Customer Churn Analysis

Gikundi Frankline Murithi and Muchai Philip Njeri

Department of Pure and Applied Sciences, Kirinyaga University

PA102: Actuarial Science

Dr. Martin Kithiji

Abstract

This paper explores the use of machine learning algorithms to predict customer churn. We compare the performance of Support Vector Machine, Logistic Regression, Random Forest and Naïve Bayes models on a real world dataset of customer behavior. We find that the Random Forest model performs the best with an accuracy of 100%. This paper aims to determine reasons as to why customer withdraw, fit appropriate models that classifies and predicts customer churning. We check the model which best predict customer status using the R software.

Keywords: customer churn, machine learning, classification

Customer Churn Analysis

Introduction

Customer Churn, refers to the phenomenon where customers or subscribers disengage from a business relationship by discontinuing the use of specific services or products.

Customer churn rate: represents the proportion of customers who have withdrawn from the business, signaling the need for effective retention strategies. Customer churn can have various causes such as dissatisfaction with the quality or value of your products or services, poor customer service, lack of engagement or loyalty, competitive offers or changing customer needs or preferences. One of the main aim of Customer Churn prediction is to help in establishing strategies for customer retention. The risk of customer churn increases exponentially with the growing competition in markets for providing services. Hence, the need to establish strategies to keep track of loyal customers is vital.

The Telecom Churns can be classified into two main categories: Involuntary and Voluntary. Involuntary churn is those customers whom the Telecom industry decides to remove as a subscriber. They are churned for fraud, non-payment and those who don‘t use the service. Involuntary are easier to identify while voluntary churn is difficult to determine because it is the decision of the customer to unsubscribe from the service provider. Voluntary churn can further be classified as incidental and deliberate churn. The former occurs without any prior planning to churn but due to change in the financial condition, location, etc. The latter occurs when customers intentionally terminate their relationship with a business to switch to a competitor for better options. Customers may deliberately churn due to various reasons such as; Better pricing, desirable features, brand loyalty etc.

Most operators are trying to deal with these types of churns mainly.

In order to tackle this problem, machine learning has proved itself as a highly efficient technique, for forecasting information on the basis of previously captured data, which includes Support Vector Machine, Naïve Bayes Classifier, logistic Regression model and Random Forest.

If the reason for churning is known, the providers can then improve their services to fulfill the needs of the customers. Churns can be reduced by analyzing the past history of the potential customers systematically.

Literature Review

Churn prediction and management have become of great concern to the mobile operators. Mobile operators wish to retain their subscribers and satisfy their needs. Hence, they need to predict the possible churners and then utilize the limited resources to retain those customers. This paper reviews the work done in customer churn prediction using machine learning techniques and mainly focuses on methodology rather than the area of application.

(Hadden, 2017)Analyze the variables that impact churn in reverence. They also provided the comparative study of three machine learning models such as neural network, regression trees and regression. The obtained results confirm that decision tree is superior than others due to its rule based architecture. The obtained accuracy can be further improved using the existing feature selection techniques.

(umman, 2016),analyzed the mass data base using logistic regression and decision tree machine learning models but, obtained accuracy was low. Therefore, further improvement is required for that other machine learning and feature selection techniques can be adopted

In response to the difficulty of churner prediction, (Chang's, 2004). study applies data mining techniques to build a model for churner prediction through an analysis result from a big Taiwan telecom provider, the results indicated that the proposed approach has pretty good prediction

accuracy by using customer demography, billing information, call detail records, and service changed log to build churn prediction mode by using Artificial Neural Networks.

(Huang, 2020),applied various classifiers on churn prediction dataset in which the obtained results confirmed that random forest performs superior than others in terms of AUC(Area Under Curve) and PR-AUC(Precision Recall- Area Under Curve) analysis. But, accuracy can be further improved using the optimization techniques for the feature extraction.

There is a research gap on the lack of adequate studies that explore the role of technology in predicting and preventing customer churn. As technology becomes more advanced and more data becomes available, there is an opportunity to explore how machine learning algorithms can be used to identify customers who are at risk of churning and develop targeted interventions to prevent them from leaving.

Data modelling using machine learning

Machine learning modeling is the process of training and building predictive or descriptive models using machine learning algorithms. The process:

1. Model Selection: Choose an appropriate machine learning algorithm or ensemble of algorithms based on the problem type (classification, regression, clustering, etc.), the nature of the data, and the performance requirements.

2. Training the Model: The selected model is trained on the prepared dataset. The data is split into training and testing sets, and the model learns from the training data to capture the underlying patterns and relationships.

3. Evaluation of the model: It assess the performance on the test datasets and reports its accuracy precision, recall and F1 score.

We evaluate various classification modelling techniques to check which best classifies and predict customer churning.

1. Support Vector Machine.
2. Logistic Regression
3. Naive Bayes Classifier
4. Random Forest.

1. Support Vector Machine

It seeks to find an optimal (best) hyperplane that best separates two classes in a dataset. Mathematically, Support vector machine involves maximizing the margin between classes while minimizing the norm of the weight of vector, subject to the constraint that each point is correctly classified within a specified margin.

Decision boundary

Decision boundary is the main separator for dividing the points into their respective classes.

The hyperplane equation dividing the points (for classifying) can be written as:

$$\boldsymbol{H}: \boldsymbol{W^T(X) + b = 0} \quad \textbf{(1)}$$

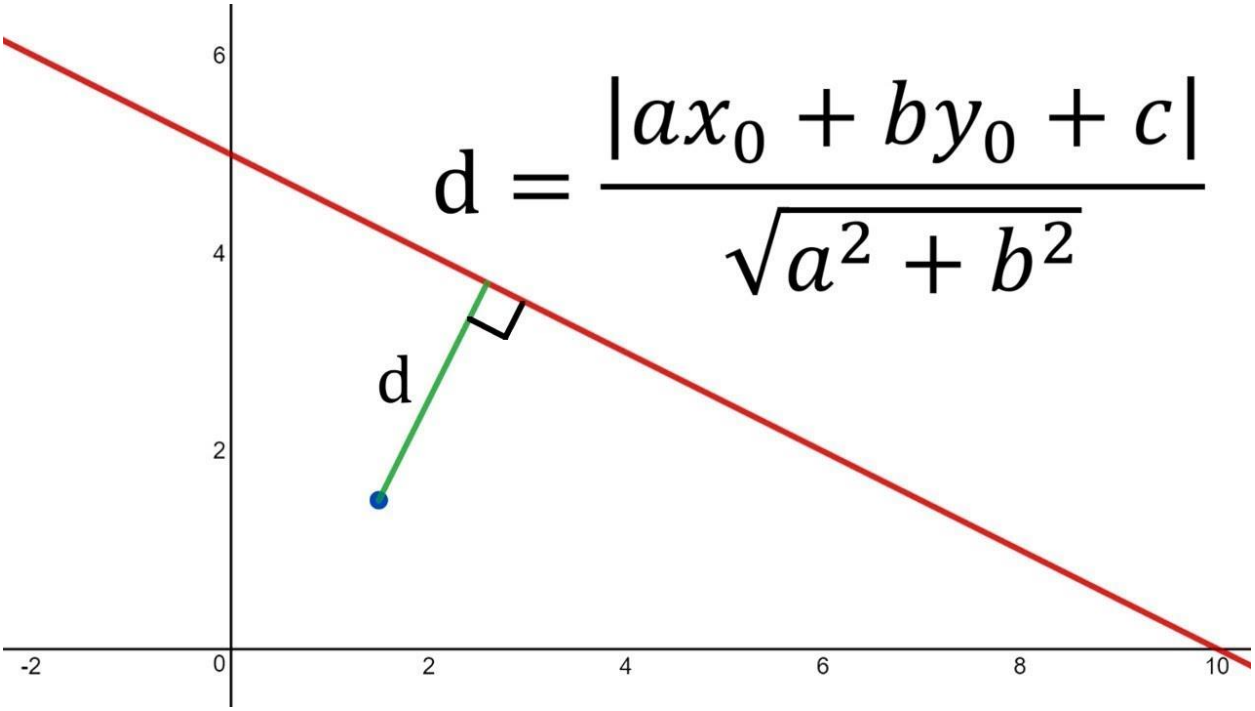
Where: b = intercept and the bias term of the hyperplane equation.

W^T = is the slope of the equation

Distance measure

Distance measure is used to calculate the distance between data points in the feature space. The distance between data points is then used to determine the decision boundary in the SVM in order to have the least errors in the classification of the data points.

+



The distance of any line, $ax + by + c = 0$ from a given point say, (x_0,y_0) is given by d.

Similarly, the distance of a hyperplane equation**:** : $\boldsymbol{W^T\phi(x) + b = 0}$

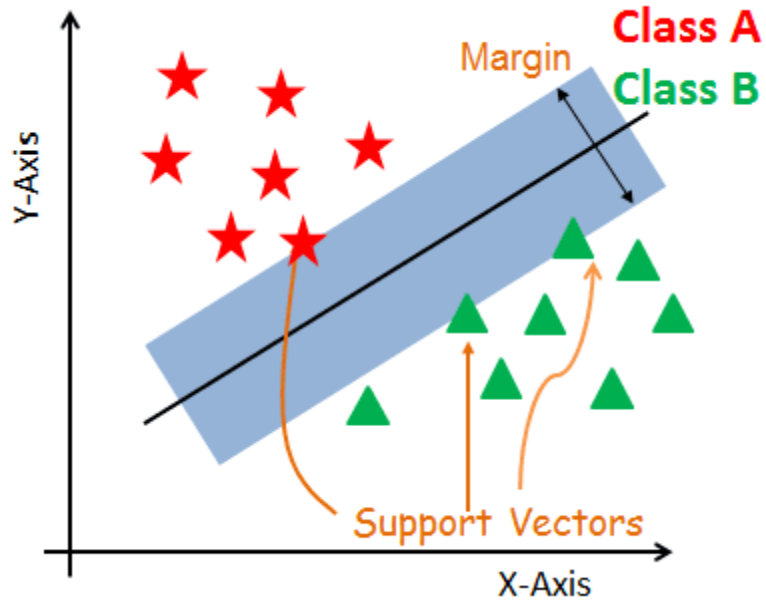
From a given point vector $\boldsymbol{\phi(x_0)}$ can be easily written as**:**

$$\boldsymbol{d_H}\left(\phi(\boldsymbol{x_0})\right) = \frac{|\boldsymbol{w^T}(\phi(\boldsymbol{x_0}))+\boldsymbol{b}|}{||\boldsymbol{w}||_2} \quad \textbf{(1)}$$

Here ||w||2 is the Euclidean norm for the length of w given by:

$$||w||_2 = \text{sqrt}(w_1^2 + w_2^2 + w_3^2 + \cdots w_n^2) \quad \textbf{(2)}$$

To choose the best hyperplane, we place the hyperplane at the center where the distance is maximum from the closest point to the least test errors further.



to sum up;

1. When given any test feature vector x to predict, mapped to a complex Φ(x) and asked to predict which is basically $W^T\phi(x)$
2. Then rewrite it using the duals so that the prediction is completely independent of complex feature basis Φ
3. And is further predicted using kernels

$$W^T\phi(x) = \sum_n \alpha_n y_n k(x_n, x) \quad (3)$$

2. Naïve Bayes Classifier

Naive Bayes classifier is a probabilistic model based on Bayes' theorem.

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)} \quad (4)$$

In this equation, using Bayes theorem we can find the probability of A, given that B occurred. A is the hypothesis and B is the evidence.

P(B|A) is the probability of B given that A is true and P(A) and P(B) are independent probabilities of A and B.

Assumptions:

- Features are conditionally independent of each other.
- Each of the features is equal in terms of weightage and importance.
- Features follows a normal distribution.
- There is no correlation among the features.

3. Logistic Regression

Logistic regression is a statistical method used to model the probability of a binary outcome based on one or more predictor variables. The logistic regression model uses the logistic function to model the relationship between the predictor variables and the probability of the outcome. Here is the mathematical description of logistic regression:

The logistic function, also known as the sigmoid function, is defined as:

$$F(x) = \frac{1}{1+e^{-x}} \quad (6)$$

Where:

(x) is the linear combination of the predictor variables and their coefficients.

The linear combination of the predictor variables and their coefficients is calculated as:

$$x = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p \quad (7)$$

Where:

(β_0) is the intercept or bias term.

($\beta_1, \beta_2, \ldots, \beta_p$) are the coefficients associated with the predictor variables.

($x_1, x_2, \ldots, x_p$) are the predictor variables.

The logistic regression model can be written as:

$$P(Y = 1|X) = \frac{1}{1+e^{-[\beta_0+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_p x_p]}} \quad (8)$$

Where:

(P(Y=1|X)) is the probability of the binary outcome being 1 given the predictor variables.

(e) is the base of the natural logarithm.

Estimating coefficients: The coefficients ($\beta_1, \beta_2, \ldots, \beta_p$) are estimated using maximum likelihood estimation, which aims to find the values of the coefficients that maximize the likelihood of observing the given data. The likelihood function (L) for the entire dataset is the product of the individual probabilities for each observation:

$$L(\beta_0,\beta_1,\beta_2,\ldots,\beta_p) = \prod_{i=1}^{n} p(y_i = 1|x_i)^{y_i} 1 - p((y_i = 1|x_i)^{(1-y_i)} \quad (9)$$

Where:

(n) is the number of observations.

(y_i) is the actual outcome for the (i)th observation.

(X_i) represents the predictor variables for the (i)th observation.

Log-likelihood function: To simplify the calculations, the log-likelihood function is often used. The log-likelihood function is the natural logarithm of the likelihood function:

$$l(\beta_0,\beta_1,\beta_2,\ldots,\beta_p) = \sum_{i=1}^{n} y_i \ln p(y_i = 1|x_i)^{y_i} + (1 - y_i) \ln(1 - p(y_i = 1|x_i) \quad (10)$$

Maximizing the log-likelihood: The goal of MLE is to find the values of the coefficients ($\beta_0,\beta_1,\beta_2,\ldots,\beta_p$) that maximize the log-likelihood function. This is typically done using numerical optimization methods such as gradient descent, Newton-Raphson, or other optimization algorithms.

The Newton-Raphson equation is given as

$$\beta^{t+1} = \beta^t - \frac{\nabla_\beta l(\beta^t)}{\nabla_{\beta\beta} l(\beta^t)} \quad (11)$$

Where,

$\nabla_\beta l(\beta^t)$ is the first order derivative(gradient)

$\nabla_{\beta\beta} l(\beta^t)$is the second order derivative

t is the current iteration number

To determine gradient we take the first-order derivative of log-likelihood function

$$\nabla_\beta l = \nabla_\beta \sum_{i=1}^{n} y_i \beta x_i - \log\left(1 + e^{\beta x_i}\right) \tag{12}$$

$$\nabla_\beta l = \sum_{i=1}^{n} \nabla_\beta [y_i \beta x_i - \log\left(1 + e^{\beta x_i}\right)] \tag{13}$$

$$\nabla_\beta l = \sum_{i=1}^{n} \nabla_\beta [y_i \beta x_i] - \nabla_\beta [\log\left(1 + e^{\beta x_i}\right)] \tag{14}$$

$$\nabla_\beta l = \sum_{i=1}^{n} y_i x_i - [\frac{1}{\left(1+e^{\beta x_i}\right)} * e^{\beta x_i} x_i] \tag{15}$$

$$\nabla_\beta l = \sum_{i=1}^{n} y_i x_i - [\frac{1}{\left(1+e^{-\beta x_i}\right)} * x_i] \tag{16}$$

Now, we replace the exponential term with probability

$$\nabla_\beta l = \sum_{i=1}^{n} y_i x_i - [p(x_i) * x_i] \tag{17}$$

$$\nabla_\beta l = \sum_{i=1}^{n} [y_i - p(x_i)] * x_i \tag{18}$$

The matrix representation of gradient will be

$$\nabla_{\beta} l = X^T (Y - \hat{Y}) \tag{19}$$

We are done with the numerator term of Newton-Raphson. Now we will calculate the denominator i.e. second-order derivative which is also called as Hessian Matrix.

To do so we will find derivate of gradient as:

$$\nabla_{\beta\beta} l = \nabla_{\beta} \sum_{i=1}^{n} [y_i - p(x_i)] * x_i \tag{20}$$

$$\nabla_{\beta\beta} l = \sum_{i=1}^{n} \nabla_{\beta} [y_i - p(x_i)] * x_i \tag{21}$$

$$\nabla_{\beta\beta} l = \sum_{i=1}^{n} \nabla_{\beta} p(x_i) * x_i \tag{22}$$

Now, we will replace probability with its equivalent exponential term and compute its derivative

$$\nabla_{\beta\beta} l = \sum_{i=1}^{n} \nabla_{\beta} [\frac{1}{(1+e^{-\beta x_i})}] * x_i \tag{23}$$

$$\nabla_{\beta\beta} l = \sum_{i=1}^{n} \left[\frac{1}{(1+e^{-\beta x_i})}\right]^2 e^{-\beta x_i} (-x_i) x_i \tag{24}$$

$$\nabla_{\beta\beta} l = \sum_{i=1}^{n} \left[\frac{e^{-\beta x_i}}{(1+e^{-\beta x_i})}\right] [\frac{1}{(1+e^{-\beta x_i})}] {x_i}^T x_i \quad (25)$$

Resubstitute exponential term as probability

$$\nabla_{\beta\beta} l = \sum_{i=1}^{n} p(x_i)(1-p(x_i)) {x_i}^T x_i \quad (26)$$

The matrix representation of the Hessian matrix will be

$$\nabla_{\beta\beta} l = -X^T P(1-P)\mathrm{X} \quad (27)$$

$$\nabla_{\beta\beta} l = -X^T W\mathrm{X} \quad (28)$$

As we have calculated gradient and Hessian matrix, plugging these two terms into the Newton-Raphson equation to get a final form

$$\beta^{t+1} = \beta^t - \frac{\nabla_\beta l(\beta^t)}{\nabla_{\beta\beta} l(\beta^t)} \quad (29)$$

$$\beta^{t+1} = \beta^t - \frac{X(Y-\hat{Y}^{(t)})}{(-X^T W^{(t)} X)} \quad (30)$$

$$\beta^{t+1} = \beta^t + X(Y - \hat{Y}^{(t)})(X^T W^{(t)} X)^{-1} \quad (31)$$

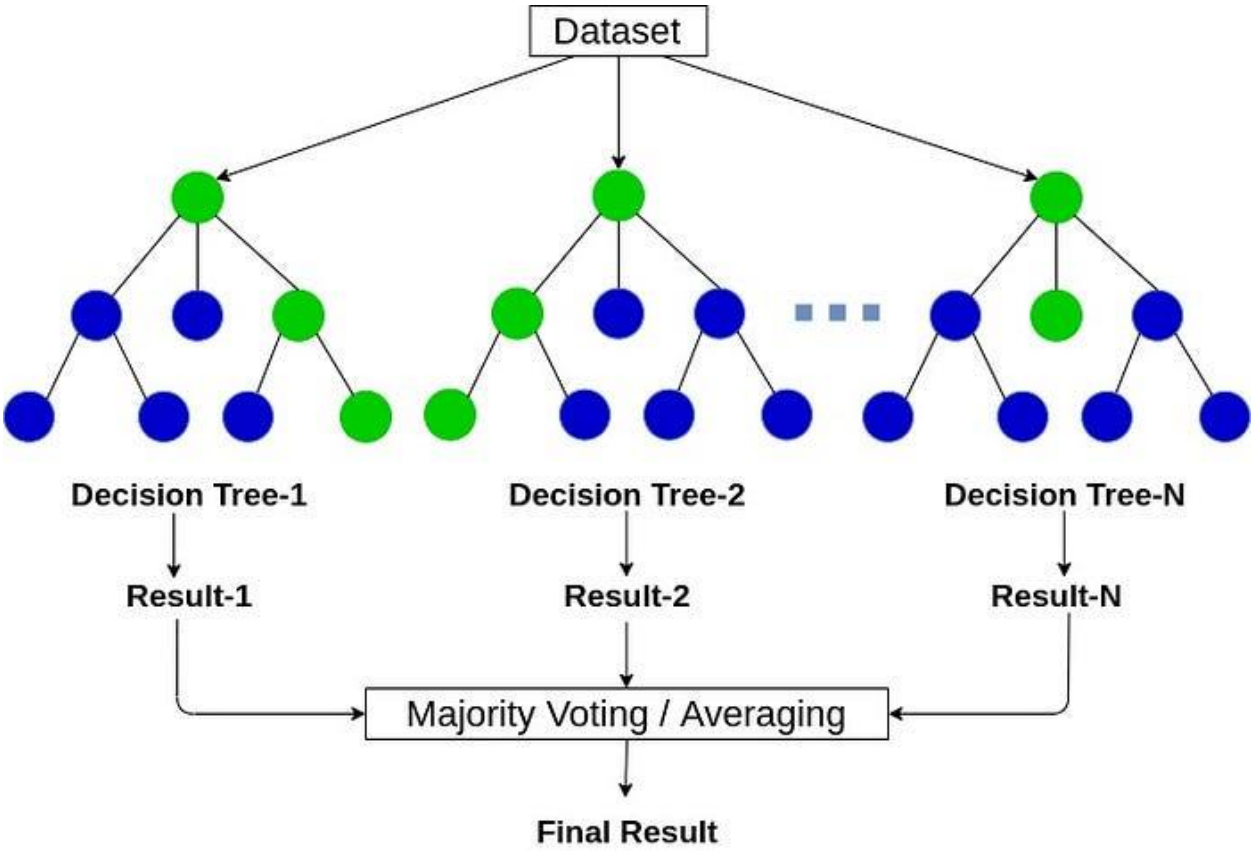
Now, we will execute the final equation for t number of iterations until the value of β converges.

Once the coefficients have been estimated we can then plug in the values of some feature vector X to estimate the probability of it belonging to a specific class.

We should choose a threshold value above which belongs to class 1 and below which is class 0.
Estimating the coefficients: Once the maximum of the log-likelihood function is found, the corresponding values of the coefficients are the MLE estimates of the logistic regression model.

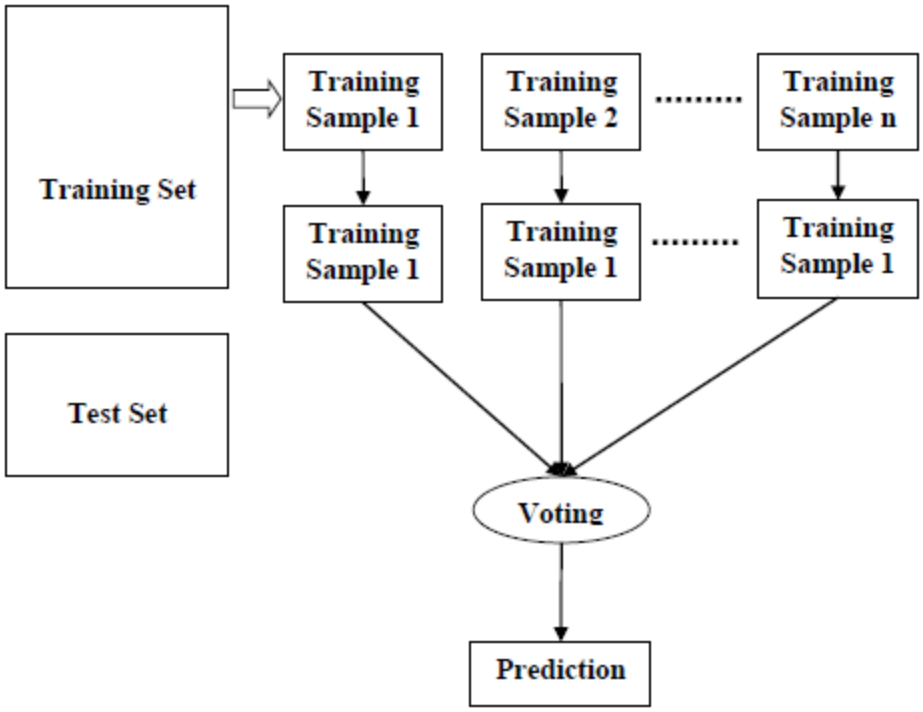
4. **Random Forest**

It works by constructing multiple decision trees from random subsets of the training data. Each decision tree is then used to make a prediction, and the final prediction is made by taking the average of all the individual predictions. This approach makes the algorithm more robust and less prone to overfitting. Random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting



Let us understand the working of Random Forest algorithm with the help of following steps −

- **Step 1** − First, start with the selection of random samples from a given dataset.
- **Step 2** − Next, this algorithm will construct a decision tree for every sample. Then it will get the prediction result from every decision tree.
- **Step 3** − In this step, voting will be performed for every predicted result.
- **Step 4** − At last, select the most voted prediction result as the final prediction result.



Random forest uses a combination of two key mathematical concepts: bagging and random feature selection. Bagging, or bootstrap aggregating, is a technique that involves training multiple decision trees on different subsets of the training data. Random feature selection is a technique that randomly selects a subset of features to use when training each individual decision tree. These two techniques work together to make random forest more robust and less prone to overfitting.

Results and Discussion

Descriptive Analysis

This data set was retrieved from kaggle website.

The data contains information about various customer attributes and their subscription services, which can be used to predict customer churn.

The structure of the data set shows that it is a data frame with 7043 observations (rows) and 38 variables (columns). Each observation represents a customer of a telecom company and each variable contains some information about the customer, such as their demographic, geographic, service, and billing details. The data set also includes the customer status, churn category, and churn reason for each customer, which indicate whether they stayed or left the company and why.

```
'data.frame':   7043 obs. of  38 variables:
 $ Customer.ID                      : chr  "0002-ORFBO" "0003-MKNFE" "0004-TLHLJ" "0011-IGKFF" ...
 $ Gender                           : chr  "Female" "Male" "Male" "Male" ...
 $ Age                              : int  37 46 50 78 75 23 67 52 68 43 ...
 $ Married                          : chr  "Yes" "No" "No" "Yes" ...
 $ Number.of.Dependents             : int  0 0 0 0 0 3 0 0 0 1 ...
 $ City                             : chr  "Frazier Park" "Glendale" "Costa Mesa" "Martinez" ...
 $ Zip.Code                         : int  93225 91206 92627 94553 93010 95345 93437 94558 93063 95681 ...
 $ Latitude                         : num  34.8 34.2 33.6 38 34.2 ...
 $ Longitude                        : num  -119 -118 -118 -122 -119 ...
 $ Number.of.Referrals              : int  2 0 0 1 3 0 1 8 0 3 ...
 $ Tenure.in.Months                 : int  9 9 4 13 3 9 71 63 7 65 ...
 $ Offer                            : chr  "None" "None" "Offer E" "Offer D" ...
 $ Phone.Service                    : chr  "Yes" "Yes" "Yes" "Yes" ...
 $ Avg.Monthly.Long.Distance.Charges: num  42.39 10.69 33.65 27.82 7.38 ...
 $ Multiple.Lines                   : chr  "No" "Yes" "No" "No" ...
 $ Internet.Service                 : chr  "Yes" "Yes" "Yes" "Yes" ...
 $ Internet.Type                    : chr  "Cable" "Cable" "Fiber Optic" "Fiber Optic" ...
 $ Avg.Monthly.GB.Download          : int  16 10 30 4 11 73 14 7 21 14 ...
 $ Online.Security                  : chr  "No" "No" "No" "No" ...
 $ Online.Backup                    : chr  "Yes" "No" "No" "Yes" ...
 $ Device.Protection.Plan           : chr  "No" "No" "Yes" "Yes" ...
 $ Premium.Tech.Support             : chr  "Yes" "No" "No" "No" ...
 $ Streaming.TV                     : chr  "Yes" "No" "No" "Yes" ...
 $ Streaming.Movies                 : chr  "No" "Yes" "No" "Yes" ...
 $ Streaming.Music                  : chr  "No" "Yes" "No" "No" ...
 $ Unlimited.Data                   : chr  "Yes" "No" "Yes" "Yes" ...
 $ Contract                         : chr  "One Year" "Month-to-Month" "Month-to-Month" "Month-to-Month" ...
 $ Paperless.Billing                : chr  "Yes" "No" "Yes" "Yes" ...
 $ Payment.Method                   : chr  "Credit Card" "Credit Card" "Bank Withdrawal" "Bank Withdrawal" ...
 $ Monthly.Charge                   : num  65.6 -4 73.9 98 83.9 ...
 $ Total.Charges                    : num  593 542 281 1238 267 ...
 $ Total.Refunds                    : num  0 38.3 0 0 0 ...
 $ Total.Extra.Data.Charges         : int  0 10 0 0 0 0 0 20 0 0 ...
 $ Total.Long.Distance.Charges      : num  381.5 96.2 134.6 361.7 22.1 ...
 $ Total.Revenue                    : num  975 610 415 1600 290 ...
 $ Customer.Status                  : chr  "Stayed" "Stayed" "Churned" "Churned" ...
 $ Churn.Category                   : chr  "" "" "Competitor" "Dissatisfaction" ...
 $ Churn.Reason                     : chr  "" "" "Competitor had better devices" "Product dissatisfaction" ...
```

Analysis

Summary distribution of the numerical features.

The Total.Revenue variable is the total amount of money that each customer has paid to the company over a year period. The boxplot shows that the median Total.Revenue is around $1,200, the interquartile range is from about $600 to $2,000, and there are some outliers above $4,000. This means that most customers have paid between $600 and $2,000 to the company, but some customers have paid much more than that. The boxplot also shows that the data is skewed to the right, meaning that there are more customers with lower Total.Revenue than higher Total.Revenue.

The Avg.Monthly.GB.Download variable is the average amount of gigabytes that each customer downloads from the internet every month. The boxplot shows that the median value is around $40. The first quartile is around $20 and the third quartile is around $60. The minimum value is around $0 and the maximum value is around $80. There are several outliers above the upper quartile, which means that there are some customers who have paid much more than the rest of the data for the average monthly gigabyte download.

The Monthly.Charge variable is the amount of money that each customer pays to the company every month for the services they use. The median value is around 60 months. The first quartile is around 40 months and the third quartile is around 80 months. The minimum value is around 20 months and the maximum value is around 100 months. This means that the lowest and highest values of the data are 20 and 100 months. The boxplot also shows that the data is symmetric, meaning that the data is evenly distributed around the median. There are no outliers in the data, meaning that there are no data points that are very far from the rest of the data.

The boxplot of Total Long Distance Charges over a certain period. The median of the data is about $1500, which means that half of the charges are above this value and half are below. The interquartile range (IQR) is about $1000, which means that 50% of the charges are between

$1000 and $2000. The minimum and maximum values of the data are about $500 and $3000, respectively.

The Avg.Monthly.Long.Distance.Charges variable is the average amount of money that each customer pays to the company every month for the long distance calls they make. The boxplot has a range from 0 to 50 months. The median value is around 30 months. The lower quartile is around 20 months and the upper quartile is around 40 months. The minimum value is around 0 months and the maximum value is around 50 months. This means that the lowest and highest values of the data are 0 and 50 months.

Relationship between categorical variable against customer churn

On marriage it shows that most churned customers are those unmarried.

On contract it's evident that those who have contract for month-to-month have churned the most compared to one year and two year contracts.

On the internet type those with fiber optic have highest number of customers who stayed, but also a significant number who churned. DSL has a balanced ratio of joined and stayed customers with very few churns. Cable has the lowest total customers with more customers having churned than those who have joined or stayed.

It appears that having streaming TV is associated with higher customer retention (staying) and lower rates compared to customers without streaming TV

Having unlimited data seems to positively impact customer retention customers without unlimited data have a higher likelihood of churning the majority of customers with unlimited data choose to stay

A higher number of females stayed as customers compared to those who churned this suggest that females tend to have better retention rates. The numbers are almost equal for both stayed and

churned male customers. There is no significant difference in retention rates between males who stayed and those who churned.

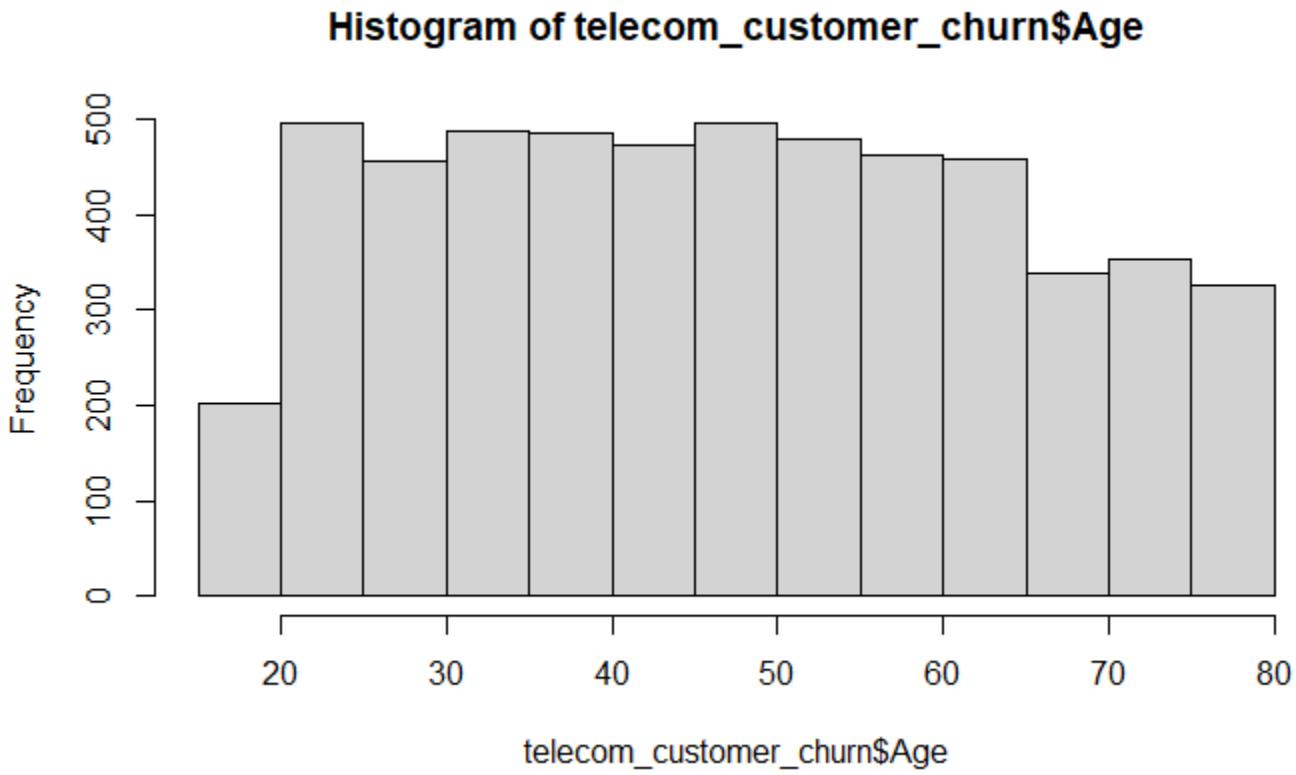
The number of customers with online backup is significantly lower than those who stayed while for the customers without online backup have a significant number that has churned almost equal to those who stayed.

Having premium tech support does not necessarily lead to better customer retention while significant number of customers without premium tech support stayed, a majority of those with premium tech support churned

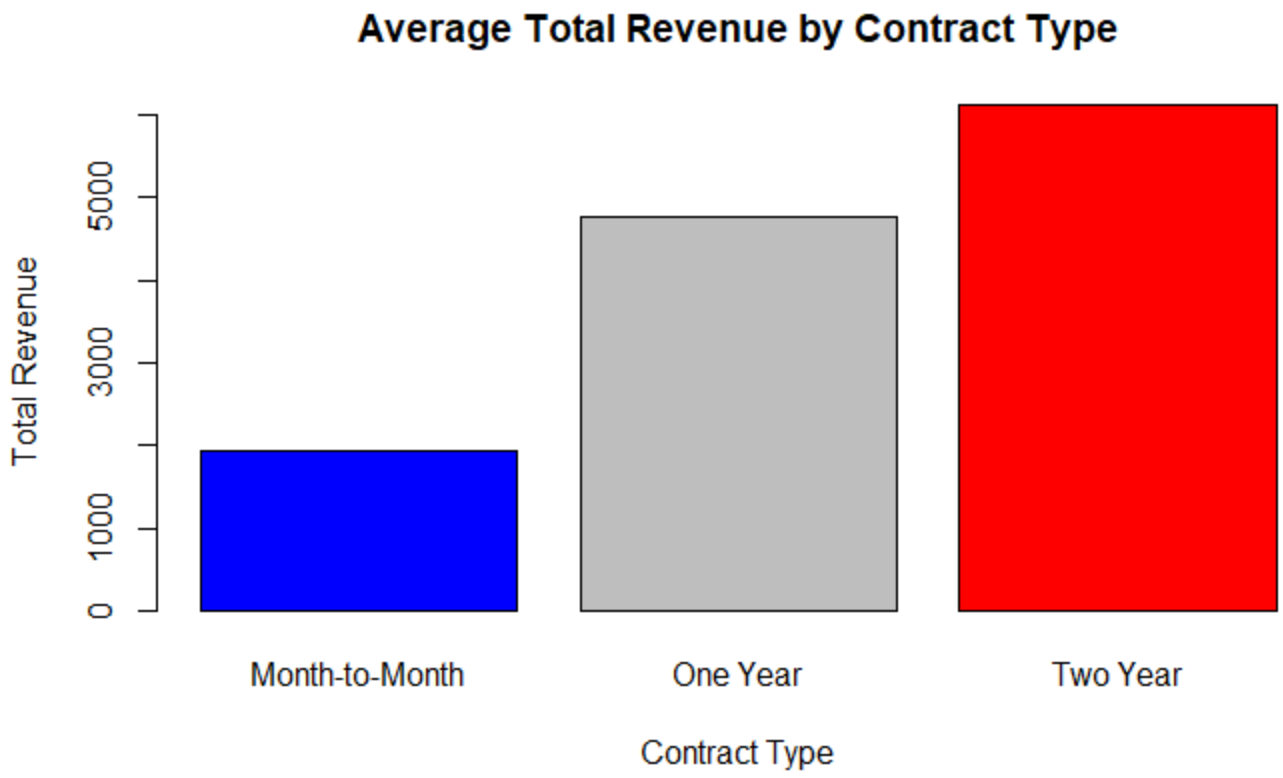
Many customers without streaming music stayed, an equal number of new joiners and stayers have streaming music and it appears that having streaming music does not significantly impact customer retention

Having online security appears to positively impact customer retention. While many customers without online securities stayed, none of those with online securities churned.

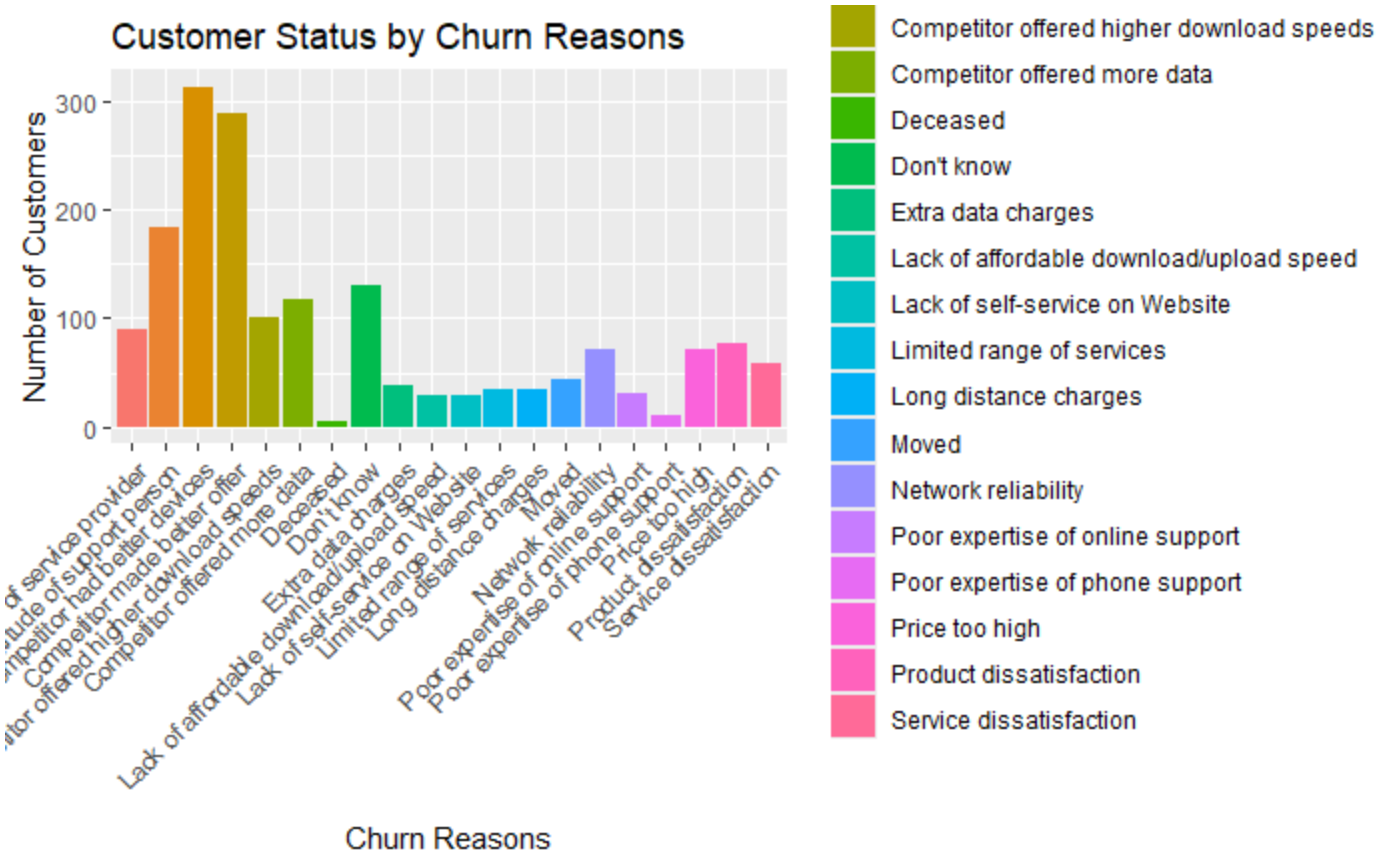
Having streaming movies appears to positively impact customer retention. While many customers without streaming movies stayed, none of those with streaming movies churned.



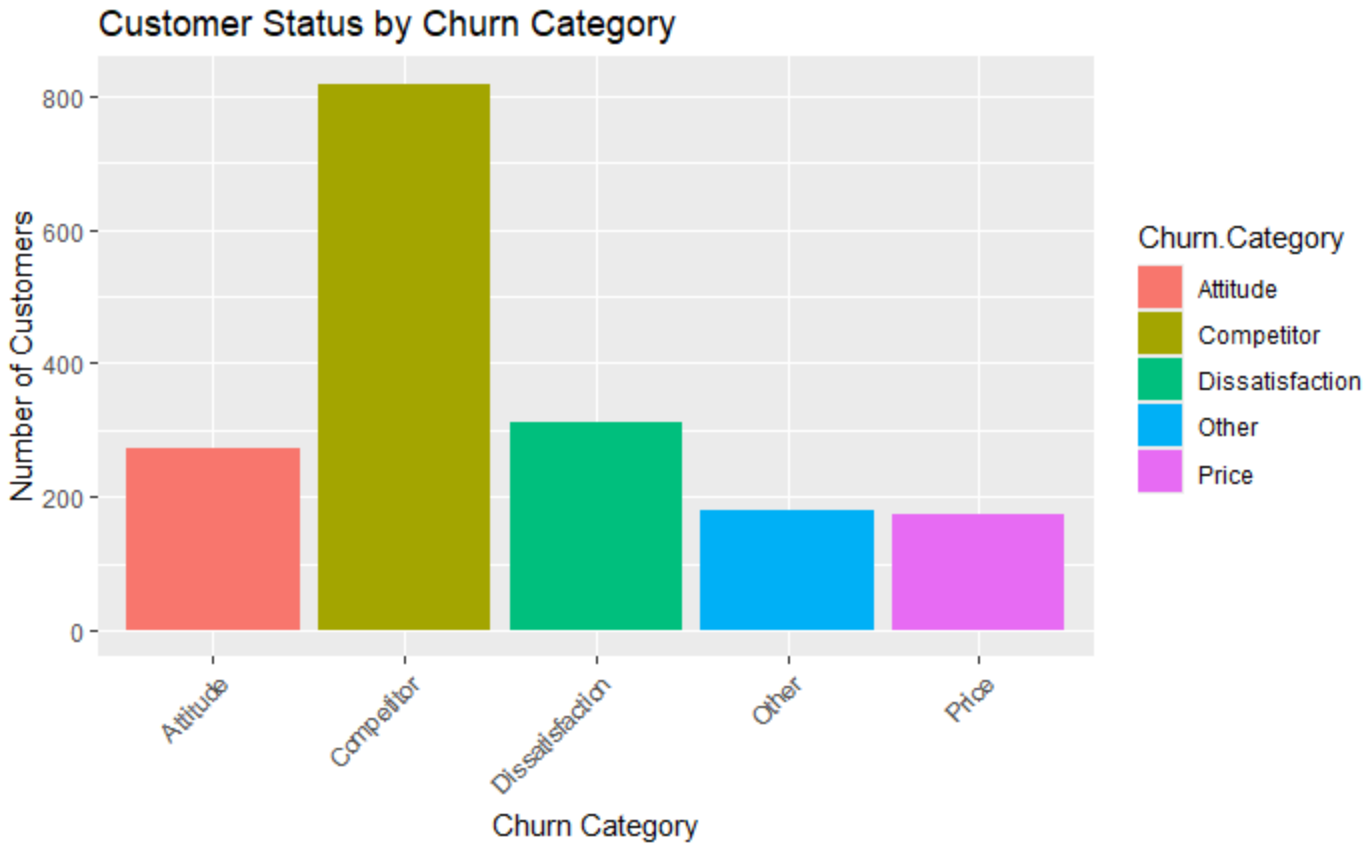
The histogram has a peak or mode at the 40-50 age group, meaning that this is the most frequent bin. This suggests that customers in their 40s and 50s are the most likely to churn, and therefore the most important target group for retention strategies.



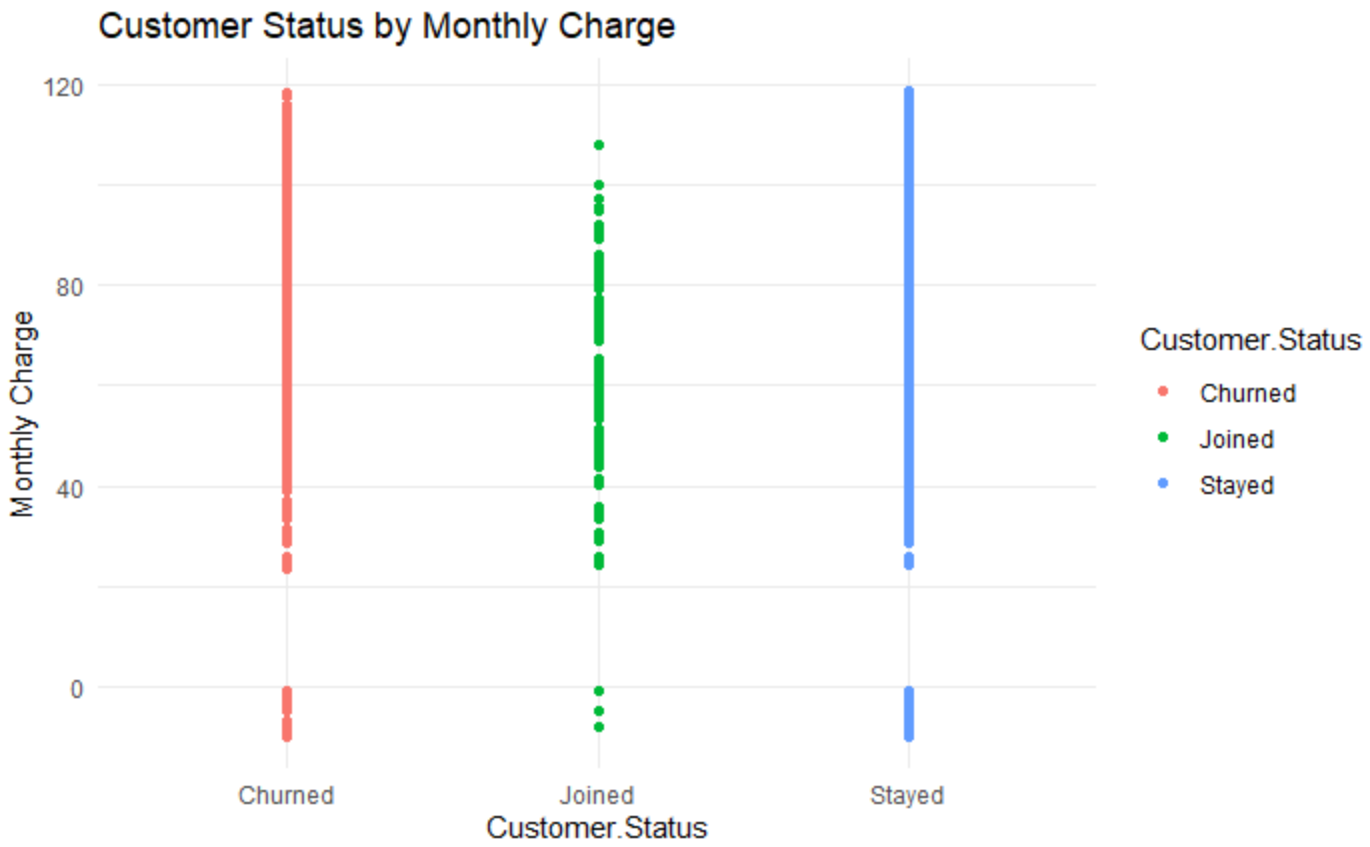
The graph reveals that customers who have longer contracts tend to pay more money to the telecom company than customers who have shorter contracts. The two year contracts generate the highest average revenue, followed by the one year contracts, and then the month-to-month contracts.



the graph reveals that attitude of service provider,attitude of support provider, competitor had better devices, competitor offered higher download speeds, competitor offered more data, don't know, network reliability, price too high, product dissatisfaction, service dissatisfaction are the one that has higher number of customers who churned.



Competitor seems to have and be the main reason why customers churn followed by attitude and dissatisfaction.



Customers who churned (red circles) tend to have higher monthly charges than customers who stayed (blue circles) or joined (green circles). This could indicate that customers who pay more are more likely to switch to other providers or cancel their service.

Customers who joined (green circles) have moderate monthly charges, mostly between 40 and 80. This could indicate that customers who sign up for the telecom service are attracted by reasonable prices or discounts.

Customers who stayed (blue circles) have lower monthly charges than customers who churned, but higher than most customers who joined. This could indicate that customers who remain loyal to the telecom company are satisfied with their service or have long-term contracts.

Data Modelling

We evaluated various classification modelling techniques to check which one classifies and predict the customers churning best.

1. Support Vector Machine.
2. Naive Bayes Classifier.
3. Random Forest.
4. Logistic Regression

To prepare data for modelling we split it in the ratio of 80:20 i.e. 80% in training set and 20% in testing where the testing is used in prediction

The models were correctly fitted and this is their performance in predicting customer churning.

Support Vector Machine

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Churned Joined Stayed
##    Churned     351      0      0
##    Joined        0     51      1
##    Stayed        0      3    697
##
## Overall Statistics
##
##                Accuracy : 0.9964
##                  95% CI : (0.9907, 0.999)
##     No Information Rate : 0.6328
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9927
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Churned Class: Joined Class: Stayed
## Sensitivity                  1.0000       0.94444        0.9986
## Specificity                  1.0000       0.99905        0.9926
## Pos Pred Value               1.0000       0.98077        0.9957
## Neg Pred Value               1.0000       0.99715        0.9975
## Prevalence                   0.3182       0.04896        0.6328
## Detection Rate               0.3182       0.04624        0.6319
## Detection Prevalence         0.3182       0.04714        0.6346
## Balanced Accuracy            1.0000       0.97175        0.9956
```

Naïve Bayes Classifier

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Churned Joined Stayed
##    Churned     262      0      0
##    Joined       66     53     24
##    Stayed       23      1    674
##
## Overall Statistics
##
##                Accuracy : 0.8966
##                  95% CI : (0.8772, 0.914)
##     No Information Rate : 0.6328
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.8003
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: Churned Class: Joined Class: Stayed
## Sensitivity                  0.7464       0.98148        0.9656
## Specificity                  1.0000       0.91420        0.9407
## Pos Pred Value               1.0000       0.37063        0.9656
## Neg Pred Value               0.8942       0.99896        0.9407
## Prevalence                   0.3182       0.04896        0.6328
## Detection Rate               0.2375       0.04805        0.6111
## Detection Prevalence         0.2375       0.12965        0.6328
## Balanced Accuracy            0.8732       0.94784        0.9532
```

Random Forest

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Churned Joined Stayed
##    Churned     351      0      0
##    Joined        0     54      0
##    Stayed        0      0    698
##
## Overall Statistics
##
##                Accuracy : 1
##                  95% CI : (0.9967, 1)
##     No Information Rate : 0.6328
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Churned Class: Joined Class: Stayed
## Sensitivity                  1.0000       1.00000        1.0000
## Specificity                  1.0000       1.00000        1.0000
## Pos Pred Value               1.0000       1.00000        1.0000
## Neg Pred Value               1.0000       1.00000        1.0000
## Prevalence                   0.3182       0.04896        0.6328
## Detection Rate               0.3182       0.04896        0.6328
## Detection Prevalence         0.3182       0.04896        0.6328
## Balanced Accuracy            1.0000       1.00000        1.0000
```

Logistic Regression

```
##
## predicted_labels Churned Joined Stayed
##                0     351      0      0
##                1       0     54    698
```

```
cat("Accuracy:", accuracy, "\n")
```

```
## Accuracy: 0.3671804
```

```
sensitivity <- confusion_matrix[2, 2] / sum(confusion_matrix[2, ])
precision <- confusion_matrix[2, 2] / sum(confusion_matrix[, 2])
specificity <- confusion_matrix[1, 1] / sum(confusion_matrix[1, ])
cat("Specificity (True Negative Rate):", specificity, "\n")
```

```
## Specificity (True Negative Rate): 1
```

```
cat("Sensitivity (Recall):", sensitivity, "\n")
```

```
## Sensitivity (Recall): 0.07180851
```

```
cat("Precision (Positive Predictive Value):", precision, "\n")
```

```
## Precision (Positive Predictive Value): 1
```

Findings:

Random Forest provides the most accuracy for predicting this dataset

Churn reasons:

- Competitor had better devices
- Competitor had better offers
- Attitude of the support persons
- Competitor offered more data
- Competitor offered best download speed

The following features had impacts on customer churning

I. Contract where those who had monthly contracts churned most

II. Internet type where customers with fibre optic churned most

III. Unlimited data where customers with unlimited data churned most

IV. Online security- customers without online security churned most

V. Premium tech support- customers without premium tech support churned most

VI. Streaming music-customers with no streaming music churned most

VII. Online backup -customers without online backup churned most

VIII. Married -customers who are not married churned most

Mitigations to retain churned customers

1) Improve on devices that are used by customers.
2) Improve the quality of the product and services – provide offers that substitute the competitor's
3) Improve on customer experience or relationship between customers and support persons
4) Educate the support person on how to handle customers with different attitudes etc

5) Improve on the technical infrastructure to enhance higher download speed.

Conclusion.

After analysis of customer churning we concluded that The customer experience you provide and your ability to engage customers—with intuitive onboarding, loyalty programs, and personalized communication and content—greatly impact a customer's desire to churn.

Recommendations

The following recommendations were made on the basis of the research findings.

1 We recommend the telecom industry to use Random Forest model for predicting customer churn.

2 The telecom industry should keep working on the customer experience

3 In future we suggest researchers to compare other models apart from the three with the Random Forest model to check which best predicts customer churning.

4 We suggest that the model used can be put in production so that they can make business decision based on data, interact with their applications.

References

Andrew H. Karp, Using logistic regression to predict customer retention

J. Pamina, B. Raja, S. SathyaBama, S. Soundarya, M. S. Sruthi, S. Kiruthika, V. J. Aiswaryadevi, and G. Priyanka, ``An effective classifier for predicting churn in telecommunication," *J. Adv. Res. Dyn. Control Syst.*, vol. 11, no. 10, pp. 221_229, Jun. 2019.

L. Yangi, C. Chiu, Subscriber Churn Prediction in Telecommunications,

N. Kamalraj, .A.Malathi, Applying Data Mining Techniques in Telecom Churn Prediction, in proc. International Journal of Advanced Research in Computer Science and Software Engineering, 10, October 2013.

S. Khotijah. (2020). *Churn Prediction*. [Online]. Available: https://www.kaggle.com/khotijahs1/churn-prediction

U. Ahmed, A. Khan, S. H. Khan, A. Basit, I. U. Haq, and Y. S. Lee, ``Transfer learning and meta classification based deep churn prediction system for telecom industry," 2019, *arXiv:1901.06091*. [Online]. Available: https://arxiv.org/abs/1901.06091

V. Umayaparvathi, K. Iyakutti, Applications of Data Mining Techniques in Telecom Churn Prediction, International Journal of Computer Applications (0975 –8887) Volume 42– No.20, March 2012

Y.T. Chang, Applying Data Mining to Telecom Churn Management. *International Journal of Reviews in Computing.* 69-77, (2009).

Customer churn analysis code

Philip and Frankline

Data Collection.

```
telecom_customer_churn <- read.csv("C:/Users/lenovo/Desktop/telecom_customer_
churn.csv")

View(telecom_customer_churn)
```

#Data preparation.#

```
sapply(telecom_customer_churn, function(x) sum(is.na(x)))
```

```
##                         Customer.ID                              Gender
##                                   0                                   0
##                                 Age                             Married
##                                   0                                   0
##               Number.of.Dependents                                City
##                                   0                                   0
##                            Zip.Code                            Latitude
##                                   0                                   0
##                           Longitude                 Number.of.Referrals
##                                   0                                   0
##                    Tenure.in.Months                               Offer
##                                   0                                   0
##                       Phone.Service Avg.Monthly.Long.Distance.Charges
##                                   0                                 682
##                     Multiple.Lines                    Internet.Service
##                                   0                                   0
##                     Internet.Type             Avg.Monthly.GB.Download
##                                   0                                1526
##                   Online.Security                     Online.Backup
##                                   0                                   0
##             Device.Protection.Plan                Premium.Tech.Support
##                                   0                                   0
```

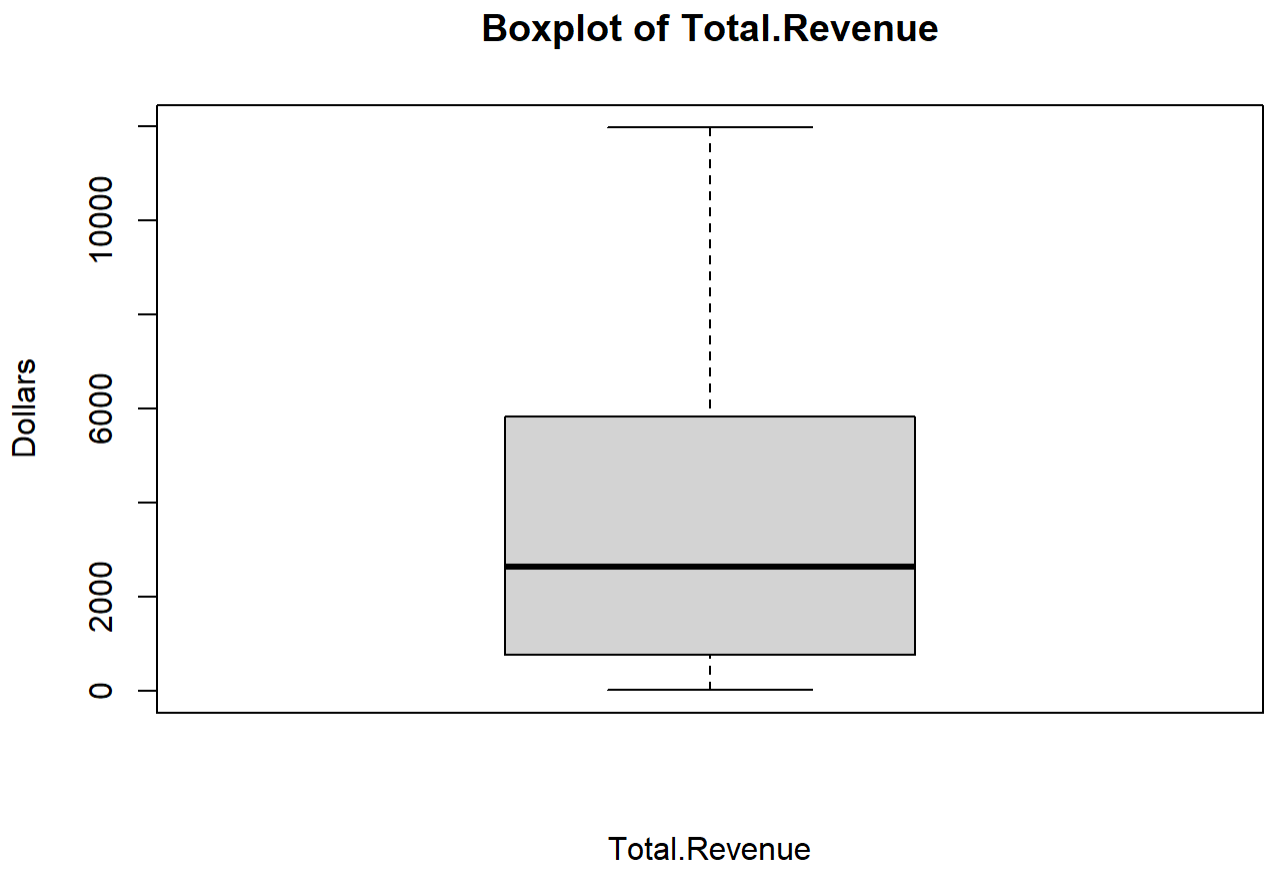
```
##                      Streaming.TV                  Streaming.Movies
##                                 0                                 0
##                    Streaming.Music                    Unlimited.Data
##                                 0                                 0
##                          Contract                 Paperless.Billing
##                                 0                                 0
##                    Payment.Method                    Monthly.Charge
##                                 0                                 0
##                     Total.Charges                     Total.Refunds
##                                 0                                 0
##         Total.Extra.Data.Charges      Total.Long.Distance.Charges
##                                 0                                 0
##                     Total.Revenue                   Customer.Status
##                                 0                                 0
##                    Churn.Category                    Churn.Reason
##                                 0                                 0
```

```
telecom_customer_churn$Avg.Monthly.GB.Download[is.na(telecom_customer_churn$Avg.Monthly.Long.Distance.Charges)]<-mean(telecom_customer_churn$Avg.Monthly.Long.Distance.Charges,na.rm = TRUE)

telecom_customer_churn$Avg.Monthly.Long.Distance.Charges[is.na(telecom_customer_churn$Avg.Monthly.Long.Distance.Charges)]<-mean(telecom_customer_churn$Avg.Monthly.Long.Distance.Charges,na.rm = TRUE)

telecom_customer_churn <- telecom_customer_churn[complete.cases(telecom_customer_churn), ]

telecom_customer_churn$Phone.Service<-as.character(telecom_customer_churn$Phone.Service)

telecom_customer_churn$Multiple.Lines<-as.character(telecom_customer_churn$Multiple.Lines)

telecom_customer_churn$Online.Security<-as.character(telecom_customer_churn$Online.Security)

telecom_customer_churn$Online.Backup<-as.character(telecom_customer_churn$Online.Backup)

telecom_customer_churn$Device.Protection.Plan<-as.character(telecom_customer_churn$Device.Protection.Plan)

telecom_customer_churn$Premium.Tech.Support<-as.character(telecom_customer_churn$Premium.Tech.Support)

telecom_customer_churn$Streaming.TV<-as.character(telecom_customer_churn$Streaming.TV)
```

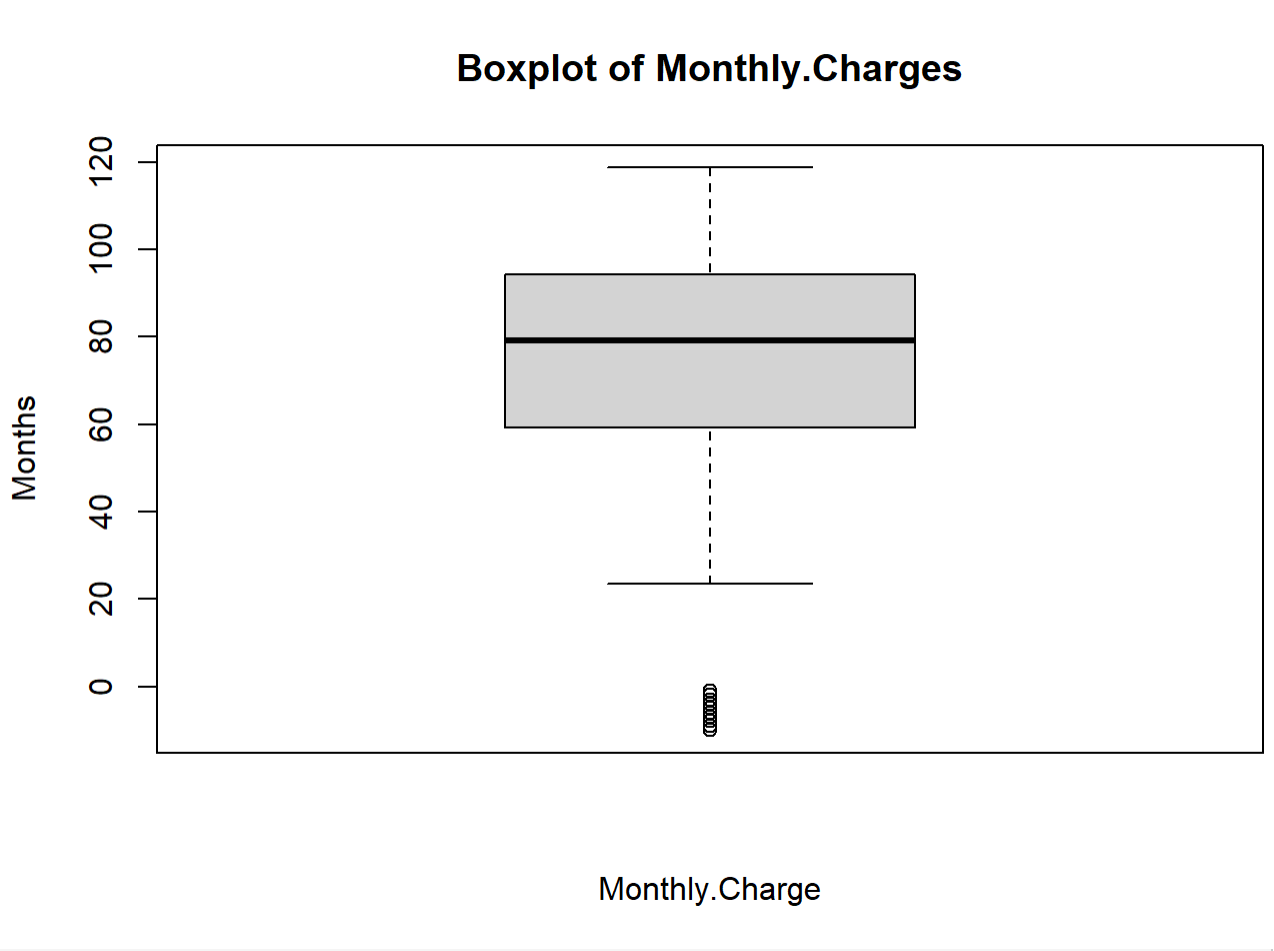
```
telecom_customer_churn$Streaming.Movies<-as.character(telecom_customer_churn$
Streaming.Movies)

telecom_customer_churn$Streaming.Music<-as.character(telecom_customer_churn$S
treaming.Music)

telecom_customer_churn$Unlimited.Data<-as.character(telecom_customer_churn$Un
limited.Data)

telecom_customer_churn$Paperless.Billing<-as.character(telecom_customer_churn
$Paperless.Billing)

telecom_customer_churn$Customer.Status<-as.factor(telecom_customer_churn$Cust
omer.Status)

telecom_customer_churn$Customer.ID<-NULL

telecom_customer_churn$Married<-as.character(telecom_customer_churn$Married)

telecom_customer_churn$Internet.Service<-NULL
```

#EXPLORATORY DATA ANALYSIS.#

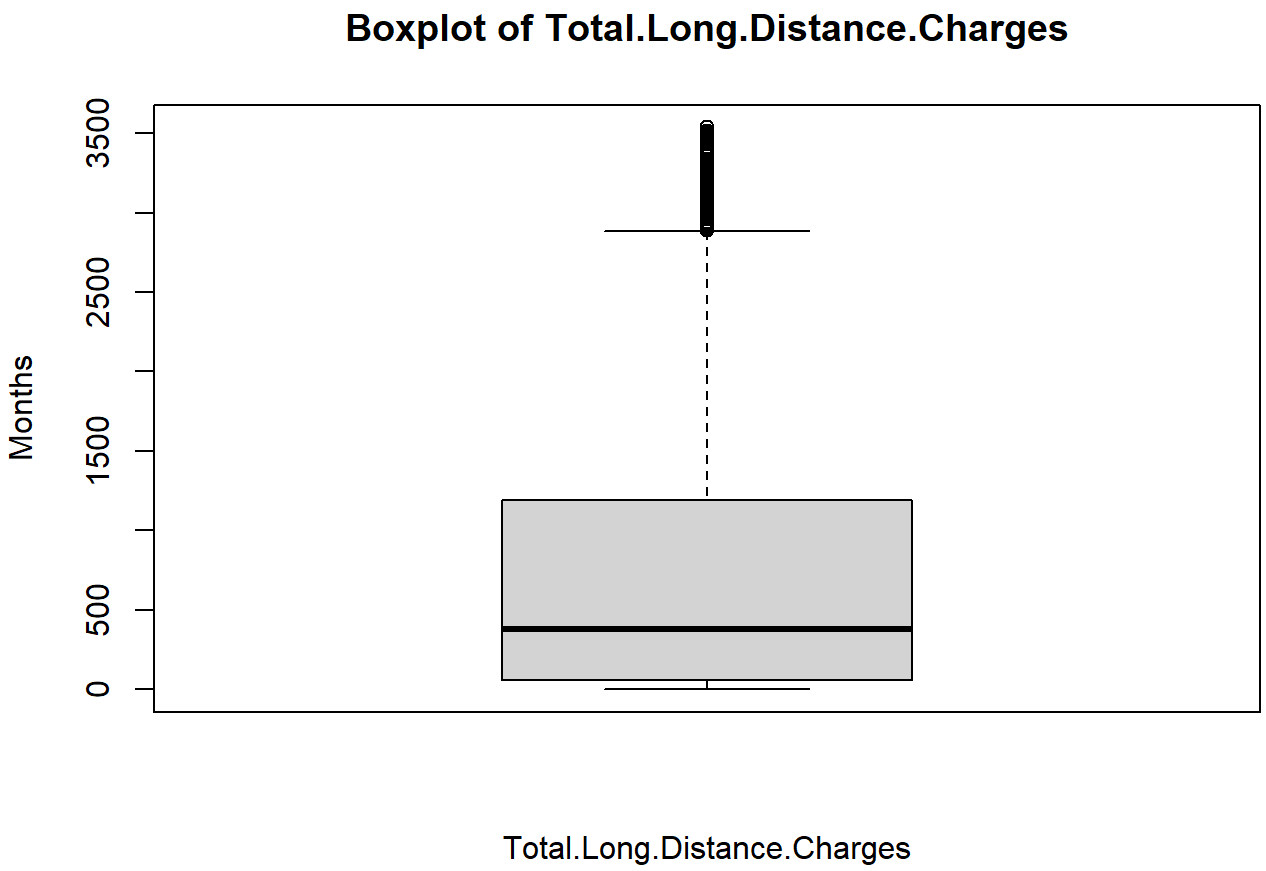
```
boxplot (telecom_customer_churn$Total.Revenue, main = "Boxplot of Total.Reven
ue", xlab = "Total.Revenue", ylab = "Dollars")
```



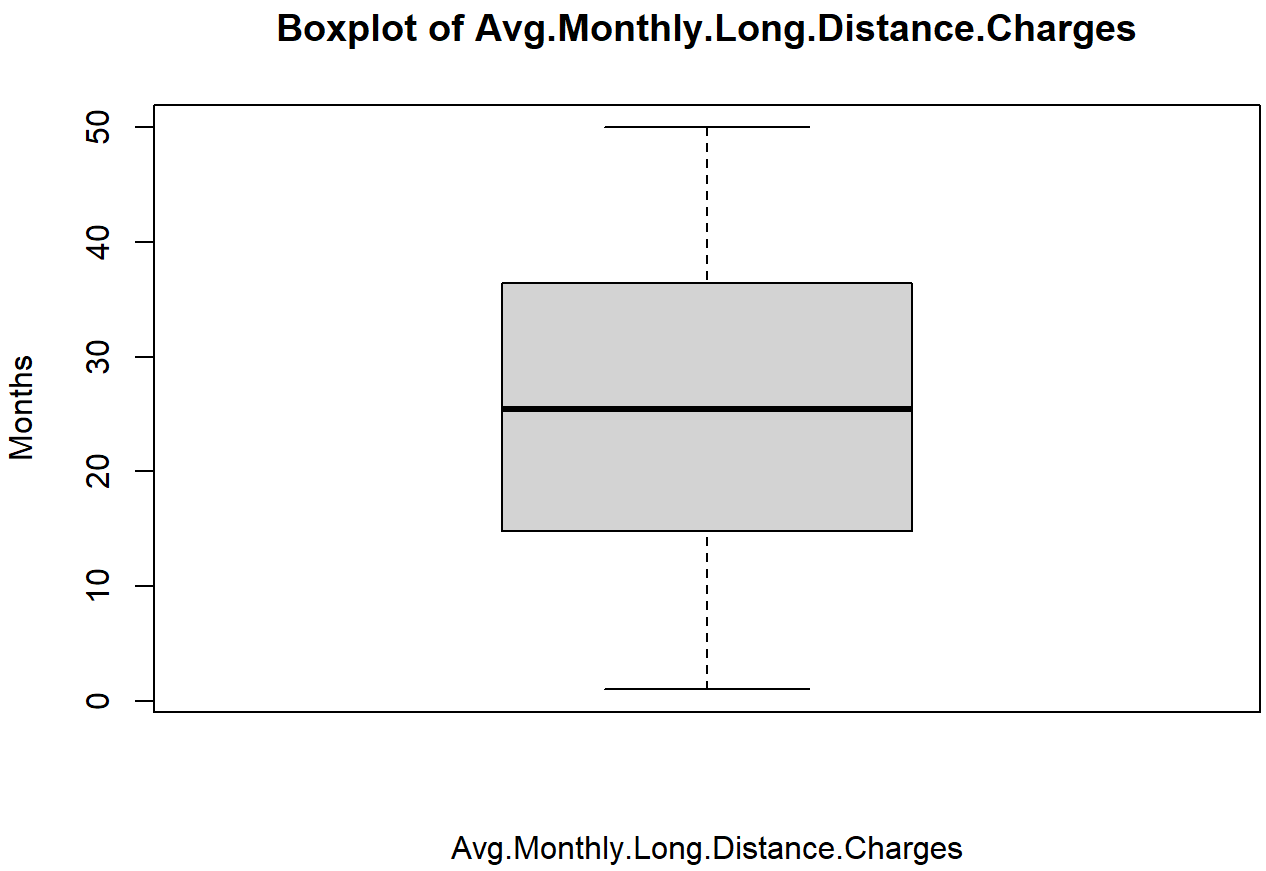
```
boxplot(telecom customer churn$Monthly.Charge, main = "Boxplot of Monthly.Cha
rges", xlab = "Monthly.Charge", ylab = "Months")
```



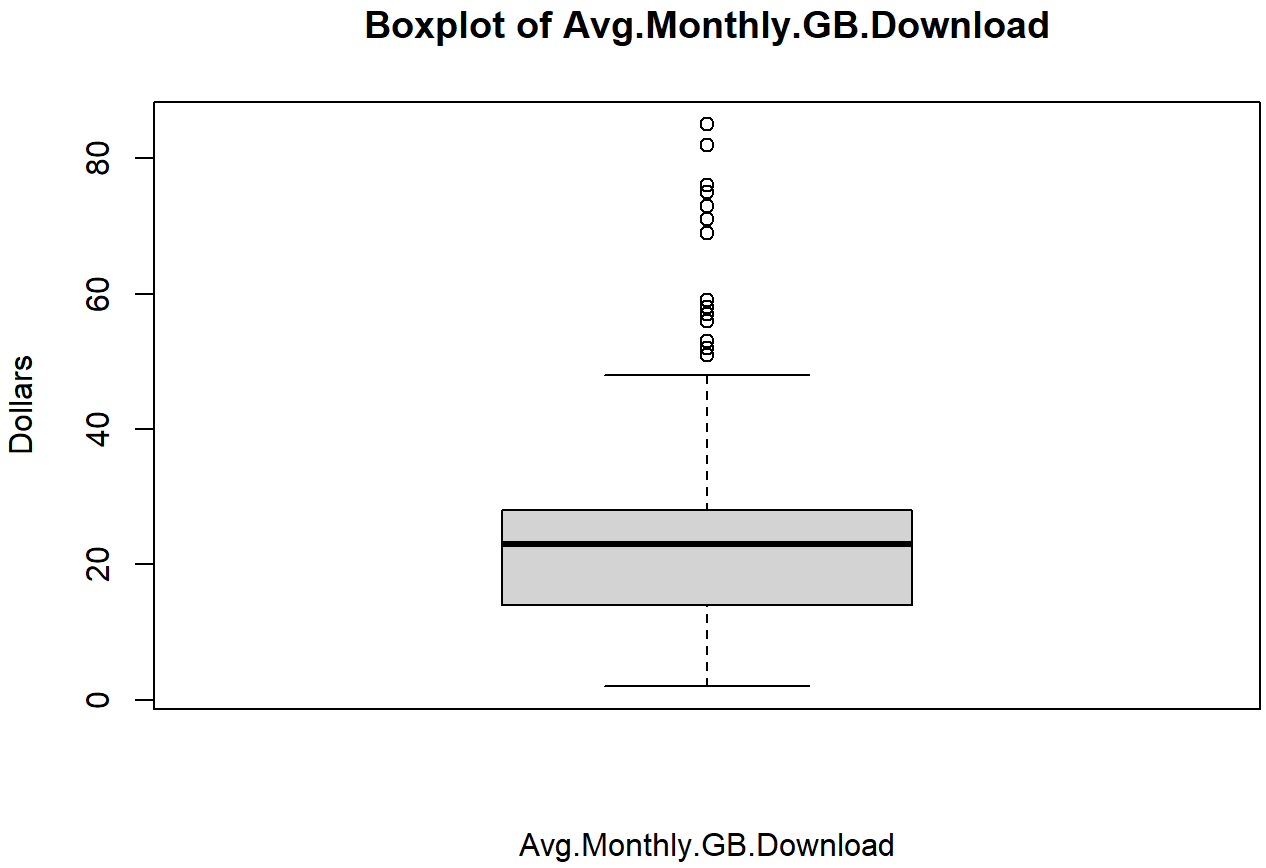

```
boxplot(telecom_customer_churn$Total.Long.Distance.Charges, main = "Boxplot o
f Total.Long.Distance.Charges", xlab = "Total.Long.Distance.Charges", ylab =
"Months")
```




```
boxplot(telecom_customer_churn$Avg.Monthly.Long.Distance.Charges, main = "Box
plot of Avg.Monthly.Long.Distance.Charges", xlab = "Avg.Monthly.Long.Distance
.Charges", ylab = "Months")
```

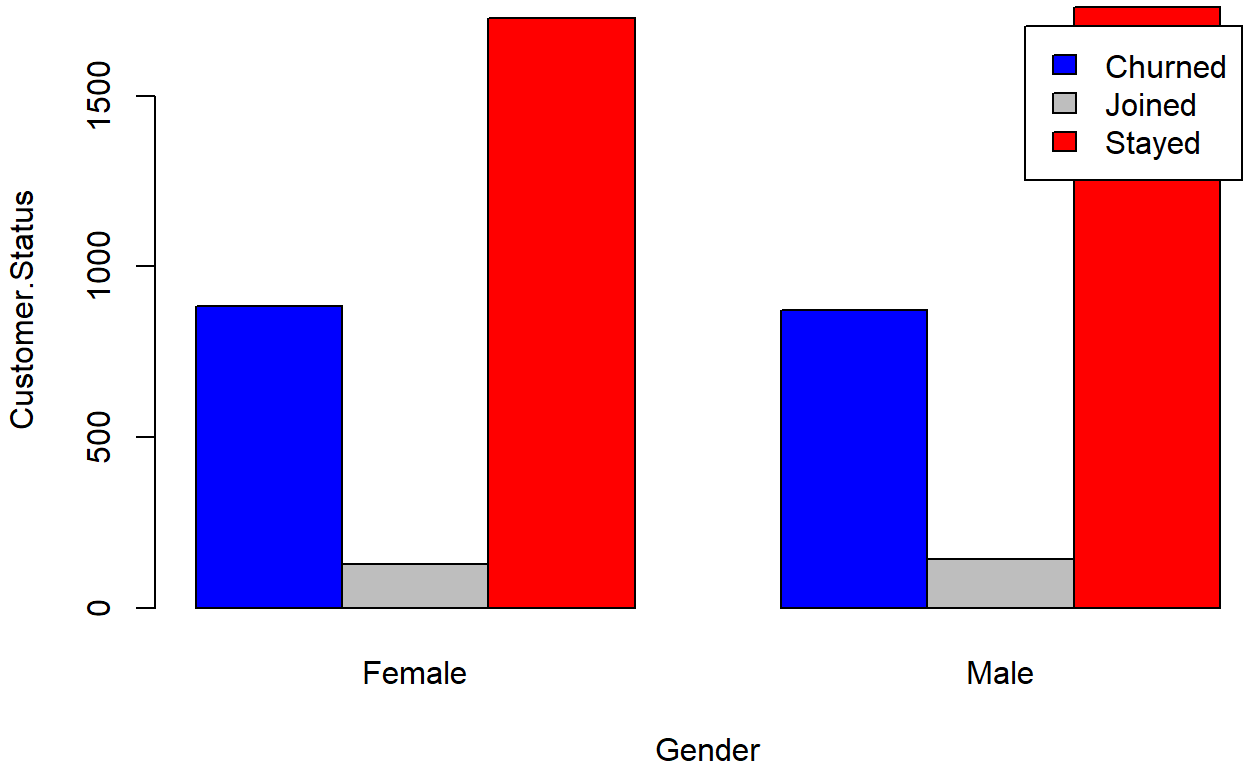



```
boxplot(telecom_customer_churn$Avg.Monthly.GB.Download, main = "Boxplot of Av
g.Monthly.GB.Download", xlab = "Avg.Monthly.GB.Download", ylab = "Dollars")
```

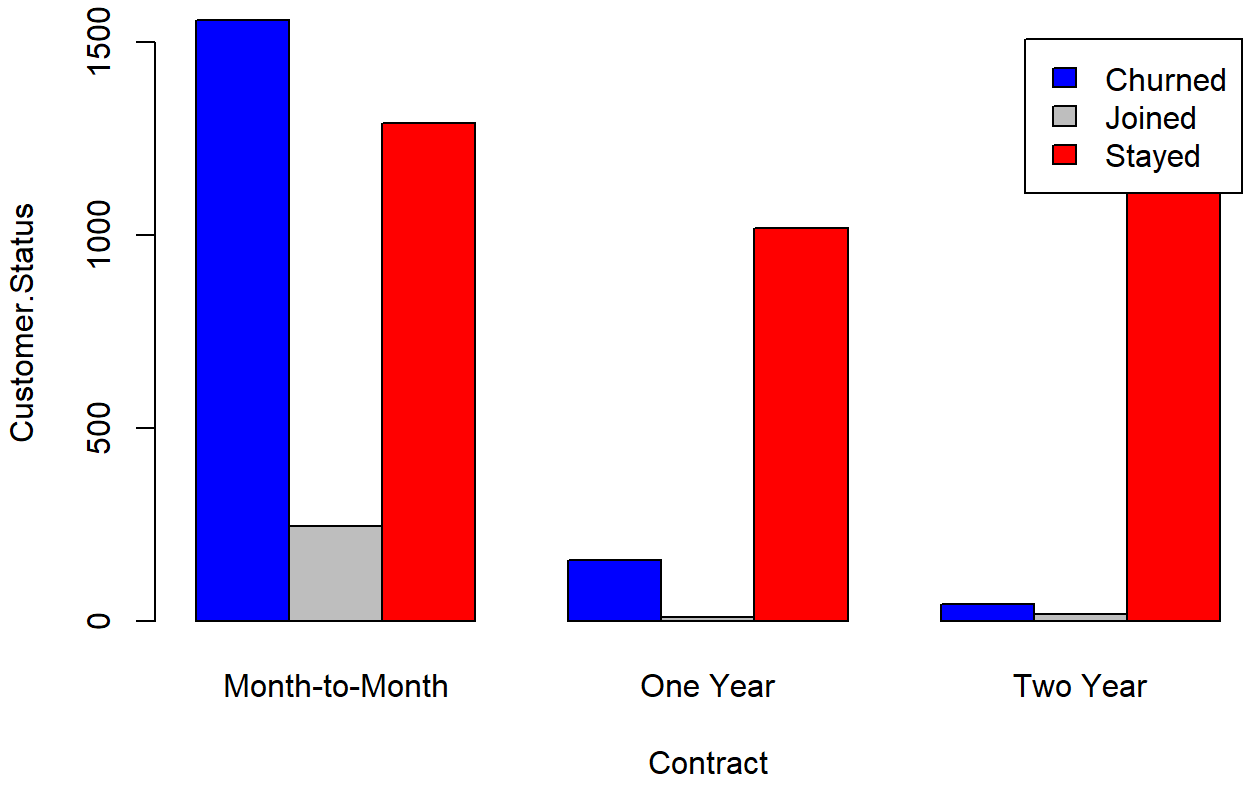


#Create Barplot to check for relationship between categorical variable aganist customer churn.#

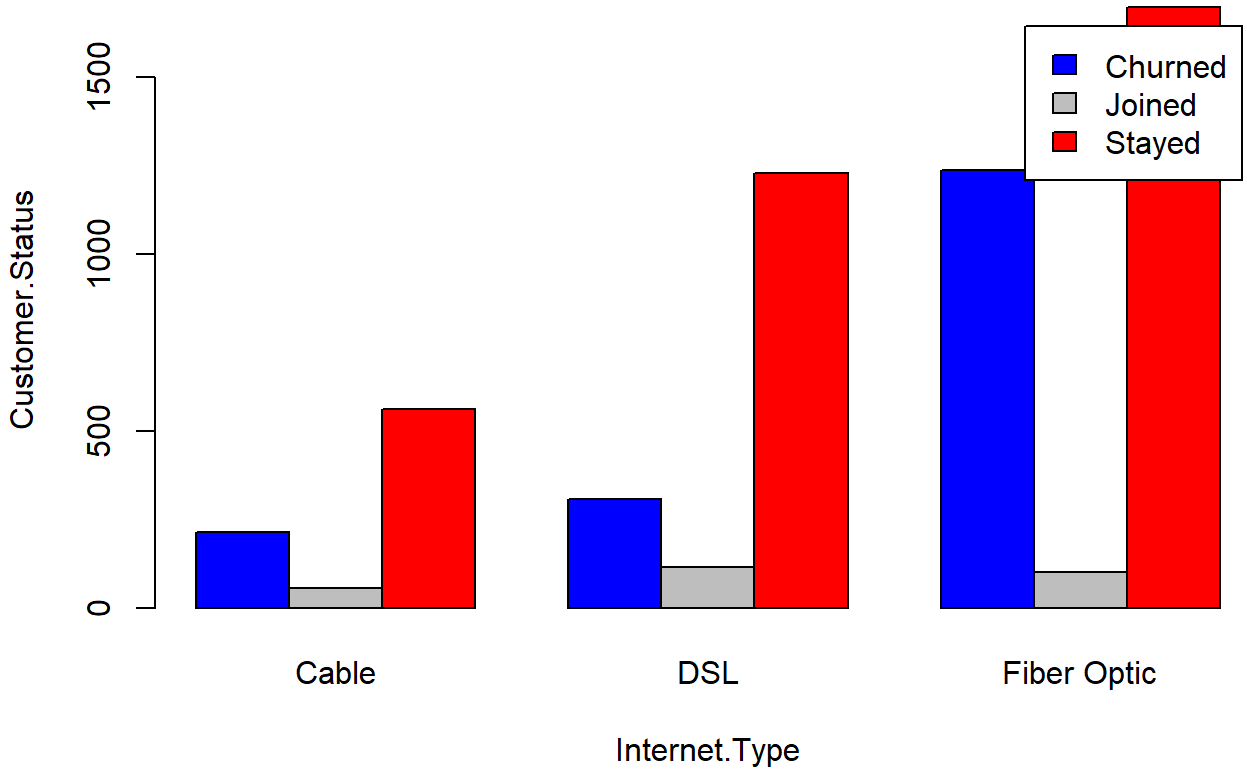
```
Customer.Status <- telecom_customer_churn$Customer.Status

Gender <- telecom_customer_churn$Gender

barplot(table(Customer.Status, Gender), beside = TRUE, col = c("blue","grey",
"red"), legend = TRUE, xlab = "Gender", ylab = "Customer.Status")
```



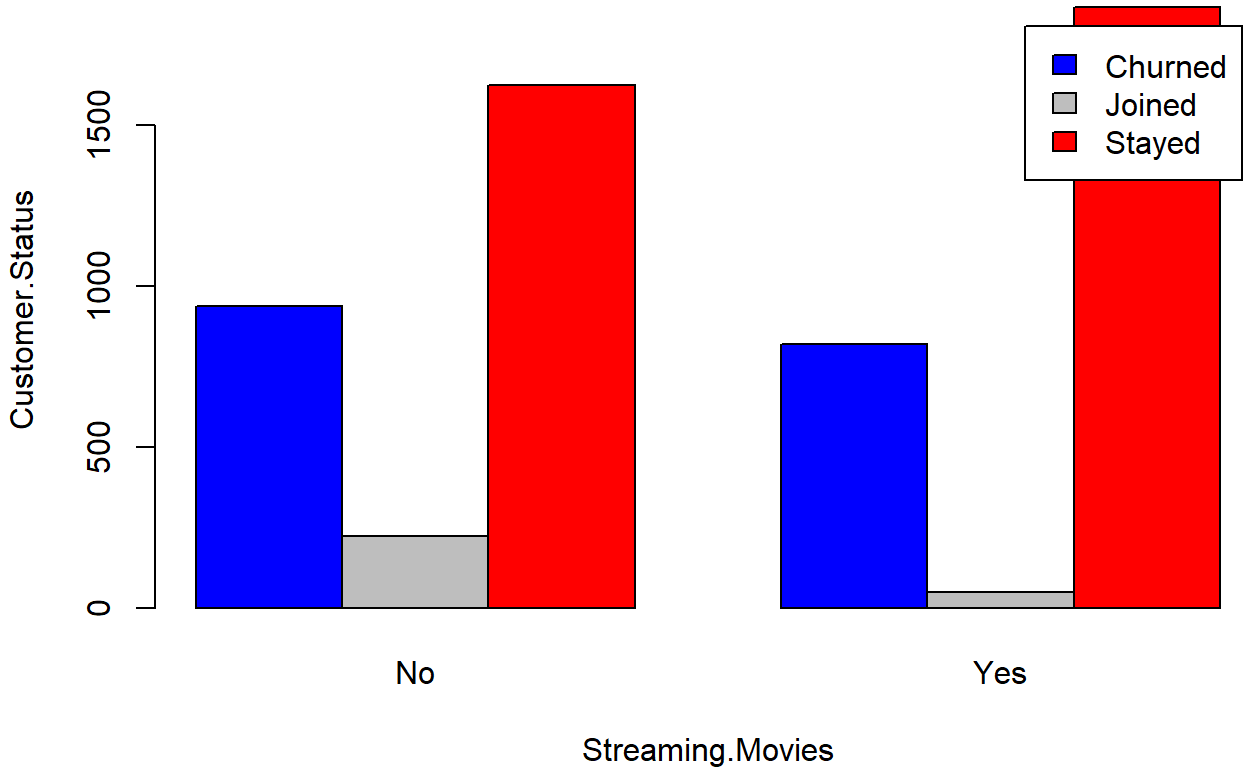

```
Contract <- telecom_customer_churn$Contract

barplot(table(Customer.Status,Contract), beside = TRUE, col = c("blue","grey"
, "red"), legend = TRUE, xlab = "Contract", ylab = "Customer.Status")
```



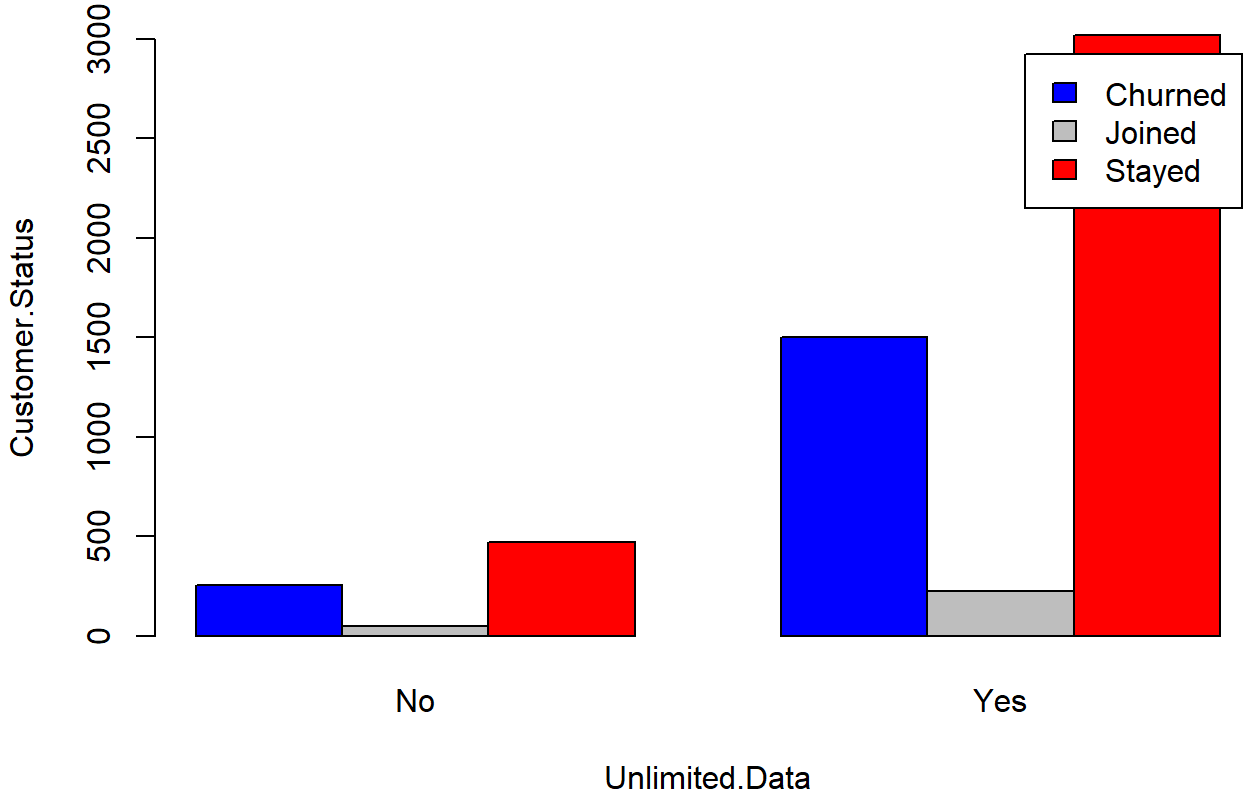

```
Internet.Type  <- telecom_customer_churn$Internet.Type

barplot(table(Customer.Status,Internet.Type), beside = TRUE, col = c("blue","
grey", "red"), legend = TRUE, xlab = "Internet.Type", ylab = "Customer.Status
")
```



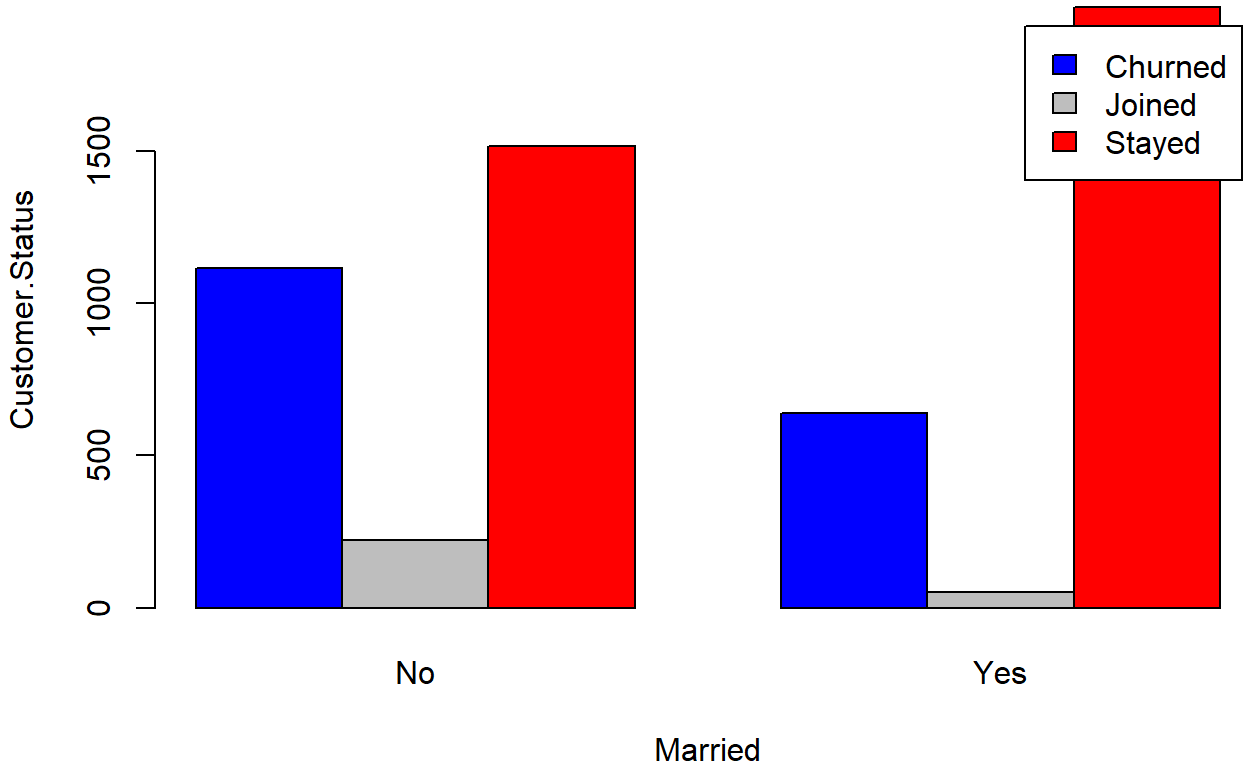

```
Streaming.Movies<-telecom_customer_churn$Streaming.Movies

barplot(table(Customer.Status, Streaming.Movies), beside=TRUE, col=c("blue","
grey","red"),legend=TRUE, xlab="Streaming.Movies", ylab="Customer.Status")
```



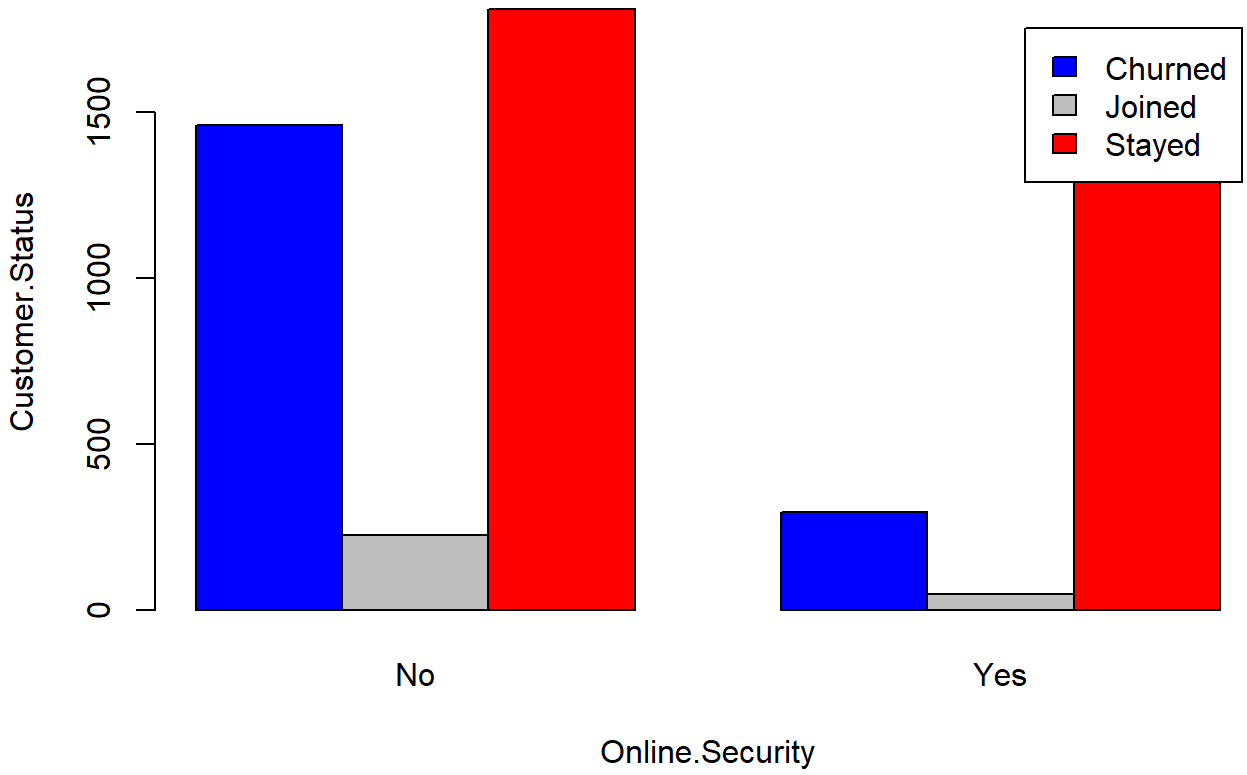

```
Unlimited.Data<-telecom_customer_churn$Unlimited.Data

barplot(table(Customer.Status,Unlimited.Data),beside=TRUE,col=c("blue","grey"
,"red"),legend=TRUE, xlab="Unlimited.Data",ylab="Customer.Status")
```



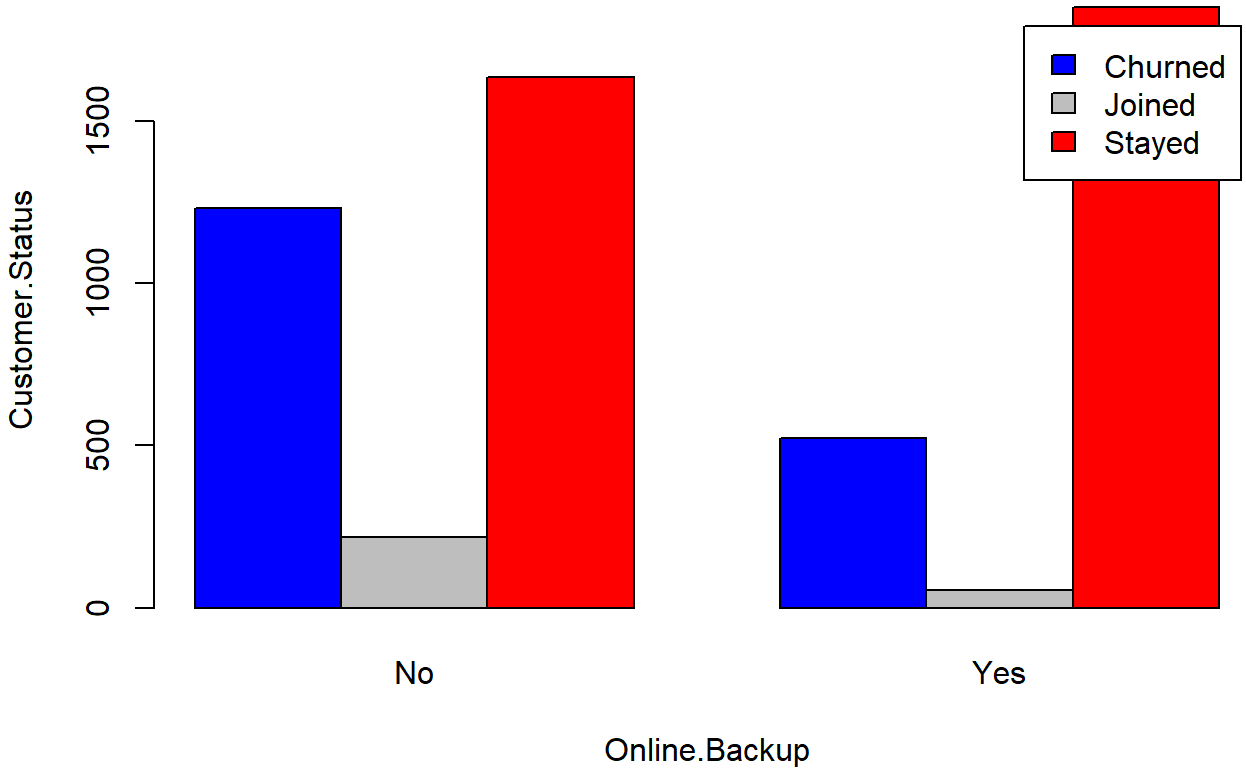

```
Married<-telecom_customer_churn$Married

barplot(table(Customer.Status,Married),beside=TRUE,col=c("blue","grey","red")
,legend=TRUE,xlab="Married", ylab="Customer.Status")
```



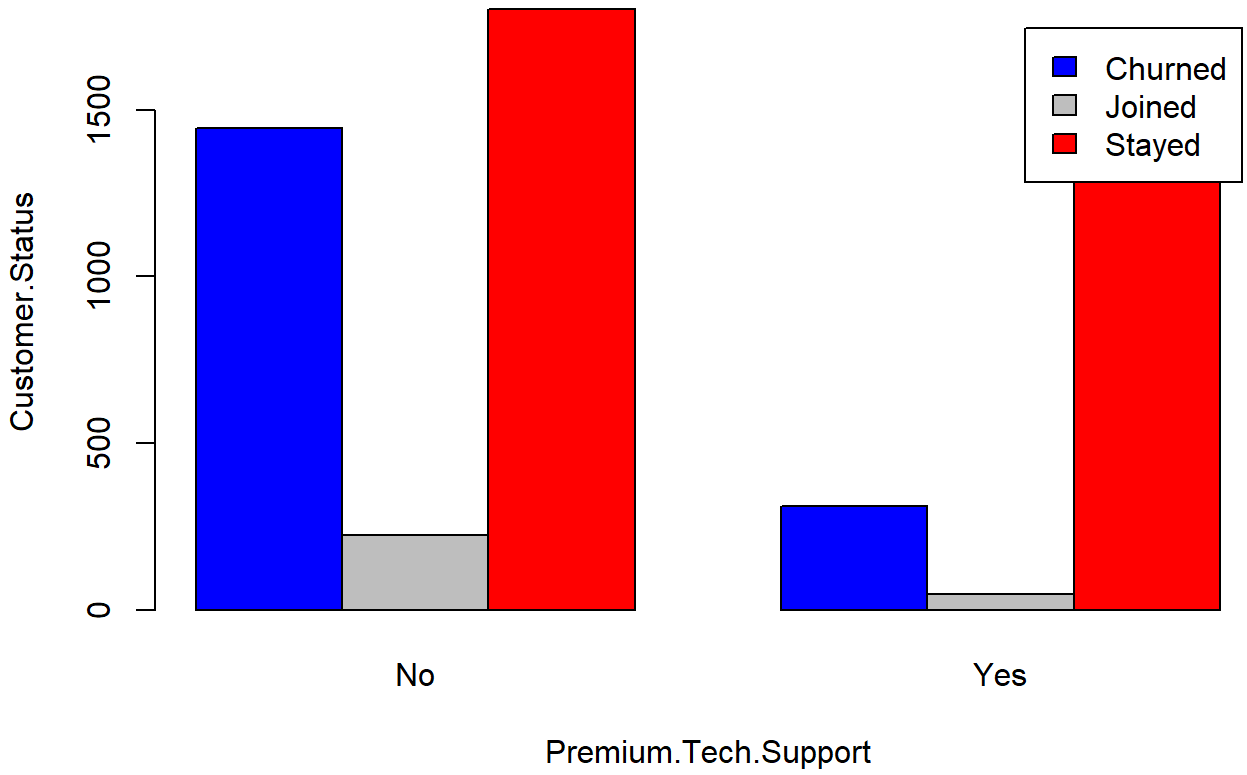

```
Online.Security<-telecom_customer_churn$Online.Security

barplot(table(Customer.Status,Online.Security),beside=TRUE,col=c("blue","grey
","red"),legend=TRUE,xlab="Online.Security",ylab="Customer.Status")
```



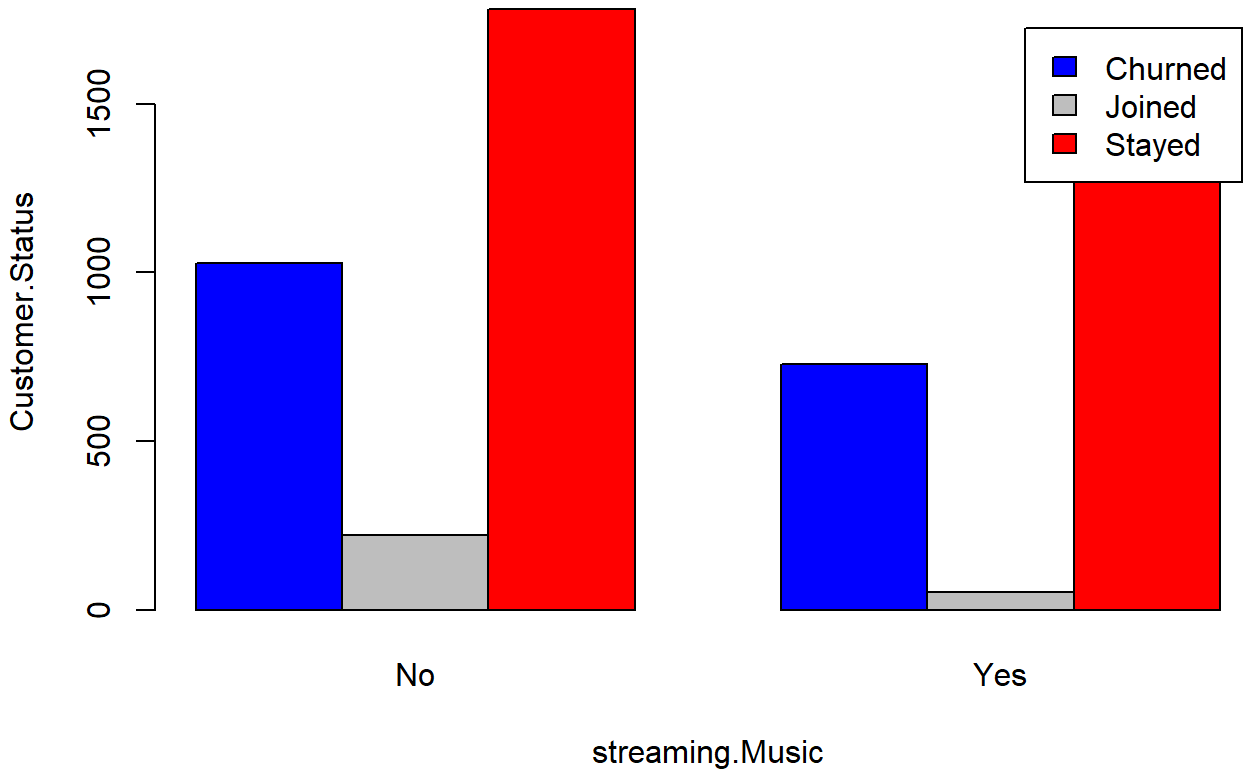

```
Online.Backup<-telecom_customer_churn$Online.Backup

barplot(table(Customer.Status,Online.Backup),beside=TRUE,col=c("blue","grey",
"red"),legend=TRUE,xlab="Online.Backup",ylab="Customer.Status")
```



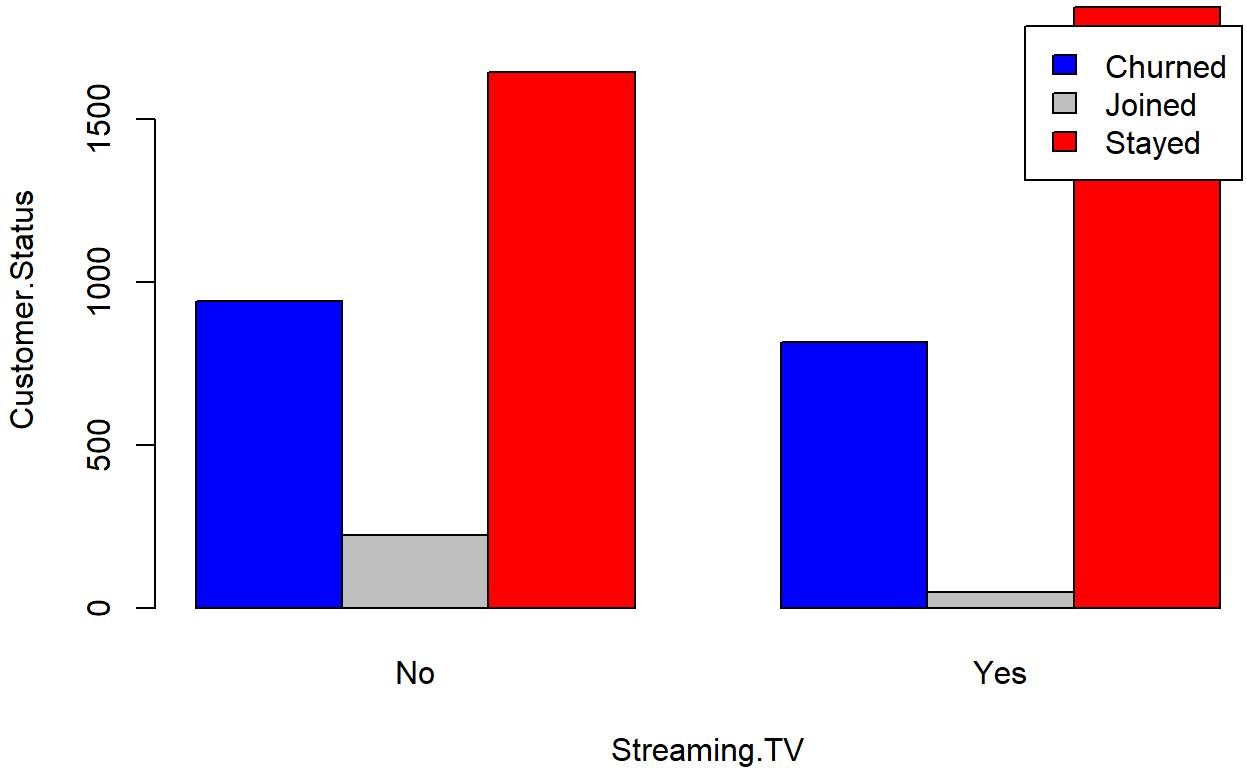

```
Premium.Tech.Support <-telecom_customer_churn$Premium.Tech.Support

barplot(table(Customer.Status,Premium.Tech.Support),beside=TRUE,col=c("blue",
"grey","red"),legend=TRUE,xlab="Premium.Tech.Support",ylab="Customer.Status")
```




```
Streaming.Music<-telecom_customer_churn$Streaming.Music

barplot(table(Customer.Status,Streaming.Music),beside=TRUE,col=c("blue","grey
","red"),legend=TRUE,xlab="streaming.Music",ylab="Customer.Status")
```

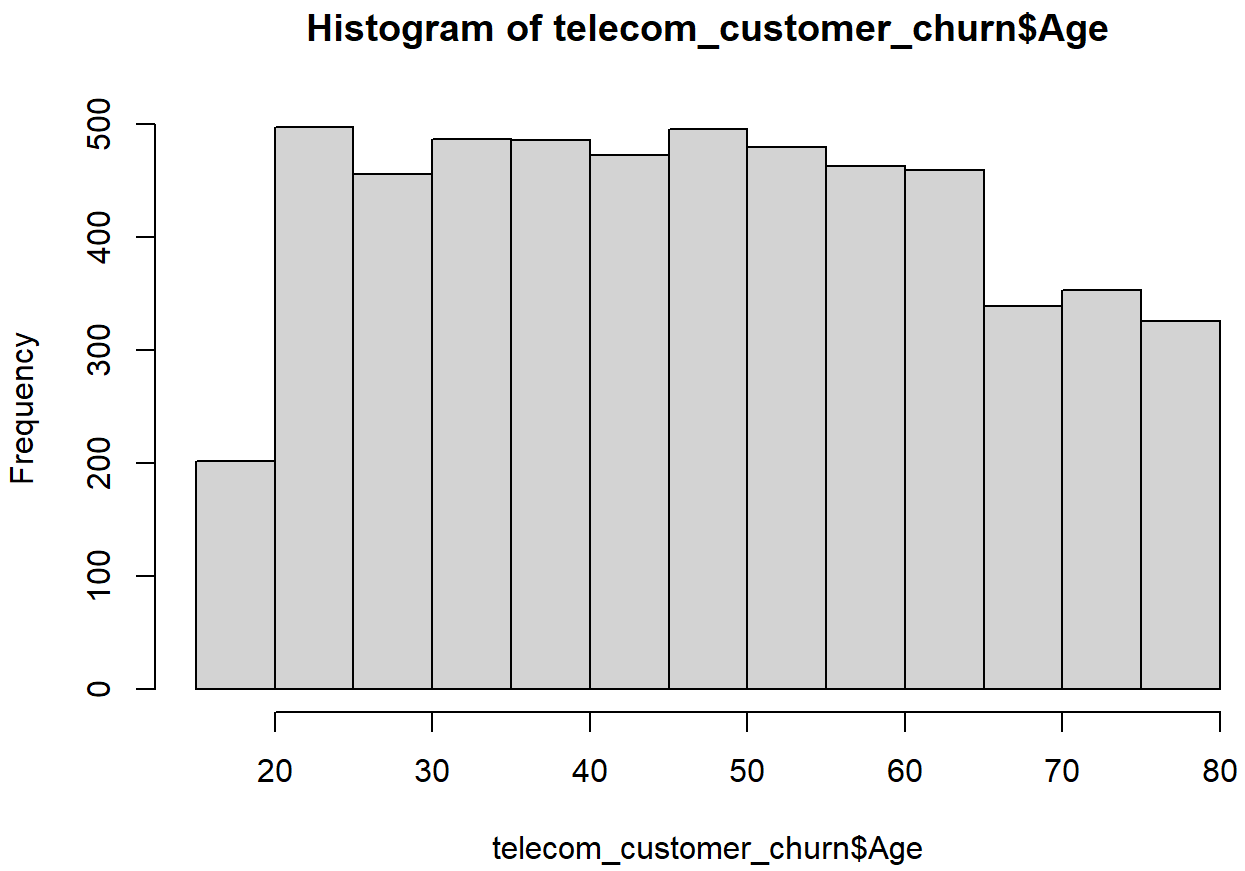



```
Streaming.TV<-telecom_customer_churn$Streaming.TV

barplot(table(Customer.Status,Streaming.TV),beside=TRUE,col=c("blue","grey","
red"),legend=TRUE,xlab="Streaming.TV",ylab="Customer.Status")
```

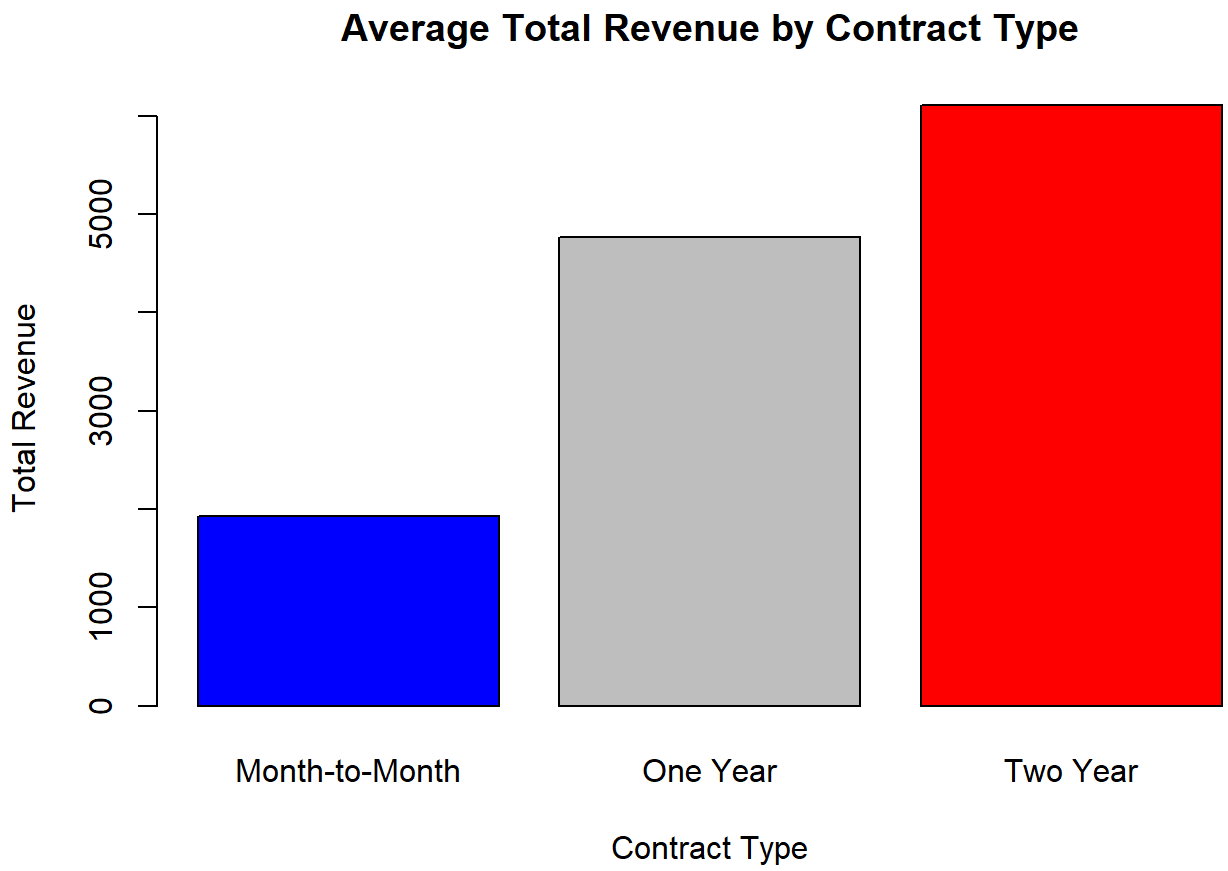


#numerical variables aganist churn.#

```
hist(telecom_customer_churn$Age)
```




```
mean_revenue <- aggregate(Total.Revenue ~ Contract, data = telecom_customer_c
hurn, mean)

barplot(mean_revenue$Total.Revenue, names.arg = mean_revenue$Contract,

        main = "Average Total Revenue by Contract Type",

        xlab = "Contract Type", ylab = "Total Revenue",

        col = c("blue", "grey", "red"))
```

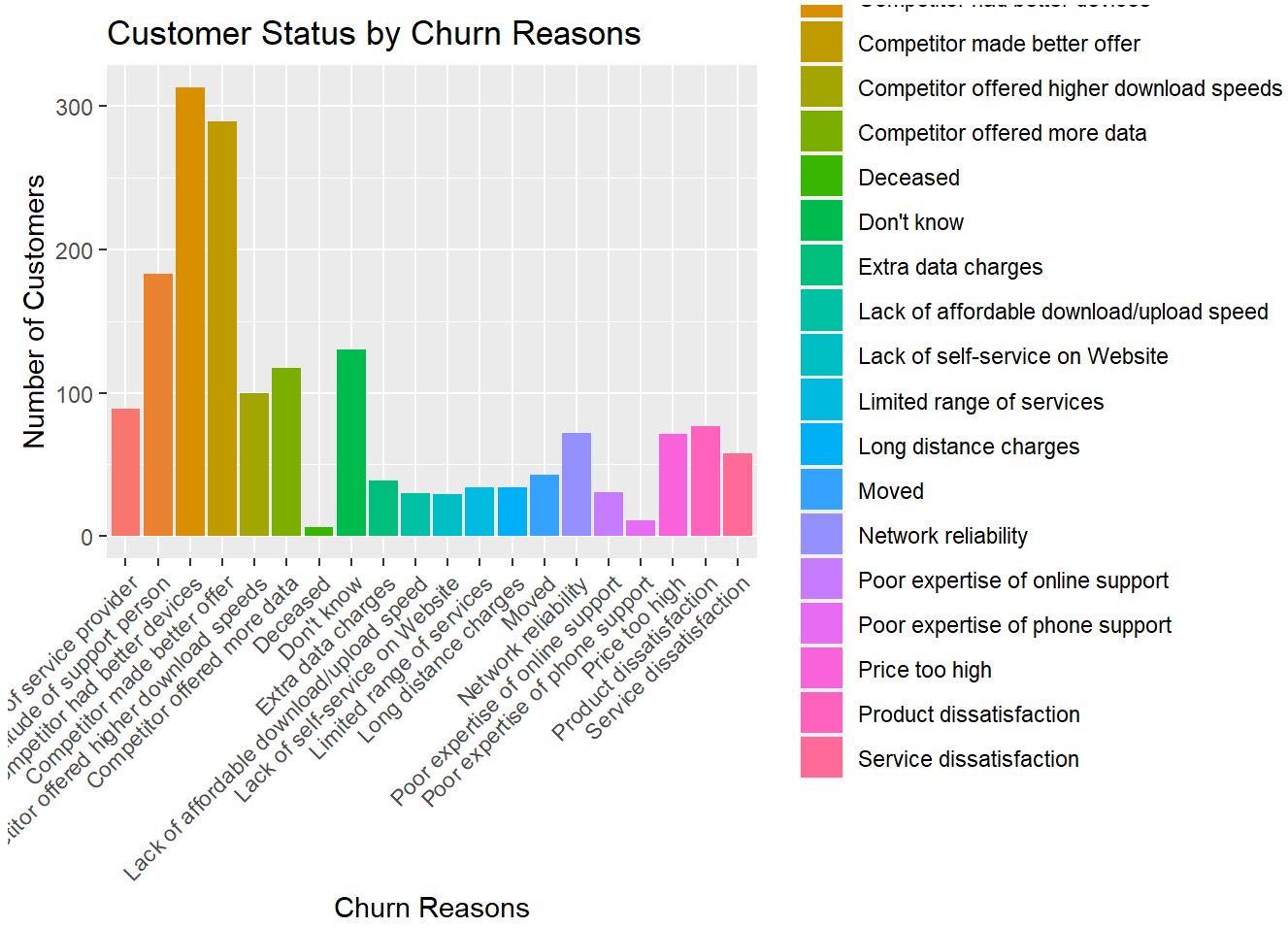


#Filtering the data for more insights.#

```
library(dplyr)
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
churn_data <- filter(telecom_customer_churn, Customer.Status == "Churned")
churn_summary1 <- group_by(churn_data, Churn.Reason) %>%
  summarize(count = n())
churn_summary1
```

```
## # A tibble: 20 × 2
##    Churn.Reason                                count
##    <chr>                                       <int>
##  1 Attitude of service provider                   89
##  2 Attitude of support person                    183
##  3 Competitor had better devices                 313
##  4 Competitor made better offer                  289
##  5 Competitor offered higher download speeds     100
##  6 Competitor offered more data                  117
##  7 Deceased                                        6
##  8 Don't know                                    130
##  9 Extra data charges                             39
## 10 Lack of affordable download/upload speed       30
## 11 Lack of self-service on Website                29
## 12 Limited range of services                      34
## 13 Long distance charges                          34
## 14 Moved                                          43
## 15 Network reliability                            72
## 16 Poor expertise of online support               31
## 17 Poor expertise of phone support                11
## 18 Price too high                                 71
## 19 Product dissatisfaction                        77
## 20 Service dissatisfaction                        58
```

```
library(ggplot2)

ggplot(churn_summary1, aes(x = Churn.Reason, y = count, fill = Churn.Reason))
+
  geom_bar(stat = "identity") +
  labs(title = "Customer Status by Churn Reasons",
       x = "Churn Reasons", y = "Number of Customers") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



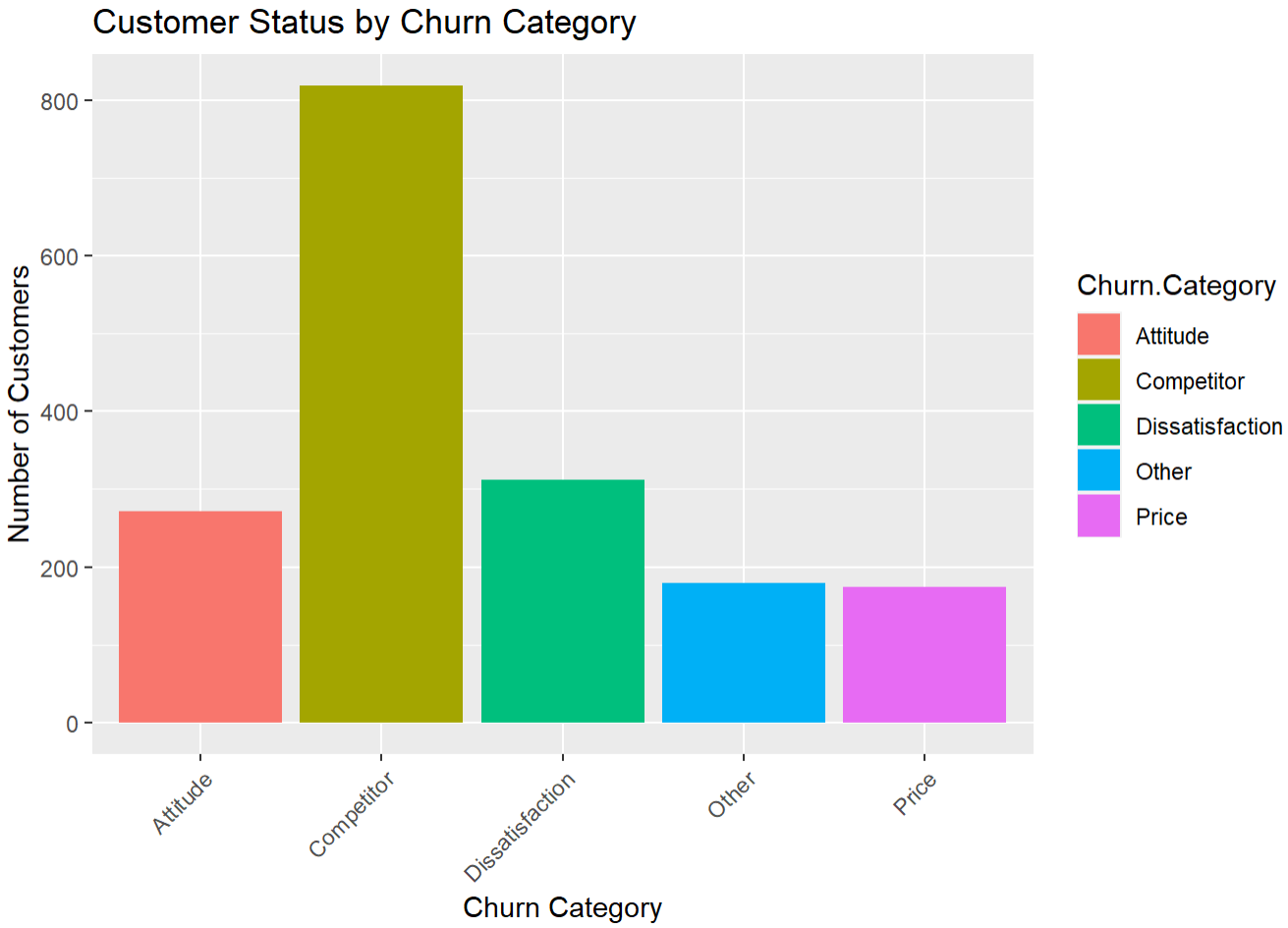
#We can also check which categories have the most churned customers.#

```
churn_data <- filter(telecom_customer_churn, Customer.Status == "Churned")
churn_summary2  <- group_by(churn_data, Churn.Category) %>%
  summarize(count = n())
churn_summary2
```

```
## # A tibble: 5 × 2
##   Churn.Category  count
##   <chr>           <int>
## 1 Attitude          272
## 2 Competitor        819
## 3 Dissatisfaction   312
## 4 Other             179
## 5 Price             174
```

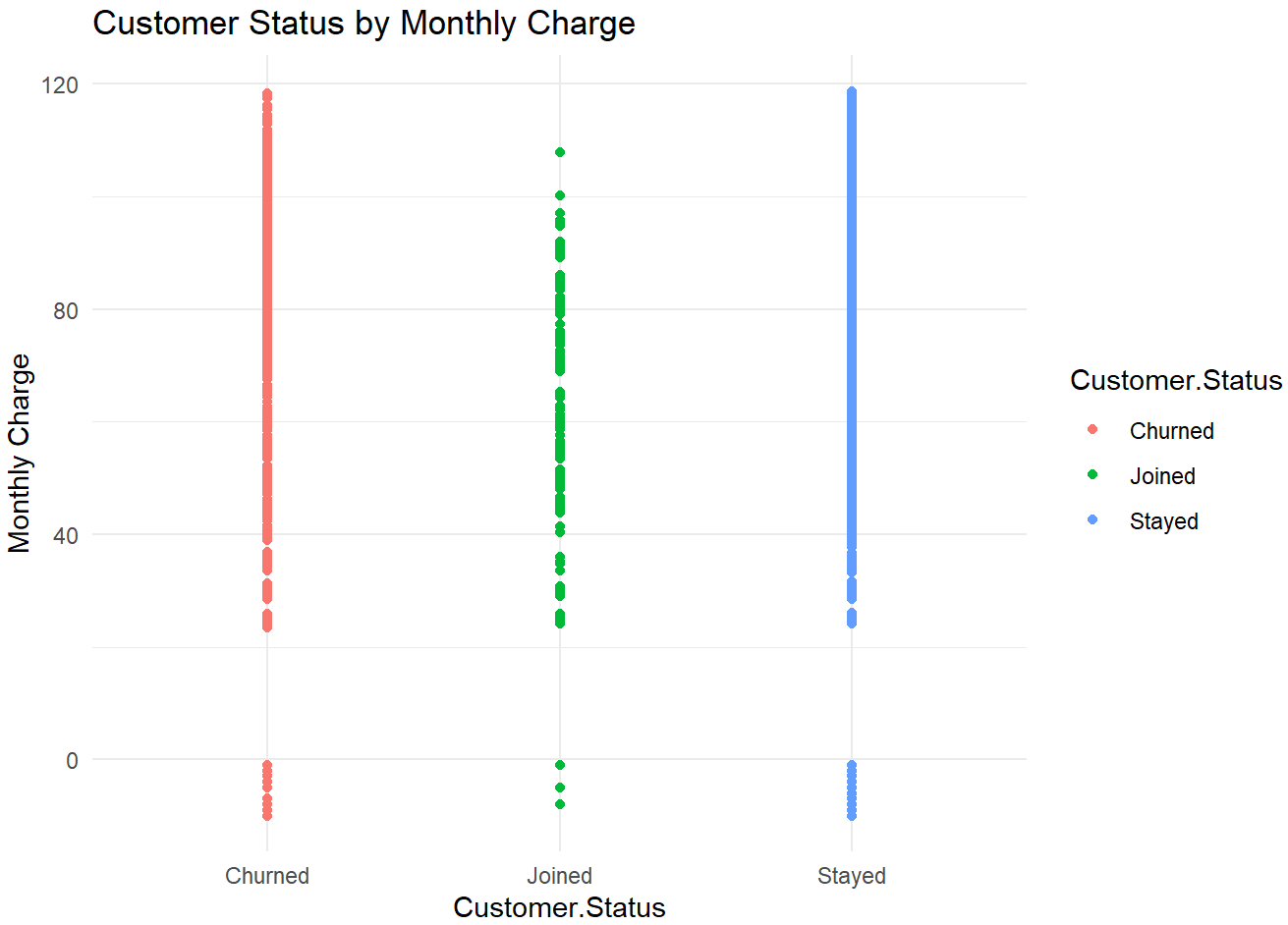
```
library(ggplot2)
```

```
ggplot(churn_summary2, aes(x = Churn.Category, y = count, fill = Churn.Catego
ry)) +
  geom_bar(stat = "identity") +
  labs(title = "Customer Status by Churn Category",
       x = "Churn Category", y = "Number of Customers") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



#check if there is relation between contract and charges.#

```
ggplot(telecom_customer_churn, aes(x = Customer.Status, y = Monthly.Charge))
+geom_point(aes(color = Customer.Status)) +labs(x = "Customer.Status", y = "M
onthly Charge") +
  ggtitle("Customer Status by Monthly Charge") +theme_minimal()
```



#DATA MODELLING USING MACHINE LEARNING.#

#Classification#

#Building classification model.#

```
set.seed(123)
library(caTools)
Sample<-sample.split(telecom_customer_churn$Customer.Status, SplitRatio = 0.8
)
trainSet<-subset(telecom_customer_churn, Sample==TRUE)
testSet<-subset(telecom_customer_churn, Sample==FALSE)
trainSet <- subset(trainSet, select = -City)
testSet <- subset(testSet, select = -City)
```

#Support Vector Machine#

```
library(e1071)
library(caret)
```

```
## Loading required package: lattice
```

```
svmod <- svm(formula = Customer.Status ~ ., data = trainSet, type = "C-classification",
    kernel = "linear")
svmtrain<-predict(svmod,trainSet,type="class")
table(svmtrain,trainSet$Customer.Status)
```

```
##
## svmtrain  Churned Joined Stayed
##   Churned    1405      0      0
##   Joined        0    214     10
##   Stayed        0      4   2781
```

```
svmtest<-predict(svmod,testSet,type="class")
table(svmtest,testSet$Customer.Status)
```

```
##
## svmtest   Churned Joined Stayed
##   Churned     351      0      0
##   Joined        0     51      1
##   Stayed        0      3    697
```

```
confusion_matrix <- confusionMatrix(svmtest, testSet$Customer.Status)
print(confusion_matrix)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Churned Joined Stayed
##    Churned     351      0      0
##    Joined        0     51      1
##    Stayed        0      3    697
##
## Overall Statistics
##
##                Accuracy : 0.9964
##                  95% CI : (0.9907, 0.999)
##     No Information Rate : 0.6328
##     P-Value [Acc > NIR] : < 2.2e-16
```

```
##
##                  Kappa : 0.9927
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Churned Class: Joined Class: Stayed
## Sensitivity                  1.0000       0.94444        0.9986
## Specificity                  1.0000       0.99905        0.9926
## Pos Pred Value               1.0000       0.98077        0.9957
## Neg Pred Value               1.0000       0.99715        0.9975
## Prevalence                   0.3182       0.04896        0.6328
## Detection Rate               0.3182       0.04624        0.6319
## Detection Prevalence         0.3182       0.04714        0.6346
## Balanced Accuracy            1.0000       0.97175        0.9956
```

#Naive Bayes Classifier#

```
library(e1071)
nbmod<-naiveBayes(Customer.Status~.,data=trainSet)
nbtrain<-predict(nbmod,trainSet,type = "class")
table(nbtrain,trainSet$Customer.Status)
```

```
##
## nbtrain   Churned Joined Stayed
##   Churned    1021      0      0
##   Joined      313    214    101
##   Stayed       71      4   2690
```

```
nbtest<-predict(nbmod,testSet,type = "class")
confusion_matrix <- confusionMatrix(nbtest, testSet$Customer.Status)
print(confusion_matrix)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Churned Joined Stayed
```

```
##    Churned     262       0       0
##    Joined       66      53      24
##    Stayed       23       1     674
##
## Overall Statistics
##
##                Accuracy : 0.8966
##                  95% CI : (0.8772, 0.914)
##     No Information Rate : 0.6328
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.8003
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: Churned Class: Joined Class: Stayed
## Sensitivity                  0.7464       0.98148        0.9656
## Specificity                  1.0000       0.91420        0.9407
## Pos Pred Value               1.0000       0.37063        0.9656
## Neg Pred Value               0.8942       0.99896        0.9407
## Prevalence                   0.3182       0.04896        0.6328
## Detection Rate               0.2375       0.04805        0.6111
## Detection Prevalence         0.2375       0.12965        0.6328
## Balanced Accuracy            0.8732       0.94784        0.9532
```

#Random Forest#

```
library(randomForest)
## randomForest 4.7-1.1
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
## The following object is masked from 'package:ggplot2':
```

```
##
##     margin
## The following object is masked from 'package:dplyr':
##
##     combine
rfmod<-randomForest(Customer.Status~.,data=trainSet,importance=T)
predtrain<-predict(rfmod,trainSet,type = "class")
table(predtrain,trainSet$ Customer.Status)
##
## predtrain Churned Joined Stayed
##   Churned    1405      0      0
##   Joined        0    218      0
##   Stayed        0      0   2791
predtest<-predict(rfmod,testSet,type = "class")
table(predtest,testSet$Customer.Status)
##
## predtest  Churned Joined Stayed
##   Churned     351      0      0
##   Joined        0     54      0
##   Stayed        0      0    698
confusion_matrix <- confusionMatrix(predtest, testSet$Customer.Status)
print(confusion_matrix)
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Churned Joined Stayed
##    Churned     351      0      0
##    Joined        0     54      0
##    Stayed        0      0    698
##
## Overall Statistics
##
##                Accuracy : 1
##                  95% CI : (0.9967, 1)
```

```
##     No Information Rate : 0.6328
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Churned Class: Joined Class: Stayed
## Sensitivity                  1.0000       1.00000        1.0000
## Specificity                  1.0000       1.00000        1.0000
## Pos Pred Value               1.0000       1.00000        1.0000
## Neg Pred Value               1.0000       1.00000        1.0000
## Prevalence                   0.3182       0.04896        0.6328
## Detection Rate               0.3182       0.04896        0.6328
## Detection Prevalence         0.3182       0.04896        0.6328
## Balanced Accuracy            1.0000       1.00000        1.0000
```

#Logistic Regression.#

```
class_lr <- glm(Customer.Status ~ ., data = trainSet, family = binomial(link
= "logit"))
```

```
## Warning: glm.fit: algorithm did not converge
```

```
predicted_prob <- predict(class_lr, newdata = testSet, type = "response")
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful case
s
```

```
predicted_labels <- ifelse(predicted_prob > 0.5, 1, 0)
confusion_matrix <- table(predicted_labels, testSet$Customer.Status)
accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
print(confusion_matrix)
```

```
##
## predicted_labels Churned Joined Stayed
##                0     351      0      0
```

```
##                1       0     54    698
cat("Accuracy:", accuracy, "\n")
## Accuracy: 0.3671804
sensitivity <- confusion_matrix[2, 2] / sum(confusion_matrix[2, ])
precision <- confusion_matrix[2, 2] / sum(confusion_matrix[, 2])
specificity <- confusion_matrix[1, 1] / sum(confusion_matrix[1, ])
cat("Specificity (True Negative Rate):", specificity, "\n")
## Specificity (True Negative Rate): 1
cat("Sensitivity (Recall):", sensitivity, "\n")
## Sensitivity (Recall): 0.07180851
cat("Precision (Positive Predictive Value):", precision, "\n")
## Precision (Positive Predictive Value): 1
```