Student number: B1102372

SOURCE CODE

```
# I will install some packages we will need for this project
install.packages('tidyverse')
install.packages('dslabs')
install.packages("lubridate")
install.packages('corrplot')
install.packages("gridExtra")
install.packages("GGally")
install.packages("knitr")
install.packages('naniar')
install.packages("caret")
install.packages("ggthemes")
install.packages("tidyr")
install.packages("cluster")
install.packages("ggplot2")
install.packages("corrgram")
install.packages("cowplot")
install.packages("caret")
install.packages("rpart.plot")
install.packages("e1071")

library(readr)
library(dplyr)
library(naniar)
library(lubridate)
library(caret)
library(corrplot)
library(tidyr)
library(cluster)
library(ggplot2)
library(corrgram)
library(ggthemes)
library(cowplot)
library(corrplot)
library(rpart.plot)

#importing our csv file into the studio

customers_data <- read_delim("C:/Users/B1102372/OneDrive - Teesside University
/Documents/CIS4047/MY ICA/marketing_campaign.csv",
                             delim = "\t", escape_double = FALSE,
                             trim_ws = TRUE)
```

```r
View(customers_data)



# checking for missing value
n_miss(customers_data)

# checking the variables with the missing value
miss_var_summary(customers_data)

# removing the missing values
customers_data = na.omit(customers_data)
dim(customers_data)


# Getting Age from Year Birth
customers_data = customers_data %>%
  mutate(Age = 2021 - Year_Birth)

# visualizing the Age column
customers_data %>%
  ggplot(aes(1:length(ID), Age)) +
  geom_point() +
  theme_bw() +
  labs(x = "ID")

#setting the age to maximum of 95 and in the process removing the three outlie
rs in Variable Age
customers_data = customers_data %>%
  filter(Age < 95)

# visualizing the income column
customers_data %>%
  ggplot(aes(1:length(ID), Income)) +
  geom_point() +
  theme_bw() +
  labs(x = "ID")

#Removing the outlier from the Income variable by putting a cap of 300,000
customers_data = customers_data %>%
  filter(Income < 300000)


# modifying the Recency column to Active(0-30days) and Not active(31-100)
customers_data = customers_data %>%
  mutate(Recency = ifelse(Recency <= 30,
                          "Active",
                          "Not Active") )
```

```r
# converting the marital status into two categories(single and partner).
customers_data = customers_data %>%
  mutate(Marital_Status = replace(Marital_Status, Marital_Status =="Divorced"
| Marital_Status == "Widow" | Marital_Status == "Alone" | Marital_Status == "A
bsurd" | Marital_Status == "YOLO", "Single"))

customers_data = customers_data %>%
  mutate(Marital_Status = replace(Marital_Status, Marital_Status == "Together"
 | Marital_Status == "Married", "Partner"))


# converting Education into two categories too
customers_data = customers_data %>%
  mutate(Education = replace(Education, Education == "2n Cycle" | Education ==
 "Basic", "Non-Graduate"))

customers_data = customers_data %>%
  mutate(Education = replace(Education, Education == "PhD" | Education == "Gra
duation" | Education == "Master", "Graduate"))

#creating a total spending for each household by adding the number of products
bought on wines, fruits,meats,fish,sweet and Gold together
customers_data = customers_data %>%
  mutate(Total_spending = MntWines+MntFruits+MntMeatProducts+MntFishProducts+M
ntSweetProducts+MntGoldProds)


# getting the number of children in each household and convert into two
categories
customers_data = customers_data %>%
  mutate(Children_in_household = Kidhome + Teenhome)



# removing some columns that are not needed for our project
customers_data = customers_data %>%
  select(-ID, -Year_Birth, -Teenhome, -Dt_Customer, -Kidhome, -
Z_CostContact, -Z_Revenue)



#  doing data visualization and checking how data is distributed
mar_plot = ggplot(data = customers_data, aes(Marital_Status, fill = Marital_St
atus))
mar_plot + geom_histogram(stat = "count")

Edu_plot =  ggplot(data = customers_data, aes(Education, fill = Education))
Edu_plot + geom_histogram(stat = "count")
```

```r
# products bought distribution visualization
customers_data %>%
  pivot_longer(
    cols = starts_with("Mnt")
  ) %>%
  select(name, value) %>%
  ggplot(aes(value, fill = name)) +
  geom_histogram() +
  facet_wrap(vars(name), scales = "free") +
  labs(x = "",
       y = "",
       subtitle = "Products Purchased distribution")


# Response, our target column distribution visualization
Resp_plot =  ggplot(data = customers_data, aes(Response, fill = Response))
Resp_plot + geom_histogram(stat = "count")




# some attributes Vs response visualization
plot_grid(
  ggplot(customers_data)+geom_bar(aes(x=Education,y=Response),stat = "identity
"),
  ggplot(customers_data)+geom_bar(aes(x=Marital_Status,y=Response),stat = "ide
ntity"),
  ggplot(customers_data)+geom_bar(aes(x=Children_in_household,y=Response),stat
 = "identity"),
  ggplot(customers_data)+geom_bar(aes(x=Recency,y=Response),stat = "identity")
,
  ggplot(customers_data)+geom_bar(aes(x=Response,y=Complain),stat = "identity"
)
)


# converting the factor columns, we want to use for our training to numeric.
# starting with the Education column,then the Marital status, Children in hous
ehold, Recency, Total spending.
customers_data = customers_data %>%
  mutate(Education = recode(Education,
                            "Graduate" = 1,
                            "Non-Graduate" = 0))

customers_data = customers_data %>%
  mutate(Marital_Status = recode(Marital_Status,
                                 "Single" = 1,
                                 "Partner" = 2))
```

```r
customers_data = customers_data %>%
  mutate(Recency = recode(Recency,
                          "Not Active" = 0,
                          "Active" = 1))



# checking the data type of all attributes
sapply(customers_data, class)



# doing the correlation matrix on our cleaned dataset

customers_data_cor = cor(customers_data[,1:25])
corrplot(customers_data_cor, method = "circle")


#Now we split the dataset into two set, training set and testing set
#The target variable is Response, and we will allocate 70% for training and 30
% testing
intrain = createDataPartition(y = customers_data$Response, p=0.7, list = FALSE
)
training = customers_data[intrain,]
testing = customers_data[-intrain,]

# Viewing the columns and rows we have for training and testing
dim(training)
dim(testing)

summary(customers_data)


# training the data for Svm

training[["Response"]] = factor(training[["Response"]])

trctrl = trainControl(method = "repeatedcv", number = 10, repeats = 3)


svm_Linear = train(Response ~., data = training, method = "svmLinear",trContro
l=trctrl,preProcess = c("center", "scale"),tuneLength = 10)

# testing the data
svm_Linear
svm_test = predict(svm_Linear, newdata = testing)
svm_test
```

```r
# the confusion matrix for Svm model
svm_cm = print(confusionMatrix(table(svm_test, testing$Response)))


# Tuning
grid = expand.grid(C = c(0,0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.7
5, 2,5))
svm_Linear_Grid <- train(Response ~., data = training, method = "svmLinear",tr
Control=trctrl,preProcess = c("center", "scale"),tuneGrid = grid,tuneLength= 1
)

svm_Linear_Grid
plot(svm_Linear_Grid)

# testing the tuned data
svm_test_grid = predict(svm_Linear_Grid, newdata = testing)
svm_test_grid

# confusion matrix of the tuned Svm
svm_tune_cm = print(confusionMatrix(table(svm_test_grid, testing$Response)))




# Decision tree model
customers_data$Response = as.factor(customers_data$Response)
set.seed(1)
inTrain = createDataPartition(customers_data$Response, p = .7)[[1]]
# Assign the 70% of observations to training data
training <- customers_data[inTrain,]
# Assign the remaining 30% of observations to testing data
testing <- customers_data[-inTrain,]
# Setting_the_seed (in order all results to be fully reproducible) and apply a
prediction Model with all variables
set.seed(2)
model.all <- train(Response ~ ., method="rpart", data = training)
# Applying the prediction
prediction <- predict(model.all, newdata= testing)
#Check the accuracy of the prediction model by printing the confusion matrix
dt_cm = print(confusionMatrix(prediction, testing$Response), digits=4)


tree = rpart(Response~., data = customers_data, cp=.05)
#Plotting the Classification Tree
rpart.plot(tree, box.palette = "RdBu", shadow.col = "gray", nn = TRUE, main =
"Classification Tree of Marketing Campaingn")
```

```r
# Random forest model
rf.fit <- train(Response~., data = customers_data, method = "rf", trainControl
 = trainControl)

rf.predict <- predict(rf.fit, newdata = testing)

rf_cm = print(confusionMatrix(rf.predict, testing$Response), digits=4)



# comparing the models getting the data from confusion matrix
svm_data = c(svm_cm$byClass)
svm_tune_data = c(svm_tune_cm$byClass)
dt_data = c(dt_cm$byClass)
rf_data = c(rf_cm$byClass)


# new data frame for the models
modeldf = data.frame( SVM=c(svm_data), SVMs=c(svm_tune_data), DT=c(dt_data), R
F=c(rf_data))


library(RColorBrewer)
#plotting performance for each model
barplot(t(as.matrix(modeldf)), beside=TRUE,col=brewer.pal(4, 'Spectral') ,
        legend=c('SVM', 'SVMs', 'DT', 'RF'), args.legend = list(x="topright"),
 ylim = c(0,1.5),
        names.arg = rownames(modeldf), main = 'Predicting Performance for the
models', las = 2)
```

SCREENSHOTS

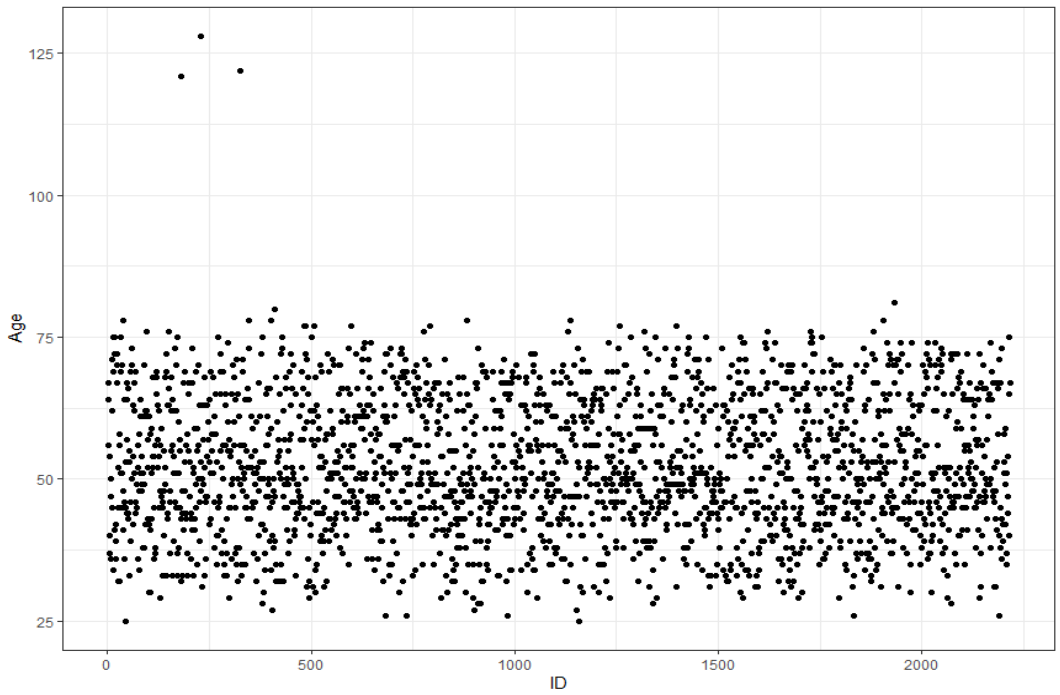

Figure 1: Age Distribution

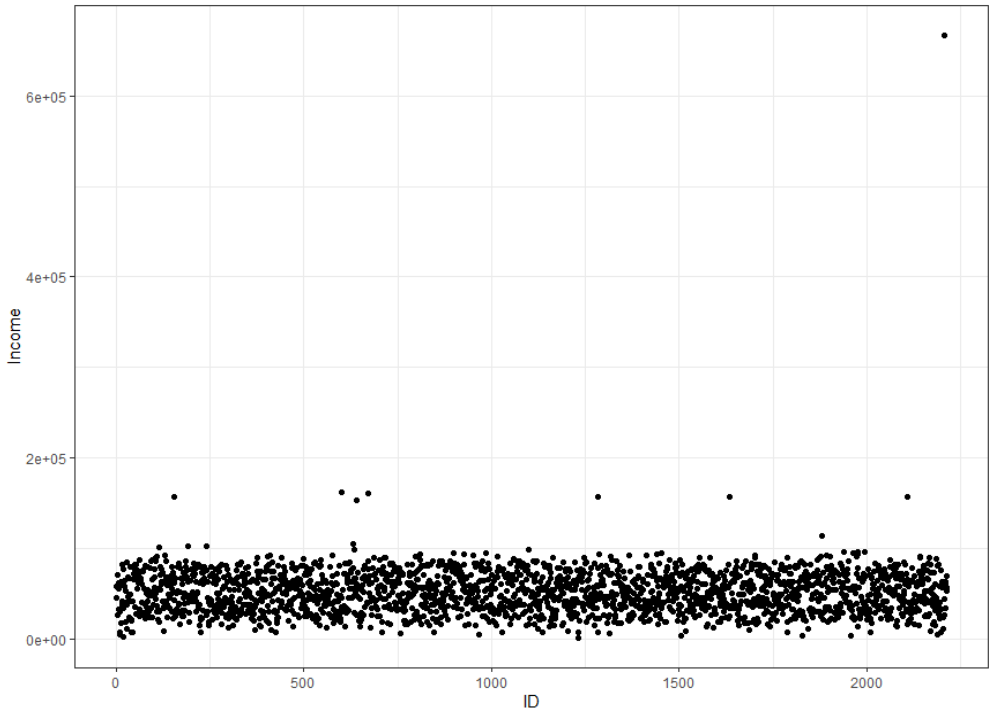

Figure 2: Income Distribution

customers.R × customers_data ×

Filter

	Education	Marital_Status	Income	Recency	MntWines	MntFruits	MntMeatProducts	MntFishProducts	MntSweetProducts	MntGoldProds	Nu
1	Graduate	Single	58138	Not Active	635	88	546	172	88	88	
2	Graduate	Single	46344	Not Active	11	1	6	2	1	6	
3	Graduate	Partner	71613	Active	426	49	127	111	21	42	
4	Graduate	Partner	26646	Active	11	4	20	10	3	5	
5	Graduate	Partner	58293	Not Active	173	43	118	46	27	15	
6	Graduate	Partner	62513	Active	520	42	98	0	42	14	
7	Graduate	Single	55635	Not Active	235	65	164	50	49	27	
8	Graduate	Partner	33454	Not Active	76	10	56	3	1	23	
9	Graduate	Partner	30351	Active	14	0	24	3	3	2	
10	Graduate	Partner	5648	Not Active	28	0	6	1	1	13	
11	Non-Graduate	Partner	7500	Not Active	6	16	11	11	1	16	
12	Graduate	Single	63033	Not Active	194	61	480	225	112	30	
13	Graduate	Single	59354	Not Active	233	2	53	3	5	14	
14	Graduate	Partner	17323	Not Active	3	14	17	6	1	5	
15	Graduate	Single	82800	Active	1006	22	115	59	68	45	
16	Graduate	Partner	41850	Not Active	53	5	19	2	13	4	
17	Graduate	Partner	37760	Active	84	5	38	150	12	28	
18	Graduate	Partner	76995	Not Active	1012	80	498	0	16	176	
19	Non-Graduate	Single	33812	Not Active	4	17	19	30	24	39	
20	Graduate	Partner	37040	Not Active	86	2	73	69	38	48	
21	Graduate	Partner	2447	Not Active	1	1	1725	1	1	1	

Figure 3:Dataset showing the new modified Education, marital status and Recency column

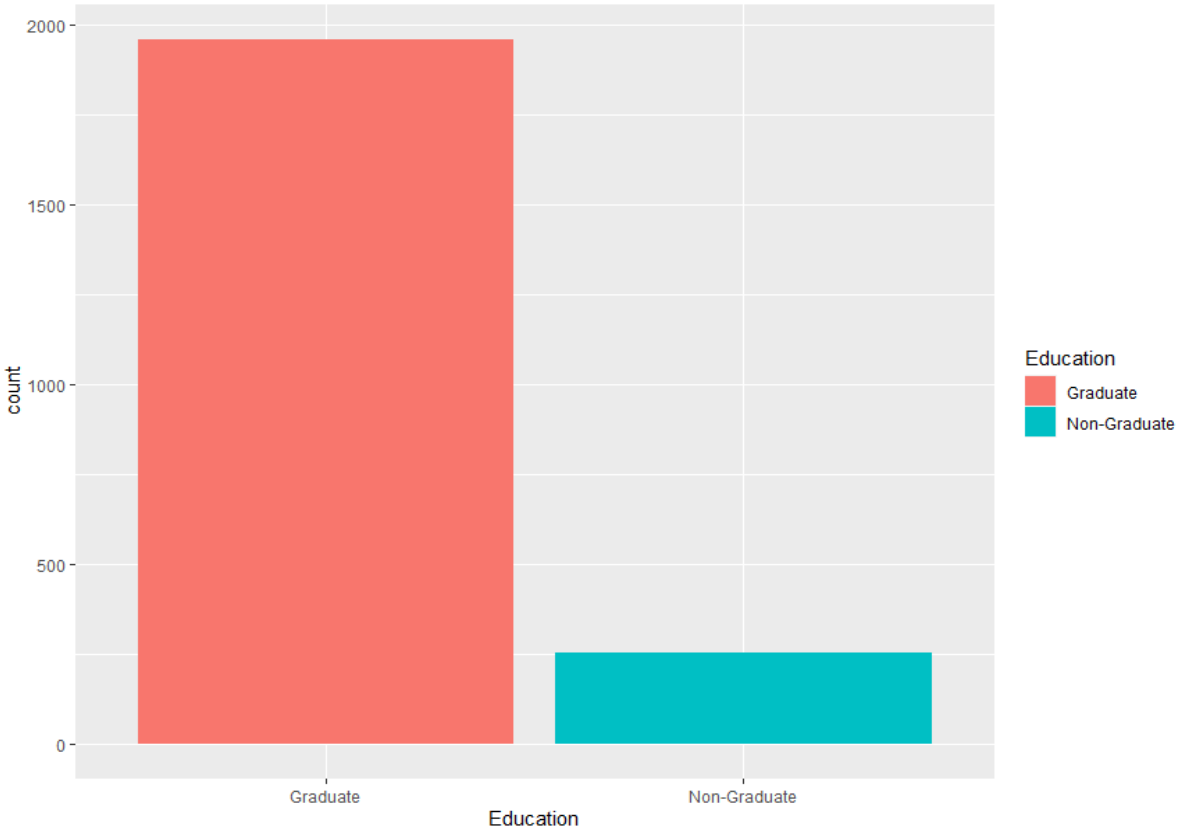

Figure 4: Education level distribution

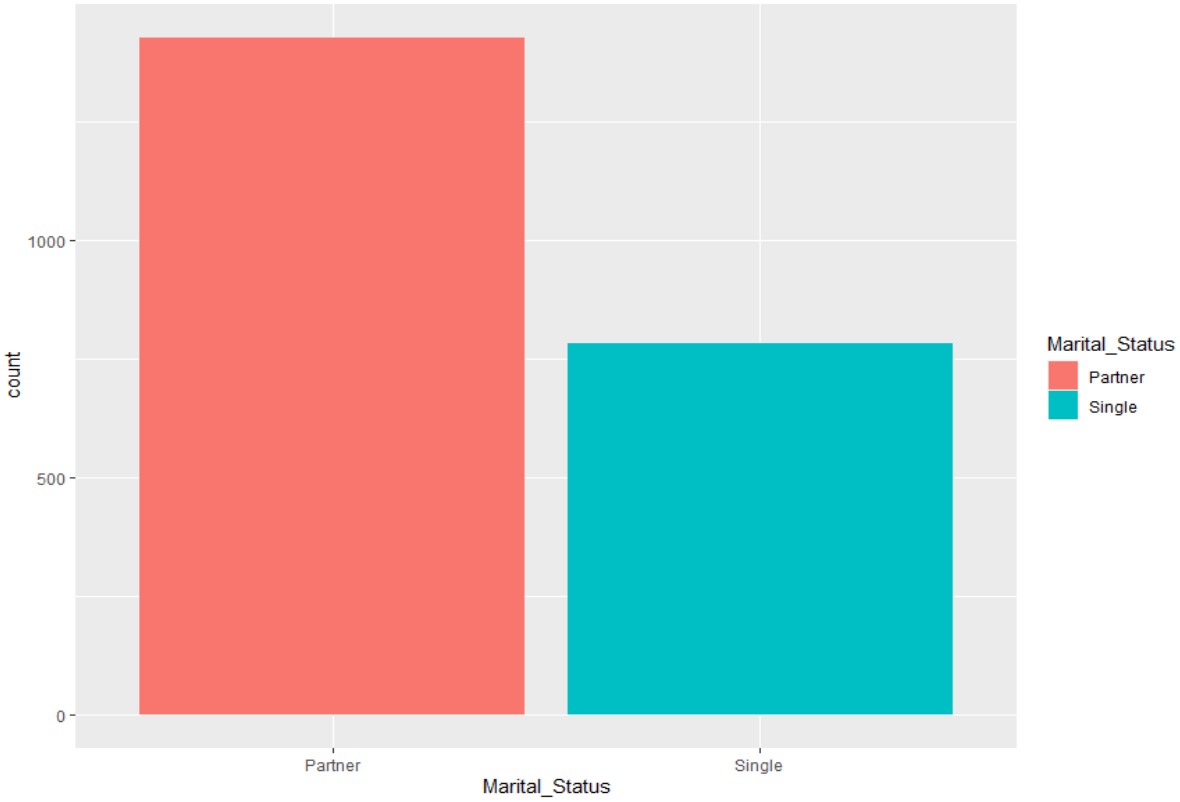

Figure 5: Marital status distribution

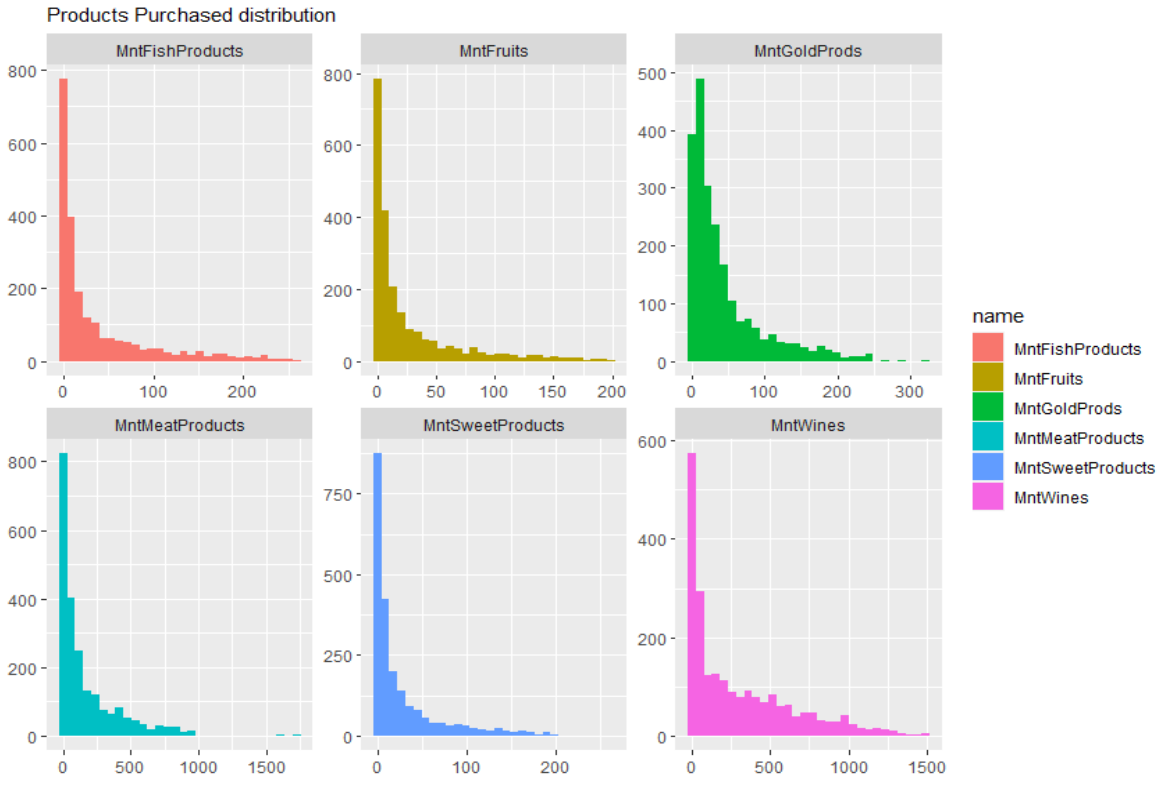

Figure 6: products bought distribution

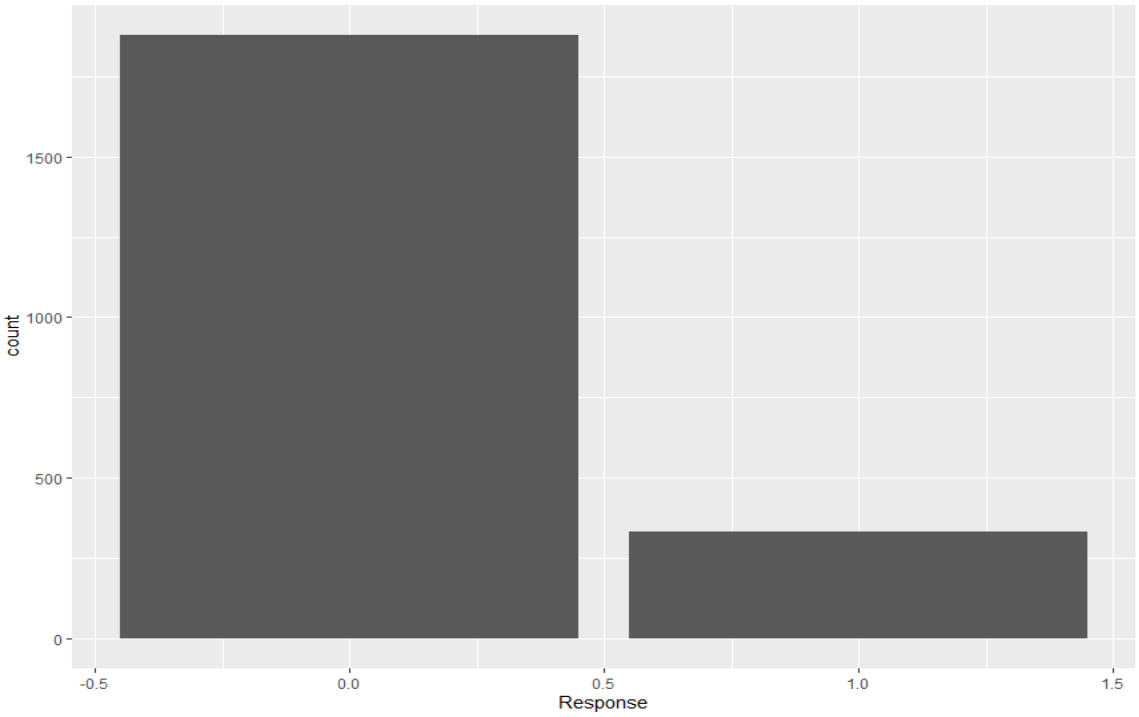

Figure 7: Response distribution

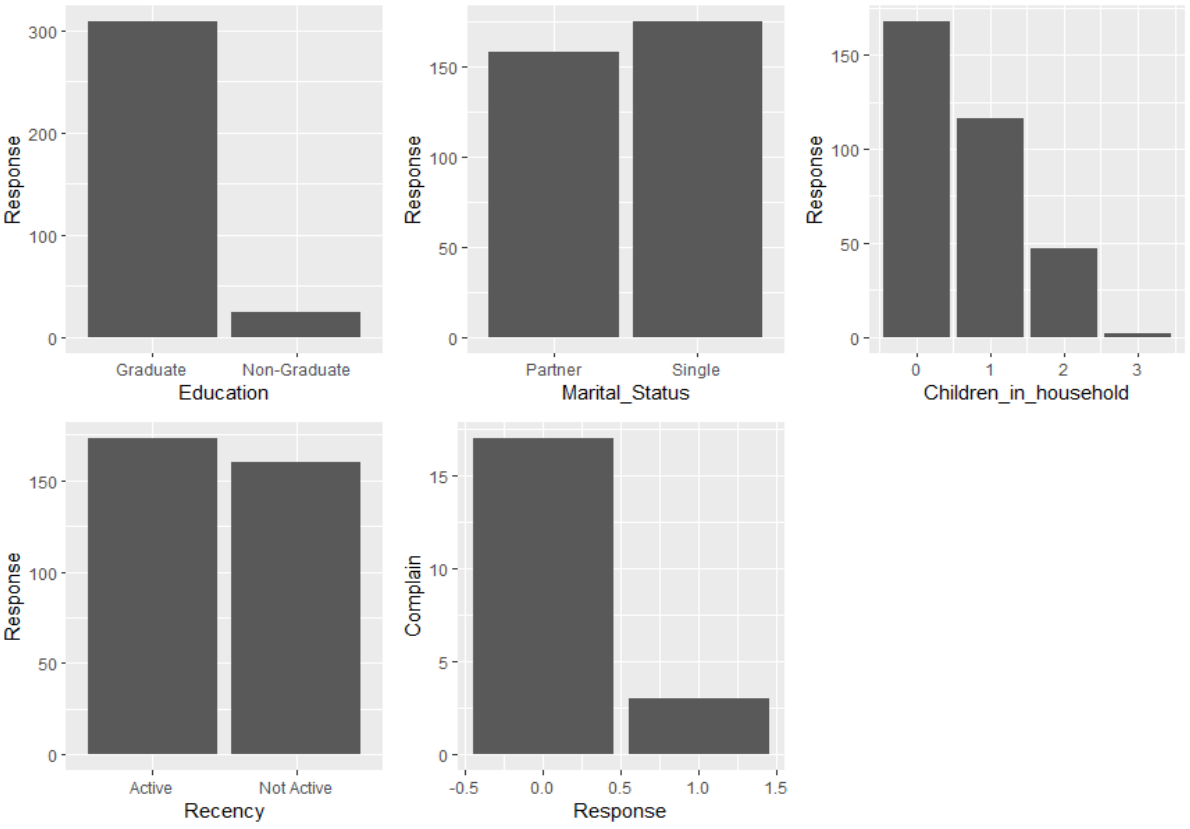

Figure 8: Some columns Vs response distribution

```
> sapply(customers_data, class)
        Education      Marital_Status              Income             Recency            MntWines 
        "numeric"           "numeric"           "numeric"           "numeric"           "numeric" 
        MntFruits     MntMeatProducts      MntFishProducts     MntSweetProducts      MntGoldProds 
        "numeric"           "numeric"           "numeric"           "numeric"           "numeric" 
NumDealsPurchases     NumWebPurchases  NumCatalogPurchases     NumStorePurchases   NumWebVisitsMonth 
        "numeric"           "numeric"           "numeric"           "numeric"           "numeric" 
     AcceptedCmp3        AcceptedCmp4        AcceptedCmp5        AcceptedCmp1        AcceptedCmp2 
        "numeric"           "numeric"           "numeric"           "numeric"           "numeric" 
         Complain            Response                 Age     Total_spending Children_in_household 
        "numeric"           "numeric"           "numeric"           "numeric"           "numeric" 
```

Figure 9: Classes of dataset attributes in numeric

```
> summary(customers_data)
   Education      Marital_Status       Income          Recency          MntWines         MntFruits      MntMeatProducts 
 Min.   :0.0000   Min.   :1.000   Min.   :  1730   Min.   :0.0000   Min.   :   0.0   Min.   :  0.00   Min.   :   0.0  
 1st Qu.:1.0000   1st Qu.:1.000   1st Qu.: 35234   1st Qu.:0.0000   1st Qu.:  24.0   1st Qu.:  2.00   1st Qu.:  16.0  
 Median :1.0000   Median :2.000   Median : 51371   Median :0.0000   Median : 175.5   Median :  8.00   Median :  68.0  
 Mean   :0.8861   Mean   :1.646   Mean   : 51959   Mean   :0.3237   Mean   : 305.3   Mean   : 26.33   Mean   : 167.0  
 3rd Qu.:1.0000   3rd Qu.:2.000   3rd Qu.: 68487   3rd Qu.:1.0000   3rd Qu.: 505.0   3rd Qu.: 33.00   3rd Qu.: 232.2  
 Max.   :1.0000   Max.   :2.000   Max.   :162397   Max.   :1.0000   Max.   :1493.0   Max.   :199.00   Max.   :1725.0  
 MntFishProducts  MntSweetProducts  MntGoldProds    NumDealsPurchases NumWebPurchases  NumCatalogPurchases
 Min.   :  0.00   Min.   :  0.00   Min.   :  0.00   Min.   : 0.000    Min.   : 0.000   Min.   : 0.000     
 1st Qu.:  3.00   1st Qu.:  1.00   1st Qu.:  9.00   1st Qu.: 1.000    1st Qu.: 2.000   1st Qu.: 0.000     
 Median : 12.00   Median :  8.00   Median : 24.50   Median : 2.000    Median : 4.000   Median : 2.000     
 Mean   : 37.65   Mean   : 27.05   Mean   : 43.93   Mean   : 2.325    Mean   : 4.088   Mean   : 2.672     
 3rd Qu.: 50.00   3rd Qu.: 33.00   3rd Qu.: 56.00   3rd Qu.: 3.000    3rd Qu.: 6.000   3rd Qu.: 4.000     
 Max.   :259.00   Max.   :262.00   Max.   :321.00   Max.   :15.000    Max.   :27.000   Max.   :28.000     
 NumStorePurchases NumWebVisitsMonth  AcceptedCmp3      AcceptedCmp4      AcceptedCmp5      AcceptedCmp1      
 Min.   : 0.000    Min.   : 0.000    Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.0000    
 1st Qu.: 3.000    1st Qu.: 3.000    1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.0000    
 Median : 5.000    Median : 6.000    Median :0.00000   Median :0.00000   Median :0.00000   Median :0.0000    
 Mean   : 5.807    Mean   : 5.321    Mean   :0.07369   Mean   :0.07414   Mean   :0.07278   Mean   :0.0642    
 3rd Qu.: 8.000    3rd Qu.: 7.000    3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.0000    
 Max.   :13.000    Max.   :20.000    Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.0000    
  AcceptedCmp2        Complain           Response           Age         Total_spending   Children_in_household
 Min.   :0.00000   Min.   :0.000000   Min.   :0.0000   Min.   :25.00   Min.   :   5.0   Min.   :0.0000       
 1st Qu.:0.00000   1st Qu.:0.000000   1st Qu.:0.0000   1st Qu.:44.00   1st Qu.:  69.0   1st Qu.:0.0000       
 Median :0.00000   Median :0.000000   Median :0.0000   Median :51.00   Median : 397.0   Median :1.0000       
 Mean   :0.01356   Mean   :0.009042   Mean   :0.1505   Mean   :52.09   Mean   : 607.3   Mean   :0.9476       
```

Figure 10: Summary of the cleaned dataset

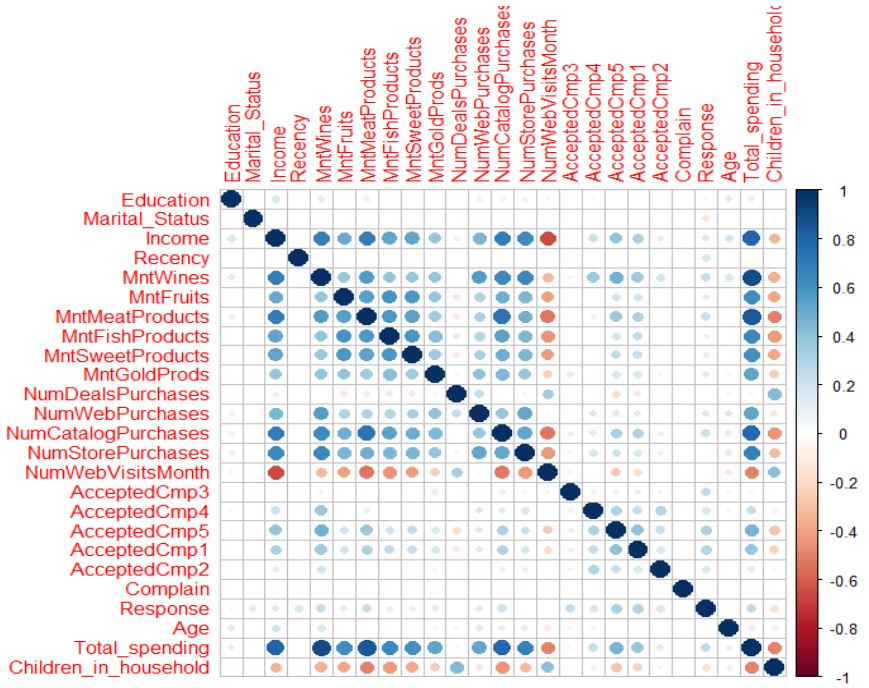

Figure 11: Correlation matrix

```
svm_test   0   1
       0 544  79
       1  11  29

               Accuracy : 0.8643
                 95% CI : (0.8358, 0.8894)
    No Information Rate : 0.8371
    P-Value [Acc > NIR] : 0.03064

                  Kappa : 0.3332

 Mcnemar's Test P-Value : 1.636e-12

            Sensitivity : 0.9802
            Specificity : 0.2685
         Pos Pred Value : 0.8732
         Neg Pred Value : 0.7250
             Prevalence : 0.8371
         Detection Rate : 0.8205
   Detection Prevalence : 0.9397
      Balanced Accuracy : 0.6243

       'Positive' Class : 0
```

Figure 12: Svm confusion matrix

```
svm_test_grid   0   1
             0 544  79
             1  11  29

                Accuracy : 0.8643
                  95% CI : (0.8358, 0.8894)
    No Information Rate : 0.8371
    P-Value [Acc > NIR] : 0.03064

                   Kappa : 0.3332

 Mcnemar's Test P-Value : 1.636e-12

            Sensitivity : 0.9802
            Specificity : 0.2685
         Pos Pred Value : 0.8732
         Neg Pred Value : 0.7250
             Prevalence : 0.8371
         Detection Rate : 0.8205
   Detection Prevalence : 0.9397
      Balanced Accuracy : 0.6243

       'Positive' Class : 0
```

Figure 13: Tuned Svm confusion matrix

```
          Reference
Prediction   0   1
         0 552  80
         1  11  19

               Accuracy : 0.8625
                 95% CI : (0.8339, 0.8879)
    No Information Rate : 0.8505
    P-Value [Acc > NIR] : 0.2081

                  Kappa : 0.2418

 Mcnemar's Test P-Value : 1.016e-12

            Sensitivity : 0.9805
            Specificity : 0.1919
         Pos Pred Value : 0.8734
         Neg Pred Value : 0.6333
             Prevalence : 0.8505
         Detection Rate : 0.8338
   Detection Prevalence : 0.9547
      Balanced Accuracy : 0.5862

       'Positive' Class : 0
```

Figure 14: Decision tree confusion matrix

```
Prediction   0   1
         0 563  15
         1   0  84

               Accuracy : 0.9773
                 95% CI : (0.9629, 0.9873)
    No Information Rate : 0.8505
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.905

 Mcnemar's Test P-Value : 0.0003006

            Sensitivity : 1.0000
            Specificity : 0.8485
         Pos Pred Value : 0.9740
         Neg Pred Value : 1.0000
             Prevalence : 0.8505
         Detection Rate : 0.8505
   Detection Prevalence : 0.8731
      Balanced Accuracy : 0.9242

       'Positive' Class : 0
```

Figure 15: Random forest confusion matrix

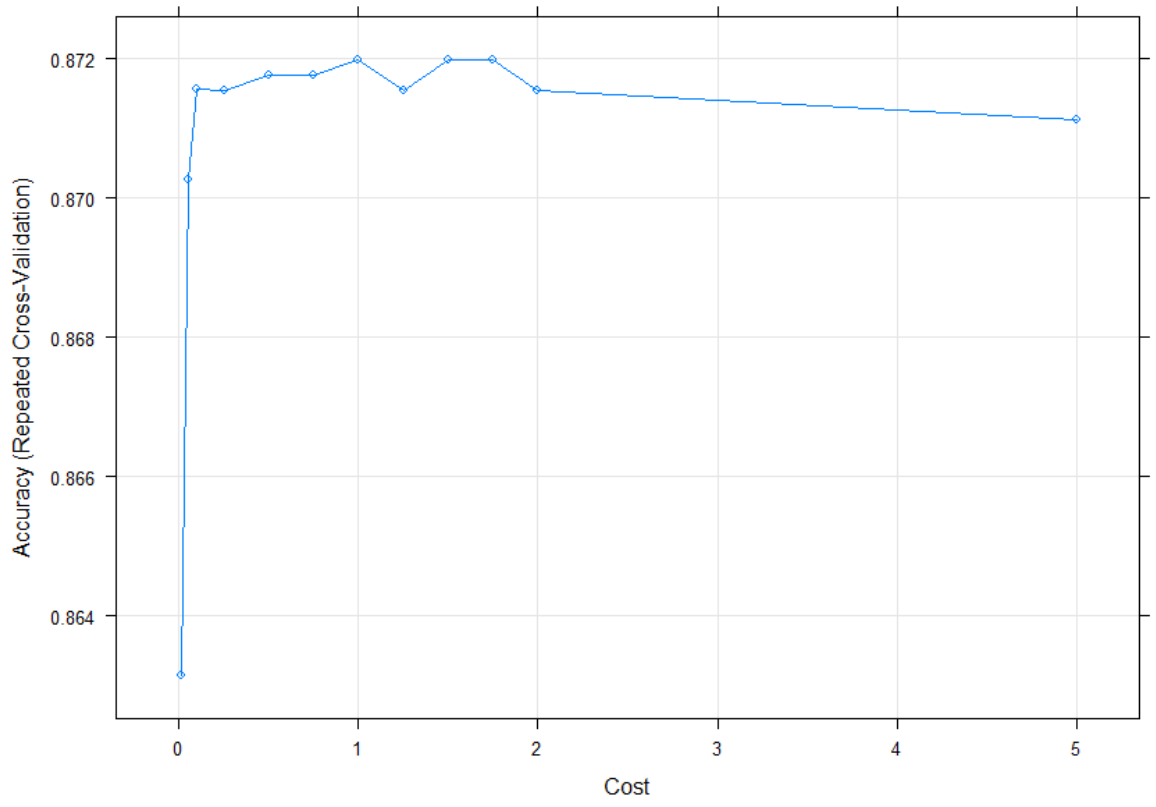

Figure 16: Svm Accuracy plot

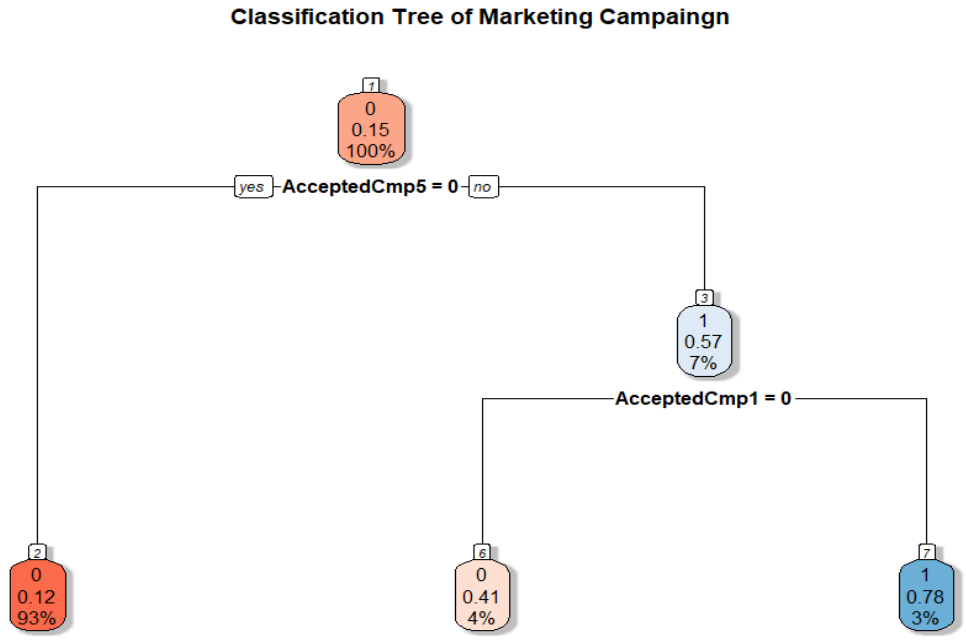

Figure 17: Classification tree

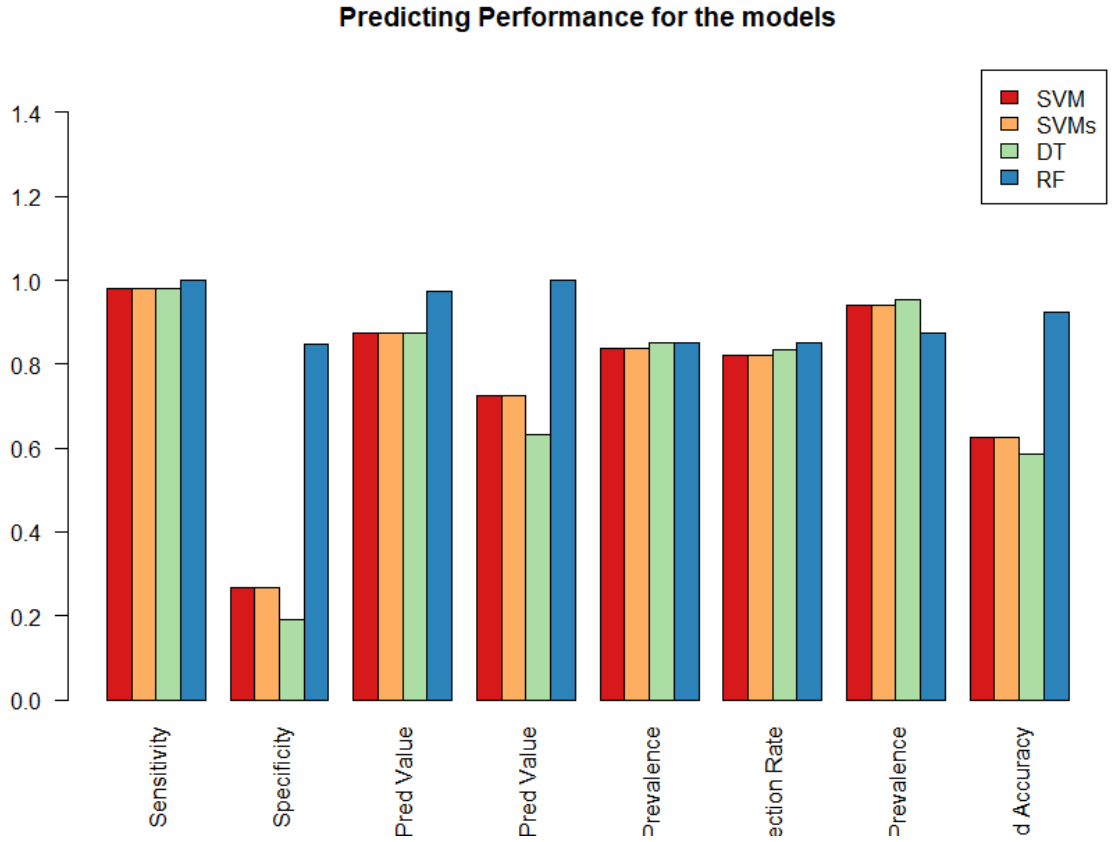

Figure 18: Comparing the models performance