



Student Level Prediction using Machine Learning

Dept of Computer Science Engineering

Machine Learning Module (CIS4035)



A Project Report

By

Krishna Gopal Sharma

UID: S3454618

Semester 2: 2024 – 2025

Word Count: 2623

Content

1. Abstract	3
2. Introduction	3
3. Literature Review	3
4. Methodology	4
4.1 Data Collection and Dataset Overview	4
4.2 Data Preprocessing	6
4.3 Model Selection and Training	8
4.4 Evaluation Metrics	9
4.5 Expandable AI (LIME)	9
5. Results	9
5.1 Data Preprocessing Result	9
5.2 Performance Metrics	10
5.3 ROC AUC Analysis	12
5.4 Confusion Matrix Analysis	13
5.5 Expandable AI (LIME) Result	18
6. Discussion and Conclusion of Results	21
7. List of Tables and Figures	21
8. References	22

1. Abstract

Machine learning (ML) has become increasingly important in predicting and evaluating student performance. This study examines the use of ML to predict student success in the UAE, using the "Student Level Prediction in UAE" dataset. The research compares models such as Decision Trees, Support Vector Machines (SVM), Artificial Neural Networks (ANN), and Ensemble Methods to predict performance and identify at-risk students. The findings show that ensemble models are more accurate, stable, and interpretable than traditional methods. The study also uses LIME (Local Interpretable Model-agnostic Explanations) to explain how different models make predictions and highlight key factors affecting outcomes. This research offers valuable insights to improve educational strategies, personalise learning, and better predict student success.

2. Introduction

Advancements in technology have transformed education, prompting a shift towards using data and machine learning (ML) to predict student outcomes. Traditional methods like grades and assessments often overlook factors such as behaviour and engagement. ML can analyse large datasets, offering insights into student performance and enabling early intervention for at-risk students. While models like Decision Trees and SVM are commonly used, recent advancements in ensemble learning and graph-based methods have shown promise in improving prediction accuracy, ultimately helping to personalise learning and enhance student success.

3. Literature Review

The use of machine learning (ML) in predicting student performance has gained significant attention, especially with the growing amount of educational data. This technology is being used not only to predict academic success but also to identify at-risk students early, enabling timely interventions. In diverse educational settings, such as the UAE, where multiple curriculums exist, ML is helping to address challenges like student levelling. Ghareeb et al. [2] propose a framework using ML algorithms to place students in the correct academic year, considering differences in curriculums, assessment methods, and schedules.

Qureshi and Lokhande [3], highlight the role of Educational Data Mining (EDM) in predicting student outcomes. Their study demonstrates how factors such as academic performance and personal background influence academic results. Ahmed [4] extends this by using ML models like K-means clustering and Support Vector Machines in e-learning environments, showing how student interactions with learning platforms can be used to predict success.

Chandra and Kumar [5] explore the use of ML in predicting job placement based on academic scores and skills, while Lagrazon et al. [6] apply ML to predict success in licensure exams for engineering graduates. Both studies highlight the potential of ML in forecasting future academic and career outcomes. Rimpay et al. [7] provide a broader view on EDM, focusing on early identification of struggling students, which can lead to better support and improved academic performance.

In higher education, Bird [8] and Issah et al. [9] review the effectiveness of predictive analytics, emphasizing how academic attributes and demographics are key to understanding student success. Asthana et al. [10] introduce the concept of 'Learning Coefficients' to guide interventions, while Mubarak et al. [11] use Graph Convolutional Networks to classify students

based on their engagement with course materials, offering a more personalised approach to prediction. The review by Oppong [12] and Fazil et al. [13] contribution has also a great significant in this study regarding the insight for student level predictions.

4. Methodology

This section outlines the methodology used to predict student performance, covering data collection, preprocessing, feature engineering, model selection, training, evaluation, and analysis to choose the best-performing model.

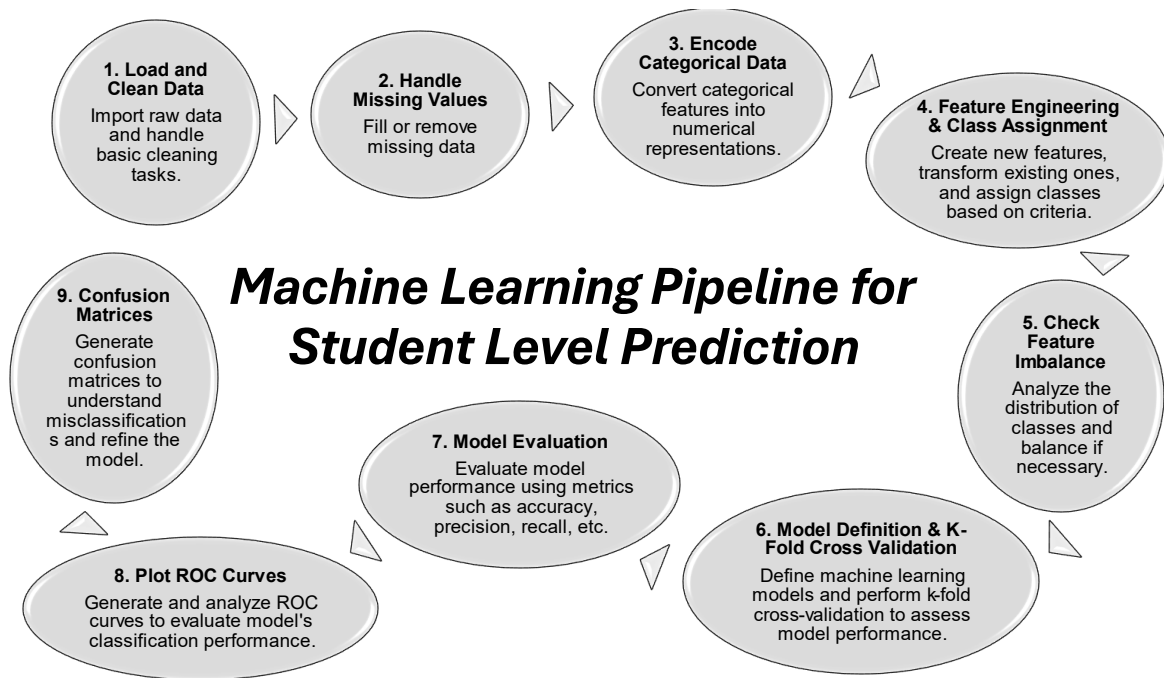


Fig 1: ML Pipeline for this project.

4.1 Data Collection and Dataset Overview

The dataset used for this study is real time primary Dataset of student level prediction in UAE. *Data in Brief*, [1]. 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. Artificial Intelligence Ethics are also considered in this dataset, consist of 2 types of data.

The categorical data encompasses student demographics (e.g., age, gender), academic records (e.g., previous and current grades), attendance, participation in online forums, assignment submission rates, and other behavioural indicators. These features help understand how various factors impact student performance.

Attribute name	Value	Description
Gender	Male/Female	Gender of the student
Student age (as of 2017/18)	6, 7, 8, 9,.....etc	Age of the student calculated from 2017/18 academic year
Year of admission	Old BBS Student, Old GEMS Student, New Admission 18/19	The data collected is for two or more academic years; 2017/18 and before academic years + 2018/19 academic year
Current year (17/18)	Foundation Stage 1, Foundation Stage 2, Year 1–12/Grade 1–11	This is the year or grade group assigned to the student by the school
Proposed year/grade (18/19)	Foundation Stage 1, Foundation 2, Year 1–13/Grade 1–12	This is the year or grade group assigned to the student by the school
Previous school (17/18)	Many schools in UAE	Previous schools that the student was in before this study
Previous curriculum	British/American/MOE/Canadian/Indian/Australian/CBSE/German	The curriculum the student transferred from
Current school	GEMS, Belvedere	Name of the school that the data has been collected
Current curriculum	Binary value	The curriculum the student transferred to

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](#). 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".

Attribute name	Value	Description
Math-exam	Mark out of 100	Exam marks for school entry exam in math
Science-exam	Mark out of 100	Exam marks for school entry exam science
English-exam	Mark out of 100	Exam marks for school entry exam English
Math19-1	Percentage out of 100%	Term 1 student Maths Exam marks during academic year 2018/19
Science19-1	Percentage out of 100%	Term 1 student science Exam marks during academic year 2018/19
English19-1	Percentage out of 100%	Term 1 student English Exam marks during academic year 2018/19
Math19-2	Percentage out of 100%	Term 2 student Maths Exam marks during academic year 2018/19
Science19-2	Percentage out of 100%	Term 2 student science Exam marks during academic year 2018/19
English19-2	Percentage out of 100%	Term 2 student English Exam marks during academic year 2018/19
Math19-3	Percentage out of 100%	Term 3 student Maths Exam marks during academic year 2018/19
Science19-3	Percentage out of 100%	Term 3 student science Exam marks during academic year 2018/19
English19-3	Percentage out of 100%	Term 3 student English Exam marks during academic year 2018/19
Math20-1	Percentage out of 100%	Term 1 student Maths Exam marks during academic year 2019/20
Science20-1	Percentage out of 100%	Term 1 student science Exam marks during academic year 2019/20
English20-1	Percentage out of 100%	Term 1 student English Exam marks during academic year 2019/20
Math20-2	Percentage out of 100%	Term 2 student Maths Exam marks during academic year 2019/20
Science20-2	Percentage out of 100%	Term 2 student science Exam marks during academic year 2019/20
English20-2	Percentage out of 100%	Term 2 student English Exam marks during academic year 2019/20
Math20-3	Percentage out of 100%	Term 3 student Maths Exam marks during academic year 2019/20
Science20-3	Percentage out of 100%	Term 3 student science Exam marks during academic year 2019/20
English20-3	Percentage out of 100%	Term 3 student English Exam marks during academic year 2019/20

Table 2: Numerical features, content values and description.

4.2 Data Preprocessing

Raw datasets often contain missing values, irrelevant features, or imbalanced classes that can affect model performance. Therefore, the following preprocessing steps were applied

- **Cleaning the Data:** The dataset underwent a cleaning process to remove issues such as spelling errors, extra whitespace, and non-alphanumeric characters. This step ensures that the dataset is unique and free from errors, allowing the models to function effectively. A Python script was used to automatically detect and correct these issues.

Column Name	Missing Values	Percentage
Science-exam	3	0.1937
English-exam	1	0.0646
English19-1	1	0.0646
Math19-2	1	0.0646
Science19-2	1	0.0646
English19-2	2	0.1291
Science19-3	1	0.0646
Math20-1	1	0.0646
English20-1	1	0.0646
Math20-2	1	0.0646

Table 3: List of Missing values in features.

- **Handling Missing Values:** Missing data is a common problem. In this study, missing numerical values were imputed using the mean of the respective column, while categorical data were imputed using the mode value (most frequent entry). Additionally, some missing values were artificially added to test the robustness of the models.
- **Feature Engineering:** This technique modifies or creates new features based on existing ones. In this study, the average marks across six terms (plus entrance marks) were calculated for each student. The average score was then categorised into two classes: Class 1 (students with averages greater than 80%) and Class 2 (students with averages below 80%). This categorisation helps in identifying students who might need extra attention or support.
- **Encoding Categorical Features:** As machine learning models require numerical input, categorical variables such as gender and year of admission were encoded using One-Hot Encoding, converting categorical data into binary vectors. This ensures that important information is retained and used efficiently in the models.

Gender	Age_as_of_Academic_Year_1718	Current_Year_1718	Proposed_YearGrade_1819	Previous_Curriculum_1718	Current_School	Current_Curriculum	Previous_yearGrade
1	4	13	13	0	0	0	0
1	4	13	13	0	0	0	0
1	4	13	13	0	0	0	0
0	4	13	13	0	0	0	0
1	3	13	13	0	0	0	0

Table 4: Categorical features values converted into numerical values

- **Normalization/Standardization:** Features with different scales were normalised using Min-Max Scaling to ensure that each feature contributes equally to the learning process.
- **Feature Selection:** To improve model accuracy, features that were most relevant to predicting student performance were selected. This was done using correlation analysis and mutual information techniques, which helped reduce the dataset to the most significant features.
- **Handling Feature Imbalance:** In the “Year_of_Admission” column, two schools' data were merged into a single category, and rows from the “New Admission 2019-20” category were removed due to the imbalance. Dimensionality reduction techniques were then applied to remove the unnecessary column.

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 supervised learning model that splits the dataset into subsets based on binary decisions, making it easy to interpret.
2. **Support Vector Machine (SVM):** A binary classification algorithm that identifies the optimal hyperplane to separate classes, useful for high-dimensional data.
3. **Random Forest Classifier:** An ensemble method that builds multiple decision trees and combines their predictions to reduce overfitting and improve accuracy.
4. **Gradient Boosting Machine (GBM):** A sequential ensemble method that builds trees in stages, each one correcting the errors of the previous one.
5. **Ensemble Learning:** A technique that combines predictions from multiple models. In this study, a Voting Classifier was used to aggregate predictions from all models and choose the final output based on the majority vote.

The models were trained using an 80/20 train-validation split. Hyperparameter optimisation techniques like Grid Search and Random Search were used to fine-tune the models and determine the best parameters.

4.4 Evaluation Metrics

Assess the performance of the models, several evaluation metrics were employed. [16] 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 total predictions. While common, accuracy alone is not always sufficient, especially in imbalanced datasets.
2. **Precision:** The proportion of true positives out of all predicted positive instances, useful for reducing false positives.
3. **Recall (Sensitivity):** The proportion of true positives out of all actual positives, important for identifying at-risk students.
4. **F1 Score:** The harmonic means of precision and recall, useful in balancing both false positives and false negatives.
5. **Confusion Matrix:** A matrix showing actual vs predicted classifications, helping to better understand model performance and errors.

4.5 Expandable AI (LIME)

This study also explores the use of Explainable AI (XAI), specifically LIME (Local Interpretable Model-agnostic Explanations). LIME helps explain individual model predictions, enhancing the transparency of machine learning models. By analysing the specific features associated with each instance, it shows their contribution to the prediction. This approach is model agnostic, meaning it can be applied to any machine learning model, including decision trees, SVMs, and neural networks. LIME's focus on local interpretability is particularly useful when it is important to understand why a model made a particular prediction for a specific student.

5. Results

In this study the models were evaluated using several performance metrics, including accuracy, precision, recall, F1 score, and ROC AUC. The models tested include Decision Tree (DT), Support Vector Machine (SVM), Random Forest (RF), Gradient Boosting Machine (GBM), and an Ensemble Learning Model. Additionally, a Graph-Based Ensemble Learning approach was explored to enhance prediction accuracy and stability.

5.1 Data Pre-Processing Result

Class Distribution in Training Data	
Class	Count
0	594
1	524
Total	1118

Table 5: Class distribution in Training data.

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

Class Distribution in Test Data	
Class	Count
0	148
1	131
Total	279

Table 6: Class distribution in Testing data.

In The testing data was also perfectly balanced. The total dataset of 1397 records were divided in a 4:1 ratio for training and testing, ensuring that class bias was removed. Initially, the data contained bias towards new students, with only 103 records for them. After removing these rows and the "Year_Of_Admission" feature, the data was balanced. As AI practitioners suggest, using larger datasets typically improves model performance, but it's essential to ensure that the data is unbiased, as biased data can adversely affect the model's predictions. Therefore, it is always necessary to remove the presence of any kind of bias.

5.2 Performance Metrics

The performance of each model was evaluated using accuracy, recall, precision, and F1-score, each of which plays a crucial role in understanding a model's effectiveness.

Model	Accuracy	F1-Score	Precision	Recall	ROC AUC
SVM	0.989	0.989	0.988	0.989	1.000
Voting Classifier	0.928	0.919	0.962	0.881	0.991
XGBoost	0.893	0.883	0.906	0.861	0.966
AdaBoost	0.870	0.858	0.882	0.838	0.952
Stacking	0.860	0.842	0.890	0.800	0.918
KNN	0.858	0.829	0.948	0.739	0.940
Random Forest	0.850	0.827	0.901	0.765	0.949
Extra Trees	0.848	0.825	0.903	0.760	0.946
ANN (MLP)	0.780	0.758	0.779	0.744	0.842
Bernoulli Naive Bayes	0.773	0.738	0.808	0.679	0.780

Table 7: Performance metrics for each model.

1. Random Forest

- Precision: 0.901
- Recall: 0.765
- F1-Score: 0.827

Random Forest performs well, with high precision (indicating fewer false positives) but moderate recall (missing 23.5% of positive cases). Its overall performance is balanced but could improve in recall.

2 ANN (MLP)

- Precision: 0.779
- Recall: 0.744
- F1-Score: 0.758

ANN performs reasonably well, but the precision and recall values are relatively low, suggesting room for improvement in capturing positive cases.

3. SVM (Support Vector Machine)

- Precision: 0.988
- Recall: 0.989
- F1-Score: 0.989

SVM excels, with high precision and recall, indicating excellent classification performance. Its ROC AUC score of 1.0 further highlights its strength in separating classes.

4. KNN (K-Nearest Neighbours)

- Accuracy: 0.858
- Precision: 0.948
- Recall: 0.739
- F1-Score: 0.829

KNN has high precision, meaning it accurately predicts positive cases but struggles to identify all true positives, reflected in the lower recall.

5. Bernoulli Naive Bayes

- Precision: 0.808
- Recall: 0.679
- F1-Score: 0.738

The performance of Naive Bayes is suboptimal, with relatively low recall, meaning it misses many positive cases. Its assumption of feature independence limits its ability to capture complex patterns.

6. AdaBoost

- Precision: 0.882
- Recall: 0.838
- F1-Score: 0.858

AdaBoost performs well, combining multiple weak learners to create a stronger model. However, it still misses some positive cases, affecting recall.

7. XGBoost

- Precision: 0.906
- Recall: 0.861
- F1-Score: 0.883

XGBoost provides solid performance with high precision and recall, making it a strong contender for classification tasks.

8. Extra Trees

- Precision: 0.903
- Recall: 0.760
- F1-Score: 0.825

Extra Trees perform similarly to Random Forest but with slightly lower recall, which affects its overall performance.

9. Stacking

- Precision: 0.890
- Recall: 0.800
- F1-Score: 0.842

Stacking, which combines multiple models, performs well overall but is not the top performer compared to SVM and Voting Classifier.

10. Voting Classifier

- Precision: 0.962
- Recall: 0.881
- F1-Score: 0.919

The Voting Classifier performs excellently, combining multiple models to achieve high precision and recall, making it one of the best-performing models

Let's summarize the models result in short:

1. SVM stands out with exceptional performance in precision, recall, and F1-score, making it the top performer.

2. Voting Classifier also performs very well due to the combination of multiple strong models.
3. XGBoost, AdaBoost, and Random Forest deliver strong results, particularly in terms of precision.
4. ANN and Bernoulli Naive Bayes have lower performance, especially in recall, indicating they are not as effective in capturing all positive cases.

5.3 ROC AUC Analysis

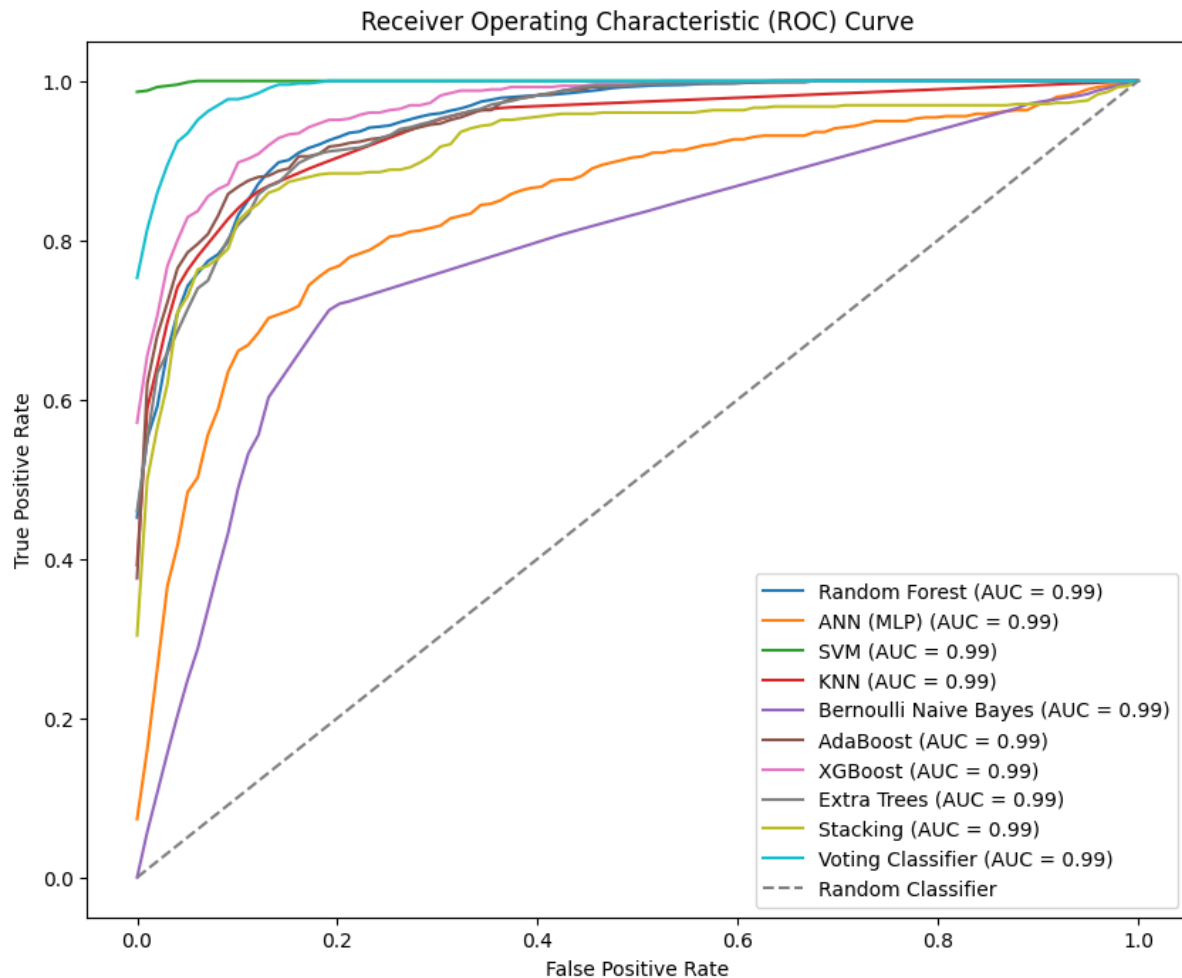


Fig 2: ROC curve for each model.

Model	SVM	Voting Classifier	XGBoost	AdaBoost	Stacking	KNN	Random Forest	Extra Trees	ANN (MLP)	Bernoulli Naive Bayes
ROC AUC	1.000	0.991	0.966	0.952	0.918	0.940	0.949	0.946	0.842	0.780

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

The ROC AUC scores indicate how effectively each model discriminates between the two classes. SVM leads with a perfect score of 1.0, followed by Voting Classifier (0.991) and XGBoost (0.966). Other models like AdaBoost (0.952), Stacking (0.918), and KNN (0.940) show strong

performance. Models like Random Forest (0.949) and Extra Trees (0.946) also perform well. However, ANN (0.842) and Bernoulli Naive Bayes (0.780) struggle to separate classes effectively.

5.4 Confusion Matrix Analysis

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

Model Name	FP	FN	TP	TN	Total
Random Forest	7	28	103	142	280
ANN (MLP)	29	25	106	120	280
SVM	1	0	131	147	279
KNN	5	30	101	143	279
Bernoulli Naive Bayes	16	38	93	133	280
AdaBoost	8	23	108	140	279
XGBoost	6	19	112	142	279
Extra Trees	6	27	104	143	280
Stacking	15	16	115	134	280
Voting Classifier	4	10	121	145	280

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.

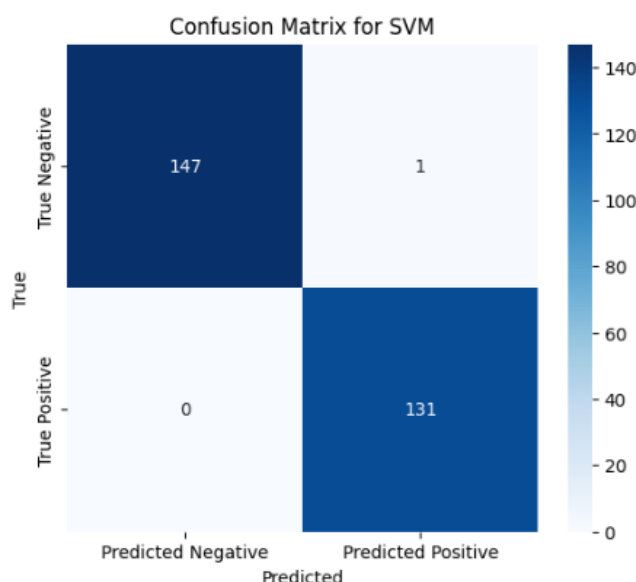


Fig 3: Confusion Matrix for SVM.

The SVM model performs excellently with minimal errors, particularly excelling in tasks with clear decision boundaries. It has a low number of false positives and no false negatives, making it ideal for high-stakes applications such as medical diagnosis. However, SVM may struggle with complex datasets and non-linear boundaries.

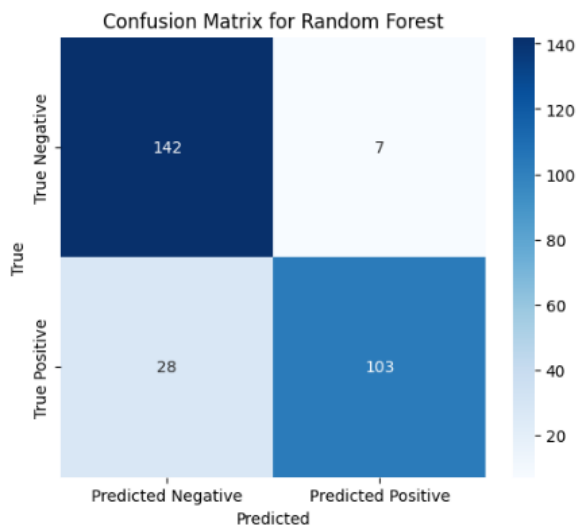


Fig 4: Confusion Matrix for Random Forest.

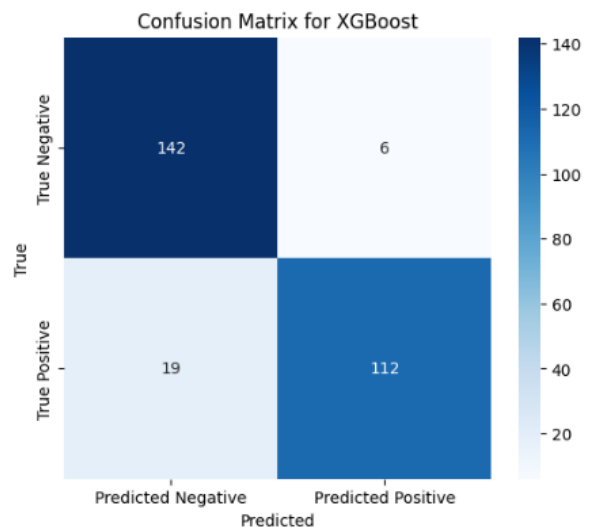


Fig 5: Confusion Matrix for XGBoost.

Random Forest demonstrates strong performance but has more false negatives, indicating it may miss some positive cases. This can be improved through hyperparameter tuning or feature engineering. XGBoost shows solid results, though it is also prone to false negatives, suggesting a cautious approach in predicting the positive class

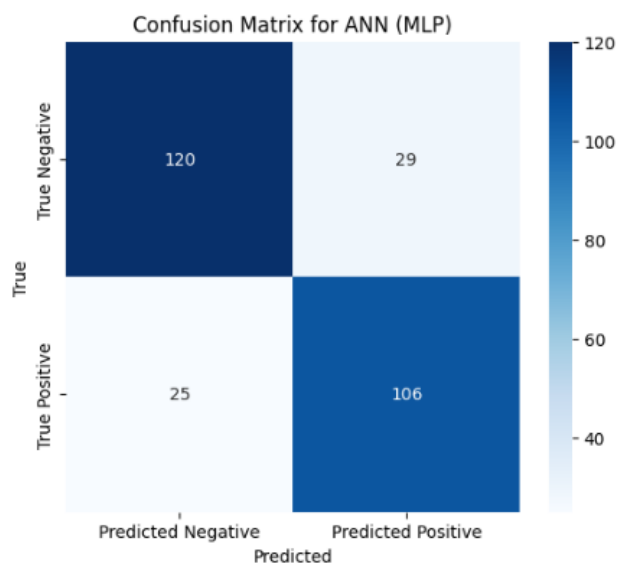


Fig 6: Confusion Matrix for ANN (MLP).

The MLP model performs similarly to Random Forest but struggles with false positives, possibly due to overfitting. Fine-tuning the model may enhance its performance.

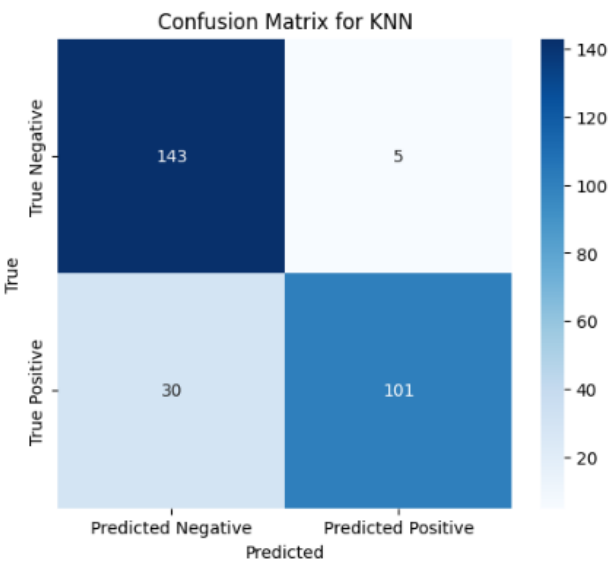


Fig 7: Confusion Matrix for KNN.

KNN has moderate performance, but its sensitivity to irrelevant features can result in poor performance, particularly in cases with complex class boundaries.

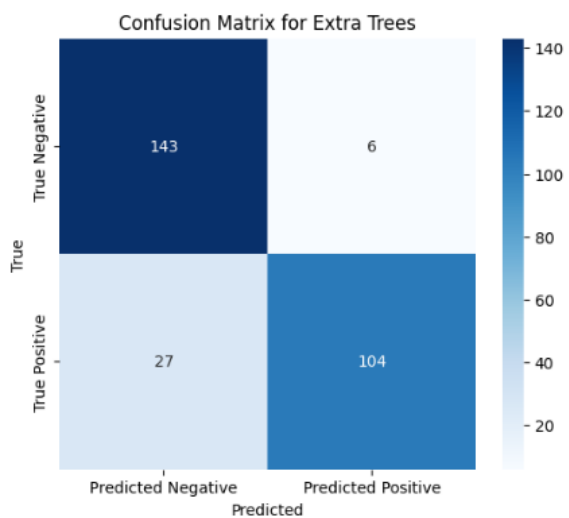


Fig 8: Confusion Matrix for AdaBoost.

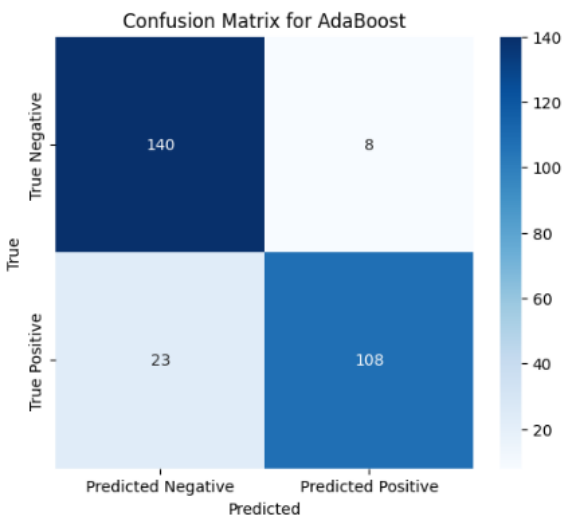


Fig 9: Confusion Matrix for Extra Trees.

AdaBoost has more false negatives, and its performance can be improved with adjustments to the learning rate. Extra Trees has a similar weakness to AdaBoost, with false negatives affecting performance.

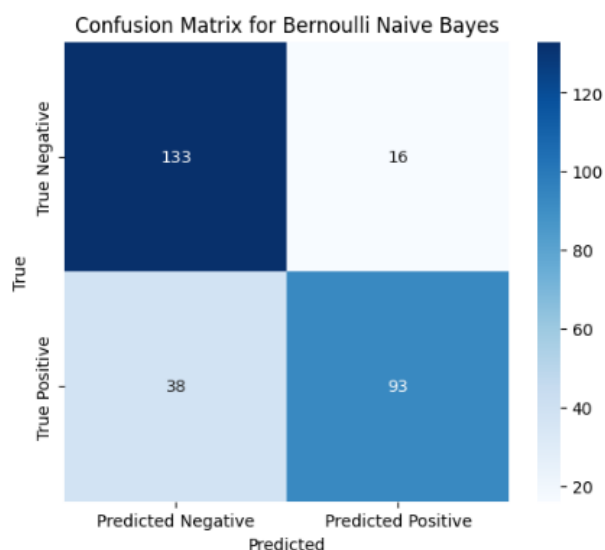


Fig 10: Confusion Matrix for Bernoulli Naive Bayes.

This model struggles with distinguishing between classes, resulting in higher false negatives.

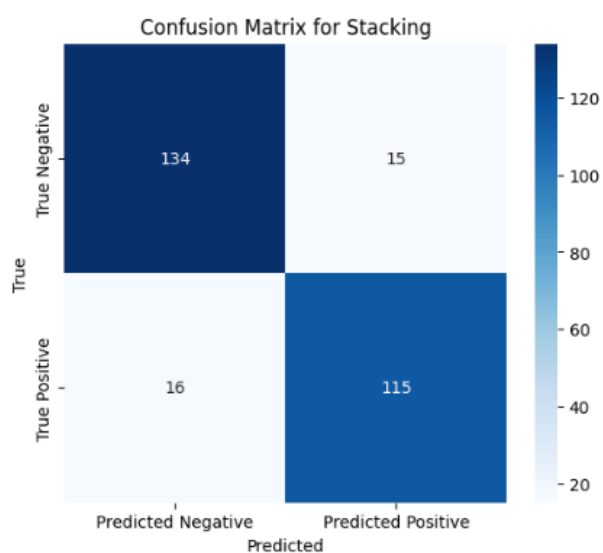


Fig 11: Confusion Matrix for **Stacking**.

Stacking combines multiple models and works well overall, but still suffers from some false positives and false negatives..

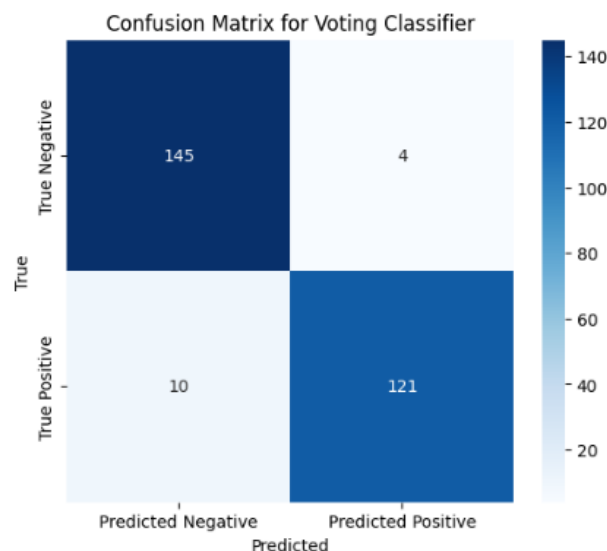


Fig 12: Confusion Matrix for **Voting Classifier**.

The Voting Classifier combines multiple base models to improve robustness but does not outperform models like SVM.

In summary, ensemble methods like SVM, Random Forest, and XGBoost generally perform better than single models such as Naive Bayes and KNN, especially in complex classification tasks.

5.5 Expandable AI (LIME) Result

The study uses the LIME (Local Interpretable Model-agnostic Explanations) technique to analyse the predictions of five popular models—SVM, Random Forest, Voting Classifier, AdaBoost, and XGBoost—on instance 219. Although the true value for this instance is 1, all the models incorrectly predict it as 0. LIME helps to uncover why each model makes this wrong prediction by highlighting the most important features that influence their decisions. This approach allows a deeper understanding of the individual model behaviours and their prediction errors for this specific instance.

```
Chosen Instance 219:
Actual Value: 1
SVM Predicted Value: 0 (Incorrect)
Voting Classifier Predicted Value: 0 (Incorrect)
AdaBoost Predicted Value: 0 (Incorrect)
Random Forest Predicted Value: 0 (Incorrect)
XGBoost Predicted Value: 0 (Incorrect)
```

Fig 13: LIME setup for instance 219.

The LIME results for each model are displayed in the corresponding figures (Fig. 14, Fig. 15, Fig. 16, Fig. 17, and Fig. 18).

1. SVM (Support Vector Machine) [\[18\]](#)

SVM		
Feature	Importance	
Math193_ <= 70.00	0.1721	
Mathexam > 84.00		-0.1861
Math202_ > 92.00		-0.1393
Science202_ <= 78.00	0.1080	
English193_ > 88.80		-0.1361
80.20 < Math191_ <= 91.00		-0.0833
English201_ > 88.40		-0.1043
83.00 < Science193_ <= 90.00		-0.0674
83.00 < Science191_ <= 89.07		-0.0720
82.70 < Science201_ <= 88.30		-0.0750

Fig 14: LIME explanation for SVM Model.

2. Random Forest [\[19\]](#)

Random Forest		
Feature	Importance	
Math193_ <= 70.00	0.1371	
0.00 < Previous_yearGrade <= 1.00		-0.1027
0.00 < Current_Curriculum <= 1.00		-0.0889
0.00 < Current_School <= 1.00		-0.0679
0.00 < Previous_Curriculum_17182 <= 1.00		-0.0571
English193_ > 88.80		-0.0409
English201_ > 88.40		-0.0375
Mathexam > 84.00		-0.0394
85.00 < English203_ <= 91.30		-0.0264
85.00 < Science193_ <= 90.00		-0.0262

Fig 15: LIME explanation for Random Forest Model.

3. Voting Classifier reference by [\[20\]](#).











Voting Classifier		
Feature	Importance	
Math193_ <= 70.00	0.1848	
English193_ > 88.80		-0.0953
Mathexam > 84.00		-0.0770
80.20 < Math191_ <= 91.00		-0.0554
0.00 < Previous_yearGrade <= 1.00		-0.0548
0.00 < Current_Curriculum <= 1.00		-0.0488
83.00 < Science193_ <= 90.00		-0.0425
0.00 < Previous_Curriculum_17182 <= 1.00		-0.0408
0.00 < Current_School <= 1.00		-0.0371
Math202_ > 92.00		-0.0362

Fig 16: LIME explanation for Voting Classifier Model.

4. AdaBoost Kalaiselvi et al. [21].











AdaBoost		
Feature	Importance	
0.00 < Current_Curriculum <= 1.00		-0.0633
Math193_ <= 70.00	0.0493	
85.00 < English203_ <= 91.00		-0.0228
Mathexam > 84.00		-0.0214
Science202_ <= 78.00	0.0256	
80.20 < Math191_ <= 91.00		-0.0174
English201_ > 88.40		-0.0167
71.00 < Science192_ <= 82.70	0.0139	
English193_ > 88.80		-0.0145
68.00 < Englishexam_ <= 75.00	0.0122	

Fig 17: LIME explanation for AdaBoost Model.

5. XGBoost











XGBoost		
Feature	Importance	
Math193_ <= 70.00	0.2644	
0.00 < Previous_Curriculum_17182 <= 1.00		-0.1710
English193_ > 88.80		-0.1002
Math202_ > 92.00		-0.0893
Mathexam > 84.00		-0.0871
Science202_ <= 78.00	0.0785	
85.00 < Science201_ <= 90.30		-0.0878
85.00 < Science193_ <= 90.00		-0.0751
English201_ > 88.40		-0.0745
76.00 < Math203_ <= 87.60	0.0666	

Fig 18: LIME explanation for XGBoost Model.

6. Discussion and Conclusion of Results

This study explores the use of machine learning models, including SVM, Random Forest, Voting Classifier, XGBoost, and others, to predict student performance and identify at-risk students. SVM performed exceptionally well in terms of precision, recall, and F1-Score, while ensemble methods like Voting Classifier and XGBoost showed strong results. LIME analysis for instance 219 revealed that different models assigned varying importance to features, influencing their predictions. These insights emphasize the need for Explainable AI in educational applications.

7. List of Tables and Figures

7.1 List of Figures

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

7.2 List of Table

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

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

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

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

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

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

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

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

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

8. 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. Rimpay, 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/iciict57646.2023.10134519>.
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/iciet59348.2024.10616372>.
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>
25. Zahra Elhamraoui. “Introduction into Deep Learning - Zahra Elhamraoui - Medium.” Medium, 27 May 2020, medium.com/@zahraelhamraoui1997/introduction-into-deep-learning-8b72bf5cd9f3. Accessed 19 Apr. 2025.