**USE CASE STUDY REPORT**

**Group No**.: Group 15

**Student Names**: Hansika Karkera and Vrunda Shah

Executive Summary:

Customer attrition (also known as customer churn or customer turnover) is the most important metric that directly affects a business’s profitability. The profitability of any service is directly proportional to the growth of its customer base. The goal of this case study is to evaluate the customer information data retrieved from kaggle.com for Telco industry, to analyze customer churn and the factors that affect it. We begin with data preprocessing following which we use data mining techniques like Classification, Regression and Prediction to create various data models and compare their accuracy, to find the best suited model.

# I. Background and Introduction

Customers are the life force of every business and in this competitive era finding a new customer is as difficult as retaining an old one. Hence analyzing customer data proves essential for a company to understand it’s customer’s need and reduce customer attrition. Customer attrition or customer churn is the term used for loss of customers. It can be powered by several factors and even a small month-on-month rise in the churn can prove to be a huge loss to a company. One of the industries in which analyzing customer attrition proves to be profitable is the telecom industry. In this case we will be evaluating the customer information of Telco industry to analyze their customer defection and the factors that affect it.

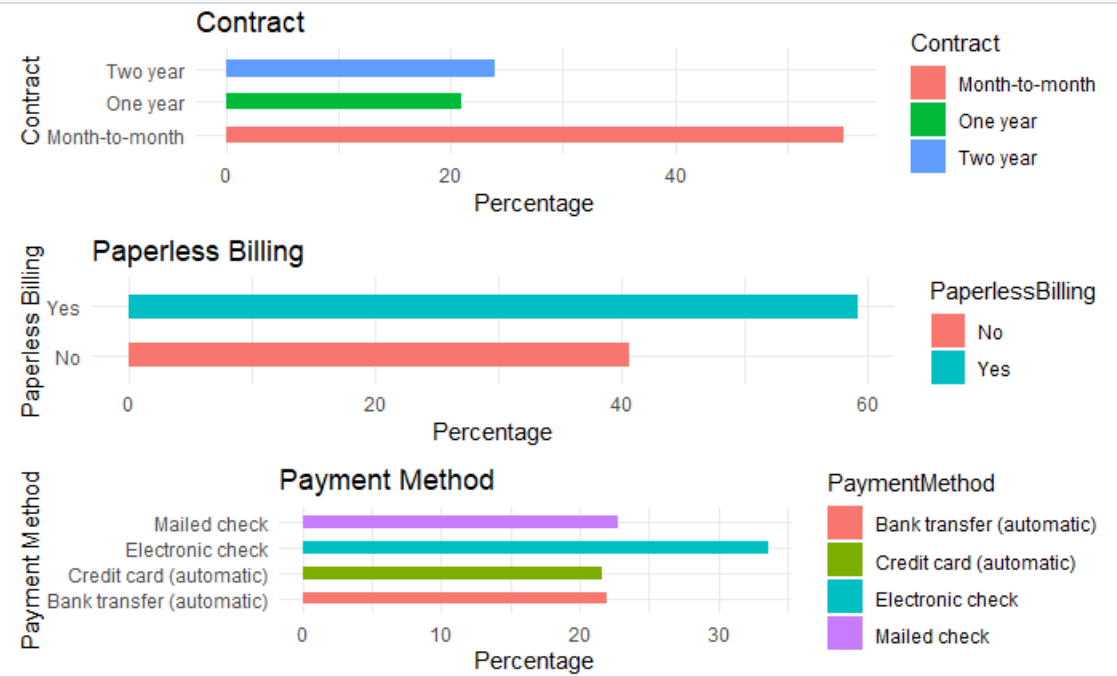
The aim of this case study is to identify the features that affect customer turnover in Telco. Secondly, to predict the tenure around which a customer is most likely to drop out of service.

Deriving intelligence from the data will help the service provider understand their customers better and enable them to customize the billing plans according to their customers flexibility. It will help them distinguish between the customers they need to work on for retention and provide their most profitable customers with suitable benefits.

# II. Data Exploration and Visualization

This is the very first stage of engaging with the dataset, it guides us towards how to begin with the data cleaning. We begin with checking if there are any missing values in the data with help of summary function. We found that there are 11 missing values in the dataset. To get a complete data for all the attributes we remove all the records with a missing value. Moving on, we use categorical plots like bar plots and pie charts to check the distribution of the categorical variable.

These plots help with the variable derivation and selection, they can help determine if there are any redundant variables that can be excluded from the analysis. Below is one of the bar plots from which we can see that all the categorical variables have a wide distribution and hence cannot be dropped

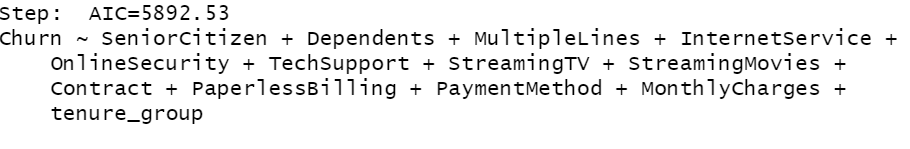


**III. Data Preparation and Preprocessing**

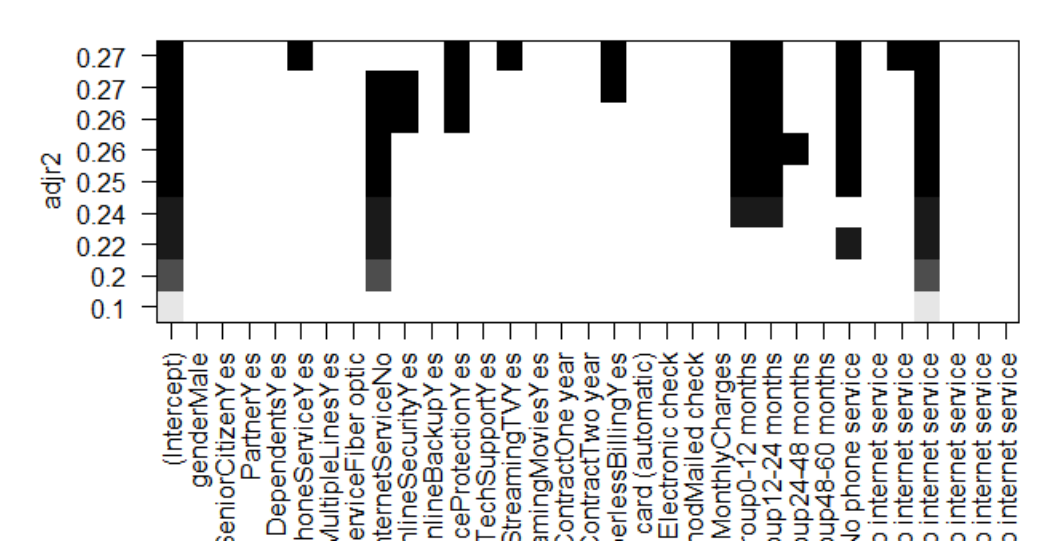
From the summary of the dataset we can observe that all the predictors except monthly charges and total charges are categorical variables. In real world, total charges for every customer is an aggregate of the monthly charges, showcasing correlation between the two variables. Hence, we plot the correlation between the two numerical variables and observe they have a correlation of 0.65. Since they are highly correlated, we can drop one of the two variables, in this case we drop total charges. From the summary, we also see that most of the attributes have values of “No internet service” or “No phone service” which can be replaced by “No” because either of the values denote the same meaning. After making the above changes to the categories, all the predictors have a uniform set of values. The value of a customer’s Id won’t be having any effect on the analysis we drop the predictor from the analysis.

Since most of the independent variables are categorical, we begin with building a generalized logistic model for the whole data for variable selection. The two most popular approaches to selecting a final set of predictors from a larger pool are stepwise method and all-subset regression.

The first method that we use for variable selection is backward stepwise regression. Backward stepwise regression begins with including all the predictor variables, and then deletes variable one after the other until removing variables would degrade the quality of the model. Below is the resulting model after backward stepwise regression. This would be one of the two models we will be using for classification.



The choose the second set of variables we use all subsets regression. This method presents us with all possible combination of predictors, it uses the adjusted R-squared criterion for reporting the best models. The adjusted R-squared attempts to provide a more honest estimate of the population.



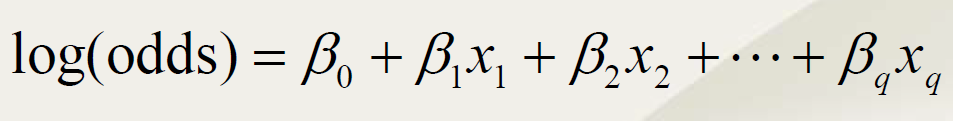
# IV. Data Mining Techniques and Implementation

To begin with data mining, we will be using supervised methods of Classification and Prediction.

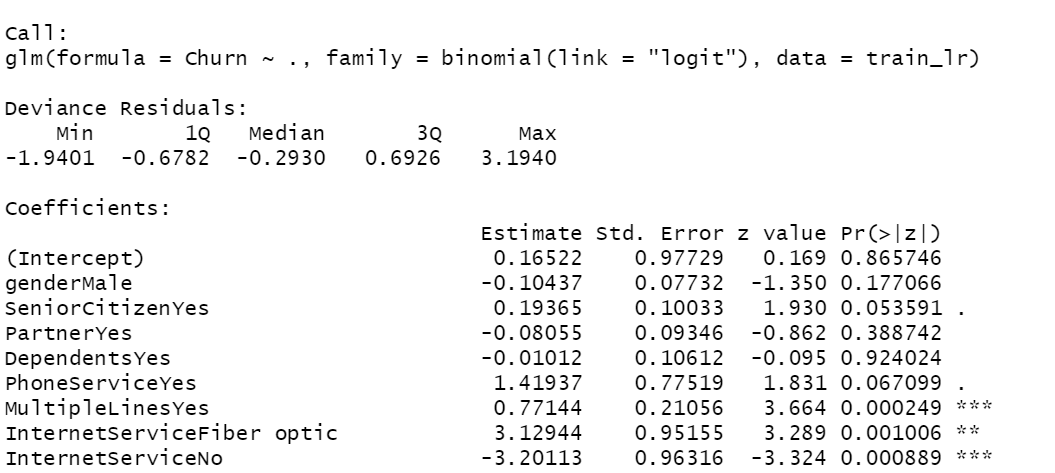
Classification in data mining is the method that is used to classify each record in the dataset into a predefined class. We will be using various classification methods on the telco dataset. We will start with splitting the data into training and testing set to avoid overfitting. Using the training data, we will build a model for each method and observe the impact of each input variable on the target variable, this will help us remove the variables that have the least impact and have no contribution towards the accuracy of the model. Further on we will compare the accuracies for the classification models, build classification matrix and compute the root-mean square error. In this case study since we are predicting if a customer will Churn or not, Churn will be our dependent variable and the variables selected after variable selection will be the independent variables. As we move on towards building new models using different methods ,we will be dropping the least contributing predictor variables to see if they have any impact on the accuracy of the model.

**1.Logistic Regression**

Logistic Regression is the first model that comes to mind when the target variable is categorical. In this case we will be predicting if Churn value is Yes or No for all the records in the training data and using it to create the confusion matrix and ROC curve. The glm function in R is used to estimate the coefficient value β0, β1…. βq for all the predictors in the logistic model.



However, the values estimated by the glm function are in logits. So a unit change in genderMale produces approximately -0.10437 unit change in the log odds.



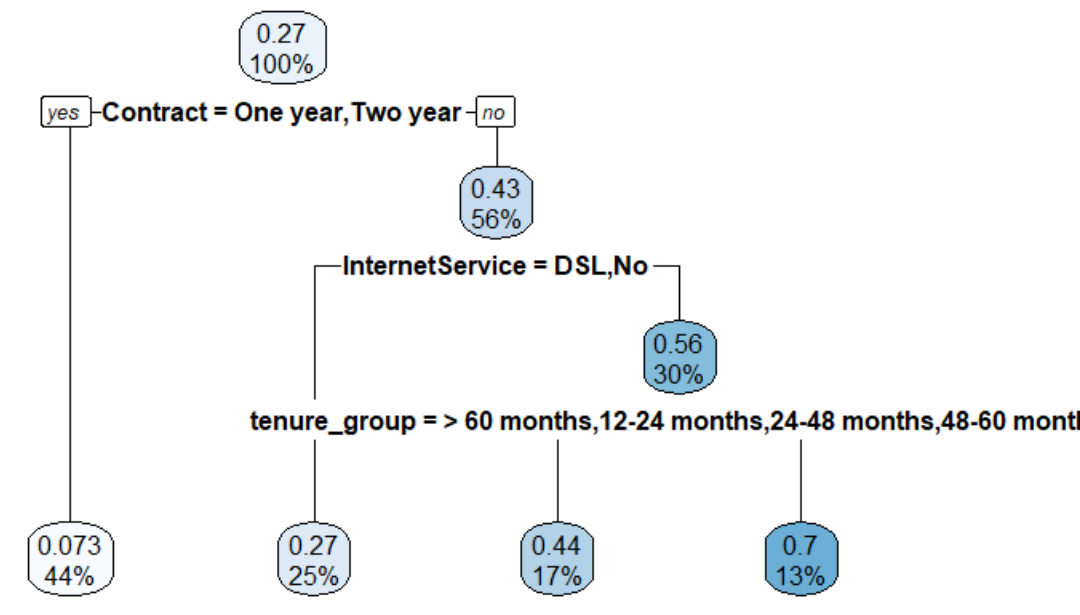
We create the logistic regression model for two set of variables selected in the previous section.

On observing the p-values for the logistic model we can say that the predictors that affect Churn the most are Tenure, Paperless Billing, Contract and Internet Services. In addition, we build confusion matrix using validation dataset, ROC-AUC curve, and find accuracy of the model.

**2. Decision Tree**

The decision tree model builds classification models in the form of a tree-like structure which can be used to classify both numerical and categorical variables. The root node is the most significant predictor and the sub-nodes formed after the split are the decision nodes. The node selection is decided by computing the gini index for the predictor variables at every split.

The input variables are same for the first decision tree, the next two decision tree inputs are selected by dropping the least important variables as per the logistic model, to see if there is any improvement or loss in accuracy. From the classification trees plotted using rpart we can observe that the significant variables towards predicting churn are Contract, Tenure Group and Internet Services. Further, with the help of the models created we construct the classification matrix.



**3. Random Forest**

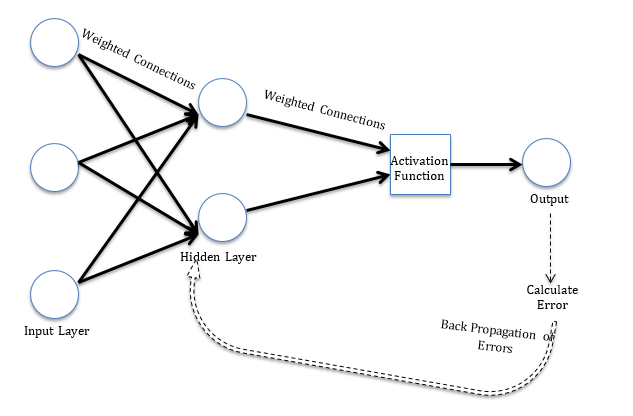
Random Forest model is built upon many decision trees. It uses random sampling of training data when building trees and random subsets of features when splitting into sub-nodes. Since it is an advanced version of decision tree, its performance is said to be better than decision tree. In R the in-built randomForest package not only helps create the classification model but also plot the variable importance plot for the corresponding random forest model. The input variables for random forest are same as the decision tree.

**4. K-Nearest Neighbors**

The k-nearest algorithm is a simple, supervised learning algorithm to solve both classification and regression problems. It’s not the most ideal algorithm to be used for categorical variables but we can create dummy variables for the categories and use them for classification. Post converting, we normalize the data and find the appropriate k-value to build the model. In this case the two k-nn models built have a k value of 68 and 46. Further on we build the classification matrix using the validation dataset and check the accuracy.

**5. Neural Networks**

Neural Networks is a series of algorithms that improvises on the model as it builds. It adapts to changing input, so the network generated the best possible result without needing to redesign the model after every iteration. Using the neural network package in R we build the network model that is used to classify the validation model. Further, we build the confusion matrix and caluculte the accuracy.



**6. Linear Discriminant Analysis**

Linear Discriminant Analysis is a model-based approach in which classification is based on the distance of an observation from each class average.

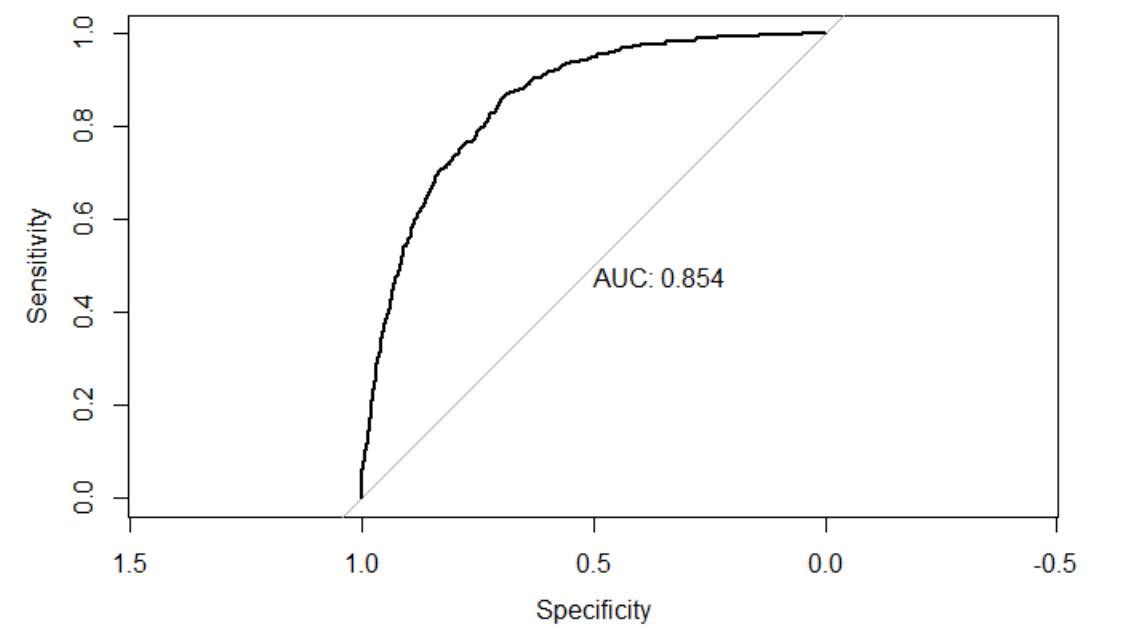
# V. Performance Evaluation

**1.Logistic Model :**

For a sample of training data, the third logistic model predicts the positive class of ‘0’ i.e. No churn with an accuracy better than the other two models. For all the three model the sensitivity of the model is high, meaning that the number of correct positive predictions is better than that of the number of correct negative predictions.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **Predictors** | **RMSE** | **Accuracy** | **Sensitivity** | **Specificity** |
| Logistic Model -1 | All predictor variables after Data cleaning. | 2.425 | 79.00%% | 0.9573 | 0.3517 |
| Logistic Model -2 | Variables selected after backward stepwise regression. | 2.418 | 78.66% | 0.9580 | 0.3233 |
| Logistic Model -3 | Variables selected after all-subsets regression. | 0.376 | 79.29% | 0.9354 | 0.4014 |

Also, from the ROC-AUC curve we can see that the AUC value for all the three models is above 0.80. Meaning that the model has a the efficiency of predicting the classes right for the given validation dataset.



**2.Decision Tree :**

For the training dataset generated, the second model predicts the positive class of ‘0’ i.e. No churn with an accuracy better than the first model. For all the models the sensitivity of the model is high, meaning that the number of correct positive predictions is better than that of the number of correct negative predictions.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **Predictors** | **RMSE** | **Accuracy** | **Sensitivity** | **Specificity** |
| Decision Tree -1 | Variables selected after all-subsets regression. | 0.379 | 79.05% | 0.9476 | 0.3588 |
| Decision Tree -2 | Dropping the variables that have least impact on the output as per the logistic model – Dependents, Multiple Lines | 0.379 | 79.28% | 0.9504 | 0.3327 |

**3.Random Forest :**

The second random forest model predicts the positive class of ‘0’ i.e. No churn with an accuracy better than the first model. For all the models the sensitivity of the model is high, meaning that the number of correct positive predictions is better than that of the number of correct negative predictions. However, on comparing the decision tree model and random forest model, we can see that for this dataset the decision tree model classifies more accurately than the random forest model for a given set of data.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **Predictors** | **RMSE** | **Accuracy** | **Sensitivity** | **Specificity** |
| Random Forest - 1 | Same as decision tree - 2 | 0.392 | 77.22% | 0.8783 | 0.4983 |
| Random Forest - 2 | 5 important variables | 0.38 | 78.67 | 0.8862 | 0.5305 |

**4. K – Nearest Neighbors :**

The second K-NN model predicts the positive class of ‘0’ i.e. No churn with an accuracy better than the first model. For all the models the sensitivity of the model is better than the specificity, meaning that the number of correct positive predictions is better than that of the number of correct negative predictions.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **Predictors** | **RMSE** | **Accuracy** | **Sensitivity** | **Specificity** |
| K-NN -1 | Same as decision tree - 2 | - | 79.48 | 0.8987 | 0.5071 |
| K-NN -1 | Same as random forest -2 | - | 80.28 | 0.8866 | 0.5699 |

**5. Neural Networks :**

The first network model predicts the positive class of ‘0’ i.e. No churn with an accuracy better than the second model. For all the models the sensitivity of the model is better than the specificity, meaning that the number of correct positive predictions is better than that of the number of correct negative predictions. However, in this case the model with the better accuracy does not have the better sensitivity, but the difference in value is quite low.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **Predictors** | **RMSE** | **Accuracy** | **Sensitivity** | **Specificity** |
| Neural Network -1 | Same as decision tree - 2 | 0.36 | 79.95 | 0.8926 | 0.5451 |
| Neural Network -2 | Same as random forest - 2 | 0.38 | 78.87 | 0.9070 | 0.4751 |

**6. Linear Discriminant Analysis:**

The second analytics model predicts the positive class of ‘0’ i.e. No churn with an accuracy better than the second model. For all the models the sensitivity of the model is better than the specificity.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **Predictors** | **RMSE** | **Accuracy** | **Sensitivity** | **Specificity** |
| LDA -1 | Same as decision tree - 2 | 0.68 | 79.62 | 0.9065 | 0.4911 |
| LDA - 2 | Same as random forest - 2 | 0.64 | 80.24 | 0.9130 | 0.4946 |

All above measurements are for a sample, with every new sample the values vary slightly but the best performing model remains the same.

Post creating the classification model, we predict the Churn value for a new data set using the Neural network model and the LDA model.

# VI. Discussion and Recommendation

Provide discussion of the overall approach, including advantages and shortcomings. Based on your results, make recommendations for solution, potential improvement, etc.

# VII. Summary

Important predictors – Tenure, Contract, Internet Service and Paperless Billing.

No strong relation between Churn and Gender, Multiple Lines, Device Protection, Tech Support, Online Security, Online Backup.

Customers between the tenure of 0-12 months or 12-24 months are more likely to Churn.

# Appendix: R Code for use case study

# Read File

churn\_df <- read.csv("Telco\_Customer\_Churn.csv")

# Check for NAs

summary(churn\_df)

# Remove NAs

churn\_df <- na.omit(churn\_df)

cc <- churn\_df

# Data Wrangling

churn\_df$MultipleLines[churn\_df$MultipleLines=="No phone service"] <- as.factor("No")

churn\_df$OnlineSecurity[churn\_df$OnlineSecurity=="No internet service"] <- as.factor("No")

churn\_df$OnlineBackup[churn\_df$OnlineBackup=="No internet service"] <- as.factor("No")

churn\_df$DeviceProtection[churn\_df$DeviceProtection=="No internet service"] <- as.factor("No")

churn\_df$TechSupport[churn\_df$TechSupport=="No internet service"] <- as.factor("No")

churn\_df$StreamingTV[churn\_df$StreamingTV=="No internet service"] <- as.factor("No")

churn\_df$StreamingMovies[churn\_df$StreamingMovies=="No internet service"] <- as.factor("No")

churn\_df$SeniorCitizen[churn\_df$SeniorCitizen==0]<- "No"

churn\_df$SeniorCitizen[churn\_df$SeniorCitizen==1]<- "Yes"

library(gridExtra)

library(ggplot2)

# Categorical Data Plots

p1 <- ggplot(churn\_df, aes(x=gender,fill = gender)) + ggtitle("Gender") + xlab("Gender") + geom\_bar(aes(y = 100\*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") + coord\_flip() + theme\_minimal()

p2 <- ggplot(churn\_df, aes(x=SeniorCitizen,fill = SeniorCitizen)) + ggtitle("Senior Citizen") + xlab("Senior Citizen") + geom\_bar(aes(y = 100\*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") + coord\_flip() + theme\_minimal()

p3 <- ggplot(churn\_df, aes(x=Partner,fill=Partner)) + ggtitle("Partner") + xlab("Partner") + geom\_bar(aes(y = 100\*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") + coord\_flip() + theme\_minimal()

p4 <- ggplot(churn\_df, aes(x=Dependents,fill = Dependents)) + ggtitle("Dependents") + xlab("Dependents") + geom\_bar(aes(y = 100\*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") + coord\_flip() + theme\_minimal()

grid.arrange(p1, p2, p3, p4, ncol=2)

library(plotly)

library(dplyr)

fig2 <- plot\_ly()

fig2 <- fig2 %>%

add\_pie(data = count(churn\_df,PhoneService),labels = ~PhoneService, values = ~n,name="PhoneService",domain = list(row = 0, column = 0))

fig2 <- fig2 %>%

add\_pie(data = count(churn\_df,MultipleLines),labels = ~MultipleLines, values = ~n,name="MultipleLines",domain = list(row = 0, column = 1))

fig2 <- fig2 %>%

add\_pie(data = count(churn\_df,InternetService),labels = ~InternetService, values = ~n,name="InternetService",domain = list(row = 1, column = 0))

fig2 <- fig2 %>%

add\_pie(data = count(churn\_df,OnlineSecurity),labels = ~OnlineSecurity, values = ~n,name="OnlineSecurity",domain = list(row = 1, column = 1))

fig2 <- fig2 %>% layout(annotations = list(

list(x=0.2,y= 1.07,text="PhoneService",showarrow = F, xref='paper', yref='paper'), list(x=0.8, y=0.95, text="MultipleLines", xref='paper', yref='paper'), list(x=0.25,y=0.5, text="InternetService", showarrow = F, xref='paper', yref='paper'), list(x=0.8, y=0.5, text="OnlineSecurity", showarrow = F, xref='paper', yref='paper')

), showlegend = T, grid=list(rows=2, columns=2))

fig2

fig3 <- plot\_ly()

fig3 <- fig3 %>%

add\_pie(data = count(churn\_df,OnlineBackup),labels = ~OnlineBackup, values = ~n,name="OnlineBackup",domain = list(x = c(0, 0.4), y = c(0.4, 1)))

fig3 <- fig3 %>%

add\_pie(data = count(churn\_df,DeviceProtection),labels = ~DeviceProtection, values = ~n,name="DeviceProtection",domain = list(x = c(0.6, 1), y = c(0.4, 1)))

fig3 <- fig3 %>%

add\_pie(data = count(churn\_df,TechSupport),labels = ~TechSupport, values = ~n,name="TechSupport",domain = list(x = c(0.25, 0.75), y = c(0, 0.6)))

fig3 <- fig3 %>% layout(annotations = list(

list(x=0.2,y= 1.07,text="OnlineBackup",showarrow = F, xref='paper', yref='paper'), list(x=0.8, y=0.95, text="DeviceProtection", xref='paper', yref='paper'), list(x=0.5,y=0.7, text="TechSupport", showarrow = F, xref='paper', yref='paper')), showlegend = T)

fig3

fig4 <- plot\_ly()

fig4 <- fig4 %>%

add\_pie(data = count(churn\_df,StreamingTV),labels = ~StreamingTV, values = ~n,name="StreamingTV",domain = list(x = c(0, 0.4), y = c(0.4, 1)))

fig4 <- fig4 %>%

add\_pie(data = count(churn\_df,StreamingMovies),labels = ~StreamingMovies, values = ~n,name="StreamingMovies",domain = list(x = c(0.6, 1), y = c(0.4, 1)))

fig4 <- fig4 %>% layout(annotations = list(

list(x=0.15,y= 1.07,text="StreamingTV",showarrow = F, xref='paper', yref='paper'), list(x=0.8, y=0.95, text="StreamingMovies", xref='paper', yref='paper')), showlegend = T)

fig4

p1 <- ggplot(churn\_df, aes(x=Contract,fill=Contract)) + ggtitle("Contract") + xlab("Contract") +

geom\_bar(aes(y = 100\*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") + coord\_flip() + theme\_minimal()

p2 <- ggplot(churn\_df, aes(x=PaperlessBilling,fill=PaperlessBilling)) + ggtitle("Paperless Billing") + xlab("Paperless Billing") +

geom\_bar(aes(y = 100\*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") + coord\_flip() + theme\_minimal()

p3 <- ggplot(churn\_df, aes(x=PaymentMethod,fill=PaymentMethod)) + ggtitle("Payment Method") + xlab("Payment Method") +

geom\_bar(aes(y = 100\*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") + coord\_flip() + theme\_minimal()

grid.arrange(p1,p2,p3,nrow=3)

# Categorizing tenure

group\_tenure <- function(tenure){

if (tenure >= 0 & tenure <= 12){

return("0-12 months")

}else if(tenure > 12 & tenure <= 24){

return("12-24 months")

}else if (tenure > 24 & tenure <= 48){

return("24-48 months")

}else if (tenure > 48 & tenure <=60){

return("48-60 months")

}else if (tenure > 60){

return("> 60 months")

}

}

churn\_df$tenure\_group <- sapply(churn\_df$tenure,group\_tenure)

churn\_df$tenure\_group <- as.factor(churn\_df$tenure\_group)

churn\_df$tenure <- NULL

library(dplyr)

library(corrplot)

library(RColorBrewer)

# Correlation plot b/w numerical columns

churn\_corr <- select(churn\_df,MonthlyCharges,TotalCharges)

corr <- cor(churn\_corr)

corrplot(corr,method="number",type="upper", order="hclust")

# Since total charges and monthly charges are highly correlated, reomove total

churn\_df$TotalCharges <- NULL

churn\_df$customerID <- NULL

# Variable Selection - Stepwise Regression

fit <- glm(Churn~.,data = churn\_df,family = binomial(link = "logit"))

library(MASS)

stepAIC(fit,direction = "backward")

# Variable Selection - All subsets Regression

library(leaps)

leaps <- regsubsets(Churn~., data = churn\_df)

plot(leaps,scale = "adjr2")

library(caret)

# Split to training and test

churn\_lr <- churn\_df

churn\_lr$Churn <- 1 \* (churn\_lr$Churn == "Yes")

intrain<- createDataPartition(churn\_lr$Churn,p=0.7,list=FALSE)

# set.seed(20)

train\_lr<- churn\_lr[indices,]

test\_lr<- churn\_lr[-indices,]

# Building logistic regression model with all the attributes

lr\_model <- glm(Churn~.,data = train\_lr,family = binomial(link = "logit"))

summary(lr\_model)

test\_lr\_predict1<- predict(lr\_model,test\_lr)

test\_lr\_predict <- factor(ifelse(test\_lr\_predict1 >= 0.5, "1", "0"))

#Confusion Matrix for Logistic Model -1

confusionMatrix(data = test\_lr\_predict, reference = as.factor(test\_lr$Churn))

library(pROC)

roc(test\_lr$Churn,test\_lr\_predict1,plot=TRUE,print.auc=TRUE)

library(MASS)

# Logistic model - using stepAIC

lr\_aic\_model <- glm(formula = Churn ~ SeniorCitizen + Dependents + MultipleLines + InternetService + OnlineSecurity + TechSupport + StreamingTV + StreamingMovies + Contract + PaperlessBilling + PaymentMethod + MonthlyCharges + tenure\_group, family = binomial(link = "logit"), data = train\_lr)

summary(lr\_aic\_model)

# Predict test data using the AIC model

test\_lr\_predict2 <- predict(lr\_aic\_model,test\_lr)

test\_pred2 <- factor(ifelse(test\_lr\_predict2 >= 0.5, "1", "0"))

confusionMatrix(data = test\_pred2, reference = as.factor(test\_lr$Churn))

library(pROC)

roc(test\_lr$Churn,test\_lr\_predict2,plot=TRUE,print.auc=TRUE)

# Logistic model using the variables selected from all subset regression.

lr\_model3 <- glm(Churn~PhoneService+DeviceProtection+StreamingTV+PaperlessBilling+tenure\_group+InternetService+Contract,data = train\_lr)

summary(lr\_model3)

lr\_predict3 <- predict(lr\_model3,test\_lr)

test\_pred3 <- factor(ifelse(lr\_predict3 >= 0.5, "1", "0"))

confusionMatrix(data = test\_pred3, reference = as.factor(test\_lr$Churn))

roc(test\_lr$Churn,lr\_predict3,plot=TRUE,print.auc=TRUE)

# Decision Tree – 01

# Constructing the decision tree for the all-subset regression variables

library(rpart)

library(rpart.plot)

r1<- rpart(Churn ~ PhoneService+DeviceProtection+StreamingTV+PaperlessBilling+tenure\_group+InternetService+Contract,train\_lr,parms=list(split=c("information","gini")))

rpart.plot(r1)

# Confusion matrix for the first decision tree

p1 <- predict(r1,test\_lr)

p1\_factor <- factor(ifelse(p1 >= 0.5, "1", "0"))

confusionMatrix(p1\_factor,as.factor(test\_lr$Churn))

# Decision Tree - 02

library(rpart)

library(rpart.plot)

r2<- rpart(Churn~SeniorCitizen+InternetService+StreamingTV+StreamingMovies+Contract+PaperlessBilling+PaymentMethod+tenure\_group+MonthlyCharges,train\_lr,parms=list(split=c("information","gini")))

rpart.plot(r2)

p2 <- predict(r2,test\_lr)

p2\_factor <- factor(ifelse(p2 >= 0.5, "1", "0"))

confusionMatrix(p2\_factor,as.factor(test\_lr$Churn))

# Creating Random Forest

library(randomForest)

churn\_rf <- churn\_df

churn\_rf$Churn <- 1 \* (churn\_rf$Churn == "Yes")

churn\_rf <- churn\_rf %>%

mutate\_if(is.character, as.factor)

train.index <- sample(row.names(churn\_rf), 0.7\*dim(churn\_rf)[1])

valid.index <- setdiff(row.names(churn\_rf), train.index)

train\_rf <- churn\_rf[train.index, ]

valid\_rf <- churn\_rf[valid.index, ]

rf1 <- randomForest(Churn~SeniorCitizen+InternetService+StreamingTV+StreamingMovies+Contract+PaperlessBilling+PaymentMethod+tenure\_group+MonthlyCharges, data = train\_rf, ntree = 500,mtry = 4, nodesize = 5, importance = TRUE)

varImpPlot(rf1)

## confusion matrix

rf.predict\_1 <- predict(rf1, valid\_rf)

rf.pred <- factor(ifelse(rf.predict\_1 >= 0.5, "1", "0"))

confusionMatrix(rf.pred, as.factor(valid\_rf$Churn))

rf3 <- randomForest(Churn~InternetService+Contract+PaperlessBilling+PaymentMethod+tenure\_group, data = train\_rf, ntree = 500,mtry = 4, nodesize = 5, importance = TRUE)

## confusion matrix

rf.predict\_3 <- predict(rf3, valid\_rf)

rf.pred3 <- factor(ifelse(rf.predict\_3 >= 0.5, "1", "0"))

confusionMatrix(rf.pred3, as.factor(valid\_rf$Churn))

#K-NN

library(class)

library(FNN)

library(caret)

## creating dummy variables and data binnig the catrgorical data

churn\_df$Contract<-as.factor(churn\_df$Contract)

churn\_knn <- churn\_df

churn\_knn[,c("month-to-month","one year","two year")]<-model.matrix(~Contract-1,data = churn\_knn)

churn\_knn$tenure\_group<-as.factor(churn\_knn$tenure\_group)

churn\_knn[,c("0-12 months","12-24 months","24-48 months","48-60 months","> 60 months")]<-model.matrix(~tenure\_group-1,data = churn\_knn)

##converting payment method

churn\_knn$PaymentMethod<-as.factor(churn\_knn$PaymentMethod)

churn\_knn[,c("Electronic Check","Mailed Check","Bank Transfer","Credit card")]<-model.matrix(~PaymentMethod-1,data = churn\_knn)

##converting internet service

churn\_knn$InternetService<-as.factor(churn\_knn$InternetService)

churn\_knn[,c("DSL","Fiber optic","No Internest service")]<-model.matrix(~InternetService-1,data = churn\_knn)

##converting paperless billing and churn

churn\_knn$PaperlessBilling <- 1\* (churn\_knn$PaperlessBilling == "Yes")

churn\_knn$Churn <- 1\* (churn\_knn$Churn == "Yes")

churn\_knn$SeniorCitizen <- 1\* (churn\_knn$SeniorCitizen == "Yes")

churn\_knn$StreamingMovies <- 1\* (churn\_knn$StreamingMovies == "Yes")

churn\_knn$StreamingTV <- 1\* (churn\_knn$StreamingTV == "Yes")

## selecting the predictors

churn\_knn1 <- churn\_knn[,c(2,32,33,34,12,13,20,21,22,15,28,29,30,31,23,24,25,26,27,17,18)]

churn\_knn1[,21]<-as.factor(churn\_knn1[,21])

## diving the data into (70%)training and (30%)validation data

indices= sample(nrow(churn\_knn1), 0.7\*nrow(churn\_knn1))

train1 = churn\_knn1[indices, ] #70% of the data

val1 = churn\_knn1[-indices,] #30% of the data

## normalizing the data

train\_label1 <- train1

val\_label1 <- val1

churn\_knn\_label1 <- churn\_knn1

library(caret)

norm<-preProcess(train1[,-21],method = c("center","scale"))

train\_label1[,-21] <- predict(norm,train1[,-21])

val\_label1[,-21] <- predict(norm,val1[,-21])

churn\_knn\_label1[,-21] <- predict(norm,churn\_knn1[,-21])

library(class)

## calculating the best value of k

accuracy\_df1 <- data.frame(k = seq(1,65,1), accuracy = rep(0, 65))

for(i in 1:65) {

knn.pred1 <- knn(train\_label1[,-21], val\_label1[,-21], cl = train\_label1[,21], k=i)

accuracy\_df1[i, 2] <- confusionMatrix(knn.pred1, val\_label1[,21])$overall[1]

}

plot(accuracy\_df1,type='b')

## creating prediction model for knn

knn\_pred1 <-knn(train=train\_label1[,-21],test=val\_label1[,-21],cl=train\_label1[,21],k=40)

## calculating confusion matrix and accuracy of the prediction model

confusionMatrix(knn\_pred1 ,val\_label1[,21])

## selecting the predictors

churn\_knn2 <- churn\_knn[,c(32,33,34,20,21,22,15,28,29,30,31,23,24,25,26,27,18)]

churn\_knn2[,17]<-as.factor(churn\_knn2[,17])

## diving the data into (70%)training and (30%)validation data

indices= sample(nrow(churn\_knn2), 0.7\*nrow(churn\_knn2))

train2 = churn\_knn2[indices, ] #70% of the data

val2 = churn\_knn2[-indices,] #30% of the data

## normalizing the data

train\_label2 <- train2

val\_label2 <- val2

churn\_knn\_label2 <- churn\_knn2

library(caret)

norm2<-preProcess(train2[,-17],method = c("center","scale"))

train\_label2[,-17] <- predict(norm2,train2[,-17])

val\_label2[,-17] <- predict(norm2,val2[,-17])

churn\_knn\_label2[,-17] <- predict(norm2,churn\_knn2[,-17])

library(class)

## calculating the best value of k

accuracy\_df2 <- data.frame(k = seq(1,50,1), accuracy = rep(0,50))

for(i in 1:50) {

knn.pred2 <- knn(train\_label2[,-17], val\_label2[,-17], cl = train\_label2[,17], k=i)

accuracy\_df2[i, 2] <- confusionMatrix(knn.pred2, val\_label2[,17])$overall[1]

}

plot(accuracy\_df2,type='b')

## creating prediction model-2 for knn

knn\_pred2 <-knn(train=train\_label2[,-17],test=val\_label2[,-17],cl=train\_label2[,17],k=43)

## calculating confusion matrix and accuracy of the prediction model

confusionMatrix(knn\_pred2 ,val\_label2[,17])

library(dplyr)

churn\_nn1 <- select(churn\_df,SeniorCitizen,InternetService,StreamingTV,StreamingMovies,Contract,PaperlessBilling,PaymentMethod,tenure\_group,MonthlyCharges,Churn)

churn\_nn1$SeniorCitizen <- 1\* (churn\_nn1$SeniorCitizen == "Yes")

churn\_nn1$StreamingTV <- 1\* (churn\_nn1$StreamingTV == "Yes")

churn\_nn1$StreamingMovies <- 1\* (churn\_nn1$StreamingMovies == "Yes")

churn\_nn1$PaperlessBilling <- 1\* (churn\_nn1$PaperlessBilling == "Yes")

churn\_nn1$Churn <- 1\* (churn\_nn1$Churn == "Yes")

churn\_nn1$MonthlyCharges <- scale(churn\_nn1$MonthlyCharges)

vars=c("InternetService","Contract","PaymentMethod","tenure\_group")

library(neuralnet)

library(BART)

Data <- cbind(churn\_nn1[,c(vars)],

class.ind(churn\_nn1[,]$InternetService),

class.ind(churn\_nn1[,]$Contract),

class.ind(churn\_nn1[,]$PaymentMethod),

class.ind(churn\_nn1[,]$tenure\_group))

names(Data)=c(vars,

paste("InternetService\_", c(1, 2, 3, 4), sep=""), paste("Contract\_", c(1, 2, 3), sep=""),paste("PaymentMethod\_", c(1,2,3,4), sep=""),paste("tenure\_group", c(1,2,3,4), sep=""))

Data[,1:4]<-NULL

input\_nn1 <- cbind(churn\_nn1,Data)

input\_nn1$InternetService<-NULL

input\_nn1$Contract<-NULL

input\_nn1$PaymentMethod<-NULL

input\_nn1$tenure\_group<-NULL

indices= sample(nrow(input\_nn1), 0.7\*nrow(input\_nn1))

train1 = input\_nn1[indices, ] #70% of the data

val1 = input\_nn1[-indices,] #30% of the data

nn1 <- neuralnet(Churn~.,data = train1,hidden = 2)

plot(nn1)

library(caret)

nn\_predict1 <- predict(nn1, val1)

nn\_pred1 <- factor(ifelse(nn\_predict1 >= 0.5, "1", "0"))

confusionMatrix(nn\_pred1, as.factor(val1$Churn))

# Neural Networks -2

library(dplyr)

churn\_nn2 <- select(churn\_df,InternetService,Contract,PaperlessBilling,PaymentMethod,tenure\_group,Churn)

churn\_nn2$PaperlessBilling <- 1\* (churn\_nn2$PaperlessBilling == "Yes")

churn\_nn2$Churn <- 1\* (churn\_nn2$Churn == "Yes")

vars=c("InternetService","Contract","PaymentMethod","tenure\_group")

library(neuralnet)

library(BART)

Data <- cbind(churn\_nn2[,c(vars)],

class.ind(churn\_nn2[,]$InternetService),

class.ind(churn\_nn2[,]$Contract),

class.ind(churn\_nn2[,]$PaymentMethod),

class.ind(churn\_nn2[,]$tenure\_group))

names(Data)=c(vars,

paste("InternetService\_", c(1, 2, 3, 4), sep=""), paste("Contract\_", c(1, 2, 3), sep=""),paste("PaymentMethod\_", c(1,2,3,4), sep=""),paste("tenure\_group", c(1,2,3,4), sep=""))

Data[,1:4]<-NULL

input\_nn2 <- cbind(churn\_nn2,Data)

input\_nn2$InternetService<-NULL

input\_nn2$Contract<-NULL

input\_nn2$PaymentMethod<-NULL

input\_nn2$tenure\_group<-NULL

indices= sample(nrow(input\_nn2), 0.7\*nrow(input\_nn2))

train2 = input\_nn2[indices, ] #70% of the data

val2 = input\_nn2[-indices,] #30% of the data

nn2<- neuralnet(Churn~.,data = train2,hidden = 3)

plot(nn2)

```

```{r}

library(caret)

nn\_predict2 <- predict(nn2, val2)

nn\_pred2 <- factor(ifelse(nn\_predict2 >= 0.5, "1", "0"))

confusionMatrix(nn\_pred2, as.factor(val2$Churn))

#LDA

library(MASS)

linear1 <- lda(Churn ~. , data = train\_label1)

linear1

## calculating the accuracy

predictions1 <- linear1 %>% predict(val\_label1)

mean(predictions1$class==val\_label1$Churn)

## finding the classification using default cuttoff 0.5

sum(predictions1$posterior[ ,1] >=.5)

library(caret)

confusionMatrix(predictions1$class, val\_label1$Churn)

#LDA - 02

library(MASS)

linear2 <- lda(Churn ~. , data = train\_label2)

linear2

## calculating accuracy

predictions2 <- linear2 %>% predict(val\_label2)

mean(predictions2$class==val\_label2$Churn)

## finding the classification using default cuttoff 0.5

sum(predictions2$posterior[ ,1] >=.5)

library(caret)

confusionMatrix(predictions2$class, val\_label2$Churn)