USE CASE STUDY REPORT

ANALYZING CUSTOMER ATTRITION

Group No.: Group 15

Student Names: Hansika Karkera and Vrunda Shah

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 us to determine if there are any redundant variables that can be excluded from the analysis. Below is one of the bar plots and pie chart depicting wide range of distribution. Therefore, it cannot be dropped from the dataset for the analysis and creating prediction model.

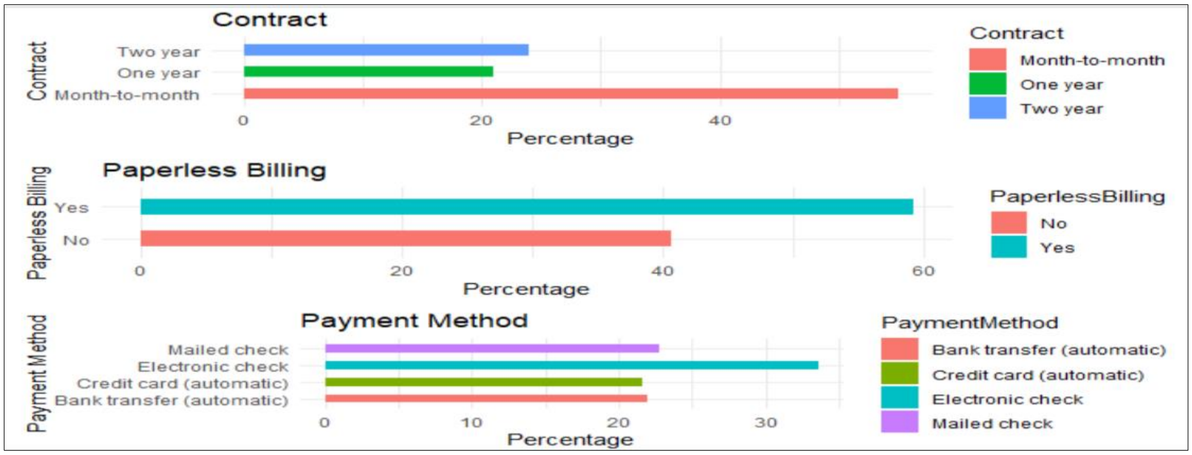


Fig 2.1

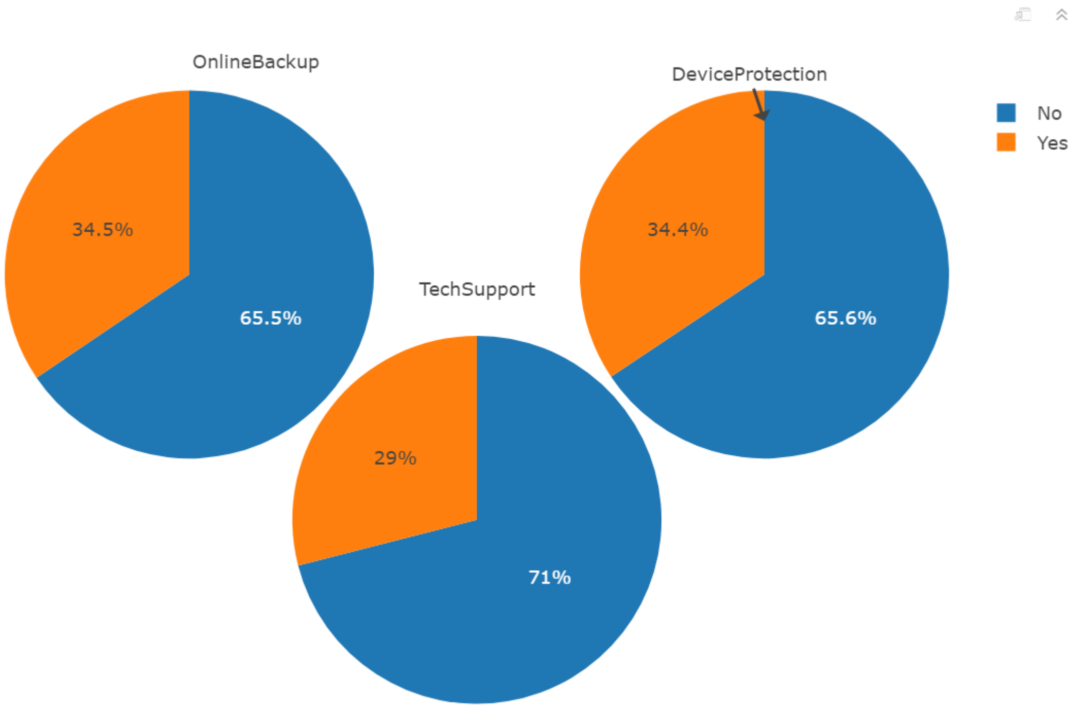


Fig 2.2

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.

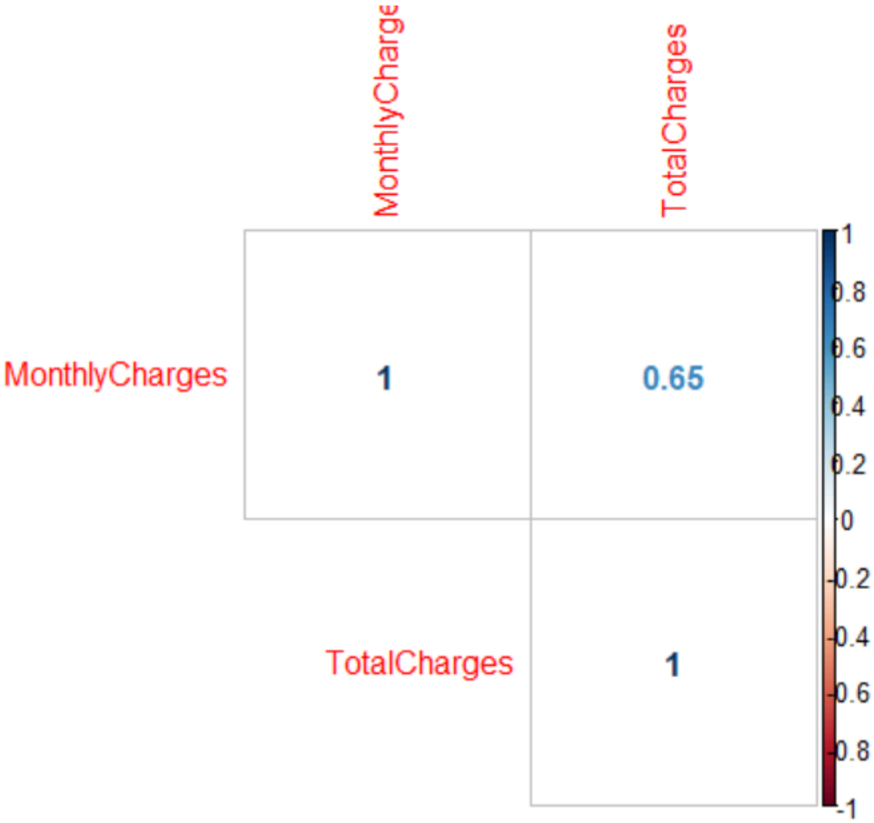
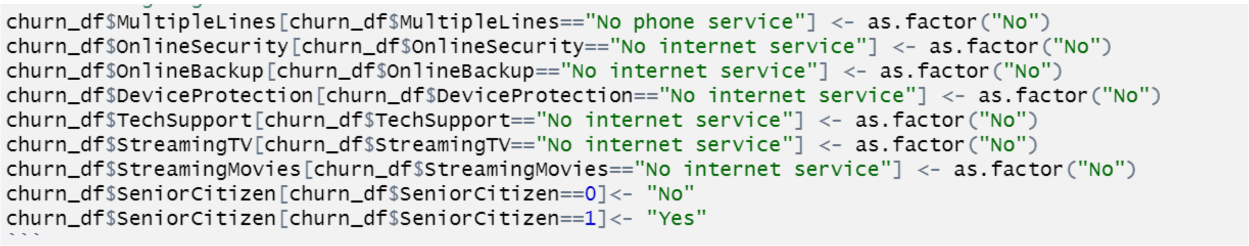


Fig 3.1

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.

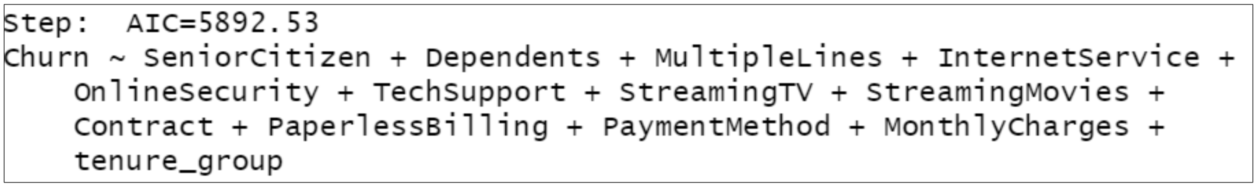


```
churn_df$MultipleLines[churn_df$MultipleLines=="No phone service"] <- as.factor("No")
churn_df$OnlineSecurity[churn_df$OnlineSecurity=="No internet service"] <- as.factor("No")
churn_df$OnlineBackup[churn_df$OnlineBackup=="No internet service"] <- as.factor("No")
churn_df$DeviceProtection[churn_df$DeviceProtection=="No 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"
```

Fig 3.2

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.



```
Step:  AIC=5892.53
Churn ~ SeniorCitizen + Dependents + MultipleLines + InternetService +
    OnlineSecurity + TechSupport + StreamingTV + StreamingMovies +
    Contract + PaperlessBilling + PaymentMethod + MonthlyCharges +
    tenure_group
```

Fig 3.3

To 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.

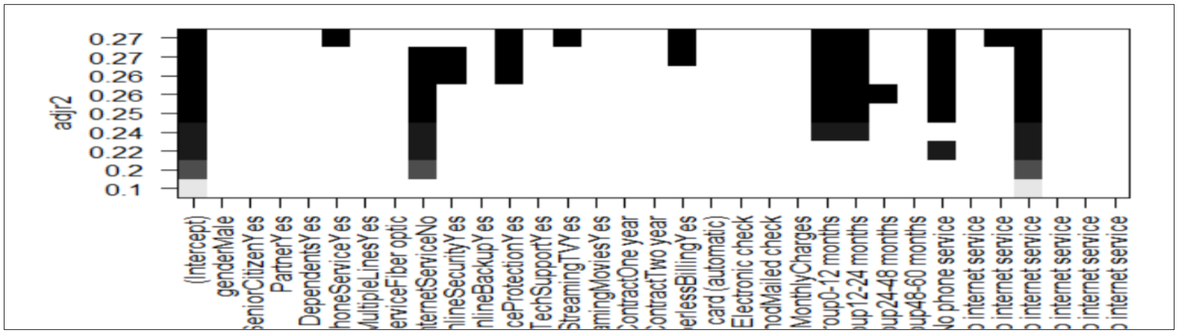


Fig 3.4

IV. Data Mining Techniques and Implementation

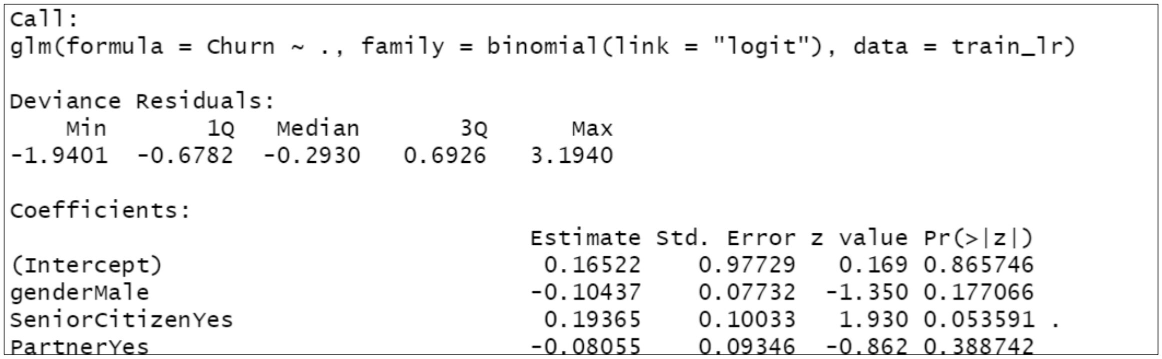
To begin with data mining, we will be using supervised methods of Classification and Prediction. Classification in data mining is the method that is used to classify each record in the dataset into a predefined class. We will be using various classification methods on the telco dataset. We will start 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 $\beta_{0,}$ β_{1}…. β_q for all the predictors in the logistic model.

$$\log(\text{odds}) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_q x_q$$

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.



```
Call:
glm(formula = Churn ~ ., family = binomial(link = "logit"), data = train_lr)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9401  -0.6782  -0.2930   0.6926   3.1940

Coefficients:
                                    Estimate Std. Error z value Pr(>|z|)
(Intercept)                          0.16522    0.97729   0.169 0.865746
genderMale                          -0.10437    0.07732  -1.350 0.177066
SeniorCitizenYes                     0.19365    0.10033   1.930 0.053591 .
PartnerYes                          -0.08055    0.09346  -0.862 0.388742
```

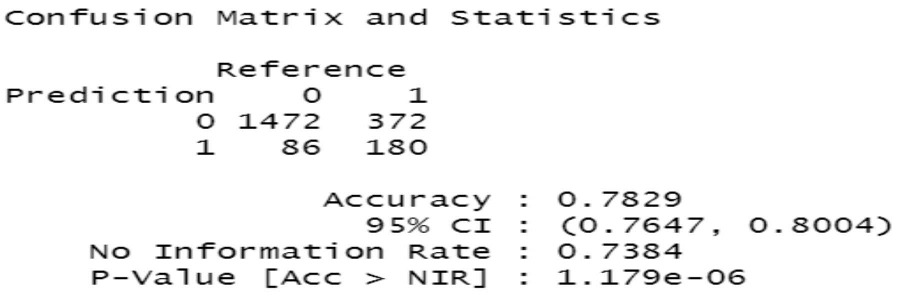
Fig 4.1.1

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.



```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1472  372
         1   86  180

               Accuracy : 0.7829
                 95% CI : (0.7647, 0.8004)
    No Information Rate : 0.7384
    P-Value [Acc > NIR] : 1.179e-06
```

Fig 4.2.1

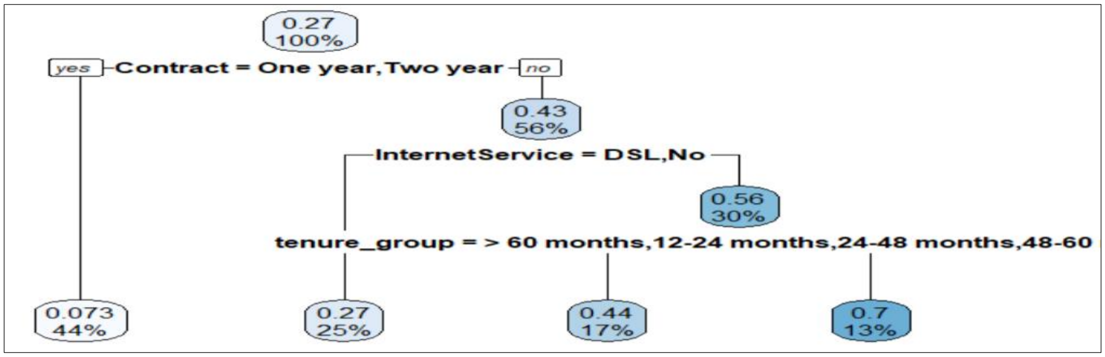

0.27
100%
yes
Contract = One year,Two year
no
0.43
56%
InternetService = DSL,No
0.56
30%
tenure_group = > 60 months,12-24 months,24-48 months,48-60
0.073
44%
0.27
25%
0.44
17%
0.7
13%

Fig 4.2.2

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.

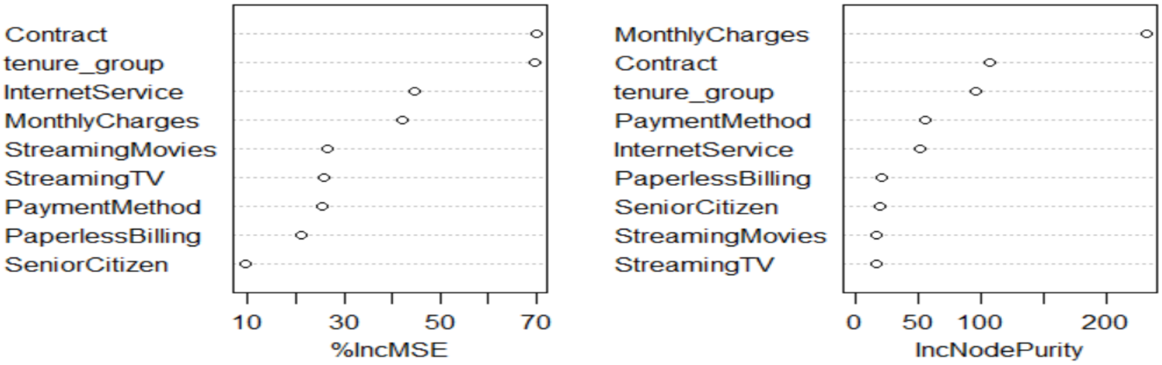
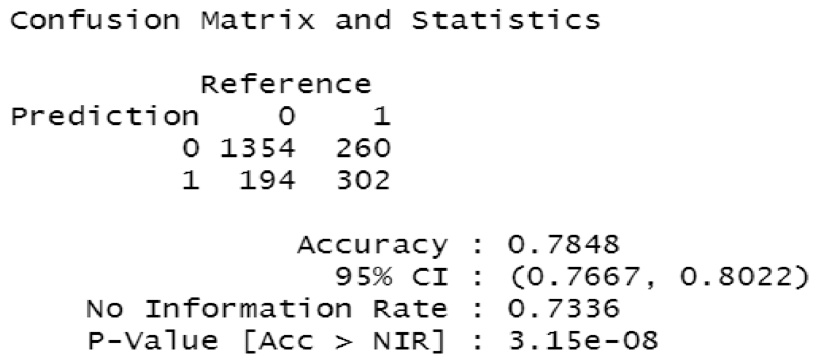


Fig 4.3.1



```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1354  260
         1  194  302

               Accuracy : 0.7848
                 95% CI : (0.7667, 0.8022)
    No Information Rate : 0.7336
    P-Value [Acc > NIR] : 3.15e-08
```

Fig 4.3.2

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 by iterating the value if k from 1 to 65. 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.

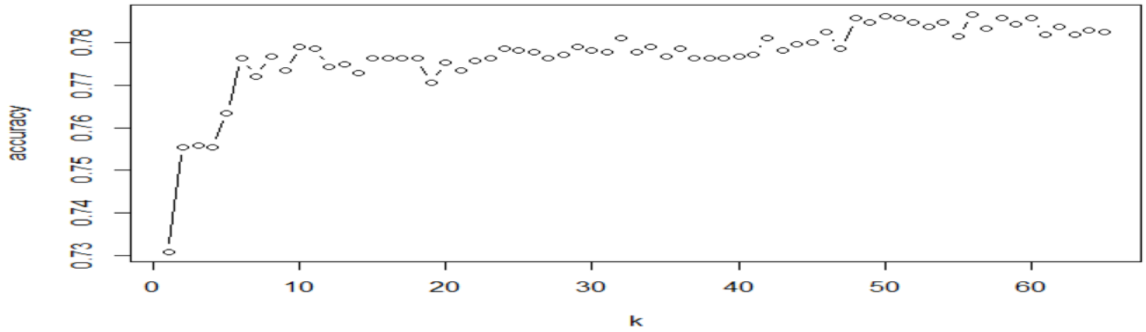
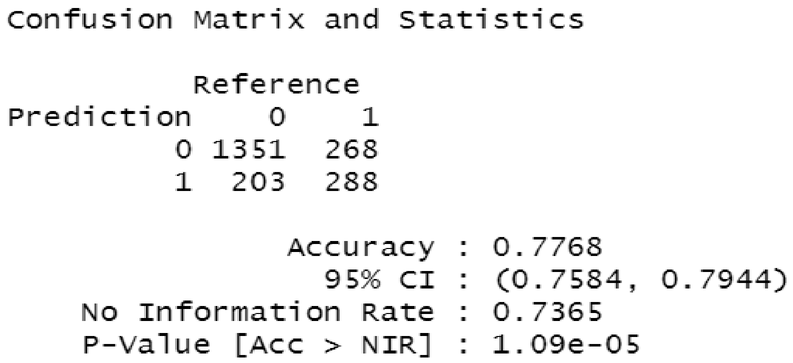


Fig 4.4.1



```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1351  268
         1  203  288

               Accuracy : 0.7768
                 95% CI : (0.7584, 0.7944)
    No Information Rate : 0.7365
    P-Value [Acc > NIR] : 1.09e-05
```

Fig 4.4.2

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.

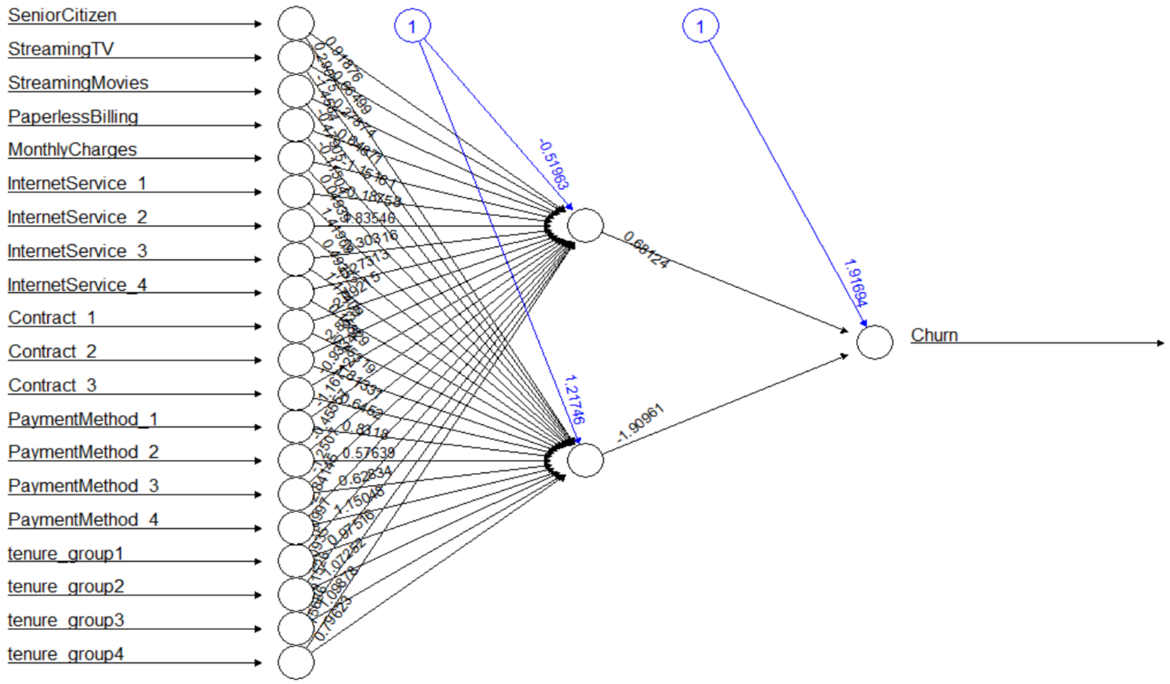


Fig 4.5.1

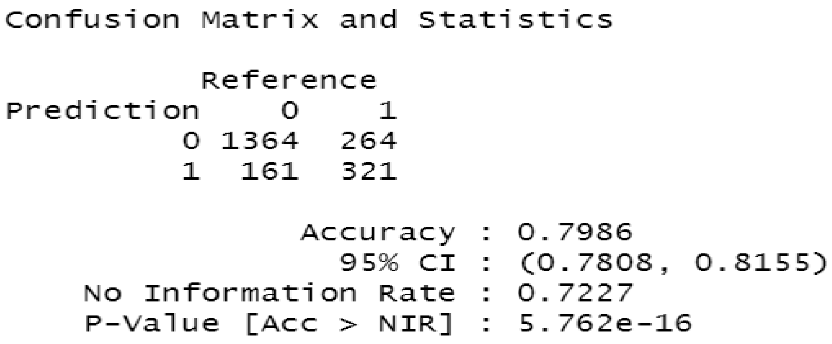
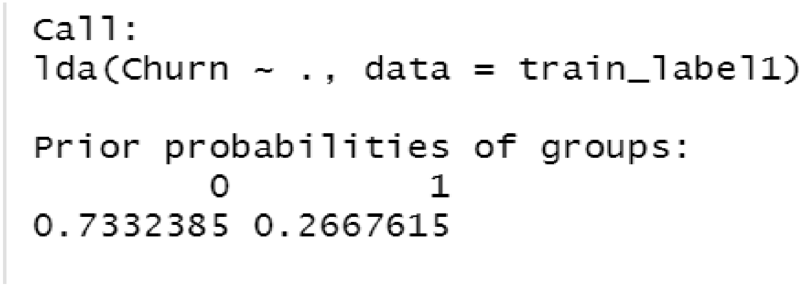

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1364  264
         1  161  321

               Accuracy : 0.7986
                 95% CI : (0.7808, 0.8155)
    No Information Rate : 0.7227
    P-Value [Acc > NIR] : 5.762e-16
```

Fig 4.5.2

6. Linear Discriminant Analysis

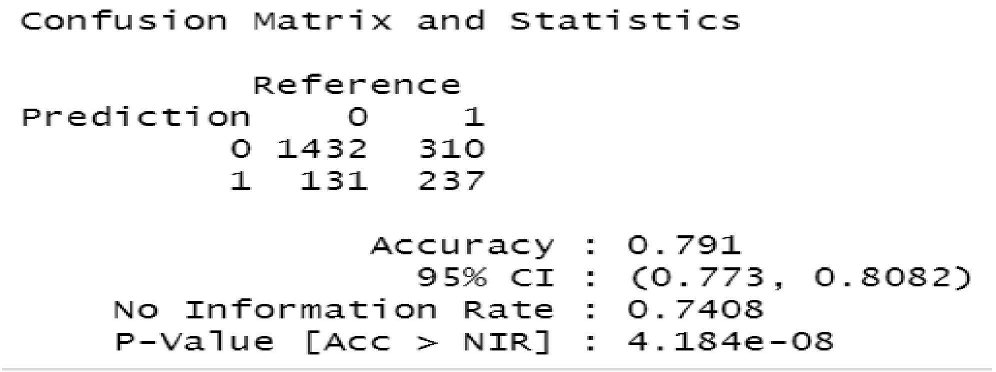
Linear Discriminant Analysis is a model-based approach in which classification is based on the distance of an observation from each class average. For classifying new record, it measures the distance from center of the class known as centroid**.** For measuring the distance statistical or

Mahalanobis distance is used. For implementing in R , we demonstrated lda() function available in MASS packages of R to create the model for linear discriminant analysis and found the accuracy for the model. We also We also added default cutoff probability of 0.5.



```
Call:
lda(Churn ~ ., data = train_label1)

Prior probabilities of groups:
        0         1
0.7332385 0.2667615
```

Fig 4.6.1



```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1432  310
         1  131  237

               Accuracy : 0.791
                 95% CI : (0.773, 0.8082)
    No Information Rate : 0.7408
    P-Value [Acc > NIR] : 4.184e-08
```

Fig 4.6.2

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 an efficiency of predicting the classes right for the given validation dataset.

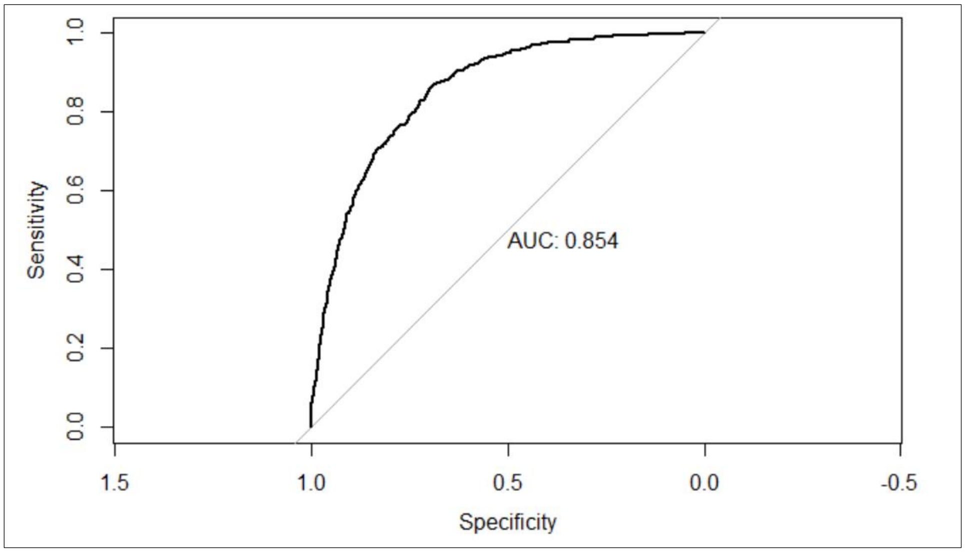


Fig 5.1.1

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	78.29%	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	78.48%	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	78.86%	0.9070	0.4751

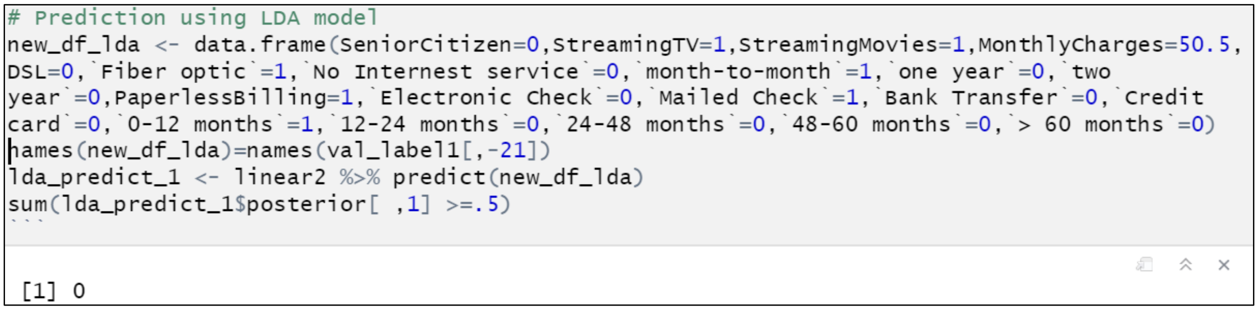
6. Linear Discriminant Analysis:

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	79.10%	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.



```
# Prediction using LDA model
new_df_lda <- data.frame(SeniorCitizen=0,StreamingTV=1,StreamingMovies=1,MonthlyCharges=50.5,
DSL=0,`Fiber optic`=1,`No Internest service`=0,`month-to-month`=1,`one year`=0,`two
year`=0,PaperlessBilling=1,`Electronic Check`=0,`Mailed Check`=1,`Bank Transfer`=0,`Credit
card`=0,`0-12 months`=1,`12-24 months`=0,`24-48 months`=0,`48-60 months`=0,`> 60 months`=0)
names(new_df_lda)=names(val_label1[,-21])
lda_predict_1 <- linear2 %>% predict(new_df_lda)
sum(lda_predict_1$posterior[ ,1] >=.5)
```

```
 [1] 0
```

Fig 5.6.1

VI. Discussion and Recommendation

We have successfully implemented data visualization and exploration using various techniques available in R like omitting records with missing values, implementing various bar plots and pie charts profiling variables. Further, we implemented data wrangling and preprocessing to make database suitable for implementing data mining techniques. Also, we created correlation plot to find correlation between numerical variables and eliminating one which are highly correlated. For selecting the predictors we applied two most popular methods namely 1) stepwise and 2) all subset regression.

Further, for finding the best model for classification we implemented several models, for instance logistic regression model, decision tree model, k-nearest neighbor model, Random Forest, neural net model and linear discriminant analysis. Accuracy of the all models were found to be almost equivalent for the dataset.

By analysis and implementing data mining techniques on telco_customer_churn dataset, we realized that dynamic dataset would give more accuracy instead of using historical dataset because dynamic dataset would give us day to day changing of the statistics related to customer churns.

VII. Summary

Throughout the project we came across several things :-

1) Accuracy is almost equivalent for all the models.

2) Most important predictors for this dataset are : Tenure , Contract , Internet Service and PaperlessBilling.

3) Customer between the tenure of 0-12 months and 12-24 months are more likely to leave the network(Churn)

4) There is no strong relation between Churn and Gender , Multiple Lines , Device Protection , Tech Support , Online Security and Online Backup . It has no impact on customer leaving or staying on the network

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)

# 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+Int
ernetService+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+Paperle
ssBilling+PaymentMethod+tenure_group+MonthlyCharges,train_lr,parms=list(split=c("informat
ion","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_grou
p, 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,Paperles
sBilling,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)


# RMSE for all models
library(Metrics)
rmse1 <- rmse(test_lr_predict1,test_lr$Churn)
rmse2 <- rmse(test_lr_predict2,test_lr$Churn)
rmse3 <- rmse(lr_predict3,test_lr$Churn)
rmse4 <- rmse(p1,test_lr$Churn)
rmse5 <- rmse(p2,test_lr$Churn)
rmse6 <- rmse(rf.predict_1,valid_rf$Churn)
rmse7 <- rmse(rf.predict_3,valid_rf$Churn)
rmse8 <- rmse(nn_predict1,val1$Churn)
rmse9 <- rmse(nn_predict2,val2$Churn)
rmse10 <- rmse(mean(predictions1$class==val_label1$Churn)
,val2$Churn)
rmse11 <- rmse(mean(predictions2$class==val_label2$Churn)
,val2$Churn)
```

```
rmse_df <-
data.frame("Model"=c("LogisticModel1","LogisticModel2","LogisticModel3","DecisionTree1",
"DecisionTree2","RandomForest1","RandomForest2","NeuralNetwork1","NeuralNetwork2","L
DA1","LDA2"),
"RMSE"=c(rmse1,rmse2,rmse3,rmse4,rmse5,rmse6,rmse7,rmse8,rmse9,rmse10,rmse11))
rmse_df

# Prediction using Neural Network
new_df_nn <-
data.frame(SeniorCitizen=0,StreamingTV=1,StreamingMovies=1,PaperlessBilling=1,MonthlyC
harges=50.5,InternetService_1=0,InternetService_2=1,InternetService_3=0,InternetService_4=0,
Contract_1=1,Contract_2=0,Contract_3=0,PaymentMethod_1=0,PaymentMethod_2=1,Payment
Method_3=0,PaymentMethod_4=0,tenure_group1=1,tenure_group2=0,tenure_group3=0,tenure_
group4=0)

nn_predict_1<-predict(nn1,new_df)
nn_pred1 <- factor(ifelse(nn_predict_1 >= 0.5, "1", "0"))
nn_pred1

# Prediction using LDA model
new_df_lda <-
data.frame(SeniorCitizen=0,StreamingTV=1,StreamingMovies=1,MonthlyCharges=50.5,DSL=0
,`Fiber optic`=1,`No Internest service`=0,`month-to-month`=1,`one year`=0,`two
year`=0,PaperlessBilling=1,`Electronic Check`=0,`Mailed Check`=1,`Bank Transfer`=0,`Credit
card`=0,`0-12 months`=1,`12-24 months`=0,`24-48 months`=0,`48-60 months`=0,`> 60
months`=0)
names(new_df_lda)=names(val_label1[,-21])
lda_predict_1 <- linear2 %>% predict(new_df_lda)
sum(lda_predict_1$posterior[ ,1] >=.5)
```