Data Mining Project
Part 1 : EDA and classification

Le Gall Mattéo

1 Introduction

This project aims to use various data analysis methods to solve a specific problem. I've chosen to work with a Bank marketing dataset from a Portuguese banking institution's marketing campaign. Using the available information, the goal is to predict if a client will sign up for a term deposit.

The first part of this project involves exploring the dataset thoroughly, looking for patterns and insights through data analysis. We will also use different methods to predict client behavior and carefully measure how accurate these predictions are. We will need to take into account different factors such as class imbalance.

Better understanding the clients behavior and how they react to different marketing strategies will allow the bank to use its resources, like time and money, wisely. This understanding helps identifying the specific factors that drive clients to subscribe to a term deposit, ensuring efficient resource utilization and avoiding unnecessary efforts.

Ultimately, this project aims to show how useful these data analysis techniques are in solving real-world problems, especially in predicting client behavior for banking services.

Preamble

All of the following graphics along with more and the code are available on the two .ipynb attached. They are also available on the following github repository with the original .png files in better quality.

2 Methods

2.1 Exploratory Data Analysis

Multiple methods were used for the exploratory data analysis to gain information in the dataset, as shown in the Results section.

Univariate analysis For univariate analysis, we used bar plots and box plots to gain information on the different features. It allows us to see the existence of outliers and to see the repartition and the frequency of each feature. Plots were also made by grouping the data depending on their class to see which feature has the best discriminative abilities.

Multivariate analysis To get a better understanding on the relations between the features, we first computed a correlation matrix between the numerical features using Pearson's correlation coefficient. It allows us to understand if some features are correlated to others. This can decrease the training time and the computational power needed as highly correlated features can be projected into a smaller dimension.

We also computed Cramér's V coefficient between the categorical variable to get their correlation. This coefficient is based on Pearson's chi-squared test and is computed by taking the square root of the chi-squared statistic divided by the sample size and the minimum dimension minus 1. As for the Pearson's correlation coefficient, a score of 1 means total association between two features and a score of 0 means no association between the variables.

Data preparation After having done the exploratory data analysis, the data was processed to be ready for classification. 'pdays' feature was transformed to a binary variable that is equal to 1 if a client had already been contacted. The categorical features were transformed with one-hot encoding that uses dummy variables to transform categorical variables to multiple binary variables. We could have used a simple encoding for 'education' that has a logical ordering but the presence of the 'unknown' feature caused a problem so it was not done.

In a second part, we also removed 3 of the 4 highly correlated features to see if it would greatly impact the results. An other feature (day_of_week) was deleted as it did not show any discriminative ability during exploratory data analysis.

2.2 Classification

Imbalanced data Multiple methods exist to deal with imbalanced data. We will use the two most common methods along with a control run where we will not change anything to see if those methods works and are useful.

The first method used will be oversampling. We will oversample the minority class by duplicating with replacements minority class' observations until the

minority class has as many observations as the majority class. This increase the size of the dataset but its advantage is that it does not lose useful information. This is the basic method for oversampling, more advanced methods exist like SMOTE (Synthetic minority oversampling technique) where synthetic data is created with k-nearest neighbors instead of duplicating existing entries but we will not use this kind of methods.

The second method used does the opposite and is called undersampling. We randomly sample a part of the majority class data to have the same number of observations for both classes. This method reduces the number of entries so it loses information but also reduces the size of the dataset and can be helpful to reduce the training time. In small datasets, the information loss can be very important but in bigger ones, it may not have a significant impact.

Moreover, we will use the balanced accuracy metric that is defined as the average of recall obtained on each class. This metric takes into account the imbalanced class problem and is more useful in our problems to show what happens when you do not tackle class imbalance.

Cross-validation We use a standard stratified repeated k-fold with 5 repeats and 2 folds for the cross-validation. This is a very popular method and is used in most research papers. Two folds means that each time the data is cut in two equal parts for testing and training and both folds takes in turn the role of training and testing. We repeat that procedure 5 times to finally get 10 different run of an algorithm. The stratified keyword means that the ratios of each classes will be respected in each fold. This is especially useful when you do not use undersampling or oversampling to get more coherent results with less noise.

Algorithms used The classification was done with 5 methods : Gaussian naive bayes, linear discriminant analysis, quadratic discriminant analysis, K-nearest neighbors (where K=5) and Random Forest.

The Gaussian naive bayes algorithm is a simple method based on Bayes theorem and the assumption that the features are independant and follow a gaussian distribution. It is a very simple and well known algorithm that can still give good results in a lot of situations, and can be very good with small datasets.

Linear and quadratic discriminant analysis (LDA and QDA) are two methods that we discussed in class with a linear and a quadratic decision surface. LDA also assumes gaussian distribution and aims to find a linear combination of features to separate the classes in the best way. QDA is similar but does not assume that the covariance matrix are equal for each class, thus it leads to more parameters to estimate but can be more flexible.

K-nearest neighbors is a well-known algorithm for classification. It classifies new data entries by looking at the classes of the k-closest neighbors. It is simple to implement and comprehend, in our problem we decided to use K=5 which is a very common choice for this algorithm.

Random Forest is an ensemble learning method similar to bagging that also select a subset of features at each splitting of the tree and constructs multiple

trees, with each tree learning on a subset of the data. The decision is made through a majority vote with all the trees. It is a robust method that is often simpler to use and train than boosting.

3 Results

3.1 Exploratory Data Analysis

Data characteristics This dataset comprises 41188 entries and 20 columns of features plus the last column that represents the class of each input, either yes or no.

We can observe multiple quantitative and qualitative features, mostly continous and nominal features. There are no missing values in the dataset, but 6 of the 10 categorical features have an option "unknown" that is used when the information about the feature is not known. These unknown values will not be considered as missing values and will not be deleted as not knowing an information about a client could have an impact on the classification.

We can also observe the fact that the data is not balanced with approximately 11% of positive cases and 89% of negative cases, we will need to take that into account for the classification.

Univariate analysis There are 10 numerical features as we can see on the figures 1 and 2, most of them seem to have asymetric distributions. We can see that 'pdays' is split between values smaller than 100 and other values that represent the majority of entries where pdays=999. This is because 999 is used when the client has never been called before. We will need to modify that before proceeding. There will be a need to standardize/normalize the data as the ranges of values are very different from one feature to another.

We get a lot of "outliers" on 'duration' and 'campaign' features as these values are quite small for the vast majority of client. However, these don't seem like "wrong" values or errors as it is possible that some clients had long conversations or were contacted a lot of times if the campaign lasted a long time. We get the same problem with 'previous', where the majority of clients were not contacted before but it does not mean that the ones who were are wrong entries.

So we will not delete these outliers as they seem like normal values and not errors. The same can be said about the 'age' feature as there are no shocking values with the largest being 98 years old. We can see a data point shown as an outlier on cons.conf.idx. As it is the only one, we may erase this entry before classification as it will not impact the data comprised of 41188 entries.

By grouping with the class figure 3, we observe small differences for the majority of features. We can still observe some difference on the median for

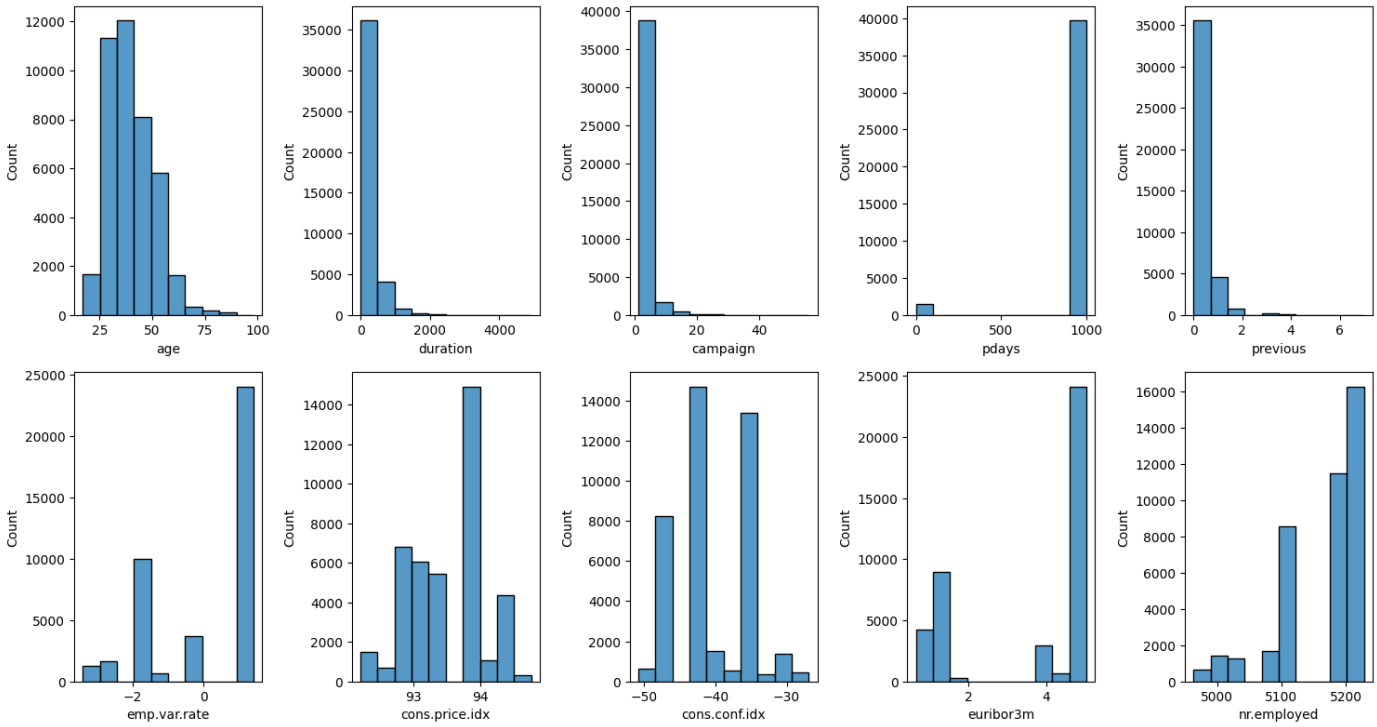

Fig. 1. Bar plot of the numerical variables

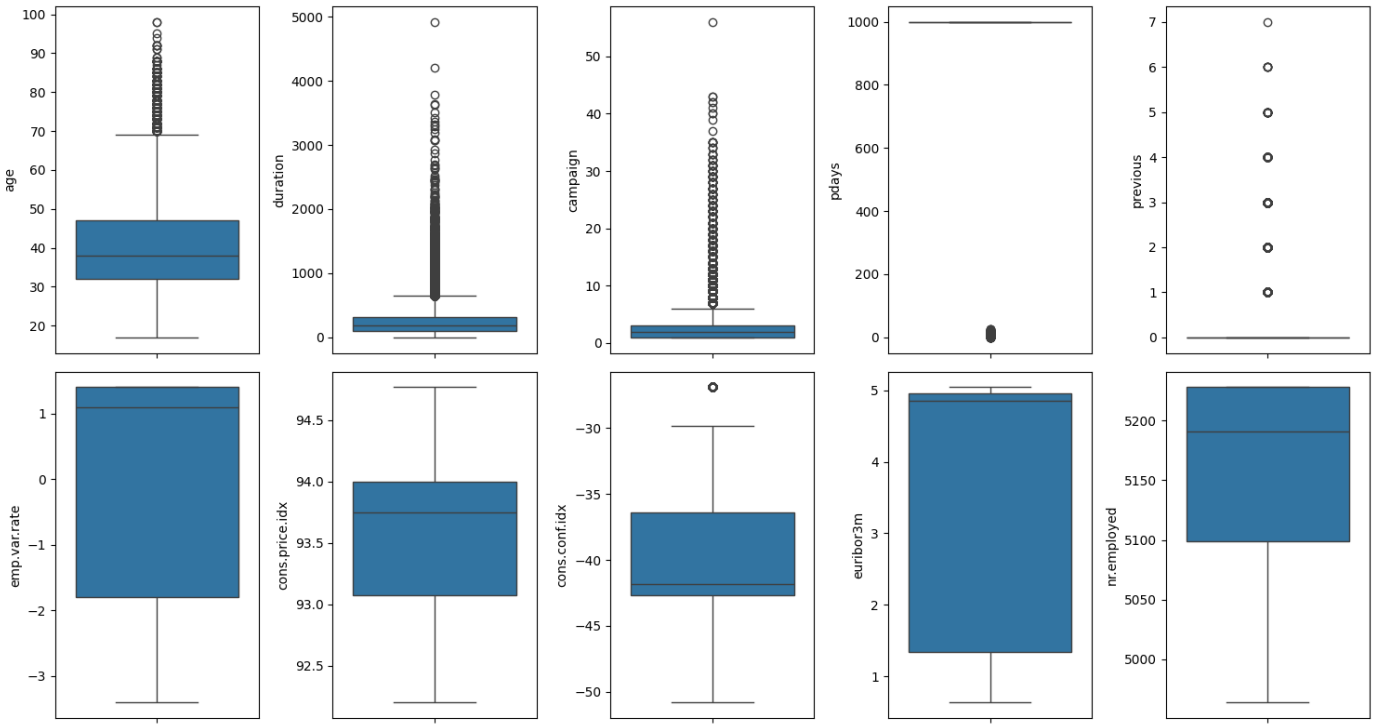

Fig. 2. Box plot of the numerical variables

'euribor3m' and 'cons.price.idx'. 'nr.employed' also seems to have some discriminative abilities.

The 'duration' feature seems to be the best to discriminate with a clear separation of the boxplots for both classes.

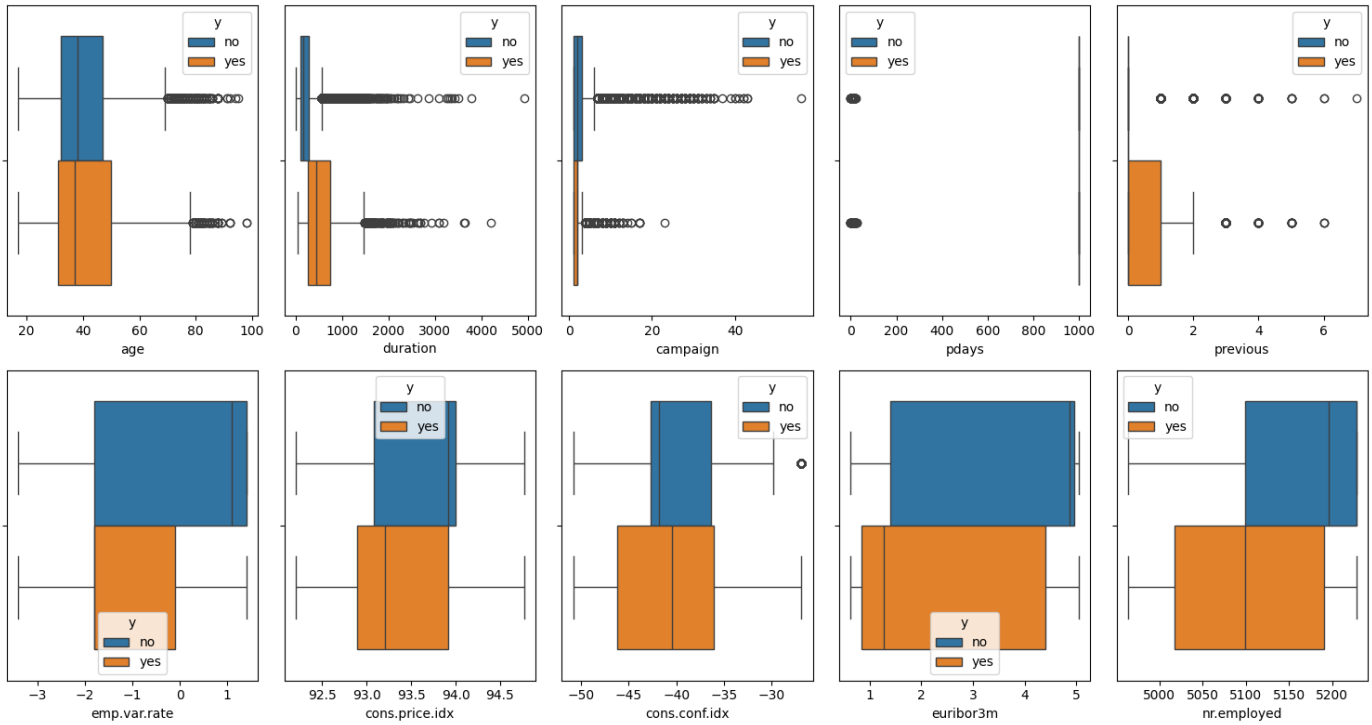

Fig. 3. Box plot of the numerical variable grouped by class

The bar plot of the categorical features figure 4 shows us the imbalance of the class labels with the vast majority being no, meaning people that did not subscribe a term deposit.

We can think that day_of_week doesn't seem to bring much to the table as the ratio yes/no seems to be the same for each day. We will confirm it later but it may be a feature that we can delete.

Other features seems to show some variability with different frequency for categories and different ratios of yes and no for those categories. Especially we can directly see on the figure 5 that the feature 'contact' seems to be able to discriminate well. The discriminative ability of the other features is not clear by just looking at these graphics.

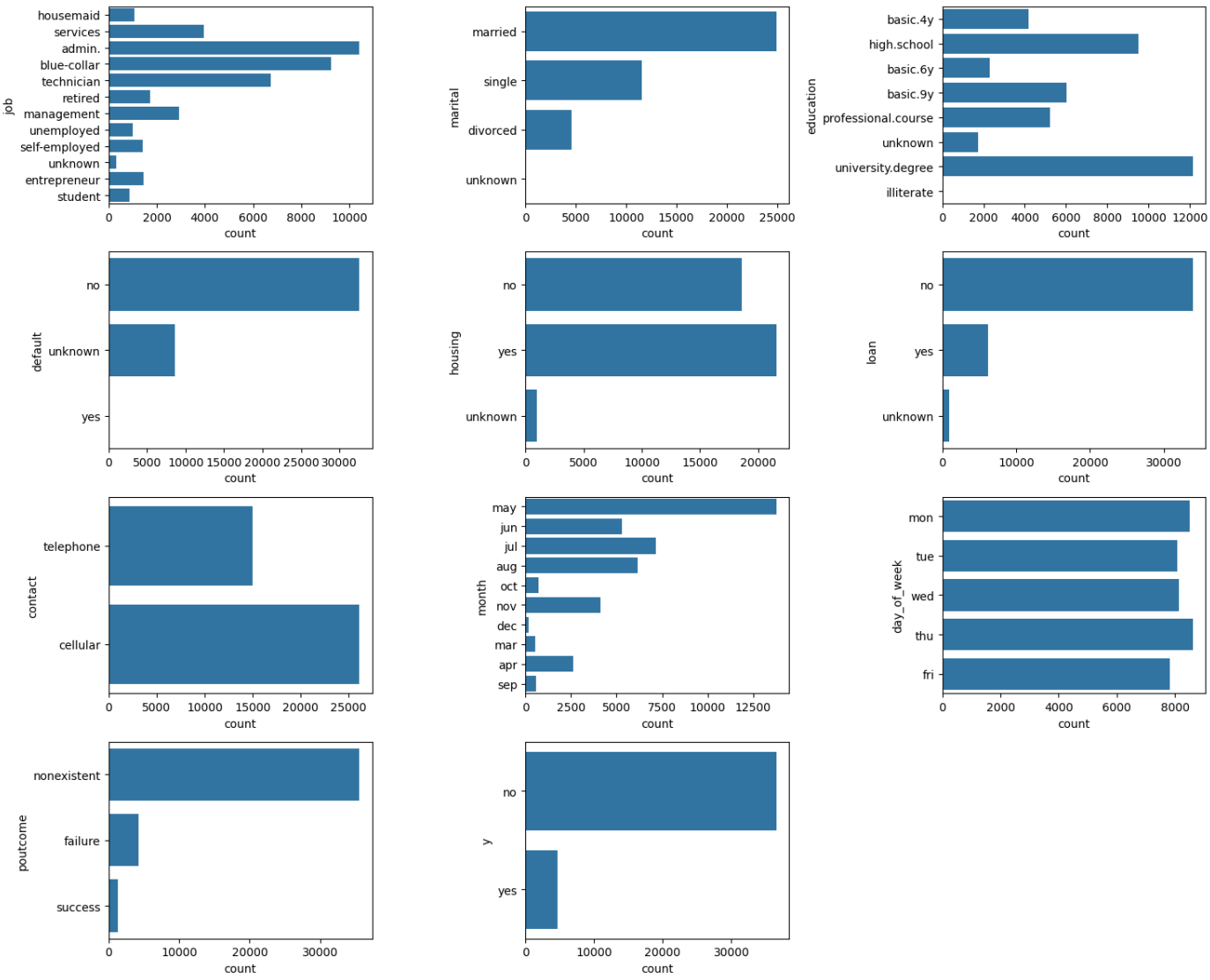

Fig. 4. Bar plot of the categorical data

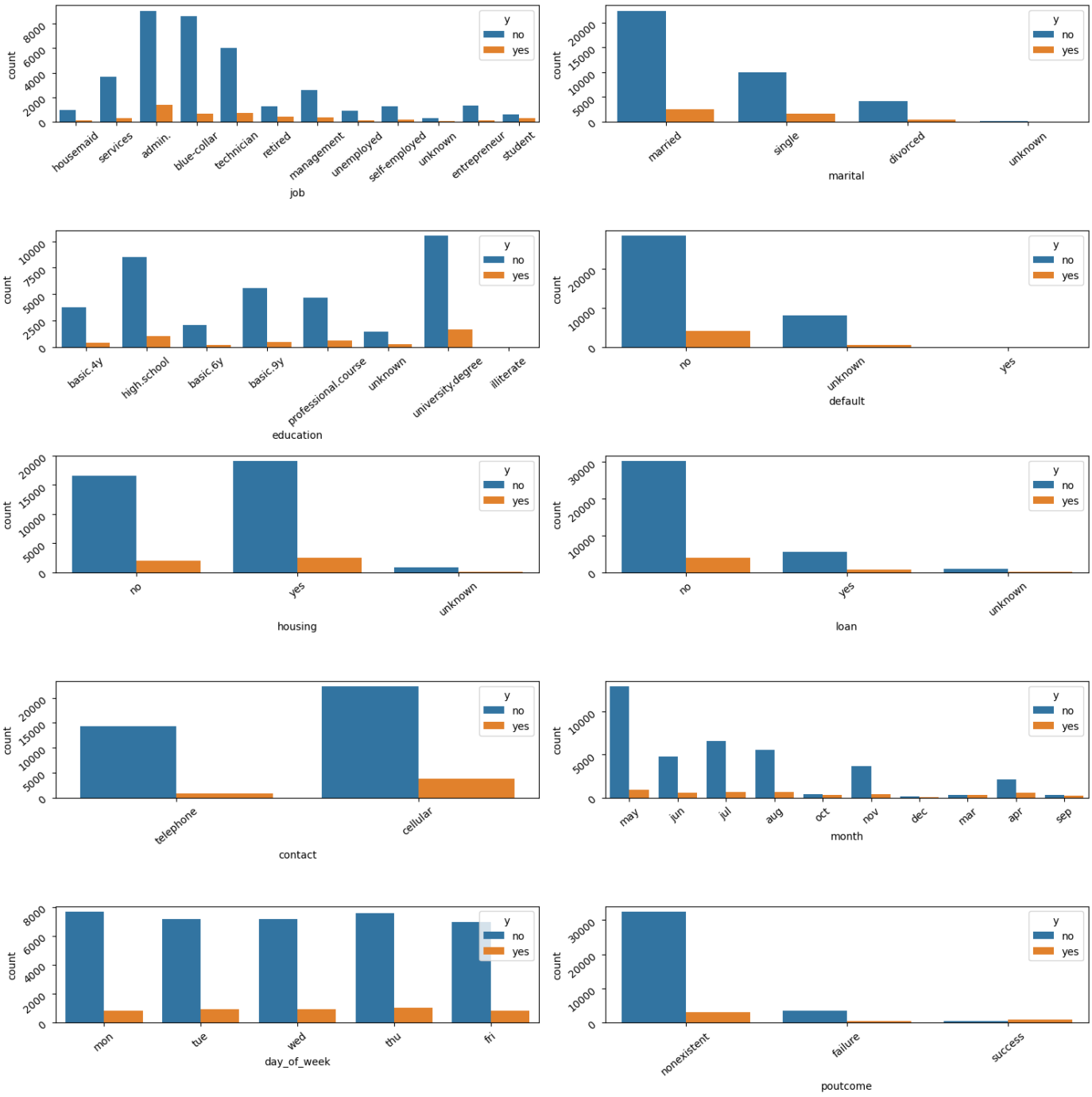

Fig. 5. Bar plot of the categorical data grouped by class

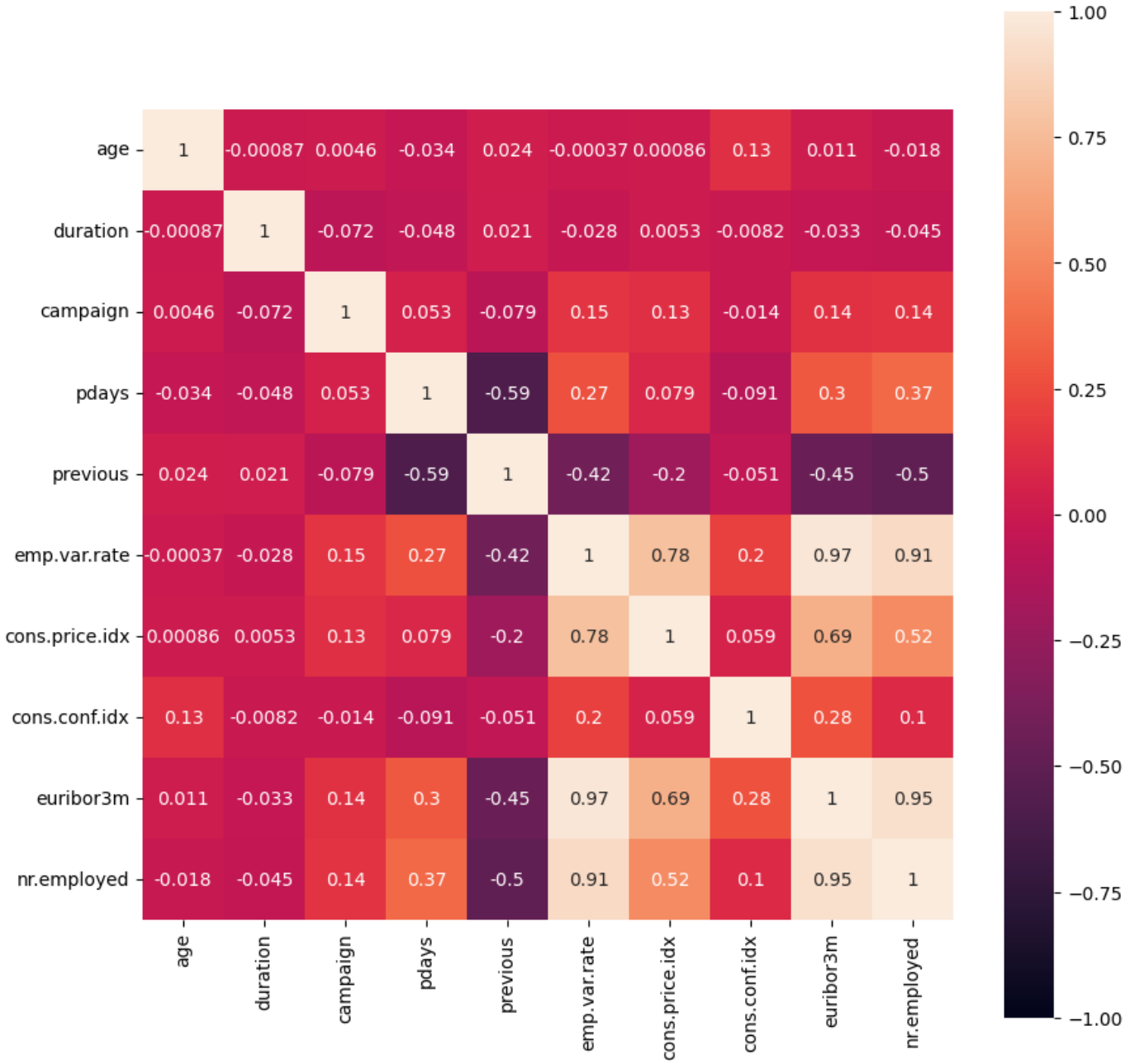

Fig. 6. Correlation matrix of the numerical variables, using Pearson's correlation

Multivariate analysis We present figure 6 the pairwise correlation of the numerical features using Pearson's standard correlation coefficient. We can observe that most of the features aren't correlated.

However there exist very strong correlation between 'emp.var.rate', 'nr.employed' and 'euribor3m' with coefficients greater than 0.9. There is also correlation between these three categories and 'cons.price.idx', albeit a little bit smaller.

There is also non-negligible inverse correlation between 'previous' and 'pdays', and 'previous' is correlated to multiple features, even though the coefficients are smaller.

	job	marital	education	default	housing	loan	contact	month	day_of_week	poutcome
job	1	0.18359	0.359526	0.152101	0.0106304	0.0102178	0.127856	0.109835	0.0164577	0.0995598
marital	0.18359	1	0.11624	0.0954341	0.00916995	0	0.0719935	0.0501735	0.0108883	0.0366305
education	0.359526	0.11624	1	0.170355	0.013316	0	0.123302	0.0947177	0.0197251	0.0422816
default	0.152101	0.0954341	0.170355	1	0.0105731	0.00156678	0.135554	0.111926	0.0113136	0.0766283
housing	0.0106304	0.00916995	0.013316	0.0105731	1	0.707852	0.0846035	0.0542431	0.0146348	0.0169577
loan	0.0102178	0	0	0.00156678	0.707852	1	0.0242059	0.0198297	0.00610372	0
contact	0.127856	0.0719935	0.123302	0.135554	0.0846035	0.0242059	0.999948	0.609087	0.0549063	0.242419
month	0.109835	0.0501735	0.0947177	0.111926	0.0542431	0.0198297	0.609087	1	0.0665687	0.2424
day_of_week	0.0164577	0.0108883	0.0197251	0.0113136	0.0146348	0.00610372	0.0549063	0.0665687	1	0.0145787
poutcome	0.0995598	0.0366305	0.0422816	0.0766283	0.0169577	0	0.242419	0.2424	0.0145787	1

Fig. 7. Correlation matrix of the categorical variable using Cramér's V statistic

Using Cramér's V statistic, we get the correlation between the categorical features figure 7. We can observe a correlation between 'housing' and 'loan', but also between 'month' and 'contact'. There is also a weaker correlation between 'education' and 'job'.

3.2 Classification

We can see the average results for each of the five algorithms used for classification figure 8. We achieved almost 85% of balanced accuracy with the best method : Random Forest. 5-nearest neighbors and linear discriminant analysis follow closely with a score greater than 80%. The gaussian naive bayes achieves 73% of accuracy, and we can also note that the standard deviation is really small which means that this method is less impacted by other parameters (randomness with repeated stratified k-fold, different subsets of data or different sampling methods). Finally, quadratic discriminant analysis has the worst results, barely exceeding 60% of balanced accuracy.

The plot figure 9 shows the results depending on the sampling method (doing nothing, undersampling or oversampling). We can immediatly see that not taking into account the imbalanced class problem leads to worse results. Indeed we only get 68% of precision that way when we get more than 80% using oversampling or undersampling. In our case both methods worked almost as well. In problems

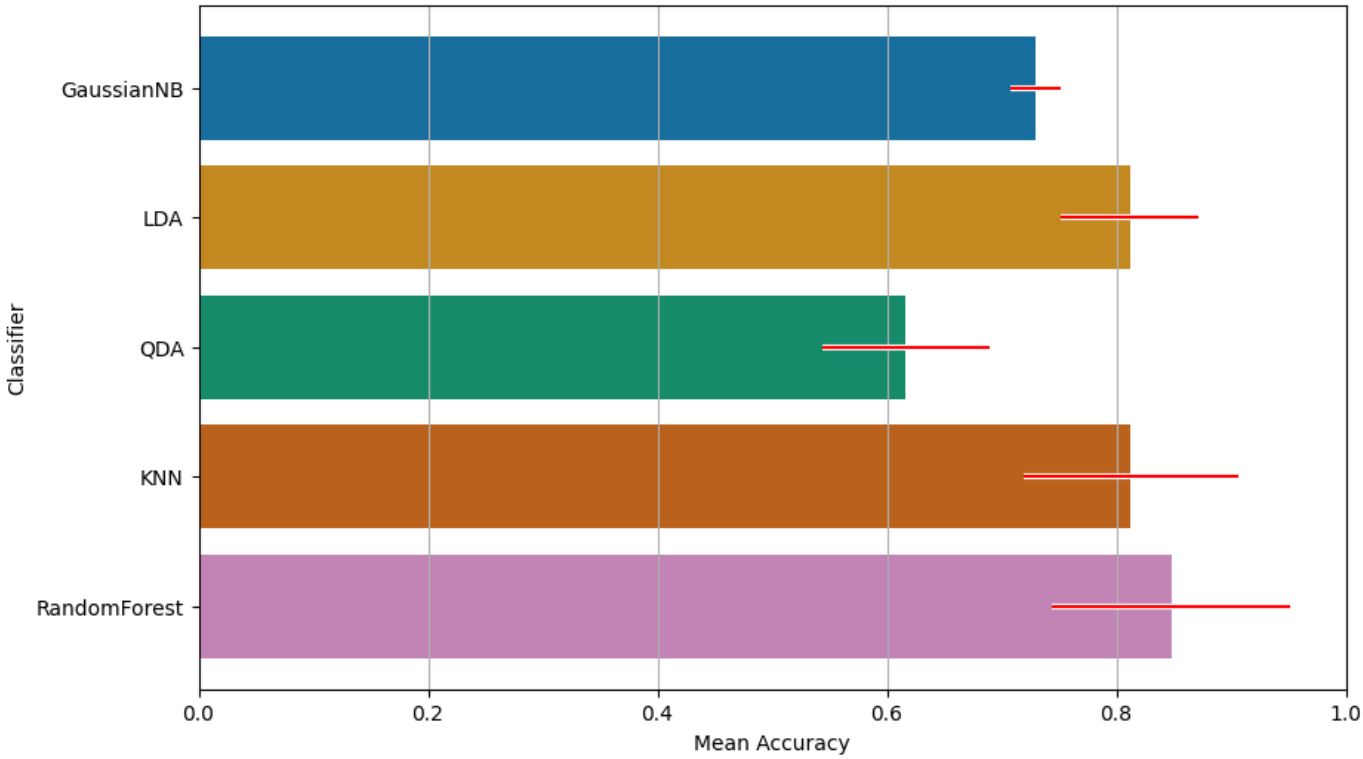

Fig. 8. Mean accuracies with standard deviations for different classifiers

where we do not have a lot of data, undersampling would probably get poorer results as it causes a loss of information.

By averaging the results depending on the dataset we used, we can see that we get very similar results (a difference of 0.5%) which means that the columns we removed were indeed correlated with others and it did not lead to a loss of information. That kind of methods can lead to a decrease in computational power and training time, even though it was not necessary in our case as we did not use computationally expensive algorithms.

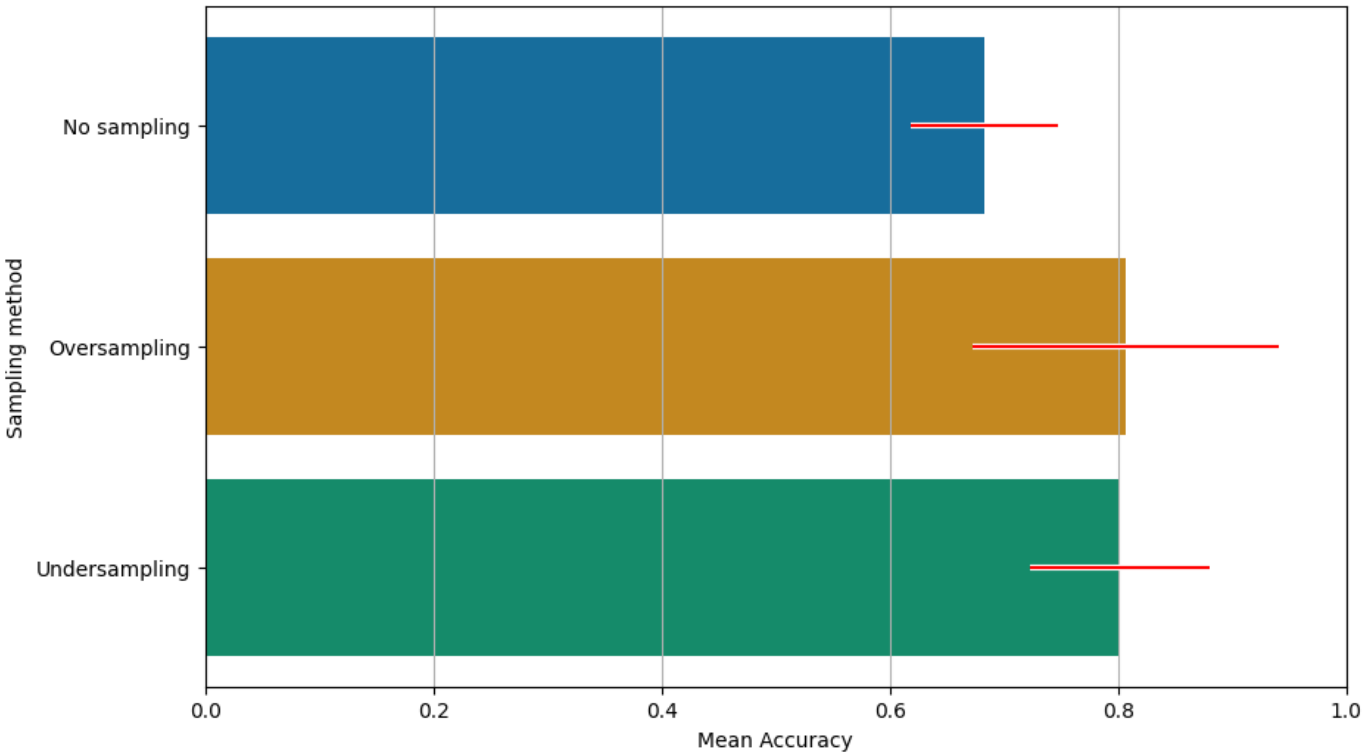

Fig. 9. Mean accuracies with standard deviations for different sampling methods

4 Conclusion

Finally, in that first part of the project we worked on understanding the dataset and using multiple classification methods to be able to predict. We now understand that the problem of class imbalance should not be ignored and that oversampling and undersampling can greatly help to improve the results. We have also shown that some classifier are better than others and random forest gets the best results in our problem.

Moreover, this first part of the project underscores the importance of precision in resource allocation. Understanding the important elements that drive client subscriptions allows for a more targeted and efficient approach, enabling the optimization of the bank's efforts and resources.

In a more advanced work, we could have used more methods to tackle data imbalance (like SMOTE), more algorithms for classification (neural network, boosting, ...) and different subsets of the data. We could also have created new features from the existing one to see if it would improve the results. We could also have tuned the hyperparameters of the algorithms we used to increase our accuracy scores.