Journal of Engineering Education Transformations,
Volume No. 37, January 2024 Special Issue, eISSN 2394-1707

Enhancing Student Performance Prediction in Higher Education: A Data-Driven Approach

R. Parkavi[1], P. Karthikeyan[2]
[1,2]Department of Information Technology, Thiagarajar College of Engineering, Madurai
[1] rpit@tce.edu [2] karthikit@tce.edu

Abstract **— Student performance is a crucial factor in higher education institutions, as admission to a high-quality institution often relies on a good academic record. The primary objective of this paper is to predict student performance by considering their personal and academic achievements. By identifying poorly performing students, teachers can offer timely guidance and support to improve their academic outcomes. However, predicting student performance becomes challenging due to the processing of a large amount of data, including both numerical and non-numerical values. This work aims to determine the most effective prediction algorithm and identify the key variables in student data to enhance student performance and success rates through classification techniques using educational databases from both universities and schools. The proposed system leverages this data to predict a student's next year's result (GPA). The dataset used for this experiment is locally obtained from third-year students, encompassing their core subject results (6 subjects, 2 laboratories), and personal details. The methods employed in this work include the Multiple Linear Regression, Naive Bayes and Decision Tree. The extracted factors impacting the results will help students prepare better in advance. The highest accuracy achieved in this prediction is 88.44%, bringing significant benefits to students, teachers, and educational institutions.**

Keywords**— Prediction; Multiple Linear Regression; GPA; Decision Tree; Naive Bayes;**

JEET Category**— Research.**

I. INTRODUCTION

In the realm of education, the diversity of learning abilities among individuals makes it challenging to accurately predict student performance. This limitation often results in unexpected underperformance during exams. To address this issue and empower learners, this paper's central goal is to establish a predictive framework for anticipating student performance. By amalgamating personal attributes and academic accomplishments, the model seeks to offer insights into students' academic trajectories. Detecting students who might be struggling holds immense potential for educators to intervene proactively and guide them toward improved outcomes. The study encompasses a substantial number of students, focusing on transforming non-numerical data into distinct features. Success is delineated as those surpassing a threshold of 50, thereby establishing a graduation rate calculated as a percentage of successful students. The predictive model hinges on a holistic examination of both academic and non-academic factors, entailing numerical and non-numerical values. Integral to this approach are machine learning techniques like the Multiple Linear Regression Model, which extrapolate patterns and rules from user data, facilitating automated decision-making and continuous enhancement. The study's purview extends across diverse applications, encompassing search engines, image processing for object recognition, and personalized product recommendations. While exploring machine learning algorithms, the research distinguishes between supervised and unsupervised approaches, primarily utilizing the former, which leverages known target values for training. The paper underscores the pivotal role of precise performance prediction in bolstering educational guidance and mitigating student attrition rates. Methodologies such as Naive Bayes classification, Decision Trees, and Multiple Linear Regression take center stage. Performance metrics like precision, recall, and F-measure feature prominently for algorithm assessment, with feature engineering's significance highlighted. The structural layout of the paper spans chapters dedicated to methodology, dataset, evaluation metrics, and results.

This paper was submitted for review on August 31, 2023 . It was accepted on November, 15, 2023.
Corresponding author: R. Parkavi, Department of Information Technology, Thiagarajar College of Engineering, Tamil Nadu, India.
e-mail: rpit@tce.edu
Copyright © 2024 JEET.

II. LITERATURE SURVEY

Higher education institutions' primary goal is to provide students with high-quality education. The necessity to identify children who perform poorly has become increasingly important, and most teachers have depended on calculating the average of exam grades. The primary goal of the work done by Sudais, M., et al. (2022) is to forecast and identify students who may fail semester exams. This would assist teachers in offering extra assistance to such children. The data that was analyzed comprised students' transcript data, which contained their CGPA and grades in all university courses. The machine learning methods employed in this study included the Naive Bayes classifier, the Neural Network, the Support Vector Machine, and the Decision Tree classifier. A comparison of the accuracy outcomes of the algorithms utilized has been performed. This study reveals that machine learning is effective for prediction, but there is still much more work to be done with this technology.

Educational data mining has gained significant attention in recent years as a means of uncovering valuable insights from educational data. Numerous techniques have emerged to extract these insights, aiding educational institutions in refining their teaching methods and enhancing the overall learning process. Such improvements naturally lead to a boost in student performance and overall educational outcomes. Amrieh, E. A., et al (2016) introduce an innovative predictive model for anticipating student performance. It introduces a unique set of data attributes referred to as "student behavioral features," which encompass students' interactions with e-learning management systems. The efficacy of this predictive model is evaluated through various classifiers, including Artificial Neural Networks, Naïve Bayesian, and Decision trees. To further enhance the performance of these classifiers, ensemble methods such as Bagging, Boosting, and Random Forest are applied. The findings highlight a compelling connection between student behaviors and their academic achievements. By incorporating behavioral features, the proposed model demonstrates a remarkable up to 22.1% accuracy improvement compared to models lacking these features. Additionally, the utilization of ensemble methods leads to an accuracy improvement of up to 25.8%. The model's testing on newly enrolled students showcases an accuracy rate exceeding 80%, which serves to affirm the model's reliability.

The success of educational institutions is dependent on student achievement. Academic accomplishment, in particular, is one of the indicators used to rank top-tier universities. Despite the vast amount of educational data available, precisely forecasting student performance becomes more difficult. The primary reason for this is a lack of research in various machine learning methodologies. As a result, educators must investigate effective strategies for predicting and analyzing student performance while spotting flaws to improve educational outcomes. This study done by Alsariera, Y. A., et al. (2022) looked into existing machine learning (ML) techniques and crucial features for predicting student success. A comprehensive search of multiple internet databases yielded related papers published between 2015 and 2021. Thirty-nine studies were chosen and assessed. According to the findings, six ML models were primarily used: decision tree, artificial neural networks, support vector machine, K-nearest neighbor, linear regression, and Naive Bayes. The findings also showed that ANN beat other models in terms of accuracy. Furthermore, the most common input variables (e.g., predictive features) utilized to predict student achievement was academic, demographic, internal assessment, and family/personal attributes. The analysis of this study shows an increase in research on this subject as well as a wide spectrum of machine learning techniques used. Simultaneously, the available evidence revealed that can be useful in detecting and enhancing several academic performance areas.

In today's competitive world, an institute's ability to forecast student performance, classify individuals based on their talents, and seek to improve their performance in future examinations is crucial. Students should be advised ahead of time to focus their efforts on a specific area to boost their academic performance. This type of study can help an institute reduce its failure rates. Pallathadka, H., et al (2023) predict students' achievement in a course based on their prior performance in related courses. Data mining is a set of techniques for discovering hidden patterns in large amounts of existing data. These patterns could be useful for analysis and prediction. The term "education data mining" refers to a grouping of data mining applications in the field of education. These programs deal with data analysis from students and teachers. The analysis could be used to categorize or predict. Machine learning algorithms such as Naive Bayes, ID3, C4.5, and SVM are being researched. The experimental investigation makes use of the UCI machinery student performance data set. Algorithms are evaluated based on characteristics such as accuracy and error rate.

Reliable prediction of individual learning performance can help students receive timely support and improve their learning experience. In this study, two well-known machine learning approaches, support vector machine (SVM) and artificial neural network (ANN), are hybridized using a teaching-learning-based optimizer (TLBO) to predict student test performance (failure courses and final exam scores) with high accuracy. The TLBO algorithm performs the feature selection process of both ANN and SVM techniques for the given classification and regression problems, determining the ideal combination of input variables. Furthermore, the ANN architecture is determined in parallel with the feature selection process using the TLBO algorithm. Finally, four hybrid models comprising anonymized information on both discrete and continuous variables were built for learning analytics utilizing a large data set. By using hybridized machine learning models and TLBO, Arashpour, M., et al. (2022) delivers scientific utility by improving forecasts of student exam performance. Individual performance prediction in practice can assist in advising students about their academic progress and taking appropriate decisions such as dropping units in later teaching periods. It can also assist scholarship providers in tracking student progress and providing support.

Predicting student performance is one of the most critical concerns in educational data mining (EDM), which is gaining popularity. By predicting students' performance, we can identify students who are at risk of academic failure and assist instructors in taking actions such as guidance or interventions to assist learners as early as possible or carry out continuous evaluations of learners to optimize learning paths or personalized learning resource recommendations. In this survey, Xiao W, et al. (2022) reviewed the 80 most important studies on predicting student performance using EDM methods from 2016 to 2021, synthesized the procedure for developing a prediction model of student performance, which consists of four phases and ten key steps, and compared and discussed the most recent EDM methods used in each step. They examined the difficulties encountered by prior studies in three areas and proposed future directions for data collecting, EDM methodologies, and prediction model interpretation. This survey gives a thorough overview and practical assistance for researchers on this topic, as well as suggestions for future research.

Educational data mining has evolved into a powerful tool for uncovering hidden patterns in educational data and forecasting students' academic performance. This paper

presents a new model based on machine learning algorithms to predict undergraduate students' final test scores using their midterm exam grades as the source data. The performances of machine learning techniques such as random forests, nearest neighbor, support vector machines, logistic regression, naive bayes, and k-nearest neighbor were calculated and compared to predict the students' final exam marks. The dataset included academic accomplishment grades from 1854 students who took the Turkish Language-I course at a Turkish state university during the autumn semester of 2019-2020. According to the results, the proposed model attained a classification accuracy of 70-75%. Only three sorts of parameters were used to make the predictions: midterm exam marks, Department data, and Faculty data. Such data-driven research is critical for building a framework for learning analysis in higher education and contributing to decision-making processes. Finally, Yagci, M. (2022) contributes to the early identification of students at high risk of failure and identifies the most effective machine learning methods.

At the Karlsruhe Institute of Technology (KIT), the authors use two machine learning methodologies, logistic regressions, and decision trees, to predict student dropout. The models are built using examination data, which is freely available at all universities and does not require any special collecting. As a result, they present a methodical methodology that can be easily implemented in other institutions. Kemper, L., et al. (2020) discover that decision trees outperform logistic regressions marginally. However, after three semesters, both techniques produce excellent prediction accuracies of up to 95%. After the first semester, a classification with more than 83% accuracy is already attainable.

In many application scenarios, using machine learning to forecast student dropout in higher education institutions and programs has proven to be useful. There are three main factors in a machine learning-based approach to detecting students at risk of dropping out: the selection of features likely to influence a partial or total stop of the student, the selection of the algorithm to implement a prediction model, and the selection of the evaluation metrics to monitor and assess the credibility of the results. Oqaidi, K., et al. (2022) aim to provide a diagnosis of machine learning techniques used to detect student dropout in higher education programs, as well as a critical analysis of the limitations of the models proposed in the literature. The main contribution of this article is to present recommendations that may address the lack of a global model that can be generalized in all higher education institutions, at least within the same country or university.

To effectively reduce student attrition, it is critical to identify the underlying causes of attrition and which students are at risk of dropping out. Berens, J., et al. (2018) create an early detection system (EDS) that predicts student dropout based on administrative student data from a public and private university. Instead of relying on a single approach, we employ the AdaBoost Algorithm to combine regression analysis, neural networks, and decision trees to produce an EDS that can be used at any German university. The public institution's prediction accuracy at the end of the first semester is 79%, whereas the private University of Applied Sciences' prediction accuracy is 85%. After the fourth semester, the public university's accuracy improved to 90% while the private university of applied sciences' accuracy improved to 95%.

The primary goal of higher education revolves around delivering exceptional learning experiences to students. To enhance the pinnacle of quality within the realm of higher education, one pivotal approach is to proactively anticipate student enrollment trends. Yadav, S.K., et al (2012) introduce a data mining initiative aimed at constructing predictive models to bolster student retention strategies. By leveraging this approach, novel entrant information can be fed into these predictive models, subsequently generating concise yet accurate lists. These lists serve to pinpoint students who are more likely to benefit from additional assistance through targeted retention programs. This study meticulously scrutinizes the efficacy of these predictive models, forged through the utilization of advanced machine learning algorithms. The findings underscore the capability of select machine learning algorithms to formulate robust predictive models derived from historical student retention data.

From the literature review it provides valuable insights into the use of machine learning and data mining techniques for predicting student performance and identifying factors that contribute to academic success or failure. However, there are several research gaps that emerged. A comprehensive exploration of predictive analytics specifically focused on student attrition. Understanding the factors leading to dropout and developing effective preventive measures could significantly contribute to improving student retention rates.

The objective of this study to predict and enhance student performance and success rates in higher education institutions by developing an effective prediction system that utilizes personal and academic data to forecast students' future academic achievements.

III. METHODOLOGY

The process of predicting future events is called Predictive analytics, it is done on previously unseen data and by using the data model is generated. The aim is to predict the student's next semester's GPA by using previous data variables. The column GPA is the dependent variable and the remaining variable is the independent variable. The GPA column gives grade values of students and the value *"9-10"* defines grade *S*, *"8-9"* is grade *A*, *"7-8"* is grade *B*, *"6-7"* is grade *C, "5-6"* is grade *D* and *<5* is grade *E* that is failing. By using an ML algorithm student data model is generated called the prediction model that provides results from the value called a dependent variable and the remaining variable is taken as input.

The dataset used for generating the model used for prediction derived from the previously obtained dataset is called the training dataset. Then generated model is applied to another dataset to determine its performance and the data used for testing is called test data. To avoid the problem of overfitting two separate datasets are used to make the generated model more flexible for any other new dataset. The problem such as the model generated gives a good result with its data but less result with different data and this problem is called overfitting. To reduce overfitting data is divided into train and test sets.

A. Algorithms Used For Prediction

There are various algorithms and methods to generate and evaluate a predicted model. This work differentiates 3 different algorithms such as (i) multiple linear regression, (ii) decision trees, and (iii) naive bayes classifier. These methods have a similar procedure for predicting both dependent and independent variables but use different mathematical methods.

B. Sklearn

Sklearn is the machine learning algorithm included in the work where it is stored in the variable in f1_score which is the student GPA. The total number of students is given and non-numerical values are considered as features and the student whose values are greater than 50 is considered as passed and the rest failed in the evaluation. The graduation rate of the class is determined by the total number of students who are passed by a total number of students multiplied by 100. The graduation rate is determined by percentage.

C. Predictive analytics of featured non-numerical value

The performance of student prediction is determined by both academic results which are numerical value also known as continuous value and non-academic results that is non-numerical value also known as discrete values. The calculation of academic results is easy because of continuous value but prediction and calculation of discrete value are difficult because the discrete value is based on multiple factors that cannot be determined by calculation or formula so predictive analytics is used on feature columns all the non-numerical values are transformed into featured columns, it contains important and non-important columns the important column becomes the targeted column and non-important columns become support columns the target column is GPA column and others are support columns.

D. Dummy Variable

A dummy variable is a good method for categorical data that includes fixed, non-numerical, and unordered number data values such as male/female gender values. These values are represented as 0 and 1. If the data variables consist of two or more variables that are highly correlated then the condition is called Dummy Variable Trap. The method is one variable is used to predict another variable and the solution to this problem is to drop one categorical variable. In the dataset, the h dummy variable can have an h-1 variable used in the prediction model.

E. Multiple Linear Regression

MLR is used to find the relationship between two or more variables by fitting a linear equation to observed data that are explanatory. Each *x*-independent variable is associated with the y-dependent variable. Here *p* is explanatory variables of x_1, x_2, ... x_p respectively, and the regression line is defined to be (3.1)

$$Y_y = B_0 + B_1x_1 + B_2x_2 + ... + B_px_p \quad \textbf{(3.1)}$$

Here Y_y is the mean response that changes with explanatory variables. When mean Y_y changes observed values of *y* also change and have the same value of standard deviation σ. The parameters of B_0, B_1, ..., *and* B_p have fitted values of b_0, b_1, ..., and bp respectively. This change of values in both observed and mean in the MLR model is termed and expressed as (3.2)

$$DATA = FIT + RESIDUAL \quad \textbf{(3.2)}$$

where 'FIT' represents the expression (3.3)

$$B0 + B1x1 + B2x2 + ... + Bpxp \quad \textbf{(3.3)}$$

The term 'RESIDUAL' represents the deviations of Y_y mean with observed values *y* have distributed value of mean *0* and variance σ. The model deviations ε are. The MLR model for *n* given observations is (3.4)

$$yi = B0 + B1xi1 + B2xi2 + ... Bpxip + \varepsilon i \text{ for } i = 1,2, ... n \quad \textbf{(3.4)}$$

In the least-squares model, the best line for fitting for observed data is derived by using minimizing the sum of the squares of the vertical deviations from each data point to the line. The values are squared and added and positive and negative values are not canceled. The statistical software calculates least-squares estimates of b_0, b_1, ... b_p.

The equation $b_0 + b_1x_{i1} + ... + b_px_{ip}$ FIT value is denoted as $\hat{y}$i, and the residuals value e_i is equal to $y_i - \hat{y}_i$, and the difference between the observed and fitted values is equal to zero is the sum of the residuals. The variance σ^2 is defined as in (3.5)

$$S=Ee12/n-p-1 \quad \textbf{(3.5)}$$

It is called Mean-Squared Error (MSE). Standard error *S* is the square root of the MSE. It uses a finite set of data to find the relationship between dependent and independent variables. hence a continuous function is generated from these relations. When the MLR model is generated the predictions of previous unknown sets can be solved.

Algorithm 1 MLR for GPA Prediction

Input: Student mark
Output: Predicted GPA of a student
1: Import libraries and student dataset
2: Categorical data is encoded
3: Avoid Dummy Variable Trap
4: Split the dataset into train and test dataset
5: Fit MLR into the train set
6: Predict model is generated
7: Apply the model on the test set to predict student GPA

F. Decision Tree

A decision tree is based on a top-down and greedy approach and it has a tree-like structure where the rectangle is used to represent internal or parent nodes and ovals are used to represent child nodes. It is used to classify the given dataset based on the entropy optimal attribute for splitting the tree selected. Here student part is taken as root nodes and the predicted GPA result will be child modes respectively.

Algorithm 2 Decision Tree for GPA Prediction

Input: Student mark
Output: Predicted GPA of a student
1: Import library and Student dataset
2: Split the dataset into train and test dataset

3: Variable *x* contains the attribute's value of the dataset
4: Variable *y* contains the target variable of the dataset
5: A set random value is used for sampling
6: Derive Gini index and information gain
7: Measure the uncertainty using entropy
8: Accuracy score and confusion matrix is obtained
9: Result is validated

G. Naive Bayes Classifier

In the Bayes method, two different events are *A* and *B* where *P(A)* denotes a student passing an exam and *P(B)* denotes a student failing an exam is the probability of *A* and *B* respectively. *P(A|B)* is the probability of GPA results of students where *A* student's academic marks are given that *B* student's value has already been obtained. The Naive Bayes algorithm is an ML method that depends on the Bayes Theorem and the formula is defined in (3.6)

$$P(A|B) = P(B|A)\,P(A)/P(B) \quad \textbf{(3.6)}$$

This formula is used to calculate the probability of student GPA value for those who Pass/Fail in the next upcoming exam depending on the dependent variable that contains data on previous student marks. Here posterior probability is *P(A|B)* which is the probability of hypothesis *A* is the student GPA on the observed event *B* is a previously obtained student mark. *P(B|A)* is likelihood probability which is the probability that a given hypothesis is true. *P(A)* is the hypothesis before observing that is prior probability and *P(B)* is the probability of evidence that is marginal.

Algorithm 3 Naive Bayes for GPA Prediction

Input: Student mark
Output: Predicted GPA of a student
1: Import libraries and dataset
2: Convert the dataset into a frequency table
3: Create a likelihood table
4: Find the probability of given features
5: Split the dataset into train and test dataset
6: Fit NB into the training dataset and generate the model
7: Test the model on the test dataset to create a confusion matrix
8: Validate the test prediction results

H. Testing/Training Samples

The method is selected to perform student performance prediction based on numerical and non-numerical values the most commonly used method in the work is support vector machine, multi-class classification, stochastic algorithm, and clustering algorithm. The data can be predefined or real-time data we have taken the real-time data of student who are studying in our college the total number of student are 120. Now the data are processed to find the total number of the training set and testing samples and the results obtained are 110 training sets and 23 testing samples. The non-numerical values also play a major role in finding training and testing samples. The index row is founded using the targeted column and support column the final results are determined using a selected method such as Adabooster, Decision tree, SVM, and Clustering.

I. Feature Engineering

This feature is processed in the given data all the non-numerical values and evaluated in the graph every unique value is identified in the value are noted in the result example there is a total number of 479 in the given below data all are unique and each plays important role in the work in identifying the *Y* value in the regression formula.

Feature engineering is used to find the difference between a good model and a bad model. It is used for (i) creating a new variable by combining two or more variables, (ii) modifying a variable, and (iii) selecting required variable features from the dataset to improve the results and also to remove unwanted features that need information about a variable. It can be performed by testing the correlation of all variables with the dependent variable in the data.

The use of Feature engineering in this work is (i) used to identify variables that cannot improve the prediction and leads the model to overfit so unwanted variables are removed for a model to be more effective. It can be done by the user manually or automatically. (ii) used to modify variables to make the model better perform in prediction and to reduce a large number of variables, feature engineering joins two or more variables, this variable can be used to obtain the best result in categorizing and classification. It can be done on the student dataset on non-numerical values such as city, address, gender, language, etc.

Algorithm 4 Feature Engineering on Dataset

Input: Normal dataset
Output: Featured dataset
1: Load the dataset and brainstorm the features
2: Find how features work to predict the model
3: Redo until the model effectively predicts the features

J. Object Variable Determination

The data are processed and variables are identified into numerical and non-numerical category values where numerical values are identified and classified into integer, float, double, and non-numerical values are identified as object characters. All character value is considered object and the Dtype value are considered as objects and imported variables are considered as Dtype variables.

IV. DATASET

Data about the students is used to predict the student's next semester's GPA mainly using data based on their academic and personal details. The dataset can be in different types such as integer, float, and character. The training data set information about the students is taken as input. The datasets are stored as CSV format files and in table format where each row and column represents a student and details contain information. Along with a column defines the data about the student pass/fail rate of the exam. Stage ID is used as an important variable in the algorithm to predict the GPA. Two different data sets were used for this product. The details about students studying in national universities are taken as the first dataset containing information about 480 students from various countries majority in the Middle East and this dataset has a total of 17 feature variables as shown in Table 4.1.

Table 4.1 Variable descriptions for the first dataset

Variables	Description	Data Type
Gender	Gender of student	Nominal
Nationality	Country of student	Nominal
Date of Birth	DOB of student	Nominal
Stage ID	Present education state of student	Nominal
Grade ID	Current education grade	Nominal
Section ID	Present classroom of student	Nominal
Topic	Courses taken	Nominal
Semester	Current semester	Nominal
Relation	Parent responsible	Nominal
Hands Raised	The student raised their hands during class	Quantitative
Resource Visited	Education resources used by student	Quantitative
View	Number of times students visited the library	Quantitative
Discussion	Student joined discussion	Quantitative
Parent Survey	Parents answer the school survey	Nominal
School Satisfaction	Level of satisfaction	Nominal
Absent Days	Number of days the student has been absent	Quantitative
Class	Grade of student for the course	Quantitative

Variables have two data types they are (i) nominal data that have a specific set of values, and (ii) quantitative data that have values that can be ordered. The variable 'Class' is the dependent or target variable that the model is trying to predict. It has 3 different values such as 'L', 'M', and 'H'. The value 'L' means low means students have a score between 0 and 69. Value 'M' means a medium that represents a score between 70 and 89 and value 'H' represents a score between 90 and 100. The second dataset was obtained locally from third-year students studying at Thiagarajar College of Engineering and has information about 125 students and 15 different variables. Some important dependent variables are 'Sem5' and the sem value can be between 0 and 10. 'Sem3' and 'Sem4' are used as input data to predict the GPA for a student as shown in Table 4.2.

Table 4.2 Variable descriptions for the second data set

Variables	Description	Data Type
Mark 10	10^{th} public Mark	Quantitative
Mark 12	12^{th} public Mark	Quantitative
Sem 3	Third Semester GPA	Quantitative
Sem 4	Fourth Semester GPA	Quantitative
Sem 5	Fifth Semester GPA	Quantitative

V. EVALUATION METHODS

The Evaluation of the predicted model is used to find the performance of the generated model. The resulting values are compared with actual values. The criteria for the prediction model and possible results of prediction in binary values are shown in Table 5.1.

Table 5.1 Binary Value for Prediction

Value	Predicted True (PT)	Predicted False (PF)
Actual True (AT)	True Positive (TP)	False Negative (FN)
Actual False (AF)	False Positive (FP)	True Negative (TN)

Here TP is when the model predicts a positive result correct, TN is when the model predicts a negative result correct, FP is when the model predicts a positive result wrong and FN is when the model predicts a negative result wrong.

The binary value matrix is called a confusion matrix which shows the possible prediction result that can be obtained. These values are obtained from different evaluation criteria such as accuracy, precision, recall, and F-measure. To obtain better results precision and recall are used together because it is not enough to accurately predict the positive outcome. Hence effective model should have a prediction of both positive and results. F-measure is a single value that has both precision and recall and it is the final evaluation criteria for comparisons.

A. ACCURACY

Accuracy is a metric for the evaluation of the classification model and it is used to find the number of correct predictions for the prediction model. Accuracy is calculated by the number of correct predictions divided by the total number of predictions defined in (5.1).

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ prediction} \quad \textbf{(5.1)}$$

In binary classification, it is calculated using TP, TN, FP, and FN respectively defined as in (5.2).

$$Accuracy = TP+TN/TP+TN+FP+FN \quad \textbf{(5.2)}$$

B. PRECISION

Precision is calculated as the ratio between several predicted results correctly which is either positive or negative divided by the total number of positive results which is either correctly or incorrectly (5.3). If the predicted model predicts a less correct result, then increasing the detonator value will lead to a smaller precision value. If the precision value is high, then the model makes a more correct prediction.

$$Precision = TP/TP+FP \quad \textbf{(5.3)}$$

C. RECALL

The recall is calculated as the ratio between the number of predicted results positively divided by the total number of positive results (5.4). It can able to find a positive result. If the recall value is high, then a more positive result is obtained. Recall depends on the positive value and is independent of the negative value.

$$Recall = TP/TP+FN \quad \textbf{(5.4)}$$

D. F-MEASURE

F-measure is used to find performance and comparisons between the models. It used both means of precision and recall as a single value respectively described in (5.5).

$$F = 2* Precision * Recall/Precision + Recall \quad \textbf{(5.5)}$$

VI. RESULT AND DISCUSSION

The compares different ML methods such as Naive Bayes classification, Multiple Linear Regression, and Decision Tree. Feature engineering in the student performance prediction. The proposed system code is written in the Python language and uses a built-in library function applied to ML methods used for this work. ML is used to generate the required output for evaluation and storing the results of prediction. The application is executed on the Anaconda workspace. Train and test sets are separated from the dataset. The training dataset is used to generate the model and this model is applied to the test dataset.

6.1 First Dataset Results

The methods are applied to the first dataset and data processing is done to modify the dependent variables to change them into binary format. Of 480 students, 353 students have good scores in exams with an accuracy result of 74 percent. It is used as a baseline to predict a value for the first dataset to generate prediction models and different ML methods were compared to find the best predictions.

Based on baseline accuracy the dataset is divided into train and test datasets. The train set containing 80 percent of student data is used to generate the prediction model and the test set consisting of 20 percent of student data is used to test the efficiency of the model. Both sets must be in the same ratio during training and test sets.

6.2. Second Dataset Results

The methods are applied to the second dataset and data processing is done to modify the dependent variables to change them into binary format. This data stores information on 125 students from 3rd year and the baseline value is used to generate prediction models and different ML methods were compared to find the best predictions. Based on baseline accuracy the dataset is divided into train and test datasets. The train set containing 80 percent of student data is used to generate a prediction model and the test set consisting of 20 percent of student data is used to test the efficiency of the model. Both sets must be in the same ratio during training and test sets.

6.3. Machine Learning Models Results

The models are coded in Python language. MLR has input data where the user knows the target, dependent, and independent variables. The decision tree uses the CART function, it has regression and classification and has similar functional properties but uses different methods but the procedure followed in this method is common to the MLR model. The Naive Bayes model is built using input data based on both dependent and independent variables. The generated model is tested on test data and a confusion matrix is obtained. The confusion matrix is the final output which has data about actual and predicted values.

Tables 6.1, 6.2, and 6.3 show models applied to the first raw dataset. Table 6.1 describes a linear regression model with a confusion matrix that shows an accuracy of 78 percent. It has a high prediction accuracy of 63 percent when using the baseline value. Table 6.2 describes the decision tree model with an accuracy of 73 percent which is the lesser than the linear regression model. Table 6.3 describes the Naive Bayes model with a confusion matrix showing an accuracy of 74 percent.

Tables 6.4, 6.5, and 6.6 show models applied to the second raw dataset. Table 6.4 describes a linear regression model with a confusion matrix that shows an accuracy of 79 percent it has a high prediction accuracy of 64 percent when using baseline value. Table 6.5 describes the decision tree model with an accuracy of 74 percent which is less than the linear regression model. Table 6.6 describes the Naive Bayes model with a confusion matrix giving an accuracy of 76 percent.

Table 6.1 MLR confusion matrix on the first dataset

Value	PF	PT
AF	26	6
AT	2	86

Table 6.2 CART confusion matrix on the first dataset

Value	PF	PT
AF	28	4
AT	1	87

Table 6.3 NB confusion matrix on the first dataset

Value	PF	PT
AF	28	4
AT	1	86

Table 6.4 MLR confusion matrix on the second dataset

Value	PF	PT
AF	14	18
AT	7	59

Table 6.5 CART confusion matrix on the second dataset

Value	PF	PT
AF	8	24
AT	7	29

Table 6.6 NB confusion matrix on the second dataset

Value	PF	PT
AF	14	18
AT	8	58

6.4 Results from Engineered Data

The dataset is modified to improve the prediction performance. The feature selection method is used for modification in this work. The variable ranking is used to identify the necessary variables in the dataset and it is done by using the feature engineering method which gives the output between the correlation value dependent and independent variables. The trial and error approach is used in the process of selecting variables. The ML model is generated using different relevant variables in multiple sets of data and the best combination of variables is identified. The custom feature creation method for modification or creation of existing variables is generated by the combination of important variables to make the model more efficient.

6.5. Engineered Data is applied to the first and second datasets

The model is coded in Python language and has many built library functions used to determine the independent variable's correlation with the dependent variable. The variables are used to generate the prediction models. The multiple linear regression models were generated using train data and the output confusion matrix has information on actual and predicted values. The decision tree model was generated using the CART method. Feature engineering is required to improve and it is important to identify what method of modification is needed. The Naive Bayes model is the same as the linear regression model and feature engineering is used for variable selection.

Table 6.7 shows by using correlations the relevant variables are identified from the first dataset containing 16 dependent variables it describes the description and data type of variables. Table 6.8 shows by using correlations the relevant variables are identified from the second dataset containing 30 dependent variables it describes the description and data type of variables.

Table 6.9 describes the confusion matrix with an accuracy of 81 percent generated by using a feature engineering dataset on the MLR model has an improvement over the normal dataset having an accuracy of 78 percent. Table 6.10 describes the confusion matrix with an accuracy of 73 percent generated by using a feature engineering dataset on a decision tree model which has similar accuracy to the model generated from the normal dataset. Table 6.11 describes the confusion matrix generated with an accuracy of 77 percent by using a feature engineering dataset on a naïve Bayes model with a better accuracy of 73 percent obtained from using a normal dataset.

Table 6.12 describes the confusion matrix with an accuracy of 77 percent generated by using a feature engineering dataset on the MLR model has improved over the normal dataset having an accuracy of 74 percent. Table 6.13 describes the confusion matrix with an accuracy of 77.3 percent generated by using a feature engineering dataset on a decision tree model which has similar accuracy to the model generated from a normal dataset. Table 6.14 describes the confusion matrix generated with an accuracy of 73 percent by using a feature engineering dataset on a Naive Bayes model which a better accuracy of 67 percent has obtained from using a normal dataset.

Table 6.7 Relevant variables in the first data set

Variables	Description	Data Type
Hands Raised	The student raised their hands during class	Quantitative
Resource Visited	Education resources used by student	Quantitative
Discussion	Student joined discussion	Quantitative
Parent Survey	School survey answered by parents	Nominal
Absent Days	Number of absent days	Quantitative

Table 6.8 Relevant variables in the second data set

Column	Description	Type
Failures	Number of times student failed in the past	Quantitative
Age	Age of student	Quantitative
Absences	Number of times the student was absent	Quantitative
Study Time	Study time in the week	Quantitative
School Sup	Education support from the school	Quantitative
Famsup	Education support from family	Nominal

Table 6.9 MLR confusion matrix on the first engineered dataset

Value	PF	PT
AF	27	5
AT	1	87

Table 6.10 CART confusion matrix on the first engineered dataset

Value	PF	PT
AF	24	8
AT	0	88

Table 6.11 NB confusion matrix on the first dataset

Value	PF	PT
AF	29	3
AT	0	88

Table 6.12 Confusion matrix for linear regression used on the second dataset

Value	PF	PT
AF	12	20
AT	2	64

Table 6.13 Confusion matrix for decision tree used on the second dataset

Value	PF	PT
AF	14	18
AT	8	58

Table 6.14 Confusion matrix for the Naive Bayes used on the second dataset

Value	PF	PT
AF	13	19
AT	3	63

6.6. Cart Model Applied On First and Second Dataset

The CART model is used to determine the optimal number of nodes automatically to make the decision tree more effective. The number of nodes is less when compared to the number of nodes is the problem here. Creating a Custom Variable-1 (CV1) by combining multiple important variables is one possible solution. The custom variable formula is applied to the first dataset and defined as (6.1)

$$CV1 = A * AbsentDays - B * ResourcesVisited - C * handsRaised \quad \textbf{(6.1)}$$

The coefficients and importance of each variable are determined as *A*, *B*, and *C*. The following rule is pass rate is proportional to the 'Resource Visited', and 'Hands Raised' variables and inversely proportional to the 'Absent Days' variable. The *B* and *C* are subtracted. It is done by trial and error process and the predicted model is generated having to differ coefficient values. It is modified to improve the prediction results. When a custom variable is generated by using a new variable CART model is generated again.

A Custom Variable-2 (CV2) is used to improve the generated model efficiency applied to the second dataset and the formula for CV2 is defined as (6.2)

$$CV = A * failures + B * absent - C * StudytTime \quad \textbf{(6.2)}$$

The coefficients and importance of each variable are determined as *A*, *B*, and *C*. The following rule fails rate is proportional to the 'Absent Days', and 'Failures' variable and inversely proportional to the 'Study Time' variable. The B and C are subtracted. It is done by trial and error process and the predicted model is generated having to differ coefficient values. It is modified to improve the prediction results. When a custom variable is generated by using a new variable CART model is generated again.

Table 6.15 gives the confusion matrix for the CART model applied to the first dataset. Table 6.16 gives the confusion matrix for the CART model applied to the second dataset. Table 6.17 shows the result obtained from the Engineering feature on the first and second datasets giving a confusion matrix with an accuracy of 77 and 75 percent respectively and the model built with the raw data had 70 and 73 percent accuracy.

Table 6.15 CART Confusion matrix on the first dataset using a custom variable

Value	PF	PT
AF	29	3
AT	1	87

Table 6.16 CART Confusion matrix on the second dataset using a custom variable

Value	PF	PT
AF	13	19
AT	3	63

Table 6.17 Results obtained from Engineering feature

Accuracy	Confusion Matrix	Raw Data
First Dataset	77	70
Second Dataset	75	73

VI. CONCLUSION

Prediction of student performance relies on the amount of data and suitable algorithms. Selecting the best algorithm for a particular problem is important to obtain better results. But algorithm alone is not sufficient to provide good prediction results. This work aims to compare different ML algorithms and feature engineering methods to improve the results of prediction. The dataset used in this work consists of two or three different ML algorithms and the obtained results were compared using four evaluation metrics such as accuracy, precision, recall, and F-measure. The main use of feature engineering is feature selection and for classification and regression methods a custom feature generation is done and feature engineering is performed automatically or manually of data. Features can be found by the trial and error method. The obtained results of both datasets show similarities and differences. The generated models used in the second dataset show better results when compared to the generated models in the first dataset. The first dataset has accuracy values of 73 to 78 percent respectively and the accuracy values of the second dataset are 78 and 88 percent respectively. Although the first dataset has more features than the second dataset prediction result shows the importance of data for the performance of prediction. The dataset uses the same methods but the obtained results are different. This shows best method also depends on the limitations of the data. This work shows that feature engineering provides a high result of prediction when compared to method selection but the combination of both methods provides the best results. In datasets, accuracy values of high student GPA prediction have an improvement over the accuracy of baseline values. Hence ML is best for predicting student GPA performance.

REFERENCES

Alsariera, Y. A., Baashar, Y., Alkawsi, G., Mustafa, A., Alkahtani, A. A., & Ali, N. (2022). Assessment and Evaluation of Different Machine Learning Algorithms for Predicting Student Performance. Computational Intelligence and Neuroscience, 2022, 1–11. https://doi.org/10.1155/2022/4151487

Amrieh, E. A., Hamtini, T., & Aljarah, I. (2016). Mining Educational Data to Predict Student's academic Performance using Ensemble Methods. International Journal of Database Theory and Application, 9(8), 119–136. https://doi.org/10.14257/ijdta.2016.9.8.13

Arashpour, M., Golafshani, E. M., Parthiban, R., Lamborn, J., Kashani, A., Li, H., & Farzanehfar, P. (2022). Predicting individual learning performance using machine-learning hybridized with teaching learning-based optimization. Computer Applications in Engineering Education, 31(1), 83–99. Portico. https://doi.org/10.1002/cae.22572

Berens, J., Schneider, K., Görtz, S., Oster, S., & Burghoff, J. (2018). Early Detection of Students at Risk – Predicting Student Dropouts Using Administrative Student Data and Machine Learning Methods. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.3275433

Kemper, L., Vorhoff, G., & Wigger, B. U. (2020). Predicting student dropout: A machine learning approach. European Journal of Higher Education, 10(1), 28–47. https://doi.org/10.1080/21568235.2020.1718520

Oqaidi, K., Aouhassi, S., & Mansouri, K. (2022). Towards a Students' Dropout Prediction Model in Higher Education Institutions Using Machine Learning Algorithms. International Journal of Emerging Technologies in Learning (iJET), 17(18), pp. 103–117. https://doi.org/10.3991/ijet.v17i18.25567

Pallathadka, H., Wenda, A., Ramirez-Asís, E., Asís-López, M., Flores-Albornoz, J., & Phasinam, K. (2023). Classification and prediction of student performance data using various machine learning algorithms. Materials Today: Proceedings, 80, 3782–3785. https://doi.org/10.1016/j.matpr.2021.07.382

Sudais, M., Safwan, M., Khalid, M. A., & Ahmed, S. (2022). Students' Academic Performance Prediction Model Using Machine Learning. https://doi.org/10.21203/rs.3.rs-1296035/v1

Xiao W, Ji P, Hu J.(2022). A survey on educational data mining methods used for predicting students' performance. Engineering Reports, 4(5). https://doi.org/10.1002/eng2.12482

Yadav, S.K., Bharadwaj, B., and Pal, S. (2012). Mining education data to predict student's retention: A comparative study. International Journal of Computer Science and Information Security 10(2), 113-117. https://doi.org/10.48550/arXiv.1203.2987

Yagci, M. (2022). Educational data mining: prediction of students' academic performance using machine learning algorithms. Smart Learn. Environ. 9, 11. https://doi.org/10.1186/s40561-022-00192-z