A Capstone Project Report

on

BANK MARKETING ANALYSIS AND PREDICTION

USING MACHINE LEARNING AND POWERBI

IMARTICUS
LEARNING

Submitted by

NAVEEN B

ABSTRACT

The dataset that describes different bank marketing campaigns results. Conducted campaigns were based mostly on direct phone calls, offering bank's clients to place a term deposit. If after all marking affords client had agreed to place deposit - target variable marked 'yes', otherwise 'no'. To predicting the future results of marketing companies based on available statistics and, accordingly, formulating recommendations for such companies in the future. Building a profile of a consumer of banking services (deposits). Data from a marketing campaign run by different bank is examined by the bank workers. The campaign's aim was to increase customer's subscription rates to fixed-term deposit products, such as CDs. Using knowledge from the course, a number of machine learning algorithms are implemented. The banks can successfully market these products in the most efficient way possible and with the highest possible rate of success.

Dataset and Source code Link: https://github.com/19BEC4130/Capstone-Project1.git

ACKNOWLEDGEMENTS

I am using this opportunity to express my gratitude to everyone who supported me throughout the course of my capstone project. I am thankful for their aspiring guidance, invaluably constructive criticism and friendly advice during the project work. I am sincerely grateful to them for sharing their truthful and illuminating views on a number of issues related to the project.

Further, I have fortunate to have Mr. PRASAD as my mentor. He has readily shared his immense knowledge in data science and guides me in a manner that the outcome resulted in enhancing my data skills.

I certify that the work done by me for conceptualizing and completing this project is original and authentic.

Date: 28/07/2022 Name: NAVEEN B

CERTIFICATE OF COMPLETION

I hereby certify that the project titled "BANK MARKETING ANALYSIS AND PREDICTION" was undertaken and completed the project under my supervision by Mr. Prasad from the batch.

Mentor : Mr. Prasad

Date : 28/07/2022

Place : Karur

TABLE OF CONTENT

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Today organizations, which hire marketing management are analysis of organization's marketing data which is one of the most typical applications of data science and machine learning. Bank marketing is the practice of attracting and acquiring new customers through traditional media and digital media strategies. The use of these media strategies helps determine what kind of customer is attracted to a certain institution. Such analysis will definitely be a good contribution to the institution. Prediction of the results of the marketing campaign for each customer and clarification of factors which affect the campaign results. This helps to find out the ways how to make marketing campaigns more efficient. Finding out customer segments, using data for customers, who subscribed to term deposit. This helps to identify the profile of a customer, who is more likely to acquire the product and develop more targeted marketing campaigns. One result of this is that marketing campaigns are growing evermore pervasive in our daily lives. This glut of advertising has forced businesses to compete for the attention of a populace that has an ever-growing number of distractions. Based on this data, machine learning models will predict which clients will subscribe and what banks can do to increase the rate of subscription.

1.1 PROBLEM STATEMENT:

Bank marketing is a necessity nowadays, and almost every individual is linked with a government or private bank. Factors determining the amount of deposit vary from bank to bank. Improve bank marketing of a bank by analysing their past marketing campaign data and recommending which customer to target. Also, people have different age of the details that the Reserve bank of India provide certain interest to those below poverty line. It is very complex method and some rural people either deposit some private bank or do not invest money at all. Prediction is premature and does not comply with any particular bank so it must not be only criteria in selection of a predicting deposit.

1.2 OBJECTIVE:

The aim of this project is to devise such a machine leaning prediction algorithm, the bank can better target its customers and channelize its marketing efforts. Predicting the future results of marketing campaigns based on available statistics and, accordingly, formulating recommendations for such companies in the future. Building a profile of a consumer of banking services (deposits).

CHAPTER 2

DATA PREPARATION AND UNDERSTANDING

Data preparation is the process of cleaning and transforming raw data before building predictive models. Dataset for Bank Marketing prediction is found from various sources. In this project, the dataset is imported from MySQL workbench and the visualization of data in the bank marketing analysis and prediction is carried out by Power BI. The following are the definition of columns present in the dataset,

Input Variables

1.Age: numerical value

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: 'primary', 'secondary', 'Teritary', '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')

8.Related with the last contact of the current campaign:

9.Contact: contact communication type (categorical: 'cellular', 'telephone')

10.Month: last contact month of year (categorical: 'Jan', 'feb', 'mar', ..., 'nov', 'dec')

11.Day_of_week: last contact day of the week ('numerical')

12.Duration: last contact duration, in days (numeric).

Output variable (Target Variable)

14.Deposit: has the client subscribed a term deposit? (Binary: 'yes', 'no')

Read Data from MySQL Database:

```
df=pd.DataFrame(mycursor.fetchall())
df=df.copy()
df
```

	Id	age	job	marital	education	default	Acc_balance	housing	loan	contact	day	month	duration	campaign	poutcome	deposit
0	1001	45	management	married	tertiary	no	2143	yes	no	unknown	5	may	261	1	unknown	no
1	1002	44	technician	single	secondary	no	29	yes	no	unknown	5	may	151	1	unknown	no
2	1003	33	entrepreneur	married	secondary	no	2	yes	yes	unknown	5	may	76	1	unknown	no
3	1004	47	blue-collar	married	unknown	no	1506	yes	no	unknown	5	may	92	1	unknown	no
4	1005	33	unknown	single	unknown	no	1	no	no	unknown	5	may	198	1	unknown	no
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4995	5996	21	student	single	secondary	no	232	yes	no	unknown	21	may	110	2	unknown	no
4996	5997	31	management	single	tertiary	no	385	yes	no	unknown	21	may	304	1	unknown	no
4997	5998	34	blue-collar	married	secondary	no	5304	yes	no	unknown	21	may	51	2	unknown	no
4998	5999	35	management	single	tertiary	no	71	yes	no	unknown	21	may	836	12	unknown	no
4999	6000	32	technician	single	unknown	no	317	yes	no	unknown	21	may	394	2	unknown	no

5000 rows × 16 columns

Fig No 2.1: Read Data from MySQL Database

The above image is the dataset in the MySQL workbench which is imported to jupyter notebook using the following code. As we need to prepare the data for training and testing.

Understand the Dataset:

In [8]:
```
#display first few rows of data
df.head()
```

Out[8]:

	Id	age	job	marital	education	default	Acc_balance	housing	loan	contact	day	month	duration	campaign	poutcome	deposit
0	1001	45	management	married	tertiary	no	2143	yes	no	unknown	5	may	261	1	unknown	no
1	1002	44	technician	single	secondary	no	29	yes	no	unknown	5	may	151	1	unknown	no
2	1003	33	entrepreneur	married	secondary	no	2	yes	yes	unknown	5	may	76	1	unknown	no
3	1004	47	blue-collar	married	unknown	no	1506	yes	no	unknown	5	may	92	1	unknown	no
4	1005	33	unknown	single	unknown	no	1	no	no	unknown	5	may	198	1	unknown	no

In [9]:
```
#display last few rows of data
df.tail()
```

Out[9]:

	Id	age	job	marital	education	default	Acc_balance	housing	loan	contact	day	month	duration	campaign	poutcome	deposit
4995	5996	21	student	single	secondary	no	232	yes	no	unknown	21	may	110	2	unknown	no
4996	5997	31	management	single	tertiary	no	385	yes	no	unknown	21	may	304	1	unknown	no
4997	5998	34	blue-collar	married	secondary	no	5304	yes	no	unknown	21	may	51	2	unknown	no
4998	5999	35	management	single	tertiary	no	71	yes	no	unknown	21	may	836	12	unknown	no
4999	6000	32	technician	single	unknown	no	317	yes	no	unknown	21	may	394	2	unknown	no

Fig No 2.2: Understand the Dataset

CHAPTER 3

EXPLORATORY DATA ANALYSIS

In Exploratory Data Analysis, consists of Inspecting and cleansing, transforming, and modeling data. To visualize the data shows the range between the rows and columns that are taken in the data. To analysis the age shows the differential formats according to their ages. Jumping into the modelling part without knowing anything about the problem is not a good idea. So as a start, we will do some EDA to find out more about what kind of data are we dealing with, if there is any pattern to the data. We will do some univariate analysis to find out which features can help us in our classification task and which variables are not that important. The first step is to load the dataset into a data frame for easy Analysis and preparation using the pandas package The clients across the different features, both categorial (marital status, job type, education, etc.) and numeric (age, number of days since previous contact, etc.). The target variable is a binary “Yes” (client subscribed) or “No” (client did not subscribe). This feature measures the length of the phone call between the bank’s marketing representative and the customer. Since this time cannot be known until after the call has ended (when the outcome for that customer is already known), including it in a predictive model would not provide realistic results.

HISTOGRAM:

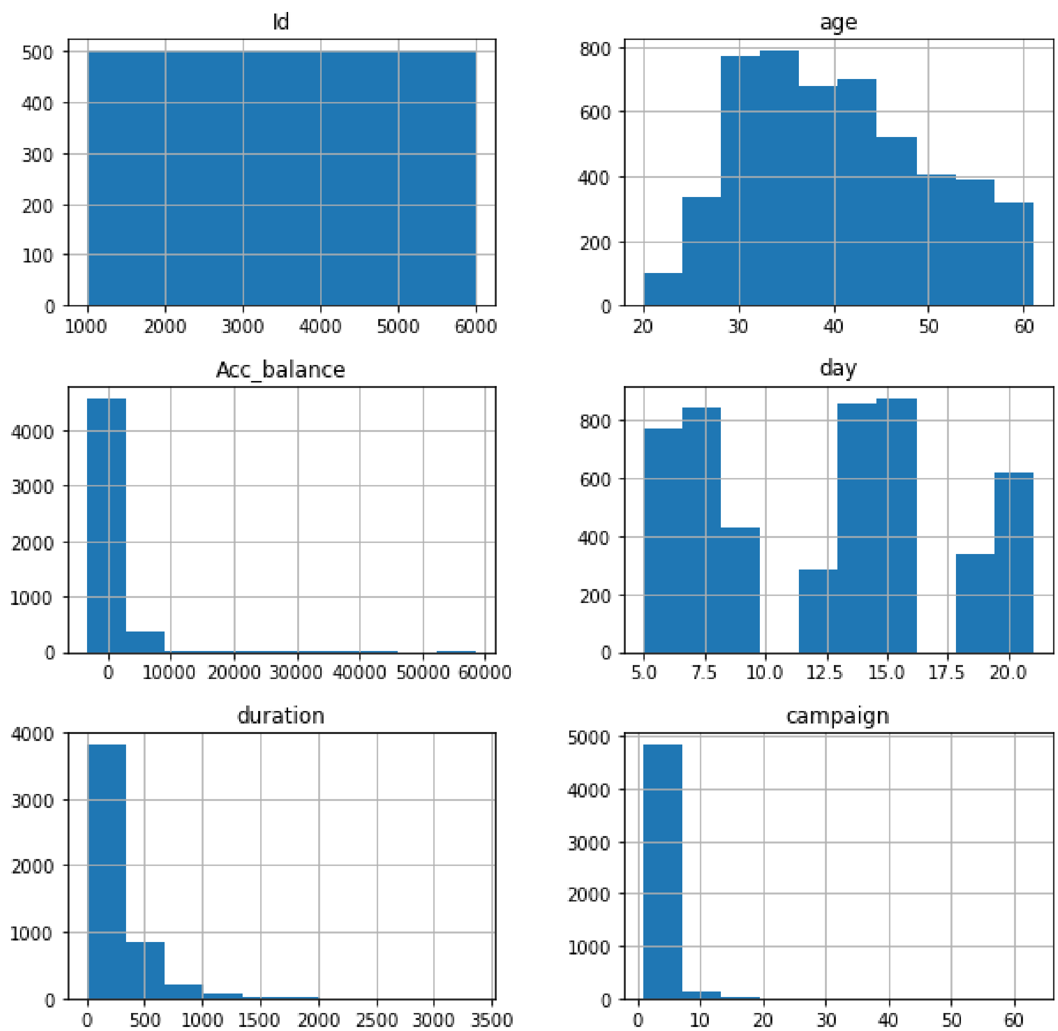

Fig No 3.1: Histogram

We plot a histogram to graphical display the data using bars of different heights. In a histogram, each bar groups numbers into ranges.

Missing Values:

Many of these features contain unknown values so the next question is how to deal with this missing data. Instead, these missing values are imputed using other independent variables to infer the missing values.

2.1 Checking Missing Values

In [18]:
```
# Check if missing values exist or not
df.isnull().sum()
```

Out[18]:
```
Id             0
age            0
job            0
marital        0
education      0
default        0
Acc_balance    0
housing        0
loan           0
contact        0
day            0
month          0
duration       0
campaign       0
deposit        0
dtype: int64
```

Fig No 3.2: Missing Values

While this does not guarantee that all the missing data will be restored, a majority of it will be. For instance, cross-tabulation between 'job' and 'education' was used based on the hypothesis that a person's job will be influenced by their education. A similar cross-tabulation process was carried out for the 'house ownership' and 'loan status' features. It's important to note that in making these imputations, care was taken to ensure the correlations made sense in the real world. Python provides quickness, ease of modifiability and ease of replacement of values throughout the dataset thanks to this tool.

3.1 VISUALIZING CATAGORICAL VARIABLE

Bar Plot:

To represents the category of data with rectangular bars with lengths and heights that is proportional to the values which they represent. This graph shows the number of observations within each given interval.

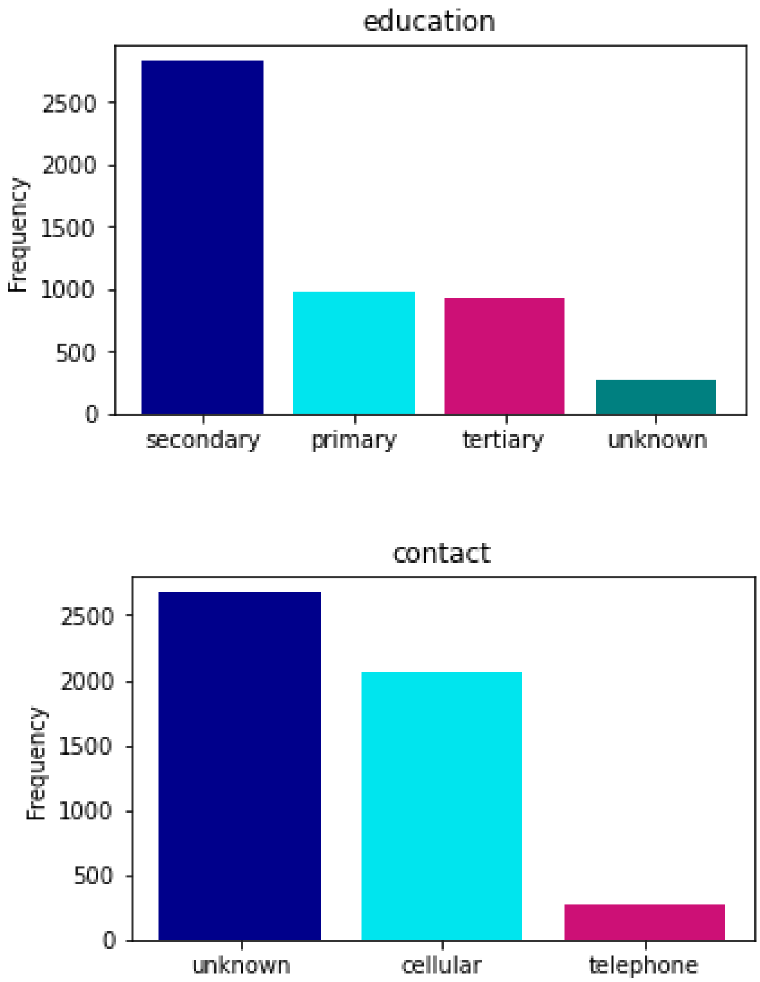

Fig No 3.3: Bar Plot

3.2 VISUALIZING NUMERICAL VALUE

Distplot:

The distplot represents the univariate distribution of data i.e., data distribution of a variable against the density distribution.

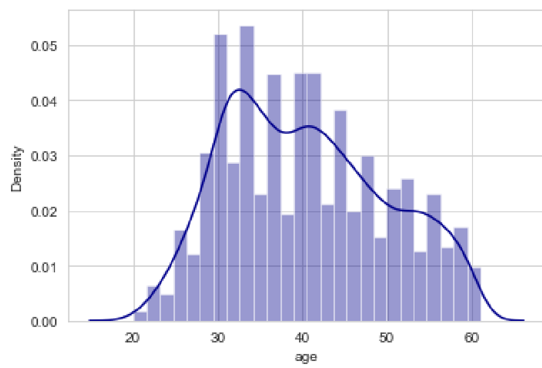

Fig No 3.4: DistPlot

Box Plot:

To display the summary of the set of data values having properties like minimum, first quartile, median, third quartile and maximum.

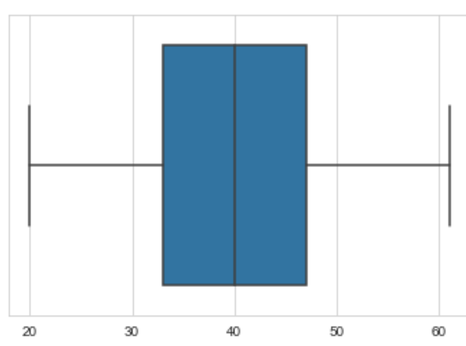

Fig No 3.5: Box Plot

3.3 VISUALIZING WITH POWER BI

Get your Power BI analytics in a Jupyter notebook with the new powerbiclient Python package. Install the powerBI Client package in your jupyter Notebook environment. The new package lets you embed Power BI reports in Jupyter notebooks easily. To Embed the power BI Embedded analytics report in an output cell. It'll be able to export data from visuals in a Power BI report to the Jupyter notebook for in-depth data exploration. It can also filter the report for quick analysis or use bookmarks to apply a saved view.

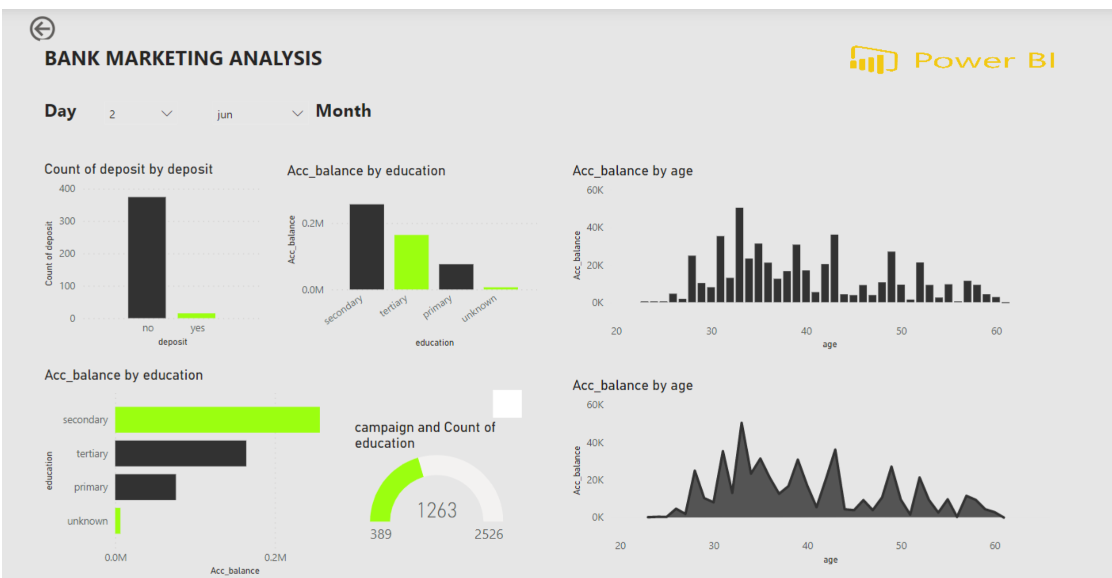

Fig No 3.6: PowerBi Analysis

CHAPTER 4

CORRELATION

Correlation is a statistic that measures the degree to which two variables move with each other. A correlation coefficient near 1 indicates the strong relationship between them; a weak correlation indicates the extent to which one variable increases as the other decreases. Correlation among multiple variables can be represented in the form of a matrix. This allows us to see which variables are correlated.

4.1 HEATMAP FOR CHECK CORRELATION OF DATA

A heatmap was created to show there is strong correlation between the target variable and any independent variables. This graphic was created using Python's seaborn package and the specially written function draw heatmap.

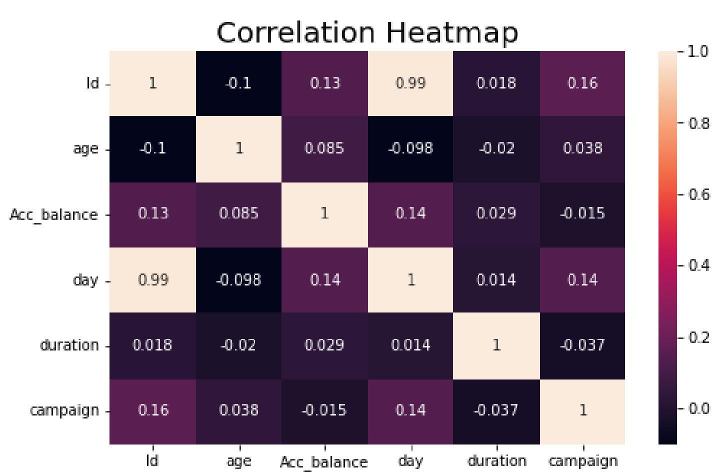

Fig No 4.1: Heat Map

CHAPTER 5

CHECKING OUTLIERS

Outliers detection is the process of identifying data that have extreme values compared to rest of the data distribution. We are checking the outliers for only the required columns in our dataset. After the execution of IQR method we are supposed to find that outliers are present in our dataset. Outliers can find their way into a dataset naturally through variability, or they can be the result of issues like human error, faulty equipment, or poor sampling. Outliers can have a big impact on statistical analysis and machine learning because they impact calculations like mean and standard deviation, and they can skew hypothesis tests.

In [40]:

```
# boxplot on numerical features to find outliers

plt.figure(figsize=(20,60), facecolor='white')
plotnumber =1
for numericVar1 in numericVar1:
    ax = plt.subplot(12,3,plotnumber)
    sns.boxplot(df[numericVar1])
    plt.xlabel(numericVar1)
    plotnumber+=1
plt.show()
```

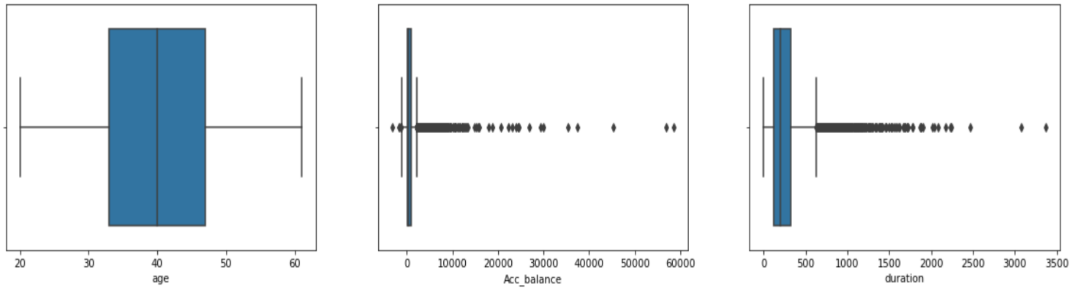

Fig No 5.1: Outliers Detection

CHAPTER 6

MODEL SELECTION

Model selection is the process of choosing one among many candidate models for a predictive modeling problem. The dataset is divided into training data and test data with the intention of using the training data to find the parameters of the particular model being used (fitting the model on the training data) and then applying this to the test data to determine the model's performance and to draw conclusions about its predictive capability.

SPLITTING DATA INTO TRAIN AND TEST SPLITS:

The train dataset and test dataset loaded separately in separate variables. Train dataset contains a greater number of data belongs to each class. Test dataset contains data belongs to each class.

6.1.1 Splitting the data into x and y

```
In [50]: x=df.drop('deposit', axis=1)
         y=df['deposit']
```

6.1.2 Splitting the data into Train and Test Splits

```
In [51]: #LogisticRegression
         #splitting
         from sklearn.model_selection import train_test_split
         x_test,x_train,y_test,y_train=train_test_split (x,y,test_size=0.25,random_state=0)
```

```
In [52]: #fitting
         from sklearn.linear_model import LogisticRegression
         model=LogisticRegression()
         model.fit(x_train,y_train)

Out[52]: LogisticRegression()
```

FITTING MODELS TO DATA:

I have fitted the model with training dataset for each data. I have tested my model with validation data which have the given the accuracy of nearly 96%.

PREDICTING THE FUTURE RESULTS:

Based on this data, machine learning models will predict which clients will subscribe and what banks can do to increase the rate of subscription. Prediction is premature and does not comply with any particular bank so it must not be only criteria in selection of a predicting deposit. Predicting the future results of marketing campaigns based on available statistics and accordingly, formulating recommendations for such companies in the future.

6.1 LOGISTIC REGRESSION:

Python provides the package sklearn.linear model.LogisticRegression for Logistic Regression. LR a is well known classification model. The logistic regression algorithm uses the same decision boundary with bit modifications. Logistic regression is used because classification is not exactly a linear function and using linear regression produces an output within $[-\infty, +\infty]$ while the probability has to be within [0, 1]. The logistic function itself does output the probability of an instance belonging to the positive class hence overcoming the drawbacks of classification using a linear model. I have fitted the model with training dataset for each data. I have tested my model with validation data which have the given the accuracy of nearly 96.82%.

```
In [55]: #to measure the accuracy of model
         from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
         accuracy_score(y_test,y_pred)

Out[55]: 0.9682666666666667

In [56]: confusion_matrix(y_test,y_pred)

Out[56]: array([[3613,   21],
                [  98,   18]], dtype=int64)

In [57]: A=classification_report(y_test,y_pred)
         print(A)

              precision    recall  f1-score   support

           0       0.97      0.99      0.98      3634
           1       0.46      0.16      0.23       116

    accuracy                           0.97      3750
   macro avg       0.72      0.57      0.61      3750
weighted avg       0.96      0.97      0.96      3750

In [58]: accuracies = {}
         accuracy_score(y_test,y_pred)
         Logistic_acc = accuracy_score(y_test,y_pred)*100
         print("Logistic Regression accuracy:",Logistic_acc)
         accuracies['Logistic Regression']=Logistic_acc

         Logistic Regression accuracy: 96.82666666666667
```

6.2 RANDOM FOREST CLASSIFIER:

Python provides the package Sklearn.ensemble.RandomForestClassifier for the Random Forest classifier. Random forest classifiers are one of the ensembles learning methods for classification. It constructs multiple decision trees (a “forest”) at the training time and the output prediction is the class which is the mode of the predictions made by the individual decision trees in the ensemble. Random forests are an ensemble learning method that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set. I have fitted the model with training dataset for each data. I have tested my model with validation data which have the given the accuracy of nearly 96.96%.

```
In [63]: #to measure the accuracy of model
         from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
         print(accuracy_score(y_test,y__pred))

         0.9696

In [64]: #making confusion matrix
         confusion_matrix(y_test,y__pred)

Out[64]: array([[3630,    4],
                [ 110,    6]], dtype=int64)

In [65]: #classification report
         B=classification_report(y_test,y__pred)
         print(B)

                       precision    recall  f1-score   support

                    0       0.97      1.00      0.98      3634
                    1       0.60      0.05      0.10       116

             accuracy                           0.97      3750
            macro avg       0.79      0.53      0.54      3750
         weighted avg       0.96      0.97      0.96      3750


In [66]: Random_Forest_acc = accuracy_score(y_test,y__pred)*100
         print("Random Forest accuracy:",Random_Forest_acc)
         accuracies['Random Forest']=Random_Forest_acc

         Random Forest accuracy: 96.96000000000001
```

6.3 DECISION TREE CLASSIFIER:

Python provides the package sklearn.tree.DecisionTreeClassifier for the decision tree classifier. Decision trees are a simple yet effective method for classification. Using a tree structure, this algorithm splits the data set based on one feature at every node until all the data in the leaf belongs to the same class. A decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements. I have fitted the model with training dataset for each data. I have tested my model with validation data which have the given the accuracy of nearly 96.02%.

```
In [69]: #to measure the accuracy of model
         from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
         accuracy_score(y_test,ypred)

Out[69]: 0.9602666666666667

In [70]: confusion_matrix(y_test,ypred)

Out[70]: array([[3573,   61],
                [  88,   28]], dtype=int64)

In [71]: C=classification_report(y_test,ypred)
         print(C)

                       precision    recall  f1-score   support

                    0       0.98      0.98      0.98      3634
                    1       0.31      0.24      0.27       116

             accuracy                           0.96      3750
            macro avg       0.65      0.61      0.63      3750
         weighted avg       0.96      0.96      0.96      3750


In [82]: DecisionTree_acc = accuracy_score(y_test,ypred)*100
         print("Decision_Tree:",DecisionTree_acc)
         accuracies['Decision_Tree']=DecisionTree_acc

         Decision_Tree: 96.02666666666667
```

6.4 K-NEAREST NEIGHBOR:

It is one of the simplest and widely used classification algorithms in which a new data point is classified based on similarity in the specific group of neighboring data points. This gives a competitive result. For a given data point in the set, the algorithms find the distances between this and all other **K** numbers of datapoint in the dataset close to the initial point and votes for that category that has the most frequency. I have fitted the model with training dataset for each data. I have tested my model with validation data which have the given the accuracy of nearly 96.8%.

```
In [75]: #to measure the accuracy of model
         from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
         accuracy_score(y_test,pred)

Out[75]: 0.968

In [76]: #making confusion matrix
         confusion_matrix(y_test,pred)

Out[76]: array([[3620,   14],
                [ 106,   10]], dtype=int64)

In [77]: #classification report
         D=classification_report(y_test,pred)
         print(D)

                       precision    recall  f1-score   support

                    0       0.97      1.00      0.98      3634
                    1       0.42      0.09      0.14       116

             accuracy                           0.97      3750
            macro avg       0.69      0.54      0.56      3750
         weighted avg       0.95      0.97      0.96      3750


In [78]: KNN_acc = accuracy_score(y_test,pred)*100
         print("KNN accuracy:",KNN_acc)
         accuracies['KNN']=KNN_acc

         KNN accuracy: 96.8
```

CHAPTER 7

COMPARISON OF ACCURACY

```
In [58]: accuracies = {}
         accuracy_score(y_test,y_pred)
         Logistic_acc = accuracy_score(y_test,y_pred)*100
         print("Logistic Regression accuracy:",Logistic_acc)
         accuracies['Logistic Regression']=Logistic_acc

         Logistic Regression accuracy: 96.82666666666667
```

```
In [66]: Random_Forest_acc = accuracy_score(y_test,y__pred)*100
         print("Random Forest accuracy:",Random_Forest_acc)
         accuracies['Random Forest']=Random_Forest_acc

         Random Forest accuracy: 96.96000000000001
```

```
In [82]: DecisionTree_acc = accuracy_score(y_test,ypred)*100
         print("Decision_Tree:",DecisionTree_acc)
         accuracies['Decision_Tree']=DecisionTree_acc

         Decision_Tree: 96.02666666666667
```

```
In [78]: KNN_acc = accuracy_score(y_test,pred)*100
         print("KNN accuracy:",KNN_acc)
         accuracies['KNN']=KNN_acc

         KNN accuracy: 96.8
```

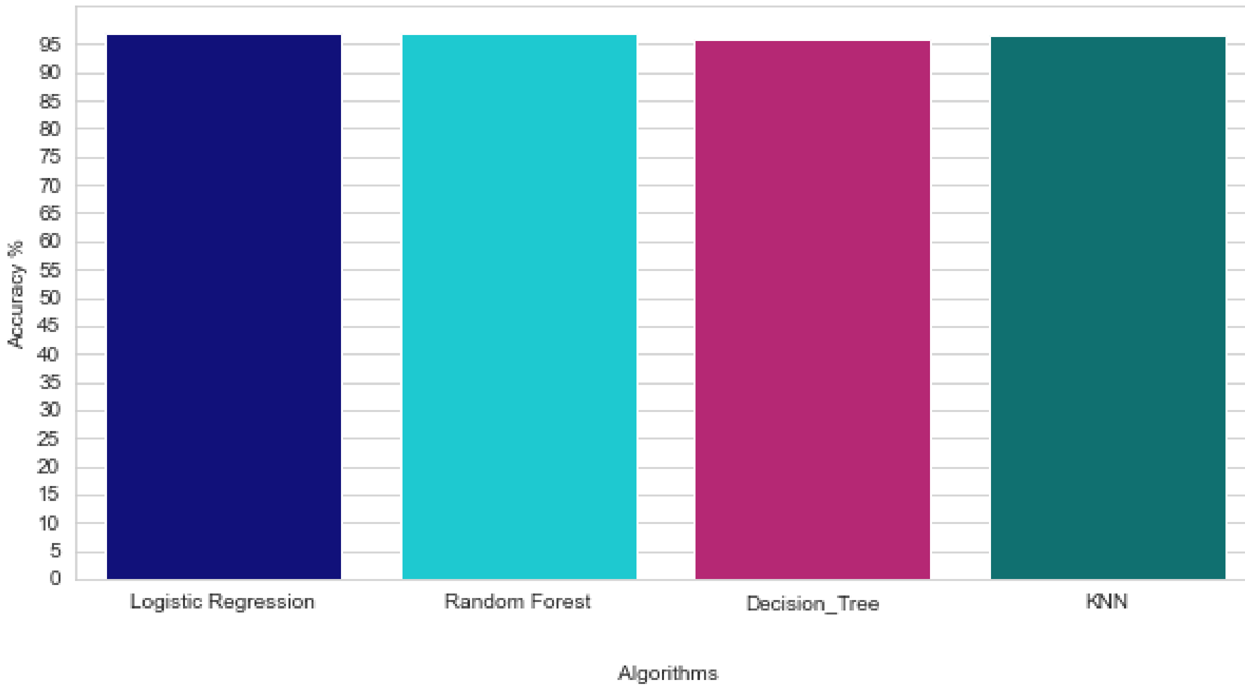

Fig No 7.1: Accuracy Comparison

```
Logistic_acc =  96.82666666666667
Random_Forest_acc =  96.96000000000001
DecisionTree_acc =  96.02666666666667
KNN_acc =  96.8
```

CHAPTER 8

CONCLUSION

From this project, we learned how banks can improve their marketing campaigns by focusing their efforts on certain prime-grade clients and also how they can recognize market conditions which are favorable to increase client subscription for the fixed-term products they are offering. All of this was possible by implementing data science and machine learning methods in Python. Tools such as data frames, arrays, for loops, etc. were all critical for the success of this project. A large number of other tools and techniques from the Python for Data Science course were used and these were invaluable for making our analyses and predictions. This project demonstrated how powerful Python can be for data science applications. The customer's account balance has a huge influence on the campaign's outcome. People with account balance above certain amount are more likely to subscribe for term deposit, so future address those customers. The customer's age affects campaign outcome as well. Future campaigns should concentrate on customers from age categories below 30 years old and above 50 years old. Number of contacts with the customer during the campaign is also very important. The number of contacts with the customer shouldn't exceed 4.

CHAPTER 9

REFERENCE

[1] Foster Provost and Tom Fawcett, Data Science for Business. O'Reilly Media, 2013.

[2] W. D. Reekie, 'Marketing in Banking', The Bankers' Magazine, (UK) (Aug. 1972) pp. 56-60 and (Sept. 1972) pp. 97-100.

[3] A. Meidan, 'The Roles of Marketing Management in Banking', Quarterly Review of Marketing, vol. 8, no. 3 (Apr. 1983) pp. 11-18.

[4] R. Hodges and 1. Tilman, Bank Marketing: Text and Cases (AddisonWesley, 1968)

[5] P. Kotler, Marketing Management: Analysis, Planning and Control, 4th edn (Prentiee-Hall, 1980) p. 31.

[6] R. B. Stall, 'Marketing in a Serviee Industry', Atlanta Economic Review (May - June 1968).