Cullen College of Engineering

UNIVERSITY OF HOUSTON

INDE 6334 – Predictive Data Analytics

FINAL REPORT
PREDICTING STUDENT DROPOUT AND ACADEMIC SUCCESS USING MACHINE LEARNING

Group 11

Member Names:

Akshay Manoj	Student ID: 2154017
Nikita Anil Yadav	Student ID: 2349683
Saad Raja Muhammad	Student ID: 2313786
Shreya Sunil Telavane	Student ID: 2398529
Vamshi Krishna Madhavan	Student ID: 2344937

Submission Date: May 4, 2025

Abstract

This project aims to predict student academic performance—specifically, whether a student graduates, falls out, or remains enrolled—using machine learning techniques on demographic, academic, and socioeconomic data. The research leverages a dataset supplied by the Portuguese university, covering an end-to-end pipeline of data preprocessing, class balancing, feature selection, dimensionality reduction, model training, and interpretability analysis.

Two preprocessing techniques were tried out: one with raw integer-coded attributes and the other with One-Hot Encoding followed by class balancing through SMOTE. Recursive Feature Elimination (RFE) was used for attribute selection, whereas dimensionality reduction was done through Principal Component Analysis (PCA). Three classification methods—Logistic Regression, Random Forest, and XGBoost—were tried out, with the respective hyperparameter tuning done with GridSearchCV.

Approach 2, using One-Hot Encoding and SMOTE, was better performing than Approach 1 across all metrics. The highest scoring model was the Tuned XGBoost model in Approach 2, with an F1-score of 0.84. SHAP analysis was utilized to interpret model predictions, and the top three predictors of student success were Admission Grade, Curricular Unit Grades, and Units Approved.

The project demonstrates how machine learning can be used to effectively predict students at risk early, enabling targeted interventions that can improve retention and academic performance.

1. Introduction

Student dropout and achievement prediction is a critical role for schools to attempt to increase the retention rate as well as contribute to student success. Identification of at-risk students early on helps universities and colleges to provide interventions, individualized assistance, as well as concentrated resources to enable students to get their education to completion.

We employ machine learning methods in this project to create forecast models using demographic, educational, and socioeconomic features. By considering historical student records, we seek to accurately assign students to one of three classes: Graduate, Dropout, Enrolled.

The project involves several important steps, including data cleaning and preprocessing, handling class imbalance, dimensionality reduction, model building, hyperparameter tuning, and model interpretation. Through this integrated approach, we demonstrate how predictive analytics can assist educational institutions in making informed decisions and realizing better educational outcomes.

2. Problem Statement and Data Sources

Problem Statement

The objective of this project is to predict student academic achievement by classifying students into either one of three categories: Graduate, Dropout, or Enrolled. Using demographic, academic, and socioeconomic data, the purpose is to develop machine learning algorithms that can successfully classify at-risk students who will drop out or fail to graduate. Accurate early predictions allow educational institutions to intervene proactively, providing additional support to students who may need it most. The challenge involves addressing issues such as missing data, class imbalance, feature selection, and ensuring that the models are both accurate and interpretable for stakeholders.

The central research question addressed in this project is:

“How accurately can machine learning models predict student dropout and academic success based on demographic, academic, and economic features?”

Importance of the Analysis

Accurately predicting student dropouts and academic success enables institutions to provide early interventions and improve retention rates. By applying machine learning to demographic, academic, and socioeconomic data, this analysis helps uncover key risk factors and supports data-driven decision-making to better assist students.

Objective of the Project

The objective of this project is to develop machine learning models that can accurately predict student academic outcomes classifying them as Graduate, Dropout, or Enrolled based on demographic, academic, and socioeconomic factors.

To achieve this, the project focuses on:

- Performing data cleaning and preprocessing, including handling missing values, label encoding, scaling, and one-hot encoding
- Addressing class imbalance using the Synthetic Minority Over-sampling Technique (SMOTE)
- Selecting important features using Recursive Feature Elimination (RFE)
- Applying dimensionality reduction using Principal Component Analysis (PCA)
- Building and evaluating classification models: Logistic Regression, Random Forest, and XGBoost
- Assessing model performance using metrics such as Accuracy, F1-score, and ROC AUC
- Interpreting model predictions through SHAP analysis and ROC curves

Data Description

Source of Data

The dataset used in this project is the Predict Students’ Dropout and Academic Success dataset, obtained from the UCI Machine Learning Repository. It provides information about students from a Portuguese higher education institution, covering their demographic, academic, and socioeconomic backgrounds.

Columns Overview

The dataset consists of 37 input features describing various student attributes:

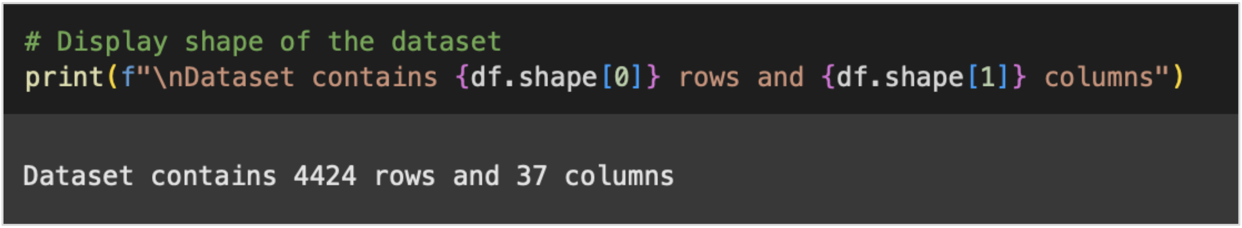
- Demographic information (e.g., Gender, Age, Marital Status, Nationality)
- Academic performance (e.g., Admission Grade, Units Enrolled, Units Approved)
- Socioeconomic factors (e.g., Unemployment Rate, GDP, Scholarship Status)

The target variable is categorical and includes three possible outcomes:

- Graduate
- Dropout
- Enrolled

Basic Dataset Summary

- Total Records: 4,424 student records
- Number of Features: 37 (including target variable)
- Target Variable Classes: Graduate, Dropout, Enrolled



```
# Display shape of the dataset
print(f"\nDataset contains {df.shape[0]} rows and {df.shape[1]} columns")
```

```
Dataset contains 4424 rows and 37 columns
```

Figure 1: Displaying the total number of records (4424) and columns (37) in the dataset

An initial exploration of the dataset showed the presence of class imbalance, with fewer dropout cases compared to students who graduated or remained enrolled. Handling this imbalance was a critical part of the preprocessing phase to ensure fair and accurate model training.

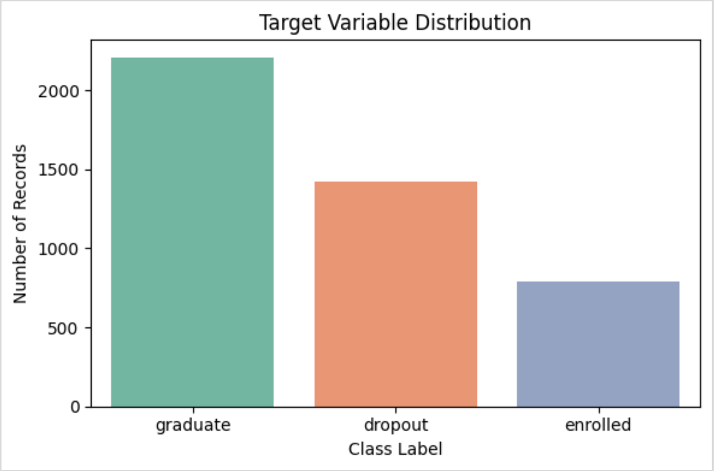


Figure 2: Distribution of the Target Variable showing class imbalance between Graduate, Dropout, and Enrolled categories

3.Methodology

Data Preprocessing

Handling Missing Values

An initial check for missing values was conducted across all features. The analysis confirmed that there were no missing values in the dataset, so no imputation or further handling was required.

Data Cleaning

After handling missing values, further data cleaning steps were performed:

- All feature names were reviewed to ensure consistency and clarity.
- Outliers and unusual values were checked visually during Exploratory Data Analysis (EDA).
- Any inconsistencies identified during initial exploration were addressed to ensure clean and ready-to-use data for modeling.

This ensured that the input data was reliable and free of major anomalies that could impact model performance.

Encoding

Categorical features were encoded to make them suitable for machine learning models:

- Label Encoding was applied to binary categorical variables (e.g., Gender, Daytime/Evening Attendance).
- One-Hot Encoding was applied to non-ordinal categorical features with more than two categories, creating new binary columns for each category.

Additionally, continuous features were scaled using StandardScaler to normalize the feature values, ensuring that models sensitive to feature scaling (e.g., Logistic Regression) would perform better.

These encoding and scaling steps prepared the dataset for effective model training across all selected algorithms.

Preprocessing Variations Across Approaches

Two preprocessing strategies were explored in this project:

- Approach 1: Used the dataset's existing integer-coded categorical features directly without applying One-Hot Encoding. SMOTE was then applied to balance the class distribution.
- Approach 2: Applied One-Hot Encoding to categorical features before applying SMOTE and model training, ensuring that the categorical nature of the data was fully captured.

Exploratory Data Analysis (EDA)

Distribution of Features

To better understand the dataset, the distribution of key features was analyzed through various plots and summary statistics.

- Numerical features such as Admission Grade, Age at Enrollment, and GPA-related attributes were plotted using histograms to observe their spread and central tendency. These distributions are illustrated in Figure 3 in the Appendix.
- Categorical features like Gender, Marital Status, Daytime/Evening Attendance, and Scholarship Holder were visualized using bar plots to examine their frequency distributions across different categories. These distributions are illustrated in Figure 4 in the Appendix.
- Special attention was given to the target variable (Graduate, Dropout, Enrolled) to verify the degree of class imbalance.

The visual analysis revealed that:

- Some numeric features (e.g., Admission Grade) were approximately normally distributed.
- Class imbalance was evident, with fewer Dropout cases compared to Graduate and Enrolled.

Feature Comparison Across Student Outcomes

- Admission Grade: Graduated students generally had higher admission grades compared to dropouts and enrolled students.
- Previous Qualification Grade: Students who graduated had slightly better previous academic qualifications.
- Curricular Units 1st Semester Grade: Graduated students achieved higher 1st semester grades compared to other groups.
- Curricular Units 2nd Semester Grade: Graduated students also performed better in the 2nd semester, indicating consistent academic success.

These academic performance features were important indicators, differentiating successful students from those at risk of dropping out or remaining enrolled. These distributions are illustrated in Figure 5 in the Appendix.

This analysis helped in understanding feature redundancy and later informed dimensionality reduction decisions like Principal Component Analysis (PCA).

Key Observations

From the exploratory data analysis, the following key insights were noted:

- Class Imbalance: The number of Dropouts was significantly lower than Graduates or Enrolled students, highlighting the need for SMOTE during preprocessing.
- Academic Performance: Students with higher Admission Grades and better Curricular Unit performance were more likely to Graduate.
- Demographic Patterns: Most of the students attended daytime classes and a higher proportion were male.
- Feature Redundancy: High correlation among several academic performance metrics indicated that dimensionality reduction techniques like PCA could improve model performance without significant loss of information.

These observations helped guide subsequent feature engineering and model development steps.

Feature Engineering

To improve model performance and reduce computational complexity, feature engineering techniques were applied to the dataset. The goal was to select the most relevant features and reduce redundancy among variables while preserving critical information.

Recursive Feature Elimination (RFE)

Recursive Feature Elimination (RFE) was employed to determine the most significant features that contribute to predicting student performance. RFE recursively removes less important features based on model,performance until the best subset of features is chosen.

Using RFE with Logistic Regression model as the base estimator, 20 important features were selected from the original set. This reduced feature set helped eliminate irrelevant or redundant information and allowed the models to give importance to the most predictive variables.

Principal Component Analysis (PCA)

Principal Component Analysis (PCA) was also applied as an alternate dimensionality reduction technique. PCA transforms the initial features into fewer uncorrelated components that maintain the majority of the variance of the data.

After PCA usage, the dimensionality was minimized while maintaining a majority of the dataset's variance. PCA indeed reduced the complexity of the model input and maintained important patterns within the data, although it introduced some interpretability concerns compared to RFE-applied features.

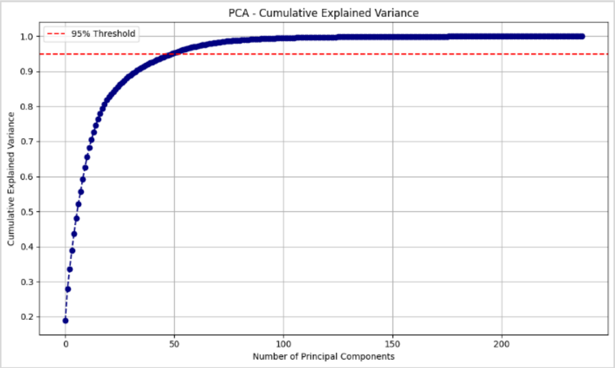


Figure 6: Cumulative Explained Variance Plot for Principal Component Analysis (PCA)

Model Building

Models Implemented

Several machine learning models were implemented to classify students into Graduate, Dropout, or Enrolled categories based on demographic, academic, and socioeconomic features.

- Logistic Regression: A Initially applied for exploratory purposes, but was not effective for this multi-class classification problem.
- Random Forest: A Random Forest Classifier was trained using the selected features. Random Forest is an ensemble model that constructs multiple decision trees and averages their predictions, providing robustness against overfitting and high variance.
- XG Boost: An XGBoost Classifier was implemented due to its strong performance with structured tabular data. XGBoost builds additive decision trees in a gradient boosting framework, optimizing model accuracy.
- Tuned Random Forest: To address class imbalance in the target variable, SMOTE (Synthetic Minority Over-sampling Technique) was applied before training the Random Forest model. SMOTE generated synthetic samples for minority classes, leading to a more balanced training dataset and improved model fairness.
- Tuned XG Boost: Similarly, XGBoost was trained after applying SMOTE. This approach aimed to combine the benefits of class balancing with the powerful predictive capabilities of the XGBoost algorithm.

Hyperparameter Tuning

The final selected hyperparameters for each approach are summarized in the table below:

Approach	Model	Hyperparameters
Approach 1	Logistic Regression	Default settings (no tuning)
	Random Forest Classifier	n_estimators=100, max_depth=None, min_samples_split=2
	XGBoost Classifier	n_estimators=100, max_depth=3, learning_rate=0.1
	Tuned Random Forest	n_estimators=200, max_depth=20, min_samples_split=2
	Tuned XGBoost	n_estimators=200, max_depth=6, learning_rate=0.1
Approach 2	Logistic Regression	Default settings (no tuning)
	Random Forest Classifier	n_estimators=100, max_depth=None, min_samples_split=2
	XGBoost Classifier	n_estimators=100, max_depth=3, learning_rate=0.1
	Tuned Random Forest	n_estimators=200, max_depth=20, min_samples_split=2
	Tuned XGBoost	n_estimators=200, max_depth=6, learning_rate=0.1

To optimize model performance, hyperparameter tuning was performed using GridSearchCV for Random Forest and XGBoost models. Different parameter combinations were evaluated based on weighted F1-score using 3-fold cross-validation.

4. Analysis and Results

Model Evaluation and Results

This section presents the evaluation of all models built under both approaches.

Performance was assessed using Accuracy, Precision, Recall, and F1-Score. Confusion matrices and ROC curves were also analyzed to understand model performance in more detail. The ROC curves and confusion matrices for the models are provided in the Appendix for reference.

Approach 1: Results

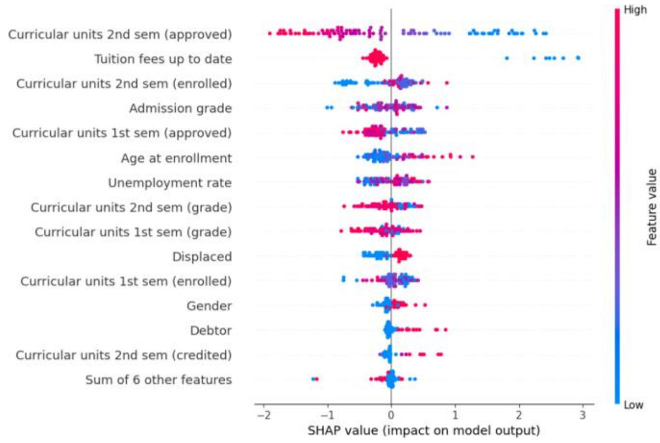
Performance Metrics

Model	Accuracy	Dropout	Enrolled	Graduate
Logistic Regression	0.76	0.78	0.41	0.86
Random Forest Classifier	0.78	0.79	0.47	0.85
XGBoost Classifier	0.74	0.74	0.40	0.85
Tuned Random Forest	0.77	0.78	0.44	0.85
Tuned XGBoost	0.76	0.77	0.43	0.86

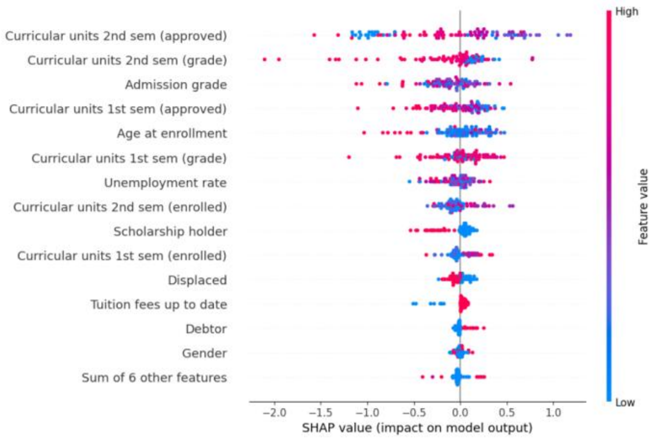
SHAP Analysis

To enhance the interpretability of the best-performing models, SHAP (SHapley Additive exPlanations) analysis was conducted. SHAP values help explain the contribution of each feature towards the model's prediction for each class.

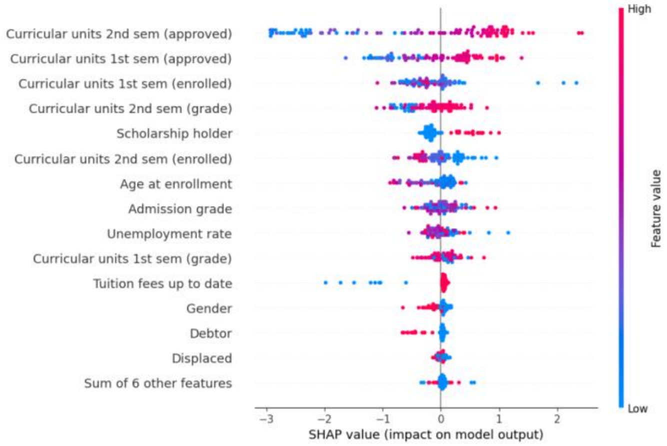
Separate SHAP plots were generated for each class:



Class 0: Graduate



Class 1: Dropout



Class 2: Enrolled

Key Observations from SHAP Analysis:

- For Class 0 (Graduate): Features such as Admission Grade, Curricular Units Approved, and Curricular Unit Grades had the highest positive contributions towards predicting graduation.
- For Class 1 (Dropout): Lower Admission Grades, fewer Units Approved, and lower Curricular Grades were major indicators associated with dropout predictions.
- For Class 2 (Enrolled): Features were more spread out, and current enrollment status and pending evaluations played a more important role.

The SHAP plots clearly illustrated how specific academic features influenced the model's classification decisions, making the model more transparent and trustworthy. The SHAP analysis uncovered a degree of uncertainty and suggested that the model was acquiring certain erroneous patterns. This prompted us to implement Approach 2 utilizing one-hot encoding, leading to more dependable feature attributions and enhanced model interpretability and performance.

Approach 2: Results

Performance Metrics

Model	Accuracy	Dropout	Enrolled	Graduate
Logistic Regression	0.77	0.76	0.71	0.82
Random Forest Classifier	0.82	0.83	0.81	0.82
XGBoost Classifier	0.83	0.84	0.80	0.84
Tuned Random Forest	0.82	0.82	0.81	0.83
Tuned XGBoost	0.84	0.84	0.82	0.85

Best Model Discussion

Based on the F1-scores and overall classification metrics:

- Tuned XGBoost from Approach 2 achieved the best weighted F1-Score.
- One-Hot Encoding combined with SMOTE and hyperparameter tuning contributed to superior performance in Approach 2.
- Approach 2 consistently outperformed Approach 1 across all models, especially after hyperparameter tuning.

Thus, Tuned XGBoost (Approach 2) was selected as the best-performing model for predicting student outcomes.

5. Conclusion

This project focused on predicting student academic outcomes—Graduate, Dropout, or Enrolled—using machine learning models applied to demographic, academic, and socioeconomic features.

Two different preprocessing approaches were explored:

- Approach 1: Using original integer-encoded categorical features.
- Approach 2: Applying One-Hot Encoding before SMOTE balancing.

A variety of machine learning models were implemented, including Logistic Regression, Random Forest, and XGBoost, with additional hyperparameter tuning using GridSearchCV for Random Forest and XGBoost models.

Through detailed evaluation using Accuracy, Precision, Recall, F1-Score, confusion matrices, ROC curves, and SHAP analysis, it was observed that:

- Approach 2 consistently outperformed Approach 1 across all models.
- Tuned XGBoost from Approach 2 achieved the best performance, demonstrating the benefit of better encoding and tuning.

Academic performance indicators, particularly Admission Grade and Curricular Unit Grades, were the most influential factors in predicting student outcomes, as revealed through SHAP analysis.

Overall, the project demonstrates that careful preprocessing, class balancing, and hyperparameter optimization can significantly improve the predictive power and interpretability of machine learning models for educational analytics.

6. Bibliography

[1] Dua, D., & Graff, C. (2017). UCI Machine Learning Repository: Student Performance Data Set. University of California, Irvine. Retrieved from https://archive.ics.uci.edu/ml/datasets/Student+Performance

[2] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, ´E. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830. Retrieved from https://scikit-learn.org

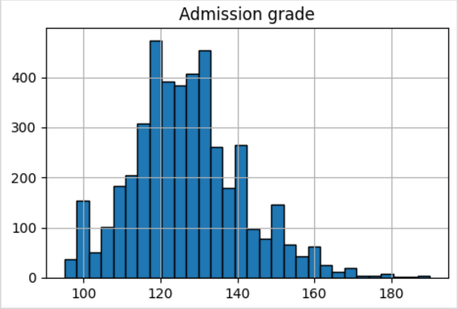
[3] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794). Retrieved from https://xgboost.readthedocs.io

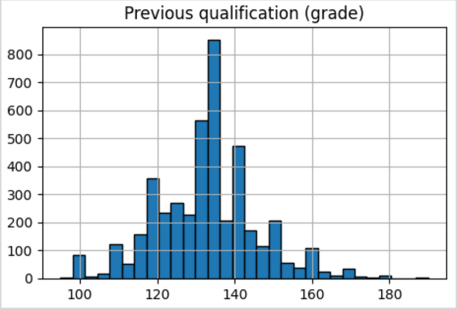
[4] Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In Advances in Neural Information Processing Systems, 30. Retrieved from https://shap.readthedocs.io

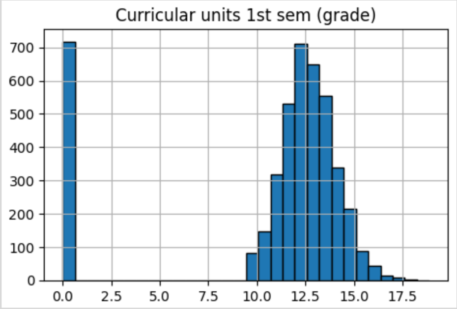
[5] Lemaitre, G., Nogueira, F., & Aridas, C. K. (2017). Imbalanced-learn: A Python toolbox to tackle the curse of imbalanced datasets in machine learning. Journal of Machine Learning Research, 18 (17), 1–5. Retrieved from https://imbalanced-learn.org

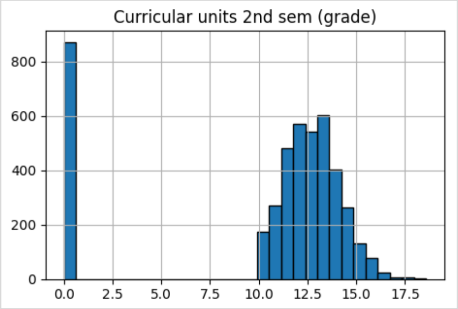
7. Appendix

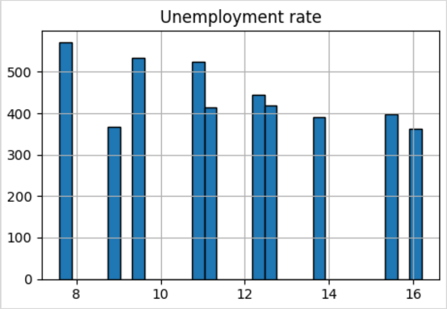
Data Visualizations:

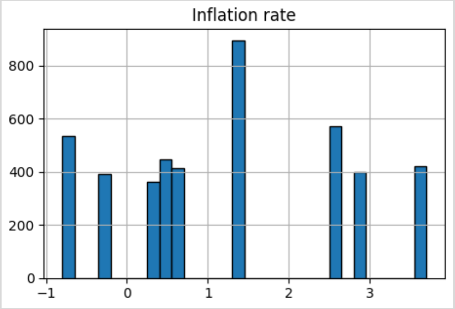












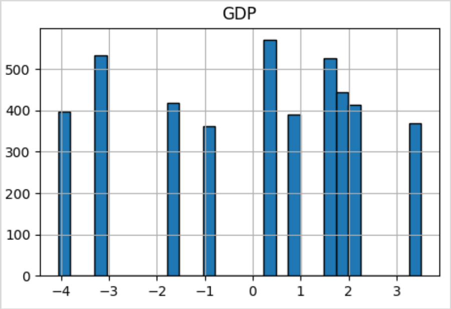
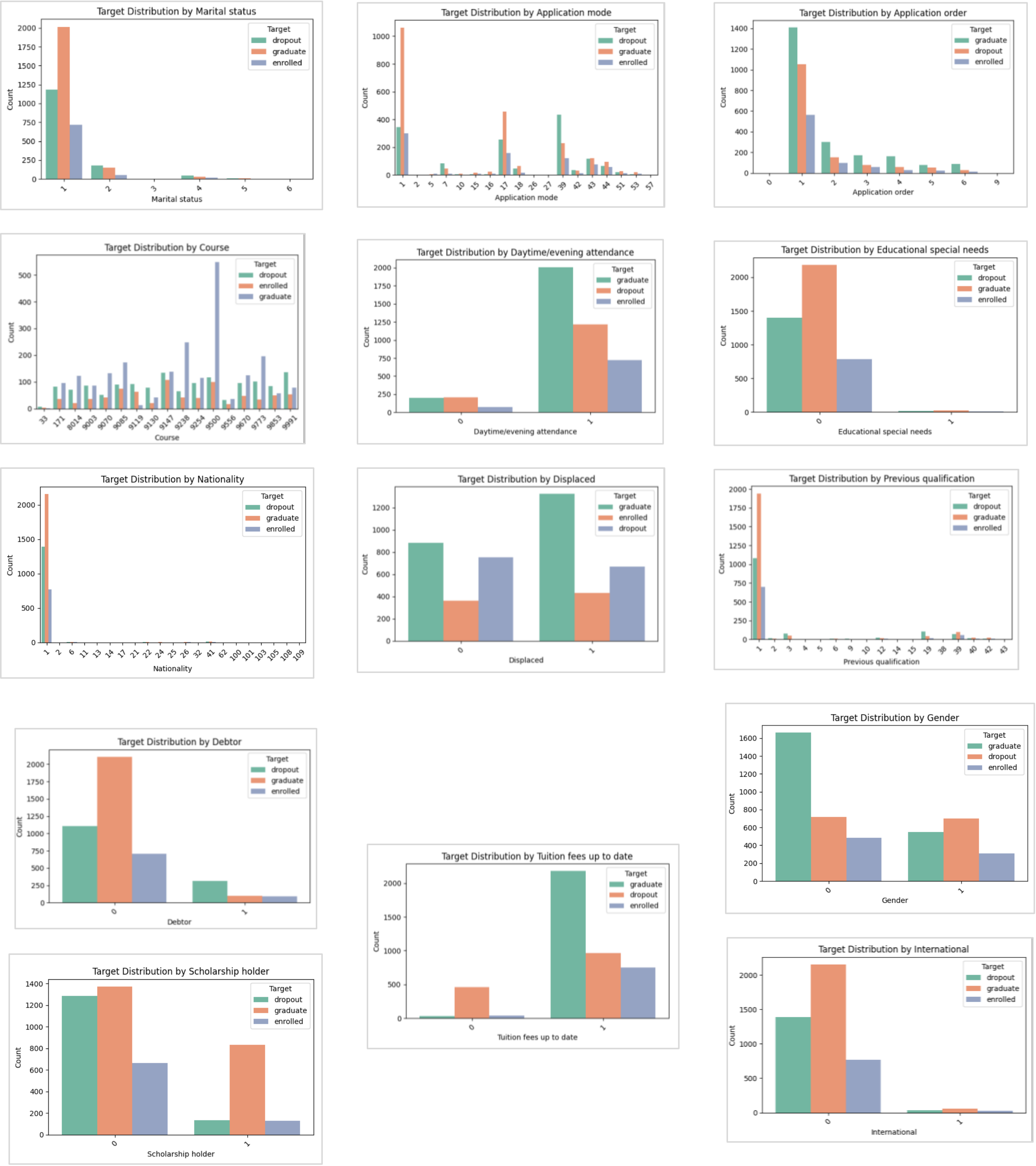


Figure 3: Histograms showing the distribution of continuous numerical features, including Admission Grade, Previous Qualification Grade, Curricular Unit Grades, Unemployment Rate, Inflation Rate, and GDP


Target Distribution by Marital status

Target Distribution by Application mode

Target Distribution by Application order

Target Distribution by Course

Target Distribution by Daytime/evening attendance

Target Distribution by Educational special needs

Target Distribution by Nationality

Target Distribution by Displaced

Target Distribution by Previous qualification

Target Distribution by Debtor

Target Distribution by Gender

Target Distribution by Tuition fees up to date

Target Distribution by Scholarship holder

Target Distribution by International

Figure 4: Bar plots showing the distribution of categorical features such as Marital Status, Application Mode, Daytime/Evening Attendance, Nationality, Gender, Scholarship Holder Status, and others with respect to student outcomes.

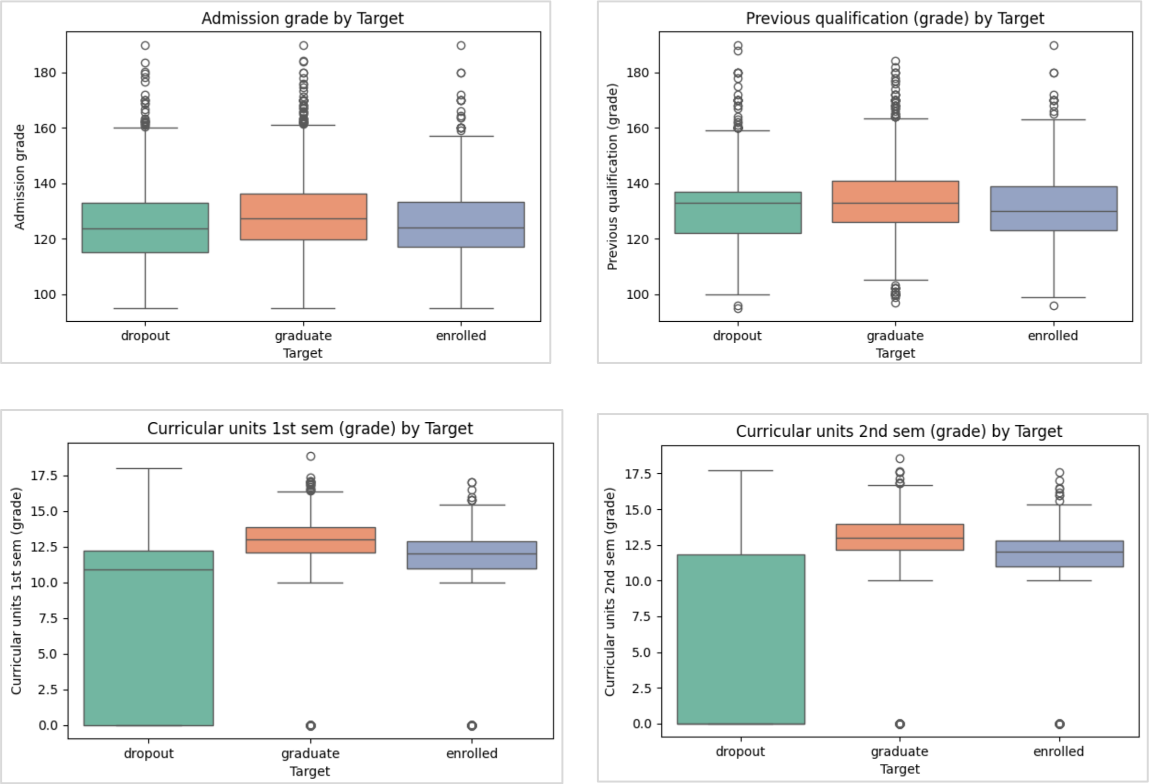
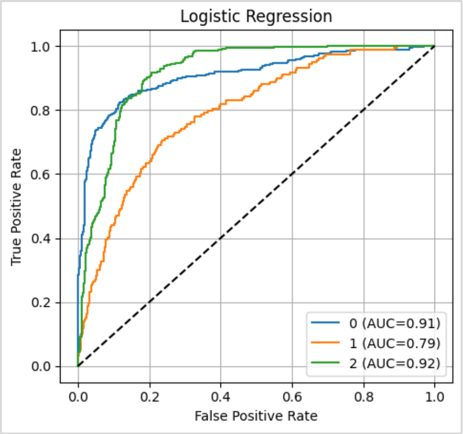
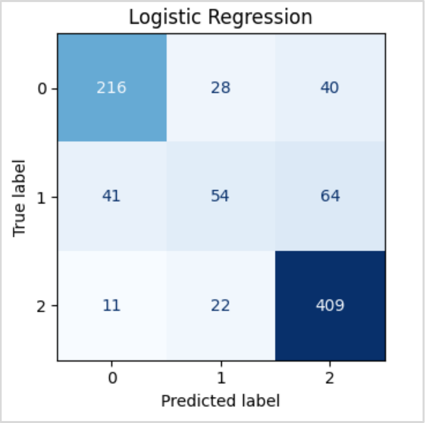


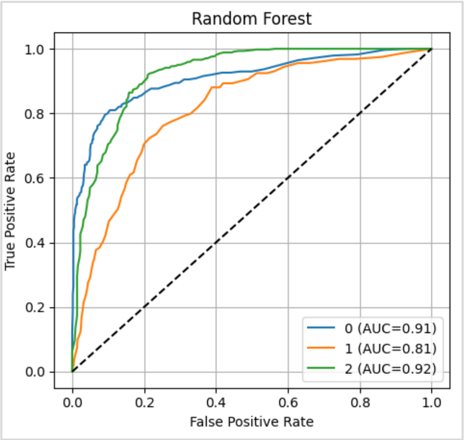
Figure 5: Box plots showing the distribution of key academic performance features

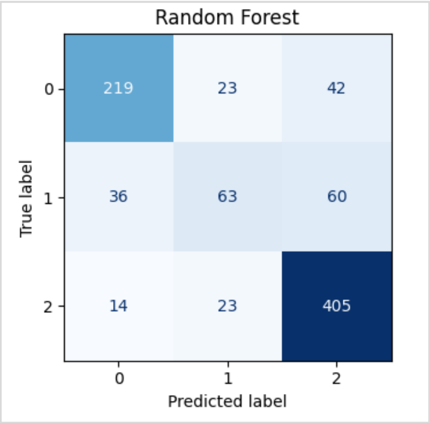
Confusion Matrices/ ROC Curves

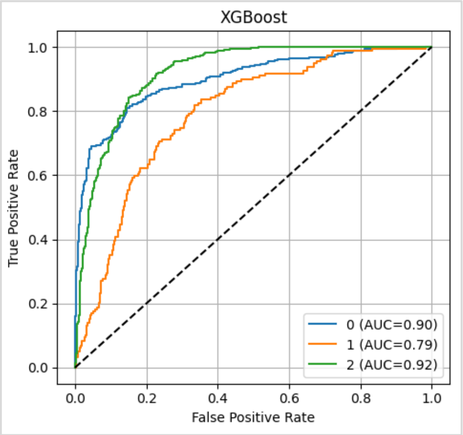
Approach 1:

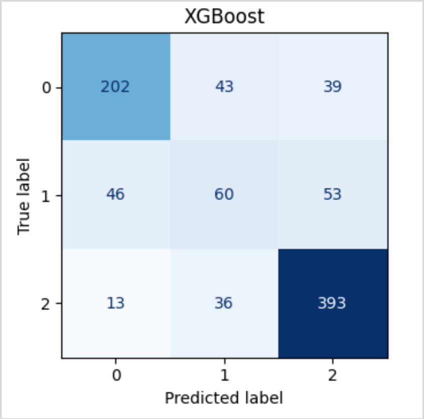


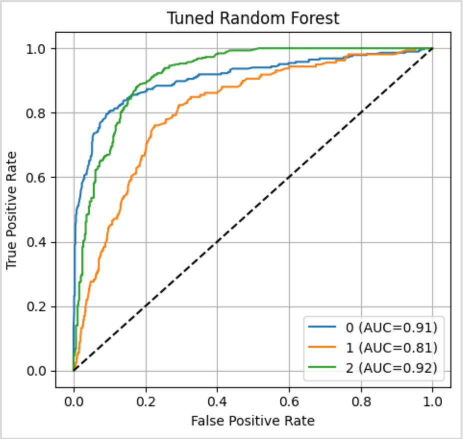


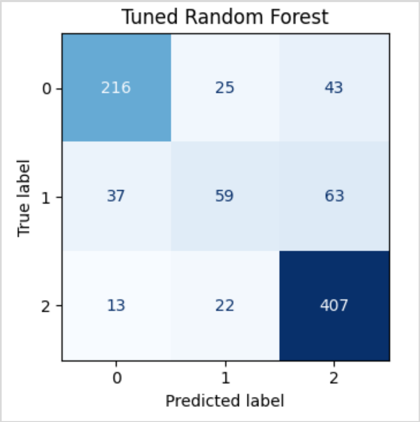

Random Forest
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
0 (AUC=0.91)
1 (AUC=0.81)
2 (AUC=0.92)

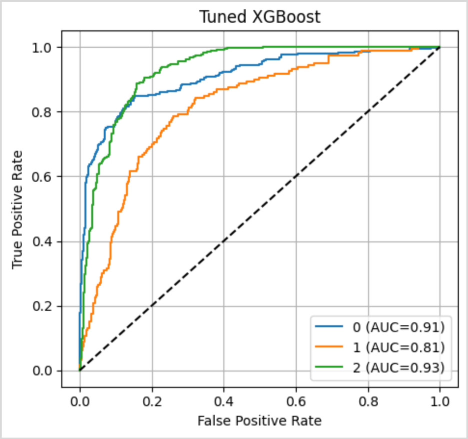

Random Forest
True label
Predicted label
0
1
2
219
23
42
36
63
60
14
23
405

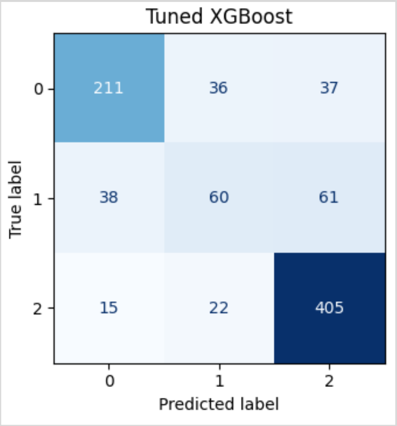

XGBoost
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
0 (AUC=0.90)
1 (AUC=0.79)
2 (AUC=0.92)


XGBoost
True label
Predicted label
0
1
2
202
43
39
46
60
53
13
36
393

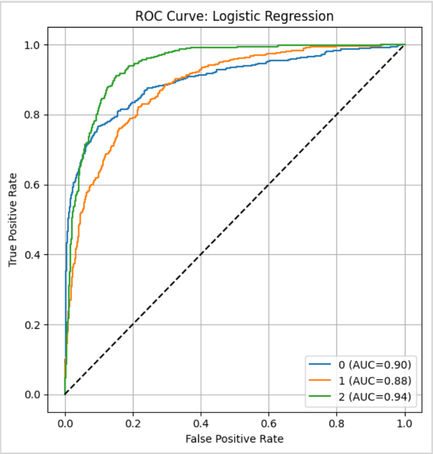

Tuned Random Forest
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
0 (AUC=0.91)
1 (AUC=0.81)
2 (AUC=0.92)

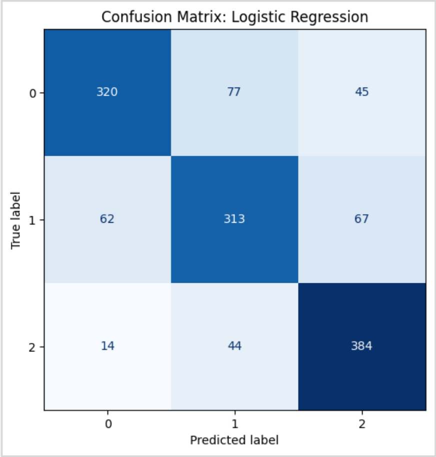

Tuned Random Forest
True label
Predicted label
0
1
2
216
25
43
37
59
63
13
22
407

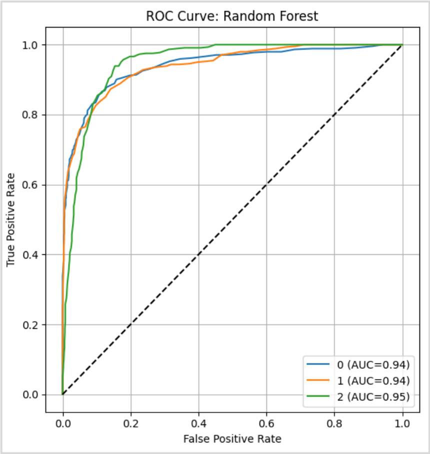

Tuned XGBoost
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
0 (AUC=0.91)
1 (AUC=0.81)
2 (AUC=0.93)

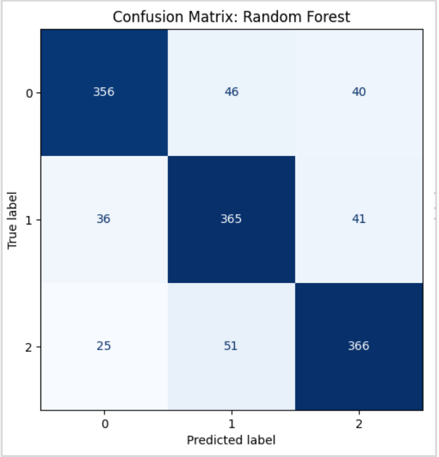

Tuned XGBoost
True label
Predicted label
0
1
2
211
36
37
38
60
61
15
22
405

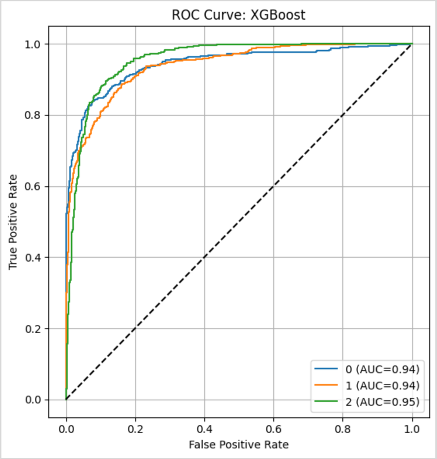
Approach 2:

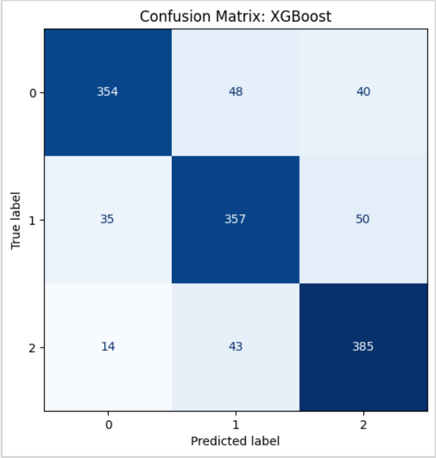

ROC Curve: Logistic Regression
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
0 (AUC=0.90)
1 (AUC=0.88)
2 (AUC=0.94)

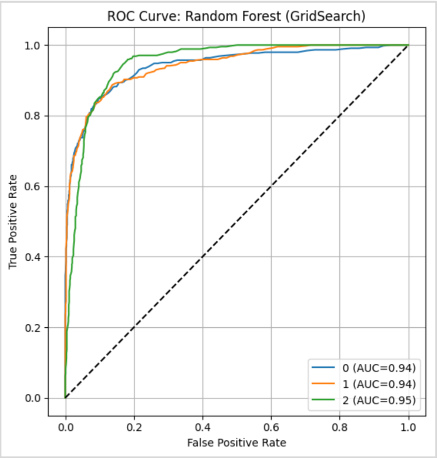

Confusion Matrix: Logistic Regression
True label
Predicted label
0
1
2
320
77
45
62
313
67
14
44
384

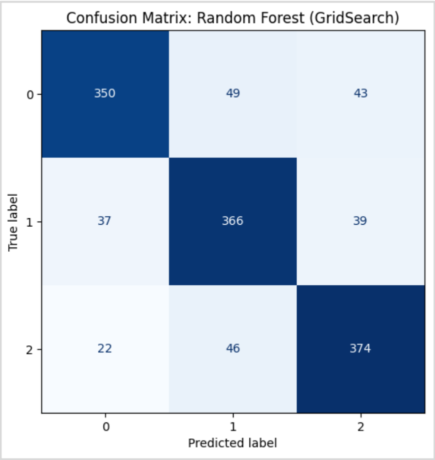

ROC Curve: Random Forest
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
0 (AUC=0.94)
1 (AUC=0.94)
2 (AUC=0.95)


Confusion Matrix: Random Forest
True label
Predicted label
0
1
2
356
46
40
36
365
41
25
51
366

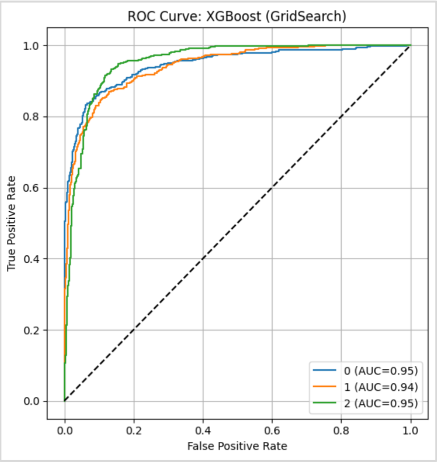

ROC Curve: XGBoost
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
0 (AUC=0.94)
1 (AUC=0.94)
2 (AUC=0.95)

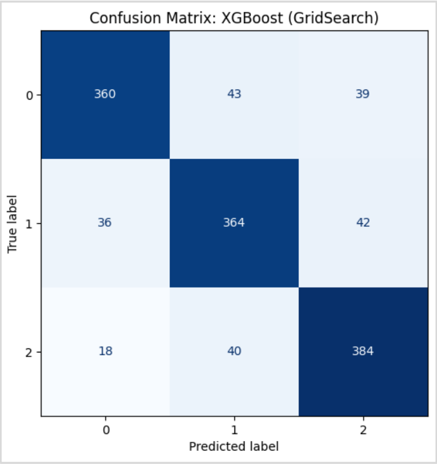

Confusion Matrix: XGBoost
True label
Predicted label
0
1
2
354
48
40
35
357
50
14
43
385





	0	1	2
0	350	49	43
1	37	366	39
2	22	46	374





	0	1	2
0	360	43	39
1	36	364	42
2	18	40	384