1019)	-1.1906(2.5668)
AUC ROC	0.4999(0.0003)	0.5(0.0)	0.514(0.0321)	0.5142(0.032)	0.608(0.2047)
AUC PR	0.4045(0.2236)	0.5045(0.0)	0.4186(0.2335)	0.4186(0.2335)	0.4307(0.2405)

Table B.70: Logistic Regression - Testing

LR - CS	None	NORM	POWER	STD	TRD
BSS	-0.003(0.0125)	-0.0086(0.0)	0.006(0.0326)	0.0071(0.0351)	0.0004(0.0201)
AUC ROC	0.5165(0.0287)	0.5(0.0)	0.5094(0.0211)	0.5089(0.0199)	0.5044(0.0099)
AUC PR	0.4893(0.0226)	0.5045(0.0)	0.5071(0.006)	0.5099(0.0122)	0.5089(0.0098)

Table B.71: Cost Sensitive Logistic Regression - Training

LR - CS	None	NORM	POWER	STD	TRD
BSS	-0.0377(0.0644)	-0.0089(0.0)	-0.0665(0.1289)	-0.0377(0.0644)	-0.0665(0.1289)
AUC ROC	0.4999(0.0003)	0.5(0.0)	0.4997(0.0006)	0.4999(0.0003)	0.4997(0.0006)
AUC PR	0.4045(0.2236)	0.5045(0.0)	0.4045(0.2236)	0.4045(0.2236)	0.4045(0.2236)

Table B.72: Cost Sensitive Logistic Regression - Testing

MNB	MNB	MNB - SMOTENC
BSS	-60.8519(28.2386)	-93.3953(14.1801)
AUC ROC	0.6338(0.0908)	0.5193(0.0525)
AUC PR	0.4117(0.0644)	0.4354(0.0162)

Table B.73: MNB - Training

MNB	MNB	MNB - SMOTENC
BSS	-60.7706(29.6569)	-93.2271(17.6402)
AUC ROC	0.6257(0.1587)	0.5094(0.1364)
AUC PR	0.4084(0.0821)	0.434(0.0723)

Table B.74: MNB - Testing

QDA	None	NORM	POWER	STD	TRD
BSS	-1.5574(0.1667)	-1.7511(0.3908)	-1.9834(0.6477)	-2.1283(0.242)	-2.4016(2.4986)
AUC ROC	0.6283(0.0273)	0.638(0.0212)	0.6902(0.0287)	0.6729(0.0208)	0.5413(0.0244)
AUC PR	0.3142(0.039)	0.2778(0.0499)	0.3074(0.0428)	0.2692(0.0137)	0.3734(0.0604)

Table B.75: QDA - Training

QDA	None	NORM	POWER	STD	TRD
BSS	-4.7072(0.7249)	-4.7072(0.7801)	-4.1307(1.0119)	-4.5631(0.6882)	-0.0089(0.0)
AUC ROC	0.8896(0.0753)	0.8613(0.0618)	0.8922(0.0738)	0.8903(0.057)	0.5(0.0)
AUC PR	0.4807(0.0811)	0.4494(0.0671)	0.4881(0.0766)	0.4824(0.0607)	0.5045(0.0)

Table B.76: QDA - Testing

QDA - SMOTENC	None	NORM	POWER	STD	TRD
BSS	-3.4379(0.2552)	-3.4518(0.2928)	-3.3664(0.388)	-3.3554(0.4015)	-3.2943(0.3712)
AUC ROC	0.8933(0.0275)	0.8859(0.0252)	0.8671(0.0146)	0.877(0.032)	0.8307(0.0477)
AUC PR	0.4997(0.0314)	0.4896(0.0263)	0.4705(0.0154)	0.4825(0.0315)	0.4318(0.0484)

Table B.77: QDA - SMOTENC - Training

QDA - SMOTENC	None	NORM	POWER	STD	TRD
BSS	-3.5831(0.5711)	-3.5254(0.5545)	-3.439(0.7163)	-3.5254(0.4512)	-0.0089(0.0)
AUC ROC	0.9088(0.0519)	0.909(0.0519)	0.8528(0.0923)	0.8949(0.0607)	0.5(0.0)
AUC PR	0.5116(0.0645)	0.5124(0.0645)	0.449(0.1045)	0.496(0.0732)	0.5045(0.0)

Table B.78: QDA - SMOTENC - Testing

SVM - CALIBRATED	None	NORM	POWER	STD	TRD
BSS	-0.0086(0.0)	-0.0298(0.0219)	0.0071(0.0315)	-0.004(0.0145)	0.0441(0.0645)
AUC ROC	0.5(0.0)	0.5104(0.0235)	0.532(0.0237)	0.5342(0.0231)	0.5437(0.0265)
AUC PR	0.5045(0.0)	0.4649(0.0251)	0.4803(0.0099)	0.4714(0.0241)	0.5046(0.049)

Table B.79: SVM - CALIBRATED - Training

SVM - CALIBRATED	None	NORM	POWER	STD	TRD
BSS	-0.0377(0.0644)	-0.153(0.2038)	-0.0377(0.0644)	-0.0665(0.0789)	-0.0089(0.0)
AUC ROC	0.4999(0.0003)	0.4994(0.0009)	0.514(0.0317)	0.5139(0.0318)	0.5(0.0)
AUC PR	0.4045(0.2236)	0.3045(0.2739)	0.3686(0.2179)	0.2686(0.252)	0.5045(0.0)

Table B.80: SVM - CALIBRATED - Testing

KNN - CALIBRATED	None	NORM	POWER	STD	TRD
BSS	-0.0086(0.0)	0.0049(0.0255)	-0.0075(0.0083)	-0.0052(0.005)	0.0665(0.0306)
AUC ROC	0.5(0.0)	0.5089(0.0139)	0.5028(0.0039)	0.5072(0.0132)	0.5499(0.0252)
AUC PR	0.5045(0.0)	0.5083(0.0148)	0.5005(0.0087)	0.4999(0.0134)	0.5335(0.0253)

Table B.81: Decision Tree - CALIBRATED - Training

	DT - CALIBRATED
BSS	-0.0665(0.0789)
AUC ROC	0.4997(0.0004)
AUC PR	0.3045(0.2739)

Table B.82: Decision Tree - CALIBRATED - Testing

	DT - CALIBRATED
BSS	0.0205(0.0183)
AUC ROC	0.526(0.019)
AUC PR	0.5053(0.0064)

Table B.83: KNN - CALIBRATED - Training

KNN - CALIBRATED	None	NORM	POWER	STD	TRD
BSS	-0.0089(0.0)	-0.0089(0.1019)	-0.0665(0.0789)	-0.0377(0.0645)	-0.0089(0.0)
AUC ROC	0.5(0.0)	0.5142(0.032)	0.4997(0.0004)	0.4999(0.0003)	0.5(0.0)
AUC PR	0.5045(0.0)	0.4186(0.2335)	0.3045(0.2739)	0.4045(0.2236)	0.5045(0.0)

Table B.84: KNN - CALIBRATED - Testing

B.4. Probability Prediction For GU. Probability Prediction For GU

GNB	None	NORM	POWER	STD	TRD
BSS	-4.388(0.4713)	-4.8125(0.2568)	-4.8491(0.2728)	-4.8221(0.2558)	-4.8123(0.2267)
AUC ROC	0.9377(0.0184)	0.9401(0.0187)	0.9371(0.0147)	0.9373(0.0103)	0.9317(0.0122)
AUC PR	0.5427(0.0222)	0.5425(0.0216)	0.5391(0.0175)	0.5395(0.0122)	0.5335(0.0137)

Table B.85: GNB - Training

GNB	None	NORM	POWER	STD	TRD
BSS	-5.1973(0.9614)	-5.399(1.1004)	-5.5143(1.1275)	-5.399(1.1004)	-0.0089(0.0)
AUC ROC	0.9299(0.0603)	0.929(0.0596)	0.9285(0.0593)	0.929(0.0596)	0.5(0.0)
AUC PR	0.5236(0.0643)	0.5219(0.063)	0.5209(0.0626)	0.5219(0.063)	0.5045(0.0)

Table B.86: GNB - Testing

GNB - SMOTENC	None	NORM	POWER	STD	TRD
BSS	-0.8806(0.1639)	-0.9357(0.1499)	-0.9627(0.1565)	-0.9346(0.1452)	-1.2638(0.1452)
AUC ROC	0.8161(0.0296)	0.8284(0.0252)	0.8329(0.0236)	0.8336(0.02)	0.8711(0.0133)
AUC PR	0.4957(0.0373)	0.4971(0.0407)	0.4983(0.0261)	0.5013(0.0243)	0.5321(0.0236)

Table B.87: GNB - SMOTENC - Training

GNB - SMOTENC	None	NORM	POWER	STD	TRD
BSS	-3.8137(0.6957)	-3.929(0.6478)	-3.9002(0.6683)	-3.929(0.6478)	-0.0089(0.0)
AUC ROC	0.9078(0.0728)	0.9073(0.0726)	0.8932(0.0932)	0.9073(0.0726)	0.5(0.0)
AUC PR	0.5083(0.0863)	0.5065(0.0855)	0.4906(0.1079)	0.5065(0.0855)	0.5045(0.0)

Table B.88: GNB - SMOTENC - Testing

GPC	None	NORM	POWER	STD	TRD
BSS	-0.5604(0.0598)	-0.019(0.0166)	-0.2792(0.1559)	-0.2568(0.1201)	-0.01(0.0)
AUC ROC	0.5014(0.007)	0.5011(0.0025)	0.6359(0.0744)	0.6346(0.0409)	0.5(0.0)
AUC PR	0.0928(0.0187)	0.4895(0.0216)	0.3296(0.1287)	0.3444(0.0741)	0.505(0.0)

Table B.89: GPC - Training

GPC	None	NORM	POWER	STD	TRD
BSS	-0.6718(0.3318)	-0.0089(0.0)	-0.6718(0.3894)	-0.6142(0.3441)	-0.0089(0.0)
AUC ROC	0.5254(0.0397)	0.5(0.0)	0.5537(0.0587)	0.5822(0.0763)	0.5(0.0)
AUC PR	0.0804(0.1093)	0.5045(0.0)	0.1244(0.1198)	0.1616(0.1168)	0.5045(0.0)

Table B.90: GPC - Testing

GPC - SMOTENC	None	NORM	POWER	STD	TRD
BSS	-3.2518(0.1874)	-1.0977(0.1306)	-0.8728(0.2384)	-0.9424(0.2144)	-0.5632(0.1023)
AUC ROC	0.5131(0.0216)	0.8307(0.0232)	0.8475(0.0287)	0.8317(0.0345)	0.7813(0.0171)
AUC PR	0.0422(0.0272)	0.4944(0.0329)	0.5262(0.0506)	0.5033(0.0531)	0.4679(0.0156)

Table B.91: GPC - SMOTENC - Training

GPC - SMOTENC	None	NORM	POWER	STD	TRD
BSS	-3.4678(0.3813)	-3.0931(0.8441)	-1.9689(0.9206)	-2.3148(0.8465)	-0.0089(0.0)
AUC ROC	0.5412(0.0596)	0.826(0.0325)	0.8169(0.0636)	0.7446(0.13)	0.5(0.0)
AUC PR	0.0763(0.0741)	0.4239(0.0428)	0.4381(0.09)	0.3404(0.1682)	0.5045(0.0)

Table B.92: GPC - SMOTENC - Testing

LDA	None	NORM	POWER	STD	TRD
BSS	-1.9226(0.2165)	-1.9294(0.2322)	-1.7255(0.1441)	-1.951(0.2212)	-1.9482(0.1826)
AUC ROC	0.837(0.0179)	0.8383(0.0178)	0.8408(0.0163)	0.8372(0.0185)	0.8372(0.0176)
AUC PR	0.465(0.0265)	0.4654(0.0244)	0.4759(0.0215)	0.4632(0.0267)	0.4665(0.0261)

Table B.93: LDA - Training

LDA	None	NORM	POWER	STD	TRD
BSS	-2.6031(0.467)	-2.6031(0.467)	-2.3725(0.5544)	-2.6031(0.467)	-0.0953(0.1289)
AUC ROC	0.7716(0.0501)	0.7716(0.0501)	0.7868(0.0769)	0.7716(0.0501)	0.5138(0.0319)
AUC PR	0.3651(0.0603)	0.3651(0.0603)	0.387(0.0884)	0.3651(0.0603)	0.2686(0.252)

Table B.94: LDA - Testing

LDA - SMOTENC	None	NORM	POWER	STD	TRD
BSS	-3.0554(0.9774)	-3.8286(0.8734)	-3.6911(0.9953)	-4.3109(1.0684)	-5.364(0.3747)
AUC ROC	0.8781(0.0386)	0.9001(0.0198)	0.9184(0.0179)	0.9151(0.0196)	0.952(0.0113)
AUC PR	0.494(0.0372)	0.507(0.0181)	0.5325(0.0099)	0.5235(0.0173)	0.5536(0.0134)

Table B.95: LDA - SMOTENC - Training

LDA - SMOTENC	None	NORM	POWER	STD	TRD
BSS	-6.8979(0.6867)	-6.2061(1.6464)	-6.2061(2.0458)	-6.4079(1.1192)	-6.4944(1.0881)
AUC ROC	0.9648(0.0031)	0.9537(0.0254)	0.9113(0.1177)	0.967(0.005)	0.9666(0.0049)
AUC PR	0.5569(0.0045)	0.548(0.0217)	0.5002(0.1235)	0.5608(0.0082)	0.5602(0.0082)

Table B.96: LDA - SMOTENC - Testing

LR	None	NORM	POWER	STD	TRD
BSS	-0.01(0.0)	0.0147(0.0323)	-0.0442(0.0936)	-0.019(0.0851)	-0.0257(0.0571)
AUC ROC	0.5(0.0)	0.5221(0.0237)	0.5565(0.0241)	0.5695(0.0159)	0.5337(0.0149)
AUC PR	0.505(0.0)	0.5104(0.019)	0.4394(0.0603)	0.4324(0.0557)	0.4383(0.057)

Table B.97: Logistic Regression - Training

LR	None	NORM	POWER	STD	TRD
BSS	-0.0377(0.0644)	-0.0089(0.0)	-0.0377(0.1579)	-0.0089(0.1019)	-1.1906(2.5668)
AUC ROC	0.4999(0.0003)	0.5(0.0)	0.514(0.0321)	0.5142(0.032)	0.608(0.2047)
AUC PR	0.4045(0.2236)	0.5045(0.0)	0.4186(0.2335)	0.4186(0.2335)	0.4307(0.2405)

Table B.98: Logistic Regression - Testing

LR - CS	None	NORM	POWER	STD	TRD
BSS	-0.0083(0.0038)	0.008(0.0123)	0.0214(0.036)	0.0253(0.0426)	0.0046(0.0281)
AUC ROC	0.5019(0.0027)	0.5133(0.0132)	0.5263(0.0404)	0.5238(0.0344)	0.5191(0.0271)
AUC PR	0.5036(0.0057)	0.5065(0.0154)	0.5095(0.0129)	0.5153(0.0097)	0.4957(0.0164)

Table B.99: Cost Sensitive Logistic Regression - Training

LR - CS	None	NORM	POWER	STD	TRD
BSS	-0.0377(0.0644)	-0.0089(0.0)	-0.0665(0.1289)	-0.0377(0.0644)	-0.0665(0.1289)
AUC ROC	0.4999(0.0003)	0.5(0.0)	0.4997(0.0006)	0.4999(0.0003)	0.4997(0.0006)
AUC PR	0.4045(0.2236)	0.5045(0.0)	0.4045(0.2236)	0.4045(0.2236)	0.4045(0.2236)

Table B.100: Cost Sensitive Logistic Regression - Testing

MNB	MNB	MNB - SMOTENC
BSS	-20.3758(5.277)	-30.0413(7.6563)
AUC ROC	0.7044(0.0423)	0.7571(0.039)
AUC PR	0.3259(0.0665)	0.4249(0.0631)

Table B.101: MNB - Training

MNB	MNB	MNB - SMOTENC
BSS	-60.7706(29.6569)	-93.2271(17.6402)
AUC ROC	0.6257(0.1587)	0.5094(0.1364)
AUC PR	0.4084(0.0821)	0.434(0.0723)

Table B.102: MNB - Testing

QDA	None	NORM	POWER	STD	TRD
BSS	-4.3034(0.5259)	-4.0447(0.7514)	-3.7373(1.1312)	-4.0472(0.7855)	-3.931(0.8457)
AUC ROC	0.9002(0.015)	0.9056(0.0155)	0.8843(0.0187)	0.9022(0.0101)	0.9011(0.0148)
AUC PR	0.5026(0.0144)	0.512(0.022)	0.4944(0.0351)	0.5076(0.0179)	0.5078(0.0234)

Table B.103: QDA - Training

QDA	None	NORM	POWER	STD	TRD
BSS	-4.7072(0.7249)	-4.7072(0.7801)	-4.1307(1.0119)	-4.5631(0.6882)	-0.0089(0.0)
AUC ROC	0.8896(0.0753)	0.8613(0.0618)	0.8922(0.0738)	0.8903(0.057)	0.5(0.0)
AUC PR	0.4807(0.0811)	0.4494(0.0671)	0.4881(0.0766)	0.4824(0.0607)	0.5045(0.0)

Table B.104: QDA - Testing

QDA - SMOTENC	None	NORM	POWER	STD	TRD
BSS	-1.5409(0.6749)	-1.1387(0.2828)	-1.0058(0.2426)	-1.0966(0.2255)	-1.3176(0.2096)
AUC ROC	0.8271(0.0242)	0.817(0.0264)	0.8157(0.0253)	0.8029(0.0222)	0.8393(0.0278)
AUC PR	0.4709(0.0424)	0.4672(0.0387)	0.4804(0.0397)	0.4618(0.0325)	0.4947(0.0371)

Table B.105: QDA - SMOTENC - Training

QDA - SMOTENC	None	NORM	POWER	STD	TRD
BSS	-3.5831(0.5711)	-3.5254(0.5545)	-3.439(0.7163)	-3.5254(0.4512)	-0.0089(0.0)
AUC ROC	0.9088(0.0519)	0.909(0.0519)	0.8528(0.0923)	0.8949(0.0607)	0.5(0.0)
AUC PR	0.5116(0.0645)	0.5124(0.0645)	0.449(0.1045)	0.496(0.0732)	0.5045(0.0)

Table B.106: QDA - SMOTENC - Testing

SVM - CALIBRATED	None	NORM	POWER	STD	TRD
BSS	-0.0128(0.0383)	-0.0139(0.0702)	-0.0274(0.0476)	0.0007(0.0432)	-0.0223(0.0639)
AUC ROC	0.5118(0.0147)	0.5201(0.014)	0.5244(0.0335)	0.53(0.031)	0.5214(0.0132)
AUC PR	0.4752(0.0384)	0.4718(0.0568)	0.4556(0.0354)	0.4822(0.037)	0.4509(0.0618)

Table B.107: SVM - CALIBRATED - Training

SVM - CALIBRATED	None	NORM	POWER	STD	TRD
BSS	-0.0377(0.0644)	-0.153(0.2038)	-0.0377(0.0644)	-0.0665(0.0789)	-0.0089(0.0)
AUC ROC	0.4999(0.0003)	0.4994(0.0009)	0.514(0.0317)	0.5139(0.0318)	0.5(0.0)
AUC PR	0.4045(0.2236)	0.3045(0.2739)	0.3686(0.2179)	0.2686(0.252)	0.5045(0.0)

Table B.108: SVM - CALIBRATED - Testing

	DT - CALIBRATED
BSS	0.0433(0.0801)
AUC ROC	0.5305(0.0376)
AUC PR	0.5269(0.0426)

Table B.109: Calibrated Decision Tree - Training

	DT - CALIBRATED
BSS	-0.0606(0.1129)
AUC ROC	0.5121(0.0278)
AUC PR	0.3674(0.2183)

Table B.110: Calibrated Decision Tree - Testing

KNN - CALIBRATED	None	NORM	POWER	STD	TRD
BSS	-0.01(0.0)	0.0136(0.0419)	-0.0038(0.0646)	-0.0021(0.0525)	0.036(0.0518)
AUC ROC	0.5(0.0)	0.5656(0.0499)	0.5355(0.0533)	0.5457(0.042)	0.5585(0.0551)
AUC PR	0.505(0.0)	0.4752(0.0237)	0.4587(0.0386)	0.4507(0.0235)	0.4952(0.0086)

Table B.111: Calibrated KNN - Training

KNN - CALIBRATED	None	NORM	POWER	STD	TRD
BSS	-0.0101(0.0)	-0.2122(0.3851)	-0.2122(0.3851)	-0.1869(0.2766)	-0.0101(0.0)
AUC ROC	0.5(0.0)	0.5237(0.0328)	0.499(0.0019)	0.5115(0.0275)	0.5(0.0)
AUC PR	0.505(0.0)	0.2909(0.225)	0.305(0.2739)	0.3424(0.2323)	0.505(0.0)

Table B.112: Calibrated KNN - Testing

B.5. Probability Prediction For MUSC. Probability Prediction For MUSC

GNB	None	NORM	POWER	STD	TRD
BSS	-4.2575(0.2417)	-4.2461(0.1935)	-4.2516(0.2652)	-4.2064(0.2154)	-4.2315(0.2167)
AUC ROC	0.9417(0.0065)	0.9407(0.0085)	0.9559(0.0058)	0.9411(0.0104)	0.9402(0.0091)
AUC PR	0.5523(0.0062)	0.5509(0.0083)	0.5682(0.0047)	0.5516(0.0109)	0.5501(0.0099)

Table B.113: GNB - Training

GNB	None	NORM	POWER	STD	TRD
BSS	-3.9665(0.586)	-4.1238(0.8761)	-4.1238(0.8761)	-4.1463(0.8535)	-0.0113(0.0)
AUC ROC	0.95(0.029)	0.9491(0.0284)	0.9601(0.0248)	0.949(0.0285)	0.5(0.0)
AUC PR	0.5606(0.0316)	0.5591(0.0309)	0.5714(0.0297)	0.5587(0.0313)	0.5056(0.0)

Table B.114: GNB - Testing

GNB - SMOTENC	None	NORM	POWER	STD	TRD
BSS	-2.8854(0.1795)	-2.6487(0.1377)	-2.6304(0.152)	-2.6096(0.1308)	-3.101(0.1477)
AUC ROC	0.8496(0.0088)	0.8472(0.0129)	0.8508(0.0106)	0.8501(0.0141)	0.9185(0.0139)
AUC PR	0.4599(0.0126)	0.4617(0.014)	0.4672(0.0144)	0.4669(0.0185)	0.5373(0.0171)

Table B.115: GNB - SMOTENC - Training

GNB - SMOTENC	None	NORM	POWER	STD	TRD
BSS	-2.8428(0.7554)	-2.5507(0.5596)	-2.5507(0.5596)	-2.5507(0.5596)	-0.0113(0.0)
AUC ROC	0.8465(0.0515)	0.8481(0.0502)	0.8481(0.0502)	0.8481(0.0502)	0.5(0.0)
AUC PR	0.4558(0.0648)	0.4615(0.064)	0.4615(0.064)	0.4615(0.064)	0.5056(0.0)

Table B.116: GNB - SMOTENC - Testing

GPC	None	NORM	POWER	STD	TRD
BSS	-0.6372(0.0955)	-0.0111(0.0)	-0.3408(0.1619)	-0.3436(0.05)	-0.0111(0.0)
AUC ROC	0.5598(0.0178)	0.5(0.0)	0.585(0.0381)	0.5706(0.0198)	0.5(0.0)
AUC PR	0.1566(0.0401)	0.5056(0.0)	0.256(0.1032)	0.2256(0.0542)	0.5056(0.0)

Table B.117: GPC - Training

GPC	None	NORM	POWER	STD	TRD
BSS	-0.7978(0.2102)	-0.0113(0.0)	-0.3034(0.2587)	-0.3034(0.1704)	-0.0113(0.0)
AUC ROC	0.5175(0.0295)	0.5(0.0)	0.5643(0.0915)	0.5423(0.0466)	0.5(0.0)
AUC PR	0.0476(0.0575)	0.5056(0.0)	0.1691(0.2273)	0.1363(0.1478)	0.5056(0.0)

Table B.118: GPC - Testing

GPC - SMOTENC	None	NORM	POWER	STD	TRD
BSS	-2.6422(0.1946)	-2.0255(0.5157)	-1.6211(0.2411)	-1.8031(0.3117)	-1.5083(0.2126)
AUC ROC	0.5799(0.0152)	0.7877(0.0126)	0.7405(0.0435)	0.7194(0.0432)	0.6559(0.0352)
AUC PR	0.1333(0.0182)	0.4039(0.0222)	0.3553(0.0596)	0.3239(0.0612)	0.2539(0.0512)

Table B.119: GPC - SMOTENC - Training

GPC - SMOTENC	None	NORM	POWER	STD	TRD
BSS	-2.9327(0.5149)	-1.8765(0.7308)	-1.4046(0.293)	-1.5394(0.6082)	-0.0113(0.0)
AUC ROC	0.5714(0.0492)	0.797(0.1167)	0.7118(0.1099)	0.7001(0.1063)	0.5(0.0)
AUC PR	0.1209(0.0647)	0.4162(0.1504)	0.3192(0.1472)	0.3026(0.1472)	0.5056(0.0)

Table B.120: GPC - SMOTENC - Testing

LDA	None	NORM	POWER	STD	TRD
BSS	-2.429(0.4722)	-2.3534(0.4674)	-1.7996(0.1869)	-2.4101(0.4563)	-2.3467(0.4673)
AUC ROC	0.8235(0.0129)	0.8323(0.0165)	0.7983(0.0168)	0.8284(0.0138)	0.8309(0.0121)
AUC PR	0.4392(0.0179)	0.4495(0.0191)	0.4227(0.0277)	0.4441(0.0181)	0.4495(0.0174)

Table B.121: LDA - Training

LDA	None	NORM	POWER	STD	TRD
BSS	-2.326(0.4174)	-2.326(0.4174)	-1.6293(0.3237)	-2.326(0.4174)	-2.0788(0.188)
AUC ROC	0.8274(0.0885)	0.8274(0.0885)	0.8204(0.0879)	0.8274(0.0885)	0.7629(0.0872)
AUC PR	0.4393(0.0988)	0.4393(0.0988)	0.4484(0.1123)	0.4393(0.0988)	0.367(0.1073)

Table B.122: LDA - Testing

LDA - SMOTENC	None	NORM	POWER	STD	TRD
BSS	-4.8654(0.3168)	-4.6015(0.4665)	-4.9103(0.284)	-4.6038(0.3992)	-5.0811(0.1537)
AUC ROC	0.9486(0.0121)	0.9341(0.0243)	0.9497(0.0104)	0.9234(0.0201)	0.9551(0.0068)
AUC PR	0.5563(0.0128)	0.5418(0.0254)	0.5568(0.0112)	0.5294(0.0193)	0.5618(0.0079)

Table B.123: LDA - SMOTENC - Training

LDA - SMOTENC	None	NORM	POWER	STD	TRD
BSS	-4.9553(0.6103)	-4.753(0.6619)	-4.9553(0.6103)	-4.7081(0.6714)	-4.7305(0.4204)
AUC ROC	0.9554(0.0255)	0.9455(0.0293)	0.9554(0.0233)	0.9458(0.029)	0.9566(0.0236)
AUC PR	0.5609(0.0293)	0.5509(0.0321)	0.5608(0.0254)	0.5514(0.0312)	0.563(0.0256)

Table B.124: LDA - SMOTENC - Testing

LR	None	NORM	POWER	STD	TRD
BSS	-0.0083(0.0049)	-0.0111(0.0)	-0.0076(0.0425)	-0.0369(0.0252)	-0.0184(0.0163)
AUC ROC	0.5256(0.0357)	0.5(0.0)	0.5585(0.0331)	0.5479(0.0234)	0.5(0.0001)
AUC PR	0.4843(0.0294)	0.5056(0.0)	0.4376(0.0437)	0.4191(0.0227)	0.4923(0.0298)

Table B.125: Logistic Regression - Training

LR	None	NORM	POWER	STD	TRD
BSS	-0.0337(0.1231)	-0.0113(0.0)	-0.0113(0.0795)	-0.1012(0.094)	-0.0113(0.0)
AUC ROC	0.5438(0.0718)	0.5(0.0)	0.5549(0.0004)	0.5324(0.0492)	0.5(0.0)
AUC PR	0.4346(0.14)	0.5056(0.0)	0.3439(0.1264)	0.2286(0.2182)	0.5056(0.0)

Table B.126: Logistic Regression - Testing

LR - CS	None	NORM	POWER	STD	TRD
BSS	-0.0094(0.0025)	-0.0111(0.0)	-0.0032(0.0131)	-0.0072(0.0062)	-0.0111(0.0)
AUC ROC	0.5124(0.022)	0.5(0.0)	0.5572(0.0348)	0.5198(0.0278)	0.5(0.0)
AUC PR	0.4947(0.0178)	0.5056(0.0)	0.4594(0.0293)	0.4827(0.0315)	0.5056(0.0)

Table B.127: Cost Sensitive Logistic Regression - Training

LR - CS	None	NORM	POWER	STD	TRD
BSS	0.0112(0.0502)	-0.0113(0.0)	-0.0113(0.0795)	-0.0562(0.0615)	-0.0113(0.0)
AUC ROC	0.5331(0.074)	0.5(0.0)	0.5439(0.0246)	0.5107(0.0247)	0.5(0.0)
AUC PR	0.4986(0.0157)	0.5056(0.0)	0.3829(0.1426)	0.2666(0.2512)	0.5056(0.0)

Table B.128: Cost Sensitive Logistic Regression - Testing

MNB	MNB	MNB/SMOTENC
BSS	-23.165(25.269)	-29.5385(26.8746)
AUC ROC	0.4926(0.0299)	0.5239(0.0529)
AUC PR	0.1293(0.169)	0.2018(0.1389)

Table B.129: MNB - Training

MNB	MNB/None	MNB-SMOTENC/None
BSS	-16.9333(29.1752)	-27.1809(30.1303)
AUC ROC	0.5032(0.0348)	0.5771(0.0817)
AUC PR	0.1085(0.194)	0.2482(0.1297)

Table B.130: MNB - Testing

QDA	None	NORM	POWER	STD	TRD
BSS	-4.1932(0.2276)	-4.205(0.2246)	-4.1391(0.2003)	-4.2223(0.2896)	-4.246(0.269)
AUC ROC	0.9202(0.0087)	0.9229(0.0199)	0.9327(0.0158)	0.9181(0.0201)	0.9186(0.0165)
AUC PR	0.5277(0.0108)	0.5315(0.0226)	0.5426(0.0173)	0.5262(0.0237)	0.5258(0.0188)

Table B.131: QDA - Training

QDA	None	NORM	POWER	STD	TRD
BSS	-4.1238(0.9183)	-4.1238(0.9183)	-4.0564(0.9232)	-4.1238(0.9183)	-0.0113(0.0)
AUC ROC	0.9271(0.0273)	0.9271(0.0273)	0.9385(0.0285)	0.9271(0.0273)	0.5(0.0)
AUC PR	0.5348(0.0343)	0.5348(0.0343)	0.5479(0.0318)	0.5348(0.0343)	0.5056(0.0)

Table B.132: QDA - Testing

QDA - SMOTENC	None	NORM	POWER	STD	TRD
BSS	-2.8825(0.1796)	-2.7168(0.1477)	-2.5106(0.1633)	-2.6151(0.1686)	-3.0129(0.1595)
AUC ROC	0.8409(0.0186)	0.8434(0.0154)	0.8224(0.0261)	0.8369(0.0302)	0.8626(0.0377)
AUC PR	0.4515(0.0235)	0.4557(0.0185)	0.4384(0.0255)	0.4509(0.0327)	0.4742(0.0419)

Table B.133: QDA - SMOTENC - Training

QDA - SMOTENC	None	NORM	POWER	STD	TRD
BSS	-2.8428(0.7637)	-2.5732(0.6019)	-2.5282(0.6386)	-2.5507(0.603)	-0.0113(0.0)
AUC ROC	0.8355(0.0661)	0.826(0.0633)	0.8263(0.0813)	0.8262(0.0633)	0.5(0.0)
AUC PR	0.4433(0.0828)	0.4358(0.0829)	0.4359(0.0976)	0.4363(0.0828)	0.5056(0.0)

Table B.134: QDA - SMOTENC - Testing

SVM - CALIBRATED	None	NORM	POWER	STD	TRD
BSS	-0.0313(0.0415)	-0.0465(0.024)	-0.0184(0.024)	-0.0369(0.0294)	-0.0212(0.0176)
AUC ROC	0.5109(0.0133)	0.5067(0.0064)	0.5313(0.0283)	0.5158(0.0123)	0.5148(0.0125)
AUC PR	0.445(0.0401)	0.4453(0.024)	0.4511(0.0375)	0.4466(0.0233)	0.466(0.0235)

Table B.135: SVM - CALIBRATED - Training

SVM - CALIBRATED	None	NORM	POWER	STD	TRD
BSS	0.0112(0.0503)	-0.0562(0.1704)	-0.0787(0.1005)	-0.0562(0.1005)	-7.6969(17.1856)
AUC ROC	0.5111(0.0248)	0.5107(0.0251)	0.5216(0.0304)	0.4997(0.0006)	0.5225(0.0503)
AUC PR	0.5166(0.0246)	0.4166(0.231)	0.2276(0.2177)	0.4056(0.2236)	0.4733(0.0724)

Table B.136: SVM - CALIBRATED - Testing

	DT_CALIBRATED
BSS	0.0024(0.0108)
AUC ROC	0.5097(0.0081)
AUC PR	0.5075(0.0078)

Table B.137: Decision Tree - CALIBRATED - Training

	DT - CALIBRATED
BSS	-0.0606(0.1129)
AUC ROC	0.5121(0.0278)
AUC PR	0.3674(0.2183)

Table B.138: Decision Tree - CALIBRATED - Testing

KNN - CALIBRATED	None	NORM	POWER	STD	TRD
BSS	-0.0111(0.0)	-0.0133(0.017)	0.0254(0.0362)	-0.0083(0.004)	-0.0167(0.0053)
AUC ROC	0.5(0.0)	0.5066(0.007)	0.5543(0.0397)	0.5041(0.0071)	0.5016(0.0022)
AUC PR	0.5056(0.0)	0.4872(0.0114)	0.4834(0.0317)	0.4998(0.0155)	0.4923(0.013)

Table B.139: KNN - CALIBRATED - Training

KNN - CALIBRATED	None	NORM	POWER	STD	TRD
BSS	-0.0101(0.0)	-0.2122(0.3851)	-0.2122(0.3851)	-0.1869(0.2766)	-0.0101(0.0)
AUC ROC	0.5(0.0)	0.5237(0.0328)	0.499(0.0019)	0.5115(0.0275)	0.5(0.0)
AUC PR	0.505(0.0)	0.2909(0.225)	0.305(0.2739)	0.3424(0.2323)	0.505(0.0)

Table B.140: KNN - CALIBRATED - Testing

B.6. Summary of Probability Prediction Results.

		Training - Testing		
Model	Best Scaling	BSS	AUC ROC	AUC PR
GNB	No Scaling	0.4903(0.0478)	0.9089(0.0185)	0.7879(0.019)
		0.4673(0.1544)	0.9052(0.0147)	0.7765(0.0503)
GNB - SMOTENC	POWER	0.4758(0.044)	0.9808(0.0045)	0.8318(0.0107)
		0.4729(0.166)	0.9799(0.013)	0.8299(0.0403)
GPC	NORM	0.4821(0.0411)	0.7768(0.0195)	0.7493(0.0219)
		0.4673(0.1273)	0.7716(0.0552)	0.7395(0.0719)
GPC - SMOTENC	NORM	0.5491(0.017)	0.9705(0.0047)	0.8408(0.0079)
		0.5353(0.0254)	0.9711(0.0211)	0.8345(0.0112)
KNN - CALIBRATED	NORM	0.5201(0.0176)	0.8443(0.0112)	0.7684(0.0094)
		0.4952(0.1123)	0.8365(0.063)	0.7525(0.0645)
LDA	POWER	0.6403(0.0239)	0.8369(0.0075)	0.8338(0.0122)
		0.6373(0.101)	0.8322(0.0455)	0.8322(0.0479)
LDA - SMOTENC	POWER	0.2852(0.0225)	0.9832(0.0005)	0.8017(0.0031)
		0.2859(0.0905)	0.9832(0.0021)	0.7981(0.0155)
Logistic Regression	POWER	0.7376(0.028)	0.9368(0.013)	0.8822(0.0134)
		0.7166(0.075)	0.924(0.0282)	0.8671(0.0371)
Cost Sensitive Logistic Regression	POWER	0.7423(0.027)	0.9457(0.0204)	0.8871(0.0142)
		0.7166(0.075)	0.9368(0.015)	0.8718(0.0292)
MNB	No Scaling	-5.6117(0.2959)	0.5498(0.0117)	0.2363(0.012)
		-5.7604(0.4973)	0.5427(0.0428)	0.2316(0.0425)
MNB - SMOTENC	No Scaling	-5.5392(0.284)	0.5521(0.0162)	0.2375(0.0188)
		-5.6414(0.6043)	0.5481(0.0518)	0.2352(0.052)
QDA	POWER	0.3267(0.0571)	0.8157(0.0122)	0.6825(0.0255)
		0.3029(0.1353)	0.8063(0.0335)	0.6641(0.0643)
QDA - SMOTENC	POWER	0.3112(0.0563)	0.8069(0.0165)	0.6735(0.0292)
		0.2916(0.1472)	0.8009(0.0431)	0.6565(0.0747)
SVM - CALIBRATED	POWER	0.6921(0.0459)	0.8995(0.0159)	0.8556(0.022)
		0.6654(0.0734)	0.8894(0.0215)	0.84(0.0349)
Decision Tree - CALIBRATED	No Scaling	0.7507(0.021)	0.9194(0.0107)	0.8833(0.0107)
		0.6824(0.0677)	0.9077(0.0224)	0.8507(0.0339)

Table B.141: Summary Results - Probability Prediction - All Defects

		Training - Testing		
Model	Best Scaling	BSS	AUC ROC	AUC PR
GNB	No Scaling	-5.463(0.2828)	0.9515(0.0097)	0.5501(0.0118)
		-5.4801(0.7148)	0.9562(0.0299)	0.5521(0.0327)
GNB - SMOTENC	STD	-3.457(1.0072)	0.8091(0.0251)	0.406(0.0168)
		-3.1406(1.3114)	0.7654(0.1017)	0.3514(0.1075)
GPC	NORM	-0.009(0.0002)	0.5(0.0)	0.5046(0.0001)
		-0.0091(0.0006)	0.5(0.0)	0.5046(0.0003)
GPC - SMOTENC	STD	-2.0656(0.5478)	0.7335(0.0453)	0.3335(0.0701)
		-1.93(0.7267)	0.7266(0.1122)	0.3259(0.1496)
KNN - CALIBRATED	STD	0.0594(0.0313)	0.5421(0.0236)	0.5281(0.0126)
		-0.0668(0.0787)	0.4997(0.0004)	0.3046(0.274)
LDA	POWER	-1.941(0.1383)	0.8646(0.0282)	0.4952(0.0374)
		-1.8435(0.5434)	0.865(0.1136)	0.4919(0.1396)
LDA - SMOTENC	POWER	-3.681(1.8581)	0.8857(0.0314)	0.4987(0.0373)
		-3.1947(1.7138)	0.8589(0.108)	0.4645(0.1187)
Logistic Regression	NORM	-0.009(0.0002)	0.5(0.0)	0.5046(0.0001)
		-0.0091(0.0006)	0.5(0.0)	0.5046(0.0003)
Cost Sensitive Logistic Regression	POWER	-0.0123(0.0244)	0.5077(0.0053)	0.4923(0.0266)
		-0.0956(0.1932)	0.4996(0.0009)	0.4046(0.2237)
MNB	No Scaling	-5.2993(0.5993)	0.5246(0.0074)	0.0602(0.0082)
		-4.8094(1.3577)	0.5348(0.0564)	0.0716(0.0688)
MNB - SMOTENC	No Scaling	-9.8787(3.0044)	0.533(0.0139)	0.0909(0.0189)
		-6.7587(2.6378)	0.5544(0.0819)	0.0991(0.1009)
QDA	No Scaling	-4.399(1.4047)	0.8654(0.1397)	0.4932(0.0756)
		-5.1666(0.8965)	0.9434(0.0387)	0.5399(0.0435)
QDA - SMOTENC	STD	-3.6887(0.595)	0.8158(0.0177)	0.4084(0.0212)
		-3.0003(1.2078)	0.8084(0.1223)	0.4035(0.1365)
SVM - CALIBRATED	POWER	0.0257(0.0473)	0.5332(0.0196)	0.5043(0.0289)
		-0.0668(0.1287)	0.4997(0.0006)	0.4046(0.2237)
Decision Tree - CALIBRATED	No Scaling	0.009(0.0101)	0.5089(0.005)	0.5134(0.0049)
		-0.0091(0.0006)	0.5(0.0)	0.5046(0.0003)

Table B.142: Summary Results - Probability Prediction - CHD

		Training - Testing		
Model	Best Scaling	BSS	AUC ROC	AUC PR
GNB	No Scaling	-5.3871(0.5222)	0.9288(0.0149)	0.5243(0.0165)
		-5.1973(0.9614)	0.9299(0.0603)	0.5236(0.0643)
GNB - SMOTENC	No Scaling	-3.7386(0.3115)	0.8687(0.0304)	0.4676(0.035)
		-3.8137(0.6957)	0.9078(0.0728)	0.5083(0.0863)
GPC	NORM	-0.0086(0.0)	0.5(0.0)	0.5045(0.0)
		-0.0089(0.0)	0.5(0.0)	0.5045(0.0)
GPC - SMOTENC	STD	-2.2814(0.4333)	0.7599(0.0501)	0.3622(0.0674)
		-2.3148(0.8465)	0.7446(0.13)	0.3404(0.1682)
KNN - CALIBRATED	NORM	0.0049(0.0255)	0.5089(0.0139)	0.5083(0.0148)
		-0.0089(0.1019)	-0.0089(0.1019)	0.4186(0.2335)
LDA	POWER	-2.4921(0.1687)	0.7793(0.0189)	0.3788(0.0258)
		-2.3725(0.5544)	0.7868(0.0769)	0.387(0.0884)
LDA - SMOTENC	POWER	-6.3761(0.9235)	0.918(0.0371)	0.5082(0.0406)
		-6.2061(2.0458)	0.9113(0.1177)	0.5002(0.1235)
Logistic Regression	STD	0.0128(0.0456)	0.5161(0.0231)	0.5071(0.0256)
		-0.0089(0.1019)	0.5142(0.032)	0.4186(0.2335)
Cost Sensitive Logistic Regression	STD	0.0071(0.0351)	0.5089(0.0199)	0.5099(0.0122)
		-0.0377(0.0644)	0.4999(0.0003)	0.4045(0.2236)
MNB	No Scaling	-60.8519(28.2386)	0.6338(0.0908)	0.4117(0.0644)
		-60.7706(29.6569)	0.6257(0.1587)	0.4084(0.0821)
MNB - SMOTENC	No Scaling	-93.3953(14.1801)	0.5193(0.0525)	0.4354(0.0162)
		-93.2271(17.6402)	0.5094(0.1364)	0.434(0.0723)
QDA	NORM	-1.7511(0.3908)	0.638(0.0212)	0.2778(0.0499)
		-4.7072(0.7801)	0.8613(0.0618)	0.4494(0.0671)
QDA - SMOTENC	STD	-3.3554(0.4015)	0.877(0.032)	0.4825(0.0315)
		-3.5254(0.4512)	0.8949(0.0607)	0.496(0.0732)
SVM - CALIBRATED	POWER	0.0071(0.0315)	0.532(0.0237)	0.4803(0.0099)
		-0.0377(0.0644)	0.514(0.0317)	0.3686(0.2179)
Decision Tree - CALIBRATED	No Scaling	0.0205(0.0183)	0.526(0.019)	0.5053(0.0064)
		-0.0665(0.0789)	0.4997(0.0004)	0.3045(0.2739)

Table B.143: Summary Results - Probability Prediction - CNS

		Training - Testing		
Model	Best Scaling	BSS	AUC ROC	AUC PR
GNB	No Scaling	-4.388(0.4713)	0.9377(0.0184)	0.5427(0.0222)
		-5.1973(0.9614)	0.9299(0.0603)	0.5236(0.0643)
GNB - SMOTENC	No Scaling	-0.8806(0.1639)	0.8161(0.0296)	0.4957(0.0373)
		-3.8137(0.6957)	0.9078(0.0728)	0.5083(0.0863)
GPC	NORM	-0.019(0.0166)	0.5011(0.0025)	0.4895(0.0216)
		-0.0089(0.0)	0.5(0.0)	0.5045(0.0)
GPC - SMOTENC	POWER	-0.8728(0.2384)	0.8475(0.0287)	0.5262(0.0506)
		-1.9689(0.9206)	0.8169(0.0636)	0.4381(0.09)
KNN - CALIBRATED	NORM	0.0136(0.0419)	0.5656(0.0499)	0.4752(0.0237)
		-0.2122(0.3851)	0.5237(0.0328)	0.2909(0.225)
LDA	POWER	-1.7255(0.1441)	0.8408(0.0163)	0.4759(0.0215)
		-2.3725(0.5544)	0.7868(0.0769)	0.387(0.0884)
LDA - SMOTENC	No Scaling	-3.0554(0.9774)	0.8781(0.0386)	0.8781(0.0386)
		-6.8979(0.6867)	0.9648(0.0031)	0.5569(0.0045)
Logistic Regression	NORM	0.0147(0.0323)	0.5221(0.0237)	0.5104(0.019)
		-0.0089(0.0)	0.5(0.0)	0.5045(0.0)
Cost Sensitive Logistic Regression	STD	0.0253(0.0426)	0.5238(0.0344)	0.5153(0.0097)
		-0.0377(0.0644)	0.4999(0.0003)	0.4045(0.2236)
MNB	No Scaling	-20.3758(5.277)	0.7044(0.0423)	0.3259(0.0665)
		-60.7706(29.6569)	0.6257(0.1587)	0.4084(0.0821)
MNB - SMOTENC	No Scaling	-30.0413(7.6563)	0.7571(0.039)	0.4249(0.0631)
		-93.2271(17.6402)	0.5094(0.1364)	0.434(0.0723)
QDA	POWER	-3.7373(1.1312)	0.8843(0.0187)	0.4944(0.0351)
		-4.1307(1.0119)	0.8922(0.0738)	0.4881(0.0766)
QDA - SMOTENC	POWER	-1.0058(0.2426)	0.8157(0.0253)	0.4804(0.0397)
		-3.439(0.7163)	0.8528(0.0923)	0.449(0.1045)
SVM - CALIBRATED	STD	0.0007(0.0432)	0.53(0.031)	0.4822(0.037)
		-0.0665(0.0789)	0.5139(0.0318)	0.2686(0.252)
Decision Tree - CALIBRATED	No Scaling	0.0433(0.0801)	0.5305(0.0376)	0.5269(0.0426)
		-0.0606(0.1129)	0.5121(0.0278)	0.3674(0.2183)

Table B.144: Summary Results - Probability Prediction - GU

		Training - Testing		
Model	Best Scaling	BSS	AUC ROC	AUC PR
GNB	STD	-4.2064(0.2154)	0.9411(0.0104)	0.5516(0.0109)
		-4.1463(0.8535)	0.949(0.0285)	0.5587(0.0313)
GNB - SMOTENC	STD	-2.6096(0.1308)	0.8501(0.0141)	0.4669(0.0185)
		-2.5507(0.5596)	0.8481(0.0502)	0.4615(0.064)
GPC	NORM	-0.0111(0.0)	0.5(0.0)	0.5056(0.0)
		-0.0113(0.0)	0.5(0.0)	0.5056(0.0)
GPC - SMOTENC	POWER	-1.6211(0.2411)	0.7405(0.0435)	0.3553(0.0596)
		-1.4046(0.293)	0.7118(0.1099)	0.3192(0.1472)
KNN - CALIBRATED	POWER	0.0254(0.0362)	0.5543(0.0397)	0.4834(0.0317)
		-0.2122(0.3851)	0.499(0.0019)	0.305(0.2739)
LDA	POWER	-1.7996(0.1869)	0.7983(0.0168)	0.4227(0.0277)
		-1.6293(0.3237)	0.8204(0.0879)	0.4484(0.1123)
LDA - SMOTENC	STD	-4.6038(0.3992)	0.9234(0.0201)	0.5294(0.0193)
		-4.7081(0.6714)	0.9458(0.029)	0.5514(0.0312)
Logistic Regression	POWER	-0.0076(0.0425)	0.5585(0.0331)	0.4376(0.0437)
		-0.0113(0.0795)	0.5549(0.0004)	0.3439(0.1264)
Cost Sensitive Logistic Regression	POWER	-0.0032(0.0131)	0.5572(0.0348)	0.4594(0.0293)
		-0.0113(0.0795)	0.5439(0.0246)	0.3829(0.1426)
MNB	No Scaling	-23.165(25.269)	0.4926(0.0299)	0.1293(0.169)
		-16.9333(29.1752)	0.5032(0.0348)	0.1085(0.194)
MNB - SMOTENC	No Scaling	-29.5385(26.8746)	0.5239(0.0529)	0.2018(0.1389)
		-27.1809(30.1303)	0.5771(0.0817)	0.2482(0.1297)
QDA	POWER	-4.1391(0.2003)	0.9327(0.0158)	0.5426(0.0173)
		-4.0564(0.9232)	0.9385(0.0285)	0.5479(0.0318)
QDA - SMOTENC	POWER	-2.5106(0.1633)	0.8224(0.0261)	0.4384(0.0255)
		-2.5282(0.6386)	0.8263(0.0813)	0.4359(0.0976)
SVM - CALIBRATED	POWER	-0.0184(0.024)	0.5313(0.0283)	0.4511(0.0375)
		-0.0787(0.1005)	0.5216(0.0304)	0.2276(0.2177)
Decision Tree - CALIBRATED	No Scaling	0.0024(0.0108)	0.5097(0.0081)	0.5075(0.0078)
		-0.0606(0.1129)	0.5121(0.0278)	0.3674(0.2183)

Table B.145: Summary Results - Probability Prediction - MUSC

	Training - Testing		
CHD_CNS	**BSS**	**AUC ROC**	**AUC PR**
DT_CALIBRATED	0.5604(0.0642)	0.8245(0.045)	0.7744(0.0331)
	0.5892(0.1214)	0.8386(0.0777)	0.7861(0.0733)
LR	0.58(0.0265)	0.8262(0.0143)	0.7927(0.0127)
	0.6131(0.1619)	0.8304(0.0782)	0.8073(0.0817)
LR_CS	0.0881(0.0776)	0.9808(0.0078)	0.7683(0.0065)
	0.0913(0.1983)	0.9773(0.0199)	0.7518(0.0328)
SVM_CALIB	0.1693(0.0644)	0.6221(0.0215)	0.5037(0.0574)
	0.2225(0.1049)	0.6463(0.051)	0.5716(0.0881)

Table B.146: Summary Results - Probability Prediction - CHD - CNS

	Training - Testing		
CHD_GU	**BSS**	**AUC ROC**	**AUC PR**
DT_CALIBRATED	0.6723(0.0504)	0.8747(0.0378)	0.8344(0.0273)
	0.6236(0.1968)	0.8589(0.0443)	0.8035(0.1045)
LR	0.6492(0.0616)	0.8686(0.0193)	0.8276(0.0315)
	0.6752(0.1302)	0.8661(0.0663)	0.8416(0.0626)
LR_CS	0.2138(0.0546)	0.9754(0.0079)	0.7777(0.0161)
	0.2014(0.1211)	0.9725(0.0282)	0.7642(0.0133)
SVM_CALIB	0.1899(0.1116)	0.6601(0.0338)	0.5237(0.0859)
	0.3121(0.074)	0.712(0.0945)	0.6392(0.0237)

Table B.147: Summary Results - Probability Prediction - CHD - GU

	Training - Testing		
CHD_MUSC	**BSS**	**AUC ROC**	**AUC PR**
DT_CALIBRATED	0.5022(0.1312)	0.7838(0.0662)	0.7339(0.0778)
	0.5284(0.1463)	0.7967(0.0671)	0.7428(0.0878)
LR	0.5293(0.0744)	0.8139(0.0268)	0.7641(0.041)
	0.577(0.1906)	0.8165(0.0857)	0.7841(0.1137)
LR_CS	0.1298(0.0785)	0.9699(0.0086)	0.7607(0.0173)
	0.2073(0.2016)	0.9789(0.0194)	0.7707(0.0481)
SVM_CALIB	0.1512(0.0619)	0.6466(0.0116)	0.4961(0.0479)
	0.0986(0.0913)	0.6404(0.066)	0.4822(0.0543)

Table B.148: Summary Results - Probability Prediction - CHD - MUSC

	Training - Testing		
CNS_GU	**BSS**	**AUC ROC**	**AUC PR**
DT_CALIBRATED	0.5555(0.0883)	0.8092(0.0529)	0.7709(0.0486)
	0.5976(0.133)	0.8325(0.0528)	0.7847(0.076)
LR	0.6718(0.0496)	0.871(0.0119)	0.8415(0.0249)
	0.7293(0.1587)	0.8797(0.0692)	0.867(0.0792)
LR_CS	0.2759(0.0723)	0.9901(0.0044)	0.804(0.0139)
	0.2826(0.195)	0.9799(0.0191)	0.7839(0.0483)
SVM_CALIB	0.0904(0.0323)	0.6291(0.0222)	0.4456(0.0365)
	0.1304(0.1086)	0.6383(0.028)	0.4636(0.0875)

Table B.149: Summary Results - Probability Prediction - CNS - GU

	Training - Testing		
CNS_MUSC	**BSS**	**AUC ROC**	**AUC PR**
DT_CALIBRATED	0.4015(0.0944)	0.7318(0.0558)	0.6759(0.0551)
	0.5168(0.1128)	0.8043(0.0365)	0.743(0.063)
LR	0.5468(0.0207)	0.8311(0.0153)	0.7726(0.0146)
	0.5778(0.1131)	0.8435(0.051)	0.7858(0.0624)
LR_CS	0.1838(0.0519)	0.9768(0.0081)	0.7736(0.0148)
	0.2343(0.1872)	0.9731(0.0299)	0.7696(0.0502)
SVM_CALIB	0.1052(0.0491)	0.6163(0.0312)	0.4553(0.0457)
	0.0099(0.2112)	0.6145(0.036)	0.404(0.1439)

Table B.150: Summary Results - Probability Prediction - CNS - MUSC

	Training - Testing		
GU_MUSC	**BSS**	**AUC ROC**	**AUC PR**
DT_CALIBRATED	0.5101(0.0773)	0.7949(0.0442)	0.7488(0.0388)
	0.545(0.1616)	0.8338(0.057)	0.7557(0.0942)
LR	0.6075(0.0305)	0.8526(0.0126)	0.8058(0.0192)
	0.5893(0.1496)	0.8341(0.0435)	0.7883(0.0844)
LR_CS	0.2459(0.0441)	0.9695(0.0088)	0.78(0.0153)
	0.2436(0.1485)	0.9735(0.0269)	0.772(0.0321)
SVM_CALIB	0.1386(0.0394)	0.6134(0.0225)	0.4863(0.0354)
	0.172(0.0499)	0.6387(0.0667)	0.5234(0.0476)

Table B.151: Summary Results - Probability Prediction - GU - MUSC

Bibliography

[1] *World Health Organization on Congenital Anomalies.*

[2] M. Tobollik, O. Razum, D. Wintermeyer, and D. Plass, "Burden of outdoor air pollution in kerala, india—a first health risk assessment at state level," *International Journal of Environmental Research and Public Health*, vol. 12, p. 10602–10619, Aug 2015.

[3] G. Hoek, R. M. Krishnan, R. Beelen, A. Peters, B. Ostro, B. Brunekreef, and J. D. Kaufman, "Long-term air pollution exposure and cardio- respiratory mortality: a review," *Environmental Health*, vol. 12, p. 43, Dec. 2013.

[4] R. Liang, B. Zhang, X. Zhao, Y. Ruan, H. Lian, and Z. Fan, "Effect of exposure to PM2.5 on blood pressure: a systematic review and meta-analysis," *Journal of Hypertension*, vol. 32, pp. 2130–2141, Nov. 2014.

[5] L. Xiong, Z. Xu, H. Wang, Z. Liu, D. Xie, A. Wang, and F. Kong, "The association between ambient air pollution and birth defects in four cities in Hunan province, China, from 2014 to 2016," *Medicine*, vol. 98, p. e14253, Jan. 2019.

[6] J. Ren, D. Friedmann, J. Xiong, C. D. Liu, B. R. Ferguson, T. Weerakkody, K. E. DeLoach, C. Ran, A. Pun, Y. Sun, B. Weissbourd, R. L. Neve, J. Huguenard, M. A. Horowitz, and L. Luo, "Anatomically Defined and Functionally Distinct Dorsal Raphe Serotonin Sub-systems," *Cell*, vol. 175, pp. 472–487.e20, Oct. 2018.

[7] Schembari Anna, de Hoogh Kees, Pedersen Marie, Dadvand Payam, Martinez David, Hoek Gerard, Petherick Emily S., Wright John, and Nieuwenhuijsen Mark J., "Ambient Air Pollution and Newborn Size and Adiposity at Birth: Differences by Maternal Ethnicity (the Born in Bradford Study Cohort)," *Environmental Health Perspectives*, vol. 123, pp. 1208–1215, Nov. 2015.

[8] E. A. L. Gianicolo, C. Mangia, M. Cervino, A. Bruni, M. G. Andreassi, and G. Latini, "Congenital anomalies among live births in a high environmental

risk area—A case-control study in Brindisi (southern Italy)," *Environmental Research*, vol. 128, pp. 9–14, Jan. 2014.

[9] E. K.-C. Chen, D. Zmirou-Navier, C. Padilla, and S. Deguen, "Effects of Air Pollution on the Risk of Congenital Anomalies: A Systematic Review and Meta-Analysis," *International Journal of Environmental Research and Public Health*, vol. 11, pp. 7642–7668, Aug. 2014.

[10] C. Wang, S. I. Bangdiwala, S. Rangarajan, S. A. Lear, K. F. AlHabib, V. Mohan, K. Teo, P. Poirier, L. A. Tse, Z. Liu, A. Rosengren, R. Kumar, P. Lopez-Jaramillo, K. Yusoff, N. Monsef, V. Krishnapillai, N. Ismail, P. Seron, A. L. Dans, L. Kruger, K. Yeates, L. Leach, R. Yusuf, A. Orlandini, M. Wolyniec, A. Bahonar, I. Mohan, R. Khatib, A. Temizhan, W. Li, and S. Yusuf, "Association of estimated sleep duration and naps with mortality and cardiovascular events: a study of 116 632 people from 21 countries," *European Heart Journal*, vol. 40, pp. 1620–1629, May 2019.

[11] A. Waked and C. Afif, "Emissions of air pollutants from road transport in Lebanon and other countries in the Middle East region," *Atmospheric Environment*, vol. 61, pp. 446–452, Dec. 2012.

[12] N. Saliba, S. Moussa, H. Salame, and M. El-Fadel, "Variation of selected air quality indicators over the city of Beirut, Lebanon: Assessment of emission sources," *Atmospheric Environment*, vol. 40, pp. 3263–3268, June 2006.

[13] W. van der Aalst, *Process Mining*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016.

[14] Y. Luo, Z. Li, H. Guo, H. Cao, C. Song, X. Guo, and Y. Zhang, "Predicting congenital heart defects: A comparison of three data mining methods," *PLOS ONE*, vol. 12, pp. 1–14, 05 2017.

[15] Mu, Yu, Feng, Kai, Yang, Ying, and Wang, Jingyuan, "Applying deep learning for adverse pregnancy outcome detection with pre-pregnancy health data," *MATEC Web Conf.*, vol. 189, p. 10014, 2018.

[16] *Maternal Exposure to Ambient PM 10 During Pregnancy Increases the Risk of Congenital Heart Defects: Evidence From Machine Learning Models*.

[17] J. Brownlee, *Imbalanced Classification with Python: Better Metrics, Balance Skewed Classes, Cost-Sensitive Learning*. Machine Learning Mastery, 2020.

[18] *Learning from Imbalanced Data Sets*. Springer Nature Switzerland AG: Springer International Publishing, 2018.

[19] M. Zieba, J. M. Tomczak, M. Lubicz, and J. Swiatek, "Boosted svm for extracting rules from imbalanced data in application to prediction of the post-operative life expectancy in the lung cancer patients," *Applied Soft Computing*, vol. 14, pp. 99 – 108, 2014. Special issue on hybrid intelligent methods for health technologies.

[20] Y. Park, L. Luo, K. K. Parhi, and T. Netoff, "Seizure prediction with spectral power of eeg using cost-sensitive support vector machines," *Epilepsia*, vol. 52, no. 10, pp. 1761–1770, 2011.

[21] P. Cao, J. Yang, W. Li, D. Zhao, and O. Zaiane, "Ensemble-based hybrid probabilistic sampling for imbalanced data learning in lung nodule cad," *Computerized Medical Imaging and Graphics*, vol. 38, no. 3, pp. 137 – 150, 2014.

[22] U. R. Acharya, P. Chowriappa, H. Fujita, S. Bhat, S. Dua, J. E. Koh, L. Eugene, P. Kongmebhol, and K. Ng, "Thyroid lesion classification in 242 patient population using gabor transform features from high resolution ultrasound images," *Knowledge-Based Systems*, vol. 107, pp. 235 – 245, 2016.

[23] S. Garg, A. K. Sarje, and S. K. Peddoju, "Improved detection of p2p botnets through network behavior analysis," in *Recent Trends in Computer Networks and Distributed Systems Security* (G. Martínez Pérez, S. M. Thampi, R. Ko, and L. Shu, eds.), (Berlin, Heidelberg), pp. 334–345, Springer Berlin Heidelberg, 2014.

[24] C. Márquez-Vera, A. Cano, C. Romero, A. Y. M. Noaman, H. Mousa Fardoun, and S. Ventura, "Early dropout prediction using data mining: a case study with high school students," *Expert Systems*, vol. 33, no. 1, pp. 107–124, 2016.

[25] *Clustering on mixed type data.*

[26] *Clustering with optimised weights for Gower's metric.*

[27] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural Computation*, vol. 10, pp. 1895–1923, Oct. 1998.

[28] *SHAP (SHapley Additive exPlanations).*

[29] D. A. Chambers, W. G. Feero, and M. J. Khoury, "Convergence of implementation science, precision medicine, and the learning health care system," *JAMA*, vol. 315, p. 1941, May 2016.

[30] "Why does the shift from "personalized medicine" to "precision health" and "wellness genomics" matter?," *AMA Journal of Ethics*, vol. 20, pp. E881–890, Sept. 2018.

[31] A. Shaban-Nejad, M. Michalowski, and D. L. Buckeridge, "Health intelligence: how artificial intelligence transforms population and personalized health," *npj Digital Medicine*, vol. 1, Oct. 2018.

[32] A. Shaban-Nejad and M. Michalowski, eds., *Precision Health and Medicine*. Springer International Publishing, 2020.

[33] *Rocke Feller*.

[34] *Vast Biome*.

[35] L. Mueller, P. Berhanu, J. Bouchard, V. Alas, K. Elder, N. Thai, C. Hitchcock, T. Hadzi, I. Khalil, and L.-A. Miller-Wilson, "Application of machine learning models to evaluate hypoglycemia risk in type 2 diabetes," *Diabetes Therapy*, vol. 11, pp. 681–699, Feb. 2020.

[36] *COVID-19 and AI*.

[37] M. Uddin, Y. Wang, and M. Woodbury-Smith, "Artificial intelligence for precision medicine in neurodevelopmental disorders," *npj Digital Medicine*, vol. 2, Nov. 2019.

[38] C. Krittanawong, H. Zhang, Z. Wang, M. Aydar, and T. Kitai, "Artificial intelligence in precision cardiovascular medicine," *Journal of the American College of Cardiology*, vol. 69, pp. 2657–2664, May 2017.

[39] W. J. Hopp, J. Li, and G. Wang, "Big data and the precision medicine revolution," *Production and Operations Management*, vol. 27, pp. 1647–1664, June 2018.

[40] R. Mirnezami, J. Nicholson, and A. Darzi, "Preparing for precision medicine," *New England Journal of Medicine*, vol. 366, pp. 489–491, Feb. 2012.

[41] Y. J. Kim, B. P. Kelley, J. S. Nasser, and K. C. Chung, "Implementing precision medicine and artificial intelligence in plastic surgery," *Plastic and Reconstructive Surgery - Global Open*, vol. 7, p. e2113, Mar. 2019.

[42] *Google Trends*.

[43] *Insight RX*.

[44] *Foundation for Precision Medicine*.

[45] *Tempus.*

[46] *Benevolent AI.*

[47] *Emory University Study.*