An-Najah National University

Faculty of Graduate Studies

MACHINE LEARNING ALGORITHMS FOR PREDICTING STUDENTS' ACADEMIC PERFORMANCE IN EDUCATIONAL DATA MINING

By

Fatimah Najwan Fawwaz Salmiyah

Supervisors

Dr. Ahmad Awad

Dr.Emad Natsheh

This Thesis is Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Artificial Intelligence, Faculty of Graduate Studies, An-Najah National University, Nablus, Palestine.

2025

MACHINE LEARNING ALGORITHMS FOR PREDICTING STUDENTS' ACADEMIC PERFORMANCE IN EDUCATIONAL DATA MINING

By
Fatimah Najwan Fawwaz Salmiyah

This Thesis was Defended Successfully on 21/06/2025 and approved by

Dr. Ahmad Awad	
Supervisor	Signature
Dr. Emad Natsheh	
Co-Supervisor	Signature
Dr. Mohammad Hamarsheh	
External Examiner	Signature
Dr. Hamed Abd Al-Haq	
Internal Examiner	Signature

Dedication

﴿يَرْفَعِ ٱللَّهُ ٱلَّذِينَ ءَامَنُواْ مِنكُمْ وَٱلَّذِينَ أُوتُواْ ٱلْعِلْمَ دَرَجَٰتٍۚ وَٱللَّهُ بِمَا تَعْمَلُونَ خَبِيرٌ﴾ [المجادلة:11]

بسم الله على الغايات فبفضل الله وصلنا ... بسم الله على الأحلام، قد جعلها ربي حقا

يا أجود من أعطى و يا خير من سئل لك العتبى حتى ترضى يا الله

بفضل الله ومنته وكريم انعامه أناقش اليوم رسالة الماجستير، فها قد آتى يوم الحصاد وسأقطفه بيدي وأهديه لمن شاركوني الخطى، ومن كانوا دوماً يذَكِروني بأن من جد وجد

سأهديه لقرة العين وتاج الرأس والديّ الحبيبين وزوجي العزيز وجميع الأهل والاقارب/

سأهديه لكل من علمنا حرفا...

سأهديه لصديقاتي وزميلاتي

ساهديه لاهلنا في غزة العزة...

ساهديه لوطننا لشهدائنا لأسرانا لجرحانا

كما أتقدم بالشكر الجزيل مشرفيَّ الفاضلين الدكتور أحمد عواد والدكتور عماد النتشة على جهودهم المبذولة لانجاح هذه الرسالة، والشكر موصول للجنة المقيمة

كما نشكر كل من ساهم في انجاح هذا العمل والشكر للحضور الكريم فبكم تكمل فرحتي

شكراً لكل من دعوا لنا دعوة صادقة كانت مصدر قوتنا

انتهت رحلة الماجستير فاللهم وفقني في خطواتي القادمة لما تحب و ترضى واجعلها فاتحة خير وعلم نافع ينتفع به.

Declaration

I, the undersigned, declare that I submitted the thesis entitled:

MACHINE LEARNING ALGORITHMS FOR PREDICTING STUDENTS' ACADEMIC PERFORMANCE IN EDUCATIONAL DATA MINING

I declare that the work provided in this thesis, unless otherwise referenced, is the researcher's own work, and has not been submitted elsewhere for any other degree or qualification.

Student's Name: فاطمة بنوان فواز مالية

Signature: فاطمة

Date: 21/6/2025

List of Contents

List of Tables

List of Figures

List of Appendices

MACHINE LEARNING ALGORITHMS FOR PREDICTING STUDENTS' ACADEMIC PERFORMANCE IN EDUCATIONAL DATA MINING

By
Fatimah Najwan Fawwaz Salmiyah
Supervisors
Dr. Ahmad Awad
Dr.Emad Natsheh

Abstract

Student performance prediction has been one of the important works in educational data mining, due to the possibility of early detection, intervention, and informed decision-making in academics. The purpose of this study is to improve the accuracy of predicting student performance by using seven machine learning models—Decision Tree, Random Forest, Linear Regression, Neural Network, Support Vector Machines (SVM), Logistic Regression, Naive Bayes and five feature selection techniques : Particle Swarm Optimization (PSO), Lasso, Wrapper Method, SelectKBest and SelectPercentile. The research investigates how student outcomes are associated with such factors as previous educational experience, parental education, past educational failures, attendance, residence and participation in extra-curriculum activities. The results show that Linear Regression combined with SelectKBest achieved the highest accuracy of 93.5% . The performance of the model was further optimized by hyperparameter tuning (GridSearchCV) and k-fold cross-validation, which increased both prediction accuracy and model robustness. The results highlight the role of feature selection in maximizing model performance and offer practical guidance for higher education institutions interested in implementing predictive analytics to enhance student success.

Keywords: Educational Data Mining, Machine Learning Models, Student Performance Prediction, Feature Selection.

Chapter One

Introduction

No two students learn the same way, but what if there was a way to predict which students might be at risk and how best to learn? Progress in educational data mining is now making it possible to do so for improvements in learner support systems.

Education providers have recently been working to improve student outcomes through the use of data (1). One of the most significant areas among various applications is predicting academic performance. Historically, teachers have relied on intuition and historical records to flag students in jeopardy, yet those methods have often not been early or precise enough. Thanks to the recent advances in the area of educational data mining, a subset of data mining (2), it is now possible to reveal significant structures and derive useful insights from educational datasets (3) This understanding facilitates better decision-making and personalizes intervention. Accordingly, student performance prediction has become a major research issue in the field of educational data mining (4) with important implications in identifying learners at risk, improving instructional strategies, and providing personalized interventions (5).

The rise of online learning environments worldwide has significantly increased the volume and importance of digital student data. Such data is a rich source that can be mined to provide information on the patterns (e.g., enrollment trends or academic performance history), with help of which predictions of student outcomes can be made. Additionally, factors such as educational, family, socio-economic circumstances, the learning environment, and prior academic history are central in determining the probability of a student being either successful or at risk on a particular academic task.

Predicting students' performance in the academic domain is a complex, multifactorial problem, depending upon numerous issues such as domain-specific skills, study habits, and social status (6). Standard forecasting techniques are often manual analyses of limited datasets, which frequently fail to capture the various factors that influence student performance, thus leading to inaccurate or unreliable forecasts. Nevertheless, progress in machine learning methods, as well as an increasing variety of large educational data sets, provides promising solutions to these longstanding challenges (1).

The application of data mining analytics encounters many of the challenges and limitations in education, as elsewhere. A major challenge is data accuracy and validity; any errors in data collection can lead to inaccurate conclusions. Data security and privacy are also critical issues because the institutions are handling student information; they are the curators of student records, and they want to make sure that those student records are protected from unauthorized access (7).

In educational data mining, the machine learning method provides an effective way for predicting students' academic performance based on large data and ethical aspects. Machine learning algorithms offer several advantages in this thesis : (i) the capability of modelling complex relationships between multiple parameters; (ii) scalability to large datasets; (iii) adaptivity to learn from new data and find new patterns; and (iv) ease of discovering hidden relationships and automating the prediction process (8). With the help of these algorithms, this study aims to show that more accurate predictions on student academic performance can be made and more efficient interventions can be delivered, adaptive instruction can be provided, and evidence-based educational decision-making is encouraged.

This study aims to predict student performance at the end of the semester using data from secondary education students at two Portuguese schools. The objective is to anticipate final grades in order to enable educators to intervene and support students potentially at risk of academic difficulties. The research involves extensive data preprocessing to improve the accuracy of the predictive models. Various feature selection techniques, including Particle Swarm Optimization (PSO), Lasso, Wrapper Method, SelectKBest, and SelectPercentile, are employed to identify the most relevant features. Subsequently, seven classification algorithms—decision tree, random forest, linear regression, neural network, SVM, logistic regression, and Naive Bayes—are applied and evaluated using metrics such as accuracy, precision, recall, F1 score, mean squared error, mean absolute error, R^2 score, confusion matrix, and computational time. The best-performing model is then refined to achieve enhanced predictive outcomes. In addition, the study investigates the impact of binary outcome categorization (pass/fail) on the effectiveness of educational data mining.

This thesis focuses on developing an automatic student performance prediction system with machine learning algorithms. This study uses educational data to determine the

major factors that could affect the academic success or failure of students and presents educators with actionable insights to support the learning process. Better prediction of performance can support early interventions that lead to enhanced retention and success rates, and contribute to improved educational practice.

The main novelty of our findings is that careful feature selection from student datasets is critical for predicting students' academic performances using machine learning techniques. Prediction accuracy is related to the depth of data analysis for various types of data and a good knowledge of what factors are affecting student outcomes. By selecting relevant features, only the important information is preserved, while minimizing noise from irrelevant or redundant attributes. Feature selection and engineering methods serve the purpose of finding the most predictive variables. In addition, correcting for data quality problems and reducing potential biases are required to develop accurate prediction models (9). Therefore, this study is anticipated to enhance the reliability of student performance predictions based on these considerations.

1.1 Dataset Description

This study uses data from the UCI Machine Learning Repository, specifically two datasets collected from Portuguese secondary schools. The combined dataset consists of 1001 student records and 33 features, including demographic, academic, and behavioral attributes such as parental education, prior grades, internet access, and support programs. Each student record includes a final grade (G3), which was binarized into "pass" or "fail" to support classification modeling. The diversity of the features—spanning numerical and categorical types—allowed for applying various preprocessing, feature selection, and machine learning techniques. This dataset provides a practical foundation for building accurate prediction models in the context of real educational challenges.

1.2 Related work

Knowledge from relevant studies in a particular research area can be identified and appraised through literature reviews. They chart the progress of the field, synthesizing what is known, historically and currently, about this nascent body of work.

How to predict student success and detect at risk students is a commonly studied problem area. Data mining techniques have been used in several studies to predict students' academic performance accurately. A study (10) for example introduces the construction of a Decision Support System (DSS) based on a multi-class SVM approach to predict student performance in higher education. By using 7-fold cross-validation, the system gave an accuracy of 72.25%.

In another study (11), the application of the Naïve Bayes algorithm was described to mine the academic data for the prediction and analysis of student and teacher performance. In order to better support them with regard to examination interventions later on, it aimed to identify pupils who would fail.

Moreover, in (12) , they study the academic performance of students by a decision tree algorithm based on student activity and academic data. In this work, a classification experiment was performed using the decision tree algorithm on student performance and Moodle access time with WEKA With an accuracy rate of 63.64%, which helps develop student results in the module and provides stakeholders to assess and analyze how the module was well conducted.

A deep neural network to predict student performance was discussed in (13). Based on student categorization, the neural network is able to give a clear idea to educational institutions as to what performance category that student belongs to. Identifying the core of failing students is important so we can directly address their issues. The deep neural network that is proposed tries to predict whether students will pass or fail and the accuracy was 84.3%.

Another study (14) conducted the classification analysis, which furthered with three models: Decision Tree (C4.5), Multilayer Perceptron (MLP), and Naïve Bayes for student performance prediction depicts the comparison of three machine learning models, which shows that the Naïve Bayes classifier at 86% prediction accuracy was most accurate of all. With this study in hand, teachers can identify those who are most likely to be at risk for failing and can intervene directly with that student to help them improve their academic performance.

The article (15) examines the application of data mining approaches - decision trees, naive Bayes and random forests - to predict student final grades. It underlines the

significance of carrying out a feature selection process, as the predictive accuracy for the models increased particularly with binary grading systems. The two datasets employed were collected from Portuguese secondary schools with attributes that included grades, social, demographic, academic and other school-related features. Results indicated an increase in performance with the models with binary pass/fail grading and feature selection considerably improved the predictive accuracy. Overall, the research displayed the merits of data preprocessing and feature selection to predict educational performance.

The research (16) investigates the use of machine learning algorithms to forecast academic achievement by leveraging engagement metrics, internet activity, demographic data, and learning management system (LMS) activity. Within the context of machine learning, the study reinforces the necessity of interpretability by evaluating the performance of rule-based and tree-based classifiers, achieving classification accuracies in the range of 71-76%. An important conclusion of the study is that prediction models designed for specific subgroups of students, based on demographics and study modes, enhance accuracy and facilitate proactive predictive analytics in higher education.

Machine learning algorithms have previously been applied to predict students academic performance in the area of educational data mining. It is significant to fine-tune the algorithm parameters and enhance metrics, utilize ensemble methods, and adjust feature selection methods for better accuracy with increased degrees of data. Our research fills this gap by improving the accuracy by the algorithms tuning and selecting the best features. A comparative analysis of earlier studies (15),(16), (17) ,(18),(19), (20), (21),(22) ,(23) and (24) is presented in Table (A.2) in Appendix A.

In study(18) , educational data mining methods are used to evaluate undergraduate student performance from a four-year information technology program at a Pakistani university. The primary objectives are to make predictions about students' final academic achievement early in the academic program using admission marks and first semester course marks, to determine which course are key indicators of overall performance, and to analyze the typical trend patterns in student progression. This research utilized classification models, such as a decision tree and several clustering methods, with data from 210 students and found that academic data could reasonably

predict ending final academic outcomes, finding 85% accuracy from the decision tree model. There was considerable evidence for certain courses being strong indicators of student performance, and the clustering did identify a few trajectories related to student performance that would be beneficial for educating the at-risk student population and documented against support strategies.

The paper (20) assess the use of various machine learning methods, for example, decision trees (ID3, C4.5), Naive Bayes, and SVM in predicting student outcomes using educational data mining. They assess that C4.5 has an advantage over ID3 on larger datasets, that Naive Bayes has good precision and recall, and that SVM had the highest accuracy 85% in classifying students who performed well. The studies used the UCI Machinery Student Performance dataset with 33 attributes, and, 649 instances. They observed that there was significant potential for these models to identify students at risk, and use early alert systems, to intervene and improve educational outcomes.

The Multilayer Perceptron (MLP) outperformed traditional classifiers in the study (23) due to its high prediction accuracy of 98.3%, robust learning algorithm, effective feature utilization, low error rates, comparable performance, and flexibility. The MLP was trained using a back-propagation learning algorithm, which minimizes error through gradient descent. And the ability to adjust hidden layers and units make it an effective predictive tool for student academic performance. Additionally, "Decision Tree" and "Random Forest" outperform other classifications in study (24), "Random Forest" reached 99.5% on the first database, while on the second database, "Decision Tree" had an accuracy of 97.03% These algorithms typically perform well on smaller datasets, and have fewer hyper parameters to tune compared to other classifications, like MLP.

Few recent studies (in the period of interest — 2022–2024), and again using deep learning methods, have further advanced the state-of-the-art in performance prediction in education:

- (25): A survey on deep learning approaches in EDM which finds the performance of LSTM, CNN and transformer models on different prediction tasks.
- (26): Based on behavioral data collected using clickstream, an LSTM model was developed to predict student performance with an accuracy of 90.25%, with the most significant predictors found to be web pages related to quiz .

- (27) : The ProbSAP framework that incorporated Bi-LSTM and smart XGBoost was proposed by (2025) which yielded 88.23% accuracy and decreased MAE significantly.
- (28) : Finally, LSTM was applied on time-series educational big data, showing that LSTM was able to capture a long-term learning trend.
- (29): reviewed use of DL methods for knowledge tracing, known for its strong need for multimodal DL approaches, and note that a portion of DL applications are related to affect detection.
- (30): recapitulated progress on EDM and learning analytics, covering dataset heterogene- ity and model evaluation methods

Table 1

comparative Summary of Recent Studies on Student Performance Prediction (2023–2025)

Study	Year	Techniques / Models	Data Focus	Key Findings
(25)	2023	LSTM/CNN/Transformer DL	Multiple EDM tasks	67% of DL studies outperform ML baselines; reviewed several public datasets
(26)	2024	LSTM + Clickstream features	Click-thru logs	Achieved 90.25% accuracy; content, quiz pages most predictive
(27)	2025	LSTM + XGBoost (ProbSAP)	Imbalanced academic data	>88% accuracy; MAE significantly reduced
(28)	2024	LSTM over time-series behavioral data	Educational Big Data	Validated LSTM effectiveness in capturing long-term trends
(29)	2024	Survey (DL methods)	Broader EDM field	Emphasized trends like affect detection and multimodal data
(30)	2024	Survey	Learning analytics	Highlighted key methods, datasets, and future EDM directions

Comparison & Gap Analysis

These studies, however, focus on deep learning models, behavior tracking over time, or evaluation of a single algorithm, and ignore the computational cost, cross-model comparison and feature selection effect.

In comparison our work presents a holistic framework which:

- An ensemble of seven ML algorithms with five feature selection methods
- Uses GridSearchCV and k-fold cross-validation for tuning
- Performance metrics – accuracy, recall, precision, MAE, and F1
- Shows how feature selection affects accuracy and time to process

Specifically, Linear Regression with SelectKBest achieved the highest accuracy (93.5%) with notable computational efficiency. This confirms that combining optimal model–feature selection pairs provides a scalable and interpretable predictive solution that surpasses prior approaches in both accuracy and practicality.

However, in our research, Linear regression outperforms Random Forest and MLP, Linear regression is more effective than other models in linear relationships, small datasets, interpretability, multicollinearity, low noise, computational efficiency, and independence assumptions. It captures the true relationship between independent and dependent variables accurately, reduces overfitting, provides clear coefficients, and is less computationally intensive. Also, in our research, we focus on selecting the best features using various feature selection methods that achieve higher accuracy and better results, an approach not thoroughly explored in previous studies.

1.3 Problem Statement

Accurately predicting students' academic performance through machine learning algorithms is an essential task in the educational data mining area. The challenge is to create an efficient predictive model that can predict academic performances based on different data observed from educational systems.

Machine learning techniques can be applied to the massive amount of educational data which includes students' demographic information and behavioral, informational, and context features (31) . All this data we try to mine, get insights, and create a strong algorithm with maximum accuracy.

In our framework, we want to maximize accuracy and minimize prediction errors to improve student results and reduce the failure rate at an early stage. We also want to provide immediate assistance to students who need it most by using several machine

learning techniques, such as decision trees, random forests, linear regression, neural networks, SVM, logistic regression, and Naive Bayes:

Let's represent this mathematically

Let X = [x1, x2, x3, x4,..., xn]: feature matrix where each xi represents a feature related to student performance (e.g., grades, absences, age , father's education,...).

let y: vector of actual student outcomes (e.g., success or failure).

Let w: vector of model parameters (weights).

Let f(X, w) be the model's prediction function.

Objectives:

- Maximize Predictive Accuracy:
  $$\max \text{Accuracy}(X, y; w) = \frac{TP + TN}{TP + TN + FP + FN}$$
- Maximize Precision, Recall, and F1 Score:
  max Precision (X , y ; w) , Recall (X , y ; w) , F1 Score (X , y ; w)
- Minimize Prediction Errors (MAE and MSE):
  $$\min \text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |yı - f(Xi\,; w)|$$
  $$\min \text{MSE} = \frac{1}{n}\sum_{i=1}^{n} \left(yı - f(Xi\,; w)\right)^2$$

Key Challenges

1. Data Pre-processing: Educational data are commonly marked with missing values, outliers, and noises. Important note: Its highly essential to preprocess our data effectively since this will add value to the prediction models we are going to build.
2. Feature Selection: The method which determines essential features from data to decrease dimensionality while improving prediction accuracy. The process requires both deep knowledge of the data and understanding of the domain.
3. Model Selection and Evaluation: The selection of appropriate machine learning algorithms which effectively detect relationships between input features and target

variables represents a vital process. Model evaluation requires appropriate validation techniques together with correct metrics to determine predictive capabilities**.**

4. Generalize: The predictive model needs to perform well on data that the model has not seen before to prevent overfitting or underfitting we require effective model regularization techniques and cross-validation strategies**.**
5. Algorithm Tuning and Modification: The field of education benefits substantially from algorithm tuning and modification because it leads to improved model accuracy through parameter adjustment and algorithm changes. The implementation of proper tuning and adjustments results in more accurate predictions which produce better educational results.

1.4 Research Objectives

Predicting student academic performance is not just a technical challenge, but an educational necessity. In many schools, especially in resource-constrained environments, students who struggle academically are often identified too late after failure has already occurred. Traditional assessment methods fail to detect early warning signs due to their reliance on subjective judgment or limited indicators.

This research bridges this gap by building an intelligent, data-driven framework capable of identifying at-risk students early based on historical, behavioral, and socio-demographic data. Unlike previous studies that relied on a single model or basic features, this study combines multiple machine learning algorithms with advanced feature selection techniques. The goal is not only to improve prediction accuracy but also to provide actionable insights that teachers, school counselors, and academic institutions can use to design early intervention strategies, allocate support resources more efficiently, and ultimately improve student outcomes.

Accordingly, this research aims to create an accurate student performance prediction model through machine learning by using educational data mining approaches. The specific objectives are to:

1. Examine the educational data to determine which variables and features most strongly affect student academic results.
2. The educational data requires preprocessing to address missing values and outliers and noise that affects prediction modeling accuracy.

3. The research evaluates and compares machine learning algorithms which include decision tree, random forest, linear regression, neural network, SVM, logistic regression and Naive Bayes for student academic performance classification.
4. The research employs Particle Swarm Optimization (PSO) and Lasso and Wrapper Method and SelectKBest and SelectPercentile to select important factors which reduces dimensionality and improves prediction accuracy.
5. Evaluate model performance: The predictive capabilities need assessment through accuracy, precision, recall, F1 score, MAE, MSE, R2, and confusion matrix metrics.
6. Assess model generalization: The validation process includes cross-validation and performance evaluation on new data to guarantee model applicability for different student groups and educational environments.
7. Provide insights and recommendations: The analysis of predictive models reveals important performance influencers which leads to recommendations for better educational results.
8. Identify algorithm modifications: The research should investigate ways to enhance model accuracy together with methods for better student academic performance prediction.

Chapter 2 explains the methodologies employed, including data collection, preprocessing, and the machine learning techniques used. Chapter 3 presents the experimental studies, detailing the dataset, preprocessing steps, and the results of the predictive modeling. Finally, Chapter 4 offers discussion of the previous results, conclusions and directions for future research.

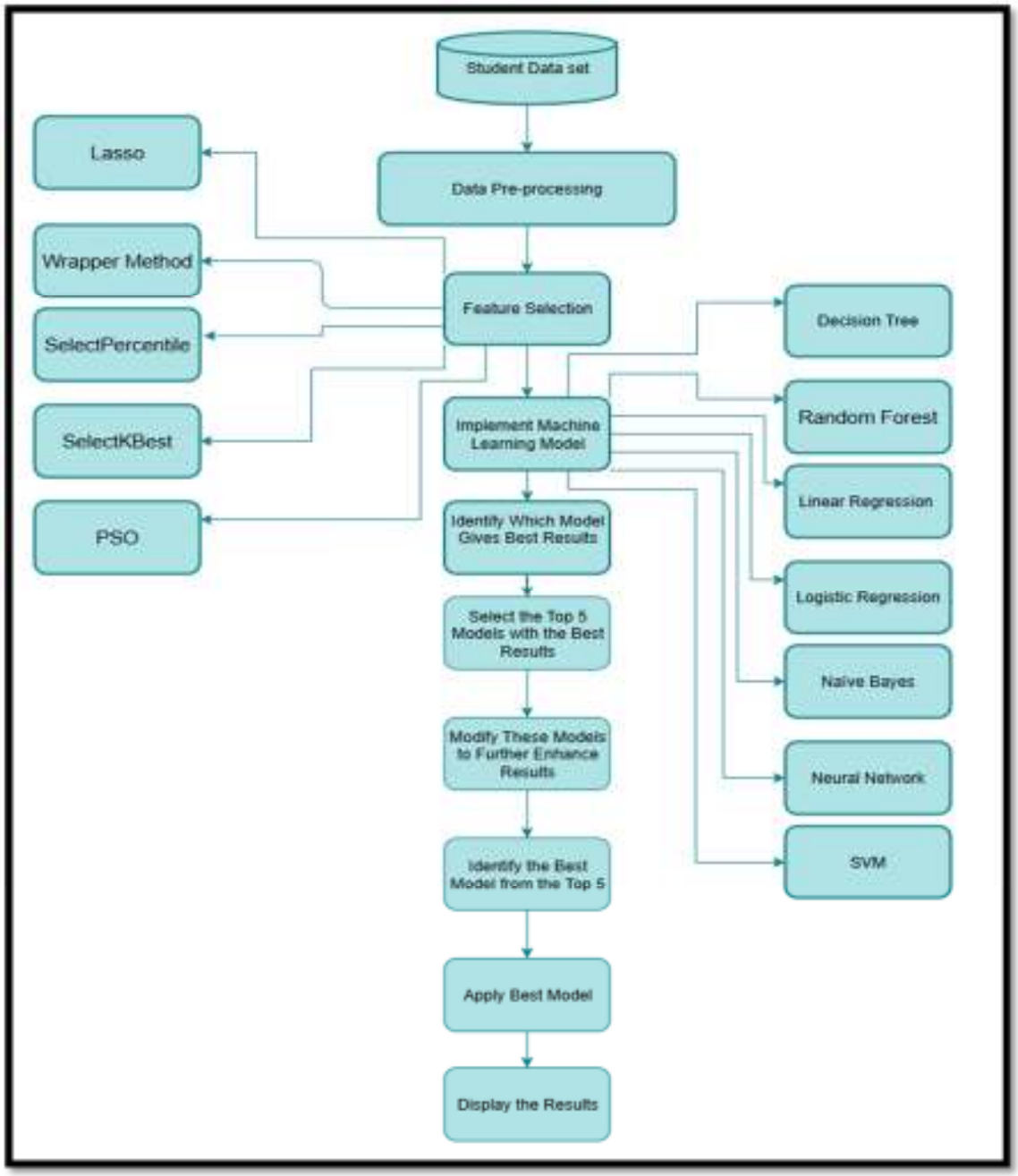
Chapter Two

Research Methodology

The research method implemented in this thesis is a systematic investigation and analysis of the machine learning algorithms to predict student's academic performance for Educational Data Mining. The figure below shows the next key steps to achieve the research objectives:

Figure 1

Flow Diagram of the Proposed Methodology



2.1 Data Collection

Our study uses two publicly available datasets, the UCI Machine Learning Repository (32) secondary education datasets of two Portuguese schools, to predict student performance for this study. The datasets consist of student grades and social, demographic and school-related features collected from questionnaires and school reports. The first dataset has information regarding student performance in Math, and the other one contains data about Portuguese language classes.

2.2 Data Description

We started with all 33 original features in the dataset, as shown in Table (A.1) in Appendix A. To support various prediction tasks, we preprocessed the dataset. These tasks included binary classification (such as pass or fail), multi-class classification (such as A, B, and C grades), and regression (to predict numerical scores). For our thesis, we chose to use binary classification. The data set sizes for the present study included the records of 1,001 students: 647 (64.6%) passed and 354 (35.4%) failed, showing mild class imbalance. This asymmetry should be taken into account in regard to the accuracy measure. Hence, alongside accuracy, other performance measures, including precision, recall, and F1-score, were computed for a more complete examination of the classification models.

2.3 Data Preprocessing

Data preprocessing is a key step in the overall Knowledge Discovery process that deals with data cleaning, feature selection, data transformation, and reduction. It refers to the transformation of unprocessed data into an appropriate format for use by a data mining algorithm.

Since the data was already clean, and we did not have any missing rows or attributes, we desired to skip the data cleaning process and headed into our first phase of data preprocessing. Here in this phase, we introduced a new column, Grade, in which the final grade is turned into a binary classification target. We considered all variables as predictors of Grade as the dependent variable. Possible problems are the first G1 and G2 grades that come with similarly high inter-correlations with G3. But G1 and G2 are not based on G3; they are bona fide academically taken prior to G3 on the school year. As they are known prior to assignment of final grade, their usage is not data leakage. They

are instead indicative of the true model and would be available if such testing was occurring in the real world: any early performance information would be used to predict final values.

Final grades were divided into two categories: Pass and Fail the levels and classifications are shown below. This binary category simplifies evaluation by identifying successful and unsuccessful students.

- Pass: ranging (10–20), marks (A, B, C, D)
- Fail: ranging (0–9), mark (F)

2.4 Exploratory Data Analysis (EDA)

Before applying machine learning models, it is important to understand the dataset, perform exploratory data analysis (EDA) to get insight into the dataset, identify patterns, detect anomalies, and inform feature selection. In this paper, we employed several visualization and analysis techniques to conduct EDA on the educational datasets.

- Distribution Visualization: to visualize the distribution of each numerical feature and assess their normality. This helps to understand the normality or spread value of distributions. The histograms tell you how the data is distributed while probability plots will check whether that distribution can be approximated with a normal one. This step is crucial in order to understand the underlying distribution of the features which can help you further in data transformation and while making modeling decisions later.

In addition, the modeling exercise needs to encode categorical data and manipulate features that help in analyzing student performance. For example, factors like family relationships and free time are frequently encoded categorically to understand how they affect grades.

Figure 2

Distribution Visualization of Students Based on Mother's Education Level

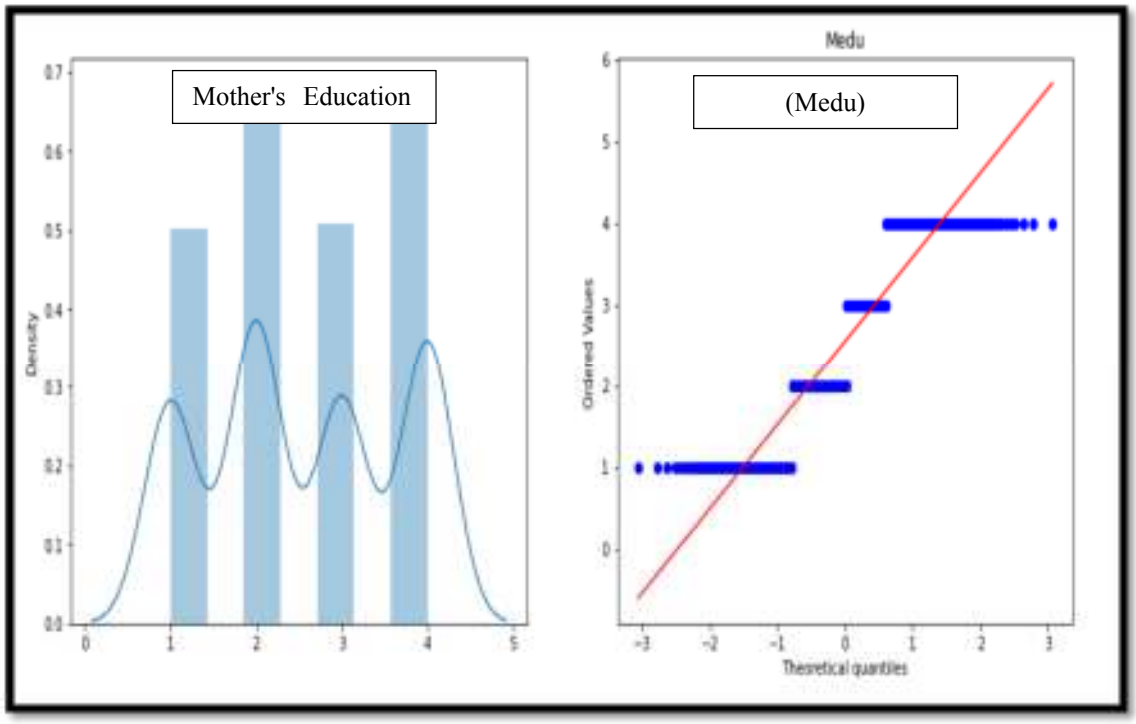
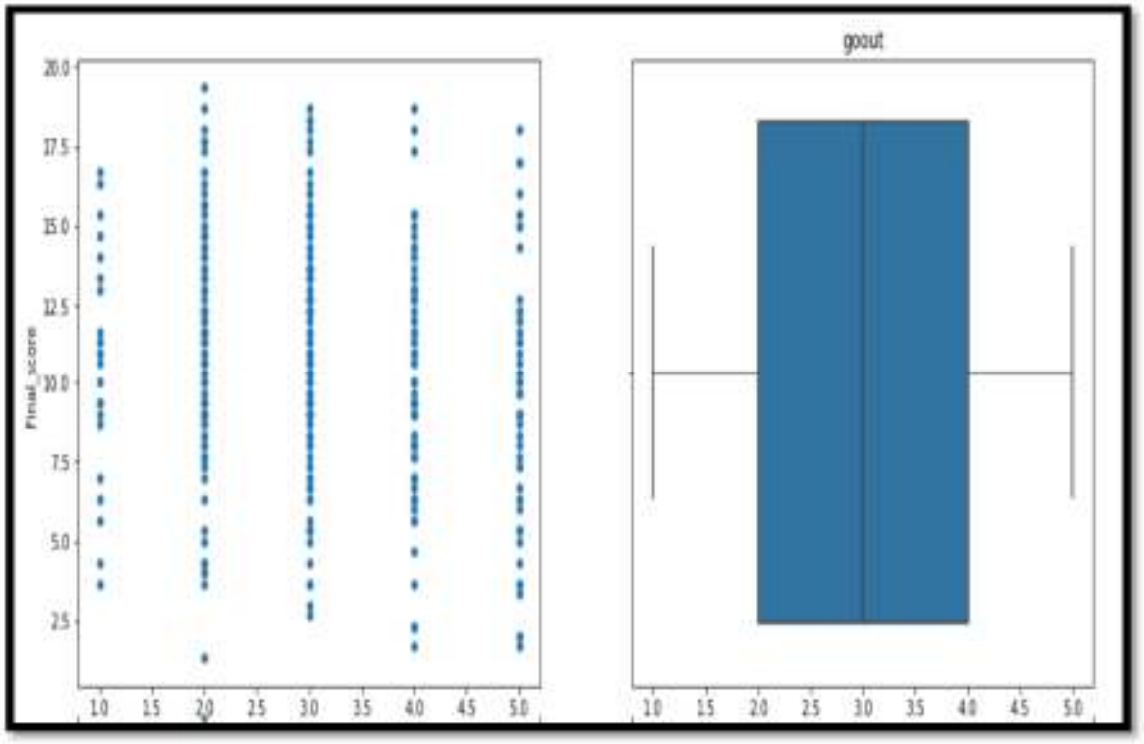


Figure 2 illustrates the distribution of the mother's education level (Medu) among the students in the dataset. The left chart shows a histogram with a KDE curve, indicating that Medu values are discrete and approximately uniformly distributed. The right chart presents a Q–Q plot for Medu, which reveals some deviation from normality, suggesting that this feature is not normally distributed.

- Outlier Detection: As part of the data preprocessing step, outlier detection was conducted using scatter plots and box plots. This process was applied to all features to identify and assess the impact of outliers on model performance. Figure 3 illustrates an example using the feature "goout", which represents how often students go out with friends. This variable was selected as a representative feature for demonstration purposes.

Figure 3

Scatter and Box Plot Depicting the Relationship Between Students' Education Level and Frequency of Going Out



The scatter plot shows that students who go out moderately (2 to 4 times per week) tend to achieve higher final scores, while those who rarely or never go out appear to perform lower. The box plot further visualizes the distribution of the "goout" variable and highlights some minor outliers.

Since only a small number of outliers were identified and they did not significantly distort the distribution, they were removed from the dataset to ensure cleaner input for machine learning models. This procedure was consistently applied across all features during the preprocessing stage.

- Multi-Collinearity Visualization: Regression Plots are good to check if features have multi-collinearity with each other or not with respect to grades. It is a plot that explains the relationship between various features with grades of each student in three periods (first period G1, second period G2, final period G3). It also helps in Identifying the most significant predictors; and filtering out unimportant predictors and redundant features.

Figure 4

Multi-Collinearity Visualization Showing the Correlation Between Study Time and Grades

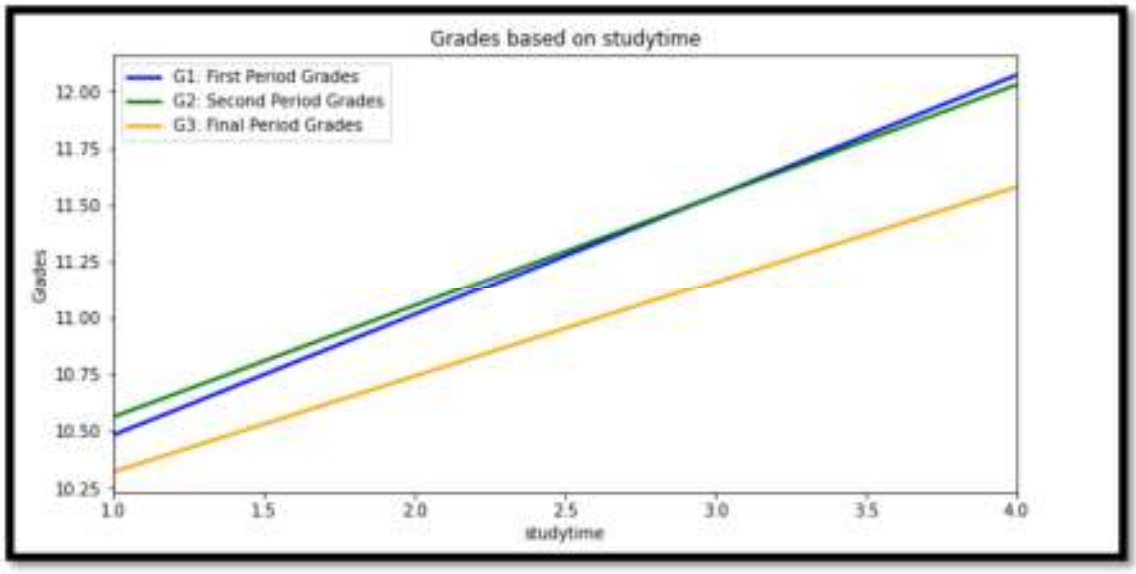
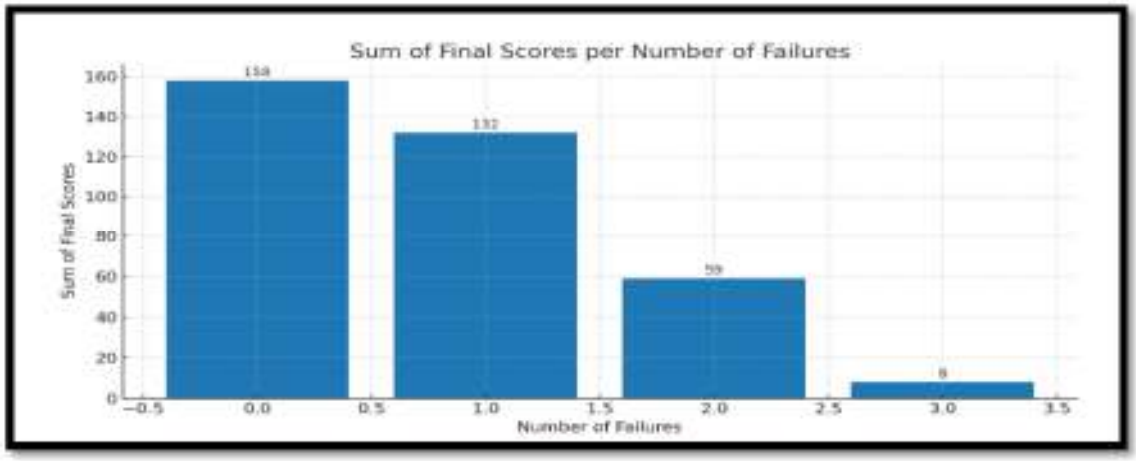


Fig. 4 illustrates that increased study time is positively correlated with better grades, indicating that students who dedicate more time to studying tend to achieve higher academic performance.

To determine the number of failures in relation to the final academic score, we binned the data according to the feature failures and evaluated the sum of Final score in which each bin was represented. The latter is represented by a bar chart below.

Figure 5

Correlation between Past Failures and Final Grades



The figure 5 shows that students with few failures have a higher cumulative final score. It implies an inverse relationship between number of failures and overall academic

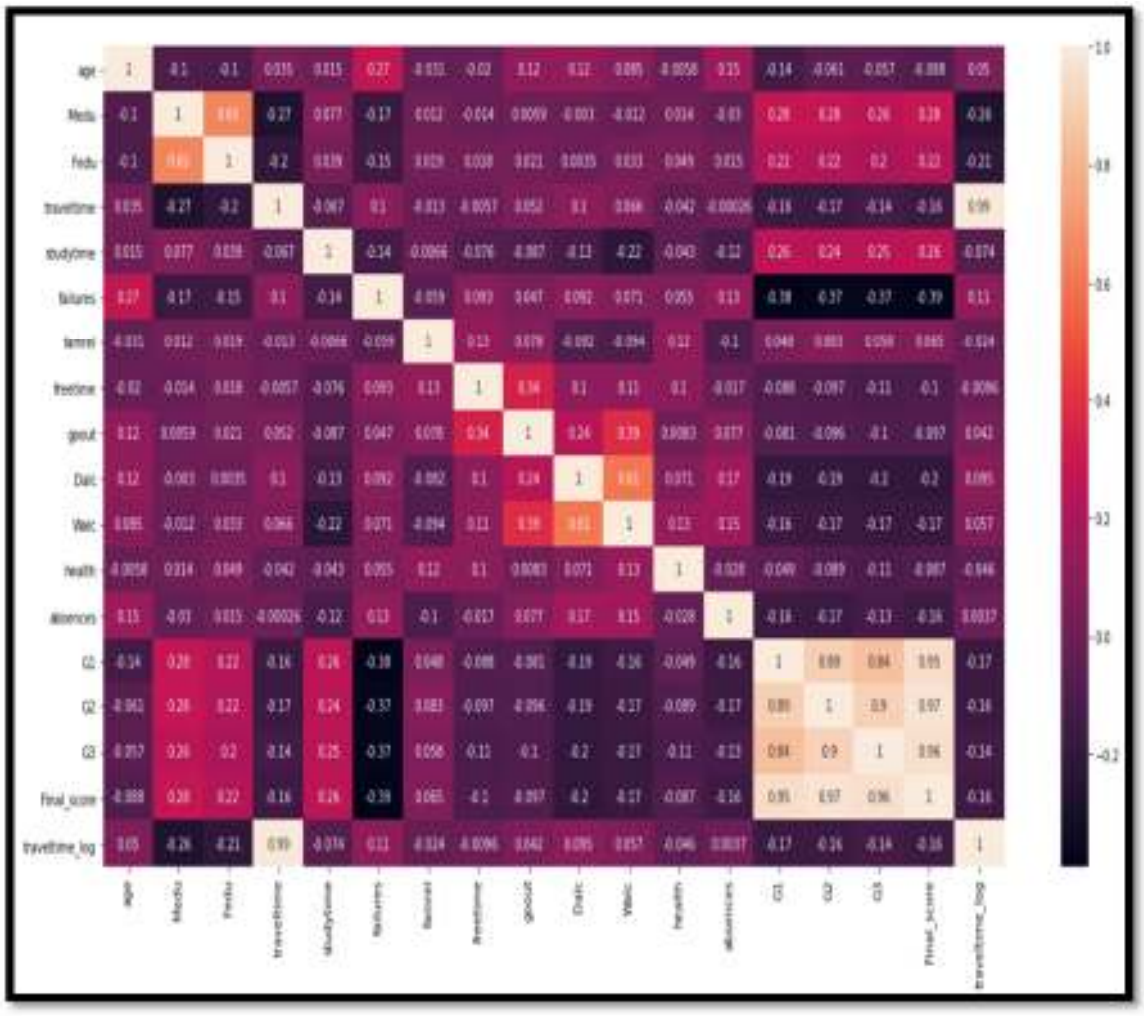
performance. The large drop in completely local scores for students with multiple failures further demonstrates the significance of early scholastic support.

Using the correlation matrix, we can make the following observations:

1. Future grades are strongly correlated with previous grades.
2. There is a substantial positive association between grades and parents' education levels, study time, relationships with family, and further education aspirations.
3. There is a considerable negative link between grades and alcohol intake, prior failures, the frequency of going out, travel time, and absence.
4. There is a strong positive correlation between mother's education and father's education, as well as between daily alcohol consumption and weekend alcohol consumption.

Figure 6

Correlation Matrix



We identified several additional correlations:

1. We noted that Urban students generally performed better than their rural counterparts. One reason for this discrepancy may relate to the lengthier commute time of rural students and its possible negative effect on academic performance.
2. Performance in a few cases has been noted to decrease with age, as a rule, the general performance decreases with higher ages.
3. We also observe a positive correlation between improved family relationships and better student performance.

2.5 Feature Selection

Feature selection is a technique used to select a subset of features from an original set of x features, where the selected subset contains fewer features ($n < x$) to optimize a specific performance metric (33). Feature Selection plays a vital part in the process of building a predictive model where we have to select among many useful variables but only a few are directly affecting our target variable thus using them will improve the accuracy as well as interpretability. To construct the most accurate and computationally efficient predictive model, we applied several state-of-the-art feature selection methods that include Particle Swarm Optimization (PSO), Lasso, Wrapper Method, SelectKBest and SelectPercentile.

Feature selection has several advantages, which includes simplifying the dataset (which helps in resolving the curse of dimensionality(34)), reducing overfitting and computational time. The feature selection is a five-phase way to achieve these objectives. The dimension of the search space is first defined by the number of original features, which is calculated during the initialization phase. Second step we call as Generation in which subset of features are going to select through different search approaches like conventional methods and meta-heuristic algorithms. Usually, this may begin with no features, all features or some random subset. In the third step i.e evaluation, this algorithm evaluates the effectiveness of selected subsets. Stopping criteria: This step applies criterions to stop the process once found a good performance. Finally, the validation step is carried out to confirm that the subset of features chosen works well through a test set.

2.5.1 Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) algorithm. PSO is a continuous real-valued algorithm, which was first proposed by Kennedy and Eberhart (35) and it is commonly known as the Standard PSO (SPSO). SPSO operates a swarm of particles moving in the D-dimensional search space seeking for the optimal solution. Every particle i further has a current velocity vector Vi=[vi1,vi2,…,viD] and a current position vector Xi=[xi1 ,xi2i,…,xiD],where D represents the number of dimensions (36). How does the SPSO work ? The control cycle of the SPSO is executed recurrence with Vi and Xi being haphazardly set to any value in a group. At every iteration, the velocity and position of particle i are updated based on two most important entities which are; the individual best position found by particle i, Pbesti=[pbesti1,pbesti2,…,pbestiD] , and the global best position found by all particles in the swarm Gbest=[gbest1,gbest2,…., ghestD], according to equations (1) and (2).

$$v_{id}(t+1) = v_{id}(t) + c_1 r_1 \left(Pbest_{id}(t) - x_{id}(t)\right) + c_2 r_2 (Gbest_d\ (t) - x_{id}\ (t)) \ \ldots.\ (1)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..\ (2)$$

where c1 and c2 are the cognitive and social acceleration coefficients, and r1 and r2 are two uniform random values generated within [0, 1] interval.

A feature selection is an example of a binary optimization problem. A solution for it is usually a vector of n dimensions and each index can have one arbitrary value 0 or 1. Model values of 0 or 1; with 0 meaning that a feature is not selected and 1 indicating select. Such a useful tool based on PSO can be a binary vector to solve the feature selection problem and practical implementation of its functionality. The idea is that continuous optimization problems in real-valued variables will be encoded into binary variables,For binary optimization, position updates involve flipping the value from 0 to 1 or from 1 to 0.Hence, for PSO to be employed towards feature selection problems, a transfer function is applied which can convert the real-valued positions of solutions into binary values (37) PSO was used to optimize features by searching the feature space and identifying the subset that delivered maximum predictive accuracy of the model.

2.5.2 Lasso (Least Absolute Shrinkage and Selection Operator)

Lasso regression, which employs L1 regularization, was applied to shrink the coefficients of less important features to zero. This method facilitates automatic feature selection by penalizing the absolute size of the regression coefficients, thereby retaining only the most significant features (38). Lasso was instrumental in reducing the complexity of the model without sacrificing performance.

Lasso regression is a linear model that uses the following cost function:

$$Cost\ Function = \sum_{i=1}^{n}(y_i - \sum_{j=1}^{p} a_j\ x_{ij})^2 + \alpha \sum_{j=1}^{p}|a_j| \quad \ldots\ldots\ldots\ldots\ldots\ldots (3)$$

Where a_j is the coefficient of the j-th feature. The final term in the equation is known as the L1 penalty and α is the hyperparameter that controls the strength of this penalty term. The cost function value increases with the feature coefficient magnitude. The goal of Lasso regression is to minimize the cost function by minimizing absolute coefficient values. The method proves successful when the features receive prior scaling treatment through standardization or alternative scaling methods. The α hyperparameter needs determination through cross-validation methods.

The process of using Lasso regression for feature selection involves running Lasso regression on a scaled dataset and selecting only the features with non-zero coefficients. We need to determine the α hyperparameter before applying Lasso regression.

2.5.3 Wrapper Method Feature Selection

The Wrapper Method tests multiple feature combinations to discover the most effective subset which matches the performance requirements of a particular machine learning algorithm(39). The approach includes three techniques: forward selection, backward elimination and stepwise selection. The techniques use incremental feature addition or removal to find an optimal set which creates a balance between model accuracy and complexity.

The thesis uses Recursive Feature Elimination (RFE) as its method for feature elimination. RFE works by recursively removing the weakest features to create a model with fewer attributes. This consists of the following steps:

1. Train the Model: We train our model with all features available.

2. Rank Features: This will rank all features by their importance scores (which can be model coefficients in linear models or feature importance scores in tree-based models)
3. Remove Features: You can remove the least important feature(s).
4. Repeat: Train the model on the remaining features and repeat steps until we reach to desired number of features or any stopping criteria are met.

RFE is especially useful to simplify models, reduce computational time and improve model performance by removing irrelevant or redundant features.

2.5.4 SelectKBest Feature Selection

SelectKBest is a univariate feature selection that selects top 'k' features based on statistical tests (40) In this study, we used SelectKBest to assess each feature individually and selected those with the highest scores. Therefore, this method acted as a quick way to get the most important features based on their statistical significance.

Here's a step-by-step explanation of how the best features are selected with SelectKBest:

1. Feature Selection Method: We provide a feature selection method while defining the SelectKBest object mentioning the scoring function. This scoring function specifies how the respective feature is related to the target variable (or whatever it is that you want to predict). Depending on what works with respect to the feature set, one of these built-in scoring functions will be used for feature selection: chi-squared, mutual information or F-value (ANOVA). We selected chi-squared scoring function in our thesis.
2. Scoring : The feature scoring is a name mapping to each individual feature on which we wish to apply a scoring function on. It measures the "goodness" of each feature in describing or predicting the target variable.
3. Ranking Features: Using the scores given by the scoring function, the model ranks all features.The higher the score, the more relevant or important the feature is; the lower the score less important.
4. Selecting the Top Features: The SelectKBest technique selects the top 'k' features with the most powerful scores, where 'k' is what we specified when creating this SelectKBest object.

2.5.5 SelectPercentile Feature Selection

Like SelectKBest, SelectPercentile selects features based on their statistical scores; the selection of features, though, depends directly on a percentile threshold level instead of a fixed number of features. We have used SelectPercentile to keep only the best features that fall within a certain percentile (41), hence selecting only those features that were most relevant for the model.

Following is how SelectPercentile works in a step-by-step manner:

1. Feature Selection Method: At the time of creation of the object SelectPercentile, a scoring function is mentioned which defines the relationship between each feature and the target variable. Some common scoring functions include Chi-squared, F-value (ANOVA), and Mutual Information. We chose the latter, Mutual information, for our thesis.
2. Scoring: Next, each feature in the dataset is subjected individually to a scoring function that determines how important the feature is towards predicting the target variable.
3. Ranking Features : The ranking of the features is obtained by their scores from the scoring function; higher scores mean higher relevance.
4. Selecting Feature by Percentile: SelectPercentile returns features within a specific range of the highest percentage of scores. For example, if the percentile threshold is set to 10%, the method retains only the top 10% of the features based on their scores. In this thesis, we tuned this parameter for this percentile threshold in order to find the best selection on our dataset.
5. Inclusion in the Model: Only those features that fall within the specific percentile will make the cut into the final model; hence, the best and most relevant features the model is built on.

2.6 Machine Learning Algorithms

Machine learning becomes an integral part of Educational Data Mining, therefore robust tools and techniques are provided for predicting academic performance. Selection of appropriate machine learning algorithms is rather crucial in order to develop an effective predictive model that could put forward valuable insight and recommendations. The various machine learning algorithms used in this study for the prediction of students' academic performance will be discussed here.

2.6.1 Decision Tree Model

Decision trees are probably the most popular techniques in data mining and they take the form of flowcharts: each internal node represents a test on an attribute; every branch represents an outcome of the test; while each terminal node indicates a class label. Several researchers have commented that decision trees are easily interpreted since they are based upon IF-THEN rules (42).

The basic decision tree algorithm starts by selecting a root node. Then, it computes information gain or entropy for all possible splits. The higher the information gain, the lower the entropy, so that is how it decides which node to split. It will continue to split nodes and recalculate until no more splits are allowed, or until the entropy is at a minimum. Entropy is a metric of impurity that designates uncertainty or randomness within data, while information gain is a measure of an entropy reduction before and after a split (43).

entropy H(S) is defined as:

$$H(S) = -\sum_{i=1}^{n} p_i \ log_2 p_i \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4)$$

where pi is the proportion of examples in class i .

information gain IG(S,A) for a dataset S and attribute A is defined as:

$$IG(S,A) = H(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} H(S_v) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (5)$$

where:

- H(S) is the entropy of the original set S.
- Sv is the subset of S where attribute A has value v.
- Values(A) is the set of all possible values of attribute A.
- |Sv|\|S| is the proportion of examples in S that have the value v for attribute A.
- H(Sv) is the entropy of the subset Sv .

Implementation Details: In this study, we implemented the Decision Tree classifier using the Gini Impurity criterion to measure node purity. The model was trained using the CART (Classification and Regression Trees) algorithm**,** with a maximum depth of 10 to prevent overfitting and Post-pruning was applied to prevent overfitting.

2.6.2 Random Forest

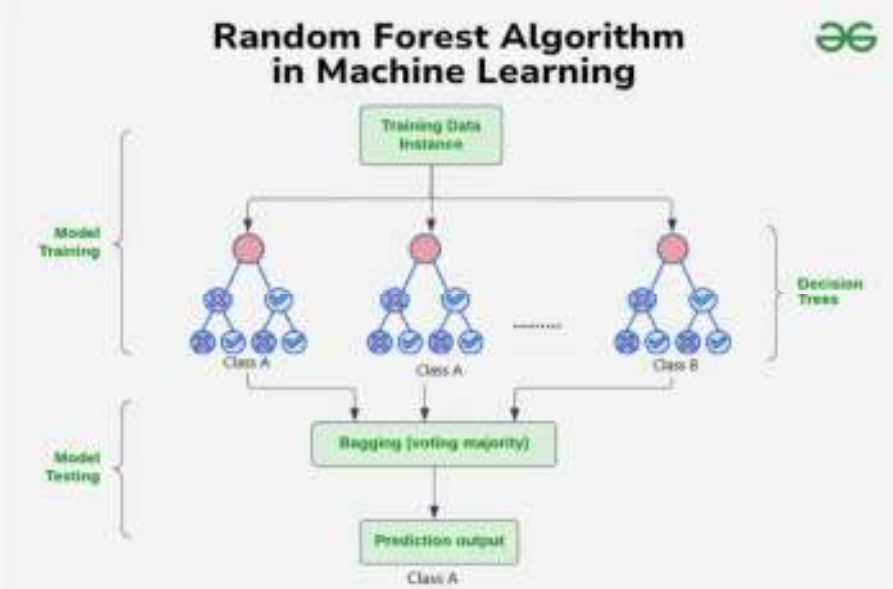
The Random Forest algorithm is one of the most robust algorithms in machine learning. It creates a forest of decision trees during training. Every tree in this forest has been created based on a random subset of the dataset and a random subset of features; hence, this results in variability and reduces the risk of overfitting (44). It predicts values by combining outputs from all the trees created by voting for classification tasks or averaging for regression analysis. This ensemble approach brings stability and higher accuracy with which the results can be reliable and reasonably accurate.

Implementation Details: The Random Forest classifier in this study was trained using 100 trees with the Gini Impurity criterion for splitting. We set the maximum depth to 10 and used bootstrap sampling to enhance model generalization.

The picture below illustrates how the Random Forest algorithm works.

Figure 7

Random Forest Algorithm



Note. See (45)

2.6.3 Linear Regression

Linear Regression is actually a supervised machine learning algorithm that models the linear relationship between a dependent variable and one or more independent variables by fitting a linear equation to observed data. If the number of independent variables is one, the technique is called Simple Linear Regression; when there is more than one

independent variable, then the technique is known as Multiple Linear Regression(46). We use Simple linear regression in our research study.

The equation for simple linear regression is:

$$y=\beta 0+\beta 1X \quad \text{...} \quad (6)$$

Where:

- Y is the dependent variable
- X is the independent variable
- β0 is the intercept
- β1 is the slope

The goal of the algorithm is to find the best-fit line equation that can predict the values of the dependent variable based on the independent variable.

Implementation Details**:** In this study, we implemented Simple Linear Regression**,** using the Ordinary Least Squares (OLS) method for parameter estimation to minimize the sum of squared errors.

2.6.4 Naive Bayes

The Naive Bayes classifier is a family of supervised learning algorithms based on Bayes' theorem, which is used to calculate the probability of an event based on prior occurrences. These methods assume that all features are conditionally independent given the class variable, an assumption that, despite its simplicity, allows Naive Bayes classifiers to perform well in real-world applications. The algorithm learns conditional and class probabilities from the training data and uses these values to classify new observations effectively (47).

The equation for Bayes' theorem is:

$$P(C|X) = \frac{P(X|C).P(C)}{P(X)} \quad \text{...} \quad (7)$$

Where:

- P(C|X) is the posterior probability of class C given feature X.
- P(X|C) is the likelihood of feature X given class C.

- P(C) is the prior probability of class C.
- P(X) is the prior probability of feature X.

Implementation Detail: Gaussian Naive Bayes model was utilized where we assume feature values have a normal distribution. This was appropriate for our dataset, which contained continuous numerical features.

2.6.5 Logistic Regression

Logistic regression is a popular algorithm for solving classification problems. Its name comes from the logistic, or sigmoid function s-shaped curve that maps real-valued numbers into values between 0 and 1, excluding the limits (48).

The logistic function is defined as:

$$\sigma(x) = \frac{1}{1+e^{-x}} \quad \text{......} \quad (8)$$

where σ(x) is the output, and x is the input value. Logistic regression can be categorized into binomial, multinomial, or ordinal types, depending on the nature of the classification task.

Implementation details: We have used Binary Logistic Regression with L2 regularization (Ridge Regression) to avoid overfitting.

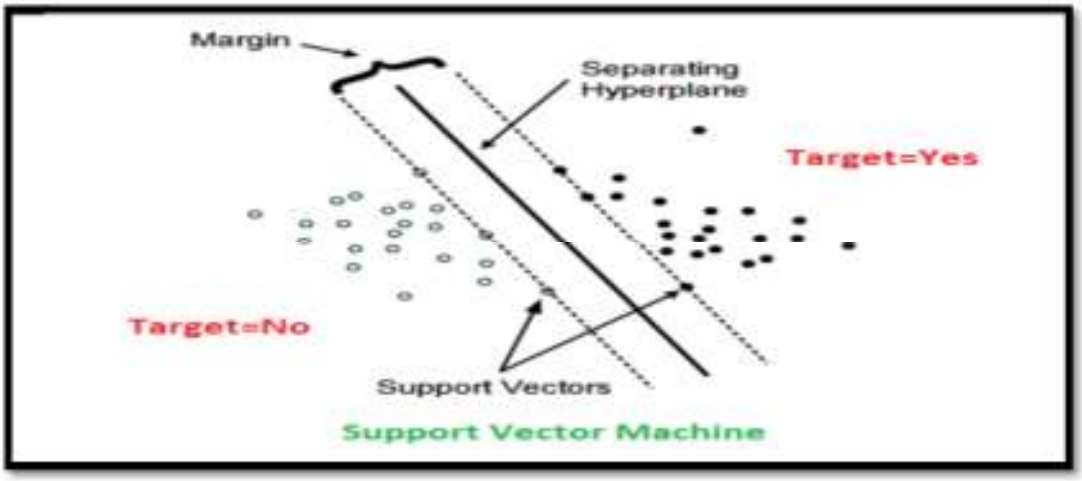
2.6.6 Support Vector Machine (SVM)

Support Vector Machines are supervised algorithms that are used more often than not in solving problems of classification. The goal of SVM is to identify a hyperplane in an N-dimensional space that optimally separates data points into distinct classes. Among many potential hyperplanes, the one that maximizes the margin—the maximum distance between the nearest data points of each class—is selected. Hyperplanes act as decision boundaries that classify data points; those on different sides of the hyperplane are assigned to their respective classes (49). Support vectors are the data points closest to the hyperplane and are critical in defining its position and orientation. The margin is maximized by focusing on these support vectors, and removing them can significantly alter the hyperplane's placement (21).

Implementation Details: In this study, we implemented SVM with the Radial Basis Function (RBF) kernel**,** which helps handle non-linearly separable data. The regularization parameter (C) was set to 1.0 for balancing margin maximization and misclassification, and the gamma parameter was set to 'scale'**,** allowing the model to adapt automatically to the dataset's variance.

Figure 8

SVM Formulation



2.6.7 Neural Networks

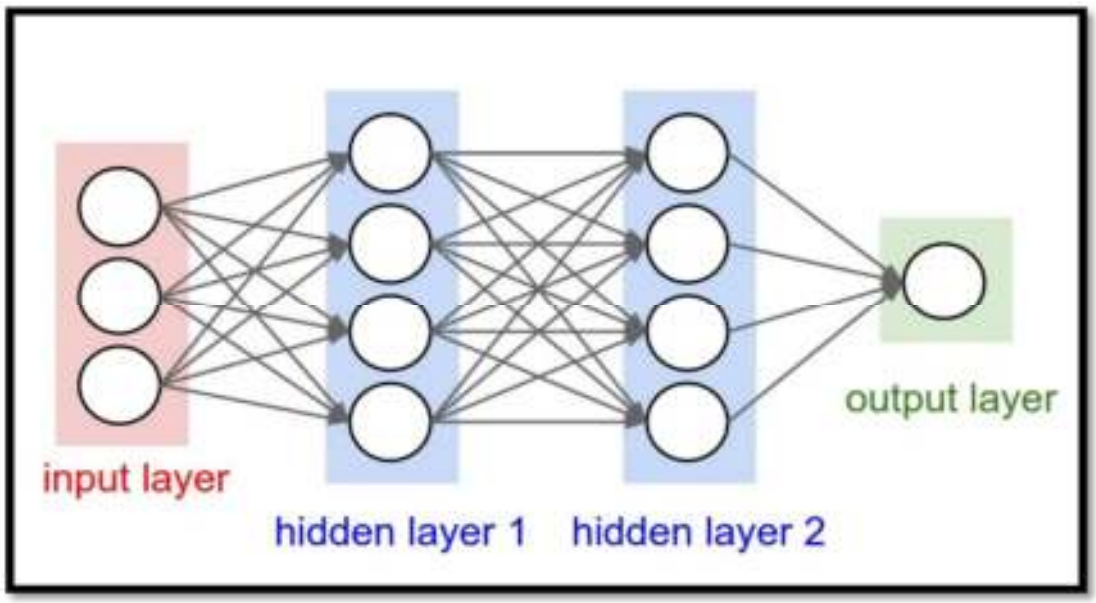
The term neural networks refers to a collection of algorithms inspired by the structure and function of the human brain and find mainly useful applications in pattern recognition within data. It accomplishes this by grouping similar data without labels, or after training, classifying data with labels. Deep neural networks -also sometimes called "Deep Learning"-represent advanced components used in wider applications of machine learning: reinforcement learning, classification, and regression. ANNs are based on hidden states, somewhat similar to neurons, which process inputs through many layers of interconnected artificial neurons to generate outputs(50). These hidden states have a probabilistic behavior and act as mediators between the input data and the final output.

Implementation Details: We implemented a feedforward neural network with three layers: an input layer, one hidden layer with 64 neurons captures enough complexity without overfitting, and an output layer. The ReLU activation function was used in the hidden layer avoids vanishing gradients, speeds up training, while the sigmoid activation function was used in the output layer for binary classification. The model was

trained using the Adam optimizer Efficient, adaptive, prevents local minima, with a learning rate of 0.001 it balances convergence speed and stability, and the binary cross-entropy loss function was used for optimization it standard for binary classification.

Figure 9

Structure of a Multi-Layer Artificial Neural Network



Note. See (51)

An Artificial Neural Network (ANN) employs activation functions within its nodes to determine the output based on the given inputs. The sigmoid function is a specific type of activation function characterized by its S-shaped curve. It's a variant of the logistic function that produces probability outputs ranging from 0 to 1 based on input values.(51).

$$Sigmoid(x) = \frac{1}{1+e^{(-x)}} = \frac{e^x}{1+e^x} \quad \text{..... (9)}$$

2.7. Performance measures

In this research, nine measures were used to evaluate the quality of classification: precision, recall, F-score, accuracy, mean squared error, mean absolute error, R^2 score, confusion matrix, and computational time. The equations and descriptions for precision, recall, F-score, accuracy, MSE, MAE, R^2 score, confusion matrix, and computational time are detailed in the table 2 below:

Table 2

Performance measures and there equations

Performance measures	Description	Equations
Precision	The ratio of correctly classified cases to the total number of classified cases.	$\frac{TP}{(TP+FP)}$
Recall	The proportion of correctly classified cases to the total number of actual cases.	$\frac{TP}{(TP+FN)}$
F1-score	Combines precision and recall, providing a balanced measure of their relationship.	$2 * \frac{Precision * Recall}{Precision + Recall}$
Accuracy	The ratio of correctly predicted instances to the total predictions.	$\frac{(TP+TN)}{(TP+TN+FP+FN)}$
Mean Squared Error (MSE)	Measures the average squared difference between the estimated values and the actual value.	$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$
Mean Absolute Error (MAE)	The average of the absolute differences between predicted and actual values.	$\frac{1}{n}\sum_{i=1}^{n}\lvert y_i - \hat{y}_i \rvert$
R^2 score	Indicates the proportion of the variance in the dependent variable explained by the independent variables.	$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}{\sum_{i=1}^{n}(y_i-\bar{y})^2}$
confusion matrix	A table that displays the performance of a classification model by showing the actual versus predicted values.	
computational time	Refers to the time needed for the algorithm to perform the classification, which is key for evaluating its efficiency and practicality	N/A.

Here are the key terminologies for understanding performance metrics:

- True Positive (TP): The actual value is Positive, and the predicted value is Positive.
- True Negative (TN): The actual value is Negative, and the predicted value is Negative.

- False Positive (FP) / Type I Error: The actual value is Negative, but the predicted value is Positive.
- False Negative (FN) / Type II Error: The actual value is Positive, but the predicted value is Negative.
- n: The number of observations.
- yi: The actual value for the i -th observation.
- y_hat: The predicted value for the i-th observation.

2.8 Algorithm Tuning and Modification

Modifications and tuning machine learning algorithms, in combination, improve model performance, parameterize, and tailor the algorithm to the nature of the input dataset. In the domain of education, these enhancements result in higher quality estimates of student performance. Optimal fine-tuning e.g., hyperparameter optimization and algorithmic adjustment e.g., using more features or more sophisticated procedures lead to better model performance, better and more reliable generalization, and decisions.

- Fine Tuning Model's Parameters using Grid Search

To further improve the predictive accuracy of this model, we used a grid search to look through various combinations of hyperparameters. We systematically evaluated each combination of hyperparameters to identify those that perform best on our dataset. This process was an important refining of model performance and the necessity of robustness of results.

- Diverse Evaluation Metrics

While evaluating the model, we make use of various metrics to assess different aspects of performance. This is not just about accuracy, but among others, precision, recall, or the F1 score, enabling a more wholesome view of the model's behavior under conditions of variation.

- K-Fold Cross-Validation for Robustness

K-fold cross-validation is a strategy we used to validate our results. We divided the dataset into k parts; training and testing were done k times by moving the test fold after each iteration. This helps in minimizing overfitting and confirming the model on generality other than from the training data.

- Improving the Model by Polynomial Feature Engineering

We also designed features to capture the non-linear relationships, for instance, designing polynomial features. This enabled the model to learn more complex patterns in the data, which was particularly useful with our linear regression model.

- Combining Techniques to Improve the Outcomes

Combine these techniques: hyperparameter tuning, rich evaluation metrics, robust validation, and feature engineering in one go to have a well-optimized model. The advantages of this approach are that it does not just increase the accuracy but also justifies the result against the objective of the study.

Chapter Three

Results

The following section presents the results of the study concerning the above-stated research objectives. Further details are provided in subsequent sections, summarizing data analyses, model performances, feature importance, and statistical findings.

3.1 Descriptive Statistics of Key Features

The purpose of this section is to give an overview of the main features inherent in the dataset, which are essential to predicting the academic performance of the students. Descriptive statistics of these features provide their central tendencies, variability, and their overall distribution. This gives insight into the underlying patterns and relationships that are inherent in the data, which will be helpful for effective feature selections and model building.

Table (A.3) in Appendix A gives the mean, median, standard deviation, minimum, and maximum values for each key feature. This statistic enables an efficient overview of the dataset, highlighting its central characteristics and the range of values.

3.2 Model Performance Metrics

We conducted an extended review of several machine learning models and feature selection methods. In Table (A.4) in Appendix A, an overview of performance metrics that are time, accuracy, MAE, MSE, R^2, recall, precision, F1 score, and confusion matrix per model-method combination is given. In this section, the main performance metrics will be discussed: time, accuracy, recall, precision, and the F1 score. Table 3 shows the accuracy values for the classifiers using different feature selection algorithms.

Table 3

Accuracy values of classifiers using different feature selection algorithms

Algorithm / Feature Selection	Random Forest	Decision Tree Classifier	Naïve Bayes	SVC	Logistic Regression	Linear Regression	MLP
PSO Feature Selection	0.64677	0.860697	0.7413	0.6617	0.900498	0.860697	0.5622
(RFE)	0.91045	0.870647	-	0.9154	0.910448	0.905473	0.9254
SelectKBest	0.91045	0.850746	0.8756	0.9104	0.915423	0.935323	0.9154
Select Percentile	0.93035	0.820896	0.8806	0.9204	0.910448	0.925373	0.9254
LASSO Regression	0.89055	0.890547	0.9005	0.9055	0.915423	0.900498	0.9104

Figure (B.1) in Appendix B provides a comprehensive, visual comparison of the accuracy values for the various models and techniques of feature selection.

The accuracy of linear regression had the highest scores among different feature selection methods and was high across, peaking at 0.9353 when applied together with SelectKBest. This agrees with the characteristics of the dataset: small, labeled, numerical features, low complexity, and low noise. These characteristics favor simpler models such as linear regression, which also requires less handling of missing values and complex feature interactions.

Table 4

The recall values of classifiers using different feature selection algorithms

Algorithm / Feature Selection	Random Forest	Decision Tree Classifier	Naïve Bayes	SVC	Logistic Regression	Linear Regression	MLP
PSO Feature Selection	0.84874	0.94958	0.8656	0.916	0.966387	0.932773	0.7143
Wrapper Method (RFE)	0.96639	0.941176	-	0.97479	0.97479	0.957983	0.9748
SelectKBest	0.96639	0.915966	0.9076	0.9664	0.98312	0.97479	0.9832
Select Percentile	0.97479	0.907563	0.916	0.9664	0.97479	0.97479	0.9664
LASSO Regression	0.94118	0.932773	0.8913	0.958	0.98312	0.966387	0.9748

Table 4 summarizes the recall scores of different classifiers when applied with various feature selection algorithms.

Figure (B.4) in Appendix B provides a comprehensive, visual comparison of the recall values for the various models and techniques of feature selection.

Both the logistic regression and the MLP classifier have the same high recall of 0.9832, with their ability to effectively utilize selected features with the aim of model sensitivity. This is probably because of how the classifier models are aligned in characteristics with the dataset-small, numerical, of low complexity, and with no handling of missing values.

Table 5

The Precision values of classifiers using different feature selection algorithms

Algorithm / Feature Selection	Random Forest	Decision Tree Classifier	Naïve Bayes	SVC	Logistic Regression	Linear Regression	MLP
PSO Feature Selection	0.640006	0.866432	0.7410	0.6527	0.877863	0.847328	0.6115
Wrapper Method (RFE)	0.891473	0.854962	-	0.8923	0.885496	0.890625	0.9063
SelectKBest	0.891473	0.844961	0.8853	0.8915	0.886364	0.920635	0.8864
Select Percentile	0.913386	0.81203	0.8862	0.9055	0.885496	0.90625	0.9127
LASSO Regression	0.88189	0.888	0.902	0.8906	0.886364	0.877863	0.8855

Table 5: Precision scores of different classifiers for various feature selection algorithms. Precision tells about the number of true positive predictions out of all the positive predictions the model has made.

Figure (B.5) in Appendix B provides a comprehensive, visual comparison of the precision values for the various models and techniques of feature selection.

Linear regression was on par with all of the feature selection methods, having achieved the highest score with a value of 0.920635 when coupled with SelectKBest. This would agree with the characteristics of the dataset—small, labeled, numerical features, low in

complexity, and low in noise not favoring complex models that require elaborate handling of missing values or interaction features.

Table 6

The F1- score of classifiers using different feature selection algorithms

Algorithm / Feature Selection	Random Forest	Decision Tree	Naïve Bayes	SVC	Logistic Regression	Linear Regression	MLP
PSO Feature Selection	0.73993	0.889764	0.7985	0.7622	0.92	0.888	0.6589
Wrapper Method (RFE)	0.92742	0.896	-	0.9317	0.928	0.923077	0.9393
SelectKBest	0.92742	0.879032	0.8963	0.9274	0.932271	0.946939	0.9323
Select Percentile	0.94309	0.857143	0.9008	0.935	0.928	0.939271	0.9388
LASSO Regression	0.91057	0.909836	0.8957	0.9231	0.932271	0.92	0.928

Table 6 shows the F1 scores of several classifiers for different feature selection approaches. The F1-score is the harmonic mean of precision and recall and can be viewed as a statistic that balances false positives and false negatives.

Figure (B.2) in Appendix B provides a comprehensive, visual comparison of the F1-score values for the various models and techniques of feature selection.

Linear regression consistently obtained high F1-scores across all feature selection methods, with the highest score of 0.9469 when applying SelectKBest. This agrees with the properties of the dataset: small in size, labeled, and numerical features of low complexity and noise. Such conditions favor simpler models, such as linear regression, that require minimal processing for missing values and complex feature interactions.

Table 7

The Time values in seconds of classifiers using different feature selection algorithms

Algorithm / Feature Selection	Random Forest	Decision Tree Classifier	Naïve Bayes	SVC	Logistic Regression	Linear Regression	MLP
PSO Feature Selection	361.4638	26.34085	2.8724	26.577	23.27806	11.31164	1127.1
RFE	27.39932	0.502014	-	3.6972	0.340429	0.107759	1798.8
SelectKBest	0.710936	0.039985	0.0144	0.0455	0.022921	0.038208	1.021
Select Percentile	1.255736	0.657924	0.1499	0.1954	0.206763	0.3761	1.4482
LASSO Regression	1.072673	0.505332	0.1388	0.2020	0.196395	0.308849	1.3474

Table 7: Computational time in seconds required by different classifiers when various feature selection methods were applied. Since many applications could be using large datasets or even requiring real-time decision-making, computational time is often one of the important metrics to evaluate the efficiency of machine learning models.

Figure (B.3) in Appendix B provides a comprehensive, visual comparison of the time values for the various models and techniques of feature selection.

Among all the models, Naïve Bayes simpler models had shorter computation times. Naive Bayes has been known to be fast and efficient in classification tasks due to its easy probabilistic model, independence assumption, efficient training, fast prediction, and low usage of memory. The application of Bayes' theorem, independent features, and one pass through the training data makes Bayes Naive so fast that it is even practical in big datasets.

3.3 Feature Indices and Names

For clarity and reference, Table (A.5) in Appendix A lists the indices and corresponding feature names used in the analysis. This helps in understanding which features contributed to the model's performance.

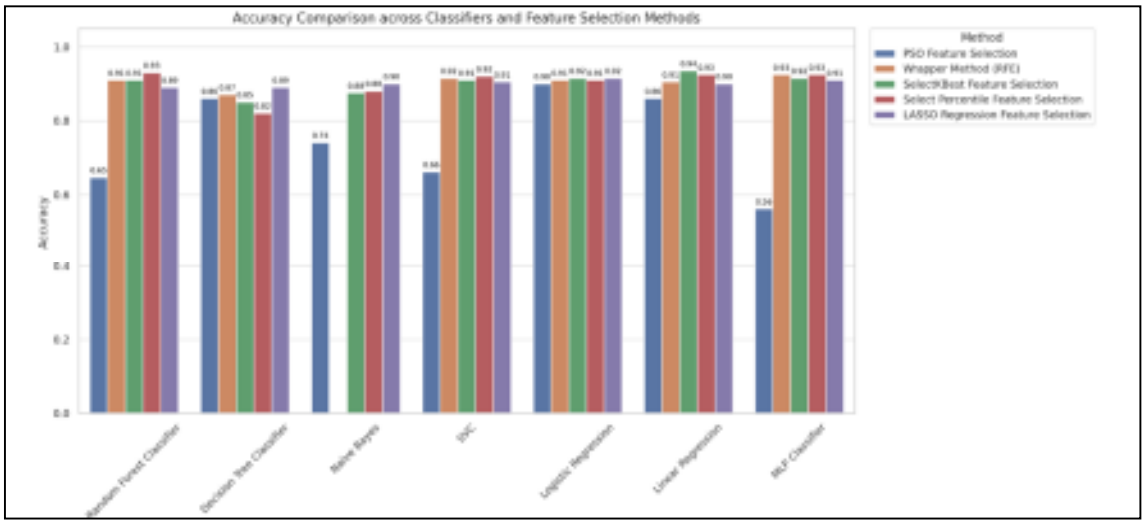
3.4 Visualization of Model Performance

- **Model Accuracy Bar Plots**

We have also visualized the performance of different models by creating bar plots of model accuracy for various feature selection techniques. The plots give an easy view of the performance of every model across various feature selection techniques.

Figure 10

Accuracy of various models for different feature selection methods



- **Computational Time Analysis**

The relevant section in Figure (B.1) in Appendix B compares the accuracies of the various machine-learning models with various feature selection methods. Each bar, as depicted within the Appendix, gives the models' respective accuracies considering the different feature selection methods, thereby allowing for the clear and visual comparisons of the performances.

In Figure (B.2) in Appendix B, F1 scores of different combinations between ML algorithms with feature selection methods are compared. The F1 score is a combination of precision and recall, thereby giving an insightful measure of model performance.

Figure (B.3) in Appendix B indicates the computation time for all ML models combined with different feature selection methods. Recognizing the computation time is key to the efficiency and practical applicability of each model in a real-life situation.

3.5 Top six models performance. Feature selection comparisons before and, after adjustments

The evaluation of the performance metrics, for the six models and the process of selecting features and, after making changes.

In this section we assess the effectiveness of the six machine learning models both before and, after implementing changes. The subsequent tables offer an overview of the performance indicators showcasing the impact of these adjustments.

The best hyperparameters chosen, amongst the ones tested for each of the machine learning classifiers involved in this research, are summarized in Table 8. All these hyperparameters were tuned using a systematic grid search to optimize the values for each model's best performance.

Hyperparameter tunning

Table 8

Hyperparameter tunning of top six models with feature selection after modifications

Classifier Name	Parameters
Random Forest	'max_depth': 10, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 5, 'n_estimators': 100 criterion: gini
Decision Tree	ccp_alpha: 0.0 ,class_weight: None ,criterion: gini ,max_depth: None max_features: None , max_leaf_nodes: None ,min_impurity_decrease: 0.0 min_samples_leaf: 1 ,min_samples_split: 2 ,min_weight_fraction_leaf: 0.0 random_state: 42 ,splitter: best
Naive Bayes	priors: None ,var_smoothing: 1e-09
Support Vector Machines	'C': 1, 'gamma': 'scale', 'kernel': 'linear'
Logistic Regression	Regularization strength (C): 1.0 , Penalty: L2 regularization ,Solver: Limited-memory Broyden–Fletcher–Goldfarb–Shanno (lbfgs) , Other parameters were kept at their default values.
Linear Regression	copy_X: True ,fit_intercept: True ,n_jobs: None ,positive: False
Neural Network for with Wrapper Method (RFE))	activation='relu' ,hidden_layer_sizes=(100,) , solver='adam' ,learning_rate_init=0.001 , max_iter=200 , random_state=42 , Other parameters were kept at their default values
Neural Network with SelectPercentile	activation='tanh', hidden_layer_sizes=(50, 50), solver='sgd',learning_rate_init=0.001, max_iter =500, momentum=0.9, nesterovs_momentum=True,random_state=42 , Other parameters were kept at their default values

Table 9

Performance metrics of top six models with feature selection before and after modifications

Model + Feature selectio		Linear Regression +SelectKBest	Random Forest+ Select Percentile	Linear Regression +Select Percentile	MLP+ Select Percentile	MLP + (RFE)	SVC+ Select Percentile
Indices	Before	[4 5 8 9 20 24 25 29 34 38]	[4 7 8 14 24 25 27 37 40]	[8 14 16 18 19 24 25 28 30]	[4 8 9 12 20 24 25 28 37]	[8 10 15 24 25 26 34 38 40 43]	[1 2 4 8 10 24 25 32 43]
	after	The same	[2 4 8 12 18 23 24 25 31]	[4 6 8 18 22 24 25 37 42]	[4 8 20 24 25]	The same	[2 8 11 16 24 25 32 38 43]
Time	Before	0.038208	1.255736	0.3761	1.448239	1798.751	0.195425
	after	0.021954	7.67652826			1216.74	
Accuracy	Before	0.935323	0.930348	0.925373	0.925373	0.925373	0.920398
	after	0.93532338	0.91542288	0.9303482587	0.925373	0.925373	0.9353234
MAE	before	0.269019	0.069652	0.274192	0.074627	0.074627	0.079602
	after	0.269019	0.0845771	0.27579957	0.074627	0.074627	0.0646766
MSE	before	0.098197	0.069652	0.100048	0.074627	0.074627	0.079602
	after	0.098197	0.084577	0.101823457	0.074627	0.074627	0.0646766
R2	before	0.593437	0.711621	0.585773	0.691023	0.691023	0.670424
	after	0.593437	0.64982578	0.5784208356	0.691023	0.691023	0.7322197
Recall	before	0.97479	0.97479	0.97479	0.966387	0.97479	0.966387
	after	0.97479	0.9747899		0.966387	0.97479	0.983193
Precision	before	0.920635	0.913386	0.90625	0.912698	0.90625	0.905512
	after	0.920635	0.89230769		0.912698	0.90625	0.9140625
F1	before	0.946939	0.943089	0.939271	0.938776	0.939271	0.934959
	after	0.946939	0.9317269		0.938776	0.939271	0.9473684
Confusion _Matrix	before	[[72 10] [3 116]]	[[71 11] [3 116]]	[[70 12] [3 116]]	[[71 11] [4 115]]	[[70 12] [3 116]]	[[70 12] [4 115]]
	after	The same	[[68 14] [3 116]]		The same	The same	[[71 11] [2 117]]

Chapter Four

Discussions and Conclusions

4.1 Discussions

4.1.1 Descriptive Statistics of Key Features

Table (A.3) in Appendix A presents the descriptive statistics of the key features used in our study. These statistics provide valuable insights into the central tendencies, variations, and distributions of the data.

The mean, median, and standard deviation of key features, such as study time, family relationship quality, and parental education levels, highlight the typical values and the spread of the data. For instance, the average study time indicates the general amount of time students dedicate to their studies, while the variability provides insights into the differences among individual students.

- Impact of Study Time on Academic Performance:
- Average study time can determine the exact time being utilized by the students towards their studies.
- Comparison of study time with academic performance can allow us to discuss whether an increase in the study time in turn means good grades given the other measures of academic performance.
- Family Relationship Quality:
- Family relationship quality statistics may show how home environments supportive of students impact their performances.
- To what extent will it affect students in terms of concentration and mental stability resulting in improved academic performance if the student comes from strong family relationships?
- Parental Education Levels:
- The mean and median parental educational values observed in the dataset provide valuable information about the general educational status of the students' families. These are conceptual tools that allow researchers to gain a sense of that learning environment into which students are likely born. Parental education, on the other hand, is generally related to increased academic support, which includes assistance

with school assignments, motivation for academic success, and access to learning materials.

4.1.2 Model Performance Metrics

The performance metrics of each of the seven machine learning models evaluated with five different feature selection methods are presented in Tables 4, 5, 6, 7, 8, and table (A.4) in Appendix A. These metrics include accuracy, precision, recall, F1 score, mean squared error (MSE), mean absolute error (MAE), R^2 score, confusion matrix, and computational time. Important observations and intuitions gained from these results are given in the following section.

1. Accuracy

- Among all models, LinearRegression with the SelectKBest Feature Selection had the best accuracy of 0.935323. This means that the linear regression model has a very strong potential to predict performance if this particular feature selection method is applied.
- The worst-accuracy score belonged to MLPClassifier with PSO Feature Selection, which hinted that some problems might occur either with the relevance of features or model suitability for this task.

2. Precision, Recall, and F1 Score:

- Precision and recall give information about the performance of the model in correctly identifying positive cases. For example, the best precision, 0.920635, was produced by linear regression combined with SelectKBest feature selection; hence, it had a low number of false positives.
- The recall was highest for the MLPClassifier with SelectKBest Feature Selection , at 0.983193, suggesting that this model is particularly good at identifying all relevant instances of the target class.
- The F1 score, balancing precision and recall, was highest for the Linear Regression combined with the SelectKBest Feature Selection. At precisely 0.946939, this suggests this combination provides a balanced and robust performance.

3. Mean Squared Error (MSE) and Mean Absolute Error (MAE):

- The lowest MSE and MAE were for RandomForestClassifier with SelectPercentile feature selection, this would then be indicative of high predictive accuracy with low deviation from actual values.

- In contrast, the highest MSE and MAE were for MLPClassifier with PSO Feature Selection, which would mean higher discrepancies between predicted values and real values.

4. R^2 Score:

- The R^2 score indicates how well the model explains the variability of the target variable. The highest R^2 score was achieved by the RandomForestClassifier with SelectPercentile Feature Selection, implying that this model accounts for the majority of variance in student performance.
- The lowest R^2 score was noted for the MLPClassifier with PSO Feature Selection, suggesting limited explanatory power.

5. Confusion Matrix:

It can be observed from the confusion matrix of each model and feature selection method which kind of error has been committed. For instance, LinearRegression with SelectKBest Feature Selection, with a high number of true positives and true negatives shows that this classifier has performed the classification job quite effectively.

6. Computational Time:

- Computational efficiency is especially important in large datasets. Therefore, Naïve Bayes with SelectKBest Feature Selection was computationally the fastest at 0.014349, followed by LogisticRegression and then LinearRegression. This makes it preferable for real-time or large dataset applications.
- Conversely, The MLPClassifier with the Wrapper Method (RFE) was the longest in computation, therefore suggesting high accuracy may be gotten at the cost of computational time.

Summary of Key Findings

- The LinearRegression with SelectKBest Feature Selection consistently performed well across multiple metrics, making it a strong candidate for predicting student performance.
- There is a notable trade-off between model accuracy and computational time, which should be considered when selecting the optimal model for practical applications.

4.1.3 Feature Indices and Names

Table (A.5) in Appendix A lists the feature indices and their corresponding names, selected by various feature selection methods used in this study, highlighting their relevance in predicting student performance. Notably, features with indices 4, 5, 11, 24, 25 and 12, among others, played a crucial role in predicting the target variable. Understanding these key features can help refine the models and improve educational outcomes.

4.1.4. Accuracy of Different Models with Various Feature Selection Methods

Figure 10: Performance of different machine learning models with different feature selection methods. This figure also compares their performance to see the performance of each model and feature selection method in performing the task of student performance prediction.

Performance of Different Models:

Decision Tree Classifier

- PSO Feature Selection: Accuracy is 0.86.
- Wrapper Method (RFE): Accuracy is 0.87.
- SelectKBest Feature Selection: Accuracy is 0.85.
- SelectPercentile Feature Selection: Accuracy is 0.82.
- LASSO Regression Feature Selection: Accuracy is 0.89.

Observation: Among the different feature selection methods, the Decision Tree model with LASSO Regression Feature Selection performs the best, with an accuracy of 0.89. It performs relatively low in the case of the SelectPercentile method with an accuracy of 0.82, thus showing sensitivity to the selection method used.

Linear Regression

- PSO Feature Selection: Accuracy is 0.86.
- Wrapper Method (RFE): Accuracy is 0.91.
- SelectKBest Feature Selection: Accuracy is 0.94.
- SelectPercentile Feature Selection**:** Accuracy is 0.93.
- LASSO Regression Feature Selection: Accuracy is 0.90.

Observation: It can be noticed that the highest accuracy, 0.94, will be provided by Linear Regression here, combined with SelectKBest Feature Selection. Most of these methods reported relatively consistent performances; the worst performance for this model was PSO, with its accuracy equal to 0.86, which reveals that proper feature selection can drastically improve this model.

Logistic Regression

- PSO Feature Selection: Accuracy is 0.90.
- Wrapper Method (RFE): Accuracy is 0.91.
- SelectKBest Feature Selection: Accuracy is 0.92.
- SelectPercentile Feature Selection: Accuracy is 0.91.
- LASSO Regression Feature Selection: Accuracy is 0.92.

Observation: Logistic regression was steadily doing well across all the feature selection methods, especially the highest accuracy in both SelectKBest and LASSO Regression feature selection methods stood at 0.92. It thereby follows that this consistency demonstrates its robustness on binary classification tasks.

Neural Network (MLP Classifier)

- PSO Feature Selection**:** Accuracy is 0.56.
- Wrapper Method (RFE): Accuracy is 0.93.
- SelectKBest Feature Selection: Accuracy is 0.92.
- SelectPercentile Feature Selection: Accuracy is 0.93.
- LASSO Regression Feature Selection: Accuracy is 0.91.

Observation: The Neural Network model accuracy falls drastically for PSO Feature Selection to 0.56 while it is the best for the Wrapper Method and the SelectPercentile Feature Selection, giving an impressive 0.93 and therefore showing its capability in capturing the complex pattern in the data when the right features are selected.

Random Forest Classifier

- PSO Feature Selection: Accuracy is 0.65.
- Wrapper Method (RFE): Accuracy is 0.91.
- SelectKBest Feature Selection: Accuracy is 0.91.

- SelectPercentile Feature Selection: Accuracy is 0.93.
- LASSO Regression Feature Selection: Accuracy is 0.89.

Observation: Random Forest has the highest value of accuracy, 0.93, with SelectPercentile Feature Selection, while in PSO Feature Selection, it has the lowest value of accuracy, 0.65, hence showing its robustness and adaptability to different sets of features.

Naive Bayes

- PSO Feature Selection: Accuracy is 0.74.
- SelectKBest Feature Selection: Accuracy is 0.88.
- SelectPercentile Feature Selection**:** Accuracy is 0.88.
- LASSO Regression Feature Selection: Accuracy is 0.90.

Observation: On the other hand, Naive Bayes yielded the best performance with LASSO Regression Feature Selection at 0.90, while its worst performance is with PSO Feature Selection at 0.74, which again suggests its dependency on effective feature selection for optimal performance.

Support Vector Machine (SVM)

- PSO Feature Selection: Accuracy is 0.66.
- Wrapper Method (RFE): Accuracy is 0.92.
- SelectKBest Feature Selection: Accuracy is 0.91.
- SelectPercentile Feature Selection: Accuracy is 0.92.
- LASSO Regression Feature Selection: Accuracy is 0.91.

Observation: It is observed that the maximum value of SVM's accuracy is with both Wrapper Method RFE and SelectPercentile Feature Selection methods, at 0.92, while the lowest value is from PSO Feature Selection, at 0.66. It just proves how crucial it is picking the feature selection technique, for SVMs performance.

Impact of Feature Selection Methods

- PSO**:** Generally leads to lower accuracy for most models, indicating that it may not be the most effective feature selection method in this context.

- SelectKBest and SelectPercentile**:** These methods resulted in competitive accuracy scores, highlighting their utility in identifying top-performing features based on statistical significance.
- Lasso and Wrapper Method**:** Both approaches demonstrated effectiveness, with models in terms of performance evaluation and feature selection enhancement, for improved model outcomes.

Feature selection methods for choosing features have an influence on the accuracy of models and underscore the importance of picking the right method, for the given machine learning job.

Comparing the Processing Time of Various Models Using Feature Selection Techniques.

It's important to consider the time it takes for machine learning models to process data along, with the feature selection techniques when assessing their practicality for real time applications.You can find information on the time taken by different models with various feature selection methods, in Figure (B.3) in Appendix B.

Random Forest Classifier:

- PSO Feature Selection: 361.46 seconds
- Wrapper Method (RFE): 27.40 seconds
- SelectKBest Feature Selection: 1.26 seconds
- SelectPercentile Feature Selection: 1.07 seconds
- LASSO Regression Feature Selection: 0.71 seconds

Observation: noticed that the Random Forest Classifier takes a lot time when combined with PSO Feature Selection compared to Wrapper Method ,which is also slow but not as much as PSO method; however, SelectKBeset, SelectPercentile and LASSO Regression Feature Selection techniques are more efficient ,with LASSO being the fastest among them all.

Decision Tree Classifier:

- PSO Feature Selection: 26.34 seconds
- Wrapper Method (RFE): 0.50 seconds

- SelectKBest Feature Selection: 0.04 seconds
- SelectPercentile Feature Selection: 0.66 seconds
- LASSO Regression Feature Selection: 0.50 seconds

Observation: The Decision Tree Classifier operates efficiently with SelectKBest Feature Selection but PSO needs additional time. The SelectPercentile , Wrapper Method and LASSO Regression methods complete tasks at a relatively fast pace.

Naive Bayes:

- PSO Feature Selection: 2.87 seconds
- SelectKBest Feature Selection: 0.01 seconds
- SelectPercentile Feature Selection: 0.15 seconds
- LASSO Regression Feature Selection: 0.14 seconds

Observation: Naive Bayes is fastest with SelectKBest Feature Selection. PSO is the most time-consuming method, while SelectPercentile and LASSO Regression are also efficient.

Support Vector Machine (SVM):

- PSO Feature Selection: 26.57 seconds
- Wrapper Method (RFE): 3.70 seconds
- SelectKBest Feature Selection: 0.05 seconds
- SelectPercentile Feature Selection: 0.20 seconds
- LASSO Regression Feature Selection: 0.20 seconds

Observation: SVM shows a significant reduction in computational time with SelectKBest, SelectPercentile and LASSO Regression methods compared to PSO and Wrapper Method. Wrapper Method, though better than PSO, is still considerably more time-consuming.

Logistic Regression:

- PSO Feature Selection: 23.28 seconds
- Wrapper Method (RFE): 0.34 seconds
- SelectKBest Feature Selection: 0.02 seconds

- SelectPercentile Feature Selection: 0.21 seconds
- LASSO Regression Feature Selection: 0.20 seconds

Observation: Logistic Regression is extremely time-efficient with SelectKBest, SelectPercentile and LASSO Regression Feature Selection, and slightly higher but still efficient with Wrapper Method. PSO is the most time-consuming method.

Neural Network (MLP Classifier):

- PSO Feature Selection: 1127.12 seconds
- Wrapper Method (RFE): 1798.75 seconds
- SelectKBest Feature Selection: 1.02 seconds
- SelectPercentile Feature Selection: 1.45 seconds
- LASSO Regression Feature Selection: 1.35 seconds

Observation: The MLP Classifier has the highest computational time with PSO and Wrapper Method, indicating a substantial resource demand. In contrast, SelectKBest, SelectPercentile, and LASSO Regression methods are significantly more time-efficient.

Linear Regression:

- PSO Feature Selection: 11.31 seconds
- Wrapper Method (RFE): 0.11 seconds
- SelectKBest Feature Selection: 0.04 seconds
- SelectPercentile Feature Selection: 0.38 seconds
- LASSO Regression Feature Selection: 0.31 seconds

Observation: Linear Regression is highly time-efficient with SelectKBest, LASSO Regression, Select Percentile and Wrapper Method Feature Selection methods. PSO is the most time-consuming method.

General Observations

1. PSO Feature Selection: Generally leads to higher computational time for most models, suggesting that it may not be the most time-efficient method in this context.
2. Wrapper Method (RFE): It gives accurate results but consumes more time than other methods such as LASSO, SelectKBest, and SelectPercentile.

3. SelectKBest and SelectPercentile: These two methods combined will get a fair deal of accuracy and computational time giving them an edge for the cases where both aspects are essential.
4. LASSO Regression Feature Selection: Consistently the most time-efficient across various models while still providing competitive accuracy, making it a strong candidate for various applications.

The analysis of computation time indicates that the feature selection methods recommended are not only for their good accuracy but also for the time taken to compute them. While PSO and Wrapper Method (RFE) could provide good accuracy, they may not be reasonable options in real-time or constrained resource settings, due to their increasing computation time. Similar with LASSO, SelectKBest, and SelectPercentile , the success of those methods relies on a balance of time efficiency to performance. In practical application, time efficient methods may be preferred. It all provides a frame of reference for time and efficiency needs of the different feature selection methods and models. This reference frame will guide the choice of methods for specific use cases in machine learning projects.

Accuracy, F1, and Time Comparison Across Machine Learning Models and Feature Selection Methods

The analysis of Figures (B.1), (B.2) and (B.3) in Appendix B presents a detailed comparison of the results for different machine learning models combined with different feature selection methods.

The results in Figure (B.1) in Appendix B demonstrate that most models reach high accuracy levels through SelectKBest, SelectPercentile and LASSO Regression feature selection methods. The three models Logistic Regression, Linear Regression and SVM demonstrate the best accuracy results across almost all feature selection approaches. Random Forest achieves high accuracy levels especially when using SelectPercentile. The Naive Bayes model performs best with LASSO Regression but MLP Classifier achieves better results when using SelectPercentile and Wrapper Method (RFE)**.**

The results in Figure (B.2) in Appendix B confirm the previous findings by demonstrating equivalent F1 score patterns which show that the high accuracy rates are

supported by good precision and recall performance. The F1 score results show Logistic Regression, Linear Regression and SVM models achieving the highest values when using different feature selection approaches. The Decision Tree, Random Forest and MLP classifiers demonstrate good performance but their results vary based on the selected feature selection approach.

The results in Figure (B.3) in Appendix B demonstrate that PSO and Wrapper Method (RFE) require the longest computational time for all models. The MLP Classifier, in particular, shows extremely high computational times with these methods, indicating substantial resource demands. In contrast, SelectKBest, SelectPercentile, and LASSO Regression are much more efficient across all models, with LASSO Regression being the most time-efficient. Linear Regression and Naive Bayes models show the least computational time across all methods, making them suitable for real-time applications.

4.1.5 Performance Comparison and Model Optimization

The performance metrics in Table 9 show a detailed comparison of the top six models before and after applying various optimizations. These optimizations include hyperparameter tuning as shown in Table 8 , cross-validation, polynomial feature expansion, data augmentation using SMOTE, regularization, model stacking, feature scaling, and dynamic feature selection thresholding.

Key Observations and Implications from Table 8:

- Random Forest and Decision Tree: Parameters like max_depth and min_samples_split were critical in preventing overfitting, while a relatively large number of estimators (100) for Random Forest provided robustness.
- Naive Bayes and SVM: Simple and stable settings with minimal hyperparameters, suggesting that these models rely more on the quality of data rather than complex parameter tuning.
- Neural Networks: Different architectures and activation functions (e.g., ‘relu’ vs. ‘tanh’) demonstrated flexibility in handling various feature patterns, with the neural network using SelectPercentile achieving deeper insights through multiple layers.
- Logistic and Linear Regression: The bias-variance trade-off was managed through the optimization of regularization parameters (C and penalty) to achieve stable models which avoided overfitting..

These parameters demonstrate a customized method of model optimization which utilizes the particular advantages of each classifier to enhance predictive accuracy and generalizability.

Key Observations and Implications from Table 9:

Accuracy Comparison:

- The Linear Regression (SelectKBest) preserved its 93.53% accuracy after optimization because the first model configuration was already optimal.
- The RandomForest Classifier (SelectPercentile) experienced a 1.49% decrease in accuracy from 93.03% to 91.54% which shows that the optimization improved other metrics at the expense of slight accuracy reduction.
- Linear Regression (SelectPercentile): The Linear Regression model achieved a 0.49% improvement in accuracy from 92.54% to 93.03%, when using SelectPercentile which demonstrated the benefits of hyperparameter optimization and feature selection techniques.
- MLPClassifier (Wrapper Method (RFE)), (SelectPercentile): The MLPClassifier achieved 92.54% accuracy through the Wrapper Method (RFE) and SelectPercentile techniques which produced identical results.
- SVC (SelectPercentile): The SVC model achieved a major accuracy boost through optimization techniques which resulted in a 1.49% improvement from 92.04% to 93.53%.

F1 Score and Recall:

- SVC (SelectPercentile): The recall rate increased from 96.64% to 98.32% and precision also increased which indicates that the model became more effective at detecting positive instances and reducing false positives.
- SVC (SelectPercentile): The F1 score improved from 0.9349 to 0.9473 which indicates that the model achieved a better balance between precision and recall after optimization.

Computational Time:

- MLPClassifier (SelectPercentile and Wrapper Method (RFE)): The computational time decreased substantially after optimization from more than 1700 seconds to approximately 1200 seconds thus demonstrating better efficiency.

- Linear Regression (SelectKBest): Computational time reduced from 0.0382 seconds to 0.0219 seconds, indicating enhanced efficiency.

Mean Absolute Error (MAE) and Mean Squared Error (MSE):

- SVC (SelectPercentile): Showed improvement in both MAE (from 0.0796 to 0.0647) and MSE (from 0.0796 to 0.0647), indicating better predictive accuracy and error reduction.
- RandomForest Classifier (SelectPercentile): The MAE and MSE slightly increased, reflecting the trade-off between model complexity and accuracy.
- The Linear Regression (SelectKBest) model, maintained a consistent MAE of 0.269 and an MSE of 0.098, demonstrating its stability and accuracy in predictions.

R-Squared (R2): The R2 values which are the proportion of variance explained by the models increased in some cases after optimization.

- SVC (SelectPercentile): Improved R2 from 0.6704 to 0.7322, indicating a better fit and explaining more variance in the data.
- Linear Regression (SelectPercentile): The R2 value slightly decreased, showing a trade-off with other performance metrics.

Precision: The precision of the top models was high after optimization and the models were able to correctly identify positive instances without producing a high number of false positives. The SVC (SelectPercentile Feature Selection) model for instance had an increase in precision from 0.906 to 0.914.

Confusion Matrix: The confusion matrices of the optimized models showed a slight improvement in classification accuracy. For instance, the confusion matrix of the SVC (SelectPercentile Feature Selection) model improved which means that the overall classification performance improved.

Feature Selection: Feature selection is important in improving model performance because it helps in the identification of the most relevant features that can be used for prediction. The top six models selected different features before and after optimization with some features being selected as important predictors in both cases.

- Before Optimization: Features that were common among the models included 'Mother's Education, 'failures', 'G1', 'G2', 'Fjob_teacher', and schoolsup'. For example, Linear Regression (SelectKBest) and MLPClassifier (Wrapper Method (RFE)) both selected ' Mother's Education ', 'failures', 'schoolsup', and 'G1'.
- After Optimization: The consistency of certain features remained, with ' Mother's Education ', 'failures', 'G1', and 'G2' continuing to appear frequently. Additionally, new relevant features emerged, such as 'address', 'activities', and 'absences', which were selected by optimized models like RandomForest Classifier (SelectPercentile) and LinearRegression (SelectPercentile).

The repeated selection of features such as ' Mother's Education ' (mother's education), 'failures' (number of past class failures), 'G1' (first period grade), and 'G2' (second period grade) indicates their strong predictive power across different models and feature selection methods.

The modifications and optimization techniques applied, including hyperparameter tuning, cross-validation, data augmentation, and regularization, have generally improved or maintained the performance of the top models. The enhancements not only reduced computational time but also improved various performance metrics, particularly for models like SVC (SelectPercentile), which showed significant gains in accuracy, R2, recall, precision, and F1 score. The insights from feature selection emphasize the critical role of specific features in driving model performance, guiding future work in feature engineering and selection.

4.1.6 Supporting Research Findings with Previous Studies

For binary classification problems, such as predicting student performance, where classes can be linearly separated and sample sizes may impact the training and testing of widely-used data mining and machine learning methods, Random Forests and Linear Regression Analysis have demonstrated high accuracy, sensitivity, specificity, and discriminant power. In contrast, data mining classifiers such as Support Vector Machines, Neural Networks, and Classification Trees have exhibited lower sensitivity, suggesting they are not as suitable for classification problems where the class of interest is underrepresented (52) . Additionally, other research (53) has concluded that Random Forests work better with smaller datasets, while Support Vector Machines perform better with larger datasets . Another study(54) found that Support Vector Regression

(SVR) and Linear Regression methods outperformed Neural Networks in predicting graduation CGPA .

4.2 Conclusion

This thesis aimed to enhance the prediction of student performance using various machine learning models combined with feature selection methods. Through a comprehensive analysis, it was found that models such as Linear Regression, Random Forests, Neural Networks, and Support Vector Machines, when paired with effective feature selection techniques like SelectKBest, SelectPercentile, and LASSO Regression, achieved the highest accuracy and computational efficiency. Furthermore, hyperparameter tuning with GridSearchCV and cross-validation improved the model performance.

Student academic performance prediction depends on multiple factors, but our findings demonstrate that previous grade and parent education, together with the number of past class failures, absences, addresses, and activities, have the most significant impact on student achievement. The identification of these essential features enables a better understanding of the determinants of academic success while also informing specific student support strategies.

The present study contributes to the literature by providing a feasible methodology to choose and improve ML models for educational data. It indicates that feature selection enhances accuracy and efficiency to a large extent, and computational time matters when a real-time purpose is taken into consideration.

Although these findings contribute to the literature, several limitations exist. The models were only built with one database, and hence, the findings may not be generalizable. Hence, in future work, we recommend these models to be tested on various educational datasets, and we investigate other feature selection methods and sophisticated hyperparameter optimization algorithms to bring significant improvement.

In summary, we contribute to the educational data mining by demonstrating that incorporating feature selection and aggregation of machine learning models together into a predictive model provides a more accurate, more efficient, and more applicable model. These models can offer an educational institution powerful instruments to aid and optimize academic attainments.

List of Abbreviations

Abbreviations	Meaning
ANN	Artificial Neural Network
C4.5	Decision Tree (C4.5)
DSS	Decision Support System
DT	Decision Tree
EDA	Exploratory Data Analysis
F1	F1 score
KNN	K-Nearest Neighbors
LASSO	Least Absolute Shrinkage and Selection Operator
MAE	Mean Absolute Error
Medu	Mother's Education Level
ML	Machine Learning
MLP	Multi-Layer Perceptron
MSE	Mean Squared Error
NB	Naïve Bayes
PSO	Particle Swarm Optimization
R^2	R-Squared Score
RFE	Recursive Feature Elimination
RF	Random Forest
SPSO	Standard Particle Swarm Optimization
SVC	Support Vector Classification
SVM	Support Vector Machine

References

1. Feng G, Fan M. Research on learning behavior patterns from the perspective of educational data mining: Evaluation, prediction and visualization. Expert Syst Appl. 2024;237:121555.

2. Baker RS, Yacef K. The state of educational data mining in 2009: A review and future visions. J Educ Data Min. 2009;1(1):3–17.

3. Bogarín A, Romero C, Cerezo R, Sánchez-Santillán M. Clustering for improving educational process mining. In: Proceedings of the fourth international conference on learning analytics and knowledge. 2014. p. 11–5.

4. Qadir R, Meghji AF, Oad U, Kumari V. Exploring Learning Patterns: A Review of Clustering in Data-Driven Pedagogy. Int J Educ Manag Eng. 2022;12(1):1–9.

5. Waheed H, Hassan SU, Aljohani NR, Hardman J, Alelyani S, Nawaz R. Predicting academic performance of students from VLE big data using deep learning models. Comput Hum Behav. 2020;104:106189.

6. Ameen AO, Alarape MA, Adewole KS. Students' academic performance and dropout predictions: A review. Malays J Comput. 2019;4(2):278–303.

7. Emerging India Analytics. Big Data Analytics in Education: Improving Learning Outcomes and Student Success [Internet]. 2024. Available from: https://medium.com/@analyticsemergingindia/big-data-analytics-in-education-improving-learning-outcomes-and-student-success-a9d0589b7d8b

8. Mahesh B. Machine learning algorithms-a review. Int J Sci Res IJSR. 2020;9(1):381–6.

9. Uddin MF, Lee J, Rizvi S, Hamada S. Proposing enhanced feature engineering and a selection model for machine learning processes. Appl Sci. 2018;8(4):646.

10. Asogbon MG, Samuel OW, Omisore MO, Ojokoh B. A multi-class support vector machine approach for students academic performance prediction. Int J Multidiscip Curr Res. 2016;4:210–5.

11. Razaque F, Soomro N, Shaikh SA, Soomro S, Samo JA, Kumar N, et al. Using naive bayes algorithm to students' bachelor academic performances analysis. In: 2017 4th IEEE International Conference on Engineering Technologies and Applied Sciences (ICETAS). 2017. p. 1–5.

12. Hasan R, Palaniappan S, Raziff ARA, Mahmood S, Sarker KU. Student academic performance prediction by using decision tree algorithm. In: 2018 4th International Conference on Computer and Information Sciences (ICCOINS). 2018. p. 1–5.

13. Bendangnuksung, Prabu. Students performance prediction using deep neural network. Int J Appl Eng Res. 2018;13(2):1171–6.

14. Mueen A, Zafar B, Manzoor U. Modeling and predicting students' academic performance using data mining techniques. Int J Mod Educ Comput Sci. 2016;8(11):36–42.

15. Ünal F. Data Mining for Student Performance Prediction in Education. In: Data Mining – Methods, Applications and Systems. IntechOpen; 2021.

16. Helal S, Li J, Liu L, Ebrahimie E, Dawson S, Murray DJ, et al. Predicting academic performance by considering student heterogeneity. Knowl-Based Syst. 2018;161:134–46.

17. Roslan MHB, Chen CJ. Predicting students' performance in English and Mathematics using data mining techniques. Educ Inf Technol. 2023;28(2):1427–53.

18. Asif R, Merceron A, Ali SA, Haider NG. Analyzing undergraduate students' performance using educational data mining. Comput Educ. 2017;113:177–94.

19. Naser SA, Zaqout I, Ghosh MA, Atallah R, Alajrami E. Predicting student performance using artificial neural network: in the faculty of engineering and information technology. Int J Hosp Inf Technol. 2015;8(2):221–8.

20. Pallathadka H, Wenda A, Ramirez-Asís E, Asís-López M, Flores-Albornoz J, Phasinam K. Classification and prediction of student performance data using various machine learning algorithms. Mater Today Proc. 2023;80:3782–5.

21. Shah MB, Kaistha M, Gupta Y. Student performance assessment and prediction system using machine learning. In: 2019 4th International Conference on Information Systems and Computer Networks (ISCON). IEEE; 2019. p. 386–90.

22. Azizah EN, Pujianto U, Nugraha E. Comparative performance between C4.5 and Naive Bayes classifiers in predicting student academic performance in a Virtual Learning Environment. In: 2018 4th International Conference on Education and Technology (ICET). IEEE; 2018. p. 18–22.

23. Alsariera YA, Baashar Y, Alkawsi G, Mustafa A, Alkahtani AA, Ali NA. Assessment and evaluation of different machine learning algorithms for predicting student performance. Comput Intell Neurosci. 2022;1–12.

24. Masood MF, Khan A, Hussain F, Shaukat A, Zeb B, Ullah RMK. Towards the selection of best machine learning model for student performance analysis and prediction. In: 2019 6th International Conference on Soft Computing & Machine Intelligence (ISCMI). IEEE; 2019. p. 12–7.

25. Lin Y, Chen H, Xia W, Lin F, Wang Z, Liu Y. A comprehensive survey on deep learning techniques in educational data mining. ArXiv Prepr ArXiv230904761. 2023;

26. Liu Y, Fan S, Xu S, Sajjanhar A, Yeom S, Wei Y. Predicting student performance using clickstream data and machine learning. Educ Sci. 2022;13(1):17.

27. Kalita E, Alfarwan AM, El Aouifi H, Kukkar A, Hussain S, Ali T, et al. Predicting student academic performance using Bi-LSTM: a deep learning framework with SHAP-based interpretability and statistical validation. In: Frontiers in Education. Frontiers Media SA; 2025. p. 1581247.

28. Wang C, Chen J, Xie Z, Zou J. Research on education big data for student's academic performance analysis based on machine learning. In: Proceedings of the 2024 Guangdong-Hong Kong-Macao Greater Bay Area International Conference on Education Digitalization and Computer Science. 2024. p. 223–7.

29. Pinto JD, Paquette L. Deep learning for educational data science. In: Trust and inclusion in AI-mediated education: Where human learning meets learning machines. Cham: Springer Nature Switzerland; 2024. p. 111–39.

30. Romero C, Ventura S. Educational data mining and learning analytics: An updated survey. Wiley Interdiscip Rev Data Min Knowl Discov. 2020;10(3):e1355.

31. Hernández-Blanco A, Herrera-Flores B, Tomás D, Navarro-Colorado B. A systematic review of deep learning approaches to educational data mining. Complexity. 2019;2019:1306039.

32. UCI Machine Learning Repository. Student Performance Data Set [Internet]. 2019. Available from: https://archive.ics.uci.edu/ml/datasets/student+performance

33. Nguyen BH, Xue B, Zhang M. A survey on swarm intelligence approaches to feature selection in data mining. Swarm Evol Comput. 2020 May;54:100663.

34. Anuragi A, Sisodia DS, Pachori RB. Mitigating the curse of dimensionality using feature projection techniques on electroencephalography datasets: an empirical review. Artif Intell Rev. 2024;57(3):75.

35. Kennedy J, Eberhart R. Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks (ICNN). 1995. p. 1942–8.

36. Gao J, Wang Z, Jin T, Cheng J, Lei Z, Gao S. Information gain ratio-based subfeature grouping empowers particle swarm optimization for feature selection. Knowl-Based Syst. 2024;286:111380.

37. Shami TM, El-Saleh AA, Alswaitti M, Al-Tashi Q, Summakieh MA, Mirjalili S. Particle swarm optimization: A comprehensive survey. IEEE Access. 2022;10:10031–61.

38. Sethi JK, Mittal M. An efficient correlation based adaptive LASSO regression method for air quality index prediction. Earth Sci Inform. 2021;14(4):1777–86.

39. Pudjihartono N, Fadason T, Kempa-Liehr AW, O'Sullivan JM. A review of feature selection methods for machine learning-based disease risk prediction. Front Bioinforma. 2022;2:927312.

40. Tariq MA. A Study on Comparative Analysis of Feature Selection Algorithms for Students Grades Prediction. 2024;

41. Güner M. Retail data predictive analysis using machine learning models. 2020.

42. Roy K. S, Roopkanth K, Uday Teja V, Bhavana V, Priyanka J. Student career prediction using advanced machine learning techniques. Int J Eng Technol. 2018;7(2.20):26–9.

43. Bansal M, Goyal A, Choudhary A. A comparative analysis of K-nearest neighbor, genetic, support vector machine, decision tree, and long short term memory algorithms in machine learning. Decis Anal J. 2022;3:100071.

44. Zheng J, Xin D, Cheng Q, Tian M, Yang L. The Random Forest Model for Analyzing and Forecasting the US Stock Market in the Context of Smart Finance. ArXiv Prepr ArXiv240217194. 2024;

45. GeeksforGeeks. Random Forest Algorithm. 2024.

46. James G, Witten D, Hastie T, Tibshirani R, Taylor J. Linear regression. An introduction to statistical learning: With applications in python. Cham: Springer International Publishing; 2023. 69–134 p.

47. Taheri S, Mammodov M. Learning the naives Bayes classifier with optimization models. Int J Appl Math Comput Sci. 2014;23(4):787–95.

48. Zaidi A, Al Luhayb ASM. Two statistical approaches to justify the use of the logistic function in binary logistic regression. Math Probl Eng. 2023;2023(1):5525675.

49. Roy A, Chakraborty S. Support vector machine in structural reliability analysis: A review. Reliab Eng Syst Saf. 2023;233:109126.

50. Mfetoum IM, Ngoh SK, Molu RJJ, Nde Kenfack BF, Onguene R, Naoussi SRD, et al. A multilayer perceptron neural network approach for optimizing solar irradiance forecasting in Central Africa with meteorological insights. Sci Rep. 2024;14(1):3572.

51. Good Audience. Artificial Neural Networks Explained [Internet]. 2019. Available from: https://blog.goodaudience.com/artificial-neural-networks-explained436fcf36e75

52. Maroco J, Silva D, Rodrigues A, Guerreiro M, Santana I, de Mendonça A. Data mining methods in the prediction of Dementia: A real-data comparison of the accuracy, sensitivity and specificity of linear discriminant analysis, logistic regression, neural networks, support vector machines, classification trees and random forests. BMC Res Notes. 2011;4:1–14.

53. Alamri L, Almuslim R, Alotibi M, Alkadi D, Khan IU, Aslam N. Predicting student academic performance using support vector machine and random forest. In: Proceedings of the 2020 3rd International Conference on Education Technology Management. 2020. p. 100–7.

54. Obsie EY, Adem SA. Prediction of student academic performance using neural network, linear regression and support vector regression: a case study. Int J Comput Appl. 2018;180(40):39–47.

Appendices

Appendix A

Tables

Table A.1

The primary features of the data set

Feature	Explanation	Data Type	Possible Values
Gender	Indicates whether the student is male or female	Binary	Male , Female
Age	Student's age in years	Numeric	Ranges From 15 to 22
School Name	The institution the student attends	Binary	MS (Mousinho da Silveira), GP (Gabriel Pereira)
Residential Area	Type of student's home location	Binary	Urban , Rural
Parental Status	Reflects whether the parents live together or separately	Binary	Together, Apart
Mother's Education	Level of education attained by the student's mother	Numeric	Scale from 0 (none) to 4 (higher education)
Mother's Job	Occupational field of the student's mother	Nominal	Health sector, Services, Teacher, At home, Others
Father's Education	Level of education attained by the student's father	Numeric	Scale from 0 to 4
Father's Job	Occupation of the student's father	Nominal	Health sector, Services, Teacher, At home, Others
Guardian	Primary guardian responsible for the student	Nominal	Father,Mother, Otherd
Family Size	Number of individuals in the family	Binary	"LE3" (three or fewer) , "GT3" (more than three)
Family Relationship	Student's assessment of family bonding quality	Numeric	Scale from 1 (very poor) to 5 (excellent)
School Choice Reason	Main factor influencing school selection	Nominal	Proximity, Reputation, Course preference, Other
Commute Time	Time required to travel from home to school	Numeric	1: <15 mins, 2: 15–30 mins, 3: 30–60 mins, 4: >1 hour
Weekly Study Time	Total study hours per week	Numeric	1: <2 hours, 2: 2–5 hours, 3: 5–10 hours, 4: >10 hours

Past Failures	Count of previous course failures	Numeric	1–3, or 4 if more than 3
School Support	Indicates if student receives academic support from school	Binary	Yes , No
Family Support	Indicates if student receives educational help from family	Binary	Yes , No
Extra Activities	Participation in extracurricular activities	Binary	Yes , No
Private Lessons	Attendance in paid extra classes	Binary	Yes , No
Home Internet	Whether internet access is available at home	Binary	Yes , No
Nursery Attendance	If the student attended nursery school	Binary	Yes , No
Higher Education Aspiration	Whether the student plans to pursue higher education.	Binary	Yes , No
Romantic Relationship	Indicates if student is in a romantic relationship	Binary	Yes , No
Free time	Amount of free time the student has after school	Numeric	Scale from 1 (very low) to 5 (very high)
Social Activity(Go out)	Frequency of going out with friends	Numeric	Scale from 1 (very low) to 5 (very high)
Weekend Alcohol Use	Level of alcohol consumption during weekends	Numeric	Scale from 1 (very low) to 5 (very high)
Weekday Alcohol Use	Alcohol consumption on school days	Numeric	Scale from 1 to 5
Health Status	Student's self-evaluation of current health status	Numeric	Scale from 1 (very poor) to 5 (excellent)
Absenteeism	Total number of absences from school	Numeric	From 0 to 93 days
Grade1 (G1)	Academic performance in the first term	Numeric	Range from 0 to 20
Grade2 (G2)	Academic performance in the second term	Numeric	Range from 0 to 20
Final Grade (G3)	Student's final academic score	Numeric	Range from 0 to 20

Table A.2

Comparative analysis of previous studies

References	Attributes	Objective	Level	Dataset	Algorithms	Accuracy	
						Min	Max
(15)	33 Attributes : the school, student's gender, age, living address, size of the family, parental status, Mother's Education , Guardian,….	Predicting students' performance	High school students	Two datasets were collected from secondary education of two Portuguese schools	DT(J48),RF, NB	DS1 :NB (89.68%) DS2: NB (89.11%)	DS1: RF (93.22%) DS2: RF (93.67%)
(23)	Each study has different attributes such that :Demographic, Academic, Internal assessment, Family/personal, Behavioral, Communication, Psychological	Assessment and Evaluation of Different Machine Learning Algorithms for Predicting Student Performance	Above 17□years; all educational levels. Academic institutions; university; college; high school.	The data supporting this review are from previously reported studies between 2015 and 2021 and online datasets	LinR,NB,SVM,KNN,DT, ANN	LinR (76%).	ANN (98.3%)
(16)	Incorporates socio-demographic (age, gender and economic status) and academic data (attendance type and delivery mode) gathered during student enrolment, and activity data	Predicting academic performance by considering student heterogeneity	Obtained from the Australian university learning management system . LMS - Moodle.	DS1: Enrolment:4010 DS2:LMS activity: 10968 DS3: Combined : 4010	NB , SMO, J48, JRip	DS1:NB (70.05%) DS2: NB (75.12%) DS3: NB (78.44%)	DS1: JRip (78%) DS2:SMO (80.7%) DS3:JRip (83.%)
(17)	Past academic performance, demographics such as	Predicting students' performance in	Secondary school	DS1: 159	DT,NN,SVM, NB	DS1:NN (71.0%)	DS1:DT (87.1%)

	parents' educational level and gender as well as psychological factors such as diversity and self-criticism	English and Mathematics using data mining techniques		DS2:159		DS2: SVM (71.0%)	DS2: NB (83.9%)
(24)	DS1: 17 attributes including "Gender", "Nationality", "Stage ID", "Grade ID", "Place of birth", "Section ID"… DS2: 33 attributes including "School", "Sex", "Age", "Address", "Family Size", "Parent Status"…	Select of Best ML Model for Student Performance	High school students	DS1:480 DS2: 395 20 studies (2012-2019)	RF,DT,NN,NB,LR,LogR,AdaB,KNN,QDA,MLP, SVM	DS1: QDA (91.40%) DS2: NB (80.86%)	DS1: RF (99.50%) DS2:DT (97.03%)
(18)	The marks for all the courses that are taught in the 4- years bachelor degree of a public sector engineering university in Pakistan	Predicting Students' academic achievement at the end of a four-year study programme	Undergraduate students	210	DT, 1-NN, NB, NN, RF	DT (60.58%)	NB (83.65%)
(19)	High School score,Math I,Math II,Electrical Circuit I,Electronics I,Number of credits passed,CGPA of freshman Year,Zone of High school attended,Type of High School,Gender	Predicting students' performance	Faculty of Engineering in Al-Azhar University	Do not report dataset count	ANN	-	84.6%
(20)	33 Attributes : personal and academic information about students such that : school,age,sex, address , famsize , mother's education , father's education ,…	Predicting students' performance	University of Minho, Portugal	649	SVM,NB, C4.5,ID3	ID3 (60%)	SVM (85%)

(21)	33 Attributes : the school, student's gender, age, living address, size of the family, parental status,….	Predicting students' performance	High school students in Portugal	649	DT,SVM,RF,LogReg,GB,XGBoost,AdaBoost,ANN,RNN	LogReg (59.8%)	GBoost (93.80%)
(5)	Students' demographics, clickstream events and assessment performance	Analyse student performance prediction, Pass-fail, withdrawn-pass, distinction-fail, distinction-pass	Open University Learning Analytics (OULA) provided by Open University	32,593	SVM, ANN, LR	SVM (78.08 %)	ANN (93.23%)
(22)	7 Attributes : region, education, credit, disability, web, klik, result	Identifying academic performance in a Virtual Learning Environment	Open University England	17000	NB,C4.5 Tree	C4.5 (63,6 %)	NB (63,8 %)

Table A.3

Descriptive Statistics of Key Features

Feature	count	mean	std	min	25%	50%	75%	max
Gender	1001.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0
Age	1001.0	17.0	1.0	15.0	16.0	17.0	18.0	21.0
Residential Area	1001.0	1.0	0.0	0.0	0.0	1.0	1.0	1.0
Family Size	1001.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0
Mother's Education	1001.0	3.0	1.0	1.0	2.0	3.0	4.0	4.0
Father's Education	1001.0	2.0	1.0	1.0	2.0	2.0	3.0	4.0
Commute Time	1001.0	2.0	1.0	1.0	1.0	1.0	2.0	4.0
Weekly Study Time	1001.0	2.0	1.0	1.0	1.0	2.0	2.0	4.0
Past Failures	1001.0	0.0	1.0	0.0	0.0	0.0	0.0	3.0
School Support	1001.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
Family Support	1001.0	1.0	0.0	0.0	0.0	1.0	1.0	1.0
Guardian	1001.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
Extra Activities	1001.0	0.0	1.0	0.0	0.0	0.0	1.0	1.0
Nursery Attendance	1001.0	1.0	0.0	0.0	1.0	1.0	1.0	1.0
Higher Education Aspiration	1001.0	1.0	0.0	0.0	1.0	1.0	1.0	1.0
Home Internet	1001.0	1.0	0.0	0.0	1.0	1.0	1.0	1.0
Romantic Relationship	1001.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0
Family Relationship	1001.0	4.0	1.0	1.0	4.0	4.0	5.0	5.0
Free time	1001.0	3.0	1.0	1.0	3.0	3.0	4.0	5.0
Social Activity(Go out)	1001.0	3.0	1.0	1.0	2.0	3.0	4.0	5.0
Weekday Alcohol Use	1001.0	1.0	1.0	1.0	1.0	1.0	2.0	5.0
Weekend Alcohol Use	1001.0	2.0	1.0	1.0	1.0	2.0	3.0	5.0
Health Status	1001.0	4.0	1.0	1.0	3.0	4.0	5.0	5.0
Absenteeism	1001.0	4.0	5.0	0.0	0.0	2.0	6.0	40.0
Grade1 (G1)	1001.0	11.0	3.0	3.0	9.0	11.0	13.0	19.0
Grade2 (G2)	1001.0	11.0	3.0	4.0	9.0	11.0	13.0	19.0

Table A.4

Model performance metrics

Category	Method	Indices	Time	Accuracy	MAE	MSE	R2	Recall	Precision	F1	Confusion _Matrix
RandomForest Classifier	PSO Feature Selection	[0 1 3 5 7 9 11 14 16 18 19 23 29 31 34 37 38 40 42 43]	361.4638	0.646766	0.353234	0.353234	-0.46249	0.848739	0.640006	0.739927	[[29 53] [18 101]]
RandomForest Classifier	Wrapper Method (RFE)	[1 4 8 18 19 21 24 25 27 28]	27.39932	0.910448	0.089552	0.089552	0.629227	0.966387	0.891473	0.927419	[[68 14] [4 115]]
RandomForest Classifier	SelectKBest Feature Selection	[4 5 8 9 20 24 25 29 34 38]	0.710936	0.910448	0.089552	0.089552	0.629227	0.966387	0.891473	0.927419	[[68 14] [4 115]]
RandomForest Classifier	SelectPercentile Feature Selection	[4 7 8 14 24 25 27 37 40]	1.255736	0.930348	0.069652	0.069652	0.711621	0.97479	0.913386	0.943089	[[71 11] [3 116]]
RandomForest Classifier	LASSO Regression Feature Selection	[8 11 13 14 24 25 29]	1.072673	0.890547	0.109453	0.109453	0.546833	0.941176	0.88189	0.910569	[[67 15] [7 112]]
DecisionTree Classifier	PSO Feature Selection	[2 5 7 9 10 11 12 16 17 22 24 25 26 28 34 37 40 42]	26.34085	0.860697	0.139303	0.139303	0.423242	0.94958	0.866432	0.889764	[[60 22] [6 113]]
DecisionTree Classifier	Wrapper Method (RFE)	[0 1 17 18 19 22 23 24 25 28]	0.502014	0.870647	0.129353	0.129353	0.464439	0.941176	0.854962	0.896	[[63 19] [7 112]]
DecisionTree Classifier	SelectKBest Feature Selection	[4 5 8 9 20 24 25 29 34 38]	0.039985	0.850746	0.149254	0.149254	0.382046	0.915966	0.844961	0.879032	[[62 20] [10 109]]
DecisionTree Classifier	SelectPercentile Feature Selection	[0 8 11 18 20 24 25 26 32]	0.657924	0.820896	0.179104	0.179104	0.258455	0.907563	0.81203	0.857143	[[57 25] [11 108]]

Category	Method	Indices	Time	Accuracy	MAE	MSE	R2	Recall	Precision	F1	Confusion _Matrix
DecisionTree Classifier	LASSO Regression Feature Selection	[8 11 13 14 24 25 29]	0.505332	0.890547	0.109453	0.109453	0.546833	0.932773	0.888	0.909836	[[68 14] [8 111]]
Naïve Bayes	PSO Feature Selection	[5 6 8 14 21 30 39 40 41 43]	2.872444	0.741294	0.258706	0.258706	-0.07112	0.865546	0.741007	0.79845	[[46 36] [16 103]]
Naïve Bayes	SelectKBest Feature Selection	[4 5 8 9 20 24 25 29 34 38]	0.014349	0.875622	0.124378	0.124378	0.485038	0.907563	0.885246	0.896266	[[68 14] [11 108]]
Naïve Bayes	SelectPercentile Feature Selection	[4 8 14 24 25 28 29 34 38]	0.149904	0.880597	0.119403	0.119403	0.505636	0.915966	0.886179	0.900826	[[68 14] [10 109]]
Naïve Bayes	LASSO Regression Feature Selection	[8 11 13 14 24 25 29]	0.138831	0.900498	0.099502	0.099502	0.58803	0.89132	0.901947	0.895725	[[69 13] [7 112]]
SVC	PSO Feature Selection	[3 5 7 9 10 11 12 14 20 23 28 29 31 32 33 38 39 41 43]	26.57689	0.661692	0.338308	0.338308	-0.4007	0.915966	0.652695	0.762238	[[24 58] [10 109]]
SVC	Wrapper Method (RFE)	[24 25 27 28]	3.697233	0.915423	0.084577	0.084577	0.649826	0.97479	0.892308	0.931727	[[68 14] [3 116]]
SVC	SelectKBest Feature Selection	[4 5 8 9 20 24 25 29 34 38]	0.045522	0.910448	0.089552	0.089552	0.629227	0.966387	0.891473	0.927419	[[68 14] [4 115]]
SVC	SelectPercentile Feature Selection	[1 2 4 8 10 24 25 32 43]	0.195425	0.920398	0.079602	0.079602	0.670424	0.966387	0.905512	0.934959	[[70 12] [4 115]]
SVC	LASSO Regression Feature Selection	[8 11 13 14 24 25 29]	0.202043	0.905473	0.094527	0.094527	0.608629	0.957983	0.890625	0.923077	[[68 14] [5 114]]

Category	Method	Indices	Time	Accuracy	MAE	MSE	R2	Recall	Precision	F1	Confusion _Matrix
LogisticRegression	PSO Feature Selection	[0 2 4 5 10 11 13 14 18 21 22 25 26 27 35 38 40 41]	23.27806	0.900498	0.099502	0.099502	0.58803	0.966387	0.877863	0.92	[[66 16] [4 115]]
LogisticRegression	Wrapper Method (RFE)	[4 6 8 13 24 25 27 28 29 35]	0.340429	0.910448	0.089552	0.089552	0.629227	0.97479	0.885496	0.928	[[67 15] [3 116]]
LogisticRegression	SelectKBest Feature Selection	[4 5 8 9 20 24 25 29 34 38]	0.022921	0.915423	0.084577	0.084577	0.649826	0.983193	0.886364	0.932271	[[67 15] [2 117]]
LogisticRegression	SelectPercentile Feature Selection	[0 5 8 12 13 14 19 24 25]	0.206763	0.910448	0.089552	0.089552	0.629227	0.97479	0.885496	0.928	[[67 15] [3 116]]
LogisticRegression	LASSO Regression Feature Selection	[8 11 13 14 24 25 29]	0.196395	0.915423	0.084577	0.084577	0.649826	0.983193	0.886364	0.932271	[[67 15] [2 117]]
MLPClassifier	PSO Feature Selection	[0 4 5 11 12 16 17 18 19 31 35 42]	1127.119	0.562189	0.437811	0.437811	-0.81267	0.714286	0.611511	0.658915	[[28 54] [34 85]]
MLPClassifier	Wrapper Method (RFE)	[8 10 15 24 25 26 34 38 40 43]	1798.751	0.925373	0.074627	0.074627	0.691023	0.97479	0.90625	0.939271	[[70 12] [3 116]]
MLPClassifier	SelectKBest Feature Selection	[4 5 8 9 20 24 25 29 34 38]	1.021082	0.915423	0.084577	0.084577	0.649826	0.983193	0.886364	0.932271	[[67 15] [2 117]]
MLPClassifier	SelectPercentile Feature Selection	[4 8 9 12 20 24 25 28 37]	1.448239	0.925373	0.074627	0.074627	0.691023	0.966387	0.912698	0.938776	[[71 11] [4 115]]
MLPClassifier	LASSO Regression Feature Selection	[8 11 13 14 24 25 29]	1.347423	0.910448	0.089552	0.089552	0.629227	0.97479	0.885496	0.928	[[67 15] [3 116]]

Category	Method	Indices	Time	Accuracy	MAE	MSE	R2	Recall	Precision	F1	Confusion _Matrix
LinearRegression	PSO Feature Selection	[1 6 9 11 14 16 17 22 24 26 27 29 30 33 34 35 36 41 43]	11.31164	0.860697	0.289467	0.117505	0.513496	0.932773	0.847328	0.888	[[62 20] [8 111]]
LinearRegression	Wrapper Method (RFE)	[8 13 14 24 25 27 28 29 32 33]	0.107759	0.905473	0.263349	0.095555	0.604376	0.957983	0.890625	0.923077	[[68 14] [5 114]]
LinearRegression	SelectKBest Feature Selection	[4 5 8 9 20 24 25 29 34 38]	0.038208	0.935323	0.269019	0.098197	0.593437	0.97479	0.920635	0.946939	[[72 10] [3 116]]
LinearRegression	SelectPercentile Feature Selection	[8 14 16 18 19 24 25 28 30]	0.3761	0.925373	0.274192	0.100048	0.585773	0.97479	0.90625	0.939271	[[70 12] [3 116]]
LinearRegression	LASSO Regression Feature Selection	[8 11 13 14 24 25 29]	0.308849	0.900498	0.264471	0.097169	0.597693	0.966387	0.877863	0.92	[[66 16] [4 115]]

Table A.5

Feature indices and names

Index	Feature Name
0	'sex'
1	'age'
2	'address'
3	'famsize'
4	' Mother's Education '
5	' father's education '
6	'traveltime'
7	'studytime'
8	'failures'
9	'schoolsup'
10	'famsup'
11	'paid'
12	'activities'
13	'nursery'
14	'higher'
15	'internet'
16	'romantic'
17	'famrel'
18	'freetime'
19	'goout'
20	'Dalc'
21	'Walc'
22	'health'
23	'absences'
24	'G1'
25	'G2'
26	'traveltime_log'
27	'absences_log'
28	'std_abs'

29	'school_MS'
30	'Pstatus_T'
31	'Mjob_health'
32	'Mjob_other'
33	'Mjob_services'
34	'Mjob_teacher'
35	'Fjob_health'
36	'Fjob_other'
37	'Fjob_services'
38	'Fjob_teacher'
39	'reason_home'
40	'reason_other'
41	'reason_reputation'
42	'guardian_mother'
43	'guardian_other'

Appendix B

Figures

Figure B.1

Accuracy comparison across machine learning models and feature selection methods

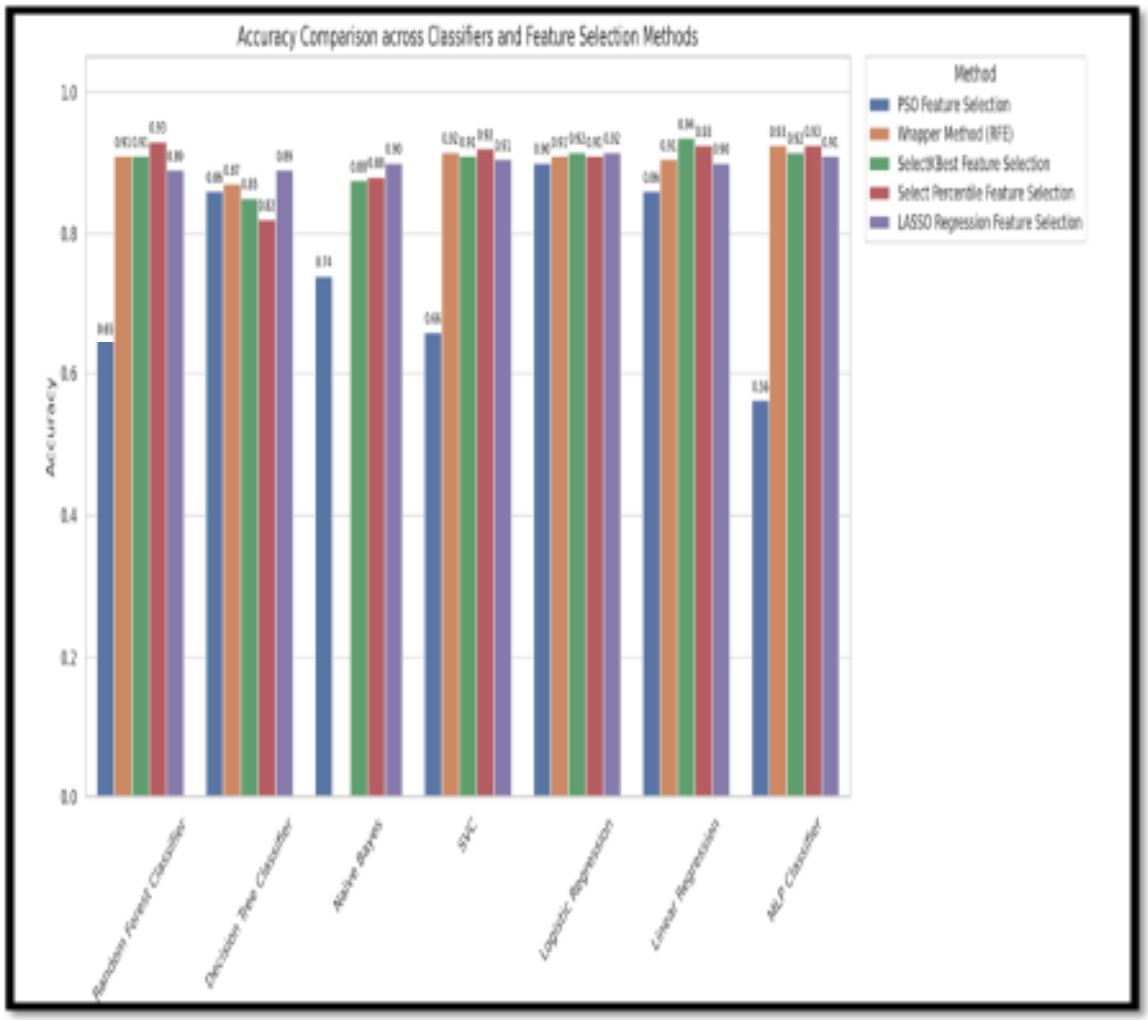


Figure B.2

F1 score comparison across machine learning models and feature selection methods

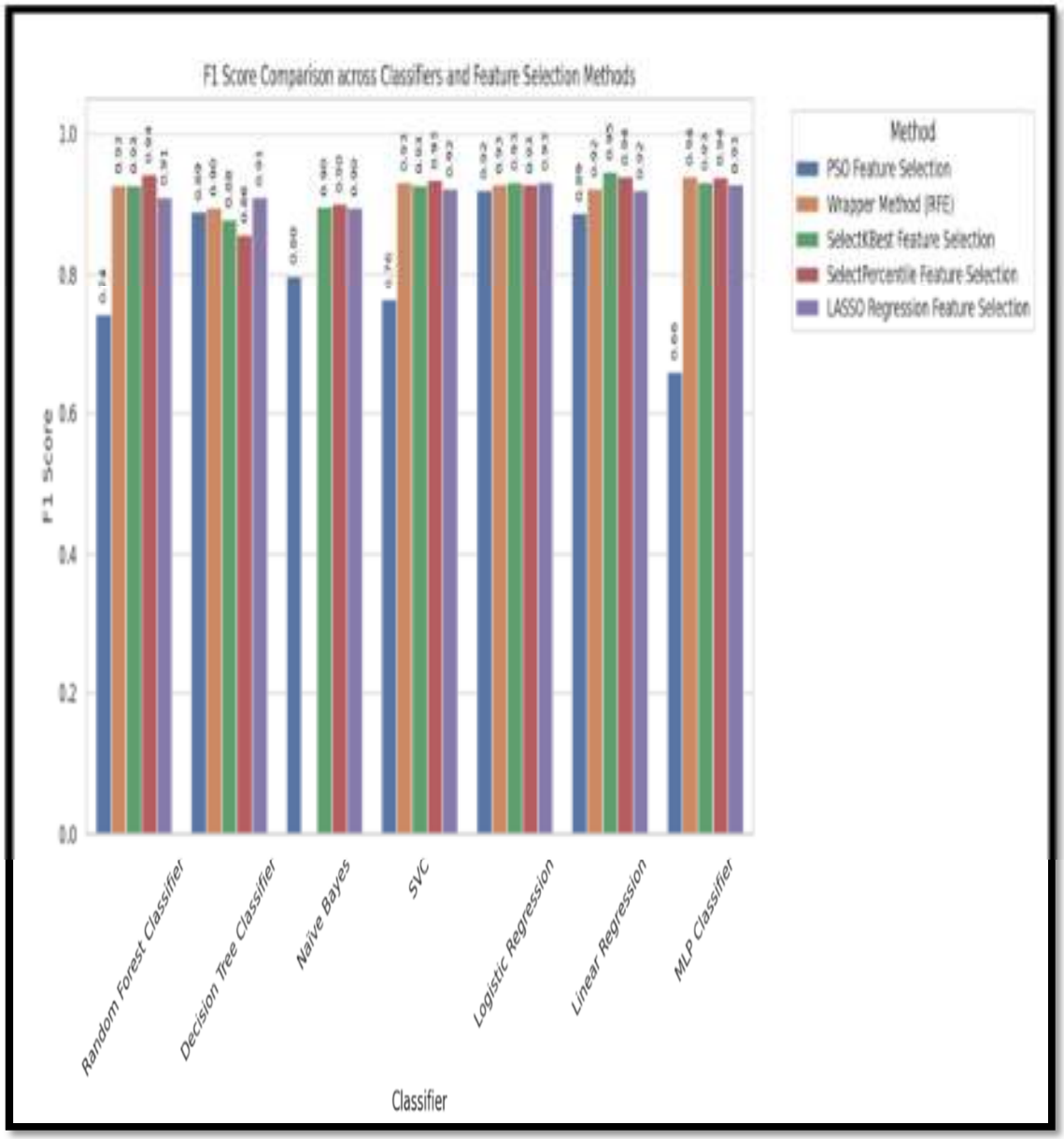


Figure B.3

Computational time comparison across machine learning models and feature selection methods

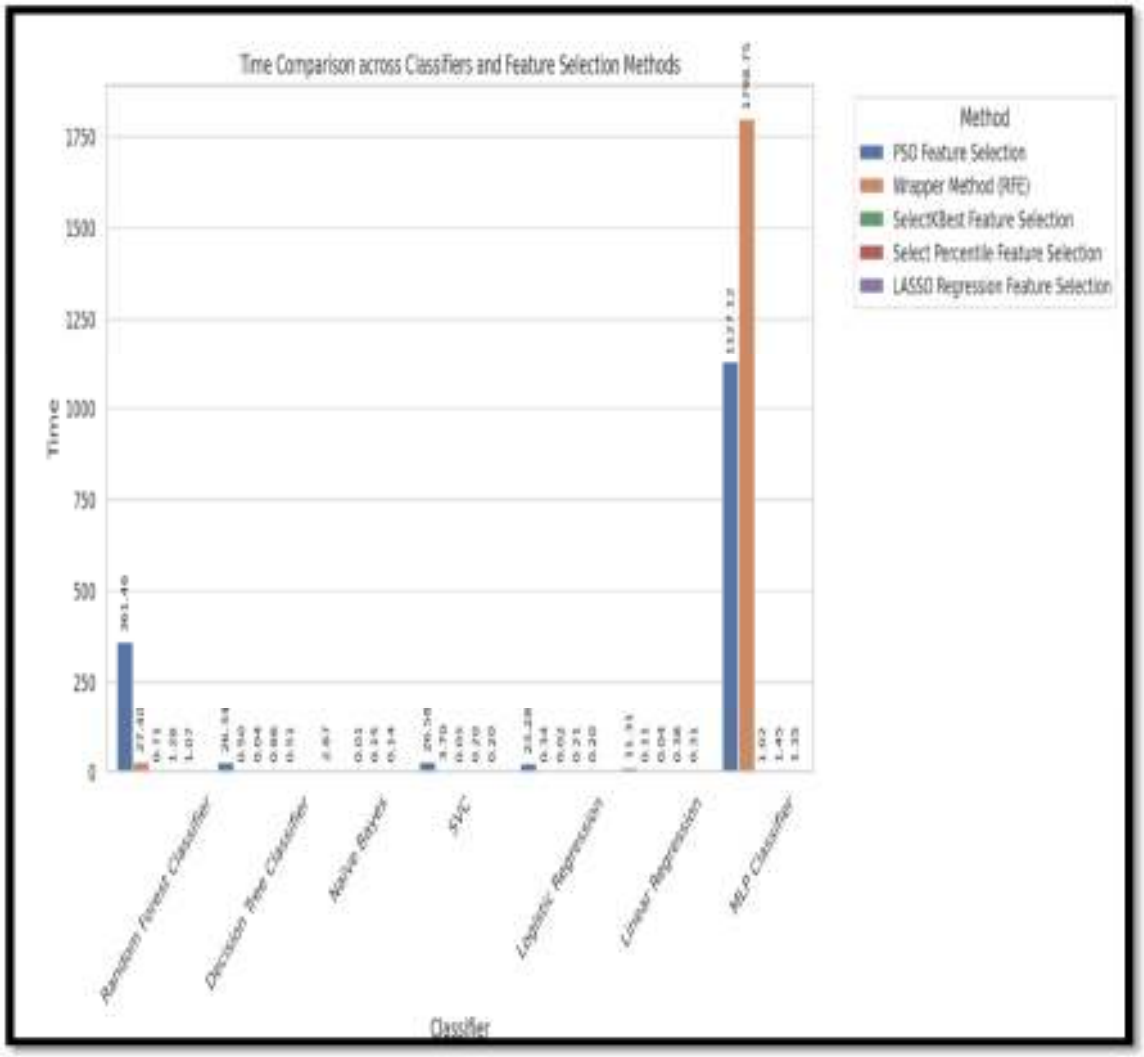


Figure B.4

Recall comparison across machine learning models and feature selection methods

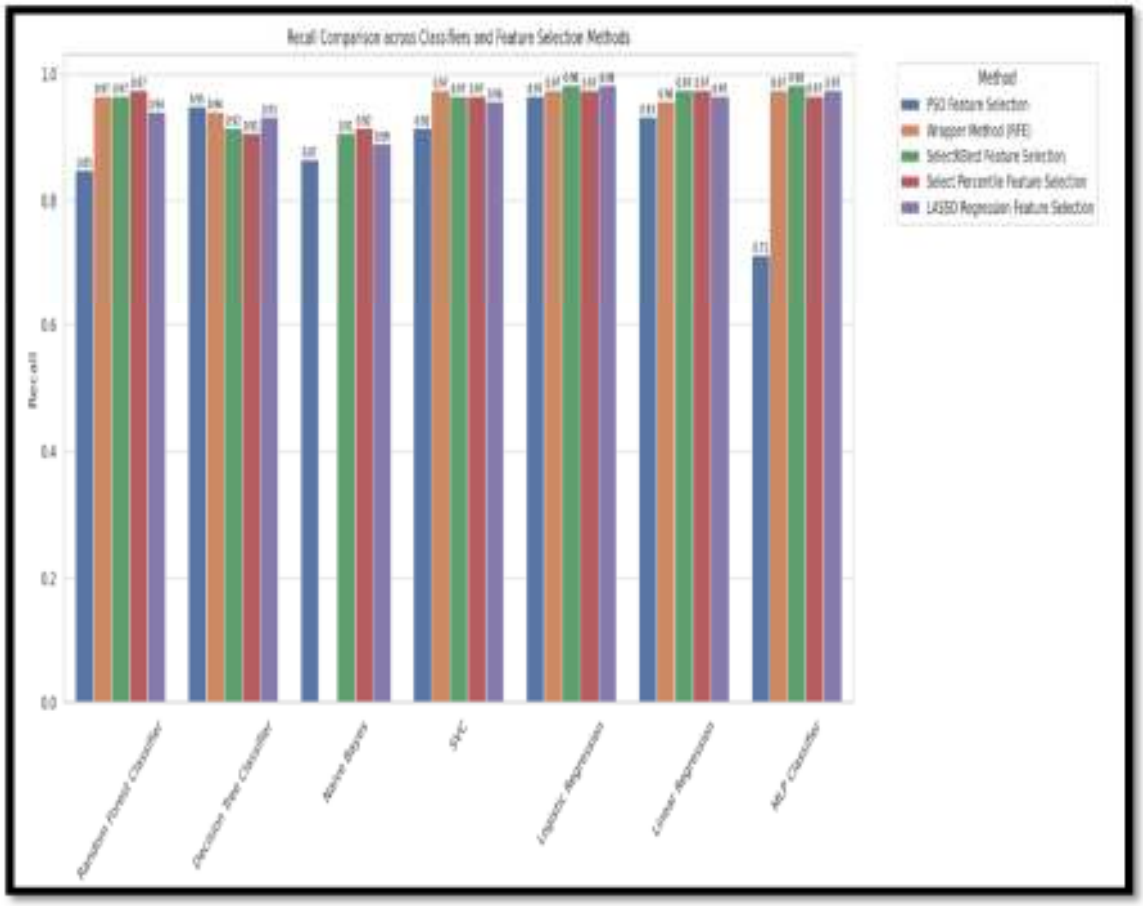
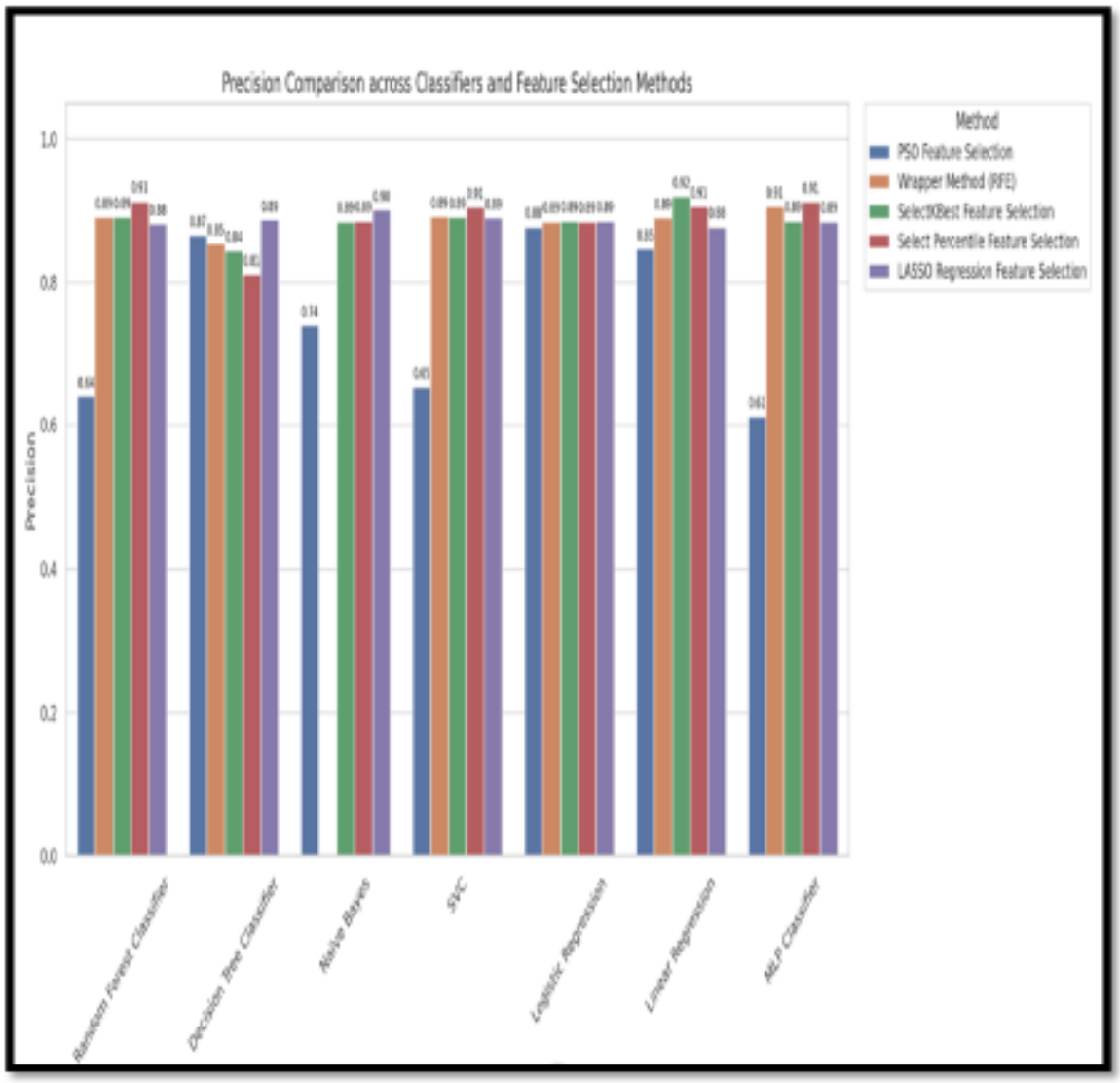


Figure B.5

Presision comparison across machine learning models and feature selection methods



جامعة النجاح الوطنية
كلية الدراسات العليا

خوارزميات التعلم الآلي لتوقع الأداء الأكاديمي للطلاب في التنقيب عن البيانات التعليمية

إعداد
فاطمة نجوان فواز سالمية

إشراف
د. أحمد عواد
د. عماد النتشة

قدمت هذه الرسالة استكمالا لمتطلبات الحصول على درجة الماجستير في الذكاء الاصطناعي،
من كلية الدراسات العليا، في جامعة النجاح الوطنية، نابلس – فلسطين.

2025

خوارزميات التعلم الآلي لتوقع الأداء الأكاديمي للطلاب في التنقيب عن البيانات التعليمية

إعداد

فاطمة نجوان فواز سالمية

إشراف

د. أحمد عواد

د. عماد النتشة

الملخص

يعد التنبؤ بأداء الطلبة من أهم مجالات التنقيب في البيانات التعليمية، نظراً لما يتيحه من إمكانيات للكشف المبكر، والتدخل، واتخاذ القرارات الأكاديمية المستنيرة. يهدف هذا البحث إلى تحسين دقة التنبؤ بالأداء الأكاديمي للطلبة باستخدام سبعة نماذج من خوارزميات التعلم الآلي: شجرة القرار (Decision Tree)، الغابة العشوائية (Random Forest)، الانحدار الخطي (Linear Regression)، الشبكات العصبية (Neural Network)، آلات الدعم الناقل (Support Vector Machines - SVM)، الانحدار اللوجستي (Logistic Regression)، و خوارزمية نايف بايز (Naive Bayes)، إضافة إلى خمس تقنيات لاختيار الميزات وهي: تحسين سرب الجسيمات (Particle Swarm Optimization - PSO)، لاسو (Lasso)، طريقة الغلاف ((Wrapper Method، SelectKBest و SelectPercentile.

يبحث هذا البحث في كيفية ارتباط مخرجات الطلبة بعوامل مثل الخبرات التعليمية السابقة، مستوى تعليم الوالدين، الإخفاقات الدراسية السابقة، الحضور، مكان السكن، والمشاركة في الأنشطة اللامنهجية. أظهرت النتائج أن نموذج الانحدار الخطي (Linear Regression) مع تقنية SelectKBest قد حقق أعلى دقة بلغت 93.5%. كما جرى تحسين أداء النموذج من خلال ضبط معاملات الضبط (GridSearchCV) واستخدام أسلوب التحقق المتقاطع (k-fold cross-validation)، مما أسهم في زيادة دقة التنبؤ وتعزيز متانة النماذج. وتبرز هذه النتائج الدور المحوري لاختيار الميزات في رفع كفاءة النماذج، كما تقدّم إرشادات عملية للمؤسسات التعليمية الراغبة في تطبيق تقنيات التحليلات التنبؤية لتعزيز نجاح الطلبة.

الكلمات المفتاحية: التنقيب في البيانات التعليمية، نماذج التعلم الآلي، التنبؤ بأداء الطلبة، اختيار الميزات.