ALY6020_Final_Project

Yuhe Gu & Zhenchao Zhang

3/27/2019

Summary

Telecommunication company, also know as a telco, is a kind of company to provide telecommunications service and also provides the Television servise and Internet services. With the development of the global requirement in telecommunication and connection service, the telecommunications company is needed to expand there service and acquire more customers.

Customer Churn is one of the biggest problems facing most businesses to solve. According to Harvard Business Review, it costs between 5 times and 25 times as much to find a new customer than to retain an existing one. In other words, your existing customers are worth their weight in gold (Heintz, 2018).

If we have the model to predict a customer, or a group of customer have high probability to churn, the telecommunication company may make some business strategies, such as putting out new discount packages to these customers. Moreover, the results from the predictive model could also provide the prediction of the profits.

To gain profits, It is important to retain customers. Therefore, the goal of this project is to predict behaviors of churn or not churn to help retain customers.

Introduction

In the Telco Customer Churn dataset, each row refers to a single customer with 20 different attributes.

The attributes include:

Churn: Customers who left within the last month

Services that each customer has signed up for – phone, multiple lines, internet, online security, online backup, device protection, tech support, and streaming TV and movies.

Customer account information – how long they've been a customer (tenure), contract, payment method, paperless billing, monthly charges, and total charges.

Demographic information about customers – gender, age range, partners, and dependents.

In the following analysis, we will research on these attributes whether they influence the result (churn or not churn), and how much they influence.

Implementation of the project

```r
#Library
library(readr)
library(ggplot2)
library(DataExplorer)
library(dplyr)
library(tidyr)
library(corrplot)
```

```r
library(caret)
#install.packages("rms")
library(rms)
library(MASS)
library(e1071)
#install.packages("ROCR")
library(ROCR)
library(gplots)
library(pROC)
library(rpart)
library(rpart.plot)
library(randomForest)
#install.packages("ggpubr")
library(ggpubr)
```

Data Manipulation

Import the Data

```r
telecom <- read.csv("WA_Fn-UseC_-Telco-Customer-Churn.csv")
```

Show the summary of the dataset

```r
str(telecom)
```

```
## 'data.frame':    7043 obs. of  21 variables:
##  $ customerID      : Factor w/ 7043 levels "0002-ORFBO","0003-MKNFE",..: 5376 3963 2565 5536 6512 655
##  $ gender          : Factor w/ 2 levels "Female","Male": 1 2 2 2 1 1 2 1 1 2 ...
##  $ SeniorCitizen   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Partner         : Factor w/ 2 levels "No","Yes": 2 1 1 1 1 1 1 1 2 1 ...
##  $ Dependents      : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 2 1 1 2 ...
##  $ tenure          : int  1 34 2 45 2 8 22 10 28 62 ...
##  $ PhoneService    : Factor w/ 2 levels "No","Yes": 1 2 2 1 2 2 2 1 2 2 ...
##  $ MultipleLines   : Factor w/ 3 levels "No","No phone service",..: 2 1 1 2 1 3 3 2 3 1 ...
##  $ InternetService : Factor w/ 3 levels "DSL","Fiber optic",..: 1 1 1 1 2 2 2 1 2 1 ...
##  $ OnlineSecurity  : Factor w/ 3 levels "No","No internet service",..: 1 3 3 3 1 1 1 3 1 3 ...
##  $ OnlineBackup    : Factor w/ 3 levels "No","No internet service",..: 3 1 3 1 1 1 3 1 1 3 ...
##  $ DeviceProtection: Factor w/ 3 levels "No","No internet service",..: 1 3 1 3 1 3 1 1 3 1 ...
##  $ TechSupport     : Factor w/ 3 levels "No","No internet service",..: 1 1 1 3 1 1 1 1 3 1 ...
##  $ StreamingTV     : Factor w/ 3 levels "No","No internet service",..: 1 1 1 1 1 3 3 1 3 1 ...
##  $ StreamingMovies : Factor w/ 3 levels "No","No internet service",..: 1 1 1 1 1 3 1 1 3 1 ...
##  $ Contract        : Factor w/ 3 levels "Month-to-month",..: 1 2 1 2 1 1 1 1 1 2 ...
##  $ PaperlessBilling: Factor w/ 2 levels "No","Yes": 2 1 2 1 2 2 2 1 2 1 ...
##  $ PaymentMethod   : Factor w/ 4 levels "Bank transfer (automatic)",..: 3 4 4 1 3 3 2 4 3 1 ...
##  $ MonthlyCharges  : num  29.9 57 53.9 42.3 70.7 ...
##  $ TotalCharges    : num  29.9 1889.5 108.2 1840.8 151.7 ...
##  $ Churn           : Factor w/ 2 levels "No","Yes": 1 1 2 1 2 2 1 1 2 1 ...
```

```r
summary(telecom)
```

```
##       customerID      gender     SeniorCitizen     Partner     Dependents
```

```
## 0002-ORFBO:   1   Female:3488   Min.   :0.0000   No :3641   No :4933
## 0003-MKNFE:   1   Male  :3555   1st Qu.:0.0000   Yes:3402   Yes:2110
## 0004-TLHLJ:   1                 Median :0.0000
## 0011-IGKFF:   1                 Mean   :0.1621
## 0013-EXCHZ:   1                 3rd Qu.:0.0000
## 0013-MHZWF:   1                 Max.   :1.0000
## (Other)   :7037
##     tenure      PhoneService          MultipleLines     InternetService
## Min.   : 0.00   No : 682     No              :3390   DSL        :2421
## 1st Qu.: 9.00   Yes:6361     No phone service: 682   Fiber optic:3096
## Median :29.00                Yes             :2971   No         :1526
## Mean   :32.37
## 3rd Qu.:55.00
## Max.   :72.00
##
##             OnlineSecurity              OnlineBackup
## No                 :3498   No                 :3088
## No internet service:1526   No internet service:1526
## Yes                :2019   Yes                :2429
##
##
##
##
##            DeviceProtection              TechSupport
## No                 :3095    No                 :3473
## No internet service:1526    No internet service:1526
## Yes                :2422    Yes                :2044
##
##
##
##
##              StreamingTV             StreamingMovies
## No                 :2810   No                 :2785
## No internet service:1526   No internet service:1526
## Yes                :2707   Yes                :2732
##
##
##
##
##           Contract    PaperlessBilling                   PaymentMethod
## Month-to-month:3875   No :2872         Bank transfer (automatic):1544
## One year      :1473   Yes:4171         Credit card (automatic)  :1522
## Two year      :1695                    Electronic check         :2365
##                                        Mailed check             :1612
##
##
##
## MonthlyCharges    TotalCharges    Churn
## Min.   : 18.25   Min.   :  18.8   No :5174
## 1st Qu.: 35.50   1st Qu.: 401.4   Yes:1869
## Median : 70.35   Median :1397.5
## Mean   : 64.76   Mean   :2283.3
## 3rd Qu.: 89.85   3rd Qu.:3794.7
## Max.   :118.75   Max.   :8684.8
```

```
##                     NA's   :11
```

Observations with Missing Values

According to the summary above, there are 11 missing values in the TotalCharges column, which account for 0.16% of the observations, which is a small number, and removing those 11 rows with missing values will not bring large influence to the final results.

The follwing code is removing the missing values from the datasets.

```
telecom <- telecom[complete.cases(telecom),]
```

Check Churn Rate for the full dataset

```
telecom %>%
  summarise(Total = n(), n_Churn = sum(Churn == "Yes"), p_Churn = n_Churn/Total)
```

```
##   Total n_Churn  p_Churn
## 1  7032    1869 0.265785
```

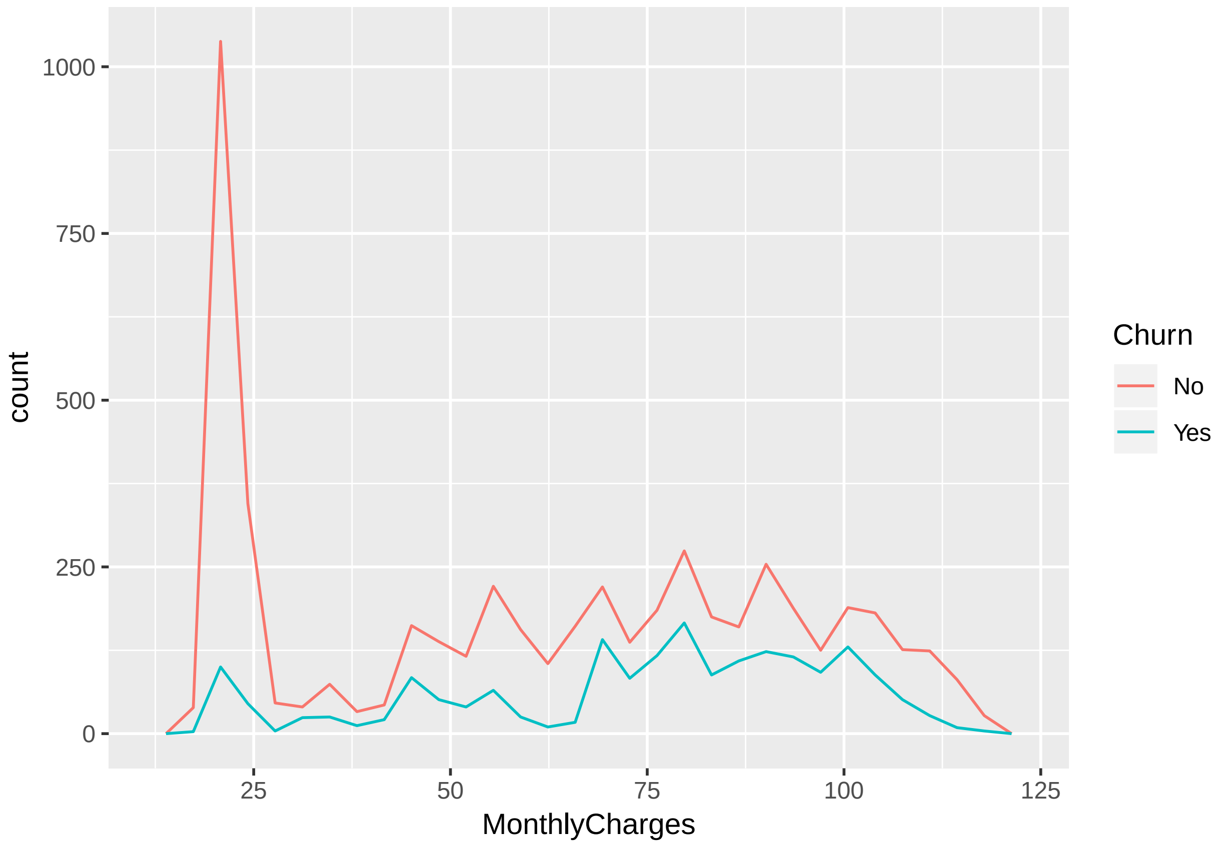
There are about 26.6% of customers churn.

Exploratory Data Analysis

Data Distributions

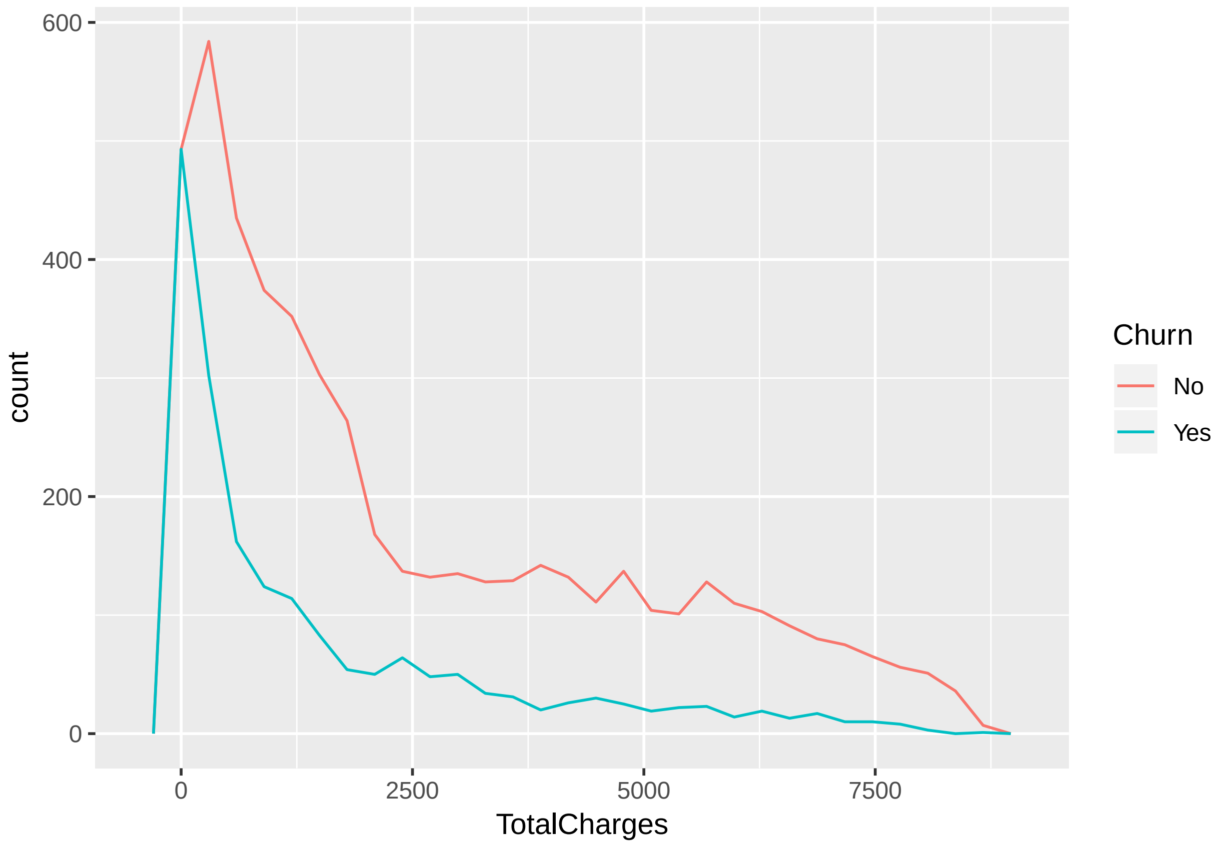
In this part, we will visulize the distributions of continuous variables to make some comparison.

```
ggplot(data = telecom, aes(MonthlyCharges, col = Churn))+
  geom_freqpoly()
```



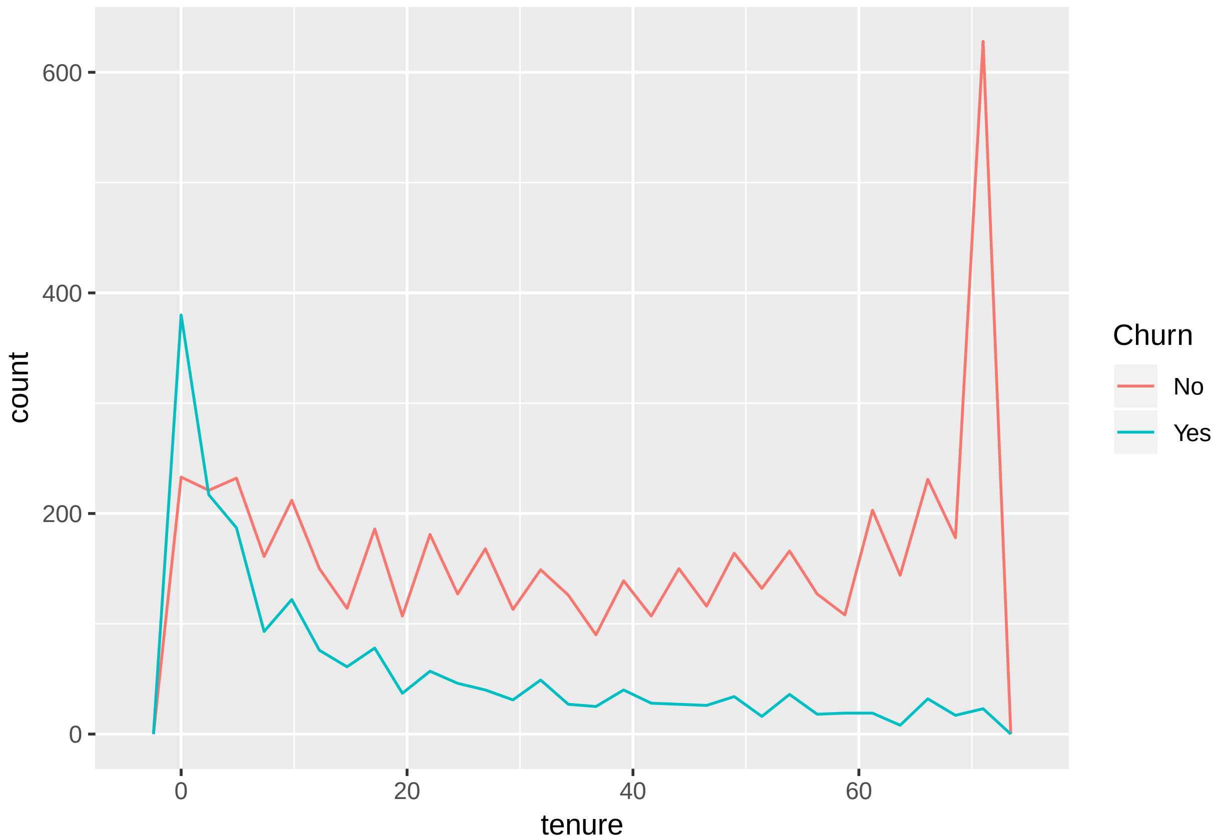
From the plot above, we can conclude that if a customer with less than 25 dollars Monthly charge, they have high probability to churn. On the other hand, if the customer with larger than 30 dollars monthly charge, the distributions of the customers who churn or not are similar (and the churn rate is lower than not churn).

```
ggplot(data = telecom, aes(TotalCharges, col = Churn))+
  geom_freqpoly()
```



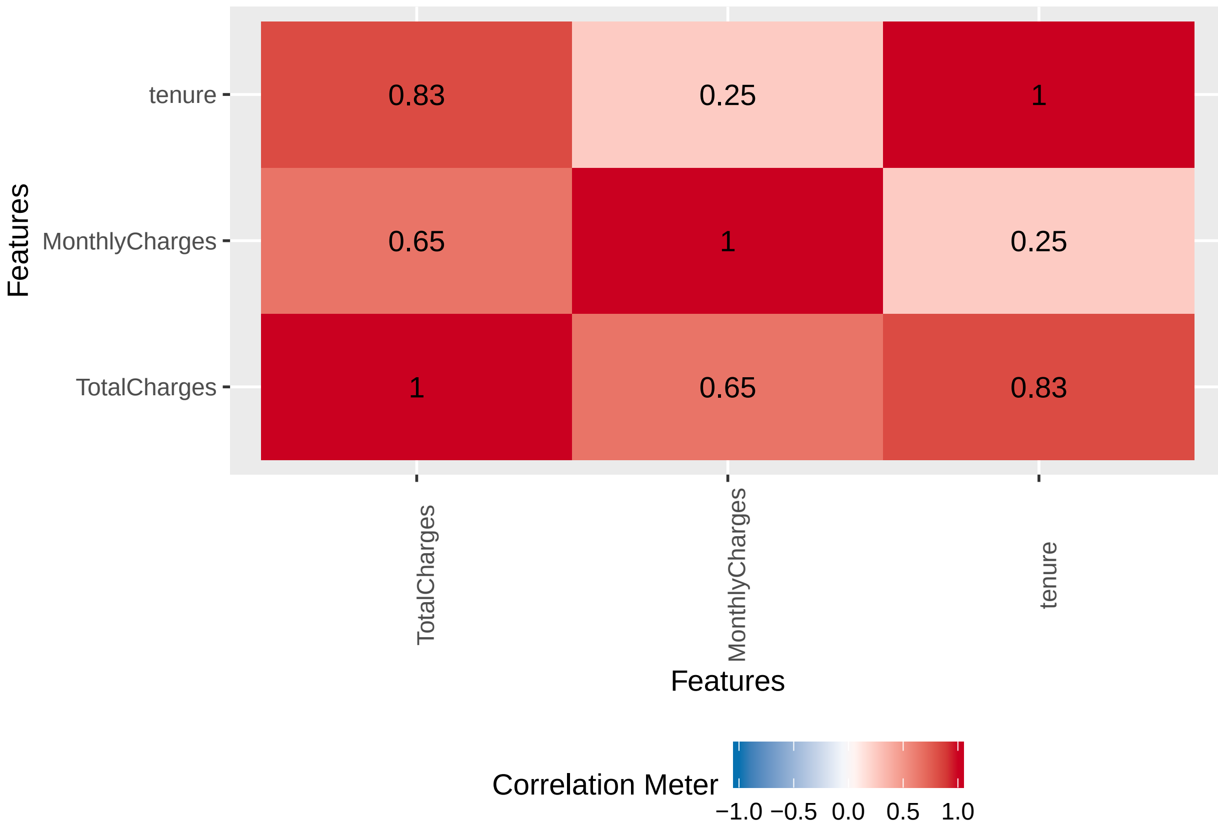
In terms of the TotalCharges, it is highly positive skew for all customers no matter whether they churned or not.

```
ggplot(data = telecom, aes(tenure, col = Churn))+
  geom_freqpoly()
```



In terms of the tenure, the distributions are very different between customers who churned and who didn't churn. From the plot, we can conclude that a customer are more likely to quit the telecommunication company in the first few month, and the more they have used the service, they will mot quit the seiverce. Moreover, this company has a huge number of customers who have been in the service more than sixty months, which means more than five years. These group of customer is the "old customer" for the business.

```
plot_correlation(telecom[,c("TotalCharges", "MonthlyCharges", "tenure")])
```

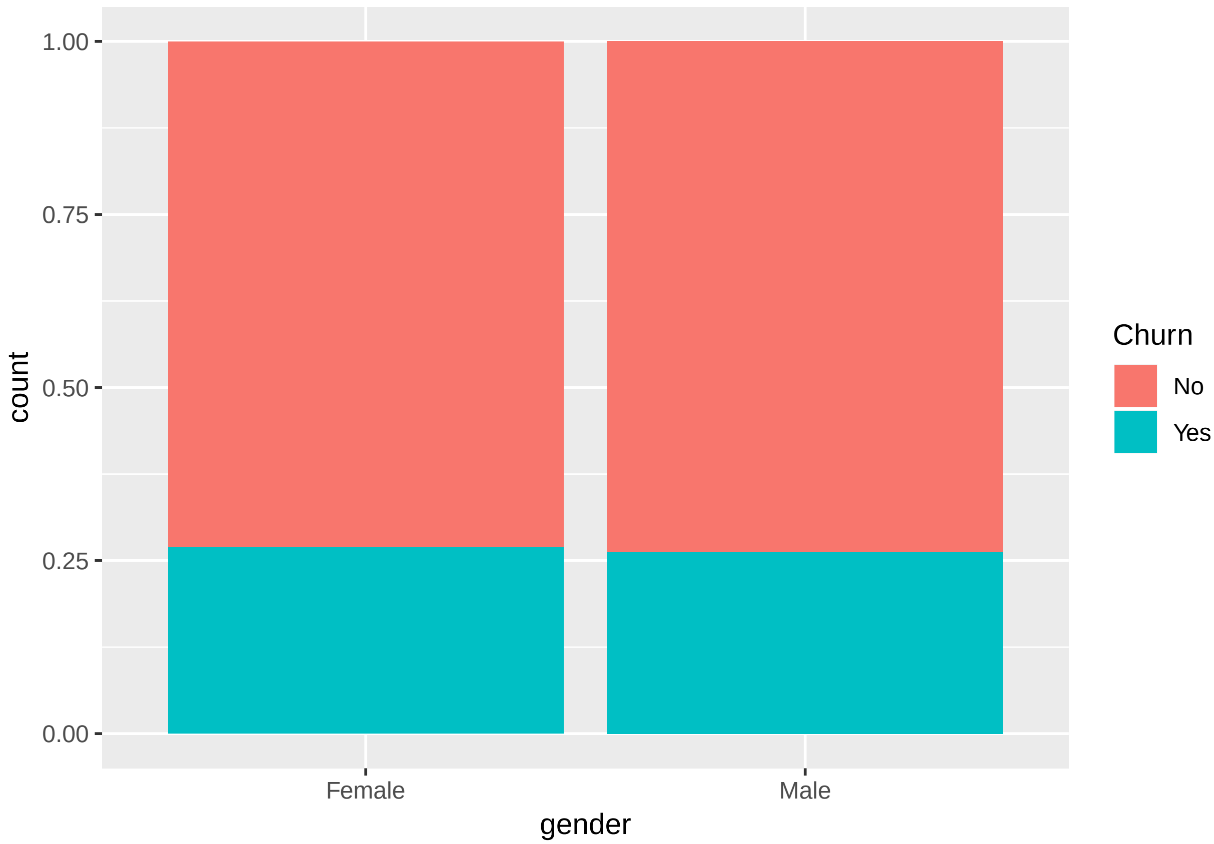


The plot shows high correlations between Totalcharges & tenure and between TotalCharges & MonthlyCharges. In the modeling part, we will consider the correlation when we build the model to increase the models' accuracy for prediction.

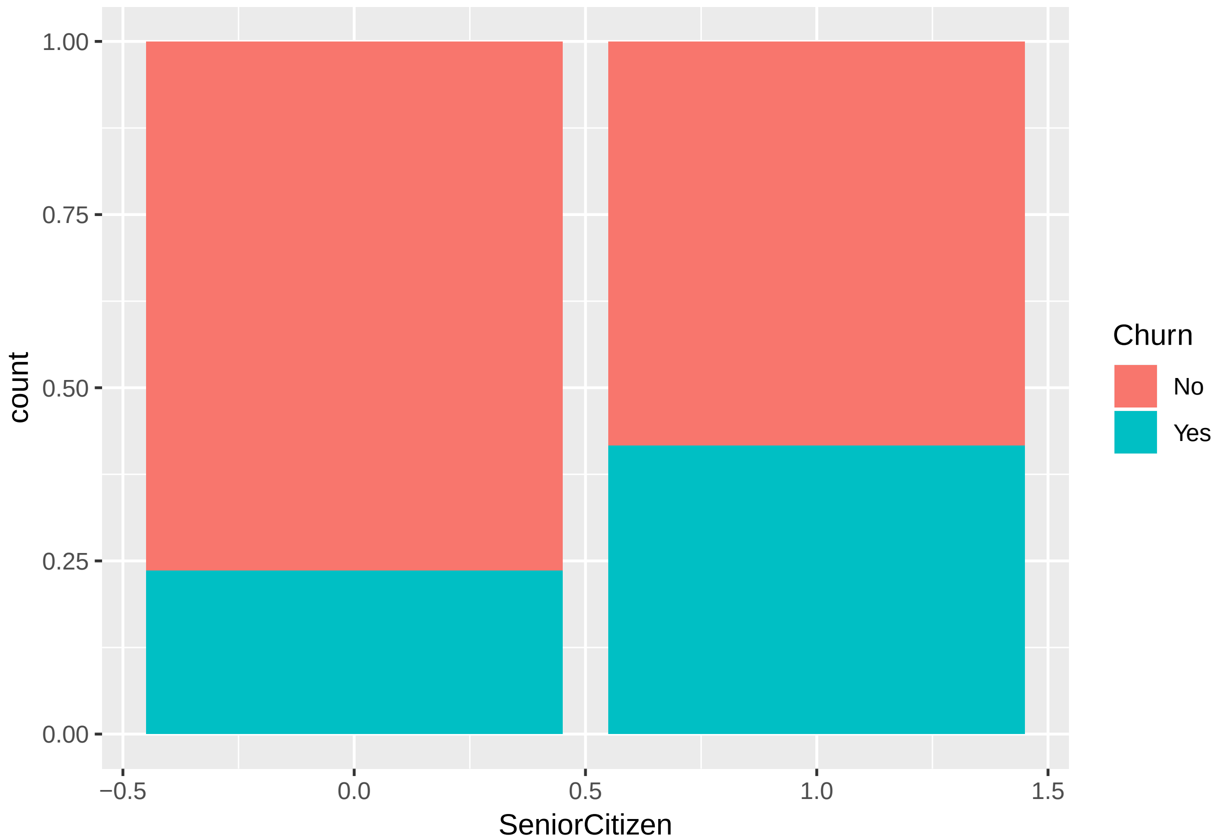
Categorical Variables

In this part, we will research on how the customers' demographic information influence on the customer churn.

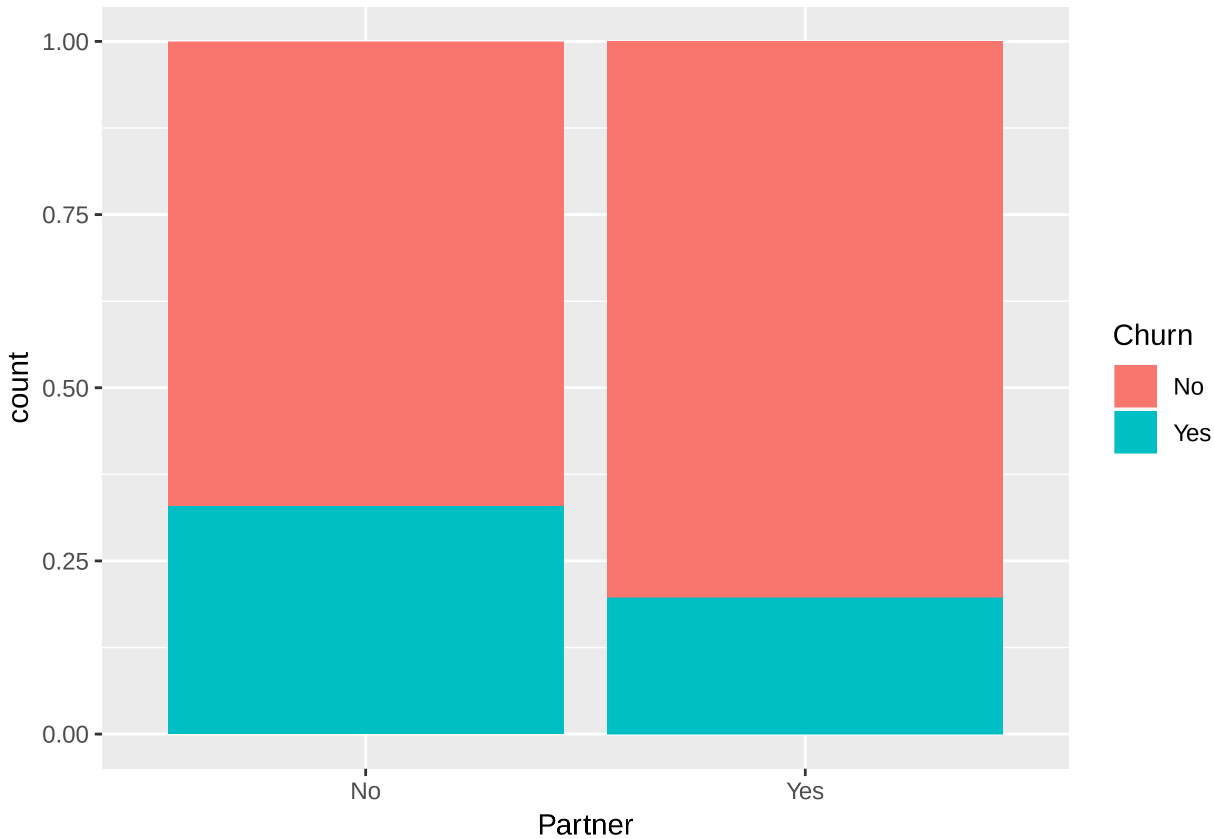
```
ggplot(data = telecom) +
  geom_bar(mapping = aes(x = gender, fill = Churn), position = "fill", stat = "count")
```



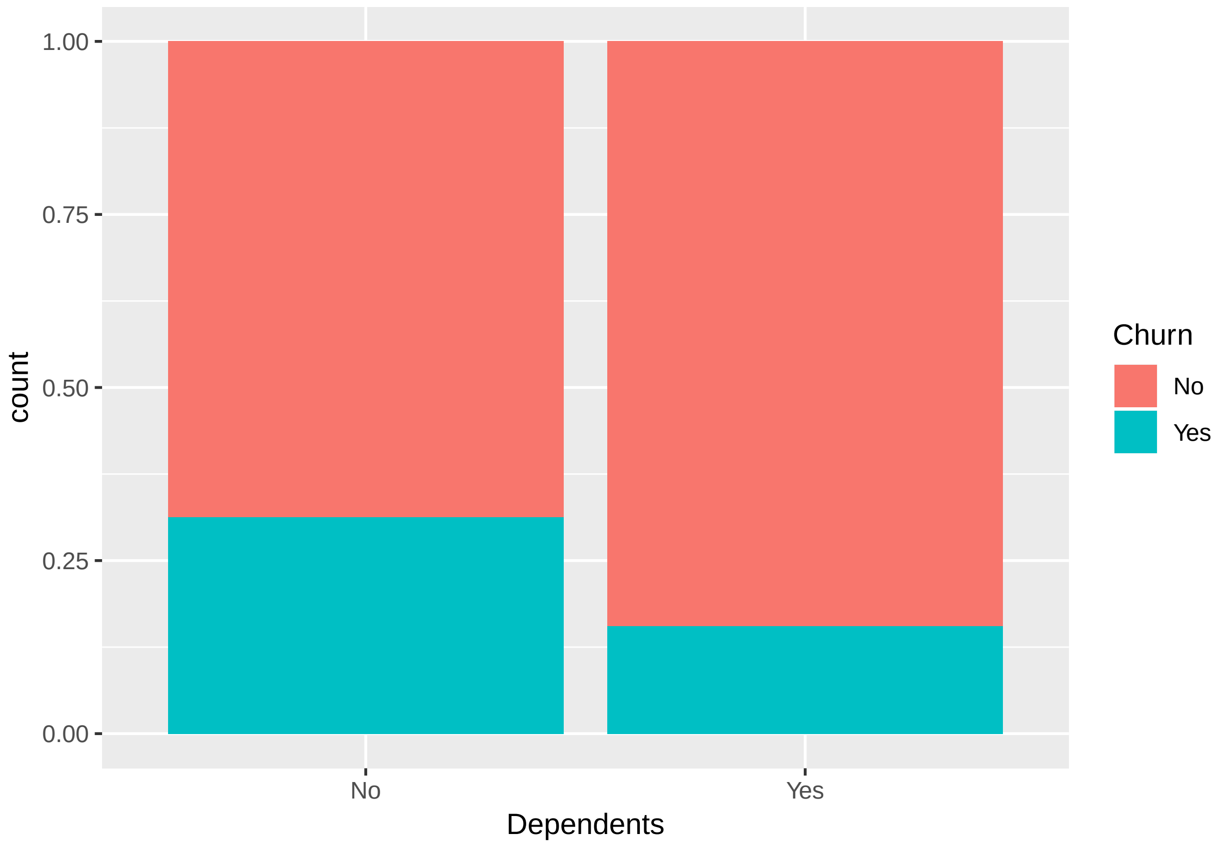

```
ggplot(data = telecom) +
  geom_bar(mapping = aes(x = SeniorCitizen, fill = Churn), position = "fill", stat = "count")
```



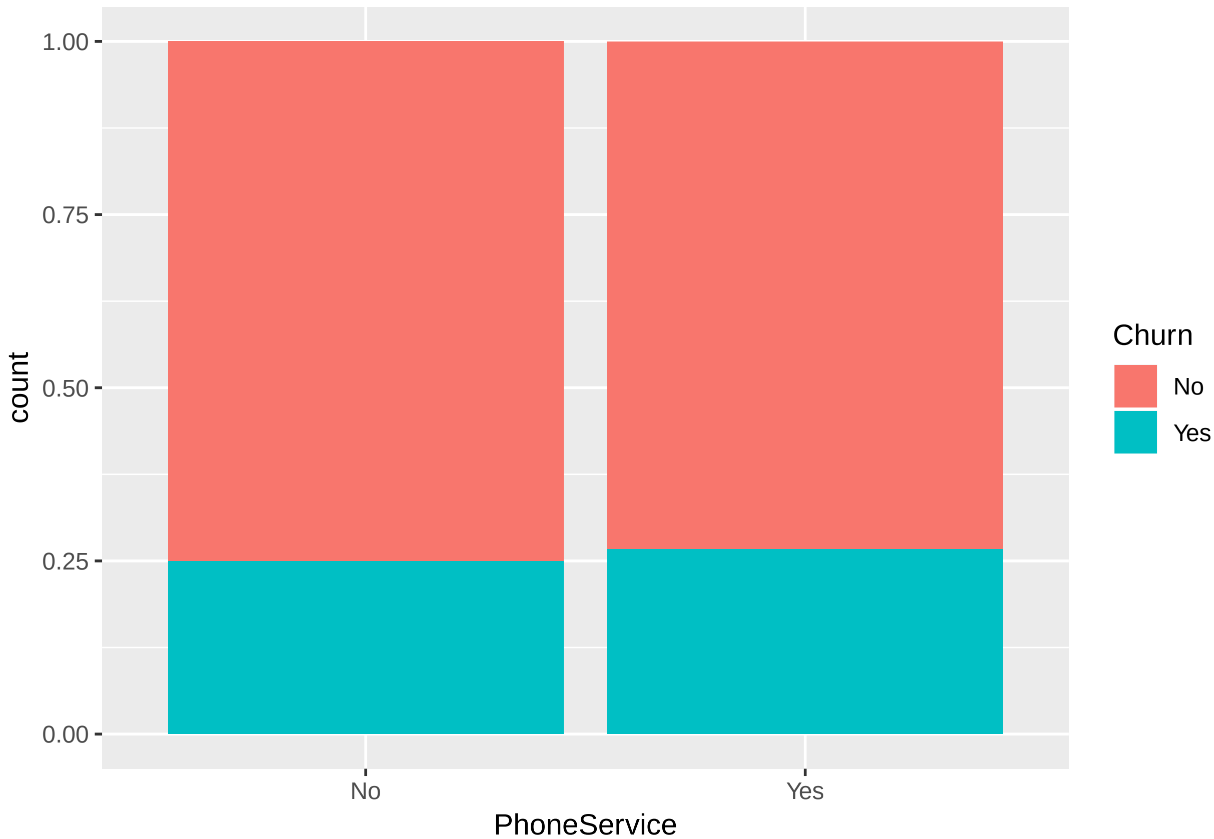

```r
ggplot(data = telecom) +
  geom_bar(mapping = aes(x = Partner, fill = Churn), position = "fill", stat = "count")
```



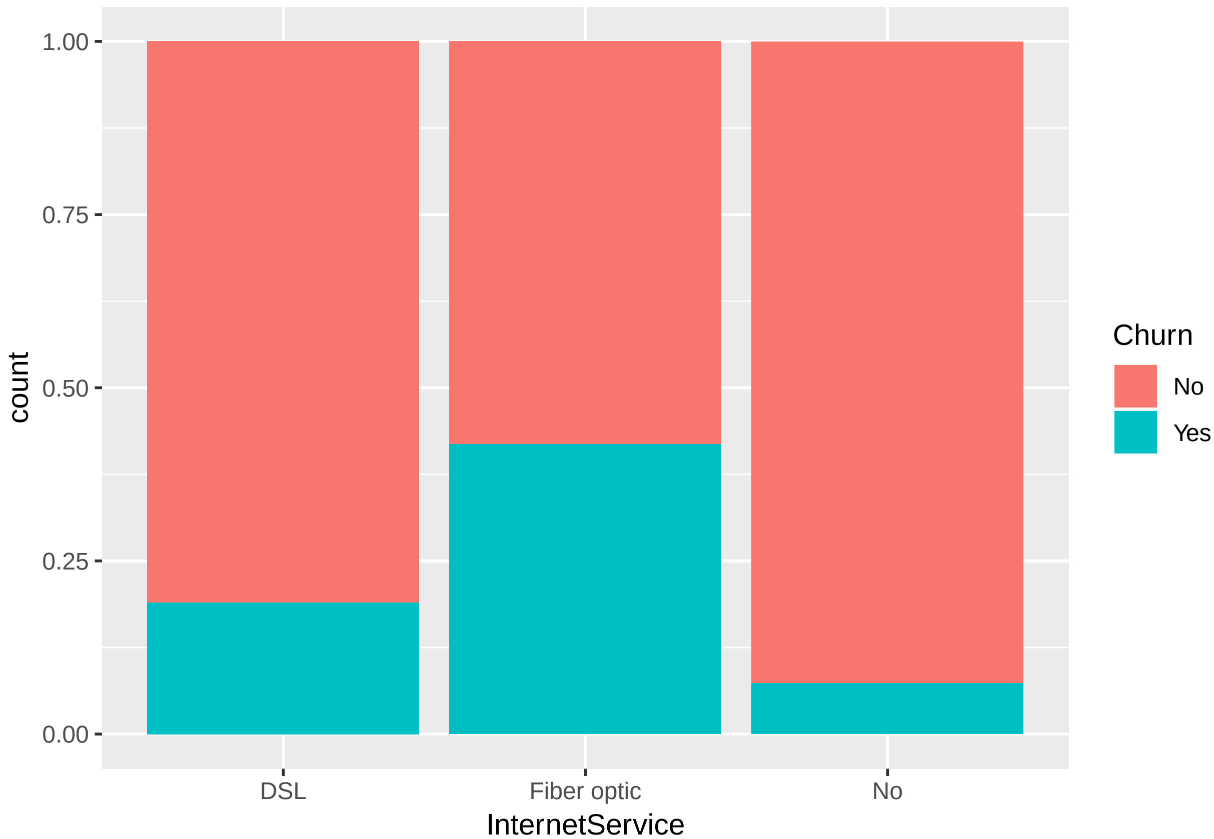

```
ggplot(data = telecom) +
  geom_bar(mapping = aes(x = Dependents, fill = Churn), position = "fill", stat = "count")
```




```r
ggplot(data = telecom) +
  geom_bar(mapping = aes(x = PhoneService, fill = Churn), position = "fill", stat = "count")
```




```r
ggplot(data = telecom) +
  geom_bar(mapping = aes(x = InternetService, fill = Churn), position = "fill", stat = "count")
```



From the plot, we can conclude:

Genders and phone service have no influences on the customer churn.
The senior customers have higher churn rate.
The customers who have partners or dependents have lower churn rate.

The tenure refers to how many months that a customer been in the service. In order to get better analysis, we change the column to a factor with 5 levels, with each level represents a bin of tenure in years.

```
telecom %>%
  mutate(tenure_year = case_when(tenure <= 12 ~ "0-1",
                                 tenure > 12 & tenure <= 24 ~ "1-2",
                                 tenure > 24 & tenure <= 36 ~ "2-3",
                                 tenure > 36 & tenure <= 48 ~ "3-4",
                                 tenure > 48 & tenure <= 60 ~ "4-5",
                                 tenure > 60 & tenure <= 72 ~ "5-6")) -> telecom
telecom$tenure <-NULL
table(telecom$tenure_year)
```

```
##
##  0-1  1-2  2-3  3-4  4-5  5-6
## 2175 1024  832  762  832 1407
```

Data Analysis

Logistic Regression Model

In order to build the logistic regression model, we change the categorical content such as "yes" and "no" into 1 and 0. The columns we modify are: Churn, gender, Partner, PhoneService, Dependents, PaperlessBilling

```
telecom_LR <- telecom
telecom_LR$Churn <- ifelse(telecom_LR$Churn == "Yes", 1, 0)
telecom_LR$gender <- ifelse(telecom_LR$gender == "Female", 1, 0)
telecom_LR$Partner <- ifelse(telecom_LR$Partner == "Yes", 1, 0)
telecom_LR$PhoneService <- ifelse(telecom_LR$PhoneService == "Yes", 1, 0)
telecom_LR$Dependents <- ifelse(telecom_LR$Dependents == "Yes", 1, 0)
telecom_LR$PaperlessBilling <- ifelse(telecom_LR$PaperlessBilling == "Yes", 1, 0)
#remove the columns we will not use
telecom_LR <- telecom_LR[,-c(1, 7, 8, 9, 10, 11, 12, 13, 14, 15, 17)]
str(telecom_LR)
```

```
## 'data.frame':    7032 obs. of  10 variables:
##  $ gender          : num  1 0 0 0 1 1 0 1 1 0 ...
##  $ SeniorCitizen   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Partner         : num  1 0 0 0 0 0 0 0 1 0 ...
##  $ Dependents      : num  0 0 0 0 0 0 1 0 0 1 ...
##  $ PhoneService    : num  0 1 1 0 1 1 1 0 1 1 ...
##  $ PaperlessBilling: num  1 0 1 0 1 1 1 0 1 0 ...
##  $ MonthlyCharges  : num  29.9 57 53.9 42.3 70.7 ...
##  $ TotalCharges    : num  29.9 1889.5 108.2 1840.8 151.7 ...
##  $ Churn           : num  0 0 1 0 1 1 0 0 1 0 ...
##  $ tenure_year     : chr  "0-1" "2-3" "0-1" "3-4" ...
```

Create the data into traning and testing datasets (80% vs 20%)

```
set.seed(1)
trainindex = createDataPartition(telecom_LR$Churn, p=0.80, list=FALSE)
train = telecom_LR[trainindex,]
test = telecom_LR[-trainindex,]
```

Train Model

```
model <- glm(Churn ~., family = "binomial", data = train)
summary(model)
```

```
##
## Call:
## glm(formula = Churn ~ ., family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1646  -0.6858  -0.3877   0.6603   2.8054
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -1.689e+00  1.539e-01 -10.970  < 2e-16 ***
## gender            7.146e-03  7.060e-02   0.101    0.919
```

```
## SeniorCitizen     5.284e-01  9.205e-02   5.740 9.48e-09 ***
## Partner          -2.859e-03  8.338e-02  -0.034    0.973
## Dependents       -3.758e-01  9.610e-02  -3.911 9.20e-05 ***
## PhoneService     -7.754e-01  1.259e-01  -6.160 7.26e-10 ***
## PaperlessBilling  5.466e-01  7.963e-02   6.864 6.67e-12 ***
## MonthlyCharges    3.535e-02  2.266e-03  15.601  < 2e-16 ***
## TotalCharges     -2.593e-04  6.147e-05  -4.219 2.45e-05 ***
## tenure_year1-2   -9.150e-01  1.117e-01  -8.190 2.62e-16 ***
## tenure_year2-3   -1.284e+00  1.537e-01  -8.352  < 2e-16 ***
## tenure_year3-4   -1.143e+00  2.023e-01  -5.649 1.61e-08 ***
## tenure_year4-5   -1.501e+00  2.615e-01  -5.740 9.46e-09 ***
## tenure_year5-6   -2.205e+00  3.418e-01  -6.451 1.11e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6487.9  on 5625  degrees of freedom
## Residual deviance: 4967.8  on 5612  degrees of freedom
## AIC: 4995.8
##
## Number of Fisher Scoring iterations: 5
```

Testing Model

```
train_prob <- predict(model, data = train, type = "response")
test_prob <- predict(model, newdata = test, type = "response")
```

Set the cut-off value as 0.5.

```
train_pre <- factor(ifelse(train_prob >= 0.5, "Yes", "No"))
train_actual <- factor(ifelse(train$Churn == 1, "Yes", "No"))
test_pre <- factor(ifelse(test_prob >= 0.5, "Yes", "No"))
test_actual <- factor(ifelse(test$Churn == 1, "Yes", "No"))
```

Confusion Matrix and AUC for the logistic regression model

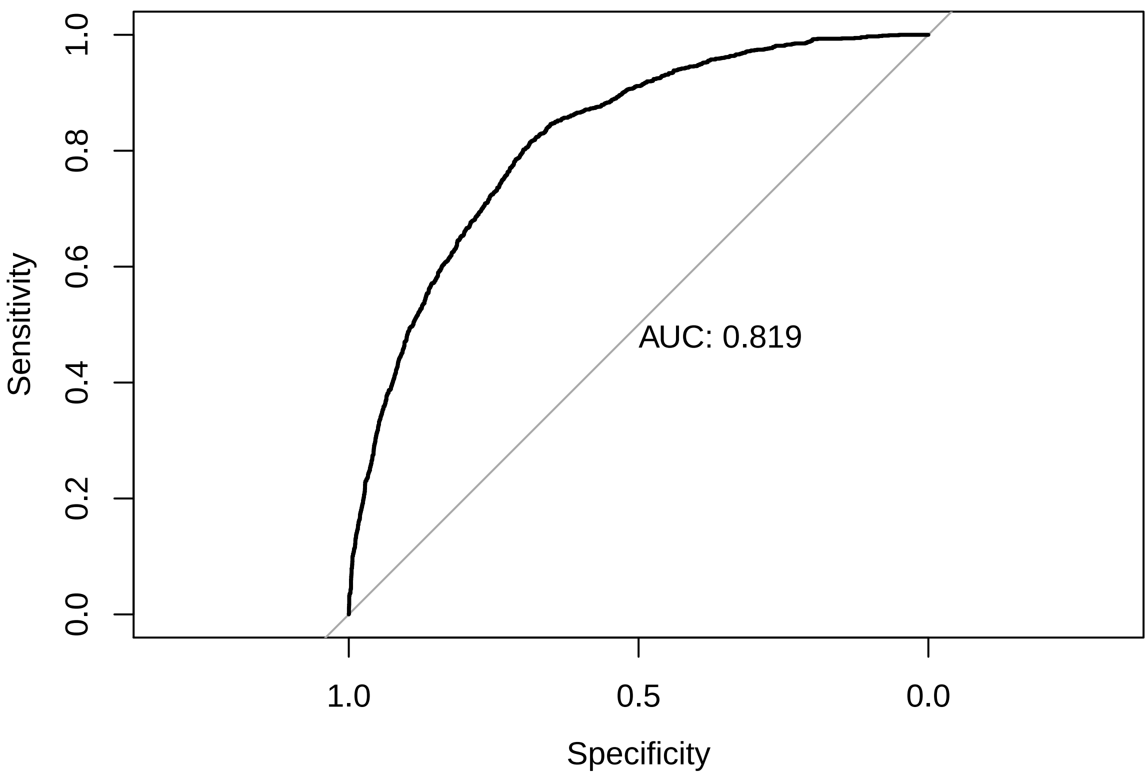
For the Training Set:

```
confusionMatrix(data = train_pre, reference = train_actual)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  3784  829
##        Yes  360  653
##
##                Accuracy : 0.7887
##                  95% CI : (0.7778, 0.7993)
##     No Information Rate : 0.7366
##     P-Value [Acc > NIR] : < 2.2e-16
##
```

```
##                   Kappa : 0.3938
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9131
##             Specificity : 0.4406
##          Pos Pred Value : 0.8203
##          Neg Pred Value : 0.6446
##              Prevalence : 0.7366
##          Detection Rate : 0.6726
##    Detection Prevalence : 0.8199
##       Balanced Accuracy : 0.6769
##
##        'Positive' Class : No
##
```

```
roc <- roc(train$Churn, train_prob, plot= TRUE, print.auc=TRUE)
```



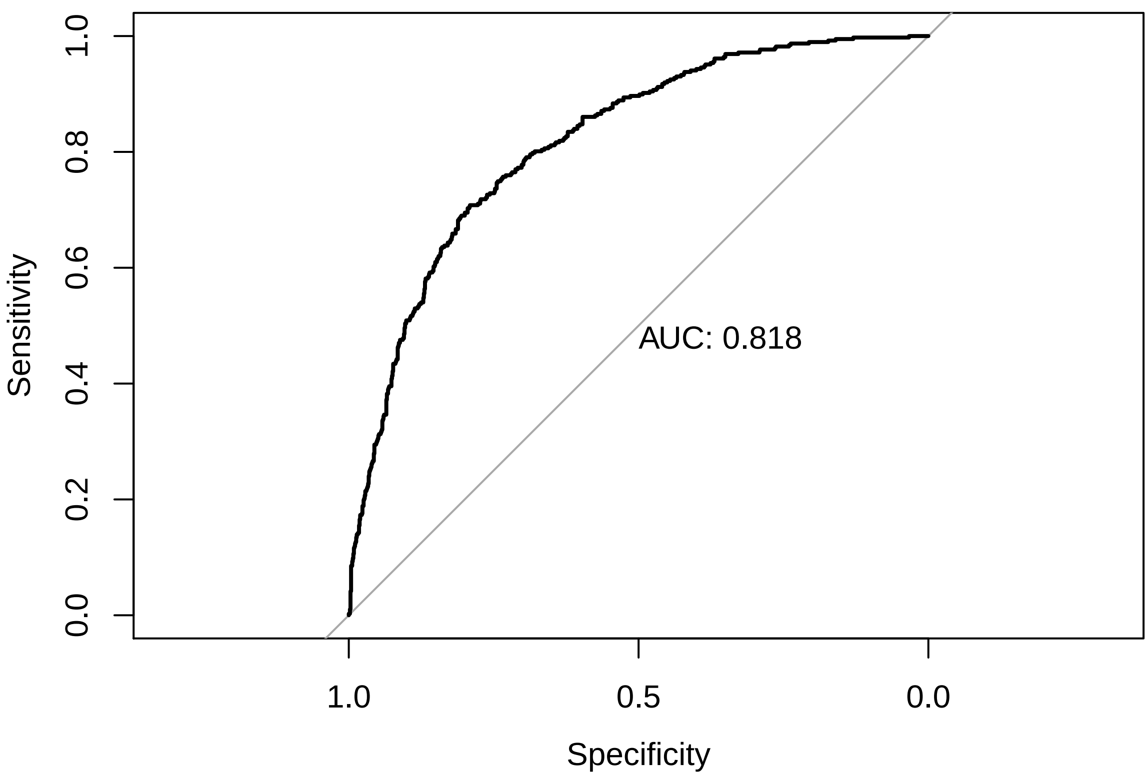
For the Testing Set:

```
confusionMatrix(data = test_pre, reference = test_actual)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  927 203
##        Yes  92 184
##
##                Accuracy : 0.7902
##                  95% CI : (0.768, 0.8112)
##     No Information Rate : 0.7248
```

```
##     P-Value [Acc > NIR] : 9.975e-09
##
##                   Kappa : 0.4228
##  Mcnemar's Test P-Value : 1.509e-10
##
##             Sensitivity : 0.9097
##             Specificity : 0.4755
##          Pos Pred Value : 0.8204
##          Neg Pred Value : 0.6667
##              Prevalence : 0.7248
##          Detection Rate : 0.6593
##    Detection Prevalence : 0.8037
##       Balanced Accuracy : 0.6926
##
##        'Positive' Class : No
##
```

```
roc <- roc(test$Churn, test_prob, plot= TRUE, print.auc=TRUE)
```



For the training set, the accuracy is 0.79 and the AUC is 0.82. For the testing set, the accuracy is 0.79 and the AUC is 0.82. It's a good model because the accuracy and AUC do not have big difference between the training and testing sets, and it has high sensitivity, and relatively low specificity.

Decision Tree

Data Preparation

```
telecomDT <- telecom
telecomDT <- telecom[, -c(1)]
telecomDT %>%
  mutate_if(is.character, as.factor) -> telecomDT
str(telecomDT)
```

```
## 'data.frame':	7032 obs. of  20 variables:
##  $ gender          : Factor w/ 2 levels "Female","Male": 1 2 2 2 1 1 2 1 1 2 ...
##  $ SeniorCitizen   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Partner         : Factor w/ 2 levels "No","Yes": 2 1 1 1 1 1 1 1 2 1 ...
##  $ Dependents      : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 2 1 1 2 ...
##  $ PhoneService    : Factor w/ 2 levels "No","Yes": 1 2 2 1 2 2 2 1 2 2 ...
##  $ MultipleLines   : Factor w/ 3 levels "No","No phone service",..: 2 1 1 2 1 3 3 2 3 1 ...
##  $ InternetService : Factor w/ 3 levels "DSL","Fiber optic",..: 1 1 1 1 2 2 2 1 2 1 ...
##  $ OnlineSecurity  : Factor w/ 3 levels "No","No internet service",..: 1 3 3 3 1 1 1 3 1 3 ...
##  $ OnlineBackup    : Factor w/ 3 levels "No","No internet service",..: 3 1 3 1 1 1 3 1 1 3 ...
##  $ DeviceProtection: Factor w/ 3 levels "No","No internet service",..: 1 3 1 3 1 3 1 1 3 1 ...
##  $ TechSupport     : Factor w/ 3 levels "No","No internet service",..: 1 1 1 3 1 1 1 1 3 1 ...
##  $ StreamingTV     : Factor w/ 3 levels "No","No internet service",..: 1 1 1 1 1 3 3 1 3 1 ...
##  $ StreamingMovies : Factor w/ 3 levels "No","No internet service",..: 1 1 1 1 1 3 1 1 3 1 ...
##  $ Contract        : Factor w/ 3 levels "Month-to-month",..: 1 2 1 2 1 1 1 1 1 2 ...
##  $ PaperlessBilling: Factor w/ 2 levels "No","Yes": 2 1 2 1 2 2 2 1 2 1 ...
##  $ PaymentMethod   : Factor w/ 4 levels "Bank transfer (automatic)",..: 3 4 4 1 3 3 2 4 3 1 ...
##  $ MonthlyCharges  : num  29.9 57 53.9 42.3 70.7 ...
##  $ TotalCharges    : num  29.9 1889.5 108.2 1840.8 151.7 ...
##  $ Churn           : Factor w/ 2 levels "No","Yes": 1 1 2 1 2 2 1 1 2 1 ...
##  $ tenure_year     : Factor w/ 6 levels "0-1","1-2","2-3",..: 1 3 1 4 1 1 2 1 3 6 ...
```

Split the data into training and test sets.

```
set.seed(1)
trainindex = createDataPartition(telecom_LR$Churn, p=0.80, list=FALSE)
trainDT = telecomDT[trainindex,]
testDT = telecomDT[-trainindex,]
```

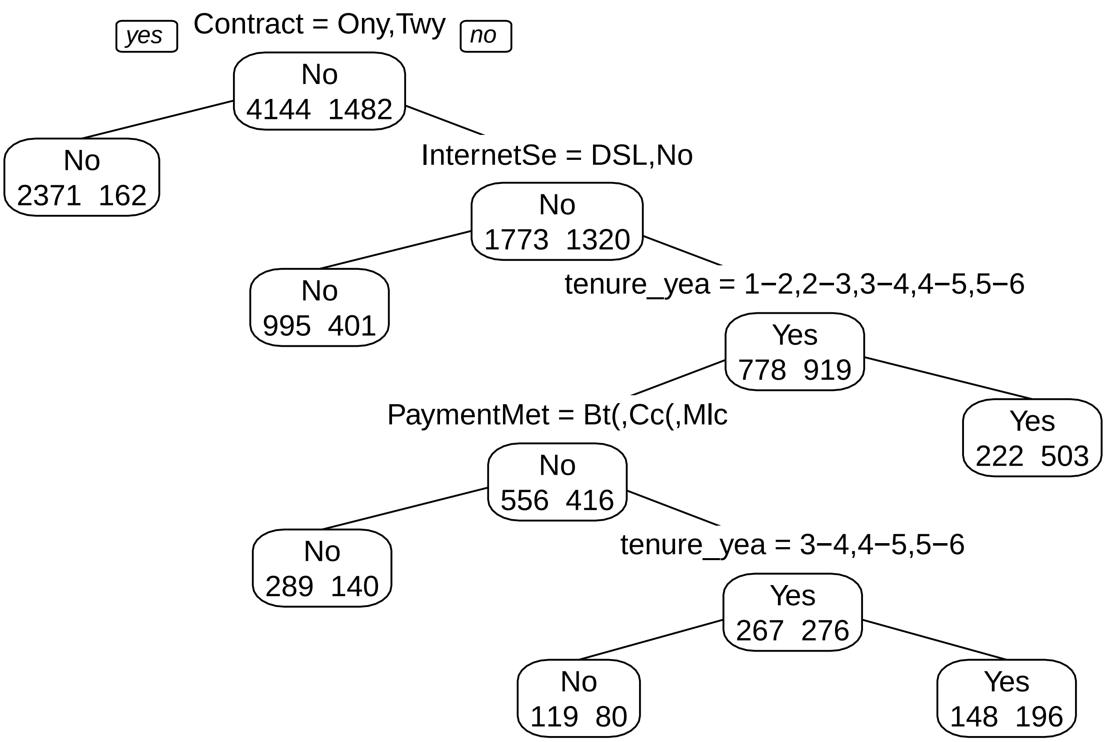
Train Model

That Totalcharges, MonthlyCharges and tenure are highly correlated, which may effect the performance of the decision tree models, so I remove the TotalCharges column to train the decision tree model.

```
modelDT <- rpart(formula = Churn ~ gender + SeniorCitizen + Partner + Dependents + PhoneService +
                       MultipleLines + InternetService + OnlineSecurity + TechSupport +
                       OnlineBackup + DeviceProtection + StreamingTV + StreamingMovies +
                       Contract + PaperlessBilling + tenure_year +
                       PaymentMethod + MonthlyCharges, data = trainDT,
                       method = "class", parms = list(split = "gini"))
```

Plot the Tree

```
prp(modelDT, type = 1, extra = 1, split.font = 1, varlen = -10)
```



Test Model

```
trainDT_pre <- predict(modelDT, data = trainDT, type = "class")
trainDT_prob <- predict(modelDT, data = trainDT, type = "prob")
testDT_pre <- predict(modelDT, newdata= testDT, type = "class")
testDT_prob <- predict(modelDT, newdata = testDT, type = "prob")
```

Confusion Matrix and AUC for the decision tree model

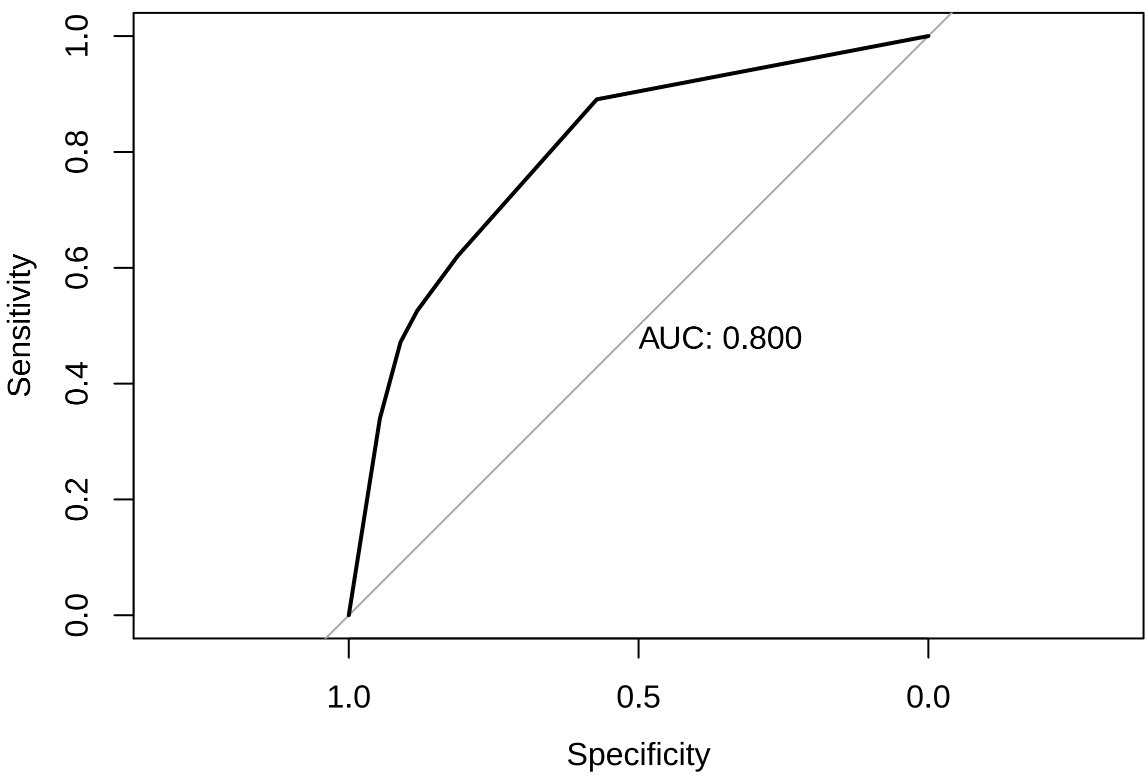
For the Training Set

```
confusionMatrix(data = trainDT_pre, reference = trainDT$Churn)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  3774  783
##        Yes  370  699
##
##                Accuracy : 0.7951
##                  95% CI : (0.7843, 0.8055)
##     No Information Rate : 0.7366
##     P-Value [Acc > NIR] : < 2.2e-16
```

```
##
##                   Kappa : 0.42
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9107
##             Specificity : 0.4717
##          Pos Pred Value : 0.8282
##          Neg Pred Value : 0.6539
##              Prevalence : 0.7366
##          Detection Rate : 0.6708
##    Detection Prevalence : 0.8100
##       Balanced Accuracy : 0.6912
##
##        'Positive' Class : No
##
```

```r
trainDT_actual <- ifelse(trainDT$Churn == "Yes", 1,0)
roc <- roc(trainDT_actual, trainDT_prob[,2], plot= TRUE, print.auc=TRUE)
```



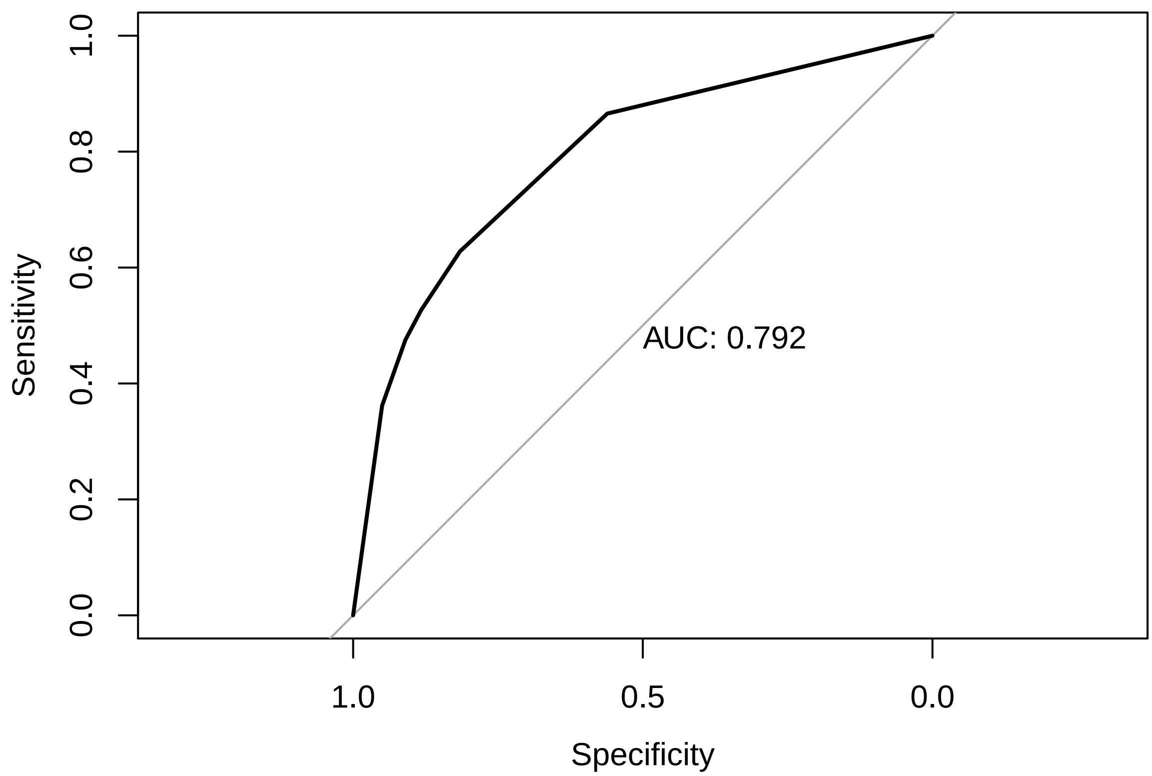
For the Testing Set:

```r
confusionMatrix(data = testDT_pre, reference = testDT$Churn)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  927 203
##        Yes  92 184
##
##                Accuracy : 0.7902
```

```
##                  95% CI : (0.768, 0.8112)
##     No Information Rate : 0.7248
##     P-Value [Acc > NIR] : 9.975e-09
##
##                   Kappa : 0.4228
##  Mcnemar's Test P-Value : 1.509e-10
##
##             Sensitivity : 0.9097
##             Specificity : 0.4755
##          Pos Pred Value : 0.8204
##          Neg Pred Value : 0.6667
##              Prevalence : 0.7248
##          Detection Rate : 0.6593
##    Detection Prevalence : 0.8037
##       Balanced Accuracy : 0.6926
##
##        'Positive' Class : No
##
```

```
testDT_actual <- ifelse(testDT$Churn == "Yes", 1,0)
roc <- roc(testDT_actual, testDT_prob[,2], plot = TRUE, print.auc = TRUE)
```



For the training set, the Accuracy is 0.795 and the AUC is 0.800. For the testing set, the accuracy is 0.790 and the AUC is 0.792. Therefore, the model is good.

Random Forest

Train Model

```
modelRF <- randomForest(formula = Churn ~., data = trainDT, ntree = 300)
print(modelRF)
```

```
##
## Call:
##  randomForest(formula = Churn ~ ., data = trainDT, ntree = 300)
##                Type of random forest: classification
##                      Number of trees: 300
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 20.85%
## Confusion matrix:
##       No Yes class.error
## No  3720 424   0.1023166
## Yes  749 733   0.5053981
```

Variable Importance

```
varImpPlot(modelRF,type=2)
```

Test Model

```r
trainRF_pre <- predict(modelRF, trainDT, type = "class")
trainRF_prob <- predict(modelRF, trainDT, type = "prob")
testRF_pre <- predict(modelRF, newdata = testDT, type = "class")
testRF_prob <- predict(modelRF, newdata = testDT, type = "prob")
```

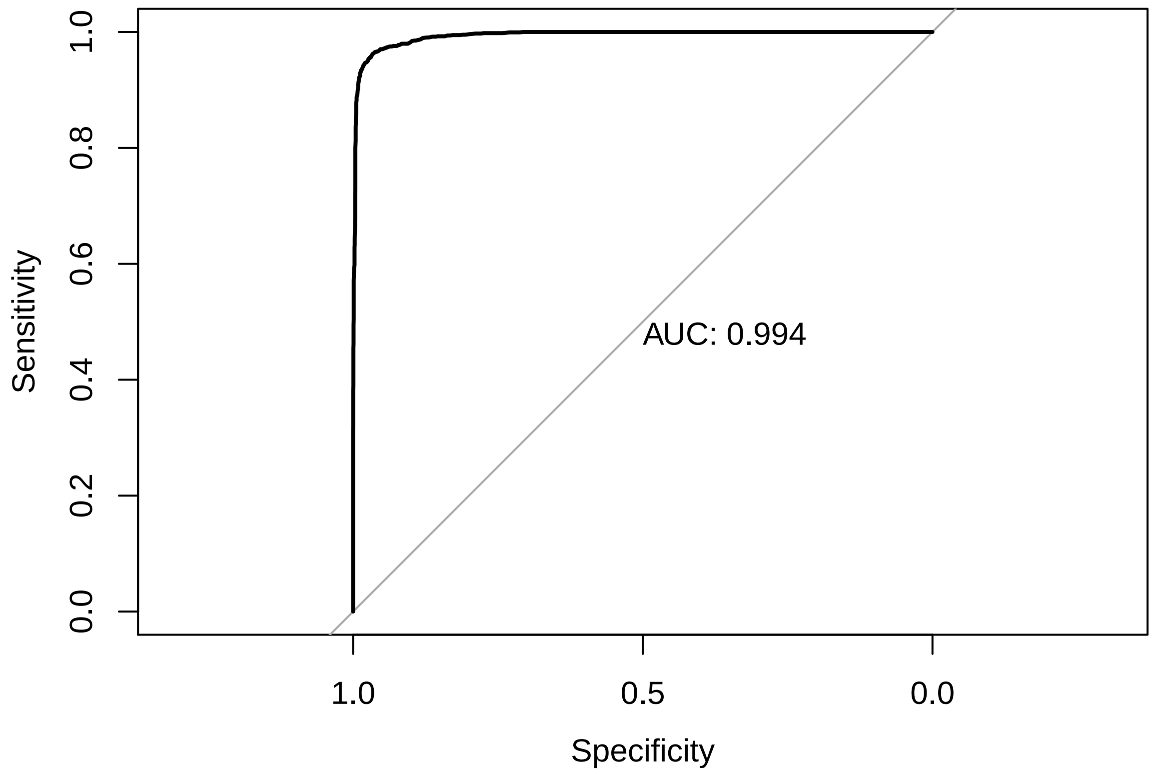
Cross Validation for the random forest model

For the Training Set:

```r
confusionMatrix(data = trainRF_pre, reference = trainDT$Churn)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  4091  103
##        Yes   53 1379
##
##                Accuracy : 0.9723
##                  95% CI : (0.9676, 0.9764)
##     No Information Rate : 0.7366
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9278
##  Mcnemar's Test P-Value : 8.74e-05
##
##             Sensitivity : 0.9872
##             Specificity : 0.9305
##          Pos Pred Value : 0.9754
##          Neg Pred Value : 0.9630
##              Prevalence : 0.7366
##          Detection Rate : 0.7272
##    Detection Prevalence : 0.7455
##       Balanced Accuracy : 0.9589
##
##        'Positive' Class : No
##
```

```r
trainRF_actual <- ifelse(trainDT$Churn == "Yes", 1,0)
roc <- roc(trainRF_actual, trainRF_prob[,2], plot= TRUE, print.auc=TRUE)
```

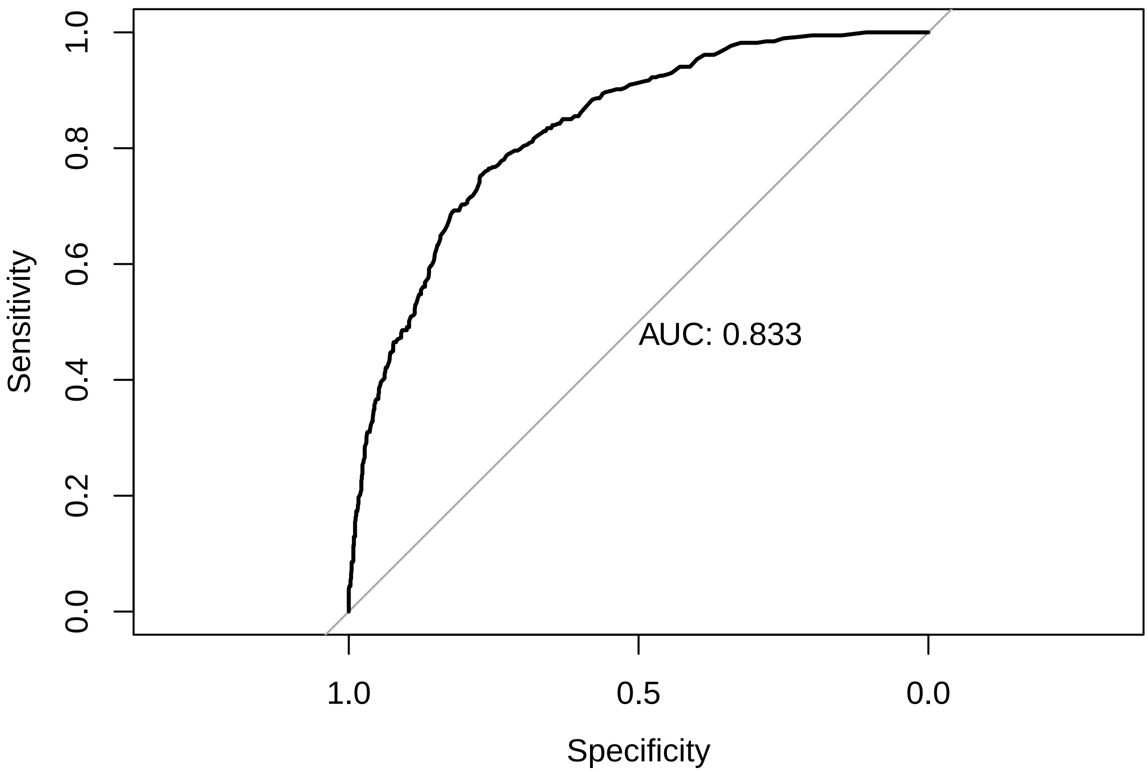


For the Test Set:

```
confusionMatrix(data = testRF_pre, reference = testDT$Churn)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  916 197
##        Yes 103 190
##
##                Accuracy : 0.7866
##                  95% CI : (0.7643, 0.8078)
##     No Information Rate : 0.7248
##     P-Value [Acc > NIR] : 5.905e-08
##
##                   Kappa : 0.4216
##  Mcnemar's Test P-Value : 7.902e-08
##
##             Sensitivity : 0.8989
##             Specificity : 0.4910
##          Pos Pred Value : 0.8230
##          Neg Pred Value : 0.6485
##              Prevalence : 0.7248
##          Detection Rate : 0.6515
##    Detection Prevalence : 0.7916
##       Balanced Accuracy : 0.6949
##
##        'Positive' Class : No
##
```

```r
testRF_actual <- ifelse(testDT$Churn == "Yes", 1,0)
roc <- roc(testRF_actual, testRF_prob[,2], plot = TRUE, print.auc = TRUE)
```



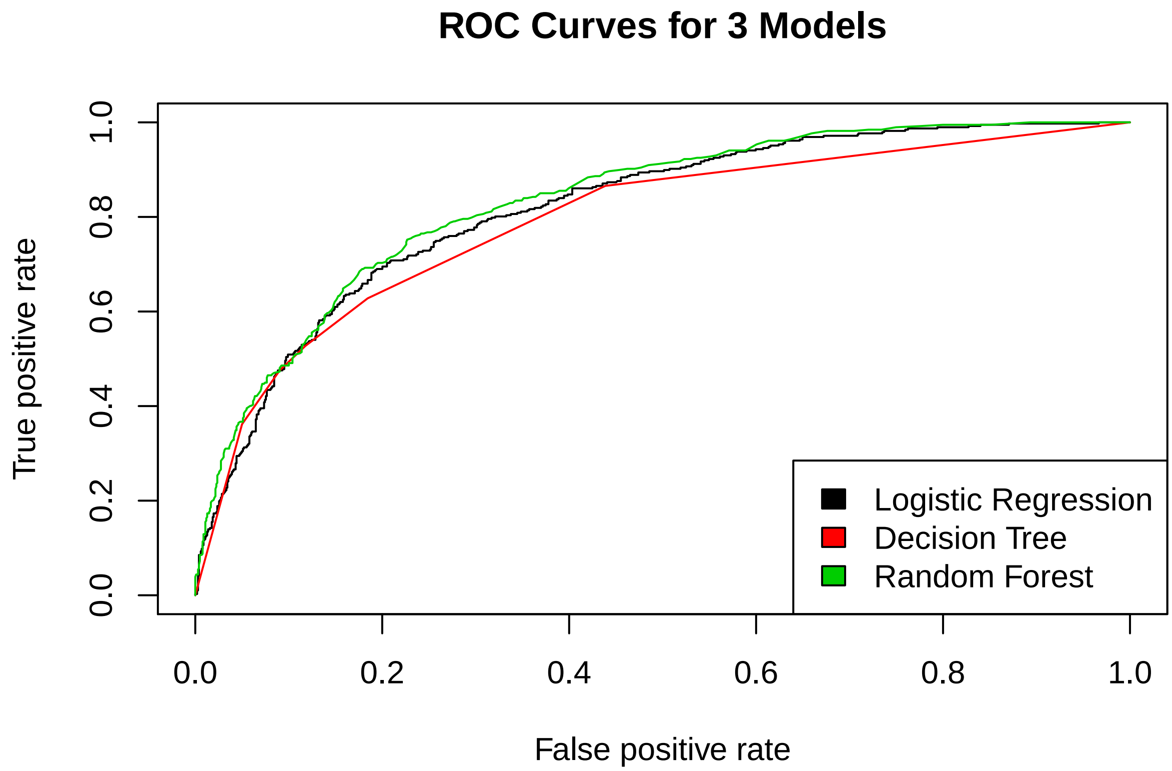
For the training set, the Accuracy is 0.974 and the AUC is 0.994. For the testing set, the Accuracy is 0.792 and the AUC is 0.832. Therefore, the model is overfitting.

Comparison of ROC for the three models

For this project, we are more willing to focus on the customer who quit the service, so it is important to research on the "yes" group. Therefore, ROC Curve is important for us.

```r
pre_list <- list(test_prob, testDT_prob[,2], testRF_prob[,2])
m <- length(pre_list)
testDT$Churn <- ifelse(testDT$Churn == "Yes", 1, 0)
actual_list <- rep(list(testDT$Churn), m)

pre <- prediction(pre_list, actual_list)
rocs <- performance(pre, "tpr", "fpr")
plot(rocs, col = as.list(1:m), main = "ROC Curves for 3 Models")
legend(x = "bottomright",
       legend = c("Logistic Regression", "Decision Tree", "Random Forest"),
       fill = 1:m)
```



Each point in ROC curve represents classification result (probability) compared to a predetermined cut-off value; AUC is the probability that randomly chosen positive samples is ranked above randomly chosen nagative ones. From the plot above, we can conclude that random forest and logistic regression perform better than decision tree. In total, these three model all perform good for the testing dataset. In the future analysis, it is better to usage some other ensamble skills to increase the accuracy and AUC value.

Discussion

In this project, we build three models for prediction of customer churn in teleco, and logistic regression performs best of the three models. Although random forest is an overfitting model in this project, it also has high accuracy for testing dataset, so we can also use random forest model in prediction. Random forest gives better results with the increasing number of examples. It might be used for clustering, statistical inference and feature selection as well, and it Works good with numerical and categorical data.

For the logistic regression model, in the future analysis, we can try and test different cut-off values' performance, and then we can choose the cut-off value with the highest accuracy.

In addition, except for the random forest model, we can use other resemble skills, such as bagging the support vector machine, logistic regression, or other data mining method to got a better results of prediction.

Reference

Heintz, Brenner. (2018). Cutting the Cord: Predicting Customer Churn for a Telecom Company. Retrieved from https://towardsdatascience.com/cutting-the-cord-predicting-customer-churn-for-a-telecom-company-268e65f177a5

Telco Customer Churn. https://www.kaggle.com/blastchar/telco-customer-churn