**Content**

1. [Abstract](#Abstract_1)
2. [Introduction](#Introduction_2)
3. [Literature Review](#Literature_Review_3)
4. [Methodology](#Methodology_4)
   1. [Data Collection and Dataset Overview](#Data_Collection_and_Dataset_Overview_4_1)
   2. [Data Preprocessing](#Data_Preprocessing_4_2)
   3. [Model Selection and Training](#Model_Selection_and_Training_4_3)
   4. [Evaluation Metrics](#Evaluation_Metrics_4_5)
   5. [Expandable AI (LIME)](#Expandable_AI_LIME_4_5)
   6. Introduce a Web-Application
5. [Results](#Results_5)
   1. [Data Preprocessing Result](#Data_preprocessing_result_5_1)
   2. [Performance Metrics](#Performance_Metrics_5_2)
   3. [ROC AUC Analysis](#ROC_AUC_Analysis_5_3)
   4. [Confusion Matrix Analysis](#Confusion_Matrix_Analysis_5_4)
   5. [Expandable AI (LIME) Result](#Expandable_AI_LIME_Result_5_5)
   6. The Web-Application
6. [Discussion of Results](#Discuss_the_result6)
7. [Conclusion of Results](#Conclusion_of_Results7)
8. [List of Tables and Figures](#List_of_Tables_and_Figures8)
9. [References](#References_9)

# **1.** **Abstract**

In recent years, machine learning (ML) techniques to predict and evaluate student performance has gained a strong significant attention in the field of education. This cloud be for higher studies or for kinder garden. It has been recognized by many researchers that students struggle with their academic when they face a curricula transfer. This study aims to explore the potential of machine learning in predicting student levels and assessing academic outcomes in next higher grade/year. The dataset “Student Level Prediction in UAE” [[1]](#Ref1), analysing a range of student data, including demographic, behavioural, and academic performance indicators, various ML models such as Decision Trees, Support Vector Machines (SVM), Artificial Neural Networks (ANN), and Ensemble Methods are employed to predict student success and identify at-risk learners early. The study highlights the advantages of using data-driven approaches in educational institutions to enhance personalized learning experiences, improve retention rates, and inform targeted interventions. Results from extensive experiments demonstrate the superiority of ensemble-based models in achieving higher accuracy, stability, and interpretability compared to traditional approaches. Furthermore, the study emphasizes the importance of integrating machine learning models into the educational framework, as they offer valuable insights for curriculum design, teaching methodologies, and student support services. In addition, this study also focusses on LIME (Local Interpretable Model-agnostic Explanations) as well, it takes 5 best and most used models to study the LIME and understand how these models predict the value, and which features effect the result and how much.

Ultimately, this research contributes to the growing body of literature on the application of machine learning in education, providing a foundation for future research aimed at further enhancing academic performance prediction and student evaluation systems.

# **2.** **Introduction**

The rapid advancements in technology over the past decade have revolutionized various fields, and education is no exception. As educational institutions continue to evolve and expand, there is a growing need for innovative approaches to support and enhance student learning. Traditionally, academic success has been evaluated based on grades, test scores, and teacher assessments. However, these methods fail to capture the full spectrum of factors influencing a student’s academic journey, such as behavioural patterns, personal circumstances, previous and current curricula and engagement with learning resources. This has led to a growing interest in leveraging data and machine learning (ML) algorithms to predict student performance and develop more personalized, data-driven educational strategies.

Machine learning, a subset of artificial intelligence, has proven to be highly effective in extracting patterns from large datasets, making it an ideal tool for analysing student data. The ability to predict academic outcomes before they manifest can enable educators to implement timely interventions, personalize learning experiences, and identify at-risk students long before they fail or disengage. By utilizing data from various sources, including student demographics, attendance records, participation in online learning environments, and prior academic performance, machine learning algorithms can provide valuable insights into student success and failure factors. These insights are crucial for educational institutions aiming to improve retention rates, optimize teaching methods, and tailor curriculum designs to better meet student needs.

Despite the promise of machine learning in education, many challenges remain in its application to student performance prediction. Traditional machine learning models such as Decision Trees, Support Vector Machines (SVM), and Neural Networks are frequently used for predicting academic outcomes. However, these models often struggle with issues such as overfitting, lack of interpretability, and inability to handle diverse datasets with complex relationships between features. As a result, many studies in the field of student performance prediction have highlighted the need for more robust and accurate models that can overcome these limitations.

Recent developments in ensemble learning methods have addressed some of these challenges. Ensemble methods, which combine multiple models to improve predictive performance, have demonstrated significant potential in various domains, including student performance prediction. By integrating the strengths of multiple machine learning algorithms, ensemble models can produce more stable, accurate, and interpretable results. Moreover, the application of graph-based methods in combination with ensemble learning offers an exciting avenue for improving prediction accuracy. Graph-based ensemble methods allow for the propagation of information through connected data points, creating a more coherent and reliable prediction model that can better reflect the dynamic nature of student performance.

This study aims to explore the potential of using machine learning, particularly ensemble and graph-based methods, for predicting and evaluating student performance. By developing and evaluating multiple ML models on diverse student data, this research seeks to determine the most effective approach for predicting student success, identifying at-risk students, and supporting tailored interventions. The results of this research will provide a deeper understanding of how data-driven approaches can revolutionize student evaluation, offer practical applications for educators, and contribute to the growing body of knowledge on the role of artificial intelligence in education. Not just focused on the outcomes of each model, but this study also insight about the expandable AI XAI and give a brief introduction to LIME technology. (Local Interpretable Model-agnostic Explanations), and how it works to explain the prediction of machine learning models.

The remainder of this paper is organized as follows: Section 2 reviews the existing literature on machine learning applications in student performance prediction, highlighting the strengths and limitations of current approaches. Section 3 describes the research methodology, including the datasets used, the machine learning models implemented, and the evaluation metrics employed. Section 4 presents the results of the experiments and discusses the findings. Finally, Section 5 concludes with recommendations for future research and the practical implications of the study’s findings for educational institutions.

In addition to the machine learning models developed for student performance prediction, this study also introduces a web application designed to provide an interactive and user-friendly interface for students and educators. The web application allows users to input various student-related data, such as demographics, academic performance, and curriculum details, to receive predictions regarding the student's future academic success. By integrating the machine learning model into the web app, this tool aims to bring data-driven insights directly to students, helping them understand their academic trajectory and potential areas for improvement. This integration not only enhances the practical application of the study but also empowers students and educators with actionable insights to support personalized learning strategies and timely interventions.

# **3.** **Literature Review**

The use of machine learning (ML) in predicting student performance has grown significantly in recent years, driven by the vast amounts of educational data generated across institutions. As educational systems evolve, it becomes increasingly important to develop methods that not only predict academic success but also identify at-risk students early and provide appropriate interventions.

In multicultural settings, such as the UAE, where multiple education curriculums exist, machine learning has been applied to address challenges in student levelling. Ghareeb et al. [[2]](#Ref2) propose a framework that uses ML algorithms to help place students in the right academic year group. This is crucial as curriculums vary significantly in terms of assessment techniques, exam boards, and academic year schedules. By implementing machine learning classifiers such as Random Forest and Artificial Neural Networks, their approach helps ease the transition for students switching between curriculums, ensuring smoother integration and better tracking of their academic progress.

Similar concerns regarding student performance prediction are tackled by Qureshi and Lokhande [[3]](#Ref3), who focus on the use of Educational Data Mining (EDM) to predict academic outcomes. The paper evaluates six classification algorithms (including Random Forest, Decision Trees, and Support Vector Machines) and identifies how factors such as academic performance, personal traits, and family background influence a student's academic results. Their work underscores the importance of using diverse datasets to understand the various factors impacting student success.

Ahmed [[4]](#Ref4) builds on this by exploring the predictive power of ML models in higher education, specifically in e-learning environments. Through techniques such as K-means clustering and Support Vector Machines, this study highlights how machine learning can be used to predict student success based on their interaction with learning platforms. By identifying the features that contribute to success, the study emphasizes how these models can improve student outcomes and institutional rankings.

In a similar vein, Chandra and Kumar [[5]](#Ref5) investigate the role of ML in student placement prediction, looking at factors like technical and communication skills, as well as academic scores. They use various preprocessing and visualization techniques to better understand how these factors contribute to students' chances of securing job placements after graduation. This research highlights how academic performance influences career outcomes and how ML can aid in predicting these opportunities, ultimately enhancing job placement strategies.

Lagrazon et al. [[6]](#Ref6) explore the application of ML in predicting success in licensure examinations for Electronics Engineering graduates. They demonstrate how ensemble models—by combining predictions from multiple algorithms can enhance accuracy in forecasting licensure exam results. Their study highlights the potential of machine learning to improve educational strategies and curriculum design by providing actionable insights into students’ academic progress.

Rimpy et al. [[7]](#Ref7) provide a comprehensive review of EDM techniques used for performance prediction, noting how machine learning helps identify patterns in educational data. They stress the importance of early identification of weak students, enabling schools to intervene before poor performance impacts their academic journey. Their research serves as a reminder of the power of data mining in transforming educational systems by providing insights that can inform decision-making at various levels.

In higher education, machine learning has also been employed to predict performance based on both academic and non-academic factors. Bird [[8]](#Ref8) discusses the promises and challenges of predictive analytics in improving student success. While not providing specific data, it highlights how ML can inform decisions about student support and academic strategies. Similarly, Issah et al. [[9]](#Ref9) conduct a systematic review of various ML methods used to determine the factors influencing student performance. Their review confirms that academic attributes, including grades and demographic factors, are the most influential in predicting outcomes. They also point out the gap in research on prescriptive intervention strategies, urging further exploration into how predictive models can be used to prevent poor academic performance.

The application of regression-based machine learning models in predicting student performance is also highlighted by Asthana et al. [[10].](#Ref10) Their study proposes the concept of ‘Learning Coefficients’, a measure of students’ learning potential, which can guide targeted interventions. By employing models like Random Forest, Support Vector Regression, and Artificial Neural Networks, the study reveals that linear regression achieved the highest accuracy in predicting academic success. This demonstrates how regression models can provide valuable metrics not only for prediction but also for helping students improve their performance.

Graph-based approaches to student performance prediction are explored by Mubarak et al[. [11]](#Ref11), who use Graph Convolutional Networks (GCN) to classify students based on their engagement with course materials. Their study uses a semi-supervised approach to classify students into behavioural categories such as "high engagement" and "at-risk", showcasing the power of GCNs in handling complex student interaction data. This model can help educators identify students who might need additional support based on their engagement patterns, thus providing more personalised and targeted interventions.

The review by Oppong [[12]](#Ref12) brings attention to the broader use of machine learning in predicting student performance, with an emphasis on neural networks. The study reveals that supervised learning techniques dominate the field, with neural networks yielding the best prediction accuracy. This finding highlights the importance of using appropriate algorithms for various prediction tasks, particularly when dealing with complex datasets that include multiple variables influencing student success.

Finally, the novel deep learning model introduced by Fazil et al. [[13]](#Ref13) offers an innovative approach to performance prediction by incorporating student behaviour data, such as interaction with virtual learning environments (VLEs). Their system, ASIST, combines attention mechanisms with convolutional and bidirectional LSTM networks to predict student performance. The model's ability to process various behavioural and assessment data enables it to classify students into different performance categories, providing early predictions of academic success.

**Conclusion**

In summary, the studies reviewed here illustrate the significant progress made in applying machine learning to predict and evaluate student performance. While traditional models like SVM, Decision Trees, and Neural Networks have been valuable, they are often hindered by issues such as overfitting and lack of interpretability. Recent advances in ensemble learning and graph-based techniques offer promising solutions, improving prediction accuracy and model stability. Moreover, the integration of personalized learning paths and early intervention systems shows how machine learning can support at-risk students by providing tailored educational experiences. Moving forward, future research should focus on refining these models, expanding their use in different educational contexts, and addressing the ethical challenges related to data privacy and transparency.

# **4.** **Methodology**

This section outlines the approach used in this study to predict and evaluate student performance using machine learning techniques. The process includes data collection, preprocessing, feature engineering, model selection, training, evaluation, and results analysis. The methodology was developed to ensure that all aspects of student performance are captured effectively and that the best model is selected based on performance metrics.

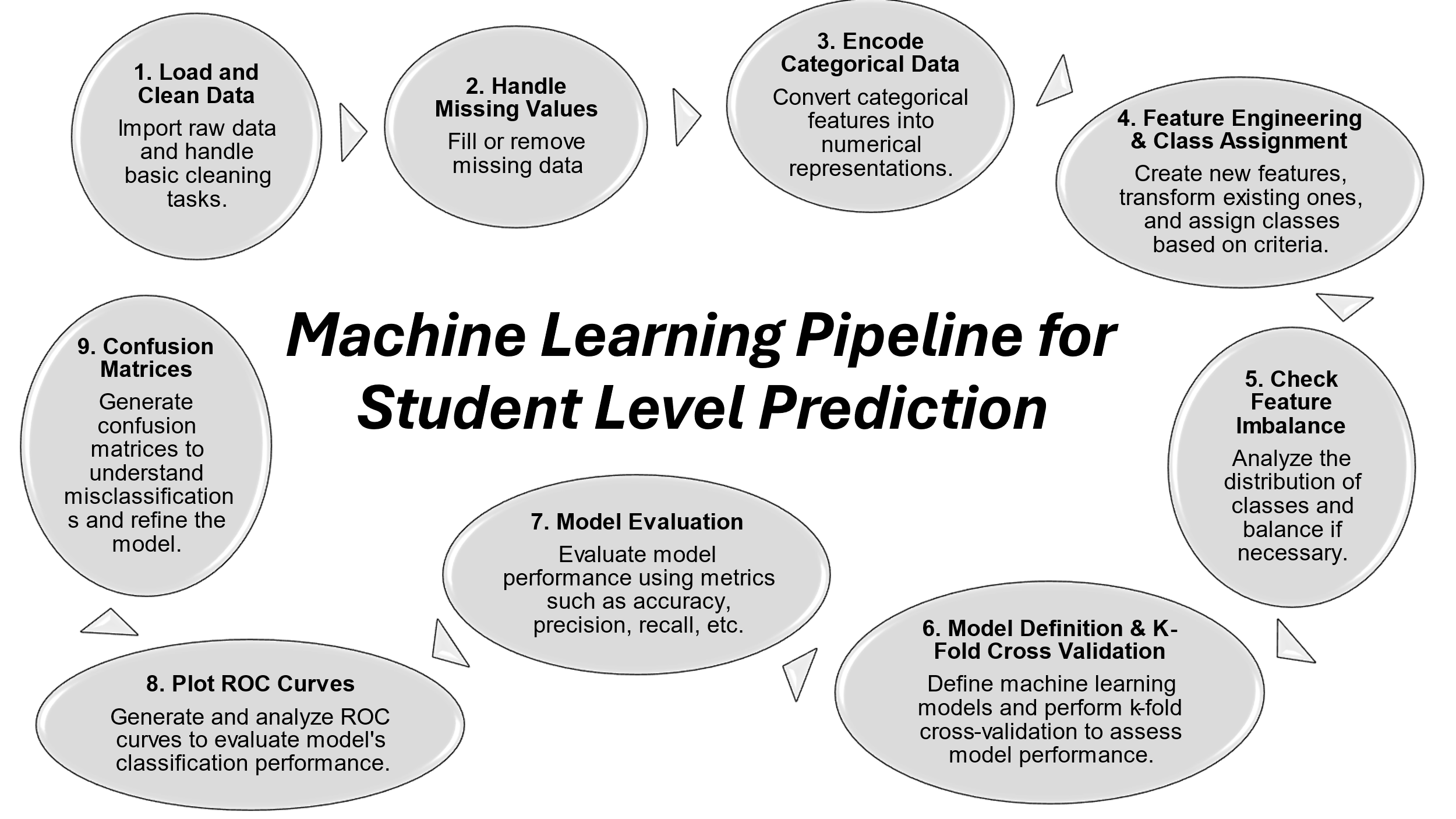


Fig 1: ML Pipeline for this project.

## **4.1** **Data Collection and Dataset Overview**

The dataset used for this study is real **time dataset collected** by Dr. Shatha Ghareeb. It consists of various student data points from an educational institution. There are 1500 rows (unique student information), and 30 columns. The data doesn’t include any personal information, and dataset deals all Artificial Intelligence Ethics, and these columns consist of 2 types of data.

**The categorical data i.e.** These data points include student demographics (e.g., age, gender), academic records (e.g., previous grades, current grades), attendance records, participation in online forums, assignment submission rates, and other behavioural indicators. This data is essential for evaluating how varied factors impact student performance.This data is very crucial for machine learning models because this provides a direct information about the student’s previous and current curricula including the school and additional information about the student.

A table of information

AI-generated content may be incorrect.

Table 1: Categorical features, content values and description.

Another data type is numerical. First the entrance marks of subjects Mathematics, Science and English are written out of 100 and then for next 2 years (2018-19 and 2019-20), all the 3 terms per year marks are written in percentage. These numerical values are further used in feature engineering this will be discuss in[**Feature Engineering part methodology section.**](#Feature_Engineering) attendance, quiz scores, etc.). The data is used to predict the students' academic success or failure, and the target variable is typically a categorical label such as "Pass" or "Fail”.

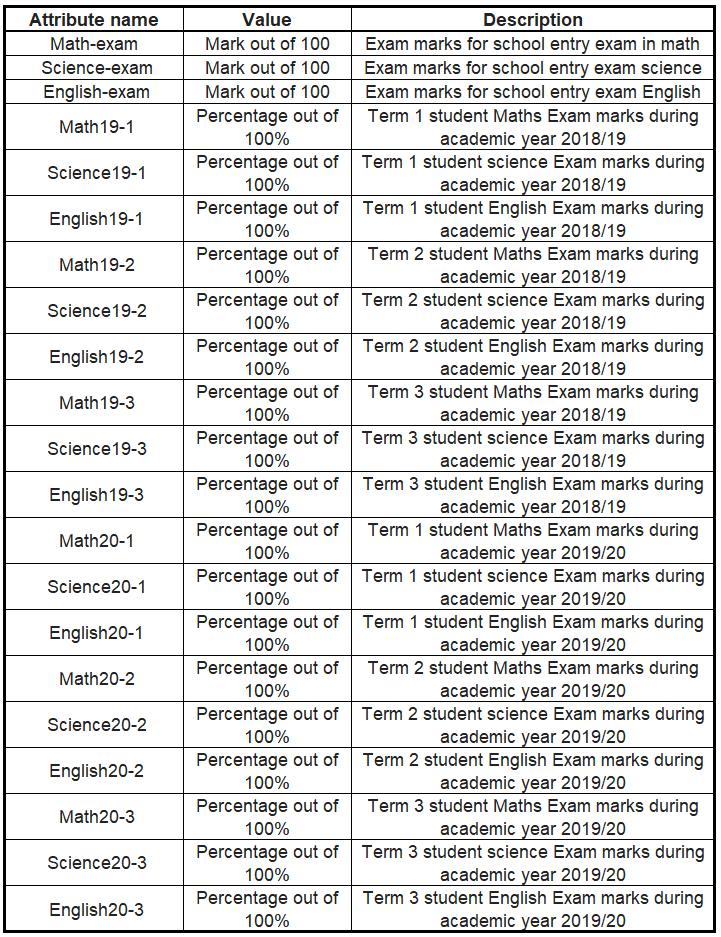


Table 2: Numerical features, content values and description.

## **4.2** **Data Preprocessing**

The raw dataset often contains missing values, irrelevant features, or unbalanced classes that can hinder the performance of machine learning models. The preprocessing steps performed are as follows:

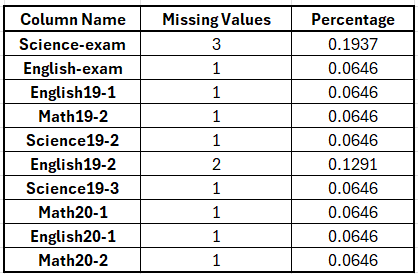
* **Cleaning the Data:** The data should clean as it contains something like, wrong spelling, extra whitespace, any symbol or anything. As been the AI study, a small python code is required to check and clear all the white space present at the end or beginning of the column name and check if there are any spelling mistakes. There are leading/trailing spaces, replace spaces with underscores, remove non-alphanumeric characters. This is common problem which is needed to be solve.
* ****At last, the cleaning function is applied on the dataset. The point to be noted here is this process is different for different dataset, features and values. For cleaning one should be check the dataset understand it and locate the errors and then apply the function to clean/remove it. This makes the dataset unique and free of all errors.

Table 3: List of Missing values in features.

* **Handling Missing Values:** Missing data is a common problem in real-world datasets. In this study, missing values were managed by employing **imputation techniques**. For numerical data, **mean imputation** was used to replace missing values with the mean of the respective column. For categorical data, the most frequent value i.e. mode value was used for imputation.

For further experiment and understanding, this study added some missing values by removing some actual values. There are no missing values in categorical features but still for the safe side it still considers handling the missing values in both categorical and numerical data.

* **Feature Engineering:** This technique is used to modify the existing features or create new feature(s) using old feature. This study also examines the students’ marks obtained by them in last 2 years and entrance exam performance marks as well. [[14]](#Ref14), [[15]](#Ref15) These are random numbers for a model and model would not work on these random numbers until there is one link between them. To create a link, the “Feature Engineering” technique is used to create a new feature from existing one. First, it takes average of all the marks obtained by each student all the subjects in 6 terms and include the entrance marks. The average has in range between 89% to 50%. Then, the categorisation is done on this average into two class, class 1 contain those students who get average greater than 80% and rest are in class 2. This is the most crucial foundation of this project and for this study because when model predict the student is in class 2 then, student or guardian got an idea that the student will need more attention if he/she chose the certain curricula.
* **Encoding Categorical Features:** Machine learning algorithms require numerical input. Therefore, categorical variables such as gender, year of admission, current year or other categorical labels were encoded using **One-Hot Encoding**. This method converts categorical variables into binary vectors, ensuring they can be used in models without losing any significant information.

A table with numbers and text

AI-generated content may be incorrect.

Table 4: Categorical features values converted into numerical values

All the categorical data converted to numerical data, which make the model to perform better and more efficiently.

*“As per this study, there are multiple experiments done on this data, and it has been realized that when categorical data used directly on model, is not just affect the result but also effect the computational time of models as well. Although the effect was in micro-seconds, but still the difference was spotted.”*

* **Normalization/Standardization:** Some features have different scales, which can negatively affect model performance. To address this, **feature scaling** was applied using **Min-Max Scaling** to normalize the values between 0 and 1. This step ensures that all features contribute equally to the learning process. (“Best Practices for Data Quality in Machine Learning - Anomalo”).
* Feature Selection: Not all features are equally important for predicting student performance. Feature selection techniques were employed to identify and retain the most significant features. Using correlation analysis and mutual information and reduced the dataset to the most relevant features, thereby improving the model's accuracy and interpretability.

*“In the dataset there is one column name as “Year\_of\_Admission”, in this column 3 unique values are there, ‘School 1 Current Student’, ‘School 2 Current Student’ and ‘New admission 2019-20’. First, both school data are merged into one unique value i.e. ‘Current Student’ because “Current School” column has this data already. Then, check the feature\_imbalance in each column and find out the ‘Current Student’ value had 1397 rows, and ‘New student’ value had 103 rows in dataset, which caused a* ***biased*** *in model. So, the decision was to remove the 103 rows contains ‘New admission 2019-20’ value and for that dimensionality reduction techniques is used to remove the column “Year\_of\_Admission”.*

## **4.3** **Model Selection and Training**

In this study, several machine learning algorithms were considered for predicting student performance. These include both traditional models and advanced ensemble techniques. The following models were implemented:

1. **Decision Tree Classifier (DT):** A decision tree is a supervised machine learning model that is simple to understand and interprets data based on a series of binary decisions. It is used to predict student performance by analysing the hierarchical relationships between input features.
2. **Support Vector Machine (SVM):** SVM is a classification algorithm that works well for binary classification tasks. It seeks to find the optimal hyperplane that maximizes the margin between the two classes (e.g., Pass/Fail). This method is effective for handling complex datasets with nonlinear boundaries.
3. **Random Forest Classifier:** Random Forest is an ensemble method based on decision trees. It builds multiple decision trees using bootstrapped samples of the dataset and aggregates their predictions to improve accuracy. This model reduces the risk of overfitting compared to individual decision trees and provides robust predictions.
4. **Gradient Boosting Machine (GBM):** GBM is an ensemble method that builds trees sequentially, each correcting the errors of the previous tree. This approach often leads to highly accurate results, especially for complex datasets. It is particularly useful for student performance prediction, as it can learn the relationships between features more effectively.
5. **Ensemble Learning Model:** This research also includes to explore **Ensemble Learning**, which combines the predictions of multiple models to produce a final prediction. A **Voting Classifier** was implemented, which aggregates predictions from all the models above, and the final output is determined based on a majority vote from each model's prediction.

The models were trained using the pre-processed dataset, and the **training-validation split** was used, where 80% of the data was used for training the model and 20% for validation. The models were then fine-tuned using **hyperparameter optimization techniques** (e.g., Grid Search, Random Search) to identify the best configuration of parameters for each model.

## **4.4** **Evaluation Metrics**

Assess the performance of the models, several evaluation metrics were employed. [[16]](#Ref16) These metrics help determine how well the models predict student performance and whether they generalize well to unseen data:

1. **Accuracy:** The ratio of correct predictions to the total number of predictions. (“Uncertainty Quantification and Interpretability for Clinical Trial ...”) While commonly used, accuracy alone may not be sufficient in cases with imbalanced datasets (e.g., when most students pass).
2. **Precision:** The proportion of true positives (correct predictions of passing students) out of all predicted positive instances. This metric is essential when the focus is on minimizing false positives.
3. **Recall (Sensitivity):** The proportion of true positives out of all actual positive instances. (“Accuracy, Precision, Recall, and F1-Score”) Recall is crucial when it is important to capture as many true positives as possible, such as identifying students who are at risk of failing.
4. **F1 Score:** The harmonic means of precision and recall. It provides a balance between the two metrics and is useful in situations where both false positives and false negatives are critical.
5. **Confusion Matrix:** A matrix showing the actual vs. predicted classifications, allowing for a deeper understanding of model performance and identifying the types of errors made (e.g., false positives, false negatives).

The models were evaluated on the validation set, and the best-performing model was selected for further testing and comparison with other models.

## **4.5** **Expandable AI (LIME)**

The study provides essential insights into the field of Explainable AI (XAI) [[17]](#Ref17), [[23]](#Ref23), focusing on methods that help make machine learning models more understandable and transparent. One such method discussed in the paper is LIME, which stands for Local Interpretable Model-agnostic Explanations. LIME is a technique designed to explain how machine learning models make predictions, helping us understand the reasoning behind their decisions.

LIME primarily emphasizes explaining individual predictions rather than providing an explanation for the entire model. This is important because while a machine learning model might make accurate predictions across a broad dataset, study may need to understand why the model predicted a specific outcome for a particular instance. For example, suppose study used a Support Vector Machine (SVM) model to predict a binary outcome, such as whether a person will develop a certain medical condition (TRUE or FALSE). If the model predicts a TRUE value for a specific instance (say row 219) and the actual value is also TRUE, study may want to understand how the model arrived at this prediction.

In such a case, LIME works by analysing the specific features associated with that instance (row 219) and how they contributed to the model's prediction. For example, in medical predictions, features could include factors like age, blood pressure, and lifestyle choices. LIME then generates an explanation for that individual prediction, showing which features had the most impact on the prediction. This is typically done by presenting a graphical representation of the features, along with numerical values that reflect their influence.

One of the key characteristics of LIME is its focus on local interpretability. This means that LIME only explains the prediction for a specific instance based on its individual features, rather than attempting to explain the entire model’s behaviour. This localized approach helps provide more accurate and relevant explanations for individual cases. Essentially, it allows users to zoom in on a particular prediction and understand the factors that led to that decision, making it easier to trust the model's output.

Furthermore, LIME is "model-agnostic," meaning that it can be applied to any machine learning model, whether it's a decision tree, a support vector machine (SVM), or a neural network. It doesn’t require knowledge of the inner workings of the model, which makes it a versatile and widely applicable tool in the field of Explainable AI.

In summary, LIME enhances transparency in machine learning by breaking down complex predictions into understandable components, offering insights into the influence of different features on the model’s decision for a specific instance. This makes it easier for both developers and end-users to trust the model, ensuring that decisions are based on clear, explainable reasoning.

## **4.6 Introduce a Web-Application**

Incorporating the web application into the study adds a practical dimension to the research. The web app serves as an interactive platform where users (students, parents, and educators) can input the student related data, such as demographic information, academic performance, and exam scores, to receive personalized academic predictions. The app communicates directly with the machine learning models running in the backend, feeding the inputs through the models, which generate the predictions in real-time, and answer this question, “Weather the chosen grade is align with student interest or not?”

The web app is designed with user-friendliness in mind, featuring a simple, intuitive interface that guides users through the data input process. Once the data is submitted, the app returns a prediction about the student’s future academic performance, along with suggestions for improvement, based on the model's findings. For students, the web app provides an opportunity for self-assessment and awareness of areas that may require additional focus. Parents can use the app to stay informed about their child's academic trajectory, enabling them to provide timely support. Schools can leverage this tool to identify at-risk students and take proactive measures to enhance their educational experience.

By integrating the machine learning models into a web application, this study bridges the gap between theory and practice, making the predictive models accessible, interactive, and actionable for all stakeholders involved in the educational process. The web app not only makes the study more robust but also empowers users with actionable insights to enhance student learning outcomes.

# **5.** **Results**

The results of the machine learning models implemented in this study are presented in this section. The models were evaluated using several performance metrics, including accuracy, precision, recall, F1 score, and confusion matrix. The models tested included Decision Tree (DT), Support Vector Machine (SVM), Random Forest (RF), Gradient Boosting Machine (GBM), and an Ensemble Learning Model. In addition, this paper also explored the performance of a Graph-Based Ensemble Learning approach, which integrates both ensemble learning and graph-based methods to improve prediction accuracy and stability.

## **5.1** **Data Pre-Processing Result**

A number of blue squares

AI-generated content may be incorrect.

Table 5: Class distribution in Training data.

For training data, the target is perfectly balanced after using Dimensionality reduction method of PCA.

A number of blue squares

AI-generated content may be incorrect.

Table 6: Class distribution in Testing data.

In a same way the testing data is also perfectly balanced. The total data 1397 was divided into 4:1 in training and testing respectively and by checking the balance gives 100 sureties that the biased in the class has been removed. This balancing is a result of preprocessing what this study had explained previously, before this the data was little bias with the new student as there were only 103 rows for them, this crated a bias in the model, if study consider the student is new or old, but after removing those rows and “Year\_Of\_Admission” feature. As per experienced people says, in AI, it is always recommended to use more data, “The larger the data is, the better result the model can give”, but it is also keep in mind that the data should free of biased in any aspect, because the biased data not only impact the performance of the model but also effect the result and predictions on real-time data. Therefore, it is always necessary to remove the presence of any kind of bias.

## **5.2 Performance Metrics**

choice of performance metrics is basically dependent upon the model type, choice or regression or classification etc. Here, in this project the performance of the individual models is measured by Accuracy, Recall, Precision and AUC Curve. These all metrics are important in their own. Let’s see and discuss the result what this study gets from each model.

A table with numbers and a number of objects

AI-generated content may be incorrect.

Table 7: Performance metrics for each model.

**1. Random Forest**

* **Precision** (0.901) is quite high, meaning that when the model predicts a positive class, it is correct 90.1% of the time. This indicates that the model is good at avoiding false positives.
* **Recall** (0.765) is moderate, suggesting that the model correctly identifies 76.5% of the actual positive cases but misses around 23.5% of them.
* **F1-Score** (0.827) balances precision and recall, indicating the model performs reasonably well overall, though there is room for improvement in recall.

**Why this is happening:**

Random Forest is a robust ensemble model that typically performs well in handling data with high dimensionality and imbalanced classes. Its relatively high precision shows that it’s conservative in predicting positives, leading to fewer false positives.

**2 ANN (MLP)**

* **Precision** (0.779) and **Recall** (0.744) are reasonably balanced, but both are relatively low, suggesting that the model does a fairly decent job at predicting the positive class, but there’s still significant room for improvement.
* **Recall** (0.744) is moderate, suggesting that the model correctly identifies 76.5% of the actual positive cases but misses around 23.5% of them.
* **F1-Score** (0.758) indicates that the model strikes a balance between precision and recall, but its overall performance is not stellar.

**Why this is happening:**

Artificial Neural Networks (ANNs) (specifically, a Multi-Layer Perceptron) are flexible and powerful models, but their performance can be sensitive to the architecture, hyperparameters, and data preprocessing. The relatively lower scores could be due to suboptimal configurations or insufficient training data for such a complex model.

Not just accuracy, but the value of Recall, Precision and F1-Score all are low.

**3. SVM (Support Vector Machine)**

* **Precision** (0.988) and **Recall** (0.989) are both very high, indicating that the SVM is very effective at correctly identifying positive cases and avoiding false positives or negatives.
* **F1-Score** (0.989) reflects an excellent balance between precision and recall.

**Why this is happening:**

SVM models, especially with appropriate kernel functions, are known for their ability to create clear decision boundaries, suggest even in complex datasets. The SVM's high accuracy and ROC AUC of 1.0. The model can generalize very well on the test data, leading to few errors.

**4. KNN (K-Nearest Neighbours)**

* **Accuracy** (0.858) is high, indicating that when KNN predicts the positive class, it is very accurate.
* **Precision** (0.948) is high, indicating that when KNN predicts the positive class, it is very accurate.
* **Recall** (0.739) is lower, meaning that the model is missing a significant portion of the actual positive cases.
* **F1-Score** (0.829) reflects this imbalance, showing that while precision is good, recall still needs improvement.

**Why this is happening:**

KNN works by comparing new data points to a set of neighbours, and its performance often depends on the choice of the number of neighbours (K). In this case, the model is good at avoiding false positives but struggles to capture all true positives, leading to a lower recall. The accuracy is slightly better than others but in FN reduced its overall ranking.

**5. Bernoulli Naive Bayes**

* **Precision** (0.808) indicates that when the model predicts the positive class, it is correct most of the time.
* **Recall** (0.679) is relatively low, indicating that the model is missing many actual positive cases.
* **F1-Score** (0.738) reflects this imbalance, showing that while precision is decent, recall is not strong enough to boost the overall performance.

**Why this is happening:**

Naive Bayes models assume conditional independence between features, which can be a limitation when features are highly correlated. This leads to suboptimal performance, especially with more complex datasets where this assumption doesn't hold.

***“****In this journal the lowest accuracy is given by Bernoulli Naive Bayes, because it assumes that all features are independent, which cloud be true in real-world data. If it gives the lowest accuracy among these algorithms, it means that the feature dependencies in the dataset are significant, and its simple probabilistic approach is not capturing complex patterns well. Other algorithms like Random Forest, SVM, or Neural Networks handle feature interactions better, leading to higher accuracy****”***

**6. AdaBoost**

* **Precision** (0.882) is relatively high, meaning that AdaBoost does a good job of predicting positive cases accurately.
* **Recall** (0.838) is also quite good, though slightly lower than precision, indicating that it does miss a few positive cases.
* **F1-Score** (0.858) suggests a strong overall performance in balancing precision and recall.

**Why this is happening:**

AdaBoost works by combining multiple weak learners (often decision trees) to create a strong model. Its performance tends to improve with more iterations, making it effective for classification tasks, especially when combined with classifiers that may underperform on their own. better but not best. Although Recall cause issues which reflect in overall result and so finally, one should not use AdaBoost as the model.

**7. XGBoost**

* **Precision** (0.906) is high, indicating that XGBoost makes accurate positive predictions.
* **Recall** (0.861) is also quite strong, though it still misses some positive cases.
* **F1-Score** (0.883) reflects a solid balance between precision and recall.

**Why this is happening:**

XGBoost is an efficient gradient boosting algorithm that works well for classification tasks. It is known for handling imbalances in datasets and for improving performance through regularization. Its strong ROC AUC and F1-Score reflect its overall effectiveness.

**8. Extra Trees**

* **Precision** (0.903) is high, indicating that it is accurate when predicting the positive class.
* **Recall** (0.760) shows that it misses some positive cases, which slightly lowers its F1-Score (0.825).

**Why this is happening:**

Extra Trees is an ensemble method like Random Forest, but it differs in that it introduces more randomness in how trees are built. This randomness can help avoid overfitting but sometimes leads to slightly lower recall compared to models like Random Forest. It is better than AdaBoost but still but fair enough to give the excepted result.

**9. Stacking**

* **Precision** (0.890) is high, but **Recall** (0.800) shows that it still misses some positive cases.
* **F1-Score** (0.842) is a good balance between precision and recall, indicating that the model performs well overall.

**Why this is happening:**

Stacking involves combining multiple models to form a stronger predictor, but the effectiveness depends on the individual models in the ensemble and how well they complement each other. In this case, stacking provides a well-rounded performance, though not as stellar as some of the top models.

**10. Voting Classifier**

* **Precision** (0.962) is extremely high, showing that when the model predicts positive cases, it is correct most times.
* **Recall** (0.881) is also strong, though not perfect, meaning that some positive cases are missed.
* **F1-Score** (0.919) shows that the model strikes a good balance between precision and recall.

**Why this is happening:**

Voting Classifiers combine the predictions of multiple models, and their performance reflects the strength of the individual models in the ensemble. The high ROC AUC, precision, and recall indicate that the ensemble method is highly effective in this scenario, resulting in excellent overall performance.

**Let’s summarize the models result in short:**

1. SVM has exceptional performance across all metrics, particularly excelling in precision, recall, and ROC AUC, making it the top performer here.
2. Voting Classifier also performs very well due to the combination of multiple strong models.
3. XGBoost, AdaBoost, and Random Forest all provide strong results, and the precision being their strength.
4. ANN and Bernoulli Naive Bayes have relatively lower performance, particularly in recall, indicating that they are not as good at capturing all positive cases,

## **5.3** **ROC AUC Analysis**

A graph of a function

AI-generated content may be incorrect.

Fig 2: ROC curve for each model.

A screenshot of a graph

AI-generated content may be incorrect.

Table 8: Tabular representation of ROC curve for each model.

The Receiver-operating characteristic curve (ROC) Area under the curve (AUC) scores show how effectively each model differentiates between the two classes. SVM leads with a perfect score of 1.000, indicating flawless classification. The Voting Classifier (0.991) and XGBoost (0.966) also perform exceptionally well, with scores that suggest near-perfect separation of classes. AdaBoost (0.952), Stacking (0.918), and KNN (0.940) provide solid, reliable performance. Models like Random Forest (0.949) and Extra Trees (0.946) follow closely, showing good discrimination between classes. However, the ANN (MLP) with a score of 0.842 and Bernoulli Naive Bayes at 0.780 struggle more, highlighting weaker class separation and less accurate predictions.

## **5.4** **Confusion Matrix Analysis**

The confusion matrix for each model is discussed here with graph value

A table with numbers and letters

AI-generated content may be incorrect.

Table 9: The confusion matrix representation for each model.

The results reflect the performance of a range of machine learning models under a classification task, and each model has its unique strengths and weaknesses.

A blue squares with white text

AI-generated content may be incorrect.Fig 3: Confusion Matrix for SVM.

The **SVM model**, with its minimal errors, shows the high efficiency of support vector machines in classification tasks where the decision boundary is clear and well-separated. SVM’s exceptional performance, especially with a single false positive and zero false negatives, indicates that it is very effective at both minimizing type I and type II errors. This suggests that the SVM could be an ideal choice for high-stakes applications where both false positives and false negatives could lead to significant consequences, such as medical diagnosis or fraud detection.

**A blue squares with white text

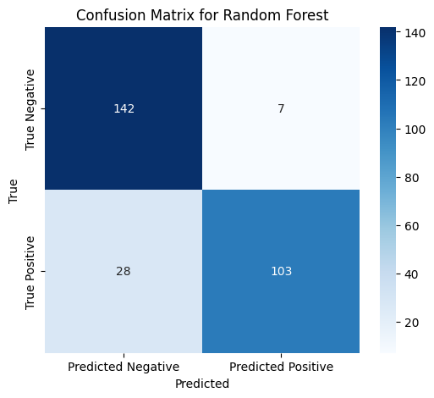
AI-generated content may be incorrect.**However, SVMs can also struggle with larger datasets or more complex, non-linear decision boundaries. It's worth noting that SVM's performance might degrade as the complexity of the dataset increases or when computational resources are constrained.

Fig 4: Confusion Matrix for Random Forest. Fig 5: Confusion Matrix for XGBoost.

**Random Forest** and **XGBoost** show strong, but not perfect, results. These models rely on ensemble learning techniques, which generally provide a higher degree of robustness and handle non-linear relationships better than individual models. Random Forest performs well but has a higher number of false negatives, suggesting that while it identifies negative cases accurately, it might miss some positive instances that have more subtle features. The model’s performance can be further enhanced by tuning hyperparameters or by using feature engineering to reduce the complexity of the data. XGBoost, another ensemble method, works well overall but like AdaBoost, is prone to some level of false negatives, which may indicate that it is more cautious about predicting the positive class.

**A blue squares with white text

AI-generated content may be incorrect.**

Fig 6: Confusion Matrix for ANN (MLP).

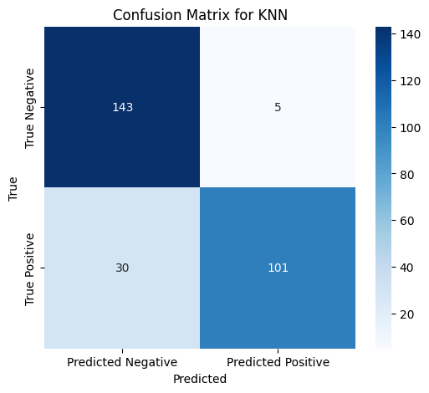
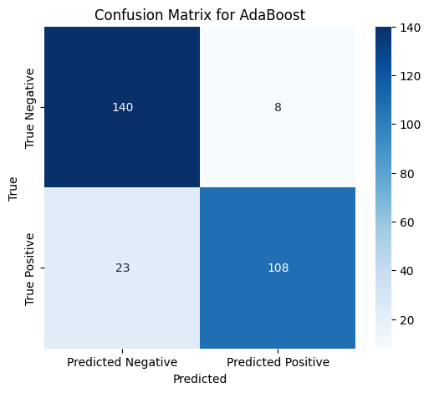
**ANN (MLP)**, on the other hand, performed similarly to Random Forest but struggled more with false positives. This points to the possibility that the MLP model is overfitting to certain features or patterns in the dataset, leading it to misclassify some instances. In future work, fine-tuning the architecture of the neural network, adjusting its learning rate, or exploring regularization techniques could improve its results.

Fig 7: Confusion Matrix for KNN.

A graph of negative and negative results

AI-generated content may be incorrect.The **KNN model** exhibits moderate performance but struggles with false negatives. KNN is highly sensitive to the distance metric and can be impacted by irrelevant features or noisy data, leading to poor performance in cases with complex boundaries between classes. It might be worth experimenting with feature selection or using weighted KNN to better address these limitations.

Fig 8: Confusion Matrix for AdaBoost. Fig 9: Confusion Matrix for Extra Trees.

**AdaBoost** and **Extra Trees** are slightly weaker performers, mainly due to their susceptibility to false negatives. These models can be improved by fine-tuning their parameters, particularly the learning rate in AdaBoost, or adjusting the number of estimators in Extra Trees. In general, these models are not as robust as the top performers like SVM or Random Forest, but they can still be effective in certain contexts, especially when coupled with proper feature engineering or in scenarios with highly imbalanced data.

A graph with blue squares

AI-generated content may be incorrect.Fig 10: Confusion Matrix for Bernoulli Naive Bayes.

A blue squares with white text

AI-generated content may be incorrect.The **Bernoulli Naive Bayes** model, while often a strong candidate for text classification or binary classification tasks, falls short here. Its higher false negative count indicates that it has trouble distinguishing between the positive and negative classes. This model is simple and fast, but for tasks requiring high accuracy in identifying the positive class, more sophisticated models like SVM or XGBoost should be considered.

Fig 11: Confusion Matrix for **Stacking**.

**Stacking** works by combining the outputs of multiple models and can offer a good compromise, especially when individual models have complementary strengths. While it performs reasonably well, it still suffers from false positive and false negative predictions. This could be a useful approach when no single model excels across all metrics, as it blends different perspectives into one final decision.

A graph of a blue box

AI-generated content may be incorrect.Fig 12: Confusion Matrix for **Voting Classifier**.

The **Voting Classifier** model combines multiple base learners, and while it performs decently, it does not outperform models like SVM. Its ability to mitigate individual model errors by voting makes it a strong contender for applications where robustness is more important than sheer accuracy. However, its performance could be further enhanced by refining the individual base models.

In general, the models that rely on ensemble methods (like SVM, Random Forest, XGBoost, and Voting Classifier) demonstrate better performance due to their ability to aggregate knowledge from multiple base models. These techniques tend to outperform single-model approaches such as Naive Bayes or KNN, particularly in more complex classification tasks.

## **5.5** **Expandable AI (LIME) Result**

As this study comprehensive discuss about the LIME and here is the result. This study has chosen 5 best and most used models i.e. SVM, Random Forest Classification, Voting Classier, AdaBoost and XGBoost. As all the models have different way to predicting the output so when the study chooses the instance (Row 219) the effect of those features is in different way for different models. Here is the result for chosen instance 219.

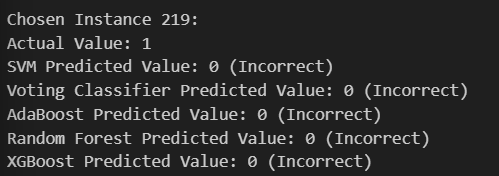


Fig 13: LIME setup for instance 219.

The instance 219 has 1 actual value but all the models predict it wrongly as 0. Now let’s see why the models predict it wrongly and what are the reason for this wrong prediction.

**1. SVM (Support Vector Machine)**

Support Vector Machine (SVM) is a powerful classifier that finds the optimal hyperplane to separate different classes in the data. It works well, especially with complex or high-dimensional datasets. SVM often provides the best results when data is well-structured, as it maximizes the margin between classes, leading to accurate predictions. [[18]](#Ref18)

A graph with numbers and a bar chart

AI-generated content may be incorrect.

Fig 14: LIME explanation for SVM Model.

The SVM model uses the following features with their respective importance values:

* **Math193\_ <= 70.00**: 0.1721: This feature has the highest positive impact. It suggests that whether a student's math score in the 193 class is below or equal to 70 significantly influences the model's prediction in a positive way.
* **Mathexam > 84.00**: 0.1861: This feature has a negative influence. If the student's math exam score is greater than 84, it tends to push the model's prediction in the opposite direction.
* **Math202\_ > 92.00**: 0.1393: Similar to the previous feature, this indicates a negative impact, where a score above 92 in Math202 leads to a negative influence on the outcome.
* **Science202\_ <= 78.00**: 0.1080: A lower score in Science202 (below or equal to 78) contributes positively to the prediction.
* **English193\_ > 88.80**: 0.1361: A higher score in English193 (above 88.80) negatively influences the prediction.
* **80.20 < Math191\_ <= 91.00**: 0.0833: This indicates a moderate negative influence on the prediction when Math191 is between 80.20 and 91.
* **English201\_ > 88.40**: 0.1043: A higher English score (above 88.40) also pushes the prediction in the negative direction.
* **83.00 < Science193\_ <= 90.00**: 0.0674: A Science193 score between 83 and 90 leads to a slight negative influence.
* **83.00 < Science191\_ <= 89.07**: 0.0720: Similar to the previous feature, this has a slight negative influence.
* **82.70 < Science201\_ <= 88.30**: 0.0750: A similar pattern with Science201, contributing negatively to the model’s prediction.

**2. Random Forest**

A Random Forest is an ensemble method that builds multiple decision trees and combines their predictions to improve accuracy and reduce overfitting. It works by averaging the results of many trees, which helps make the model more reliable. [[19]](#Ref19)

**A graph with red bars and numbers

AI-generated content may be incorrect.** Fig 15: LIME explanation for Random Forest Model.

The Random Forest model provides similar but slightly different feature importances:

* **Math193\_ <= 70.00**: 0.1371: This feature also appears to have a positive effect on the model’s prediction.
* **0.00 < Previous\_yearGrade <= 1.00**: 0.1027: This suggests that the previous year's grade plays a role in the prediction, with lower grades (between 0 and 1) contributing negatively.
* **0.00 < Current\_Curriculum <= 1.00**: 0.0889: If the student is following the current curriculum (score between 0 and 1), it has a slight negative impact.
* **0.00 < Current\_School <= 1.00**: 0.0679: Similar to the previous feature, the school attendance or performance has a minor negative effect.
* **0.00 < Previous\_Curriculum\_17182 <= 1.00**: 0.0571: Previous curriculum also has a slight negative influence on the model.
* **English193\_ > 88.80**: 0.0409: Higher scores in English193 again lead to a slight negative impact.
* **English201\_ > 88.40**: 0.0375: Higher English scores also contribute negatively, but slightly less so than in SVM.
* **Mathexam > 84.00**: 0.0394: If the math exam score is high (above 84), it has a small negative influence.
* **85.00 < English203\_ <= 91.30**: 0.0264: A moderate negative impact for English203 scores between 85 and 91.
* **85.00 < Science193\_ <= 90.00**: 0.0262: This feature also has a slight negative influence on the prediction.

**3. Voting Classifier**

As per the reference by [[20]](#Ref20), a **Voting Classifier** combines multiple machine learning models to improve prediction accuracy by considering their collective votes. This helps make the model more robust and reduces the risk of overfitting.

**A screen shot of a graph

AI-generated content may be incorrect.** Fig 16: LIME explanation for Voting Classifier Model.

The Voting Classifier is an ensemble of several models, so it combines the feature importances from multiple models.

* **Math193\_ <= 70.00**: 0.1848: Like SVM, this feature has a high positive influence on the prediction.
* **English193\_ > 88.80**: 0.0953: A higher English score has a moderate negative influence on the outcome.
* **Mathexam > 84.00**: 0.0770: This feature also has a negative impact, similarly in SVM and Random Forest.
* **80.20 < Math191\_ <= 91.00**: 0.0554: This feature contributes negatively, but to a smaller degree.
* **0.00 < Previous\_yearGrade <= 1.00**: 0.0548: Previous grades have a slight negative effect on the prediction.
* **0.00 < Current\_Curriculum <= 1.00**: 0.0488: The effect of current curriculum is minor, with a negative influence.
* **83.00 < Science193\_ <= 90.00**: 0.0425: This feature has a slight negative effect on the prediction.
* **0.00 < Previous\_Curriculum\_17182 <= 1.00**: 0.0408: Previous curriculum also contributes negatively, but slightly less.
* **0.00 < Current\_School <= 1.00**: 0.0371: Similar to the others, the current school status has a slight negative impact.
* **Math202\_ > 92.00**: 0.0362: High scores in Math202 have a minor negative influence.

**4. AdaBoost**

According to the journal by Kalaiselvi et al. [[21]](#Ref21), the AdaBoost is a great choice when the study is related to classification. it is mentioned that AdaBoost, along with LIME, helps to identify how the AdaBoost model behaves in the background.

**A screenshot of a computer

AI-generated content may be incorrect.**

Fig 17: LIME explanation for AdaBoost Model.

AdaBoost is a boosting algorithm, which focuses on improving the weak learners.

* 0.00 < Current\_Curriculum <= 1.00: 0.0633: This feature has a moderate negative influence.
* Math193\_ <= 70.00: 0.0493: This feature has a positive effect on the prediction, though it’s smaller compared to other models.
* 85.00 < English203\_ <= 91.00: 0.0228: A small negative influence.
* Mathexam > 84.00: 0.0214: Similar to other models, this feature has a small negative impact.
* Science202\_ <= 78.00: 0.0256: A positive influence, indicating that lower Science202 scores are slightly better.
* 80.20 < Math191\_ <= 91.00: 0.0174: A minor negative impact.
* English201\_ > 88.40: 0.0167: A slight negative influence for high English201 scores.
* 71.00 < Science192\_ <= 82.70: 0.0139: A small positive influence.
* English193\_ > 88.80: 0.0145: Negative influence for high English193 scores.
* 68.00 < Englishexam\_ <= 75.00: 0.0122: A very small positive effect.

**5. XGBoost**

By referring to the work of Li and Zhou (2023) [[22]](#Ref22), which helps to gain a deeper understanding of the XGBoost algorithm. Using LIME with XGBoost, this study able to identify key features influencing the model’s predictions, such as the impact of Math193\_ and English193\_ scores. This approach helped clarify the relationships between various features and student performance, aligning with the findings in their study. Here is the LIME with XGBoost.

**A screenshot of a graph

AI-generated content may be incorrect.**

Fig 18: LIME explanation for XGBoost Model.

XGBoost, known for its powerful performance in classification tasks, provides the following feature importances:

* Math193\_ <= 70.00: 0.2644: This feature has the highest positive importance in XGBoost, meaning it's a strong indicator of the prediction.
* 0.00 < Previous\_Curriculum\_17182 <= 1.00: 0.1710: This feature has a strong negative impact.
* English193\_ > 88.80: 0.1002: Higher scores in English193 negatively affect the prediction.
* Math202\_ > 92.00: 0.0893: This feature also has a negative influence, though weaker than the previous two.
* Mathexam > 84.00: 0.0871: Like in other models, higher math exam scores have a small negative influence.
* Science202\_ <= 78.00: 0.0785: A positive influence for lower Science202 scores.
* 85.00 < Science201\_ <= 90.30: 0.0878: Negative effect for Science201 scores in this range.
* 85.00 < Science193\_ <= 90.00: 0.0751: Negative influence for Science193 scores in this range.
* English201\_ > 88.40: 0.0745: Slight negative influence for higher English201 scores.
* 76.00 < Math203\_ <= 87.60: 0.0666: Small positive influence for Math203 scores in this range.

## **5.6 The Web Application**

The web application developed as part of this study uses the Support Vector Machine (SVM) model, which was chosen due to its strong performance in predicting student outcomes. After testing multiple machine learning models, SVM proved to be the most reliable, providing high accuracy and consistency in predicting academic performance based on a combination of numerical and categorical data.

To further validate the model’s effectiveness, 500 random new data points were generated and tested using the SVM model. Of these, 321 instances predicted that the chosen grade was appropriate for the student, while 179 instances indicated potential issues with the grade selection. The significant difference between these two groups highlights how often the model’s predicted grade aligns with the student’s other academic data and subject scores.

A white and blue box with black text

AI-generated content may be incorrect.

Fig 19: 500 Random New Dataset for WebApp.

The decision to use the SVM model in the web app was reinforced by the findings from Local Interpretable Model-agnostic Explanations (LIME), which showed that the numerical features, such as student scores in Maths, Science, and English, had a larger impact on the model's predictions than the categorical features. This insight confirmed that the SVM model is effective at handling key academic data, making it the most suitable choice for providing accurate predictions.

The web app allows students, parents, and educators to input student data and receive immediate feedback on whether the proposed grade is appropriate. The app not only provides predictions but also offers transparency by explaining which factors contributed to the result. This helps users understand the basis of the predictions, building trust in the model’s accuracy.

A screenshot of a login screen

AI-generated content may be incorrect.

Fig 20: WebApp User input pages

The User can input all the required details about the student in this page, and submit the form for model evaluation, at backend the model will predict the outcome.A screenshot of a test results

AI-generated content may be incorrect.

Fig 21: WebApp Result shown after model prediction

The model outcome is dependent upon the scores of students and categorical data, if the chose grade is matching with the Model output them it will says the positive part else the model will suggest like which subject is needed to be improve or user have option to choose the different grade for the student.

By integrating the SVM model, the web app offers an intuitive and informative tool for all users. It gives students personalised insights into their academic progress, helps parents make informed decisions, and assists schools in identifying students who may need additional support.

# **6.** **Discussion of Results**

This study explores the potential of various machine learning models to predict student performance in an academic setting, aiming to facilitate early identification of at-risk students. Among the models tested, Support Vector Machine (SVM) demonstrated exceptional performance, achieving high precision, recall, and F1-Score. This result is particularly significant given that SVM’s ability to handle complex decision boundaries was key to minimizing false positives and false negatives. The Voting Classifier, which aggregates predictions from multiple models, also showcased strong results, validating the effectiveness of ensemble approaches in this domain.

One of the most compelling aspects of this research is the consistent importance of precision and recall over simple accuracy. In the context of educational predictions, where the stakes are high (e.g., identifying students at risk of failing), a model that balances these metrics effectively is crucial. XGBoost and Random Forest also provided competitive results, further reinforcing the power of ensemble learning methods in handling high-dimensional, imbalanced datasets typical of educational data.

Despite these successes, the study highlighted limitations inherent in simpler models like ANN and Bernoulli Naive Bayes. While these models are computationally efficient, they struggled to capture complex relationships in the data, especially when feature dependencies were significant. This aligns with the broader trend that more sophisticated models, especially ensemble-based methods, are better suited for tasks involving complex data structures and non-linear relationships.

The findings from this study have practical implications for educational institutions, especially in the context of early intervention programs. By leveraging machine learning models, schools can not only predict student outcomes more accurately but also provide personalized support to students who need it most. This, in turn, can lead to higher graduation rates, reduced dropout rates, and a more tailored educational experience for students.

To better understand the models' decisions, LIME (Local Interpretable Model-Agnostic Explanations) was used to analyse feature importance for instance 219. The results showed that key features had different influences across models, sometimes leading to incorrect predictions. For example, the SVM model placed significant weight on Math193\_ <= 70.00, making it a strong indicator, while English and Science scores negatively influenced predictions. Random Forest and XGBoost followed similar patterns but assigned varying degrees of importance to features like previous grades and curriculum details. These differences highlight the challenge of model interpretability in education—while machine learning can identify patterns, the reasoning behind individual predictions can vary significantly, making Explainable AI essential for ensuring trust in model decisions.

# **7.** **Conclusion of Results**

This research demonstrates that machine learning, particularly SVM, Random Forest, and XGBoost, can significantly enhance the prediction of student performance, providing valuable insights for educators and policymakers alike. Similar studies, such as those by Karim et al. [[23]](#Ref23) (2024), have also highlighted the potential of machine learning in the educational context, where their hybrid CNN-SVM model was able to achieve exceptional results in predicting anxiety levels among university students. This aligns with the current research, where machine learning models are used to predict various student outcomes, showcasing the broader applicability of AI in education.

By identifying at-risk students early, educational institutions can intervene proactively, improving academic outcomes and fostering a more supportive learning environment. Hasib et al. (2022) [[24]](#Ref24) emphasised the importance of using machine learning to assess secondary school student performance, with their study showing that SVM achieved the best accuracy at 96.89%. This reinforces the findings of the current study, where SVM also played a critical role in achieving high accuracy in student performance prediction. These studies collectively demonstrate that the predictive power of machine learning can be harnessed to better understand student success factors and improve intervention strategies.

While the models performed well in terms of accuracy and predictive power, there remains room for further optimisation, particularly in refining hyperparameters, exploring hybrid models, and incorporating additional data sources such as behavioural metrics or socio-economic factors. Future work should focus on improving model interpretability, allowing stakeholders (e.g., teachers, counsellors) to understand why certain predictions are made, thus enhancing the trust and adoption of machine learning systems in real-world educational settings. The use of LIME, as seen in both Karim et al. (2024) and Hasib et al. (2022), has been instrumental in offering deeper insights into model decisions, helping to make machine learning more transparent and actionable in educational settings.

Ultimately, this study underscores the transformative potential of machine learning in education, not only for predicting student performance but also for driving data-driven decision-making. By embracing these advanced analytics, schools and universities can ensure that every student receives the support they need to succeed academically.

As highlighted by both Karim et al. (2024) and Hasib et al. (2022), the integration of explainable AI LIME, techniques can enhance the effectiveness and adoption of machine learning models, ensuring that educators can make informed, data-backed decisions to support students’ academic and personal development. The use of LIME provided deeper insights into why models misclassified certain instances, particularly instance 219. By breaking down feature importance, LIME revealed that models relied heavily on maths and English scores, sometimes in conflicting ways. For example, while Math193\_ <= 70.00 had a strong positive influence in multiple models, high English and Science scores often contributed negatively to predictions. This suggests that while machine learning models capture general trends well, they may struggle with nuanced cases where multiple academic factors interact in unexpected ways. These findings reinforce the importance of Explainable AI in education, teachers and administrators must understand why models make specific predictions to make informed, data-driven decisions that genuinely benefit students.

The integration of the SVM model into the web application provides a practical and accessible tool for predicting student performance. By leveraging machine learning, the app empowers students, parents, and educators with valuable insights into academic outcomes, fostering informed decision-making. The SVM model’s strong predictive capabilities, validated through both the random data testing and LIME analysis, ensure that the app delivers accurate and reliable results. This innovative approach not only enhances the student evaluation process but also contributes to the ongoing development of data-driven educational strategies.

# **8.** **List of Tables and Figures**

## **8.1 List of Figures**

Fig 1[: ML Pipeline for the project.](#Fig1)

Fig 2: [ROC curve for each model](#Fig2).

Fig 3: [Confusion Matrix for SVM.](#Fig3)

Fig 4: [Confusion Matrix for Random Forest.](#Fig4)

Fig 5: [Confusion Matrix for XGBoost.](#Fig5)

Fig 6: [Confusion Matrix for ANN (MLP).](#Fig6)

Fig 7: [Confusion Matrix for KNN.](#Fig7)

Fig 8: [Confusion Matrix for AdaBoost.](#Fig8)

Fig 9: [Confusion Matrix for Extra Trees.](#Fig9)

Fig 10: [Confusion Matrix for Bernoulli Naive Bayes.](#Fig10)

Fig 11: [Confusion Matrix for Stacking.](#Fig11)

Fig 12: [Confusion Matrix for Voting Classifier.](#Fig12)

Fig 13: [LIME setup for instance 219.](#Fig13)

Fig 14: [LIME explanation for SVM Model.](#Fig14)

Fig 15: [LIME explanation for Random Forest Model.](#Fig15)

Fig 16: [LIME explanation for Voting Classifier Model.](#Fig16)

Fig 17: [LIME explanation for AdaBoost Model.](#Fig17)

Fig 18: [LIME explanation for XGBoost Model.](#Fig18)

Fig 19: [500 Random New Dataset for WebApp.](#Fig19)

Fig 20: [WebApp User input pages.](#Fig20)

Fig 21: [WebApp Result shown after model prediction.](#Fig21)

## **8.2 List of Table**

Table 1: [Categorical features, content values and description.](#Table1)

Table 2: [Numerical features, content values and description.](#Table2)

Table 3: [List of Missing values in features.](#Table3)

Table 4: [Categorical features values converted into numerical values.](#Table4)

Table 5: [Class distribution in Training data.](#Table5)

Table 6: [Class distribution in Testing data.](#Table6)

Table 7: [Performance metrics for each model.](#Table7)

Table 8: [Tabular representation of ROC curve for each model.](#Table8)

Table 9: [The confusion matrix representation for each model.](#Table9)

# **9.** **References**

1. Ghareeb, S., Hussain, A., Khan, W. and Al-Jumeily, D. (2021). Dataset of student level prediction in UAE. *Data in Brief*, [online] 35, p.106908. doi:<https://doi.org/10.1016/j.dib.2021.106908>.
2. Ghareeb, S., Hussain, A.J., Al-Jumeily, D., Khan, W., Al-Jumeily, R., Baker, T., Al Shammaa, A. and Khalaf, M., 2022. Evaluating student levelling based on machine learning model’s performance. *Discover Internet of Things*, 2(1), p.3. doi:<https://doi.org/10.1007/s43926-022-00023-0>.
3. Qureshi, R. and Lokhande, P.S. (2024). A Comprehensive Review of Machine Learning techniques used for Designing An Academic Result Predictor And Identifying The Multi-Dimensional Factors Affecting Student’s Academic Results. [online] pp.1–6. doi:<https://doi.org/10.1109/idicaiei61867.2024.10842901>.
4. Ahmed, E. (2024). Student Performance Prediction Using Machine Learning Algorithms. Applied Computational Intelligence and Soft Computing, [online] 2024, p.e4067721. doi:<https://doi.org/10.1155/2024/4067721>.
5. Chandra S. K and K Santhosh Kumar (2022). Data Preprocessing and Visualizations Using Machine Learning for Student Placement Prediction. 2022 2nd International Conference on Technological Advancements in Computational Sciences (ICTACS). doi:<https://doi.org/10.1109/ictacs56270.2022.9988247>
6. Lagrazon, G.G., Edytha, J., Rossana, M. and Maaliw, R.R. (2023). Ensemble-Based Prediction Model for Enhanced Electronics Engineering Licensure Examination Results Using Student Performance Analysis. [online] doi:<https://doi.org/10.1109/iceeie59078.2023.10334657>.
7. Rimpy, Dhankhar, A. and Solanki, K. (2022). Educational Data Mining tools and Techniques used for Prediction of Student’s Performance: A Study. 2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), [online] pp.1–5. doi:<https://doi.org/10.1109/icrito56286.2022.9965023>.
8. Bird, K. (2023). Predictive Analytics in Higher Education: The Promises and Challenges of Using Machine Learning to Improve Student Success. *AIR Professional File*, [online] (Fall 2023). doi:<https://doi.org/10.34315/apf1612023>.
9. Issah, I., Appiah, O., Appiahene, P. and Inusah, F. (2023). A systematic review of the literature on machine learning application of determining the attributes influencing academic performance. *Decision Analytics Journal*, 7, p.100204. doi:<https://doi.org/10.1016/j.dajour.2023.100204>.
10. Asthana, P., Mishra, S., Gupta, N., Derawi, M. and Kumar, A. (2023). Prediction of Student’s Performance With Learning Coefficients Using Regression Based Machine Learning Models. *IEEE Access*, 11, pp.72732–72742. doi:<https://doi.org/10.1109/access.2023.3294700>.
11. Mubarak, A.T., Cao, H., Hezam, I.M. and Hao, F. (2022). Modeling students’ performance using graph convolutional networks. 8(3), pp.2183–2201. doi:<https://doi.org/10.1007/s40747-022-00647-3>
12. Stephen Opoku Oppong (2023). Predicting Students’ Performance Using Machine Learning Algorithms: A Review. *Asian Journal of Research in Computer Science*, 16(3), pp.128–148. doi:<https://doi.org/10.9734/ajrcos/2023/v16i3351>.
13. Mohd Fazil, Angélica Rísquez and Halpin, C. (2024). A Novel Deep Learning Model for Student Performance Prediction Using Engagement Data. *Journal of learning analytics*, pp.1–19. doi:<https://doi.org/10.18608/jla.2024.7985>.
14. S, A., V, D., S, M.S. and Srikanth, R. (2023). Systematic Review on Real-Time Students Behavior Monitoring using Machine Learning. *2023 International Conference on Inventive Computation Technologies (ICICT)*, pp.233–237. doi:<https://doi.org/10.1109/icict57646.2023.10134519>.
15. Luza, J.C.J. and Rodriguez, C. (2024). Predictive Attributes in Machine Learning for University Academic Performance: A Feature Engineering Approach. *2024 IEEE 16th International Conference on Computational Intelligence and Communication Networks (CICN)*, [online] pp.443–456. doi:<https://doi.org/10.1109/cicn63059.2024.10847424>.
16. Pistorius, F., Grimm, D., Florian Erdosi and Sax, E. (2020). Evaluation Matrix for Smart Machine-Learning Algorithm Choice. *2020 1st International Conference on Big Data Analytics and Practices* [online] doi:<https://doi.org/10.1109/ibdap50342.2020.9245610>.
17. De Padmasiri, P. and Kasthuriarachchi, S. (2024). Interpretable Prediction of Student Dropout Using Explainable AI Models. *2024 International Research Conference on Smart Computing and Systems Engineering (SCSE)*, pp.1–7. doi:<https://doi.org/10.1109/scse61872.2024.10550525>.
18. Retno, D.M., Nastiti, V.R.S., Aditya, C.S.K. and Azkiya, M.A. (2024). Comparative Analysis of Machine Learning Techniques Student Final Grade Classification Accuracy. *2024 International Conference on Smart Computing, IoT and Machine Learning (SIML)*, [online] pp.250–255. doi:<https://doi.org/10.1109/siml61815.2024.10578074>.
19. Pardos Kumar, D., Arnav Kothiyal, Kumar, R., C Hemantha and Maranan, R. (2024). Random Forest approach optimized by the Grid Search process for predicting the dropout students. *Random Forest Approach Optimized by the Grid Search Process for Predicting the Dropout Students*, 8, pp.1–6. doi:<https://doi.org/10.1109/icicet59348.2024.10616372>.
20. Mohammad T; Kahakashan A; Md.Hamid H; Sadia N; Safa A (2024). Predictive Modelling of Anxiety Levels in Bangladeshi University Students: A Voting-Based Approach with LIME and SHAP Explanations. (2024). *Ieee.org*. [online] doi:<https://doi.org/10.1109/iCACCESS61735.2024.10499576>.
21. Kalaiselvi, B. and Geetha, S. (2022). Ensemble Machine Learning AdaBoost with NBTree Model for Placement Data Analysis. *2022 2nd International Conference on Intelligent Technologies* pp.1–4. doi:<https://doi.org/10.1109/conit55038.2022.9847993>
22. M Li, G. and Zhou, H. (2023). Modeling and Estimation Methods for Student Achievement Recognition Based on XGBoost Algorithm. *XGBoost Algorithm*, [online] pp.1–6. doi:<https://doi.org/10.1109/easct59475.2023.10392502>.
23. Karim, Md.R., Syeed, M.M.M., Fatema, K., Hossain, S., Khan, R.H. and Uddin, M.F. (2024). AnxPred: A Hybrid CNN-SVM Model with XAI to Predict Anxiety among University Students. *2024 IEEE 17th International Scientific Conference on Informatics (Informatics)*, [online] pp.132–137. doi: <https://doi.org/10.1109/informatics62280.2024.10900931>.
24. Hasib, K.Md., Rahman, F., Hasnat, R. and Alam, Md.G.R. (2022). *A Machine Learning and Explainable AI Approach for Predicting Secondary School Student Performance*. [online] IEEE Xplore. doi:<https://doi.org/10.1109/CCWC54503.2022.9720806>