MINI PROJECT

Topic : Predict student dropout and Success adcademic

Group 4
Members :

Som Chann Reaksmey	e20220981
Sophat Vitou	e20220813
Sour Alya	e20220969
Yory Sreykhuoch	e20220389

CONTENT

01 Introduction

What is Student Dropout?

- Many students leave school before completing their education
- Affects both students and society (job opportunities, economy, etc.).

Why is this important?

- High dropout rates lead to unemployment and poverty.
- Schools need to identify at-risk students early

02 Problem Statemen

- High dropout rates harm students' futures and school reputation.
- Manual identification of at-risk students is slow and inaccurate.
- Need for an automated, reliable prediction system using student data.
- Goal: Build a model to accurately predict dropout vs. success.

Objective

Main Goal

To build a machine learning system that predicts student dropout and academic success based on personal, academic, and socio-economic data.

Special Goal

- To explore and analyze student-related data
- To train and evaluate various ML models
- To select the most effective model for prediction
- To deploy the model using a web application for user interaction

03 Methodology

Data collection

Dataset sourced from Kaggle and public records.
Includes key variables: academic performance, attendance, socioeconomic factors.

Data Preprocessing:

- Clean data: Handle missing values, remove duplicates, normalize features.
- Feature Engineering: Create meaningful variables (e.g., average grade, attendance rate).

Explore Data Analysis

- Chi-square Variable Test on Categorical features
- ANOVA test on Numerical features
- Spearman Correlation Matrix for analyse and visualize features

Model Selection:

- Random Forest
- Extreme Gradient Boost
- Logistic Regression
- K-Nearest Neighbor
- Support-Vector Classification
- Naive Bayes

Data Collection

Input Data

dataset.csv (470.86 kB)

Detail | Compact | Column | 10 of 35 columns

# 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 ... Whether the student attends classes during the day or in the evening. (Categorical)	# Previou The qualif obtained before en educatior
1 – 6	1 – 18	0 – 9	1 – 17	0 – 1	1
1	8	5	2	1	1
1	6	1	11	1	1

📊 Dataset Overview

This dataset, collected from Kaggle, contains information students, such as:

👩‍🎓 Demographics

💵 Socio-economic background

📚 Academic performance

04 Data Preprocessing

Clean column names

```
[ ]  df.rename(columns = {"Nacionality": "Nationality",
                          "Mother's qualification": "Mother_qualification",
                          "Father's qualification": "Father_qualification",
                          "Mother's occupation": "Mother_occupation",
                          "Father's occupation": "Father_occupation",
                          "Age at enrollment": "Age"}, inplace = True)

     # Replace white space in he columns name with underscore
     df.columns =df.columns.str.replace(' ','_')

     # Remove the parenthesis

     df.columns = df.columns.str.replace('(','')
     df.columns = df.columns.str.replace(')','')
     df.columns
```

MISSING VALUE DETECTION

```
Standard missing values:
Series([], dtype: int64)

Disguised missing values:
Age                                  0
Curricular_units_1st_sem_grade     718
Curricular_units_2nd_sem_grade     870
dtype: int64
```

DATA TYPE CONVERSION

```
[ ]  categorical_cols = ['Marital_status', 'Application_mode', 'Application_order', 'Course',
           'Daytime/evening_attendance', 'Previous_qualification', 'Nationality',
            'Mother_qualification', 'Father_qualification', 'Mother_occupation',
            'Father_occupation', 'Displaced', 'Educational_special_needs', 'Debtor',
            'Tuition_fees_up_to_date', 'Gender', 'Scholarship_holder',
           'International', 'Target']

     df[categorical_cols]= df[categorical_cols].astype('category')
     df.info()
```

Handle graduates with 0 grades (invalid data)

```
df = df[~((df['Target'] == 'Graduate') &
          ((df['Curricular_units_1st_sem_grade'] == 0) |
           (df['Curricular_units_2nd_sem_grade'] == 0)))]
```

05 Explore data analysis

There are 34 features in this dataset. I will examine their relationship with the target variable, which is a three-class categorical data. The features that have no association with the label will be the potential variables to be removed from modeling.

Chi-Square Independence Test for Categorical Variables

	Variable	P_value
0	Marital_status	0.00000
1	Application_mode	0.00000
2	Application_order	0.00000
3	Course	0.00000
4	Daytime/evening_attendance	0.00000
5	Previous_qualification	0.00000
7	Mother_qualification	0.00000
8	Father_qualification	0.00000
13	Debtor	0.00000
9	Mother_occupation	0.00000
10	Father_occupation	0.00000
11	Displaced	0.00000
15	Gender	0.00000
14	Tuition_fees_up_to_date	0.00000
16	Scholarship_holder	0.00000
6	Nationality	0.24666
17	International	0.52608
12	Educational_special_needs	0.63042

Identify which numerical features are statistically different between the target groups (e.g., “Graduate” vs. “Dropout” vs. “Enrolled”) using ANOVA (Analysis of Variance) and effect size (Eta Squared).

ANOVA Results (Sorted by P-value):

	Variable	P_value	Eta_squared
0	Age	0.00000	0.06313
2	Curricular_units_1st_sem_enrolled	0.00000	0.04833
3	Curricular_units_1st_sem_evaluations	0.00000	0.01880
4	Curricular_units_1st_sem_approved	0.00000	0.32558
5	Curricular_units_1st_sem_grade	0.00000	0.30773
10	Curricular_units_2nd_sem_approved	0.00000	0.44263
9	Curricular_units_2nd_sem_evaluations	0.00000	0.04409
8	Curricular_units_2nd_sem_enrolled	0.00000	0.06297
12	Curricular_units_2nd_sem_without_evaluations	0.00000	0.00867
11	Curricular_units_2nd_sem_grade	0.00000	0.40359
7	Curricular_units_2nd_sem_credited	0.00001	0.00527
6	Curricular_units_1st_sem_without_evaluations	0.00002	0.00503
1	Curricular_units_1st_sem_credited	0.00008	0.00433
13	Unemployment_rate	0.00259	0.00274
14	GDP	0.00711	0.00227

Features Important

```
[ ]  important_feature=stud_selected.drop(['Curricular_units_1st_sem_credited',
                          'Curricular_units_2nd_sem_credited',
                          'Curricular_units_1st_sem_without_evaluations',
                          'Curricular_units_2nd_sem_without_evaluations',
                          'Unemployment_rate',
                          'GDP'],axis=1)
     important_feature.columns
```

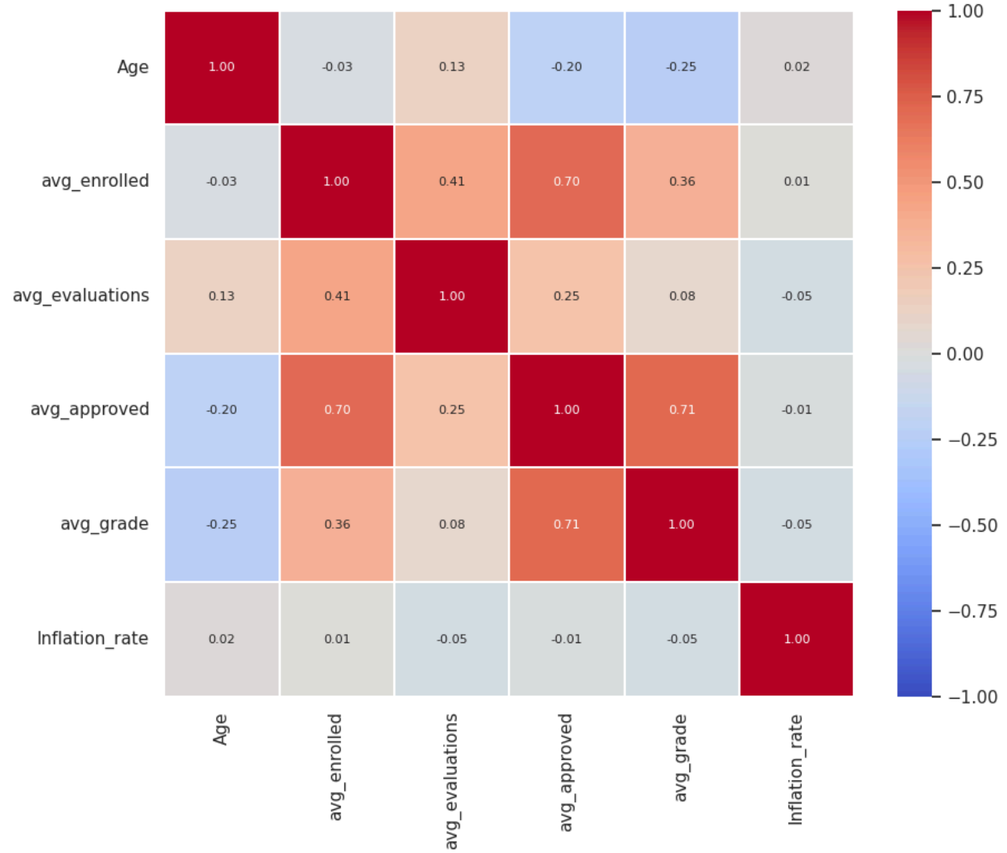
```
stud_selected = df.drop(['Nationality', 'International'
                         'Educational_special_needs'],
                              axis = 1)
stud_selected.columns
```

```
Index(['Marital_status', 'Application_mode', 'Application_order', 'Course',
       'Daytime/evening_attendance', 'Previous_qualification',
       'Mother_qualification', 'Father_qualification', 'Mother_occupation',
       'Father_occupation', 'Displaced', 'Debtor', 'Tuition_fees_up_to_date',
       'Gender', 'Scholarship_holder', 'Age',
       'Curricular_units_1st_sem_enrolled',
       'Curricular_units_1st_sem_evaluations',
       'Curricular_units_1st_sem_approved', 'Curricular_units_1st_sem_grade',
       'Curricular_units_2nd_sem_enrolled',
       'Curricular_units_2nd_sem_evaluations',
       'Curricular_units_2nd_sem_approved', 'Curricular_units_2nd_sem_grade',
       'Inflation_rate', 'Target_encoded'],
      dtype='object')
```

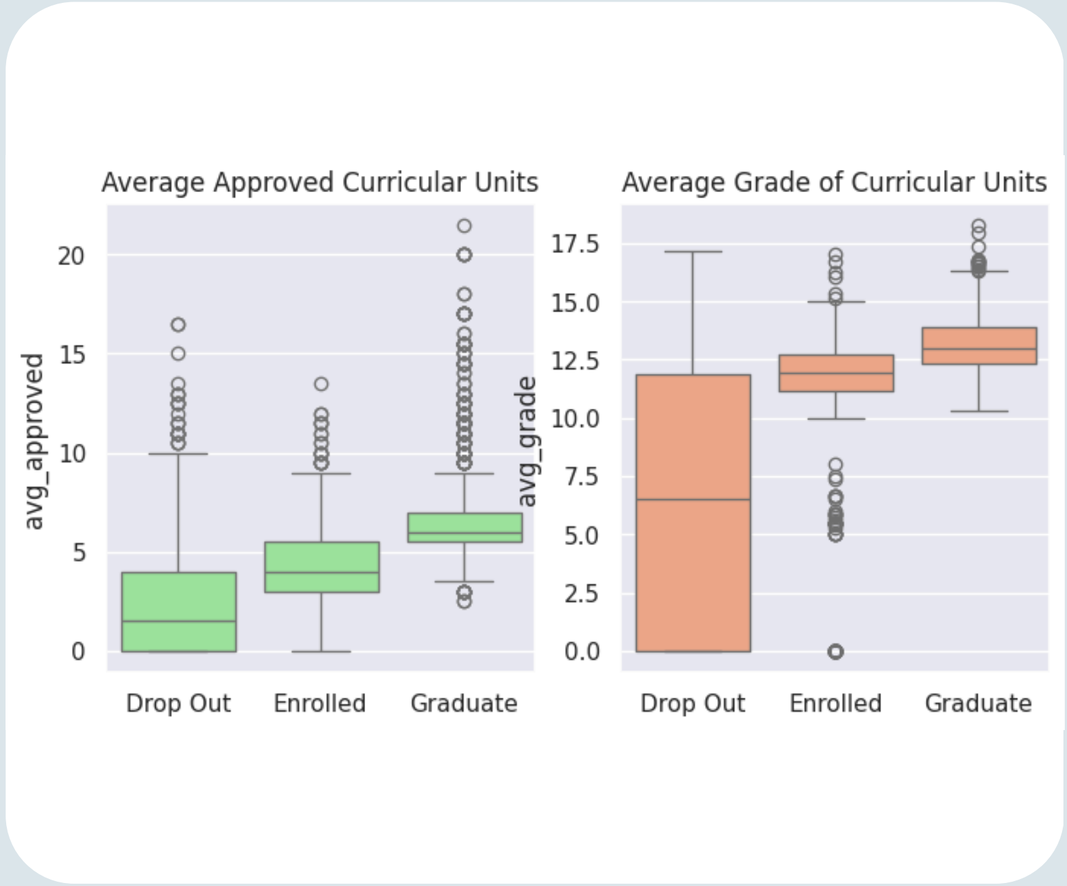
After cleaning the data, I removed irrelevant or independent variables like identifiers (e.g., student IDs) and only kept variables that could influence the target (student dropout or success).

Data Analysis

Spearman Correlation Matrix:
measures the strength and direction of the monotonic relationship between two variables, making it robust to outliers and suitable for both linear and non-linear associations.

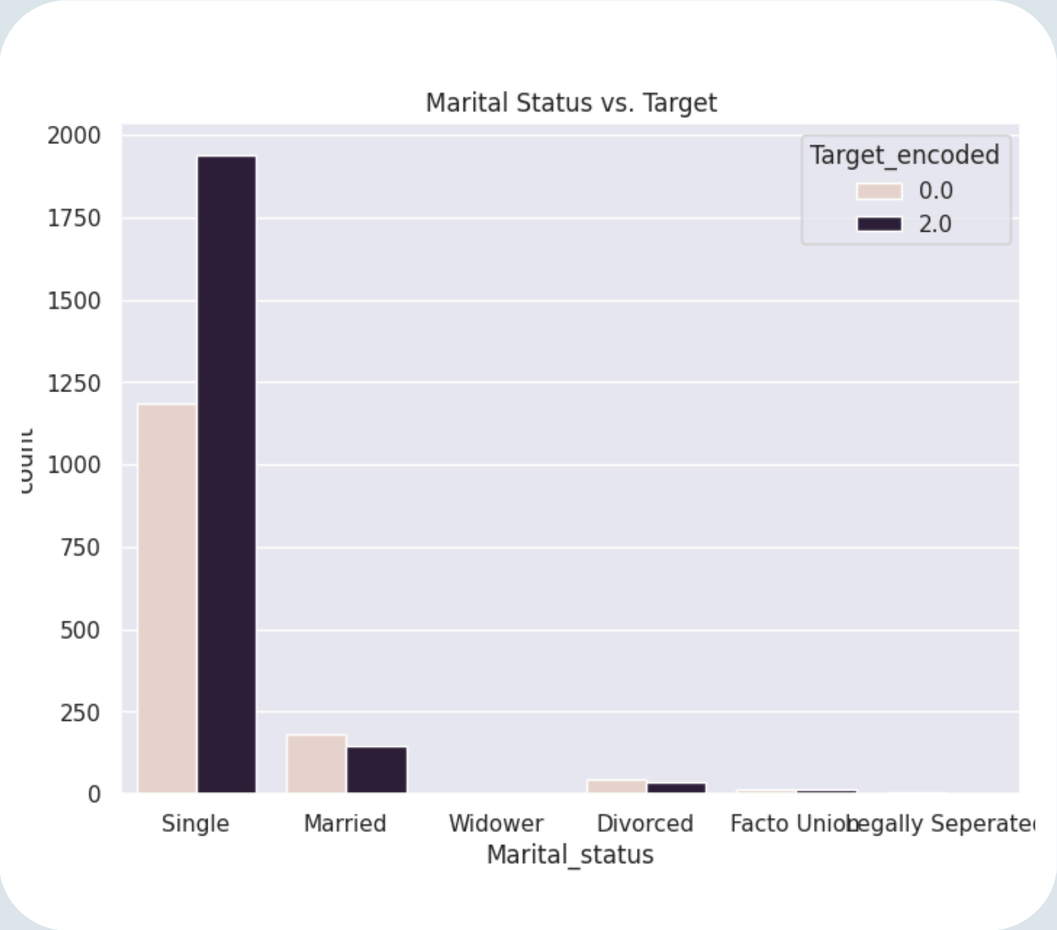


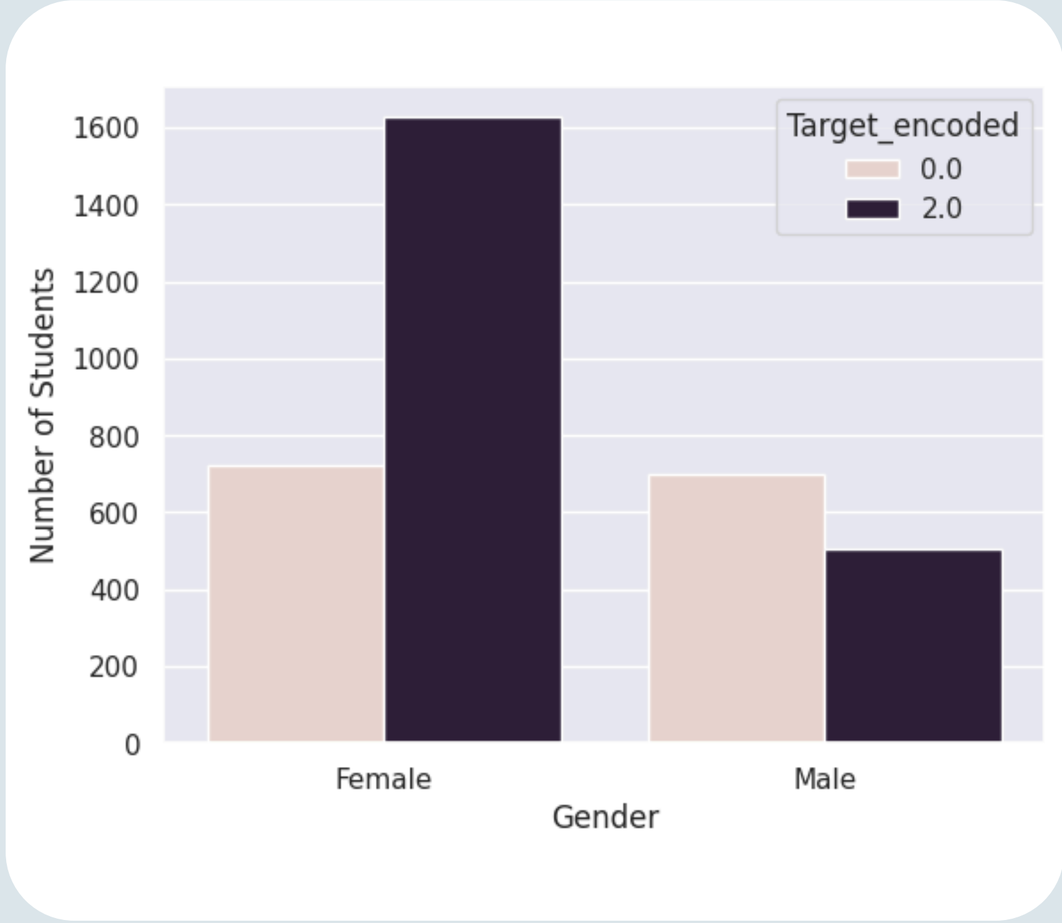
Boxplots:
compare te central tendency, spread, and presence of outliers across groups



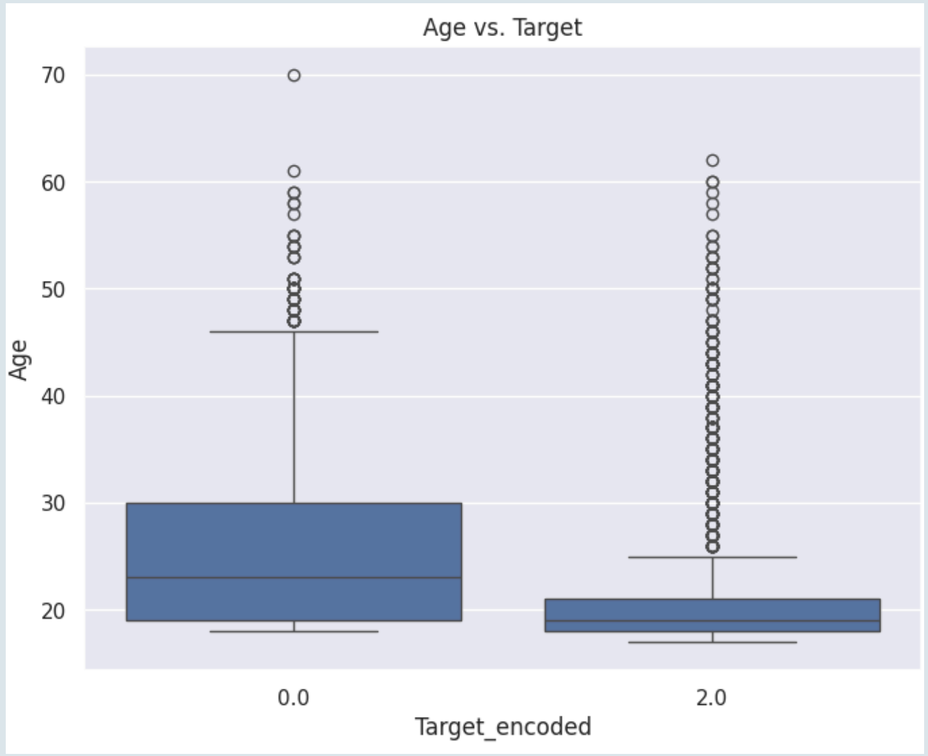
Data Analysis

Grouped Barchart

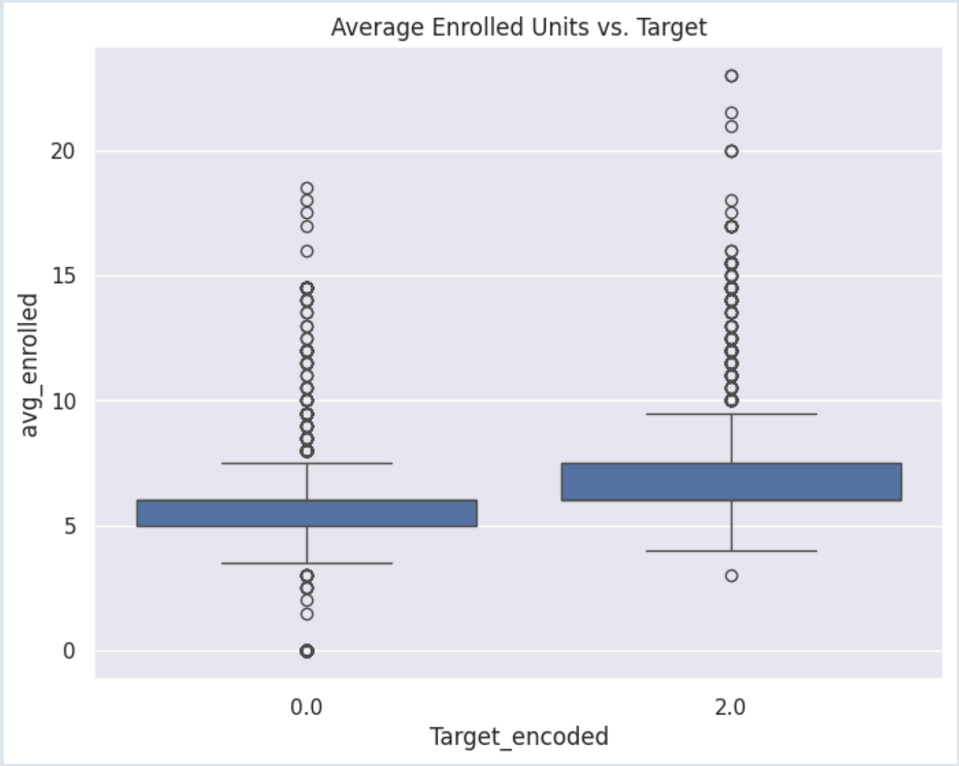




Visualization



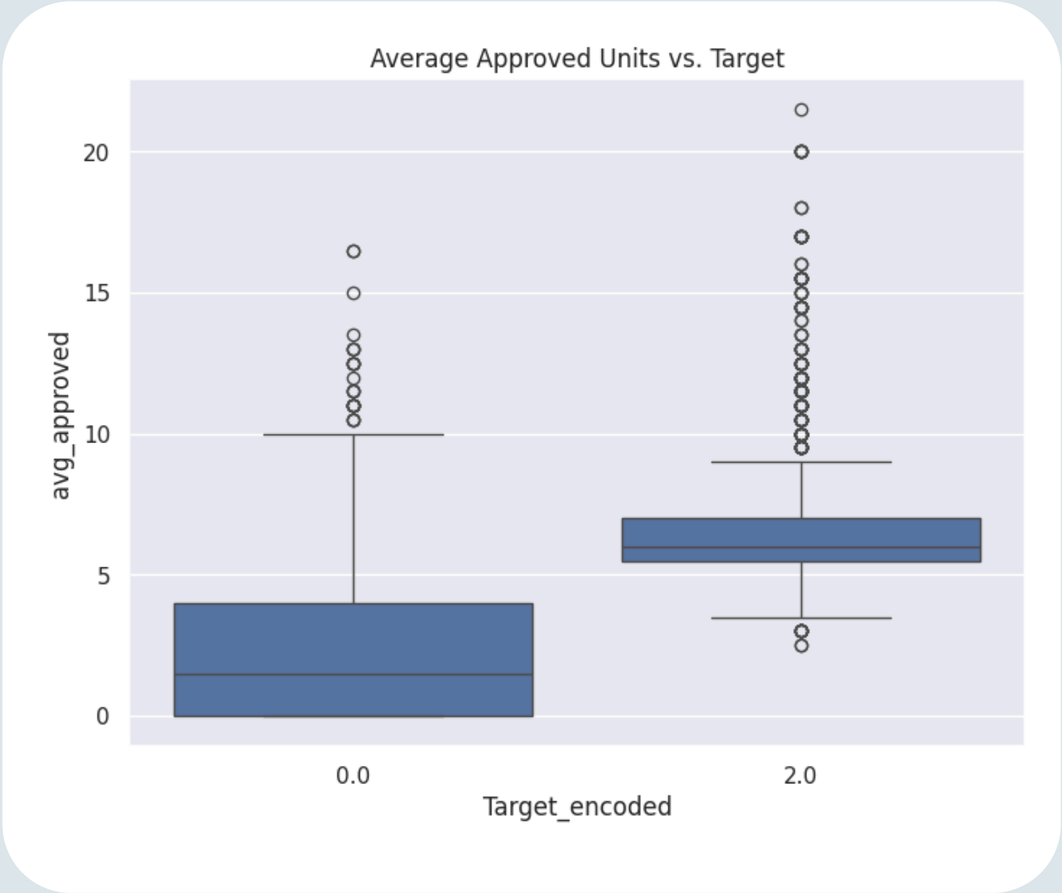
This boxplot visualizes how the distribution of student ages varies across the different student outcome categories.

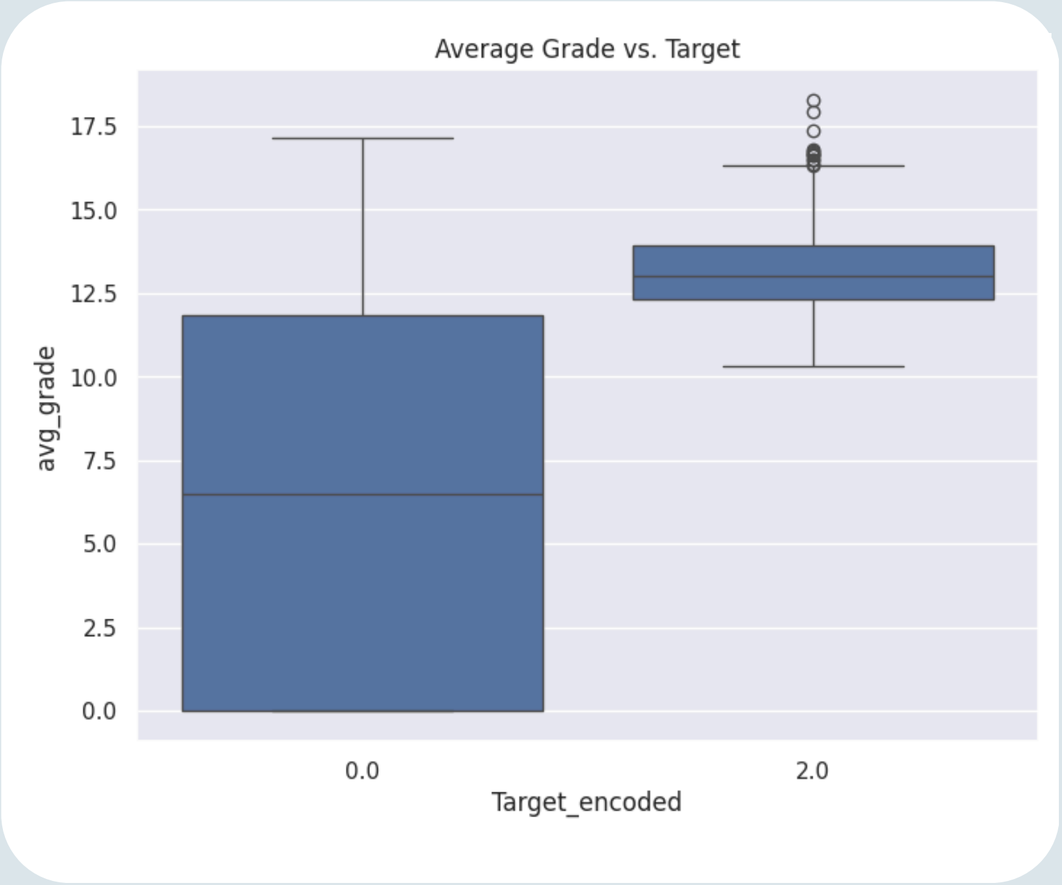


this code segment visually compares the distribution of the average number of enrolled curricular units for students based on their final academic outcome

Data Analysis

Boxplot:



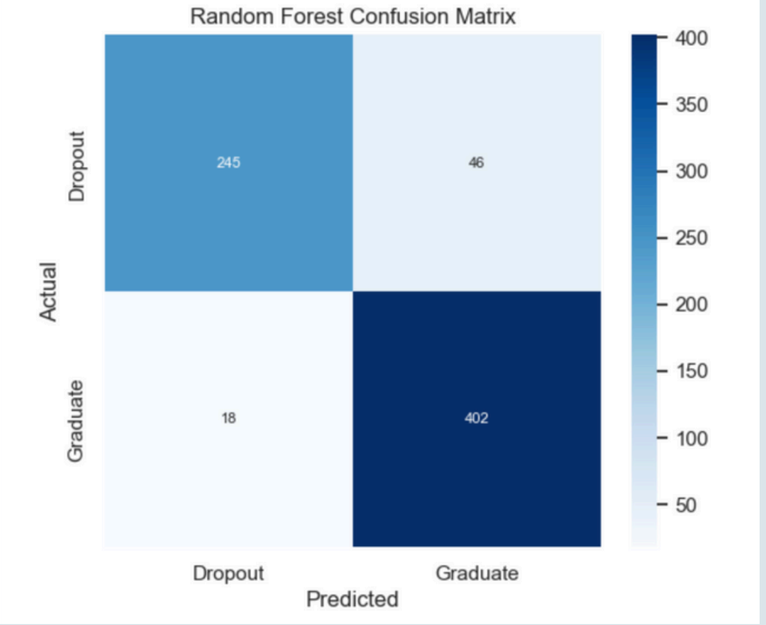


07 Model Building and Evaluation

- We fit the testing set into 6 models, each with hyperparameter tuning, and evaluate each one to determine the best model
- Model used:
  - Randon Forest
  - Extreme Gradient Boost
  - Logistic Regression
  - K-Nearest Neighbor
  - Support-Vector Classification
  - Naive Bayes
- Evaluation metrics:
  - Balanced Accuracy, F1-Score, AUC Score
  - Classification report
  - Confusion Matrix
  - 5-folds Nested Cross-Validation
  - Cost-sensitive metrics

```
tuned_rf_bi Performance:
Balanced Accuracy: 0.9
F1 Score: 0.926
AUC score: 0.958
```

```
Random Forest Classification Report:
              precision    recall  f1-score   support

         0.0       0.93      0.84      0.88       291
         1.0       0.90      0.96      0.93       420

    accuracy                           0.91       711
   macro avg       0.91      0.90      0.91       711
weighted avg       0.91      0.91      0.91       711
```

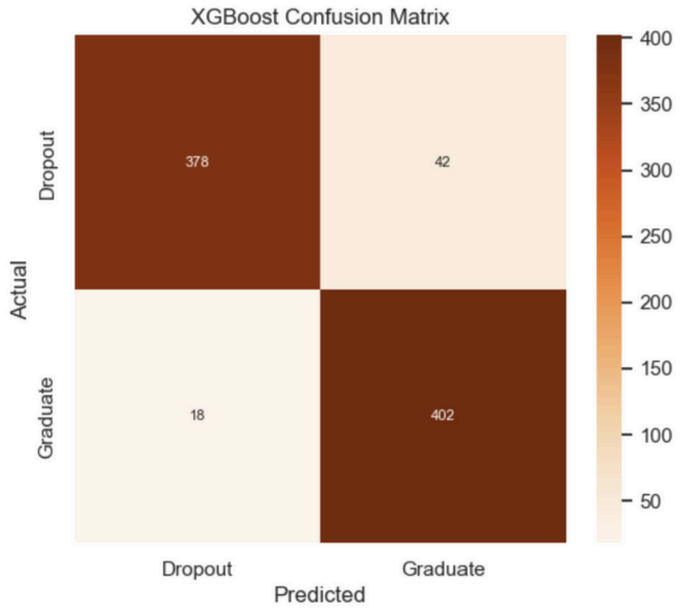


07 Model Building and Evaluation

```
tuned_xgb_bi performance:
Balanced accuracy: 0.929
F1 score: 0.931
AUC score: 0.969
```

XGBoost Classification Report:

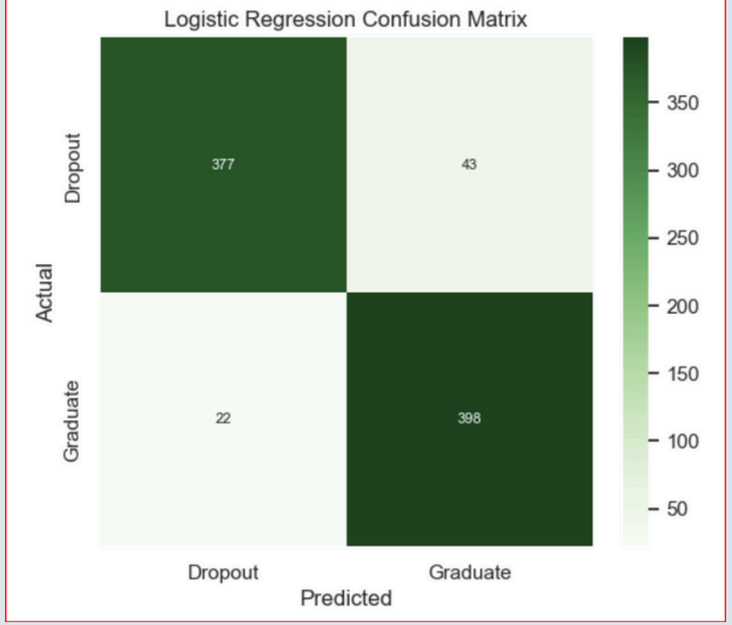
	precision	recall	f1-score	support
0.0	0.95	0.90	0.93	420
1.0	0.91	0.96	0.93	420
accuracy			0.93	840
macro avg	0.93	0.93	0.93	840
weighted avg	0.93	0.93	0.93	840




```
Tuned Logistic Regression Performance:
Balanced Accuracy: 0.923
F1 Score: 0.925
AUC score: 0.966
```

Logistic Regression Classification Report:

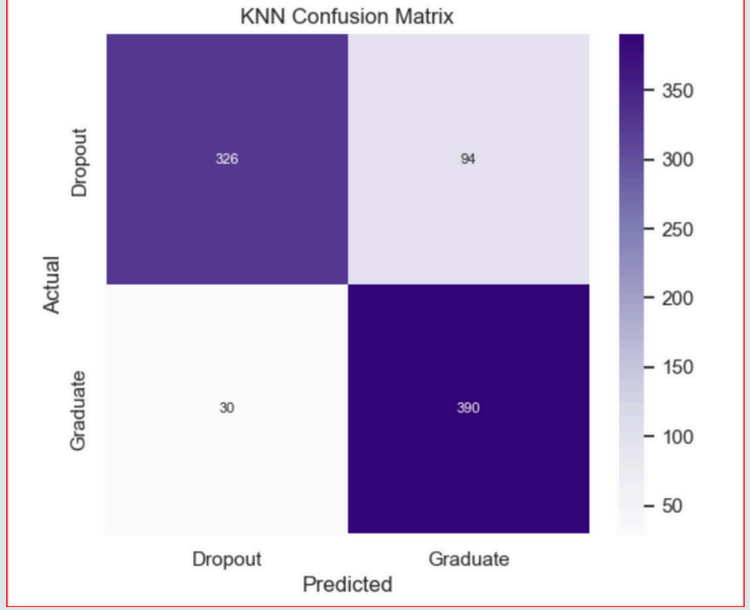
	precision	recall	f1-score	support
0.0	0.94	0.90	0.92	420
1.0	0.90	0.95	0.92	420
accuracy			0.92	840
macro avg	0.92	0.92	0.92	840
weighted avg	0.92	0.92	0.92	840




```
Tuned KNN Performance:
Balanced Accuracy: 0.852
F1 Score: 0.863
AUC score: 0.922
```

KNN Classification Report:

	precision	recall	f1-score	support
0.0	0.92	0.78	0.84	420
1.0	0.81	0.93	0.86	420
accuracy			0.85	840
macro avg	0.86	0.85	0.85	840
weighted avg	0.86	0.85	0.85	840



07 Model Building and Evaluation

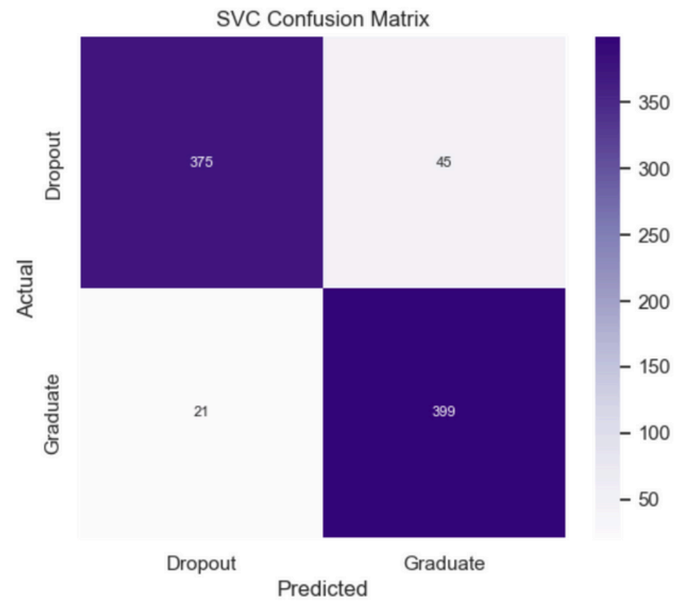
```
Tuned SVC Performance:
Balanced Accuracy: 0.921
F1 Score: 0.924
AUC score: 0.964
```

```
tuned_rf_bi Performance:
Balanced Accuracy: 0.914
F1 Score: 0.918
AUC score: 0.969
```

```
Tuned Naive Bayes Performance:
Balanced Accuracy: 0.862
F1 Score: 0.869
AUC score: 0.928
```

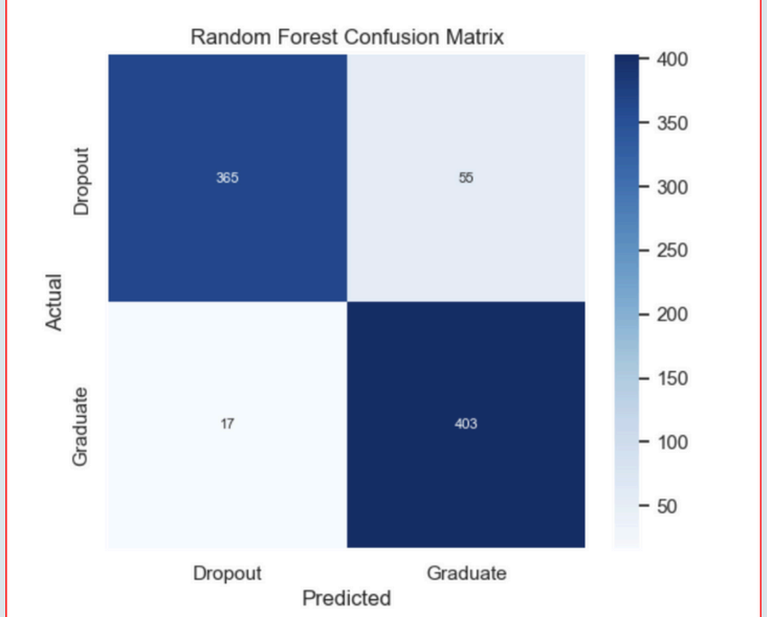
SVC Classification Report:

	precision	recall	f1-score	support
0.0	0.95	0.89	0.92	420
1.0	0.90	0.95	0.92	420
accuracy			0.92	840
macro avg	0.92	0.92	0.92	840
weighted avg	0.92	0.92	0.92	840



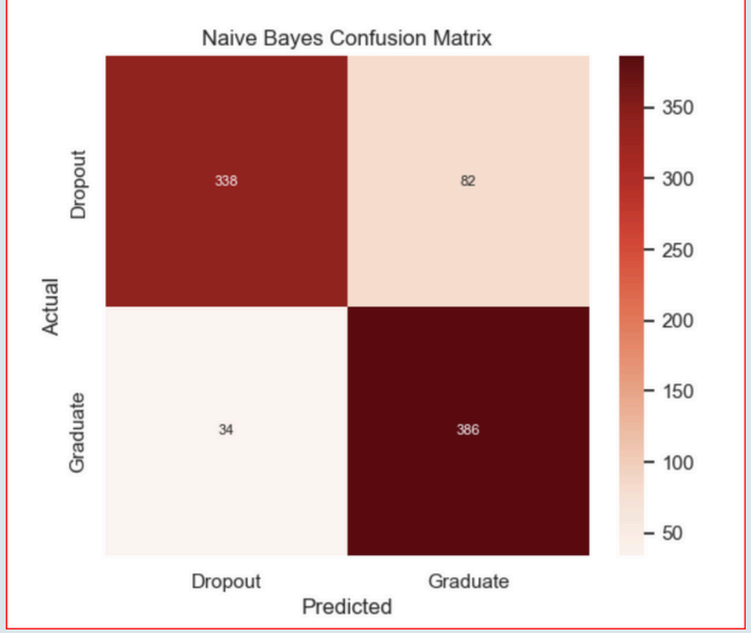
Random Forest Classification Report:

	precision	recall	f1-score	support
0.0	0.96	0.87	0.91	420
1.0	0.88	0.96	0.92	420
accuracy			0.91	840
macro avg	0.92	0.91	0.91	840
weighted avg	0.92	0.91	0.91	840



Naive Bayes Classification Report:

	precision	recall	f1-score	support
0.0	0.91	0.80	0.85	420
1.0	0.82	0.92	0.87	420
accuracy			0.86	840
macro avg	0.87	0.86	0.86	840
weighted avg	0.87	0.86	0.86	840



Evaluation of Model

Comparing performance metrics on all 6 models

	Model	Balanced Accuracy	F1 Score	AUC
0	XGBoost	0.915	0.938	0.968
1	Logistic Regression	0.915	0.934	0.960
2	Random Forest	0.900	0.926	0.958
3	SVC	0.908	0.928	0.957
4	KNN	0.839	0.894	0.923
5	Naive Bayes	0.840	0.881	0.912

```
Best model based on AUC: XGBoost
```

Cost-sensitive metrics

```
Random Forest - Total Cost: 136, Average Cost per Sample: 0.1913
XGBoost - Total Cost: 110, Average Cost per Sample: 0.1547
SVC - Total Cost: 161, Average Cost per Sample: 0.2264
```

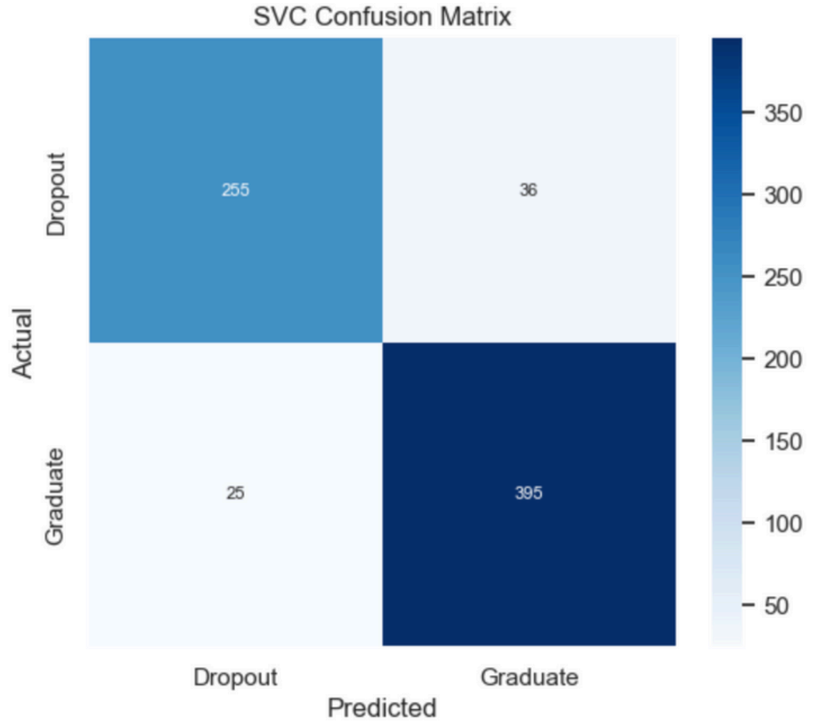
Perform 5-Folds Nested Cross Validation on the best 3 models

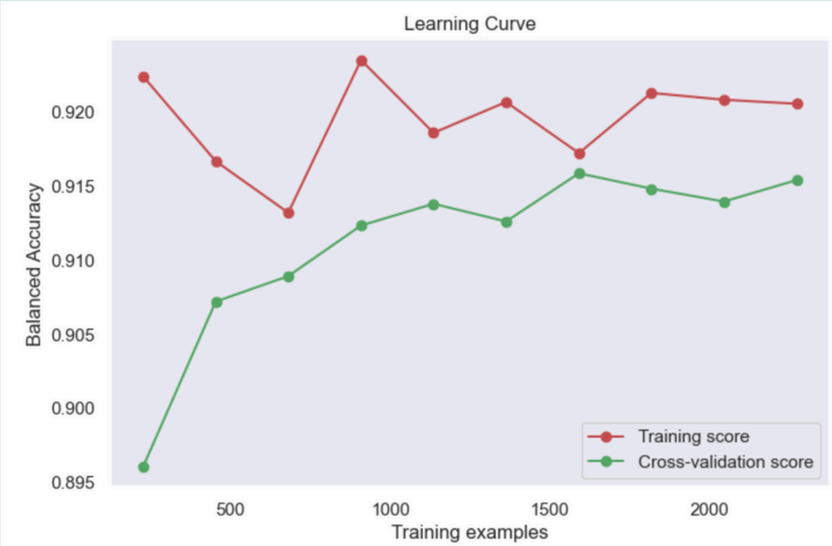
```
Random Forest Nested CV Balanced Accuracy: Mean=0.907, Std=0.012
XGBoost Nested CV Balanced Accuracy: Mean=0.909, Std=0.009
SVC Nested CV Balanced Accuracy: Mean=0.916, Std=0.008
```

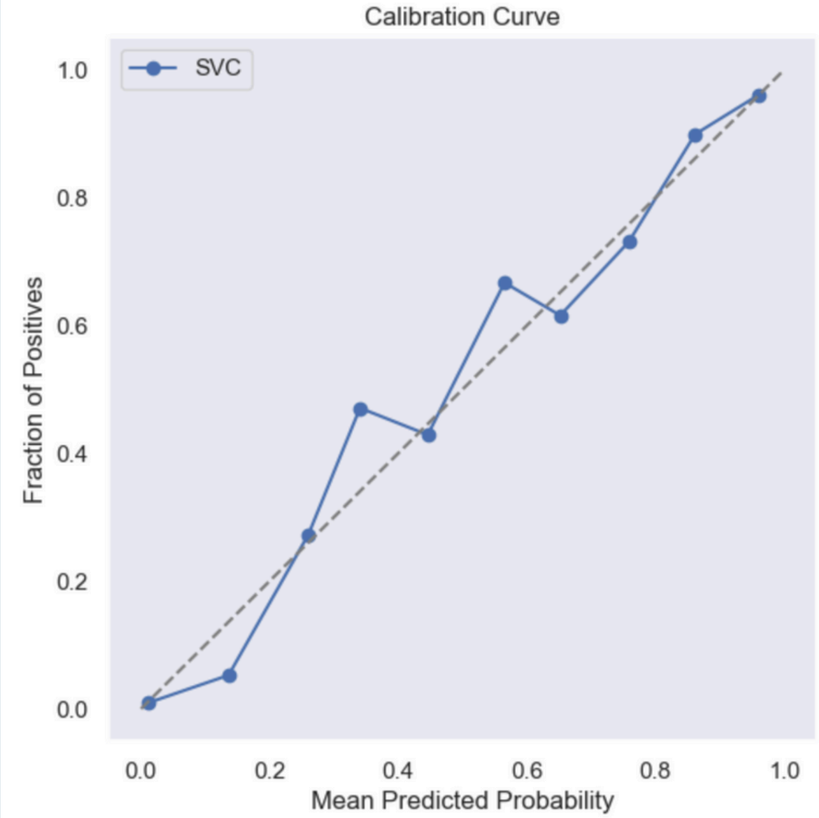
	Random Forest	XGBoost	SVC
0	0.889587	0.894063	0.910202
1	0.910150	0.906531	0.911711
2	0.925607	0.921971	0.932322
3	0.899925	0.908736	0.913238
4	0.907908	0.914545	0.910948

08 Model Training

```
SVC Classification Report:
              precision    recall  f1-score   support

         0.0       0.91      0.88      0.89       291
         1.0       0.92      0.94      0.93       420

    accuracy                           0.91       711
   macro avg       0.91      0.91      0.91       711
weighted avg       0.91      0.91      0.91       711

Balanced Accuracy: 0.9083824251350024
F1 Score: 0.9283196239717979
AUC Score: 0.957380134184258
```







CONCLUSION

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis vel dolor ante. Nullam feugiat egestas elit et vehicula. Proin venenatis, orci nec cursus tristique, nulla risus mattis eros, id accumsan massa elit eu augue. Mauris massa ipsum, pharetra id nibh eget, sodales facilisis enim.