**Machine Learning Assignment-2**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Prepared By** | **SID** | **TAID** |
| **Name** | Rajib Mandal | 118310 | TA17002 |
| **Submitted Date** | 19-FEB-2018 | | |

Table of Contents

[1. Data Set Information 3](#_Toc506759869)

[**2.** Attribute Information 3](#_Toc506759870)

[3. Project Objective 3](#_Toc506759871)

[4. Abstract 3](#_Toc506759872)

[4.1 Check the structure of the Data Set 4](#_Toc506759873)

[4.2 VISUALIZATION OF Data 4](#_Toc506759874)

[4.3 Data Preparation 7](#_Toc506759875)

[4.4 Standardize All The Continuous Field 9](#_Toc506759876)

[5. Preparing Training and Test data 10](#_Toc506759877)

[6. Support Vector Machine (SVM) 10](#_Toc506759878)

[6.1 Svm Model Building 10](#_Toc506759879)

[6.2 Accuracy Of The Model 10](#_Toc506759880)

[6.3 ROC CURVE & performance of the model 11](#_Toc506759881)

[6.4 SENSIVITY and SPECIFICITY in SVM 12](#_Toc506759882)

[6.5 Model evaluation parameters – 12](#_Toc506759883)

[7. KNN 13](#_Toc506759884)

[7.1 KNN Model Building 13](#_Toc506759885)

[7.2 Accuracy Of The Model 13](#_Toc506759886)

[7.3 ROC CURVE & performance of the model 14](#_Toc506759887)

[7.4 SENSIVITY and SPECIFICITY in KNN 14](#_Toc506759888)

[7.5 Model evaluation parameters – 15](#_Toc506759889)

[8. Advance Neural Network 15](#_Toc506759890)

[8.1 Building A Model By Using Advance Neural Network (ANN) 15](#_Toc506759891)

[8.2 Accuracy Of The ANN Model 16](#_Toc506759892)

[8.3 SENSIVITY and SPECIFICITY in ANN 16](#_Toc506759893)

[8.4 ROC CURVE & performance of the model 16](#_Toc506759894)

[8.5 Model evaluation parameters – 17](#_Toc506759895)

[9. Conclusion 17](#_Toc506759896)

[10. ANNEXTURE 18](#_Toc506759897)

# Data Set Information

The Telecom customer data set provides information to help to predict behavior to retain customers. There are 7043 records in the dataset.

It is very important for a business to ensure that their churn rate is as low as possible. Higher churn rate directly impacts the reduction in the earning of the business and increased cost of sales & marketing.

# Attribute Information

|  |  |  |
| --- | --- | --- |
| Name | Type | Data Type |
| customerID | Categorical | Text |
| Gender | Categorical | Text |
| SeniorCitizen | Categorical | integer |
| Partner | Categorical | Text |
| Dependents | Categorical | Text |
| Tenure | Continuous | integer |
| PhoneService | Categorical | Text |
| MultipleLines | Categorical | Text |
| InternetService | Categorical | Text |
| OnlineSecurity | Categorical | Text |
| OnlineBackup | Categorical | Text |
| DeviceProtection | Categorical | Text |
| TechSupport | Categorical | Text |
| StreamingTV | Categorical | Text |
| StreamingMovies | Categorical | Text |
| Contract | Categorical | Text |
| PaperlessBilling | Categorical | Text |
| PaymentMethod | Categorical | Text |
| MonthlyCharges | Continuous | Numeric |
| TotalCharges | Continuous | Numeric |
| Churn | Categorical | Text |

There are 3 numerical variables and 18 categorical variables. The detail is given below.

# Project Objective

The dataset contains 7043 customer data. The goal is to develop a machine learning model to predict behavior of customer to retain customers.

# Abstract

To predict churn rate of customers, we have analysed data using 3 different models: **KNN**, Support **Vector Machine(SVM) & Artificial Numerical Network(ANN).** Overall, **Artificial Numerical Network(ANN)** proved to be the best in predicting the churn based on the customer profiles with an 80.59 % accuracy rate.

## Check the structure of the Data Set

telecomcustomerdataframe<-read.csv("Telecom Customer Data.csv",header=TRUE, na.strings=c("",'NA'))

str(telecomcustomerdataframe)

dim(telecomcustomerdataframe)

summary(telecomcustomerdataframe)

From the output, it is seen that there are 3 continuous variables,17 categorical variables excluding the variable that predicts, “churn”, which will be used as the response variable the models are attempting to predict.

There are 11 missing values in the dataset.

Looking at the data structure, some data columns need recoding. For instance, changing values from “No phone service” and “No internet service” to “No”, for consistency

## VISUALIZATION OF Data

We found there is no outlier for the following continuous variables by exploring boxplot

* Tenure
* MonthlyCharges
* TotalCharges

The code snippet is given below

boxplot(telecomcustomerdataframe$tenure,col = "bisque",xlab="Tenure")

boxplot(telecomcustomerdataframe$MonthlyCharges,col = "bisque",xlab="Monthly Charges")

boxplot(telecomcustomerdataframe$TotalCharges,col = "bisque",xlab="Total Charges")

The output for box plot is given below.



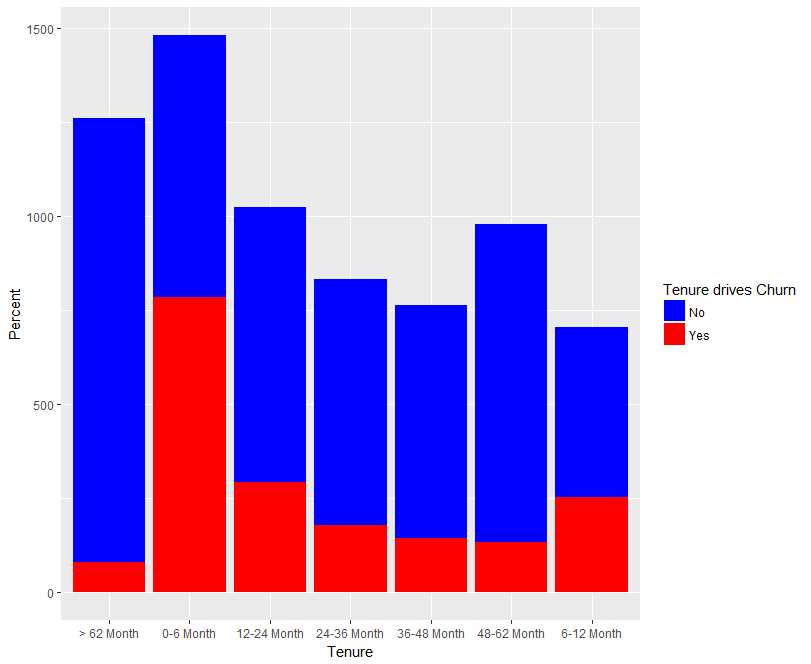
* From the Tenure box plot we identified that median is very close to 30 and it left skewed. But we could not see any outlier on the tenure variable.
* From the Monthly Charges box plot we identified that median is very close to 70 and it right skewed. But we could not see any outlier on the monthly charges variable.
* From the Total Charges box plot we identified that median is very close to 1800 and it left skewed. But we could not see any outlier on the tenure variable.

**Analysis on numeric data**

We explored relationship between customer churn with following numeric variables

* Tenure
* MonthlyCharges
* TotalCharges

The corresponding bar plot is given below



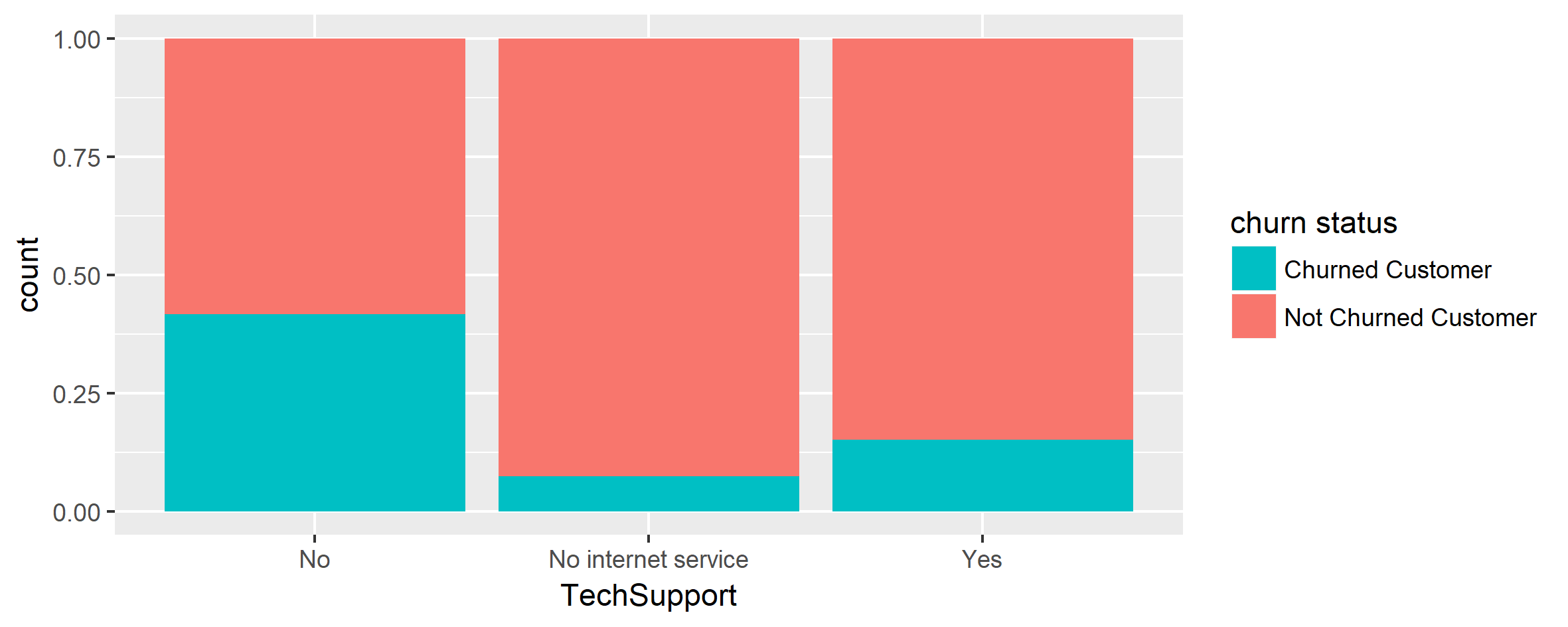
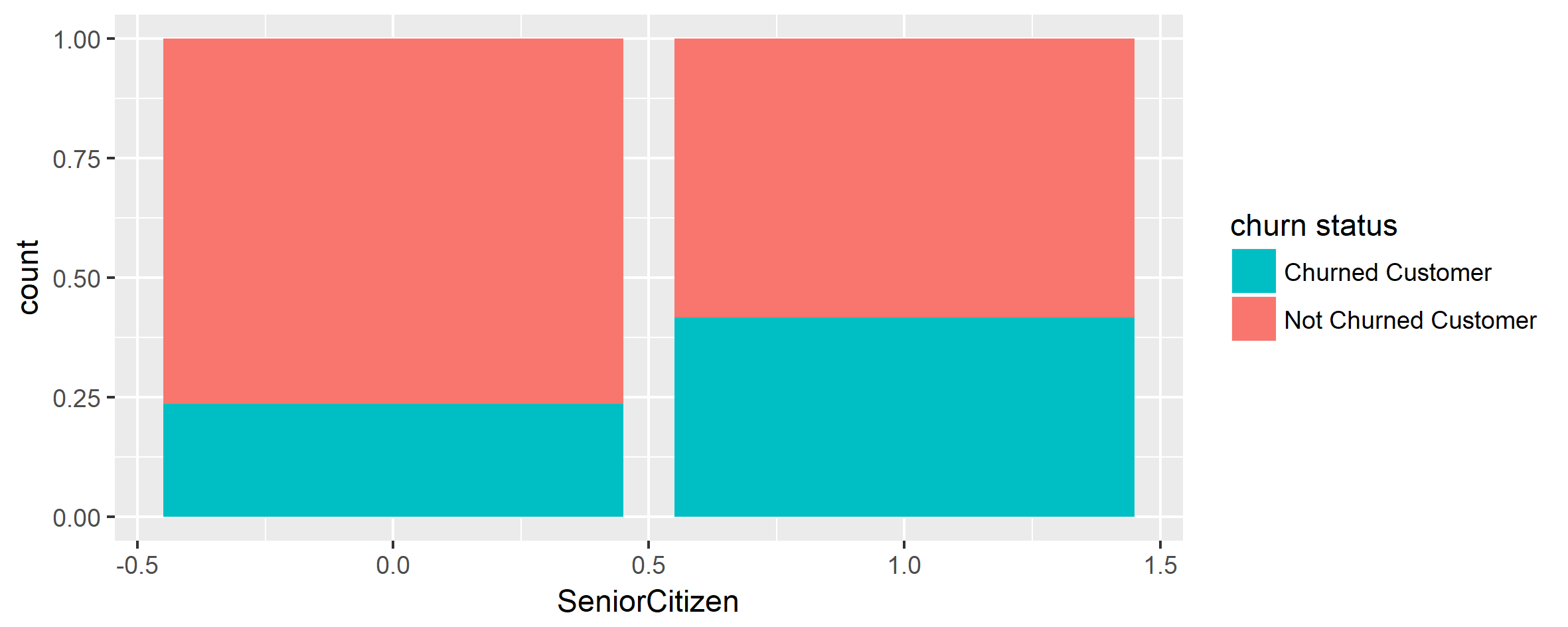
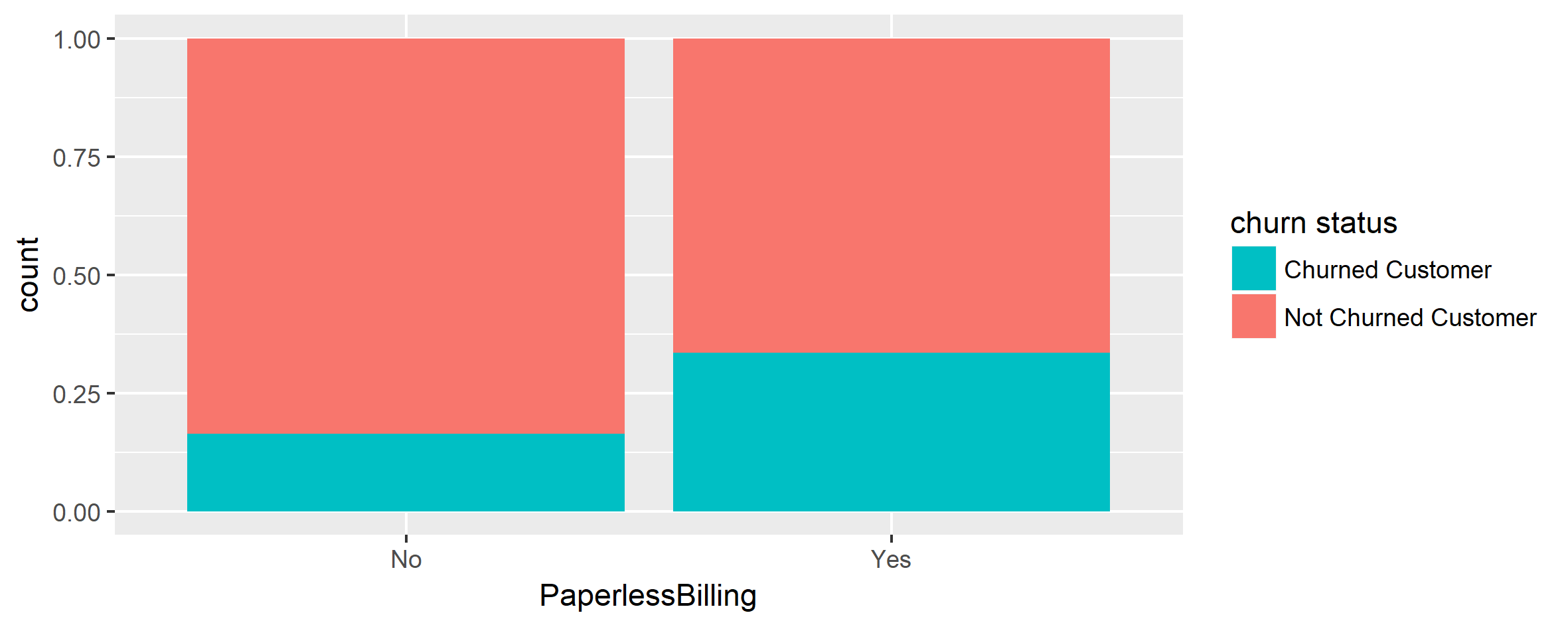
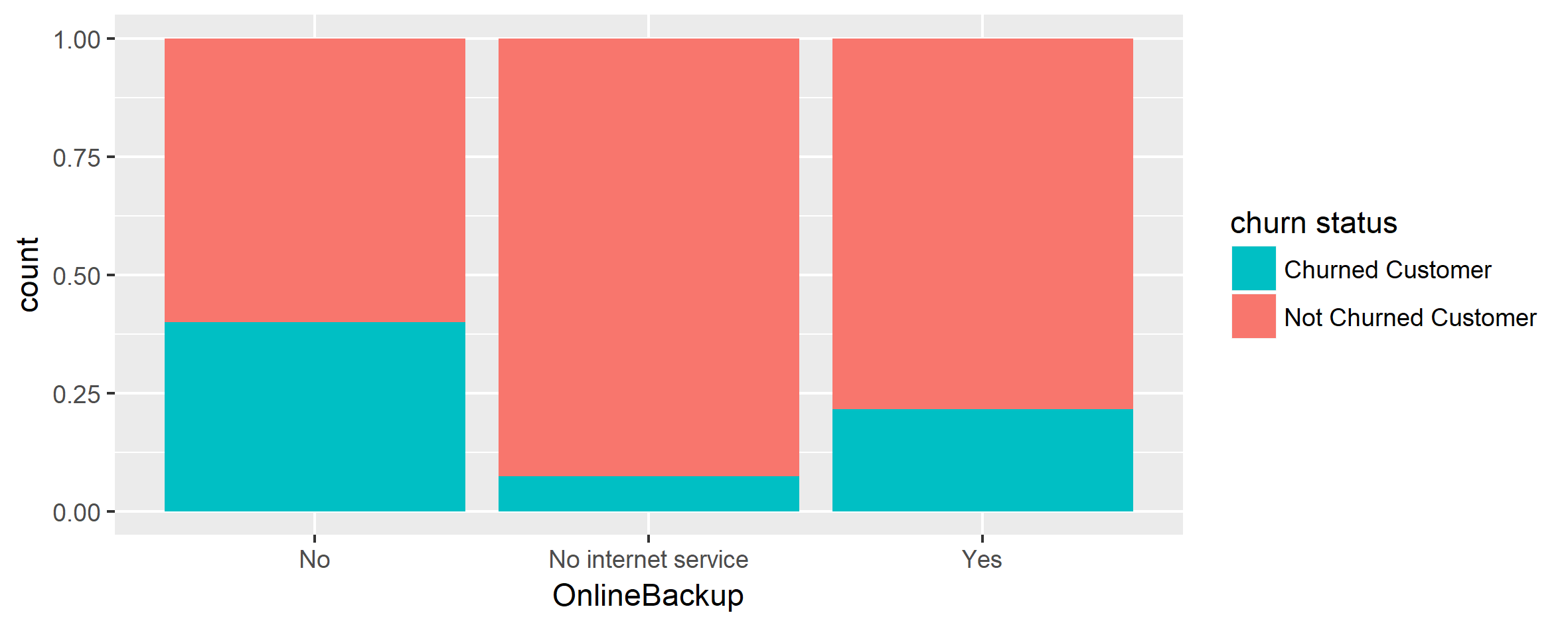
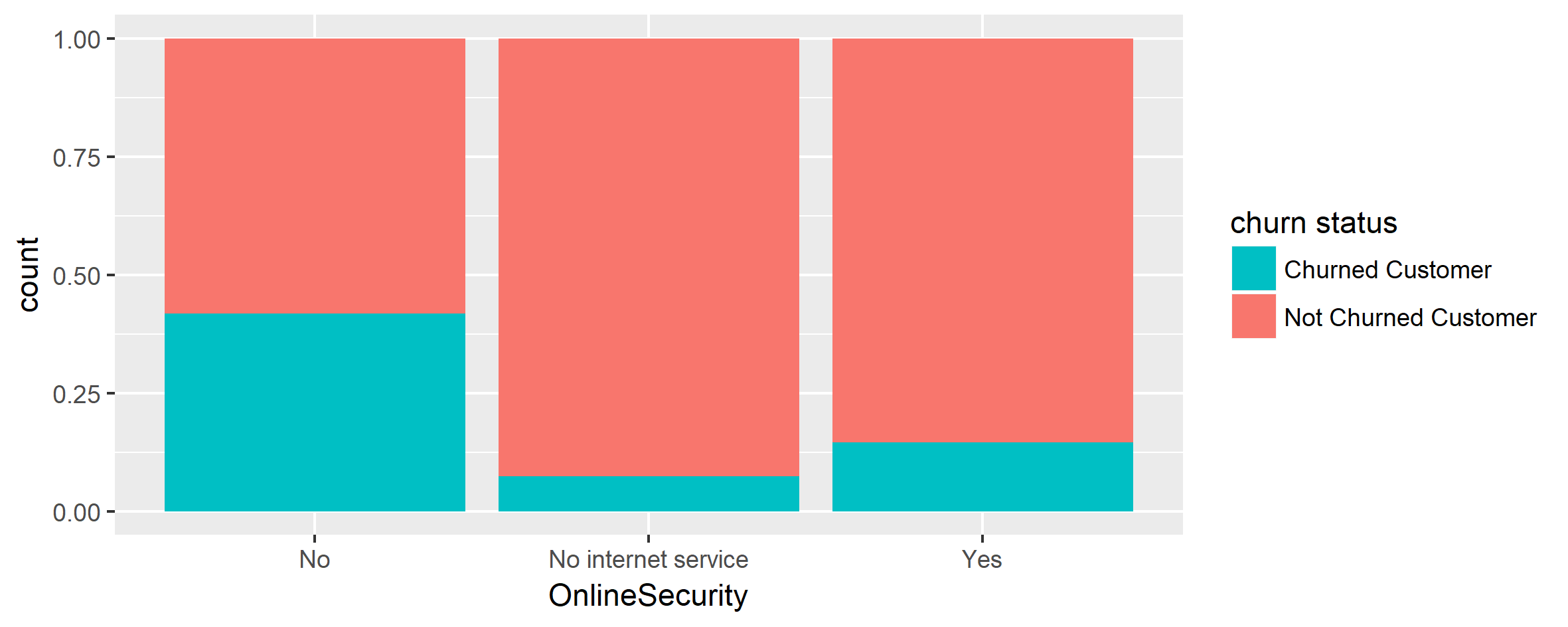
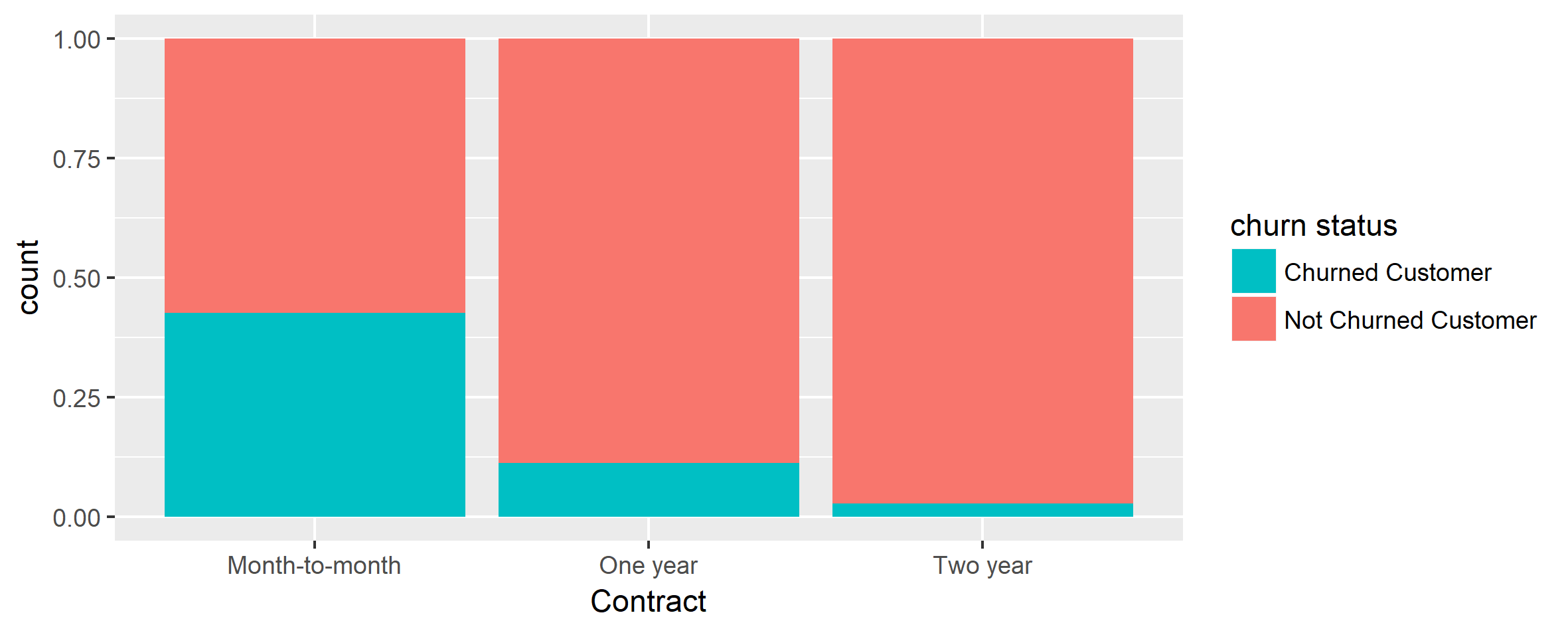
* Customer who is churing, their tenure are comparatively lesser as compare to those who are not churning

The code snippet is given below

ggplot() + geom\_bar(data = telecomcustomerdataframe,

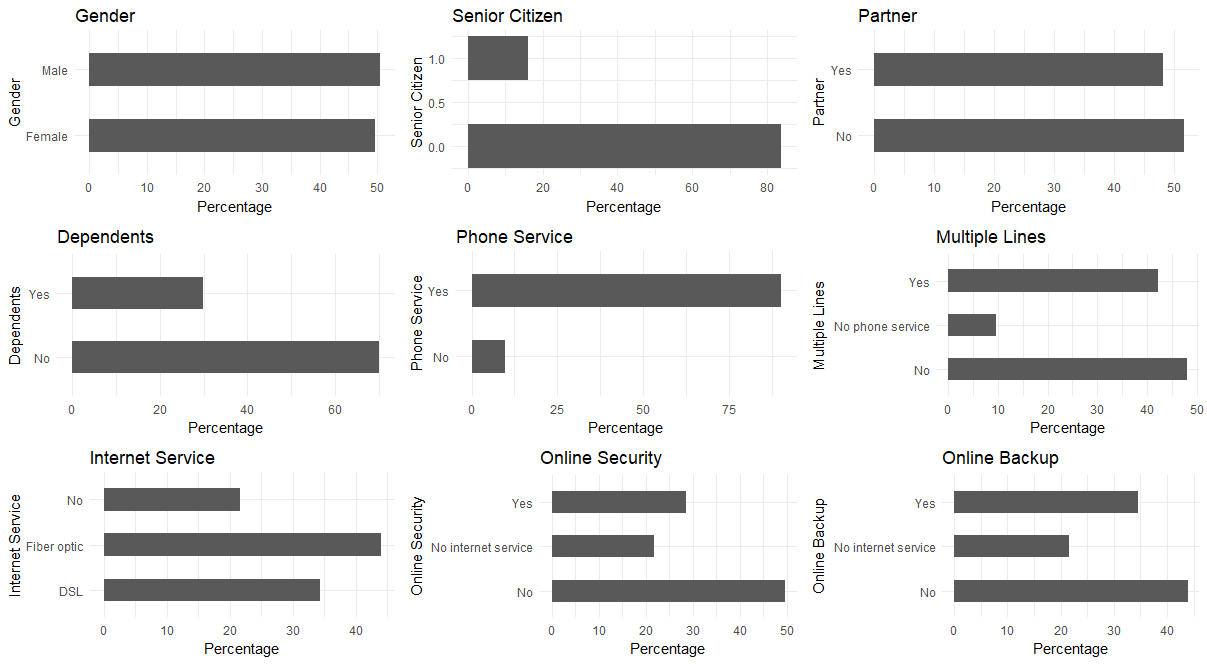
aes(x = factor(telecomcustomerdataframe$tenureinterval), fill =factor(telecomcustomerdataframe$Churn)), position = "stack") +scale\_x\_discrete("Tenure") + scale\_y\_continuous("Percent") + guides(fill=guide\_legend(title="Tenure drives Churn")) + scale\_fill\_manual(values=c("blue","red"))

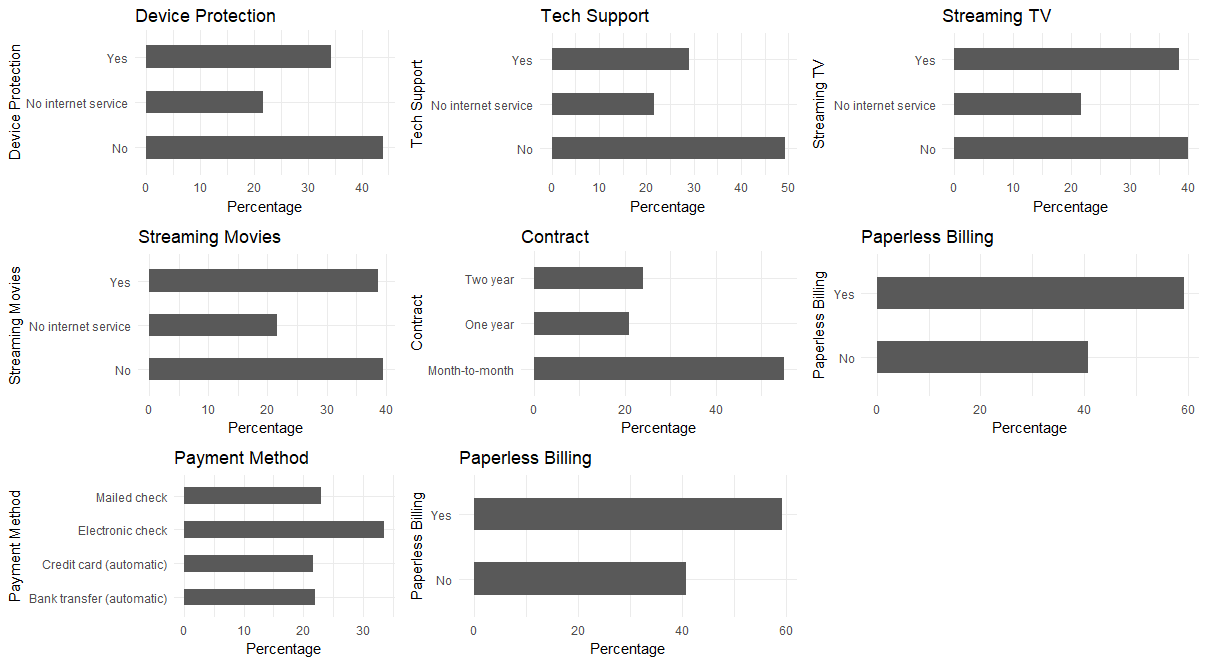
Some Insights from the data



* Monthly plan customer the most likely to churn
* Person having no dependents are churning more
* Multiple lines have no impact on churning
* Person having no online backup is more likely to churn
* Person using no online security is more likely to churn
* Senior Citizens are churning more
* No tech support, Streaming movies, Streaming TV more likely to churn

From the below graph, we can see that all the categorical variables seem to have a reasonably broad distribution, therefore, all of them will be kept for the further analysis.





## Data Preparation

There are 11 missing values in Total Charges field.We impute missing value with mean of the TotalCharges.

The code snippet is given below

telecomcustomerdataframe$TotalCharges[is.na(telecomcustomerdataframe$TotalCharges)]<-mean(telecomcustomerdataframe$TotalCharges, na.rm = TRUE)

For consistency we have changed values from “No phone service” and “No internet service” for the following 7 fields.

* MultipleLines
* OnlineSecurity
* OnlineBackup
* DeviceProtection
* TechSupport
* StreamingTV
* StreamingMovies

The code snippet is given below

telecomcustomerdataframe$MultipleLines[telecomcustomerdataframe$MultipleLines=="No phone service"] <- "No"

telecomcustomerdataframe$OnlineSecurity[telecomcustomerdataframe$OnlineSecurity=="No internet service"] <- "No"

telecomcustomerdataframe$OnlineBackup[telecomcustomerdataframe$OnlineBackup=="No internet service"] <- "No"

telecomcustomerdataframe$DeviceProtection[telecomcustomerdataframe$DeviceProtection=="No internet service"] <- "No"

telecomcustomerdataframe$TechSupport[telecomcustomerdataframe$TechSupport=="No internet service"] <- "No"

telecomcustomerdataframe$StreamingTV[telecomcustomerdataframe$StreamingTV=="No internet service"] <- "No"

telecomcustomerdataframe$StreamingMovies[telecomcustomerdataframe$StreamingMovies=="No internet service"] <- "No"

We convert following categorical variables into binary 1 and 0 since KNN works only in numeric values.

* + gender
  + Partner
  + Dependents
  + PhoneService
  + MultipleLines
  + OnlineSecurity
  + OnlineBackup
  + DeviceProtection
  + TechSupport
  + StreamingTV
  + StreamingMovies
  + PaperlessBilling
  + Churn

telecomcustomerdataframe$gender<-ifelse(telecomcustomerdataframe$gender=='Male',1,0)

telecomcustomerdataframe$Partner<-ifelse(telecomcustomerdataframe$Partner=='Yes',1,0)

telecomcustomerdataframe$Dependents<-ifelse(telecomcustomerdataframe$Dependents=='Yes',1,0)

telecomcustomerdataframe$PhoneService<-ifelse(telecomcustomerdataframe$PhoneService=='Yes',1,0)

telecomcustomerdataframe$MultipleLines<-ifelse(telecomcustomerdataframe$MultipleLines=='Yes',1,0)

telecomcustomerdataframe$OnlineSecurity<-ifelse(telecomcustomerdataframe$OnlineSecurity=='Yes',1,0)

telecomcustomerdataframe$OnlineBackup<-ifelse(telecomcustomerdataframe$OnlineBackup=='Yes',1,0)

telecomcustomerdataframe$DeviceProtection<-ifelse(telecomcustomerdataframe$DeviceProtection=='Yes',1,0)

telecomcustomerdataframe$TechSupport<-ifelse(telecomcustomerdataframe$TechSupport=='Yes',1,0)

telecomcustomerdataframe$StreamingTV<-ifelse(telecomcustomerdataframe$StreamingTV=='Yes',1,0)

telecomcustomerdataframe$StreamingMovies<-ifelse(telecomcustomerdataframe$StreamingMovies=='Yes',1,0)

telecomcustomerdataframe$PaperlessBilling<-ifelse(telecomcustomerdataframe$PaperlessBilling=='Yes',1,0)

telecomcustomerdataframe$Churn<-ifelse(telecomcustomerdataframe$Churn=='Yes',1,0)

We have transformed following variables with dummy variables (binary) since the numbers provided in the following variable are categorical in nature and hence transformed by dummy variable (binary)

* InternetService
* Contract
* PaymentMethod

The code snippet is given below

for(level in unique(telecomcustomerdataframe$InternetService)){

telecomcustomerdataframe[paste("InternetService", gsub("-","\_", gsub(" ","\_",level, fixed=TRUE), fixed=TRUE), sep = "\_")] <- ifelse(telecomcustomerdataframe$InternetService == level, 1, 0)

}

for(level in unique(telecomcustomerdataframe$Contract)){

telecomcustomerdataframe[paste("Contract", gsub("-","\_", gsub(" ","\_",level, fixed=TRUE), fixed=TRUE), sep = "\_")] <- ifelse(telecomcustomerdataframe$Contract == level, 1, 0)

}

for(level in unique(telecomcustomerdataframe$PaymentMethod)){

telecomcustomerdataframe[paste("PaymentMethod", gsub("-","\_", gsub(" ","\_",level, fixed=TRUE), fixed=TRUE), sep = "\_")] <- ifelse(telecomcustomerdataframe$PaymentMethod == level, 1, 0)

}

We have removed following fields from the data set since

* customerID
* InternetService
* Contract
* PaymentMethod

telecomcustomerdataframe$customerID<-NULL

telecomcustomerdataframe$InternetService<-NULL

telecomcustomerdataframe$Contract<-NULL

telecomcustomerdataframe$PaymentMethod<-NULL

telecomcustomerdataframe$tenure<-NULL

## Standardize All The Continuous Field

We have standardized below two continuous variable into min-max scale.

* TotalCharges
* MonthlyCharges

Below is the code snippet.

minmaxstandard <- function(x)

{

(x-min(x))/(max(x)-min(x))

}

telecomcustomerdataframe$TotalCharges<-as.numeric(minmaxstandard(telecomcustomerdataframe$TotalCharges))

telecomcustomerdataframe$MonthlyCharges <-as.numeric(minmaxstandard(telecomcustomerdataframe$MonthlyCharges))

# Preparing Training and Test data

Create training and test data sets to run the model.   Split 70% of the observations into the training set and the remaining 30% into the test set.

The code snippet is given below

sample <- sample.split(telecomcustomerdataframe$Churn,SplitRatio=0.70)

trainData <- subset(telecomcustomerdataframe,sample==TRUE)

testData <- subset(telecomcustomerdataframe,sample==FALSE)

# Support Vector Machine (SVM)

## Svm Model Building

The model definition is given below

svmmodel <- svm(formula = Churn~.,

data = trainData,

type = 'C-classification',

kernel = 'radial',

cost=0.01,

probability=TRUE

)

summary(svmmodel)

**Model development steps**

* Categorical variables were converted to numeric by creating dummy variables
* Converted the data type of response variable to factor
* Scaled the numerical variables
* Split the data into train and test data in the proportion of 70 and 30 percent ensuring equal proportion of class labels in both train and test data set
* Various evaluation metrics like accuracy, sensitivity and specificity for the model is checked
* Then radial kernel is selected and model is tuned with various parameters of cost and sigma
* The best cost and sigma values are taken and model was created using it

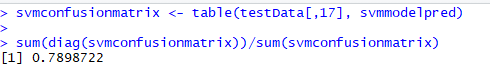
## Accuracy Of The Model

Below is the code snippet for finding out accuracy of the SVM model

svmconfusionmatrix <- table(testData[,17], svmmodelpred)

sum(diag(svmconfusionmatrix))/sum(svmconfusionmatrix)

The sample output is given below



The accuracy of the SVM model is 78.98%

## ROC CURVE & performance of the model

Below is the code snippet

modelprediction<-prediction(as.numeric(svmmodelpred),as.numeric(testData$Churn))

modelprediction

modelperformance<-performance(modelprediction,"tpr","fpr")

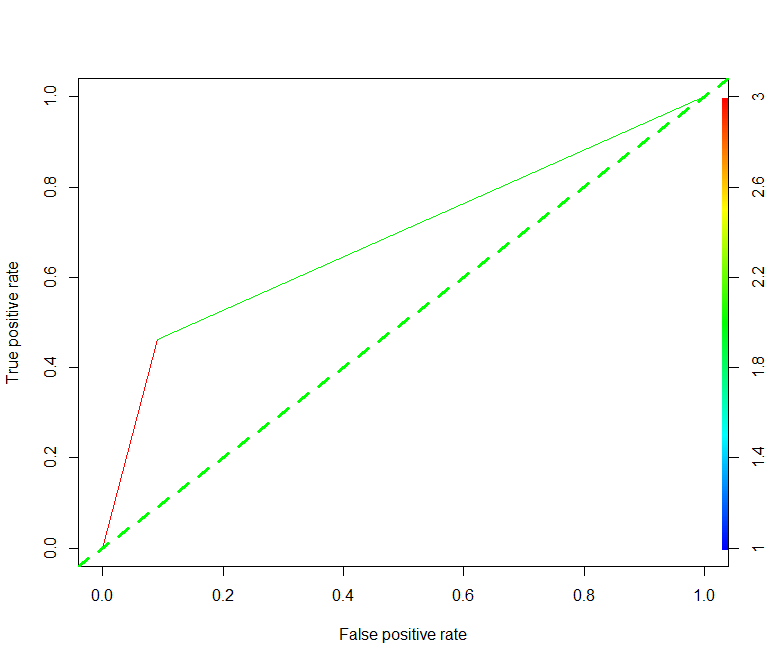
plot(modelperformance)

abline(a=0,b=1,lwd=2,lty=2)

svmaccrcy<-performance(modelprediction,"auc")

svmaccrcy

The area under the curve (AUC) is used to summarize the performance of the model, and it is currently about 0.6850. This is a pretty high value, which is an indicator of a good performance.



## SENSIVITY and SPECIFICITY in SVM

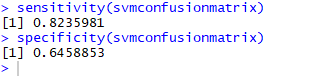
We are finding the recall and precision of SVM and below is the observation :

Here is the code snippet

sensitivity(svmconfusionmatrix)

specificity(svmconfusionmatrix)

The output is given below



The sensitivity of the model is 82.23% and specificity of the model is 64.58%

## Model evaluation parameters –

Here is the summary of the SVM model

|  |  |
| --- | --- |
| **Threshold value** | **Values (Numeric)** |
| Accuracy | 78.98% |
| Sensitivity | 82.23% |
| Specificity | 64.85% |
| AUC(ROC) | **68.50%** |
| cost | **0.01** |

# KNN

## KNN Model Building

The model definition is given below

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

set.seed(3333)

knnfit <- train(Churn ~., data = trainData, method = "knn",

trControl=trctrl,

preProcess = c("center", "scale"),

tuneLength = 10)

knnfit

Accuracy was used to select the optimal model using the largest value.

The final value used for the model was k = 23.

knnmodel = knn(train = trainData[,-17],

test = testData[,-17],

cl = trainData[,17],

k = 23,

prob=TRUE,

use.all = TRUE

)

summary(knnmodel)

**Model Development Steps –**

* Categorical variables were converted to numeric by creating dummy variables
* Converted the data type of response variable to factor
* Scaled the numerical variables
* Split the data into train and test data in the proportion of 70 and 30 percent ensuring equal proportion of class labels in both train and test data set
* After repeated cross validation, model suggested 23 as optimal value of K

## Accuracy Of The Model

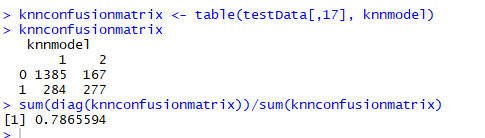
Below is the code snippet for finding out accuracy of the KNN model

knnconfusionmatrix <- table(testData[,17], knnmodel)

knnconfusionmatrix

sum(diag(knnconfusionmatrix))/sum(knnconfusionmatrix)

Here is output



The accuracy of the KNN model is 78.65%

## ROC CURVE & performance of the model

Below is the code snippet

knnmodel<-as.numeric(knnmodel)

modelprediction<-prediction(knnmodel,testData$Churn)

modelperformance<-performance(modelprediction,"tpr","fpr")

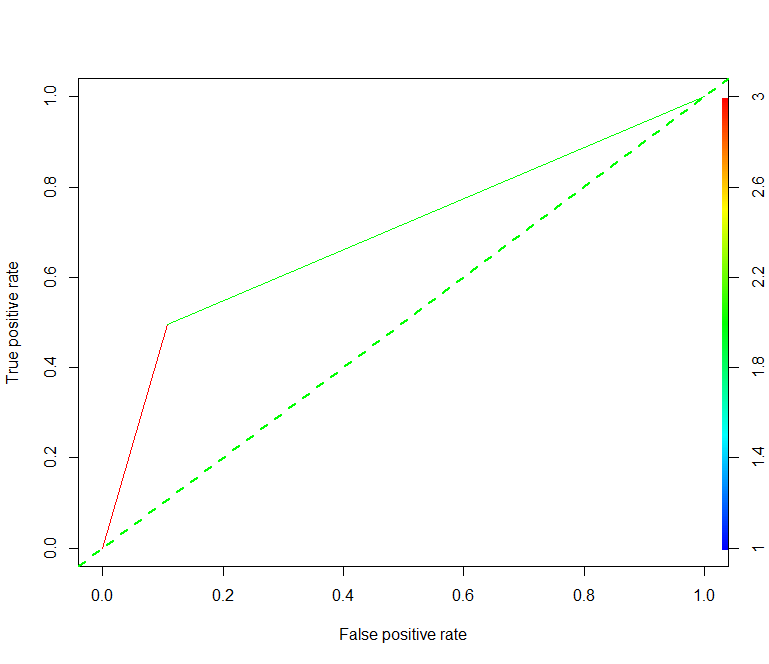
plot(modelperformance,col="blue")

abline(a=0,b=1,lwd=2,lty=2)

knnacrcy<-performance(modelprediction,"auc")

knnacrcy

The area under the curve (AUC) is used to summarize the performance of the model, and it is currently about **0.6850**. This is a pretty high value, which is an indicator of a good performance.



## SENSIVITY and SPECIFICITY in KNN

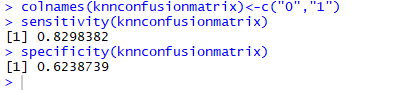
Hereis the code snippet

colnames(knnconfusionmatrix)<-c("0","1")

sensitivity(knnconfusionmatrix)

specificity(knnconfusionmatrix )

Here is the output



## Model evaluation parameters –

Here is the summary of the KNN model

|  |  |
| --- | --- |
| **Threshold value** | **Values (Numeric)** |
| Accuracy | 78.65% |
| Sensitivity | 82.98% |
| Specificity | 62.38% |
| AUC(ROC) | 68.50% |
| K | 23 |

# Advance Neural Network

## Building A Model By Using Advance Neural Network (ANN)

Create training and test data sets to run the model. Split 70% of the observations into the training set and the remaining 30% into the test set.

classifier = h2o.deeplearning(y = 'Churn',

training\_frame = as.h2o(trainData),

activation = 'Rectifier',

hidden = c(128,128),

epochs = 100,

shuffle\_training\_data=TRUE,

replicate\_training\_data=TRUE,

adaptive\_rate=TRUE,

overwrite\_with\_best\_model=TRUE,

train\_samples\_per\_iteration = -2,

seed=100)

**Model development steps**

* Categorical variables were converted to numeric by creating dummy variables
* Converted the data type of response variable to factor
* Scaled the numerical variables
* Split the data into train and test data in the proportion of 70 and 30 percent ensuring equal

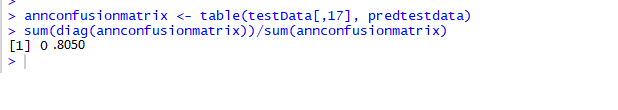
## Accuracy Of The ANN Model

Here is the code snippet

annconfusionmatrix <- table(testData[,17], predtestdata)

sum(diag(annconfusionmatrix))/sum(annconfusionmatrix)

The output is given below



The accuracy of the ANN model is 80.50%

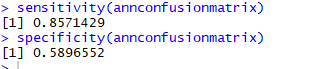
## SENSIVITY and SPECIFICITY in ANN

Here is the code snippet

sensitivity(annconfusionmatrix)

specificity(annconfusionmatrix)

The output is given below



## ROC CURVE & performance of the model

Here is the code snippet

modelprediction<-prediction(predtestdata,testData$Churn)

modelperformance<-performance(modelprediction,"tpr","fpr")

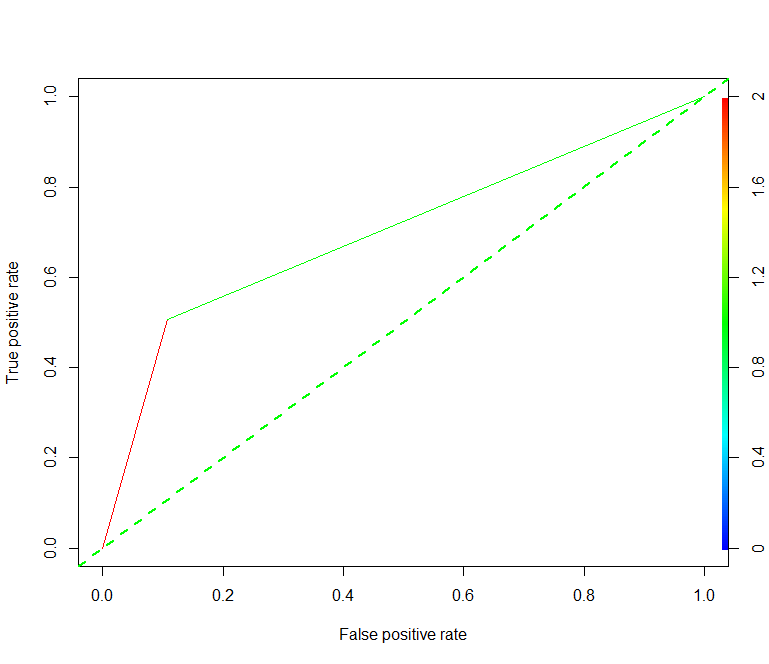
plot(modelperformance,col = "blue")

abline(a=0,b=1,lwd=2,lty=2,col = "green")

annaccry<-performance(modelprediction,"auc")

annaccry

The area under the curve (AUC) is used to summarize the performance of the model, and it is currently about **0.7281**. This is a pretty high value, which is an indicator of a good performance.



## Model evaluation parameters –

Here is the summary of the ANN model

|  |  |
| --- | --- |
| **Threshold value** | **Values (Numeric)** |
| Accuracy | 80.50% |
| Sensitivity | 85.71% |
| Specificity | 58.96% |
| AUC(ROC) | 72.81% |

# Conclusion

Comparison between different Models

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Accuracy(%)** | **Sensitivity (%)** | **Specificity (%)** | AUC(ROC) |
| Support Vector Machine | 78.98% | 82.23% | 64.85% | 68.50% |
| Advance Neural Network | 80.50% | 85.71% | 58.96% | 72.81% |
| KNN | 78.65% | 82.98% | 62.38% | 68.50% |

So we can conclude by comparing different parameters of the above three models (SVM ,ANN & KNN) that **ANN** performs better compared to other two models.

Sensitivity of ANN is high which is of utmost importance of selecting the model. Sensitivity indicates percentage of correctly identified churned customers and its value is comparatively good as compare to specificity so will be considering ANN model for this problem

# ANNEXTURE

1. Machine\_Learning\_Assignment-2\_TA17002.R**-Source code**