Predicting Success of Bank Telemarketing calls using ML Algorithms

Abhishek Laxman Joshi

CB.EN.P2DSC21033

I-Mtech-DS

Batch:2021-23

Course:21DS602(21-22(Odd))

Date: 31st Jan 2022

Abstract- **One of the field that is being changed the most by the boom of Machine Learning(ML) propels is the financial field. Be it foreseeing the stocks, or in our case anticipating if a client or customer will subscribe to a term deposit, Machine Learning can be staggeringly helpful for giving better benefit. It is actually a binary classification problem. Of the two classes "yes" denotes the customer subscribing to a term deposit and "no" denotes that the customer does not subscribe. In this paper, we propose various machine learning algorithms to predict the success of the telemarketing calls for subscription to term deposit (the dependent variable y). We compared different machine learning models: self-implemented logistic regression, sklearn logistic regression, Naïve-Bayes algorithm, decision trees, K-Nearest Neighbour(KNN), Support vector machine(SVM) with linear kernel, support vector machine with sigmoid kernel. Using the metric, area of the receiver operating characteristic curve(AUC), the different models are tested. The sklearn logistic regression model presented the best results with AUC= 0.91. The experiment was also taken under several measures like precision, recall, accuracy, F-score.**

I. INTRODUCTION

In the present world, data is ruling. We need to use data in the best way and can great a tremendous impact on the business. If one does not hold on to data, such a organization or business will be left behind in this fast paced world in no time. One of the techniques through which an industry or organization can improve its performance in the market is to obtain and do analysis of customer data in the most effective way so that they can be the customers happy by improving their experience. Marketing campaigns are a typical strategy of the most organizations to improve performance in business. Marketing can be done directly by the organizations when they are thinking of targeting specific customers. So one of the most common way of contacting the customer is by making a call on their landline or mobile. Be it any marketing strategy the main goal is to maximize the sales or the maximize the service subscriptions for the particular product. Telemarketing is one of the ways where the subscriber of the product and the seller come in direct contact through telephonic conversation so that the seller can make an good influence to sell the product. If we see the trends, this field of telemarketing is rapidly growing. In financial domain or the domain of banking, telemarketing is the most used technique to improve the customer experience. This marketing depends on extensive details of the market and client requirements.

So actually it's basically a Decision Making System (DSS) which is helping us know if the customer will subscribe or not. Machine Learning is one of the famous technologies which can be used in DSS by combining data and computer applications for precisely predicting the outcome [1]. In general, we have two types of models in ML, one is the supervised learning models and other being the unsupervised learning models. In supervised learning model, the output is known which will be used for predictions. In the unsupervised learning model, it makes use of input data and no output variable is previously known [1].

The dataset given, consists of direct marketing campaigns data of a particular banking organization [3]. There are different variants of the dataset of which we have considered "bank-additional-full.csv" consisting of 41188 data with 20 independent variable or features. In these 20 features, 10 are categorical and the rest 10 are numerical. The list of features available to us are: 1) age(numeric) 2) job: type of job (categorical) 3) marital: marital status (categorical) 4) education (categorical) 5) default: has credit in default ? (categorical) 6) housing: has housing loan? (categorical) 7) loan: has personal loan? (categorical) 8) contact: contact communication type(categorical) 9) month: last contact month of year (categorical) 10) day_of_week:last contact day of the week (categorical) 11) duration: last called duration, in seconds(numeric) 12) campaign: number of calls done during this campaign and for this client (numeric) 13) pdays: total days passed since the customer was last called from a previous campaign (numeric) 14) previous : number of calls done before this campaign and for this client (numeric) 15) poutcome :output of previous marketing campaign (categorical) 16) emp.var.rate : employment variation rate (numeric) 17) cons.price.idx : consumer price index (numeric) 18) cons.conf.idx : consumer confidence index (numeric) 19) euribor3m : euribor 3 month rate (numeric) 20) nr.employed : number of employees (numeric)

II. LITERATURE

Different papers have implemented different ML models for the same dataset. In the Decision Support Systems paper mentioned as the [1] reference of this paper, the best results authors obtained is 0.85. They made use of three models namely Naïve Bayes, Random forest and J48 classifier. In the paper mentioned as reference [4] of this research paper models such as logistic regression, decision trees, neural networks, and support vector machines have been used and they have obtained an AUC of 0.8. The research paper mentioned in reference [5] have obtained an accuracy of 94.39% using decision trees. There are many other open source projects done on GitHub and other websites where a similar accuracy has obtained. So our goal is obtaining similar results.

III. Objectives

- To design a DSS to predict the success of bank telemarketing calls for selling Long Term Deposits.
- Selecting the best set of customers or targeting the right segments of customers that those who are more likely to subscribe to a product.
- The given problem can be posed as a Classification Problem that is supervised learning
- Assigning dependent variable Y, value Yes/No, based on whether a client has subscribed a term deposits or not
- As referred papers have obtained an AUC value of 0.8, we also try to achieve the same

IV. Theoretical Background

Let's see some information related to the algorithms that we have made use of

- Logistic Regression:

Logistic regression is a supervised machine learning algorithm which is a process of modeling the probability of a discrete outcome given an input variable. The most common logistic regression models a binary output something that can take two values such as true/false, yes/no, and so on. Multinomial logistic regression can model cases where there are more than two possible discrete outputs. Logistic regression is a useful analysis method for classification problems, where you are trying to determine if a new sample fits best into a category [5].

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

$$\mathcal{L}(\hat{y}, y) = -(y \log \hat{y} + (1 - y) \log(1 - \hat{y}))$$

- Naïve Bayes

It is a classification algorithm based on Bayes Theorem which assumes that the features are independent. This is when the features may or may not be independent. Naïve Bayes is easy to build and specifically its very useful for large datasets [6]. In addition to how simple it is, Naïve Bayes has been found to outperform many highly sophisticated classification algorithms. Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c)

$$\underset{\text{Posterior Probability}}{P(c \mid x)} = \frac{\overset{\text{Likelihood}}{P(x \mid c)}\,\overset{\text{Class Prior Probability}}{P(c)}}{\underset{\text{Predictor Prior Probability}}{P(x)}}$$

$$P(c \mid \mathrm{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

P(c|x) is the posterior probability of class (c, target) given predictor (x, attributes).
P(c) is the prior probability of class.
P(x|c) is the likelihood which is the probability of predictor given class.
P(x) is the prior probability of predictor.[1]

- KNN

KNN is one of the simplest supervised learning machine learning technique. It first assumes similarity between the new data point and the currently available data points and puts the new data point into the class that is most similar to the available classes. It is also called a lazy learner algorithm because it does not learn from the training set immediately instead of that, it stores the dataset and at the time of classification, it performs an action on the dataset [7]

Algorithm for KNN:

1. Calculate the distance between new data point and remaining training data points with help of any distance measure like Euclidean, Manhattan distance.
2. Now based on the calculated distances , sort them in ascending order.
3. Choose the top K rows
4. Now, assign it to a class to the test point based on most frequently occurring class

- Support Vector Machines(SVM)

SVM is also a supervised ML algorithm that can be used for both regression and classification problems. In SVM, we plot each of the data point in n-dimensional space with the value of each feature being the value of a particular coordinate (n is number of features) [8]. Then we classify by finding the hyper-plane that differentiates the two classes very well. Support vectors are the coordinates of individual observation. In SVM, it is easy to have a linear hyper-plane between the two classes. But if the data is not linearly separable? Here it uses something known as kernel. SVM kernel is a function that takes as input of lower dimension and transforms it to a higher dimensional space that is its nothing but converting non linearly separable problem to separable one. Kernel functions may be polynomial kernel function, Gaussian kernel function, sigmoid kernel function etc.

- Decision Trees

It is a tool used for both regression and classification problems. As per the name , it uses flowchart like a tree structure to show the predictions that result from a series of feature-based splits. It starts with a root node and ends with a decision made by leaves [9]. The amount of uncertainty is calculated using Entropy. The formula for Entropy is as follows

$$E(S) = -p_{(+)}\log p_{(+)} - p_{(-)}\log p_{(-)}$$

p_ is the probability of negative class

S is the subset of the training example

Information Gain: It measures the reduction of entropy given some feature and it is also decides for which feature should be selected as a decision node or root node.

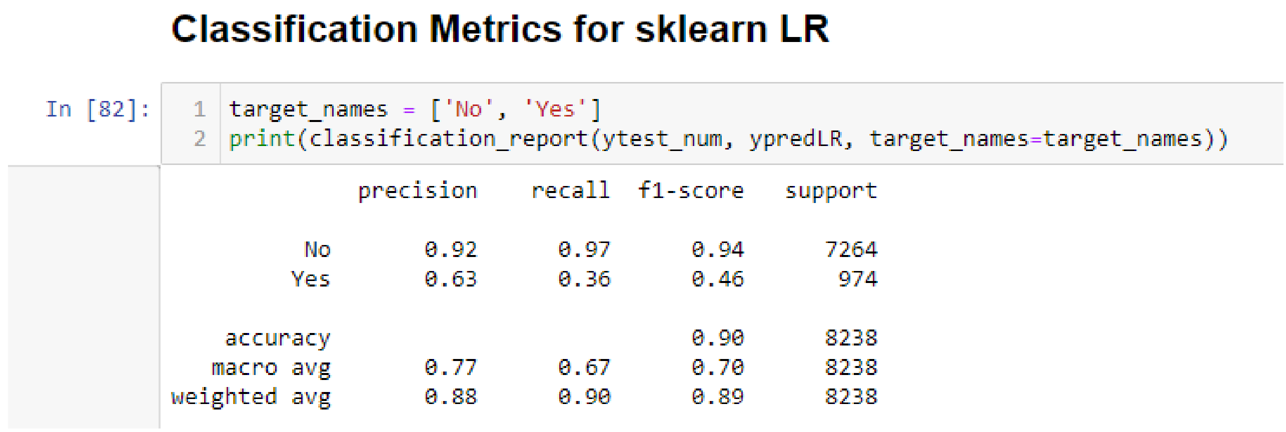
Information Gain = E(Y) – E(Y|X)

V. **Methodology**

1. First we do Exploratory Data Analysis (EDA) to find what kind of data we are dealing with. Then we perform some univariate analysis to find out which features can help us in our task of classifying and variables which are not so important.
2. We check if the dataset is balanced or not. The number of negative classes is about 8 times that of positive class, so its imbalanced.
3. We look at the count plots of each categorical feature to find its class distribution.
4. Then Univariate analysis on numerical features is done.
5. Correlation matrix of the numerical features is found out to check which features are more correlated and which are not.
6. Data Pre-Processing is done as we have 10 categorical features, we need to encode them in numerical way. We even deal with the missing values and duplicate values
7. Data is split into training and testing datasets with 80% and 20% respectively.
8. Generating model from training data
9. Fit different models such as logistic regression, KNN, Naïve Bayes, Decision trees, SVM.
10. Obtain the classification report
11. Calculate the accuracy
12. Plot the ROC for each model and display the ROC score.
13. Compare all the ROC
14. Apply K Fold cross validation for different models and see the score
15. Check which model is giving the best score.

VI. Experimental Results and Discussion

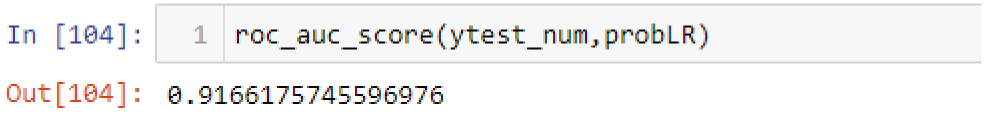
Results of Logistic Regression



Classification Metrics for sklearn LR

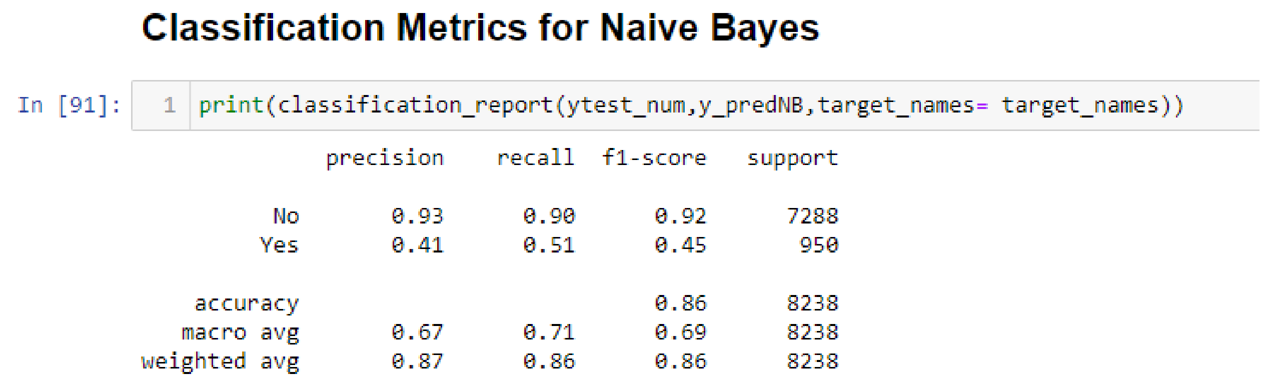
```
In [82]:  target_names = ['No', 'Yes']
           print(classification_report(ytest_num, ypredLR, target_names=target_names))
```

```
              precision    recall  f1-score   support

          No       0.92      0.97      0.94      7264
         Yes       0.63      0.36      0.46       974

    accuracy                           0.90      8238
   macro avg       0.77      0.67      0.70      8238
weighted avg       0.88      0.90      0.89      8238
```



```
In [104]:  roc_auc_score(ytest_num,probLR)

Out[104]:  0.9166175745596976
```

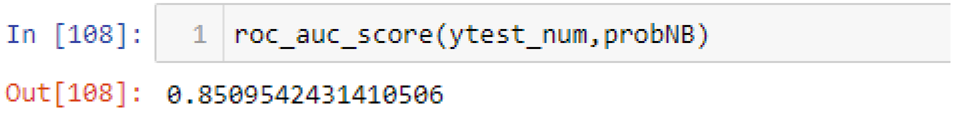
Results of Naïve Bayes Classifier:



Classification Metrics for Naive Bayes

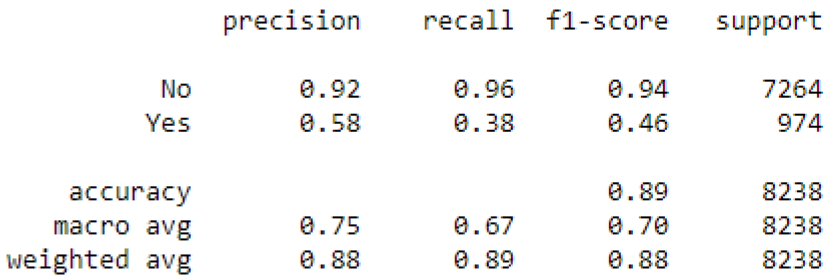
```
In [91]:  print(classification_report(ytest_num,y_predNB,target_names= target_names))
```

```
              precision    recall  f1-score   support

          No       0.93      0.90      0.92      7288
         Yes       0.41      0.51      0.45       950

    accuracy                           0.86      8238
   macro avg       0.67      0.71      0.69      8238
weighted avg       0.87      0.86      0.86      8238
```

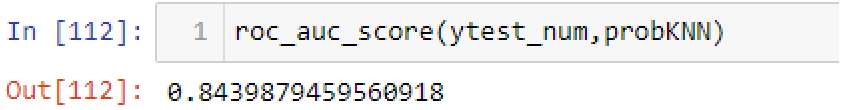


```
In [108]:  roc_auc_score(ytest_num,probNB)

Out[108]:  0.8509542431410506
```

Results of KNN

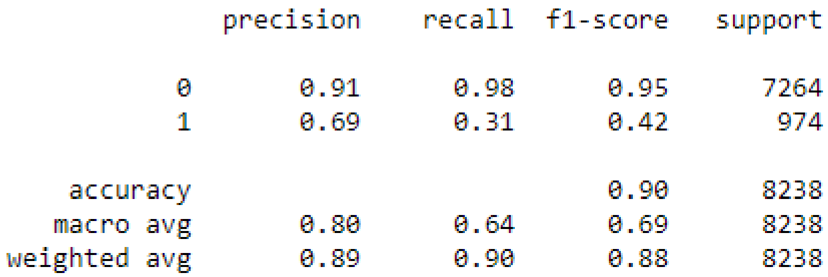


```
              precision    recall  f1-score   support

          No       0.92      0.96      0.94      7264
         Yes       0.58      0.38      0.46       974

    accuracy                           0.89      8238
   macro avg       0.75      0.67      0.70      8238
weighted avg       0.88      0.89      0.88      8238
```

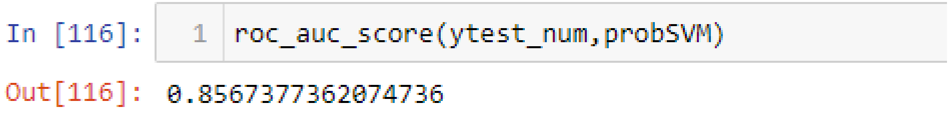


```
In [112]: roc_auc_score(ytest_num,probKNN)
Out[112]: 0.8439879459560918
```

Results of Linear SVM:

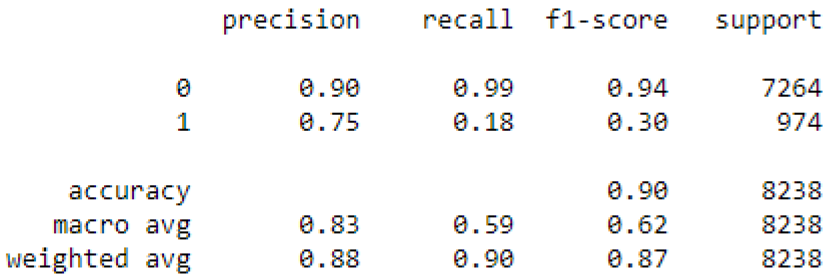


```
              precision    recall  f1-score   support

           0       0.91      0.98      0.95      7264
           1       0.69      0.31      0.42       974

    accuracy                           0.90      8238
   macro avg       0.80      0.64      0.69      8238
weighted avg       0.89      0.90      0.88      8238
```

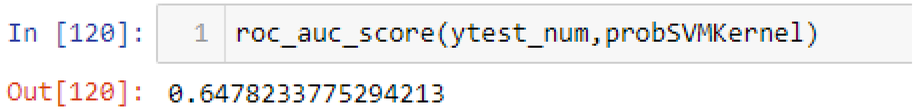


```
In [116]: roc_auc_score(ytest_num,probSVM)
Out[116]: 0.8567377362074736
```

Results of SVM with kernel:

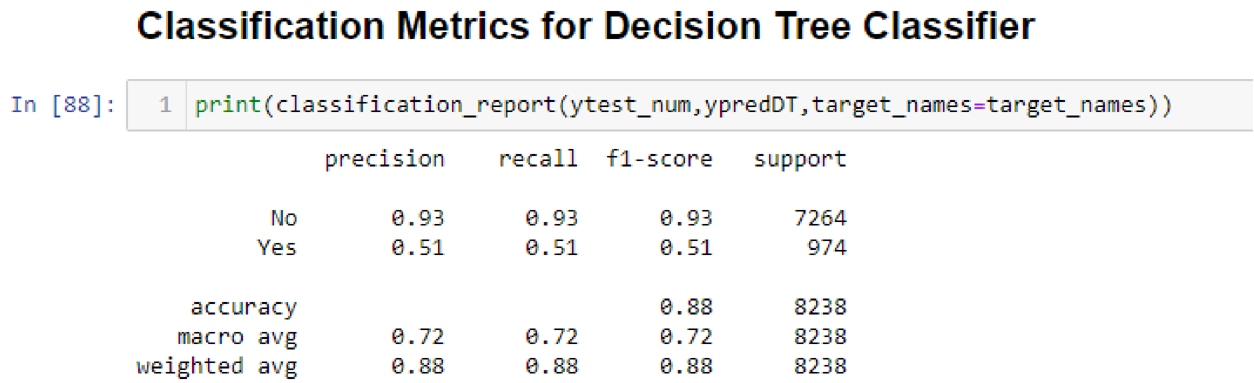


```
              precision    recall  f1-score   support

           0       0.90      0.99      0.94      7264
           1       0.75      0.18      0.30       974

    accuracy                           0.90      8238
   macro avg       0.83      0.59      0.62      8238
weighted avg       0.88      0.90      0.87      8238
```



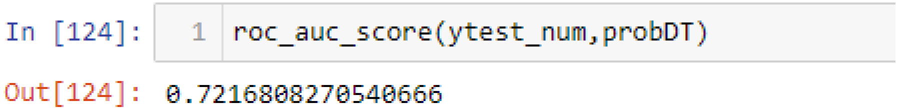
```
In [120]: roc_auc_score(ytest_num,probSVMKernel)

Out[120]: 0.6478233775294213
```

Results of Decision Tree Classifier:



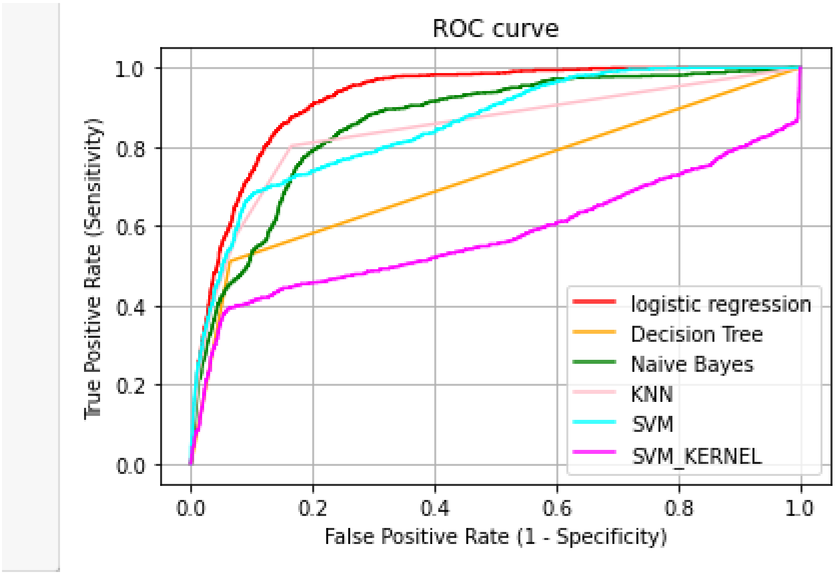
Classification Metrics for Decision Tree Classifier

```
In [88]: print(classification_report(ytest_num,ypredDT,target_names=target_names))

              precision    recall  f1-score   support

          No       0.93      0.93      0.93      7264
         Yes       0.51      0.51      0.51       974

    accuracy                           0.88      8238
   macro avg       0.72      0.72      0.72      8238
weighted avg       0.88      0.88      0.88      8238
```



```
In [124]: roc_auc_score(ytest_num,probDT)

Out[124]: 0.7216808270540666
```

Comparison of All the models:



If we observe we all the results above, we have obtained the highest AUC for the logistic regression model with an AUC of 0.92

Then we compute cross- validation scores using K-Fold cross validation techniques.

Logistic Regression: Average Cross Validation score for Logistic Regression : 0.9057359635811837

Decision Tree: Average Cross Validation score for Decision Tree :0. :0.8884370257966616

Average Cross Validation score for Naive Bayes :0.8546585735963582

Average Cross Validation score for KNN :0.903216995447648

Average Cross Validation score for SVM :0.9051593323216997

SVM with Kernel: Average Cross Validation score for SVM :0.9017905918057663

•

VII. CONCLUSION

In the current scenario, making calls through telephone is one of the subtlest way as it is one of the least money consuming way to stay in touch with the customer. In finance domain, marketing is the core coming to advertising its services and products. DSS makes use of statistical analysis for resolving the drawbacks and assists the decision makers to go for the correct decision [1]. This study investigates the application of various ML algorithms for predicting the output or to know whether the customer is willing to subscribe to the term deposit or not. The models applied to predict this output are logistic regression, KNN, Naïve Bayes Classifier, SVM, Decision trees. The data applied is Portuguese retail bank data. On applying the mentioned models on the dataset, the task of classifying is done. Then the simulation experiment takes place under several measures like precision, recall, accuracy, f1-score, support, AUC. We get fair amount of scores for each of the model, the best being for logistic regression which shows the model can have used for real-time predictions. We even deduced from information from the EDA of each features. We found that days_of_week feature will not be very helpful as all the days had similar distribution for both the classes. Similarly, age also wasn’t a good indicator because the boxplot for both overlapping. We found that duration would be very helpful to predict the output. Thus the predications have been made.

VIII. REFERENCES

[1] R. Pradeep, Dr. P. Kamaludeen "Machine Learning models for Bank Telemarketing Classification and Prediction", IJAEMA December 2019, pp. 962-967.

[2] Sergio Moro, Paulo Cortez, Paulo Rita "A data driven approach to predict the success of bank telemarketing", ELSEVIER, November 2013, pp. 22-31.

[3] https://archive.ics.uci.edu/ml/datasets/Bank+Marketing#

[4] Fereshteh Safarkhani, Sergio Moro "Improving the Accuracy of Predicting Bank Depositors behavior using a Decision Tree Classifier", Applied Sciences, September 2021, pp. 1-13.

[5] Thomas W Edgar "Research Methods for Cyber Security" ,2017.

[6] https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/

[7] https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning

[8] https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/

[9] https://www.analyticsvidhya.com/blog/2021/08/decision-tree-algorithm/