AMERICAN UNIVERSITY OF BEIRUT

APPLICATION OF MULTI-LABEL CLASSIFICATION IN PROJECT MANAGEMENT FOR RISK IDENTIFICATION

by
YASER OMAR SHAMSEDDINE

A thesis
submitted in partial fulfillment of the requirements
for the degree of Master of Engineering
to the Department of Industrial Engineering and Management
of the Maroun Semaan Faculty of Engineering and Architecture
at the American University of Beirut

Beirut, Lebanon
September, 2020

AMERICAN UNIVERSITY OF BEIRUT

APPLICATION OF MULTI-LABEL CLASSIFICATION IN PROJECT MANAGEMENT FOR RISK IDENTIFICATION

by

YASER OMAR SHAMSEDDINE

Approved by:

Sept. 2, 2020

Dr. Jimmy Azar, Assistant Professor — Advisor
Department of Industrial Engineering and Management

Maher Nouiehed

Sept. 4, 2020

Dr. Maher Nouiehed, Assistant Professor — Member of Committee
Department of Industrial Engineering and Management

Nadine Marie Moacdieh

Sept. 6, 2020

Dr. Nadine Moacdieh, Assistant Professor — Member of Committee
Department of Industrial Engineering and Management

Date of thesis/dissertation defense: September 01, 2020

AMERICAN UNIVERSITY OF BEIRUT

THESIS, DISSERTATION, PROJECT RELEASE FORM

Student Name:
_____________Omar_________Shamseddine________________Yaser____________
Middle Last First

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.

Yaser Shamseddine_______________ September 07, 2020__________________
Signature Date

This form is signed when submitting the thesis, dissertation, or project to the University Libraries

ACKNOWLEDGMENTS

First and foremost, I thank the almighty God to whom I owe my existence for providing me the health, strength, and capability to proceed successfully. Second, with immense gratitude and profound, sincere thanks to my advisor Dr. Jimmy Azar, whose guidance, support, and encouragement have been invaluable throughout this research. My appreciation to him for putting his trust in me even with his knowledge of my different background. Third, I would also like to thank my thesis committee members, Dr. Maher Nouiehed and Dr. Nadine Moachdieh, for their assistance and motivation while pursuing my research.

Special thanks to the Department of Industrial Engineering and Management with its chairperson Dr. Bacel Maddah to grant me the opportunity to proudly be one of the AUB community members and fuel my ambition to attain my desires.

Finally, with deep gratitude, I would like to acknowledge my family members for being my backbone, guiders, and support system throughout my ever life.

Thank you.

AN ABSTRACT OF THE THESIS OF

Yaser Omar Shamseddine for: Master of Engineering
Major: Engineering Management

Title: Application of Multi-Label Classification in Project Management for Risk Identification

Artificial Intelligence (AI) and Machine Learning (ML) have undoubtedly been rising technologies, and it is expected that their prevalence will only continue to increase. These technologies have changed the way of doing business in various industries, and project management is not an exemption. AI technology will strongly influence throughout its breakthroughs the future of project management in how its tasks and milestones will be delivered and controlled.

The goal of this study is to assist project managers in better-identifying their project risks at the milestone level in complex projects to optimize success rates. The process is steeped in utilizing machine learning algorithms that would accurately identify problem types and facilitate project risk analysis. The contribution of this work is two-fold: (1) we present a dataset that can serve as a benchmark for project management risk assessment in the absence of a publicly available dataset at the time of writing this thesis, and (2) we present a proof-of-concept for the applicability and use ML methods in risk assessment using this dataset.

As such, the research project starts with an overview of how AI will heavily influence the future of project management, in addition to the evolution of AI in the discipline of project management. Furthermore, the research project identifies AI potential risks and limitations. Following this, we envision a dataset that serves as a clarificatory template for risk identification via ML. The data was set up in tabular format where each data row represents a milestone associated with data variables. Subsequently, we introduced patterns into the dataset and identify problem types manually based on specific criteria. To our knowledge, there is no publicly available dataset on project management milestone/projects and their associated problem types. Therefore, the annotated dataset we created in this work serves as a benchmark for assessing risk and for future effort in this area. The dataset will be made publicly available. As a proof-of-concept, two suitable machine learning models, each utilizing a different classification algorithm such as Decision Tree (DT) and Support Vector Machine (SVM), were trained using the dataset for predicting potential problem types. Lastly, we examined both models' performance through a test set and compare them by employing confusion matrices and various associated ML performance measures.

The evaluation performance metrics outcomes proved that the DT model outperformed the SVM model for the dataset examined.

Keywords: Artificial Intelligence, Machine learning, Project management, Project risks, Milestones, Risk identification, Classification, Decision Trees, and Support Vector Machines.

CONTENTS

ILLUSTRATIONS

TABLES

CHAPTER I

OBJECTIVE

This work aims to assist project managers in better-identifying and analyzing risks on the milestone level in complex projects to increase project success rates. Previously, the techniques employed in risk management had promoted an optimal ideal that did not hold up when scrutinized thereby creating the need for introducing an alternative approach. The objective here becomes utilizing machine learning algorithms to predict project risks based on historical data of previous projects data.

We will show this by generating a dataset that includes multiple milestones, where each milestone includes eleven features that will be considered as the independent variables in the dataset. These independent variables will be set based on research and experience. Based on these aforementioned variables we will be in a better position to identify problem types. The problem types are determined by the results of our independent variables and as such are dependent variables. The milestones will then cede a number of problem types be they zero, one or several. Thereby, the function of the algorithm becomes one where the project manager identifies those features for a new milestone in a project, the algorithm can then identify whether this milestone contains a problem or not, and will identify the problem type(s). Finally, we will show through the literature review and statistics how AI technology has become invaluable for project management practices and how it is transforming the discipline of project management.

CHAPTER II

RATIONALE

As risk management has been considered one of the most critical areas in project management, and as risk identification and analysis phases have been considered the most essentials for their direct effect on project success, this study focuses on identifying risks in projects while adopting the new AI approach. The current study attempts to exploit the power of AI in using machine learning techniques to simplify risk management practices as the size and complexity of projects have risen, and as market competition has increased.

CHAPTER III

INTRODUCTION AND LITERATURE

A. Project Management

A project is a temporary endeavor undertaken to create a unique product, service, or result (*A Guide to the Project Management Body of Knowledge (PMBOK Guide)* [PMBOK], 2017, p. 4). A Project's primary purpose is to fulfill the organization's objectives that move it from its current state into a new one while achieving business value to stakeholders. Organizations struggle for project success through the adoption of traditional project management tools and measures.

Traditionally, a project has been categorized as successful if it accomplished the Triple Constraint: scope, budget and schedule (Pinto & Slevin, 1988). However, those gears are no longer sufficient in our competitive environment. *PMI's Pulse of the Profession* [PPP] (2019), showed that engaging executive sponsors, aligning projects to organizational strategies, and having control over scope creep are factors of utmost importance that indicates potential project success. Yet despite all the talk, project performance isn't getting any better (PPP, 2019). This is a disquieting state that has a considerable impact on the economy, where 9.9% of every dollar is wasted due to poor project management (PPP, 2018). Thus, a new ingredient should be incorporated into the project management traditional tools and practices, which unsurprisingly would be the use of Artificial Intelligence (AI).

B. Artificial Intelligence

Over the previous decade, AI and machine learning have been popular words across our business surroundings. The world of AI is entering our businesses areas without any exception, and definitely, AI will have a great impact on the course of project management. AI professionals are predicting that AI will change the means we produce, manufacture, and deliver (Marr, 2016). The Project Management Institute [PMI] (2018), ranked AI as the third top factor that will affect the discipline of project management after Cloud Computing and Internet of Things which are ranked as a first and second factor respectively, Figure 1: Top 3 Disruptors used for competitive advantag.

Figure 1: Top 3 Disruptors used for competitive advantage

Note. From “Next practices: Maximizing the benefits of disruptive technologies on projects,” Pulse of the Profession, 2018, p. 4.

A research held by Al Najjar & Al-Sarraj (2019), revealed that 85 percent of respondents of its CEOs 2019 survey agreed that AI will significantly change the way they do business in the next five years. Deloitte (2018), mentioned that 83 percent of AI adopters recognized benefits on the return on investment that were rated either “moderate” or “substantial”. Besides, Gartner, Inc. (2019), predicts that by 2030, 80 percent of today's project management tasks will be eliminated as AI takes over. Globally, the overall sentiment is that AI will be a catalyst for transformation across regions (Al Najjar & Al-Sarraj, 2019).

So, what is AI? "Artificial Intelligence is the designing and building of intelligent agents that receives percepts from the environment and takes actions that affect that environment." (Russell & Norvig, 2016). Although the buzz word "Artificial Intelligence" is being widely used in the last couple of years, however, it dates back to year 1956 where the term "Artificial Intelligence" was formally coined by John McCarthy at a two-month workshop at Dartmouth College in Hanover, New Hampshire (Russell & Norvig, 2016).

1. Evolution of AI in Project Management

AI main ideas converge to the point that machines could have once the ability to learn by themselves without being progressively fed or explicitly programmed by a human. AI will evolve from simple task automation to predictive project analytics, advice, and actions (Lahmann, Keiser, & Stierli, 2018). The authors continued; however, AI cannot be a human. They lighted on the four phases in the evolution of AI, in the discipline of project management, which are consecutively listed in the report as integration and automation, chatbots assistants, machine learning-based project management, and autonomous project management Figure 2: Evolution of AI in Project Management.

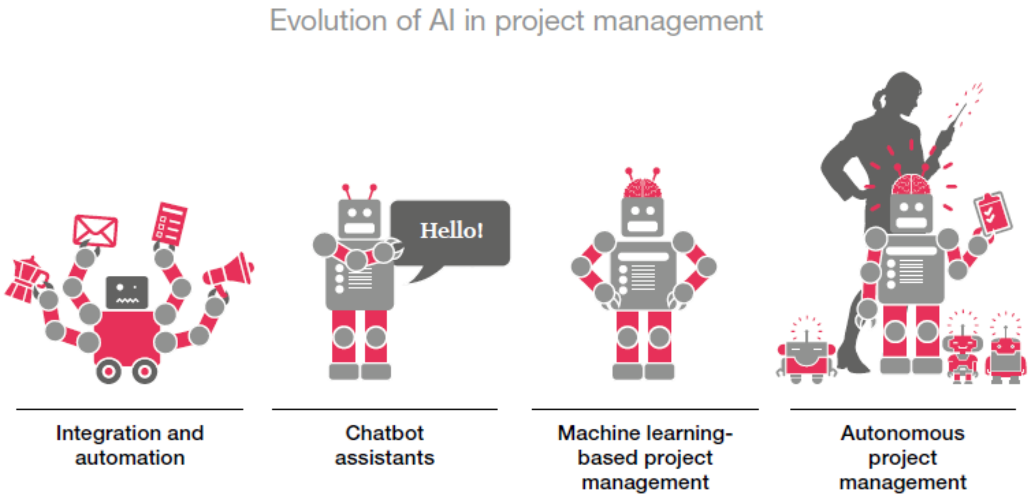

Figure 2: Evolution of AI in Project Management

Note. From "AI will transform project management. Are you ready?" by M. Lahmann, P. Keiser, and A. Stierli, 2019, p. 3.

Many project management activities have already been integrated into the phase of integration and automation, where different practices such as streamlining and automating standardized project tasks through workflow integration and process automation have been adopted. Some activities are related to updating budget forecast reports upon project budget updates, auto-scheduling, tracking, and alerting by embracing predetermined programs and rules. Current real-life practices include the use of online templates and workflows, sending alerts when recognizing budgeting or scheduling issues, and the collaboration between MS Project Online and Wunderlist for task creation and scheduling. The resulting actions offer project managers the ability to devote their time to more valuable activities that would increase efficiency and reduce costs.

AI chatbots incorporation is set as the second phase in the mentioned evolution, where bots will serve project managers as their assistants in speech and text recognition.

Chatbots can carry administrative actions as organizing meetings, reminding project team members of scheduled activities, plan versus progress checks. Also, it can include some intuitions on the available data. Some real well-noticed cases include “Fireflies.ai,” which generates notes and distinguishes tasks and assignments based on conversations processing, and “Stratejos.ai,” which notifies team members, monitors their performance, and helps project managers, based on specific measurables, to distinguish top contributors. Accordingly, as the first phase, chatbots assistants will eliminate various project management elementary tasks and allows project managers to emphasis on highly valued tasks.

Machine learning-based project management is considered as the third phase in the evolution, where machine learning is introduced into project management exercises. Machine learning will empower predictive analytics and will provide project managers with expert judgments, recommendations, and risk management support based on what worked in past projects. Some real-life examples include altering scheduling views based on user consent and preferences, identifying the best team for a task, predicting the expected net promoter score (NPS), and predicting the write-off for projects. Machine learning prediction efficiency will predict the future for project managers and allow them to anticipate what might happen in a project and what a project status might be. Predictive analytics in projects is considered as the most influential innovation in project management in the next ten years.

Autonomous project management is the fourth phase in the evolution of AI in the project management discipline, where only limited intervention of project managers or humans is required. This phase is the summation of the three previously mentioned phases and the mastering of the comprehensive project environment and stakeholders.

Therefore, sentimental analysis algorithms for understanding customer satisfaction and managing communications should be applied. Presently, there are no real-life use cases that support autonomous project management. Lahmann et al. (2018), noted that for the next 10-20 years, there would be no self-driven autonomous project managers for the reason of human control on project budgets and portfolios to manage the risk of autonomous investment decision. The first two phases were classified as weak or simple AI, whereas the second two phases were classified as advanced AI.

2. AI Potential Risks

As the new technology is invading our businesses on various levels, it is inevitable to understand its potential risks and limitations. Unfortunately, AI predictions and diagnosis never come without any risk. A human or a project manager should never trust an algorithm without an explanation or understanding of whether this algorithm is following the right process or not. Moreover, a human should have the ability to test the algorithm along different risk dimensions in case the human would disagree with the possible outcomes. In order to be capable of challenging those results, with no doubt, AI risk dimensions, as defined by Lahmann & Stierli (2019), should be understood, where safeguards can be created. They identified ten risk dimensions listed as (1) Security, (2) Privacy, (3) Autonomy, (4) Employment, (5) Accountability, (6) Power/Inequality, (7) Justice/Bias, (8) Diversity, (9) Human vigor, (10) Wisdom. Detailed information is recorded in Table 1: Key Dimensions of AI Risk.

Table 1: Key Dimensions of AI Risk

AI risk dimension	Description
Security	AI technology may not follow your company's security standards. Self-learning machines might have the wrong parameter settings or come to the wrong conclusions.
Privacy	Artificial intelligence cannot appropriately distinguish between approved and restricted data, and therefore violates the right of privacy.
Autonomy	AI technology becomes so dominant that people feel like 'slaves' to the machine. As a result, the machine-based learnings of AI may become decoupled from a project manager's sense of what is right and wrong.
Employment	Despite the fact that legal standards for terms of employment must be built into AI algorithms, AI might still regard workers as a means to an end without considering the broader context.
Accountability	The legal accountability of AI-based decisions is currently far from being settled, and has yet to be resolved.
Power/Inequality	Incomplete and missing data can reduce the statistical power of a prediction and produce biased estimates leading to invalid conclusions.
Justice/Bias	AI might include irrelevancies and bias (e.g. discriminatory name filters in an application process due to decision-making that was already biased in the past) in the decision-making process.
Diversity	AI might come to the wrong conclusions when interpreting historical data for forecasts (e.g. underrepresentation of certain nationalities/names as a misconceived predictor of their future potential).

Human vigor	Robots lack the natural instinct and therefore do not have the same urge as humans to think outside the box.
Wisdom	The direction of AI's development is reliant on a set of 'valuable, fragile, and hard-won human wisdom'. Whereas scholars agree that AI should evolve towards a more meaningful, human-like existence, there is still a widespread fear of a completely unconstrained autonomy of AI exceeding the barriers we create for it.

Note. From "How can we prevent project management from falling into the AI darkness?" by M. Lahmann and A. Stierli, 2019, pp. 6,7.

Among the mentioned risks, often, biased data took the highest priority in an AI-based project management system environment, for the reason that human cognitive bias would be inherited right away into AI through data, algorithms, and interaction. Moreover, the authors focused on three main biased pitfalls when using AI in project management. Firstly, unavailability or incompleteness of data, secondly, biased data used to train AI, and finally, reliance, redundancy, and inconsistency of data.

Regarding the first pitfall, incomplete data would reduce the statistical power of estimations leading to bias and worthless conclusions. This pitfall would occur because the information in projects derives from various knowledge areas such as schedule, cost, and resource project management that are recorded in different documents and formats. More often, projects information is not stored in well-organized structured forms.

Regarding the second pitfall, different players in the environment of project management generate biased data due to different experiences and thoughts. Moreover, every project is unique in its purpose and context. As a result, different types of data biases could be used in training AI-based projects. Different types of biased data include implicit stereotypes, priming due to perceptual identification in the memory of project managers from the past, confirmation bias that occurs as a result of the direct influence of desire on beliefs, gambler's fallacy, bandwagon effect, selective perception, and observational selection.

Regarding the third pitfall, redundancy and inconsistency would be inherited in a dataset even though some technologies provide methods that can deal with unstructured and massive data. Data inconsistency exists where data sources conflict with each other at the data value level because the same data exists in different formats in multiple tables. However, they continued; on the other hand, data redundancy occurs when the same piece of data exists in multiple places in a set of data. This kind of bias deceives ML algorithms, where an algorithm receives data without knowing if it is either reliable, accurate data without redundancy and inconsistency, or not.

AI risks and its limitations will be imposing weighty challenges on project managers for using data in the right context. Besides, risk literacy, statistical thinking, and data science are not critical areas in the widely recognized certificates of project, program, and portfolio management. Gerd Gigerenzer (2012), a German psychologist, said "Experts without risk literacy skills are part of the problem rather than the solution" as he defined risk literacy as the ability to deal with uncertainties in an informed way. He said, "Without it, people jeopardize their health and money and can be manipulated into experiencing unwarranted, even damaging hopes and fears." Moreover, the author suggested that statistical thinking is the ability to understand and critically evaluate uncertainties and risks. The importance of understanding AI risk, and its restrictions put risk literacy on the front interests of project managers. Lahmann & Stierli (2019), also mentioned in their article the controversy around ethical compliance and AI systems. The authors refer to the overruling of human values to advanced AI, machine learning-based project management, and autonomous project management, that would be used in the project management disciplines. AI-driven programs potentially implement risky actions without inferring if these actions are moral or immoral, right or wrong, or acceptable or unacceptable.

C. Project Risks Management

All projects are risky since they are unique undertakings with varying degrees of complexity that aim to deliver benefits. They do this in a context of constraints and assumptions while responding to stakeholder expectations that may be conflicting and changing (PMBOK, 2017, p. 397). Managing project risks is becoming a growing area of concern as project sizes, and complexity is increasing, and as competition between businesses is growing incessantly. Royer (2000), proposes that unmanaged or unmitigated risks are the primary driving factors of project failure. PMI (2017), suggested that risk exists within every project at two levels, “Individual project risk” and “Overall project risk,” where it defines an individual project risk in its PMBOK as “an uncertain event or condition that, if it occurs, has a positive or negative effect on one or more project objectives.” On the other hand, it defines the overall project risk as “the effect of uncertainty on the project as a whole, arises from all sources of uncertainty including individual risks, representing the exposure of stakeholders to the implications of variations in project outcome, both positive and negative.” (PMBOK, 2017, p. 397). Through this research, we will be using individual project risks like the ones that would be encountering our project activities and milestones.

Project risk management is a set of processes that aim to achieve project success through proper identification, analysis, response, and monitoring of project risks in a well and adequately systematic approach. Different guides and risk literature (Raftery, 2003; Maytorena, Winch, Freeman, & Kiely, 2007; Lester, 2006; PMI, 2017) consent on the idea that risk management is divided into several processes that aim to upsurge the likelihood of project success. Risk management processes could be generally listed as (1) Risk Identification, (2) Risk Analysis, (3) Risk Response, and (4) Risk Monitoring. The risk identification process will be of our most interest in this research.

Although each project is unique, however, it has been documented by PMI that almost all projects follow a typical life cycle (PMBOK, 2017, p. 19). As been identified by PMI, projects pass through a series of four phases from its start to its completion, identified as (1) Starting the Project, (2) Organizing and Preparing, (3) Carrying Out the Project, and (4) Ending the Project, (PMBOK, 2017, p. 18). These phases establish a basic framework where projects could be managed, and risk resources could be identified. Cohen & Palmer (2004), stated that for any project to meet its goals, risk must be managed and integrated through the overall project management approach. As well, PMI suggests that managing risk and applying its processes should be performed throughout the project, thus, throughout the prelisted project phases (PMBOK, 2017, p. 395).

Risk identification and risk analysis processes have been identified by research as the most important among overall project risk management processes as they can immensely affect the precision of the risk assessment exercises (Chapman, 1998; Bajaj, Oluwoye, & Lenard, 1997; Chapman, 2001). However, the risk analysis process acquired the extensive work of research, while the identification process has had a little rigorous evaluation (Chapman, 2018). Also, Williams (1995), noted that risk identification had been subjected relatively to a small intense of research work. If risks are not identified, they cannot be analyzed and managed (Maytorena et al., 2007).

As this research is going to deal with the process of risk identification in a manner where a machine learning algorithm will be adopted, it is vital to have an overlook of previous methods that been used in risk identification.

Chapman (2001), displays techniques that can be used for the risk identification process such as brainstorming, Nominal Group Technique (NGT), Delphi, and historical records. Maytorena et al. (2007), mentioned in their literature review for project risk identification that over the past decades, brainstorming, checklists, and interview sessions have been commonly used as tools and techniques for identifying risks. However, risk registers and risk breakdown structures (RBS) were recognized as assisting tools utilized in risk identification.

Brainstorming is a technique used for recognizing ideas or potential risks in a period in an environment of experienced practitioners. After risks have been identified, further analysis is progressed. A checklist is a list of items or actions to be considered, and it is often used as a reminder. Risk checklists depend significantly on previous information and knowledge that has been developed from analogies (PMBOK, 2017, p. 414). An interview is a way used to identify risks by talking to experts, stakeholders, and project participants directly. Interviewees favored a confidential environment where honesty is guaranteed. NGT is a modified brainstorming technique with a voting process for ranking the most useful ideas either for prioritization or for further brainstorming. The Delphi technique is an information-gathering technique where participants participate anonymously. Experts use it in order to reach a consensus when dealing with a particular matter. Previous records can help team members identify risks based on their previous risk agreements.

Regarding the risk identification assisting tools, a risk register is a document that includes details on all identified risks through a project. A risk register may contain limited or extensive information of risks as risk analysis, risk owners, causes, effects, probability, and impacts based on the complexity and variables of a project (PMBOK, 2017, p. 417). On the other hand, RBS is a hierarchical representation of potential project sources that assists project team members to consider the full range of risk sources in a project (PMBOK, 2017 p. 405).

Charette (1989), pointed out that historical information is the best fundamental source for identifying risks. Risk identification techniques depend on the historical information and experience of team members considerably. A successful project manager on both sides, organizationally and personally, is a manager who can learn from experience and better from others' experience (Royer, 2000). The author continued to say in his research that a significant contributor to the determination for uninitiated project managers was their naivete about risk management. Although Maytorena et al. (2007), came to a result which claims that the role of experience in the risk identification process has a slightly significant impact on project risk management, however, experience and risks encountered in previous projects and analogies would argue its importance in future projects success criterion.

In their research, Cohen & Palmer (2004) identified the most common sources of risk in projects. The following sources were recognized as (1) changes in project scope and requisition, (2) design errors and omissions, (3) inadequately defined roles and responsibilities, (4) inaccurate cost and schedule estimates, (5) insufficient skilled staff, (6) force majeure, and (7) new technology. Further information and analysis of the preceded risk sources were mentioned in the research.

D. Machine Learning

In this research paper, our goal is to predict potential problems in a project at the milestone level based on the project manager's previous experience. ML, which is one of the most powerful tools used in the predictive data analytics area, was embraced in this research.

ML is a core subarea in the field of AI. It focuses on analyzing and interpreting patterns and structures in data to enable predicting useful insights for making repeated decisions outside of human interaction. ML algorithms possess the ability to deal with data variables having different relationships as linear, non-linear, complex high-order, and even disjunctive ones. Moreover, ML utilize optimization techniques to attain better prediction performance by increasing model predictive accuracy.

With the continuous generation of data, ML has become an essential technique for solving real business problems in various fields like construction, retail, banking, transportation, healthcare, etc. Machine learning algorithms can always improve with continuous exposure to data. In this study, ML was used to analyze historical project data to identify potential risks before they might occur.

The classification technique we have used is of the supervised learning type. This technique depends on providing a labeled dataset that facilitates the algorithm's process to learn and map new input records (e.g., milestone characteristics) into particular dependent output classes (e.g., problem types, no problem) based on the actual labels of the data. In this technique, an ML model is trained to learn and detect underlying patterns and relationships, enabling it to yield good results once presented with never previously seen data. This study focused on using classification technique to predict and identify milestones that would carry problems before their execution time.

The DT and SVM, two classification algorithms, were applied in this study. The rationale behind choosing a DT algorithm was that it is proven to be robust against noisy data and able to learn disjunctive variable relations. It presents excellent performance with small datasets (Gondia et al., 2019). It does not require additional information besides that provided in the training data, as prior knowledge of the distribution on the data or classes (Fayyad, 1991). Furthermore, as shown by Mitchie et al. (1994), it displays good classification accuracy compared to other techniques. In addition to the DT algorithm, we focused on the SVM algorithm because of its increased performance in pattern recognition and regression estimation. It guarantees the presence of a unique, optimal, and global solution for solving a linearly constrained quadratic programming problem (Shin et al. 2005). Moreover, it is capable to generate an ideal solution with small training datasets. It has a good generalization performance on unseen data (Deris et al. 2011). Our generated dataset is not trivial but requires complex non-linear decision boundaries as those that can an SVM algorithm generate because of utilizing the kernel function.

The two algorithms were trained and tested on the same training and test set data, respectively. After model fitting and prediction, evaluation metrics such as accuracy, precision, recall, F1Score, Jaccard Index, log loss, 0/1 Loss, and Hamming loss were applied to investigate each of the models' performance and efficiency in such problems.

1. Decision Tree Classifier

Regarding the DT classifiers, they are a type of supervised machine learning, used to learn decision rules and data internal structures from data feature(s) and labels. They use the training dataset to build models and classify new data whose class labels are unknown. DT classifiers are one of the possible multistage approaches, where the basic idea involves the breakdown of a complex decision-making problem process into a collection of simplified decisions, thus providing a solution that is often more comprehensible (Safavian & Landgrebe, 1991). DT classifiers are used immensely in diverse areas such as character recognition, default loan detection, customer data analysis, medical diagnosis, to name only a few.

DT Models are proven to be most useful in data mining because they obtain reasonable accuracy and are relatively inexpensive to compute (Du & Zhan, 2002). Most often, DT classifiers such as Classification and Regression Trees (CART) and C4.5 form optimal decision trees via Tree Building and then Tree Pruning. In the tree building process, starting from the root node, data are recursively split to form new tree levels where each of the levels contains internal nodes connected via branches. In addition to the root node, internal nodes, and branches, the tree consists of leaf nodes that are the lowest level of the tree, at the end of the last branches, these play a crucial role when it comes to prediction.

Intermediate nodes and root nodes can either predict by setting the majority value in the node or split based on a particular feature. The split at each node is mainly based on a metric called purity. A pure node contains data of the same class; however, an impure node or noisy node contains two or more label types.

To improve tree generalization and to prevent overfitting, tree pruning is used to prune the leaves and branches of less importance and are responsible for the classification of single or very few data vectors. There are two types of pruning, pre-pruning and post-pruning. Pre-pruning is also known as the early stopping criterion, which aims to stop the DT from going deep when meeting one of the conditions such as (1) maximum depth of tree (longest path from a root node to a leaf node), (2) minimum number of examples that should be present in a node for the split to happen, and (3) minimum number of examples that should be present in a leaf node/terminal node. Post-pruning, however, becomes necessary when the DT is built entirely without any early stopping conditions. Thus, the tree can be susceptible to overfitting. The process is conducted by observing the complexity parameter (CP) and then prune the tree with optimal CP value associated with the lowest cross-validation error. CP is the value of improvement of fitting a DT should at least maintain within each split; else, the split is not attempted.

On one hand, DT algorithms' advantages are the following: (1) Easy to implement, interpret, and visualize, (2) Implicitly perform variable screening or feature selection to test against data, thus increasing the efficiency through eliminating unnecessary computations, (3) Can hold numerical and categorical data with multioutput problems, (4) requires somehow little effort from users for data preparation, and (5) are active not only with variables with linear/non-linear relationships but also with variables having complex and even disjunctive relationships.

On the other hand, DT algorithms would carry some disadvantages such as (1) decision trees are prone to overfitting, where learners can create over-complex trees making them unable to generalize over unforeseen data, (2) decision trees could be vulnerable in a way that a small change in the data would result in a totally different tree, (3) Greedy algorithms do not guarantee global optimal decision trees; things are more reliable when generating multiple trees with different samples of features and sampling with replacement, and (4) decision trees could be biased in the case some classes dominate over others in the data sample.

2. Support Vector Machine Classifier

Regarding the SVM algorithm, it is a powerful algorithm used either for data classification or regression challenges in machine learning. The algorithm outputs an optimal hyperplane to segregate different class labels into separate categories. It is also known as a maximum margin classifier where it forms support vectors that maximize the distance between nearest distinct labels. Several types of kernel functions are used to train SVM training parameters such as linear, polynomial, radial basis function (RBF), and sigmoid (Meyer & Wien, 2015).

SVM classifier cannot handle categorical variables; hence, each data instance should be represented as a numeric vector since it is based on Euclidean distance. In case any categorical attribute is presented, dummy variables should be used to express that attribute.

Feature normalization is crucial when applying the SVM classifier, particularly for a non-linear kernel classifier, that depends on Euclidean distance or inner product of feature vectors when implementing the kernel trick. The main advantage of normalization is to avoid features with extensive numeric ranges from dominating features having relatively small numeric ranges. Consider our dataset containing features like "Standard Deviation of Duration of Tasks" and "Project Duration" in days. The first feature ranges from 0.5 to 4, and the second feature ranges from 180 to 540, it is apparent that the latter feature is around a hundred times larger than the first one. Therefore, "Project Duration" will intrinsically influence the result because of its larger values.

To train an SVM classifier, choosing the kernel function is considered a primary step to make. The choice of kernel affects the capability of the classifier. In fact, there is no direct rule that would tell which must be chosen. Instead, it all depends on the data in hand, where different kinds of data would require different kernels.

Therefore, SVM classifier has its benefits such as (1) It works well when there is a linear separation between different classes because of the largest margin between them, (2) It handles non-linear classification efficiently using the kernel trick, (3) It is stable and is not significantly affected by a small change of data, (4) It works with regression and classification, and (5) SVM is capable of generalizing to unseen data because of the regularization feature.

On the other hand, SVM classifier has drawbacks such that (1) It is difficult to choose the appropriate kernel function, where many kernels might be tested, (2) It requires feature scaling for all features to be equally equivalent, (3) It needs all feature vectors to be represented as numeric vectors; thus dummy variables usage might be necessary, (4) SVM model is challenging to interpret, unlike decision trees, and (5) SVM performance corresponds to the valid selection of hyperparameters like cost and gamma for the classifier to predict accurately unknown data; thus, it is computationally expensive.

CHAPTER IV

METHODOLOGY

We envisioned a dataset that serves as a clarificatory template for risk identification via AI. First, we identified a set of milestone features which we believe have a direct effect on the milestone status. In our case, the features were eleven including (1) Number of Users, (2) Number of Tasks, (3) Duration, (4) Type, (5) Average Duration of Tasks, (6) Order of Milestone, (7) Standard Deviation of the Duration of Tasks, (8) Task Type 1, (9) Task Type 2, (10) Task Type 3, and (11) Project Duration. Second, we structured a table to feed our data into it. The table includes five hundred rows (instances) each representing a single milestone and sixteen columns. The first twelve columns represent milestone numbers and features, while the last four represent milestone statuses (labels), whether the milestone includes problem(s) or not. Accordingly, our first twelve columns signify our independent variables, whereas the remaining ones signify our dependent variables.

Feature values were chosen based on previous projects' data, either from practice or from research papers. We fed the values of each independent feature as a range of values using a continuous uniform distribution. Based on that, we set a criterion to identify our dependent variable labels by specifying the threshold split of each independent variable. Noise was taken into consideration by assigning some of the labels randomly to make the dataset more challenging. The dataset was designed to be non-linearly separable in order to imitate real data distributions; thus, requiring complex non-linear boundaries to classify the distinct labels efficiently.

Furthermore, to ensure data quality and behavior, special considerations were taken into account. An exploratory data analysis was conducted to represent the correlation between different variables visually. Then, the data was adjusted manually for our visualized graphs to mimic real case scenarios.

Subsequently, the data table was used as a data frame within the R programming language in the interest of fitting classifiers, DT and SVM, to the data to use them as predictive models for potential problem(s) identification. After that, predictive performance analysis and evaluation were carried out to examine the algorithms' efficiency and validity based on various evaluation metrics resulting in specifying the optimal classifier to be standardized later for such type of data.

CHAPTER V

RESULTS AND ANALYSIS

Once milestone features were identified, a data frame which contains five hundred rows (instances) associated with sixteen columns was configured. The first column relates the number of milestones; however, the remaining ones relate to data features and labels. Exceptionally, the "Task Type" feature is separated into three variables, task types 1, 2, and 3. Each milestone is represented by an eleven-dimensional vector of numeric features.

Moreover, four categorical labels A, B, C, and D, each taking the value as zero or one, are associated with every instance in our dataset. That makes the problem a kind of a multi-labeled classification problem, where each milestone can carry one problem type or more, or even none.

Feature values are fed with values of different ranges using a continuous uniform distribution. Ranges of values were chosen based on real example projects. After that, all values were normalized using the linear transformation technique "Max-Min Normalization" (Patro & Sahu, 2015). The general form of the equation is depicted as the following:

$$x_j^i = \frac{f_j^i - \min(f_j)}{\max(f_j) - \min(f_j)} \quad (1)$$

Where, x_j^i is the normalized value of f_j^i; "j" being the feature in the i^{th} training example.

Data labels were then identified by setting different criteria for each label and then adding some noise to make our dataset more realistic.

In order to better understand the data, we carried out an exploratory analysis to gather insights by visualizing the relationship of different variables by utilizing the R programming language which us is an open-source programming language and powerful tool used for statistical computation and graphics. It supports the implementation of various ML techniques, such as regression, classification, clustering, etc. R version 4.0.2 was used in the present study. To present the data better, it was crucial to modify the data manually to imitate real data distributions.

The accompanying figures (3) – (6) attached in the appendices section reveal how our data is distributed after several amendments. To better present those figures, we revealed the regression between the variables plotted on each two-dimensional coordinate system. Figure 3: Standard Deviation of Duration of Tasks versus Number of Tasks. demonstrates a high positive correlation between the two variables, “Standard Deviation of Duration of Tasks” and “Number of Tasks”, with a correlation coefficient value (r) of 0.69. The result indicates that the former variable increases as the latter variable increases. Figure 4: Project Duration versus Standard Deviation of Duration of Tasks. reveals that “Project Duration” variable is affected by the “Standard Deviation of Duration of Tasks” positively; however, with a weaker relationship. Figure 5: Project Duration versus Number of Users shows a different relation for the dependency of “Project Duration” variable on the “Number of “Users” variable. This relation ends with a positive correlation with an r-value of 0.56. Figure 6: Project Duration versus Order of Milestone, on the other hand, reveals almost no correlation between the two variables, “Project Duration” and “Order of milestone”, with an r-value of 0.28.

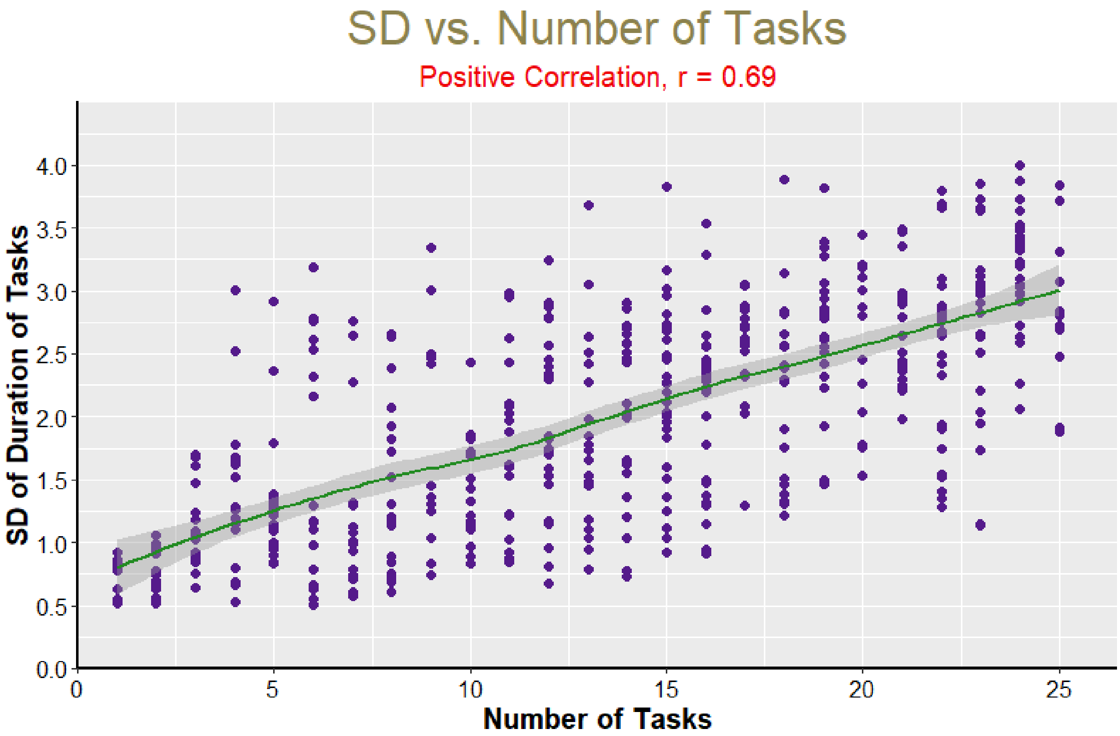

Figure 3: Standard Deviation of Duration of Tasks versus Number of Tasks.

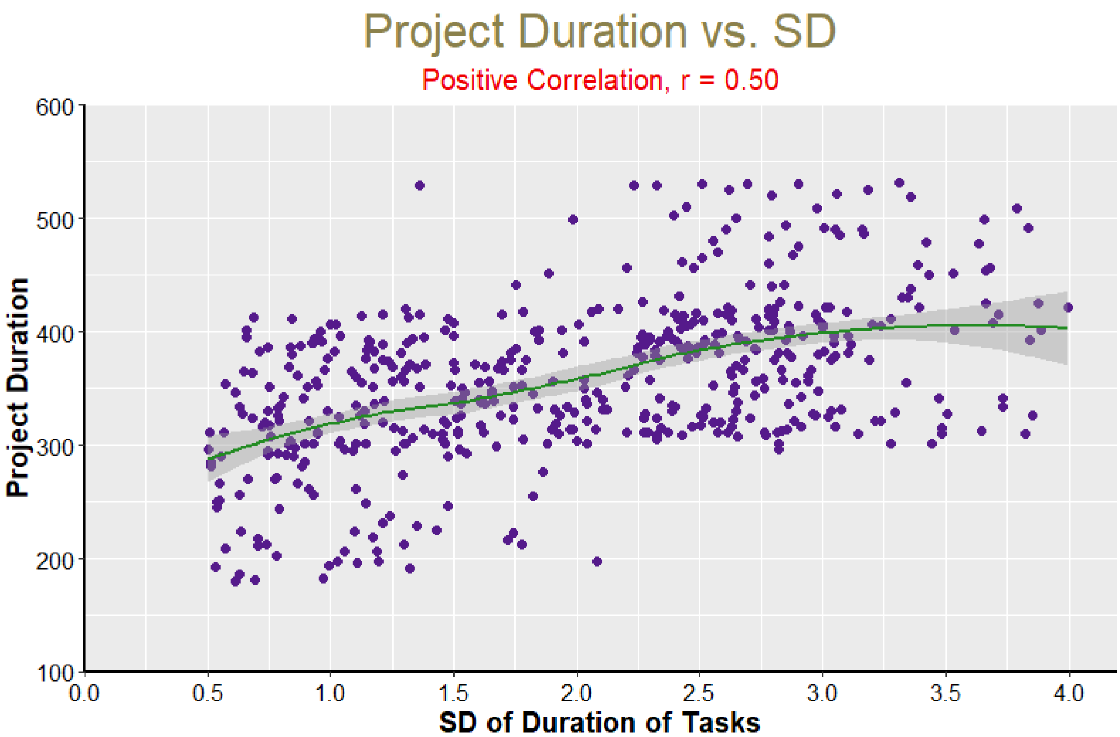

Figure 4: Project Duration versus Standard Deviation of Duration of Tasks.

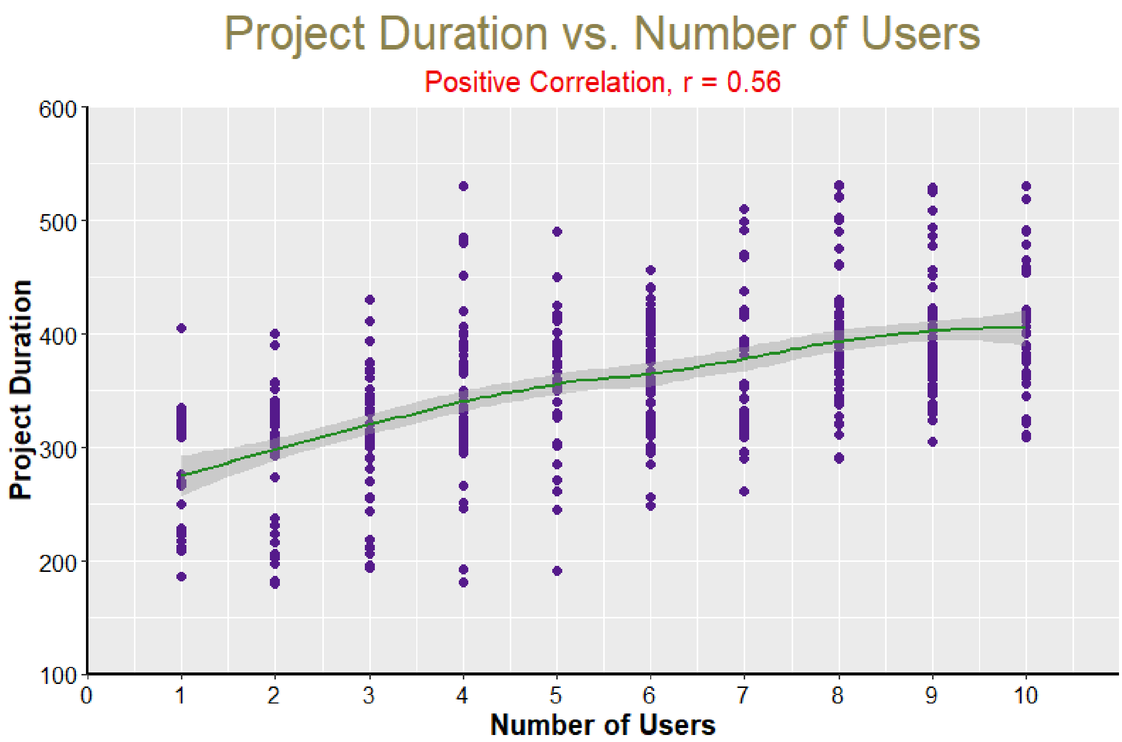

Figure 5: Project Duration versus Number of Users

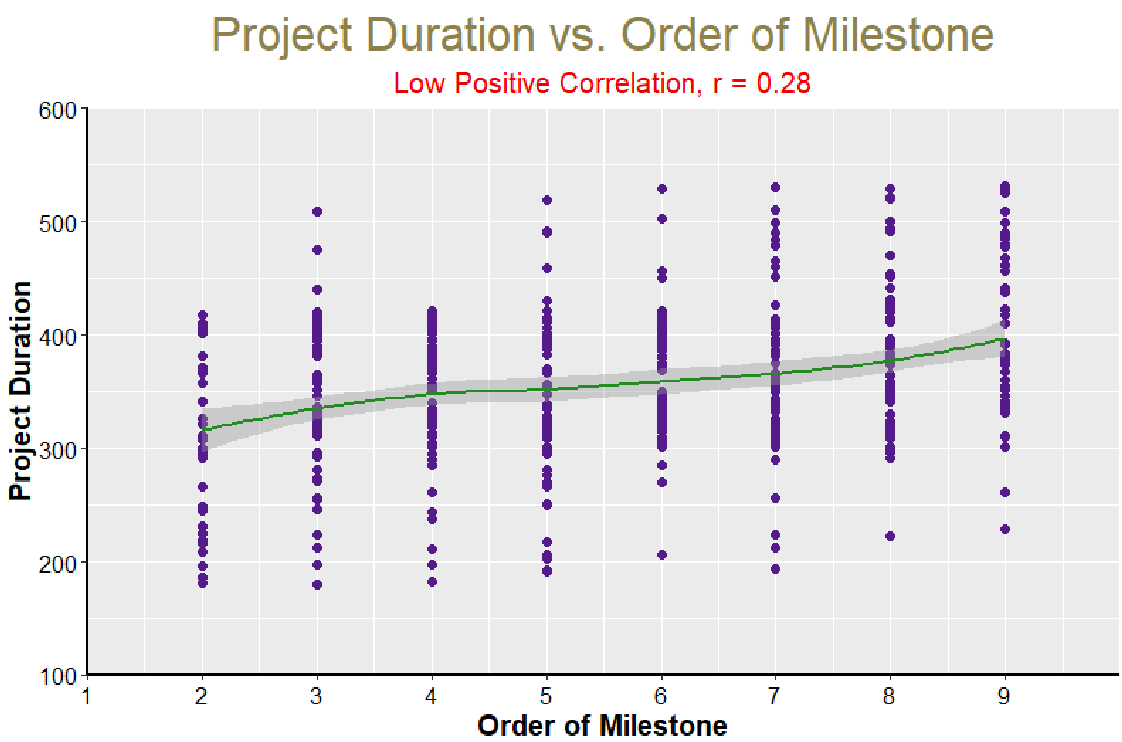

Figure 6: Project Duration versus Order of Milestone

Below is the coding work documentation as executed in the R platform for the DT and SVM models, respectively.

A. Decision Tree Models Analysis

To build our DT, we have adopted the "Recursive Partitioning and Decision Trees" (rpart) function to fit the model. In this case, we want to classify the problem type(s)/label(s) using the eleven previously mentioned features (predictors). "rpart" implements the CART algorithm for building decision trees in R. CART can handle both classification and regression tasks. The algorithm uses the “Gini Index” or “Gini Impurity” metric to create tree nodes for classification tasks.

Gini Index measures the impurity of a particular variable in a dataset. Its range of values is between zero and one, with zero indicating a pure classification. The Gini Index can be expressed with the following formula:

$$Gini\ Index = 1 - \sum_{i=1}^{L} (P_i)^2 \quad (2)$$

“L” indicates the number of classes with “Pi” indicating the probability of an object being classified to a particular class at the decision node.

Data preprocessing was then conducted by naming the variables, splitting the data, then excluding unnecessary columns/rows.

After naming all variables and labels, the data was split by a ratio of 80 percent for training and 20 percent for testing. Thus, forming a training set of four hundred labels and a test set of one hundred. Following this, the dataset was kept with only useful columns after excluding the first one from it. Each label is considered a single subproblem to solve; therefore, we had to build four models in our code. Each of the models was then treated as a binary classification model.

Giving model A as an example, we subset the training and test sets to include label A only with all the features. Factoring label A column was applied afterward to identify label levels to handle classification. Following that, a DT classifier was fit to the training set of label A by employing the "rpart" function version "4.1.15". After the model has been configured, predictions on the test set of label A became applicable. Thus, model A was used to predict the labels of test set A.

The models' performance evaluation has been investigated as a step after the models' predictions were completed. To further understand the model's performance on the test data, a confusion matrix was added. Columns in the matrix indicate real classes and their totals, while rows show the number of predicted ones in addition to their totals. All correct predictions are represented in the main diagonal of the table. That would facilitate the indication of incorrect predictions, which are the non-zero numbers laid outside the diagonal.

In order to enhance understanding of the matrix, key terms should be clarified model performance evaluation. (1) True positives (TP) the model correctly predicts the positive class, (2) True negatives (TN) the model correctly predicts the negative class, (3) False positive (FP) the model incorrectly predicts the positive class, and (4) False negative (FN) the model incorrectly predicts the negative class. Precision, Recall, and F1Score can be calculated as shown by the following equations:

$$Accuracy_{Individual-Label} = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

$$Precision_{Individual-Label} = \frac{TP}{TP+FP} \quad (4)$$

$$Recall_{Individual-Label} = \frac{TP}{TP+FN} \quad (5)$$

$$F1Score_{Individual-Label} = \frac{2*Precision*Recall}{Precision+Recall} \quad (6)$$

Hossin & Sulaiman (2015), defined the equations mentioned above as follows (1) Accuracy is the ratio of correct predictions over the total number of instances, (2) Precision measures the positive patterns that are correctly predicted from the total predicted patterns in a positive class, (3) Recall measures the fraction of positive instances that are correctly classified, and (4) F1Score represents the harmonic mean between recall and precision values.

Jaccard Index (Niwattanakul, Singthongchai, Naenudorn, & Wanapu, 2013), which measures the similarity between the predicted and actual labels, was also calculated. It is considered perfect when its value equals one where all predictions are correct and deemed worst when it equals zero, where is no intersection between the two sets. Actual and predicted vectors are numeric vectors of zeros and ones. The equation's mathematical representation is written as:

$$Jaccard\ Index_{Individual-Label} = \frac{|y_{pred} \cap y_{actual}|}{|y_{pred} \cup y_{actual}|} = \frac{|y_{pred} \cap y_{actual}|}{|y_{pred}|+|y_{actual}| - |y_{pred} \cap y_{actual}|} \quad (7)$$

Furthermore, the Log Loss metric was calculated to evaluate the model's quality of predictions (Read, Pfahringer, Holmes & Frank, 2011). It indicates zero when it is perfect, and all of the probability values in the predicted vector perfectly equal the ground-truth values in the actual vector. As the probabilities are less confident, the higher the loss is. The log loss equation is denoted as the follows:

$$Log\ Loss_{Individual-Label} = -\frac{1}{N}\sum_{i=1}^{N}(y_{actual} * \log(y_{pred}) + (1 - y_{actual}) * \log(1 - y_{pred})) \quad (8)$$

N is the length of the predicted vector. In other words, it is the number of instances in the test set. The predicted vector is the estimated probabilities of predictions.

Probabilities in the predicted vector or the soft labels are desirable in situations where probability information is useful. Soft classification estimates the class conditional probabilities explicitly and then makes the class prediction based on the largest estimated probability (Liu et al., 2011).

Lastly, the DT of model A was plotted for further interpretations, Figure 7: Decision Tree Diagram of Model A. As depicted in the tree, the classification of carrying Label A or not was based only on four features which are (1) Milestone Duration, (2) Number of Users, (3) Standard Deviation of Duration of Tasks, and (4) Task Type 2. As shown, the splitting was made on threshold splits defined by the classifier at each node. Moreover, the probabilities of selecting the activity at each node were also revealed.

Models B, C, and D were built similarly as model A using the same steps, functions, and formulas. The performance measures of the four DT models A, B, C, and D are represented in Table 2: DT Models Performance. Moreover, figures (Figure 8: Decision Tree Diagram of Model B, Figure 9: Decision Tree Diagram of Model C, and Figure 10: Decision Tree of Model D) portray the DT diagrams of models B, C, and D, respectively.

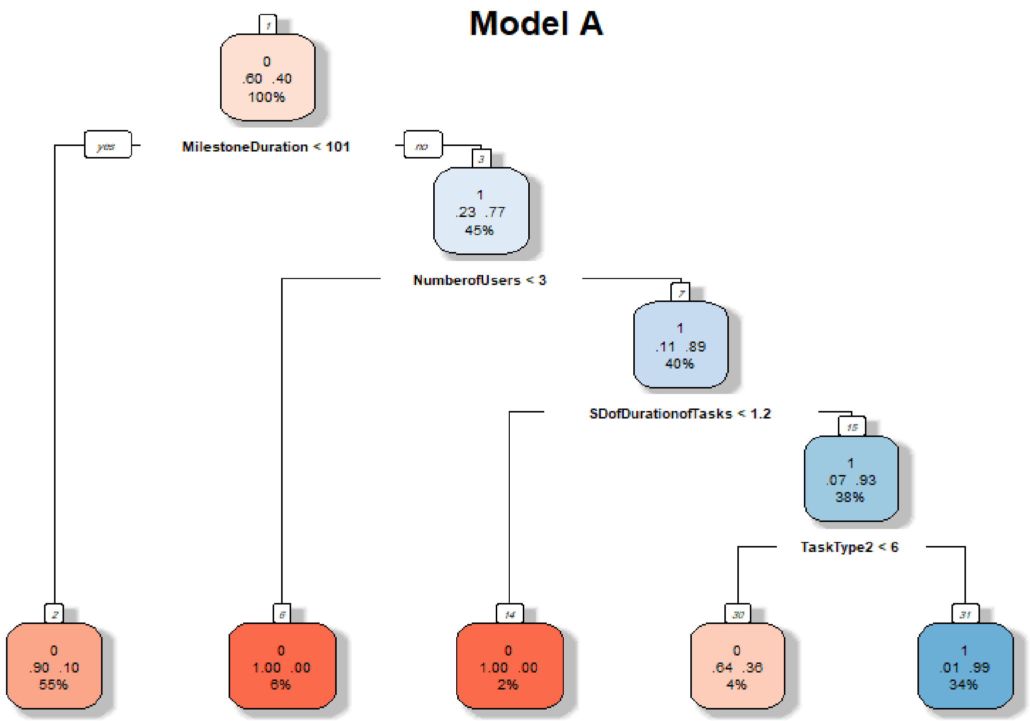

Figure 7: Decision Tree Diagram of Model A

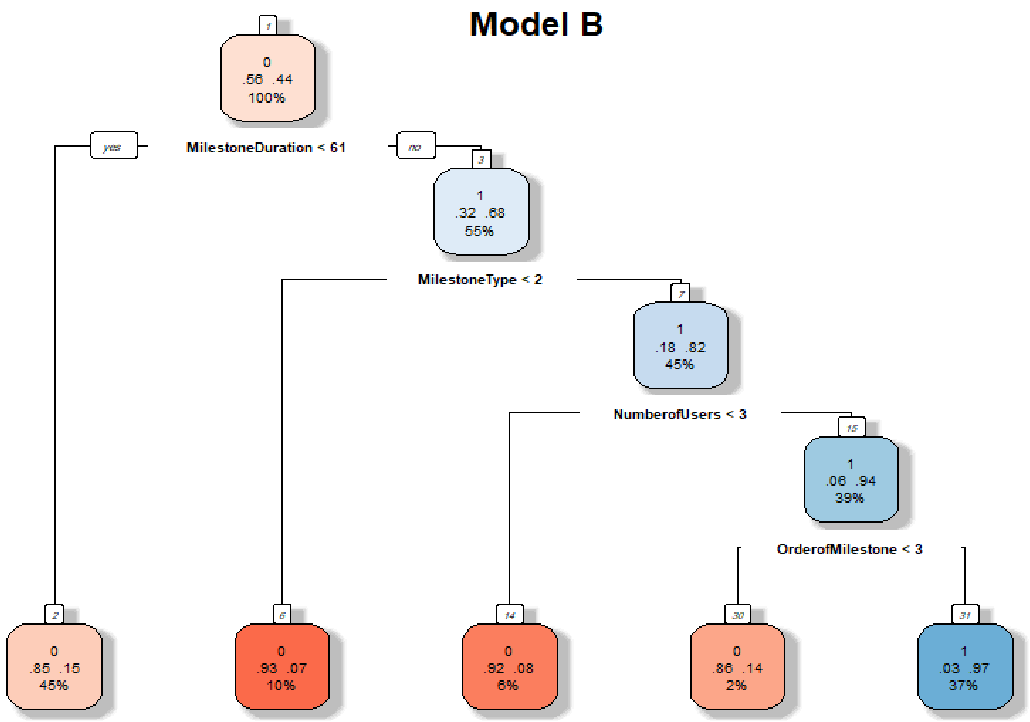

Figure 8: Decision Tree Diagram of Model B

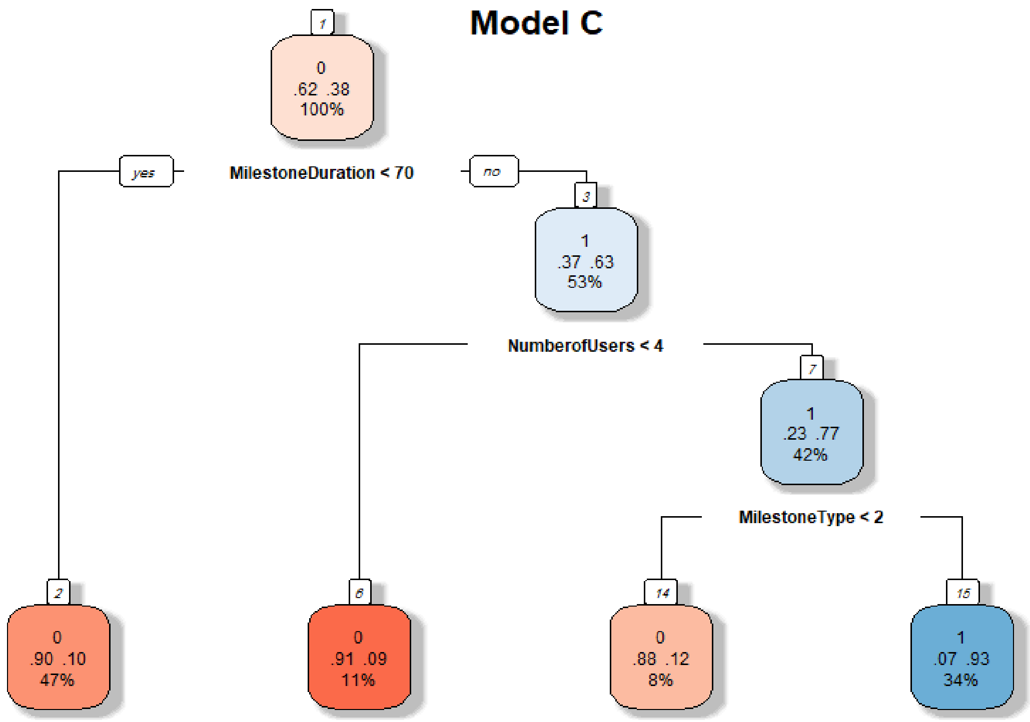

Figure 9: Decision Tree Diagram of Model C

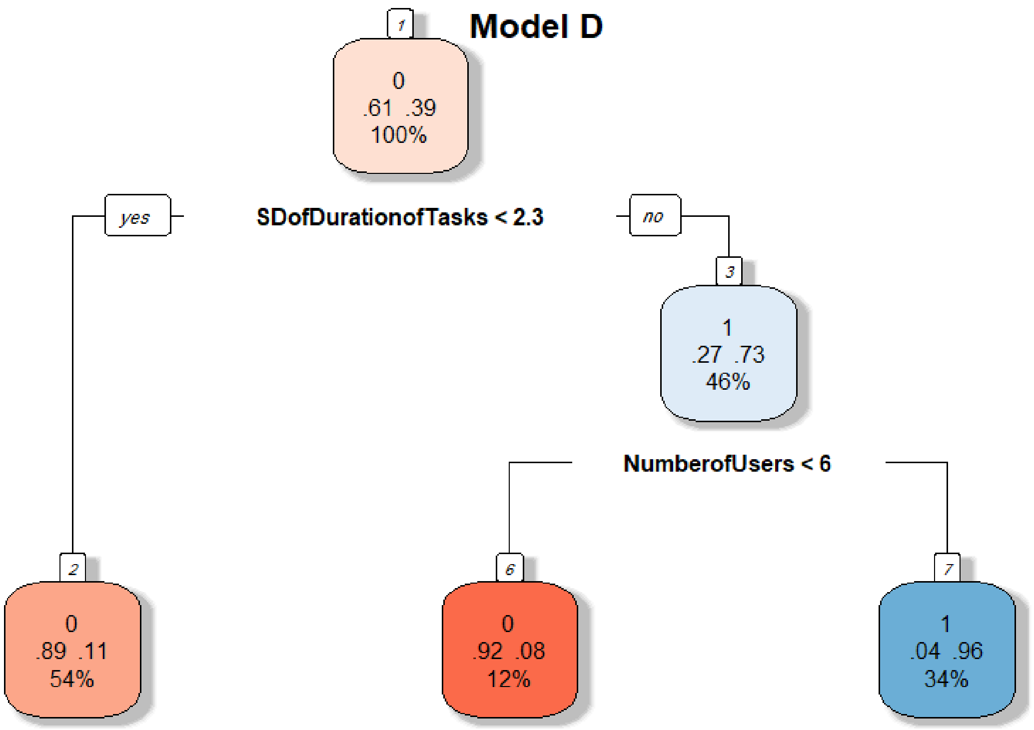

Figure 10: Decision Tree of Model D

Table 2: DT Models Performance

	Model A	Model B	Model C	Model D
Accuracy	0.920	0.870	0.890	0.940
Precision	1.000	0.914	0.931	1.000
Recall	0.818	0.826	0.836	0.860
F1Score	0.900	0.868	0.881	0.925
Jaccard Index	0.818	0.767	0.788	0.860
Log Loss	0.408	0.409	0.344	0.212

After the four binary classifiers had been built, some evaluation metrics related to multi-label classification problems were followed. Those metrics evaluate the entire performance when all sub models are integrated.

Therefore, two matrices of actual and predicted labels were constructed. The first matrix contains the four vectors of the ground-truth 0-1 labels as numeric vectors.

The second matrix is the one, including the predicted vectors, as numeric vectors of 0-1 predictions. Subsequently, having the two matrices, multi-label evaluation metrics were applied such as (1) Accuracy, (2) Precision, (3) Recall, (4) F1Score, (5) 0/1 Loss, and (6) Hamming Loss for further interpretation of the model.

There are two types of measuring performance in the case of multi-label classification. Type 1, measurement "per instance," is where each instance is considered separately, then all indices across the test objects are averaged. Type 2, the "global measure," is where the entire instances are measured simultaneously. In this research, Accuracy, Precision, Recall, and F1Score, were computed in the form of type 2; however, 0/1 Loss and Hamming Loss were computed following type 1.

Godbole and Sarawagi (2004), defined Accuracy, Precision, Recall, and F1Score as follows:

$$Accuracy_{Multi-Label} = \frac{|Actual_{labels} \cap Predicted_{labels}|}{|Actual_{labels} \cup Predicted_{labels}|} \quad (9)$$

$$Precision_{Multi-Label} = \frac{|Actual_{labels} \cap Predicted_{labels}|}{|Predicted_{labels}|} \quad (10)$$

$$Recall_{Multi-Label} = \frac{|Actual_{labels} \cap Predicted_{labels}|}{|Actual_{labels}|} \quad (11)$$

$$F1Score_{Multi-Label} = \frac{2*Precision*Recall}{Precision+ Recall} \quad (12)$$

0/1 Loss measures how the predicted labels differ from the actual ones. This is known also as the exact match measure (Read, Pfahringer, Holmes & Frank, 2011). 0/1 Loss can be very harsh, since every instance which is not predicted perfectly contributes to the loss value. The equation can be expressed in the following form:

$$0/1\ Loss_{Multi-Label} = \frac{1}{N}\sum_{I=1}^{N} I(Actual_{labels} \neq Predicted_{labels}) \quad (13)$$

Tsoumakas, & Katakis (2007), applied hamming loss which is defined as:

$$Hamming\ Loss_{Multi-Label} = \frac{1}{N}\sum_{i=1}^{N}\frac{|Actual_{labels}\,\Delta\,Predicted_{labels}|}{|L|} \quad (14)$$

Where Δ stands for the symmetric difference of the two sets and corresponds to the XOR operation. Hamming Loss is the fraction of labels that are incorrectly predicted. It is more tolerant than 0/1 Loss, as it penalizes only the individual labels.

The overall model performance results are shown in Table 3: Decision Tree Models Performance on Multi-Labeled Dataset.

Table 3: Decision Tree Models Performance on Multi-Labeled Dataset

Decision Tree Models Performance on Multi-Labeled Dataset					
Accuracy	Precision	Recall	F1Score	Zero One Loss	Hamming Loss
0.805	0.957	0.835	0.892	0.330	0.095

Traditional methods accuracy for risk identification and assessment are not well-defined in the literature. Model performance can be evaluated by accuracy; although, it would not be enough to measure a model's performance. There is no threshold yet that would say a particular accuracy is accepted or not. It all depends on the type of problem and the accepted error rate from users. Based on our results, the DT model exhibits relatively better accuracy than SVM with values of 80.5% and 72%, respectively, on the test set. We consider that those values are reasonable as compared to the accuracy obtained by other papers 47.2% (Gondia et al., 2019) and 81.4% (Suresh & Dillibabu, 2018).

B. Support Vector Machine Models Analysis

In this research, the RBF kernel was adopted to train our SVM classifiers.

The RBF function is defined as:

$$K(x,z) = \exp(-\gamma\|x - z\|^2) \quad (15)$$

$$\gamma = \frac{1}{2\sigma^2} \quad (16)$$

The value of the function ranges between zero and one, where if the two inputs are similar, the value becomes one. However, if the two inputs are too far, the value tends to zero.

Hyperparameters that must be specified, are cost "C" and gamma "ϒ". The user provides possible ranges of "C" and "ϒ" with a grid space. Different points of (C, ϒ) are tried to find the one giving the highest accuracy on the cross-validation dataset. After that, users use the best parameters to fit the algorithm on the training set and generate the best model (Chang & Lin 2011).

"C" controls the tradeoff between the correct classification of training examples and the maximization of the classifier's margin. A large value of "C" indicates that the classifier would accept a small margin for classifying all training examples correctly. However, if "C" is low, the classifier would encourage a large margin rather than obtaining high training accuracy.

"ϒ" is responsible for determining the influence of a single training example. A high value of "ϒ" means only close instances influence the decision boundary; however, low value means that even far instances are considered and affect the decision boundary.

As in the DT coding work, one SVM model was built for every label type in the dataset. Thus, four SVM classifiers A, B, C, and D were trained. The dataset was primarily entered, and then, data preprocessing was performed as a step ahead.

Then, the dataset of five hundred instances was split into 80 percent training set and 20 percent testing set. After that, the first column containing the row numbers was excluded to keep the data with valuable data only. It is crucial here to handle data splitting on the same column at the beginning of the code to secure the identity of similar training and test set elements.

Feature scaling was carried out as support vector machines are distance-based classifiers. Feature scaling aims to guarantee that features are on almost the same scale; thereby, each attribute is equally important in computing distances and makes it easier to process by most machine learning algorithms. This can be achieved by adopting standardization (Z-score normalization) to rescale data and ensure that means and standard deviations carry a value approximate or equal to zero and one, respectively.

The Z-score normalization equation (Jung & Lease, 2011) is defined as follows:

$$z_j^i = \frac{f_j^i - \mu_j}{\sigma_j} \quad (17)$$

z_j^i is the normalized value of f_j^i, where f_j^i is the value of feature "j" in i^{th} training example. μ_j and σ_j denote the mean and standard deviation, respectively, for feature f_j.

On that basis, it became possible to fit and test the different SVM classifiers. Having "classifier A" as an example, the training and test sets were subset to include only label A with all the features associated with it; then, label A column was factored to indicate label levels for the classifier to perform classification. Following this, an SVM classifier was fit to the training set of label A.

Subsequently, a confusion matrix had been developed before the model's performance was investigated based on Accuracy, Precision, Recall, F1Score, Jaccard Index, and Log Loss. All performance metrics were computed based on the equations defined previously in the DT model section. The performance measures of the four SVM models A, B, C, and D are depicted in Table 4: SVM Models Performance.

Table 4: SVM Models Performance

	Model A	Model B	Model C	Model D
Accuracy	0.860	0.850	0.840	0.880
Precision	0.857	0.930	0.923	0.897
Recall	0.818	0.769	0.734	0.813
F1Score	0.837	0.842	0.818	0.853
Jaccard Index	0.720	0.727	0.692	0.744
Log Loss	0.349	0.353	0.417	0.382

After the four binary SVM models were completed, some multi-label classification metrics were exploited to gain further information about the models' performance when they are integrated. Thus, two matrices at the beginning were constructed, where the first one contains the four ground-truth 0-1 labels numeric vectors, while the second includes the four predicted 0-1 numeric ones. Thereafter, having the two matrices, it became permissible to apply multi-label evaluation metrics like Accuracy, Precision, Recall, F1Score, 0/1 Loss, and the Hamming Loss to measure overall model's quality. The outcomes are revealed in Table 5: Support Vector Machine Models Performance on Multi-Labeled Dataset.

Table 5: Support Vector Machine Models Performance on Multi-Labeled Dataset

Support Vector Machine Models Performance on Multi-Labeled Dataset					
Accuracy	Precision	Recall	F1Score	Zero One Loss	Hamming Loss
0.720	0.901	0.781	0.837	0.470	0.142

CHAPTER VI

CONCLUSION

A project is a compile of series tasks and milestones that are needed to be accomplished to attain a particular goal. Traditionally, a project was categorized as successful if certain constraints like scope, budget, and schedule are met, besides many other factors. Yet, still and all, project success rates are not getting any better.

In this paper, we proposed the use of ML to assist project managers in planning complex projects more efficiently by means of building ML models that allow project managers to identify project risks on the milestone level based on historical data. The work came across the different traditional techniques employed in risk management for risk identification, and the various practices and traditional measures that would increase project success rates. However, despite all of that, it revealed that those are no longer sufficient.

We trained two classifiers: the DT and SVM classifiers to a manually constructed dataset containing five hundred milestones and eleven features with four labels for the aim of predicting potential problem(s) identification. Prior to fitting, we transformed the dataset from randomly selected values into one that carries patterns through its data. Subsequently, predictive performance analysis and evaluation of the two models were conducted to compare the classifiers' efficiency based on various evaluation metrics such as Accuracy, Precision, Recall, F1Score, 0/1 Loss, and Hamming Loss.

Based on the evaluation results obtained, the DT model provides better performance on the data examined. That is clearly shown through further development, where a bar plot was constructed to compare models' evaluation metrics. Figure 11: DT/SVM Metrics Comparison clearly depicts that the DT model outperforms the SVM model in terms of overall performance. The DT model showed accurate results upon prediction, about 80.5 percent. The F1Score was higher, with a value of 89.2 percent. Furthermore, the 0/1 Loss and Hamming Loss functions of DT were relatively lower than those of the SVM model with values closer to 33 percent and 9.5 percent, respectively. Based on those results, we recommend the use of DT models on like types of datasets.

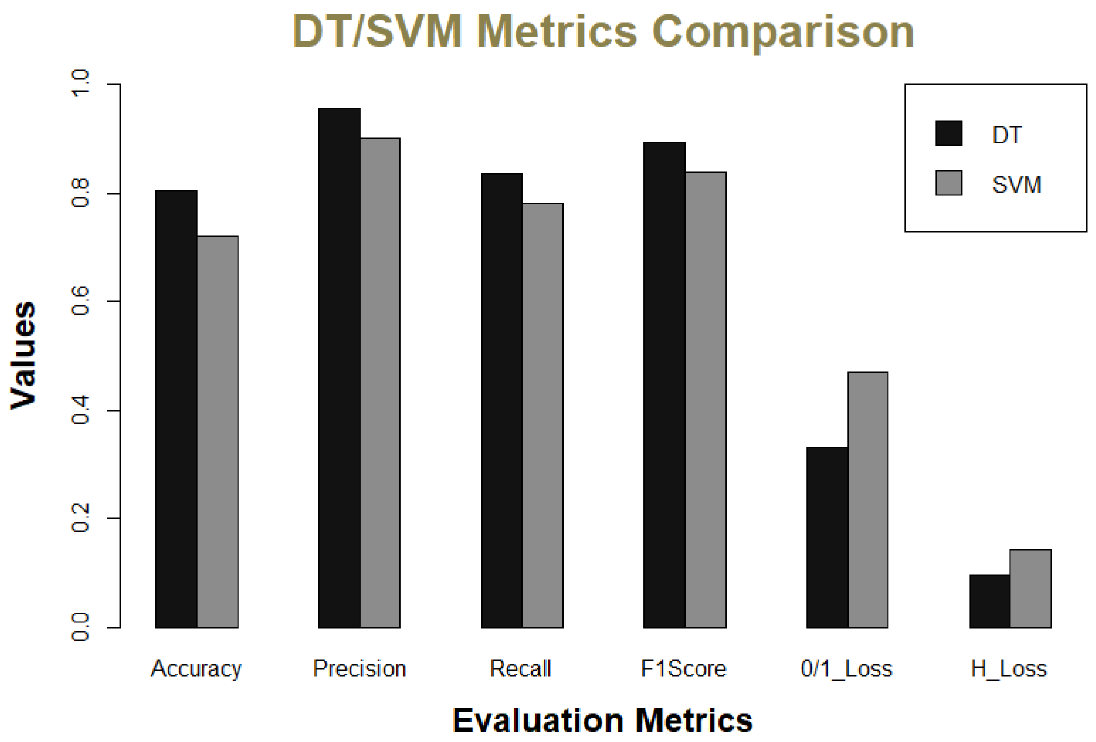

Figure 11: DT/SVM Metrics Comparison

We consider that our work can be improved further. That would be achieved by compiling different sets of real-world project data to train various DT models where each is relevant to specific types of projects. Moreover, other kinds of classifiers can be tried and evaluated against DT and SVM to standardize the usage of best performing classifiers on such kind of data. We recommend the use of algorithms capable of identifying and representing non-linear relationships in the dataset as its labels cannot be separated proficiently by linear classifiers.

REFERENCES

Al Najjar, R. & Al-Sarraj, F. (2019). A virtual partnership? How artificial intelligence will disrupt project management and change the role of project managers. https://www.pwc.com/m1/en/publications/virtual-partnership-artificial-ntelligence disrupt-project-management-change-role-project-managers.html

Assaf, S. A., & Al-Hejji, S. (2006). Causes of delay in large construction projects. *International journal of project management*, *24*(4), 349-357.

Bajaj, D., Oluwoye, J., & Lenard, D. (1997). An analysis of contractors' approaches to risk identification in New South Wales, Australia. *Construction Management & Economics*, *15*(4), 363-369.

Chang, C. C., & Lin, C. J. (2011). LIBSVM: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, *2*(3), 1-27.

Chapman, R. J. (2001). The controlling influences on effective risk identification and assessment for construction design management. *International Journal of Project Management*, *19*(3), 147-160.

Chapman, R. J. (1998). The effectiveness of working group risk identification and assessment techniques. *International Journal of Project Management, 16*(6), 333-343.

Charette, R. N. (1989). *Software engineering risk analysis and management* (p. 325). New York: McGraw-Hill.

Cohen, M. W., & Palmer, G. R. (2004). Project risk identification and management. *AACE International Transactions*, IN11.

Du, W., & Zhan, Z. (2002). Building decision tree classifier on private data.

Gigerenzer, G. (2012). Risk literacy. In *This will make you smarter: New scientific concepts to improve your thinking* (pp. 259-261). Harper Perennial.

Godbole, S., & Sarawagi, S. (2004, May). Discriminative methods for multi-labeled classification. In *Pacific-Asia conference on knowledge discovery and data mining* (pp. 22-30). Springer, Berlin, Heidelberg.

Gondia, A., Siam, A., El-Dakhakhni, W., & Nassar, A. H. (2019). Machine Learning Algorithms for Construction Projects Delay Risk Prediction. *Journal of Construction Engineering and Management*, *146*(1), 04019085.

Hossin, M., & Sulaiman, M. N. (2015). A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process*, *5*(2), 1.

Ismail, I., Memon, A. H., & Rahman, I. A. (2013). Expert opinion on risk level for factors affecting time and cost overrun along the project lifecycle in Malaysian construction projects. *International Journal of Construction Technology and Management*, *1*(2), 2289.

Jung, H. J., & Lease, M. (2011, August). Improving consensus accuracy via z-score and weighted voting. In *Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence.*

Lahmann, M. (2018). AI will transform project management. Are you ready? https://www.gartner.com/en/newsroom/press-releases/2019-03-20-gartner-says-80-percent-of-today-s-project-management

Lahmann, M. & Stierli A. (2019). How can we prevent project management from falling into the AI darkness? https://www.pwc.ch/en/insights/risk/controlling-the-dark-side-of-ai.html

Lester, A. (2006). *Project management, planning and control: managing engineering, construction and manufacturing projects to PMI, APM and BSI standards*. Elsevier.

Loucks, J., Davenport, T., & Schansky, D. (2018). State of A.I. in the enterprise, 2nd edition. Retrieved from Deloitte Consulting LLP website: https://www2.deloitte.com/content/dam/insights/us/articles/4780_State-of-AI-in-the-enterprise/DI_State-of-AI-in-the-enterprise-2nd-ed.pdf

Mahamid, I. (2011). Risk matrix for factors affecting time delay in road construction projects: owners' perspective. *Engineering, Construction and Architectural Management*, *18*(6), 609-617

Marr, B. (2016). Why Artificial Intelligence Will Change Our World and Why It Needs to Be Purposeful.

Maytorena, E., Winch, G. M., Freeman, J., & Kiely, T. (2007). The influence of experience and information search styles on project risk identification performance. *IEEE Transactions on Engineering Management, 54*(2), 315-326.

Meyer, D., & Wien, F. T. (2015). Support vector machines. *The Interface to libsvm in package e1071*, *28*.

Niwattanakul, S., Singthongchai, J., Naenudorn, E., & Wanapu, S. (2013, March). Using of Jaccard coefficient for keywords similarity. In *Proceedings of the international multiconference of engineers and computer scientists* (Vol. 1, No. 6, pp. 380-384).

Patro, S., & Sahu, K. K. (2015). Normalization: A preprocessing stage. *arXiv preprint arXiv:1503.06462*.

Pedroso, M. (2017). Application of machine learning techniques in project management tools.

Pinto, J. K., & Slevin, D. P. (1988). 20. Critical Success Factors in Effective Project implementation*. *Project management handbook*, *479*, 167-190.

Project Management Institute. (2017). *A guide to the project management body of knowledge (Pmbok® Guide)*. Newtown Square, PA.

Pulse of the Profession (2018). Next practices: Maximizing the benefits of disruptive technologies on projects.

Pulse of the Profession 2018 (2018). Success in disruptive times: Expanding the value delivery landscape to address the high cost of low performance.

Pulse of the Profession 2017 (2017). Success rate rise: Transforming the high cost of low performance.

Pulse of the Profession 2019 (2019). The future of work: leading the way with pmtq.

Raftery, J. (2003). *Risk analysis in project management*. Routledge.

Rastogi, R., & Shim, K. (2000). PUBLIC: A decision tree classifier that integrates building and pruning. *Data Mining and Knowledge Discovery*, *4*(4), 315-344.

Read, J., Pfahringer, B., Holmes, G., & Frank, E. (2011). Classifier chains for multi-label classification. *Machine learning*, *85*(3), 333.

Royer, P. S. (2000). Risk management: The undiscovered dimension of project management. *Project Management Journal*, *31*(1), 6-13.

Russell, S. J., & Norvig, P. (2016). *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited, Safavian, S. R., & Landgrebe, D. (1991). A survey of decision tree classifier methodology. *IEEE transactions on systems, man, and cybernetics*, *21*(3), 660-674.

Sri Susmitha, A. R. S., Hemantha Raja, K., and Asadi SS. (2018). A statistical approach for ranking of factors impacting the timeline of residential construction projects using important index method (impi). *IJCIET*, *9*(5), 1075 – 1083.

Suresh, K., & Dillibabu, R. (2018). Designing a machine learning based software risk assessment model using Naïve Bayes algorithm. *TAGA Journal*, *14*, 3141-3147.

Tsoumakas, G., & Katakis, I. (2007). Multi-label classification: An overview. *International Journal of Data Warehousing and Mining (IJDWM)*, *3*(3), 1-13.

Williams, T. (1995). A classified bibliography of recent research relating to project risk management. *European journal of operational research*, *85*(1), 18-38.