PH 125.9x Chose Your Own Project- Bank Marketing

Mayank Gaur

17/11/2020

1. Executive Summary

This report is made for Harvardx Data Science Professional Certificate course. It is part of the CYO (Chose Your Own) project assessment of the Capstone project module. The data is related to the direct marketing campaign of a Portuguese banking institution. The classification goal is to predict if the client will subscribe a term deposit offered by the banking institution. The project also aims to suggest how the effectiveness of the marketing campaign can be improved by suggesting which attributes of the client are the most important to focus on from a marketing campaign perspective. The marketing campaign to which the project is related to is based on the calls made to the clients in order to ascertain if the clients would opt for the term deposit.The attributes related to client information such as age, job, balance, housing loan, whether having personal loan, housing loan and attributes related to current marketing campaign in terms of number of contacts performed during the current campaign, month in which the contact was made were analyzed and interpreted in relation with target variable.

The project comprised of two sections. In the first section, the data was automatically downloaded from the github repository, it was processed, feature engineering was performed for preparing the data for further analysis. The data set was then visualized and analyzed. Distribution of different attributes- numerical and categorical were studied and also their impact on the target categorical variable (Subscription to term deposit) was analyzed. Modeling analysis was then performed by deploying different Machine Learning models. For the purpose of machine learning, the data set was split into train and test set. The train set was used to train the model and test set was used exclusively to evaluate the performance of the model. The optimum model or the best model was selected by interpreting different indicators such as Model Accuracy, Sensitivity.The Random Forest model provided the best accuracy of 71% and sensitivity of 62% and was selected as the best performing model.

2. Introduction

Marketing campaigns play an important role in the globalized commercial environment. There is stiff competition in private banking industry to increase the customer base. The banks design new financial products, come up with attractive loan repayment schemes. In this context, marketing campaigns play an important role for banks to reach out to clients with their financial products. Banks are increasingly making use of advanced machine learning and data science techniques to continuously improve their marketing strategies.

In different sectors such as banking, insurance, e-commerce, companies try to use marketing campaigns to reach out to potential customers and sell them their products. In case of banking industry, the product could be the new term deposit scheme launched by the bank or financial institution. In direct marketing campaigns, the companies try to contact clients directly through phone campaigns or other modes. Sometimes more than one communication could be made before the customer finally decides whether he would subscribe to the deposit or not. For a bank or Financial Institution using direct market campaigns, there would be some important attributes or factors that could determine if the client will opt for the financial product. The Banks and Financial institutions look to analyze the marketing campaigns performed and identify patterns that will help to improve strategies for future campaigns.

3. Overview

Data set used for the project was publicly available at the Kaggle website (https://www.kaggle.com/janiobachmann/bank-marketing-dataset). The R script automatically downloaded the data set from Git hub repository. 10% of the data set was carved out as test set so that there is as much data as possible to train. Train set was used for training the model and test set was used to evaluate the performance of the model.

4.Initial Data Exploration

4.1 Loading the data set The data set (bank_marketing) is loaded automatically from the Github repository as below:-

```
bank_marketing<-read.csv(
  "https://raw.githubusercontent.com/mgaur2009/Capstone-Own-Project/master/bank.csv")
```

The first six rows of the data set are: -

```
head(bank_marketing)
```

```
##   age        job marital education default balance housing loan contact day
## 1  59     admin. married secondary      no    2343     yes   no unknown   5
## 2  56     admin. married secondary      no      45      no   no unknown   5
## 3  41 technician married secondary      no    1270     yes   no unknown   5
## 4  55   services married secondary      no    2476     yes   no unknown   5
## 5  54     admin. married  tertiary      no     184      no   no unknown   5
## 6  42 management  single  tertiary      no       0     yes  yes unknown   5
##   month duration campaign pdays previous poutcome deposit
## 1   may     1042        1    -1        0  unknown     yes
## 2   may     1467        1    -1        0  unknown     yes
## 3   may     1389        1    -1        0  unknown     yes
## 4   may      579        1    -1        0  unknown     yes
## 5   may      673        2    -1        0  unknown     yes
## 6   may      562        2    -1        0  unknown     yes
```

The structure of the data set

```
str(bank_marketing)
```

```
## 'data.frame':    11162 obs. of  17 variables:
##  $ age      : int  59 56 41 55 54 42 56 60 37 28 ...
##  $ job      : chr  "admin." "admin." "technician" "services" ...
##  $ marital  : chr  "married" "married" "married" "married" ...
##  $ education: chr  "secondary" "secondary" "secondary" "secondary" ...
##  $ default  : chr  "no" "no" "no" "no" ...
##  $ balance  : int  2343 45 1270 2476 184 0 830 545 1 5090 ...
##  $ housing  : chr  "yes" "no" "yes" "yes" ...
##  $ loan     : chr  "no" "no" "no" "no" ...
##  $ contact  : chr  "unknown" "unknown" "unknown" "unknown" ...
##  $ day      : int  5 5 5 5 5 6 6 6 6 6 ...
##  $ month    : chr  "may" "may" "may" "may" ...
##  $ duration : int  1042 1467 1389 579 673 562 1201 1030 608 1297 ...
##  $ campaign : int  1 1 1 1 2 2 1 1 1 3 ...
```

```
##  $ pdays    : int  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
##  $ previous : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ poutcome : chr  "unknown" "unknown" "unknown" "unknown" ...
##  $ deposit  : chr  "yes" "yes" "yes" "yes" ...
```

Summary statistics of the data set

```
summary(bank_marketing)
```

```
##       age            job              marital           education
##  Min.   :18.0   Length:11162       Length:11162       Length:11162
##  1st Qu.:32.0   Class :character   Class :character   Class :character
##  Median :39.0   Mode  :character   Mode  :character   Mode  :character
##  Mean   :41.2
##  3rd Qu.:49.0
##  Max.   :95.0
##    default             balance        housing              loan
##  Length:11162       Min.   :-6847   Length:11162       Length:11162
##  Class :character   1st Qu.:  122   Class :character   Class :character
##  Mode  :character   Median :  550   Mode  :character   Mode  :character
##                     Mean   : 1529
##                     3rd Qu.: 1708
##                     Max.   :81204
##    contact               day          month              duration
##  Length:11162       Min.   : 1.0   Length:11162       Min.   :   2
##  Class :character   1st Qu.: 8.0   Class :character   1st Qu.: 138
##  Mode  :character   Median :15.0   Mode  :character   Median : 255
##                     Mean   :15.7                      Mean   : 372
##                     3rd Qu.:22.0                      3rd Qu.: 496
##                     Max.   :31.0                      Max.   :3881
##     campaign          pdays          previous       poutcome
##  Min.   : 1.00   Min.   : -1.0   Min.   : 0.00   Length:11162
##  1st Qu.: 1.00   1st Qu.: -1.0   1st Qu.: 0.00   Class :character
##  Median : 2.00   Median : -1.0   Median : 0.00   Mode  :character
##  Mean   : 2.51   Mean   : 51.3   Mean   : 0.83
##  3rd Qu.: 3.00   3rd Qu.: 20.8   3rd Qu.: 1.00
##  Max.   :63.00   Max.   :854.0   Max.   :58.00
##    deposit
##  Length:11162
##  Class :character
##  Mode  :character
##
##
##
```

4.2 Description of Data set

There are 11162 observations (clients contacted in the marketing campaign) and 17 variables(attributes) in the bank_marketing data set. The data is in tidy format as each row as one observation and column names are features. The variables /attributes are described below: -

Client related information

-age: numeric variable describes the age of the client

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

-marital: marital status of client (categorical: 'divorced', 'married', 'single', 'unknown')

-education: highest educational level of client (categorical: 'secondary', 'tertiary', 'primary', 'unknown')

-default: Whether the client has credit in default (categorical: 'no', 'yes')

-balance: Average yearly balance of the client (integer variable)

-housing: Whether client has a housing loan (categorical: 'no', 'yes')

-loan: Whether the client has a personal loan (categorical: 'no', 'yes')

Information related to current campaign

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

-day: Last contact day of the month

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

-duration: Last contact duration in seconds (numeric)

-campaign: No of contacts performed during this campaign and for this client (numeric, includes last contact)

-pdays: No of days passed by after the client was last contacted in the previous campaign (numeric, -1 indicates that client was not previously contacted)

-previous: No of contacts performed before this campaign and for this client

-poutcome: Outcome of the previous marketing campaign (categorical: 'unknown', 'other', 'faliure', 'success')

Output dependent variable

-deposit: has the client subscribed to the term deposit (binary: 'yes', 'no')

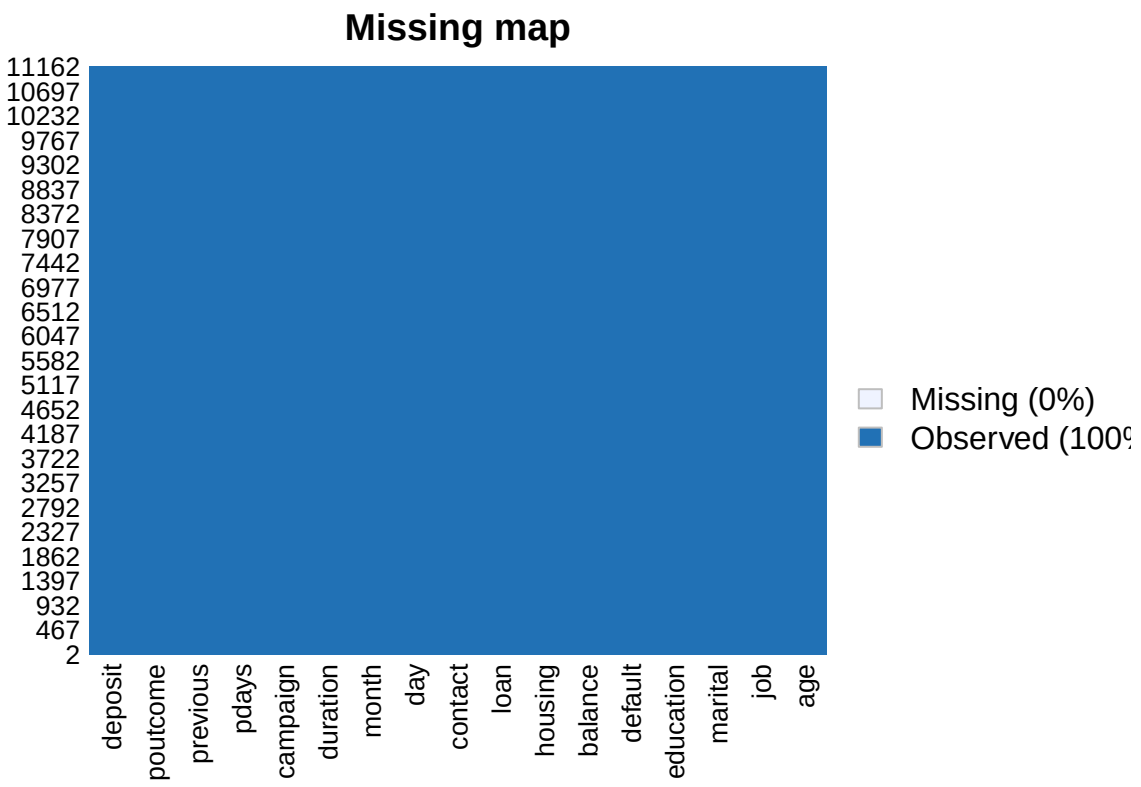
4.3 Checking for missing and NA values

```
sapply(bank_marketing,function(df){
  sum(is.na(df)==TRUE)
})
```

```
##       age       job   marital education   default   balance   housing      loan
##         0         0         0         0         0         0         0         0
##   contact       day     month  duration  campaign     pdays  previous  poutcome
##         0         0         0         0         0         0         0         0
##   deposit
##         0
```

The below table confirms that there are no NA values in our data set. We can also check for missing and NA values by analyzing the missing map

```
missmap(bank_marketing,main = "Missing map")
```



4.4 Attributes selection

```
bank_marketing%>%group_by(pdays)%>%
  summarize(n=n())%>%
  arrange(desc(n))%>%head()%>%
  knitr::kable()
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

pdays	n
-1	8324
92	106
182	89
91	84
181	81
183	73

```
table(bank_marketing$poutcome)
```

```
##
## failure   other success unknown
##    1228     537    1071    8326
```

```
table(bank_marketing$previous)%>%head()
```

```
##
##    0    1    2    3    4    5
## 8324  887  693  435  244  165
```

74% of the clients(8324) who are contacted in the current campaign were not previously contacted so most of the clients in current campaign are contacted for the first time. For 75% of the clients, the outcome of previous marketing campaign is unknown. For 74% of the clients(8324), no contacts were performed before this campaign. Large proportion of the clients in the data set are those that are contacted for the first time. Hence attributes related to previous marketing campaign – 'pdays', 'poutcome', 'previous' were not considered for prediction model.

Similarly, the contact variable which describes the mode of communication i.e cellular or telephone is not an important factor for predicting whether the client would subscribe or not. It can also be seen that 72% of the clients were contacted through cellular mode.

```
bank_marketing<-bank_marketing%>%select(-c("contact","pdays","poutcome","previous"))
```

4.5 Feature Engineering

1. Converting the 'month' variable to Quarter The 'month' variable contains information for twelve months. For more effective analysis, the information would be grouped in terms of quarter i.e. four categorical variables- 'Qtr1_jan-mar', 'Qtr2_apr-jun', 'Qtr3_july-aug', 'Qtr4_oct-dec'.

```
table(bank_marketing$month)
```

```
##
##  apr  aug  dec  feb  jan  jul  jun  mar  may  nov  oct  sep
##  923 1519  110  776  344 1514 1222  276 2824  943  392  319
```

```
bank_marketing$month<-as.factor(bank_marketing$month)

levels(bank_marketing$month)[c(4,5,8)] <- c("Qtr1_jan-march")
levels(bank_marketing$month)[c(1,7,6)] <- c("Qtr2_apr-june")
levels(bank_marketing$month)[c(2,5,8)] <- c("Qtr3_july-sep")
levels(bank_marketing$month)[c(3,5,6)] <- c("Qtr4_oct-dec")

colnames(bank_marketing)[c(10)]<-c("Qtr")
```

2. Imputing the 'unknown' values in 'job' variable

There are only 70 clients whose job category is 'unknown'. These are just 0.6% of the values in these variables. We will take off these observations and our data set would still have 11092 observations.

```
table(bank_marketing$job)
```

```
##
##        admin.   blue-collar  entrepreneur     housemaid    management
##          1334          1944           328           274          2566
```

```
##       retired self-employed      services       student    technician
##           778            405           923           360          1823
##    unemployed       unknown
##           357             70
```

```
bank_marketing<-bank_marketing%>%filter(job!="unknown")
```

3. Imputing the 'unknown' values in education variable

There are 458 values in the education column that are 'unknown'. These are 4% of the values in the column. As most of values (50%) in the education column belong to 'secondary' category so we have assigned the 'unknown' values to the 'secondary' category.

```
table(bank_marketing$education)
```

```
##
##   primary secondary  tertiary   unknown
##      1493      5461      3680       458
```

```
bank_marketing$education<-factor(bank_marketing$education)
levels(bank_marketing$education)[c(4)]<-c("secondary")
levels(bank_marketing$education)
```

```
## [1] "primary"   "secondary" "tertiary"
```

```
table(bank_marketing$education)
```

```
##
##   primary secondary  tertiary
##      1493      5919      3680
```

4. Changing the categorical variables to factor class The 'job', 'marital', 'default', 'housing', 'loan', 'deposit' variables are converted to factor class.

```
bank_marketing$job<-factor(bank_marketing$job)
bank_marketing$marital<-factor(bank_marketing$marital)
bank_marketing$default<-factor(bank_marketing$default)
bank_marketing$housing<-factor(bank_marketing$housing)
bank_marketing$loan<-factor(bank_marketing$loan)
bank_marketing$deposit<-factor(bank_marketing$deposit)
```

5. Renaming the levels of categorical variables

We have renamed the levels of the three categorical variables for better depiction of the information contained in the variable -housing: categorical ('no_housing_loan', 'housing_loan) -loan: categorical ('no_personal_loan', 'personal_loan) -deposit: categorical ('not_subscribed', 'subscribed')

```r
levels(bank_marketing$housing)
```

```
## [1] "no"  "yes"
```

```r
table(bank_marketing$housing)
```

```
##
##   no  yes
## 5814 5278
```

```r
levels(bank_marketing$housing)[c(1,2)]<-c("no_housing_loan","housing_loan")
levels(bank_marketing$housing)
```

```
## [1] "no_housing_loan" "housing_loan"
```

```r
table(bank_marketing$housing)
```

```
##
## no_housing_loan    housing_loan
##            5814            5278
```

```r
levels(bank_marketing$loan)
```

```
## [1] "no"  "yes"
```

```r
table(bank_marketing$loan)
```

```
##
##   no  yes
## 9634 1458
```

```r
levels(bank_marketing$loan)[c(1,2)]<-c("no_personal_loan","personal_loan")

levels(bank_marketing$loan)
```

```
## [1] "no_personal_loan" "personal_loan"
```

```r
table(bank_marketing$loan)
```

```
##
## no_personal_loan    personal_loan
##             9634             1458
```

```r
levels(bank_marketing$deposit)
```

```
## [1] "no"  "yes"
```

```r
table(bank_marketing$deposit)
```

```
##
##   no  yes
## 5837 5255
```

```r
levels(bank_marketing$deposit)[c(1,2)]<-c("not_subscribed","subscribed")
levels(bank_marketing$deposit)
```

```
## [1] "not_subscribed" "subscribed"
```

```r
table(bank_marketing$deposit)
```

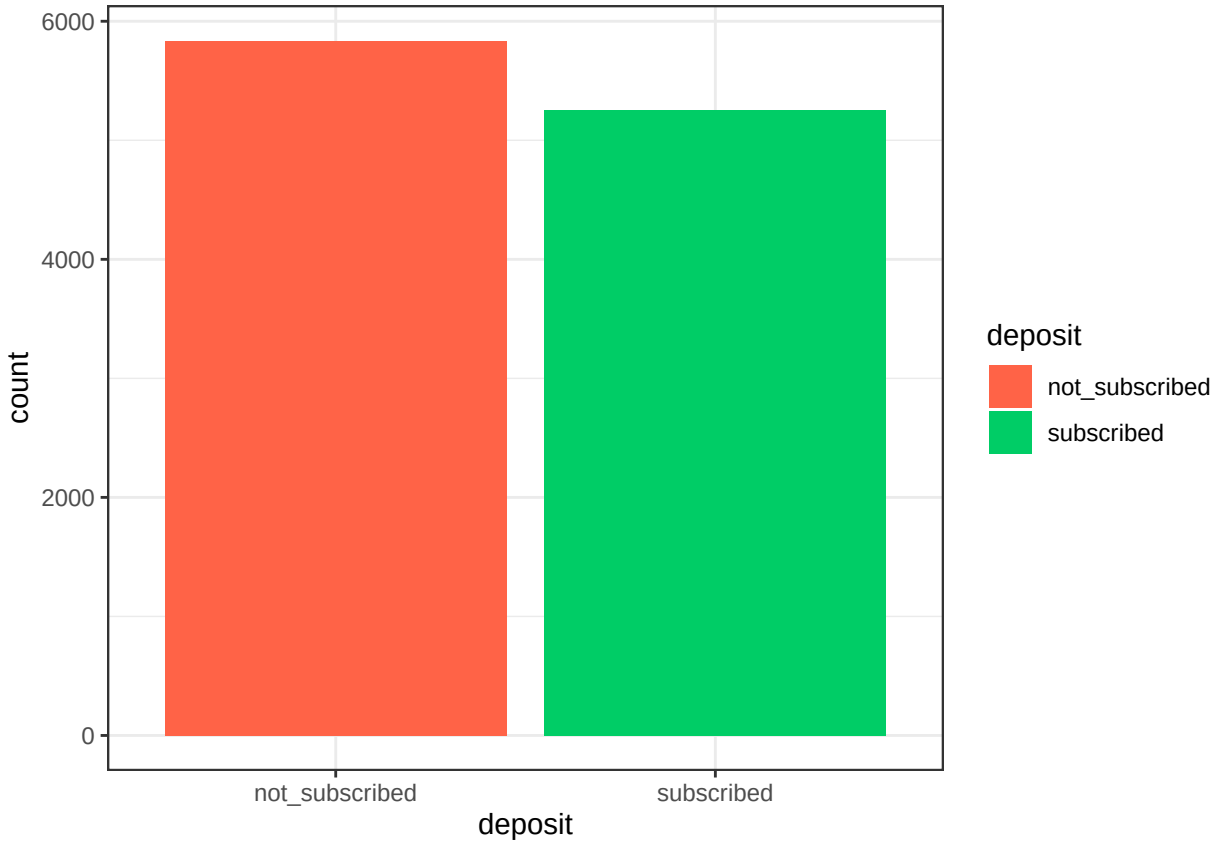
```
##
## not_subscribed     subscribed
##           5837           5255
```

5.Exploratory Data Analysis

5.1 Target variable (deposit)

Below is the bar graph plot for the distribution of the target output variable- 'deposit'. The bar graph depicts the number of clients in the data set who subscribed to the term deposit and those that didn't.

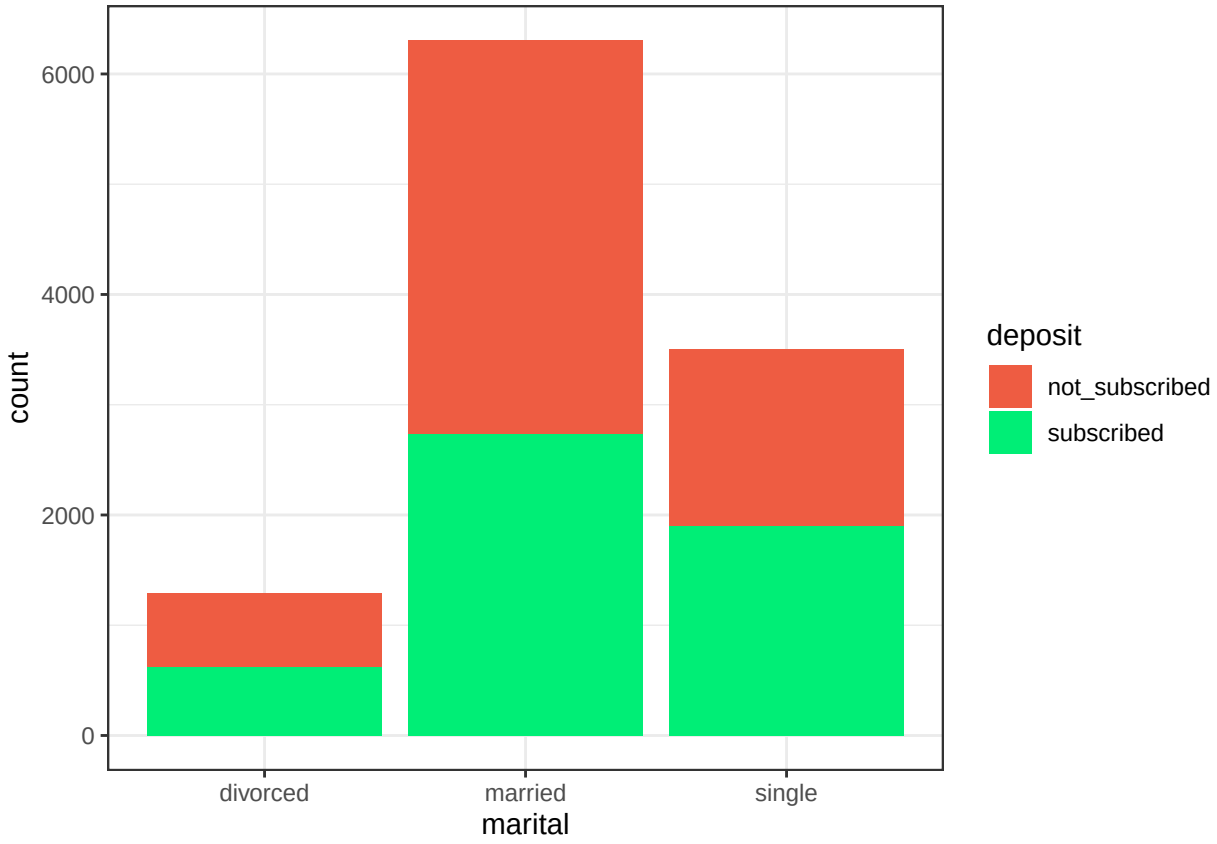
```
ggplot(bank_marketing)+geom_bar(aes(deposit,fill=deposit))+
  scale_fill_manual(values = c("tomato","springgreen3"))+theme_bw()
```



5.2 Marital Status

Below is the bar graph for the distribution of marital status of the clients in the data set and also the distribution of marital status for the deposit variable. About 57% of the clients (6302) that were contacted were married followed by 31 % that were single (3499). The no of clients who had taken the term deposit are also more in the married category.

```
bank_marketing%>%ggplot(aes(marital))+
  geom_bar(aes(fill=deposit))+
  scale_fill_manual(values = c("tomato2","springgreen2"))+
  theme_bw()
```




```
prop.table(table(bank_marketing$marital))
```

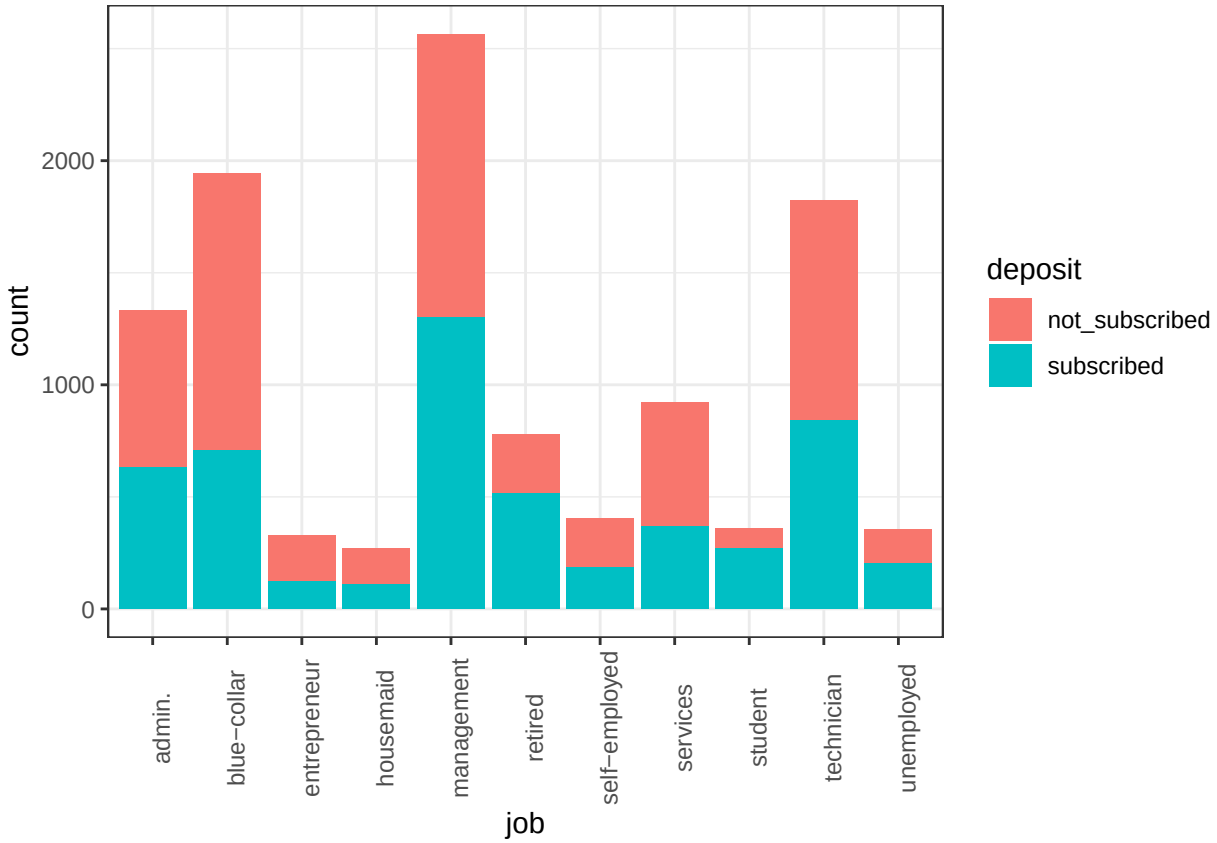
```
##
## divorced  married   single
##   0.1164   0.5682   0.3155
```

5.3 Job

Below is the bar graph for the distribution of job status of the clients in the data set and also the count of clients that subscribed to the deposit within different job categories. Most number of clients contacted in the marketing campaign belonged to the Management job, admin, blue-collar or technician jobs. Very few entrepreneurs, retired persons or self-employed were contacted. For four job categories- management, retired, student, unemployed the no of persons who subscribed to the deposit were more than those that didn't subscribe. In all the other categories the no of persons who didn't subscribe were more.

Job vs Deposit

```
bank_marketing%>%ggplot(aes(job))+
  geom_bar(aes(fill=deposit))+
  scale_fill_discrete()+
  theme_bw()+
  theme(axis.text.x = element_text(angle = 90))
```

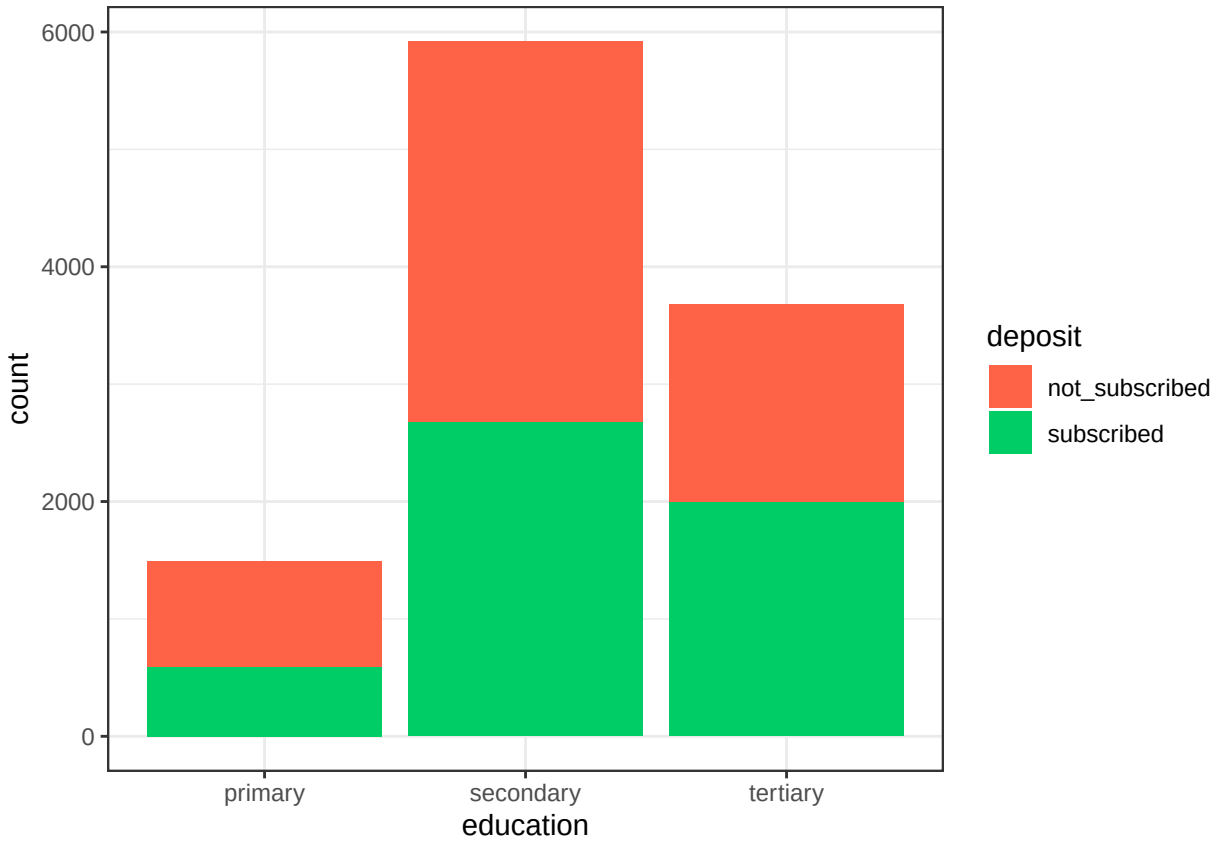


5.4 Education

Below is the bar graph for the distribution of highest education level of the clients in the data set. 53% of the clients in the data set had the highest education as 'secondary'. Also, the maximum no of clients who opted for the deposit had the highest education level as 'secondary'. Only 13% of the clients had the highest education as 'primary'.

Education vs Deposit

```
bank_marketing%>%ggplot(aes(education))+
  geom_bar(aes(fill=deposit))+
  scale_fill_manual(values = c("tomato1","springgreen3"))+
  theme_bw()
```



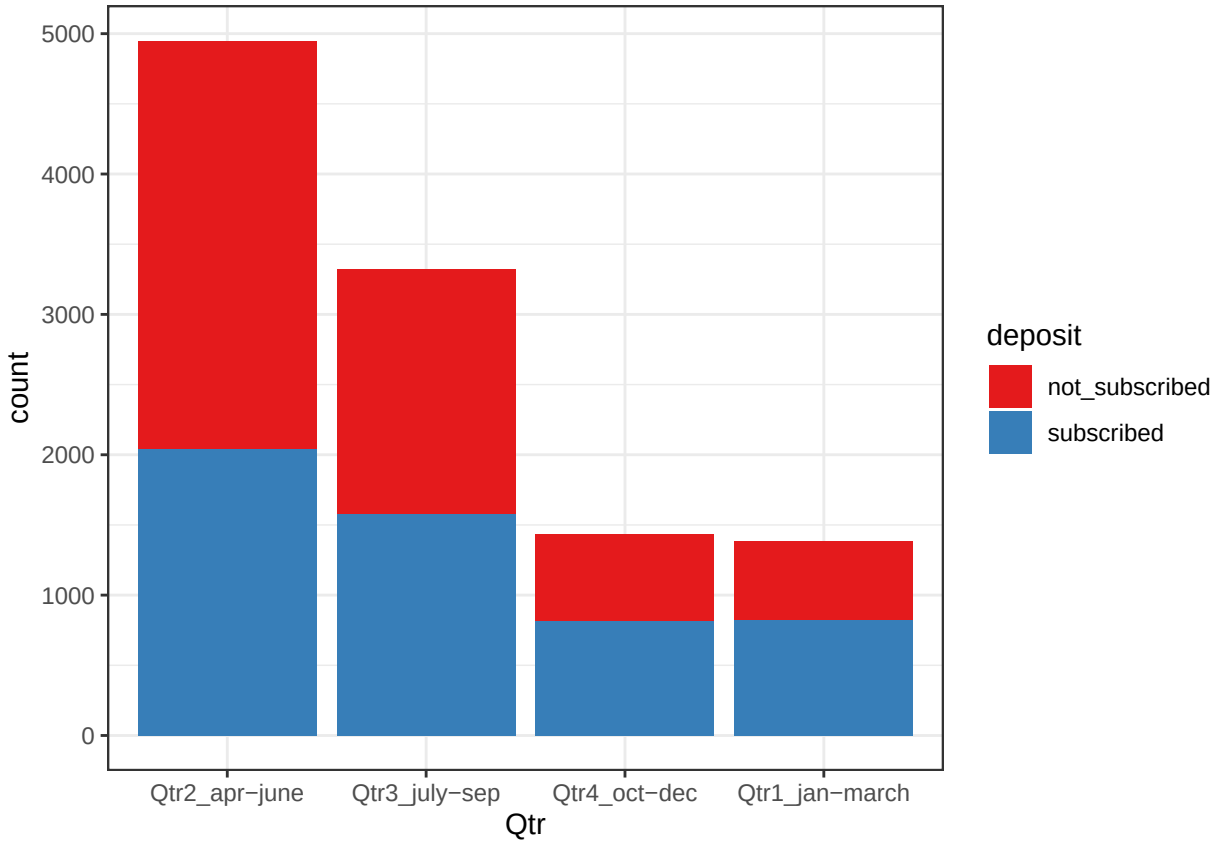

```
prop.table(table(bank_marketing$education))
```

```
##
##   primary secondary  tertiary
##    0.1346    0.5336    0.3318
```

5.5 Qtr

Below is the bar graph for the distribution of Qtr of the year in which the clients were contacted in the year. 44% of the clients were contacted in the marketing campaigns in the months of Apr-June. Also most of the clients subscribed in this qtr(Apr-June). This could be due to the fact that at the start of the Financial year more working-class people are in need for a loan or deposit schemes.

```
bank_marketing%>%ggplot(aes(Qtr))+
  geom_bar(aes(fill=deposit))+
  scale_fill_brewer(palette = "Set1")+
  theme_bw()
```




```
prop.table(table(bank_marketing$Qtr))
```

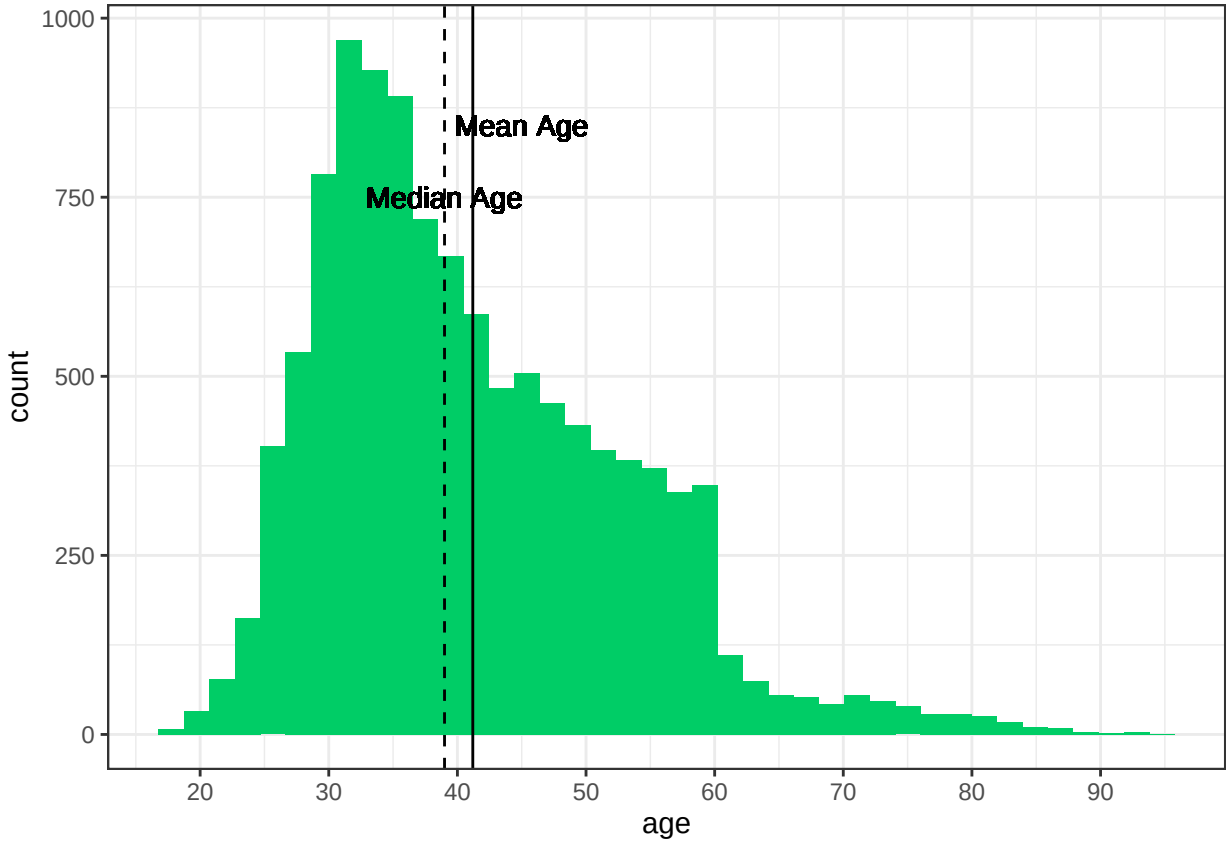
```
##
##  Qtr2_apr-june  Qtr3_july-sep   Qtr4_oct-dec Qtr1_jan-march
##         0.4463         0.2995         0.1294         0.1249
```

5.6 Age

Below is the histogram for the distribution of age of the clients in the data set. The average age of clients in the data set is 42 years and the median age of the clients is 39 years. Most of the clients contacted were in 30-50-year age group.Very few persons, only 5% clients over the age of 60 years were contacted during the campaign

Age distribution of clients

```
ggplot(bank_marketing,aes(age))+
  geom_histogram(bins = 40,alpha=1,fill="springgreen3")+
  theme_bw()+geom_vline(aes(xintercept = mean(age)))+
  geom_text(x=45,y=850,label="Mean Age")+
  geom_vline(aes(xintercept=median(age)),linetype=2)+
  geom_text(x=39,y=750,label="Median Age")+
  scale_x_continuous(breaks=seq(min(0),max(100),by=10))
```




```
mean(bank_marketing$age)
```

```
## [1] 41.2
```

```
median(bank_marketing$age)
```

```
## [1] 39
```

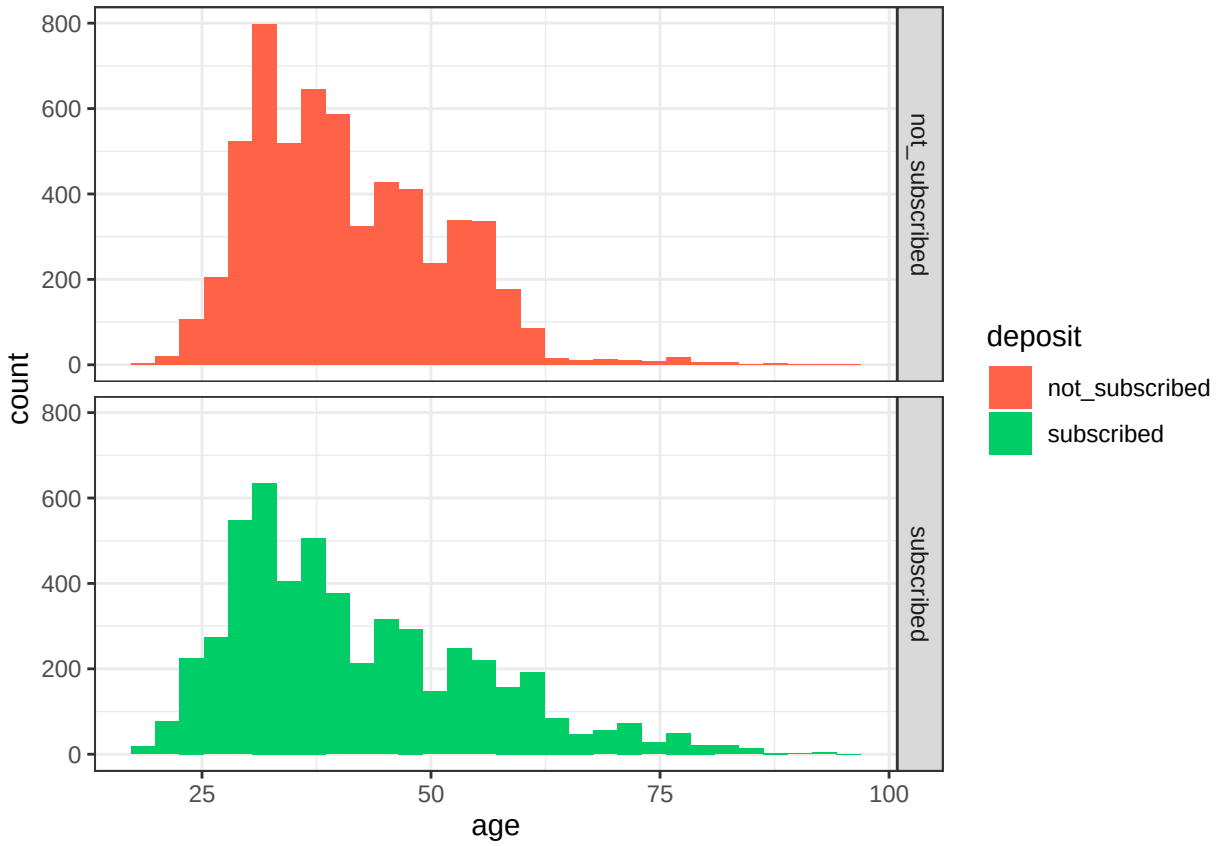
```
sum(bank_marketing$age>60)/length(bank_marketing$age)
```

```
## [1] 0.05463
```

Age vs Deposit

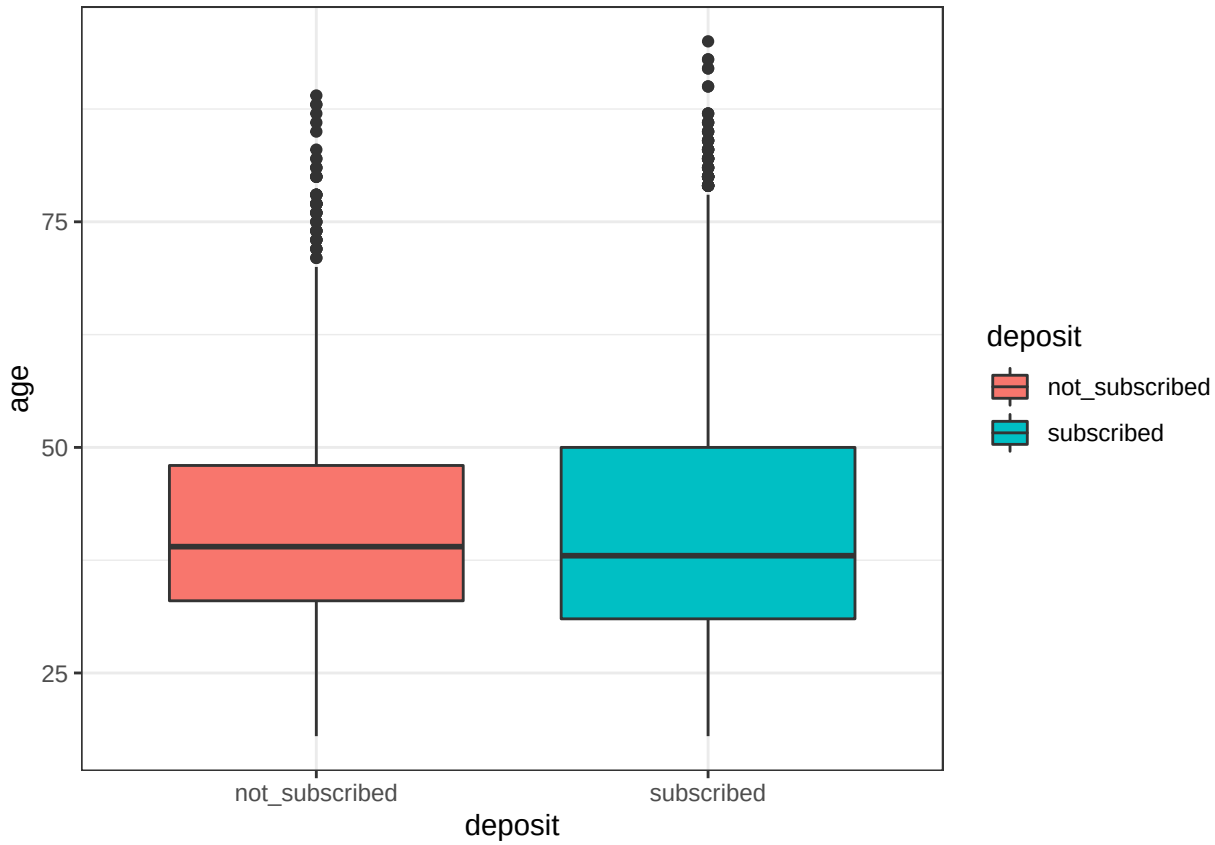
The age distribution is similar for clients who subscribed and those who didn't, with most persons in the age group of 30-50 years

```
ggplot(bank_marketing,aes(age,fill=deposit))+
  scale_fill_manual(values = c("tomato1","springgreen3"))+
  geom_histogram(bins=30)+
  facet_grid(deposit~.)+theme_bw()
```



From the boxplot below,it can be interpreted that the range of age group of clients who subscribed to the deposit is slightly more than the range of age groups for clients who didn't subscribe.

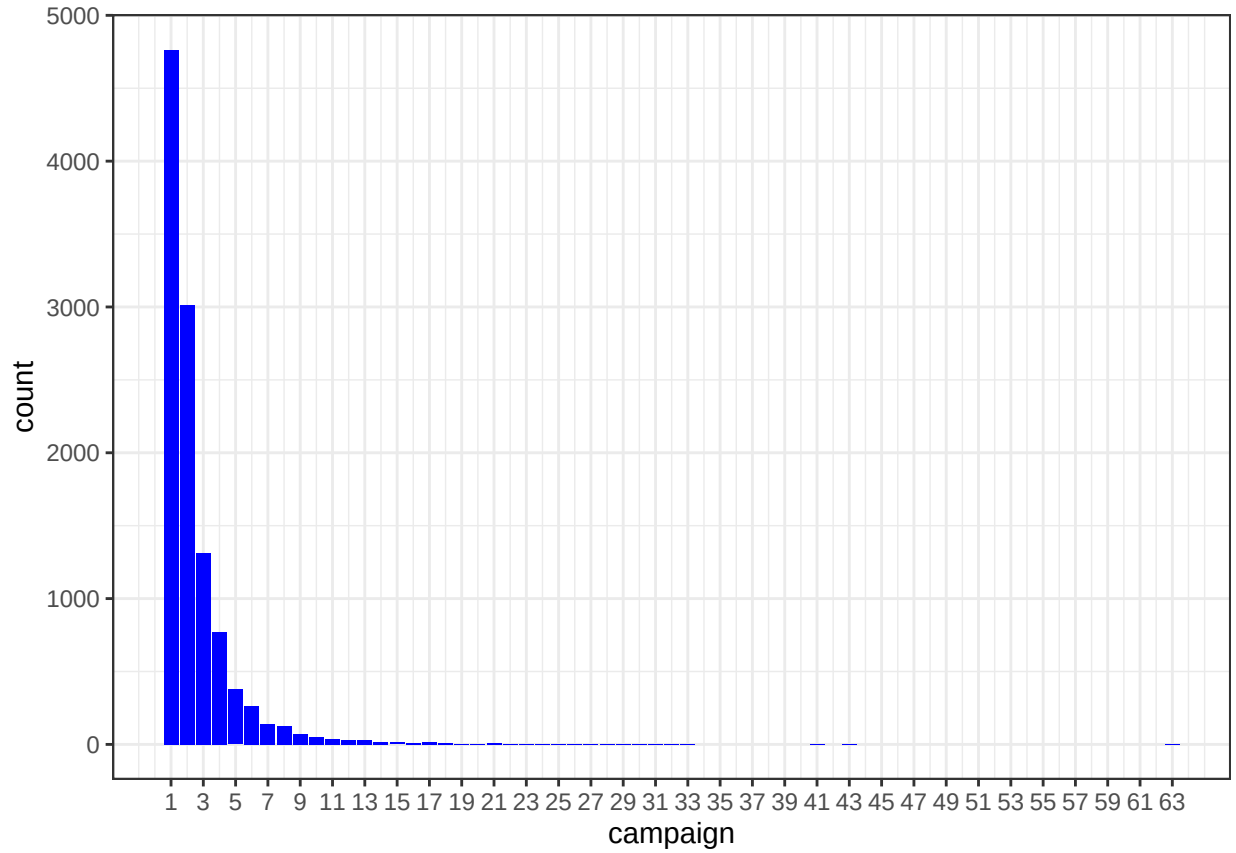
```
ggplot(bank_marketing,aes(deposit,age))+
  geom_boxplot(aes(fill=deposit))+theme_bw()
```


75
50
25
age
not_subscribed
subscribed
deposit
deposit
not_subscribed
subscribed

5.7 Campaign

Below is the bar graph for the count of no of contacts made to the client in the current marketing campaign. 70% of the clients were contacted once or twice. Most clients made their decisions in the first few calls.

```
ggplot(bank_marketing)+
  geom_bar(aes(campaign,fill=campaign),fill="blue")+
  scale_x_continuous(breaks=seq(min(1),max(63),by=2))+
  scale_fill_brewer(palette = "Set1")+theme_bw()
```



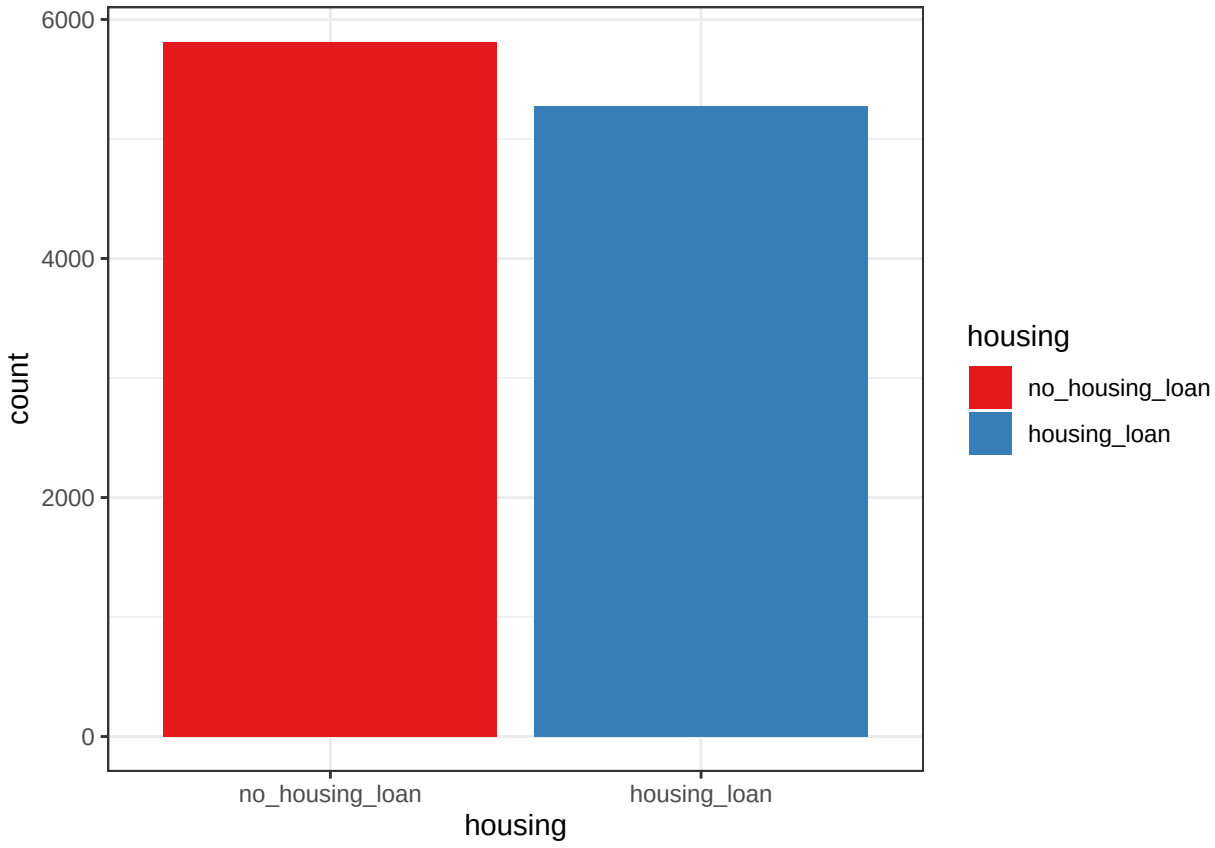

```
prop.table(table(bank_marketing$campaign))%>%head()
```

```
##
##       1       2       3       4       5       6
## 0.42959 0.27164 0.11828 0.06933 0.03390 0.02371
```

5.8 Housing Loan

Below is the bar graph distribution for the count of housing loan variable for deposit variable. We can see the more persons tend to opt for the term deposit if they haven't already taken a housing loan. 52% of clients in the data set didnt have a housing loan.

```
ggplot(bank_marketing)+geom_bar(aes(housing,fill=housing))+
  scale_fill_brewer(palette = "Set1")+theme_bw()
```



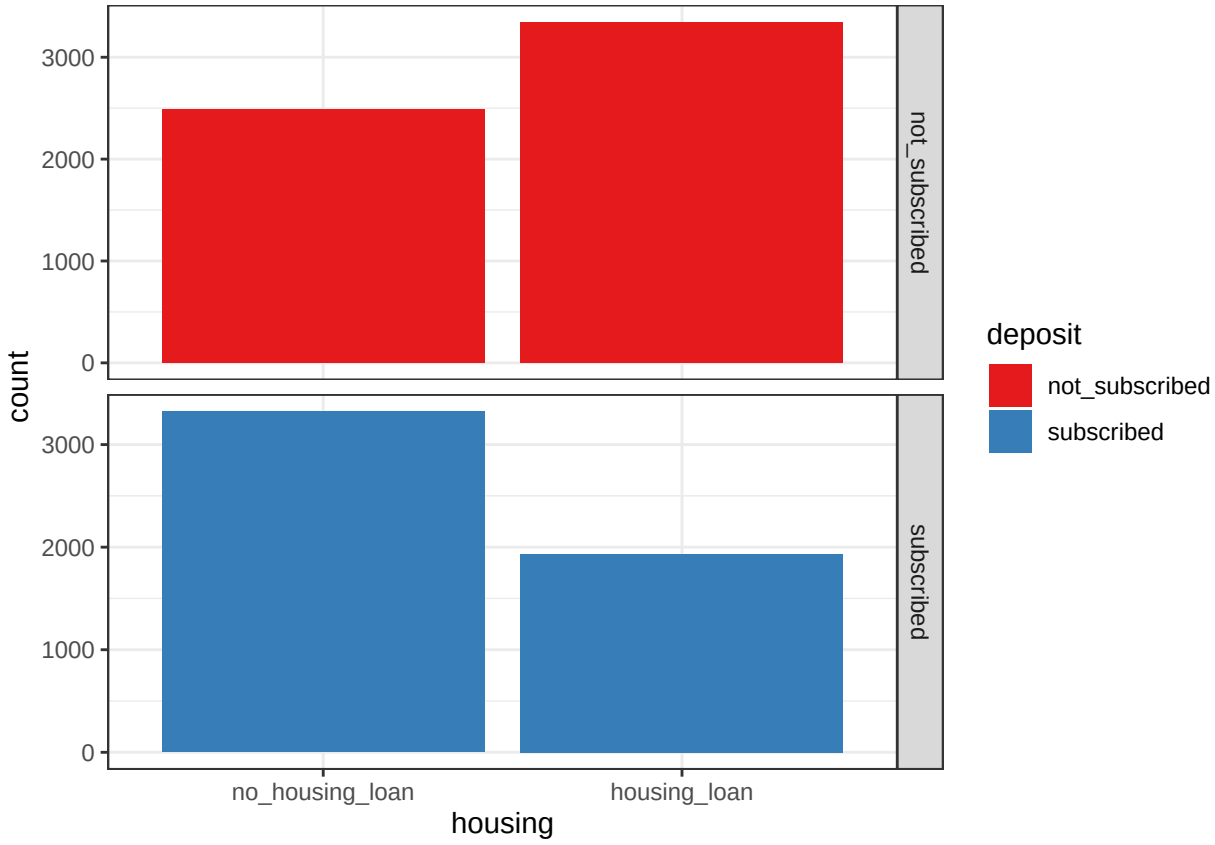

```
prop.table(table(bank_marketing$housing))
```

```
##
## no_housing_loan    housing_loan
##          0.5242          0.4758
```

Housing loan v Deposit

From the below bar graph where the distribution of housing loan is shown seperately for clients who took the term deposit and those that didnt, it can be inferred that When there is already a housing loan, less no of clients would opt for term deposit offered by the bank

```
bank_marketing%>%ggplot(aes(housing,fill=deposit))+
  geom_bar()+facet_grid(deposit~.,scales = "free")+
  scale_fill_brewer(palette = "Set1")+
  theme_bw()
```

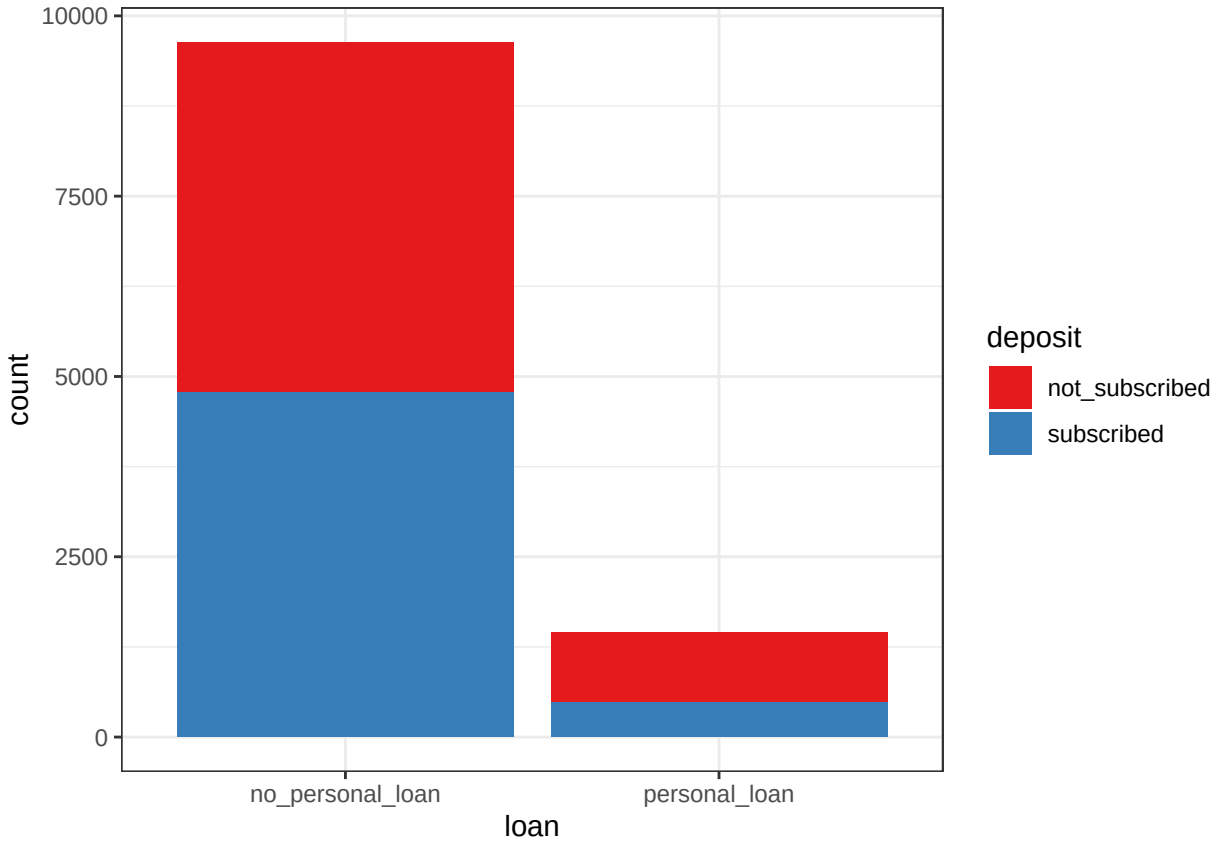


5.9 Personal Loan

Below is the bar graph distribution of personal loan for the deposit variable. 87% of the clients contacted by the bank had not taken a personal loan. We can also see that just like housing loan, if the client already has taken a personal loan, the likelihood of subscribing to term deposit would be less. But on the other hand, it doesn't necessarily mean that if the client doesn't have a personal loan, he would subscribe for term deposit. As we can see that there are almost the same proportion of clients in the no_personal_loan category that opt for term deposit and those that don't opt for term deposit.

Personal Loan v Deposit

```
bank_marketing%>%ggplot(aes(loan,fill=deposit))+
  geom_bar()+
  scale_fill_brewer(palette = "Set1")+
  theme_bw()
```



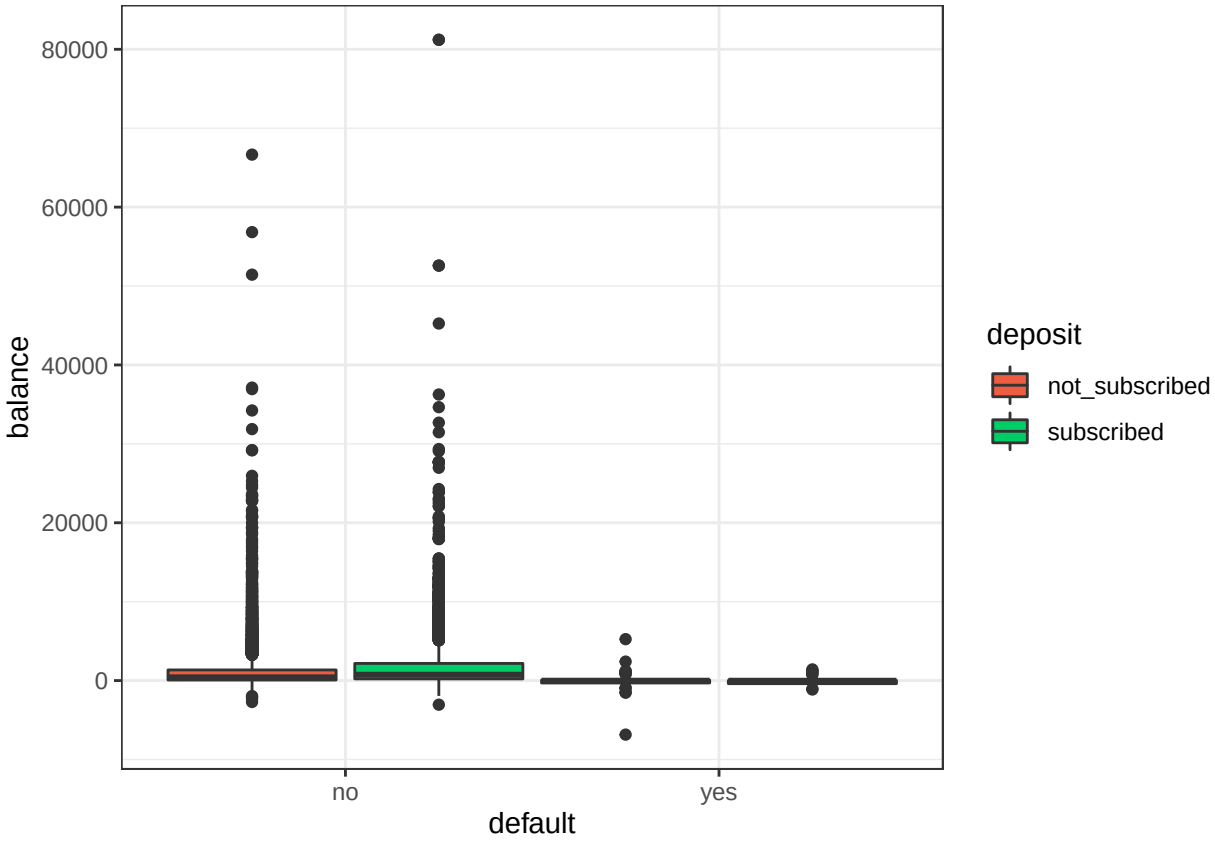

```
prop.table(table(bank_marketing$loan))
```

```
##
## no_personal_loan    personal_loan
##           0.8686           0.1314
```

5.10 Visualization of Balance, Default status and Deposit

If we analyze the box plot distribution for yearly balance for defaulters and non-defaulters, we can see that persons who have no credit in default have a slightly higher yearly balance and are more likely to subscribe to term deposit.

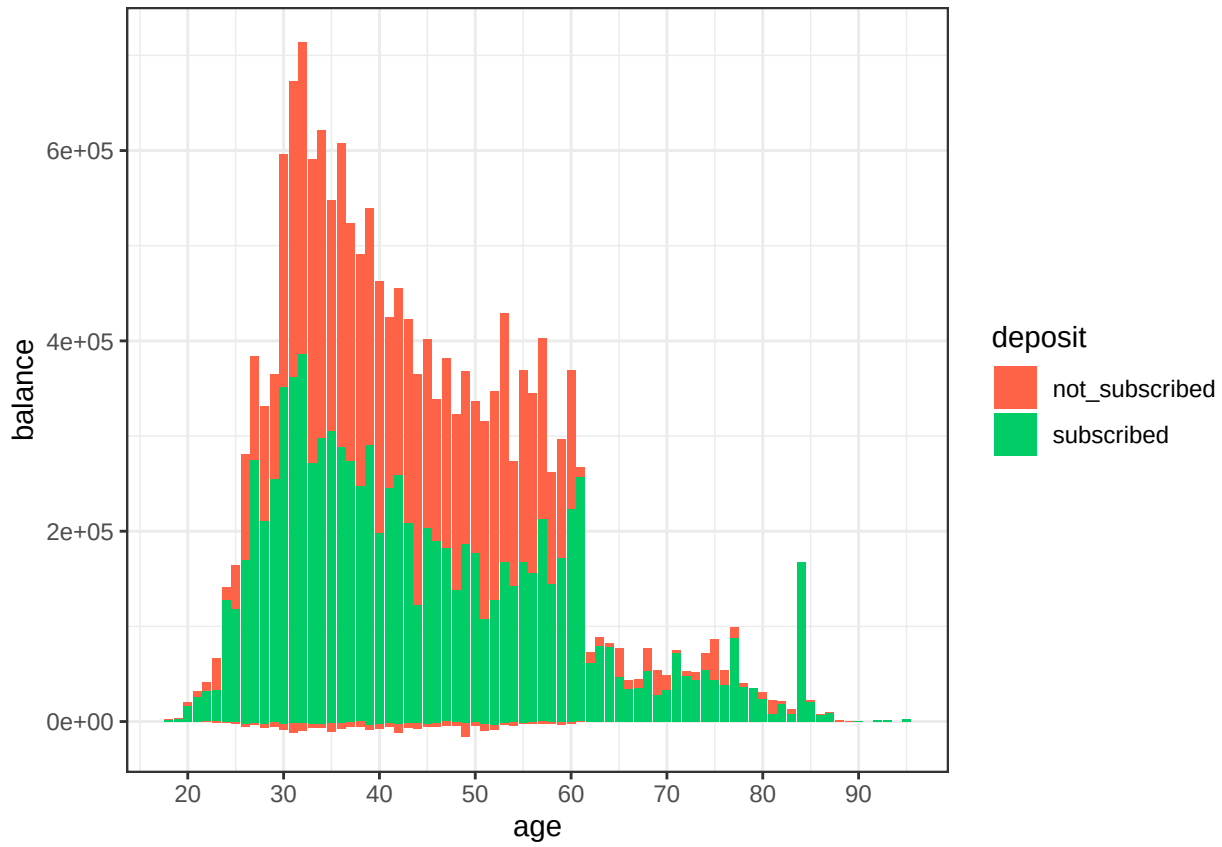
```
ggplot(bank_marketing,aes(default,balance,fill=deposit))+
  geom_boxplot(width=1)+
  scale_fill_manual(values = c("tomato2","springgreen3"))+
  theme_bw()
```



5.11 Visualization of balance and age

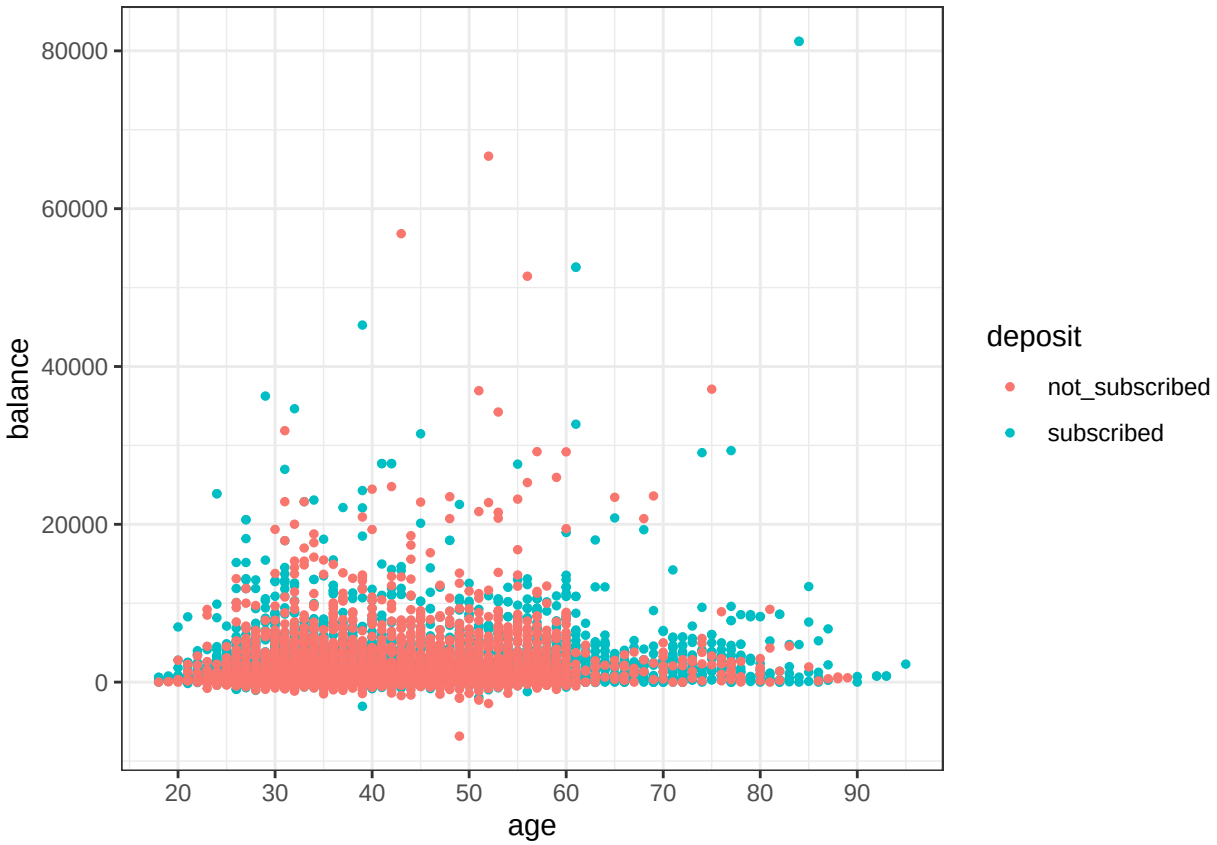
Below is the scatter and bar plot for distribution of the numeric variables- age and balance. Although there is no definite trend, but still it can be seen that the yearly balance is more in the age group 30-55 years. In the same age groups, we find that for clients with more balance, the likelihood to subscribe to deposit is high.

```
ggplot(bank_marketing,aes(age,balance))+
  geom_bar(aes(fill=deposit),binwidth=30,stat = "identity")+
  scale_x_continuous(breaks=seq(min(0),max(100),by=10))+
  scale_fill_manual(values = c("tomato","springgreen3"))+
  theme_bw()
```



The blue coloured points in the below scatter plot represent the clients who subscribed to deposit and it can be seen that subscription for term deposit is more for higher balance. As the clients engage in jobs, their yearly balance also increases and as the clients pass their working age, their balance decreases.

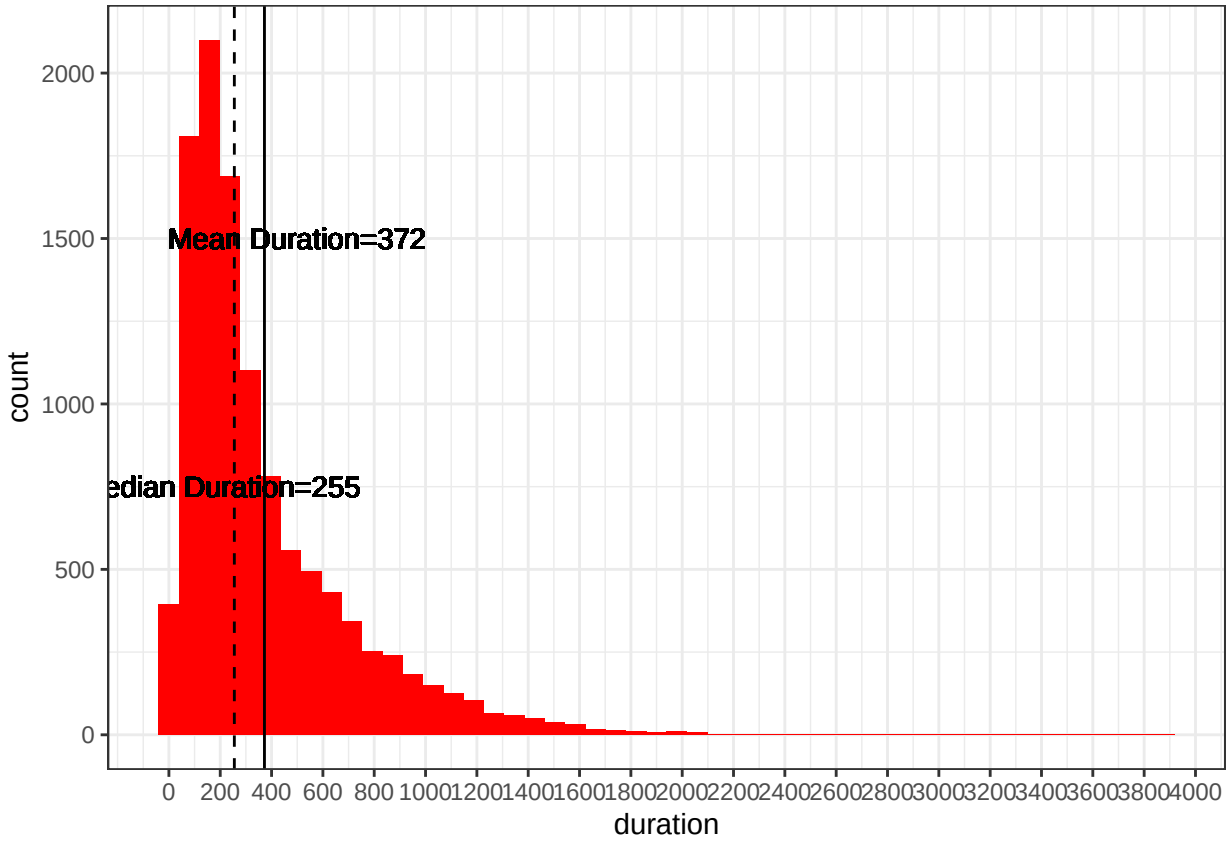
```
ggplot(bank_marketing,aes(age,balance))+
  geom_point(aes(color=deposit),alpha=1,size=1)+
  scale_x_continuous(breaks=seq(min(0),max(100),by=10))+
  theme_bw()
```


80000
60000
40000
20000
0
balance
20
30
40
50
60
70
80
90
age
deposit
not_subscribed
subscribed

5.12 Duration

Below is the histogram distribution of the duration of call(seconds) made to the clients. Most the calls made in the marketing campaign are of duration 100 seconds to 500 seconds. The average duration of the calls is 372 seconds i.e around 6 minutes and median duration of the calls is 255 seconds.

```
ggplot(bank_marketing,aes(duration))+
  geom_histogram(bins = 50,alpha=1,fill="red")+
  scale_x_continuous(breaks=seq(min(0),max(4000),by=200))+
  theme_bw()+
  geom_vline(aes(xintercept = mean(duration)))+
  geom_text(x=500,y=1500,label="Mean Duration=372")+
  geom_vline(aes(xintercept=median(duration)),linetype=2)+
  geom_text(x=200,y=750,label="Median Duration=255")
```




```
mean(bank_marketing$duration)
```

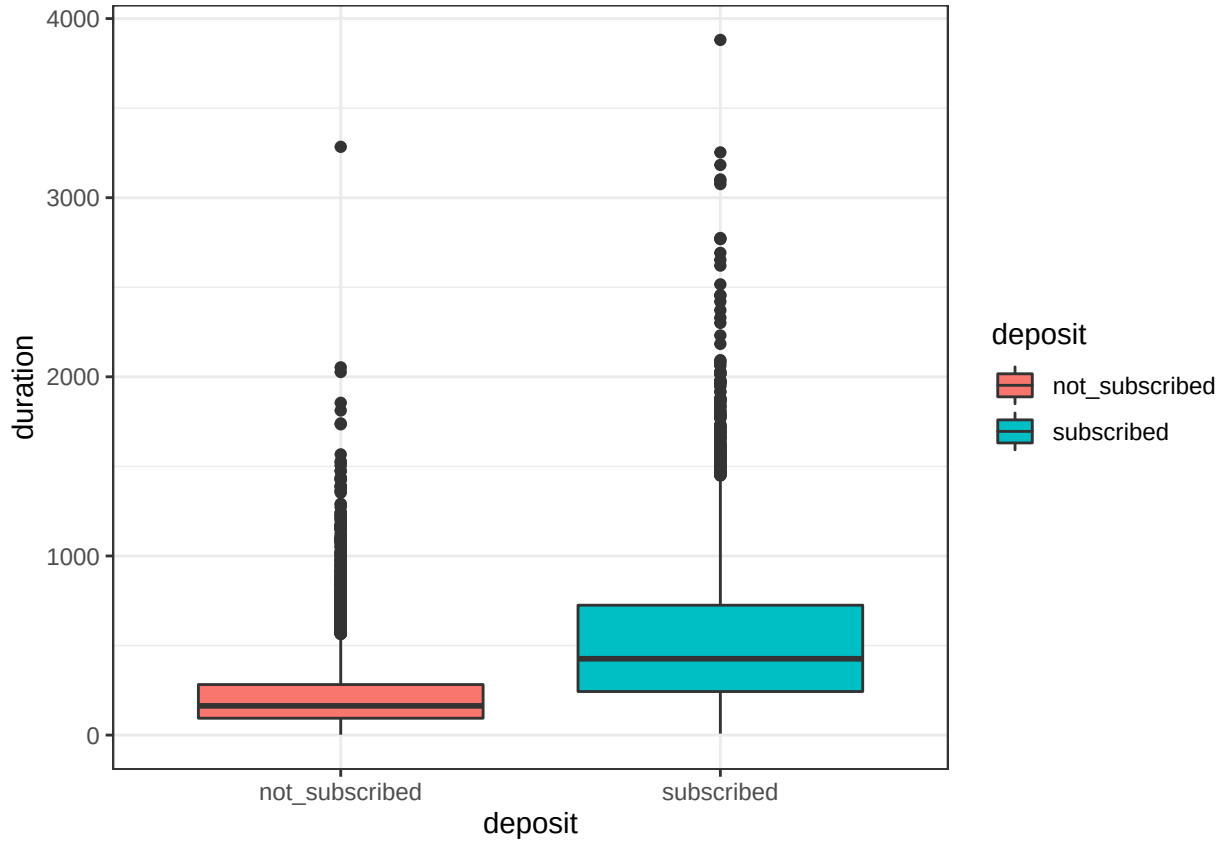
```
## [1] 372.3
```

```
median(bank_marketing$duration)
```

```
## [1] 255
```

Below is the boxplot for the distribution of call duration for the clients who subscribed to the deposit and those that didnt. As expected, from the boxplot, it can be seen duration for the calls for clients subscribed to term deposit is longer. The median duration is also more for the clients who subscribed to deposit.

```
ggplot(bank_marketing,aes(deposit,duration))+
  geom_boxplot(aes(fill=deposit))+theme_bw()
```



Removing the duration variable

Duration, numeric variable which states the duration of the call made to the client was not considered for the model. The duration is not known before the call is made and at the end of the call, it is obviously known, if the client would subscribe or not. The 'duration' variable may highly affect the results as when the duration is 0, the output is 'no'.

So to build a realistic predictive model, duration variable was not included.

Also, our objective is to build an effective prediction model based on client related attributes.

Therefore, we have considered features related to client and attributes related to current marketing campaign only.

```
bank_marketing<-bank_marketing%>%select(-c("duration"))
```

Final Data set for modeling analysis

The data set after preparation and feature engineering that would be used for model building and prediction is as below. It has 11092 observations and 12 variables.

```
str(bank_marketing)
```

```
## 'data.frame':    11092 obs. of  12 variables:
##  $ age      : int  59 56 41 55 54 42 56 60 37 28 ...
##  $ job      : Factor w/ 11 levels "admin.","blue-collar",..: 1 1 10 8 1 5 5 6 10 8 ...
##  $ marital  : Factor w/ 3 levels "divorced","married",..: 2 2 2 2 2 3 2 1 2 3 ...
##  $ education: Factor w/ 3 levels "primary","secondary",..: 2 2 2 2 3 3 3 2 2 2 ...
##  $ default  : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ balance  : int  2343 45 1270 2476 184 0 830 545 1 5090 ...
##  $ housing  : Factor w/ 2 levels "no_housing_loan",..: 2 1 2 2 1 2 2 2 2 2 ...
##  $ loan     : Factor w/ 2 levels "no_personal_loan",..: 1 1 1 1 1 2 2 1 1 1 ...
##  $ day      : int  5 5 5 5 5 5 6 6 6 6 ...
##  $ Qtr      : Factor w/ 4 levels "Qtr2_apr-june",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ campaign : int  1 1 1 1 2 2 1 1 1 3 ...
##  $ deposit  : Factor w/ 2 levels "not_subscribed",..: 2 2 2 2 2 2 2 2 2 2 ...
```

```
head(bank_marketing)
```

```
##   age        job marital education default balance         housing
## 1  59     admin. married secondary      no    2343    housing_loan
## 2  56     admin. married secondary      no      45 no_housing_loan
## 3  41 technician married secondary      no    1270    housing_loan
## 4  55   services married secondary      no    2476    housing_loan
## 5  54     admin. married  tertiary      no     184 no_housing_loan
## 6  42 management  single  tertiary      no       0    housing_loan
##               loan day           Qtr campaign    deposit
## 1 no_personal_loan   5 Qtr2_apr-june        1 subscribed
## 2 no_personal_loan   5 Qtr2_apr-june        1 subscribed
## 3 no_personal_loan   5 Qtr2_apr-june        1 subscribed
## 4 no_personal_loan   5 Qtr2_apr-june        1 subscribed
## 5 no_personal_loan   5 Qtr2_apr-june        2 subscribed
## 6    personal_loan   5 Qtr2_apr-june        2 subscribed
```

6. Modeling Approach

The dependent variable (deposit) we are trying to predict is categorical (has the client subscribed to the deposit- binary: 'not_subscribed', 'subscribed'). Since we would be classifying the target variable into two categories using classification-based Machine Learning models, our prediction problem would be Binary classification.

For the purpose of machine learning, the data set was split into train and test set. The train set was used to train the model and test set was used exclusively to evaluate the performance of the model. The classification model will try to draw conclusion from the training set and predict the categories for the test data set. We have used 90/10 split for splitting the bank_marketing set to train and test set. Train set would 90% of the bank_marketing set and test set would be 10% of bank_marketing set. 10% of the data set was carved out as test set so that there is as much data as possible to train.

6.1 Creating train and test sets

```
set.seed(1, sample.kind="Rounding") # if using R 3.5 or earlier, use 'set.seed(1)'
test_index <- createDataPartition(y = bank_marketing$deposit, times = 1, p = 0.1, list = FALSE)
train_set <- bank_marketing[-test_index,]
test_set<- bank_marketing[test_index,]
```

6.2 Matrices for Evaluation of Model

-Accuracy: Accuracy of the model we build would be the overall proportion of the output variable(deposit)that were predicted correctly.

Simply building a model that gives us high accuracy may not be enough. Since we are looking to build an optimum model that would work in all situations, and even for data sets that are imbalanced, we have to consider other matrices to evaluate the model like- Sensitivity and Specificity

-Sensitivity: Sensitivity is defined as proportion of actual positives that are predicted as positives. In this project, it would indicate the proportion of clients who subscribed the term deposit that were predicted correctly.

-Specificity: Specificity is defined as proportion of actual negatives that are called negatives. In this project, it would indicate the proportion of clients who did not subscribe the term deposit that were predicted correctly.

In this project, the performance matrices considered for evaluating the Machine Learning Algorithms used are:-Accuracy and Sensitivity

7. Methods and Analysis

7.1 Logistic Regression

Logistic Regression is a classification algorithm. It estimates discrete binary values based on the independent variables and predicts probability by fitting the data into a logit function. This classification algorithm makes use of logistic transformation that converts probability to log odds.

Since the project is based on a classification problem of predicting whether the client would subscribe or not, the model is build up by using Logistic regression to analyze a set of independent variables in the train set and used to predict the deposit variable in the test set. Train function is used with method ='glm' to build the logistic regression model

Training and evaluation of model

```r
set.seed(1,sample.kind = "Rounding")
glm_model<-train(deposit~.,method="glm",data=train_set)
glm_hat<-predict(glm_model,test_set)
```

summary of the Logistic Regression model

```r
summary(glm_model)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.257  -1.054  -0.676   1.093   2.963
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)           1.89e-01   1.73e-01    1.10  0.27276
## age                   2.74e-03   2.53e-03    1.08  0.27904
## `jobblue-collar`     -1.80e-01   8.27e-02   -2.17  0.02999 *
## jobentrepreneur      -5.19e-01   1.43e-01   -3.64  0.00027 ***
## jobhousemaid         -4.34e-01   1.55e-01   -2.80  0.00504 **
## jobmanagement        -2.08e-01   8.66e-02   -2.41  0.01606 *
## jobretired            5.45e-01   1.20e-01    4.56  5.2e-06 ***
## `jobself-employed`   -2.91e-01   1.28e-01   -2.27  0.02301 *
## jobservices          -1.82e-01   9.53e-02   -1.91  0.05556 .
## jobstudent            5.58e-01   1.46e-01    3.82  0.00014 ***
## jobtechnician        -1.52e-01   7.99e-02   -1.90  0.05713 .
## jobunemployed         4.80e-02   1.33e-01    0.36  0.71892
## maritalmarried       -1.65e-01   6.86e-02   -2.41  0.01605 *
## maritalsingle         1.89e-01   7.89e-02    2.39  0.01689 *
## educationsecondary    2.43e-01   7.27e-02    3.34  0.00083 ***
## educationtertiary     5.15e-01   8.72e-02    5.91  3.5e-09 ***
## defaultyes           -3.20e-01   1.82e-01   -1.75  0.07927 .
## balance               3.52e-05   7.57e-06    4.65  3.3e-06 ***
## housinghousing_loan -6.10e-01   4.66e-02  -13.07  < 2e-16 ***
## loanpersonal_loan    -5.00e-01   6.64e-02   -7.53  5.0e-14 ***
## day                  -1.15e-02   2.57e-03   -4.45  8.5e-06 ***
## `QtrQtr3_july-sep`    1.49e-01   5.37e-02    2.78  0.00542 **
## `QtrQtr4_oct-dec`     4.14e-01   6.83e-02    6.05  1.4e-09 ***
## `QtrQtr1_jan-march`   3.98e-01   7.03e-02    5.66  1.5e-08 ***
## campaign             -1.02e-01   1.04e-02   -9.78  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 13810  on 9981  degrees of freedom
## Residual deviance: 12759  on 9957  degrees of freedom
## AIC: 12809
##
## Number of Fisher Scoring iterations: 4
```

The summary statistics of the model indicate which variables/features are more important in predicting the output. This is indicated by the p value statistic. p value< .05 indicates that the feature variable is important in predicting the output

Results analysis of Logistic Regression model

Confusion Matrix

```
confusionMatrix(glm_hat,test_set$deposit,positive = "subscribed")
```

```
## Confusion Matrix and Statistics
##
##                 Reference
## Prediction       not_subscribed subscribed
##   not_subscribed            438        220
##   subscribed                146        306
##
##                Accuracy : 0.67
##                  95% CI : (0.642, 0.698)
##     No Information Rate : 0.526
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.334
##
##  Mcnemar's Test P-Value : 0.000136
##
##             Sensitivity : 0.582
##             Specificity : 0.750
##          Pos Pred Value : 0.677
##          Neg Pred Value : 0.666
##              Prevalence : 0.474
##          Detection Rate : 0.276
##    Detection Prevalence : 0.407
##       Balanced Accuracy : 0.666
##
##        'Positive' Class : subscribed
##
```

```
cm_logit<-confusionMatrix(glm_hat,test_set$deposit,positive = "subscribed")
accuracy_logit_all<-cm_logit$overall["Accuracy"]
sensitivity_logit_all<-cm_logit$byClass["Sensitivity"]
specificity_logit_all<-cm_logit$byClass["Specificity"]
```

Prediction table

```
table(glm_hat,test_set$deposit)
```

```
##
## glm_hat          not_subscribed subscribed
##   not_subscribed            438        220
##   subscribed                146        306
```

Model Evaluation Metrices

-Overall Accuracy of model

```
accuracy_logit_all
```

```
## Accuracy
##   0.6703
```

-Sensitivity of the model: proportion of clients who subscribed the term deposit that were predicted correctly

```
sensitivity_logit_all
```

```
## Sensitivity
##      0.5817
```

Results Table

```
results<-tibble(Method="Logistic Regression",
                Accuracy=accuracy_logit_all,
                Sensitivity=sensitivity_logit_all,
                Specificity=specificity_logit_all)
results%>%knitr::kable()
```

Method	Accuracy	Sensitivity	Specificity
Logistic Regression	0.6703	0.5817	0.75

With Logistic Regression model, an over all accuracy of 67% obtained. Sensitivity of 58% was obtained which indicates that we were able to predict 58% of the clients who subscribed to term deposit correctly.

7.2 SVM (Support Vector Machine)

An SVM model is a representation of data points mapped so that separate categories/classes are divided by a clear wide gap. New data points are then assigned to class based on which side of the gap they fall on. SVM model is implemented by finding a hyperplane to classify the data points. The aim is to find a plane that has the maximum margin, i.e a hyperplane that maximizes the margin between the classes. Data points falling on either side of the hyperplane can be attributed to different classes. Support vectors are data points that are nearest to the hyperplane Using these support vectors, we maximize the margin of the classifier which is the distance between the data points and the hyperplane.

In logistic regression, the output of the linear function is assigned value within the range of (0-1) using the sigmoid function. If the assigned value is greater than a threshold value (0.5) it is assigned to one class else to other class in a binary classification model. In SVM, if output of the linear function is greater than 1, it is assigned to one class and if the output is -1, it is assigned to another class.

Training and evaluation of model

```
set.seed(1,sample.kind = "Rounding")
svm_model<-svm(deposit~.,data = train_set,method="class")
svm_pred<-predict(svm_model,newdata = test_set,type="class")
```

Summary of the SVM model

```
summary(svm_model)
```

```
##
## Call:
## svm(formula = deposit ~ ., data = train_set, method = "class")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##
## Number of Support Vectors:  7721
##
##  ( 3848 3873 )
##
##
## Number of Classes:  2
##
## Levels:
##  not_subscribed subscribed
```

Tuning the SVM model

In the initial SVM model, we did not use the 'cost' and 'gama' parameter. However, for effective model, optimum cost and gamma parameters need to be obtained by tuning the SVM model.

Higher value of gama parameter may lead to trying to exact fit as per the training data set leading to over-fitting.

A Low-cost parameter makes the decision surface smooth, while a high cost parameter looks to classify all the data points correctly by giving the model freedom to select more data points as support vectors. It is a trade off between smooth decision boundary and classifying the training points correctly.

tune function is used to tune the SVM model

```
svm_tune<-tune(svm,train.x = deposit~.,data = train_set,
               kernel="radial",
               ranges = list(cost=c(.1,1),gamma=c(.5,1)))
```

Summary of tuned SVM model

```
summary(svm_tune)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost gamma
##     1   0.5
##
## - best performance: 0.3326
##
## - Detailed performance results:
##   cost gamma  error dispersion
## 1  0.1   0.5 0.3397    0.01483
## 2  1.0   0.5 0.3326    0.01027
## 3  0.1   1.0 0.4078    0.01602
## 4  1.0   1.0 0.3460    0.01409
```

Analyzing the summary statistics of the tuned SVM model, it can be seen that optimum parameters are cost=0.1 and gama=0.5

Training and evaluation using tuned model parameters of cost and gama

```
set.seed(1,sample.kind = "Rounding")
svm_tune_model<-svm(deposit~.,data = train_set,cost=.1,gamma=0.5,method="class")
svm_tune_pred<-predict(svm_tune_model,newdata = test_set,type="class")
```

```
summary(svm_tune_model)
```

```
##
## Call:
## svm(formula = deposit ~ ., data = train_set, cost = 0.1, gamma = 0.5,
##     method = "class")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  0.1
##
## Number of Support Vectors:  8681
##
```

```
##  ( 4303 4378 )
##
##
## Number of Classes:  2
##
## Levels:
##  not_subscribed subscribed
```

Results analysis of the SVM Model

```
cm_svm_tune<-confusionMatrix(svm_tune_pred,test_set$deposit,
                             positive = "subscribed")
accuracy_svm<-cm_svm_tune$overall["Accuracy"]
sensitivity_svm<-cm_svm_tune$byClass["Sensitivity"]
specificity_svm<-cm_svm_tune$byClass["Specificity"]
```

Confusion Matrix

```
cm_svm_tune
```

```
## Confusion Matrix and Statistics
##
##                 Reference
## Prediction       not_subscribed subscribed
##   not_subscribed            483        264
##   subscribed                101        262
##
##                Accuracy : 0.671
##                  95% CI : (0.643, 0.699)
##     No Information Rate : 0.526
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.33
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.498
##             Specificity : 0.827
##          Pos Pred Value : 0.722
##          Neg Pred Value : 0.647
##              Prevalence : 0.474
##          Detection Rate : 0.236
##    Detection Prevalence : 0.327
##       Balanced Accuracy : 0.663
##
##        'Positive' Class : subscribed
##
```

Prediction table

```
table(svm_tune_pred,test_set$deposit)
```

```
##
## svm_tune_pred    not_subscribed subscribed
##   not_subscribed            483        264
##   subscribed                101        262
```

Model Evaluation Metrices

Overall Accuracy of SVM model

```
accuracy_svm
```

```
## Accuracy
##   0.6712
```

Sensitivity of the SVM Model: Proportion of clients who subscribed to term deposit that were predicted correctly

```
sensitivity_svm
```

```
## Sensitivity
##      0.4981
```

Updating results table

```
results<-results%>%add_row(Method="SVM",
                           Accuracy=accuracy_svm,
                           Sensitivity=sensitivity_svm,
                           Specificity=specificity_svm)
results%>%knitr::kable()
```

Method	Accuracy	Sensitivity	Specificity
Logistic Regression	0.6703	0.5817	0.7500
SVM	0.6712	0.4981	0.8271

With SVM model, we obtained an over all accuracy of 67%. Sensitivity of 50% was obtained which was lower than that obtained with Logistic regression model. The SVM model did not provide improvement over the Logisitc Regression model.

7.3 KNN (K Nearest Neighbors)

KNN (K nearest Neighbor) algorithm is a model that classifies data points based on the points that are most similar to it. The new data point is assigned a class based on how closely it matches the points in the training set. If K=1, then the new data point point is assigned a value or category depending the class of the only nearest point in its neighborhood. If K=3, then the new data point is assigned a class belonging to the class of the most of the three nearest neighbors. The K nearest neighbors of the new data point are calculated on the basis of Euclidean distance.

Tuning the KNN model

With k=1, there is problem over-training. If k is large then it does not permit enough flexibility and leads to over-smoothing. For the KNN method, different values of K can be tried, k=3,5,7,9. We can change this using the tuneGrid parameter.

To prevent over training that could happen if use the same data set to optimize the parameter k and for evaluation, 10-fold cross validation has been used with the trainControl argument.

One of the requirements in the KNN model is to the scale the training data set. Scaling is required as the predictor variables may have different ranges, and when implementing KNN model, this needs to be addressed so that certain attributes do not influence the model. We have used preProcess argument to scale the train set. We have used preProcess argument with 'center' and 'scale' options to scale the train set.

Training and Evaluation using KNN

```
set.seed(1, sample.kind="Rounding")#if using R 3.5 or earlier, use `set.seed(1)`

knn_model<-train(deposit ~ .,
                 method = "knn",
                 preProcess=c("center","scale"),
                 data = train_set,
                 tuneGrid = data.frame(k = seq(1, 53, 2)),
                 trControl = trainControl(method = "cv", number = 10, p = 0.9))

knn_pred<-predict(knn_model,test_set)
```

KNN Model

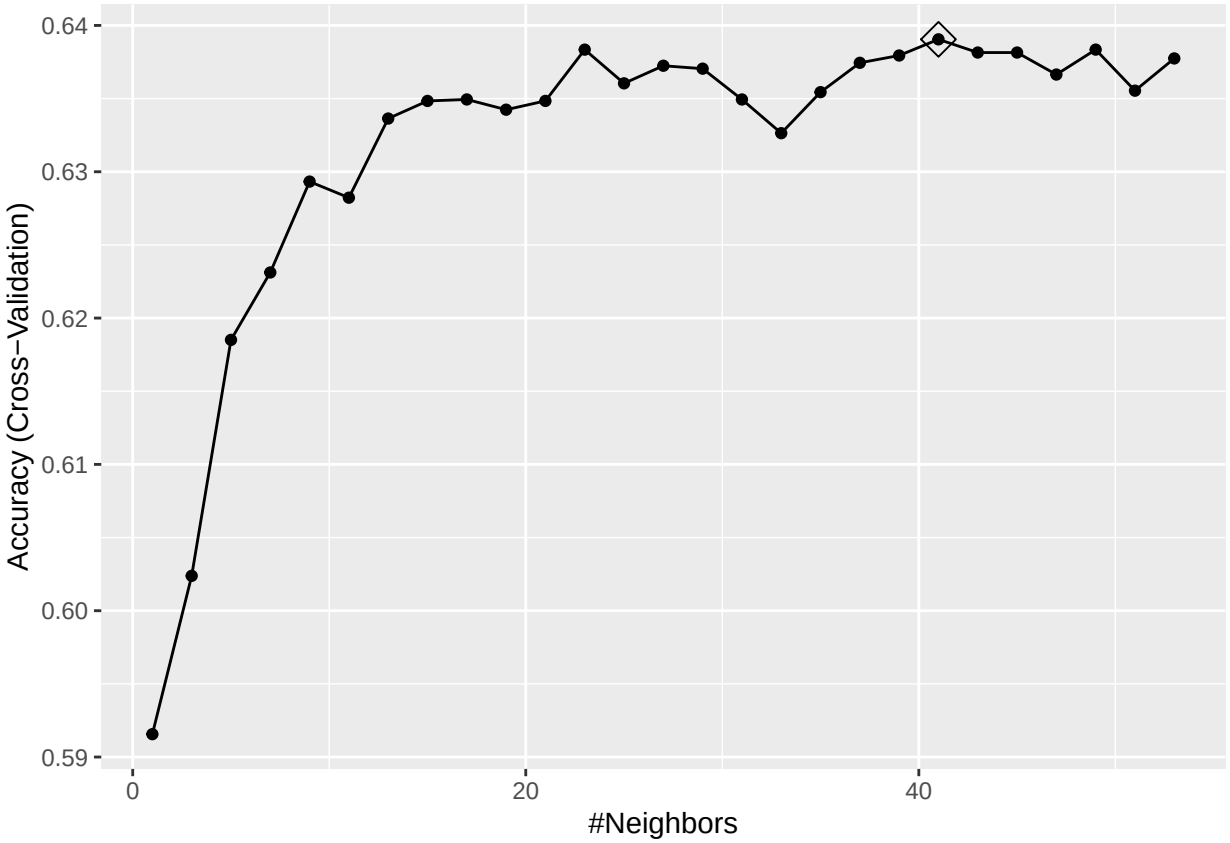
```
knn_model
```

```
## k-Nearest Neighbors
##
## 9982 samples
##   11 predictor
##    2 classes: 'not_subscribed', 'subscribed'
##
## Pre-processing: centered (24), scaled (24)
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 8983, 8984, 8984, 8983, 8984, 8984, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy  Kappa
##    1  0.5916    0.1802
##    3  0.6024    0.2003
##    5  0.6185    0.2317
##    7  0.6231    0.2406
```

```
##    9  0.6293    0.2528
##   11  0.6282    0.2505
##   13  0.6336    0.2611
##   15  0.6348    0.2633
##   17  0.6349    0.2630
##   19  0.6342    0.2615
##   21  0.6348    0.2626
##   23  0.6383    0.2697
##   25  0.6360    0.2650
##   27  0.6372    0.2672
##   29  0.6370    0.2666
##   31  0.6349    0.2625
##   33  0.6326    0.2578
##   35  0.6354    0.2633
##   37  0.6374    0.2673
##   39  0.6379    0.2682
##   41  0.6391    0.2702
##   43  0.6381    0.2684
##   45  0.6381    0.2683
##   47  0.6366    0.2651
##   49  0.6383    0.2685
##   51  0.6355    0.2628
##   53  0.6377    0.2671
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 41.
```

Plotting the Accuracy obtained with the model with different values of k

```
ggplot(knn_model, highlight = TRUE)
```




```
knn_model$bestTune
```

```
##     k
## 21 41
```

The optimum value of k is the value that gives the highest accuracy

Results analysis of the KNN model

```
cm_knn<-confusionMatrix(knn_pred,test_set$deposit,positive = "subscribed")

accuracy_knn<-cm_knn$overall["Accuracy"]
sensitivity_knn<-cm_knn$byClass["Sensitivity"]
specificity_knn<-cm_knn$byClass["Specificity"]
```

Confusion Matrix

```
cm_knn
```

```
## Confusion Matrix and Statistics
##
##                 Reference
## Prediction       not_subscribed subscribed
##   not_subscribed            456        264
```

```
##   subscribed                128        262
##
##                Accuracy : 0.647
##                  95% CI : (0.618, 0.675)
##     No Information Rate : 0.526
##     P-Value [Acc > NIR] : 2.78e-16
##
##                   Kappa : 0.283
##
##  Mcnemar's Test P-Value : 9.20e-12
##
##             Sensitivity : 0.498
##             Specificity : 0.781
##          Pos Pred Value : 0.672
##          Neg Pred Value : 0.633
##              Prevalence : 0.474
##          Detection Rate : 0.236
##    Detection Prevalence : 0.351
##       Balanced Accuracy : 0.639
##
##        'Positive' Class : subscribed
##
```

Prediction Table

```
table(knn_pred,test_set$deposit)
```

```
##
## knn_pred         not_subscribed subscribed
##   not_subscribed            456        264
##   subscribed                128        262
```

Model Evaluation metrices

Overall Accuracy with KNN model

```
accuracy_knn
```

```
## Accuracy
##   0.6468
```

Sensitivty

```
sensitivity_knn
```

```
## Sensitivity
##      0.4981
```

Updating the results table

```
results<-results%>%add_row(Method="KNN",
                           Accuracy=accuracy_knn,
                           Sensitivity=sensitivity_knn,
                           Specificity=specificity_knn)
results%>%knitr::kable()
```

Method	Accuracy	Sensitivity	Specificity
Logistic Regression	0.6703	0.5817	0.7500
SVM	0.6712	0.4981	0.8271
KNN	0.6468	0.4981	0.7808

With KNN model, an overall accuracy of 65% was achieved. Also we were able to predict only 50% (sensitivity is 50%)of the clients who subscribed to term deposit correctly. The KNN model did not provide improvement in overall accuracy or in sensitivity.

7.4 Recursive Partitioning and Regression Trees (RPART) Decision Tree

Decision Tree is a Non-Linear classification model that uses a tree structure to classify. They are easy to interpret and visualize. In Decision Tree Machine Learning Algorithm, the data is split into two or more homogeneous sets based on the most significant independent variables to make as distinct groups as possible.

Training and evaluation with RPART model

Arguments of the function

method: method="class" as the dependent variable(deposit) is categorical

```
set.seed(1, sample.kind="Rounding")#if using R 3.5 or earlier, use `set.seed(1)

rpart_model<-rpart(deposit~.,data = train_set,method = "class")

rpart_pred<-predict(rpart_model,newdata = test_set,type = "class")
```

Tuning the cp parameter

The Complexity Parameter(cp) is used to control the size of the decision tree and to select the optimum tree size.

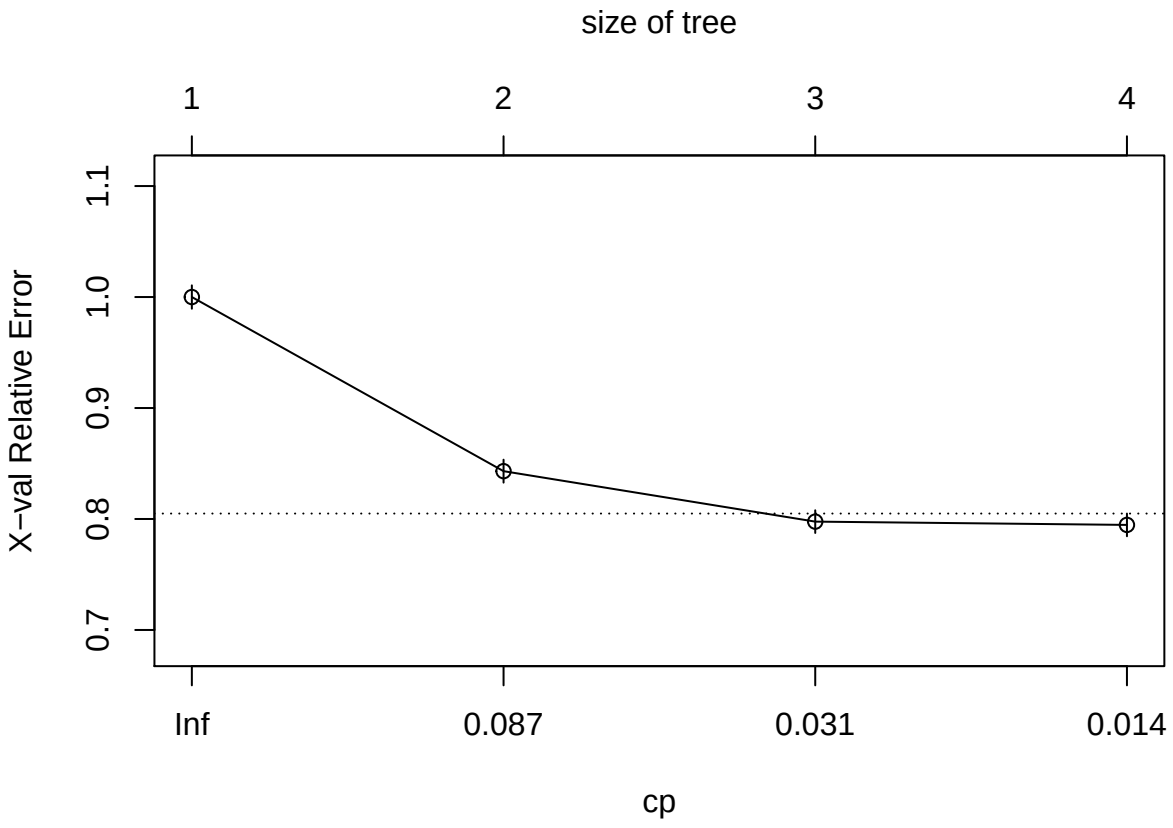
The printcp and plotcp functions provide the cross-validated error (xerror) for each split and is used to prune the tree.

The cp value that gives the lowest xerror is selected as the optimal cp.

```
printcp(rpart_model)
```

```
##
## Classification tree:
## rpart(formula = deposit ~ ., data = train_set, method = "class")
##
## Variables actually used in tree construction:
## [1] balance housing loan
##
## Root node error: 4729/9982 = 0.47
##
## n= 9982
##
##      CP nsplit rel error xerror  xstd
## 1 0.157      0      1.00   1.00 0.011
## 2 0.048      1      0.84   0.84 0.010
## 3 0.020      2      0.80   0.80 0.010
## 4 0.010      3      0.78   0.79 0.010
```

```
plotcp(rpart_model)
```




```
min(rpart_model$cptable[,"xerror"])
```

```
## [1] 0.7947
```

```
which.min(rpart_model$cptable[,"xerror"])
```

```
## 4
## 4
```

Complexity Paramter that minimized the xerror

```
cpmin<-rpart_model$cptable[4,"CP"]
cpmin
```

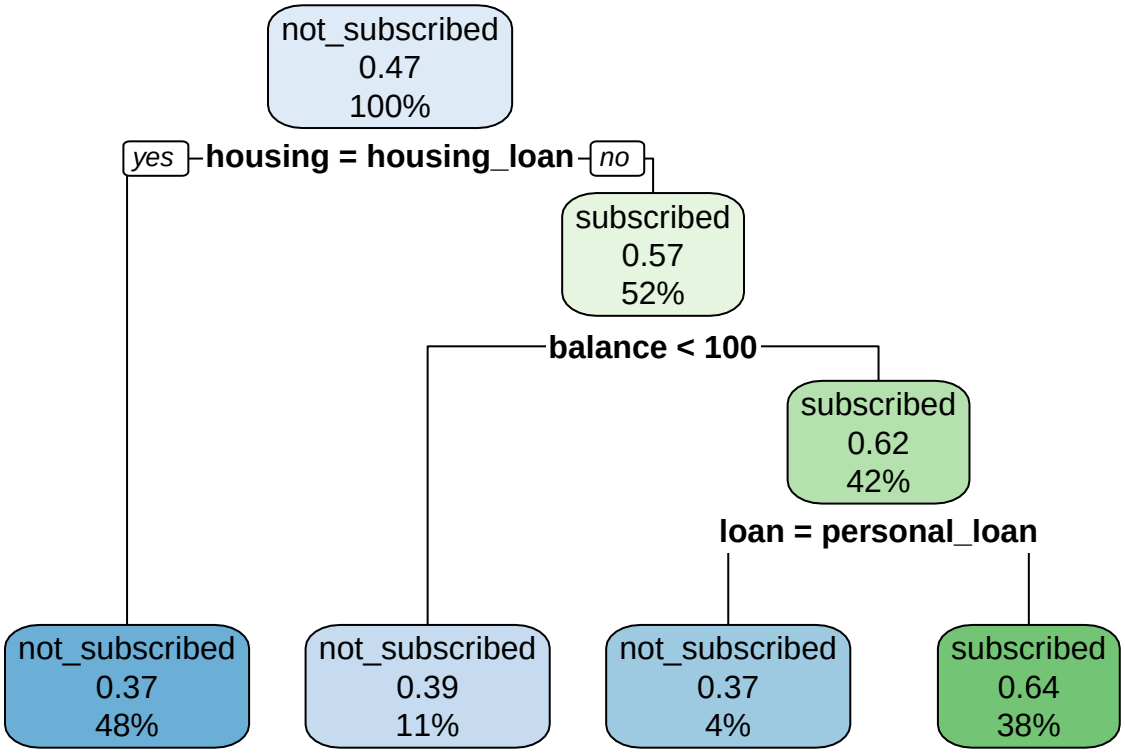
```
## [1] 0.01
```

Pruned RPART model with optimum cp(Complexity Parameter)

```
rpart_pruned<-prune(rpart_model,cp=cpmin)
```

Plot of Tuned RPART (Decision Tree) Model

```
rpart.plot(rpart_pruned)
```



Analysis of Decision tree

The decision tree can be analyzed to see which features are used to predict the output. Housing loan is the most important feature as indicated by decision tree. If the client has a housing loan then they would not subscribe to term deposit. However if the client doesnt have a housing loan, the decision to subscribe to term deposit would depend on the yearly balance and whether the client has a personal loan. If the yearly balance is less than 100 then the client will not subscribe. On the other hand if the balance is more than 100, then client will subscribe to term deposit if there is no personal loan with the client.

Evaluation with test set using pruned RPART model

```
prune_pred<-predict(rpart_pruned,newdata = test_set,type = "class")
```

Results analysis of RPART model

```
cm_rpart<-confusionMatrix(prune_pred,test_set$deposit,positive = "subscribed")
accuracy_prune<-cm_rpart$overall["Accuracy"]
sensitivity_prune<-cm_rpart$byClass["Sensitivity"]
specificity_prune<-cm_rpart$byClass["Specificity"]
```

Confusion Matrix

```
cm_rpart
```

```
## Confusion Matrix and Statistics
##
##                 Reference
## Prediction       not_subscribed subscribed
##   not_subscribed            437        248
##   subscribed                147        278
##
##                Accuracy : 0.644
##                  95% CI : (0.615, 0.672)
##     No Information Rate : 0.526
##     P-Value [Acc > NIR] : 1.25e-15
##
##                   Kappa : 0.279
##
##  Mcnemar's Test P-Value : 4.87e-07
##
##             Sensitivity : 0.529
##             Specificity : 0.748
##          Pos Pred Value : 0.654
##          Neg Pred Value : 0.638
##              Prevalence : 0.474
##          Detection Rate : 0.250
##    Detection Prevalence : 0.383
##       Balanced Accuracy : 0.638
##
##        'Positive' Class : subscribed
##
```

Prediction Table

```
table(prune_pred,test_set$deposit)
```

```
##
## prune_pred       not_subscribed subscribed
##   not_subscribed            437        248
##   subscribed                147        278
```

Model Evaluation metrices

Overall Accuracy with pruned Decision Tree model

```
accuracy_prune
```

```
## Accuracy
##   0.6441
```

Sensitivity

```
sensitivity_prune
```

```
## Sensitivity
##      0.5285
```

Updating the results table

```
results<-results%>%add_row(Method="RPART Decision Tree",
                           Accuracy=accuracy_prune,
                           Sensitivity=sensitivity_prune,
                           Specificity=specificity_prune)
results%>%knitr::kable()
```

Method	Accuracy	Sensitivity	Specificity
Logistic Regression	0.6703	0.5817	0.7500
SVM	0.6712	0.4981	0.8271
KNN	0.6468	0.4981	0.7808
RPART Decision Tree	0.6441	0.5285	0.7483

With the RPART model, an overall accuracy of 64% was obtained. Sensitivity was 53% indicating that 53% of the clients who subscribed to term deposit were predicted correctly for the test set.Although, there was as slight improvement in sensitivity from SVM and KNN, but both the accuracy and sensitivity were lower than those achieved with Logistic Regression. Hence the RPART model was not able to improve performance over the Logistic Regression model.

7.5 Random Forest

Random Forest is a supervised classification algorithm. The Random Forest model addresses the shortcomings in the decision tree model. Random Forest builds multiple decision trees and then averages these trees to get a more accurate and stable prediction. The goal is to improve the prediction performance and reduce instability by averaging multiple decision trees.

Training and evaluating the model

-ntree: this argument states the number of trees to grow. The value is set at 520

-mtry: No of variables randomly sampled as candidates at each split. The default value for classification is sqrt(p) where p is the number of variables. In our data set, there are 11 predictor variables, so default value of mtry is 3.

-importance: TRUE value indicates that importance of the predictors would be assessed.

```
set.seed(42,sample.kind = "Rounding")

rf_model<-randomForest(deposit~.,data=train_set,importance=TRUE,ntree=520, mtry=3)

rf_pred<-predict(rf_model,test_set)
```

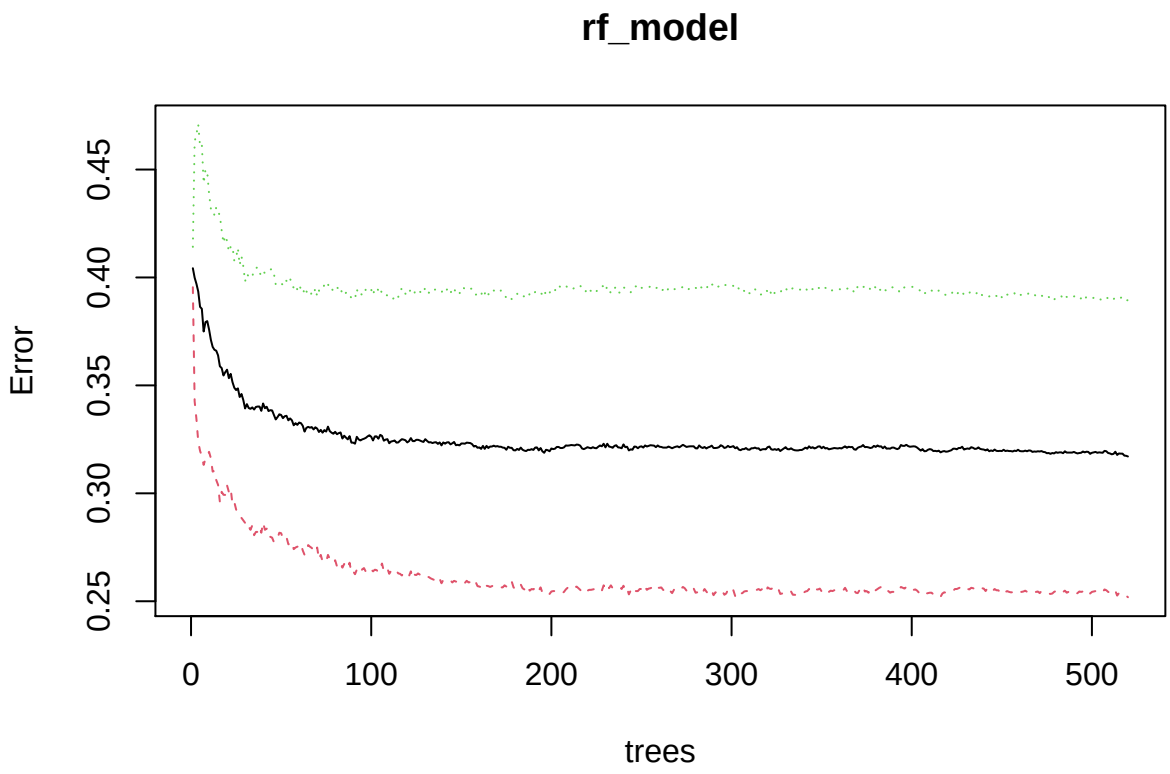
Random Forest model

```
rf_model
```

```
##
## Call:
##  randomForest(formula = deposit ~ ., data = train_set, importance = TRUE,      ntree = 520, mtry = 3)
##                Type of random forest: classification
##                      Number of trees: 520
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 31.71%
## Confusion matrix:
##                not_subscribed subscribed class.error
## not_subscribed           3930       1323      0.2519
## subscribed               1842       2887      0.3895
```

Plot of error rate and ntree

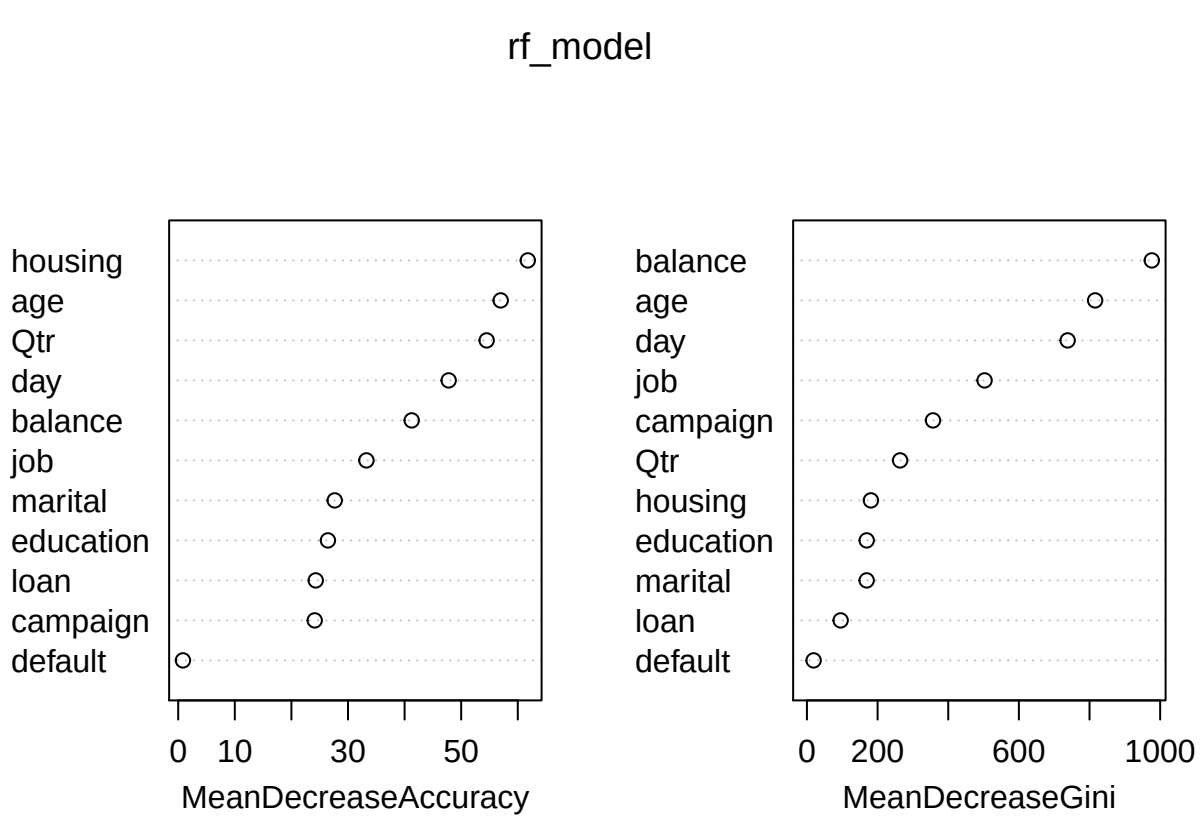
```
plot(rf_model)
```



The plot shows impact of adding more no of trees to the error rate. As can be seen, the addition of more no of trees to the model initially reduces the error rate or improves the accuracy. But after a certain point, adding more trees does not lower the error rate.By growing the tree beyond a point,improvement in overall accuracy is not achieved. In the model, ntree value of 520 has been considered.

Plot of Importance of various predictors to the random forest model

```
varImpPlot(rf_model)
```



From the plot it can be seen that which are the important variables to the random forest model. Some of the important variables are 'housing', 'balance', 'age'.

Results analysis of Random Forest Model

```
cm_rf<-confusionMatrix(rf_pred,test_set$deposit,positive = "subscribed")

accuracy_rf<-cm_rf$overall["Accuracy"]
sensitivity_rf<-cm_rf$byClass["Sensitivity"]
specificity_rf<-cm_rf$byClass["Specificity"]
```

Confusion Matrix

```
cm_rf
```

```
## Confusion Matrix and Statistics
##
##                 Reference
## Prediction       not_subscribed subscribed
##   not_subscribed            464        201
##   subscribed                120        325
##
##                Accuracy : 0.711
##                  95% CI : (0.683, 0.737)
##     No Information Rate : 0.526
##     P-Value [Acc > NIR] : <2e-16
```

```
##
##                   Kappa : 0.416
##
##  Mcnemar's Test P-Value : 8e-06
##
##             Sensitivity : 0.618
##             Specificity : 0.795
##          Pos Pred Value : 0.730
##          Neg Pred Value : 0.698
##              Prevalence : 0.474
##          Detection Rate : 0.293
##    Detection Prevalence : 0.401
##       Balanced Accuracy : 0.706
##
##        'Positive' Class : subscribed
##
```

Prediction table

```
table(rf_pred,test_set$deposit)
```

```
##
## rf_pred          not_subscribed subscribed
##   not_subscribed            464        201
##   subscribed                120        325
```

RF Model Evaluation metrices

OVerall Accuracy with Random Forest

```
accuracy_rf
```

```
## Accuracy
##   0.7108
```

Sensitivity

```
sensitivity_rf
```

```
## Sensitivity
##      0.6179
```

Updating the results table

```
results<-results%>%add_row(Method="Random Forest",
                           Accuracy=accuracy_rf,
                           Sensitivity=sensitivity_rf,
                           Specificity=specificity_rf)
results%>%knitr::kable()
```

Method	Accuracy	Sensitivity	Specificity
Logistic Regression	0.6703	0.5817	0.7500
SVM	0.6712	0.4981	0.8271
KNN	0.6468	0.4981	0.7808
RPART Decision Tree	0.6441	0.5285	0.7483
Random Forest	0.7108	0.6179	0.7945

Random Forest model provided the best performance among the five models. Not only it provided the best overall Accuracy of 71%, the sensitivity of 62% was also the highest among all the Machine Learning models used. Random Forest model is therefore selected as the best performing model among the five models.

Advantages of Random Forest model over Decision Tree

Decision trees may lead to over fitting as they lack accuracy. The approach followed in random forest is that of bootstrap or bagging. The aim is to generate many predictors, each using classification trees, and then forming a final prediction based on the average prediction of the trees. To make sure that the individual trees are not the same bootstrap is used to induce randomness thereby ensuring the trees are randomly different.

8. Conclusion

The Banks and Financial institutions look to analyze the marketing campaigns performed and identify patterns that will help to improve strategies for future campaigns. The data for the project is related to the marketing campaign of a Portuguese banking institution. The classification goal of the project was to predict if the client will subscribe to a term deposit offered by the banking institution. The project also aims to suggest how the effectiveness of marketing campaign be improved and what are the key client attributes that the banking company should focus on while they reach out to clients. The original data set was comprised 11162 clients (observations in the data set) and 17 variables(attributes). After initial data exploration and feature engineering, the 'contact' attribute and the attributes of previous marketing campaign(pdays,previous,poutcome) were removed because we wanted to focus on attributes related to current marketing campaign. Also, Large proportion of the clients in the data set are those that are contacted for the first time. The 'duration' variable may highly affect the results as when the duration is 0, the output is 'no'. Yet, the duration is not known, before the contact(call) is made. After the call duration is obviously known. Hence to build a realistic model, the 'duration' variable was also not considered. After removing the variables- (pdays,previous,poutcome,day,contact,duration) and imputing the unknown values in job and education variable, the final data set used for model building had 11092 observations and 12 variables. Following models were used for classification objective-

-Logistic Regression

-SVM (Support Vector Machines)

-KNN (K Nearest Neighbors)

-RPART (Recursive Partitioning and Regression Trees)

-Random Forest

The objective was to build an optimum model that would work in all situations, and even for data sets that are imbalanced. Therefore the performance of the Models was evaluated with two metrics- Overall Accuracy (Proportion of the output variable values that were predicted correctly) and Sensitivity-(Proportion of positive outcomes predicted correctly i.e proportion of clients who subscribed the term deposit that were predicted correctly.)

The final table for performance metrics for the five models used is as below:-

```
final_evaluation_table<-tibble(method=c("Logistic Regression","SVM","KNN",
                                        "RPART","Random Forest"),
                               Accuracy=c(accuracy_logit_all,accuracy_svm,
                                          accuracy_knn,
                                          accuracy_prune,accuracy_rf),
                               Sensitivity=c(sensitivity_logit_all,
                                             sensitivity_svm,sensitivity_knn,
                                             sensitivity_prune,sensitivity_rf))


final_evaluation_table%>%knitr::kable(caption = "Evaluation Matrics Table")
```

Table 7: Evaluation Matrics Table

method	Accuracy	Sensitivity
Logistic Regression	0.6703	0.5817
SVM	0.6712	0.4981
KNN	0.6468	0.4981
RPART	0.6441	0.5285
Random Forest	0.7108	0.6179

Random Forest provided the best accuracy (71%) and sensitivity (62%) of all the Machine Learning models used.

Application of the project

The banking company can utilize the model to assess what segments of clients will subscribe to the term deposit.The banking company or financial institution can deploy the model to select potential customers for their financial product if the the demographic profile of the clients such as age, education, job, marital status and the credit history of the client such as previous loans, credit default status is available. Based on the model,it was inferred that if the clients already have housing loan or personal or any other credit, then the propensity to subscribe to the term deposit would be less. Also, clients who have less income balance with them, they will be less inclined to opt for term deposit.

The model will help banking institutions in the private sector to improve the effectiveness of their marketing campaign, and design it in a flexible way where their outreach and focus is more to the potential clients that have more propensity to opt for the term deposit. The banks can deploy the model and optimize the marketing campaign cost by increasing the outreach or contacts to clients who are more likely to subscribe to the term deposit. These could be clients who have no recent loan history, adequate income balance. The banks may not only look at the duration of the call made to the clients as the long duration calls may not lead to any conclusion whether the clients will take the product or not.

9. Limitations

Some of the classification models were computationally expensive. For eg, the tuning of SVM model was a time taking exercise. Hence the tuning for SVM model was performed with two cost(0.1, 1) and two gama(0.5,1) parameters. More advanced machine Learning models could not be deployed due to the computational and technological limitations.

Duration,a numeric variable which states the duration of the call made to the client was not considered for the model. The duration is not known before the call is made and at the end of the call, it is obviously known, if the client would subscribe or not. So to build a realistic predictive model, duration variable was not included.

Random forest model that provided the best performance in terms of accuracy though address the issue of over fitting with Decision Tree (RPART) model but at the same time, the Random forest model has some limitations. Since Random Forest model creates many trees and combines their outputs, it requires much more computational capability and resources.

The project is developed for the data collected by the Portuguese banking institution. The financial decisions taken by the client are influenced by the macro economic factors and the client's demographic factors. Now in a different macroeconomic scenario, the demographic profile of the clients, previous loan history of the clients may be different which may affect the results or selection of clients who subscribe to term deposit.

10. References

a. https://www.kaggle.com/janiobachmann/bank-marketing-dataset

b. https://rafalab.github.io/dsbook/introduction-to-machine-learning.html

c. https://rafalab.github.io/dsbook/examples-of-algorithms.html#k-nearest-neighbors

d. https://rafalab.github.io/dsbook/caret.html#caret-cv

e. https://www.edureka.co/blog/machine-learning-algorithms/

f. https://topepo.github.io/caret/pre-processing.html#cs

g. https://cran.r-project.org/web/packages/caret/caret.pdf

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

i. https://cran.r-project.org/web/packages/caret/caret.pdf

j. https://towardsdatascience.com