IART- PROJECT 2

Topic 1: Student's dropout and academic success

Supervised Learning

Sofia Germer up201907461

Sérgio Estêvão up201905680

Pedro Silva up201907523

Approach

This is a **Supervised Learning** Problem, so the main point of this project is to learn how to classify examples in terms of the concept under analysis.
In this project, we will analyze a dataset containing information known at the time of student enrollment (academic path, demographics, and social-economic factors) and the students' academic performance at the end of the first and second semesters. The data is used to build classification models to predict students' dropout and academic success.

Our approach to this problem:

1- Data Analysis

Explore the raw dataset to identify missing or wrong information and to figure out how to use it and which problems it may have

2- Algorithm Application

Apply classification algorithms to obtain experimental results. We used Decision Trees, Neural Networks, KNN, SVM.

3- Evaluation and refinement

Evaluate and refine the results

Tools and libraries

Python

Jupyter Notebook

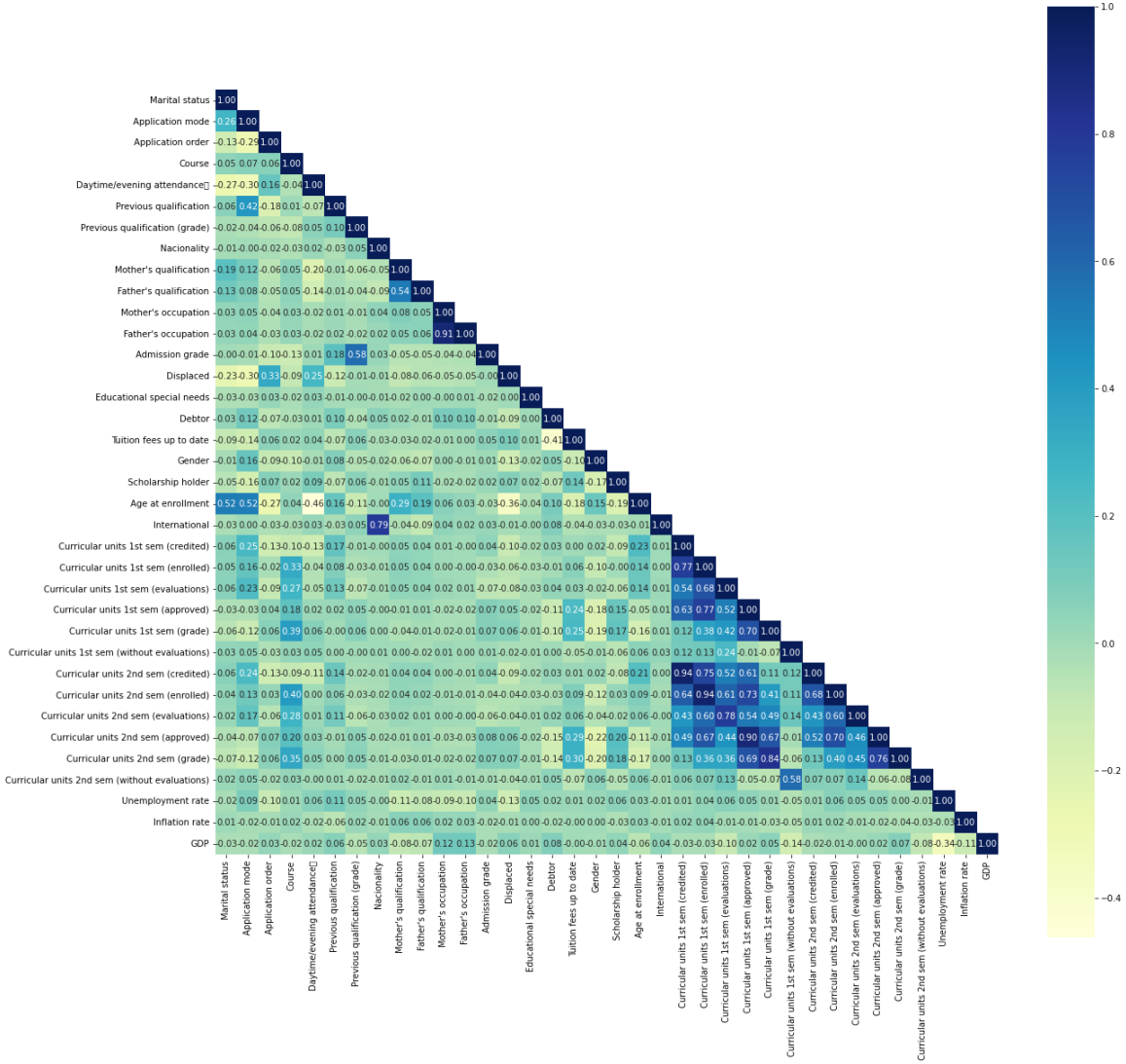
Pandas

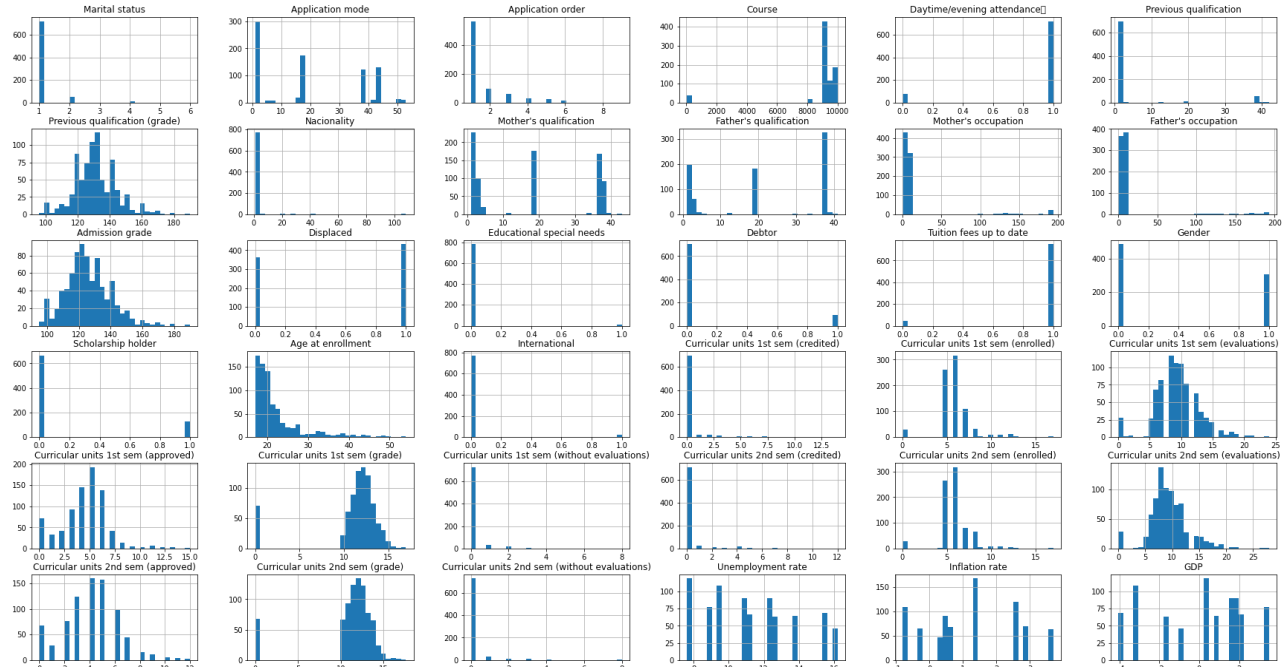
MatPlotLib

Seaborn

Sklearn

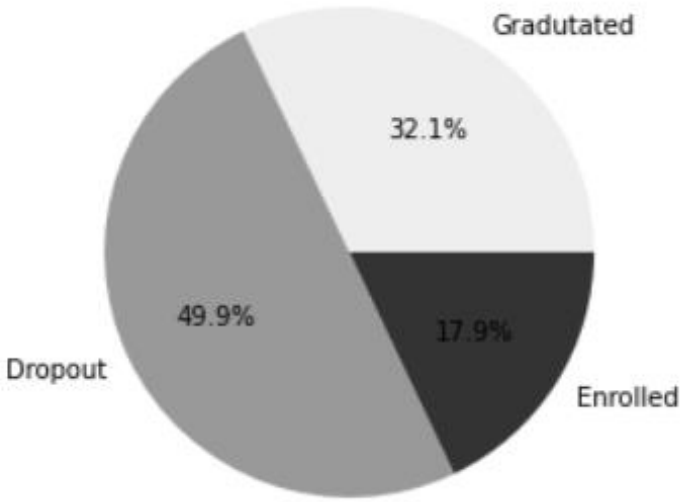
Data Preprocessing





Resampling

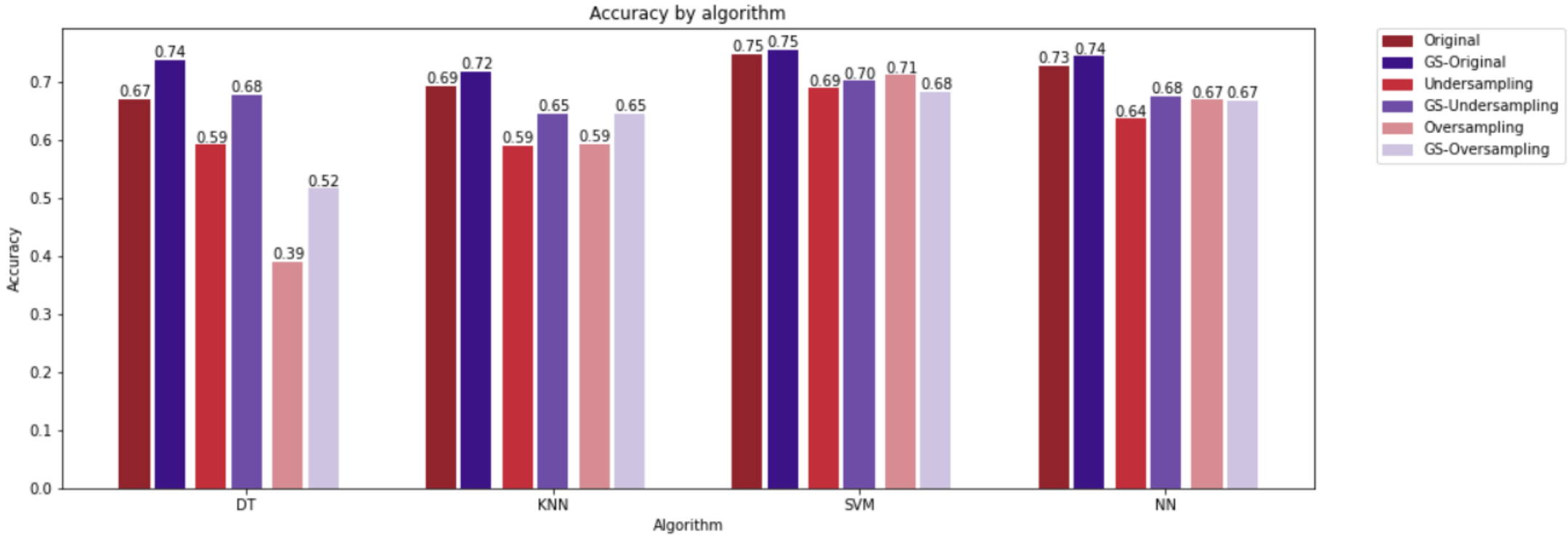
What we first observed was that the class distribution was very unbalanced: 49.9% graduated, 32.1% were dropout and only 17.9% were enrolled. Oversampling and undersampling were applied to the training data to see how they would fair.

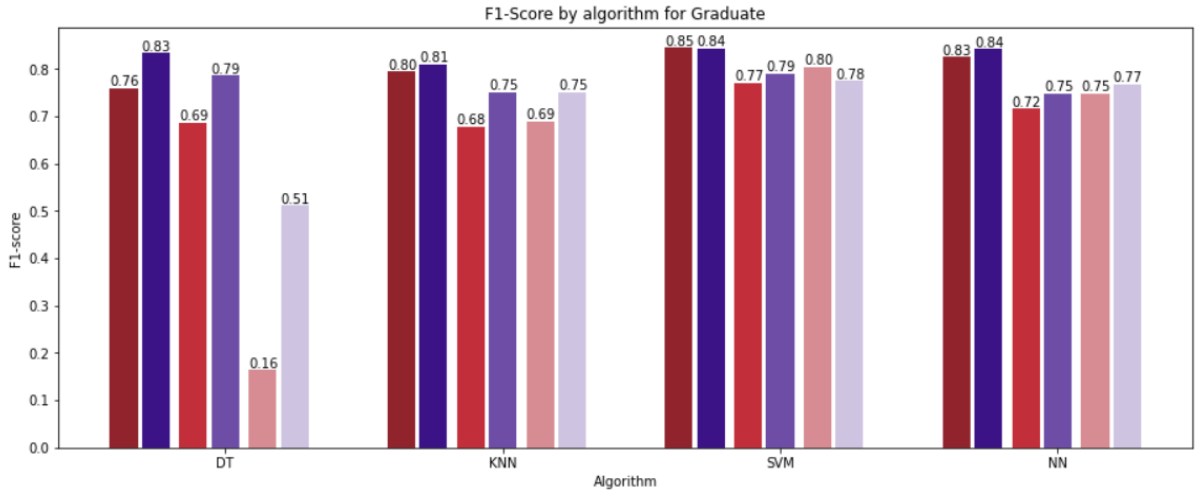


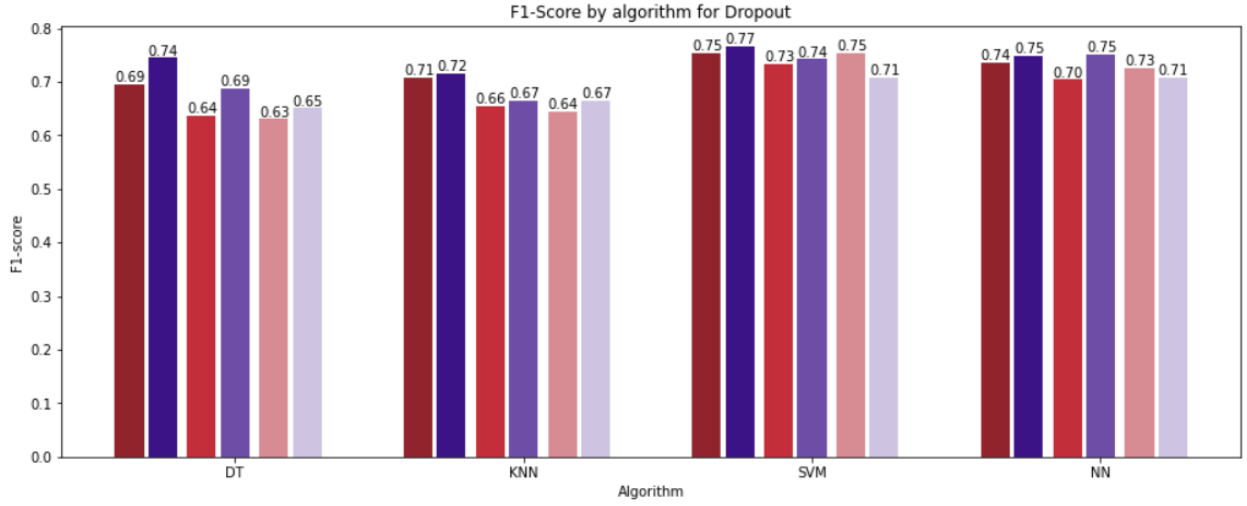

```
from imblearn.under_sampling import RandomUnderSampler

rus = RandomUnderSampler()

us_inputs, us_labels = rus.fit_resample(train_in, train_classes)

print(Counter(us_labels))
```

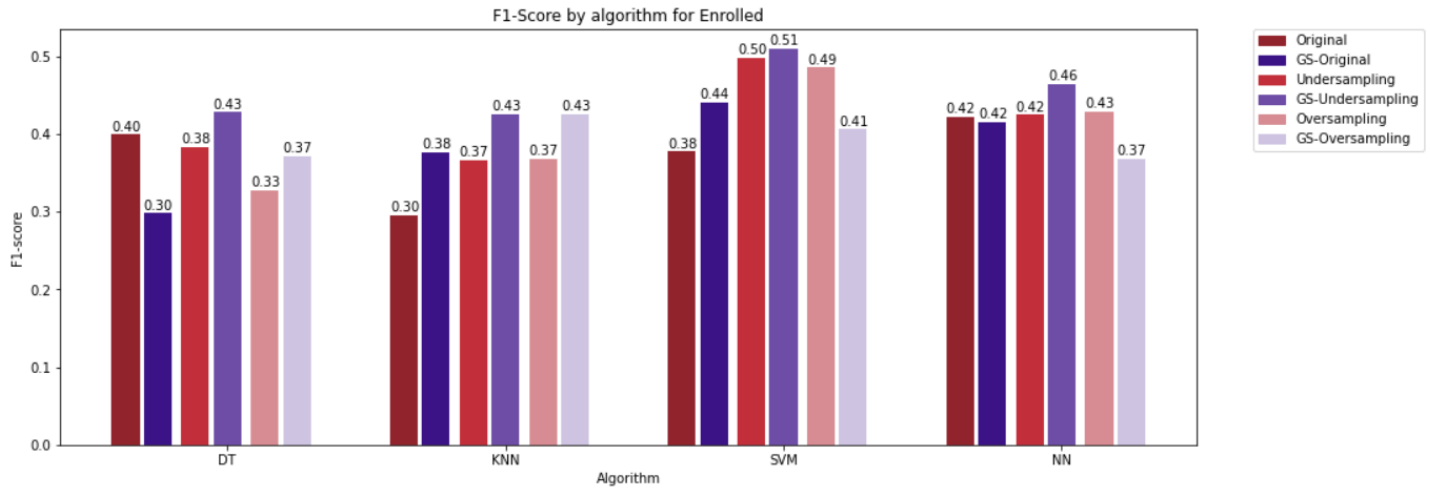
```
from imblearn.over_sampling import SMOTE

ros = SMOTE()

os_inputs, os_labels = ros.fit_resample(train_in, train_classes)

print(Counter(os_labels))
```

Result Comparison




F1-Score by algorithm for Graduate
F1-score
Algorithm
DT
KNN
SVM
NN
0.76 0.83 0.69 0.79 0.16 0.51
0.80 0.81 0.68 0.75 0.69 0.75
0.85 0.84 0.77 0.79 0.80 0.78
0.83 0.84 0.72 0.75 0.75 0.77


F1-Score by algorithm for Dropout
F1-score
Algorithm
DT
KNN
SVM
NN
0.69 0.74 0.64 0.69 0.63 0.65
0.71 0.72 0.66 0.67 0.64 0.67
0.75 0.77 0.73 0.74 0.75 0.71
0.74 0.75 0.70 0.75 0.73 0.71


F1-Score by algorithm for Enrolled
F1-score
Algorithm
DT
KNN
SVM
NN
0.40 0.30 0.38 0.43 0.33 0.37
0.30 0.38 0.37 0.43 0.37 0.43
0.38 0.44 0.50 0.51 0.49 0.41
0.42 0.42 0.42 0.46 0.43 0.37
Original
GS-Original
Undersampling
GS-Undersampling
Oversampling
GS-Oversampling

Conclusions

- We were able to reach **75%** accuracy;
- The best results were obtained using the SVN classification with the original dataset:
  - 75% accuracy;
  - overall best F1-scores.
- Results were limited by the low number of rows, compared to columns, of the dataset;
- With this project, we were able to learn about different supervised learning models used in Supervised Learning.

Bibliography

- RAM000574, Dataset "Students' dropout and academic success" used, URL: https://www.kaggle.com/datasets/tulasiram574/students-dropout-and-academic-success ;
- NumPy Developers, Numpy documentation, URL: https://numpy.org/doc/stable/user/index.html#user ;
- pandas development team, pandas documentation, URL: https://pandas.pydata.org/docs/user_guide/index.html#user-guide ;
- Matplotlib Development team, Matplotlib documentation, URL: https://matplotlib.org/stable/index.html ;
- scikit-learn developers, scikit-learn documentation, URL: https://scikit-learn.org/0.18/documentation.html ;
- Michael Waskom, seaborn tutorial, URL: https://seaborn.pydata.org/tutorial.html ;
- imbalanced-learn developers, imbalanced-learn documentation, URL: https://imbalanced-learn.org/stable/user_guide.html ;
- George Liu, Optimizing Neural Networks — Where to Start?, URL: https://towardsdatascience.com/optimizing-neural-networks-where-to-start-5a2ed38c8345 .