```
---
title: "Predictive Project"
author: "Rohan_Srivastava, Noel Abraham, Ramesh, Vaibhav, Saurav"
date: '2022-05-12'
output:
  word_document: default
  pdf_document: default
---

## Importing packages
```{r}
library(tidyverse)
library(car)
library(sjPlot)
library(sjmisc)
library(sjlabelled)
library(MASS)
library(caTools)
library(ROCR)
library(precrec)
library(pROC)
library(randomForest)
library(ggplot2)
library(lessR)
library(dplyr)
library(psych)
```

Importing the data
df <- read.csv("C:/Users/Rohan.000/Desktop/Predictive/WA_Fn-UseC_-Telco-
Customer-Churn.csv")
head(df)
nrow(df)
Drop customer ID
df <- df[, !(colnames(df) %in% c("customerID"))]
```

count and remove null values
names(which(colSums(is.na(df))>0))
sum(is.na(df$TotalCharges))
df <- na.omit(df)
Count number of unique values in each column
ulst <- lapply(df, unique)
k <- lengths(ulst)
k
Find unique values for each variable
unique(df$gender)
unique(df$SeniorCitizen)
unique(df$Partner)
unique(df$Dependents)
unique(df$PhoneService)
unique(df$MultipleLines)
unique(df$PhoneService)
unique(df$InternetService)
```
```

```
unique(df$OnlineSecurity)
unique(df$OnlineBackup)
unique(df$DeviceProtection)
unique(df$OnlineBackup)
unique(df$TechSupport)
unique(df$StreamingTV)
unique(df$StreamingMovies)
unique(df$Contract)
unique(df$PaperlessBilling)
unique(df$PaymentMethod)
unique(df$Churn)
## Summary of Data
summary(df)
## Convert all No internet service to No
df <- data.frame(lapply(df, function(x) {
  gsub("No internet service", "No", x)}))

## Split data into two categories: categorical and continuous
int <- c("tenure", "MonthlyCharges", "TotalCharges")
df[int] <- sapply(df[int], as.numeric)
df_int <- df[,c("tenure", "MonthlyCharges", "TotalCharges")]
df_int <- data.frame(scale(df_int))                              ## Scaling
the numeric data

df_cat <- df[,-c(5,7,8,15,17,18,19)]
df_dummy<- data.frame(sapply(df_cat,function(x)
data.frame(model.matrix(~x-1,data =df_cat))[,-1]))
```

```
#create dummy variables for for than 2 categories
df_cat2 <- df[,c(7,8,15,17)]

df_cat2$MultipleLines[df_cat2$MultipleLines == "Yes"] <- 1      # Replace
"Yes" by 1
df_cat2$MultipleLines[df_cat2$MultipleLines == "No"] <- 0      # Replace
"No" by 0
df_cat2$MultipleLines[df_cat2$MultipleLines == "No phone service"] <- 0
# Replace "No phone service" by 0
df_cat2$MultipleLines <- as.factor(df_cat2$MultipleLines)

df_cat2$InternetService[df_cat2$InternetService == "DSL"] <- 1      #
Replace "DSL" by 1
df_cat2$InternetService[df_cat2$InternetService == "No"] <- 0      #
Replace "No" by 0
df_cat2$InternetService[df_cat2$InternetService == "Fiber optic"] <- 2
# Replace "Fiber optic" by 2
df_cat2$InternetService <- as.factor(df_cat2$InternetService)

df_cat2$Contract[df_cat2$Contract == "Month-to-month"] <- 1      #
Replace "Month-to-month" by 1
df_cat2$Contract[df_cat2$Contract == "One year"] <- 0      # Replace "One
year" by 0
df_cat2$Contract[df_cat2$Contract == "Two year"] <- 2      # Replace "Two
year" by 2
```

```
df_cat2$Contract <- as.factor(df_cat2$Contract)

df_cat2$PaymentMethod[df_cat2$PaymentMethod == "Electronic check"] <- 1
# Replace "Electronic check" by 1
df_cat2$PaymentMethod[df_cat2$PaymentMethod == "Mailed check"] <- 0
# Replace "Mailed check" by 0
df_cat2$PaymentMethod[df_cat2$PaymentMethod == "Bank transfer
(automatic)"] <- 2      # Replace "Bank transfer (automatic)" by 2
df_cat2$PaymentMethod[df_cat2$PaymentMethod == "Credit card (automatic)"]
<- 3      # Replace "Credit card (automatic)" by 3
df_cat2$PaymentMethod <- as.factor(df_cat2$PaymentMethod)

nrow(df_cat2)
nrow(df_int)
nrow(df_dummy)
```

```
## recombining final data set : data
data <- cbind(df_int,df_dummy,df_cat2)
sapply(data, class)
head(data)
describe(data)
unique(data$MultipleLines)
## Considering Out-liers
boxplot(data$tenure~data$Churn)
boxplot(data$MonthlyCharges~data$Churn)


quartiles <- quantile(data$tenure, probs=c(.25, .75), na.rm = FALSE)
IQR <- IQR(data$tenure)

Upper <- quartiles[2] + 0.369*IQR

data_no_outlier <- subset(data,data$tenure < Upper)
nrow(data_no_outlier)
boxplot(data_no_outlier$tenure~data_no_outlier$Churn)
## 3% data lost for outliers which are not too far out, therefore we keep
original data with outliers.

## Variable Selection
library(reshape2)
## creating correlation matrix
corr_mat <- round(cor(df_int),2)

## reduce the size of correlation matrix
melted_corr_mat <- melt(corr_mat)
head(melted_corr_mat)

## plotting the correlation heatmap
library(ggplot2)
ggplot(data = melted_corr_mat, aes(x=Var1, y=Var2,
                                   fill=value)) +
  geom_tile() +
  geom_text(aes(Var2, Var1, label = value),
```

```
            color = "white", size = 4)
## We can observe a high correlation for Total charge, therefore we can
drop it.
data <- data[, !(colnames(data) %in% c("TotalCharges"))]
head(data)

## EDA

## Churn percent
data %>%
  group_by(Churn) %>%
  summarise(Count = n())%>%
  mutate(percent = prop.table(Count)*100)%>%
  ggplot(aes(reorder(Churn, -percent), percent), fill = Churn)+
  geom_col()+
  theme_bw()+
  xlab("Churn") +
  ylab("Percent")+
  ggtitle("Churn Percent")
## Chart for demographic data
ggplot(df, aes(x=gender,fill=Churn))+ geom_bar()
ggplot(df, aes(x=SeniorCitizen,fill=Churn))+ geom_bar()
ggplot(df, aes(x=Partner,fill=Churn))+ geom_bar()
ggplot(df, aes(x=Dependents,fill=Churn))+ geom_bar()
```


```
## Chart for Service data
ggplot(df, aes(x=PhoneService,fill=Churn))+ geom_bar()
ggplot(df, aes(x=MultipleLines,fill=Churn))+ geom_bar()
ggplot(df, aes(x=InternetService,fill=Churn))+ geom_bar()
ggplot(df, aes(x=OnlineSecurity,fill=Churn))+ geom_bar()
ggplot(df, aes(x=TechSupport,fill=Churn))+ geom_bar()
ggplot(df, aes(x=StreamingTV,fill=Churn))+ geom_bar()
## Comaprong churn rate for continuous variables

Churn_by_Tenure <- df$tenure[df$Churn == "Yes"]
tenchn <- df$tenure[df$Churn == "No"]
a <- hist(Churn_by_Tenure, plot = FALSE)
b <- hist(tenchn, plot = FALSE)
c1 <- rgb(173,216,230,max = 255,  names = "lt.blue")
c2 <- rgb(255,192,203, max = 255, alpha = 80, names = "lt.pink")

Churn_by_MonthlyCharges<- df$MonthlyCharges[df$Churn == "Yes"]
Monthlycharges <- df$MonthlyCharges[df$Churn == "No"]
a <- hist(Churn_by_MonthlyCharges, plot = FALSE)
b <- hist(Monthlycharges, plot = FALSE)
c1 <- rgb(173,216,230,max = 255,  names = "lt.blue")
c2 <- rgb(255,192,203, max = 255, alpha = 80, names = "lt.pink")
plot(b, col = c2)
plot(a, col = c1, add = TRUE)

## Logit Model

split1<- sample(c(rep(0, 0.7 * nrow(data)), rep(1, 0.3 * nrow(data))))
```

```
train <- data[split1 == 0, ]
test <- data[split1== 1, ]
```
with all varibles

```
glm <- glm(Churn ~., data = train)
summary(glm)
tab_model(glm)
vif(glm)
```

High VIFs, insignificant variables, so we can also use step (naive method)

```
model_2<- stepAIC(glm, direction="both")
summary(model_2)
tab_model(model_2)
vif(model_2)
```

high VIF for monthly charge and Internet Service, we try two models by removing both, one at a time and choose least AIC

```
glm2 <- glm(Churn ~.-InternetService, data = train)
summary(glm2)
tab_model(glm2)
vif(glm2)
```

heteroskedasticity check for best model from above

```
par(mfrow = c(2, 2))
plot(glm)

par(mfrow = c(2, 2))
plot(glm2)

# Probit Model

glmp <- glm(Churn ~.-InternetService, family=binomial(link="probit"), data = train)
summary(glmp)
tab_model(glmp)
vif(glmp)


#Random Forest Classifier

data_rf <- data
data_rf$Churn <- as.factor(data$Churn)
indices = sample.split(data_rf$Churn, SplitRatio = 0.7)
train1 = data_rf[indices,]
test1 = data_rf[!(indices),]
```

```
model.rf <- randomForest(Churn ~ ., data=train1,
proximity=FALSE,importance = FALSE,
                         ntree=500,mtry=4, do.trace=FALSE)
model.rf
accuracy = (3254+644)/(3254+360+664+664)
accuracy

RFPred <- predict(model.rf, newdata=test[,-24])
```
ROC Curves and accuracy

logit

```
glm_result <- predict(glm2, newdata = test, type = "response")

pred_log <- prediction(glm_result, test$Churn)
table(test$Churn, glm_result>0.5)
accuracy = (1418+147)/(1418+147+270+275)
accuracy

glmpred <- predict(glm2, type = "response", newdata = test[,-24])

glm_roc <- performance(pred_log, "tpr", "fpr")
plot(glm_roc)
```

Probit

```
prb_result <- predict(glmp, newdata = test, type = "response")

pred_log_prb <- prediction(prb_result, test$Churn)
table(test$Churn, glm_result>0.6)
accuracy = (1505+145)/(1505+145+60+400)
accuracy

glmppred <- predict(glmp, type = "response", newdata = test[,-24])

glmp_roc <- performance(pred_log_prb, "tpr", "fpr")
plot(glmp_roc)
```

ROC and accuracy comparison

```
roc1 <- roc(response = test$Churn, predictor = as.numeric(RFPred))
roc2 <- roc(response = test$Churn, predictor = as.numeric(glmpred))
roc3 <- roc(response = test$Churn, predictor = as.numeric(glmppred))
roc.test( roc3, roc2)

plot(roc1, legacy.axes = TRUE)
plot(roc2, col = "blue", add = TRUE)
plot(roc3, col = "red" , add = TRUE)
legend("bottom", c("Probit Regression", "Logistic Regression", "Random
Forest"),
       lty = c(1,1), lwd = c(2, 2), col = c("red", "blue", "black"), cex
= 0.75)
```