POLITECNICO DI TORINO

MATHEMATICS IN MACHINE LEARNING

BANK MARKETING ANALYSIS

CANDIDATE:

ALBERTO MARIA FALLETTA S277971

PROFESSORS:

FRANCESCO VACCARINO

MAURO GASPARINI

TABLE OF CONTENTS

1. INTRODUCTION

Marketing, sector devoted to designing strategies to enhance business by identifying, anticipating and satisfying customers' needs, is one of many fields benefitting from machine learning techniques for data driven decision-making processes. The focus is on maximizing customer lifetime value through the evaluation of available information and customer metrics. Using machine learning algorithms on data gathered from marketing campaigns, in fact, it is possible to select the best set of clients for a give product or service (i.e. the ones more likely to subscribe).

The current analysis [i] is based on real data gathered by a Portuguese retail bank [ii], from May 2008 to June 2013, in a total of 41.188 phone contacts related to a marketing campaign about the promotion of long-term deposits. Within a contact, the client was asked to subscribe the deposit therefore the result is a binary unsuccessful ('no') or successful contact ('yes'). The dataset is unbalanced, as only 4.640 (11.27%) records are related with successes.

2. DATA OVERVIEW AND DESCRIPTION

The dataset in analysis contains 41.188 distinct contacts between the bank and the client, each of which associated with several client's personal data as well as data from previous contacts:

- Age (numeric)
- Job (categorical: 'admin.', 'blue-collar', 'entrepreneur', 'housemaid', 'management', 'retired', 'self-employed', 'services', 'student', 'technician', 'unemployed', 'unknown')
- Marital status (categorical: 'divorced', 'married', 'single', 'unknown'; note: 'divorced' means divorced or widowed)
- Education level (categorical: 'basic 4y', 'basic 6y', 'basic 9y', 'high school', 'illiterate', 'professional course', 'university degree', 'unknown')
- Has credit in default (categorical: 'no', 'yes', 'unknown')
- Has housing loan (categorical: 'no', 'yes', 'unknown')
- Has personal loan (categorical: 'no', 'yes', 'unknown')
- Last contact type (categorical: 'cellular', 'telephone')
- Last contact month (categorical: 'mar', 'apr', ..., 'nov', 'dec')
- Last contact day (categorical: 'mon', 'tue', 'wed', 'thu', 'fri')
- Last contact duration in seconds (numeric)
- Number of contacts during this campaign for this client (numeric)
- Number of days passed by previous contact (numeric, '-1' means the client was never contacted before. Original dataset has '999' instead but was mapped to '-1' for visualization reasons)
- Number of contacts before this campaign and for this client (numeric)
- Outcome of previous contact (categorical: 'failure', 'nonexistent', 'success')
- Other socio-economic numeric attributes ('emp.var.rate', 'cons.price.idx', 'cons.conf.idx', 'euribor', 'nr.employed'. Numeric and all normalized)
- Output variable: has the client subscribed (binary: 'yes', 'no')

3. DATA EXPLORATION

3.1 BALANCE

Data exploration is the first step of the analysis, being a mandatory practice to acquire a thorough understanding of data and their distributions. In this sense, the class label distribution is of paramount importance since the balance or imbalance of the dataset depends on it. It is "balanced" a dataset containing equal or almost equal number of samples from each class, "unbalanced", as in the case of the current analysis, if the samples from one of the classes outnumbers the others (Figure 1).

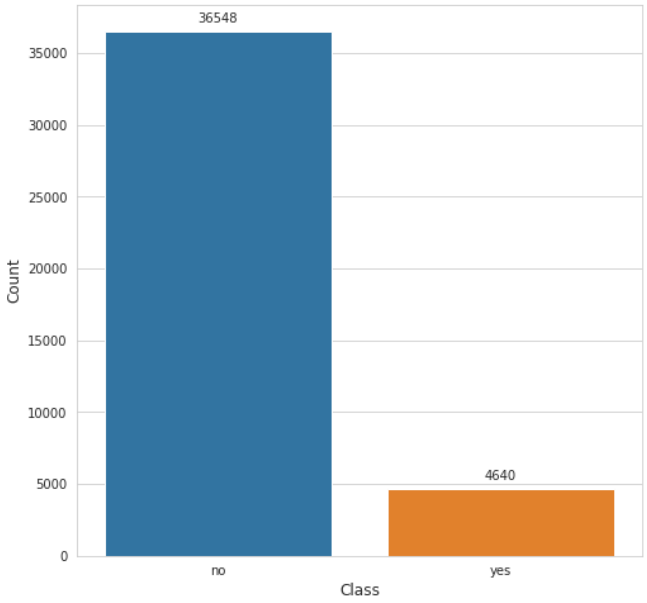

Figure 1: Class Distribution

As the figure shows, the data is skewed in favor of one class and this may cause problems in training. Machine learning classifiers, such as Random Forests, in fact, fail to cope with unbalanced training datasets as they are sensitive to the proportions of classes. Consequently, these algorithms tend to favor the class with the largest proportion of observations (majority class), which may lead to misleading accuracies. This may be particularly problematic when we are interested in the correct classification of the "rare" class (minority class) but we find high accuracies which are the product of the correct classification of the only majority class.

3.2 NAN / NULL

Another information to look for in data exploration is the existence of Nan / Null values since their presence is to be handled whether replacing them with a default value, replacing them with a custom value (i.e. most present class value, mean, median..) or dropping all the records containing them. While for this dataset is true that no invalid or missing values are present, on the other hand many features accept 'unknown' as value which sometimes may be the exact same thing. In this dataset there are 12.718 unknown values divided in 10.700 distinct records.

3.3 DOMAINS AND DISTRIBUTIONS

In this section of the analysis takes place an in-depth exploration of the features both as occurrence count, to show the exact composition of the dataset in terms of domain and distribution for each feature grouped by class, as well as visualizing the distributions as probability density functions separately for each class in order to be able to compare the two classes. It is important to point out, however, that in the case of this latest type of plot one might assume that for some feature values the probability of class 'yes' is higher than the one for class 'no' but that would be a mistake, the probability of 'no' is always higher for every feature, as shown by the occurrence plots. What this comparison tells us, in fact, is not the probability of 'yes' for a give feature value with respect to the probability of 'no' for the same feature value but, instead, it points out the segment of clients to target with the marketing campaign in order to maximize the 'yes' probability.

For the few numerical features in the dataset have been used boxplots, standardized method of displaying groups of numerical data based on a five-number summary (minimum, maximum, sample median, first and third quartiles) where the different parts of the box indicate the degree of dispersion (spread), skewness in the data and show outliers.

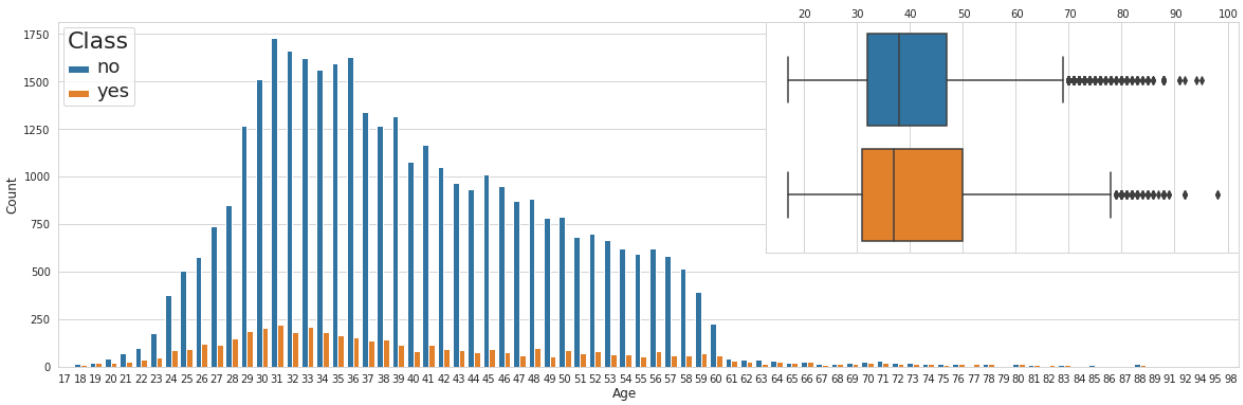

Figure2: Age Countplot & Boxplot

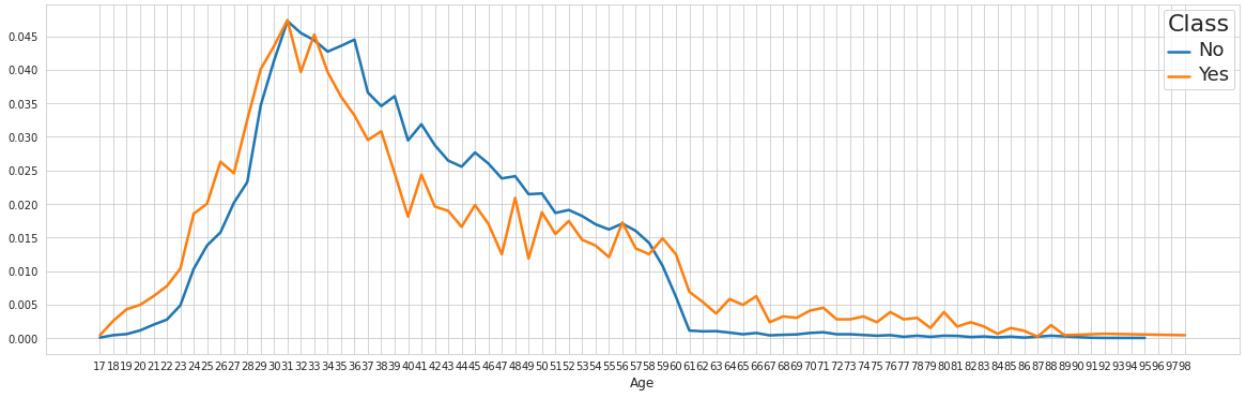

Figure3: Age Probability density function grouped by class

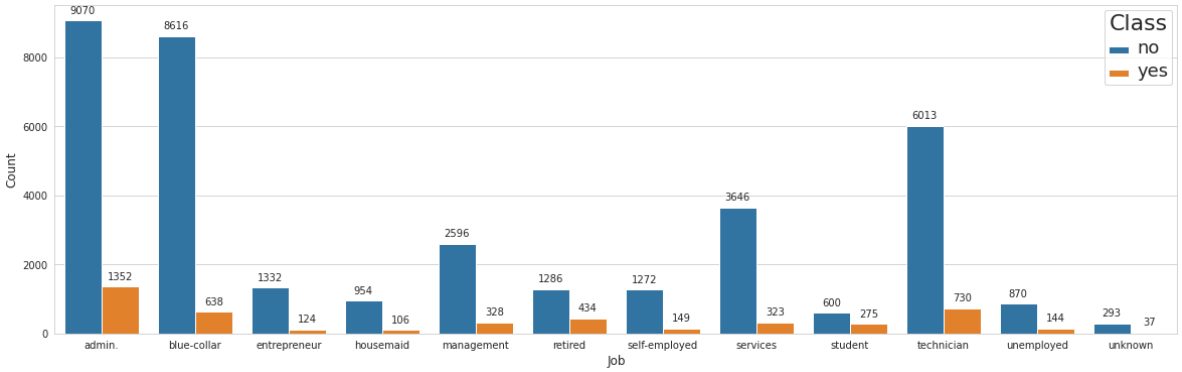

Figure 4: Job Countplot

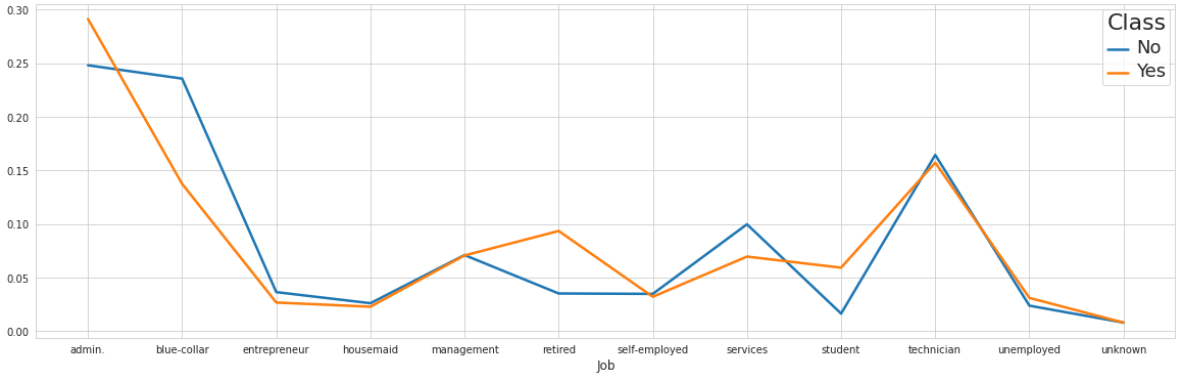

Figure 5: Job Probability density function grouped by class

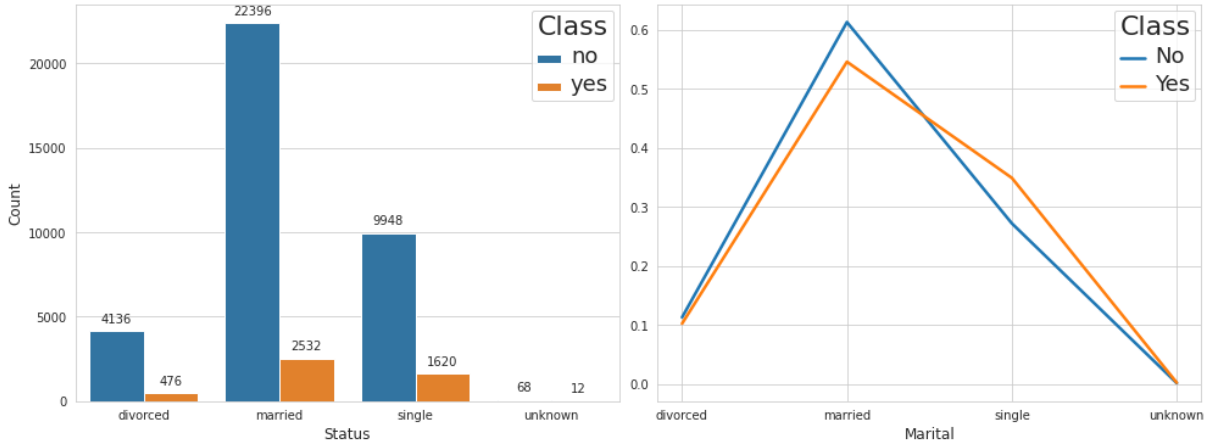

Figure 6: Status Countplot & Probability density function grouped by class

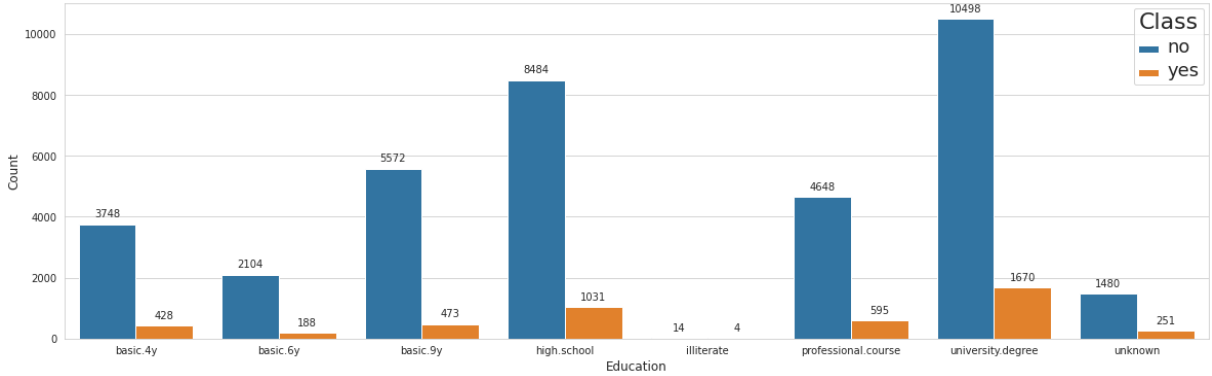

Figure 7: Education Countplot

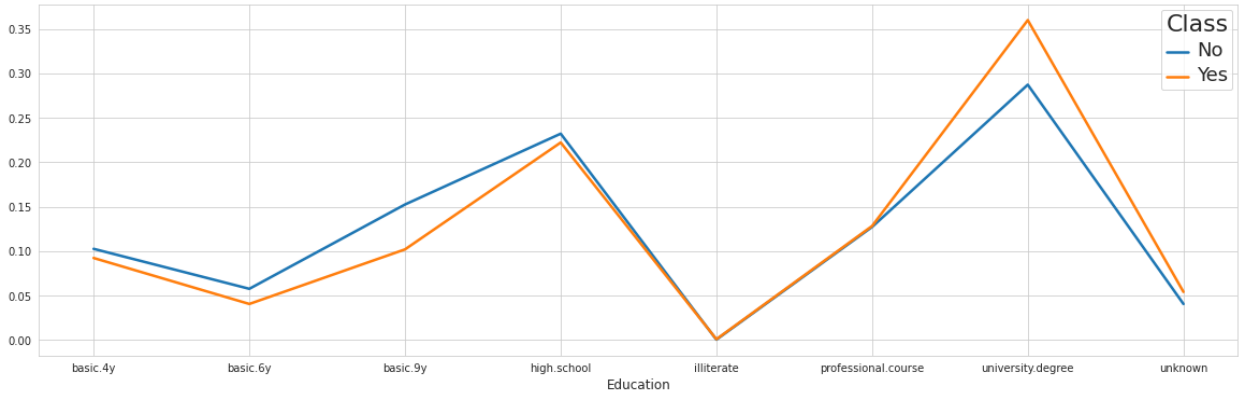

Figure 8: Education Probability density function grouped by class

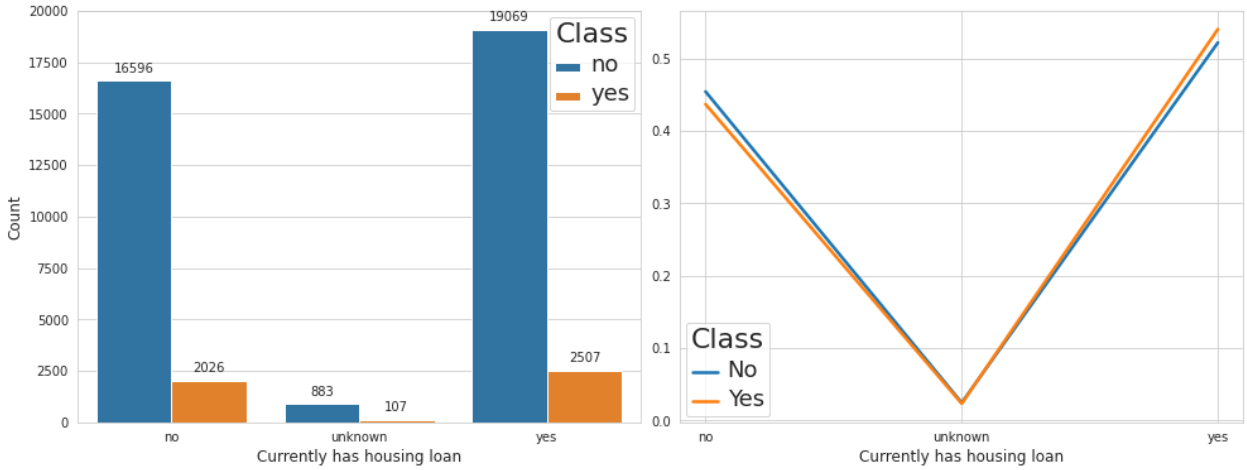

Figure 9: Housing Loan Countplot & Probability density function grouped by class

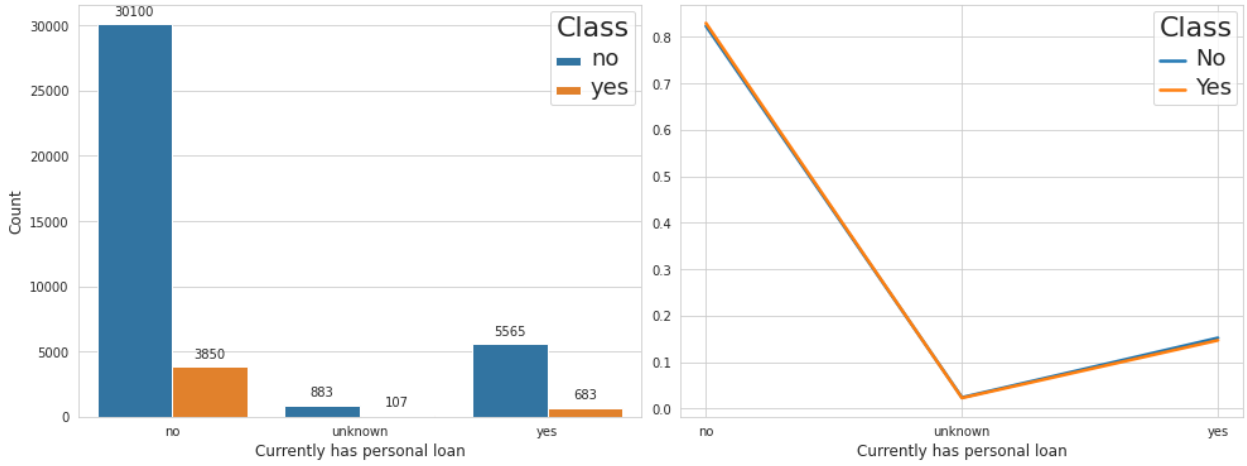

Figure 10: Personal Loan Countplot & Probability density function grouped by class

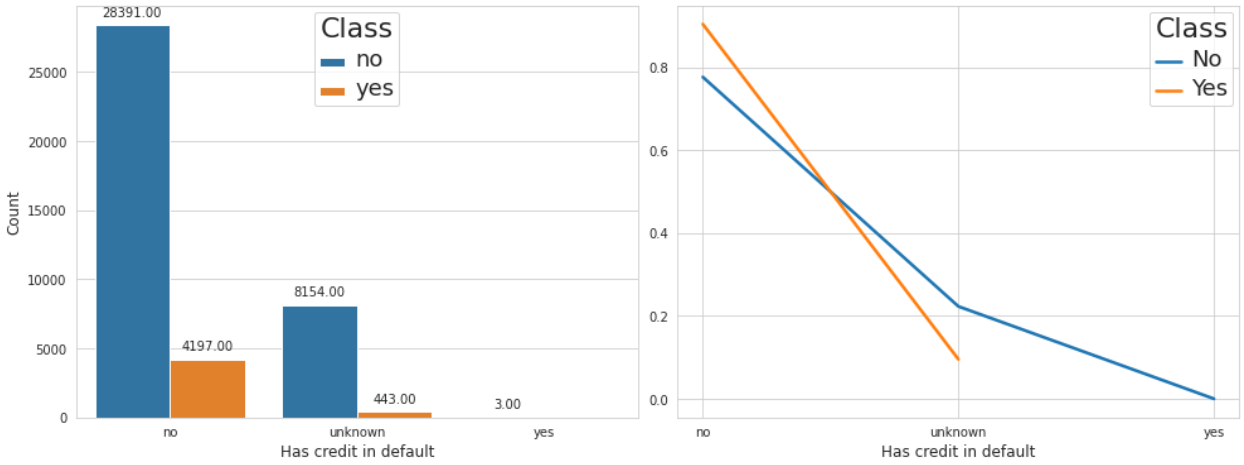

Figure 11: Credit in Default Countplot & Probability density function grouped by class

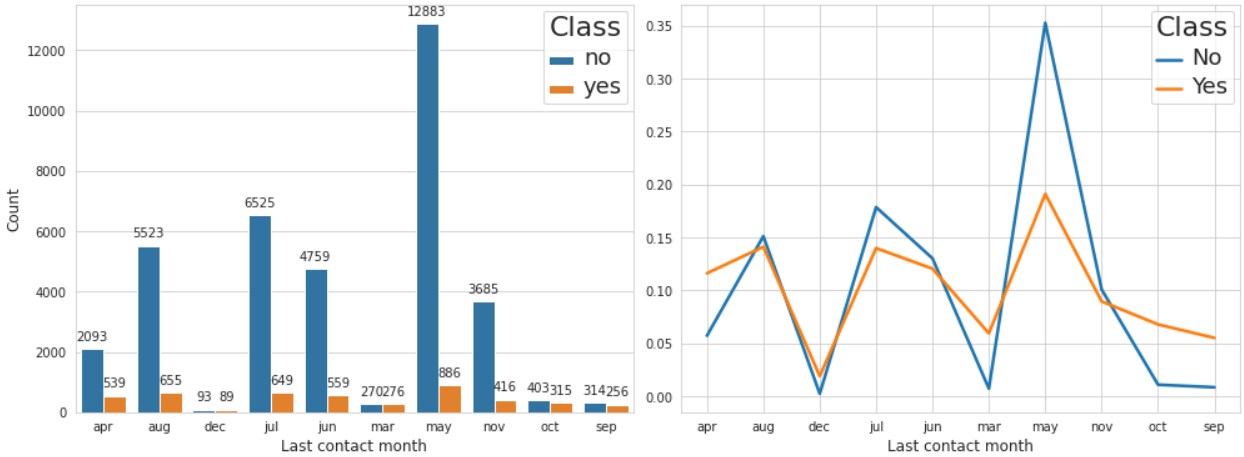

Figure 12: Contact Month Countplot & Probability density function grouped by class

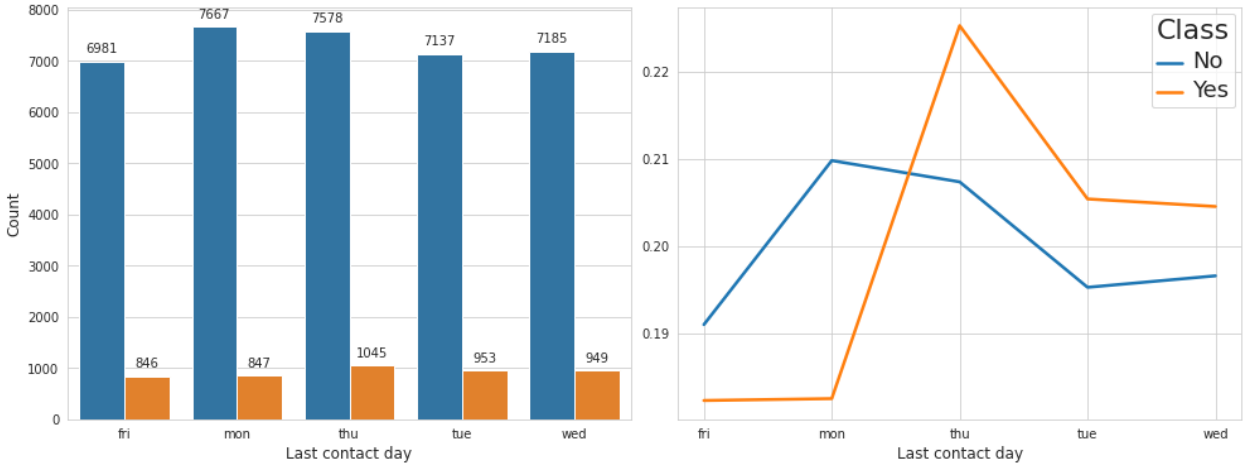

Figure 13: Contact Day Countplot & Probability density function grouped by class

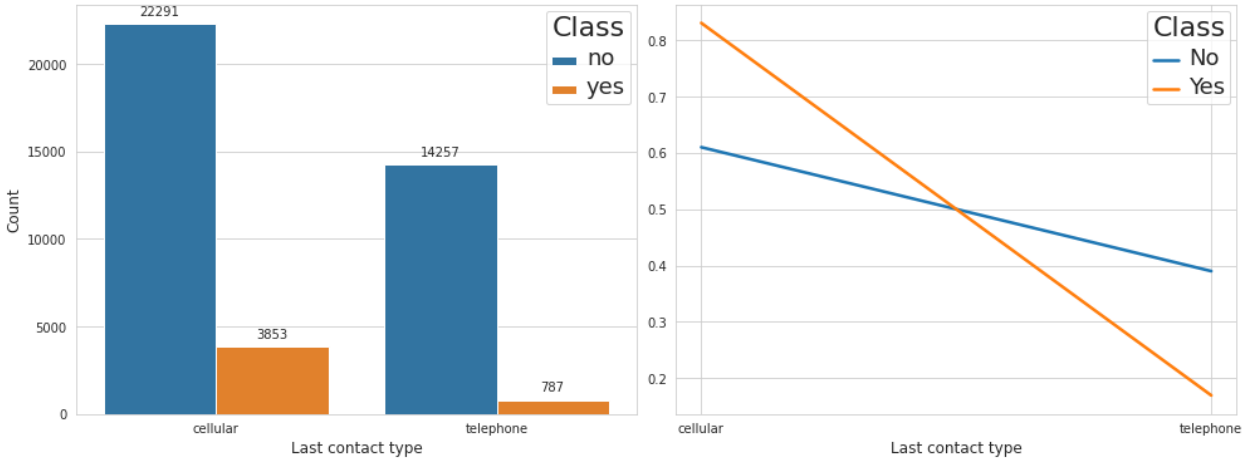

Figure 14: Contact Type Countplot & Probability density function grouped by class

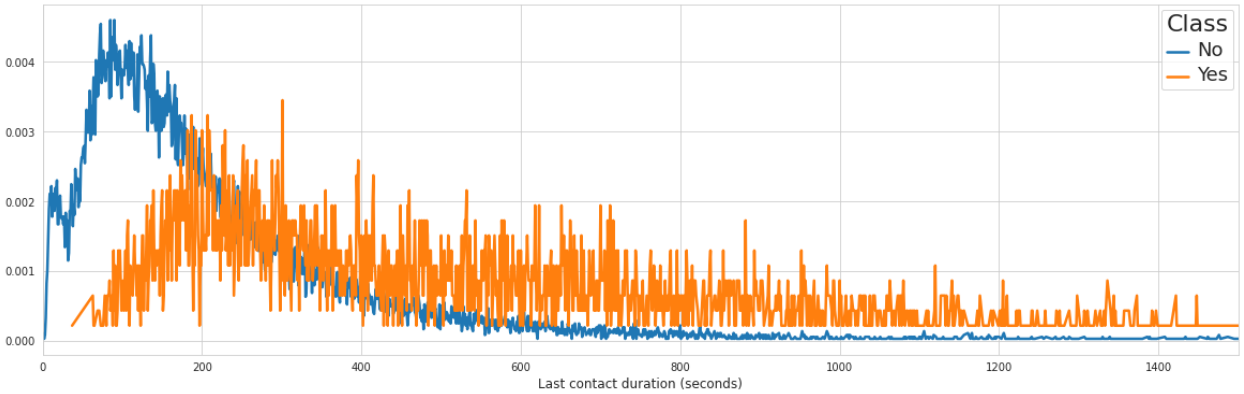

Figure15: Contact Duration Probability density function grouped by class

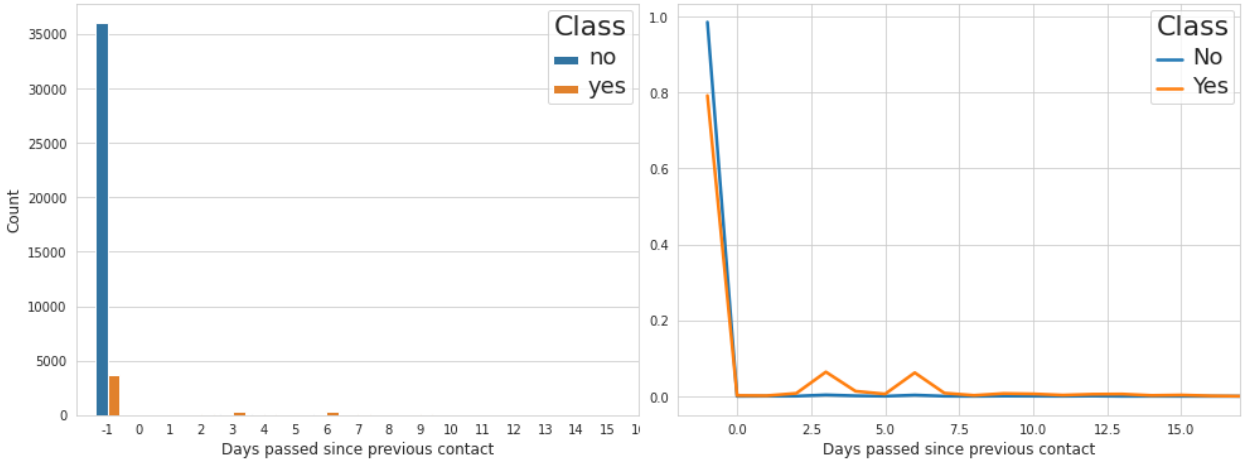

Figure 16: Days since previous contact Countplot & Probability density function grouped by class

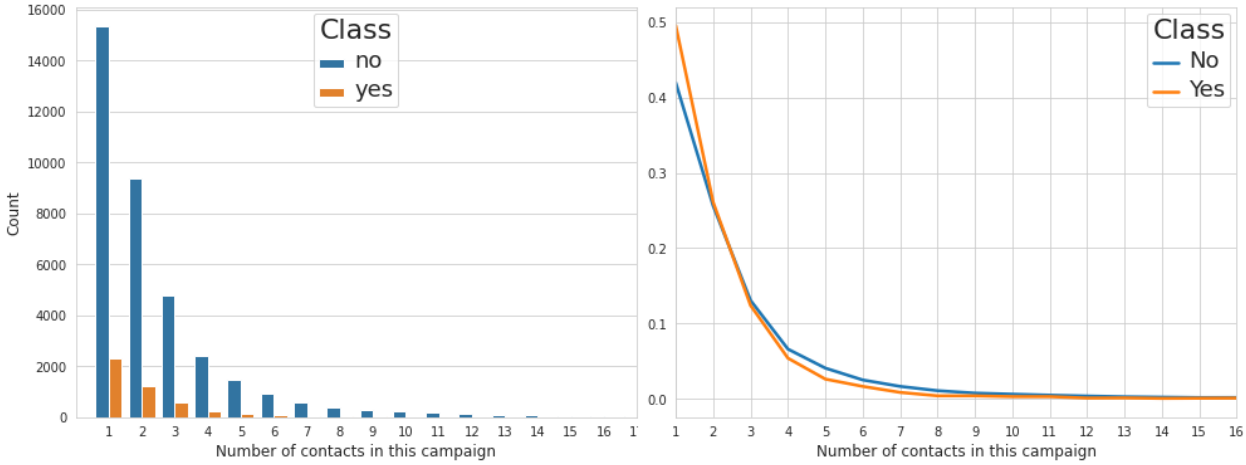

Figure 17: Number of current campaign's contacts Countplot & Probability density function grouped by class

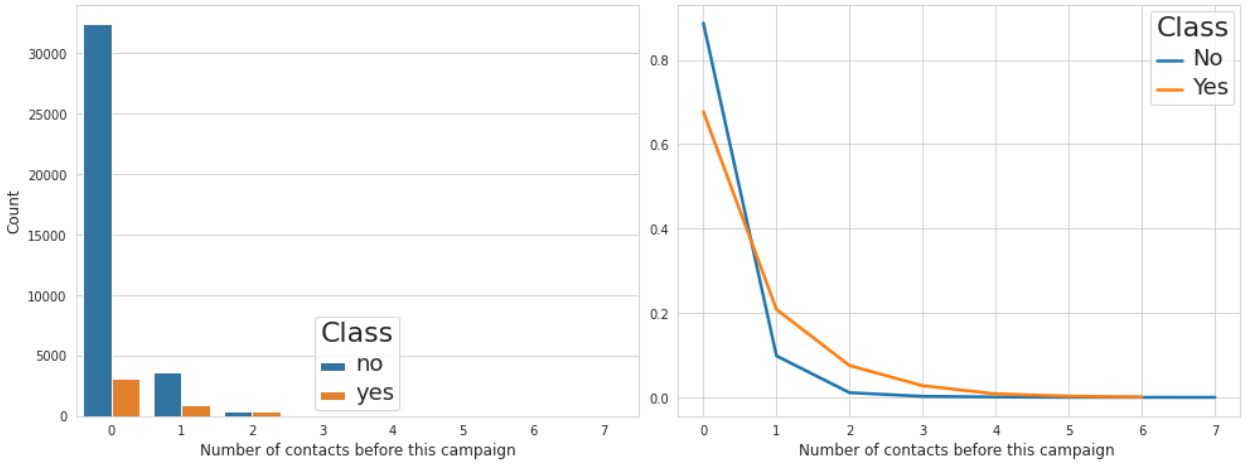

Figure 18: Number of previous campaign's contacts Countplot & Probability density function grouped by class

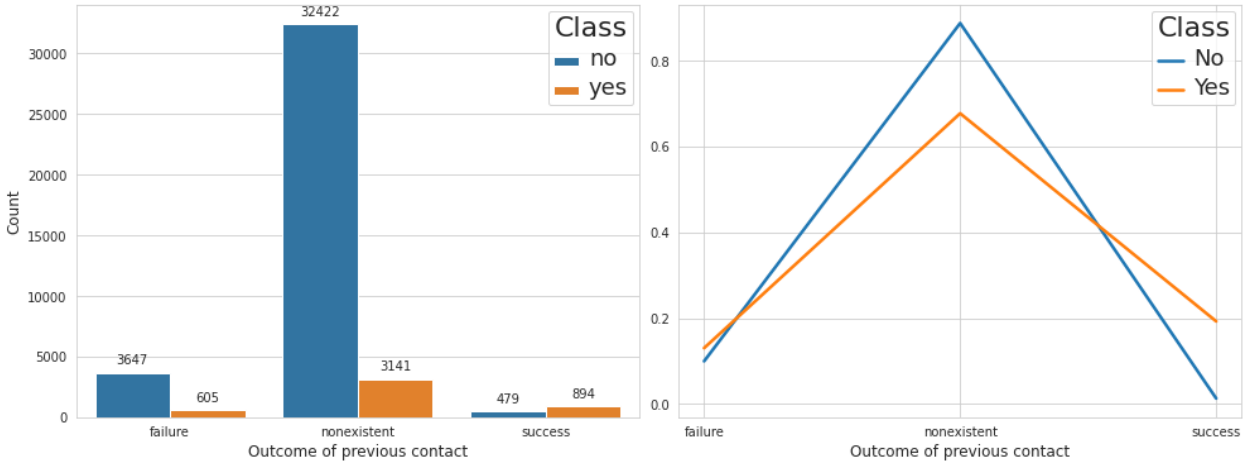

Figure 19: Last Contact Outcome Countplot & Probability density function grouped by class

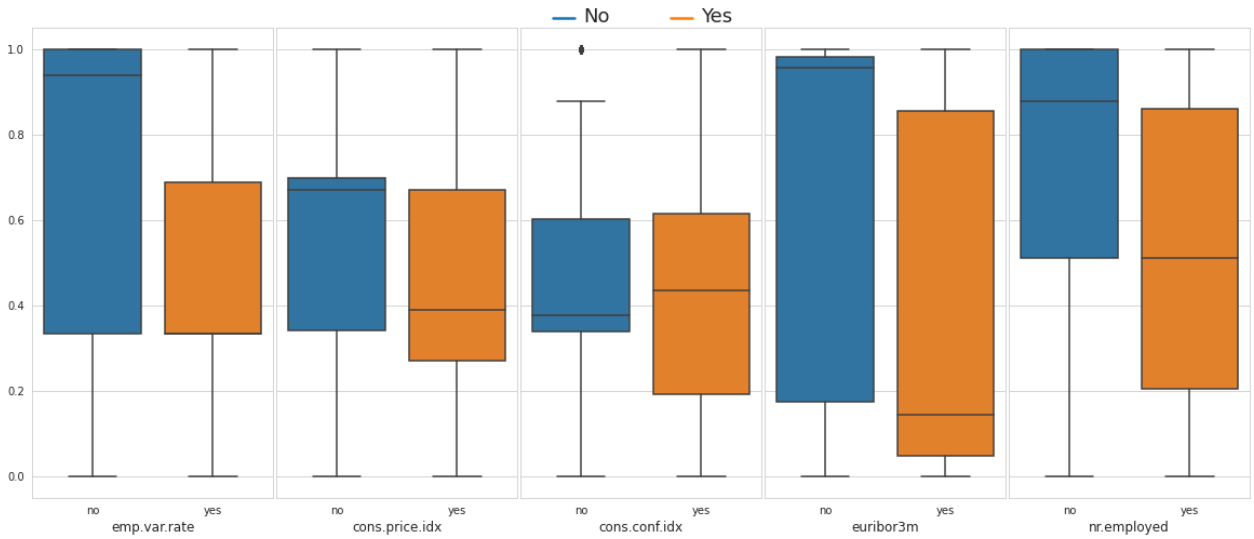

Figure 20: Normalized socio-economic indicators

The importance of data exploration is clear since, with the help of the plots, we are already able to distinguish some interesting characteristics of data both for an informed marketing decision making process (in case we wouldn't want to proceed with machine learning models) as well as behaviors to be exploited in data processing and consequent machine learning algorithms for a more complete and automated analysis.

We can, in fact, extract information on who to target (under 30 (student) or over 60(retired), with an high education level, 'job' is correlated with 'age' except for the insight of moderate success among administrators and finally clients with an active housing loan or with a previous successful contact), when (September and October during the days in the middle of the week) and how (cellular rather than telephone) to target clients in order to maximize the 'yes' rate. We can finally see how people are a little more likely to say 'yes' if they have been previously contacted at least once.

Finally, we can notice:

- No correlation between the class and the client having or not a personal loan.
- No usefulness of the knowledge about whether the credit of a given client is in default or not since the 'unknown' values are too many and the 'default = yes' are too few.

4. DATA PREPARATION

To feed the machine learning models, the second step consists in preparing data, that means we now have to go through selection, in order to maintain only relevant data, Nan values management, possible feature engineering, outliers management, categorical data encoding (not every model needs this but we will do it once for all), normalization, under/over-sampling, possible dimensionality reduction and division of the dataset in training, validation and test sets.

4.1 RELEVANT DATA SELECTION

To keep only relevant data, the first thing to do is to drop 'Duration' feature, in fact, it is true that this attribute highly affects the output target (e.g., if duration=0 then y='no') but, the duration is not known before a call is performed and, more importantly, the outcome (y) of the contact is known after the end of the call. Thus, this input should be discarded if the intention is to have a realistic predictive model.

Secondly, we are interested in dropping the features with doubtful relationship with the class variable that is the case of "personal loan" and "default". As stated before, indeed, the first of the two seems completely uncorrelated with the class variable, while the second gives no useful knowledge since the 'unknown' values are too many and 'yes' values too few.

4.2 VALUE MANAGEMENT

The next step consists in outliers removal (clients with age greater that a given value shown by figure 2) and the 'unknown' value management. As stated above, before data preparation, there were 12.718 unknown values divided in 10.700 distinct records, too high numbers to allow the deletion of such records, but now, after dropping a few precarious features and removing outliers we can notice that the number of 'unknown' has drastically decreased to only 393 divided between 370 records for the 'yes' class (8.12% of 'yes' records) and 2702 divided between 2540 records for the 'no' class (7% of 'no' records) due to the concentration of such value in 'default' feature. Given these numbers we can proceed with the removal of records with at least one 'unknown' value without feeling too guilty about wasting data.

4.3 ENCODING, PCA & PARTITIONING

At this point we are interested in numerically encoding all the categorical features in the dataset (machine learning models generally require all input and output variables to be numeric) and, if needed, we could perform dimensionality reduction. One way of doing so is Principal Component Analysis (PCA).

PCA is defined as an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by some scalar projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on. The main idea is therefore to seek the most accurate data representation in a lower dimensional space.

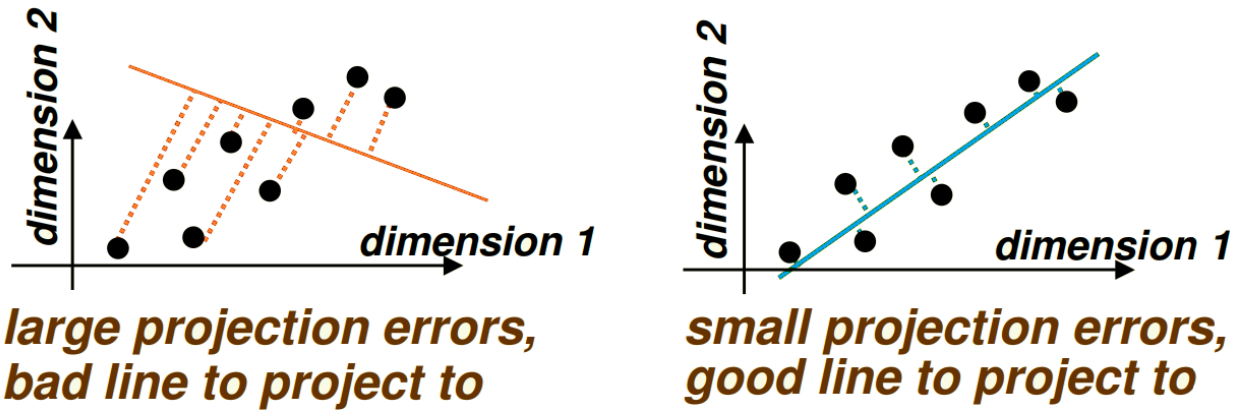

Figure 21: PCA finds the sub-dimension maximizing variance

It is fair to point out that PCA finds the most accurate data representation in a lower dimensional space, however, the directions of maximum variance may be useless for classification, we could therefore prefer Fisher Linear Discriminant since it projects data to a line preserving class separation useful for data classification (Data Representation vs. Data Classification).

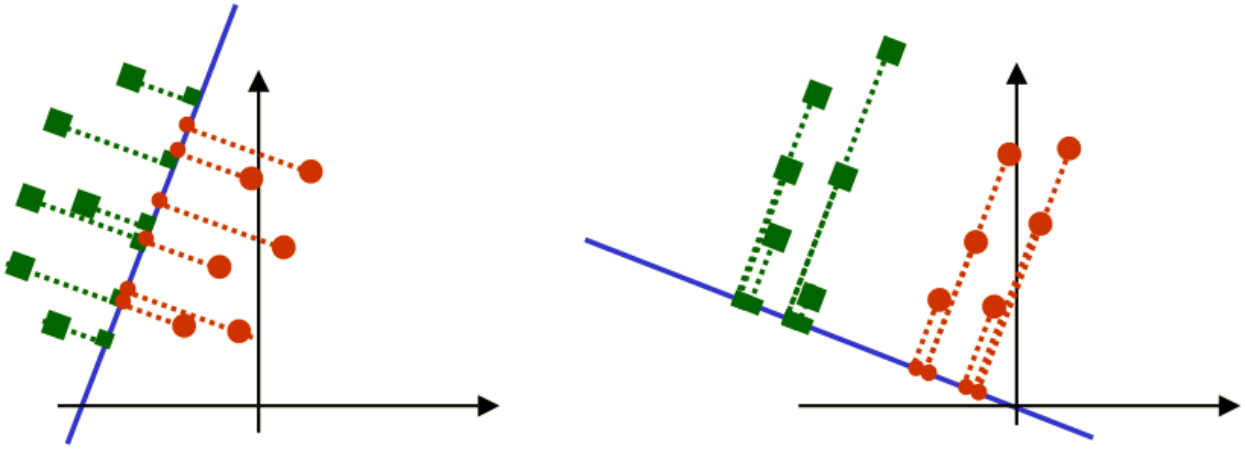

Figure 22: PCA vs LDA

Given this dataset we can avoid dimensionality reduction since the features are not so many and more importantly the number of instances is three orders of magnitude bigger than the number of features.

We can finally partition the dataset in two (training set + validation set and test set) maintaining the class proportionality.

5. SETTING & METRICS

Before going into model selection and consequent trainings and evaluations, it is important to understand how to handle class unbalance and specify the training setting, modalities and metrics used to evaluate the models.

5.1 CROSS VALIDATION

Stratified K-Fold Cross Validation has been used to perform hyperparameter tuning and to evaluate the models. This is a validation technique for assessing how the results of a statistical analysis will generalize to an independent data set, mainly used in settings where the goal is prediction, and one wants to estimate how accurately a predictive model will perform in practice.

According to this method, the original training + validation set is partitioned into k equal sized subsamples. Of the k subsamples, a single subsample is retained as the validation data for tuning the model, and the remaining $k - 1$ subsamples are used as training data. The cross-validation process is then repeated k times, with each of the k subsamples used exactly once as the validation data. The k results can then be averaged to produce a single estimation. The advantage of this method is that all observations are used for both training and validation, and each observation is used for validation exactly once.

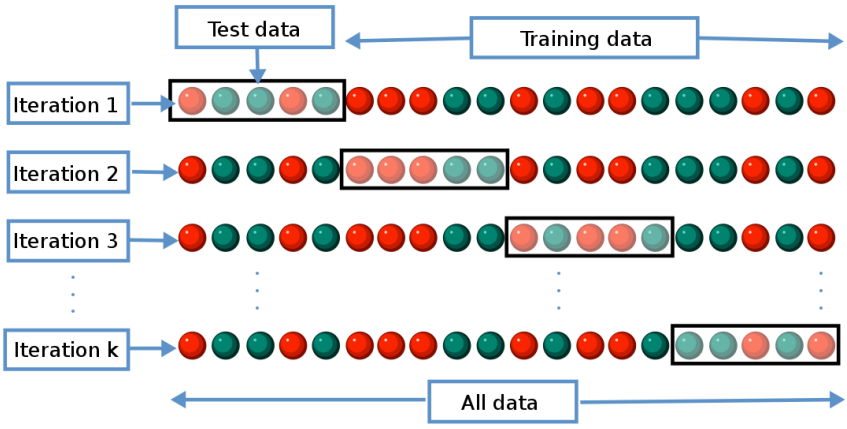

Figure 23: K-Fold Cross Validation

"Stratified" means the partitions are selected so that the mean response value is approximately equal in all the partitions. In the case of binary classification, this means that each partition contains roughly the same proportions of the two types of class labels, important since we want to validate on a subsample whose underlying distribution is as close as possible to the one of the test sample and finally as close as possible to the whole dataset's underlying distribution.

5.2 SMOTE OVERSAMPLING

The purpose of over-sampling and under-sampling is the creation of a balanced dataset, this is particularly important since many machine learning techniques make more reliable predictions from being trained with balanced data. With unbalanced data, in fact, the classifiers are more sensitive to detecting the majority class and less sensitive to the minority class leading to a biased classification output in many cases resulting in always predicting the majority class.

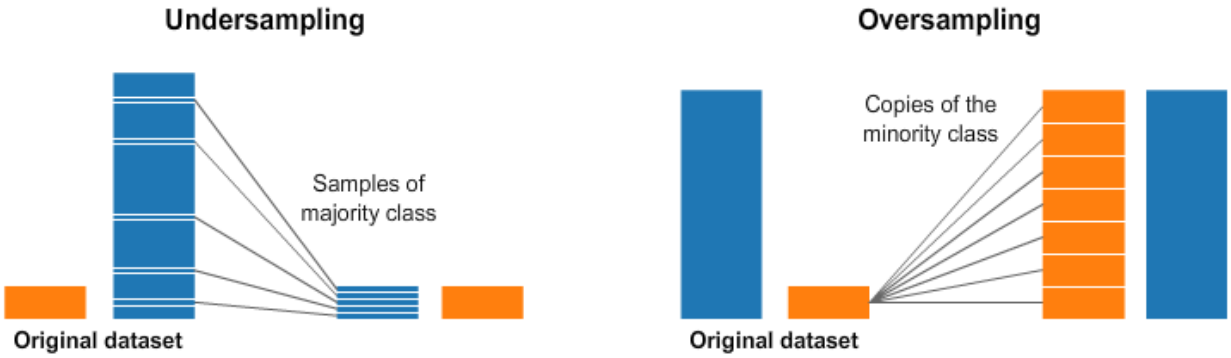

Figure 24: Effects of undersampling & oversampling

Since under-sampling, in the current situation would lead to a huge information loss, over-sampling seems a more advisable option (under-sampling is advisable only when the amount of data is so big, its processing constitute a too expensive computational cost). Among many techniques, one of the most commons is SMOTE: Synthetic Minority Over-sampling Technique, described by Nitesh Chawla, et al. in their 2002 paper [iii].

SMOTE first selects a minority class instance 'A' at random and finds its k nearest minority class neighbors. The synthetic instance is then created by choosing one of the k nearest neighbors 'B' at random and connecting 'A' and 'B' to form a line segment in the feature space. The synthetic instances are generated as a convex combination of the two chosen instances 'A' and 'B'.

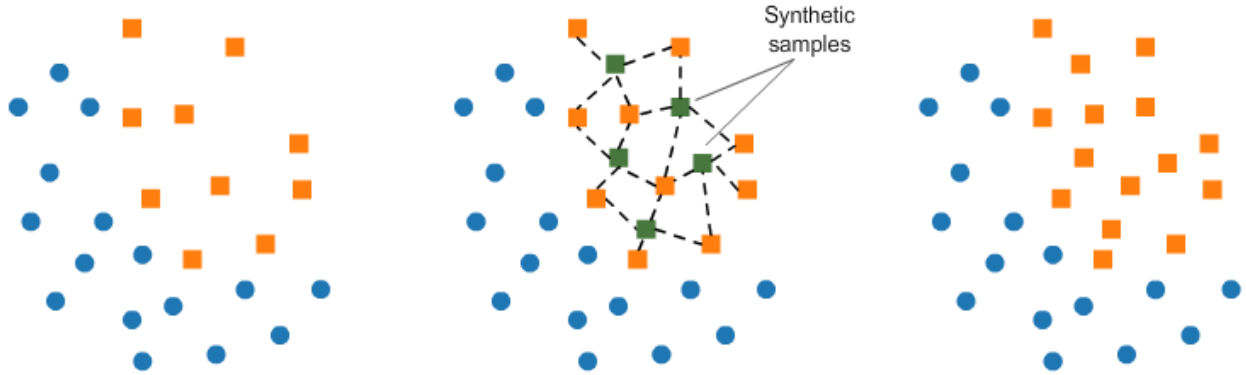

Figure 25: SMOTE: Synthetic Minority Over-sampling Technique

However, once decided between over and under-sampling and once selected the technique, a non-trivial and rather important thing to understand is when to perform the dataset balancing. Oversampling before cross validation, for instance, can lead to overfitting problems. As it is possible to see from figure 26, in fact, oversampling before cross validation, would make the model be trained on instances that are the same of the ones used for validating the model voiding the purpose of the validation phase. Therefore, it is important to oversample only the training data.

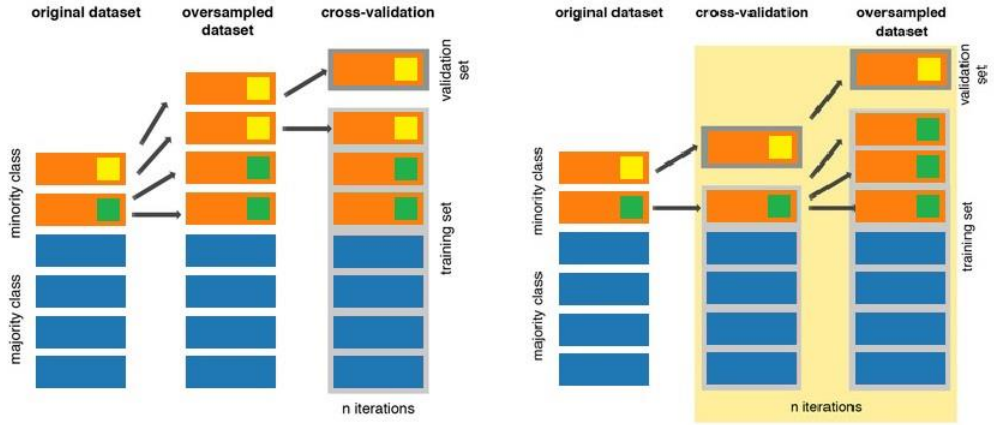

Figure 26: When to oversample

5.3 METRICS

A metric is a function that defines a distance between each pair of elements of a set (For classification problems, metrics involve comparing the expected class label to the predicted class label) and by doing so it gives us the knowledge of how well the model performs. Different metrics, however, measure different performance aspects that means the performance of a model with respect to a given metric does not tell us anything about how it performs with respect to another metric. Furthermore, choosing the wrong metric could lead to a poor model, or in the worst case, to be misled about its expected performance.

For this reason, two metrics has been taken into consideration:

- Accuracy: defined for a binary classification task as (TP + TN) / (TP + TN + FP + FN), measures the number of correct predictions over the total number of predictions.
  Even though this measure is rather important and probably the most famous and used, it does not tell the whole story. In the case of unbalanced data, as for example 99% vs 1%, predicting completely wrong the entire minority class would still lead to an impressive 99% accuracy score. This so called “Accuracy Paradox” is particularly dangerous when dealing with disease diagnosis where the disease is on one hand rare, but whose detection is of vital importance.
- Recall: defined for a binary classification task as (TP) / (TP + FN), used along with accuracy allows us to thoroughly understand the scenario by giving us information on how the model performs on the minority class, basically, filling the gaps left by the accuracy paradox. Recall measures, in fact, how many instances of the minority class has been correctly identified with respect to the total number of instances of that class.

6. MODEL SELECTION

6.1 LOGISTIC REGRESSION

Logistic Regression is a statistical learning technique as well as one of the simplest and most used Supervised Machine Learning algorithms for two-class classification based on probability, whose goal in its binary form is to train a classifier to make a binary decision about the class of a new input observation, that is, given a single input observation x, which can be represented by a feature vector $[x_1,\ x_2, \dots, x_n]$ (with 'n' number of features) and a target variable y that can be 1 or 0, we want to know the probability $\mathbb{P}(y = 1|x)$ that this observation is a member of class 1.

Logistic Regression does this by learning, from a training set, a vector of weights and a bias term. Each weight w_i is a real number associated with one of the input features x_i. The weight w_i represents how important that input feature is to the classification decision, and can be positive (meaning the feature is associated with the class) or negative (meaning the feature is not associated with the class).

The bias term b, also called 'intercept', is another real number added to the weighted inputs. To make a decision on a test instance, after we've learned the weights in training, the classifier first multiplies each x_i by its weight w_i, sums up the weighted features, and adds the bias term b. The resulting single number z expresses the weighted sum of the evidence for the class.

$$z = \left(\sum_{i=1}^{n} w_i\, x_i\right) + b = w \cdot x + b$$

At this moment, nothing forces z to be a legal probability (that is, to lie between 0 and 1), in fact, since weights are real values, the output might even be negative. To create a probability, z is passed through the sigmoid function, $\sigma(x)$, also called logistic function (giving Logistic Regression its name) with formula:

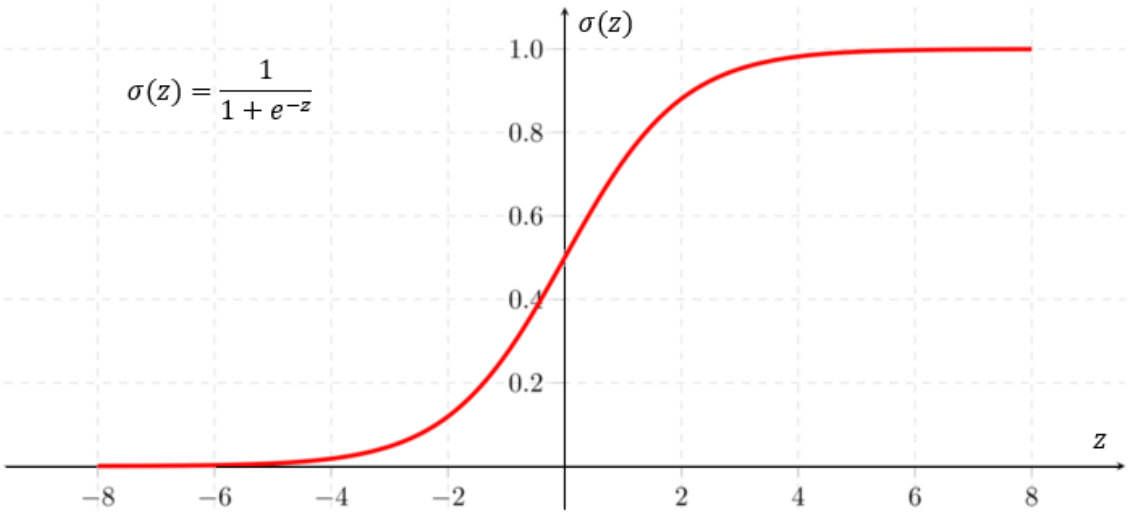

Figure 27: Logistic Function

Applying the sigmoid to the sum of the weighted features a number between 0 and 1 is obtained. To make it a probability, we just need to make sure that the two cases, $\mathbb{P}(y = 1)$ and $\mathbb{P}(y = 0)$ sum to 1. We can do this as follows:

$$\mathbb{P}(y = 1) = \sigma(w \cdot x + b) = \frac{1}{1 + e^{-(w \cdot x + b)}}$$

$$\mathbb{P}(y = 0) = 1 - \mathbb{P}(y = 1)$$

Now all that is missing for the probability computed by the algorithm to be transformed into a prediction is what is called 'decision boundary', in fact, Logistic Regression becomes a classification technique only when a decision threshold is brought into the picture:

$$\hat{y} = \begin{cases} 1 \; if \; \mathbb{P}(y = 1|x) > 0.5 \\ 0 \; if \; \mathbb{P}(y = 1|x) \leq 0.5 \end{cases}$$

All the combinations of the following hyperparameters have been evaluated in hyperparameter tuning:

- "penalty": ["l1", "l2"]
- "C": [1, 10, 100]
- "max_iter": [1000]

Best performing combination: C=1, max_iter=1000, penalty="l2"

6.2 DECISION TREE

Decision Tree is a supervised learning algorithm whose goal is to create a training model to predict the class or value of the target variable by learning simple decision rules (that are splitting rules used to segment the predictor space) inferred from training data.

In training phase, unfortunately, since it is computationally infeasible to consider every possible partition of the feature space, we take a top-down (splitting the predictor space from the top of the tree), greedy approach (at each step the local best split is performed, rather than the one that would lead to the best global result) that is known as recursive binary splitting. "Best split" is the split that minimizes entropy the most (or Gini index according to the chosen hyperparameter), considering all the features. The process continues until a stopping criterion is reached (for instance, we may continue until no region contains more than a give number of observations).

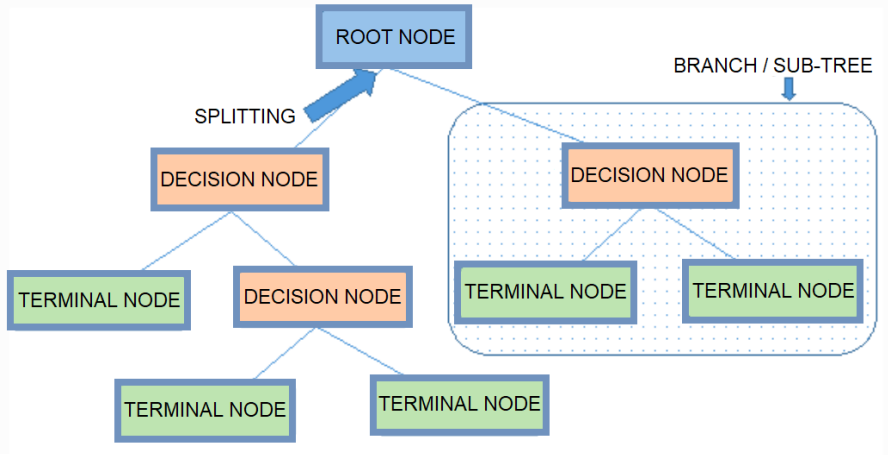

Figure 28: Decision Tree Structure

This process may produce good predictions on the training set but worse on a test set (overfitting), leading to a poor model. A possible solution is to use a smaller tree with fewer splits to lower variance and improve generalization, achievable by stopping the growth of the tree by avoiding splits that would lower the entropy no more than a threshold. Unfortunately, in this case, a seemingly worthless split and consequent stop of the growth early on, might be followed by a very good split that would not be performed.

In prediction phase each node in the tree acts as a test case for some feature, and each edge descending from the node corresponds to the possible answer to the test case. This process is recursive in nature and is repeated for every subtree rooted at the new node until a terminal node, associated with a given target variable, is reached.

Tree-based methods are simple, highly interpretable and can handle qualitative predictors without the need to create dummy variables. However, they typically are not competitive with the best supervised learning approaches in terms of prediction accuracy, this is why we introduce methods such as Random Forest, that consist in growing multiple trees then combined to produce a prediction based on majority voting. Combining many trees results in dramatic improvements in prediction accuracy, at the expense of some loss interpretation.

All the combinations of the following hyperparameters have been evaluated in hyperparameter tuning:

- "criterion": ["gini", "entropy"]
- "max_depth": [None, 15]
- "max_features": [None, "sqrt"]

Best performing combination: criterion="entropy", max_depth=15, max_features=None

6.3 RANDOM FOREST

Random Forests provide a huge improvement of the simple decision tree by bagging (building several decision trees on bootstrapped training samples) and by choosing a random subset of all predictors as split candidates from the full set of predictors. This decorrelates the trees from one another and reduces the variance when computing averages, resulting in more accurate and robust models. The final prediction is then reached by majority voting among all trees.

All the combinations of the following hyperparameters have been evaluated in hyperparameter tuning:

- "criterion": ["gini", "entropy"]
- "n_estimators": [100]
- "max_depth": [None, 15]
- "max_features": [None, "sqrt"]

Best performing combination: criterion="entropy", max_depth=15, max_features="sqrt", n_estimators=100

6.4 SUPPORT VECTOR MACHINE

Support Vector Machines (SVMs) are supervised learning models whose aim is to find the hyperplane that best separates data, that means the hyperplane with biggest margins from the closest data points of each class, since in general, the larger the margin, the lower the generalization error of the classifier. This hyperplane is the result of the optimization problem:

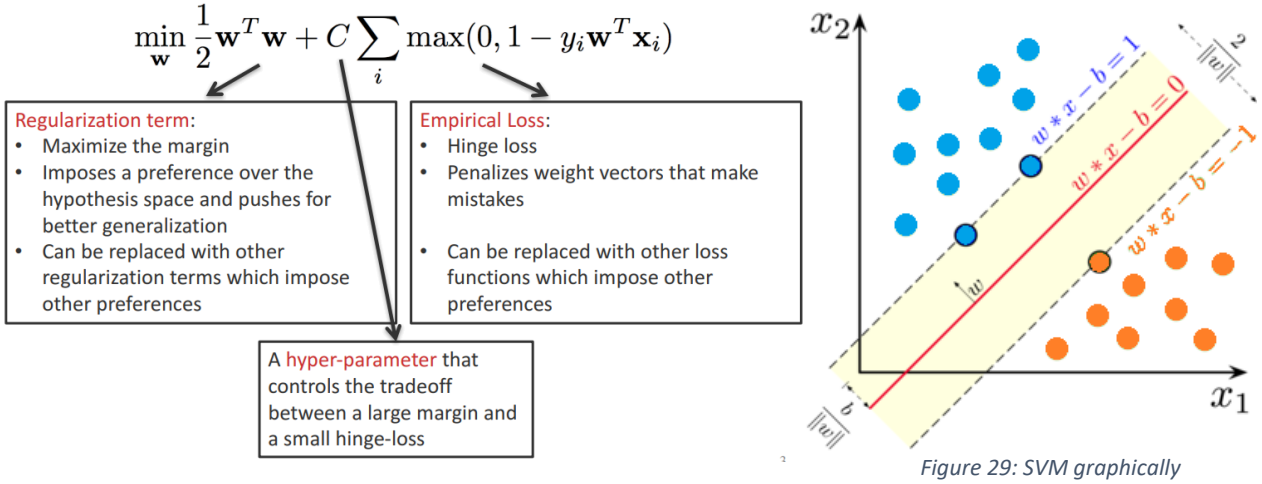

Figure 29: SVM graphically

It often happens that data are not linearly separable in their feature space, for this reason, it was proposed that the original space be mapped into a much higher-dimensional space, making the separation easier in that space, called kernel trick. To keep the computational load reasonable, the mappings used by SVM schemes are designed to ensure that dot products of pairs of input data

vectors may be computed easily in terms of the variables in the original space, by defining them in terms of a kernel function $k(x, y)$ selected to suit the problem.

The C parameter trades off correct classification of training examples against maximization of the decision function's margin. For larger values of C, a smaller margin will be accepted if the decision function is better at classifying all training points correctly. A lower C will encourage a larger margin, therefore a simpler decision function, at the cost of training accuracy. In other words, C behaves as a regularization parameter in the SVM.

Gamma parameter defines how far the influence of a single training example reaches, with low values meaning 'far' and high values meaning 'close'. The gamma parameters can be seen as the inverse of the radius of influence of samples selected by the model as support vectors.

All the combinations of the following hyperparameters have been evaluated in hyperparameter tuning:

- "C": [1, 50]
- "kernel": ["linear", "rbf"]
- "gamma": ["scale", "auto"]

Best performing combination: C=1, gamma="auto", kernel="rbf"

7. CONCLUSIONS

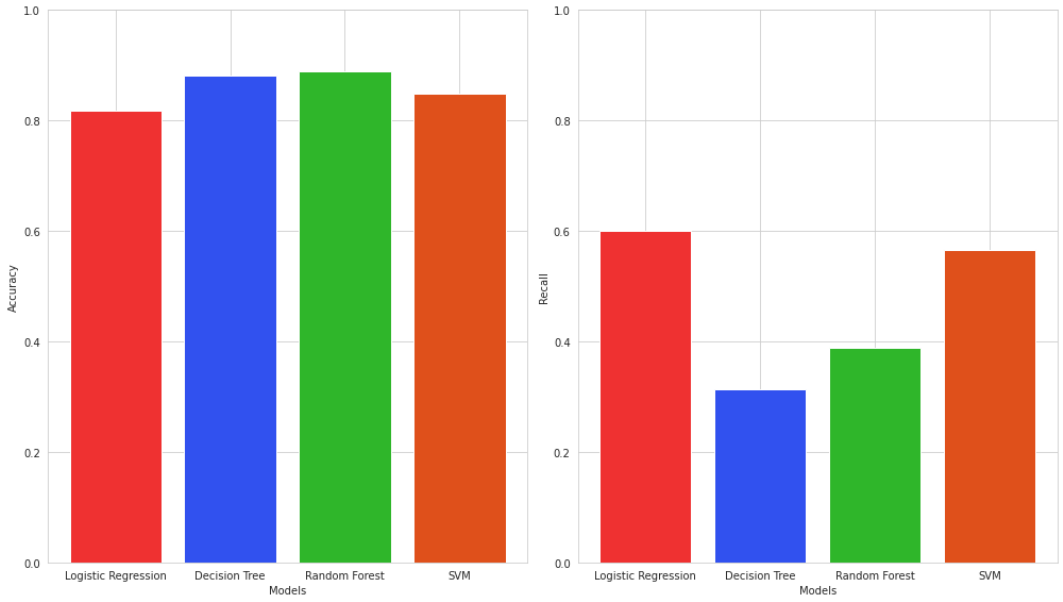

Figure 30: Accuracy and Recall Results

As we can see from figure 30 all the models have a pretty satisfying accuracy score ranging from the 81.8% of Logistic Regression to 88.9% of Random Forest, however, only Logistic Regression have an almost acceptable recall score with 60.1% of 'yes' class recognized among all the instances of that class. The first observation is that Logistic Regression is the best performing in recognizing the minority class and this is not really a surprise since its capabilities are known in the field of binary unbalanced datasets as "Fraud detection", "Credit card default" and "Cancer detection". A second observation is that recall score is overall low, all models considered, but this also does not come as a surprise due to the separation between the two classes being extremely subtle.

Within marketing, optimizing client targeting is a key issue, being a sector under a growing pressure to increase profits and reduce costs. In this context, the use of a decision support system based on a data driven models to make predictions is a valuable tool to support client selection decisions by business owners. In this study have been proposed machine learning techniques for data driven decision-making processes for the selection of bank telemarketing clients, a large Portuguese bank dataset of 41.188 records has been analyzed, four models compared (Logistic Regression, Decision Tree, Random Forest and Support Vector Machine (SVM)) and two metrics considered (Accuracy and Recall).

8. REFERENCES

[i] Data Exploration code available at:
https://github.com/albeffe/polytechnic_university_of_turin/blob/master/Mathematics%20in%20Machine%20Learning%202020/Mathematics_in_Machine_Learning_Thesis_(Part1_Visualization).ipynb

Data Preparation & Model Selection code available at:
https://github.com/albeffe/polytechnic_university_of_turin/blob/master/Mathematics%20in%20Machine%20Learning%202020/Mathematics_in_Machine_Learning_Thesis_(Part2_Preparation_%26_Models).ipynb

[ii] Dataset downloadable at: https://archive.ics.uci.edu/ml/machine-learning-databases/00222/bank-additional.zip

[iii] SMOTE paper: https://arxiv.org/pdf/1106.1813.pdf