Index

| 1 | Introduction | 2 |
| --- | --- | --- |
| 2 | Feature Description | 2 |
| 3 | Libraries used in the notebook | 3 |
| 4 | Data Cleaning | 3 |
| 5 | EDA | 3 |
|  | 1. Proportion of Labels | 4 |
|  | 1. Analysis of Categorical Features | 4 |
|  | 1. Analysis of Numerical Features | 7 |
|  | 1. EDA Summary | 11 |
| 6 | Modeling | 12 |
|  | 1. Evaluation Metrics | 12 |
|  | 1. First Iteration of Modeling | 13 |
|  | 1. Second Iteration of Modeling | 14 |
|  | 1. Feature Importance Calculation | 15 |
|  | 1. Third Iteration of Modeling | 16 |
| 7 | Results & Analysis | 18 |
|  | 1. Performance over three Iteration | 18 |
|  | 1. Feature Importance of Final Model | 19 |
|  | 1. Probability of Dropping out in a Simulated Dataset | 22 |
| 8 | Conclusion | 24 |

Student Dropout Prediction Report

# Introduction

Student dropout, graduation, and continued enrollment represent significant educational outcomes impacting both individual futures and broader societal well-being. Dropouts experience diminished economic opportunities, while graduates often enjoy better employment prospects. Accurately predicting these outcomes—dropout, enrolled, and graduate—enables institutions to intervene proactively, supporting at-risk students.  
  
This project aims to predict student outcomes by identifying patterns and key predictors within a dataset of 4424 records featuring 35 demographic, academic, and socioeconomic variables, including attendance rates, GPA, gender, family income, and parental education.

**Feature Description**

* Marital status: The marital status of the student. (Categorical)
* Application mode: The method of application used by the student. (Categorical)
* Application order: The order in which the student applied. (Numerical)
* Course: The course taken by the student. (Categorical)
* Daytime/evening attendance: Whether the student attends classes during the day or in the evening. (Categorical)
* Previous qualification: The qualification obtained by the student before enrolling in higher education. (Categorical)
* Nationality: The nationality of the student. (Categorical)
* Mother's qualification: The qualification of the student's mother. (Categorical)
* Father's qualification: The qualification of the student's father. (Categorical)
* Mother's occupation: The occupation of the student's mother. (Categorical)
* Father's occupation: The occupation of the student's father. (Categorical)
* Displaced: Whether the student is a displaced person. (Categorical)
* Educational special needs: Whether the student has any special educational needs. (Categorical)
* Debtor: Whether the student is a debtor. (Categorical)
* Tuition fees up to date: Whether the student's tuition fees are up to date. (Categorical)
* Gender: The gender of the student. (Categorical)
* Scholarship holder: Whether the student is a scholarship holder. (Categorical)
* Age at enrollment: The age of the student at the time of enrollment. (Numerical)
* International: Whether the student is an international student. (Categorical)
* Curricular unit’s 1st sem (credited): The number of curricular units credited by the student in the first semester. (Numerical)
* Curricular unit’s 1st sem (enrolled): The number of curricular units enrolled by the student in the first semester. (Numerical)
* Curricular unit’s 1st sem (evaluations): The number of curricular units evaluated by the student in the first semester. (Numerical)
* Curricular unit’s 1st sem (approved): The number of curricular units approved by the student in the first semester. (Numerical)

**Libraries used in the notebook.**

1. **Import important packages**

* import pandas as pd
* import numpy as np
* import seaborn as sns
* import matplotlib.pyplot as plt

1. **Data Preprocessing and EDA**

* from sklearn.preprocessing import OrdinalEncoder
* from scipy.stats import chi2\_contingency

**Data Cleaning**

* Renamed columns to remove whitespace and parentheses for consistency.
* Handled whitespace and formatting inconsistencies in categorical variables.
* Converted categorical variables (e.g., Gender, School\_Type) to the 'category' data type, optimizing memory and facilitating accurate analysis.

# Exploratory Data Analysis (EDA)

## Overall Approach

I systematically explored relationships between features and the multivariate target variable (dropout, enrolled, graduate) using visualizations and statistical tests to uncover significant predictors.

**Proportion of the Labels**

**A pie chart with text on it

Description automatically generated**

From the pie chart above we can see that the data is imbalanced: about 50% of the labels are 'Graduate', 32% are 'Dropout', and 18% are 'Enrolled'. The labels are encoded as ordinal data -- 0 represents 'Dropout', 1 represents 'Enrolled', and 2 represents 'Graduate' -- since most classification models only handle numeric values.

**Analysis of Categorical Features**

Univariate Analysis - Count plots were generated for categorical features like 'Gender', 'School\_Type', and 'Parent\_education\_level'. Etc.

A graph of a number of blue and white bars

Description automatically generated with medium confidence A graph of a number of blue and white bars

Description automatically generated with medium confidence

**Interpretation:**

* Marital Status is heavily skewed toward one group, which may reduce its impact unless the minority classes have distinct dropout patterns.
* Application Mode and Order show clear preference for a few categories. These could signal motivation or institutional processes and might be useful predictors.
* Course shows a broad distribution, which is promising—it could help the model differentiate dropout risk across programs.
* Daytime/Evening Attendance is mostly daytime, but evening students might represent a unique group (e.g., working students), possibly influencing outcomes.
* Previous Qualification and Nationality have low variability, especially nationality, which may limit their predictive power.
* Parental Education and Occupation have high cardinality and long tails. They might capture socioeconomic factors if grouped smartly.
* Displaced, Special Needs, Debtor, and Tuition Fees are binary and imbalanced, but they likely capture key risk factors—especially financial-related ones.
* Gender, Scholarship Holder, and International Status are slightly imbalanced. Their usefulness will depend on how they interact with other features.
* Overall, I’ll revisit these after initial model runs to evaluate their importance and decide if any features need transformation, grouping, or exclusion.

**Bivariate Analysis (Chi-Square Test)** - Chi-Square tests were conducted to determine associations between categorical features and the three outcomes:  
(Insert Chi-Square test results table here)  
**Interpretation:**

A screenshot of a phone

Description automatically generated

Most of the p-values are close to zero, except for three variables ('Nationality', 'International', 'Educational\_special\_needs') with very high p-values (0.24, 0.53, 0.73), indicating that no statistically significant association between these three features and the label. They are excluded from the model.

## Analysis of Numerical Features

## Estimates of Location and Variability –

<https://colab.research.google.com/drive/1wYYqiduWaAtsj4tIQ3XHqQgSl8U3n1cd#scrollTo=gezQNhXN6Zew&line=1&uniqifier=1>

* Looking at Age, Curricular\_units\_1st\_sem\_credited and Curricular\_units\_1st\_sem\_enrolled, it's evident that there are outliers in the data.
* Age - 75% quartile is up-to 25 and max is 70 years of age, which is a huge jump.
* Curricular\_units\_1st\_sem\_credited - 75% quartile is up-to 0 and max is at 20 credits, which is a huge jump. Similarly, it is the same with Curricular\_units\_1st\_sem\_enrolled.

## Univariate Analysis - Histograms to visualize the distribution of numerical features.

A graph of a graph

Description automatically generated with medium confidence

A graph of a graph

Description automatically generated with medium confidence

A graph of a graph

Description automatically generated with medium confidence

**Interpretation:**

Looking at these log-transformed histograms, I can tell that a lot of the data was originally skewed — probably with long tails or a lot of zeros. After applying the log transformation, the shapes of the distributions look a lot more normalized or at least smoothed out.

* Age: has a right skew originally, but the log transformation helped compress the extreme values. Now, I can clearly see that most students fall into a narrow age band.
* Curricular Units (1st & 2nd Sem): For things like curricular\_units\_1st\_sem\_approved, grade, evaluations, and similar for the 2nd semester, I see clear peaks and groupings. The log scale helped reveal subtleties, like small groups with low or zero completions or grades. Some of the variables still have sharp peaks — maybe a lot of students scored zeros or maxed out the units.
* Unemployment Rate & Inflation: These were categorical or rounded values originally, so even after log-transforming, they still look like bar charts. They seem to cluster around specific national benchmarks, which might reflect broader economic conditions.

**Bivariate Analysis** - Spearman's rank correlation between numerical features and the Target

A screenshot of a graph

Description automatically generated

**Interpretation:**

* As can be seen from the heat map, there are four features ('Curricular\_units\_2nd\_sem\_approved’, 'Curricular\_units\_2nd\_sem\_grade’, ’Curricular\_units\_1st\_sem\_approved’, 'Curricular\_units\_1st\_sem\_grade’) that have relatively high and positive correlations with the label, while some have very low correlations(e.g., 'Unemployment\_rate', 'Inflation\_rate')
* The heat map also reveals multicollinearity among the features related to curricular units. These features represent students' academic performance at the end of the first and second semesters. I will aggregate them to get the average value between the two semesters.4. Numerical Feature Analysis

A screenshot of a graph

Description automatically generated

* The new correlation matrix above shows that 'curri\_avg\_approved' and 'curri\_avg\_grade' still have a relatively high correlation with the labels ('Target\_encoded'), while 'curri\_avg\_credited' and 'curri\_avg\_evaluations', along with 'the macroeconomic data ('Unemployment\_rate', 'Inflation\_rate'), have very low correlations, all between -0.02 and 0.02. I will exclude these four features.
* The multicollinearity still exists among the academic data. I'll take it into account when selecting the models.
* Let's check how 'curri\_avg\_approved' and 'curri\_avg\_grade' are associated with students' situation at the end of the normal duration of the course.

A comparison of a graph

Description automatically generated with medium confidence

It's not surprising that 'Graduate' is associated with more approved curricular units and higher grades. However, there are some instances of a 0 value for average grade and average approved curricular units in the 'Graduate' class.

# EDA Summary

1. **Feature Aggregation:**

* Academic performance data from two semesters were aggregated into features such as `avg\_credited`, `avg\_enrolled` to enhance predictive accuracy as shown above.

1. **Re-evaluation of Correlations:**

* Post-aggregation correlations demonstrated stronger associations with the multivariate outcomes, affirming the aggregation approach as shown above in heatmap.

1. **Outlier Removal:**

* Statistical criteria identified and removed outliers, ensuring data integrity and model effectiveness.

1. **Feature Selection:**

* Irrelevant or redundant features were dropped to streamline the dataset for modeling:  
  The final dataset focused on highly predictive variables, ensuring robust modeling.

1. **Class Imbalance:**

* The dataset clearly shows a significant class imbalance: approximately 50% Graduates, 32% Dropouts, and 18% currently Enrolled.
* Class imbalance may bias predictive models towards the majority class, potentially reducing prediction accuracy for minority classes (particularly "Enrolled").

1. **Categorical Features:**

* Several categorical features like Parental Education, Course, and Daytime/Evening Attendance appear promising due to clear variations across classes.
* Low variability in Nationality, International status, and Educational special needs justify their exclusion based on Chi-Square tests.

1. **Numerical Features and Transformations:**

* Log transformations effectively normalized skewed distributions (e.g., Age, Curricular units approved), enhancing interpretability and likely predictive accuracy.
* Strong correlations identified (Curricular\_units\_approved, Curricular\_units\_grade) are meaningful and expectedly crucial for modeling dropout or graduation outcomes.
* Macroeconomic indicators (Unemployment rate, Inflation rate) showed negligible correlations and rightly excluded.

1. **Multicollinearity:**

* Multicollinearity is observed among academically related features (e.g., curricular units). Aggregating these features is appropriate and improves correlation with outcomes.

**Modeling**

Based on the dataset characteristics— multiclass classification, approximately 50% categorical data, imbalanced classes, and multicollinearity— Random Forest and XGBoost are strong candidate models. They are known to handle these characteristics well and can achieve good performance on similar problems.

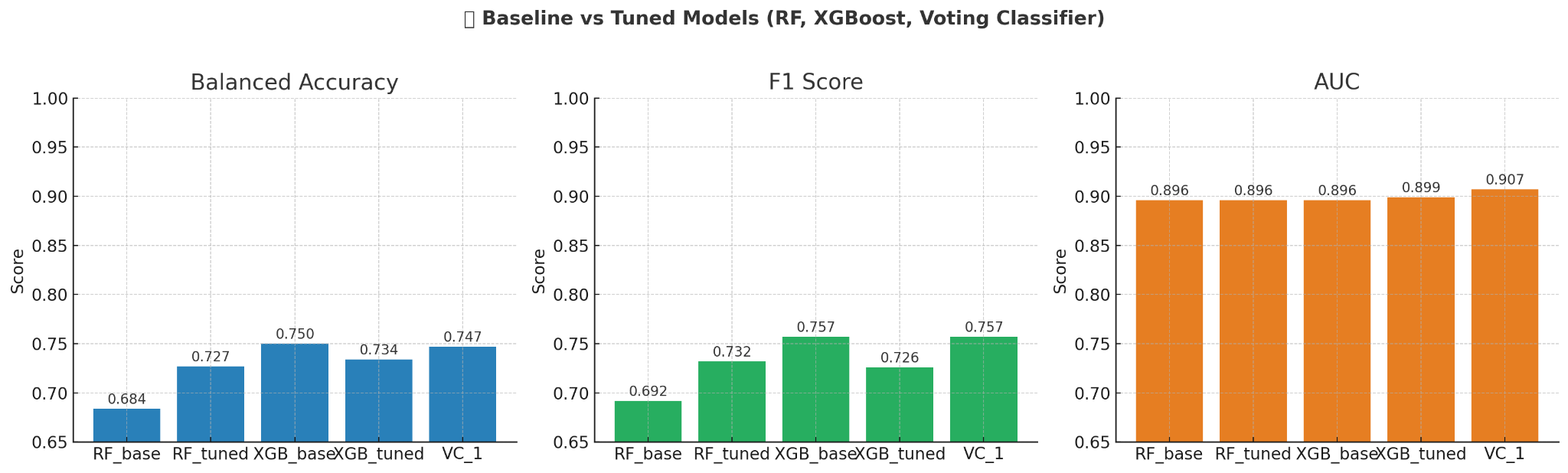
**Evaluation Metrics**

Due to the imbalanced nature of the dataset, employing Balanced Accuracy (the average recall obtained for each class), F1-score (the harmonic mean of precision and recall), and AUC (the area under the ROC curve, which plots True Positive Rate against False Positive Rate) as evaluation metrics.

Note: Dataset split ratio is 80:20 - 80% for training & 20% for testing.

**First Iteration of Modeling (21 features, 3-class label)**

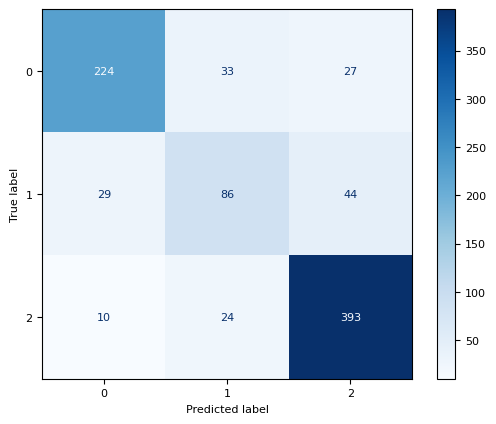
Trained the Random Forest and XGBoost models separately as base models(‘rf\_base’, ‘xgb\_base’), tuned their hyperparameters using RandomizedSearchCV with the default 5-fold cross validation (‘tuned\_rf’, ‘tuned\_xgb’), combined the two best-performing models from the previous four using VotingClassifier (‘vc\_soft’), and compared the performance of the five models.



From the performance plots above, we can see that 'xgb\_base' and 'vc\_soft' perform better than the others. The performance of these two models is very close: they have the same F1 score; 'xgb\_base' performs a bit better in balanced accuracy, and 'vc\_soft' performs a bit better in AUC.

It's also noticeable that the AUC scores in all models are much higher than the balanced accuracy and F1 score. This may indicate a potential issue with the model's ability to predict a certain class accurately.

**Confusion Matrix (XGBoost\_base\_model)**

****

* We used a confusion matrix to check how well the XGBoost base model predicts student outcomes across three categories: Dropout (0), Enrolled (1), and Graduate (2).
* The model does a great job identifying Dropouts and Graduates. It correctly predicted 224 dropouts and 393 graduates with very few mistakes between the two.
* It had a harder time with the Enrolled group—many enrolled students were misclassified as either dropout or graduate. That’s understandable since being "enrolled" is a middle state, and those students could move in either direction.
* This tells us the model is strong at spotting students at risk or on track, which is great for targeting support. But it struggles more with those in between, where extra data or context could help make better predictions.

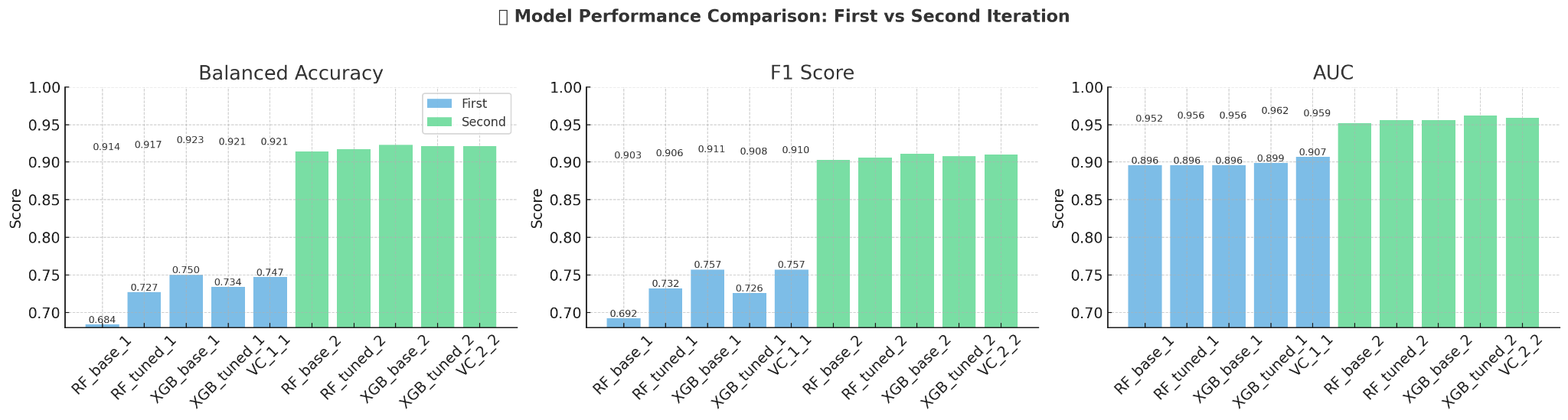
**Second Iteration of Modeling (21 features and 2 targets - dropout & graduate)**

Our goal is still the same — spot students who are likely to drop out and get them help early. To see if a simpler setup performs better, we switched from three classes to two by removing the “Enrolled” group and training fresh models with binary labels: 1 = Dropout, 0 = Graduate.

**What Changed**

| **Step** | **Rationale** |
| --- | --- |
| Kept 21 features | Same feature set as Iteration 1, so the only major change is the target label. |
| Re‑trained RF, XGBoost, and a soft‑voting ensemble | Both RF and XGB handle imbalance well; the VotingClassifier blends their strengths. |
| Quick hyperparameter search | For efficiency, XGB tuning was capped at 50 iterations (enough for a fair, but not exhaustive, search). |

**Result comparison at glance**

****

* Overall Lift: All five binary models beat their tri‑class versions—especially on Balanced Accuracy and F1.
* Top Performers: xgb\_bi edges out on Balanced Accuracy & F1, while vc\_soft\_bi posts the best AUC.
* Tuning Note: Extra XGB tuning didn’t move the needle; a deeper search may squeeze out a bit more, but the gains are already strong.

**Feature Importance Calculation**

The objective of this step was to improve our student dropout prediction model by identifying which features (variables) contributed the most to model performance. Knowing which features are most influential helps us simplify the model, improve interpretability, and reduce noise from less relevant data.

**Methods Used to Calculate Feature Importance**

1. **Random Forest’s Built-in Method**

* Tuned Random Forest trained on binary labels. It measures how much each feature decreases Gini impurity across all trees. Features that reduce impurity more often are ranked as more important.

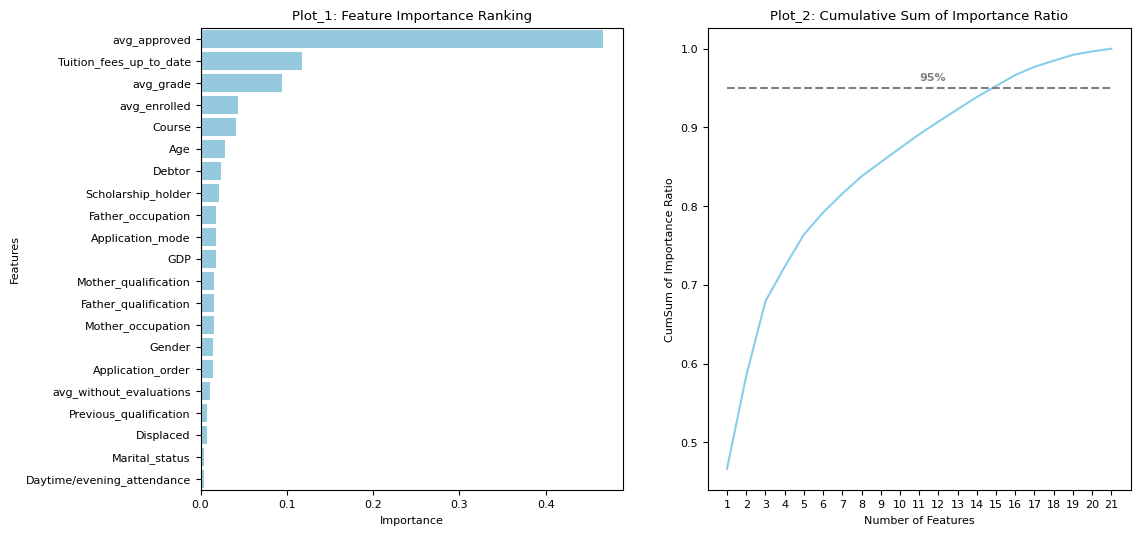
1. **XGBoost Built-in Method**

* Base XGBoost classifier trained on binary labels. Calculates feature importance based on the number of times a feature is used to split trees and how much those splits improve the model (gain-based importance).

1. **Permutation Importance**

* Soft Voting Classifier combining Random Forest and XGBoost. Measures the drop in model performance (typically accuracy or AUC) when a feature’s values are randomly shuffled. A larger drop indicates greater importance.

We calculated importance scores from each method for all 21 features in our dataset, averaged them, and then ranked them. This gave us a reliable, aggregated view of which features mattered most to predicting dropout.



After calculating the average importance scores from all three methods (Random Forest, XGBoost, and Permutation), we visualized the results using two plots:

**Feature Importance Ranking**

* This horizontal bar chart ranks all 21 features from most to least important based on the averaged scores.
* The top feature, avg\_approved, stands out with the highest importance—suggesting that the average number of approved courses plays a key role in predicting dropout.
* Other important features include tuition\_fees\_up\_to\_date, avg\_grade, and avg\_enrolled.
* Features near the bottom like marital\_status or daytime/evening attendance contributed very little and may be candidates for removal in future iterations.

**Cumulative Sum of Importance Ratio**

* This line plot shows how feature importance accumulates as we move down the ranking.
* The dashed gray line at 95% helps us decide how many features we need to retain to capture most of the model's predictive power.
* In this case, we see that the top 15 features contribute to 95% of the total importance.
* This gave us a data-driven justification to keep the first 16 features (since the 15th and 16th features have nearly equal importance) and remove the remaining 5.

**Third Iteration of Modeling (16 features & 2 targets)**

After identifying and removing the five least important features from our feature importance analysis, we retrained our models using the top 16 features. The goal here was to simplify the model while retaining high predictive performance.

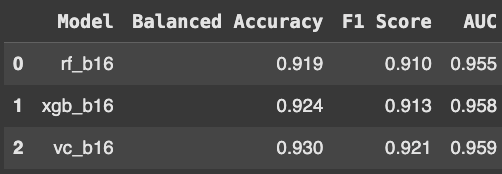
**Final 16 Features Used:**

1. avg\_approved
2. tuition\_fees\_up\_to\_date
3. avg\_grade
4. avg\_enrolled
5. course
6. age
7. debtor
8. scholarship\_holder
9. father\_occupation
10. application\_mode
11. GDP
12. mother\_qualification
13. father\_qualification
14. mother\_occupation
15. gender
16. application\_order

These features were selected based on a combined importance score derived from Random Forest, XGBoost, and permutation importance methods.

**Models Trained**

* rf\_b16: Random Forest with 16 features
* xgb\_b16: XGBoost with 16 features
* vc\_b16: Voting Classifier combining the above two (soft voting)
* Each model was evaluated on the same binary-labeled dataset (Dropout = 1, Graduate = 0), and metrics were computed using Balanced Accuracy, F1 Score, and AUC.



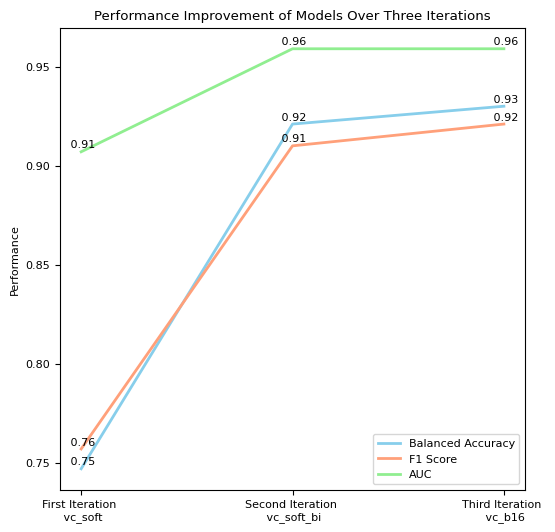
The Voting Classifier vc\_b16, trained with just the top 16 features, achieved the best performance overall—a balanced accuracy of 0.93, an F1 score of 0.921, and an AUC of 0.959. This makes it the optimal model for deployment, combining interpretability, speed, and predictive power.

**Results & Analysis**

**Performance Over Three Iterations**

To compare how the model evolved over time, we visualized the performance of the best-performing model from each iteration. The models compared were:

* Iteration 1: vc\_soft (Voting Classifier using multi-class label)
* Iteration 2: vc\_soft\_bi (Voting Classifier using binary label)
* Iteration 3: vc\_b16 (Voting Classifier using binary label and top 16 features)



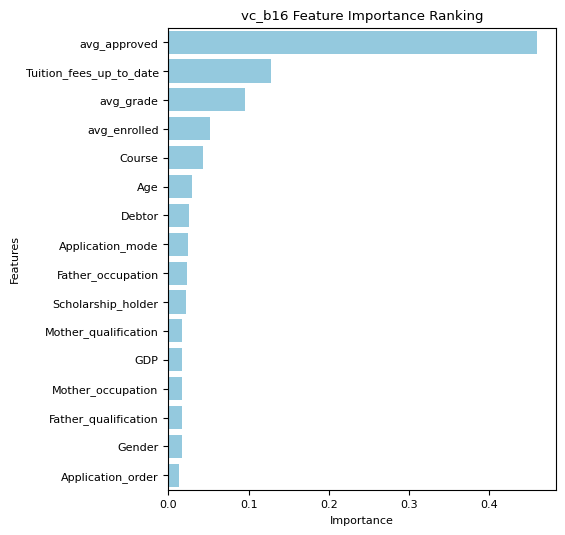
* Balanced Accuracy improved by 24%, rising from 0.75 to 0.93.
* F1 Score rose by 21%, from 0.76 to 0.92.
* AUC increased by 5%, from 0.91 to 0.96.
* This shows consistent performance gains with each refinement: first by simplifying the label from multi-class to binary, and then by reducing the number of features.
* Our final model vc\_b16 offers the best balance of simplicity and performance.

**Feature Importance of Final Model: vc\_b16**

To understand what drives predictions in our best model, we recalculated feature importance using:

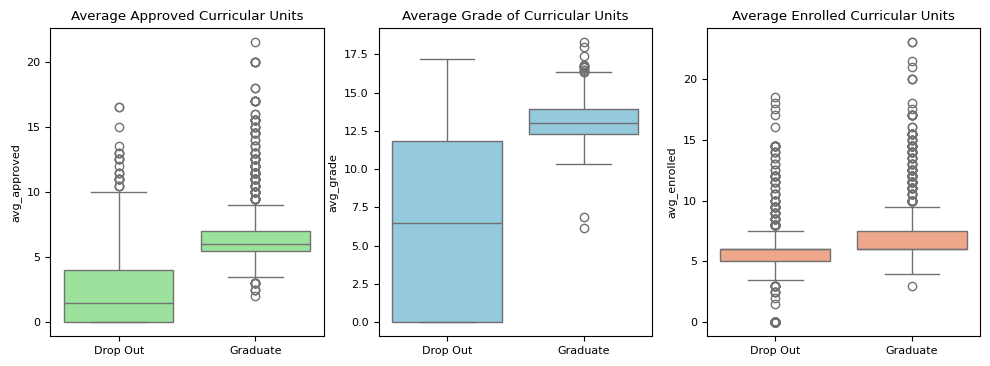
* Random Forest (rf\_b16)
* XGBoost (xgb\_b16)
* Permutation importance on vc\_b16

The scores were averaged and plotted.



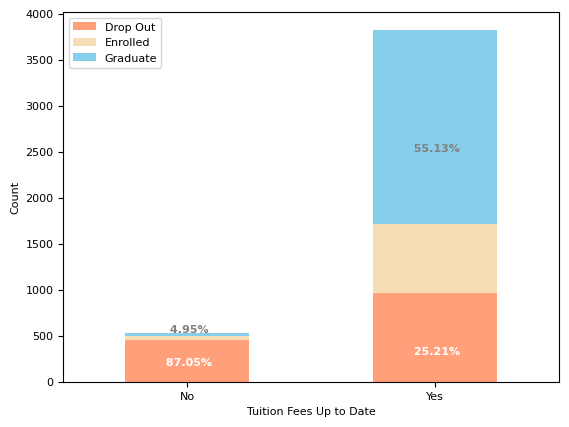
* These represent a mix of Academic performance (approved units, grades, enrollment), Financial standing (tuition status) & Demographics (age)

**Curricular Performance Metrics**



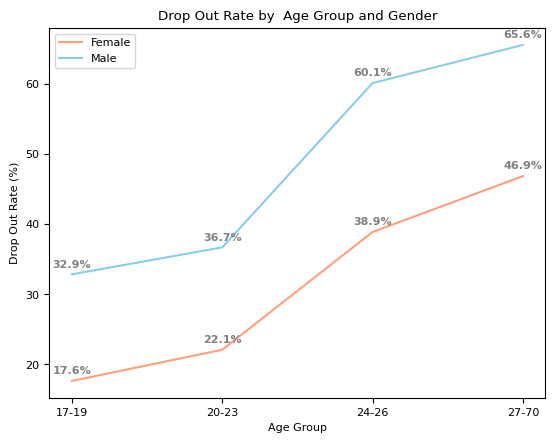
* Boxplots show that students who graduate have higher approved units, higher grades and more curricular enrollments. This confirms that consistent academic performance is a strong driver of retention.

**Tuition Fees and Dropout**



* A bar chart of tuition payment status vs. outcomes showed 87.05% of students who haven’t paid tuition drop out. Only 4.95% of them graduate. Tuition payment is a major risk factor for dropout.

**Age and Gender Trends in Dropout**

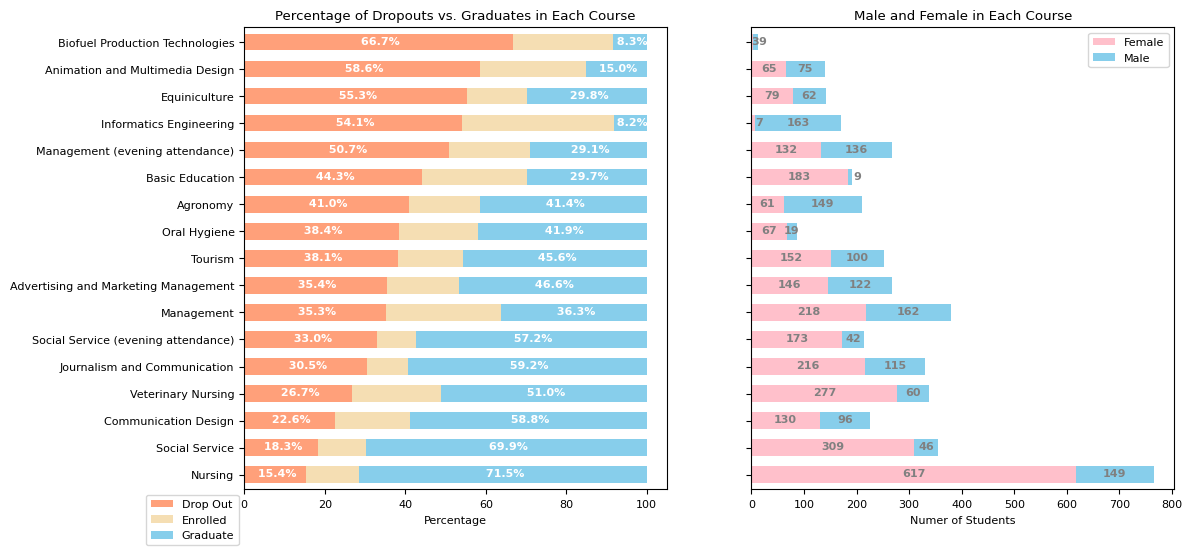
****

* We broke age into four groups (17–19, 20–23, 24–26, 27–70) and examined dropouts by gender.
* Dropout rates increase with age for both genders. However, females consistently have lower dropout rates than males across all age groups.
* Age and gender both play important roles in dropout risk, and should be considered when designing support interventions.

**Dropout rate by course.**

**Note: The course designation was not available in the dataset for better understanding the course name for the respective course code was given hypothetically**

* We looked at dropout vs. graduation rates by course, alongside male-female composition.
* Dropout rates ranged from 15.4% to 66.7% across 17 courses.
* 5 courses had dropout rates above 50%.
* Female students dominate enrollment in most courses, but dropout rates don’t always follow gender proportions, suggesting other factors (course difficulty, support) may play a role.



**Probability of Dropping out in a Simulated Dataset**

1. **Simulating the Impact of Average Grade on Dropout Probability**

In this section, I created a simulated dataset to isolate the effect of a student's average grade on dropout probability. I held most features constant—such as application mode, course, and parental background—using their most frequent values. This created a controlled environment, allowing me to focus solely on how changes in average grade influence dropout risk.

I systematically varied average grades from 0 to 20, repeating each value four times to account for gender and tuition payment status differences. This ensured a broad and balanced representation.

Using np.repeat() and the pre-trained model (vc\_b16), I predicted dropout probabilities across the simulated records. This approach provided clear insights into how academic performance alone impacts dropout risk, free from the noise of other confounding factors.

1. **Simulating the impact of average number of approved courses on dropout probability**

In this section, I created a simulated dataset to isolate the impact of a student's average number of approved courses on dropout probability. I held all other features constant—using the most common or average values from the original dataset—to create a controlled environment where the primary variable of interest was avg\_approved.

I varied avg\_approved systematically from 0 to 25, repeating each value across four combinations of gender and tuition payment status. This ensured a comprehensive view of how approval count interacts with key student demographics.

Using np.repeat() and the trained model (vc\_b16), I predicted dropout probabilities for each simulated record. I also formatted the gender and tuition variables into readable labels and combined them into a new column Gen\_Tui for easier analysis.

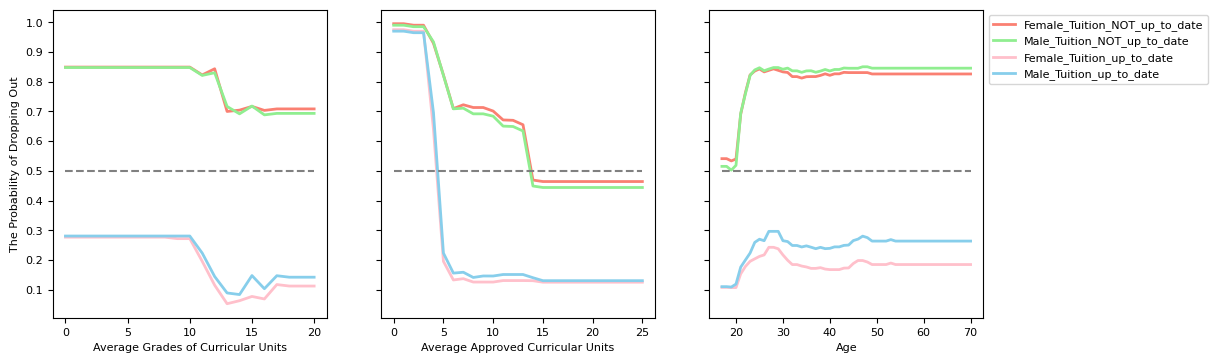
1. **Simulating the impact of student's age influence on dropout probability**

In this section, I created a simulated dataset to explore how a student’s age influences their likelihood of dropping out. To isolate the effect of age, I held all other variables constant by using the most frequent or average values from the original dataset. This ensured a controlled environment where the only changing variable was Age.

I systematically varied age from 17 to 70, repeating each value across four combinations of gender and tuition payment status. This setup allowed me to examine how dropout probability changes with age across different student demographics.

Using np.repeat() and the trained model (vc\_b16), I predicted dropout probabilities for each simulated student. I also formatted gender and tuition status into readable categories and combined them into a new column Gen\_Tui for easier comparison.

**Analysis of the three simulated dataset**



The simulated datasets provide strong evidence that tuition payment status is a key driver of dropout risk in our final model. Across nearly all scenarios, students with unpaid tuition consistently show a predicted dropout probability above 0.5, reinforcing the weight this feature carries.

While our analysis cannot definitively answer whether non-payment leads to disengagement or if institutions preemptively remove students for non-payment, the effect is clear either way—tuition status is a major red flag.

Beyond tuition, the simulations also revealed distinct, interpretable patterns:

* Average Grade: Dropout risk remains high when grades are below 10, then sharply declines between 10 and 13, and levels off beyond 13. This highlights a performance threshold that may trigger intervention.
* Approved Units: There are two noticeable drop points in dropout risk—once students exceed 5 approved units and again past 14. This suggests progress milestones that reduce risk substantially.
* Age at Enrollment: Students under 20 show consistently low dropout risk. However, the risk climbs quickly between ages 20 and 23, then stabilizes at a higher level beyond that.

These simulations confirm that academic performance, engagement (via course approvals), and age all influence dropout risk—but financial standing appears to be the most immediate and powerful signal the model picks up on.

**Conclusion**

In this project, I developed and evaluated multiple classification models to predict student dropout and academic success. Through three iterative modeling phases, I gradually simplified the model—reducing the feature set from 21 to 16 and switching from multi-class to binary classification. These changes led to significant performance gains:

* Balanced Accuracy improved from 0.75 to 0.93 (+24%)
* F1 Score increased from 0.76 to 0.92 (+21%)
* AUC rose from 0.91 to 0.96 (+5%)

These improvements highlight a key lesson: removing noise—such as low-impact features and ambiguous labels (like 'Enrolled')—can both simplify a model and enhance its accuracy. The final model, vc\_b16, proved to be the most effective and interpretable version.

**What Drives Dropout? Key Feature Insights**

To understand which factors most influence dropout, I visualized the top features identified by the final model and explored their relationships with student outcomes:

* Academic performance is a strong predictor: graduates consistently had more approved curricular units and higher average grades.
* Tuition payment status stood out as a critical factor:
  + Students with unpaid tuition had a dropout rate of 87.05% and graduation rate of just 4.95%.
  + In contrast, those who paid on time had a significantly lower dropout rate of 25.21% and a graduation rate of 55.13%.
* Age trends revealed that dropout risk increases as students get older. However, female students consistently showed lower dropout rates across all age groups.
* Course of enrollment also matters: dropout rates ranged from 15.4% to 66.7%, and five out of 17 courses had dropout rates above 50%.

**Simulated Analysis: Deeper Understanding of Dropout Risk**

To further isolate the impact of key variables, I built three simulated datasets that varied average grade, approved units, and age, while keeping other factors fixed. These simulations revealed:

* Students with unpaid tuition showed dropout probabilities above 0.5 in nearly all scenarios—confirming this as a dominant feature in the model.
* Average grade: Dropout probability drops sharply between grades 10 and 13, then levels off.
* Approved units: Significant drops in dropout risk appear when students exceed 5 and 14 units.
* Age: Risk increases noticeably as enrollment age moves from 20 to 23, then plateaus.

I’m not sure if a student's failure to pay tuition on time strongly indicates that they are likely to drop out, or if the university deregisters students from their courses when they have not paid all the required tuition fees by the specified deadlines. Nevertheless, regardless of the tuition fee factor, the patterns of dropout probability is clear in the simulated datasets

**Students Most at Risk: Who Needs Support?**

* Tuition fees not up to date
* Average grade below 10 (particularly in the first two semesters)
* Fewer than 14 approved units, especially if below 5
* Age over 23 at the time of enrollment
* Male students under these same conditions
* Students enrolled in high-risk courses (with >50% dropout rates)

This project not only resulted in a high-performing predictive model but also offered actionable insights that institutions can use to better support at-risk students—and ultimately reduce dropout rates.

**References**

1. Random Forest Feature Importance Scikit-learn. (n.d.). RandomForestClassifier. Retrieved from <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>
2. XGBoost Feature Importance XGBoost Documentation. (n.d.). XGBClassifier API Reference. Retrieved from <https://xgboost.readthedocs.io/en/latest/python/python_api.html>
3. Permutation Importance Scikit-learn. (n.d.). Permutation features importance. Retrieved from <https://scikit-learn.org/stable/modules/permutation_importance.html>
4. Voting Classifier (Hard vs. Soft Voting) Scikit-learn. (n.d.). VotingClassifier. Retrieved from <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.VotingClassifier.html>
5. Class Weight Balancing Scikit-learn. (n.d.). compute\_sample\_weight. Retrieved from <https://scikit-learn.org/stable/modules/generated/sklearn.utils.class_weight.compute_sample_weight.html>
6. RandomForestClassifier Library Scikit-learn. (n.d.). RandomForestClassifier. Retrieved from <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>
7. XGBClassifier Library XGBoost Documentation. (n.d.). XGBClassifier API Reference. Retrieved from <https://xgboost.readthedocs.io/en/latest/python/python_api.html>
8. compute\_sample\_weight Function Scikit-learn. (n.d.). compute\_sample\_weight. Retrieved from <https://scikit-learn.org/stable/modules/generated/sklearn.utils.class_weight.compute_sample_weight.html>
9. RandomizedSearchCV Scikit-learn. (n.d.). RandomizedSearchCV. Retrieved from <https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html>
10. Simulated Dataset Concept MOSTLY AI. (n.d.). What is synthetic data?. Retrieved from <https://mostly.ai/blog/data-simulation>

**Note: LLM like DeepSeek and ChaGPT were used to learn concepts, rationalize and brainstorm. These models were treated as tutors throughout the project. Debugging, syntax errors, any other errors and code structures were all solved with the help of LLM models.**

**Group Member Contribution**

| **Team Member** | **Contribution Highlight** |
| --- | --- |
| **Balram Iyengar** | - Led overall model design and iteration planning  - Implemented Random Forest, XGBoost, and Voting Classifier models  - Designed and analyzed the simulated datasets  - Managed feature importance strategy and performance optimization across iterations |
| **Maneesh Rao** | - Assisted with EDA and data preprocessing  - Tuned hyperparameters for XGBoost  - Contributed to interpretation of dropout trends by course and gender |
| **Dhanush** | - Supported feature selection and visualization  - Helped interpret simulation results  - Drafted summary points and refined visual insights |