

**Machine Learning I**

**Report**

**Wizardry School Enrollment**

Enrollment predictions

**Group 05**

**1 December 2023**

**Dinis Fernandes - 20221848**

**Inês Santos - 20221916**

**Sara Ferrer - 20221947**

**Vidhi Rajanikante - 20221982**

**Yehor Malakhov - 20221691**

**NOVA Information Management School**

**Instituto Superior de Estatística e Gestão de Informação**

Universidade Nova de Lisboa

Abstract

The main purpose of this project, concerning machine learning, was to anticipate the enrollment of a magic student into the esteemed Wizardry School, based on the hidden patterns in the dataset, provided to us. The given dataset hat was used to build the predictive algorithm consisted of a sample which was methodically arranged into definite training and test sets consisting of roughly 700 and 200 observations, respectively, with both datasets described in 11 features. The training set included an additional feature, “Admitted in School”, whereas the test set didn’t as that feature contained the predictions to be made.

In order to accomplish this goal, various machine learning models with numerous combinations of hyperparameters were implemented and compared with our studies, by splitting the training data, revealed that the best model combination, for the predictions, was the AdaBoost Classifier ensemble method along with a Decision Tree Classifier. The models created were evaluated on their F1 and Accuracy scores. The best model created, as further discussed towards the end of the report, revealed an accuracy score of 81% in the training set and 80% in the validation set.

For the generation of the models, mentioned above, vital steps were executed and the outcomes, of the steps, were recorded in the report, divided into 5 sections. The Introduction outlines the issue and reveals the solution approach, using a predictive model to forecast the enrollment of the magic student into the Wizardry School. The Methodology explains the procedures used to build a flexible final model, applicable to diverse datasets. The Results highlighted analytical outcomes, as well as the chosen features for the final model and its correlations with the outcome variable. The Discussion explores these outcomes, from the Results section, while the Conclusion summarizes the entire report.

Overall, our final machine learning model delivers significant outcomes for the Wizardry School’s decision on the admission of the students. With an accuracy of 81% for the validation set, our final model offers a reliable forecast regarding the admission of the students into the prestigious of magical institution.

Keywords

OneHotEncoder; GridSearch; DecisionTreeClassifier; AdaBoostClassifier; Accuracy; RobustScaler; Boosting Ensemble Classifier; Hyperparameters; F1 Score; Bagging Ensemble Classifier

# 

# Introduction

In this mystical world, young wizards and witches, each possessing their own unique potential, yearn to attend the most prestigious of institutions – the esteemed Wizardry School. These schools are known not only for their grandeur but also for their ability to unlock the hidden powers within each student.

The research outlined in this report explores a detailed analysis conducted on a representative sample from the Wizardry School's student database. This sample includes individuals who have either been accepted or rejected, meticulously organized into distinct training and test sets. It is consisted of around seven hundred and two hundred rows of information respectively, both composed of eleven features that describe every student who have previously been accepted or not, with the training set including an additional one - the outcome feature.

The main goal of the presented project is to create a predictive model using machine learning techniques and methods, so that this model will play a crucial role in determining which students are suitable for enrollment in the Wizardry school, ensuring that only the most deserving candidates gain admission to the realm of magical education.

In this report, we will go through all the steps that were executed to achieve the model which best predicts whether or not, the student will be accepted in the Wizardry school.

# Background

**Datatype Transformation[1]**

Besides Machine Learning I, one of the subjects that the group had this semester was Programming for Data Science where we learned how and why to change datatypes, changing data types is crucial for optimizing memory usage, improving computational efficiency, and ensuring accurate data representations.

**Winsorization[2]**

A statistical technique that addresses outliers in a dataset by restricting extreme values to a set threshold. Rather than removing outliers, winsorization adjusts them to the nearest values within an acceptable range. This prevents extreme values from disproportionately influencing analyses, ensuring more reliable results, and as we do not delete them we get more powerful observations and insights.

**One-hot encoding[3]**

A method used in machine learning to represent categorical variables as binary vectors. Each category is assigned a unique binary code, and only one bit is set to 1 in the vector, indicating the presence of that category. This binary representation allows algorithms to process categorical data by converting it into a numerical format. We would also like to point out that usually if there exists n categories for a variable we encode n-1 if all of them are zero it means it is the other one. However in our project we decided not to encode n-1 but n as we believed it would bring us more insights and significance to our data.

**Lambda[4]**

Also known as an anonymous function, it may have many arguments yet are syntactically restricted to only one expression. It is very similar to the def function but have their differences as follows:

|  |  |  |
| --- | --- | --- |
| Aspects | Def Functions | Lambda Functions |
| Type | Defined using the keyword **‘def’** | Defined using the keyword **‘lambda’** |
| Function Naming | Holds a function name in the local namespace | Doesn’t necessarily hold a function name in the namespace |
| Interpretation Clarity | Easy to interpret | Interpretation might be tricky |
| Number of Execution Statements | Can consist of numerous execution statements | Limited operation capability |
| Return Statements | **‘return’** should be explicitly defined to return an output | No need to use the **‘return’** statement |
| Execution Time | Relatively slower for the same operation | Faster execution for the same operation |

*Table 1*

We mainly used the lambda function in the notebook for outliers and its treatment thus, condition checking. The power of lambda is better shown when you use them as an anonymous function inside another function, as shown above.

**Swarm Plot – Seaborn as sns[5]**

Seaborn, a Python library built on top of Matplotlib, specializes in data visualization. Providing a sophisticated interface statistical graphics plotting with attractive default styles and color palettes and being closely integrated with the Pandas data structure makes it rather better to use with more complex statistical plots though it has a lesser control and steep learning curve, in comparison to Matplotlib. A specific type of plot was used to check the outliers amongst the categorical variables in the dataset provided. As this plot style is also known as ‘beeswarm’, it has a better representation of the distributions of the values in the variables. It acts as a good complement to the box or violin plot in cases where you aim to display every data point while also representing the underlying distribution visually.

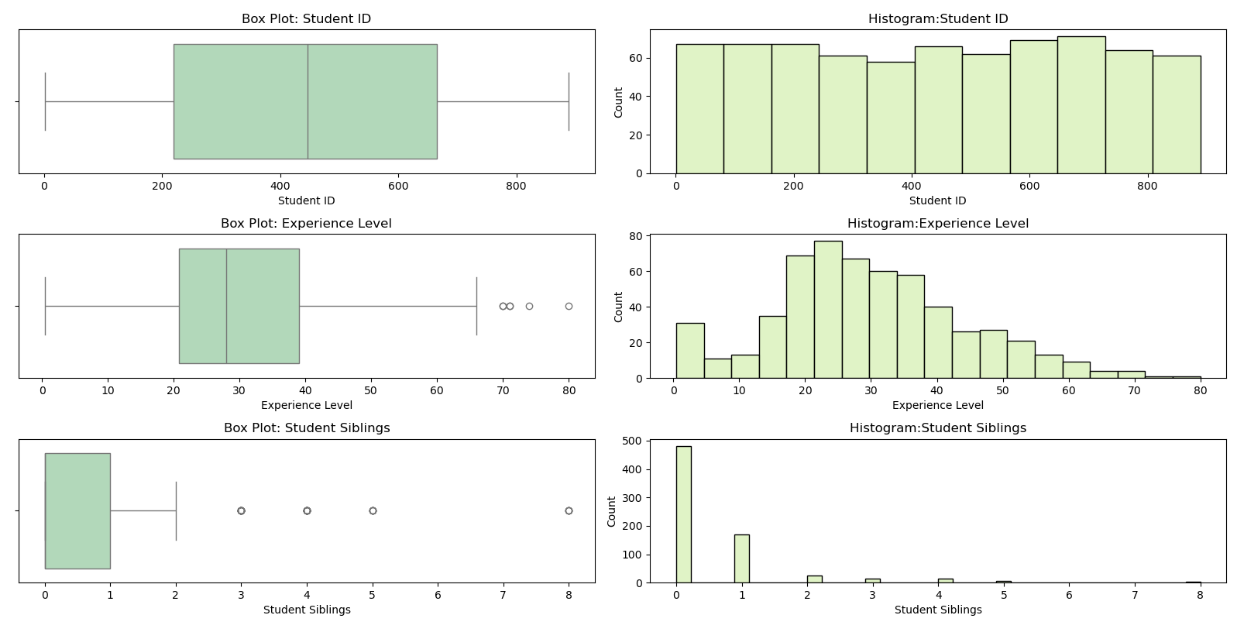
# Methodology

In a project that seeks to create an accurate predictive model, the initial step is always to go through understanding and exploring the data that will be used.

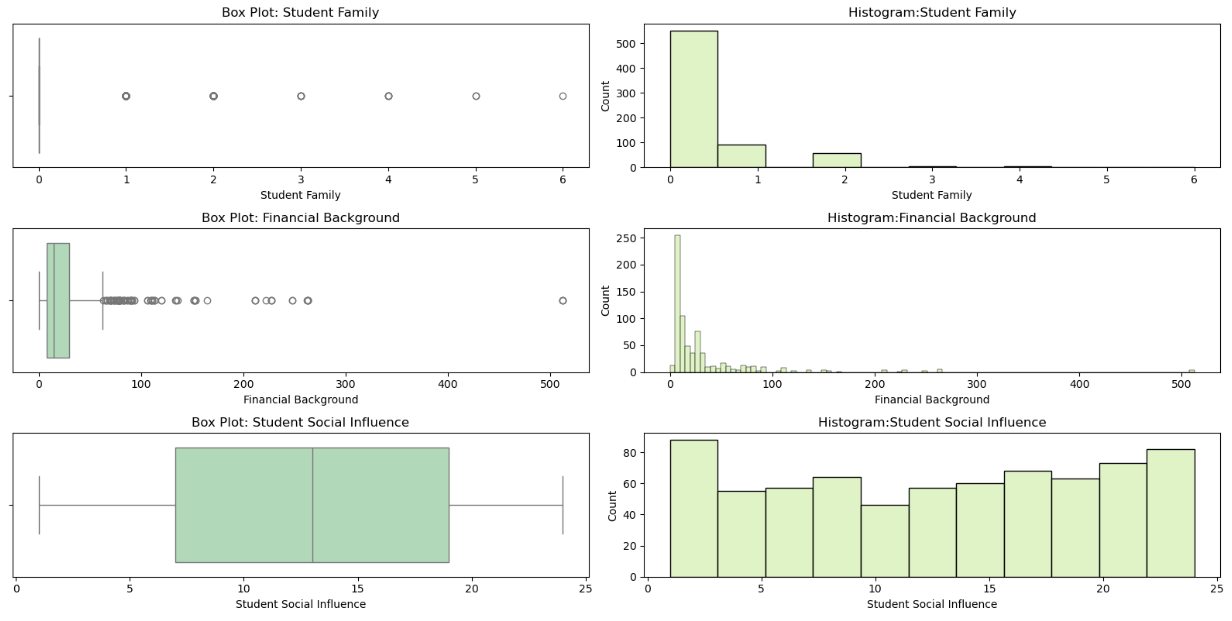
To initiate the analysis, the first step involves importing the provided data. Subsequently, the analysis can begin. Following the descriptive statistics, certain datatype changes were implemented to optimize memory usage. Upon completing these stages, the data undergoes pre-processing and feature selection processes. Once these initial phases are concluded, the development of the predictive model can commence.

In the initial phase of data exploration, the objective is to acquire insights regarding the characteristics of the dataset. These insights were obtained by taking a look at the possible existence of missing values, data types, distribution and outliers, in the dataset provided, as well as the correlations between the dataset's features.

Our group employed Pandas built-in methods, namely ‘.info()’ and ‘.describe()’, to evaluate missing values and data types in our dataset. To identify missing values, we compared the count of non-null observations (extracted from the ‘.info()’ method) with the total number of available observations. Inequality between these values indicated the existence of missing data. Additionally, by examining the 3rd quartile and maximum values for each feature, we detected potential outliers.

In order to gain a deeper understanding, we visualized feature distributions through various chart types: bar charts, histograms, box plots, swarm plots[5], count plots and scatter plots. Moreover, to explore correlations among variables, we constructed a Spearman correlation matrix and enhanced practicality by representing it as a heatmap in the correlation plot.

*Visualizations 1*



*Visualizations 2*

To start the pre-processing and the develop of the model, the training set was divided into two different 'subsets' - a train set and a validation set using a 70/30 split percentage. This division was made so that we could train our models on a sample of the dataset that had the same split percentage as our entire dataset which allowed us to get a better understanding of how the model was performing and would eventually perform when applied to the entire dataset.

Following the partition of the data into training and validation datasets we removed the outliers using the Winsorization[2] method, as they would have an impact on the improvement of our future model’s predictive capabilities. However, we decided not to apply this method to the "Financial Background" variable due to its significant impact on the outcome. We found that the higher the "Financial Background", the higher the probability of acceptance. Therefore, we applied a limit to the number itself, rather than a percentile.

Given the presence of four categorical variables, each with more than two possible values, we applied the one-hot encoding method[3] to eliminate the risk of the model inferring ordinal relationships between categories.

In addressing the missing data, our attention was focused on two variables: “School Dormitory” and “Experience Level”. For the missing data in "Experience Level," we opted to employ the KNN Imputer, with 7 neighbors in order to avoid ties when making decisions based on majority vote. As for "School Dormitory", being a categorical variable with 79% missing data and a minimal correlation with the outcome variable, we decided to drop the variable, to avoid potential biases launched by imputing a substantial number of missing values in a variable.

Then different scaling methods were performed to see which one functioned better in our dataset, more explicitly, Min Max Scaler, which scaled the data between 0 and 1, Standard Scaler, which scaled based on the mean and their variance and finally Robust Scaler, which scaled our data based on interquartile range.

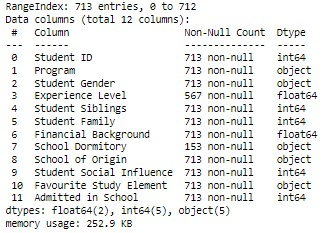
After dealing with the pre-processing, we then proceeded to what is another crucial step of any machine learning project - Feature selection. We determined which features should be kept for the future model and saw which ones play a more significant role, four different methods were employed namely the Correlation, the RFE and the Lasso Regression tests and to untie the results Random Forest. Having concluded the feature selection, we excluded the variables that showed weaker results in the above-mentioned tests.

After all these steps were done, we tested various models to determine what would be the final and best one. After some research, we found that the most promising method, in terms of enhancing the predictive accuracy of a model, was the usage of the Boosting Ensemble Classifier based on a basic decision tree of a maximum depth of 1. To reach this final model, various combinations of models were tested as well as different sets of hyperparameters,based on GridSearch. Also in some cases where the tested models did not perform well, even after the tuning of the hyperparameters was made, we tried to restart the process from the feature selection step, until a good model was realized.

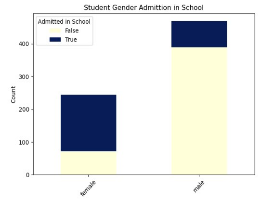
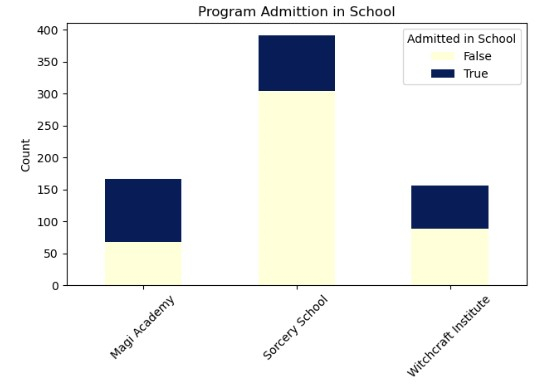
# Results

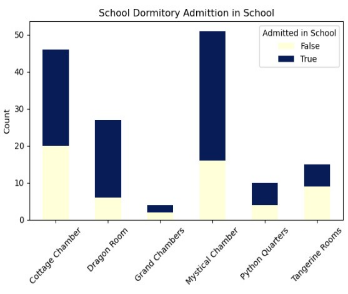
**Data Exploration**

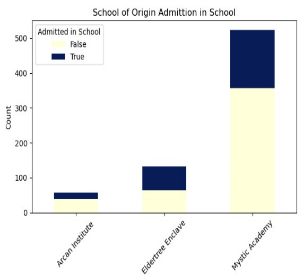
During the analysis of the original dataset, in addition to having an initial look at every variable’s data type, we observed that missing values exist in two of the features: “Experience Level” and “School Dormitory”.



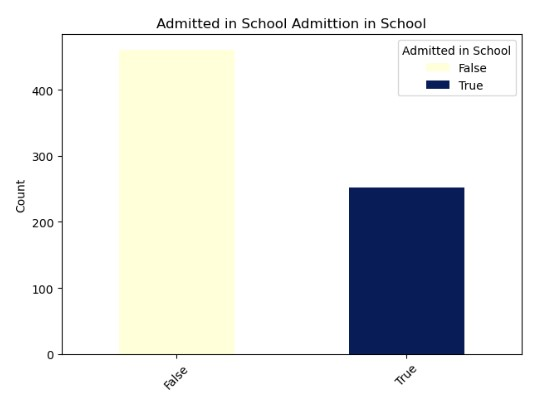
In regards to the categorical variables (except the “Favorite Study Element” variable that has a very similar distribution):





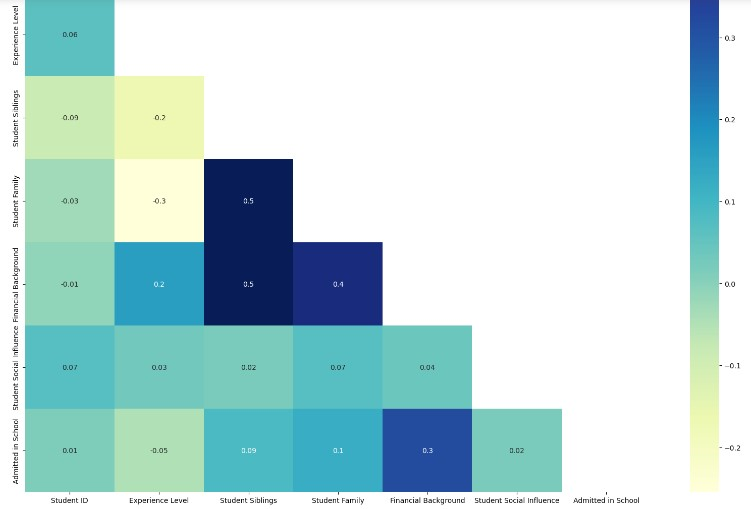


*Visualizations 3*

In regards to the outcome variable, we conclude that the data is unbalanced, having more students that were not admitted in school then the ones that were .

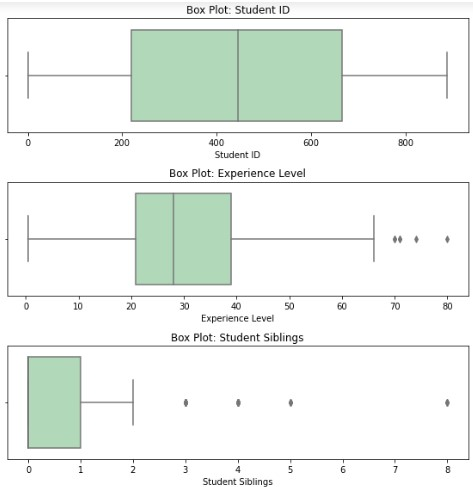
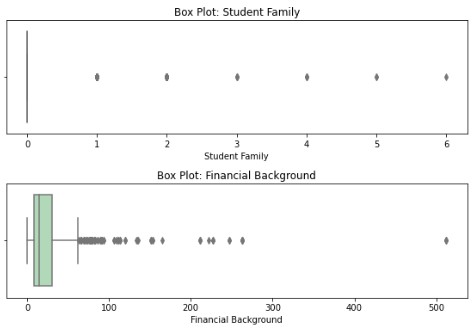
*Visualizations 4*

When examining the correlation heatmap, it is evident that there are significant dependencies between certain features, particularly in the cases of:

* Student Family and Student Siblings
* Financial Background and Student Siblings
* Admitted in School and Financial Background

*Visualizations 5*

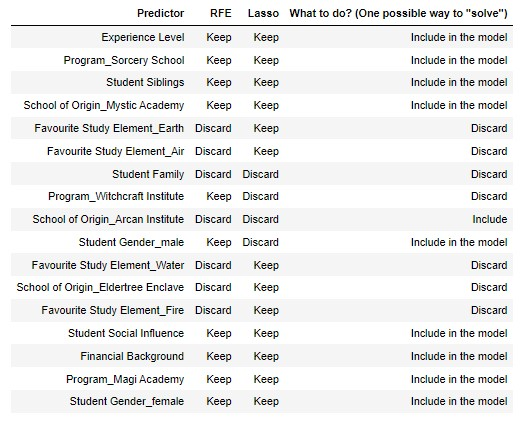
Analyzing the distribution of numerical variables, we realized that the following variables have potential outliers: 'Experience Level', 'Student Siblings', 'Student Family' and ‘Financial Background'. These conclusions were supported by the use of box plots, shown below.



*Visualizations 6*

**Features’ Datatypes**

By analyzing the features’ datatypes, we realize that some should be changed[1], specifically. The features ”Program”, ”Student Gender”, “School Dormitory”, ”School of Origin” and “Favorite Study Element” should be considered as categorical and, on the other hand, the outcome variable, “Admitted in School”, should be considered as Boolean. The datatypes of these variables were modified to ensure accurate representation.

**Feature Selection**

*Table 2*

**Final Model**

After various combinations of hyperparameters were tested, there was one that stood out from the remaining ones. The model that gave us the best results and less Overfitting was the ‘AdaBoostClassifier()’. As the AdaBoost is our final model, we need a previous model to use as a base model that the AdaBoost can use to boost its performance from, using the algorithm ‘SAMME’, a learning rate of 0.8, and 100 estimators,to make better predictions with an improved accuracy score.

A black and white text with numbers

Description automatically generated with medium confidence

*Figure 1 - Scores of the Boosting Ensembling*

While examining the difference, in F1, between the training and validation sets, we can come to a conclusion that there is no overfitting in the final model. Though it is crucial to bear in mind that the dataset displays a slight imbalance. Additionally, evaluating the individual F1-score for each potential outcome is highly significant. Notably, this model demonstrates remarkable accuracy in scenarios where the target value is a numerical Boolean value of 1.

# discussion

As previously stated in the Results section, that the dataset provided to us, to make the predictions, is slightly unbalanced, thus our models predicting more ‘1’s than ‘0’s on the outcome variable. However, we have two models who present overfitting, those are, Knn and Random Forest, the first with a difference of 30% between train and validation data and the second one with a difference of 11%. This is can be due to the hyperparameter tunning of the models, the features that we selected or even because the models are too complex for the dataset. As we only have overfitting on those two models we can surely state that there is no data leakage. Finally we choose the Boosting Ensemble Classifier based on a decision tree with depth of 1 because it presented a 0.73% of F1-score on validation data, having less than 1% difference to the train data and also presented 84,21% on f1-score on 11% of our test data. In a close second came a Bagging Ensemble Classifier also based on a decision tree, this classifier had a highest score on train, and also 73% on validation, however we chose the Boosting Classifier as the difference between train and validation was smaller.

From what was deduced in the previous section, the final model is able to predict the enrolment of a magic student into the Wizardry School, which in our case should be the priority, considering that that is the main purpose of the project. Though, as mentioned above, we do have one variable which has influence on the value of the output variable, which could make it ‘biased’. Since we have concluded that we have a slightly unbalanced dataset, which is why the model is predicting more acceptance than rejection, we may also include that the “Financial Background” feature may also have an impact on that.

Furthermore, there is another detail in the fact that our solution is not the friendliest in terms of computational complexity,as we used gridsearch for all parameters for all models. We would also like to note that highly correlated variables were kept, namely the Gender Male and Female, but also the encoding being done with n categories and not n-1. These two issues are at the top of the team’s difficulties in carrying out this project, because the decisions we ended up making go against what is considered to be good practice in the development of a project of this kind. However, the sacrifice of the conventional ideology was necessary.

This happened because, if we work with either models with identical train and validation accuracies or models without highly correlated variables, accuracy is lost - which goes clearly against the main goal of the project. Having this in mind, the risk had to be taken. We believe that it would be very beneficial for the company to perform this same study with a dataset with more observations. This could lead to a bigger amount of information being available in terms of defining the impact of the dataset’s features and consequently to a better understanding of their predictive importance.

# Conclusion

Overall, the final model used was the AdaBoost Classifier, in which the Decision Tree Classifier was used as an estimator with a learning rate of 0.8(having the parameters of the Decision Tree Classifier as the following: max\_depth=1, random\_state=15). In comparison o the other models, AdaBoost was the one that gave us the highest accuracy score with no overfitting, and on Kaggle our model presented a 84,21% on F1-Score, as a public score. This model parameters was were solely defined by Gridsearch, we do recognize that this might not be the best practice.

Though the model was good for our predictions, we may not be quite happy with the predictions because we have a slightly unbalanced dataset, and our model is now predicting more acceptance and denial of students. This is because, assuming that we are doing predictions on the esteemed Wizardry School, being on of the most prestigious of all institutions, clearly the acceptance rate of Wizardry School should be low. Having this into consideration, we can say that our predictions would be better if our dataset would be balanced, and with more observations as one of our biggest challenges was the reduced number of observations. Thus, this will help Wizardry School make appropriate decisions on the students’ enrollment. It is crucial for the school to use this work as support for future admissions and attempts of evolving a similar model in order to improve further decision-making on the admissions into the prestige Wizardry School.

# References

[1] Carina Albuquerque (2023), *Handling Large Datasets.* <https://elearning.novaims.unl.pt/pluginfile.php/211551/mod_resource/content/3/08_Handling%20Large%20Datasets.pdf>

[2] GeeksforGeeks (2021), *Winsorization.* <https://www.geeksforgeeks.org/winsorization/>

[3] Arthur Lamblet Vaz (2018), *One-hot-encoding, o que é?* <https://arthurlambletvaz.medium.com/one-hot-encoding-o-que-%C3%A9-cd2e8d302ae0>

[4] GeeksforGeeks (2023), *Python Lambda Functions.* <https://www.geeksforgeeks.org/python-lambda-anonymous-functions-filter-map-reduce/>

[4] Ibrahim Abayomi Ogunbiyi (2022), *How the Python Lambda Function Works – Explained with Examples.* <https://www.freecodecamp.org/news/python-lambda-function-explained/>

[4] GeeksforGeeks (2021), *Difference between Normal def defined function and Lambda.* <https://www.geeksforgeeks.org/difference-between-normal-def-defined-function-and-lambda/>

[5] Carina Albuquerque (2023), *Data Visualization with Matplotlib and Seaborn.* <https://elearning.novaims.unl.pt/pluginfile.php/209950/mod_resource/content/1/06_Matplotlib%20and%20Seaborn.pdf>

[5] GeeksforGeeks (2022), *Swarmplot using Seaborn in Python.* <https://www.geeksforgeeks.org/swarmplot-using-seaborn-in-python/>

[5] Waskom M. L., (2021). *seaborn.swarmplot.* <https://seaborn.pydata.org/generated/seaborn.swarmplot.html>

[6] Ray S. (2023). *8 Ways to Improve Accuracy of Machine Learning Models (Updated 2023).* <https://www.analyticsvidhya.com/blog/2015/12/improve-machine-learning-results/>

