Prediction of prospect customers response “YES” or “NO” to open a term deposit account of Bank Marketing Campaign.

Using Classification and Regression Machine Learning tools

In Python.

By: Marina Golberg 501072689

marina.golberg@ryerson.ca

Ryerson University: The Chang School of Continuing Education

CIND820: Big Data Analytics Project

Date: 12/03/2021

Instructor: Dr. Tamer Abdou

Table of Contents

Section 1: Abstract

Businesses these days are using every available platform to do marketing of their company, products, and services. They use telemarketing, email marketing, advertisements, and many other different tools to gain sales and expand their business. Telemarketing is still the most used method of increasing sales of small to big companies. It is a cost-effective and flexible marketing strategy that offers a high level of personal contact.

Predictive analytics uses data models, statistics, and machine learning to predict future events. It's a discipline that helps you to analyze your marketing campaigns, assess their efficiency, and see possible improvements to lead an increase in sales in future.

In this capstone project, the theme is Classification and Regression. The goal of these classification models is to help the company more reliably predict future customer subscription before it occurs to secure deposits more effectively and increase customer satisfaction by reducing undesirable advertisements for certain customers.

This project will be able to answer the following questions:

- Will prospective customers respond "yes" or "no" to term deposit subscription?
- What type of customers are more likely to subscribe to term deposit and which feature has higher influence?
- What is the best time of year for a marketing campaign?

Data Set Information:

The dataset is publicly available for research from the UCI Machine Learning repository. It is about a direct marketing campaign of a Portuguese banking institution, based on more than one phone calls to access if the bank term deposit would be ('yes') or not ('no') subscribed. The details are described in (Moro et al., 2011).

Data can be found here: https://archive.ics.uci.edu/ml/datasets/bank+marketing

Code and documentation for this project on GitHub repository as following:

https://github.com/marinagolberg/CIND820-MarGolb.git

The tools that used for this project are:

Jupyter notebook for Python environment, Pandas to perform data manipulation and analysis, NumPy will be used to perform a wide variety of mathematical operations on arrays, Seaborn, Plotly and Matplotlib will be used for data visualization statistical graphing and plotting, Sklearn will be used for classification and regression, these library contains a lot of efficient tools for machine learning and statistical modeling.

Summary of technics that will be used to answer the questions:

- **Subscription of customers to a term deposit.**

To answer first question, all data will be splited to training and testing set. Categorical data encoded to numerical labels and SMOTE will be used as an oversampling technique to balance the data. Classifiers to predict the results of "yes" or "no" are Logistic Regression, Decision Tree Classifier, Random Forest. Then I will compare the scores of the algorithms to choose the best one.

- **Customer profiling, feature influence on subscription, and campaign profiling.**

To answer the second and third questions. Feature selection techniques of Filter, Wrapper and Embedded Methods will be used to check the contribution of each attribute on explaining the client's subscription then the technics will be compared and tested to choose the best combination of attributes. Exploratory data analysis will be used to visually see the relationship between variables and the dependent variable "y" to build a campaign profile and customer profile.

Will prospective customers respond "yes" or "no" to term deposit subscription?	What type of customers are more likely to subscribe to term deposit and which feature has higher influence?	What is the best time in a year for marketing campaign?
Logistic Regression	Exploratory Data Analysis for customer profiling	Exploratory Data Analysis for campaign profiling
Decision Tree Classifier	Feature selection(Filter, Wrapper, Embedded) for feature influence on subscription	
Random Forest		

Figure 1.0: Summary of technics

Section 2: Literature Review

This literature review will discuss techniques and results that were found in previous researches and what will be the best approach for further analysis.

Predicting term deposit subscription from similar datasets

The purpose of the first study was to find if the use of the Random Forest improves the performance of the Decision Tree for the bank customer marketing response prediction (Olatunji,2016). Classification algorithms used for modelling were; Logistic Regression, Decision Tree, Naive Bayes and the Random Forest ensemble. These algorithms were applied to both the balanced and original bank data by ten-fold cross-validation method. Results derived from the experiment showed that the performance of the Random Forest improved when the data was balanced. The Decision Tree algorithm returned 76.6% area under Curve (AUC) and Classification Accuracy (CA) compared to Logistic Regression 75.7% and Naive Bayes 75.6%. The Random Forest had an AUC and CA value of 74.2%. There were no found improvements in Random Forest (Olatunji, 2016). Therefore, second experiment was conducted and the results showed that the performance metrics of Random Forest increased with an increase of "n" to 200. (Olatunji, 2016). The second study found that changing the number of trees has no significant effects on mean accuracy of the Random forest. Random Forest and k-Nearest Neighbor are proved to be the best classifiers for any type of dataset. (Singh et al., 2017).

A time-ordered split was performed, where the data was divided into four years of training (May 2008 to June 2012,) and one year of testing (July 2012 to June 2013). The authors' decision was to merge the two data sources that led to a large data set of 150 attributes, which could be potentially useful features (Moro et al., 2014). Using a semi-automated feature selection procedure within the modeling stage, researchers selected a reduced set of 22 relevant features. The dataset was unbalanced. Four DM learning techniques were explored: logistic regression, decision trees, support vector machines (SVMs), and a neural network. The best result was achieved by the neural network AUC = 0.8 and ALIFT = 0.7(Moro et al., 2014). Such a model was then improved by including customer lifetime-value-related features, increasing the performance to 83% of subscribers with the half better classified contacts (Moro et al., 2015b).

The goal of Ghatasheh et al.’s research is to improve the performance of predicting the readiness of customers to subscribe for a term deposit in a highly imbalanced dataset. It proposes improved Artificial Neural Network models (i.e., cost-sensitive) to facilitate the dramatic influence of highly imbalanced data, without changing the original data samples. Authors created models that was compared to different machine-learning models evaluated and validated. Telemarketing dataset from a Portuguese bank is used. The model achieved the greatest prediction of 79% geometric mean and minimized misclassification errors to 0.192 Type I, and 0.229 of Type II. In conclusion, the Meta-Cost method improved the performance of the prediction model without imposing important processing overhead or changing original data samples (Ghatasheh et al., 2020). Experiment proved that MetaCost reduces cost by large amounts compared to error-based classification according to Domingos (1999).

The authors collected two variants of the bank marketing data set to predict whether a client subscribes to a term deposit in Portuguese financial institution (Krishna et al., 2019). The two experimental results show that the Deep Neural Network classifier has outperformed four existing classifiers. On the first data set after preprocessing, researchers applied feature selection using attribute subset selection (that include Forward Selection, Backward Elimination, Decision Tree), a method based on information gain parameter. The top 10 features have been chosen. Results are as follows: Decision Tree 88.99%, Naïve Bayes 86.64%, Support Vector Machines 89.82%, K-NN 88.65%, Deep Neural Network 91.15%. On the second data set after preprocessing, researchers applied feature extraction using Principal component analysis method based on a cumulative variance parameter. The top 3 principal have been chosen. Results are as follows: Decision Tree 83.15%, Naïve Bayes 84.16%, Support Vector Machines 87.95%, K-NN 86.76% (Krishna et al., 2019).

Feature contribution on the subscription success, campaign and customer profiling.

Olatunji, divided attributes to two groups, numerical and categorical, then he tested their contribution to success of deposit subscription. By running numerical features on Decision Tree (DT), Olatunji found that feature duration is the root node that is the most important variable. The next feature selected by DT were poutcome, then month and contact. Similar results were obtained for the Logistic Regression. The author also demonstrated categorical features contribution by conducting Correspondence analysis. Results showed that customers in the management and technician cadre responded positively to a term deposit. Divorced and single clients responded more positively to the campaign than married clients. Clients with post-secondary education have a better subscription percentage than customers with elementary education. The months of September, November, March, and April were found to have higher subscription rates than other months. Finally, clients without bank loans were more likely to correspond with the "yes” response than those with bank loans (Olatunji, 2016).

Guo et al.’s constructed a Decision Tree algorithm to analyze factors that affect customer subscription to fixed deposits of an Australian bank. The data set was similar to Portuguese banks. The attributes were selected by the highest information gain Entropy. The author found three factors that significantly affect customers' subscriptions are the number of employees, duration, and month, which greatly reduce the range of clients that banks push to subscribe for long-term deposits, and are beneficial to improving the efficiency of banks (Guo et al., 2019).

Parlar et al. used two feature selection methods, information gain and Chi-square, to select the important features. The results were compared with Naive Bayes's supervised machine learning algorithm. This study found that a reduced set of features improves classification performance. The ten highest-ranked features are duration, poutcome, month, pdays, contact, previous, age, job, housing, and balance (Parlar et al., 2017).

Most machine learning algorithms designed for classification assume that there is an equal number of examples for each observed class. This is not always the case in practice, and datasets that have a skewed class distribution are referred to as imbalanced classification problems. Most algorithms are overloaded by the majority class at a time they are learning from highly unbalanced data, so the false-negative (FN) measure is always high. In the past, most researchers have introduced many methods to deal with unbalanced data, most of them focus on resampling techniques, and another one was cost-sensitive learning (CSL). Thai-Nghe et al. demonstrated in their study that one of the technics improve the classifier performance, and another one reduce the misclassification costs (Thai-Nghe et al., 2010). The studies mentioned above showed better performance results when the different classification algorithms were optimized. The most common performance evaluation metric is the AUC, but some authors indicated classification error rates. Random Forest ensemble, Decisions Tree, and Logistic Regression considered as a good classifiers for any type of dataset. Authors found that a reduced set of features improves classification performance and defines importance of attributes that explains the customer behavior on subscriptions for long-term deposits.

Tools and Technics

Data mining refers to extracting or mining knowledge from large amounts of data, which is stored in various repositories. One of the popular tasks of data mining is Classification, which assigns items in a collection to target categories or classes. The goal of classification is to accurately predict the target class for each case in the data (Krishna et al., 2019). Introduction to data mining was reviewed to obtain fundamental concepts and background for understanding each data mining technique, followed by more advanced concepts and algorithms (Steinbach et al., 2005).

Silipo et al. explores some of the most commonly used techniques for dimensionality reduction, for example removing data columns with too many missing values, removing low variance columns, reducing highly correlated columns, applying Principal Component Analysis (PCA), investigating Random Forests, Backward Feature Elimination, and Forward Feature Construction.

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering, and dimensionality reduction via a consistence interface in Python. The Python programming language is one of the most popular languages for scientific computing. Documentation of Scikit-learn provides a ~300 page user guide including narrative documentation, class references, a tutorial, installation instructions, as well as more than 60 examples, some featuring real-world applications. It has a wide assortment of well-established algorithms, with integrated graphics. It's relatively easy to install, learn, and use (https://scikit-learn.org/stable/).

Pandas is an open source, library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. It offers data structures and operations for manipulating numerical tables and time series (https://pandas.pydata.org/docs/).

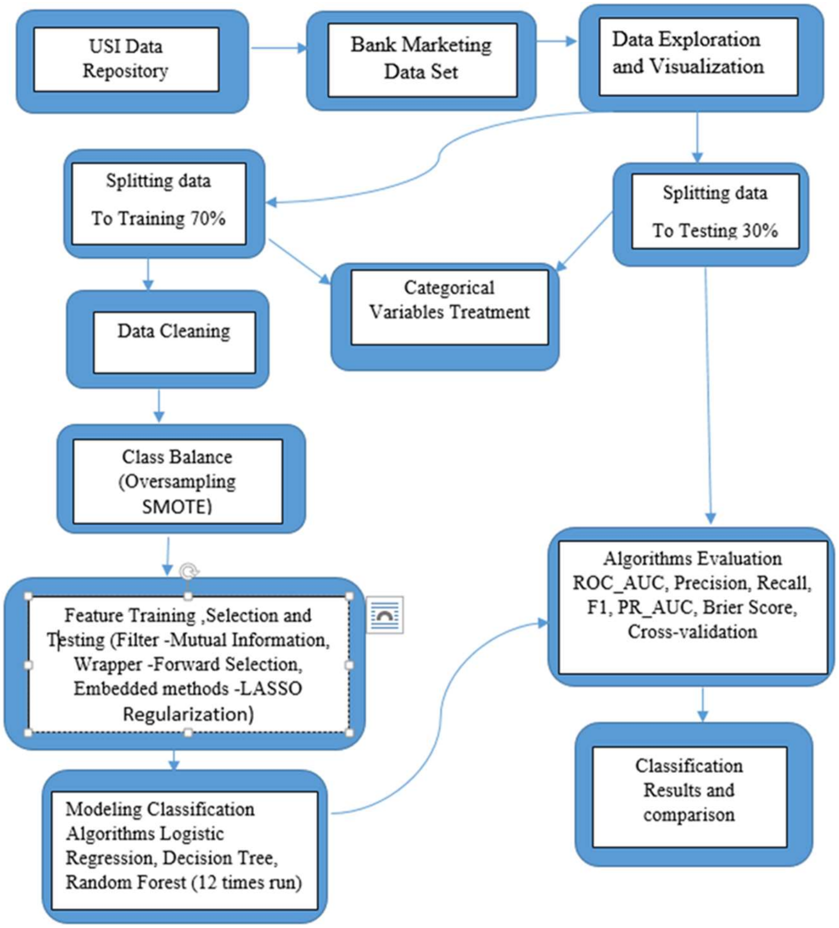
The NumPy library contains multidimensional array and matrix data structures. NumPy can be used to perform a wide variety of mathematical operations on arrays. It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices and it supplies an enormous library of high-level mathematical functions that operate on these arrays and matrices (https://numpy.org/doc/).

Data mining is a useful tool, an approach that combines exploration and discovery with confirmatory analysis. I will use Data mining tools for the exploration and analysis of medium data set in order do to answer research questions. Python is free and available software that will be used for implementing each step of modeling. In order to implement each step in Python, different packages were downloaded and installed as below.

Methodology

To answer the given research question, methods on Figure 2.0 would be most appropriate to find an answer. Literature on related topics suggests that those methods are most appropriate.

Figure 2.0: Overall Research Procedure


USI Data Repository
Bank Marketing Data Set
Data Exploration and Visualization
Splitting data To Training 70%
Splitting data To Testing 30%
Categorical Variables Treatment
Data Cleaning
Class Balance (Oversampling SMOTE)
Feature Training ,Selection and Testing (Filter -Mutual Information, Wrapper -Forward Selection, Embedded methods -LASSO Regularization)
Modeling Classification Algorithms Logistic Regression, Decision Tree, Random Forest (12 times run)
Algorithms Evaluation ROC_AUC, Precision, Recall, F1, PR_AUC, Brier Score, Cross-validation
Classification Results and comparison

I will be running for each Feature Selection technic RF, DT and LR and separately running all features without Feature selection (12 times RF, DT and LR will run) than I will built 2X2 confusion matrix and calculate the rates as follows ROC_AUC, Precision, Recall, F1, PR_AUC, Brier Score for comparition of the models results and the best combination will be chosen..

Data Description

The data is related with direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to access if the product (bank term deposit) would be ('yes') or not ('no') subscribed. The classification goal is to predict if the client will subscribe (yes/no) a term deposit (variable y). Data can be found here:

https://archive.ics.uci.edu/ml/datasets/bank+marketing

Data has 41,188 rows and 21 attributes with highly unbalanced class label.

The positive class “Yes” of target variable “Y” is 4,640 observations that is 11.3%

The negative class “No” of target variable “Y” is 36,548 observations that is 89.7%

Input variables (independent variables):

# bank client data:

1 - age (numeric)

2 - job : type of job (categorical: 'admin.','blue-collar','entrepreneur','housemaid','management','retired','self-employed','services','student','technician','unemployed','unknown')

3 - marital : marital status (categorical: 'divorced','married','single','unknown'; note: 'divorced' means divorced or widowed)

4 - education (categorical: 'basic.4y','basic.6y','basic.9y','high.school','illiterate','professional.course','university.degree','unknown')

5 - default: has credit in default? (categorical: 'no','yes','unknown')

6 - housing: has housing loan? (categorical: 'no','yes','unknown')

7 - loan: has personal loan? (categorical: 'no','yes','unknown')

# related with the last contact of the current campaign:

8 - contact: contact communication type (categorical: 'cellular','telephone')

9 - month: last contact month of year (categorical: 'jan', 'feb', 'mar', ..., 'nov', 'dec')

10 - day_of_week: last contact day of the week (categorical: 'mon','tue','wed','thu','fri')

11 - duration: last contact duration, in seconds (numeric). Important note: this attribute highly affects the output target (e.g., if duration=0 then y='no'). Yet, the duration is not known before a call is performed. Also, after the end of the call y is obviously known. Thus, this input should only be included for benchmark purposes and should be discarded if the intention is to have a realistic predictive model.

Other attributes:

12 - campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)

13 - pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted)

14 - previous: number of contacts performed before this campaign and for this client (numeric)

15 - poutcome: outcome of the previous marketing campaign (categorical: 'failure','nonexistent','success')

# social and economic context attributes

16 - emp.var.rate: employment variation rate - quarterly indicator (numeric)

17 - cons.price.idx: consumer price index - monthly indicator (numeric)

18 - cons.conf.idx: consumer confidence index - monthly indicator (numeric)

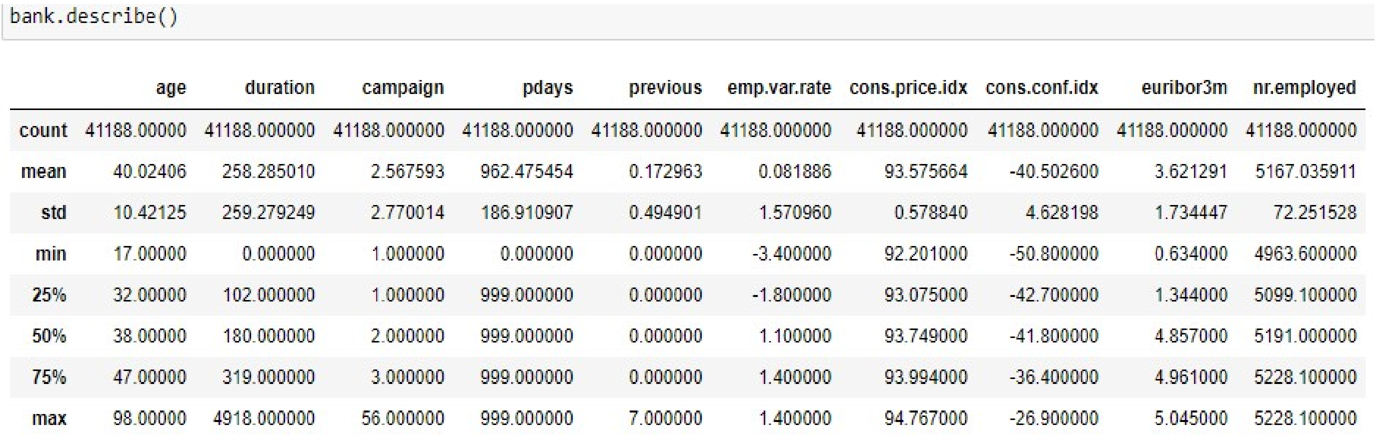
19 - euribor3m: euribor 3 month rate - daily indicator (numeric)

20 - nr.employed: number of employees - quarterly indicator (numeric)

Output variable (dependent variable):

21 - y - has the client subscribed a term deposit? (binary: 'yes','no')

Below you can see descriptive statistics for numerical variables.



```
bank.describe()
```

	age	duration	campaign	pdays	previous	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed
count	41188.00000	41188.000000	41188.000000	41188.000000	41188.000000	41188.000000	41188.000000	41188.000000	41188.000000	41188.000000
mean	40.02406	258.285010	2.567593	962.475454	0.172963	0.081886	93.575664	-40.502600	3.621291	5167.035911
std	10.42125	259.279249	2.770014	186.910907	0.494901	1.570960	0.578840	4.628198	1.734447	72.251528
min	17.00000	0.000000	1.000000	0.000000	0.000000	-3.400000	92.201000	-50.800000	0.634000	4963.600000
25%	32.00000	102.000000	1.000000	999.000000	0.000000	-1.800000	93.075000	-42.700000	1.344000	5099.100000
50%	38.00000	180.000000	2.000000	999.000000	0.000000	1.100000	93.749000	-41.800000	4.857000	5191.000000
75%	47.00000	319.000000	3.000000	999.000000	0.000000	1.400000	93.994000	-36.400000	4.961000	5228.100000
max	98.00000	4918.000000	56.000000	999.000000	7.000000	1.400000	94.767000	-26.900000	5.045000	5228.100000

Figure 3.0: Descriptive statistics

SD bigger then mean (duration, campaign, previous, emp.var.rate, cons.conf.idx) - high variation between values, and abnormal distribution for data. A smaller standard deviation indicates that more of the data is clustered about the mean while a larger once indicates the data are more spread out.

What Is a Term Deposit?

A term deposit is a fixed-term investment that includes the deposit of money into an account at a financial institution. Term deposit investments usually carry short-term maturities ranging from one month to a few years and will have varying levels of required minimum deposits.

The investor must understand when buying a term deposit that they can withdraw their funds only after the term ends. In some cases, the account holder may allow the investor early termination—or withdrawal—if they give several days notification. Also, there will be a penalty assessed for early termination (https://www.investopedia.com/terms/t/termdeposit.asp).

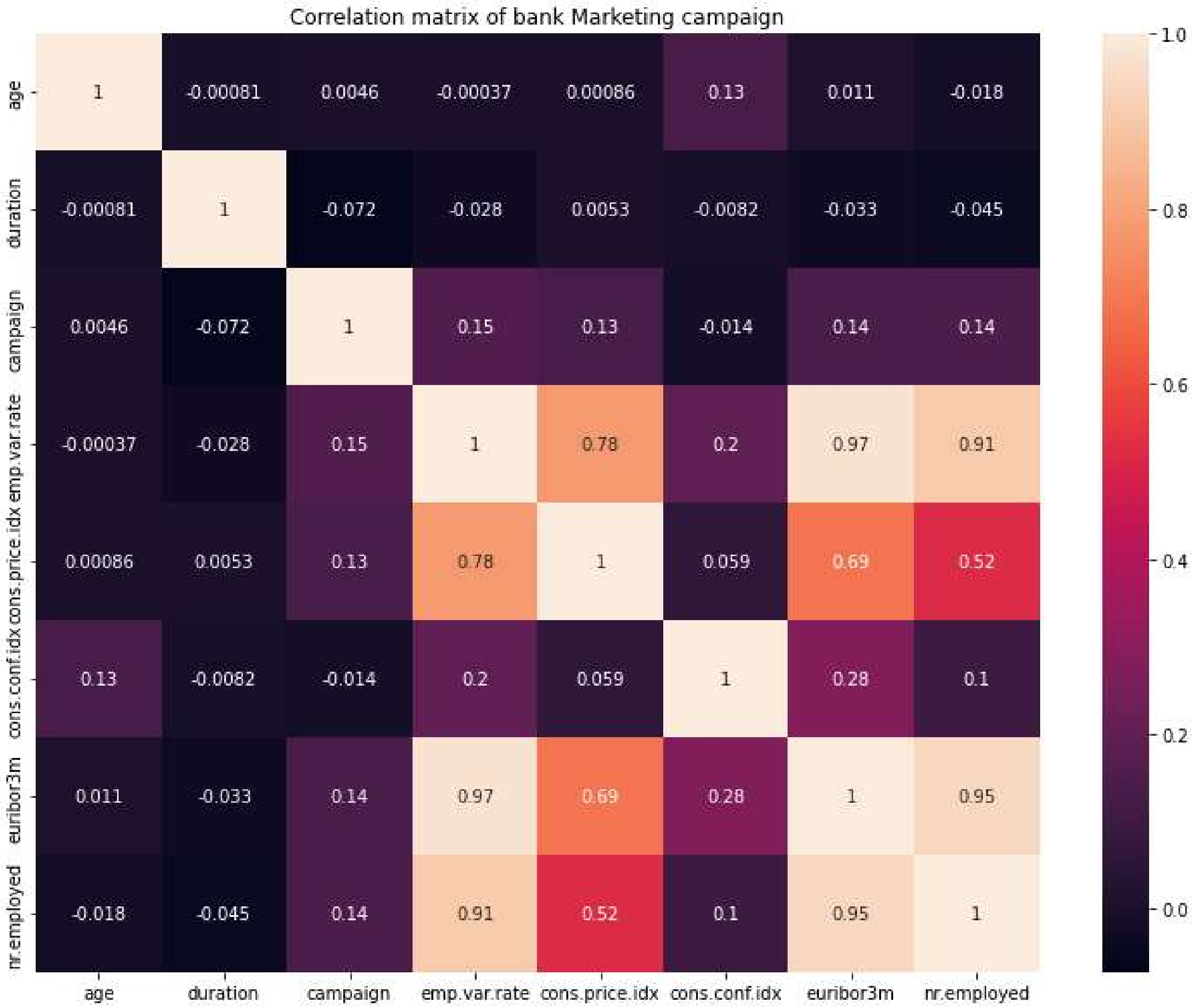
Section 3: Final Results and Project Report

Exploratory Data Analysis and Cleaning

There are no missing values in the data set, but there was 0 values found in attributes, appearing 4 times in “duration”, 15 times in “pdays”, and 35,563 times in “previous”. For “duration”, I calculated the mean of 258.29 and I replaced all nulls with the mean of that column. I decided to drop the attribute of “pdays”, as there is not enough information for further analysis. It is observed that 999 makes 96% of the values of the column, from attribute information 999 means the client was not previously contacted. The “previous” attribute has null 35,563 from the total observation of 41,188, so I dropped this attribute off. In this research I used 2 same datasets that first one was used to train and test the model and the second data set was used to validate the model.

Correlation

The dataset has been cleansed by removing attributes that show a high correlation 'emp.var.rate','euribor3m'. 'cons.conf.idx' which has a weak correlation but was deleted because I had an error with this variable when modeling. The social and economic context attributes have correlation among themselves. All columns with a high correlation will be removed to prevent Multicollinearity, it happens when predictor variable can be linearly predicted from the others with a high degree of accuracy. This can lead to skewed or misleading results.



	age	duration	campaign	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed
age	1	-0.00081	0.0046	-0.00037	0.00086	0.13	0.011	-0.018
duration	-0.00081	1	-0.072	-0.028	0.0053	-0.0082	-0.033	-0.045
campaign	0.0046	-0.072	1	0.15	0.13	-0.014	0.14	0.14
emp.var.rate	-0.00037	-0.028	0.15	1	0.78	0.2	0.97	0.91
cons.price.idx	0.00086	0.0053	0.13	0.78	1	0.059	0.69	0.52
cons.conf.idx	0.13	-0.0082	-0.014	0.2	0.059	1	0.28	0.1
euribor3m	0.011	-0.033	0.14	0.97	0.69	0.28	1	0.95
nr.employed	-0.018	-0.045	0.14	0.91	0.52	0.1	0.95	1

Figure 4.0: Correlation matrix

Following is the matrix that shows correlation between the attributes after removing the high correlated ones:

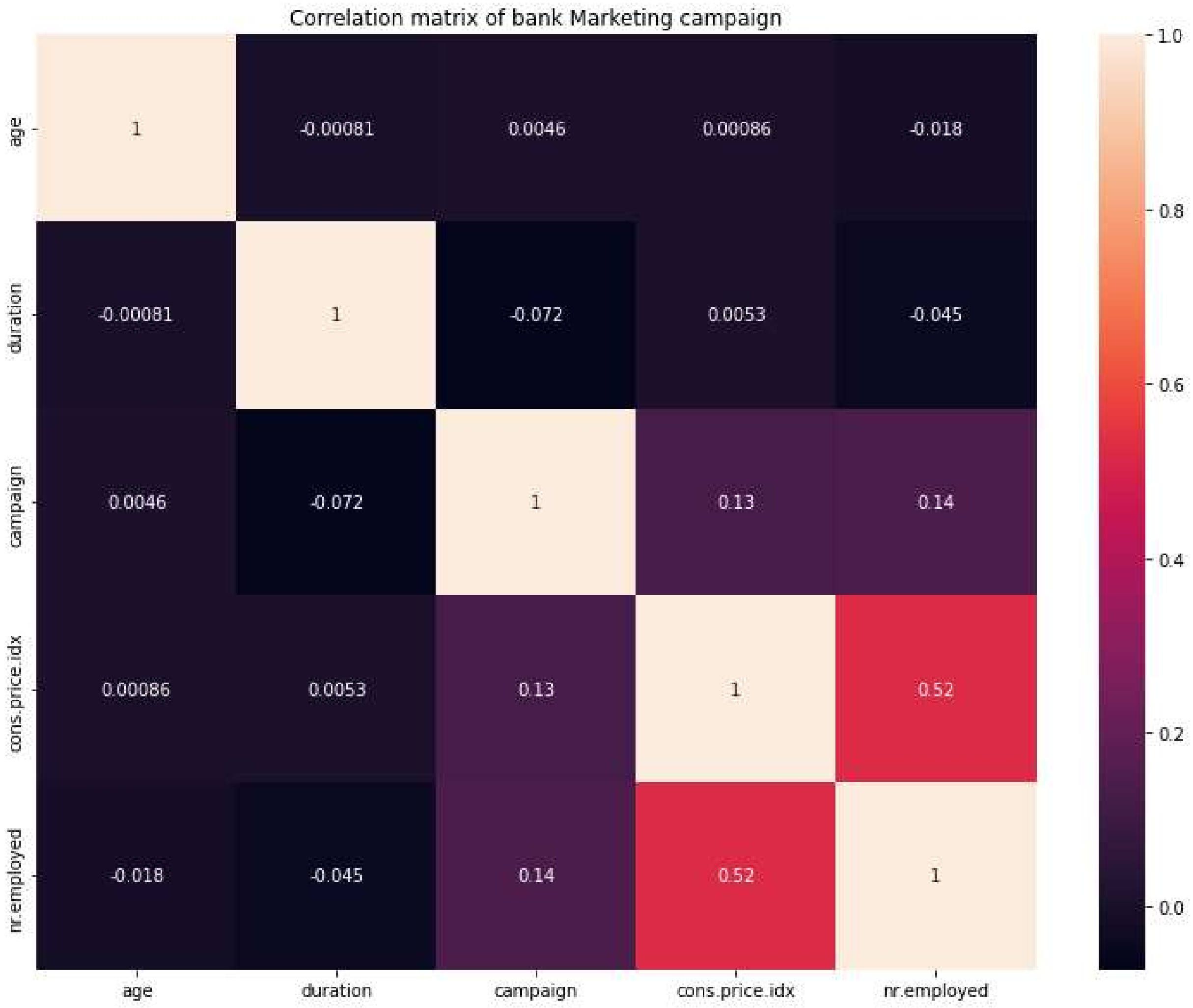


Figure 5.0: Correlation matrix

Creating different data samples for training and testing.

In this way, we can use the training set for training our model and testing set to help evaluate whether the model can generalize well to new, unseen data. I would divide the data set into 2 portions in a ratio of 70:30. My target variable is ‘y’ which is included in training and testing dataset samples and will be further separated from the training and testing sets. Training examples include 28,831 observations and testing examples include 12,357 observations. The cleaning was done on the training set, while the testing set stayed raw for testing proposes. All duplicates, outliers, and “unknown” categories were deleted only from the training set to prevent linkage of knowledge to the test set.

Outliers Treatment

Outliers are observations that are numerically distant from the rest of the data. When reviewing the box plots, as shown below, there are data points located outside of the so called “whiskers” of the boxplot i.e. outside 1.5 times the interquartile range above the upper quantile and below the lower quantile. On the boxplot below looks like there are outliers for “age”, “duration” and “campaign”. The values of the attribute “age” are appropriate for the context of the attribute, for example, minimum 17 and maximum 98. “Duration” is the last contact to the client in seconds. For example, in my data the maximum value is 4,918 which is 82 minutes for call and it's too long. The maximum of “campaign” is 56 calls to the same customer, which is very high. Standard deviations are bigger than the means of “duration” and “campaign”. There are high variation between values and abnormal distribution for data. I removed only the 10th percentile and the 90th percentile because different deletion percentiles caused algorithms to perform worse. I recommend removing the percentile 10 and percentile 90 outliers as they could be an indication of incorrectly collected information. I believe that the sample size is not materially impacted by dropping these questionable outliers. Percentile deletion was tested on algorithms performants. These found that the deletion of outlier’s of percentile 10 and percentile 90 are the perfect numbers for this data set.

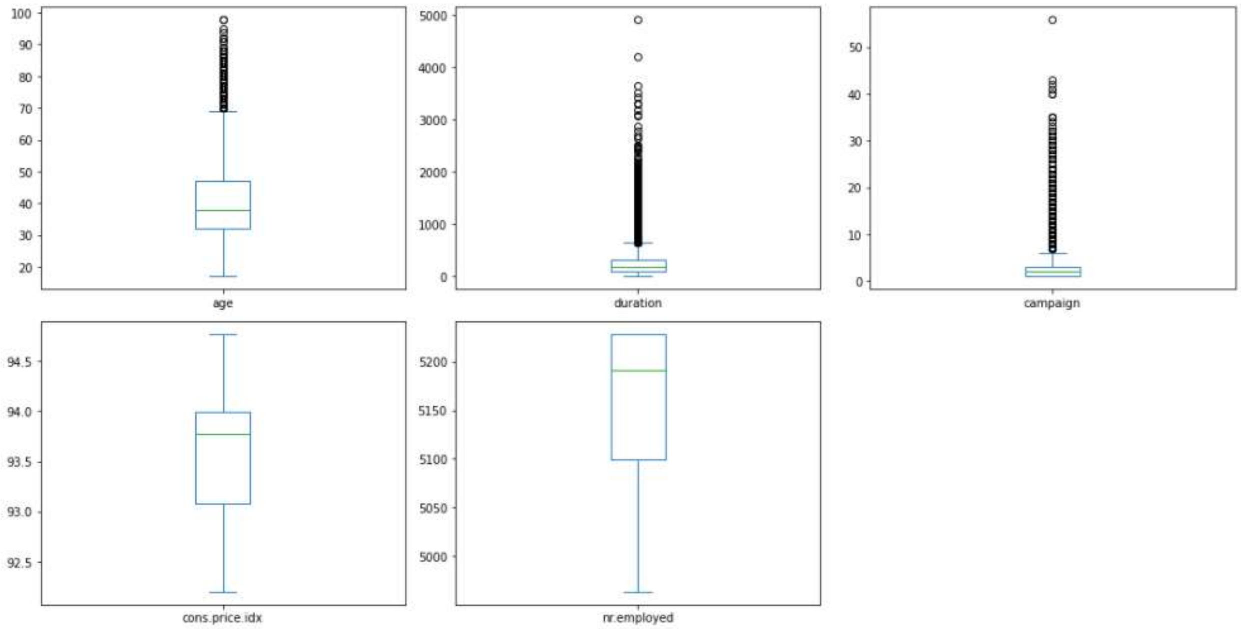
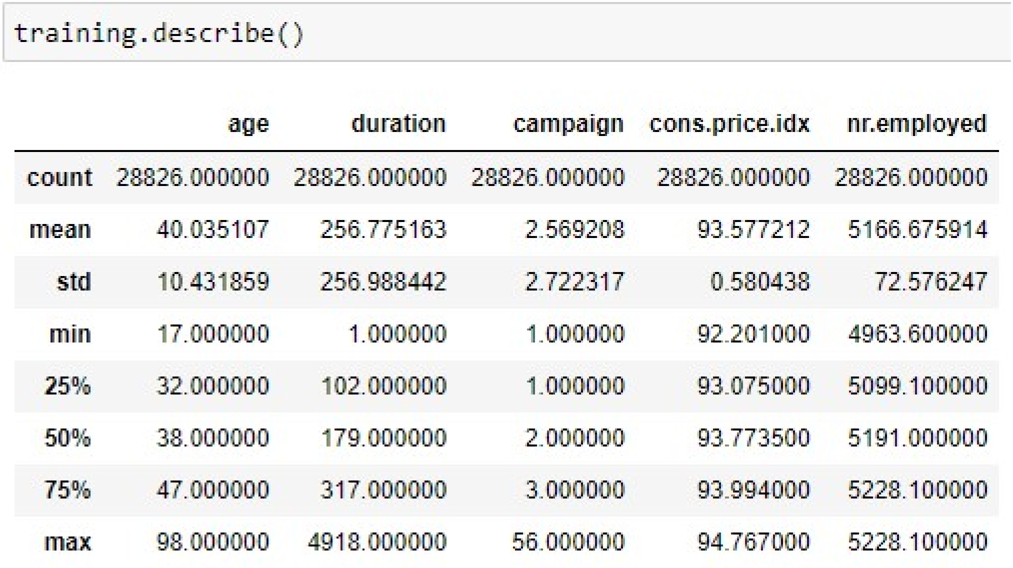


Figure 6.0: Outliers box plots

Below is the descriptive statistics before removing outliers on “duration” and “campaign”.

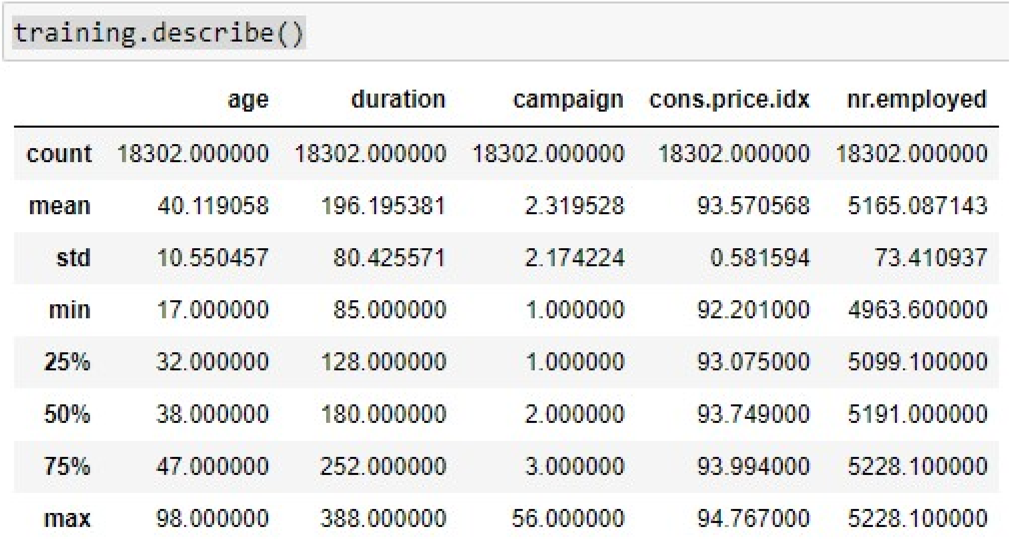


```
training.describe()
```

	age	duration	campaign	cons.price.idx	nr.employed
count	28826.000000	28826.000000	28826.000000	28826.000000	28826.000000
mean	40.035107	256.775163	2.569208	93.577212	5166.675914
std	10.431859	256.988442	2.722317	0.580438	72.576247
min	17.000000	1.000000	1.000000	92.201000	4963.600000
25%	32.000000	102.000000	1.000000	93.075000	5099.100000
50%	38.000000	179.000000	2.000000	93.773500	5191.000000
75%	47.000000	317.000000	3.000000	93.994000	5228.100000
max	98.000000	4918.000000	56.000000	94.767000	5228.100000

Figure 7.0: Descriptive statistics before removing outliers

Following is descriptive statistics data after removing outliers on “duration” and “campaign”.



```
training.describe()
```

	age	duration	campaign	cons.price.idx	nr.employed
count	18302.000000	18302.000000	18302.000000	18302.000000	18302.000000
mean	40.119058	196.195381	2.319528	93.570568	5165.087143
std	10.550457	80.425571	2.174224	0.581594	73.410937
min	17.000000	85.000000	1.000000	92.201000	4963.600000
25%	32.000000	128.000000	1.000000	93.075000	5099.100000
50%	38.000000	180.000000	2.000000	93.749000	5191.000000
75%	47.000000	252.000000	3.000000	93.994000	5228.100000
max	98.000000	388.000000	56.000000	94.767000	5228.100000

Figure 8.0: Descriptive statistics after removing outliers

When comparing the before and after cleansed datasets for each attribute, it is noticeable that by removing the outliers, the minimum values are increasing and the maximum values are decreasing. These bring both the minimum and maximum closer to the mean, reducing the standard deviation for “duration” and for “campaign”. Therefore, the data is more normally distributed after removing outliers, bringing the shape of the training data to 15,371 rows and 16 attributes.

Visualization of the relationship between feature category and dependent variable y

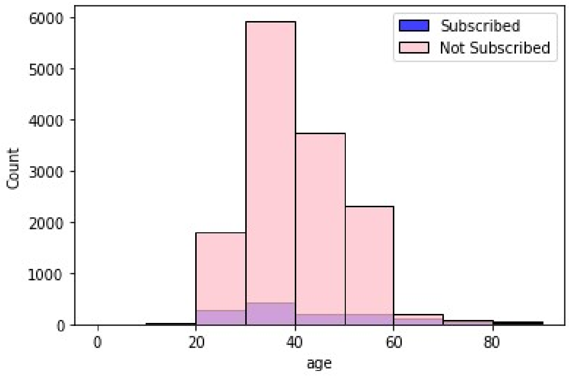


Figure 9.0: Age and deposit (y)

Customers aged 30-40, 20-30, and 40-50 had a higher percentage of subscription to a deposit account.

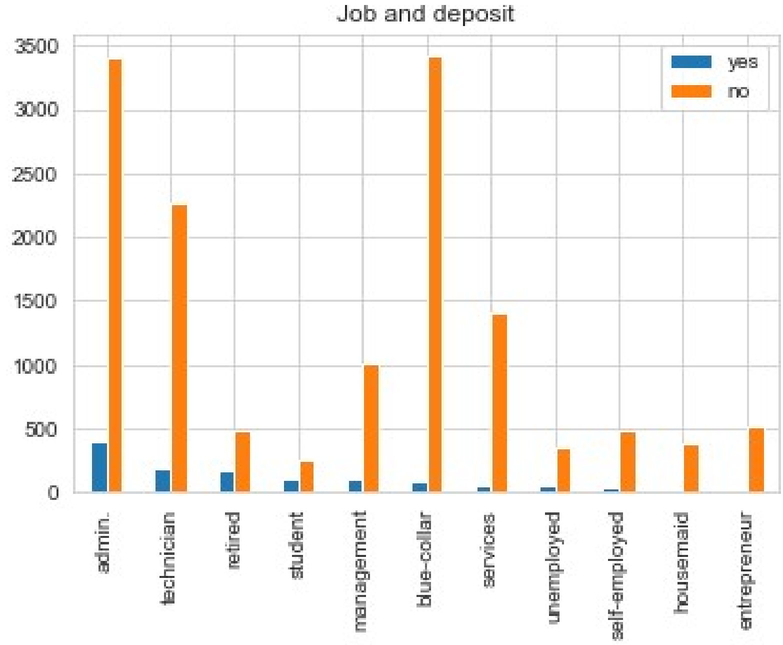


Figure 10: Job and deposit(y)

Customers who worked in administrative position followed by technicians and blue collar had a higher percentage of subscription to a deposit account.

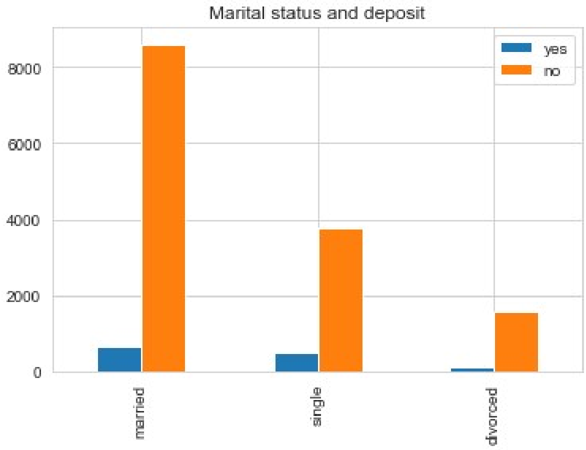


Figure 11: Marital status and deposit(y)

Married customers followed by single customers had a higher percentage of subscription to a deposit account.

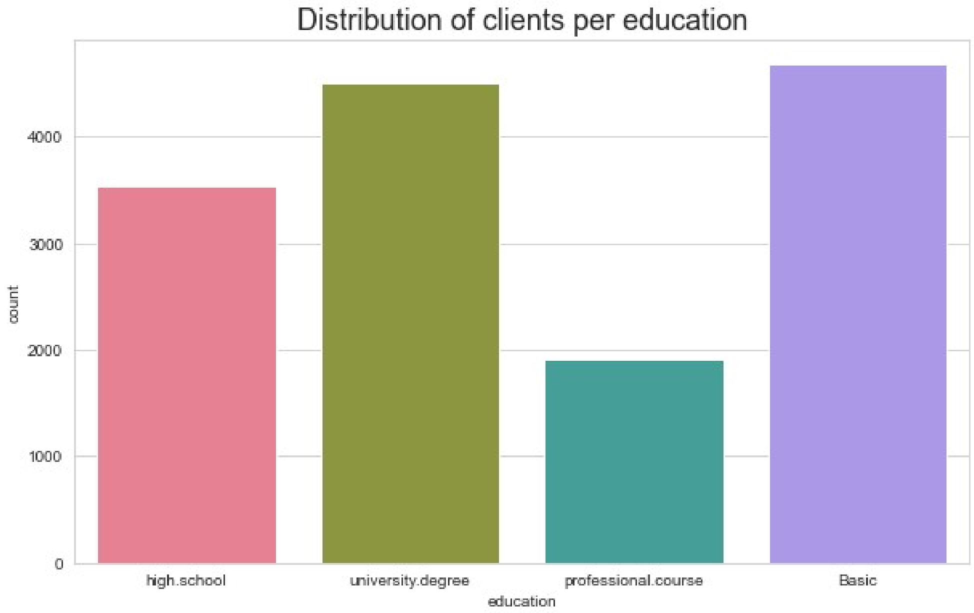


Figure 12: Education and deposit(y)

I deleted illiterate and unknown as it had small quantities and then I grouped basic.4y", "basic.9y" and "basic.6y" together and called them "basic".

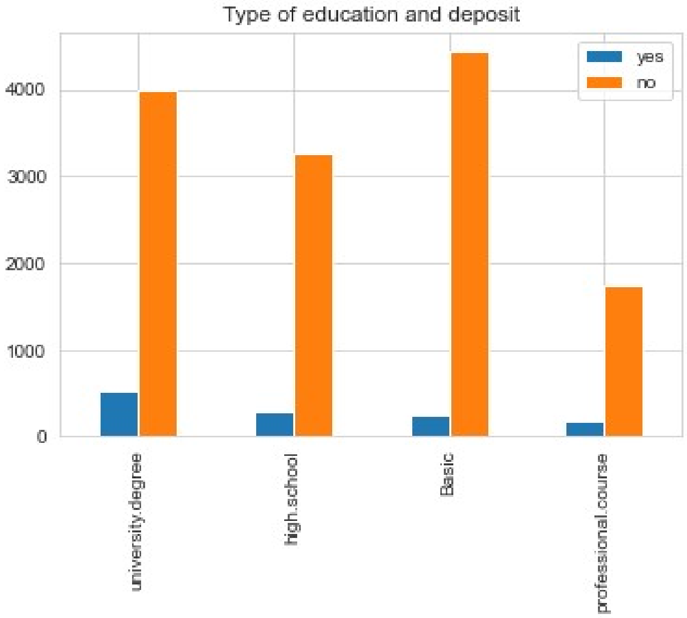


Figure 13: Education and deposit(y)

Customers who had a university degree followed by high school diploma had a higher chance of making a deposit.

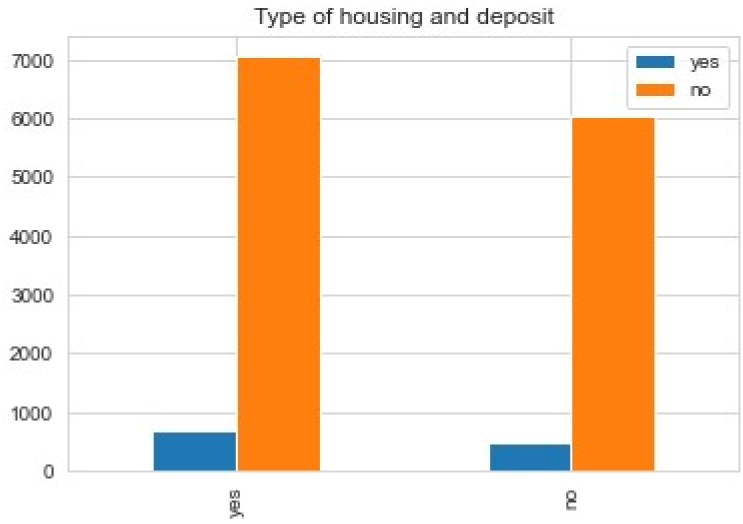


Figure 14: Housing and deposit(y)

Customers who live in a house had a higher chance in making a deposit.

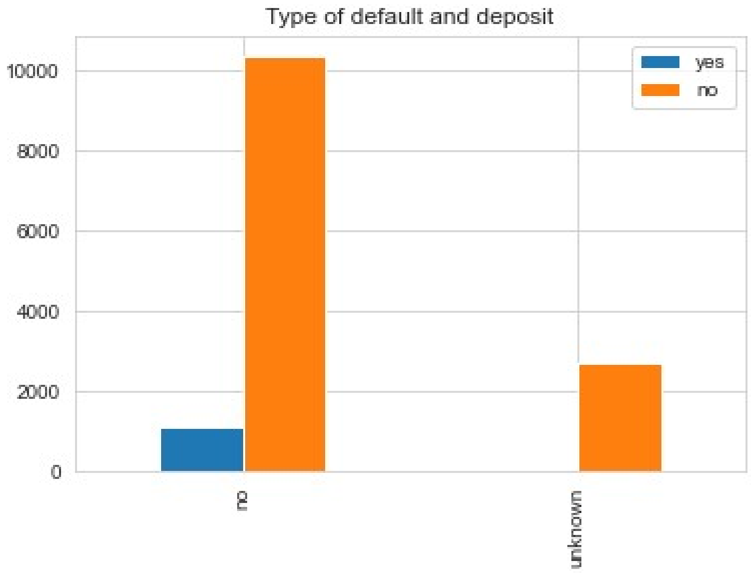


Figure 15: Default (has credit in default) and deposit(y)

Customers who had no default had a higher chance in making a deposit.

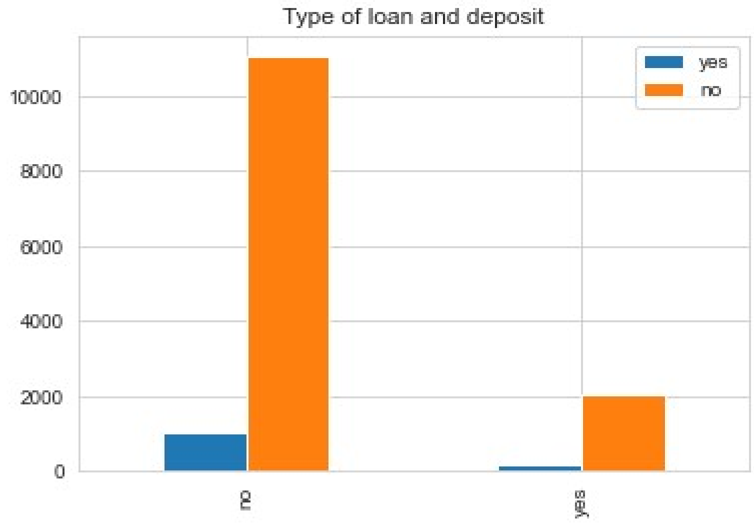


Figure 16: Loan and deposit(y)

Unknown category was dropped.

Clients that had no loan had a higher chance to subscribe to term deposits.

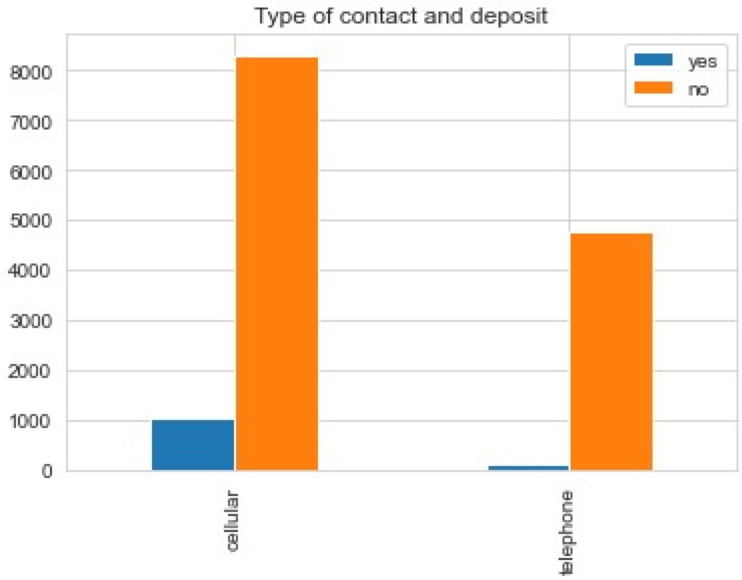


Figure 17: Type of contact and deposit(y)

Clients that were contacted by cellular had a higher chance to subscribe for a term deposit.

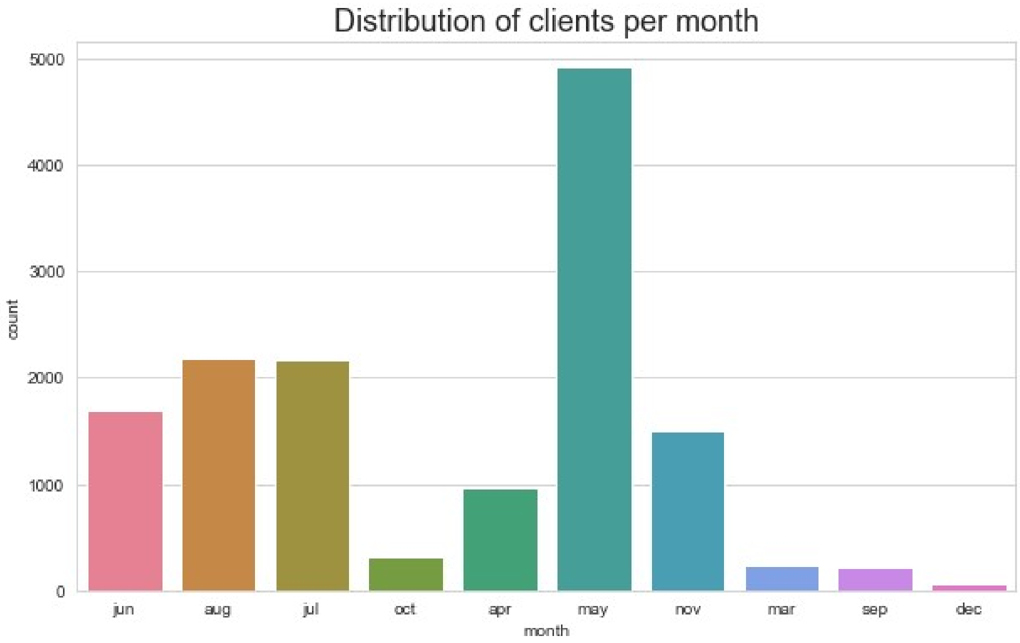
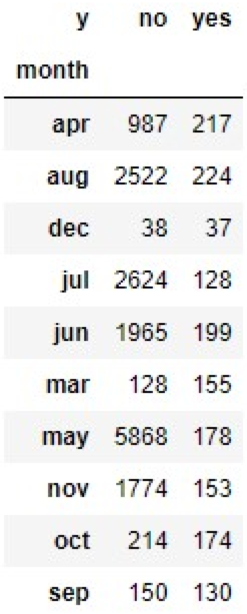


Figure 18: Distribution of clients per month

Month and deposit (y)



y	no	yes
month		
apr	987	217
aug	2522	224
dec	38	37
jul	2624	128
jun	1965	199
mar	128	155
may	5868	178
nov	1774	153
oct	214	174
sep	150	130

Figure 19: Distribution of clients per month

Most of the deposits were made during August following April and June.

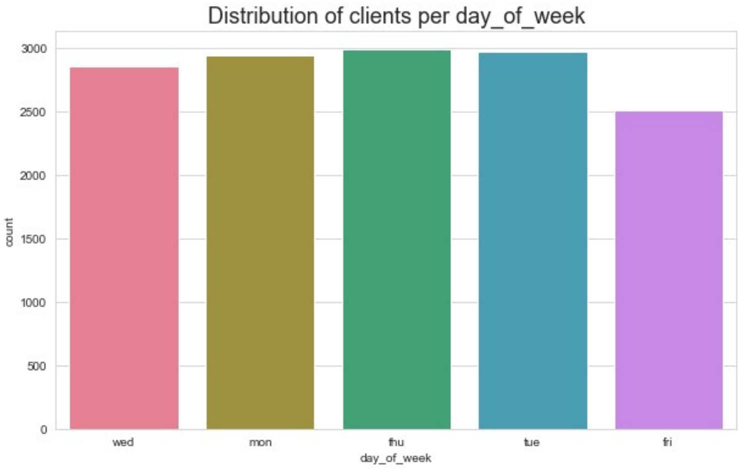
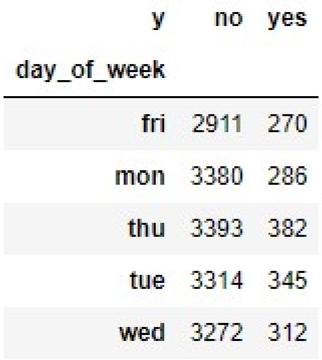


Figure 20: Distribution of clients per day of week



y	no	yes
day_of_week		
fri	2911	270
mon	3380	286
thu	3393	382
tue	3314	345
wed	3272	312

Figure 21: Day of the week and deposit (y)

Less deposits were made on Friday followed by Monday.

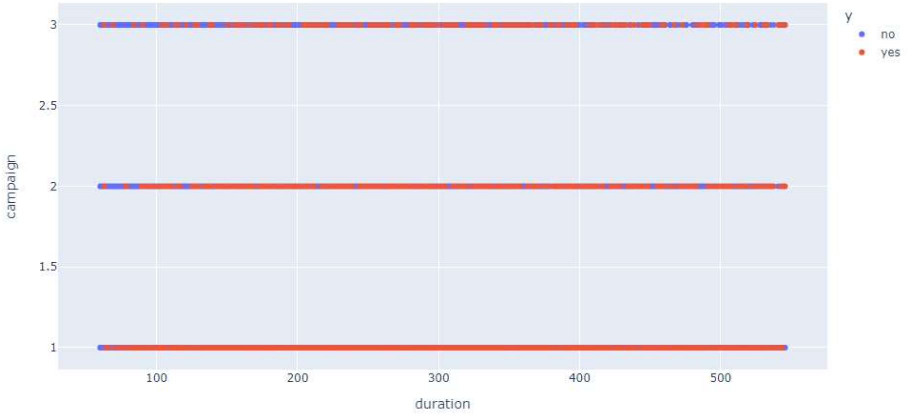


Figure 22: Duration and Campaign by deposit (y)

“Campaign” is the number of contacts performed and “duration” is the call per second. From the graph above I can see that as more employees contacted customers, the less likely they made a deposit.

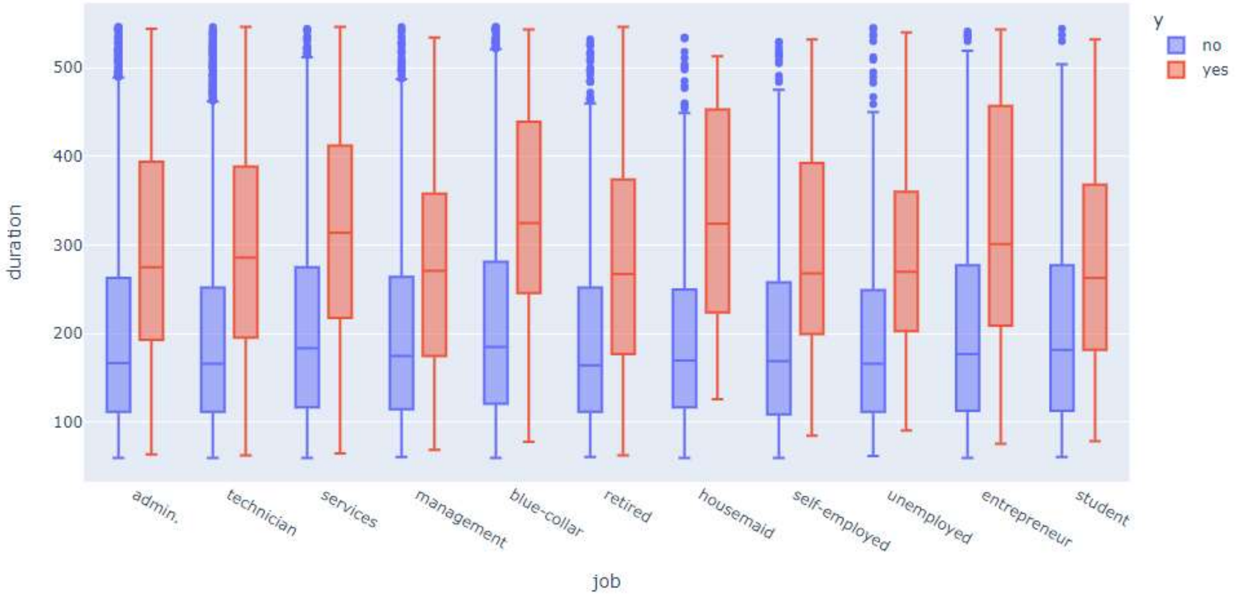


Figure 22: Job and duration by deposit(y)

The longer the conversation was with clients, the more likely they were to make a deposit.

The median of the blue collar, entrepreneur, and services had the higher duration of calls.

Categorical Treatment

The dataset contains 11 object type variables. The month and the day were converted by its corresponding number for training and testing set. All other variables were converted to numerical labels using a custom function from Scikit-learn preprocessing tool for training and testing sets allowing further data-analysis.

Scaling I have tried to rescale with Standard Scaler which is centering the variable at zero and standardizing the variance at 1, was no effect on the algorithms. I have tried Power Transformer with the method='yeo-johnson', had no prediction of class 1 of Recall and Precision whech was not working well with these dataset. Normalization which is Min-Max Scalar technic will not work as this data doesn't need to suppress outliers, I already dropped outliers.

Imbalanced class distribution

The dataset has an imbalanced class distribution with the majority of the class being the “no” that is 89% and minority class of “yes that is 11%. Working with imbalanced datasets can be problematic if there are too few examples of the minority class to incorporate into the decision boundary. Subsequently, the model becomes extremely good at predicting the majority class but does not do so well with the minority class. I improved models' performance by balancing the dataset, so it has equal numbers of both classes. To do this, I implemented an “oversampling” technique, which is achieved by oversampling the minority class which was applied only on the training dataset so the knowledge will not leak to the test set. Examples are drawn from the minority class and duplicated to match their occurrence with the majority class. This only serves to balance the dataset without amending or including any additional information. The most widely used algorithm that I have used for my dataset is called the Synthetic Minority Oversampling Technique, or SMOTE algorithm that was applied on training data, so no information goes to testing data. The purpose of resampling the training data is to better represent the minority class so the classifier would have more samples to learn from. Not only must the test data be untouched during oversampling but also validation data.

Before SMOTE the shape of y_train was for 0 class “no” 16,270 and 1 class “yes” 1,595

After SMOTE the shape of y_train was for 0 class “no” 16,270 and 1 class “yes” 16,270

Modeling

Classification algorithms

Three classification algorithms have been selected to predict future subscription:

- Random Forest (RF)
- Decision Tree (DT)
- Logistic Regression (LR)

The Random Forest algorithm is an ensemble method used mainly for classification and regression. Random Forests grow a multitude of decision trees. Each tree gives a classification, and “votes” for that class, after which the classification with the most votes is selected from all the trees within the “forest”. Random Forests do not over fit as decision trees and are able to balance error in classification caused by imbalanced data sets. RF works well with both categorical and continuous variables since no feature scaling is required. Likewise, it handles non-linear parameters efficiently, algorithm is very stable. Random Forest is comparatively less impacted by noise on the other hand it complex and requires much more computational power and resources.

Decision Tree is a very popular machine learning algorithm. DT solves the problem of machine learning by transforming the data into a tree representation. Each internal node of the tree representation denotes an attribute and each leaf node denotes a class label. A decision tree algorithm can be used to solve both regression and classification problems. It can handle both continuous and categorical variables, is simple and easy to understand, has no feature scaling required, and the non linear parameters don't affect the performance of a DT unlike curve based algorithms. In addition, there is less training period compared to Random Forest, but on the other hand, the main problem of the Decision Tree that it is overfitting, unstable, is affected by noise, is a weak learner, and not suitable for large datasets.

Logistic Regression is a classification algorithm used to find the probability of event success and event failure. It is used when the dependent variable is binary (0/1) in nature. It supports categorizing data into discrete classes by studying the relationship from a given set of labelled data. It learns a linear relationship from the given dataset and then introduces a non-linearity in the form of the Sigmoid function or also known as the ‘logistic function’ instead of a linear function. The hypothesis of logistic regression tends it to limit the cost function between 0 and 1. Therefore linear functions fail to represent it as it can have a value greater than 1 or less than 0 which is not possible as per the hypothesis of logistic regression. Logistic regression is also known as Binomial Logistic regression and it is It is very fast at classifying unknown records, good accuracy for many simple data sets and it performs well when the dataset is linearly separable. On the other hand, the major limitation of Logistic Regression is the assumption of linearity between the dependent variable and the independent variables. It can only be used to predict discrete functions.

Feature Selection Technics

Feature Selection is the process where you automatically or manually select those features which contribute most to your prediction variable or output. Data features used to train machine learning models and have a huge influence on the performance of machine learning models. Performing feature selection before modeling will benefit modeling with reduced overfitting, improved accuracy, and reduced training time. The feature selection algorithms can be divided in to three categories: filter method, wrapper method, and embedded method.

Filter method pick up the intrinsic properties of the features that is the “relevance” of the features, measured via univariate statistics instead of cross-validation performance. Individual features are ranked according to specific criteria. The top N features are then selected. Different types of ranking criteria are used for univariate filter methods, for example, mutual information gain (IG) which was used in my research. One of the major disadvantages of univariate filter methods is that they may select redundant features because the relationship between individual features is not taken into account.

Wrapper method require some method to search the space of all possible subsets of features, assessing their quality by learning and evaluating a classifier with that feature subset. For example, I used forward feature selection with the algorithm Random Forest Classifier. Forward feature selection is used to select the best important features from the bank marketing dataset concerning the target output. It is an iterative method in which we start having no feature in the model. In each iteration, we keep adding the feature which best improves our model till an addition of a new variable does not improve the performance of the model. The evaluation criteria are AUC.

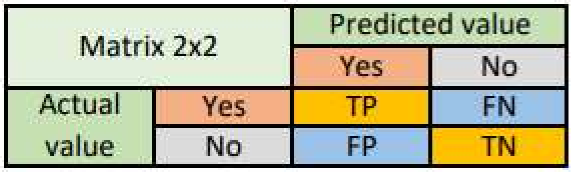
Embedded method contain the benefits of both the wrapper and filter methods, by including interactions of features but also maintaining reasonable computational cost. Embedded methods are iterative in the sense that takes care of each iteration of the model training process and carefully extract those features which contribute the most to the training for a particular iteration. In my research I used LASSO Regularization (L1) as it works better and achieve best results for my dataset compered to RIDGE Regularization (L2). Regularization consists of adding a penalty to the different parameters of the machine learning model to reduce the freedom of the model that is to avoid over-fitting. The penalty is applied over the coefficients that multiply each of the predictors. L1 has the property that is able to shrink some of the coefficients to zero. So, that feature can be removed from the model.

Approach

Divided dataset by technic of Train-Test Split which is 70% of the data goes towards training algorithms and 30% for testing same algorithms. I built a total of 12 models. I used Train-Test Split to analyze how well our supervised learning models are performing on the dataset that was not part of the data utilized to train the model. First, I run machine learning algorithms with entire features set (Test 1) after cleaning and evaluating its performance using selected metrics on the test set which will be discussed in the section below. Secondly, trained the same classification algorithm on features that were selected by Filter Method of Mutual Information Gain (Test 2) evaluated its performance on the test set using the same selected metrics. Thirdly, trained the same classification algorithm on features that were selected by Embedded Method of LASSO Regularization (L1) (Test 3) evaluated its performance on the test set using the same selected metrics. Finally, trained the same classification algorithm on features that were selected by Wrapper Feature Selection of Forward feature selection (Test 4). Compared the performance of different training models by using selected performance metrics.

Performance measures

A confusion matrix, also known as an error matrix, is a specific table layout that allows visualization of the performance of an algorithm. Each row of the matrix represents the instances in an actual class while each column represents the instances in a predicted class. To measure how well the algorithms are performing and predicting binary class, I applied evaluation matrix 2x2, which shows the correct and incorrect (i.e. true or false) predictions on each class.



Matrix 2x2		Predicted value	
		Yes	No
Actual value	Yes	TP	FN
	No	FP	TN

Figure 23: Confusion matrix 2x2

True positive (TP): Predicting positive class as positive (ok) - indicates that the outcome of the model or predicted value matches that of the actual value.

True negative (TN): Predicting negative class as negative (ok) - also indicates how well the model is performing.

False positive (FP): Predicting negative class as positive (not ok) - measures a mismatch between the actual value and the predicted value by the model. Also known as type I error.

False negative (FN): Predicting positive class as negative (not ok) - measures a mismatch between the actual value and the predicted value by the model. Also known as type II error. Confusion matrix is used to calculate precision and recall. It is not possible to maximize both precision and recall because there is a trade-off between them. Increasing precision decreases recall. Below I will discuss which one is more important for this dataset.

Precision measures how good our model is when the prediction is positive. The focus of precision is positive predictions. It indicates how many positive predictions are true.

$$Precision = \frac{TP}{TP + FP}$$

Precision can be seen as a measure of quality, and recall as a measure of quantity. Higher precision means that an algorithm returns more relevant results than irrelevant ones, and high recall means that an algorithm returns most of the relevant results (whether or not irrelevant ones are also returned). Precision is a good evaluation metric to use when the cost of a false positive is very high and the cost of a false negative is low. FP means, in bank marketing complain, that bank employees will contact clients that are predicted as 1 (“yes”) class but actually, they are class 0 (“no”), and ask them to subscribe to term deposit, so it is inconvenient. But in this case, it is not as high cost as, for example, telling people they were sick when they were not.

Recall measures how good our model is at correctly predicting positive classes. The focus of recall is actual positive classes. It indicates how many of the positive classes the model is able to predict correctly.

$$\textbf{TPR /Recall / Sensitivity} = \frac{\textbf{TP}}{\textbf{TP + FN}}$$

Recall calculates the percentage of actual positives a model correctly identified (True Positive). When the cost of a false negative is high, you should use recall. FN means, in bank marketing complain, that bank employees will not contact clients that are predicted as 0 (‘no”) class but actually they are class 1 (“yes”). They could subscribe to term deposit but no one contacted them so the bank will lose sales. In my knowledge, it is going to be a high cost for the bank to lose potential subscription but on the other hand to bother customers with inconvenienced calls are costly as well. So there is another measure that combines precision and recall into a single number and that is the F1 score.

F1 score represents the harmonic mean of the precision and recall. It is a more useful measure than accuracy for problems with uneven class distribution because it takes into account both false positive and false negatives. For a high F1-score, both precision and recall must be high.

$$F1\_score = 2\,\frac{Precision * Recall}{Precision + Recall}$$

AUC - ROC Curve is a performance measurement for the classification problems at various threshold settings. ROC is a probability curve and AUC represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes. The higher the AUC, the better the model is at predicting 0 classes as 0 and 1 classes as 1. The ROC curve is plotted with TPR against the FPR where TPR is on the y-axis and FPR is on the x-axis. When AUC is approximately 0.5, the model has no discrimination capacity to distinguish between positive class and negative class. When AUC is approximately 0, the model is predicting a negative class as a positive class. For example, the AUC of the best model is 0.88, it means there is a 88% chance that the model will be able to distinguish between positive class and negative class.

$$\text{Specificity} = \frac{TN}{TN + FP} \qquad \text{FPR} = 1 - \text{Specificity} = \frac{FP}{TN + FP}$$

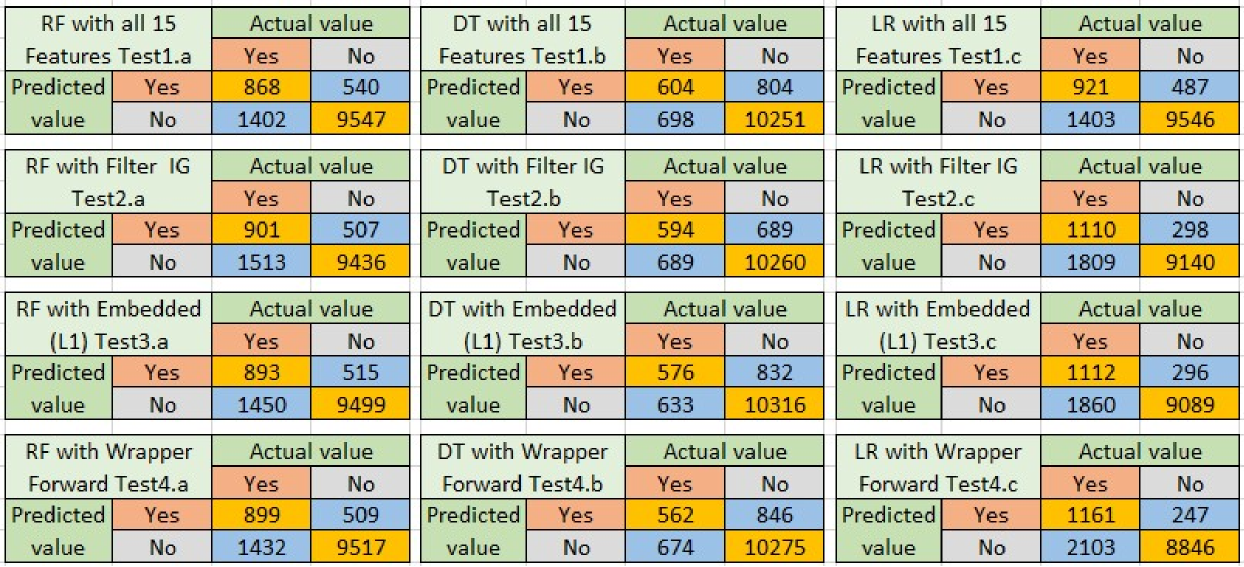
PR AUC | Average Precision for every threshold, you calculate positive predictive value PPV and true positive rate TPR and plot it. The higher on y-axis your curve is the better your model performance. You can also think of PR AUC as the average of precision scores calculated for each recall threshold. I decided to check PR AUC in addition to AUC - ROC Curve because the positive class is more important than negative and the dataset is imbalanced. For example, the rate of PR AUC in my best model is 52%.

F-measure does not, in general, take into account the True Negatives (TN). **Brier score** calculates the mean squared error between predicted probabilities and the expected values (actuals). The Brier Score-perfect skill has a score of 0 and the worst has a score of 1. For example, since the best model Brier score is 0.13, I can infer that this model has a good performance or skill.

The k-fold cross-validation procedure is a method for estimating the performance of a machine learning algorithm. In k-fold cross-validation, the original sample is randomly divided into k equal sized subsamples. Of the k subsamples, a single subsample is retained as the validation data for testing the model, and the remaining k − 1 subsamples are used as training data. Repeated k-fold cross-validation provides a way to improve the estimated performance of a machine learning model. This involves simply repeating the cross-validation procedure multiple times and reporting the mean result across all folds from all runs. I ran cross validation on the whole dataset which was split to X and y. The data set was not cleaned to correctly represent the real life data. The number of attributes and that data type was corrected to match the shape of the file which was used for training and testing purposes.

Section 4: Results of the applied techniques

Below you can see confusion matrix results for all 12 tests which is Figure 24. Tables below describe the performance of the classification models and summarizes the classifier performance. I organized the results of confusion matrix as in the order of the books, therefore, Python coding Sklearn metrics library that was used for these research prints TP, FN, TN, FP in a different order. Features that were selected by Filter Method of Mutual Information Gain (Test 2) are 'age', 'job', 'default', 'contact', 'month', 'duration', 'campaign', 'poutcome', 'cons.price.idx', and 'nr.employed'. Features that were selected by Embedded Methods of LASSO Regularization (Test 3) are 'age',' contact', 'month', 'day _of _week', 'duration', ‘campaign’, ‘cons.price.index’ , ‘nr.employed’, and features that were selected by Wrapper Feature Selection of Forward feature selection (Test 4) which are 'age', 'job', 'marital', 'education', 'month', 'day_of_week', 'duration', 'campaign', 'poutcome', 'cons.price.idx', and 'nr.employed'.



RF with all 15 Features Test1.a		Actual value	
		Yes	No
Predicted value	Yes	868	540
	No	1402	9547

DT with all 15 Features Test1.b		Actual value	
		Yes	No
Predicted value	Yes	604	804
	No	698	10251

LR with all 15 Features Test1.c		Actual value	
		Yes	No
Predicted value	Yes	921	487
	No	1403	9546

RF with Filter IG Test2.a		Actual value	
		Yes	No
Predicted value	Yes	901	507
	No	1513	9436

DT with Filter IG Test2.b		Actual value	
		Yes	No
Predicted value	Yes	594	689
	No	689	10260

LR with Filter IG Test2.c		Actual value	
		Yes	No
Predicted value	Yes	1110	298
	No	1809	9140

RF with Embedded (L1) Test3.a		Actual value	
		Yes	No
Predicted value	Yes	893	515
	No	1450	9499

DT with Embedded (L1) Test3.b		Actual value	
		Yes	No
Predicted value	Yes	576	832
	No	633	10316

LR with Embedded (L1) Test3.c		Actual value	
		Yes	No
Predicted value	Yes	1112	296
	No	1860	9089

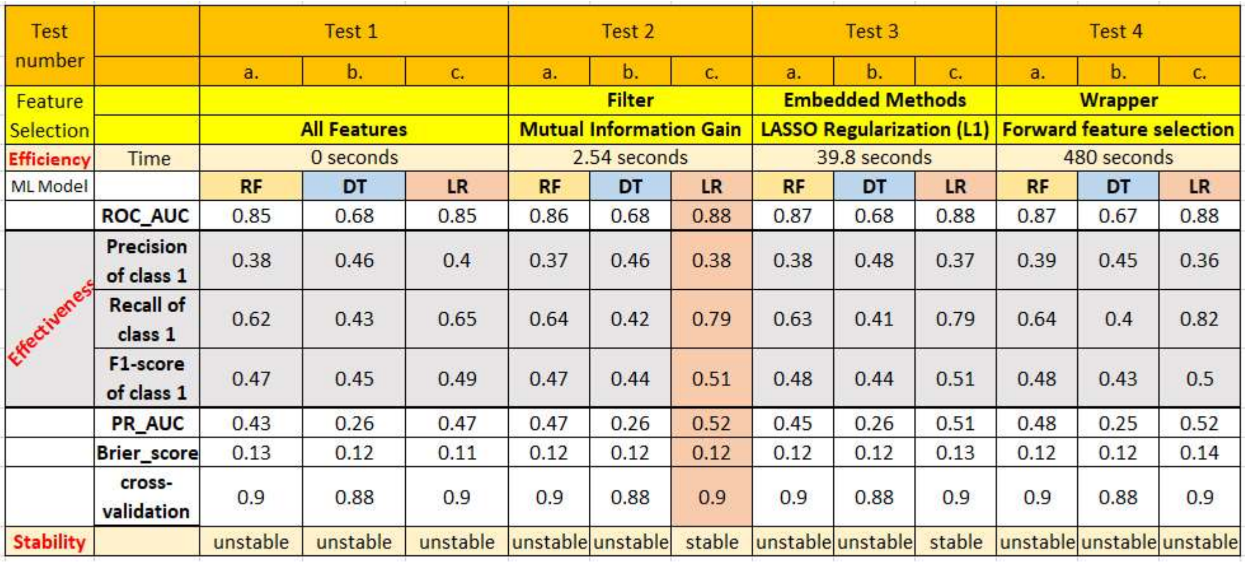
RF with Wrapper Forward Test4.a		Actual value	
		Yes	No
Predicted value	Yes	899	509
	No	1432	9517

DT with Wrapper Forward Test4.b		Actual value	
		Yes	No
Predicted value	Yes	562	846
	No	674	10275

LR with Wrapper Forward Test4.c		Actual value	
		Yes	No
Predicted value	Yes	1161	247
	No	2103	8846

Figure 24: Confusion matrix results for all 12 tests

By looking at Figure 24 we can see 12 binary confusion matrixes which represent the correct and incorrect prediction of 3 classifiers. By looking at the numbers of Test4.c we can see that TP has the highest number but on the other hand, has the highest error FP. To better understand the confusion matrixes I ran Precision, Recall, and F1 score (Figure 25). In test2.c Recall is 79% and Recall of Test4.c is 82% but the Precision is 38% compared with Precision of Test4.c which is 36% but there is a trade-off between those two, one increase and the other one decrease. As explained above in the bank marketing campaign the Recall and Precision are both important because the FP and FN are both costly to the company so the best score to compare tests is the F1 score. Test4.c has 50% of F1 score and Test2.c has 51% which is the highest, therefore, the best performance achieved by Test2.c. By looking at the Test2.c on the Confusion matrix table the total actual yes’s in the dataset is the sum of the values on the yes column (1110 + 1809). The total actual No’s in the dataset is the sum of values in the No column (298 +9140). The correct values are organized in a diagonal line from top left to bottom-right of the matrix (1110+9140). More errors were made by predicting Yes as No than predicting No as Yes.

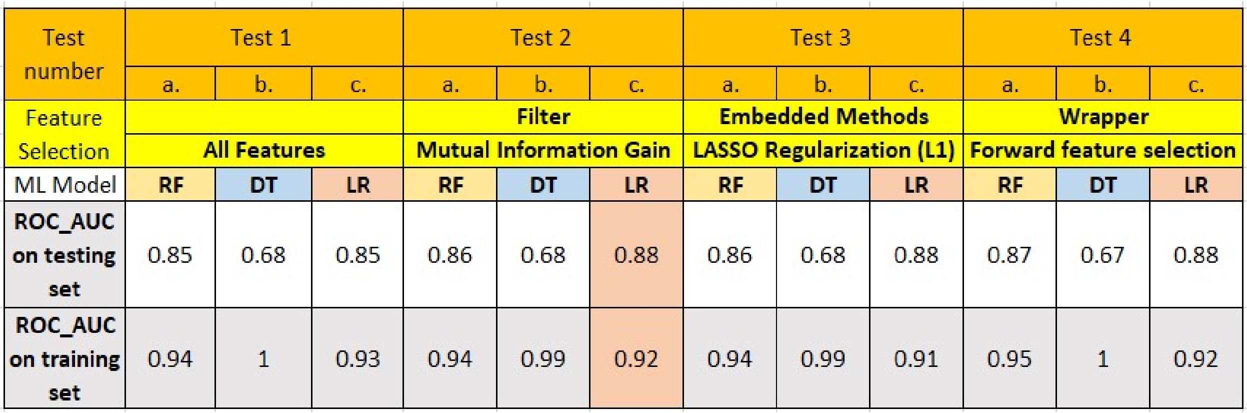


Test number		Test 1			Test 2			Test 3			Test 4		
		a.	b.	c.	a.	b.	c.	a.	b.	c.	a.	b.	c.
Feature					Filter			Embedded Methods			Wrapper		
Selection		All Features			Mutual Information Gain			LASSO Regularization (L1)			Forward feature selection		
Efficiency	Time	0 seconds			2.54 seconds			39.8 seconds			480 seconds		
ML Model		RF	DT	LR	RF	DT	LR	RF	DT	LR	RF	DT	LR
	ROC_AUC	0.85	0.68	0.85	0.86	0.68	0.88	0.87	0.68	0.88	0.87	0.67	0.88
Effectiveness	Precision of class 1	0.38	0.46	0.4	0.37	0.46	0.38	0.38	0.48	0.37	0.39	0.45	0.36
	Recall of class 1	0.62	0.43	0.65	0.64	0.42	0.79	0.63	0.41	0.79	0.64	0.4	0.82
	F1-score of class 1	0.47	0.45	0.49	0.47	0.44	0.51	0.48	0.44	0.51	0.48	0.43	0.5
	PR_AUC	0.43	0.26	0.47	0.47	0.26	0.52	0.45	0.26	0.51	0.48	0.25	0.52
	Brier_score	0.13	0.12	0.11	0.12	0.12	0.12	0.12	0.12	0.13	0.12	0.12	0.14
	cross-validation	0.9	0.88	0.9	0.9	0.88	0.9	0.9	0.88	0.9	0.9	0.88	0.9
Stability		unstable	unstable	unstable	unstable	unstable	stable	unstable	unstable	stable	unstable	unstable	unstable

Figure 25: Evaluation scores.

The best combination is Test 2 c., which is a Filter Feature selection using the method of Mutual Information Gain and the classifier Logistic Regression. The filter is the fastest among Wrapper and Embedded Methods, which means that it is very efficient. Logistic Regression has the best effective scores among all other tests which are Recall 0.79, F1 0.51, and the Precision of 0.38. Precision is lower than Recall because there is a trade-off between recall and precision. Increased Recall decreases Precision. F1 is the highest score among all tests except for LR with the embedded method that has the same numbers. I chose the Filter method which has an execution time of 2.54 seconds, the embedded method execution time of 39.8 seconds, and wrapper which takes 480 seconds. An AUC score for the best performed combination is 0.88, which means that LR classifier predicts very well and distinguishes between the classes. Logistic Regression shows the best results among other classifiers and is known as low-variance machine learning algorithm. On the other hand, DT shows the worst results, all tests had AUC around 68% which means poor performance and the model has almost no class separation capacity. All Decision Trees in different tests are overfitted, meaning that the algorithm is performing very well on training data, but not performing well on testing data. I can see these by looking at AUC’s of training data which are 1, 0.99, 0.99, 1 and AUC’s on testing data which are 0.68, 0.68, 0.68, and 0.67 (Figure 26). Moreover, Decision Tree is a weak learner in this dataset which was expected before the model was run and Decision Tree is known as a high variance model. Due to an imbalanced dataset I decided to run PR_AUC. Test 2.c has the highest PR_AUC among other tests and which is 0.52. The Brier score is 0.12 which is good as well. Stability is another important indicator which was stable for our best performed model. When running a couple of times, the algorithms received same scores and the same feature selection output.

To optimize Random Forest performants I tested n_estimators for 40, 50, 100, 200, 10000 with max_depth of 2, 3 and 4, then found that n_estimators=50, with max_depth=3 gets the best results for all tests. I fitted a Decision Tree with default parameters as I tested max_depth of 2, 10 and 32 with criterion=’entropy’ and “gini”, and found that Gini is the best one and works well with max_depth=None. I tested to find the best accuracy performants of Logistic Regression tried different combination of C = 0.002, 0.003, 0.01, 0.1, 0.5 with max_inter = 10000, 200, 1000, 500 found that default parameters and solver ='liblinear' is the best combination for all tests. I replicated the procedure 50 times with Repeated k-fold cross-validation, used to estimate model efficacy. 10-fold cross-validation was run and repeated 5 times. The best result was for Logistic Regression and Rendon Forest which both got 90% accuracy see Figure 25.



Test number	Test 1			Test 2			Test 3			Test 4		
	a.	b.	c.	a.	b.	c.	a.	b.	c.	a.	b.	c.
Feature Selection				Filter			Embedded Methods			Wrapper		
	All Features			Mutual Information Gain			LASSO Regularization (L1)			Forward feature selection		
ML Model	**RF**	**DT**	**LR**	**RF**	**DT**	**LR**	**RF**	**DT**	**LR**	**RF**	**DT**	**LR**
ROC_AUC on testing set	0.85	0.68	0.85	0.86	0.68	0.88	0.86	0.68	0.88	0.87	0.67	0.88
ROC_AUC on training set	0.94	1	0.93	0.94	0.99	0.92	0.94	0.99	0.91	0.95	1	0.92

Figure 26: Comparison between training AUC and testing AUC

As a comparison to past research, Olatunji's (2016) best results were achieved by Decision Tree with Area under Curve (AUC) value of 76.6%. Moro et al.'s (2015) best results were achieved by Neural Network with AUC = 83%, and Ghatasheh et al.'s (2020) best results were achieved by Artificial Neural Network with AUC = 79%. My research outperform Olatunji's (2016), Moro's, and Ghatasheh's (2020) with the best results achieved by Logistic Regression with AUC = 88% for same balanced dataset.

Feature importance

The bar chart below (Figure 27) shows features that used in building Logistic Regression Test 2 c. which is Filter Feature selection with the method of Mutual Information Gain, the model ranked according to importance. This ranking is based on evaluation of the gain of each variable in the context of the target variable. These computed values describe how important the features are for the machine learning model and can explain how important those features are in the overall approximation of the relationship between the predictor variables and target variable. The scores on the y-axis of the bar chart represent relative importance, meaning the percent importance of each feature. In the bar chart, we see here that the most important feature in my mutual_info_classifier to explain and predict subscription is "cons.price.idx". Consumer price index has a positive effect on subscription since the lower the consumer price index, the more willingly a customer will spend their money on financial tools. Consumer price index contributes close to 45%, followed by “duration” with percent of importance of 44%, has a positive effect on subscription to a term deposit. This is because the longer the conversations on the phone, the higher interest the customer will show to the term deposit. The next one is “nr.employed” with 30% of importance which is the number of employees in the bank, has a positive effect on convincing people to subscribe to the term deposit. The fourth is the “month” with 18%, this one answers my third question, that the best time in a year for marketing campaign is August followed by April and June. The fifth powerful feature is the “contract” with 9% of importance. Those are the most powerful features and then comes, “default”, “job”, “poutcome” , “campaign” and “age”. Type of customers who are more likely to subscribe to term deposit are clients aged 20-50 working in administration jobs or technicians or blue collar workers who had no default account. Those are the top 10 performing features which answers my second question “What type of customers are more likely to subscribe to term deposit and which feature has higher influence?”

As a comparison to past research my research found similar results as Olatunji (2016), Guo et al. (2019) and Parlar et al.(2017), best features that significantly affect customers' subscriptions are “duration”, “month”, “number of employees”, and “contact”. All authors found that a reduced set of features improves classification performance.

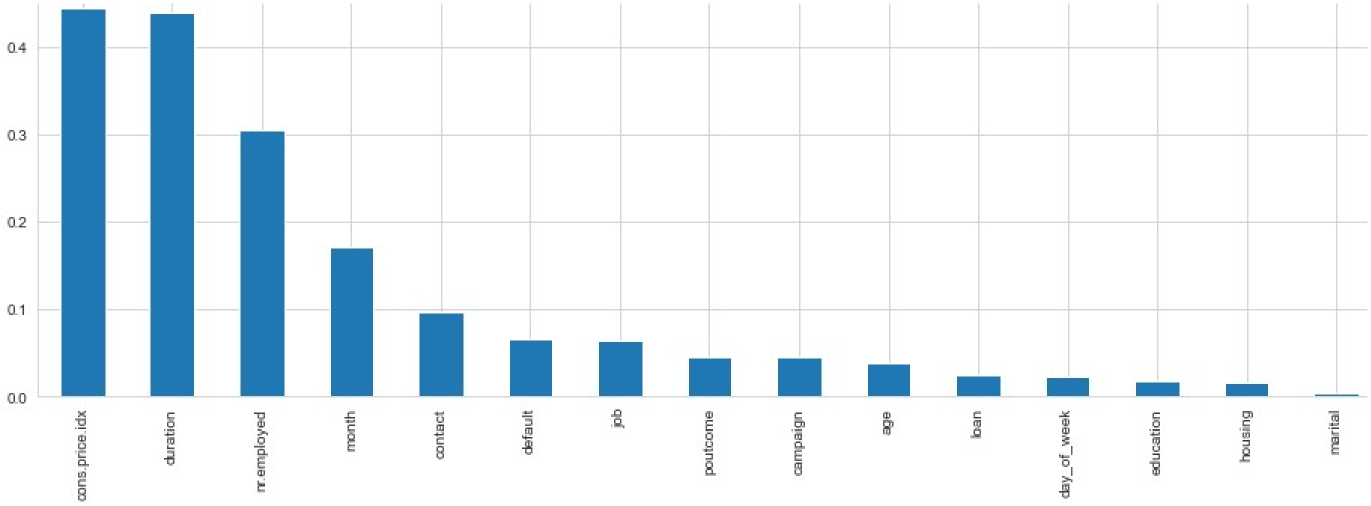


Figure 27: Filter Feature selection with Mutual Information Gain

Recommendation

I recommend to bank management to hire more people to work for a bank, improve the quality of conversation on the phone, and close the deal from the first call so there is no need to contact them many times. I also recommend running their campaigns in the spring and summer with an emphasis on August, April, and June. When interest rates are high and the macroeconomic environment is stable is the best time for a Marketing campaign.

Conclusion

The main contributions of this work are:

- Analyzed a large dataset (41,188 records) from a Portuguese bank.
- Compared 3 Classification models Logistic Regression, Decision Tree, Random Forest with 3 feature engineering techniques Filter, Wrapper, and Embedded method. Built a total of 12 models.
- Selected with Feature selection technic the most important features that contributes to subscription of bank Marketing campaign.

- Found the best combination Test 2 c. which is Filter Feature selection with the method of Mutual Information Gain and the classifier Logistic Regression. Scores as follows were found ROC_AUC of 88%, Precision of 38%, Recall of 79%, F1-score of 51%, PR-AUC of 52%, and Brier score of 12% and cross validation is 90%. Found the most powerful features that were selected by Filter Method of Mutual Information Gain, which ranked based on evaluation of the gain of each variable in the context of the target variable. The model ranked accordingly to importance with the best 5 feature which are 'cons.price.idx', 'duration', 'nr.employed', 'month', and 'contact'.

The limitation of this research was working with Scikit-learn (Sklearn) library in Python this means I have not tried other libraries that are available in Python. I have not tried other classification models. In the future, I would like to compare the classification performances of different datasets from different domains with different methods and different packages in Python.

Section 5: References

Olatunji, A. (2016). Evaluation of classification and ensemble algorithms for bank customer marketing response prediction. Journal of International Technology and Information Management, 25(4), 85. https://www.proquest.com/docview/1926957604?pq-origsite=summon&accountid=13631.

Moro, S., Cortez, P., & Rita, P. (2014). A data-driven approach to predict the success of bank telemarketing. Decision Support Systems, 62, 22-31. https://doi.org/10.1016/j.dss.2014.03.001

Moro, S., Cortez, P., & Rita, P. (2014;2015;). Using customer lifetime value and neural networks to improve the prediction of bank deposit subscription in telemarketing campaigns. Neural Computing & Applications, 26(1), 131-139. https://doi.org/10.1007/s00521-014-1703-0

Ghatasheh, N., Faris, H., AlTaharwa, I., Harb, Y., & Harb, A. (2020). Business analytics in telemarketing: Cost-sensitive analysis of bank campaigns using artificial neural networks. Applied Sciences, 10(7), 2581. https://doi.org/10.3390/app10072581

Domingos, P. (1999). Meta cost a general method for making classifiers cost sensitive. Artificial Intelligence Group Instituto Superior Tecnico Lisbon 1049-001, Portugal. https://homes.cs.washington.edu/~pedrod/papers/kdd99.pdf

Thai-Nghe, N., Gantner, Z., & Schmidt-Thieme, L. (2010). Cost-sensitive learning methods for imbalanced data. Paper presented at the 1-8. https://doi.org/10.1109/IJCNN.2010.5596486

Singh, A., Malka N. Halgamuge, Lakshmiganthan, R. (2017). Impact of Different Data Types Classifier Performance of Random Forest, Naïve Bayes, and K-Nearest Neighbors Algorithms. (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 8, No. 12, 2017 https://minervaaccess.unimelb.edu.au/bitstream/handle/11343/216910/2017_Asmita_Different_Data.pdf

Parlar, T., & Acaravci, l. K. (2017). Using data mining techniques for detecting the important features of the bank direct marketing data. International Journal of Economics and Financial Issues, 7(2), 692-696. https://www.proquest.com/docview/2270076746?pq-origsite=summon&accountid=13631

Guo, J., & Hou, H. (2019). Statistical decision research of long-term deposit subscription in banks based on decision tree. Paper presented at the 614-617. https://doi.org/10.1109/ICITBS.2019.00153

Krishna, C., L., & Reddy, P., V., S., (2019). Deep Neural Networks for the Classification of Bank Marketing Data using Data Reduction Techniques. International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8 Issue-3, September 2019. https://www.researchgate.net/profile/Leela-Krishna-Chittem/publication/336287356_Deep_Neural_Networks_for_the_Classification_of_Bank_Marketing_Data_using_Data_Reduction_Techniques/links/5d9a1d5da6fdccfd0e7eeeb

1/Deep-Neural-Networks-for-the-Classification-of-Bank-Marketing-Data-using-Data-Reduction-Techniques.pdf

Steinbach, M., Tan, P., N., & Kumar, V., (2005). Introduction to Data Mining. 1-769.

Silipo, R., Adae, I., Hart, A., Berthold, M., (2014). Seven Techniques for Dimensionality Reduction, KNIME.com, knime_seventechniquesdatadimreduction.pdf

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel., O, Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E., (2011). Scikit-learn: Machine Learning in Python, by Fabian Pedregosa , Gael Varoquaux . Journal of Machine Learning Research 12 (2011) 2825-2830, https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf Source code, binaries, and documentation can be downloaded from: https://scikit- learn.org/stable/

Pandas development team. (2021, Oct 17). Pandas documentation. pydata. https://pandas.pydata.org/docs/

NumPy. (2021, Jun 22). NumPy v1.21 Manual. numpy. https://numpy.org/doc/stable/

When Should You Delete Outliers from a Data Set? (2018, March 06). humansofdata. Retrieved November 25, 2021, from https://humansofdata.atlan.com/2018/03/when-delete-outliers-dataset/

Brownlee, J. (2020, January 17). SMOTE for Imbalanced Classification with Python. Machine

Learning Mastery. https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002;2011;). SMOTE: Synthetic minority over-sampling technique. The Journal of Artificial Intelligence Research, 16, 321-357. https://doi.org/10.1613/jair.953

Shaikh, R. (2018, October 28). Feature Selection Techniques in Machine Learning with Python. Towards data science. https://towardsdatascience.com/feature-selection-techniques-in-machine-learning-with-python-f24e7da3f36e

Kuhn, M., & Johnson, K. (2020;2019;). Feature engineering and selection: A practical approach for predictive models (1st ed.). CRC Press, Taylor & Francis Group. https://doi.org/10.1201/9781315108230

Confusion matrix. (2021, November 28). In Wikipedia. https://en.wikipedia.org/wiki/Confusion_matrix

Yildirim, S. (2020, Mar 16). How to Best Evaluate a Classification Model. Towards data science. https://towardsdatascience.com/how-to-best-evaluate-a-classification-model-2edb12bcc587

Precision and recall. (2021, October 18). In Wikipedia. https://en.wikipedia.org/wiki/Precision_and_recall

Erika, D. (2019, Dec 08). Accuracy, Recall & Precision.

Medium. https://medium.com/@erika.dauria/accuracy-recall-precision-80a5b6cbd28d

Narkhede, S. (2018, Jun 26). Understanding AUC - ROC Curve.

Towards data science. https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

Brownlee, J. (2018, August 31). How to Use ROC Curves and Precision-Recall Curves for

Classification in Python. Machine Learning Mastery.

https://machinelearningmastery.com/roc-curves- and-precision-recall-curves-for-classification-in-python/

Dash, S. (2020, December 27). Brier Score – How to measure accuracy of probabilistic

predictions. Machine learning gplus. https://www.machinelearningplus.com/statistics/brier-score/

Brownlee, J (2020, August 3). Repeated k-Fold Cross-Validation for Model Evaluation in

Python https://machinelearningmastery.com/repeated-k-fold-cross-validation-with-python/