Springboard Data Science

Bank Marketing for Term Deposit - Capstone I

Nov. 2019

Bhargavsinh Ravalji

Contents

1. Introduction

The essential business of a financial institution can be largely classified as lending and borrowing. Lending produces revenue to the bank in the form of interest from customers with some level of default risk involved. Borrowing, or rather attracting public's savings into the bank is another source of revenue generation, which can be less risky than the former. A bank usually invests the customer's long-term deposits into riskier financial assets which can earn the better return than what they pay to their customer. The customer, on the other hand, is guaranteed a risk-free return on his/her deposit. However, the return on the fixed-term deposit is better than the savings account as the customer is lacking off the rights to use the fund prior to the maturity unless one is ready to reward the bank as per the pre required agreements on the particular term deposit scheme.

In this project, we apply machine learning algorithms to build a predictive model of the Portuguese Bank Marketing data set to provide a necessary suggestion for marketing campaign team. The goal is to predict whether a client will subscribe a term deposit with the help of a given set of dependent variables. This is a real dataset collected from a Portuguese bank that used its own contact-center to do direct marketing campaigns to motivate and attract the clients for their term deposit scheme to enhance the business.

1.1 Problem Statement

Portuguese bank lost their revenue, and they wanted to investigate why their revenue declined. So, they can take necessary steps to solve bank problems. After deep analysis, they discovered that the main reason is that their clients are not depositing as frequently as before. Expressive to term deposits allow banks to hold onto a deposit for certain amount of time, so banks can invest in higher gain financial products to make a profit. Furthermore, banks also hold better chance to encourage term deposit clients into buying other products such as funds or insurance to

further increase their revenues. Consequently, the Portuguese bank would like to identify existing clients that have higher chance to subscribe for a term deposit and focus marketing efforts on such client.

1.2 Client

Our goal is to build a classifier to predict whether or not a client will subscribe a term deposit. If the classifier has high accuracy, the banks can arrange a better management of available resources by focusing on the potential customers selected by the classifier, which will improve their efficiency a lot. Besides, we plan to find out which factors are influential to customers' decision, so that a more efficient and precise campaign strategy can be designed to help to reduce the costs and improve the profits. The Portuguese bank will focus marketing effort on existing clients which have more likely to subscribe for a term

1.3 Dataset Summary

The data on Bank Marketing Data Set can be obtained from the UCI - Machine Learning Repository (http://archive.ics.uci.edu/ml/datasets/Bank+Marketing) which has file bank-additional-full.csv with all examples (41188) and 20 inputs, ordered by date (from May 2008 to November 2010).

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 whole data set is the bank's client database consisting of 17 different variables/attribute which is elaborated below.

Attribute Information:

Input 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 (desired target):

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

2. Data Analysis

2.1 Missing Data Points

Bank dataset was collected from phone call interviews, many clients refused to provide their personal information due to the privacy issue. The existence of missing data may hide the real pattern hidden in the data thus making it more difficult to extract information. Therefore, we chose three methods to deal with those missing data for different attributes.

```
age               0
job               0
marital           0
education         0
has_credit        0
housing_loan      0
personal_loan     0
contact           0
month             0
day_of_week       0
duration          0
campaign          0
pdays             0
previous          0
poutcome          0
emp.var.rate      0
cons.price.idx    0
cons.conf.idx     0
euribor3m         0
nr.employed       0
subscribed        0
dtype: int64
```

Figure 1: Missing value count

The education attribute has 1731 missing lines marked as “unknown” in this attribute. A chi-square test was applied to check the independence and the p-value of the result is less than 0.01. In this case, the unknown status is obviously related to our target response and we cannot simply ignore those missing values. Therefore, we use the rest known data to impute the missing terms.

The “default” attribute, the total amount of “yes” response is very small, only have three clients. Though, the number of “unknown” status is quite large which is 8,598 in total. After chi-square test, there is also a strong evidence shows that there is a relationship between this unknown status and our target response. In this case, we cannot make suggestion because of the rare

population of “yes” response. Therefore, we decided to keep the “unknown” status as a new type and use it in our algorithm.

2.2 Outliers of Input Features

An outlier is data point which is far away from the distribution of data. Therefore, this data point will not be useful in classification model, instead it can lead to incorrect training of the classifier model. These data point can be dropped or ignored. By looking at below boxplots of the different features, we can see ‘age’, ‘previous’ and ‘campaign’ features have outliers. we can remove these features by applying thresholds, but Random forest classifier will use for this project which handles outliers by essentially binning them. It is also indifferent to non-linear features.

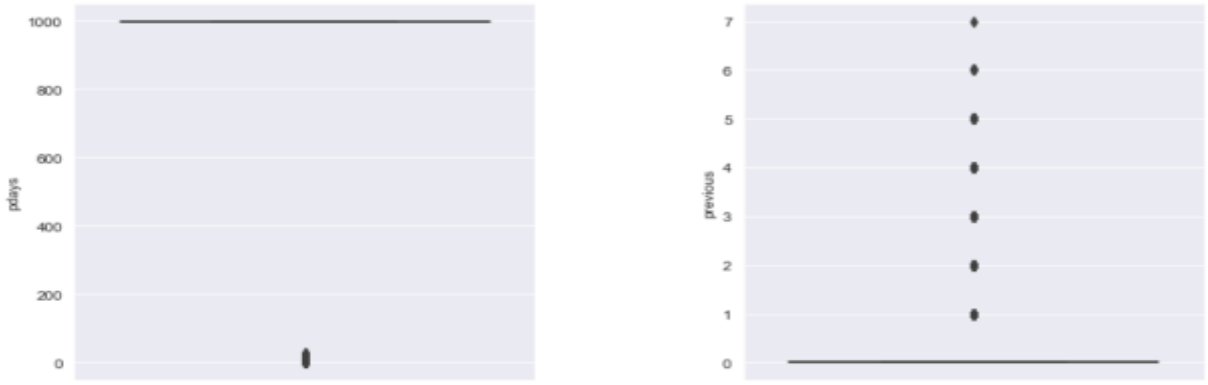

Figure 2: Outliners of Pdays and Previous Features

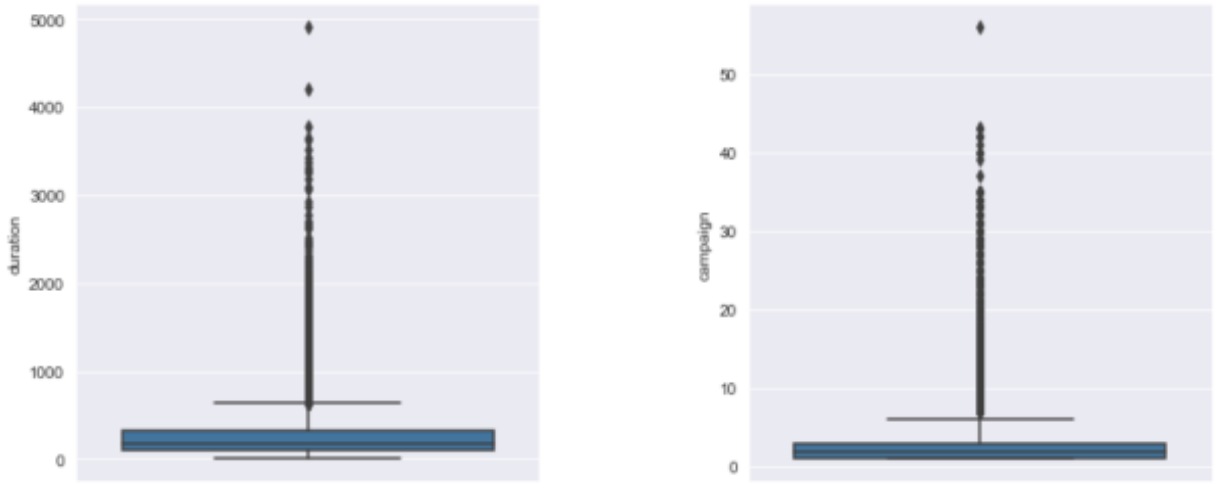

Figure 3: Outliners of Duration and Previous Features

2.3 Correlation Heatmap

The Correlation Heatmap Matrix can be seen below Figure 4 for all the numerical attributes. As shown in Heatmap, nr.employed and euribor3m also have a strong positive correlation with value of 0.95. emp.var.rate and nr.employed have a strong positive correlation with value of 0.91. Also, emp.var.rate and euribor3m have a very strong correlation with the value of 0.97. Pdays and previous has weak negative correlation with value of -0.59.

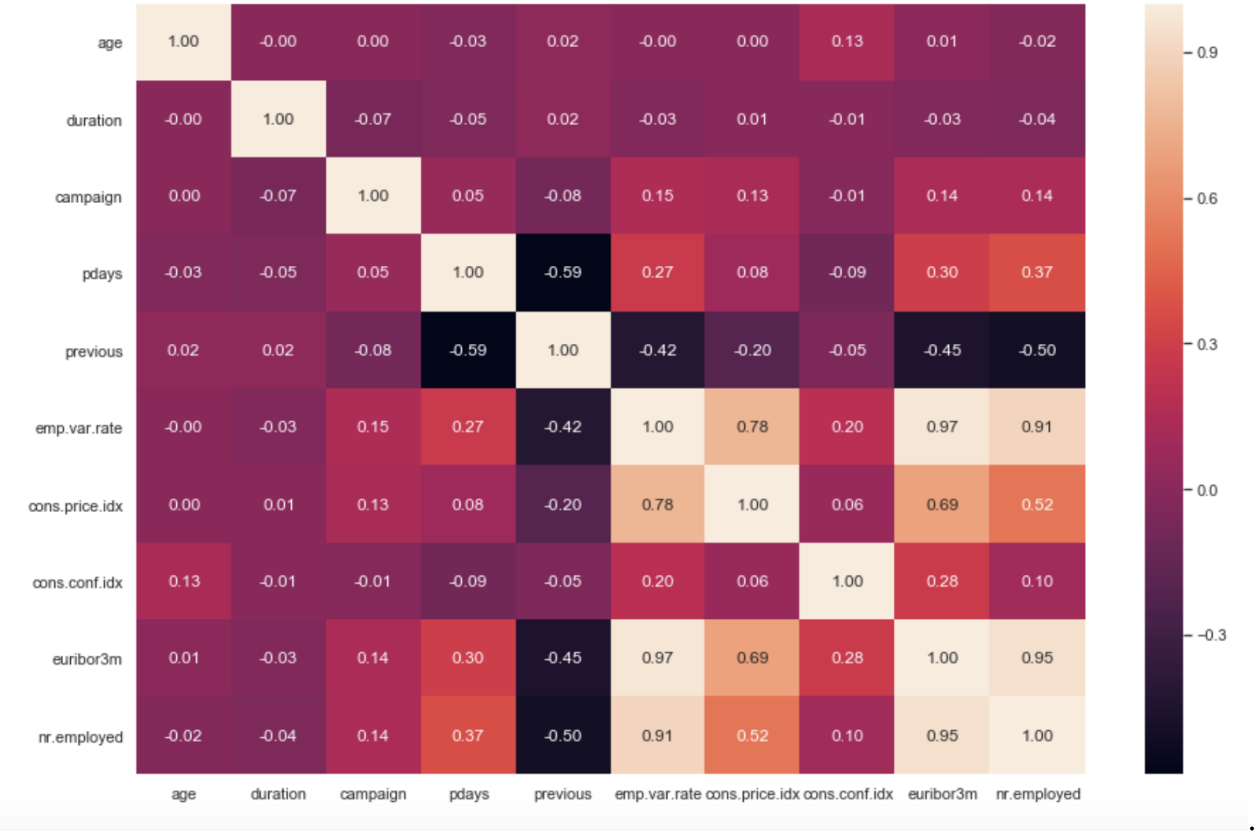

Figure 4: Correlation Heatmap

3. Preprocessing Data

3.1 Handling Imbalance dataset

At beginning, model was created with random forest classifier. Figure 4 shows Imbalance data of target variable which can provides model with misleading accuracy. Due to Imbalance data of target variable, over sampling or under sampling operations have to be performed. For this dataset, the oversampling operation applied to increase the data points which provided more data for the classifier to classify. To increase the data points, weights are given to the classes so that the data points of both the classes are increased which definitely increases the accuracy.

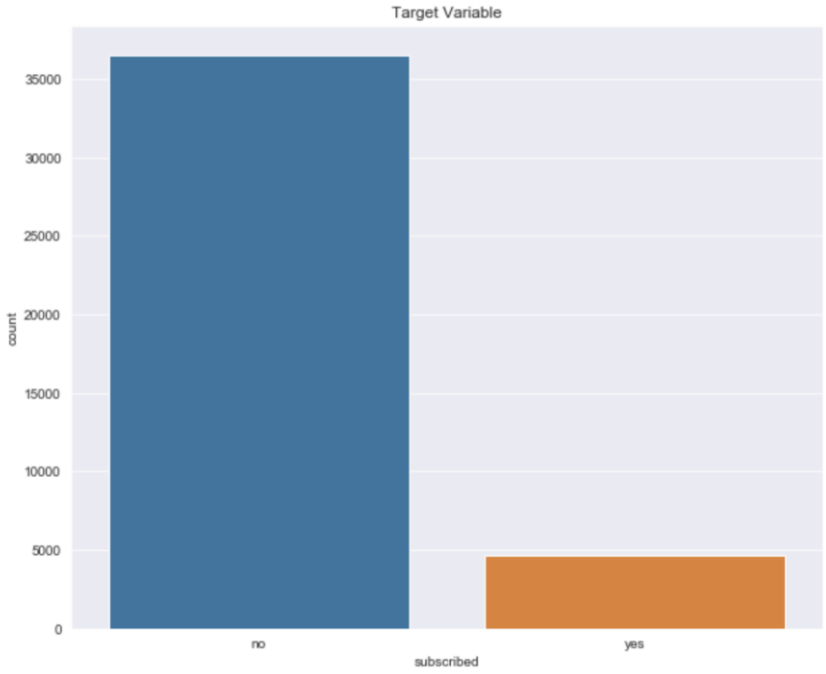

Figure 5: Imbalance data of target variable

For this dataset, **SMOTE**, oversampling technique called Synthetic Minority Over-Sampling Technique, used to balance out our dataset. SMOTE has become one of the most popular algorithms for oversampling. The simplest case of oversampling is simply called oversampling or upsampling, meaning a method used to duplicate randomly selected data observations from the outnumbered class.

SMOTE Algorithm has oversampled the minority instances and made it equal to majority class. Both categories have equal amount of records. More specifically, the minority class has been increased to the total number of majority class. Below figure 5 shows minority class has same number of data points after applying SMOTE technique.

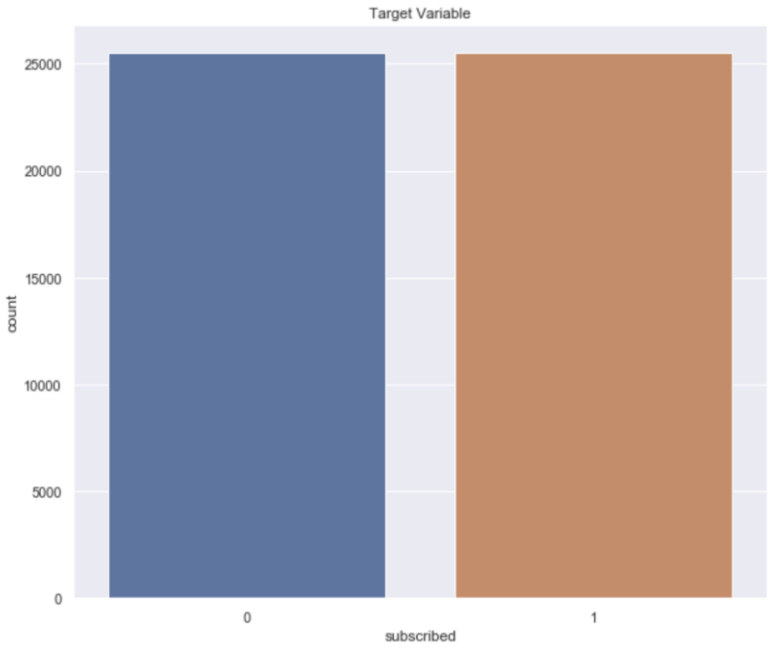

Figure 6: Balance data of target variable after applying SMOTE

3.2 Label Encoding and One Hot Encoding of Categorical Features

Label Encoding in Python can be achieved using Sklearn Library. Sklearn provides a very efficient tool for encoding the levels of categorical features into numeric values so as to convert it into the machine-readable form. Machine learning algorithms can then decide in a better way on how those labels must be operated. It is an important pre-processing step for the structured dataset in supervised learning.

Depending upon the data values and type of data, label encoding induces a new problem since it uses number sequencing. The problem using the number is that they introduce comparison between them. To overcome this problem, we use One Hot Encoder. The unordered categorical

features which have no predefined sequence but are string features have to be transformed to one hot vectors. By using one hot encoding, the feature space is increased. The numbers are replaced by 1s and 0s, depending on which column has what value.

3.3 Normalization of Dataset

To normalize the data, StandardScaler is used to standardize a feature by subtracting the mean and then scaling to unit variance. Unit variance means dividing all the values by the standard deviation. StandardScaler will normalize the features so that each feature will have mean = 0 and standard deviation = 1.

$$\text{Standardization:}$$
$$z = \frac{x-\mu}{\sigma}$$
$$\text{with mean:}$$
$$\mu = \frac{1}{N}\sum_{i=1}^{N}(x_i)$$
$$\text{and standard deviation:}$$
$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i-\mu)^2}$$

4. Building Models

4.1 LR, NB, and DT models

For this capstone project, Logistic regression, Naïve bayes, and Decision tree were applied. Cross validation tested to check which model will be best for Bank Marketing dataset. After cross validation, mean score for logistic regression, Naïve bayes, and Decision tree was 0.88, 0.76, and 0.93 accordingly. Therefore, Decision tree model will be best among other model classifier.

4.2 Random Forest Classifier

Random forest classifier creates a set of decision trees from randomly selected subset of training set. Then, it combines the votes from different decision trees to decide the final class of the test object. Instead of Considering equal voting from all the trees, the weights on the tress can be changed, Hence, trees having low error can be given more weight whereas trees having high error will be given less weight. Because of this, the classification will be properly work. The parameters in random forest are the total number of decision trees, the minimum split etc.

I used the Randomclassifier() function of sklearn library with applying different number o f parameters. The most import parameters are the total numbers of trees to be used. The random for est uses features while deciding the split a node or not which brings in a lot of randomness while cr eating the trees. This in turn helps in classification of unknown data points. The main reason why R andom forests are used is because they prevent the problem of overfitting as they create smaller tre es and combine them together to generate an ensemble. The best accuracy and F1 score for the ban k dataset was obtained using Random Forest which is intuitively correct as Random Forest classifie r works very well for imbalanced datasets. The F1 score was 0.95 with training set accuracy 96.98 % and Test set accuracy: 94.98%

5. Performance Evaluation

5.1 Hyper parameter tuning with RandomizedSerchCV

Random search is a method to parameter tuning that will sample algorithm parameters from a random distribution for a fixed number of iterations. A model is created and evaluated for each combination of parameters chosen. The parameters of the estimator used to apply these methods are improved by cross-validated search over parameter settings. In RandomizedSearchCV, all parameter values are not tried out, but rather a fixed number of parameter settings is sampled from the specified distributions.

GridSearchCV can be computationally expensive, especially if you are searching over a l arge hyperparameter space and dealing with multiple hyperparameters. A solution to this is to us e RandomizedSearchCV, in which not all hyperparameter values are tried out. Instead, a fixed num ber of hyperparameter settings is sampled from specified probability distributions. After applying, RandomizedSearchCV with following best parameters'bootstrap': True, ‘criterion': 'entropy’, ‘max_ depth': 80, 'max_features': 2, ‘min_samples_leaf': 1, 'min_samples_split': 12, 'n_estimators': 150, m odel Training set accuracy appeared 96.98%, and Test set accuracy occurred 94.98%.

5.2 Precision, Recall, and AUC

To evaluate the performance of classifier, certain performance metrics are used such as C onfusion Matrix, Accuracy score, F1 score, Area Under the Curve (AUC), Receiver Operating Cha racteristics (ROC) etc. The classifier is trained using the training data and the unknown testing data is given to it to check the classification for this unknown data. By providing the unknown test data, the classifier produces certain predictions. As this is a supervised learning method, the test labels ar e already provided as ground truths. Hence, these test labels are matched to the predicted labels to measure the parameters specified above. Classification Report is generated which gives the precisio n and recall values which provide an idea of the performance of the classifier.

Depending on the predictions made by the classifier, predicted labels can be distributed into four types: **True Positive, True Negative, False Positive, False Negative**

The true positive and True negative are the best cases as they coincide with the expected outputs. False Positive and False negative cases should be avoided.

The precision and recall value for the data is given as,

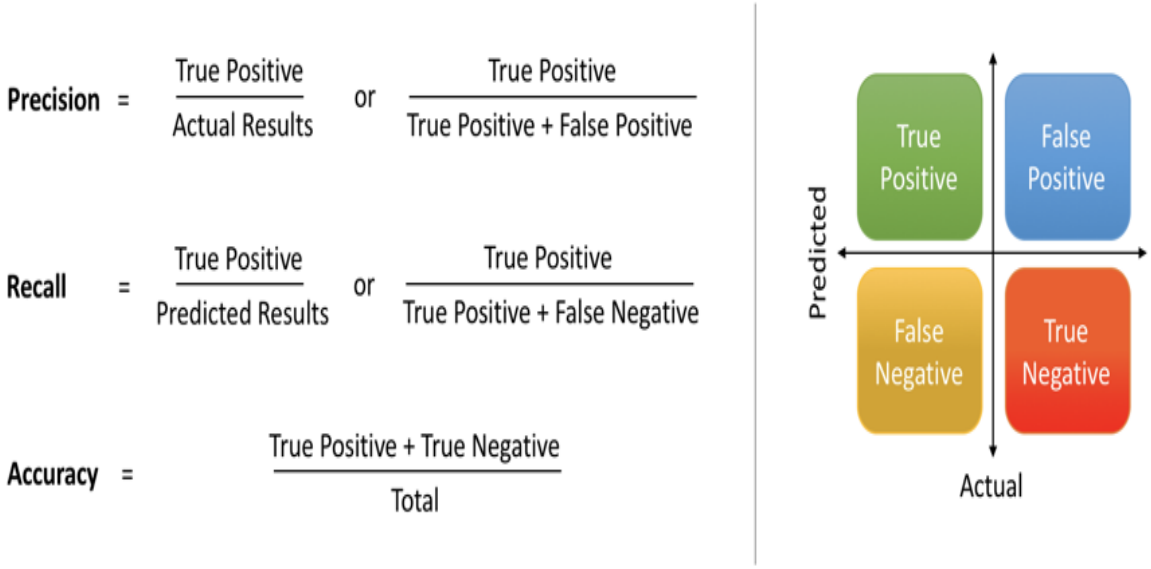

Figure 7: Precision and Recall

The F1 source is given as:

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

Different functions used in python to evaluate these performance metrics are accuracy_score(), f1_score(), classification_report(), roc_curve(), auc(). All these functions are present in the sklearn metrics library.

Classification Report:

	precision	recall	f1-score	support
0	0.93	0.97	0.95	7626
1	0.97	0.93	0.95	7682
accuracy			0.95	15308
macro avg	0.95	0.95	0.95	15308
weighted avg	0.95	0.95	0.95	15308

Figure 8: Classification report of Random forest classifier

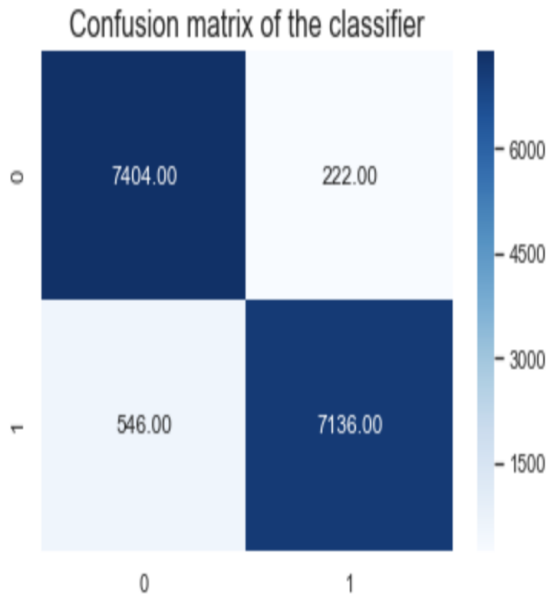

Figure 9: Confusion Matrix of Classifier

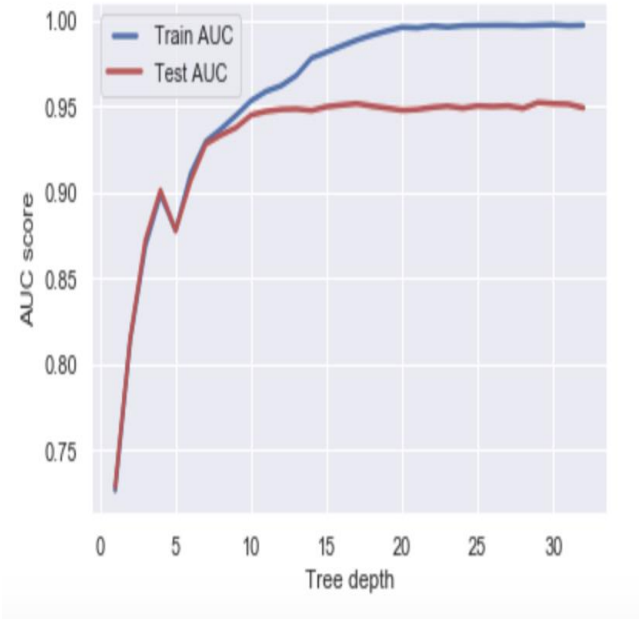

Figure 10: Tree depth Vs AUC score

6. Conclusion

Duration of the call is the feature that most positively correlates with whether a potential client will open a term deposit or not. The mode of contact with client should be telephone. The level of commitment of the potential client will lead to increase probability of subscribing to a term deposit, and then an increase in efficiency for the next marketing campaign the bank will accomplish. Focus on those customers who were part of the previous campaign.

For next marketing campaign of the bank, marketing team should target younger than 20 and older than 60 potential clients. If bank addressed these two categories for the next campaign, it will increase the probability of more term deposits subscriptions. Bank can suggest more old people for safe investment, steady income, and peace of mind as the value proportion.

Bank can increase their marketing campaigns by focusing their efforts on specific clients as mention above. As a result, next marketing campaign of the bank will be definitely more efficient than the existing bank marketing campaign.

7. References

Blog, G. (2019, June 24). How to handle Imbalanced Classification Problems in machine learning? Retrieved from https://www.analyticsvidhya.com/blog/2017/03/imbalanced-classification-problem/

Janio bachmann. (2019, March 16). Bank Marketing Campaign || Opening a Term Deposit. Retrieved from https://www.kaggle.com/janiobachmann/bank-marketing-campaign-opening-a-term-deposit

ODSC - Open Data Science. (2019, July 26). Optimizing Hyperparameters for Random Forest Algorithms in scikit-learn. Retrieved January 29, 2020, from https://medium.com/@ODSC/optimizing-hyperparameters-for-random-forest-algorithms-in-scikit-learn-d60b7aa07ead

Roy, S. (2020, January 18). Machine Learning Case Study: A data-driven approach to predict the success of bank telemarketing. Retrieved from https://towardsdatascience.com/machine-learning-case-study-a-data-driven-approach-to-predict-the-success-of-bank-telemarketing-20e37d46c31c

Saxena, S. (2018, May 13). Precision vs Recall. Retrieved from https://towardsdatascience.com/precision-vs-recall-386cf9f89488