Introduction & Problem Statement

In recent years, higher education institutions have faced growing challenges in managing student retention and graduation rates. A significant number of students fail to complete their degree programs due to various academic, financial, and personal factors. Predicting whether a student is likely to graduate or drop out has become a vital task for universities seeking to provide early support and intervention.

- □ The primary problem addressed in this study is to **predict whether a student will graduate or drop out** based on academic, demographic, and socioeconomic factors.

The dataset used includes variables such as age at enrollment, academic performance, attendance, and economic indicators. The goal is to design and evaluate machine learning models that can accurately classify students into *graduate* or *dropout* categories.

Objectives of the Project

- To analyze and preprocess the *Higher Education Predictors of Student Retention* dataset obtained from Kaggle.
- To perform **exploratory data analysis (EDA)** and identify significant factors influencing student retention.
- To develop and compare multiple **machine learning classification models** (e.g. Logistic Regression, Decision Tree, Random Forest) for predictive performance.
- To evaluate each model using appropriate performance metrics such as Accuracy, Precision, Recall, and F1-score.
- To examine **ethical implications and bias mitigation** strategies to ensure fairness and transparency in model predictions.

Dataset Description

Dataset source: Kaggle dataset “Higher Education Predictors of Student Retention”.

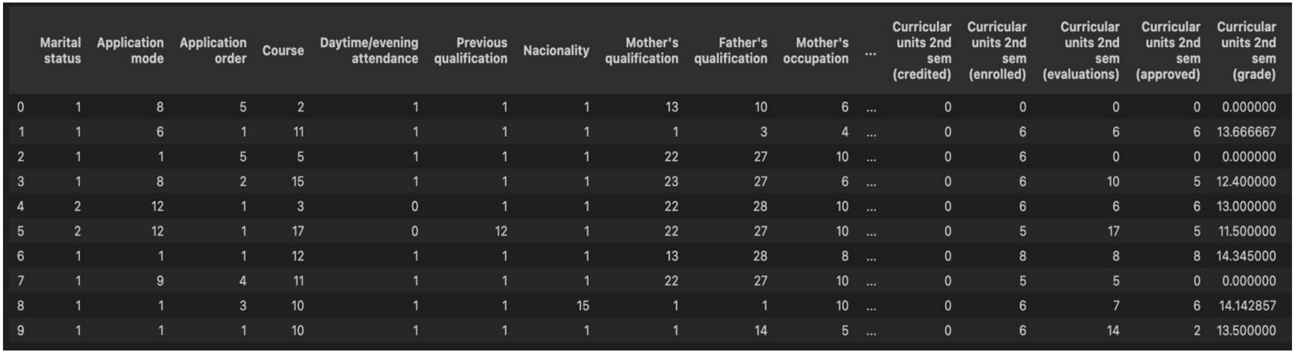
https://www.kaggle.com/datasets/thedevastator/higher-education-predictors-of-student-retention

Type of dataset: Structured Tabular Dataset (csv)

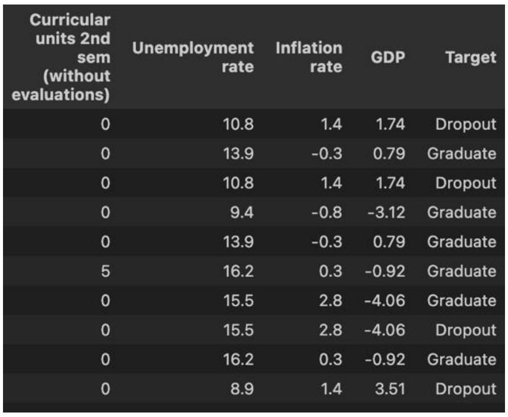
Number of records: Around 4000 student records.

Number of features: 35 features
(eg: GDP, Curricular units at different semesters, age, gender, family background, etc.)

Target: Graduate, Dropout, Enrolled (later removed Enrolled.)



	Marital status	Application mode	Application order	Course	Daytime/evening attendance	Previous qualification	Nacionality	Mother's qualification	Father's qualification	Mother's occupation	...	Curricular units 2nd sem (credited)	Curricular units 2nd sem (enrolled)	Curricular units 2nd sem (evaluations)	Curricular units 2nd sem (approved)	Curricular units 2nd sem (grade)
0	1	8	5	2	1	1	1	13	10	6	...	0	0	0	0	0.000000
1	1	6	1	11	1	1	1	1	3	4	...	0	6	6	6	13.666667
2	1	1	5	5	1	1	1	22	27	10	...	0	6	0	0	0.000000
3	1	8	2	15	1	1	1	23	27	6	...	0	6	10	5	12.400000
4	2	12	1	3	0	1	1	22	28	10	...	0	6	6	6	13.000000
5	2	12	1	17	0	12	1	22	27	10	...	0	5	17	5	11.500000
6	1	1	1	12	1	1	1	13	28	8	...	0	8	8	8	14.345000
7	1	9	4	11	1	1	1	22	27	10	...	0	5	5	0	0.000000
8	1	1	3	10	1	1	15	1	1	10	...	0	6	7	6	14.142857
9	1	1	1	10	1	1	1	1	14	5	...	0	6	14	2	13.500000



Curricular units 2nd sem (without evaluations)	Unemployment rate	Inflation rate	GDP	Target
0	10.8	1.4	1.74	Dropout
0	13.9	-0.3	0.79	Graduate
0	10.8	1.4	1.74	Dropout
0	9.4	-0.8	-3.12	Graduate
0	13.9	-0.3	0.79	Graduate
5	16.2	0.3	-0.92	Graduate
0	15.5	2.8	-4.06	Graduate
0	15.5	2.8	-4.06	Dropout
0	16.2	0.3	-0.92	Graduate
0	8.9	1.4	3.51	Dropout

The dataset contains academic, demographic, and economic indicators related to student performance. The target variable, Target, originally had three classes – Graduate, Dropout and Enrolled. But the "Enrolled" class was excluded to focus on binary prediction.

Preprocessing & EDA

Data Cleaning Process

- Missing values checked & removed.
- Duplicate rows checked & removed.
- Target variable cleaned (drop rows with "Enrolled").
- Outliers removed
  (Custom outlier remove to get value between 0-20, Normal outlier remove)
- Label encoding: mapped Graduate → 0, Dropout → 1.
- Feature Scaling (StandardScaler())
- Target column balanced. (Over Sampling)

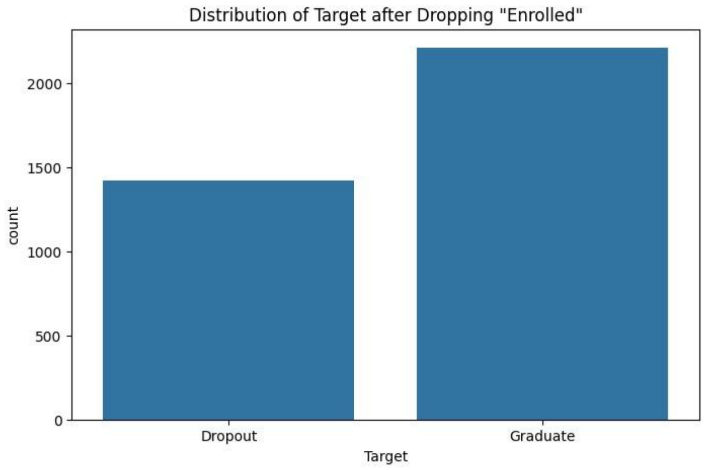
Feature Selection

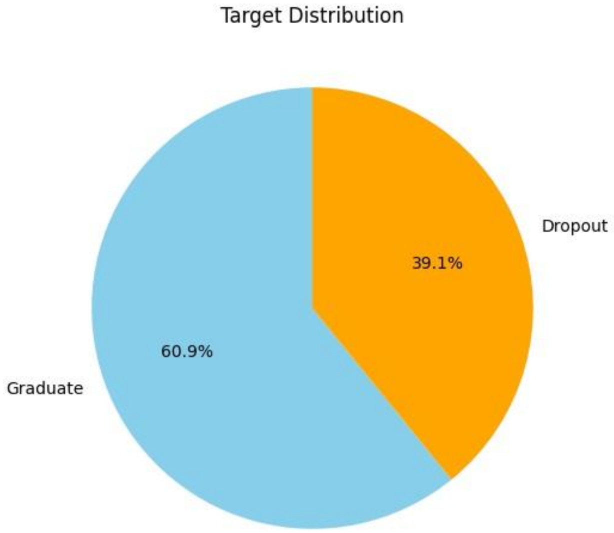
After completing data cleaning and encoding, feature selection was carried out to identify the most important factors affecting whether a student graduates or drops out. Two main methods were used: **correlation analysis** and the **Chi-square test**.

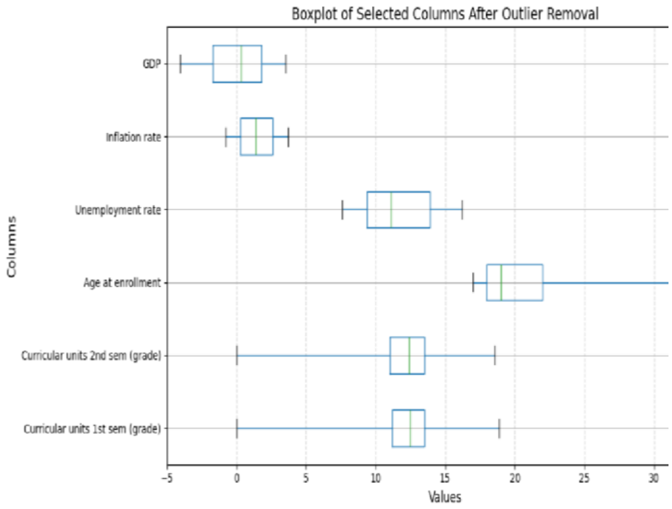
For numerical features, a correlation matrix was used to find how strongly each variable is related to the target outcome. Features that showed very weak or redundant correlations were removed to avoid unnecessary complexity. For categorical features, the Chi-square test was applied to check how strongly each feature is associated with the target variable.

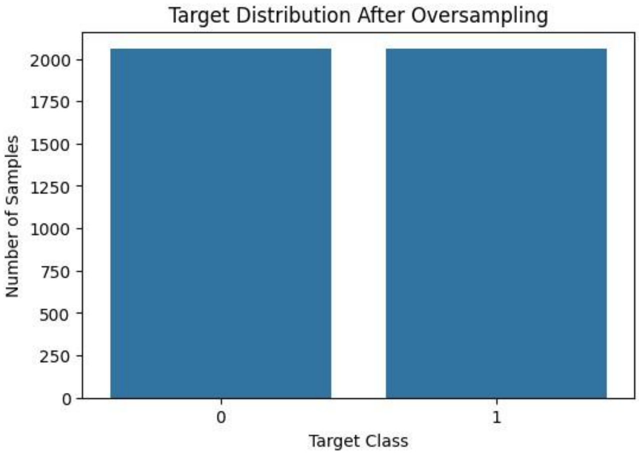
Finally, only the features that showed meaningful relationships with the target variable were kept for model training. This step helped improve model performance, reduce noise in the data, and make the results easier to interpret.

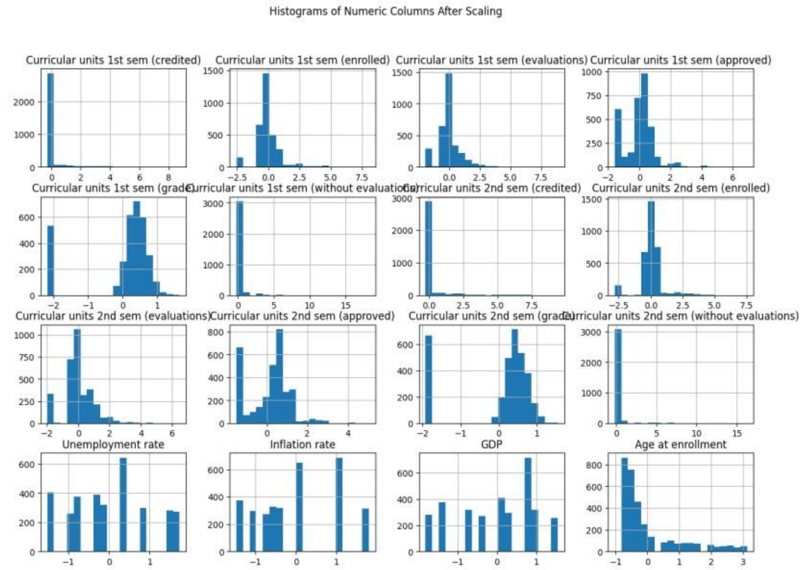
EDA Visualization after each preprocessing technique

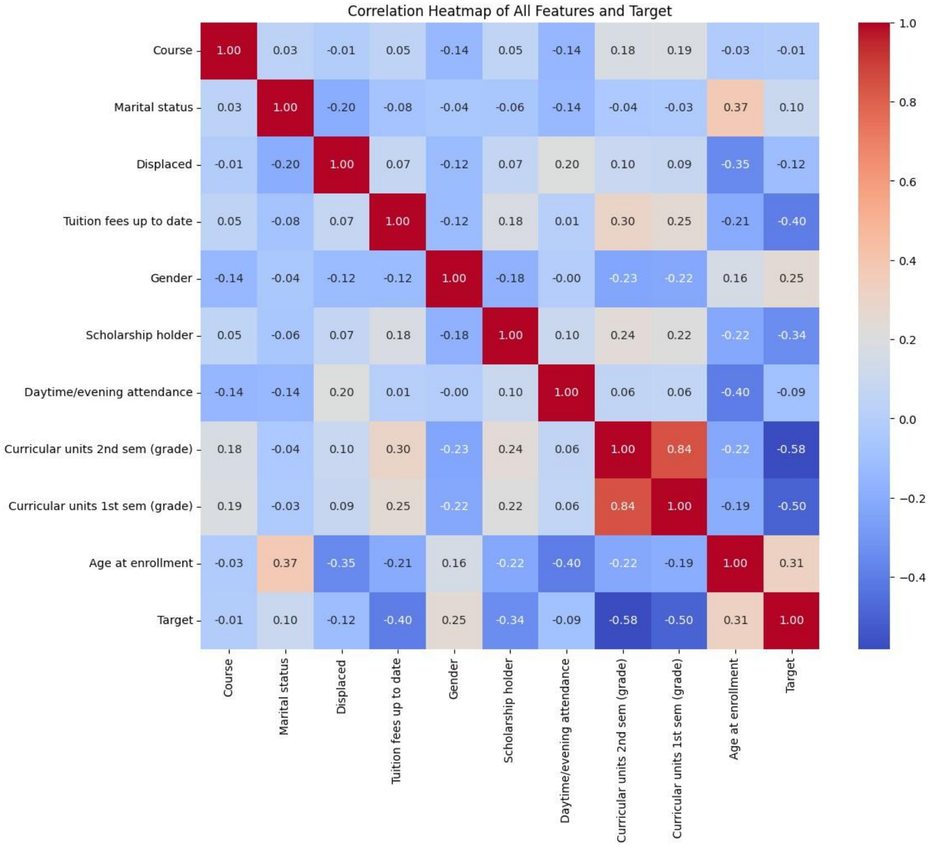





Boxplot of Selected Columns After Outlier Removal
GDP
Inflation rate
Unemployment rate
Age at enrollment
Curricular units 2nd sem (grade)
Curricular units 1st sem (grade)
Columns
Values
−5
0
5
10
15
20
25
30


Target Distribution After Oversampling
Number of Samples
Target Class
0
250
500
750
1000
1250
1500
1750
2000
0
1


Histograms of Numeric Columns After Scaling
Curricular units 1st sem (credited)
Curricular units 1st sem (enrolled)
Curricular units 1st sem (evaluations)
Curricular units 1st sem (approved)
Curricular units 1st sem (grade)
Curricular units 1st sem (without evaluations)
Curricular units 2nd sem (credited)
Curricular units 2nd sem (enrolled)
Curricular units 2nd sem (evaluations)
Curricular units 2nd sem (approved)
Curricular units 2nd sem (grade)
Curricular units 2nd sem (without evaluations)
Unemployment rate
Inflation rate
GDP
Age at enrollment


Correlation Heatmap of All Features and Target
Course 1.00 0.03 -0.01 0.05 -0.14 0.05 -0.14 0.18 0.19 -0.03 -0.01
Marital status 0.03 1.00 -0.20 -0.08 -0.04 -0.06 -0.14 -0.04 -0.03 0.37 0.10
Displaced -0.01 -0.20 1.00 0.07 -0.12 0.07 0.20 0.10 0.09 -0.35 -0.12
Tuition fees up to date 0.05 -0.08 0.07 1.00 -0.12 0.18 0.01 0.30 0.25 -0.21 -0.40
Gender -0.14 -0.04 -0.12 -0.12 1.00 -0.18 -0.00 -0.23 -0.22 0.16 0.25
Scholarship holder 0.05 -0.06 0.07 0.18 -0.18 1.00 0.10 0.24 0.22 -0.22 -0.34
Daytime/evening attendance -0.14 -0.14 0.20 0.01 -0.00 0.10 1.00 0.06 0.06 -0.40 -0.09
Curricular units 2nd sem (grade) 0.18 -0.04 0.10 0.30 -0.23 0.24 0.06 1.00 0.84 -0.22 -0.58
Curricular units 1st sem (grade) 0.19 -0.03 0.09 0.25 -0.22 0.22 0.06 0.84 1.00 -0.19 -0.50
Age at enrollment -0.03 0.37 -0.35 -0.21 0.16 -0.22 -0.40 -0.22 -0.19 1.00 0.31
Target -0.01 0.10 -0.12 -0.40 0.25 -0.34 -0.09 -0.58 -0.50 0.31 1.00
Course
Marital status
Displaced
Tuition fees up to date
Gender
Scholarship holder
Daytime/evening attendance
Curricular units 2nd sem (grade)
Curricular units 1st sem (grade)
Age at enrollment
Target
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4

Model Design and Implementation

Models Selected

Model	Reason for selecting
Logistic regression	A simple and interpretable model that works well for binary classification problems.
Random Forest	It's an ensemble of decision trees, which makes it more accurate and less prone to overfitting. It handles both numerical and categorical data well and gives feature importance scores.
Decision Tree	It's easy to understand and visualize. It splits the data based on feature values, which helps identify the most important factors influencing student outcomes.
SVM	Useful for binary classification problems. It works well with both linear and non-linear data by finding the best boundary (hyperplane) that separates the two classes. It's especially effective when the data has clear margins between classes.
KNN	It's a simple, instance-based learning method that classifies a student based on the outcomes of similar students. It's useful when the data has clear clusters or patterns.
XGBoost	It's an efficient gradient boosting algorithm that often gives high accuracy. Great for structured/tabular datasets.

Model Training

The dataset was divided into **training** and **testing** sets to evaluate model generalization. The **training set** was used for model fitting, while the **test set** evaluated real-world prediction performance. A consistent random state was used to ensure reproducibility. Each model was trained using Scikit-learn's standard library functions and XGBoost was implemented using the *xgboost* package.

Hyperparameter Tuning

To improve model performance hyperparameter tuning was carried out. This process involved adjusting key configuration parameters that influence how the model learns. The tuning was performed using **RandomizedSearchCV** with **cross-validation**, which automatically tested multiple parameter combinations and evaluated their performance based on accuracy scores. The best-performing parameter set was then selected to train the final model. The tuned XGBoost model achieved higher accuracy and reduced overfitting compared to the baseline version, demonstrating the effectiveness of parameter optimization.

Cross Validation

Cross-validation was integrated into the model training and tuning process to ensure fair and reliable evaluation of each parameter setting. During hyperparameter tuning with RandomizedSearchCV, the training data was automatically divided into multiple folds to identify the best-performing parameter combinations.

Evaluation and Comparison

To measure the performance of the developed models, several evaluation metrics were used;

Accuracy: The proportion of correctly predicted instances.

Precision: The proportion of predicted dropouts that were actually correct.

Recall: The proportion of actual dropouts correctly identified by the model.

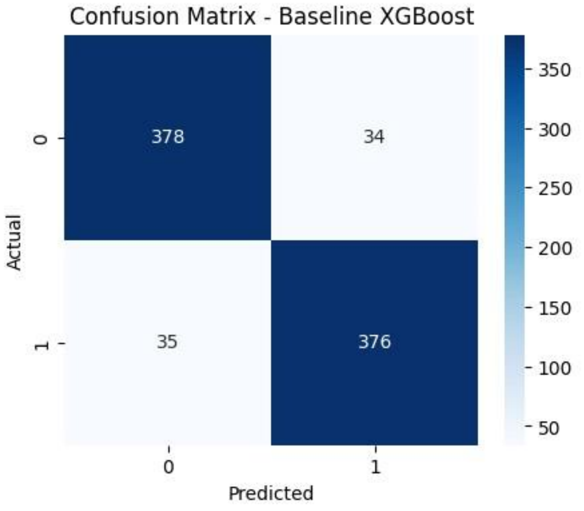
F1-Score: The harmonic mean of precision and recall, representing the overall balance between the two.

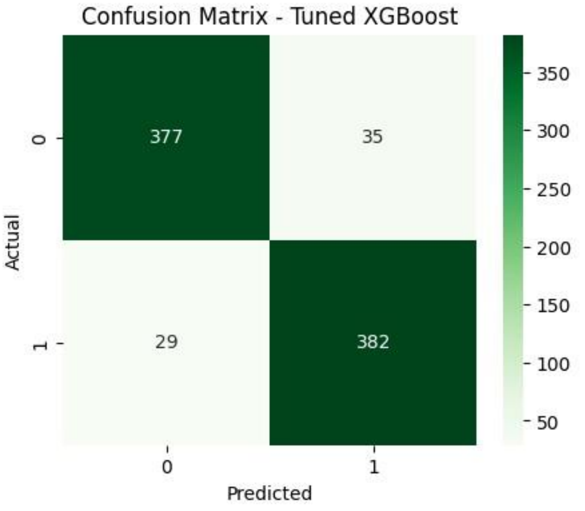
Cross-Validation Accuracy: The average performance of the model across multiple validation folds.

Confusion Matrix Analysis

Confusion matrices were generated for both the **baseline** and **tuned** versions of the models. These matrices visualize true positive, true negative, false positive, and false negative predictions, helping to understand the types of classification errors made by the model.

Eg: The tuned XGBoost model displayed fewer misclassifications compared to the baseline, confirming that parameter optimization improved predictive accuracy and class balance.

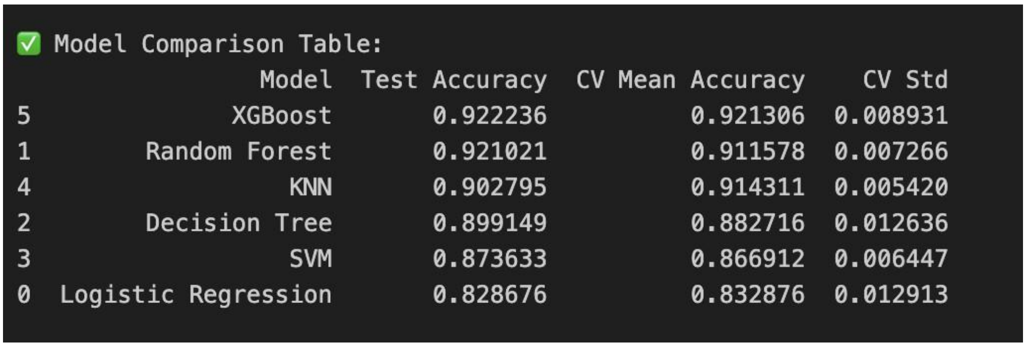




Cross-Validation Results

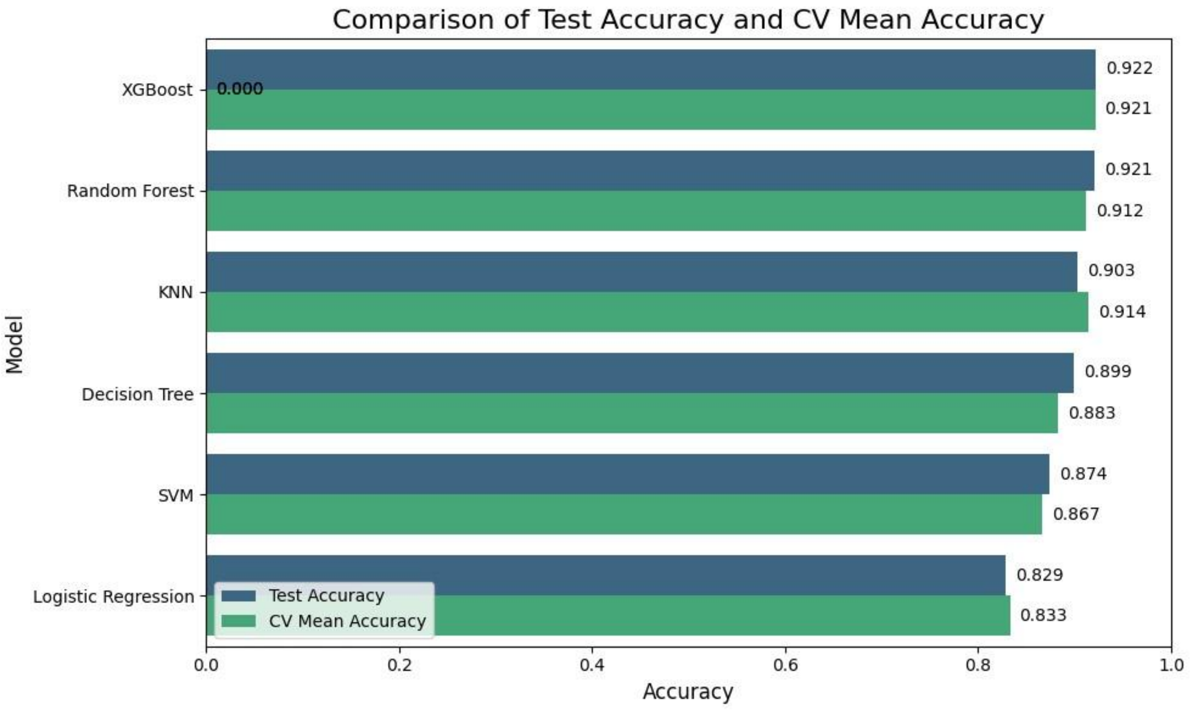
To evaluate model stability and consistency, cross-validation was applied using multiple folds of the training data. The average accuracy and standard deviation across all folds were used to assess how well each model generalized to unseen data.

Model Comparison



```
✅ Model Comparison Table:
                 Model  Test Accuracy  CV Mean Accuracy    CV Std
5              XGBoost       0.922236          0.921306  0.008931
1        Random Forest       0.921021          0.911578  0.007266
4                  KNN       0.902795          0.914311  0.005420
2        Decision Tree       0.899149          0.882716  0.012636
3                  SVM       0.873633          0.866912  0.006447
0  Logistic Regression       0.828676          0.832876  0.012913
```

These results shows that **XGBoost** achieved the highest accuracy.



The above chart clearly highlights the superior performance of the tuned XGBoost model, which maintained both high accuracy and stability across validation folds.

Therefore, the tuned **XGBoost** model was selected as the **final model** for predicting student graduation and dropout outcomes.

Ethical Considerations and Bias Mitigation

Ethical Importance in Educational Prediction

Developing predictive models for student outcomes requires careful attention to ethical concerns, as such systems may directly affect students' academic opportunities and support. Predictive analytics in education should always be used to **assist** decision-making, not to unfairly label or penalize students.

Potential Ethical Issues

- **Data Privacy:** The dataset may contain sensitive academic and demographic information. Proper handling, storage, and anonymization are essential to protect individual privacy.
- **Algorithmic Bias:** If the dataset is imbalanced or biased toward certain student groups (eg: gender, income level, or department), the model might make unfair predictions.

- **Transparency:** Students and administrators must be able to understand how predictions are made and how model decisions are used.

Bias Identification and Mitigation

During model development, the dataset was checked for **class imbalance** and **bias** in demographic features. Records were preprocessed to reduce these issues, and the "Enrolled" category was removed to focus on clearer binary outcomes.

Data balancing techniques and fairness-aware feature selection were used to ensure that no particular group dominated the training process. Additionally, evaluation metrics were interpreted carefully to confirm that high accuracy did not come at the cost of biased misclassifications.

Ethical Use of the Model

The predictive model is intended for **early intervention**, helping universities identify students who might need academic or emotional support. It should **not** be used for punishment, exclusion, or ranking purposes. Ethical use emphasizes transparency, fairness, and the improvement of student welfare.

Summary

In summary, ethical and bias considerations were integrated throughout the project to promote fairness, protect privacy, and ensure responsible use of AI in education. Continuous monitoring and stakeholder awareness are recommended to maintain these ethical standards as the model is applied in real-world scenarios.

Reflections and Lessons Learned

Throughout this project, we were able to receive experience on working with AIML techniques to solve a real-world educational problem. We understood the importance of preprocessing of data like cleaning, feature selection, and handling class imbalance, as they significantly affected towards model performance.

Testing several algorithms such as Logistic Regression, Decision Tree, Random Forest, SVM, KNN, and XGBoost provided us with understanding about how the complexity of the model affects accuracy and interpretability. Hyperparameter tuning and cross-validation also showed us the need to fine-tune parameters to improve generalization and avoid overfitting.

As for challenges we faced; managing noisy data and predicting with fairness can be mentioned, but these tasks developed our understanding of ethical AI procedures.

Overall, the project strengthened our teamwork, technical skills, and knowledge about how machine learning can be applied responsibly to aid in decision-making in a common real-world problem.

References

[1] Kaggle, “Higher Education Predictors of Student Retention,” 2024. [Online]. Available: https://www.kaggle.com/datasets/thedevastator/higher-education-predictors-ofstudent-retention

[2] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[3] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[4] W. McKinney, “Data structures for statistical computing in Python,” in *Proc. 9th Python Sci. Conf.*, 2010, pp. 51–56.

[5] J. D. Hunter, “Matplotlib: A 2D graphics environment,” *Comput. Sci. Eng.*, vol. 9, no. 3, pp. 90–95, 2007.