Prediction of Bank Marketing Phone Campaign Success

Pau Comas and Odysseas Kyparissis[a]

[a] *UPC, Facultad de Informtica de Barcelona , Barcelona, November 2023, Master of Data Science, Spain*

Abstract

This paper presents the final project for the Advanced Machine Learning (AML) course in UPC's Data Science Master's program. It focuses on classifying bank marketing phone campaigns, specifically predicting whether the contacted client will subscribe (yes) or not (no). The project consists of data exploration, development of classification models, and draws meaningful conclusions for future research (3).

Keywords: Bank marketing campaign, Explanatory data analysis, Modeling, Binary classification, Support Vector Machines

1. Introduction

It is undeniable that Machine Learning (ML) and Artificial Intelligence (AI) technologies have revolutionized today's world, by affecting most of the important scientific fields that play a crucial role in society's evolution. One of those fields is Economics and Banking. By analyzing big amounts of data, ML and AI models can provide great support to banking professionals in analyzing the trends about their clients behavior, how to address marketing campaigns or retain as much of them as possible (2). For that reason, we decided to choose the Portuguese Bank Marketing Dataset, from the UCI Repository (3). The data is related to direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls, and the classification goal is to predict if the client will subscribe (yes/no) a term deposit (variable y) based on information about the client and the contact call.

2. Motivation and goals

The task of predicting the success of a marketing campaign presents a interesting challenge with many applications across various industries, including banking and finance. By effectively utilizing potential client data and diverse features to accurately forecast the campaign's outcome, valuable insights can be provided to stakeholders, enabling them to make strategic decisions and identify crucial factors.

In addition, addressing the classification problem of predicting campaign success can provide profound insights into the underlying elements that contribute to a positive or negative response from customers, and create new opportunities for real-life businesses. This new knowledge has the potential to assist marketing teams in optimizing their strategies by identifying areas that require improvement or focusing on characteristics that resonate positively with the target audience.

Our goal is to predict the campaign outcome based on customer data and features, such as previous interaction history or demographic information. More than deeply understanding the features' effect on the marketing success, we're mainly interesting in obtaining insights that would lead to a targeted or more agressive marketing to X group of potential clients, to save costs and enhance our strategies as an hypothetical bank.

3. Data available

The data from this study was obtained from UC Irvine Machine Learning Repository (3), imported and downloaded to the project's Github repository (1). There are different versions of the dataset available, from which we chose "bank-full", with all 45211 examples and 17 inputs, ordered by date (although we do not take into account the temporal dependencies). This dataset is licensed under a Creative Commons Attribution 4.0 International (CC BY 4.0) license, which allows for the sharing and adaptation of the datasets for any purpose, provided that the appropriate credit is given.

4. Related previous work

An introductory paper for the dataset was presented in 2014, called "A data-driven approach to predict the success of bank telemarketing" (4). In this publication, the analysis spans the period from 2008 to 2013, encompassing the aftermath of the recent financial crisis. With a analysis of 150 features related to bank clients, products, and socio-economic attributes, the study employed a feature selection method, ultimately narrowing down to a focused set of 22 features selected before July 2012. It compares four data mining models: logistic regression, decision trees, neural network, and support vector machine. The neural network emerged as the most effective, exhibiting an AUC of 0.8 and ALIFT of 0.7. This model successfully reached 79 % of subscribers by targeting half of the better-classified clients. It is important to note that this seminal work employed an extended dataset with temporal variables, including meaningful indicators that we don't have such as Euribor, distinguishing it from the dataset available for the current investigation.

5. Explanatory Data Analysis

In this section of the report, an initial description of the dataset is provided, followed by the implementation of exploratory analysis, including both univariate and bivariate analyses, as well as the outlier detection procedure. Finally, the section concludes with the feature selection and extraction process. Through these exploratory steps, our goal is to uncover valuable insights and establish a solid foundation for subsequent analyses.

5.1. Description of the dataset

The dataset initially comprises 45211 observations and 17 columns (*variables*). The target value, denoted as the 17th variable (y), represents the *Subscription Status* of the client and is binary (**yes** or **no**). In accordance with the dataset's metadata, there are 17 variables in total, including 7 numerical and 11 categorical variables (Tables 1, 2, 3, 4). Notably, null values (missing values) are present in the dataset, specifically for the variables: *job*, *education*, *contact*, and *poutcome*. Prior to addressing the missing values, it is essential to examine the distribution of numerical variables and the categories of categorical variables to gain a comprehensive understanding of their content and nature, including the characteristics of any missing values.

Table 1: Numerical variables

Variable	Description
age	Age of the client
balance	Average yearly balance in euros
campaign	Number of contacts during this campaign
duration	Last contact duration, in seconds
pdays	Days since the client was last contacted
previous	Number of contacts before this campaign
poutcome	Outcome of the previous marketing campaign

Table 2: Date variables

Variable	Description
day	Last contact day of the week
month	Last contact month of the year

Table 3: Categorical variables

Variable	Description
job	Occupation type of the client
marital	Marital status of the client
education	Education level of the client
default	Client's credit default status
housing	Presence of a housing loan
loan	Presence of a personal loan
contact	Contact communication type
y	Subscription status (target variable)

The brief analysis for the description of the data reveals that numerical variables do not exhibit abnormally high maximums

Table 4: Categories of date and categorical Variables

Variable	Categories
job	management, technician, entrepreneur, blue-collar, Unknown, retired, admin., services, self-employed, unemployed, housemaid, student
marital	married, single, divorced
education	tertiary, secondary, Unknown, primary
default	no, yes
housing	yes, no
loan	no, yes
contact	Unknown, cellular, telephone
month	may, jun, jul, aug, oct, nov, dec, jan, feb, mar, apr, sep
y	no, yes

(e.g., 9999999), suggesting a lack of errors in these variables. Special cases are observed for the variables *balance* and *pdays*. The minimum value of balance (-8019) may indicate an erroneous case, to be further analyzed. Additionally, for variable *pdays*, a minimum value of -1 indicates cases where the client was never contacted before. Outliers are also identified; for example, in the variable *duration*, a value of 4918 seconds stands out, implying an unusually long call duration of around 82 minutes. Outlier analysis is conducted in subsequent sections.

Furthermore, the variable *duration*, measured in seconds, significantly influences the output target (y). However, since the duration is not known before a call and the target output becomes evident only after the call's conclusion, it has been decided to exclude this variable for realistic predictive modeling purposes. Additionally, three instances where the *duration* of a call is equal to zero have been observed. It is proposed to first remove all observations with a duration equal to zero because retaining those observations would introduce bias (since $duration = 0 \Rightarrow y = 0$), and subsequently, the entire duration column should be dropped.

On the other hand, categorical variables reveal some erroneous categories. For variables with missing values (*job*, *education*, *contact*, *poutcome*), imputation techniques are considered. If the percentage of erroneous or missing data for a specific variable is substantial (e.g., 70%), excluding it from the analysis is recommended. For that reason, the *poutcome* column is dropped due to missing values exceeding 80% of the dataset's rows. For the *contact* variable, missing values (29% of the dataset) will be replaced with a new category named *Unknown*. Finally, imputation is performed for the remaining variables in subsequent sections.

After dropping the mentioned variables, the dataset is left with 15 variables — 14 explanatory variables and a target variable (y). Of these, 5 are numerical, 2 are date variables, and 8 are categorical. The target variable y is categorical as well. Next, univariate analysis is presented.

5.2. Univariate analysis

To initiate the analysis, the column names of numerical and categorical variables are separated into two distinct lists. With

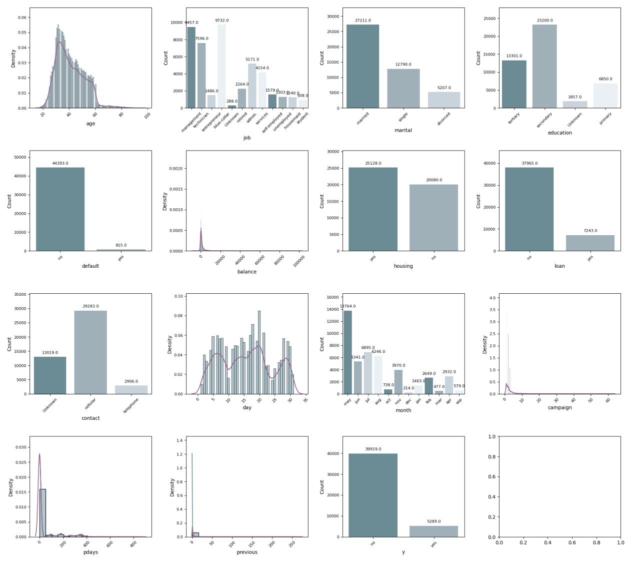
Figure 1: Univariate analysis

the variables separated, a combined plot (Figure 1) is created, incorporating histograms for numerical variables and countplots for categorical variables. This visualization provides an initial overview of the distribution of the available variables.

To begin with, from Figure 1, for the categorical variables (Tables 2, 3, 4) the diverse distribution of *jobs* is noteworthy, with *management* and *technicians* exhibiting high counts, while certain categories like students and housemaids have lower representation. *Marital status* leans heavily towards married clients, creating a substantial imbalance. *Education*-wise, the majority hold *secondary education*, with *tertiary* and *primary* levels having comparatively lower counts. The credit *default* variable indicates a significant majority without defaults. *Housing status* is relatively balanced, with more clients having housing loans. *Personal loans*, on the other hand, exhibit an imbalance, with a higher count of clients having loans. *Communication* type presents a considerable imbalance, with the *Unknown* category having the highest count. Analysis of the *last contact month* shows fluctuations, with a notable increase in call volumes between **May** and **August**. The target variable (y) also displays an unbalanced distribution, with a predominant count of ***no*** cases.

Moreover, as for the numerical variables (Table 1), *age* distribution reveals a relatively even spread across various age groups, with a mean around 40 years. The average yearly *balance* distribution is right-skewed, indicating potential outliers. Additionally, the distribution of the last contact day (*day*) shows a fair spread with some autocorrelation, suggesting repetitive patterns between weekdays. The number of contacts during this campaign (*campaign*) and the number of days since the last contact (*pdays*) both exhibit right-skewed distributions, necessitating careful treatment for high values. Similar patterns are observed for the number of contacts performed before this campaign (*previous*). To address these observations and prevent potential bias in learning algorithms, a univariate outliers analysis (5.2.2) follows this section.

5.2.1. Shapiro normality tests

Before proceeding, a *Shapiro-Wilk* normality test is performed on numerical values. The results indicate that none of the numerical variables follow a normal distribution. To mitigate this, transformations of the numerical variables are considered. While examining the numerical variables, the *balance* variable, through a logarithmic transformation, exhibits normality, while the logarithms of the variables *campaign* and *previous* seem to follow a power-law distribution. Figure 2 confirm the efficacy of these transformations. Consequently, the addition of the logarithmic version of such variables as new features in the dataset is completed in section 5.3.3.

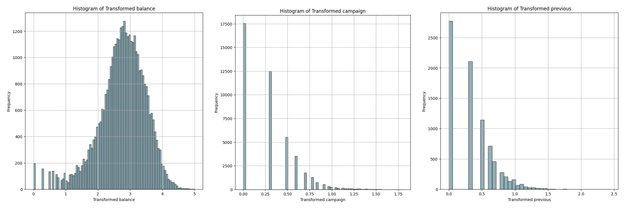

Figure 2: Logarithmic transformations of numerical variables.

Further analysis, including outlier detection and feature extraction, precedes dataset preprocessing and normalization. This step is crucial for verifying data accuracy and enabling separate treatment for training and test data in subsequent sections.

5.2.2. Univariate outliers analysis

Outliers are observations that lie far away from the majority of the data points in a dataset, often indicating unusual or exceptional behavior. They can potentially distort statistical analysis and should be carefully examined to determine whether they are due to measurement errors, rare events, or meaningful phenomena. Thus it is very important to treat them carefully, first by identifying them and finally by imputing or removing their values, or just by separating them from the remaining observations, in order to treat them separately.

Figure 3 presents an example for the *age* variable. The purple and blue dotted lines indicate the $Q1 \pm 1.5 * IQR$ and $Q3 \pm 3 * IQR$, respectively, following the interquartile range (IQR[1]) approach. It is important to mention that only extreme outliers are treated in the current implementation. For the remaining numerical variables, the results are depicted in *01.EDA-feature-extraction.html*[2].

For the *age* variable, three extreme outliers occur, describing elderly individuals. These outliers are retained in the analysis, as they are only three and they do not introduce high bias to the model. Moving on to the variable *balance*, a total of 4728 outliers are identified, with 2442 considered extreme outliers. Considering that the *balance* variable can take negative values for clients in debt, no cases are dropped. However, as the dataset's

[1]`https://en.wikipedia.org/wiki/Interquartile_range`

[2]`https://github.com/odyskypa/aml-bank-marketing-campaign/blob/main/notebooks/01.EDA-feature-extraction.html`

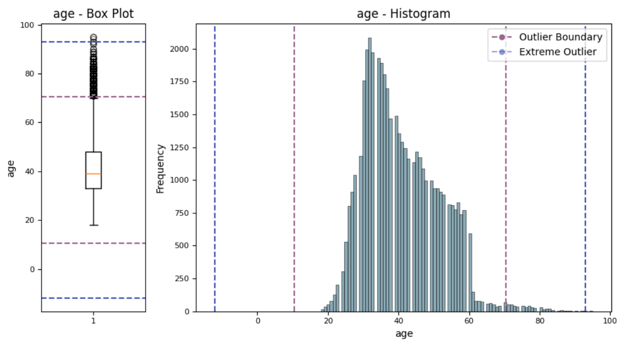

Figure 3: Outliers analysis of variable *age*.

metadata lacks information about the variable's value range, no further action is taken. Subsequently, the analysis shifts to the *day* variable, where no outliers are identified as all values fall within the range of integer values $1 - 31$. The variable *campaign* is then analyzed, revealing 3063 outliers in total, with 1461 considered extreme outliers. Upon scrutinizing statistics and specific cases of both normal and extreme outliers for the *campaign* variable, it is observed that several instances involve an exceptionally high number of calls performed for specific clients during this campaign. Extreme outliers, indicating more than 10 calls, are therefore discarded. Moving on to the variable *pdays*, a total of 8223 outliers are identified, and all are considered as extreme outliers. Given the lack of reasonability in this result, a more detailed examination is required. Leading to the fact that When *pdays* is equal to -1, it signifies that the client was never contacted before. Analysis reveals that this holds true for 35524 cases in the dataset. Further examination of cases where clients were contacted before indicates that the number of clients contacted matches the number of outliers. Consequently, excluding clients with a value of -1 for this variable, the outliers analysis is conducted again. However, no clients are discarded as they do not qualify as extreme outliers. Finally, the variable *previous* is analyzed, showing a similar pattern as observed in the *campaign* variable. Following the same methodology, all observations for clients contacted more than 13 times in previous campaigns are discarded. The rationale behind this decision is to train a model capable of predicting client subscription without excessive calls. Thus, 144 observations are discarded, resulting in a final dataset shape of $(43621, 15)$.

5.2.3. *Imputation of missing values*

Before proceeding with the exploratory bivariate analysis (section 5.3), the imputation of missing values is firstly performed. Imputation is a crucial step in machine learning projects, addressing the absence of values in certain variables to ensure a comprehensive and accurate analysis.

In the exploration of missing values, three key categorical variables demand attention: *job*, *education*, and *contact*. For the variable *job*, only 288 instances, comprising less than 1% of the dataset, exhibit missing values. A random imputation method, weighted by the proportion of each job category, is applied to maintain the integrity of the data. Similarly, for *education*, which includes 1857 missing values, accounting for less than 1% of instances, the same random imputation strategy is employed. In this case, the imputation is executed with a weighting proportional to each education level. Lastly, for the variable *contact*, acknowledging that the *Unknown* category already captures over 30% of missing values, it is retained as a new category during the imputation process. Consequently, after this step of the pipeline, the dataset no longer includes any missing values.

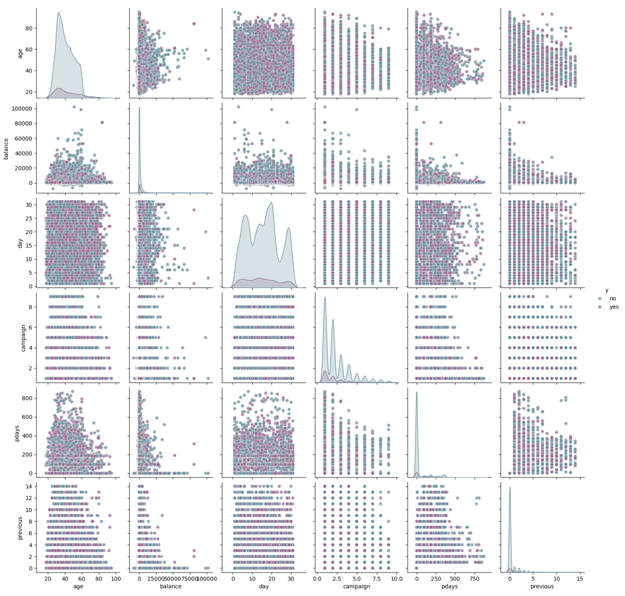

Figure 4: Bivariate scatterplot analysis with respect to *y*.

5.3. *Bivariate exploratory analysis*

In this section, a deeper look is taken into the relationship between the pairs of available variables with respect to the target variable (y). Firstly, the relations between the numerical variables with respect to the target variable is depicted in Figure 4.

At first sight, by taking a look at the scatter plot of the numerical variables, one can understand that the observations of *yes* and *no* classes are mixed when a combination of two numerical variables occurs.This means that there is no clear separation between a pair of numerical variables for distinguishing *yes* and *no* cases. For that reason, a conclusion from the scatterplot is that the decision boundary between the two classes is likely to be complex and nonlinear. Simply relying on a linear separation based on individual pairs of numerical variables may not be sufficient for accurate classification. Further analysis, involving feature engineering or the application of advanced machine learning techniques, is necessary to capture the intricate patterns and relationships within the data for a more effective classification model.

Moreover, a look at the correlation[3] heatmap (Figure 5) of the numerical variables is taken at this point. The result of the

[3]https://en.wikipedia.org/wiki/Correlation

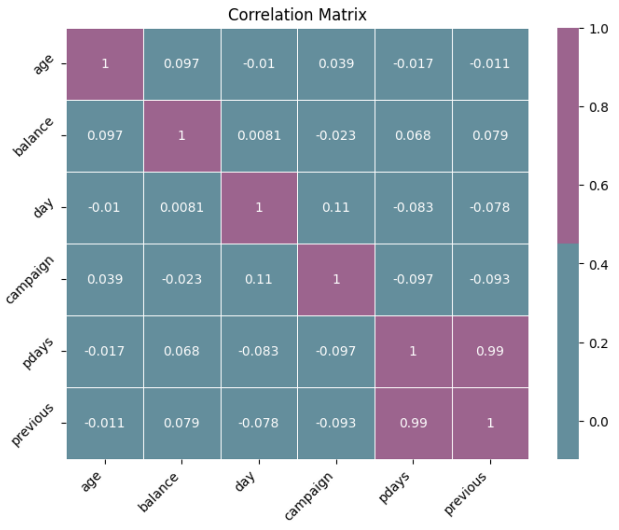

Figure 5: Spearman correlation heatmap of numerical features.

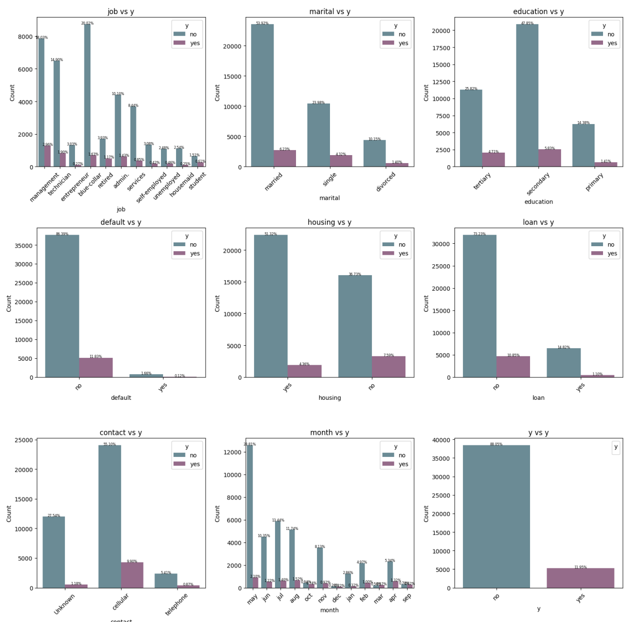
Figure 6: Bivariate barplot analysis with respect to *y*.

heatmap introduces the insight that the observed clear correlation between *pdays* and *previous*, aligns with the expected relationship. Both variables represent information about past contacts with the potential client, with one indicating the number of days that passed by after the client was last contacted from a previous campaign, and the other measuring the number of contacts performed before this campaign and for this client. Given this strong correlation and to avoid multicollinearity issues, it was decided to drop the variable *pdays*, leading to a shape of $(43621, 14)$. Conversely, the heatmap reveals that there are no significant correlations between any other pairs of numerical variables, indicating that these variables can be considered relatively independent in the context of our analysis.

To continue with a detailed look into the distribution of the categorical variables with respect to the target variable is taken, and it is presented in Figure 6. In our exploration of bivariate relationships, the barplots provide a nuanced perspective on the dataset, albeit acknowledging its inherent imbalance. The interplay of variables such as *job*, *marital*, *education*, *housing*, *loan*, *contact*, and *month* reveals intriguing patterns that offer valuable insights.

Looking at the occupational distribution in relation to subscription acceptance, we observe distinct behavior. *retired* individuals and *students* show a pronounced inclination towards accepting the offer, while those in *management* and *entrepreneur* roles lean towards declining it. Moving to *marital* status, *single* individuals exhibit a higher likelihood of accepting the marketing offer compared to their *married* counterparts. *Education* levels further illuminate the landscape, with a discernible trend indicating a higher ratio of acceptance over rejection as education levels increase, particularly for the *tertiary* category. The impact of financial considerations becomes apparent as we explore *housing* and *loan* status. Individuals without a housing loan and those without a personal loan demonstrate a greater likelihood of accepting the marketing offer. The mode of *contact* also plays a role, as individuals reached through *cellular* communication are more inclined to accept the offer compared to those contacted via *telephone*. Finally, temporal variations manifest in certain months. Notably, *sep*, *dec*, and *feb* stand out with some of the highest ratios of accepting the marketing offer over rejecting it, hinting at potential seasonality effects influencing subscription decisions. These intertwined observations paint a holistic picture, setting the stage for a more comprehensive understanding of the complex relationships within the dataset.

5.3.1. Chi-squared tests

At this point it is necessary to statistically test if correlation exists between the categorical variables, and this is accomplished by using the Chi-quared[4]. By taking a look at the final results of the tests (found here[5]), the following conclusions can be derived.

Upon examining the Chi-squared results, we uncover compelling associations among categorical variables within the dataset. Notable connections emerge, revealing robust associations between *job*, and *marital* status, *education*, and *loan*. Additionally, *marital* status demonstrates strong associations with both *education* and *loan*. Further insights unfold as *education* and *month* display a significant connection. *Default* and *loan* exhibit a strong association, while *housing* and *month* showcase a noteworthy relationship. Moreover, robust associations are revealed between *loan* and *month*, *contact* and *month*, and *month* and *y*. These findings underscore the intricate web of relationships among categorical variables, highlighting the imperative need for careful consideration in model development.

[4]`https://en.wikipedia.org/wiki/Chi-squared_testtest`

[5]`https://github.com/odyskypa/aml-bank-marketing-campaign/blob/main/notebooks/chi-2.csv`

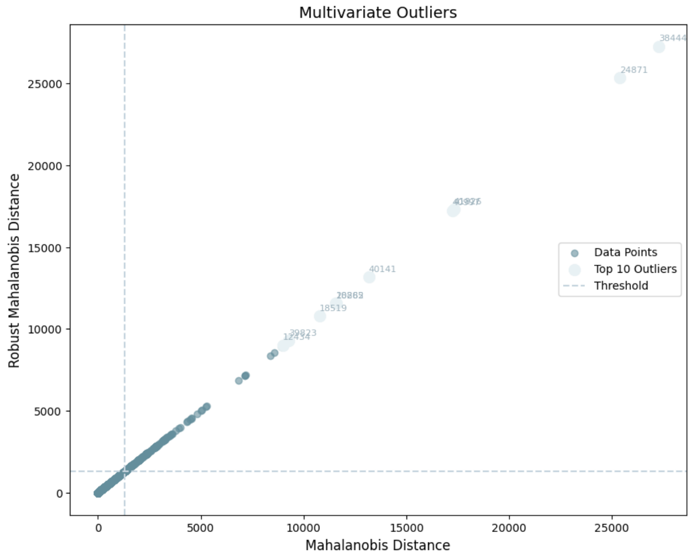

Figure 7: Multivariate outliers with MD and RMD.

Given these significant associations, adopting a dimensionality reduction approach may prove beneficial. For this reason, a try of performing kernelized Principal Components Analysis[6] (PCA) and then fitting a model was tested during the current implementation. This strategy would compress the wealth of information into a more manageable set of variables, enhancing both model efficiency and interpretability.

5.3.2. Multivariate outliers

In our pursuit to identify potential multivariate outliers within the dataset, we utilized two distinct distance measures: Mahalanobis Distance[7] (MD) and Robust Mahalanobis Distance[8] (RMD). In order to do so the calculation of the covariance matrix[9] and its inverse is necessary in order to calculate the distances between the observations of the dataset. The cutoff value for detecting outliers is set to 1% based on the *Chi-Square* distribution. Thus, 0.99 represents the desired significance level. Upon visualizing the distances, we observed multiple cases were multivariate outlier in both the MD and RMD occur (Figure 7). More specifically, 437 multivariate outliers are present, although after careful inspection of those cases, it was decided to discard only the 10 most extreme cases (indexed points in Figure 7).

5.3.3. Feature Preprocessing

In addressing the challenges posed by an imbalanced dataset, various procedures for feature selection and extraction were implemented. Firstly, to simplify the feature space for Support Vector Machine (SVM) fitting without information loss, we transformed the *education* variable from categorical to an ordinal numerical, with *primary* = 1, *secondary* = 2, and *tertiary* = 3. For the *job* variable, given its imbalance and excessive categories, we pursued two approaches. The first (named *job*) involved creating balanced categories, *business* and *others*, to alleviate imbalance. The second attempt (named *job2*), based on insights from bivariate analysis, refined categories to *business* (from initial values *admin*, *entrepreneur*, and *management*), *student*, and *retired*. Regarding the *marital* variable, while initially it was considered to reduce the categories to *married* and *not married*, we refrained due to likely significant bivariate relations between *single*, *divorced*, and the target variable (y). The *month* variable, with numerous modalities and despite imbalance, was retained in its original form, acknowledging its potential impact on marketing outcomes.

Finally, the inclusion of the transformations of the variables based on the analysis of section 5.2.1 is completed. Specifically, the logarithm of the positive values for the variables *campaign* and *previous* are applied (*pdays* were discarded). However, in the case of the *balance* variable, a decisive choice was made to preserve the original version. The consideration of transforming negative values necessitates careful deliberation, prompting the retention of the initial format.

5.4. Final dataset description

To conclude on the EDA, before moving on to Modeling (section 6), the univariate, correlation matrix, and bivariate figures were regenerated for the new version of the dataset, and can be found under Figures 8, 9, 10, 11.

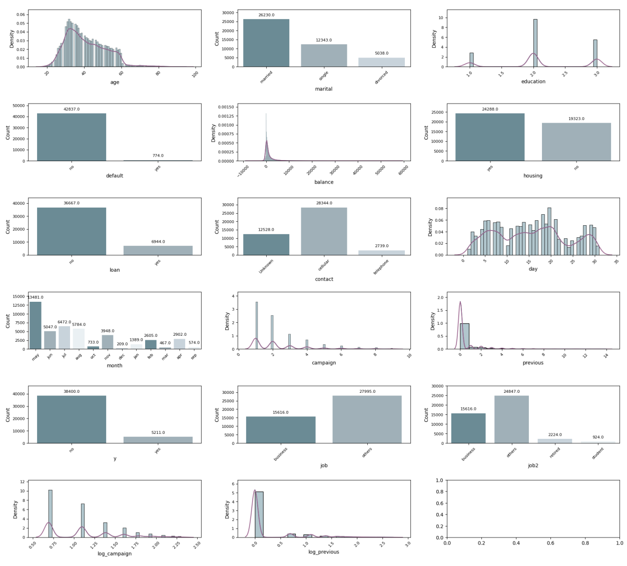

Figure 8: Univariate analysis, final dataset.

6. Modeling

In this section our methodology for designing, training and testing of several learning algorithms is described. We start showing the prior steps of preparing our data obtained from

[6]`https://en.wikipedia.org/wiki/Principal_component_analysis`
[7]`https://en.wikipedia.org/wiki/Mahalanobis_distance`
[8]`https://scikit-learn.org/stable/auto_examples/covariance/plot_mahalanobis_distances.html`
[9]`https://en.wikipedia.org/wiki/Covariance_matrix`

EDA for the different models, we then describe the main models used used in this project and finally we explain our criteria for choosing our preferred algorithm and analyze its results.

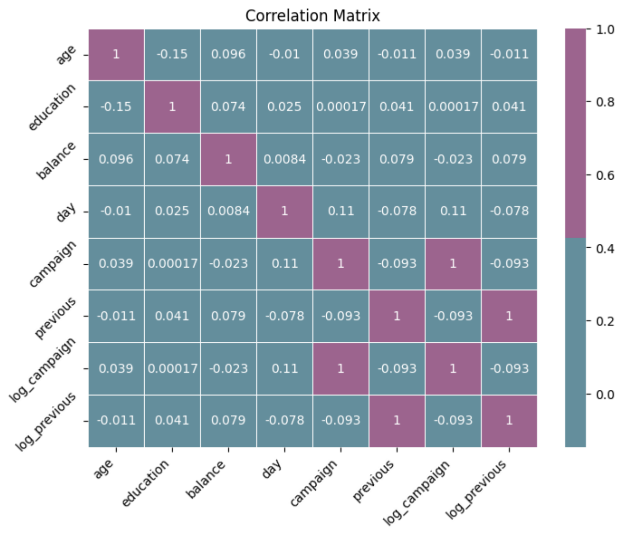

Figure 9: Spearman correlation heatmap of numerical features, final dataset.

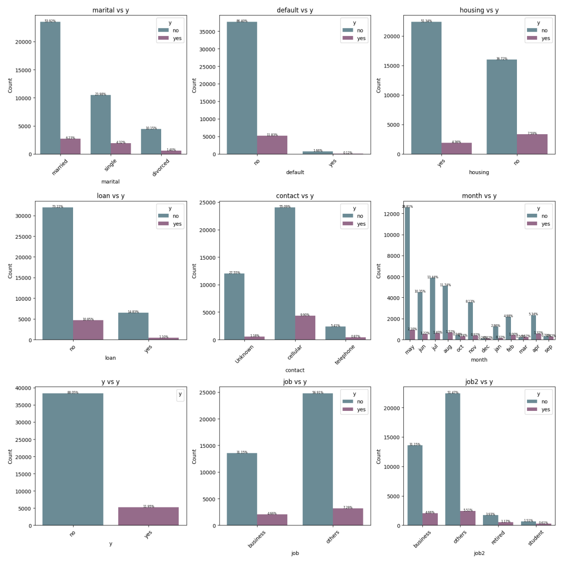

Figure 10: Bivariate barplot analysis with respect to *y*, final dataset.

6.1. Preprocessing

In order to assemble a final version of the data that is runnable in all the methods we intended to try, we decided to standardize the numerical variables using the Standard Scaler[10] and employ One-hot encoding[11] for the categorical variables.

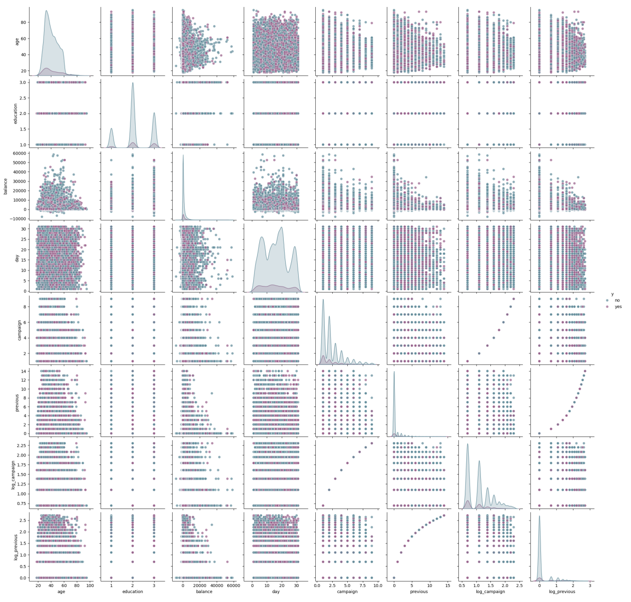

Figure 11: Bivariate scatterplot analysis with respect to *y*, final dataset.

It is very important to distinguish the preprocessing steps between the two sets, in order to avoid introducing bias. The objective is to emulate the model testing process with unseen data, mimicking a real-world scenario. The implementation of the preprocessing procedure underscores that the scaler is fitted exclusively with the training set data. Subsequently, this scaler is applied to the testing data, replicating a real-world testing scenario. It's worth noting that alternative numerical transformations, such as min-max normalization (Min-max Scaler[12]), were considered; however, their application resulted in a small decline in our final results in this particular case. The variables *job*, *campaign* and *previous* were included in some modeling tests, while in others we used *job2*, *logcampaign* and *logprevious* to obtain a comparison of the effect of using the different alternative variables.

6.2. Resampling and data splits

Initiating the modeling process requires a pivotal step – the division of the dataset into training, validation and testing sets. In our case this involved the creation of two distinct partitions, namely the *train* set, which consists of the 80% of the data, and the *test* set, with a respective 20%. The reasoning behind is that our data size is enough for allowing these split and the decision was to use the *train* split both for training and validation of the models with cross-validation. Namely, cross-validation[13] is applied exclusively to the train partition, facilitating model comparison and hyperparameter determination. The segregation of training and test sets serves as a critical measure to guarantee an impartial evaluation of the model's performance. The training

[10]`https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html`

[11]`https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html`

[12]`https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html`

[13]`https://en.wikipedia.org/wiki/Cross-validation_(statistics)`

set becomes the arena for constructing and optimizing multiple models, enabling a comparative assessment to identify the most suitable one. Meanwhile, the test set assumes the role of an unseen dataset, instrumental in evaluating the model's generalization capabilities. This approach provides a realistic estimation of the model's effectiveness on unfamiliar data. Additionally, the implementation of cross-validation on the training set presents multiple benefits. It aids in robustly fine-tuning the model's hyperparameters through iterative training and evaluation on distinct subsets of the training data. We apply it in our modeling via GridSearchCV. This iterative process identifies the optimal hyperparameter configuration, optimizing performance and mitigating overfitting. Finally, cross-validation ensures a more dependable evaluation of the model's performance, mitigating the risk of bias introduced by a single train-test split.

6.3. Undersampling and oversampling

After our univariate analysis, one of the conclusions is that *yes* was 13% of the data while a 77% of potential clients would say no to the bank marketing campaign. Because of the problems caused by the imbalance of our target variable, as we will see in the Results section, undersampling was applied at one point to mitigate the bad results for the *yes* (positive) category of the target. It was done simply by taking as much *no* instances as *yes* possible from the training set. We assessed in the real-life original test set, which was imbalanced, but also created an undersampled test set to assess the predictability capacity of our model in an hypothetical 50-50 scenario.

On the other hand, oversampling was also tried with the RandomOverSampler[14], which over-samples the minority class by picking samples at random with replacement. Nonetheless, the resulting models didn't impact with any improvements compared to the undersampling methodology, which we maintained as definitive for our final model training. By performing a quality undersampling, a benefit was also the reduction in the computation time of the learning process, without accounting for a notable reduction in the generalization capabilities of our models, in this case.

6.4. Learning Algorithms

To identify a suitable learning algorithm capable of effectively predicting the binary outcome in the dataset, various classification models were experimented with. The chosen models for this exploration encompassed Logistic Regression[15], Random Forest[16], Support Vector Machine (SVM)[17], Gradient Boosting[18], k-nearest neighbors (KNN)[19], and Naive Bayes[20].

[14]`https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.RandomOverSampler.htmlobject`

[15]`https://en.wikipedia.org/wiki/Logistic_regression`

[16]`https://en.wikipedia.org/wiki/Random_forest`

[17]`https://en.wikipedia.org/wiki/Support_vector_machine`

[18]`https://en.wikipedia.org/wiki/Gradient_boosting`

[19]`https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm`

[20]`https://en.wikipedia.org/wiki/Naive_Bayes_classifier`

These chosen algorithms were deemed suitable for their ability to handle two-class outcomes, accommodate both numerical and categorical features, and capture complex relationships between predictors and the target variable. Each algorithm possesses unique strengths, such as logistic regression's interpretability, Random Forest's robustness, and SVM's proficiency in handling high-dimensional data, providing a comprehensive set of options based on specific requirements. However, as needed by the project requirements, special attention was paid to the application of the SVM algorithmics, using multiple different kernels and hyperparameter configurations. Here we present a brief explanation of the SVM based models we used and their particularities.

6.4.1. Linear Support Vector Classifier (Linear SVC)

In our initial modeling phase, we employed the primal version of the Linear Support Vector Classifier (Linear SVC). The choice of the primal definition was motivated by the fact that we had a significantly lower number of features compared to samples, leading to enhanced model training speed. Our focus on the primal version allowed us to efficiently navigate the trade-off between correct classification of training examples and the maximization of the decision function's margin.

The primary hyperparameter we tuned for this model was C, which plays a crucial role in balancing the correct classification of training examples and the margin size of the decision function. Larger C values result in a smaller margin, accepting a trade-off for improved accuracy on training points. Conversely, lower C values encourage a larger margin, yielding a simpler decision function at the expense of training accuracy. Essentially, C acts as a regularization parameter in the Support Vector Machine (SVM) framework. While the Linear SVC proved to be a fast and competitive method for the unbalanced training set, it exhibited limitations in the context of the undersampled training set.

6.4.2. Kernelized Support Vector Classifier (Kernelized SVC)

In this phase, we explored the versatility of the SVM by testing different kernel formulations, including the *polynomial*, *radial basis function* (RBF), and *sigmoid* kernels. These formulations enable the model to capture intricate relationships within the data. Each kernel exhibits unique characteristics that make them suitable for diverse patterns and structures present in the dataset.

The polynomial kernel is good at capturing complex relationships by raising the input features to different powers, introducing non-linearity into the decision boundary. It is more used in image classification though. The radial basis function (RBF) kernel and sigmoid, on the other hand, are proficient at modeling intricate and non-linear decision boundaries, making it a capable tool for capturing complex data patterns.

A key hyperparameter we addressed for the RBF and sigmoid kernels was γ (gamma). The γ parameter determines the range of influence of a single training example, with low values signifying a broader reach and high values indicating a more localized influence. Conceptually, γ can be considered as the inverse of the radius of influence of samples designated as support vectors by the model. This parameter plays a crucial role in balancing the flexibility and generalization of the model, making it a pivotal consideration in our hyperparameter tuning process.

6.5. Model comparison & hyper-parameter tuning

As stated in the Resampling protocol section (6.2), employing cross-validation is advantageous for comparing models and tuning hyperparameters. This involves an iterative process of training and evaluating each learning algorithm with distinct training subsets. For every learning algorithm, we utilize GridSearchCV[21], a combination of grid search and cross-validation, to determine the optimal hyperparameters. In our scenario, the grid search with cross-validation internally divides the training data into training and validation subsets for each cross-validation fold, eliminating the need to create validation subsets from the original data beforehand. The process identifies the best hyperparameters based on mean performance metrics (accuracy, precision, recall, f1-score) calculated on the validation subsets, but mainly focused on optimizing the accuracy metric. Additionally, focusing on precision metric was tested as well in the current solution, although the results were not getting better. This ensures that the final evaluation of the chosen model on the test set provides an unbiased estimate of the model's performance and generalization. Training scores for each model are illustrated in Tables 5, 6.

Table 5: Best Hyperparameters for Each Model

Model	**Best Hyperparameters**
Logistic Regression	{'C': 0.001, 'class_weight': 'balanced', 'penalty': 'l2'}
Random Forest	{'class_weight': 'balanced', 'max_depth': 10, 'n_estimators': 100}
SVM_linear	{'C': 0.001, 'class_weight': 'balanced'}
SVM_poly	{'C': 10, 'class_weight': 'balanced', 'degree': 2}
SVM_rbf	{'C': 1, 'class_weight': 'balanced', 'gamma': 'scale'}
SVM_sigmoid	{'C': 0.01, 'class_weight': 'balanced', 'gamma': 'scale'}
Gradient Boosting	{'learning_rate': 0.1, 'n_estimators': 300}
KNN	{'n_neighbors': 7, 'weights': 'distance'}
Naive Bayes	{}

Table 6: Model Performance Metrics

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**	**Valid. Error**
Logistic Regression	0.683	0.684	0.680	0.682	0.011
Random Forest	0.716	0.747	0.652	0.696	0.008
SVM_linear	0.681	0.691	0.655	0.673	0.007
SVM_poly	0.710	0.777	0.589	0.670	0.010
SVM_rbf	0.712	0.771	0.603	0.677	0.010
SVM_sigmoid	0.668	0.645	0.745	0.692	0.009
Gradient Boosting	0.720	0.760	0.644	0.697	0.004
KNN	0.673	0.687	0.634	0.660	0.005
Naive Bayes	0.636	0.761	0.397	0.522	0.007

In light of the cross-validation results, with a primary focus on optimizing accuracy, it is evident that models such as Gradient Boosting, Random Forest, SVM (with sigmoid kernel), and SVM (with rbf kernel) stand out, excelling in terms of f1-score, accuracy, and precision respectively. However, given the specific objectives of the model, placing a significant emphasis on achieving a high recall value is crucial. The primary goal is to effectively classify positive outcomes in future marketing campaign calls, identifying clients more likely to subscribe to the marketing offer, which is equivalent to obtaining a high recall for the *yes* class. To choose the algorithm to focus for our final model, we looked at the table and checked which one suited more overall our use-case, to then focus and try several configurations within it (giving more or less weights to class *yes*, training on normal or undersampled training set, etc.) Consequently, the selected model for this use-case is the SVM (with sigmoid kernel). A possible hyperparametrization for optimizing accuracy and still getting recall capabilities would be 'C': 0.01, 'class_weight': 'balanced', 'gamma': 'scale'. This particular model attains the highest mean recall of **0.744802**, making it the optimal choice for the intended purpose.

[21] `https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html#sklearn.model_selection.GridSearchCV`

7. Results

In this section we take a look at some of the main metric results and model configurations we obtained throughtout our work. The most optimal featurization comprises the following 28 numerical, standarized and one-hot encoded variables: *age*, *education*, *default*, *balance*, *housing*, *loan*, *day*, *campaign*, *previous*, *y*, *job*, *marital-divorced*, *marital-married*, *marital-single*, *contact-unknown*, *contact-cellular*, *contact-telephone*, *month-apr*, *month-aug*, *month-dec*, *month-feb*, *month-jan*, *month-jul*, *month-jun*, *month-mar*, *month-may*, *month-nov*, *month-oct* and *month-sep*. We can analyze the results in Table 7, where all the models were evaluated in the real-life case test set that contains 7650 *no* and 1073 *yes* target values.

Table 7: Main modeling results

Model	**Accur.**	*no* **Pre.**	*yes* **Pre.**	*no* **Rec.**	*yes* **Rec.**	**%** *yes*
FT-Linear SVC (C=0.1)	0.88	0.88	0.59	0.99	0.06	1
FT-SVC_sigmoid	0.88	0.88	0.00	1.00	0.00	0
UT-Linear SVC (C=0.01)	0.70	0.94	0.24	0.69	0.69	35
UT-SVC_sig	0.69	0.94	0.24	0.69	0.69	35
UT-SVC_rbf	0.78	0.94	0.31	0.81	0.60	25
UT-SVC_rbf (x2 yes)	0.49	0.95	0.17	0.45	0.83	58
UT-SVC_sig (rec, x2 yes)	0.38	0.96	0.15	0.31	0.90	72
UT-SVC_sig (C=0.01)	0.63	0.94	0.21	0.62	0.74	42

Every model in the table is the optimal result of a hyperparametrization process, in Linear SVC for C and for sigmoid kernel SVC for C and γ. The columns correspond to all the models names, the precision and recall for the *yes* and *no* class and finally the percentage of data labeled as *yes* by the model.

Firstly, we can observe the results for the main models above the full training set (FT). It is clear that our models lack the capability of predicting the *yes* class, which is clearly reviled in front of the *no* class. The accuracy both with the linear and sigmoid kernel SVM is 0.88, and we can even see that the sigmoid model doesn't even try to predict the *yes* instances, with a recall of 0. As we explained, in order to mitigate this we decided to test an undersampled training set (UT). In the table we can distinguish multiple of our UT models, being Linear, RBF or sigmoid based. Other than the default RBF and sigmoid configurations hyperparametrizing γ and C, with accuracy as the scoring function, we present here other models giving 2 times the weight to the *yes* class. Also, the model UT-SVC_sig(rec,x2 yes) considers recall as the primary scoring function for choosing the best model.

The reasoning behind presenting all these models are the differences they have in obtaining a higher recall for the *yes* class in exchange for a worse precision for that class. The main hypothetical goal of the modelling is to obtain a model that can tell us how to proceed among our clients in terms of marketing, which means predicting the *yes* target for each one of them and making decisions about our strategy. For us, this is equivalent to targeting those clients that our model say they potential *yeses*.

For instance, model UT-SVC_rbf tells us that 25% of the data are potential *yeses*, from which we would get a 60% of all the *yes* clients targeted with our campaign. On the other hand, UT-SVC sig (rec, x2 yes) would say *yes* to 72% of the data, which we'd call and would account for 90% of recall of the *yes* clients. The outcome is that we can choose a model depending on our budget for market campaigning and willingness to make more calls for less *yes* ratio in total, whilst every time getting more clients.

7.1. Kernelized PCA

In an interesting test, we conducted an experiment utilizing Kernel Principal Component Analysis (KernelPCA). This experiment was designed to visualize the data boundaries after transforming all variables into numerical representations. KernelPCA is a technique that applies the kernel trick to Principal Component Analysis (PCA), allowing for the non-linear transformation of data. It is useful for revealing complex patterns and structures in high-dimensional spaces.

We can see that the decision boundaries obtained with the sigmoid kernel in 12 appear curved and irregular. The decision boundary tries to separate the classes by fitting a sigmoid-shaped curve, resulting in a complex boundary that may not generalize well to unseen data. From this example it becomes obvious, that the sigmoid kernel has very specific use cases, when dealing with data that exhibits a sigmoidal shape.

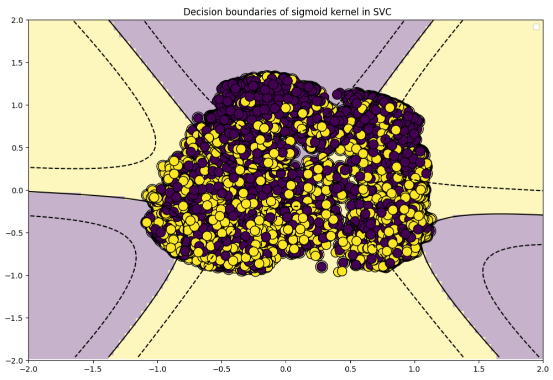

Figure 12: Decision boundaries of a sigmoid kernel over the PCA components

8. Final model

Our final model is a Support Vector Machine (SVM) with a Radial Basis Function (RBF) kernel, specifically configured with $C = 0.01$ and γ = 'scale'. The 'scale' option assigns gamma to the ratio between 1 and the length and variance product of the training set. This model was trained on our undersampled dataset, and it incorporates balanced weights for the two binary classes. The RBF kernel enables the model to effectively capture complex relationships within the data, and the tuning of hyperparameters ensures a balanced and well-generalized performance.

8.1. Performance metrics

The selected model achieved an accuracy of 0.63, showcasing its mediocre ability to make correct predictions across both classes. Specifically, the precision for the negative class (no) is impressive at 0.94, indicating a high proportion of correctly identified negative instances. On the other hand, the precision for the positive class (yes) is 0.21, emphasizing the challenge of correctly predicting positive instances.

In terms of recall, the model demonstrates a balanced performance, with recall for the negative class (no) at 0.62 and for the positive class (yes) at 0.74. The confusion matrix, as illustrated in Figure 13, provides a detailed overview of the model's performance in terms of instance clients.

However, our primary focus lies in the recall versus the proportion of data targeted as "yes," which stands at 42%. This signifies that by implementing a marketing strategy directed to this subset of potential clients, we can successfully capture 74% of all feasible clients. This strategic approach optimizes resource utilization by efficiently targeting a subset of the total potential client pool.

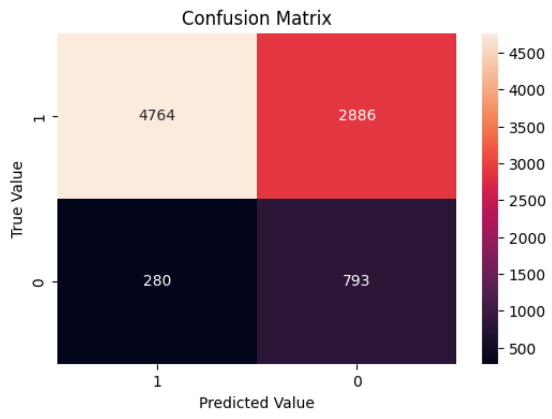

Figure 13: Confusion matrix of the final model yes=1, no=0

8.2. Feature importance

In Figure 14, we present the calculated feature importances for our final model, derived through the permutation feature importance method. This model inspection technique is applicable to any fitted estimator, particularly advantageous for non-linear or opaque models. The permutation feature importance quantifies the reduction in model performance when a single feature value is randomly shuffled. By breaking the relationship between the feature and the target, the drop in the model score reflects the dependency of the model on that specific feature.

According to our results, the top three variables in terms of importance for the classification are ”housing,” ”contact_unknown,” and ”previous.” These features play a pivotal role in influencing the model’s predictions, providing valuable insights into the factors driving the classification outcomes.

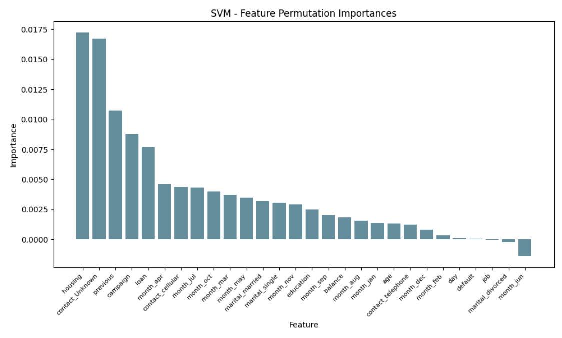

Figure 14: Permutation importances over the final model

8.3. *Bias-variance tradeoff*

The difficulties of model development involves a balance between bias and variance[22]. As we move into the final stages of our analysis, it becomes crucial to understand the dynamic interplay between these two contrasting elements. Bias represents the model’s simplifying assumptions, capturing a high-level overview of the underlying patterns within the data. On the flip side, variance accounts for the model’s sensitivity to fluctuations in the dataset, potentially leading to overfitting. Striking the right equilibrium between bias and variance is paramount for crafting a predictive model that not only captures the complexity of the underlying relationships but also maintains its adaptability to unseen data. Thus, in this section we investigate how different parameters affect the bias-variance tradeoff of the final model, aiming for a final result that maximizes predictive accuracy and generalizability.

Firstly, the change in the model’s training and testing error is investigated, when different values of the hyper-parameters C, *gamma* and *class_weight* are used. The results of the hyper-parameters, can be found in the *02.Experiments.html*[23]. Moreover, it is investigated how the change of the training sample’s size affects the bias and variance of the model (Figure 15). From Figure 15, one can understand that as the training set size which is fed into the model increases, the training misslcassification error decreases but very slightly, since the worst error achieved is approx. 0.32. As for the test misslcassification error, it is clear that in fluctuates between the values 0.03 and 0.88 when the training sample is kept at small numbers $(0 to 1600)$, althouh for higher numbers both errors tend to stabilise. The slopes of the figure suggest that the increase in the total size of the dataset, would lead the two curves to come closer and closer, meaning that the model would be able to generalize satisfactorily.

[22]https://en.wikipedia.org/wiki/Bias%E2%80%93variance_tradeoff

[23]https://github.com/odyskypa/aml-bank-marketing-campaign/blob/main/notebooks/02.Experiments.html

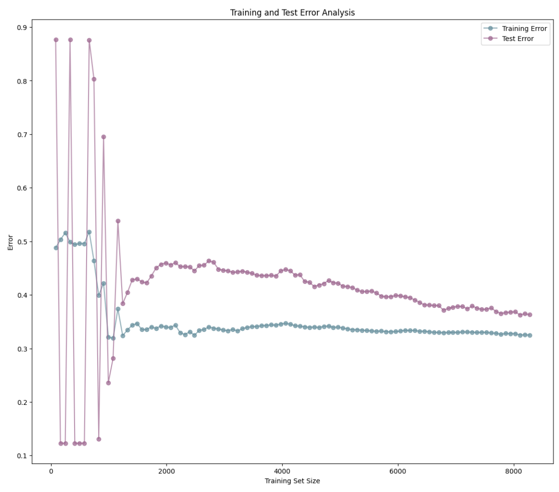

Figure 15: Training and test error analysis for different values of C.

9. Conclusion

In the course of this project, we deepened into the intricacies of a binary classification problem centered around bank marketing. Our primary focus was to navigate the challenges around in adapting the data for both linear and kernelized Support Vector Machine (SVM) models, and subsequently, compare their performance against alternative classification methods. The dataset posed a considerable challenge, yielding non-trivial and imperfect results across various machine learning models. Consequently, we engaged in a comprehensive analysis to discern the potential scenarios for our marketing campaign and identify models that offer valuable insights.

This project has been good in providing an understanding of the advantages and limitations associated with SVM modeling, especially in the context of diverse data types. It underscores the importance of thoughtful analysis and consideration of various modeling approaches when dealing with real-world datasets, putting light on the complexities involved in decision-making for marketing strategies. Through this exploration, we have learned valuable lessons into the interplay between data characteristics and model performance, contributing to our broader understanding of machine learning applications in marketing contexts.

10. Possible extensions and known limitations

Exploring potential areas for improvement and acknowledging existing constraints, this section goes into both the promising possibilities for extending the current analysis and the recognized limitations in the methodology employed.

An identified limitation of our analysis lies in the available features, as the dataset may lack certain variables that could enhance the predictive power of the model. Despite conducting a thorough exploration through univariate and bivariate analyses, the absence of specific features could impact the overall model performance, as it was shown in the results of the state-of-the-art(4), where more advanced features were available.

Additionally, it's worth noting that the simplicity of our feature engineering approach might limit the model's capability to capture more complex relationships within the data. While the current solution offers valuable insights and predictive capability, a more extensive exploration of intricate feature engineering tasks could potentially discover hidden patterns, leading to deeper improvements in the predictive model.

Unfortunately, a direct comparison with the state-of-the-art model from the literature(4) was not feasible in our analysis. The authors of that paper had access to a more extensive set of relevant features, which played a crucial role in their model's performance. Additionally, their approach involved the utilization of Neural Networks, a sophisticated technique beyond the scope of our current project. While we aimed to optimize our model with the available data and features, the comparison is limited due to the differences in feature richness and modeling methodology. However, it is considered as future work, to apply most sophisticated learning techniques, as the Neural Networks, and compare the results with the state-of-the-art.

References

[1] Odysseas Kyparissis and Pau Comas. aml-bank-marketing-campaign. `https://github.com/odyskypa/aml-bank-marketing-campaign`, 2023.

[2] Andrej Miklosik and Nina Evans. Impact of big data and machine learning on digital transformation in marketing: A literature review. *IEEE Access*, 8:101284–101292, 2020.

[3] S. Moro, P. Rita, and P. Cortez. Bank Marketing. UCI Machine Learning Repository, 2012. `https://doi.org/10.24432/C5K306`.

[4] Sérgio Moro, Paulo Cortez, and Paulo Rita. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62:22–31, 2014.