CS6375 Project - Group 17
Student Academic Performance Prediction

Likhitha Emmadi (lxe220001), Vatsalya Gorthi(vxg210008)

Motivation

The data set chosen for the ML analysis is to predict student's dropout and academic success. We choose this data set to understand the various reasons that affect the academic performance of students and result in their dropout. Academic institutions can take preventive steps to help students continue their studies if they know the factors that contribute mostly to the dropout.

The data set used in the project is taken from Kaggle. This data set provides a comprehensive view of students enrolled in various undergraduate degrees offered at a higher education institution. It has 35 features which includes demographic data, social-economic factors and academic performance information [Rea+21].

Data set Description

The data set has three classes namely Graduated, Dropout and Enrolled. There are no missing or null values in the dataset. But the data points with the enrolled class are very less compared to the other two classes. So, to reduce the class imbalance the data points with the class 'Enrolled' has been removed from the data set. And the problem is made a binary classification problem with the classes Graduated and Dropout.

The size of the initial data set is 4424 rows x 35 columns and the size of the modified data set after removing the data points with the "Enrolled" class is 3630 rows x 35 columns. The Target variables (graduate/dropout) are given as strings in the data set, that has been updated to integers (1/0).

To understand the relationship between the features, we calculated the correlation between the features and plotted a heat map as shown in the Fig 2. Selected the top 20 features that have high correlation with the target variable to train the model as shown in the Fig 1 and Fig 3

Exploratory Data Analysis

The Count plot of the Target variable shown in Fig 4 shows that there is a class imbalance in the data set, with more data in the positive class. And as shown in the Fig 5 the scatter plot of the top 4 highly correlated features with the target variable shows that the data is linearly separable.

Fig 6 to Fig 9 shows the histograms for the selected numerical Features. Most of the features in the data set like

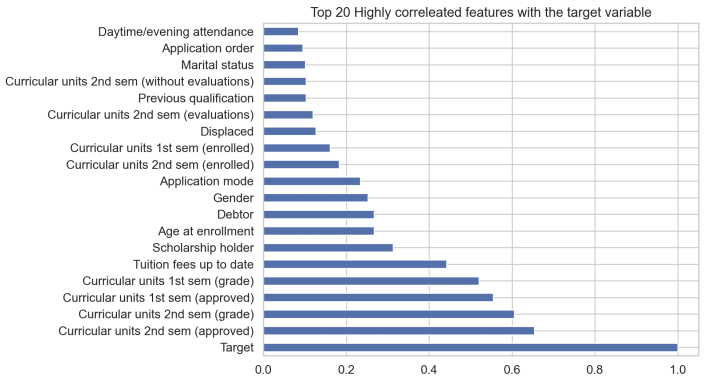


Figure 1: Top 20 highly correlated features with the target

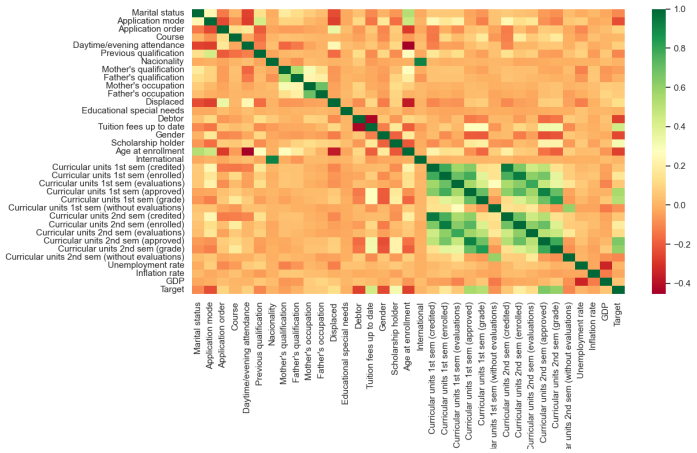


Figure 2: Heat map

```
Important features selected based on the correlation with the Target Variable
['Curricular units 2nd sem (approved)', 'Curricular units 2nd sem (grade)', 'Cur
ricular units 1st sem (approved)', 'Curricular units 1st sem (grade)', 'Tuition
fees up to date', 'Scholarship holder', 'Age at enrollment', 'Debtor', 'Gender',
 'Application mode', 'Curricular units 2nd sem (enrolled)', 'Curricular units 1s
t sem (enrolled)', 'Displaced', 'Curricular units 2nd sem (evaluations)', 'Previ
ous qualification', 'Curricular units 2nd sem (without evaluations)', 'Marital s
tatus', 'Application order', 'Daytime/evening attendance', 'Curricular units 1st
 sem (without evaluations)']
```

Figure 3: The important features selected based on the correlation

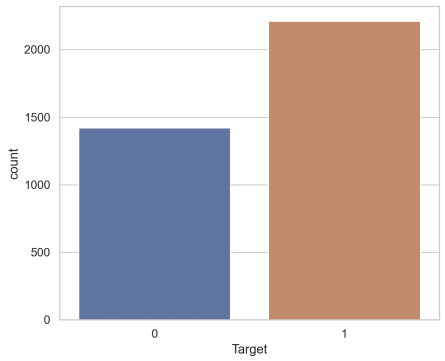


Figure 4: Count of the points in different classes

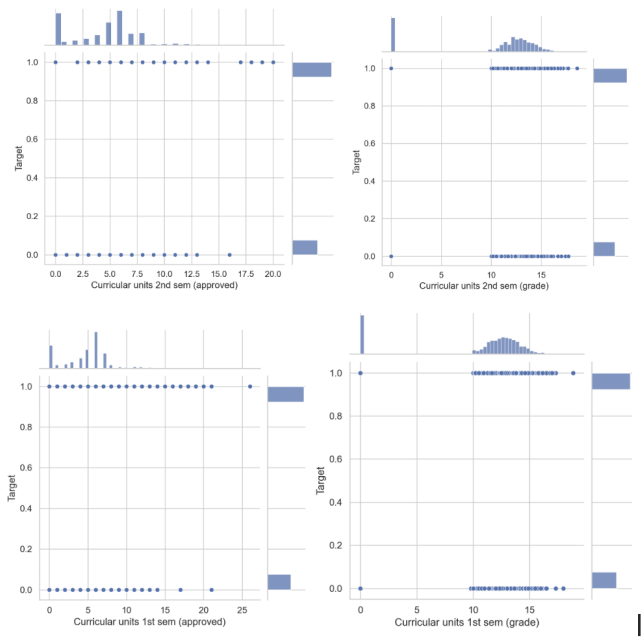


Figure 5: Scatter plot of the features with the target variable

Curricular units 2nd sem (approved), Curricular units 1st sem (approved), Age at enrollment, Application mode etc., are right skewed. From Fig 7 and Fig 11 we can infer that most the data is collected from the people of the age 17 and the application mode 1 which is 1st phase—general contingent. And enrolled curricular units for most of the them is between 5 and 8. Also From Fig 8 and Fig 9 we can infer that most of the people in the data set are single and have completed secondary education.

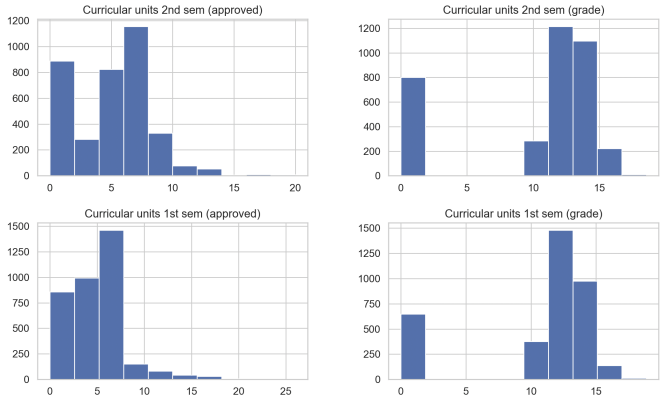


Figure 6: Histogram for the features

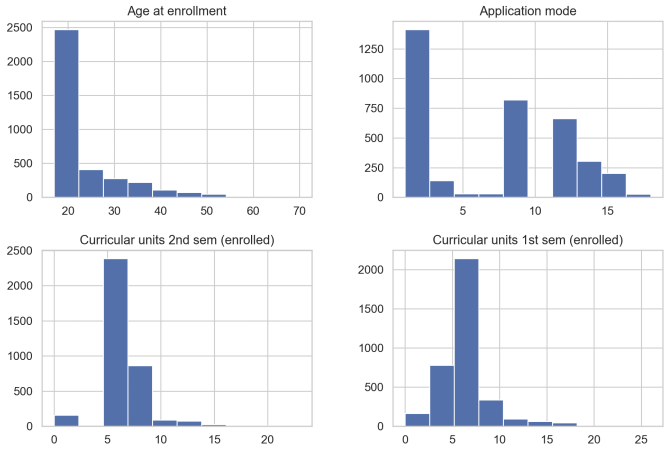


Figure 7: Histogram for the features

The Box plots in Fig 10 shows that most of the students who were approved less than 5 curricular units and whose grade was less than 12.5 in both the 1st and 2nd sem dropped out.

Model Selection

The data set is split into train, validation and test sets in the ratio of 80,10,10 respectively. The model has been trained on the training data and the hyper parameters have been tuned by testing on the validation data to avoid overfitting. The following models have been implemented as part of the project.

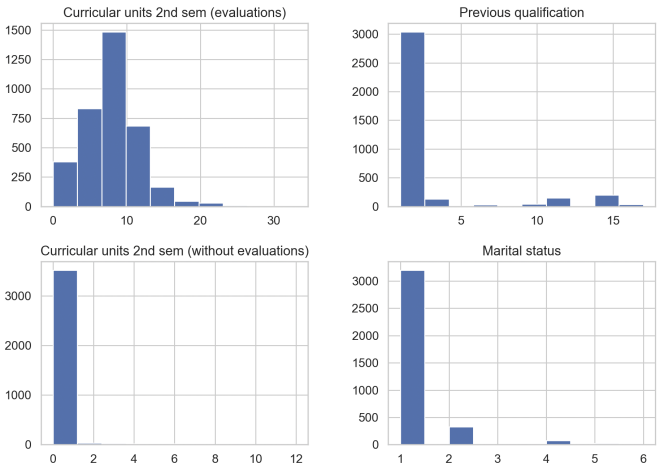


Figure 8: Histogram for the features

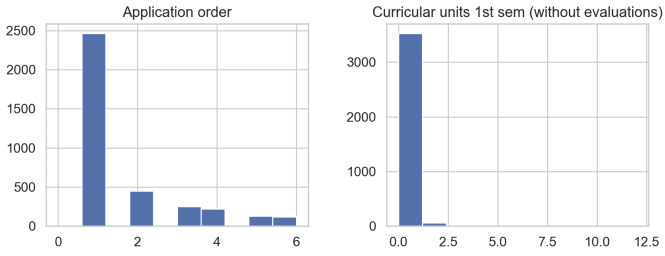


Figure 9: Histogram for the features

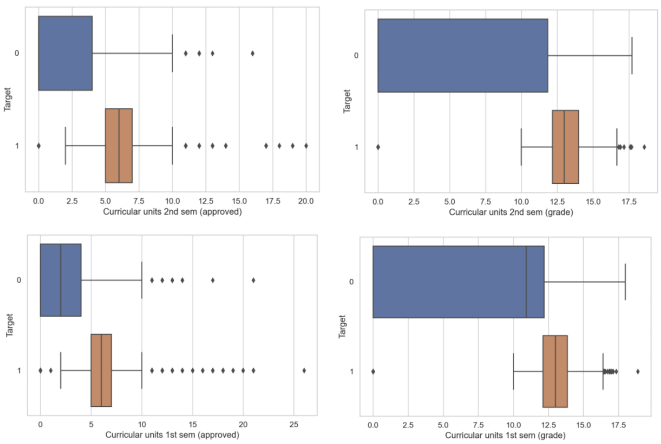


Figure 10: Box plot of the features

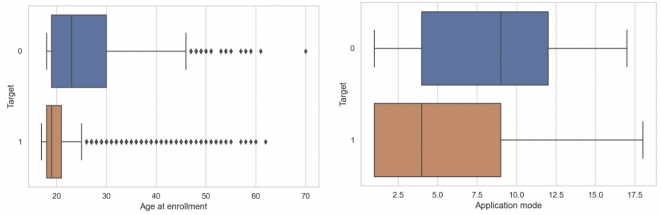


Figure 11: Box plot of the features

Logistic Regression

We started out with the basic model, logistic regression to implement binary classification. As shown in Fig 12 training the model with 0.01 learning rate and 10000 iterations gives the best results.

```
 Logistic Regression
Results for Validation Error for different number of iterations and learning rates
Iterations  Learning rate  Validation error
10          0.01           0.2259
10          0.1            0.21763
10          0.33           0.22314
100         0.01           0.14876
100         0.1            0.2011
100         0.33           0.29201
1000        0.01           0.12121
1000        0.1            0.12397
1000        0.33           0.23691
10000       0.01           0.09091
10000       0.1            0.11019
10000       0.33           0.19559

The best parameters for logistic regression that gives minimum error on the validation data is
Iterations -  10000    Learning rate -  0.01    Validation error -  0.19559228650137742
```

Figure 12: Hyperparameter selection for Logistic regression

Decision Tree

As the data set contains both numerical and categorical data, decision trees are used, as they work well with any type of data. As shown in Fig 13 decision tree with depth 7 gives the best results.

```
 Decision Tree
Results for Validation Error for different depths
Depth    Validation error
1           0.2011
2           0.14876
3           0.12397
4           0.1157
5           0.11846
6           0.11846
7           0.11295
8           0.12397
9           0.12948

The best parameters for decision tree that gives minimum error on the validation data is
Depth -  7    Validation error -  0.12947658402203857
```

Figure 13: Hyperparameter selection for the decision tree

Naive Bayes

As Naive Bayes works well most of the time, we implemented it. Multinomial naive Bayes works well on categorical data and they can also handle numerical data. Fig 14 shows the top 3 features for the dropout and the graduated classes.

K Nearest neighbors

AS KNN depends on the similarity of the data points and our data set is medium sized, we implemented that algorithm. As shown in Fig 15 KNN with k=8 gives the best results.

```
Top three words that have the highest class-conditional likelihoods for both the
"Dropout" and "Graduated" classes for our naive bayes model

{0.0: [('Curricular units 2nd sem (enrolled)', -1.1216869815570316), ('Curricular
 units 2nd sem (grade)', -2.259308447247031), ('Curricular units 2nd sem (evaluat
ions)', -2.3887313215457504)], 1.0: [('Curricular units 2nd sem (enrolled)', -1.4
76355993163042 3), ('Daytime/evening attendance', -2.016566175386843), ('Curricula
r units 2nd sem (evaluations)', -2.0196639027257337)]}
```

Figure 14: Top features for both the classes in the Naive Bayes implementation

```
 KNN
Results for Validation Error for different values of K
K        Validation error
1              0.46832
2              0.46832
3              0.47107
4              0.46006
5              0.46006
6              0.45455
7              0.46006
8              0.45179
9              0.46281

The best parameters for KNN that gives minimum error on the validation data is
K -  8    Validation error -  0.4628099173553719
```

Figure 15: Hyperparameter selection for KNN

Results

The following confusion matrix shown in Fig 16 structure is followed to visualize the results.

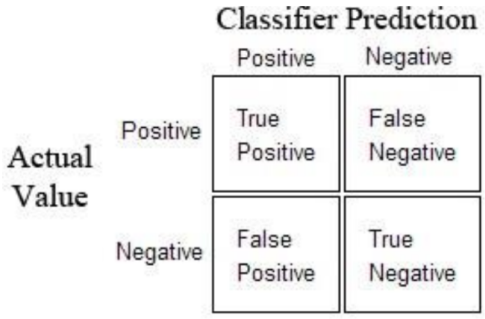


Figure 16: Confusion Matrix

Tables 1 and 2 show the performance metrics of the various algorithms that are implemented.

Out of the 4 implemented models, logistic regression performs better. And Neural networks and Stochastic gradient boosting gives good accuracy out of all the scikit learn models.

ROC and Precision-Recall curve

ROC and Precision-Recall curve for the own models implemented as shown in the Fig 17

ROC and Precision-Recall curve for the scikit models implemented as shown in the Fig 18

Conclusion

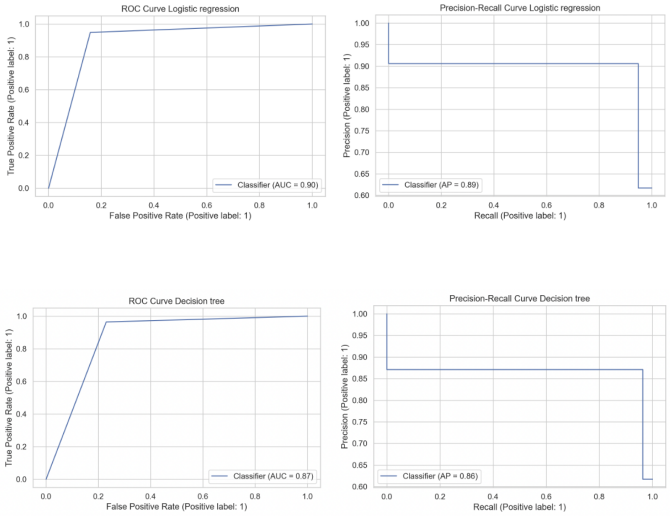
- The provided Kaggle data set contains only one semester-worth information on each admission, and this is not sufficient for the accurate prediction of the academic performance. By including more semesters worth of data the accuracy of the model can be improved.

Model	Accuracy	Training Error	Precision	Recall	F1 score	Confusion Matrix
Logistic regression	90.77%	8.74%	90.62%	94.87%	92.69%	425 23 44 234
Decision tree	88.98%	8.85%	87.1%	96.43%	91.53%	432 16 64 214
Multinomial Naive Bayes	81.68%	19.21%	80.0%	93.75%	86.33%	420 28 105 173
K Nearest Neighbors	85.95%	9.6%	84.33%	94.87%	89.28%	425 23 79 199

Table 1: Results of own implementation models

Model	Accuracy	Training Error	Precision	Recall	F1 score	Confusion Matrix
Logistic regression	90.77%	8.78%	90.97%	94.42%	92.66%	423 25 42 236
Decision tree	86.09%	0.31%	88.99%	88.39%	88.69%	396 52 49 229
Multinomial Naive Bayes	82.92%	18.58%	84.18%	89.06%	86.55%	399 49 75 203
Gaussian Naive Bayes	84.71%	15.94%	85.03%	91.3%	88.05%	409 39 72 206
Bernoulli Naive Bayes	83.61%	15.94%	81.21%	95.54%	87.8%	428 20 99 179
K Nearest Neighbors	82.23%	19.83%	81.71%	91.74%	86.44%	411 37 92 186
SVM SVC	90.5%	10.27%	88.13%	97.77%	92.7%	438 10 59 219
SVM LinearSVM	90.22%	9.52%	87.48%	98.21%	92.53%	440 8 63 215
Stochastic Gradient Boosting	91.46%	6.81%	91.06%	95.54%	93.25%	428 20 42 236
Neural network	91.46%	9.96%	92.14%	94.2%	93.16%	422 26 36 242

Table 2: Results of scikit-learn models



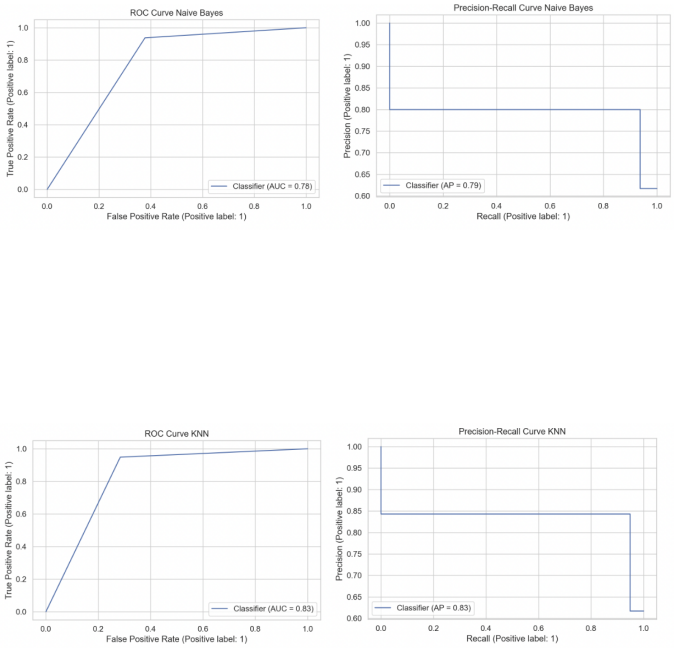
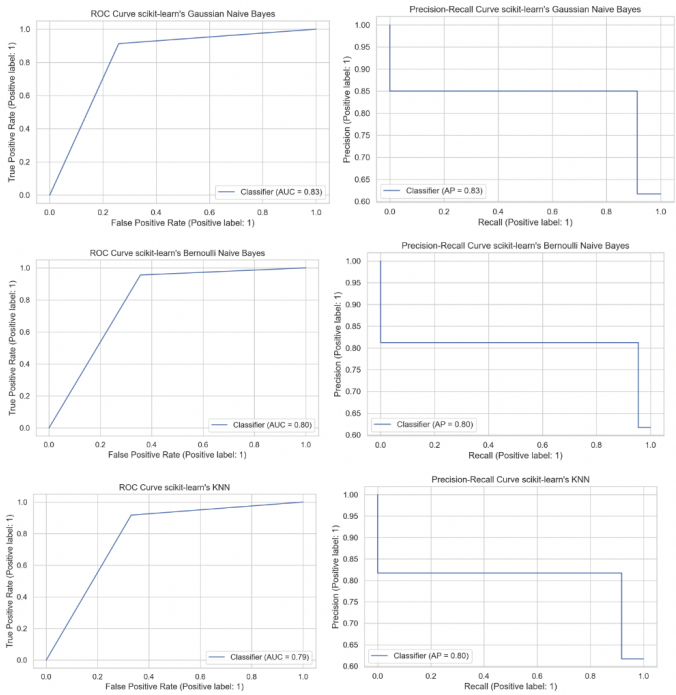
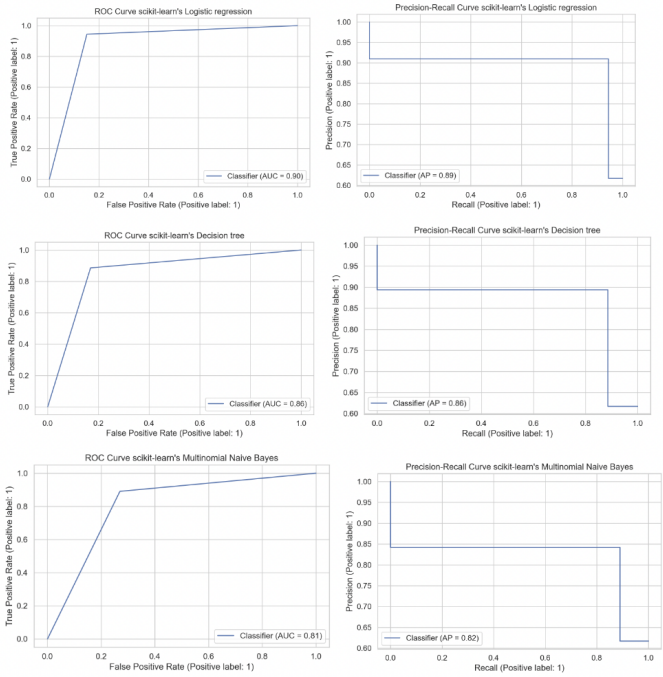
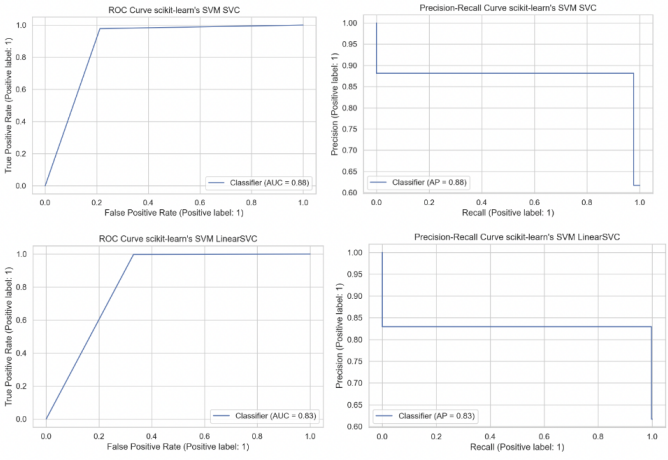


Figure 17: Roc and Precision-Recall Curve







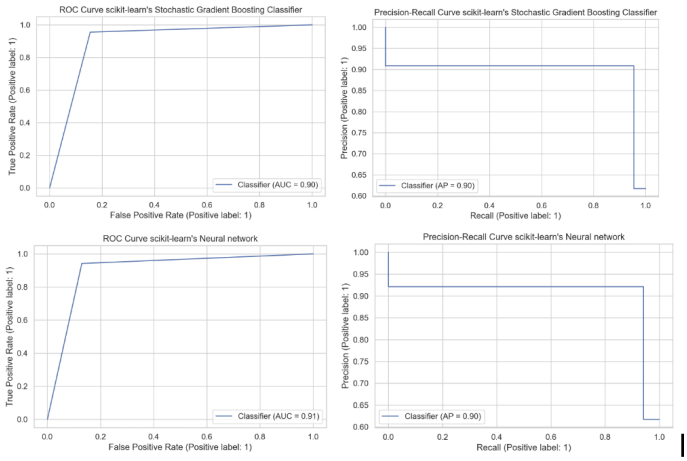


Figure 18: Roc and Precision-Recall Curve

- As the data is linearly separable and there is correlation between the features (Fig 5 and Fig 2) Logistic regression performs better.
- And for the same reason as above Linear SVM also gives good accuracy.
- As the heat map in Fig 2 shows correlation between the selected features, few of them are not independent. The feature independence assumption of the naive Bayes is not followed, that might be one of the reasons for the less accuracy of naive Bayes.
- By selecting the features that are approximately independent of each other and have a correlation with the target can be selected to improve the accuracy of Naive Bayes.
- And the lower accuracy in KNN might be because of using large number of features (20), the accuracy of KNN increased to 88% when trained the model with 10 features instead of 20 features. But the accuracy for the other models are decreasing if we reduce the number of features. And KNN is also affected by the outliers.
- And the imbalance in the class (Fig 4) might also be the reason for the low accuracy of all the models.
- And from the confusion matrices of all the models we can infer that the accuracy of the model is affected by the high false positive rate i.e, we are wrongly predicting the students who dropped out as graduated this might be because of having more data points in the positive class.
- And similar observations are made in the scikit models. Logistic regression, Linear SVM, Stochastic gradient boosting and neural networks give better accuracy and the accuracy of the other models is reduced due to high false positives.
- As Stochastic gradient boosting and neural networks are complex models they give good results, but it is preferable to use Linear models like Logistice regression or Linear SVM as the data is linearly separable.

Libraries Installed

- sklearn
- seaborn
- numpy
- pandas
- matplotlib
- graphviz

References

[Rea+21] Valentim Realinho et al. *Predict students' dropout and academic success*. Version 1.0. Zenodo, Dec. 2021. DOI: 10.5281/zenodo.5777340. URL: https://doi.org/10.5281/zenodo.5777340.