Introduction

Customer churn is a loss to any company but a competitive industry like a telco industry, it will be a challenging factor due to technology advancements. Therefore, it is recommended to analyze existing customer behavior and predict which customers would leave the company and take preventive measures to retain them. The model which predicts the churn should be accurate enough and it can be obtained with confusion matrix.

Methodology

By using the R studio, soon after loading the data frame, I removed the index column. Then I checked for missing values (attachment 1) and removed them (attachment 2). After cleaning the data, it is a good practice to understand the data by visualizing them (attachment 3). Then I developed the prediction model by using Decision Tree Algorithm and finally checked the predictions by using confusion matrix.

Findings

Total number of records	: 7043	Number of features	: 21
Number of Null Values	: 11	Missing values in	: 01 Feature (Total Charges)
Train set size (75%)	: 5274	Test set size (25%)	: 1758

Confusion Matrix – Refer Attachment 4

```
dtree_predict
        No  Yes
  No  1152  139
  Yes  252  215
```

TP – 1152	| FN – 139
FP – 252	| TN – 215

Accuracy	= 77.76%		
Precision	= 82.05%		
Recall (For No)	= 89.23%	F1 Score (For No)	= 85.49%
Recall (For Yes)	= 46.04%	F1 Score (For Yes)	= 58.98%

TPR	= 89.23%	FPR	= 19.52%
TNR	= 46.04%	FNR	= 29.76%

Conclusions

The accuracy of the model is 77.76% but when it comes to TPR & TNR, TPR is higher (89.23%) but TNR is (46.04%) is bit lower. Also, FPR & FNR is considerably giving a higher value. (19.52% & 29.76%) Therefore we can say that the dataset is balanced (up to a certain level) and the model is acceptable. (refer attachment 04)

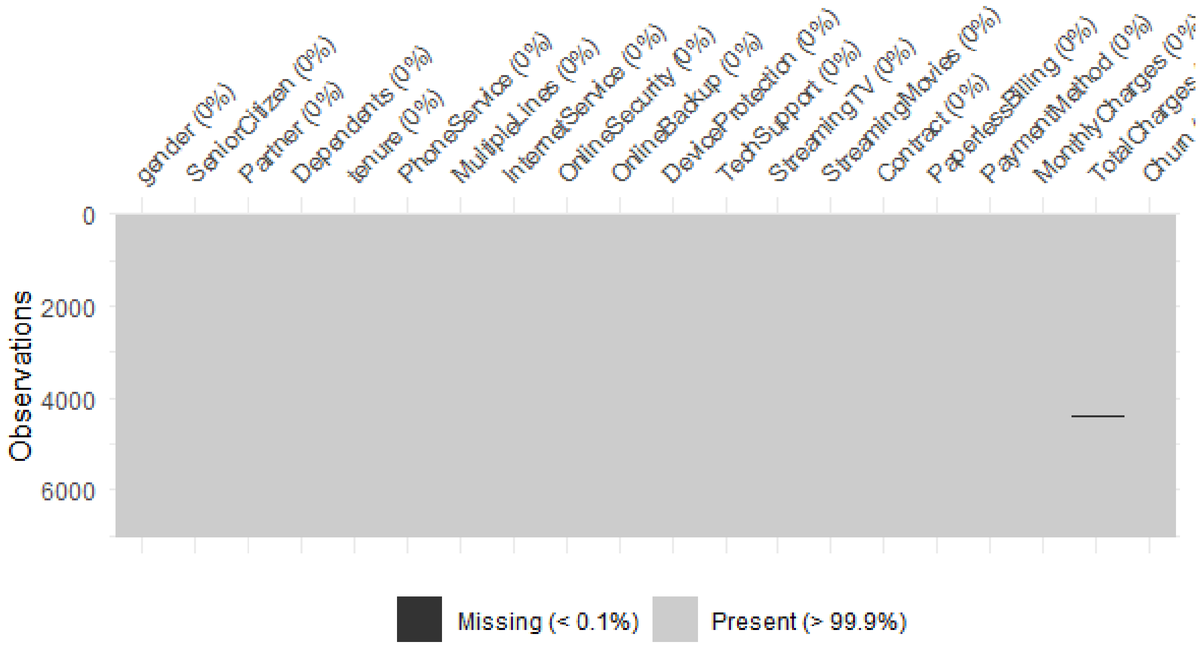
Also, the precision of the model is high and F1 score is also closer to 100% in both cases. Therefore, the model is acceptable but it's always good to test with other classification algorithms like Logistic regression, KNN, Random forest and etc.

References

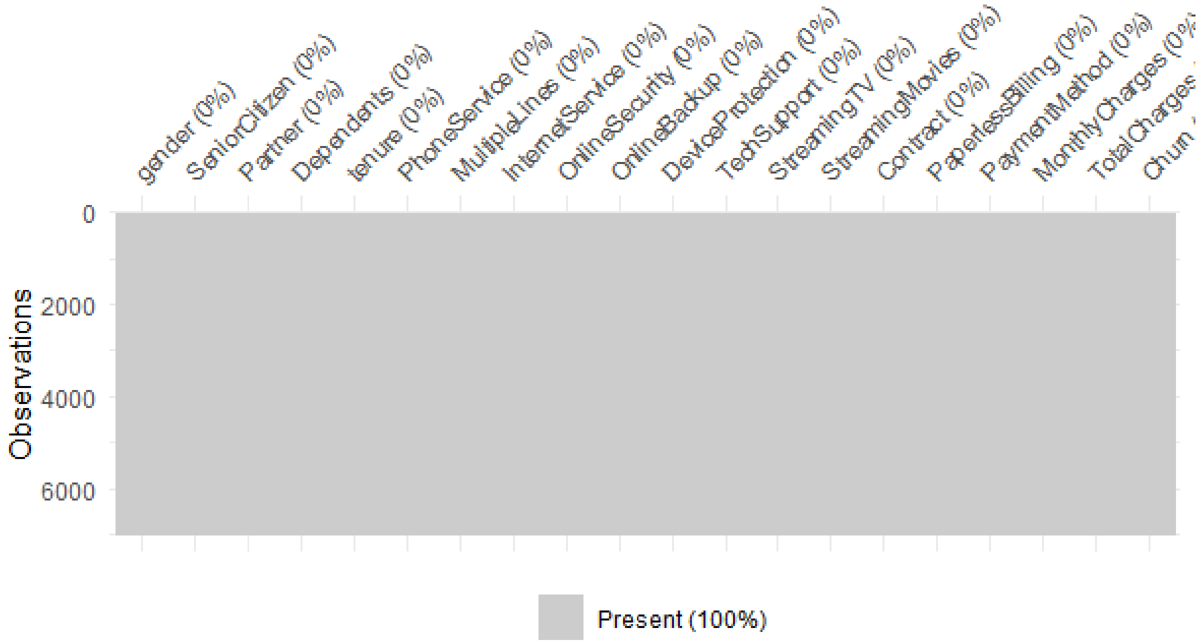
Dataset : https://www.kaggle.com/blastchar/telco-customer-churn

Appendix

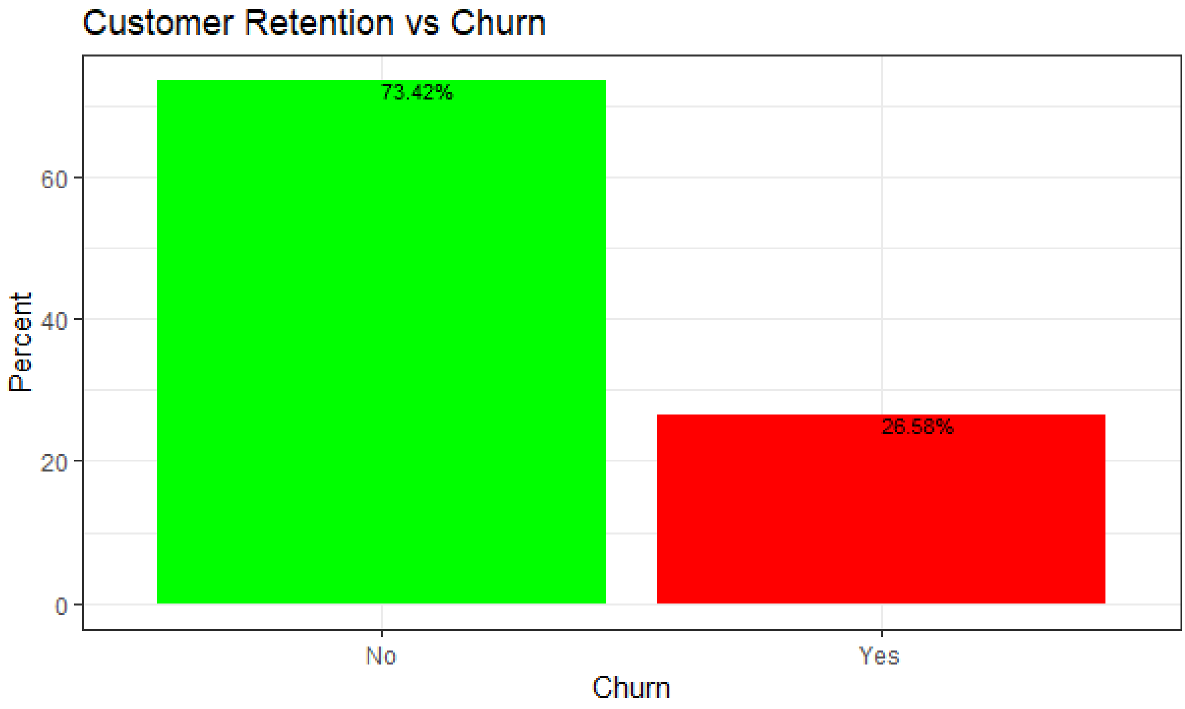
Attachment 01 – Heatmap for missing values



Attachment 2 – Heatmap after removing the missing values



Attachment 03 – Customer Retention vs Customer Churn



Attachment 4 - Confusion Matrix

```
Confusion Matrix and Statistics

     dtree_predict
        No  Yes
  No  1152  139
  Yes  252  215

               Accuracy : 0.7776
                 95% CI : (0.7574, 0.7968)
    No Information Rate : 0.7986
    P-Value [Acc > NIR] : 0.9864

                  Kappa : 0.3822

 Mcnemar's Test P-Value : 1.478e-08

            Sensitivity : 0.8205
            Specificity : 0.6073
         Pos Pred Value : 0.8923
         Neg Pred Value : 0.4604
             Prevalence : 0.7986
         Detection Rate : 0.6553
   Detection Prevalence : 0.7344
      Balanced Accuracy : 0.7139

       'Positive' Class : No


>
```

Code

```
# Customer Churn - Assignment (Telco Dataset) | L. A. C. A. Sandaruwan (199127B)
library(car)
library(e1071)
library(caret)
library(caTools)
library(heatmaply)
library(naniar)
library(rpart)
library(ggplot2)

#Load the dataset
df_telco=read.csv("datasets_13996_18858_WA_Fn-UseC_-Telco-Customer-Churn.csv",header=TRUE)
str(df_telco)
nrow(df_telco)
df_telco%>% select(-1)->df_telco
str(df_telco)

#Check for missing values
vis_miss(df_telco)

# Remove columns with missing values (Since the missing values are less than the 0.01% of the dataset)
df_telco <- na.omit(df_telco)
vis_miss(df_telco)
nrow(df_telco)

#Understand about the Churn vs Customer Retention
options(repr.plot.width = 1, repr.plot.height = 4)
```

```
df_telco %>%
  group_by(Churn) %>%
  summarise(Count = n())%>%
  mutate(percent = prop.table(Count)*100)%>%
  ggplot(aes(reorder(Churn, -percent), percent), fill = Churn)+
  geom_col(fill = c("GREEN", "RED"))+
  geom_text(aes(label = sprintf("%.2f%%", percent)), hjust = 0.01,vjust = 1, size =3)+
  theme_bw()+
  xlab("Churn") +
  ylab("Percent")+
  ggtitle("Customer Retention vs Churn")



#$Splitting the dataset as train (75%) and test (25%)
set.seed(123)
indices = sample.split(df_telco$Churn, SplitRatio = 0.75)
train_telco = df_telco[indices,]
test_telco = df_telco[!(indices),]
nrow(train_telco)
nrow(test_telco)
#test_telco$Churn = replace(test_telco$Churn,test_telco$Churn=="No"||test_telco$Churn== "Yes",NA)
#head(test_telco,5)


# By using Decesion Tree Algorithum, develop a model
dtree_train=rpart(Churn~.,data=train_telco)
summary(dtree_train)
dtree_predict=predict(dtree_train,newdata=test_telco,type = "class")
confusionMatrix(table(test_telco$Churn,dtree_predict))
```