[**Machine Learning Techniques for Predicting the Success of Bank Telemarketing**](https://www.sciencedirect.com/science/article/pii/S016792361400061X) **-**

**CRISP-DM Methodology**

Ani P. Petkova

Sofia University „St. Kliment Ohridski”,

Faculty of Economics and Business Administration

Sofia, Bulgaria

****

**Contents**

Executive Summary 3

1.Business Understanding 3

* + 1. Business Objectives 3
    2. Business Requirements 3
    3. Solution Success Criteria 4
    4. Project Plan 4

2.Data Understanding 4

* 1. Initial Data Collection. 4

2.2 Data Description 4

3.Data Preparation 5

3.1 Data Selection and Data Cleaning. 5

3.2 Data Construction 6

4. Modeling 6

4.1 Model Building. 6

5. Evaluation 7

5.1 Business Requirements. 7

5.2 Performance Metrics 7

5.3 Solution Success Criteria 8

6. Appendix 9

**Executive Summary**

The purpose of the project is to propose a Machine Learning (ML) algorithm to predict the success of telemarketing calls for selling bank long-term deposits. The data were collected from direct marketing campaigns (phone calls) of a Portuguese banking institution, consisting of 41188 observations and 20 features. In the banking sector, optimizing telemarketing targeting is a significant challenge, driven by the need to enhance profits and minimize costs. Due to the 2008 financial crisis, Portuguese banks faced pressure to raise capital requirements, for instance, by attracting additional long-term deposits. In this context, the use of the proposed ML algorithm allows banks to target their marketing efforts more effectively and helps with client selection decisions.

**1. Business Understanding**

**1.1Business Objectives**

It is clear that the goal is to increase efficiency of directed campaigns for long-term deposit subscriptions by predicting if a client will subscribe to the deposit. To achieve this most effectively, our job is to attain the highest accuracy in the classification model and enhance performance by reducing complexity through dimensionality reduction techniques.

**1.2 Business Requirements**

The classification model should meet the following requirements:

* A machine learning algorithm should be used for the classification task (e.g., SVM, KNN, Logistic Regression, etc.).
* The model should be trained on a dataset comprising no fewer than 500 observations and 5 features.
* Both a dimensionality reduction method and an analysis of the components should be performed.
* Either KFold or train-test-split should be used for partitioning the dataset.
* To assess the model's performance, metrics such as accuracy, confusion matrix, and classification report should be employed.

**1.3** **Solution Success Criteria**

* Meet all business requirements.
* Optimize the model for maximum accuracy and recall coefficients.
* Reduce the dimensionality of the dataset while retaining as much variance as possible.

**1.4 Project Plan**

1. Find suitable data that meet the project requirements.
2. Define goals and formulate an action plan to follow.
3. Data cleaning, data modeling and assessment of results.
4. Create comprehensive documentation for the entire CRISP-DM process.
5. Submit project and present results.

**2. Data Understanding**

**2.1** **Initial Data Collection**

The dataset used for the purpose of this project is related to direct marketing campaigns of a Portuguese banking institution and is available at the UCI Machine Learning Repository. It consists of 41188 instances and 20 features and is ordered by date. This dataset meets the requirements specified in *Business Requirements*.

**2.2** **Data Description**

This project considers real data collected from a Portuguese retail bank, from May 2008 to November 2010, in a total of 41188 phone contacts. This time period includes the effects of the global financial crisis that peaked in 2008. Because of the crisis, Portuguese banks were pressured to increase a financial asset. In response to this challenge, a strategy employed involves presenting enticing long-term deposit opportunities featuring favorable interest rates, primarily through targeted marketing campaigns. The competitive environment drives the banks to minimize costs and time and therefore, there is a need to enhance efficiency, aiming for fewer interactions while maintaining a consistent level of successful outcomes (clients subscribing the deposit).

The dataset is unbalanced, as only 4640 **(**11.26%**)** records are related with successes. Each record includes1 :

* The contact outcome (failure or success). This is also the target variable.
* Candidate input features. These include client information (age, marital status) and telemarketing attributes (e.g., call durations, last day of contact).
* Social and economic influence features (e.g., employment variation rate).

**3. Data Preparation**

Since we have been working with Spyder (Python 3.7) throughout the semester, choosing it for the purpose of this project was a natural decision. Python offers a comprehensive suite of libraries and classification techniques, providing everything needed to develop the required classification model. The libraries used are as follows:

* + *pandas* – used for reading the bank-additional-full.csv file
  + *numpy* – calculates cumulative sum
  + *seaborn* – used for creating count plots
* *sklearn.preprocessing* – used for scaling data and converting categorical labels into numerical format
  + *matplotlib.pyplot* – used for creating plots
  + *sklearn.decomposition* – used for performing Principal Component Analysis (PCA)
  + *sklearn.linear\_model* – used for building Logistic Regression
  + *sklearn.model\_selection* - used for implementing Kfold cross-validation
  + *sklearn.metrics* – used for visualizing confusion matrices and generating classification reports

**3.1 Data Selection and Data Cleaning**

As stated in *Data Description* the data consist of 41188 observations and 20 features (variables). The first step is to load the data using *pd.read\_csv*. There are several missing values in some categorical attributes, all coded with the "unknown" label. That is why *pandas’s .apply()* function is used to remove any observations that include the word "unknown". Then a check for any other null values in the dataset is performed using *dataset.isnull().sum().* A count of the target variable is performed using *sns.countplot2*. The target variable has two values: 'no,' which occurs 26,629 times, and 'yes,' which occurs 3,859 times. (23-43).

1­See *Appendix 1* for a detailed description of all variables

**3.2 Data** **Construction**

The *LabelEncoder()* function is used for transforming categorical labels into numeric format, making the dataset much easier to work with. The next step is to define the dependent variable 'y' by selecting the values from the last column of the dataset. After that, we define 'Y' which consists of all the other columns - the independent variables. *StandardScaler()* is used to standardize the data. This helps in centering the data and removing any bias due to varying scales (46-65).

**4. Modelling**

**4.1 Model Building**

The initial step in this section involves constructing the *Principal Component Analysis (PCA)* model. The *pca.**explained\_variance\_ratio\_* shows how much information each principal component retains from the original data. A plot of cumulative explained variance ratio is build to visually show the cumulative contribution of each principal component to the overall variance3. This helps to determine the number of principal components needed to capture a satisfactory amount of variance in the data. From the plot it is clear that 13 components explain 88% of the total variance. A PCA model with *n\_components = 13* is built (67-82).

The next step is to define the model that will be used for the classification of the data. After thoroughly evaluating different models, a final decision to use *Logistic Regression* was made. The Logistic Regression model with L2 regularization, commonly referred to as Ridge Regression, is instantiated with the parameters *penalty='l2', max\_iter=1000*, and *class\_weight='balanced'*. The *class\_weight* parameter is set to 'balanced' due to the unbalanced nature of the dataset.(85)

Then cross-validation is performed by dividing the training set into k parts. It performs k trainings with k-1 parts and always keeps one part for validation. In this case, the parameters for KFold cross-validation are set as *n\_splits = 20, shuffle = True, and random\_state = 99*. *n\_splits* is used to specify the number of folds in KFold cross-validation. Two loops are constructed to find and save the train\_maxindex and test\_maxindex, which represent the indices with the highest test and training scores (87-117).

2See *Appendix 2* for a visualization of the count plot

3See *Appendix 3 for an explained variance ratio plot*

The final part is to visualize the confusion matrix and generate a classification report for the test set4 (119-123).

1. **Evaluation**

**5.1Business Requirements**

All business requirements specified in section *1.2* are met. The model was trained on a dataset comprising 41,188 observations and 20 features. Utilizing Logistic Regression for classification and dimensionality reduction trough PCA, reducing the features to 13. A plot of the explained variance ratio was employed to analyze the components. Cross-validation was conducted to evaluate the model's performance and generalization ability. Key performance metrics such as accuracy, confusion matrix, and classification report were visualized. The next section presents the results and insights derived from the model.

**5.2 Performance Metrics**

A set of performance metrics were used to evaluate the effectiveness of the predictive model. The key metrics considered include:

* Accuracy: The overall correctness of the model's predictions.
* Recall: The ratio of correctly predicted positive observations to the total actual positives.
* Confusion Matrix: A detailed breakdown of true positive, true negative, false positive, and false negative predictions.

The current algorithm achieves an accuracy of 86%. The recall coefficient for Class 0 stands at 86%, representing the proportion of actual instances belonging to Class 0 that the model correctly identifies as Class 0 (True Negative Rate). Similarly, the recall coefficient for Class 1 is 85%, indicating the proportion of actual instances belonging to Class 1 that the model correctly identifies as Class 1 (True Positive Rate)

4See *Appendix 4 and 5 for the confusion matrix and classification report*

Note: It is important to note that several other classification models were tested (SVM, Decision Tree, Random Forest), some of them reached higher Accuracy results, however Recall Coefficients were way too unbalanced to consider the models as effective. Example: Recall Coefficient Class 0 – 96%, Recall Coefficient Class 1 – 33%. This is the reasoning behind choosing an algorithm with less overall Accuracy.

**5.3 Solution Success Criteria**

* + Meet all business requirements – as stated in section *5.1* all business requirements were successfully fulfilled.
  + Optimize the model for maximum accuracy and recall coefficients – as shown in the previous section, the selected model achieves best results in terms of accuracy and recall coefficients
  + Reduce the dimensionality of the dataset while retaining as much variance as possible – through implementing PCA, the features were reduced to 13, explaining nearly 90% of the total variance.

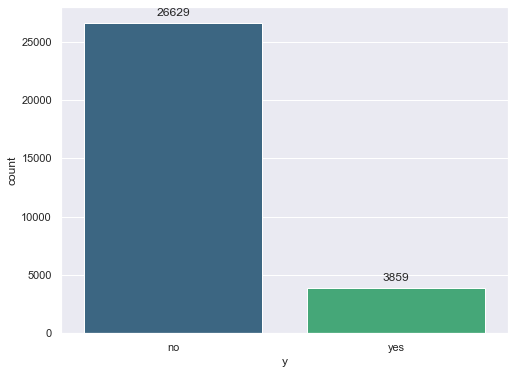
In conclusion, the project not only aligns with business needs but also optimizes model performance, striking a balance between accuracy and recall coefficients while effectively reducing dimensionality. These achievements position the project as a success in meeting its objectives.

**6. Appendix**

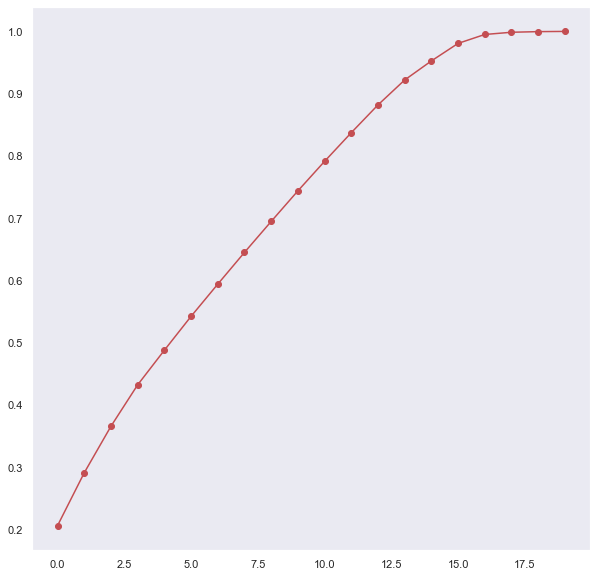
**1 List of Variables**

|  |  |  |  |
| --- | --- | --- | --- |
| Number | Name of variable | Type | Description |
| 1 | age | Numeric | Age of the individual |
| 2 | job | Categorical | Type of job |
| 3 | marital | Categorical | Marital status |
| 4 | education | Categorical | Level of education |
| 5 | default | Categorical | Credit in default |
| 6 | housing | Categorical | Housing loan |
| 7 | loan | Categorical | Personal loan |
| 8 | contact | Categorical | Communication type for last contact |
| 9 | month | Categorical | Last contact month of the year |
| 10 | day\_of\_week | Categorical | Last contact day of the week |
| 11 | duration | Numeric | Last contact duration in seconds |
| 12 | campaign | Numeric | Number of contacts during this campaign |
| 13 | pdays | Numeric | Days since the client was last contacted |
| 14 | previous | Numeric | Number of contacts before this campaign |
| 15 | poutcome | Categorical | Outcome of the previous marketing campaign |
| 16 | emp.var.rate | Numeric | Employment variation rate |
| 17 | cons.price.idx | Numeric | Consumer price index |
| 18 | cons.conf.idx | Numeric | Consumer confidence index |
| 19 | euribor3m | Numeric | Euribor 3-month rate |
| 20 | nr.employed | Numeric | Number of employees |
| 21 | y | Binary: “yes”, “no” | has the client subscribed a term deposit? |

**2 Count of Variables**



1. **Explained Variance Ratio**

****

1. **Confusion Matrix**

|  |  |  |
| --- | --- | --- |
|  | Actually Positive | Actually Negative |
| Predicted Positive | 1120 | 186 |
| Predicted Negative | 33 | 185 |

1. **Classification Report**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1-Score | Support |
| Class 0 | 0.97 | 0.86 | 0.91 | 1306 |
| Class 1 | 0.50 | 0.85 | 0.63 | 218 |
| Accuracy |  |  | 0.86 | 1524 |
| Macro Avg | 0.74 | 0.85 | 0.77 | 1524 |
| Weighted Avg | 0.90 | 0.86 | 0.87 | 1524 |