Banking Marketing strategy by Data Analysis Project

AANSH SARDANA

2 JANUARY, 2020

1 Introduction. -

The common consensus about financial institutions is that they serve as the medium linking borrowers and lenders in an economy. Sudden demand shift often cause to the problems of excess supply or excess demand for credit, which affects the profit curve of the financial institution adversely. In order to prevent themselves from such mayhem, they need to conduct market research on a continuous basis.

Market research might help the financial institutions to gather the feedbacks of the consumers on their products and services, on their desires, on the reasons behind their choices et al. This will help them to improve their production efficiency, increasing the demand.

The aim of my project is to use a data analysis method which is better than standard linear regression model to study the behavioral pattern of the future customer those who will be interested in investing term deposit in banking institution.

Here I will be using Random forest and K-NN model for data analysis and finally the best resulting model will be used to predict the customer behaviour and it will help to build a marketting stratergy to attract the targetted customers so as to increase the profit of the banking institutions.

The dataset of my model is included in one of the files and can be downloaded from Kaggle (https://www.kaggle.com/henriqueyamahata/bank-marketing).

About The Dataset:

The dataset contains data of 41,188 customer which are on direct marketing campaigns (phone calls) of a banking institution.

Variables in the dataset:

Client :
1) age
2) job
3) marital
4) education
5) default status
6) housing
7) loan

Campaign:
1) last contact type

2) last contact month of year
3) last contact day of the week
4) last contact duration

Others:
1) number of contacts performed in current campaign
2) number of days that passed by after the client was last contacted
3) number of contacts performed before this campaign
4) outcome of previous campaign
5) whether a client has subscribed a term deposit

Key Steps Performed:

1-Data Collection

It is generally a representation of data which is used for training.

2 - Data Preparation

Wrangle data and prepare it for training.
Clean to remove duplicates, correct errors, deal with missing values, normalization, data type conversions, etc.
Visualize data to help detect relevant relationships between variables or class imbalances (bias alert!), or perform other exploratory analysis
Split into training and evaluation sets

3 - Choose a model
Used random forest and K- nn models

4 - Train and evaluted the models.

5 - Make Predictions
Using test data, a better approximation or prediction of how the model will perform in the real world is made.

2.Data Analysis:

2.1 Starting Analysis

Installing and Loading the required packages:

```
rm(list = ls())
options(warn=-1)
#let's install needed packages and download dataset from my github repository
if(!require(readr)) install.packages("readr", repos = "http://cran.us.r-
project.org")
if(!require(tidyverse)) install.packages("tidyverse", repos = "http://
cran.us.r-project.org")
if(!require(GGally)) install.packages("GGally", repos = "http://cran.us.r-
project.org")
```

```
if(!require(glmnet)) install.packages("glmnet", repos = "http://cran.us.r-
project.org")
if(!require(Matrix)) install.packages("Matrix", repos = "http://cran.us.r-
project.org")
if (!require(devtools)) install.packages("devtools")
if(!require(DataExplorer)) install.packages("DataExplorer")
if(!require(corrplot)) install.packages("corrplot", repos = "http://
cran.us.r-project.org")
if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-
project.org")
if(!require(randomForest)) install.packages("randomForest", repos = "http://
cran.us.r-project.org")
if(!require(class)) install.packages("class", repos = "http://cran.us.r-
project.org")
if(!require(gmodels)) install.packages("gmodels", repos = "http://cran.us.r-
project.org")
if(!require(dplyr)) install.packages("dplyr", repos = "http://cran.us.r-
project.org")
if(!require(psych)) install.packages("psych", repos = "http://cran.us.r-
project.org")
#inititalising the libraries
library(readr)
library(tidyverse)
library(GGally)
library(devtools)
library(glmnet)
library(Matrix)
library(ggplot2)
library(DataExplorer)
library(corrplot)
library(caret)
library(randomForest)
library(class)
library(gmodels)
library(dplyr)
library(psych)
#set the seed to 1
set.seed(1)
```

Loading the dataset:

```
#reading data from my github profile
dt.df <- read.csv("https://raw.github.com/aansh-01/create_your_own/master/
bank-full.csv", header=TRUE, sep=";")
```

Viewing the names of the column in dataset:

```
names(dt.df)
```

```
 [1] "age"          "job"          "marital"        "education"      "default"        "housing"        "loan"
 [8] "contact"      "month"        "day_of_week"    "duration"       "campaign"       "pdays"          "previous"
[15] "poutcome"     "emp.var.rate" "cons.price.idx" "cons.conf.idx"  "euribor3m"      "nr.employed"    "y"
>
```

Details of columns of the dataset:

```
str(dt.df)
```

```
> str(dt.df)
'data.frame':   41188 obs. of  21 variables:
 $ age           : int  56 57 37 40 56 45 59 41 24 25 ...
 $ job           : chr  "housemaid" "services" "services" "admin." ...
 $ marital       : chr  "married" "married" "married" "married" ...
 $ education     : chr  "basic.4y" "high.school" "high.school" "basic.6y" ...
 $ default       : chr  "no" "unknown" "no" "no" ...
 $ housing       : chr  "no" "no" "yes" "no" ...
 $ loan          : chr  "no" "no" "no" "no" ...
 $ contact       : chr  "telephone" "telephone" "telephone" "telephone" ...
 $ month         : chr  "may" "may" "may" "may" ...
 $ day_of_week   : chr  "mon" "mon" "mon" "mon" ...
 $ duration      : int  261 149 226 151 307 198 139 217 380 50 ...
 $ campaign      : int  1 1 1 1 1 1 1 1 1 1 ...
 $ pdays         : int  999 999 999 999 999 999 999 999 999 999 ...
 $ previous      : int  0 0 0 0 0 0 0 0 0 0 ...
 $ poutcome      : chr  "nonexistent" "nonexistent" "nonexistent" "nonexistent" ...
 $ emp.var.rate  : num  1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 ...
 $ cons.price.idx: num  94 94 94 94 94 ...
 $ cons.conf.idx : num  -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 ...
 $ euribor3m     : num  4.86 4.86 4.86 4.86 4.86 ...
 $ nr.employed   : num  5191 5191 5191 5191 5191 ...
 $ y             : chr  "no" "no" "no" "no" ...
>
```

Dataset Summary analysis:

```
summary(dt.df)
```

```
      age            job              marital           education           default            housing              loan          
 Min.   :17.00   Length:41188       Length:41188       Length:41188       Length:41188       Length:41188       Length:41188      
 1st Qu.:32.00   Class :character   Class :character   Class :character   Class :character   Class :character   Class :character  
 Median :38.00   Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character  
 Mean   :40.02                                                                                                                    
 3rd Qu.:47.00                                                                                                                    
 Max.   :98.00                                                                                                                    
   contact             month          day_of_week          duration         campaign          pdays          previous      
 Length:41188       Length:41188       Length:41188       Min.   :   0.0   Min.   : 1.000   Min.   :  0.0   Min.   :0.000  
 Class :character   Class :character   Class :character   1st Qu.: 102.0   1st Qu.: 1.000   1st Qu.:999.0   1st Qu.:0.000  
 Mode  :character   Mode  :character   Mode  :character   Median : 180.0   Median : 2.000   Median :999.0   Median :0.000  
                                                          Mean   : 258.3   Mean   : 2.568   Mean   :962.5   Mean   :0.173  
                                                          3rd Qu.: 319.0   3rd Qu.: 3.000   3rd Qu.:999.0   3rd Qu.:0.000  
                                                          Max.   :4918.0   Max.   :56.000   Max.   :999.0   Max.   :7.000  
   poutcome          emp.var.rate      cons.price.idx  cons.conf.idx     euribor3m      nr.employed        y           
 Length:41188       Min.   :-3.40000   Min.   :92.20   Min.   :-50.8   Min.   :0.634   Min.   :4964   Length:41188      
 Class :character   1st Qu.:-1.80000   1st Qu.:93.08   1st Qu.:-42.7   1st Qu.:1.344   1st Qu.:5099   Class :character  
 Mode  :character   Median : 1.10000   Median :93.75   Median :-41.8   Median :4.857   Median :5191   Mode  :character  
                    Mean   : 0.08189   Mean   :93.58   Mean   :-40.5   Mean   :3.621   Mean   :5167                     
                    3rd Qu.: 1.40000   3rd Qu.:93.99   3rd Qu.:-36.4   3rd Qu.:4.961   3rd Qu.:5228                     
                    Max.   : 1.40000   Max.   :94.77   Max.   :-26.9   Max.   :5.045   Max.   :5228                     
>
```

2.2. Preparation of Data

Checking for any missing value

```
sum(is.na(dt.df))
```

```
> sum(is.na(dt.df))
[1] 0
>
```

There are no missing values present in our dataset.

From starting analysis we conclude that there are many variables with class=int; hence, we need to convert them into numeric class

Converting int values to numeric class:

```
dt.df$pdays <- as.numeric(dt.df$pdays)
```

```
dt.df$duration <- as.numeric(dt.df$duration)
dt.df$age <- as.numeric(dt.df$age)
dt.df$campaign <- as.numeric(dt.df$campaign)
dt.df$previous <- as.numeric(dt.df$previous)
```

Ordered the levels of month:

```
dt.df$month<- factor(dt.df$month, ordered = TRUE, levels = c("mar", "apr",
"may", "jun", "jul", "aug", "sep", "oct", "nov", "dec"))
```

As the target variable is a categorical variable which can have two possible values either yes or no. So we convert it into numerical value 1 and 0 respectively

Converting the target variable into 1 for yes and 0 for no

```
table(dt.df$y)
dt.df <- dt.df %>%
  mutate(y = ifelse(y=="yes", 1, 0))
dt.df$y <- as.factor(dt.df$y)
table(dt.df$y)
```

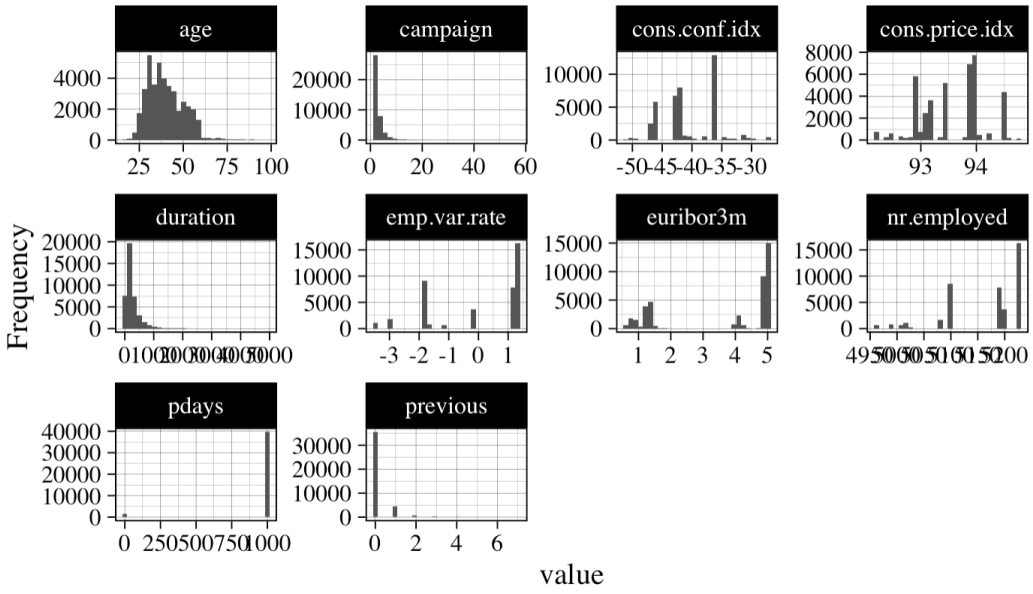
```
> table(dt.df$y)

    0     1
36548  4640
>
```

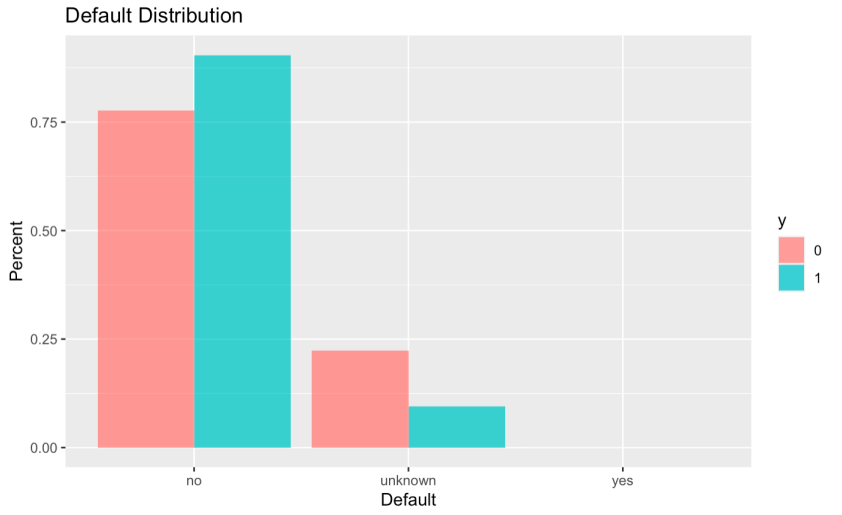
2.3.Descriptive Analysis

Creating the histogram of given input variables

```
plot_histogram(dt.df[,-21],ggtheme = theme_linedraw(base_size = 15,
base_family = "serif"))
```

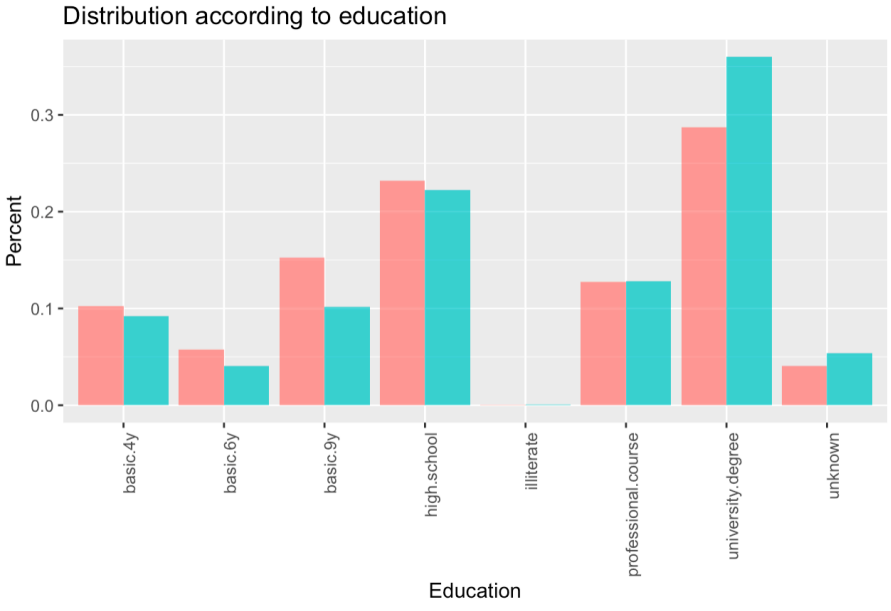


```
mtab <-
table(dt.df$default, dt.df$y)
ptt <- as.data.frame(prop.table(mtab, 2))
colnames(ptt) <-  c("default", "y", "perc")
ggplot(data = ptt, aes(x = default, y = perc, fill = y)) +
  geom_bar(stat = 'identity', position = 'dodge', alpha = 2/3) +
  labs(title = "Default Distribution", y = "Percent", x = "Default")
```



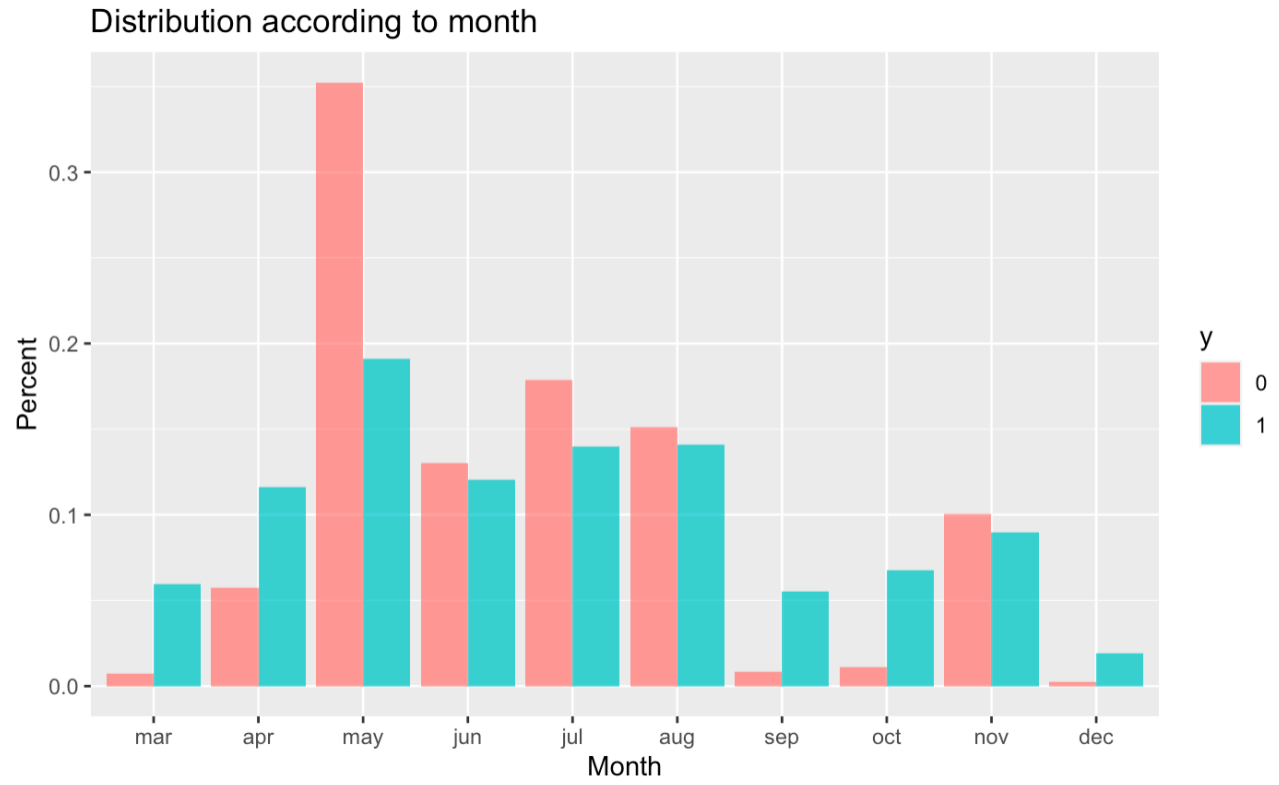
Therefore, People who are in default are higher in number

```
mtab <- table(dt.df$education, dt.df$y)
ptt <- as.data.frame(prop.table(mtab, 2))
colnames(ptt) <-  c("education", "y", "perc")
ggplot(data = ptt, aes(x = education, y = perc, fill = y)) +
  geom_bar(stat = 'identity', position = 'dodge', alpha = 2/3) +
  theme(axis.text.x=element_text(angle=90,hjust=1)) +
  labs(title = "Distribution according to education", y = "Percent", x =
"Education")
```



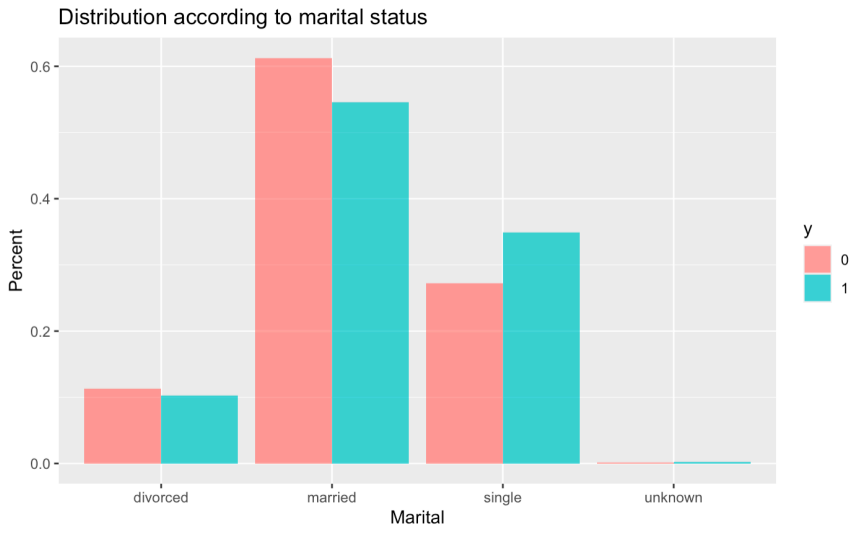
Therefore, customers who sign up for bank deposits, proportionally, have achieved a higher level of education, than those who didn't sign up.

```r
mtab <- table(dt.df$month, dt.df$y)
ptt <- as.data.frame(prop.table(mtab, 2))
colnames(ptt) <-  c("month", "y", "perc")
ggplot(data = ptt, aes(x = month, y = perc, fill = y)) +
  geom_bar(stat = 'identity', position = 'dodge', alpha = 2/3) +
  labs(title = "Distribution according to month", y = "Percent", x = "Month")
```



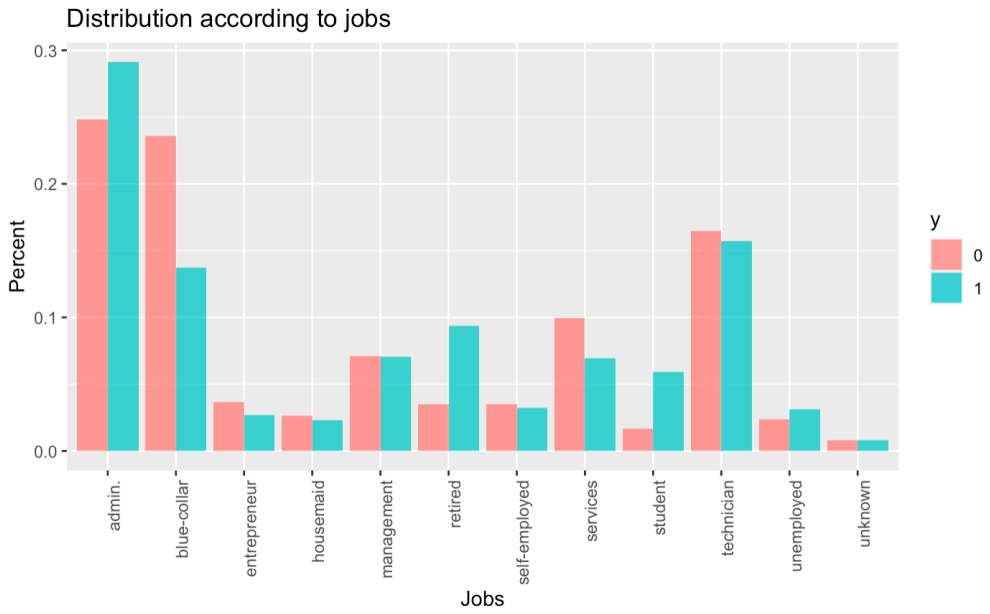
Therefore , the month of May is when the highest number of calls were placed for marketing deposit. And the following months of April, September, and October is the time when a higher proportion of people subscribed for term deposits.

```
mtab <- table(dt.df$marital, dt.df$y)
ptt <- as.data.frame(prop.table(mtab, 2))
colnames(ptt) <-  c("marital", "y", "perc")
ggplot(data = ptt, aes(x = marital, y = perc, fill = y)) +
  geom_bar(stat = 'identity', position = 'dodge', alpha = 2/3) +
  labs(title = "Distribution according to marital status", y = "Percent", x =
"Marital")
```



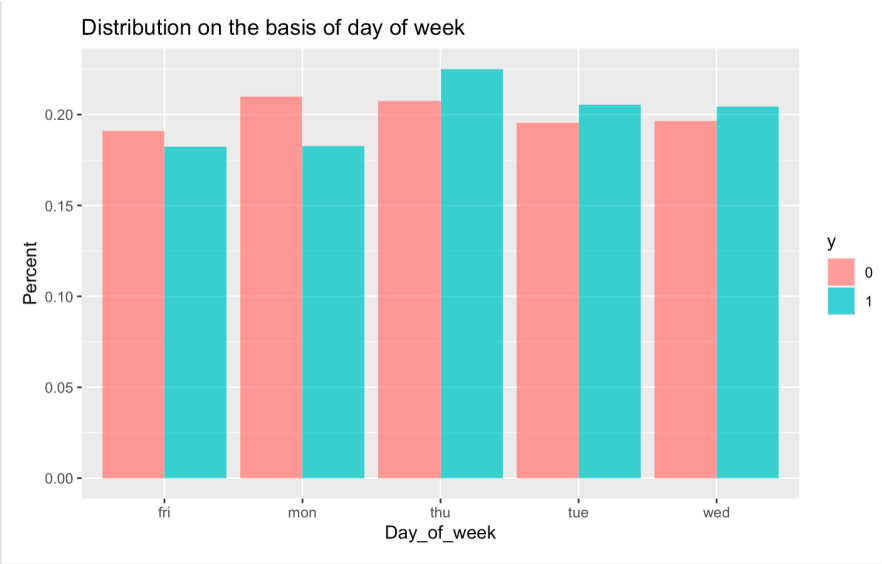
Therefore , With respect to Marital Status there is not an observed large difference in the proportion of people subscribed to term deposits and people without term deposits.

```
mtab <- table(dt.df$job, dt.df$y)
ptt <- as.data.frame(prop.table(mtab, 2))
colnames(ptt) <-  c("job", "y", "perc")
ggplot(data = ptt, aes(x = job, y = perc, fill = y)) +
  geom_bar(stat = 'identity', position = 'dodge', alpha = 2/3) +
  theme(axis.text.x=element_text(angle=90,hjust=1)) +
  labs(title = "Distribution according to jobs", y = "Percent", x = "Jobs")
```



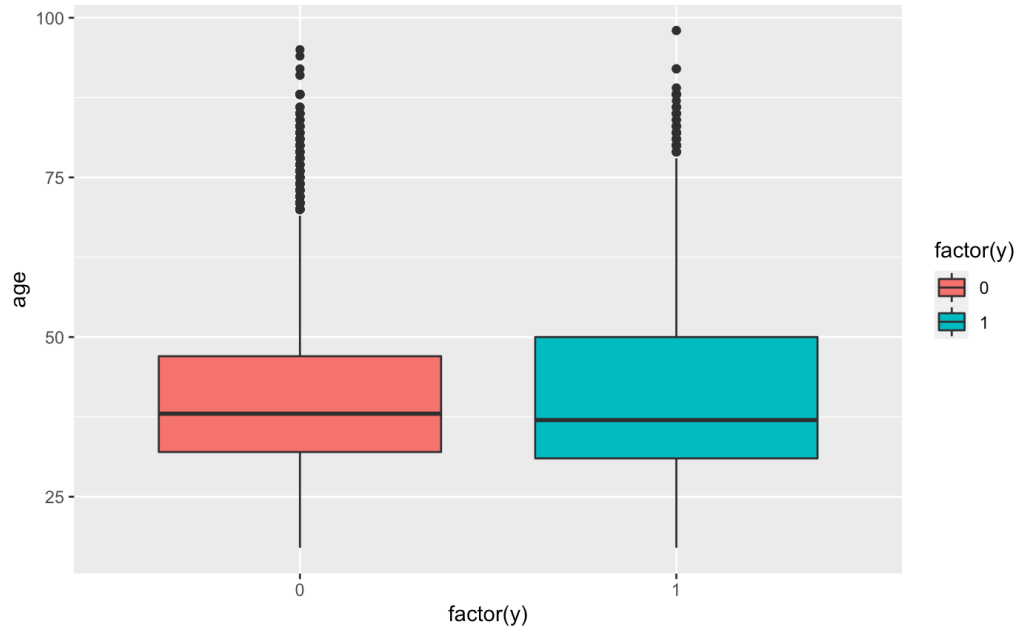
Therefore , We see there are higher proportions for customers signing up for the term deposits who have the jobs of admin, retired, and students.

```
mtab <- table(dt.df$day_of_week, dt.df$y)
ptt <- as.data.frame(prop.table(mtab, 2))
colnames(ptt) <-  c("day_of_week", "y", "perc")
ggplot(data = ptt, aes(x = day_of_week, y = perc, fill = y)) +
  geom_bar(stat = 'identity', position = 'dodge', alpha = 2/3) +
  labs(title = "Distribution on the basis of day of week", y = "Percent", x =
"Day_of_week")
```



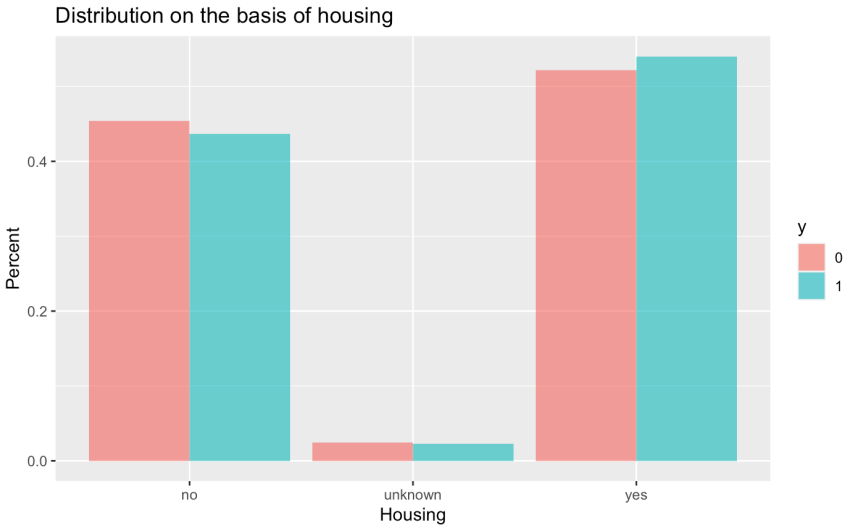
Therefore , Campaigns that were performed midweek, on Tuesdays, Wednesdays, and Thursdays had a slightly higher proportion of people who subscribed for bank deposit.

```
myboxplot<-ggplot(dt.df, aes(factor(y), age)) + geom_boxplot(aes(fill =
factor(y)))
myboxplot
```



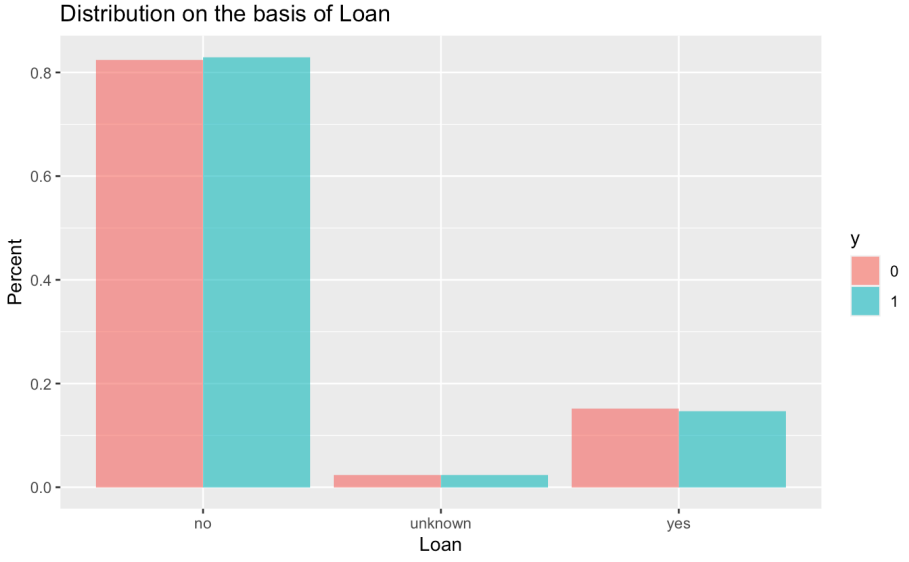
Therefore , the age range for successful conversion has a slightly lower median, but higher quartile ranges.

```
mtab <- table(dt.df$housing, dt.df$y)
ptt <- as.data.frame(prop.table(mtab, 2))
colnames(ptt) <-  c("housing", "y", "perc")
ggplot(data = ptt, aes(x = housing, y = perc, fill = y)) +
  geom_bar(stat = 'identity', position = 'dodge', alpha = 2/3) +
  labs(title = "Distribution on the basis of housing", y = "Percent", x =
"Housing")
```



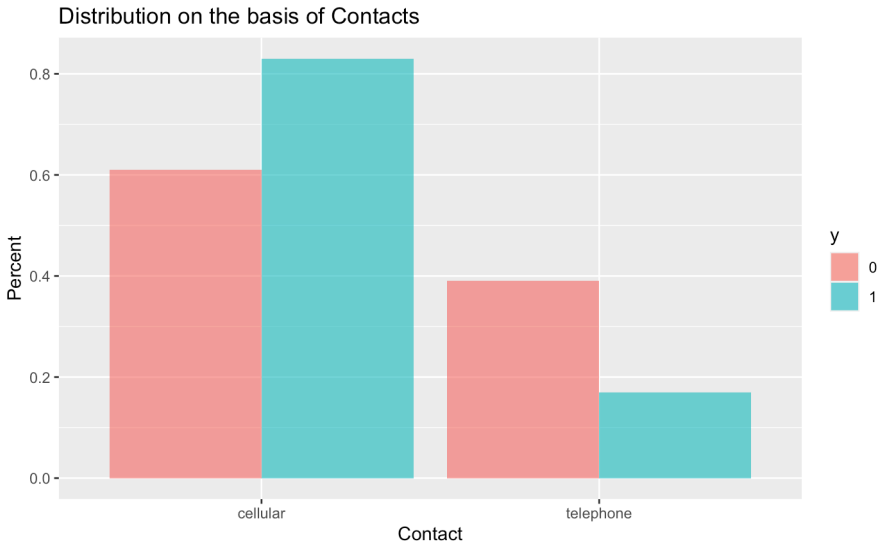
Therefore , we see that a higher proportion of people who have subscribed for bank deposit are home owners versus ones that don't own their own houses.

```
mtab <- table(dt.df$loan, dt.df$y)
ptt <- as.data.frame(prop.table(mtab, 2))
colnames(ptt) <-  c("loan", "y", "perc")
ggplot(data = ptt, aes(x = loan, y = perc, fill = y)) +
  geom_bar(stat = 'identity', position = 'dodge', alpha = 2/3) +
  labs(title = "Distribution on the basis of Loan", y = "Percent", x =
"Loan")
```



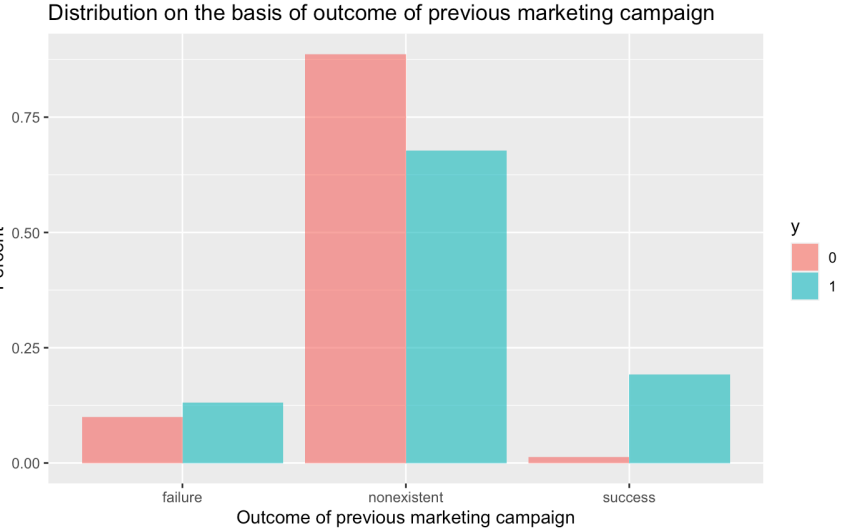
Therefore , we see the proportion of people who have subscribed and not subscribed to a term deposit is the same for categories of the Loan.

```
mtab <- table(dt.df$contact, dt.df$y)
ptt <- as.data.frame(prop.table(mtab, 2))
colnames(ptt) <-  c("contact", "y", "perc")
ggplot(data = ptt, aes(x = contact, y = perc, fill = y)) +
  geom_bar(stat = 'identity', position = 'dodge', alpha = 2/3) +
  labs(title = "Distribution on the basis of Contacts", y = "Percent", x =
"Contact")
```



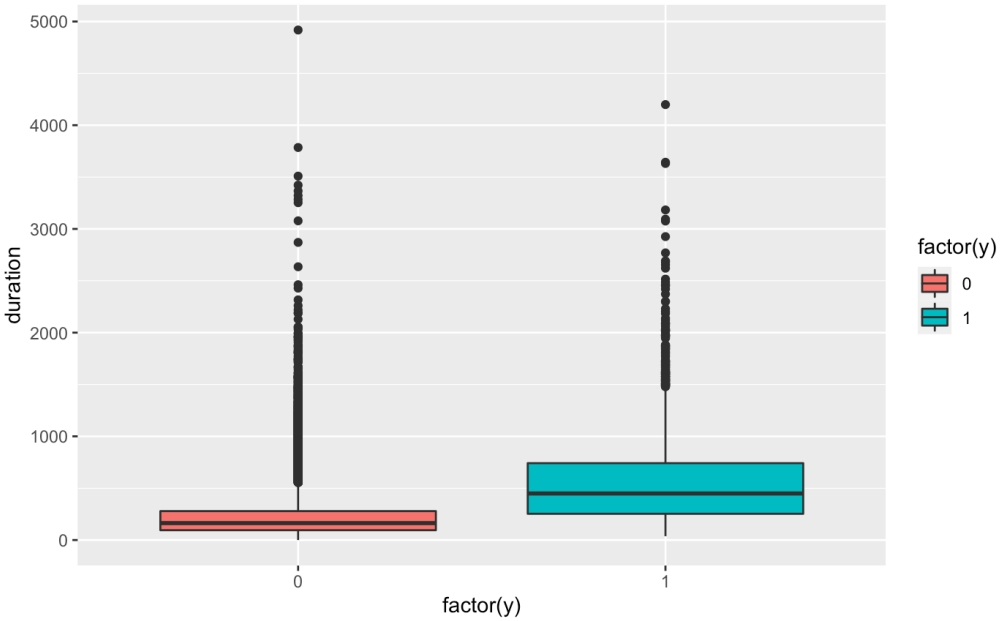
Therefore , Customers who have cell phones have a more direct way of communicating,and signed up for term deposits more than those who only had a landline telephone.

```
mtab <- table(dt.df$poutcome, dt.df$y)
ptt <- as.data.frame(prop.table(mtab, 2))
colnames(ptt) <-  c("poutcome", "y", "perc")
ggplot(data = ptt, aes(x = poutcome, y = perc, fill = y)) +
  geom_bar(stat = 'identity', position = 'dodge', alpha = 2/3) +
  labs(title = "Distribution on the basis of outcome of previous marketing
campaign", y = "Percent", x = "Outcome of previous marketing campaign")
```



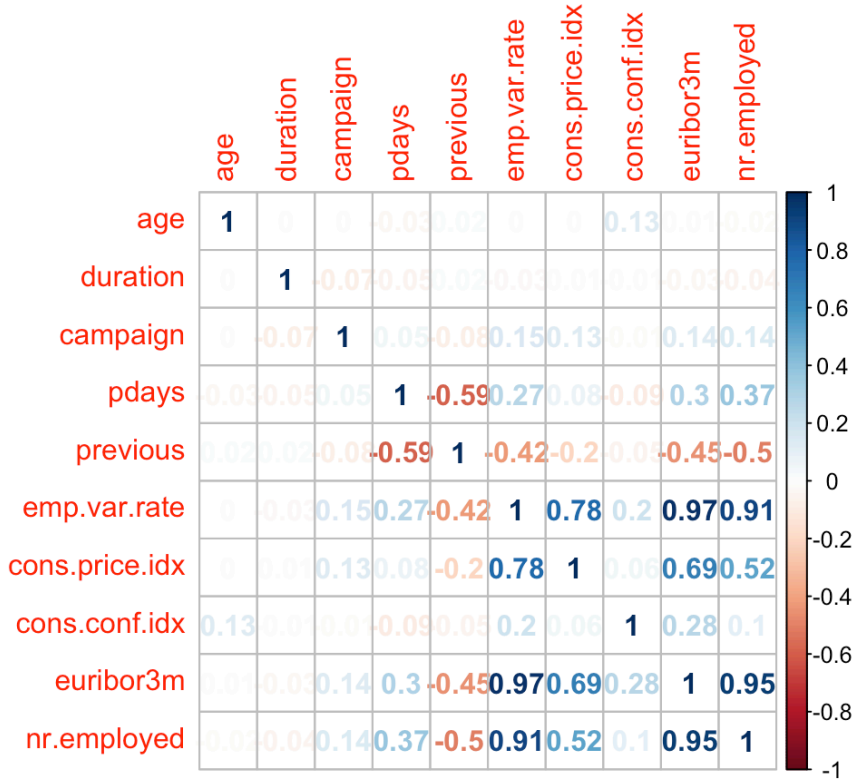
Therefore , Potential customers who successfully connected and responded in previous campaigns had a higher proportion of signing up for the term deposit.

```
myboxplot <- ggplot(dt.df, aes(factor(y), duration)) + geom_boxplot(aes(fill = factor(y)))
myboxplot
```



Therefore , longer the phone conversation the greater the conversion rate is for the potential customer to sign up for the term deposit. There are higher median and quartile ranges.

```
df_corll <- select_if(dt.df, is.numeric) %>% cor()
corrplot(df_corll, method = "number")
```



Therefore , We see our target variable has a high positive correlation with duration and if the customer was involved and connected in a previous campaign, while there's negative correlation with number of employees,number of days from last contact ,Euribor 3 month rate and employee variation rate.

3.Data Modeling and Final Results:

3.1 Data Preparation

Missing values for duration were filtered out because if duration = 0 then y = "no" . Thus, it doesn't make sense to have 0 second duration.
I have also filtered out education illiterate, and default yes because they only have 1 observation each. We can't predict these situations if they happen to be in the test data but not the train data.

```
dt.df <- dt.df %>%
  filter(duration != 0, education != "illiterate", default != "yes") %>%
  mutate(y = ifelse(y==1, 1, 0))
```

Splitting the data into training and test datasets:

```
#setting seed to 123
set.seed(123)
trainIndex <- createDataPartition(dt.df$y,
                                  p = 0.8, # training contains 80% of data
                                  list = FALSE)
dfTrainSet <- dt.df[ trainIndex,]
dfTestSet  <- dt.df[-trainIndex,]
dim(dfTrainSet)
dim(dfTestSet)
```

```
> dim(dfTrainSet)
[1] 32931    21
> dim(dfTestSet)
[1] 8232   21
```

Above output shows that the train dataset has 32931 rows and 21 columns and the test dataset has 8232 rows and 21 columns.
The number of columns remains the same because the dataset was split vertically.

3.2.Data Modeling using Random Forest:

Random forest is a supervised learning algorithm which is used for both classification as well as regression. It creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting.

The data set was categorized into training and testing data with 80%:20% split ratio respectively.
A seed value was set using set.seed() function to regenerate the split data.

A random forest model was built using training data using random forest algorithm.
We use 10 predictors for each split and grow 200 trees fully without pruning.
A subset of predictors is randomly chosen without replacement at each split which helps in reducing the variance of the model overall.

This model gives an Out-Of-Bag error rate of 8.7%. The model also outputs a confusion matrix but maintains a high accuracy in predicting the response variable i.e. term deposit(Yes/No) field.

```
set.seed(123)
# random forest
model_rndfor <- randomForest(as.factor(y)~.,
                         data = dfTrainSet,
                         ntree = 200,
                         mtry=10,
                         importance = TRUE)
print(model_rndfor)
rf_pred_prob <- predict(model_rndfor,newdata = dfTestSet)
head(rf_pred_prob)
```

Model evaluation:

```
# putting "rf_pred_prob" in a data frame
outcome_RF_test <- data.frame(dfTestSet$y)
# merging "model_rndfor" and "outcome_RF_test"
comparison_df_RF <- data.frame(rf_pred_prob, outcome_RF_test)
# specifying column names for "comparison_df_RF"
names(comparison_df_RF) <- c("RF_Predicted_y", "RF_Observed_y")
comparison_df_RF$RF_Predicted_y <- as.factor(comparison_df_RF$RF_Predicted_y)
comparison_df_RF$RF_Observed_y <- as.factor(comparison_df_RF$RF_Observed_y)
# inspect "comparison_df_RF"
head(comparison_df_RF)
str(comparison_df_RF)
confusionMatrix(comparison_df_RF$RF_Observed_y,comparison_df_RF$RF_Predicted_
y)
```

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 7050  296
         1  366  520

               Accuracy : 0.9196
                 95% CI : (0.9135, 0.9254)
    No Information Rate : 0.9009
    P-Value [Acc > NIR] : 2.793e-09

                  Kappa : 0.5663

 Mcnemar's Test P-Value : 0.007324

            Sensitivity : 0.9506
            Specificity : 0.6373
         Pos Pred Value : 0.9597
         Neg Pred Value : 0.5869
             Prevalence : 0.9009
         Detection Rate : 0.8564
   Detection Prevalence : 0.8924
      Balanced Accuracy : 0.7940

       'Positive' Class : 0

> 
```

Total RF observation - 8232

7050 cases out of 8232 have been accurately predicted (TN->True Negatives) as negative class (0) which constitutes 85%.

510 out of 8232 observations were accurately predicted (TP-> True Positives) as positive class (1) which constitutes 6%.

Thus a total of 510 out of 8232 predictions where TP i.e, True Positive in nature.

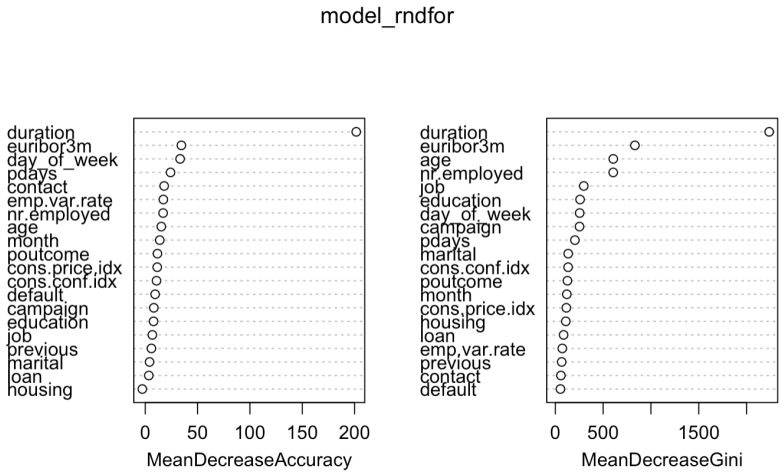
There were 520 cases of False Positives (FP) meaning 520 cases out of 8232 were actually negative but got predicted as positive.

There were 366 cases of False Negatives (FN) meaning 366 cases out of 8232 were actually positive in nature but got predicted as negative.

Accuracy of the model is the correctly classified positive and negative cases divided by all ther cases.The total accuracy of the model is 91.96%, which means the model prediction is very accurate.

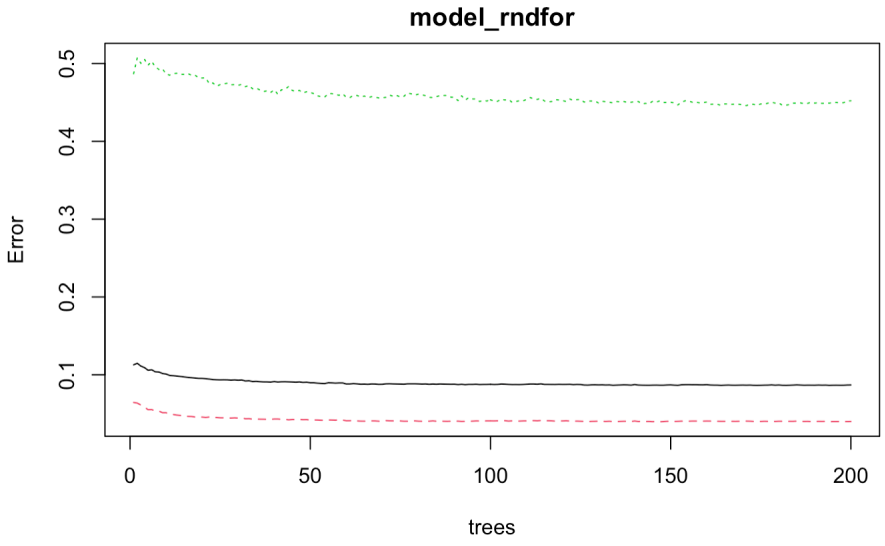
variable importance plot:

```
varImpPlot(model_rndfor)
```



By setting the importance argument on, we obtained the variable importance plot as above using varImpPlot() function and we can see that duration is highly significant in our data set.

```
plot(model_rndfor)%>%
legend("right", legend=c("OOB Error", "FPR", "FNR"),
       col=c("black", "red", "green"), lty=1:3, cex=0.8)
```



We can see that the False Negative Rate is higher compared to other error rate and False Positive Rate is lowest. The error rate starts dropping for at ntree~ 20. This says that our model is predicting 'Yes' cases more accurately than 'No' cases which is also shown confusion matrix

3.2.Data Modeling using K-NN Model

K-NN is a Non-parametric algorithm i.e it doesn't make any assumption about underlying data or its distribution. It is one of the simplest and widely used algorithm which depends on it's k value(Neighbours) and finds it's applications in many industries like finance industry, healthcare industry etc.

Making a copy of our data set for our k-NN classification.

```
data_FOR_knn <- dt.df
str(data_FOR_knn)
```

We must use numeric variables onlyBecause k-NN algorithm involves determining distances between datapoints. This is applicable only to independent variables. The target variable for k-NN classification should remain a factor variable.

Firstly , we will scale the data just in case our features are on different metrics. For example, if we had "duration" as a variable, it would be on a much larger scale than "age", which could be problematic given the k-NN relies on distances. Note that we are using the 'scale' function here, which means we are scaling to a z-score metric.

The variables "age", "duration", "campaign", "pdays", "previous", "emp.var.rate", "cons.price.idx", "cons.conf.idx", "euribor3m" and "nr.employed" are interger variables, that can be scalled.

```
data_FOR_knn[, c("age", "duration", "campaign", "pdays", "previous",
"emp.var.rate", "cons.price.idx", "cons.conf.idx",
"euribor3m","nr.employed")] <- scale(data_FOR_knn[, c("age", "duration",
"campaign", "pdays", "previous", "emp.var.rate", "cons.price.idx",
"cons.conf.idx", "euribor3m","nr.employed")])
head(data_FOR_knn)
str(data_FOR_knn)
```

```
> str(data_FOR_knn)
'data.frame':   41163 obs. of  21 variables:
 $ age           : num  1.53368 1.62966 -0.28981 -0.00189 1.53368 ...
 $ job           : chr  "housemaid" "services" "services" "admin." ...
 $ marital       : chr  "married" "married" "married" "married" ...
 $ education     : chr  "basic.4y" "high.school" "high.school" "basic.6y" ...
 $ default       : chr  "no" "unknown" "no" "no" ...
 $ housing       : chr  "no" "no" "yes" "no" ...
 $ loan          : chr  "no" "no" "no" "no" ...
 $ contact       : chr  "telephone" "telephone" "telephone" "telephone" ...
 $ month         : Ord.factor w/ 10 levels "mar"<"apr"<"may"<..: 3 3 3 3 3 3 3 3 3 3 ...
 $ day_of_week   : chr  "mon" "mon" "mon" "mon" ...
 $ duration      : num  0.0104 -0.4216 -0.1246 -0.4139 0.1878 ...
 $ campaign      : num  -0.566 -0.566 -0.566 -0.566 -0.566 ...
 $ pdays         : num  0.195 0.195 0.195 0.195 0.195 ...
 $ previous      : num  -0.349 -0.349 -0.349 -0.349 -0.349 ...
 $ poutcome      : chr  "nonexistent" "nonexistent" "nonexistent" "nonexistent" ...
 $ emp.var.rate  : num  0.648 0.648 0.648 0.648 0.648 ...
 $ cons.price.idx: num  0.722 0.722 0.722 0.722 0.722 ...
 $ cons.conf.idx : num  0.887 0.887 0.887 0.887 0.887 ...
 $ euribor3m     : num  0.712 0.712 0.712 0.712 0.712 ...
 $ nr.employed   : num  0.332 0.332 0.332 0.332 0.332 ...
 $ y             : num  0 0 0 0 0 0 0 0 0 0 ...
>
```

We can see that the variables "job", "marital", "education", "default", "housing", "loan", "contact", "month", "day_of_week" and "poutcome" are factor variables that have two or more levels.

```
data_FOR_knn$contact <- dummy.code(data_FOR_knn$contact)
```

Dummy code variables that have three or more levels.

```
default <- as.data.frame(dummy.code(data_FOR_knn$default))
housing <- as.data.frame(dummy.code(data_FOR_knn$housing))
day_of_week <- as.data.frame(dummy.code(data_FOR_knn$day_of_week))
education <- as.data.frame(dummy.code(data_FOR_knn$education))
loan <- as.data.frame(dummy.code(data_FOR_knn$loan))
month <- as.data.frame(dummy.code(data_FOR_knn$month))
job <- as.data.frame(dummy.code(data_FOR_knn$job))
marital <- as.data.frame(dummy.code(data_FOR_knn$marital))
poutcome <- as.data.frame(dummy.code(data_FOR_knn$poutcome))
```

Renamming "unknown" columns.

```
marital <- rename(marital, unknown_marital = unknown)
education <- rename(education , unknown_education  = unknown)
default <- rename(default , unknown_default  = unknown)
default <- rename(default , yes_default  = yes)
default <- rename(default , no_default  = no)
housing <- rename(housing , yes_housing  = yes)
housing <- rename(housing , no_housing  = no)
loan <- rename(loan , yes_loan  = yes)
loan <- rename(loan , no_loan  = no)
job <- rename(job, unknown_job = unknown)
housing <- rename(housing , unknown_housing  = unknown)
loan <- rename(loan , unknown_loan  = unknown)
```

Combinning new dummy variables with original data set.

```
data_FOR_knn <- cbind(data_FOR_knn, job, marital, education, default,
housing, loan, month, day_of_week,poutcome)
str(data_FOR_knn)
```

Removing original variables that had to be dummy coded.

```
data_FOR_knn <- data_FOR_knn %>% select(-one_of(c("job", "marital",
"education", "default", "housing", "loan", "month", "day_of_week",
"poutcome")))
head(data_FOR_knn)
```

We are now ready for k-NN classification. We partition 80% of the data into the training set and the remaining 20% into the test set.

Splitting the dataset into Test and Train:

```
set.seed(1234) # set the seed to make the partition reproducible
# 80% of the sample size
sample_size <- floor(0.8 * nrow(data_FOR_knn))
train_index <- sample(seq_len(nrow(data_FOR_knn)), size = sample_size)
# putting outcome in its own object
outcome_OF_knn <- data_FOR_knn %>% select(y)
# removing original variable from the data set
data_FOR_knn <- data_FOR_knn %>% select(-y)
# creating test and training sets that contain all of the predictors
knn_data_train <- data_FOR_knn[train_index, ]
knn_data_test <- data_FOR_knn[-train_index, ]
# Splitting outcome variable into training and test sets using the same
partition as above.
outcome_OF_knn_train <- outcome_OF_knn[train_index, ]
outcome_OF_knn_test <- outcome_OF_knn[-train_index, ]
```

We will run our k-NN classification on our data using 'class' package,. We have to decide on the number of neighbors (k).This is an iterative exercise as we need to keep changing the value of k to dtermine the optimum performance. In our case, we started with k=10 till k=20, and finally got an optimum performance at k=17.

```
model_knn <- knn(train = knn_data_train, test = knn_data_test, cl =
outcome_OF_knn_train, k=17)
```

Evaluating model:

```r
# putting "outcome_OF_knn_test" in a data frame
outcome_OF_knn_test <- data.frame(outcome_OF_knn_test)
# merging "model_knn" and "outcome_OF_knn_test"
knn_comparison_df <- data.frame(model_knn, outcome_OF_knn_test)
# specifying column names for "knn_comparison_df"
names(knn_comparison_df) <- c("KNN_Predicted_y", "KNN_Observed_y")
knn_comparison_df$KNN_Predicted_y <-
as.factor(knn_comparison_df$KNN_Predicted_y)
knn_comparison_df$KNN_Observed_y <-
as.factor(knn_comparison_df$KNN_Observed_y)
# inspecting "knn_comparison_df"
head(knn_comparison_df)
```

Finally we will compare our predicted values of deposit to our actual values.
The confusion matrix will give an indication of how well our model predicted the actual values.
The confusion matrix output also shows overall model statistics and statistics by class

```r
confusionMatrix(knn_comparison_df$KNN_Observed_y,knn_comparison_df$KNN_Predicted_y)
```

```
> confusionMatrix(knn_comparison_df$KNN_Obser
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 7045  214
         1  591  383

               Accuracy : 0.9022
                 95% CI : (0.8956, 0.9086)
    No Information Rate : 0.9275
    P-Value [Acc > NIR] : 1

                  Kappa : 0.437

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.9226
            Specificity : 0.6415
         Pos Pred Value : 0.9705
         Neg Pred Value : 0.3932
             Prevalence : 0.9275
         Detection Rate : 0.8557
   Detection Prevalence : 0.8817
      Balanced Accuracy : 0.7821

       'Positive' Class : 0
```

1) The K-nn test data consisted of 8233 observations.
2) 7045 cases have been accurately predicted (TN->True Negatives) as negative class (0) which constitutes 87%.
3) Also, 383 out of 8233 observations were accurately predicted (TP-> True Positives) as positive class (1) which constitutes 4%.

4) Thus a total of 383 out of 8233 predictions where TP i.e, True Positive in nature.

5) There were 591 cases of False Positives (FP) meaning 591 cases out of 8238 were actually negative but got predicted as positive.

6) There were 214 cases of False Negatives(FN) meaning 214 cases were actually positive in nature but got predicted as negative.

7) Accuracy of the model is the correctly classified positive and negative cases divided by all ther cases.The total accuracy of the model is 90.22%, which means the model

prediction is very accurate.

4.Conclusion:

Comparing Model:
Both the algorithms namely Random Forest and K Nearest Neignbor are generaing high accuracy when trained with the bank marketing dataset.

The parameter comparision for both the model is:

Parameter	Random Forest	K-nn Model
Accuracy	91.96%	90.22%
Sensitivity	95.06%	92.26%
Specificity	63.73%	64.15%
Pos Pred Value	95.97%	97.05%
Neg Pred Value	58.69%	39.32%

At an overall level, the performance of both the model is similar, however Random Forest model has a better prediction performance for Negative classes and hence, we can go forward with selecting Random Forest as a better model for our objective.

Analysis Summary:
The key insights derived from the overall analysis are:
- With respect to Marital Status, there is not an observed large difference in the proportion of people subscribed to term deposits and people without term deposits.
- Customers who sign up for bank deposits, proportionally, have achieved a higher level of education, than those who didn't sign up.
- The months of April, September, and October is the time when a higher proportion of people subscribed for term deposits.
- There are higher proportions for customers signing up for the term deposits who have the jobs of admin, retired, and students.
- People who aren't in default are a higher proportion of people who have subscribed for bank deposits.
- Higher proportion of people who have subscribed for bank deposit are home owners versus ones that don't own their own houses.
- The proportion of people who have subscribed and not subscribed to a term deposit is the same for categories of the Loan.
- Customers who have cell phones, and therefore a more direct way of communicating, signed up for term deposits more than those who only had a landline telephone.
- Campaigns that were performed midweek, on Tuesdays, Wednesdays, and Thursdays had a slightly higher proportion of people who subscribed for bank deposit.

- Potential customers who successfully connected and responded in previous campaigns had a higher proportion of signing up for the term deposit.
- The longer the phone conversation the greater the conversion rate is for the potential customer to sign up for the term deposit. There are higher median and quartile ranges.
- The age range for successful conversion has a slightly lower median, but higher quartile ranges.
- Subscribing to term deposit has a high positive correlation with duration and if the customer was involved and connected in a previous campaign, while there's negative correlation with Nr.employed (number of employees), pdays (number of days from last contact), Euribor3m (Euribor 3 month rate) and emp.var.rate (employee variation rate).

Proposed Marketing Strategy

As mentioned in the introduction, the aim of the project is to help devise a marketing strategy for
the bank based on the behavioral data collected and analyzed to increase the investments in term deposit.

The marketing strategy should be alluring the following customers
1) those who are existing account holders with the bank and have higher education.
2) those who are either employed in admin
related jobs, or are students, or retired people.
3)those who are easily accessible via mobile number i.e. answer the call. And the prerequisite is to
have a telecaller with good communication skills and capable of establishing a warm relationship with the customer so that they feel secured while investing in
a term deposit with the bank.
4) those who were part of the previous campaigns but did/ didn't invested should be contacted by the bank again as buiding relationship results in a higher number of customers.
5) Lastly but most importantly, the marketing campaigns should
be launched in the last third of the calendar year (financial year)when people are thinking of saving for the future and preparing for year-end taxes.

Future Work:
Random Forest model does not scale very well for time-series data and might need to be constantly updated in Production or trained with some Random Data that lies outside our range of Training set
Answering questions like “What would the number of term deposit for next Year?"or "How many new customers would invest in term deposit in the next three months?” becomes really difficult when using Random Forests.
And these setbacks have to be eliminated by working upon the existing models

****