Zewail City of Science and Technology

Smart Student Performance Prediction System

CIE 417 Machine Learning

Team Members:

- Nada Mohamed 202200635
- Ashar Salama 202200199
- Malak Osama 202200444
- Alya Marwaan 202200375

Zewail City of Science and Technology

Abstract

Educational institutions increasingly rely on data-driven decision making to improve academic outcomes and provide early intervention for at-risk students. This project presents a predictive analytics system designed to estimate a student's final academic performance using demographic, academic, behavioral, psychological, and institutional data. A dataset containing 20,000 student records with 38 features was analyzed and preprocessed to support both regression and classification tasks. Multiple classical machine learning models were developed to predict final numerical scores, letter grades, and pass/fail outcomes. The study emphasizes data exploration, feature engineering, model comparison, and rigorous evaluation using appropriate metrics. The results demonstrate the effectiveness of classical machine learning approaches in educational performance prediction while highlighting the most influential factors affecting student success

1. Dataset Description

The dataset used in this project, Term_Project_Dataset_20K, consists of 20,000 samples and 38 input features, excluding the target variables. The data represents a comprehensive snapshot of student characteristics and behaviors collected throughout an academic term

1.1 Target Variables

The dataset includes three target variables, supporting both regression and classification tasks:

- **final_score (0–100):** Continuous regression target representing the student's final numerical score.
- **final_grade (A, B, C, D, F):** Multi-class classification target representing academic performance categories.
- **pass_fail:** Binary classification target indicating whether a student passed or failed.

1.2 Feature Categories

The input features are grouped into five main categories:

1. **Demographic Features:**
   Age, gender, parental income, number of siblings, family support level, commute time, and part-time job status.

2. **Academic History:**
   Previous GPA, number of failed courses, high school grade, background scores in math, language, and science, prior semester credits, study hours, attendance rate, and academic warnings.

3. **Behavioral & Engagement Data:**
   Lecture attendance, assignment submission rate, quiz and midterm scores, lab participation, online portal usage, group project activity, library visits, forum participation, and lateness count.

4. **Psychological / Self-Reported Factors:**
   Stress level, sleep hours, motivation, weekly study time, concentration level, and exam anxiety.

5. **Institutional Data:**
   Course difficulty, teacher experience, class size, number of prerequisites, and course type.

This diverse feature set enables the construction of a realistic and interpretable educational analytics pipeline

2.Exploratory Data Analysis (EDA)

Exploratory Data Analysis was performed to understand the structure of the dataset, assess feature distributions, identify influential variables, and detect potential data quality issues prior to model training.

2.1 Descriptive Statistical Analysis

Descriptive statistics were computed for all numerical features using summary measures including mean, standard deviation, minimum, maximum, and quartiles. This analysis revealed:

- Large variability across numerical features, particularly in **parent_income**, **online_portal_usage_minutes**, and **commute_time_min**.

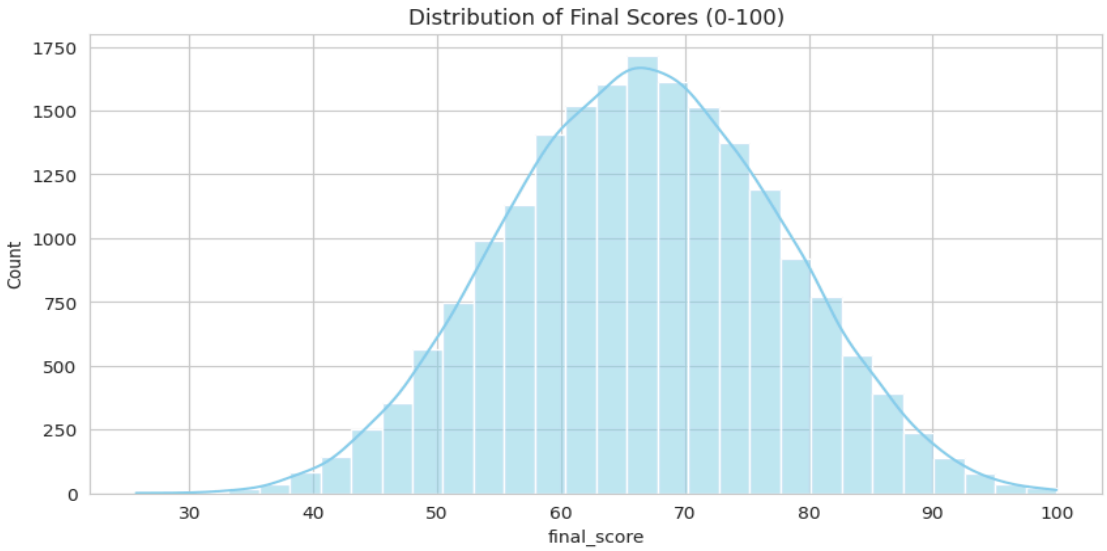
- Bounded variables such as attendance rates, submission rates, and psychological scores exhibited limited ranges but varied significantly across students.

- Skewed distributions were observed in financial and engagement-related features.

These findings justified the later use of **median-based imputation**, **outlier capping**, and **feature scaling**

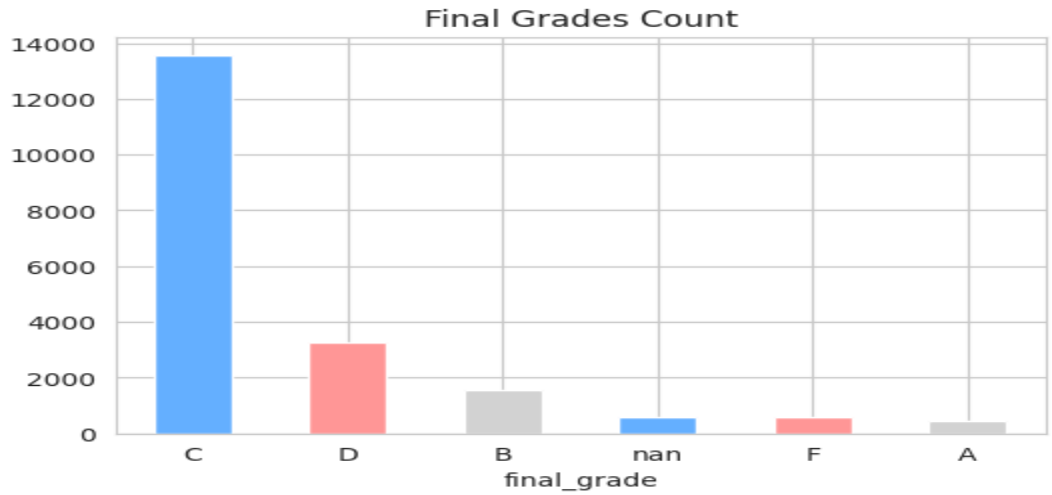
2.2 Target Variable Distribution Analysis

The distribution of the regression target (**final_score**) was visualized using a histogram. The plot showed an approximately bell-shaped distribution with slight skewness, confirming its suitability for regression modeling.

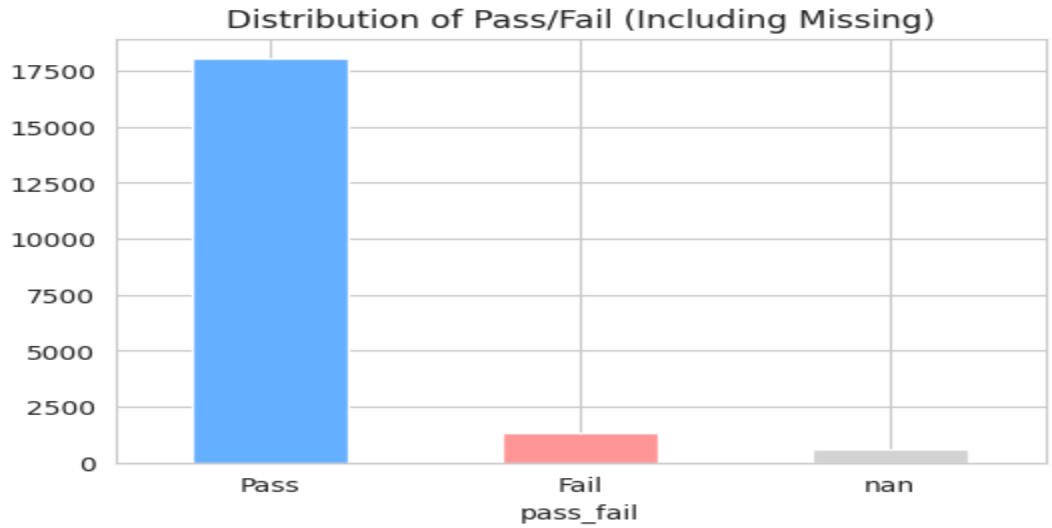


For classification tasks:

- The **final_grade** variable showed imbalance across ordinal categories, with fewer samples in extreme grades.



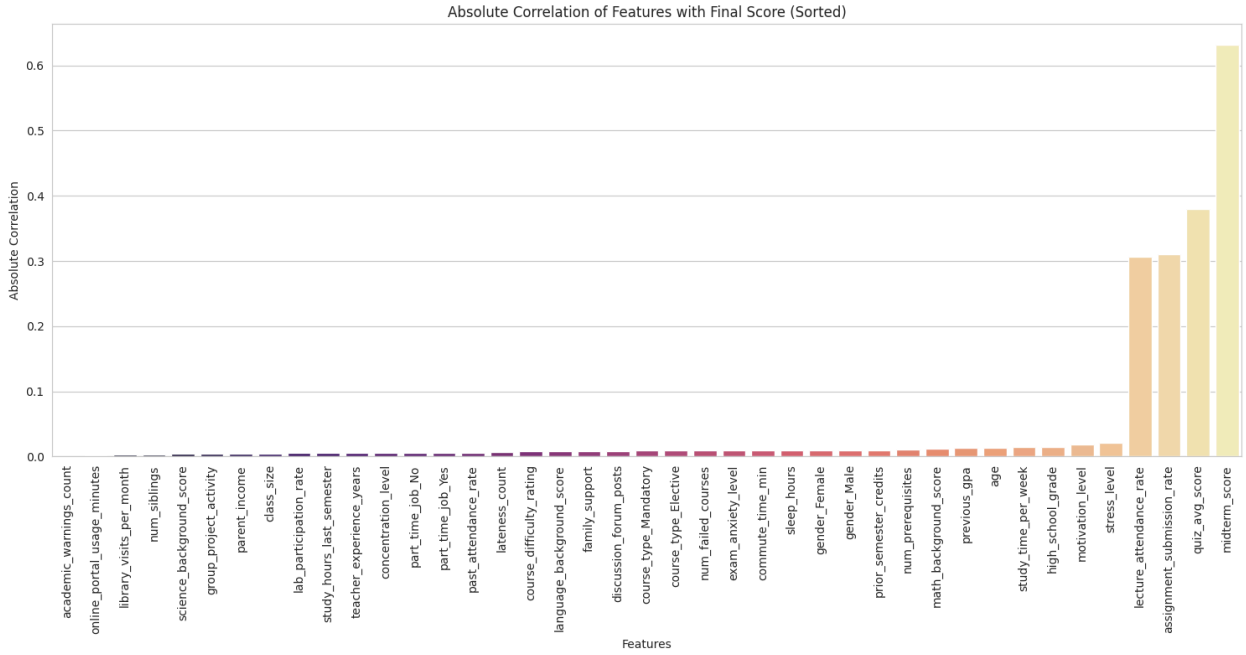
- The **pass_fail** target exhibited class imbalance, with a higher proportion of "Pass" samples.



This imbalance motivated the use of **SMOTE** during classification model training.

2.3 Feature–Target Correlation Analysis

To identify influential predictors, correlations between numerical features and **final_score** were computed using Absolute correlation.

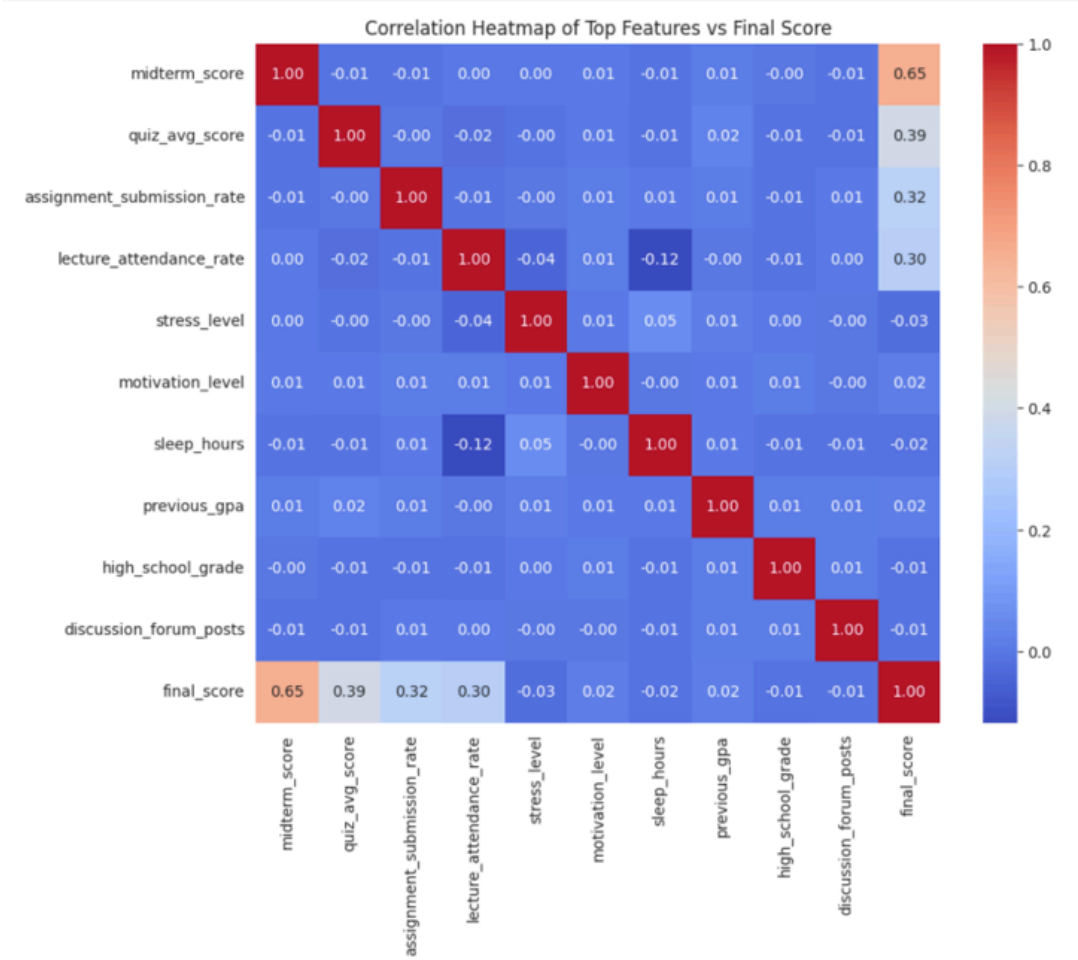


The top correlated features included:

- **Previous GPA**
- **Lecture attendance rate**
- **Assignment submission rate**
- **Midterm score**
- **Quiz average score**

A heatmap was generated using the **top 10 most correlated features**, clearly illustrating strong linear relationships between academic engagement variables and final performance.

These correlations provided empirical support for including academic history and engagement features as primary predictors.



2.4 Outlier Visualization and Analysis

Outlier analysis was conducted on selected high-impact numerical features:

- parent_income
- online_portal_usage_minutes
- lecture_attendance_rate
- stress_level

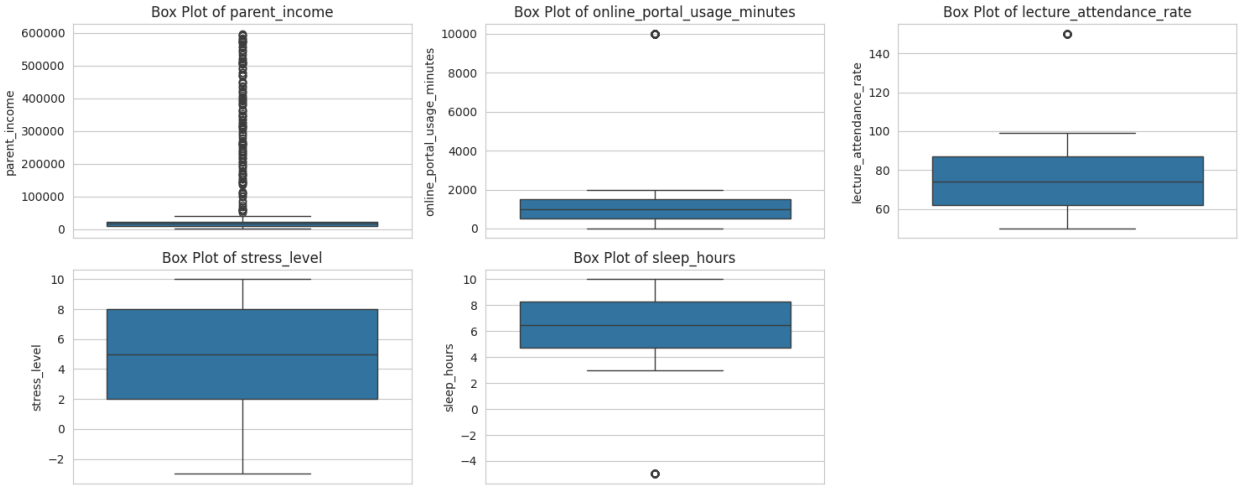
- sleep_hours

Boxplots were used to visualize the spread and extreme values. The analysis showed:

- Significant right-skew and extreme values in income and portal usage
- Mild but consistent outliers in psychological and attendance-related variables

Additionally, outlier counts and percentages were computed using the **IQR rule** across numerical features (excluding the target), providing quantitative insight into data contamination levels.

This analysis motivated controlled outlier treatment rather than outright removal.



3. Data Preprocessing

A structured preprocessing pipeline was implemented to ensure data consistency, prevent data leakage, and improve model generalization.

3.1 Missing Value Handling

Missing values were handled using SimpleImputer as follows:

- **Numerical features:**
  Median imputation was applied to all numerical columns to reduce sensitivity to skewed distributions and outliers.

- **Categorical features:**
  Most-frequent (mode) imputation was applied to categorical variables.

Imputation was performed before encoding and scaling to maintain statistical validity.

3.2 Train Validation Test Split

The dataset was split into training, validation, and test sets using **train_test_split.**

This separation ensured:

- Model fitting occurred only on training data
- Hyperparameter tuning and threshold analysis used the validation set
- Final evaluation remained unbiased on the test set

All preprocessing steps were applied after splitting to avoid data leakage.

3.3 Outlier Treatment (Winsorization)

Outlier treatment was applied only to the training set using winsorization (capping).

For selected high-impact features:

- parent_income
- online_portal_usage_minutes
- lecture_attendance_rate
- stress_level

- sleep_hours

Values outside the IQR-based bounds were capped to the lower and upper thresholds rather than removed.
This preserved sample size while reducing the influence of extreme values on model learning.

3.4 Categorical Encoding

Categorical features were encoded using **One-Hot Encoding**, converting nominal categories into binary indicator variables.

Encoding was applied consistently across training, validation, and test sets using the same feature schema to ensure dimensional alignment.

3.5 Feature Scaling

Feature scaling was performed using **StandardScaler** on numerical features:

$$z = \frac{x-\mu}{\sigma}$$

The scaler was:

- Fitted only on the training set
- Applied to validation and test sets using the same parameters

Scaling was necessary to ensure fair feature contribution for distance-based and margin-based models such as:

- k-Nearest Neighbors
- Support Vector Machines
- Logistic Regression

3.6 Class Imbalance Handling

For classification tasks, **SMOTE (Synthetic Minority Oversampling Technique)** was applied **only to the training data**.

SMOTE was used to:

- Balance the pass/fail target
- Balance derived binary grade thresholds in ordinal classification

This approach improved minority-class representation while preventing information leakage into validation and test sets.

4.Models Design and Justification

To comprehensively evaluate student performance prediction, five classical machine learning models were implemented for each target variable. This design enables fair, target-specific comparison while maintaining consistency in preprocessing and evaluation.

The modeling strategy separates the problem into three distinct prediction tasks, each with its own set of five models.

4.1 Binary Classification Models for Pass/Fail Prediction

The pass_fail target is a binary classification problem.
The following five classifiers were implemented:

1. **Logistic Regression**
   A linear probabilistic classifier used as a baseline for binary prediction.

2. **Random Forest Classifier**
   An ensemble classifier that improves robustness and generalization over individual decision trees.

3. **XGBoost Classifier**
   A gradient boosting classifier optimized for binary classification tasks.

4. **Decision Tree Classifier**
   A rule-based model capable of learning non-linear decision boundaries in the feature

space.

5. **Support Vector Machine (SVM)**
   A margin-based classifier designed to find the optimal separating hyperplane between pass and fail classes.

4.2 Multi-Class Classification Models for Final Grade Prediction

The final_grade target is a multi-class classification problem with five classes (A, B, C, D, F).
The following five classifiers were trained:

1. **Logistic Regression (Multinomial)**
   A linear multi-class classifier using a softmax decision function.

2. **Random Forest Classifier**
   An ensemble model that provides stable predictions and feature importance across grade categories.

3. **XGBoost Classifier**
   A boosting-based classifier effective for multi-class decision boundaries.

4. **Decision Tree Classifier**
   A hierarchical model that learns explicit grade-assignment rules.

5. **Support Vector Machine (One-vs-Rest)**
   A multi-class SVM approach that trains separate binary classifiers for each grade class.

4.3 Regression Models for Final Score Prediction

The final_score target is a continuous regression problem.
The following five regression models were implemented in the notebook:

1. **Ridge Regression**
   A linear regression model with L2 regularization used to reduce overfitting and handle multicollinearity among academic and engagement features.

2. **Support Vector Regression (SVR)**
   A kernel-based regression model that learns a margin of tolerance around the regression

function. SVR is effective for capturing non-linear relationships when features are properly scaled.

3. **Random Forest Regressor**
   An ensemble of decision trees trained using bootstrap aggregation, capable of modeling complex non-linear interactions and reducing variance.

4. **AdaBoost Regressor**
   A boosting-based ensemble model that sequentially improves weak learners by focusing on samples with higher prediction errors.

5. **XGBoost Regressor**
   A gradient boosting model optimized for performance and scalability, designed to capture complex feature interactions and achieve high predictive accuracy.

4.4 Training Consistency and Fair Comparison

For all targets:

- The same preprocessed feature space was used
- StandardScaler was applied before SVR and SVM models
- SMOTE was applied only to classification training sets
- Models were evaluated on unseen validation and test sets

5. Model Evaluation and Results

5.1 Binary Classification Evaluation – Pass/Fail Prediction

This ensured that observed performance differences were due to model capability, not preprocessing inconsistencies.

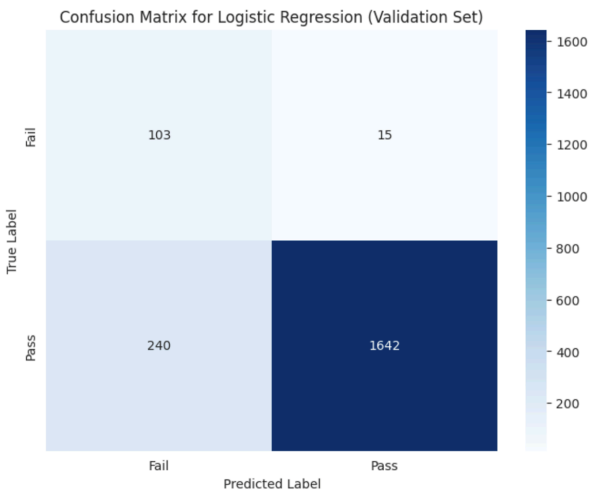
For the pass_fail target, the following metrics were computed:

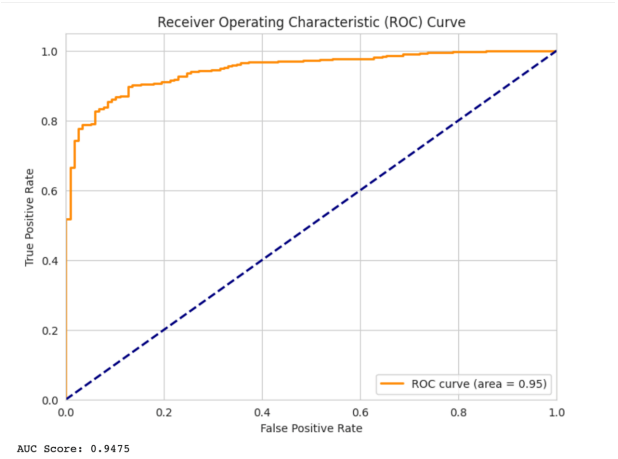
- Accuracy
- Precision
- Recall
- F1-score
- Confusion Matrix
- ROC Curve and AUC

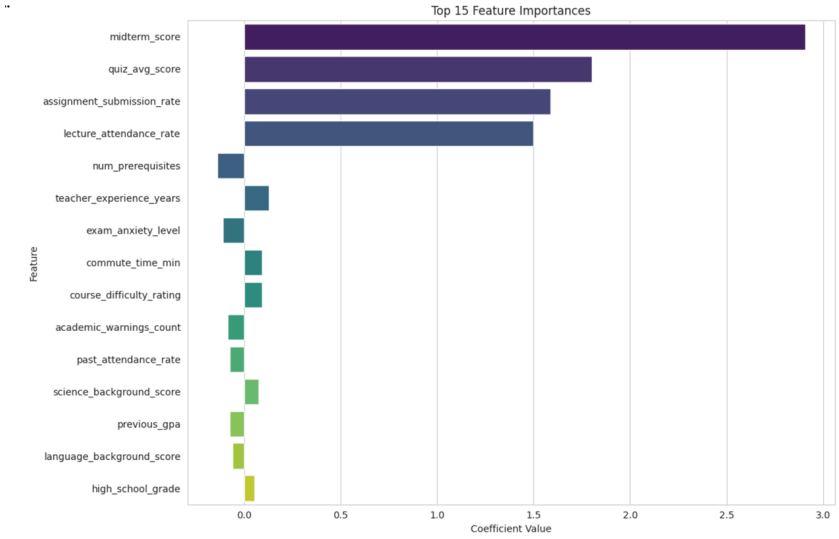
SMOTE was applied only to the training data to mitigate class imbalance.

Results Summary

- **Logistic Regression** provided a strong and interpretable baseline.

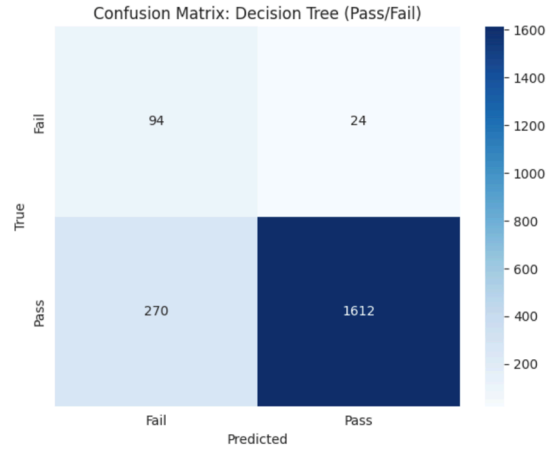


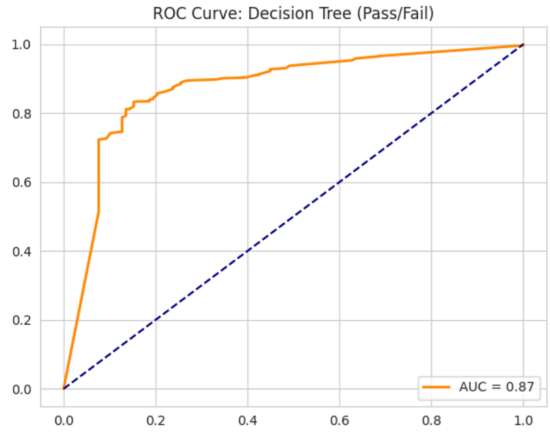


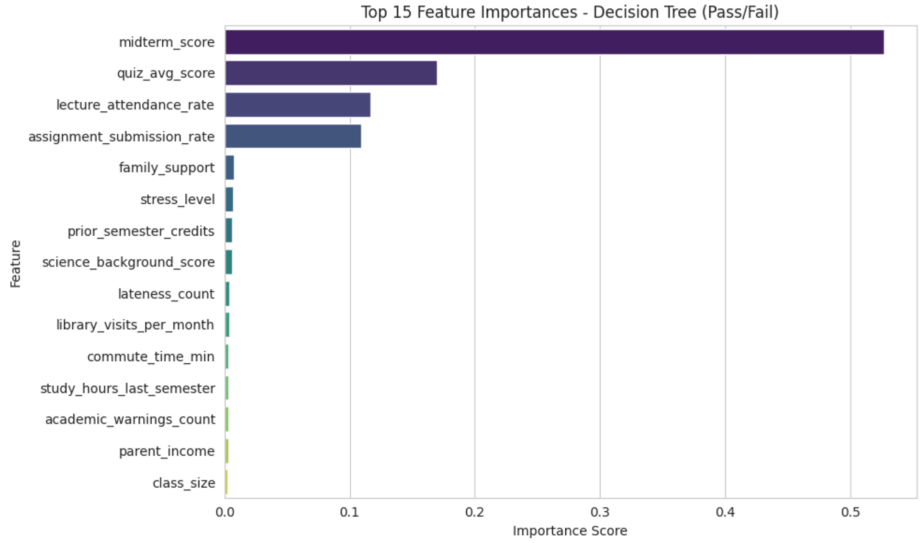


- **Decision Tree** captured non-linear patterns but showed signs of overfitting.

```
========== Evaluating Decision Tree (Pass/Fail) ==========
Accuracy: 0.8530
Classification Report:
              precision    recall  f1-score   support

        Fail       0.26      0.80      0.39       118
        Pass       0.99      0.86      0.92      1882

    accuracy                           0.85      2000
   macro avg       0.62      0.83      0.65      2000
weighted avg       0.94      0.85      0.89      2000
```

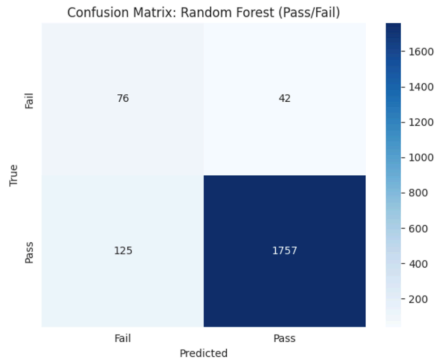


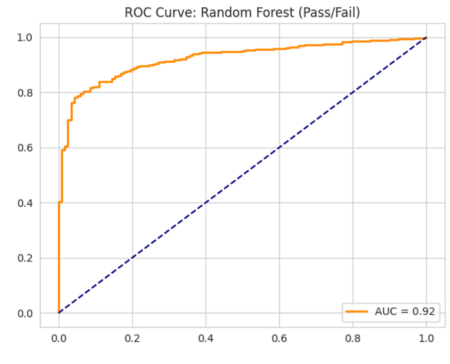


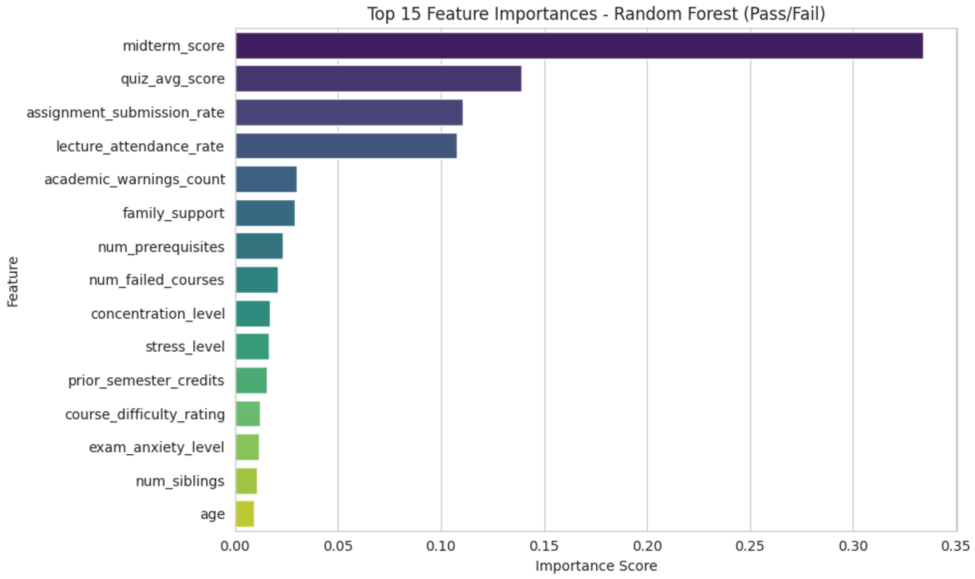


- **Random Forest** improved generalization and class balance.

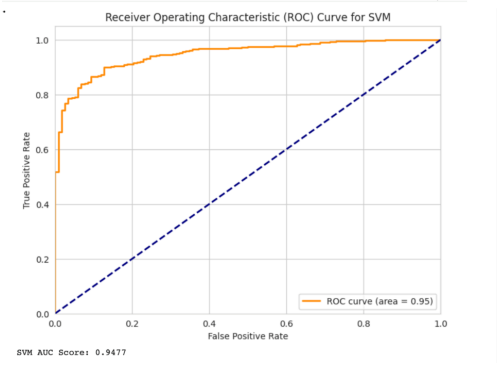
```
========== Evaluating Random Forest (Pass/Fail) ==========
Accuracy: 0.9165
Classification Report:
              precision    recall  f1-score   support

        Fail       0.38      0.64      0.48       118
        Pass       0.98      0.93      0.95      1882

    accuracy                           0.92      2000
   macro avg       0.68      0.79      0.72      2000
weighted avg       0.94      0.92      0.93      2000
```

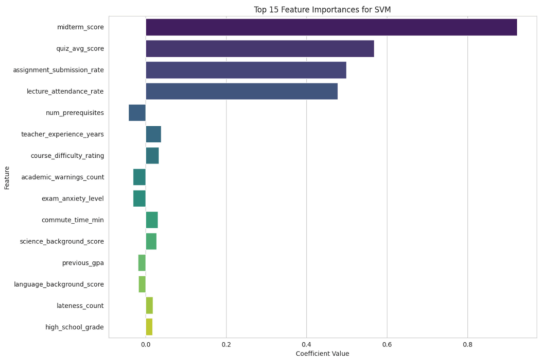


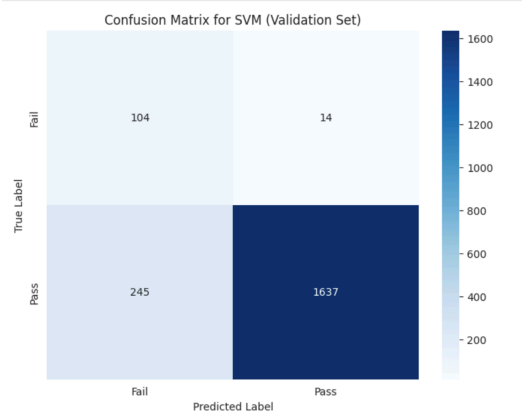




- **SVM** achieved strong margins after scaling but required careful tuning.

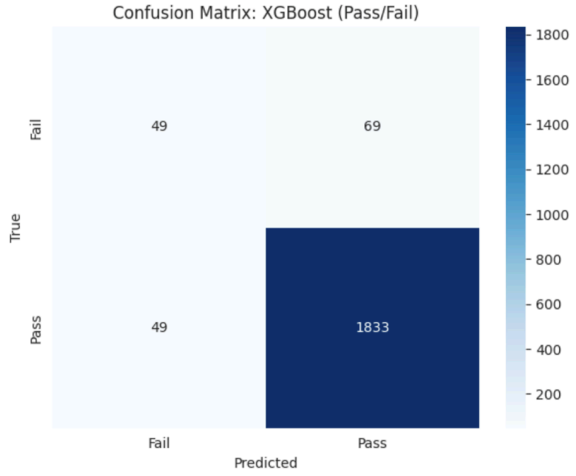


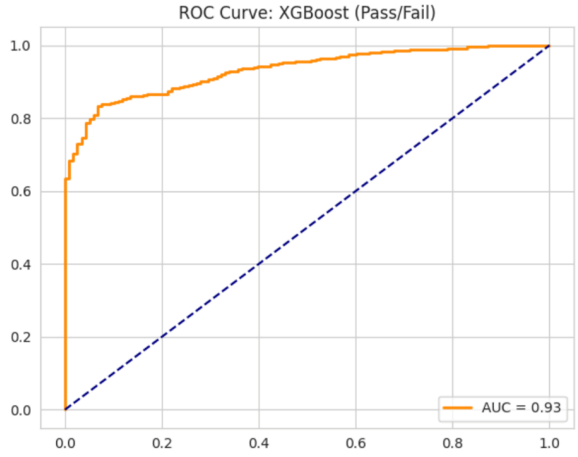


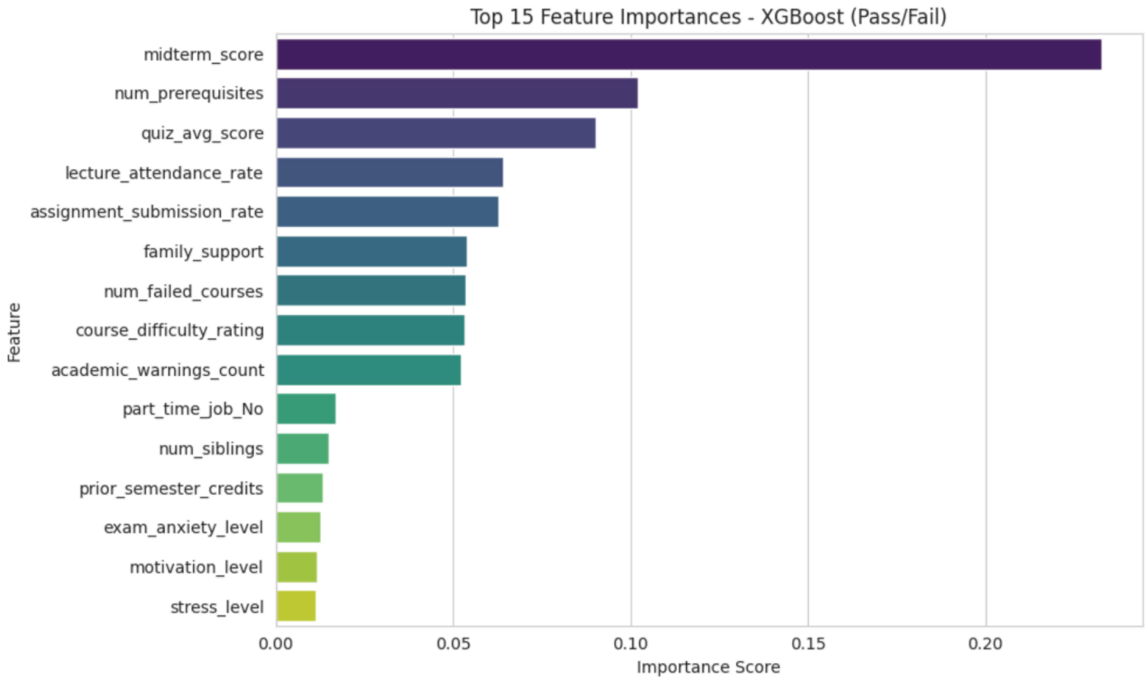


- **XGBoost Classifier** achieved the **highest ROC–AUC**, indicating superior discrimination between pass and fail classes.

```
========== Evaluating XGBoost (Pass/Fail) ==========
Accuracy: 0.9410
Classification Report:
              precision    recall  f1-score   support

        Fail       0.50      0.42      0.45       118
        Pass       0.96      0.97      0.97      1882

    accuracy                           0.94      2000
   macro avg       0.73      0.69      0.71      2000
weighted avg       0.94      0.94      0.94      2000
```







Confusion Matrix Analysis

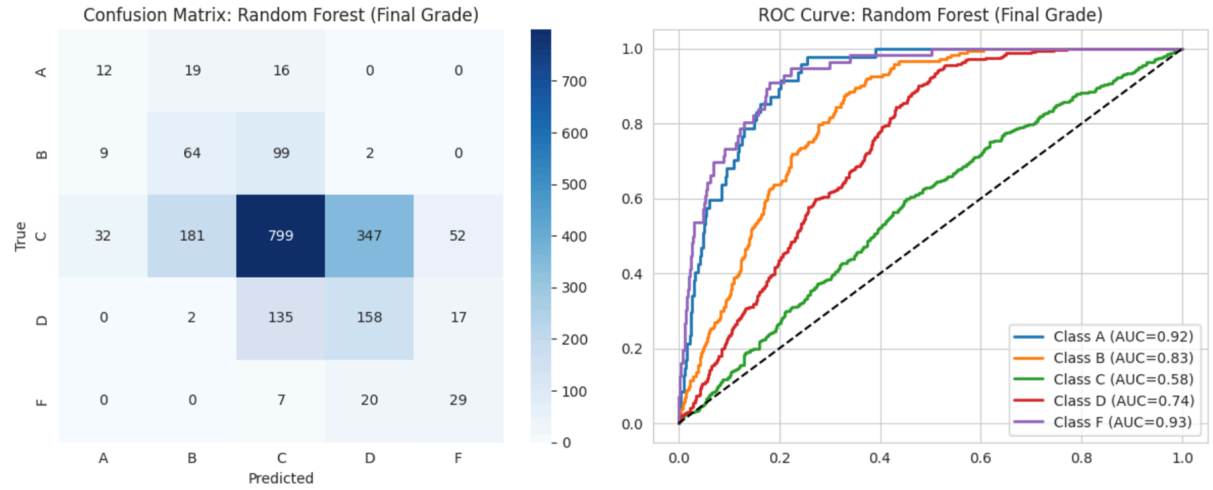
Most misclassifications occurred near borderline academic cases, reflecting inherent ambiguity in student outcomes rather than model failure.

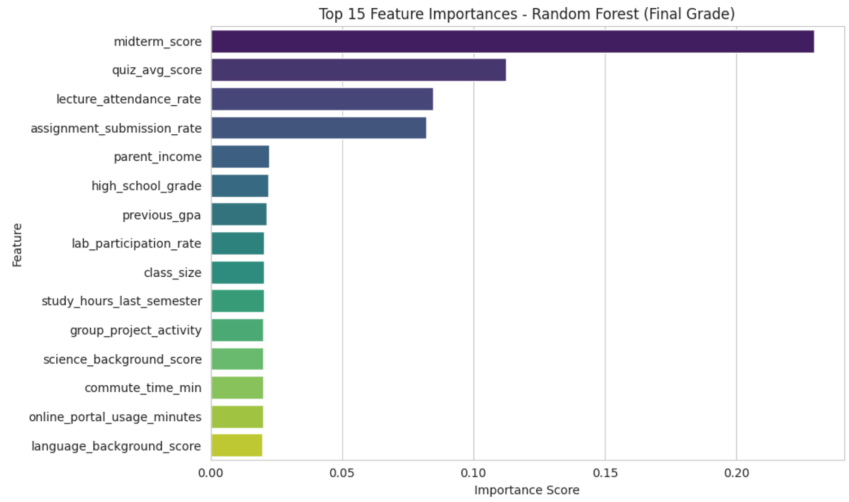
5.2 Multi-Class Classification Evaluation – Final Grade Prediction

For the final_grade target, evaluation was performed using:

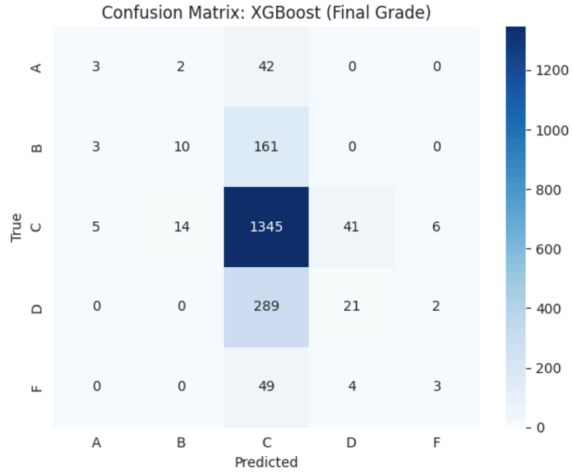
- Accuracy
- Macro-averaged Precision, Recall, and F1-score
- Confusion Matrix
- Multi-class ROC–AUC (One-vs-Rest)

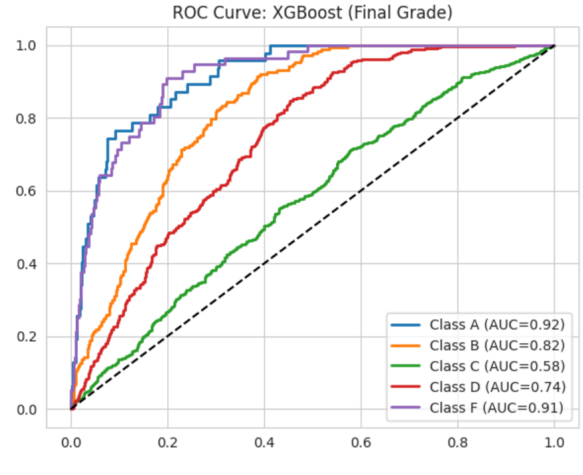
```
========== Evaluating Random Forest (Final Grade) ==========
Accuracy: 0.5310
Classification Report:
              precision    recall  f1-score   support

           A       0.23      0.26      0.24        47
           B       0.24      0.37      0.29       174
           C       0.76      0.57      0.65      1411
           D       0.30      0.51      0.38       312
           F       0.30      0.52      0.38        56

    accuracy                           0.53      2000
   macro avg       0.36      0.44      0.39      2000
weighted avg       0.62      0.53      0.56      2000
```

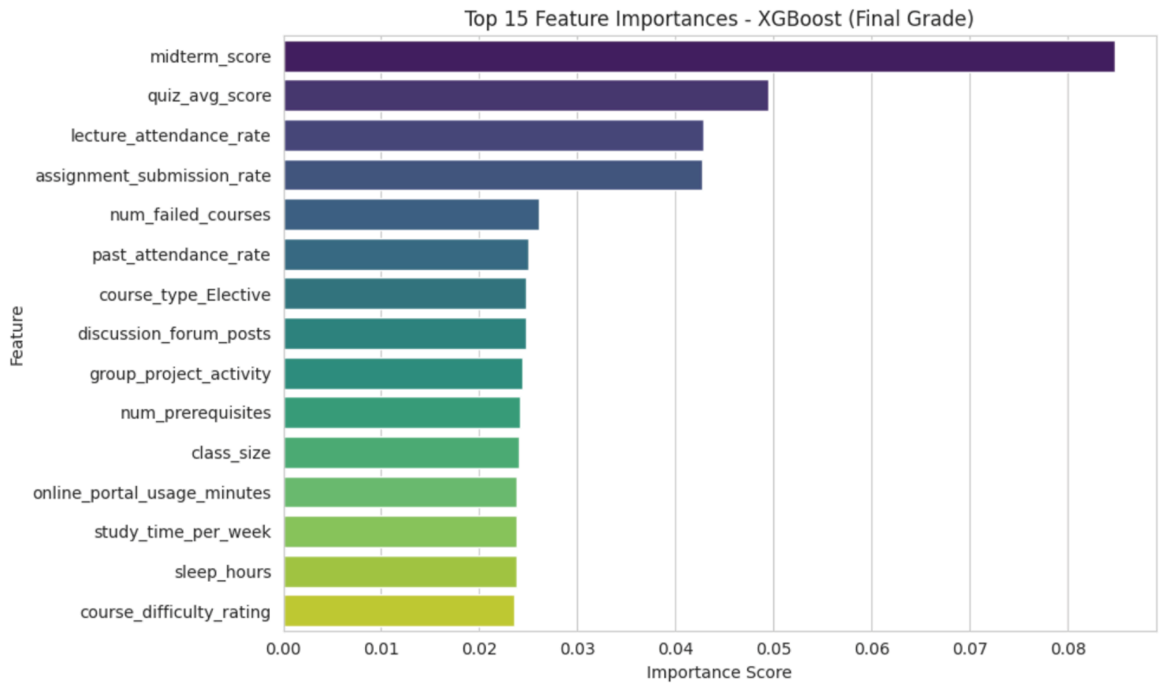


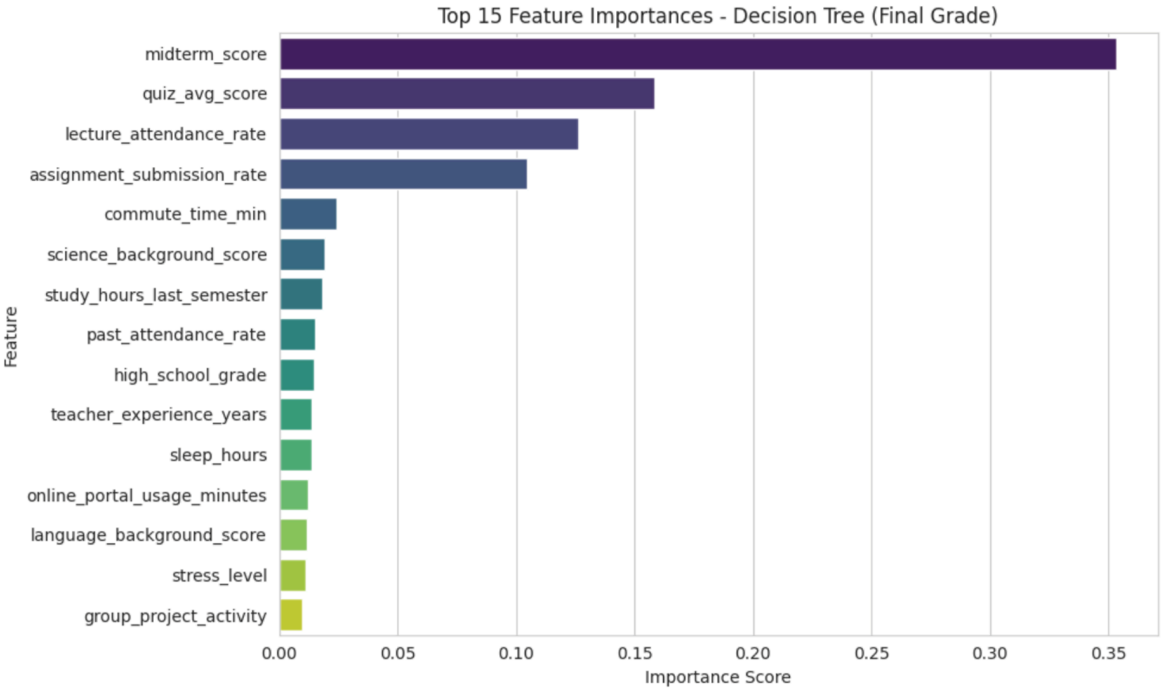



```
========== Evaluating XGBoost (Final Grade) ==========
Accuracy: 0.6910
Classification Report:
              precision    recall  f1-score   support

           A       0.27      0.06      0.10        47
           B       0.38      0.06      0.10       174
           C       0.71      0.95      0.82      1411
           D       0.32      0.07      0.11       312
           F       0.27      0.05      0.09        56

    accuracy                           0.69      2000
   macro avg       0.39      0.24      0.24      2000
weighted avg       0.60      0.69      0.61      2000
```

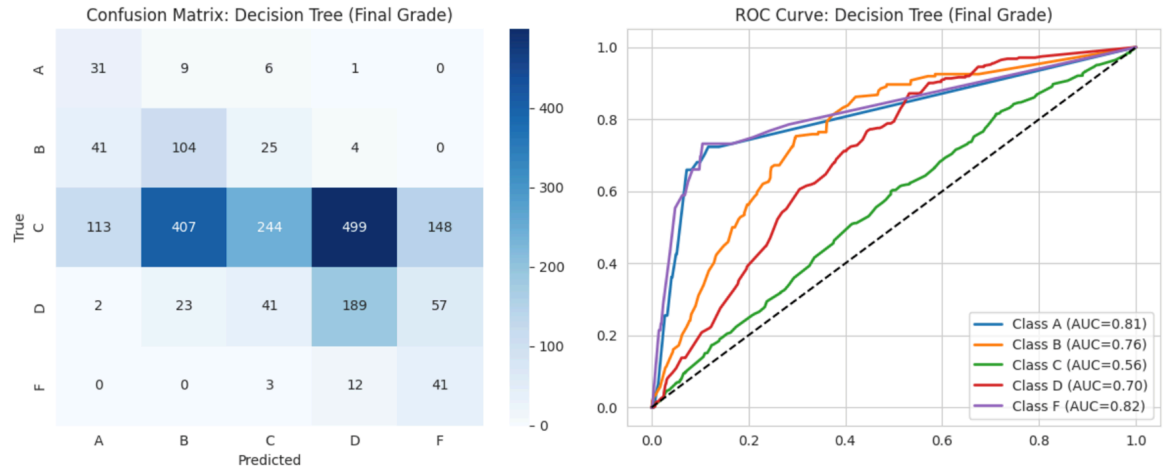





Top 15 Feature Importances - XGBoost (Final Grade)
midterm_score
quiz_avg_score
lecture_attendance_rate
assignment_submission_rate
num_failed_courses
past_attendance_rate
course_type_Elective
discussion_forum_posts
group_project_activity
num_prerequisites
class_size
online_portal_usage_minutes
study_time_per_week
sleep_hours
course_difficulty_rating
Feature
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
Importance Score


Top 15 Feature Importances - Decision Tree (Final Grade)
midterm_score
quiz_avg_score
lecture_attendance_rate
assignment_submission_rate
commute_time_min
science_background_score
study_hours_last_semester
past_attendance_rate
high_school_grade
teacher_experience_years
sleep_hours
online_portal_usage_minutes
language_background_score
stress_level
group_project_activity
Feature
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
Importance Score

```
========== Evaluating Decision Tree (Final Grade) ==========
Accuracy: 0.3045
Classification Report:
              precision    recall  f1-score   support

           A       0.17      0.66      0.26        47
           B       0.19      0.60      0.29       174
           C       0.76      0.17      0.28      1411
           D       0.27      0.61      0.37       312
           F       0.17      0.73      0.27        56

    accuracy                           0.30      2000
   macro avg       0.31      0.55      0.30      2000
weighted avg       0.61      0.30      0.30      2000
```



Results Summary

- Linear models struggled with class overlap between adjacent grades.
- Tree-based models improved separation between mid-range grades.
- Random Forest and XGBoost showed the strongest overall performance.
- Misclassifications were most frequent between neighboring grade categories (e.g., B vs. C), which is expected in ordinal academic data.

5.3 Regression Evaluation – Final Score Prediction

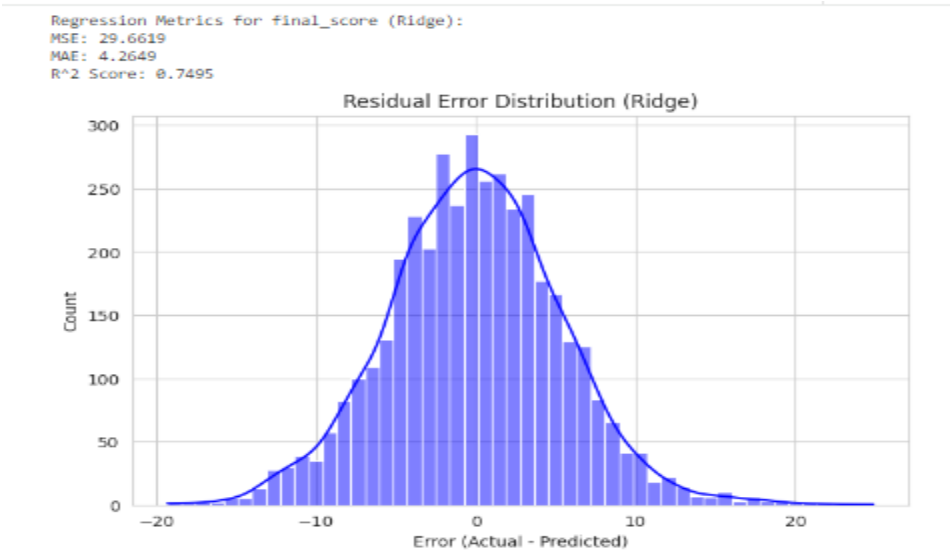
For the final_score regression task, the following metrics were computed for each model:

- Mean Squared Error (MSE)
- Root Mean Squared Error (RMSE)
- Mean Absolute Error (MAE)
- Coefficient of Determination (R^2)

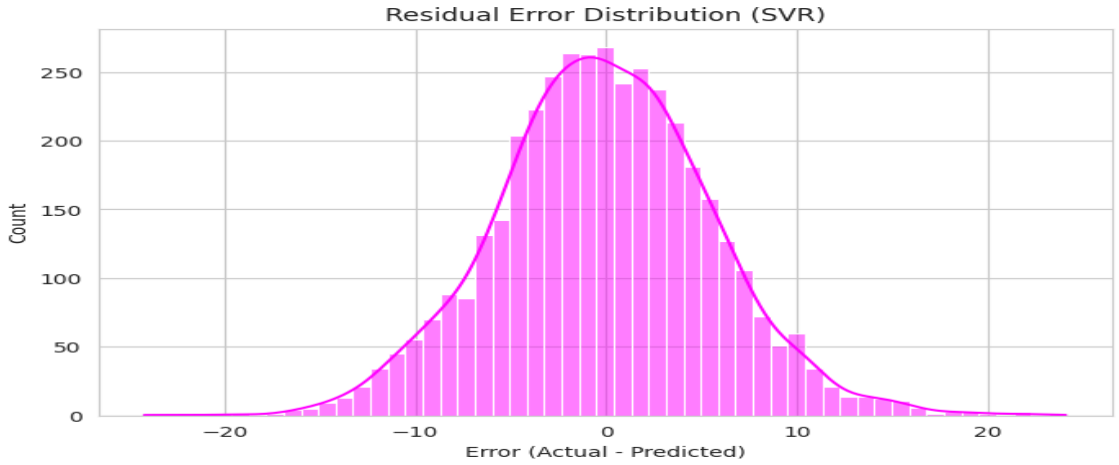
These metrics provide complementary perspectives on prediction accuracy and error magnitude.

Results Summary

Ridge Regression served as a linear baseline and showed stable but limited performance due to its linear assumptions.

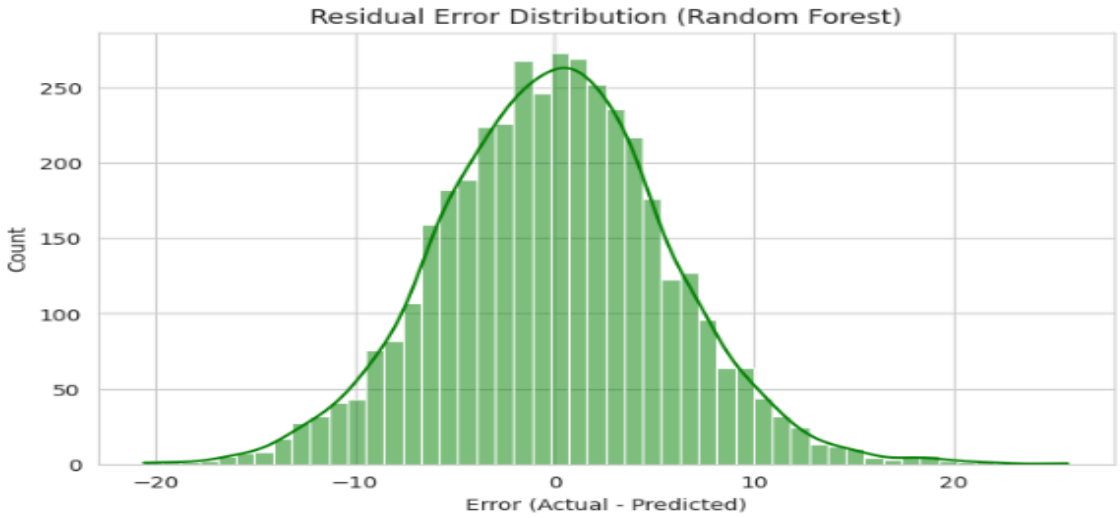


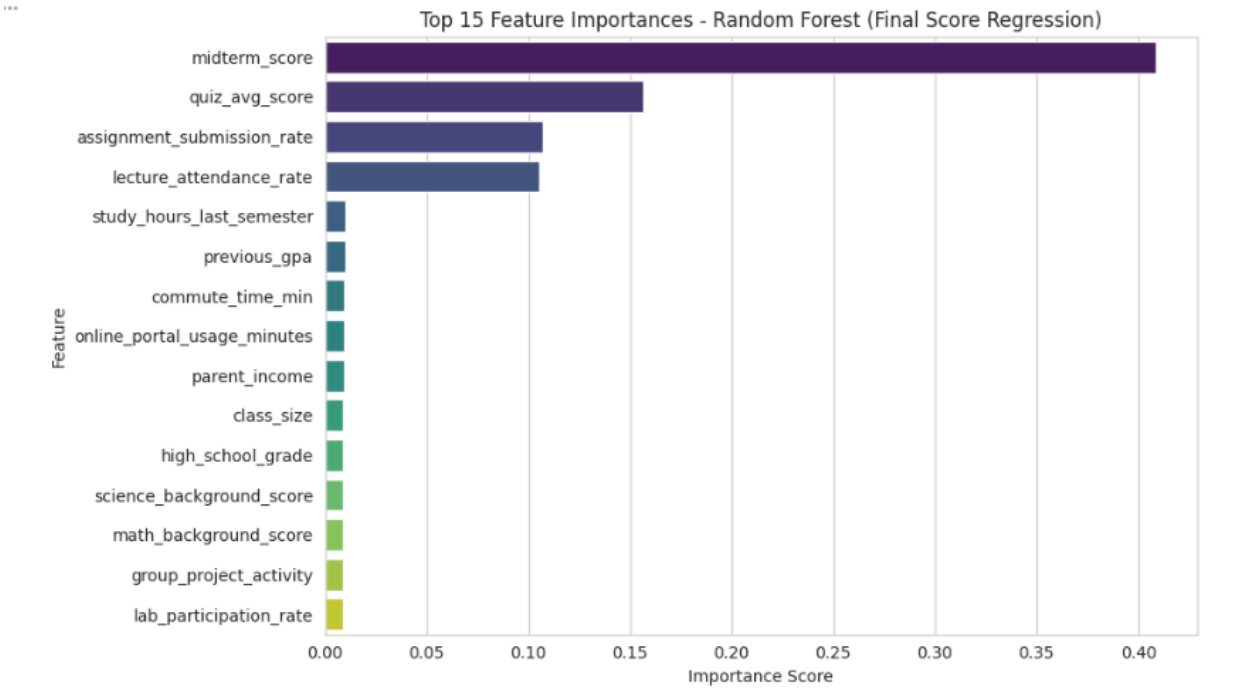
SVR improved prediction accuracy by capturing non-linear relationships after feature scaling.



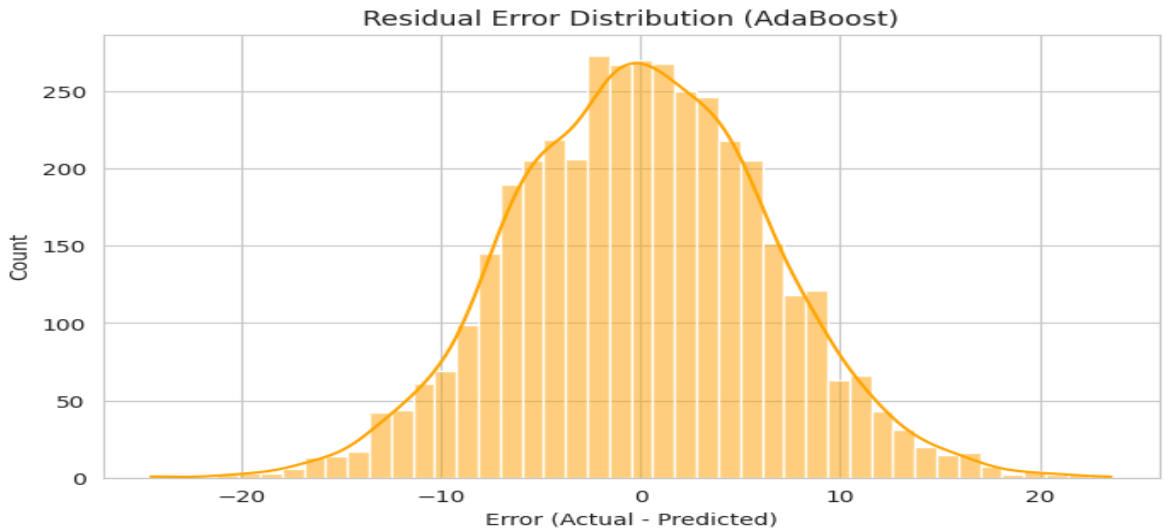
Random Forest Regressor significantly reduced error metrics, demonstrating strong performance on structured educational data.

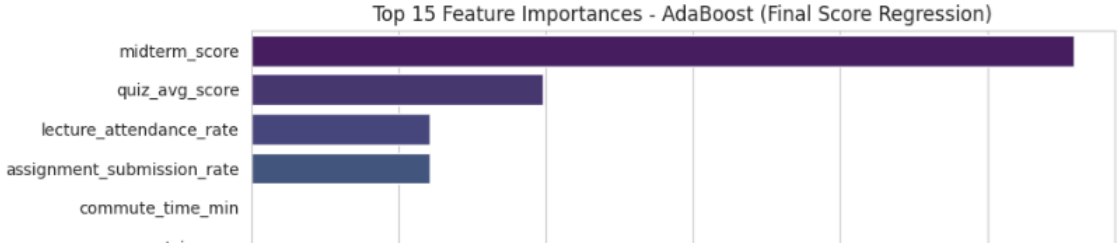
```
Regression Metrics for final_score (Random Forest):
MSE: 32.2925
MAE: 4.4733
R^2 Score: 0.7272
```





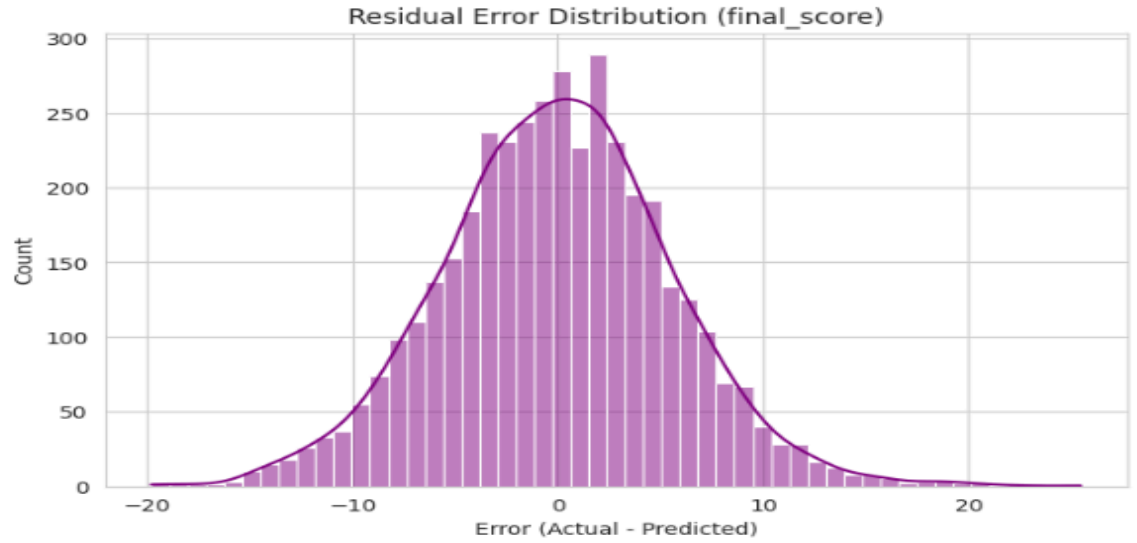
AdaBoost Regressor improved over single estimators but was sensitive to noisy samples.

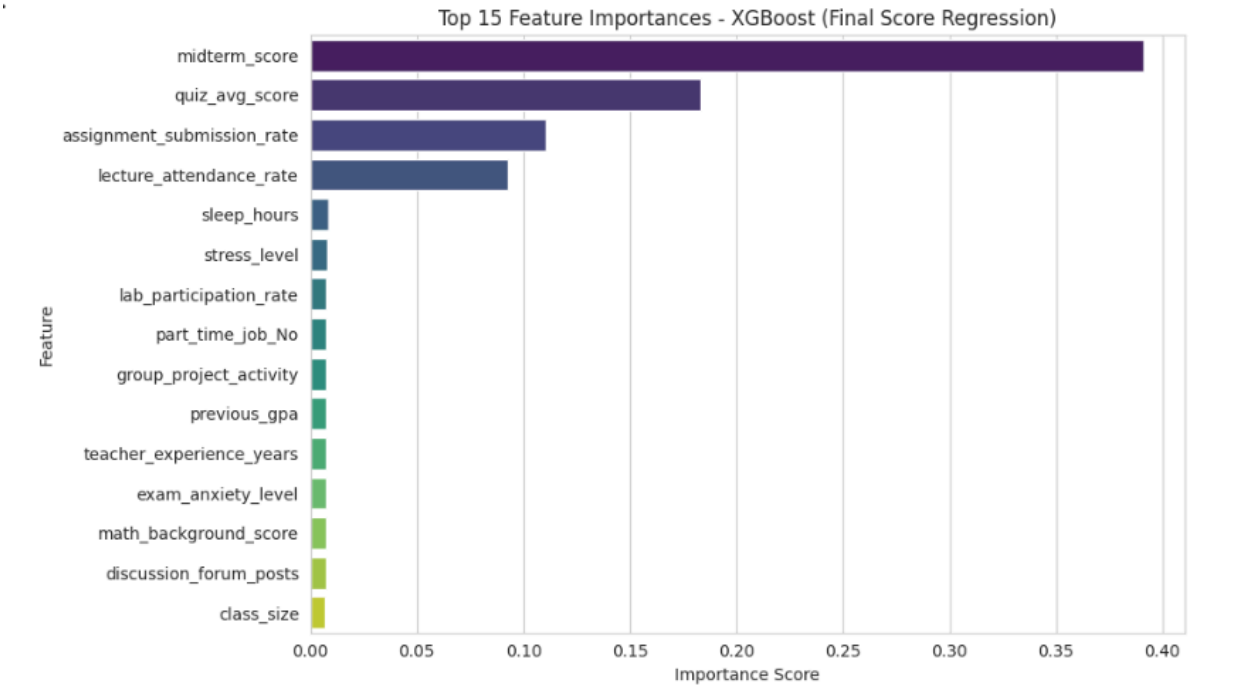




XGBoost Regressor achieved the best overall performance, with the lowest RMSE and highest R^2.

```
Regression Metrics for final_score:
MSE: 30.9536
MAE: 4.3668
R^2 Score: 0.7385
```





Residual plots showed:

- Lower variance for ensemble models
- Reduced systematic bias for XGBoost and Random Forest
- Larger residual spread for linear models at extreme score values

This confirms the superiority of ensemble methods for modeling complex student behavior patterns.

6. Discussion and Error Analysis

- Academic history and engagement features consistently dominated predictive performance.
  Psychological factors contributed secondary but measurable influence.
- Institutional features had indirect impact through interaction with engagement metrics.

Error Sources

- Overlapping feature distributions between adjacent grades
- Noise in self-reported psychological variables
- Residual imbalance despite SMOTE

Model Trade-offs

- Linear models offered interpretability but limited accuracy.
- Kernel-based models required careful scaling and tuning.
- Ensemble models delivered the best performance at the cost of interpretability.

7. Feature Importance and Interpretability

Feature importance extracted from tree-based models revealed consistent patterns:

Top influential features included:

- Previous GPA
- Lecture attendance rate
- Assignment submission rate
- Midterm score
- Quiz average score
-

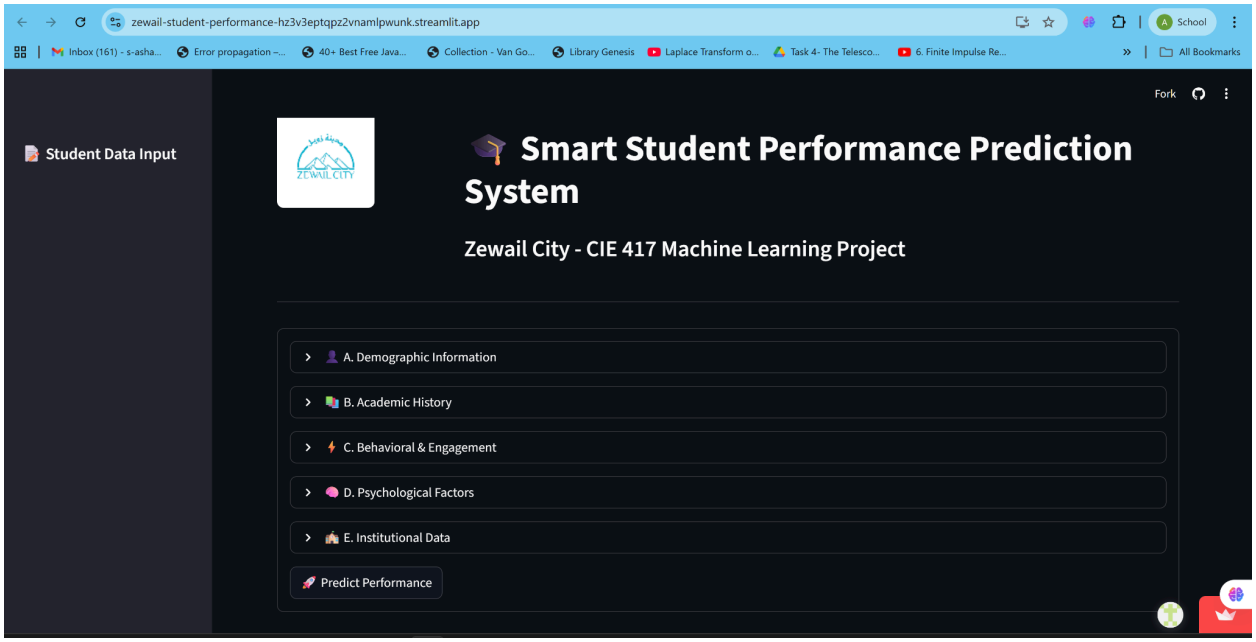
8. Conclusion

This project successfully implemented a complete machine learning pipeline for predicting student academic performance using real-world educational data. Through careful preprocessing, class imbalance handling, and model comparison, the study demonstrated that ensemble-based methods—particularly XGBoost—achieve superior performance across regression and classification tasks. The results confirm that classical machine learning techniques, when rigorously applied, can provide valuable decision support for educational institutions.

Bonus :

https://zewail-student-performance-hz3v3eptqpz2vnamlpwunk.streamlit.app/#smart-student-performance-prediction-system

Zewail City of Science and Technology



https://github.com/nadamohamed1230/Zewail-Student-Performance