**Aim :-**

The Aim of this project is to generate models provided in the dataset using various classification and regression techniques in R. Analyze the results presents in the model and optimize model which is most suitable for this dataset.

**Dataset Description : -** Dataset contains information about information used for bank marketing. Bank Marketing dataset contains 21 variables and 4119 Rows. This Dataset is provided by a Portuguese banking institution based on bank marketing campaigns and based on real world problem.

**Training Dataset : 3530**

**Testing Dataset : 589**

**Link to Dataset :**

**http://archive.ics.uci.edu/ml/datasets/Bank+Marketing/**

**Data Cleaning** :- Dataset contains 21 attributes in the training dataset and 20 in the test dataset & one attribute in the target test dataset**.** No N/A values found when summary of all dataset performed . Attribute List as follows :-

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')

Following code used for initial variable analysis :-

**# Setting up default working directory path of the project**

>setwd("D:\\Plan\\DataScience\\project\\Bank project\\bank-additional")

**#Getting working directory**

**> getwd()**

**# Read CSV file from the working directory**

**>** DataBank<-DataBank<-read.csv("bank-additional.csv", sep=";", header=TRUE,quote = "\" ")

**# file: file name**

**#** header**: 1st line as header or not, logical  
 # sep: field separator**

**# quote: quoting characters**

**# Determine structure of CSV file**

str(DataBank)

'data.frame': 4119 obs. of 21 variables:

$ age : int 30 39 25 38 47 32 32 41 31 35 ...

$ job : Factor w/ 12 levels "admin.","blue-collar",..: 2 8 8 8 1 8 1 3 8 2 ...

$ marital : Factor w/ 4 levels "divorced","married",..: 2 3 2 2 2 3 3 2 1 2 ...

$ education : Factor w/ 8 levels "basic.4y","basic.6y",..: 3 4 4 3 7 7 7 7 6 3 ...

$ default : Factor w/ 3 levels "no","unknown",..: 1 1 1 1 1 1 1 2 1 2 ...

$ housing : Factor w/ 3 levels "no","unknown",..: 3 1 3 2 3 1 3 3 1 1 ...

$ loan : Factor w/ 3 levels "no","unknown",..: 1 1 1 2 1 1 1 1 1 1 ...

$ contact : Factor w/ 2 levels "cellular","telephone": 1 2 2 2 1 1 1 1 1 2 ...

$ month : Factor w/ 10 levels "apr","aug","dec",..: 7 7 5 5 8 10 10 8 8 7 ...

$ day\_of\_week : Factor w/ 5 levels "fri","mon","thu",..: 1 1 5 1 2 3 2 2 4 3 ...

$ duration : int 487 346 227 17 58 128 290 44 68 170 ...

$ campaign : int 2 4 1 3 1 3 4 2 1 1 ...

$ pdays : int 999 999 999 999 999 999 999 999 999 999 ...

$ previous : int 0 0 0 0 0 2 0 0 1 0 ...

$ poutcome : Factor w/ 3 levels "failure","nonexistent",..: 2 2 2 2 2 1 2 2 1 2 ...

$ emp.var.rate : num -1.8 1.1 1.4 1.4 -0.1 -1.1 -1.1 -0.1 -0.1 1.1 ...

$ cons.price.idx: num 92.9 94 94.5 94.5 93.2 ...

$ cons.conf.idx : num -46.2 -36.4 -41.8 -41.8 -42 -37.5 -37.5 -42 -42 -36.4 ...

$ euribor3m : num 1.31 4.86 4.96 4.96 4.19 ...

$ nr.employed : num 5099 5191 5228 5228 5196 ...

$ y : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...

**# Determine Number of rows in the Dataset**

**> nrow(DataBank)**

**[1] 4119**

**# Number of the columns in the dataset**

**> ncol(DataBank)**

**[1] 21**

**# Determine if Dataset has any missing or not applicable value**

**> sum(is.na(DataBank))**

**[1] 0**

# import library caTools for splitting dataset

> library(caTools)

# splitting dataset into two part

# a 90% of training dataset

# b 10% testing dataset

> split<-sample.split(DataBank,SplitRatio = 0.9)

> training<-subset(DataBank,split==1)

> testing<-subset(DataBank,split==0)

> nrow(testing)

[1] 589

> nrow(training)

[1] 3530

# prepare model with training dataset

**Naive Bayes Classification Algorithm :**

**it is based on Bayesian theorem ,** it is based onassumption of

independence among predictors. Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods

**Training and Testing dataset is taken from above mentioned classification algorithm L- logistic egression**

> nrow(training)

[1] 3530

> nrow(testing)

[1] 589

>library(e1071)

> model<-naiveBayes(y~.,data = training)

**interpretation from NaiveBayes algorithm**

**Prior Probabilities** from this class for example 89 % of the people from the training dataset are not interested to subscribe term deposit scheme and only 10% person subscribed to term deposit.

A-priori probabilities:

Y

no yes

0.890085 0.109915

below represent condition al probabilities for example for age ( being numerical data). first column represent mean and second represent standard deviation. so from this data it is clear that persons having average age 39 are not subscribed to the term deposit and only average age 42 are subscribed to term deposit. second column represent SD i.e. it shows variation in age column data for example there is +-9 in the variation of age column not subscribed to term deposit.

Conditional probabilities:

age

Y [,1] [,2]

no 39.87683 9.913239

yes 42.33763 13.613602

for categorical columns it represent conditional probabilities

for example probability (55.6%) of married people subscribed to term deposit is less than the probability (60.9 %) not subscribed to term deposit.

marital

Y divorced married single unknown

no 0.114258434 0.609802673 0.273074475 0.002864418

yes 0.100515464 0.556701031 0.342783505 0.000000000

job

Y admin. blue-collar entrepreneur housemaid management retired self-employed services student technician unemployed

no 0.239656270 0.223424570 0.036600891 0.026734564 0.077975812 0.035646085 0.041693189 0.097071929 0.017823043 0.168682368 0.025779758

yes 0.273195876 0.139175258 0.018041237 0.020618557 0.069587629 0.095360825 0.028350515 0.079896907 0.043814433 0.185567010 0.038659794

job

Y unknown

no 0.008911521

yes 0.007731959

education

Y basic.4y basic.6y basic.9y high. school illiterate professional. course university. degree unknown

no 0.1066199873 0.0550604710 0.1451304901 0.2253341820 0.0003182686 0.1288987906 0.3001273074 0.0385105029

yes 0.0902061856 0.0335051546 0.1005154639 0.2139175258 0.0000000000 0.1546391753 0.3505154639 0.0567010309

default

Y no unknown yes

no 0.7963080840 0.2033736474 0.0003182686

yes 0.8969072165 0.1030927835 0.0000000000

housing

Y no unknown yes

no 0.44684914 0.02705283 0.52609803

yes 0.44072165 0.02319588 0.53608247

loan

Y no unknown yes

no 0.81063017 0.02705283 0.16231700

yes 0.81185567 0.02319588 0.16494845

contact

Y cellular telephone

no 0.6250796 0.3749204

yes 0.8273196 0.1726804

month

Y apr aug dec jul jun mar may nov oct sep

no 0.048695099 0.159770847 0.003182686 0.178548695 0.128262253 0.005410567 0.346276257 0.106938256 0.012730745 0.010184596

yes 0.074742268 0.141752577 0.025773196 0.126288660 0.162371134 0.067010309 0.198453608 0.090206186 0.056701031 0.056701031

day\_of\_week

Y fri mon thu tue wed

no 0.1826862 0.2081477 0.2103756 0.2081477 0.1906429

yes 0.1752577 0.2139175 0.2242268 0.2036082 0.1829897

duration

Y [,1] [,2]

no 219.2403 193.6475

yes 549.0567 400.0687

campaign

Y [,1] [,2]

no 2.585614 2.664784

yes 2.023196 1.414936

pdays

Y [,1] [,2]

no 982.2616 127.8080

yes 781.4175 411.3392

previous

Y [,1] [,2]

no 0.1441757 0.4281143

yes 0.5695876 0.9606215

poutcome

Y failure nonexistent success

no 0.10725652 0.87905792 0.01368555

yes 0.15463918 0.64690722 0.19845361

emp.var.rate

Y [,1] [,2]

no 0.2421388 1.486908

yes -1.1951031 1.651920

cons.price.idx

Y [,1] [,2]

no 93.60090 0.5642712

yes 93.41151 0.6904072

cons.conf.idx

Y [,1] [,2]

no -40.56238 4.408774

yes -39.68969 5.969111

euribor3m

Y [,1] [,2]

no 3.803346 1.643257

yes 2.138722 1.771841

nr.employed

Y [,1] [,2]

no 5175.533 66.23501

yes 5092.863 90.26600

**predicting the result from the testing dataset**

**>** result\_set<-predict(model,testing)

result\_set

no yes

499 90

it is clear from testing dataset

% of people subscribe to term deposit from testing dataset = 15.28%

% of people not subscribe to term deposit from testing dataset = 84.71%

result\_set<-predict(model,testing,type = "raw")

result\_set :- will return the probabilities of subscribing to term deposit from testing dataset.

result\_set<-predict(model,testing)

table(ActualValue=testing$y,PredictedValue=result\_set)

PredictedValue

ActualValue no yes

no 470 41

yes 40 38

Accuracy of this model = 508/589 = 86.24 %

precision = 38/38+40 = 48.71%

Recall(% (yes) items selected) 38/38+41 = 48.1 %

Specapacity (% of no items selected )= 470/470+41 = 93.81%