



Research Article

Prediction of failures in the project management knowledge areas using a machine learning approach for software companies

Gizatie Desalegn Taye¹  · Yibelital Alemu Feleke²

Received: 15 December 2021 / Accepted: 25 April 2022

Published online: 10 May 2022

© The Author(s) 2022 [OPEN](#)

Abstract

In this paper we propose a novel machine-learning model to predict project management knowledge areas failure for software companies using ten knowledge areas in project management based solely on the criteria of unambiguity, measurability, consistency, and practicability. The majority of software projects fail in software companies due to a lack of software project managers who are unfamiliar with the Project Management Knowledge Areas (PMKAs) that are used without considering the company's conditions or project contexts. By distributing questionnaires, we use an experimental methodology and the snowball sampling method to collect data from software businesses. We employ machine learning techniques including Support Vector Machines (92.13%), Decision Trees (90%), K-Nearest Neighbors (87.64%), Logistic Regression (76.4%), and Naive Bayes (66%) to adapt data from failed software projects. When we look at the results, Support Vector Machine outperforms the other four machine learning methods. High dimensional data is more efficient and contains nonlinear changes since Support Vector Machines deal with categorical data. The study's purpose is to improve project quality and decrease software project failure. Finally, we recommend collecting more failed project datasets from software businesses and comparing them to our findings to predict knowledge domain failure.

Article highlights

- Design a machine learning model to predict knowledge area failure in project management.

- Compare and contrast the machine learning model's performance.
- Evaluate the suggested machine learning model.

Keywords Project management · Project Management Knowledge Areas · Project failure · Machine learning

1 Introduction

An established software company's goal is to sell software products and profit from them. A project is a short-term undertaking that results in a unique deliverable [1]. The objectives of project management including initiating, planning, executing, regulating, and closing projects, as well as controlling the operations of the project team

within the defined time, scope, budget, and quality standards to achieve all agreed goals and software project management refers to the scheduling, planning, resource allocation, and execution [2]. There are ten software Project Management Knowledge Areas (PMKAs). These are Project Integration Management (PIM), Project Scope Management (PSM), Project Time Management (PTM), Project Cost Management (PCM), Project Quality Management

✉ Gizatie Desalegn Taye, gizatied2004@gmail.com | ¹Department of Computer Science, Faculty of Technology, Debre Tabor University, Debre Tabor, Ethiopia. ²Department of Database Administration, Amhara National Regional State Labour and Training Bureau, Bahir Dar, Ethiopia.



(PQM), Project Human Resource Management (PHRM), Project Risk Management (PRM), Project Procurements Management (PPM), Project Communications Management (PCCM), and Project Stakeholders Management (PSTM) [1]. The problems that cause software project failures are poor planning, lack of leadership, problems with people, vague or changing requirements, life cycle problems, inefficient communication process, inadequate funding, little attention to approval of stakeholders, lack of schedule, missed deadlines, due to the hiring of unqualified project manager. As a result, the research's goal is to forecast knowledge areas of project management failures for software firms. We develop a model based on machine learning that helps software project managers predict the failed knowledge areas that best fit the current situation (problem domain (failed motives), company characteristics, project size, indispensable nature of the project, the nature of the opportunities, and the methodology that follows). Improving the efficiency and maintaining the sustainability of a software project are obstacles that project managers face. The probability of project failure is generally due to a lack of knowledge, skills, resources, and technology during project implementation [3, 4]. The study answers the following research questions.

1. How do we design a machine learning model that predicts project management knowledge area failure?
2. Which machine learning techniques are the most effective for predicting project management knowledge areas failure?
3. How well does our model predict project management failure in terms of knowledge areas?

The study would reduce the amount of time, and effort was given would spend money (for the project managers, and software companies) to predict the failure of the knowledge areas. However, every software project is different and unique [5]. According to [6] described that a software company faces different challenges between funding, team building, and ideation to attract talent at a very early stage. Starting from this idea, the study focuses on identifying the reasons behind wariness and uncertainty in organizations. The authors [7] carried out identifies and categorizes the software engineering Project Management Knowledge Areas (PMKAs) used in software companies to map the state of the art using a systematic study method of literature mapping with the application of snowball sampling to evaluate the Software Engineering Body of Knowledge (SWEBOK) characterizes the content of the software engineering discipline and promotes a consistent view of software engineering. Our

work makes predictions not only statistics. The study presented by the Project Management Institute (PMI) identifies new domains of knowledge that contain a process to be followed for effective project management and project managers must have knowledge and skills in each of these areas or have specialists who can assist in these areas like some large projects have dedicated schedule coordinators, risk managers, communication specialists, or procurement contract officers. The authors [1] described a competent and knowledgeable project manager is vital to project success. The researchers evaluate the ten project management knowledge areas in service industries and manufacturing using the Analytic Hierarchy Process (AHP) and the Absolute Degree Grey Incidence Analysis (ADGIA) model. Both models have the result that project quality management is the most important knowledge area and also most strongly related to project communication management and least strongly related to project integration management but the literature has a gap.

The authors [8] focus on behavioral advertisement analysis, such as an individual's preferences, buying habits, or hobbies, and will employ machine-learning approaches to identify and successfully execute targeted advertising using data that reflects the user's retail activity. By building a unique framework that uses a classification model through streaming technologies, and produces a multi-class classifier to provide sector-based classification. To improve the accuracy of the model prediction task, the method uses a structured approach and multiple ensemble techniques. To forecast failure, we employed a multiclass classifier in our research. The authors [9] provided a framework for value realization. Universities must assess learning analytics (LA's) strategic role and spend carefully on the following criteria like high-quality data, analytical tools, knowledgeable people that are up to date on technology, and data-driven prospects for learning improvement. In our research, we used the four criteria to select attributes for prediction. The authors [10] investigated an efficient algorithm for predicting software reliability using a hybrid approach known as Neuro-Fuzzy Inference System, which was also applied to test data for software reliability prediction using complexity, changeability, and portability parameters in software development as input for the Fuzzy Inference System. After testing and training real-time data, forecast reliability in terms of mean relative error and mean absolute relative error. The study's findings are verified by comparing them to other state-of-the-art soft computing techniques.

From the above-mentioned related work, they have the following gaps in general. To begin with, the majority of research does not focus on making predictions. Second, the above-mentioned related works are carried out in the automotive supply sector, manufacturing, and non-governmental

organizations (NGOs). Third, they employed a different method than we did in our research. As a result, we focused our investigation on software companies. In Ethiopia, most software firms have inexperienced, unsuccessful, and less skilled project managers as compared to other experienced corporate projects. Third, when to add or reduce the criteria influence on the project management knowledge areas is self-evident. As a result, we added more factors to the mix. Finally, the datasets that are associated with them are quite modest. As a result, the output is hurried. So, we prepared the dataset as much as feasible.

The introduction section comes to a close with this paragraph. In Sect. 2, we look at the methodologies, which include everything from using datasets to predicting failed project management, as well as the design of the suggested model, data preparation, and the confusion matrices for calculating performance measures. The results, validation of the model, and discussion highlights of the performance metrics of the findings are presented in Sect. 3, and the paper is concluded with the possibility of future extension of this work.

2 Methodology

The research is based on experiments. Experimental research is a collection of research designs that employ manipulation and controlled testing to gain a better understanding of entire processes that predict outcomes depending on certain criteria. As a result, the following methods and techniques are employed to complete this study.

2.1 The designed proposed prediction model

The general description of the prediction failure model for project management knowledge areas in software companies is given in Fig. 1. The model has five major phases;

The first phase is the failure of project data collected from software development companies, the second phase is data pre-processing, which serves to refine our data cleansing, feature selection, data transformation, and data reduction tasks, the third phase consists of implementing the selected algorithms like Support Vector Machine (SVM), Decision Trees (DT), Naïve Bayes (NB), Logistic Regression (LR), and K-Nearest Neighbors (KNN), the fourth step is to perform data analysis and evaluation to calculate using the chosen data and the efficiency of the proposed models made by the accuracy, precision, F1-score, and recall of each algorithm, the fifth and final step is the end of our work, which consists of analyzing and drawing conclusions based on the graphical and aggregated experimental result. In addition, we can see in Fig. 1 that each component in the model is interconnected and sequential.

2.2 Data collection and dataset preparation

We used a questionnaire to gather data from target software companies for this study, and we produced data found by project managers working for software companies in Ethiopia. The dataset included eighteen attributes classified into three groups (project manager, project context, and business situations) that influence the prediction failure of the knowledge areas in project management, and are collected, and prepared based on the criteria of unambiguity, consistency, practicability, and measurability [11].

There are ten knowledge areas or output classes, as indicated in Table 1, namely, PCCM, PCM, PHRM, PIM, PPM, PQM, PRM, PSTM, PSM, PTM and its failure values for each class is 48, 76, 45, 82, 40, 21, 27, 36, 42, 26 out of 443 total datasets. For prediction, we employed multiclass methods.

Row failed project data: are produced based on the questionnaires from software companies. Processing failed project row data: The gathered row failed project data should be processed for three reasons: missing values should be fixed, data should be standardized, and variable sets should be optimized.

2.2.1 Analyzing attributes

2.2.1.1 Unambiguity Each attribute should have its meaning. Each attribute is subject to one and only one interpretation. The possible values are yes (Y) and no (N). Ambiguous attributes not selected.

2.2.1.2 Consistency Each attribute should be independent of the others. There are three possible values: high (H), medium (M), and low (L). The attributes with the highest consistency value were chosen.

2.2.1.3 Measurability Each attribute should be assigned a value based on the metric. There are three possible values: high (H), medium (M), and low (L). Attributes with higher ease of measurability were chosen.

2.2.1.4 Practicability Each attribute should be feasible in the sense of a particular (sudden) project. There are three possible values: high (H), medium (M), and low (L). Attributes with higher feasibility or practicability were chosen.

There are three possible values in Table 2: High (H), Medium (M), and Low (L). The final list of criteria included an attribute with a higher level of practicability. A characteristic may be added or removed from the final list of influential attributes based on the aforementioned criteria [11]. As a result, nine attributes were chosen as the input for machine learning from 18 preliminary lists of attributes. The project manager has four attributes, three of which

Fig. 1 The proposed model

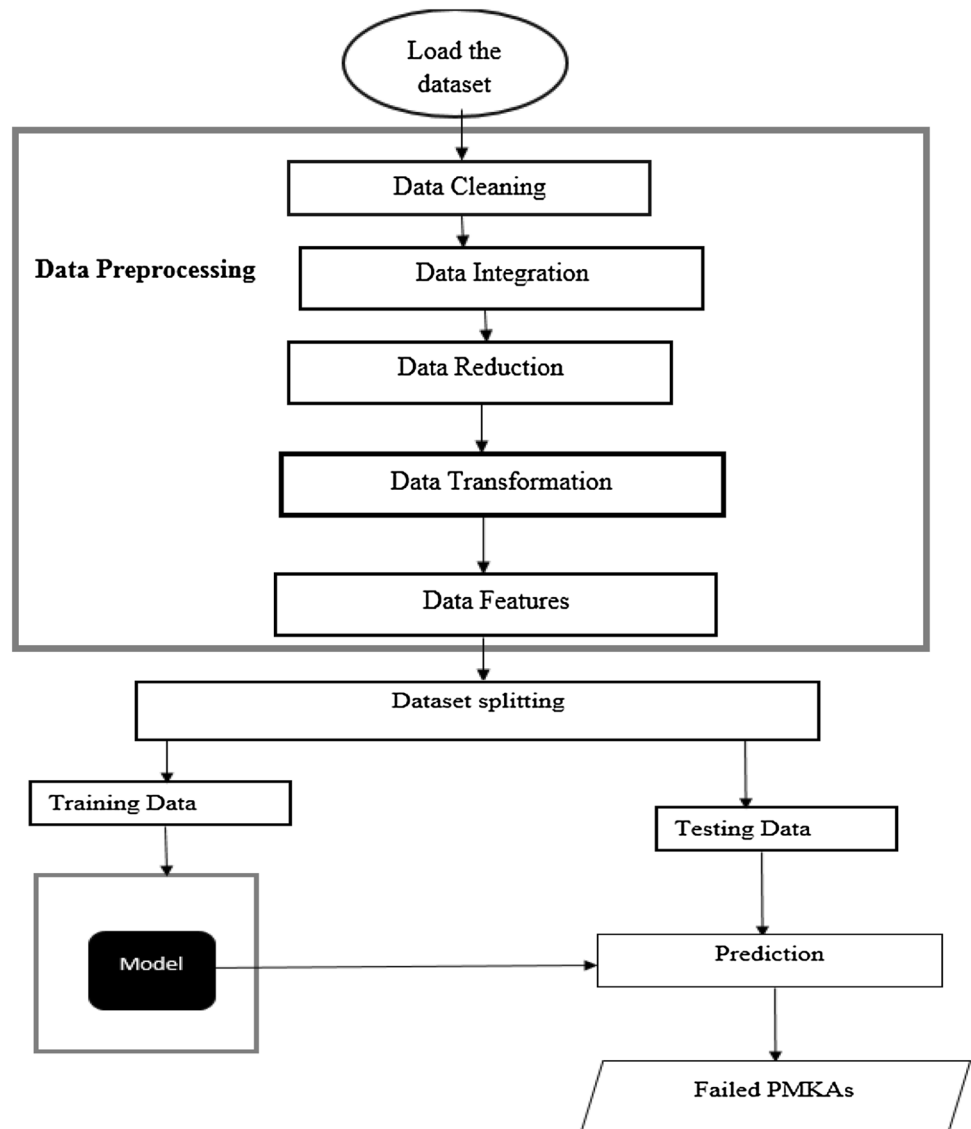


Table 1 Project management knowledge areas failures after annotating the data

Knowledge areas	No of annotated data
Project Communications Management (PCCM)	48
Project Cost Management (PCM)	76
Project Human Resource Management (PHRM)	45
Project Integration Management (PIM)	82
Project Procurements Management (PPM)	40
Project Quality Management (PQM)	21
Project Risk Management (PRM)	27
Project Stakeholders Management (PSTM)	36
Project Scope Management (PSM)	42
Project Time Management (PTM)	26
Total	443

are related to the project’s context and the remaining two to the nature of the company’s situation. Table 2 shows the list of attributes and their results ("P" denotes selected attributes that made it into the final list of attributes, while "F" denotes unselected attributes that did not make it into the final list of attributes).

2.3 Data preprocessing

The information on failed projects was gathered from software companies. As a result, data preprocessing has been completed, which includes data cleansing, duplicate value removal, null value detection, rectification, and balancing. This is where the preprocessing mapping is finished. Because we collect data from a variety of sources, data integration has become a crucial part of the process. We need to make a condensed version of the dataset that

Table 2 Identified attributes with their description and selected attribute using four criteria

Factors	Attributes	Criteria in order of its unambiguity, consistency, measurability, and practicability	Result	References	Descriptions
Project Manager	Education Level	YHHH	P	[11]	The act of doing and seeing things, as well as the act of making things happen to you. The amount of time you have spent on a particular task.: the ability or experience you have acquired from doing something (such as a particular job)
	Experience	YLMM	F	[12]	Your major and the degree you received
	Knowhow of the PMKAs	YMMM	P	[13]	Is it a term for practical knowledge about how to do something?
	Decision maker	YMLL	F	[13]	Someone who makes decisions, especially at a high level in an organization. A strategic decision-maker may be in charge of acquisitions, company growth, or capital investment
	Relevant Job	YMMM	P	[14]	The job you want in terms of the required skills or knowledge
	Education Background	YHHH	P	[11]	The highest grade obtained, or whether the individual has a secondary school (high school) diploma or equivalency certificate
Context of the project	Complexity	NMML	F	[11]	A factor that plays a role in a complex process or circumstance
	Size of the project	YMML	F	[11]	A broad term is used to define the project's overall scope
	Budget	YLMM	F	[15]	A forecast of sales and expenditures for a future period
	Reasons for failed	YMMM	P	[1]	Answers why the project failed?
	Development model followed	YMMM	P	[1]	The steps that take place after a project are completed
	Requirement elicitation Technique Followed	YMMM	P	[1]	Obtaining information about a system's specifications from users, consumers, and other stakeholders
	Number of functionalities	NLHM	F	[13]	The condition of making a profit or gaining money

Table 2 (continued)

Factors	Attributes	Criteria in order of its unambiguity, consistency, measurability, and practicability	Result	References	Descriptions
	Profitability	YMMM	P	[13]	In the minds of its target customers, a brand occupies
	Proper positioning	YLMM	F	[13]	Being one of a kind is a desirable attribute
	Uniqueness	NLMM	F	[1]	The consistency of being persuasive
	Credibility	NMML	F	[13]	The opportunity to take advantage of a deal or circumstance that may lead to a favorable outcome
	Market situations	YMMH	P	[13]	The marketing plans in place and expectations for the future

is smaller in size but retains the original's integrity. Data preparation is the process of transforming data into a format suitable for data modeling, such as converting character values to binary values.

The train test split technique is used to measure the performance of machine learning algorithms that make predictions on data that was not used to train the model.

- A training data set is a set of data that is used to fit a machine learning model.
- Test data set—used to assess the machine learning model's fit.

The purpose of splitting the dataset is to assess the machine learning model's performance on new data that hasn't been used to train the model. This is how we hope to use the model in practice. That is, to fit it to existing data with known inputs and outputs, and then make predictions about future events where we do not have the expected output or target values.

2.3.1 Experimental methods

The experimental methods are mainly aimed at achieving, identifying, and visualizing what factors contribute to project managers and building a prediction model that executes a project whether or not the failed project management knowledge areas were based on the performance of the model.

2.3.2 Model evaluation

This activity is in charge of describing the evaluation parameters of the designed model and its results. The comparison was made between the data categorized by the proposed model system and the manually labeled (categorized) data. Having a common performance appraisal metric for classification and classification accuracy (CA) is used as the final proof of performance.

2.3.2.1 Confusion matrix The confusion matrix assesses the performance of a classification or classifier model on a test dataset. Our target class was multiclass, which means classification tasks that have more than two class labels. So, our target class has ten labels that are 10X10 arrays.

The performance of a classification model is defined by a confusion matrix.

True positives (TP): cases where the classifier predicted that the true and correct class was true.

True negatives (TN): cases in which the model predicted the false and correct class was false.

False positives (FP) (type I error) - Classes predicted true but the correct class was false.

False negatives (FN) (type II error): The classifier predicted false but the correct class was false.

2.3.3 Accuracy

Accuracy means the number of all misclassified samples divided by the total number of samples in the dataset. Accuracy has the best value of one and the worst value of zero.

$$\text{Accuracy} = \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{TN} + \text{FP} + \text{FN})} \quad (1)$$

2.3.4 Precision

Precision (P)—precision is the fraction or percentage of identified or retrieved instances that the classification algorithm considers important. High precision means that most items labeled, for example, as "positive" actually belong to the class "positive" and is defined as precision characterized as the number of isolated true positives times the total sum of true positives and false positives.

$$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})} \quad (2)$$

2.3.5 Recall

A recall is considered a measure of completeness, which is the level of positive examples that are marked as positive. Cluster revision is characterized by the number of isolated true positives times the total number of components that have a place with the positive classes.

$$\text{Recall} = \frac{\text{TP}}{(\text{TP} + \text{FN})} \quad (3)$$

2.3.6 F1 score

F-Measure (F1 score) is defined as the harmonic means of precision and recall which is a measure that joins recall and precision into a single measure of performance. The F1-score was calculated by averaging precision and recall. The relative contribution of precision and recall to the F1-score are equal.

$$\text{F1 - score} = 2 * \frac{(\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})} \quad (4)$$

3 Results and discussion

Experimentation is recognized to necessitate the preparation of a dataset for training and testing purposes, as there is no free, ready-to-use dataset available on the Internet. We used 19 software companies in this study, which took the dataset and split it into three categories based on nine attributes (project manager, project context, and company situations). The collection has 443 records with 9 attributes. The remaining 20% was utilized to test the proposed model, with 80% being used to train the model.

3.1 Experimental results and analysis

After importing the necessary python modules and libraries, the second immediate task is to read the processed data frame (df) in python and check the imported rows. The ID, project manager name, education label, educational experience, relevant work, company name, knowledge about project management knowledge areas (PMKAs), model of development followed, the technique of obtaining requirements followed by market situations, the profitability of the company, reasons for failure and class. From those IDs, the project manager name and project name are not required for the study as the value of each attribute removed the remaining unique values displayed.

Feature engineering—the main goal of feature engineering is to add features that are likely to have an impact on the failed project dataset. The fundamental step in feature engineering is to split the training and test datasets. Out of the 443 rows in the dataset, we used 354 rows for training and 89 rows for tests. Because our datasets are small, we have demonstrated that the data split for training data is high, as high training data and low-test data are recommended for small datasets to get good accuracy.

3.1.1 Results of each prediction algorithm

We employed five methods to predict the failure of the project management knowledge areas in our experiment. K-Nearest Neighbors (KNN), Decision Trees (DT), Logistic Regression (LR), Naive Bayes (NB), and Support Vector Machines (SVM) are all examples of machine learning algorithms.

3.1.1.1 K-nearest neighbors (KNN) prediction algorithm results and analysis We started building a K-Nearest Neighbors model to predict knowledge area failures in software companies after finalizing the data transformation and splitting the train test. The model result is presented in Table 3, we have got the weighted average F1-Score with an accuracy of 87.64%. The values listed in the Support column are classified in the test data into 10 classes.

3.1.1.2 Decision trees prediction algorithm results and analysis As we can see from the confusion matrix report in Table 4, we have got a 90% weighted average accuracy of F1-Score for the decision tree algorithm.

3.1.1.3 Logistic regression prediction algorithm results and analysis The performance measures we have obtained during Logistic Regression findings using the

Table 3 K-Nearest Neighbors (KNN) result using the confusion matrix

No	Precision	Recall	F1-score	Support
0	0.75	0.86	0.80	7
1	0.90	1.00	0.95	18
2	0.92	0.79	0.85	14
3	0.92	0.92	0.92	12
4	0.78	0.78	0.78	9
5	1.00	1.00	1.00	3
6	1.00	0.50	0.67	4
7	0.88	0.88	0.88	8
8	0.78	1.00	0.88	7
9	1.00	0.86	0.92	7
Accuracy			0.88	89
Macro average	0.89	0.86	0.86	89
Weighted average	0.88	0.88	0.87	89

Table 4 Decision Tree (DT) result using the Confusion Matrix

No	Precision	Recall	F1-score	Support
0	1.00	0.71	0.83	7
1	0.95	1.00	0.97	18
2	0.85	0.79	0.81	14
3	0.86	1.00	0.92	12
4	0.73	0.89	0.80	9
5	0.75	1.00	0.86	3
6	1.00	0.50	0.67	4
7	1.00	1.00	1.00	8
8	1.00	1.00	1.00	7
9	1.00	0.86	0.92	7
Accuracy			0.90	89
Macro average	0.91	0.87	0.88	89
Weighted average	0.91	0.90	0.90	89

testing set are given in Table 5. Here, we achieve the performance of 76.40% weighted average F1-Score.

3.1.1.4 Results and analysis of the naïve bayes prediction algorithm The performance measures we have obtained during Naïve Bayes findings using the testing set are given in Table 6. Here, we achieve the performance of 66% weighted average F1-Score.

3.1.1.5 Support vector machine prediction algorithm results and analysis The performance of the Support Vector Machine (SVM) model was also evaluated using the testing set and the obtained performance measures are given in Table 7. From the performance report, we can see that the SVM model achieves a 92.13% weighted average F1-Score.

Table 5 Logistic regression (LR) result using the confusion matrix

No	Precision	Recall	F1-score	Support
0	0.67	0.57	0.62	7
1	0.79	0.83	0.81	18
2	0.85	0.79	0.81	14
3	0.69	0.75	0.72	12
4	0.78	0.78	0.78	9
5	0.50	0.33	0.40	3
6	1.00	0.50	0.67	4
7	0.73	1.00	0.84	8
8	0.71	0.71	0.71	7
9	0.86	0.86	0.86	7
Accuracy			0.76	89
Macro average	0.76	0.71	0.72	89
Weighted average	0.77	0.76	0.76	89

Table 6 Naïve Bayes (NB) result using the confusion matrix

No	Precision	Recall	F1-score	Support
0	0.35	1.00	0.52	7
1	0.95	1.00	0.97	18
2	1.00	0.29	0.44	14
3	0.83	0.83	0.83	12
4	1.00	0.33	0.50	9
5	0.18	1.00	0.30	3
6	1.00	0.50	0.67	4
7	0.75	0.38	0.50	8
8	1.00	0.29	0.44	7
9	1.00	0.86	0.92	7
Accuracy			0.65	89
Macro average	0.81	0.65	0.61	89
Weighted average	0.87	0.65	0.66	89

Table 7 Support vector machine (SVM) result using the confusion matrix

No	Precision	Recall	F1-score	Support
0	0.86	0.86	0.86	7
1	0.90	1.00	0.95	18
2	0.86	0.86	0.86	14
3	1.00	0.92	0.96	12
4	0.89	0.89	0.89	9
5	0.75	1.00	0.86	3
6	1.00	1.00	1.00	4
7	1.00	1.00	1.00	8
8	1.00	1.00	1.00	7
9	1.00	0.71	0.83	7
Accuracy			0.92	89
Macro average	0.93	0.92	0.92	89
Weighted average	0.93	0.92	0.92	89

3.2 Validation of the model

Validation ensures the model does not overfit or underfit during the training process. To prevent the model from learning too much or too little from the training set, a dropout layer or early stopping can be added. When a model learns too much on the training set, it performs well in the training phase but fails miserably in the testing phase. In data it has never seen before, it performs poorly. The accuracy of training is high, but the accuracy of testing is extremely low. Here is the validation for our model.

Visualizing the training vs. validation accuracy over a number of epochs is an excellent approach to see if the model has been properly trained. This is necessary to ensure that the model is not undertrained or overtrained to the point that it begins to memorize the training data, reducing its capacity to predict effectively. We employed early Stopping and epochs = 100 in our model in Fig. 2, with nine attributes as the input layer, two hidden layers, and ten classes as the output layer. Early Stopping entails keeping track of the loss on both the training and validation datasets (a subset of the training set not used to fit the model). The training process can be interrupted as soon as the validation set's loss begins to exhibit evidence of overfitting. We've increased the number of epochs and are certain that training will finish as soon as the model begins too overfit. From the plot of accuracy, as given in Fig. 2, we can see that the model could probably be trained a little more as the trend for accuracy on both datasets is still rising for the last few epochs. We can also see that the model has not yet over-learned the training dataset, showing comparable skills on both datasets.

From the plot of loss, we can see that the model has comparable performance on both train and validation datasets (labeled test). If these parallel plots start to depart consistently, it might be a sign to stop training at an earlier

epoch. The validation loss is constantly reduced throughout the training procedures, as given in Fig. 3, indicating that there is no overfitting.

3.3 Discussion of the results

Table 8 shows, that the Support Vector Machine has stood out due to its prediction accuracy.

First experiment: In the findings of the confusion matrix of the test data for the K-Nearest Neighbors (KNN) prediction model, which is presented in Table 8, 78 of them were correctly identified and the remaining 11 were mistakenly classified. Finally, K-Nearest Neighbors (KNN) was shown to be 87.64% accurate.

Second experiment: In the findings of the confusion matrix of the test data for the Decision Tree (DT) prediction model, which is presented in Table 8, 80 of them were correctly identified and the remaining 9 were mistakenly classified. Finally, Decision Trees (DT) were able to reach an accuracy of 90%.

Third experiment: In the findings of the confusion matrix of the test data for the Logistic Regression (LR) prediction model, which is illustrated in Table 8, 68 of them were correctly identified and the remaining 21 were mistakenly classified. Finally, the accuracy of the Logistic Regression (LR) was 76.4%.

Fourth experiment: In the confusion matrix findings for the Naïve Bayes (NB) prediction model, which is illustrated in Table 8, 58 of the test data were correctly identified, while the remaining 31 were mistakenly classified. Finally, the accuracy of Naive Bayes (NB) was 66%.

Fifth experiment: In the confusion matrix of the test data, 82 of them were correctly identified, while the remaining 7 were mistakenly classified, according to the Support Vector Machine (SVM) prediction model which is

Fig. 2 Validation of model accuracy

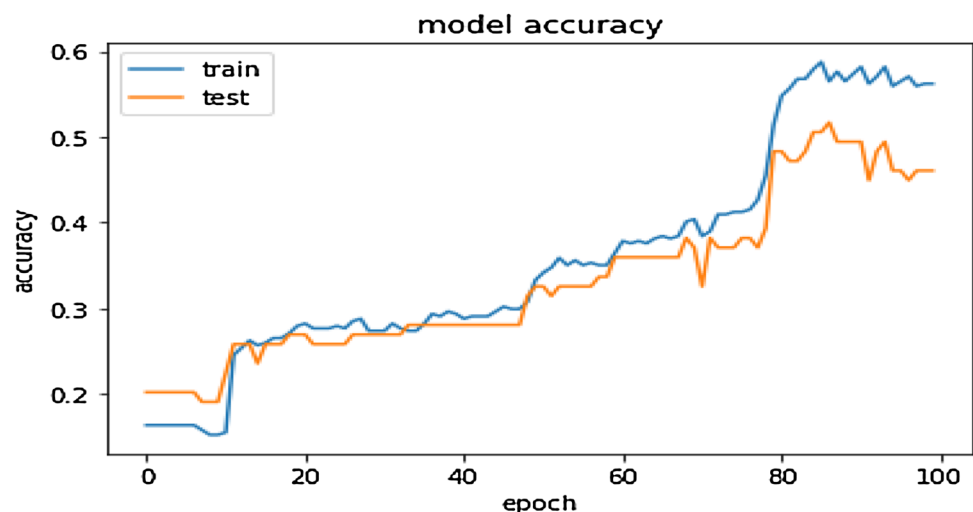
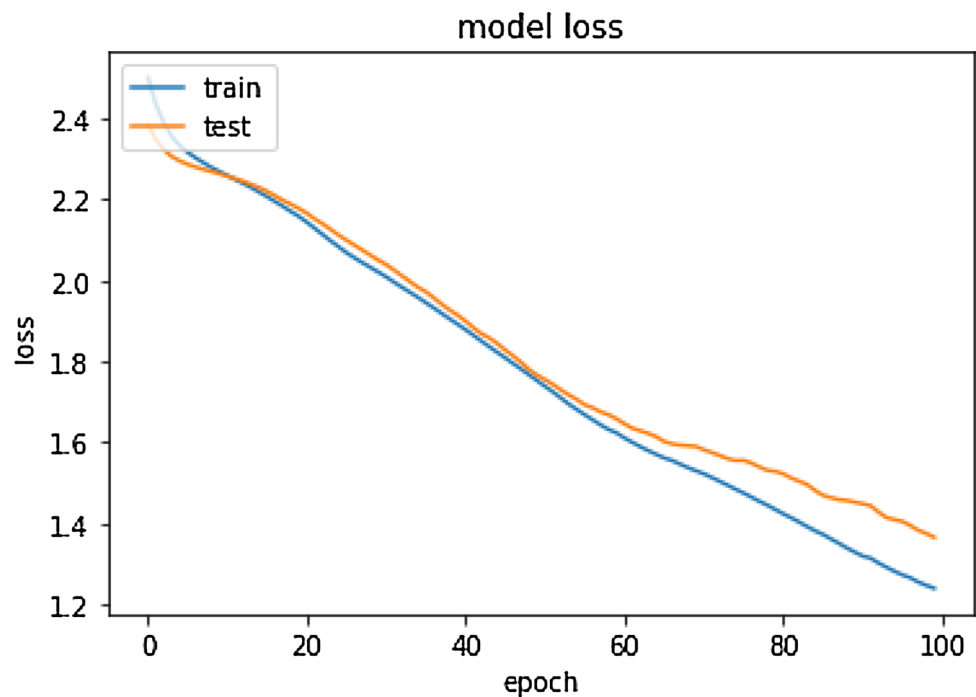


Fig. 3 Validation of model loss



included in Table 8. Finally, the Support Vector Machine (SVM) attained a 92.13% accuracy.

The following are some of the reasons why the Naive Bayes (NB) prediction performed poorly in our experiment: first, if the test dataset contains a categorical variable of a category that was not present in the training dataset, the Naive Bayes (NB) model assigns zero probability, which is known as 'frequency zero' [16]. In addition, to tackle this problem, we applied a smoothing technique. Second, the Naive Bayes (NB) algorithm is well-known for being an ineffective estimator [16]. Therefore, you should not take the probability outputs or predict probability too seriously. Third, the Naive Bayes (NB) algorithm assumes that all the features are independent classes [17].

In our experiment, Logistic Regression (LR) predicted achieving lower performance next to Naive Bayes (NB) because of the following reasons. First, the assumption of linearity between the dependent and independent

variables is a key constraint of Logistic Regression (LR) [17]. Second, Logistic Regression requires average or non-multicollinearity between independent variables [16]. Third, non-linear problems cannot be solved with logistic regression since it has a linear decision surface [18]. Linearly separable data is unusual in real-world situations. As a result, non-linear characteristics must be converted, which can be accomplished by increasing the number of features to segregate data linearly in higher dimensions. Fourth, when creating a model, only the most critical and relevant features should be employed. Otherwise, the probabilistic predictions made by the model lead to incorrect, and the model's predictive value may degrade [18]. Fifth, each training instance must be self-contained from the rest of the dataset instances [17]. If they are related in some way, the model tries to give those specific training instances. As a result, matching data or repeated measurements such as training data should not be used. Some scientific study procedures, for example, rely on several observations of the same individual. In such conditions, this method is ineffective.

In our experiment, the prediction of the K-Nearest Neighbors (KNN) achieved less performance together with the Logistics Regression (LR) and Naive Bayes (NB) due to the following reasons. First, K-Nearest Neighbors (KNN) can suffer from biased class distributions, if a certain class is very frequent in the training set, it tends to master the majority vote of the new instance (large number = more common) [17]. In our data, if the management of the integration class projects is more frequent, the

Table 8 Comparison of models on test data

Algorithms	Correctly predicted	Incorrectly predicted	Accuracy (%)
K-nearest neighbors (KNN)	78	11	87.64
Logistic regression (LR)	68	21	76.4
Support vector machine (SVM)	82	7	92.13
Decision trees (DT)	80	9	90
Naive Bayes (NB)	58	31	66

K-Nearest Neighbors (KNN), the prediction assumes that the new data is the management of project integration. Second, the accuracy of the K-Nearest Neighbors (KNN) can be severely degraded with high-dimensional data [19]. Because there is little difference between the nearest and farthest neighbor. That is why K-Nearest Neighbors (KNN) is not good for high-dimensional data. Third, the algorithm gets significantly slower as the number of features increases [17]. Fourth, needs a large number of samples for acquiring better accuracy [20]. Therefore, our data do not have a large number of samples. Fifth, the algorithm is hard to work with categorical features [16]. Therefore, our data has categorical features.

In our experiment, the predictions of the Decision Tree (DT) achieved less performance together with the K-Nearest Neighbors (KNN), the Logistic Regression (LR), and the Naïve Bayes (NB), respectively, due to the following reasons: First, Decision Trees (DT) suffer in overfitting [17]. This is the main problem of the Decision Trees (DT). It usually results in data overfitting, which leads to incorrect predictions. It keeps creating new nodes to fit the inputs (even noisy data), and the tree eventually gets too complex to interpret. It loses its ability to generalize in this way. It performs very well on the trained data but starts making many mistakes on the unseen data. Second, High variance [16] as mentioned in the first concept, the decision tree generally leads to the overfitting of data. Overfitting causes a lot of variances in the output, which leads to many inaccuracies in the final estimates and shows a lot of inaccuracy in the findings. Obtained zero bias (overfitting), resulting in significant variance. Third, Unstable [21], adding new data, the point can lead to regeneration of the overall tree and all nodes need to be recalculated and recreated. Fourth, affected by noise [17], a little bit of noise can make it unstable which leads to wrong predictions.

The prediction of the Support Vector Machine (SVM) achieved better performance among others due to the following reasons. First, it works more effectively in categorical data [21]. For this reason, our dataset is categorical. Second, it works relatively well even in smaller datasets because the algorithm does not rely upon the complete data [20].

Third, it works more effectively for high-dimensional datasets because the complexity of the training data set does not depend on the dimensionality of the dataset [18]. Fourth, a Support Vector Machine (SVM) is extremely useful when we have no prior knowledge of the data [17].

Using traditional machine learning methods rather than deep learning techniques has several advantages. The Support Vector Machine outperforms the other techniques, and it's better for small datasets with outliers and non-parametric models, as we showed in our results. Deep learning, on the other hand, is used when the complexity

grows as the number of training samples grows when large datasets are required to function well when a complicated structure necessitates learning multi-layered features, and when high experience is required. It is used in a variety of industries, from automatic leadership to medical devices. Finally, while our dataset is limited, we apply typical machine learning algorithms to achieve the best results.

4 Conclusions

Due to its profitability, the development of software-based systems and the founding of software companies have increased in recent years. However, in any business, especially a software company, some projects can fail. One way to avoid software project failure is to fill the skill gaps of software project managers to increase their knowledge areas of project management. Because knowledge areas are the key issues associated with software project management. In our country, Ethiopia, software projects are not led by professionals. The functionality, schedule, budget management, risk of software projects is not managed properly due to a lack of knowledge about Project Management Knowledge Areas (PMKAs).

The machine learning model used in this work is intended to assist project managers in predicting the failure of project management knowledge areas (PMKA) for a specific project. As a result, a literature review was conducted to identify the features, which were then evaluated using unambiguity, consistency, measurability, and practicability criteria to discover the most important attributes in predicting failed knowledge areas. Finally, a machine learning model has been developed to predict failed Project Management Knowledge Areas (PMKAs). The model included three factors: project manager context, project context, and company context. This research work had a total of 443 records and 9 attributes to predict the failure of the Project Management Knowledge Areas (PMKAs). Noise removal and management of missing values were performed to prepare the dataset for the experiments. To build the model, we have used machine learning algorithms such as Decision Trees (DT), Logistic Regression (LR), Naïve Bayes (NB), K-Nearest Neighbors (KNN), and Support Vector Machine (SVM). Accuracy, precision, and recall were used to evaluate the performance of the developed model. The model is evaluated by comparing its performance or results with the actual data (the data we have at hand) that have the values of the nine attributes and ten domains of knowledge of project management. The results demonstrated that the Support Vector Machine (SVM) technique is more efficient than other candidate algorithms at predicting failed Project Management

Knowledge Areas (PMKAs). In terms of accuracy, the significance of the produced model will change the progress of anticipating failed areas of project management expertise.

5 Future works

In terms of future research, we recommend the following:

1. Conduct various types of empirical research on predicting and reporting the effectiveness of project management knowledge areas to assist project managers, and predict project management knowledge areas failure by compiling multiple failed project datasets using deep learning approaches and comparing them with our results.
2. Test the effect of attribute reduction on the performance of selected algorithms or other machine learning algorithms by adding more features and criteria.

Funding The authors have not disclosed any funding.

Data availability The datasets and source code analyzed during the current study are publicly available at this link: <https://colab.research.google.com/drive/1k84ZYMIXW4gjpKn1BQDjiEfgPUzT4C3#scrollTo=34hjffZL2Oj9>.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal ties that may have influenced our work.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Javed SA, Liu S (2017) Evaluation of project management knowledge areas using grey incidence model and AHP. In: 2017 international conference on grey systems and intelligent services (GSIS), pp 120–120. <https://doi.org/10.1109/GSIS.2017.8077684>
2. Houston SM (2017) Project knowledge areas. In: The project manager's guide to health information technology implementation, 2nd Edition, p 16. <https://doi.org/10.1201/b22038-4/project-knowledge-areas-susan-houston>. Accessed 6 Apr 2022
3. Oun TA, Blackburn TD, Olson BA, Blessner P (2016) An enterprise-wide knowledge management approach to project management. *EMJ* 28(3):179–192. <https://doi.org/10.1080/10429247.2016.1203715>
4. Saleem N (2019) Empirical analysis of critical success factors for project management in global software development. In: 2019 ACM/IEEE 14th international conference on global software engineering (ICGSE), pp 68–71. <https://doi.org/10.1109/ICGSE.2019.00025>
5. Lehtinen TOA, Mäntylä MV, Vanhanen J, Itkonen J, Lassenius C (2014) Perceived causes of software project failures—an analysis of their relationships. *Inf Softw Technol* 56(6):623–643. <https://doi.org/10.1016/j.infsof.2014.01.015>
6. Klotins E, Unterkalmsteiner M, Gorschek T (2019) Software engineering in start-up companies: an analysis of 88 experience reports. *Empir Softw Eng* 24(1):68–102. <https://doi.org/10.1007/s10664-018-9620-y>
7. Knodel J, Manikas K (2015) Towards a typification of software ecosystems. In: Fernandes J, Machado R, Wnuk K (eds) *Software Business. ICSOB 2015. Lecture Notes in Business Information Processing*, vol 210. Springer, Cham. https://doi.org/10.1007/978-3-319-19593-3_5
8. Alojail M, Bhatia S (2020) A novel technique for behavioral analytics using ensemble learning algorithms in E-commerce. *IEEE Access* 8:150072–150080. <https://doi.org/10.1109/ACCESS.2020.3016419>
9. Sheikh RA, Bhatia S, Metre SG, Faqih AYA (2022) Strategic value realization framework from learning analytics: a practical approach. *J Appl Res High Educ* 14(2):693–713. <https://doi.org/10.1108/JARHE-10-2020-0379>
10. Gandhi P, Khan MZ, Sharma RK, Alhazmi OH, Bhatia S, Chakraborty C (2022) Software reliability assessment using hybrid neuro-fuzzy model. *Comput Syst Sci Eng* 41(3):891–902. <https://doi.org/10.32604/csse.2022.019943>
11. Ramadan N, Abdelaziz A, Salah A (2016) A hybrid machine learning model for selecting suitable requirements elicitation techniques. *Int J Comput Sci Inf Secur* 14(6):1–12
12. Komi-Sirviö S (2004) Development, and evaluation of software process improvement methods. VTT
13. Jain R, Suman U (2018) A project management framework for global software development. *ACM SIGSOFT Softw Eng Notes* 43(1):1–10. <https://doi.org/10.1145/3178315.3178329>
14. Wanberg CR, Ali AA, Csillag B (2020) Job seeking: the process and experience of looking for a job. *Annu Rev Org Psychol Org Behav* 7:315–337. <https://doi.org/10.1146/annurev-orgpsych-012119-044939>
15. Eastham J, Tucker DJ, Varma S, Sutton SM (2014) PLM software selection model for project management using hierarchical decision modeling with criteria from PMBOK® knowledge areas. *EMJ* 26(3):13–24. <https://doi.org/10.1080/10429247.2014.11432016>
16. Dey A (2022) Machine learning algorithms: a review. <https://ijcsit.com/docs/Volume%207/vol7issue3/ijcsit2016070332.pdf>. Accessed 6 Apr 2022
17. Hassanat AB, Abbadi MA, Altarawneh GA, Alhasanat AA (2014) Solving the problem of the K parameter in the KNN classifier using an ensemble learning approach. <http://sites.google.com/site/ijcsis/>
18. Osisanwo FY, Akinsola JE, Awodele O, Hinmikaiye JO, Olakanmi O, Akinjobi J (2017) Supervised machine learning algorithms: classification and comparison. *Int J Comput Trends Technol* 48(3):128–138. <https://doi.org/10.14445/22312803/JCTT-V48P126>
19. Bhatia N, Vandana A (2010) Survey of nearest neighbor techniques. *Int J Comput Sci Inf Secur*. <https://doi.org/10.48550/arXiv.1007.0085>

20. Taneja S, Gupta C, Goyal K, Gureja D (2014) An enhanced K-nearest neighbor algorithm using information gain and clustering. In: International conference on advanced computing and communication technologies, ACCT, pp 325–329. <https://doi.org/10.1109/ACCT.2014.22>
21. Mece EK, Binjaku K, Paci H (2020) The application of machine learning in test case prioritization—a review. Eur J Electr Eng Comput Sci. <https://doi.org/10.24018/ejece.2020.4.1.128>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.