Student Dropout Rates or Academic Success Prediction

Presented by Team Ignite
Deon S
Bhavika Jangid

Problem Statement

Develop a machine learning model that can predict student dropout rates or academic success based on a variety of factors, such as attendance, grades, and demographic data.

Data Collection and Data Preprocessing

Data collection is the process of gathering relevant information or data from various sources. For our analysis, we chose our data from the UC Irvine Machine Learning Repository. Our data contained 37 features which includes student grades and demographic data.

The secondary data was found to be free of null values ,duplicates or typos, so much of data cleaning wasn't required. Data was in numerical form, there was no need to convert any kind of categorical data to numerical later on for model building.

	Marital status	Application mode	Application order	Course	Daytime/evening attendance\t	Previous qualification	Previous qualification (grade)	Nacionality	Mother's qualification	Father's qualification	...	Curricular units 2nd sem (credited)	Curricular units 2nd sem (enrolled)	(e
0	1	17	5	171	1	1	122.0	1	19	12	...	0	0	
1	1	15	1	9254	1	1	160.0	1	1	3	...	0	6	
2	1	1	5	9070	1	1	122.0	1	37	37	...	0	6	
3	1	17	2	9773	1	1	122.0	1	38	37	...	0	6	
4	2	39	1	8014	0	1	100.0	1	37	38	...	0	6	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
4419	1	1	6	9773	1	1	125.0	1	1	1	...	0	6	
4420	1	1	2	9773	1	1	120.0	105	1	1	...	0	6	
4421	1	1	1	9500	1	1	154.0	1	37	37	...	0	8	
4422	1	1	1	9147	1	1	180.0	1	37	37	...	0	5	
4423	1	10	1	9773	1	1	152.0	22	38	37	...	0	6	

Representation of our data with 37 features and 4424 instances.

Data Feilds

- Marital status
- Application mode
- Application order
- Course
- Daytime/evening attendance
- Previous qualification
- Previous qualification (grade)
- Nationality
- Mother's qualification
- Father's qualification
- Mother's occupation
- Father's occupation
- Admission grade
- Displaced
- Educational special needs
- Debtor
- Unemployment rate
- Inflation rate
- Tuition fees up to date
- Gender
- Scholarship holder
- Age at enrollment
- International
- 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)
- GDP

Libraries used

matplotlib

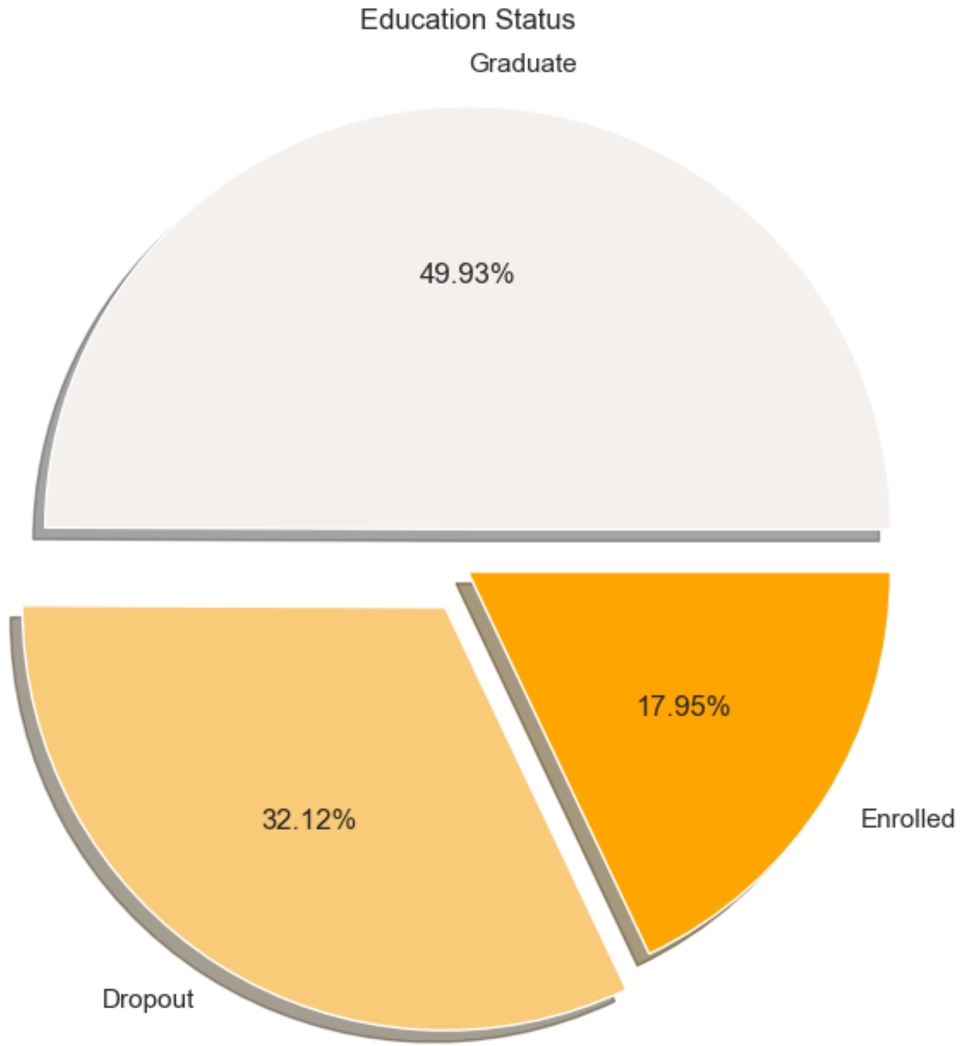
seaborn

pandas

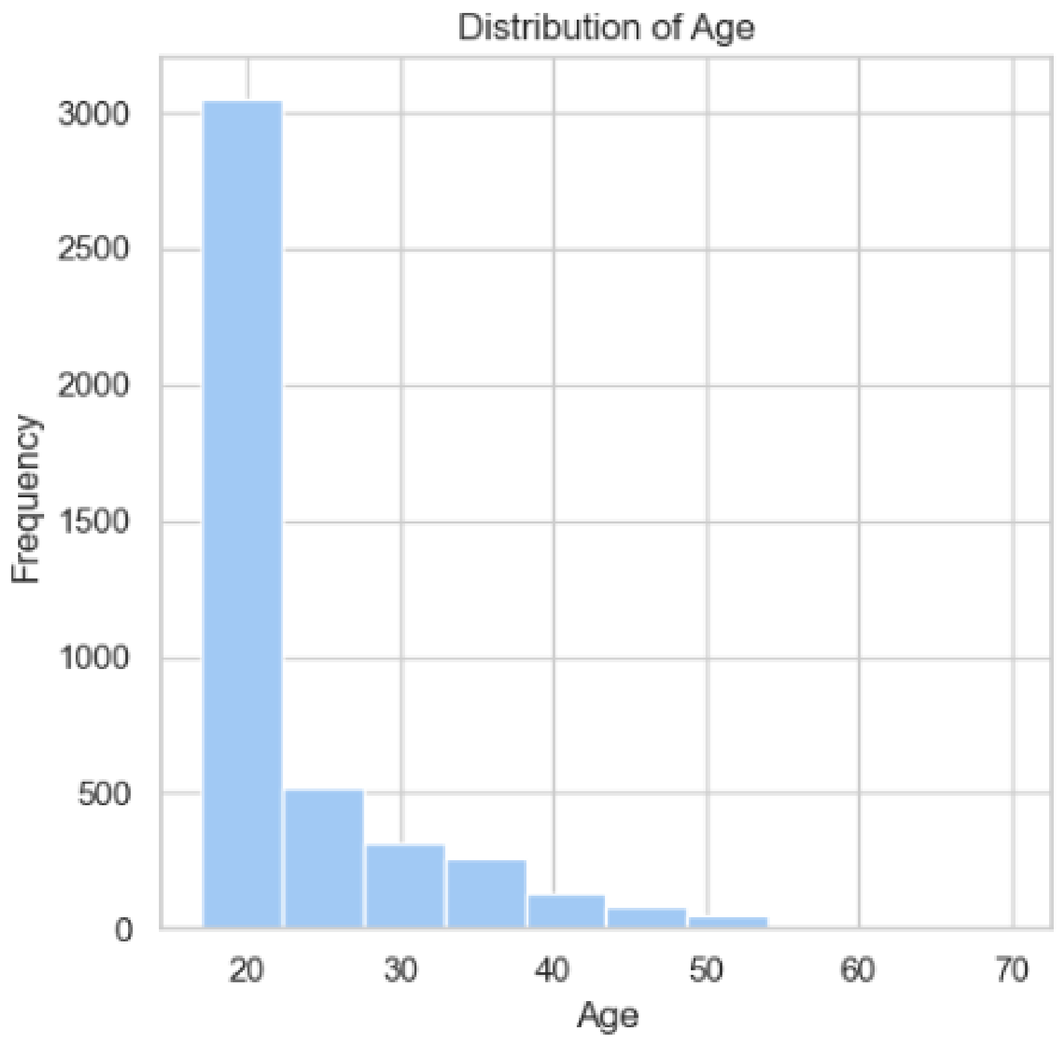
statsmodels

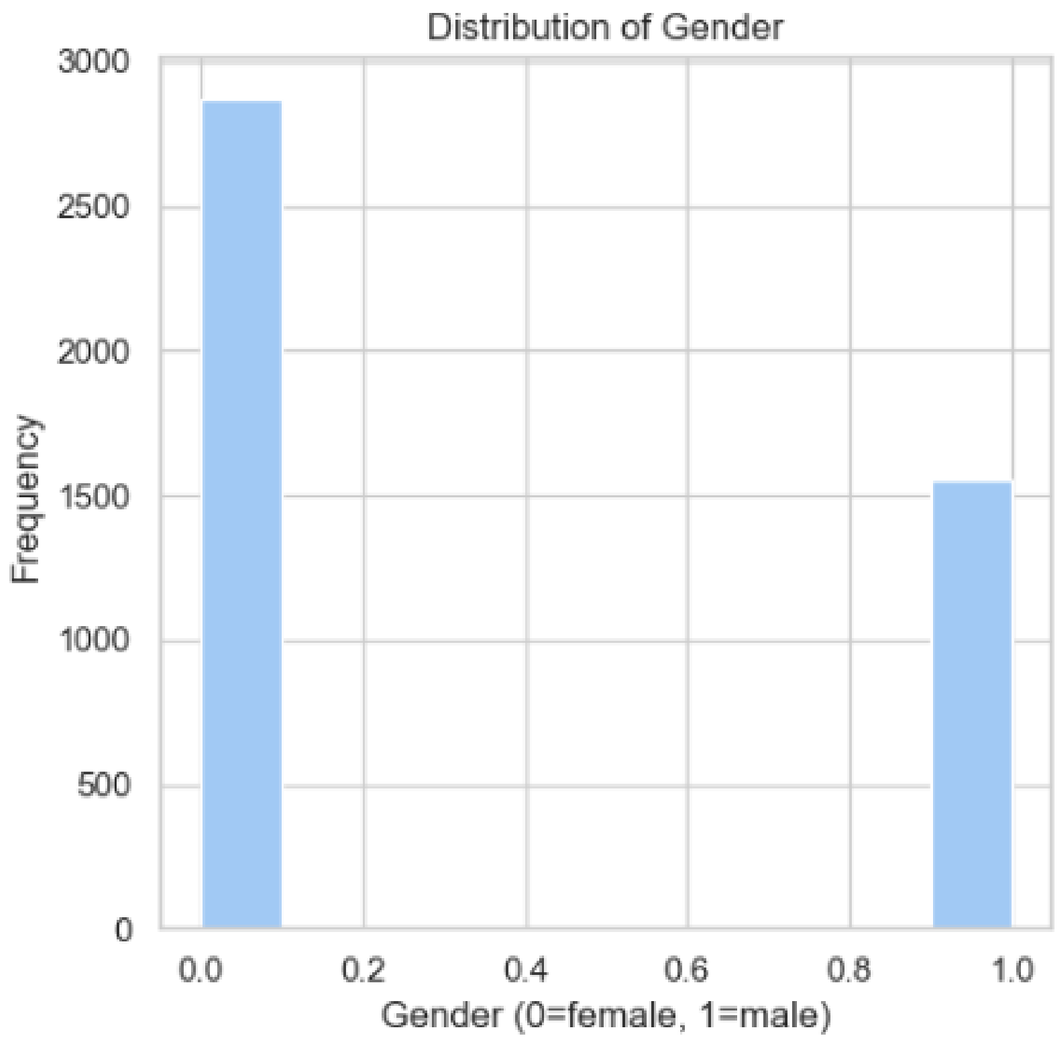
Exploratory Data Analysis

The Exploratory Data Analysis was done mainly using two libraries Matplotlib and Seaborn.

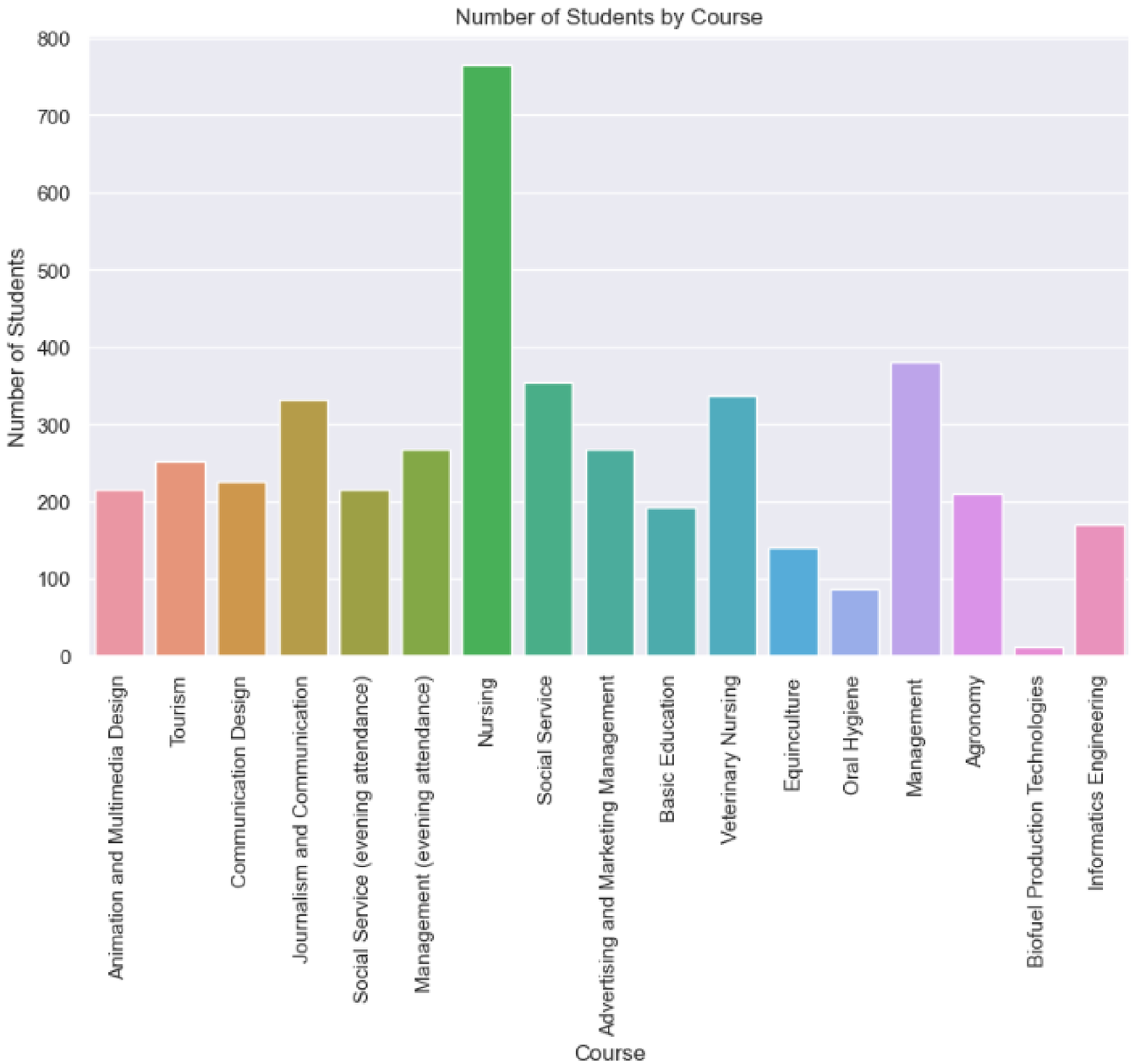


Distribution of Age and Gender

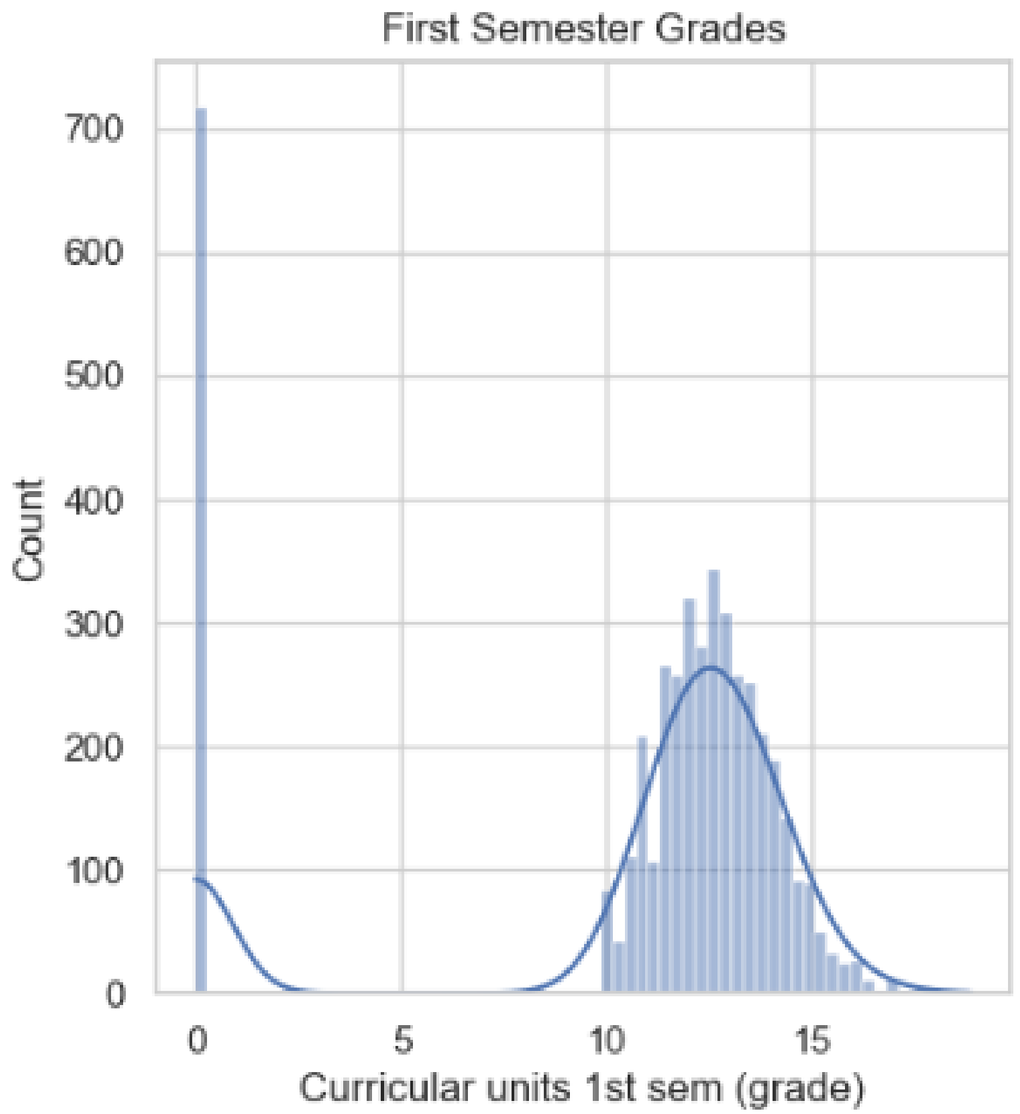


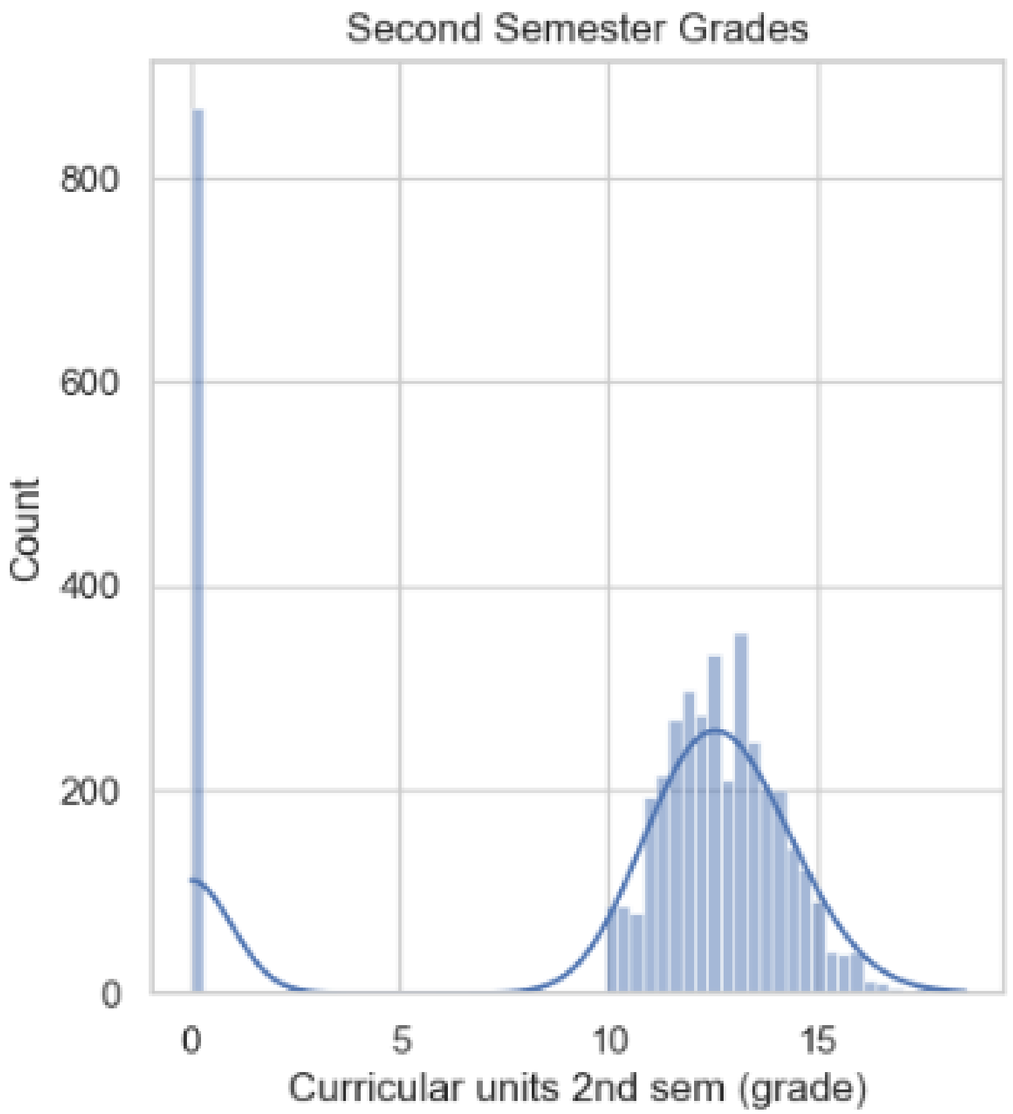


Students in each course

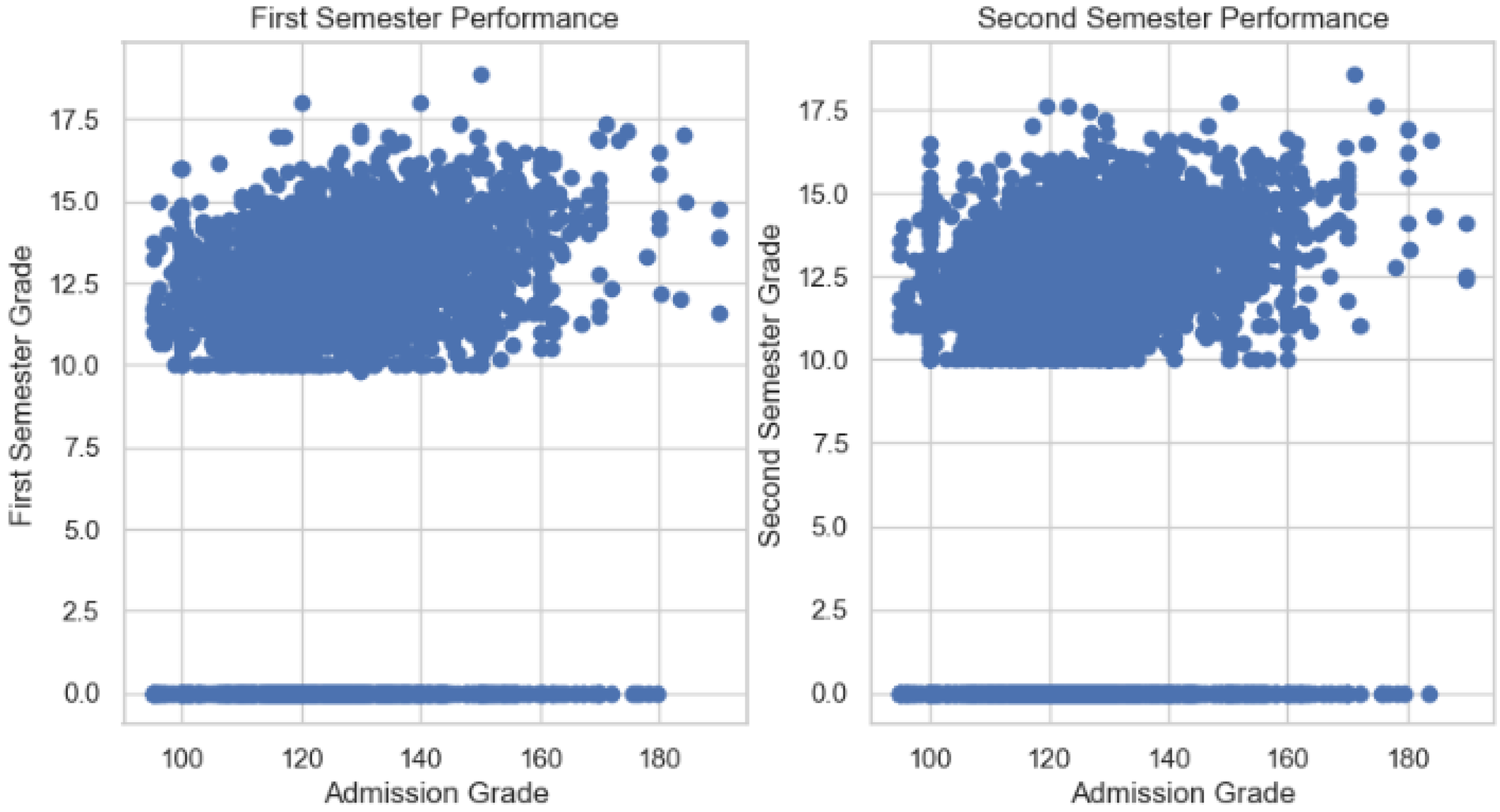


Grade comparison in 1st and 2nd Semesters

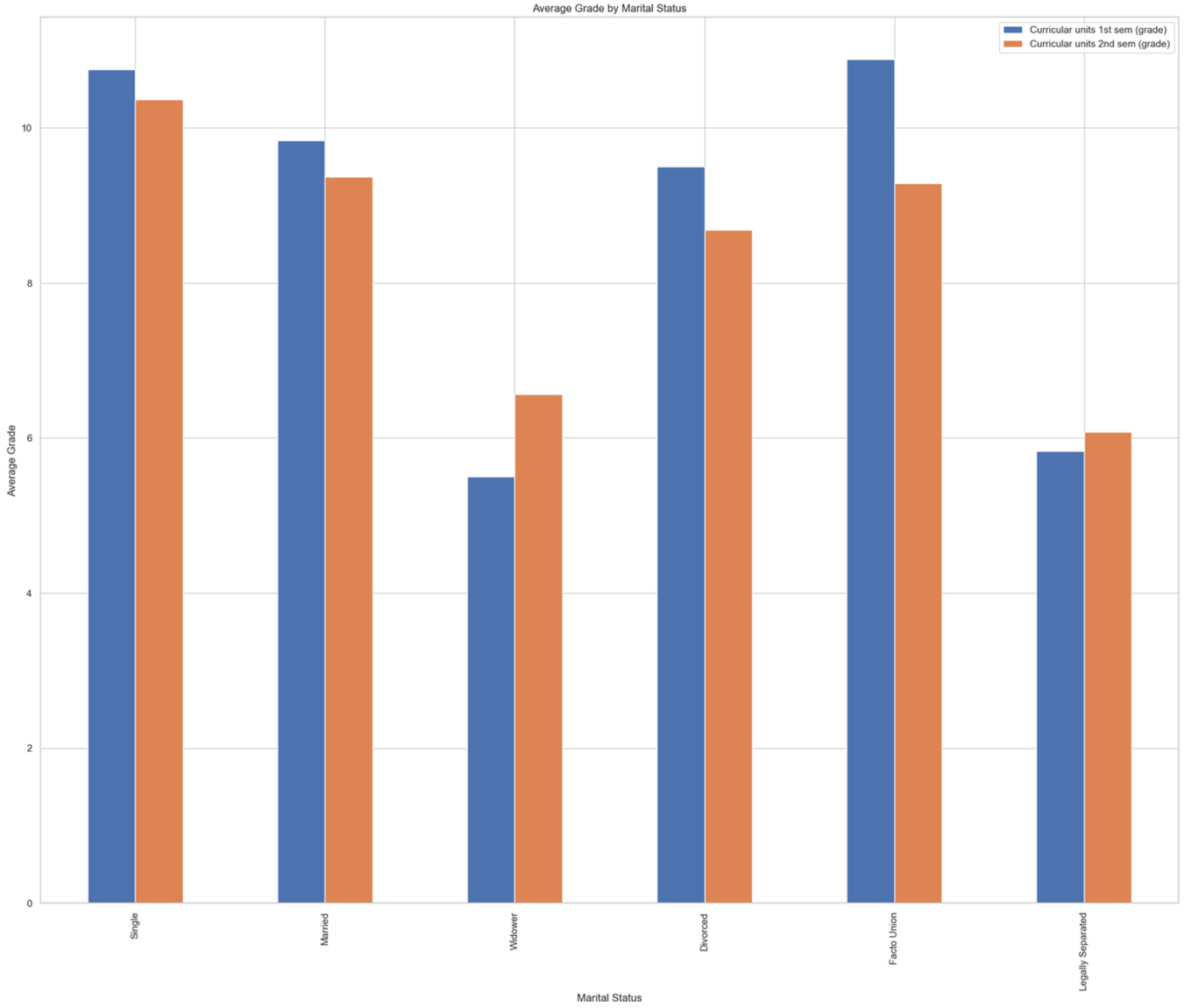




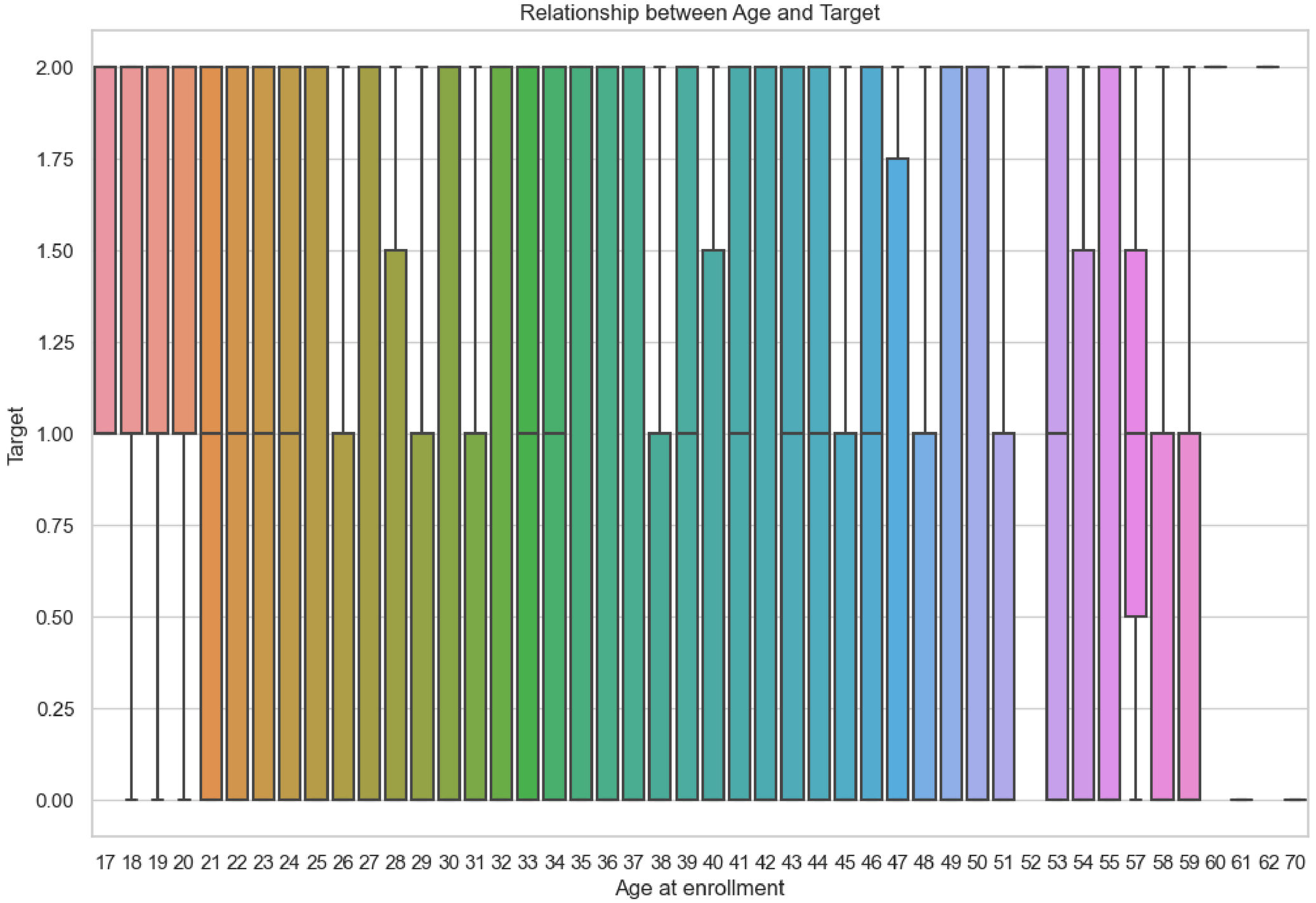
Scatter plot of Grade and Performance in both semesters

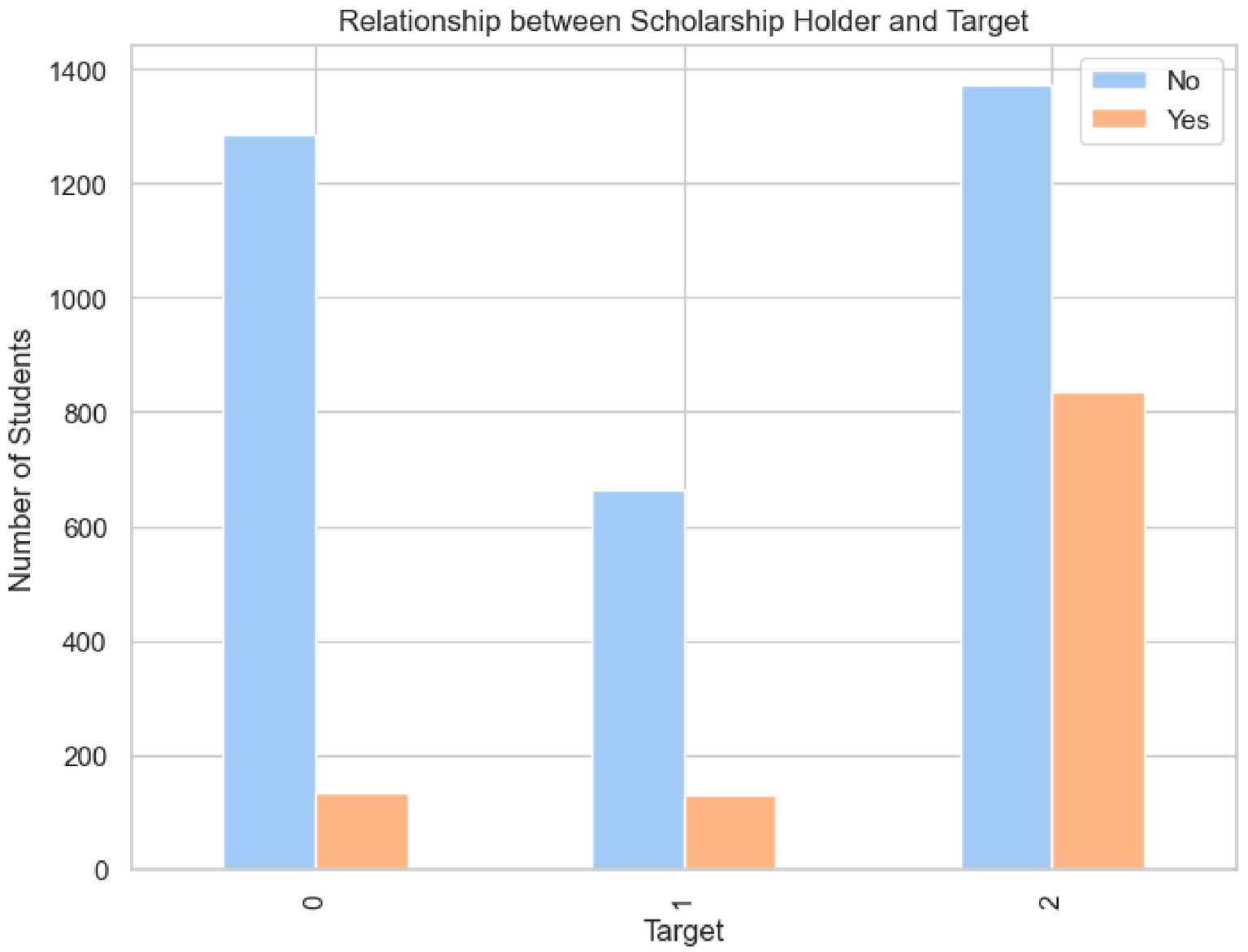


Average grades by Marital Status

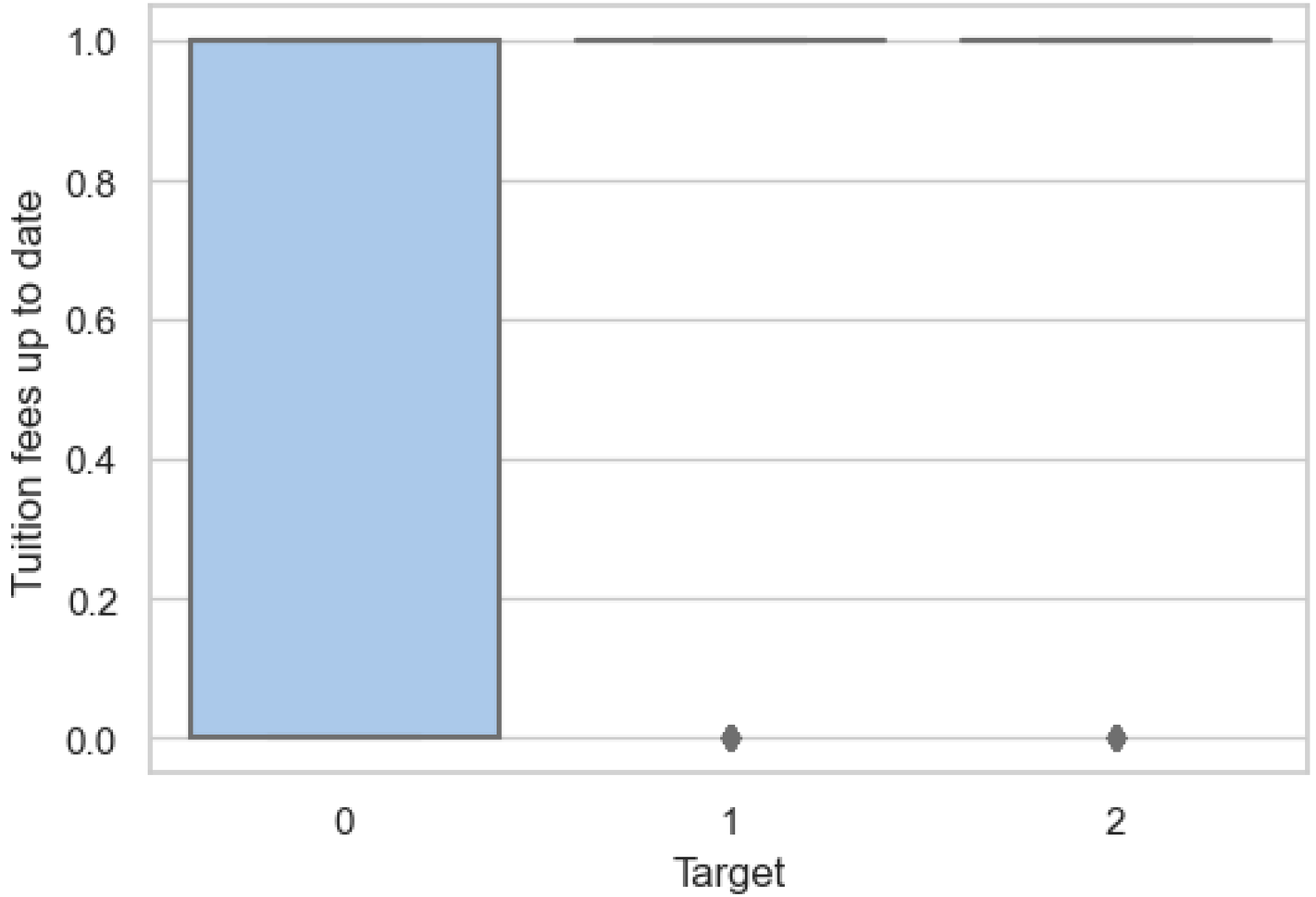


'Facto Union's' scored the highest average marks in 1st semester while 'Single' scored top average marks in semester 2. 'Widows' and 'Legally seperated' tends to score less.


Relationship between Age and Target
Target
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 57 58 59 60 61 62 70
Age at enrollment


Relationship between Scholarship Holder and Target
No
Yes
Number of Students
1400
1200
1000
800
600
400
200
0
0
1
2
Target

Tuition Fees



Students who have failed to pay the tuition fees on time have found out to be dropped.

Feature Selection

Feature selection, also known as variable selection or attribute selection, is the process of selecting a subset of relevant features (variables) from a larger set of available features. The goal of feature selection is to improve the performance of a machine learning model by reducing the dimensionality of the data, removing irrelevant or redundant features, and focusing on the most informative ones.'

In the feature selection phase, the team leveraged the Statsmodel library to identify statistically significant features. Then performed feature selection by setting a p-value threshold of $p < 0.005$.

Selected Features

Therefore the statistically significant features with respect to the target variable were

- Previous qualification
- Nationality
- Debtor
- Tuition fees up to date
- Gender
- Scholarship holder
- International
- Curricular units 1st sem (credited)
- Curricular units 1st sem (approved)
- 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)

Model Building

- Imported the necessary libraries and split the dataset into training and testing sets, with an 80:20 ratio.
- The training set contained 80% of the data and the remaining for testing. The model gave as an accuracy of 0.88705 which was good.
- With Random Forest, we obtained an accuracy of 0.9008 which was better than logistic regression.
- The model build using Gradient Boosting Classifier gave an accuracy of 0.89.
- By applying Hyperparameter tuning for Random Forest Classifier using Grid Search, we improved the score to 0.904958.

Evaluation Metrices

	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.89	0 0.93 2 0.88	0 0.80 2 0.96	0 0.86 2 0.92
Random Forest Classifier	0.90	0 0.91 2 0.89	0 0.81 2 0.95	0 0.86 2 0.92
Gradient Boosting	0.89	0 0.93 2 0.87	0 0.78 2 0.96	0 0.85 2 0.91

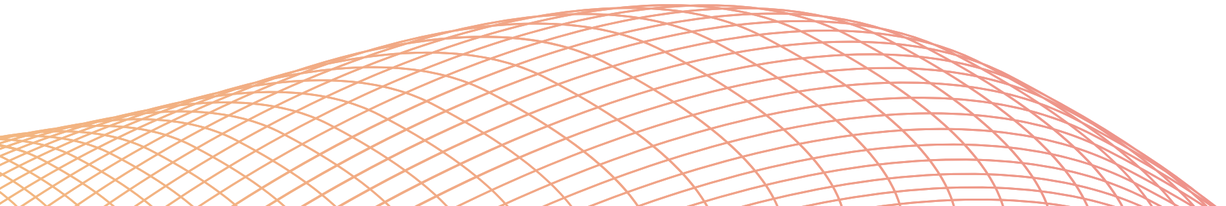
0-Dropout / 2-Graduated

Hyperparameter Tuning

It is the process of selecting the optimal values for the hyperparameters of a machine learning model.

For Random Forest classifier, some common hyperparameters include the number of trees (n_estimators), maximum depth of the trees (max_depth), minimum number of samples required to split an internal node (min_samples_split), and the number of features to consider when looking for the best split (max_features).

After hyperparameter tuning using GridSearchCV we got the following results,
Best Parameters: {'max_depth': None, 'min_samples_split': 10, 'n_estimators': 300}
Best Model Accuracy: 0.9049586776859504



Model Deployment

A sample data was taken from the testing data and was tested to know if our model predicts well.

```
# Load the new data into a pandas DataFrame
new_data = pd.read_csv(r"C:\Users\LENOVO\Dropbox\PC\Downloads\test sample.csv")

# Preprocess the new data (if required) to match the format of the training data

# Load the trained model
gbc = GradientBoostingClassifier()

# Assuming you have trained the model on X_train and y_train
# Fit the model on the training data
gbc.fit(xgb_train,ygb_train)

# Make predictions on the new data
predictions = gbc.predict(new_data)

# View the predicted labels
print(predictions)
```

```
[2 2 0 2 2 0 2 0 2 0 2 2 0 2 2 0 2 2 2]
```

Thank You

Team Ignite