# TELCO CHURN


**Team**

**Saurabh Sharma**

**Swati Chellani**

# Contents

# Introduction

Churn prediction is one of the most common applications of classification in the business settings. Most of the financial institutions are concerned with customer retention studies to prevent losing their market share and maximize their gained profit from existing customers. The primary objective of customer retention is to maximize the potential profit which can come from existing customers.

# Problem Statement

Most telecommunication companies today suffer from voluntary customer churn. The abandonment rate has great impact on the value of customers' lifetime, because it affects the duration of the service and the company's benefit. These companies spend a huge amount of money to acquire new customers and when these customers abandon them, the companies not only lose customer benefit but also the resources spent to acquire them. Also, one of the benefits of retaining old customer base is that the company already has built trust with the customer unlike in the case of a newly acquired customer. This model will predict whether or not the company will be able to retain a certain future customer.

# Objective

The objective is to create a model to predict if a customer will or not abandon the telecom company and identify factors that most affect the probability of the same.
This model can further be used by the company or any other related company to reduce the percent of customer churn and analyse the behaviour of the potentially abandoning customers.

# SEMMA Approach

We have used SEMMA Approach used in Business problems in our project and drawn results on the basis of it.

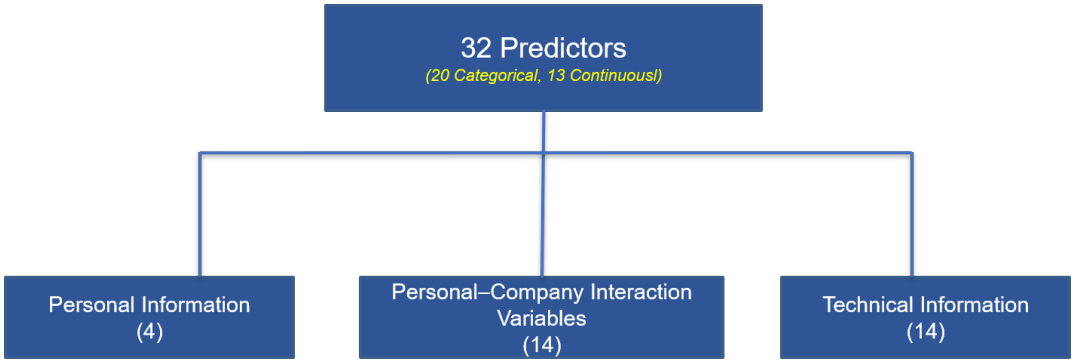
## Source

The data has been extracted from Kaggle Website - https://www.kaggle.com/pangkw/telco-churn
The data set has multiple parameters available for the past and existing customers and a response variable Churn which tells if an existing customer will abandon the company or not.

## Exploring the data

The data initially consisted of 3333 observations and 33 predictors which include 20 categorical (Gender, Senior Citizen, Marital Status, Dependent, Tenure, Phone services, etc.) and 13 continuous variables (Customer ID, Revenue, etc).

The distribution below shows the category-wise distribution of the variables.




- **Personal Information**: The personal information available about the customer. Ex: Gender
- **Personal-Company Interaction Variables:** These are the variables which record what kind of services are subscribed by the customer from the company. Ex:CustomerServiceCalls.
- **Technical Information:** This is the technical data that is only relevant for the company Ex: InternetService.

# Data Modification/Feature Engineering

## Missing Value Imputation

Missing values for **TotalRevenue** were imputed using the mean of the observations having values.

## Variable Selection

**CustomerID** – was removed as it was a unique id and wasn't helpful for prediction.
**Phone Service**- was removed as it had same value (1) for all observations.

## Converting Continuous Variables to Categorical Variables

As **Tenure** gave the value of the number of months a customer has been with the company, we converted it into **GroupTenure**, which converts months into years (0 to 6).

| Tenure | Group Tenure |
|---|---|
| 0-12 | 1 |
| 13-24 | 2 |
| 25-36 | 3 |
| 37-48 | 4 |
| 49-60 | 5 |
| 61-72 | 6 |

## Variable Levelling

StreamingMovies, the value of 'No' and 'No Internet Connection' was considered as the same level.

| StreamingMovies | |
|---|---|
| Old | New |
| Yes | 1 |
| No | 0 |
| No Internet Connection | 0 |

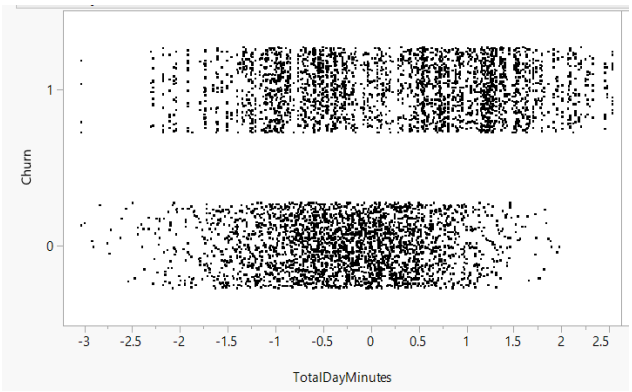
## Encoding of Categorical Variables:

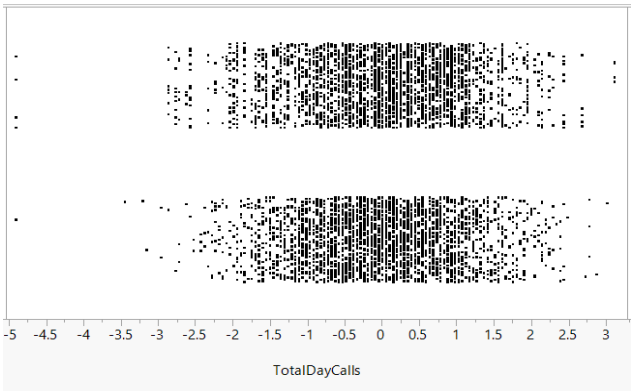
- **Gender** – Female (1), Male (0)
- For variables - **SeniorCitizen**, **MaritalStatus**, **Dependents**, **MultipleLines**, **OnlineSecurity**, **OnlineBackup**, **DeviceProtection**, **TechSupport**, **StreamingTV**, **StreamingMovies**, **PaperlessBilling**, **InternationCalling**, **VoicemailPlan**:
  1 = Yes, 0 = No
- **InternetService** – DSL (1) , Fiberoptic (2), No (0)
- **Contract** – One year(0) , Two year (2), Month-to-month (1)

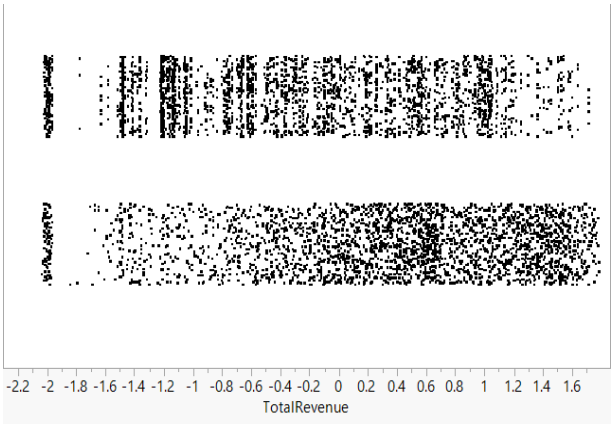
| **Encoding of Categorical Variables** | | | | | |
|---|---|---|---|---|---|
| **Gender** | Female | 1 | **TechSupport** | Female | 1 |
| | Male | 0 | | Male | 0 |
| **SeniorCitizen** | Yes | 1 | **StreamingTV** | Yes | 1 |
| | No | 0 | | No | 0 |
| **MaritalStatus** | Yes | 1 | **StreamingMovies** | Yes | 1 |
| | No | 0 | | No | 0 |
| **Dependents** | Yes | 1 | **PaperlessBilling** | Yes | 1 |
| | No | 0 | | No | 0 |
| **MultipleLines** | Yes | 1 | **InternationCalling** | Yes | 1 |
| | No | 0 | | No | 0 |
| **OnlineSecurity** | Yes | 1 | **VoicemailPlan** | Yes | 1 |
| | No | 0 | | No | 0 |
| **OnlineBackup** | Yes | 1 | **DeviceProtection** | Yes | 1 |
| | No | 0 | | No | 0 |
| **InternetService** | FibreOptic | 2 | **Contract** | Two Year | 2 |
| | DSL | 1 | | Month to Month | 1 |
| | No | 0 | | One Year | 0 |

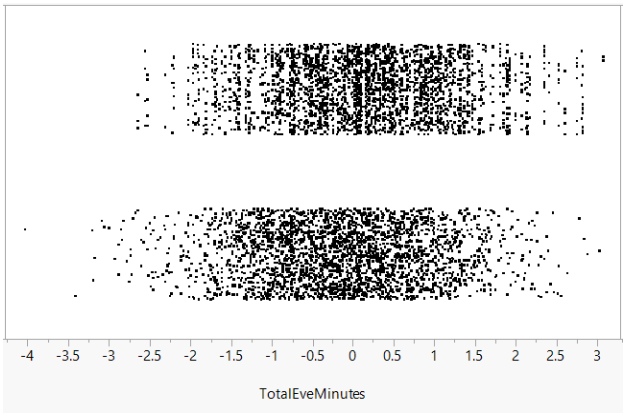
## Outlier Detection

The dataset was comparatively much cleaner as compared to the other observed datasets.
Only a handful of outliers were observed in a few variables.



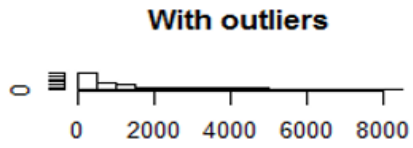



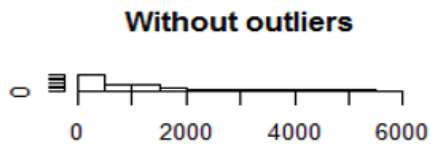




As there were only a handful of outliers and the available data size was small, we decided to perform log transformations on the data so that the data is brought closer to a normal distribution and has normal peakedness.

We preferred log transformations over removing data which had outlier values as, firstly the outliers did not majorly deviate from the distribution; also, removing data from an already small data set would have reduced the observations which might have affected the precision of prediction model.




Total Revenue




- Major outliers found on Total Revenue
- Mean without removing outliers: 1673.27
- Mean after removing outliers: 1353.66
- Even after outlier removal, data is skewed

## Normalization & Standardization

### Log Transformations

We used Log transformations mainly to reduce skewness in our data and obtain normal data. Also, algorithms like regression have an assumption of multivariate normality. Skewness and kurtosis were checked for all the variables with cut off of 0.5 for both. The variables containing 0 were transformed using “log (1 + value)” and variables not containing 0 were transformed using “log(value)”.

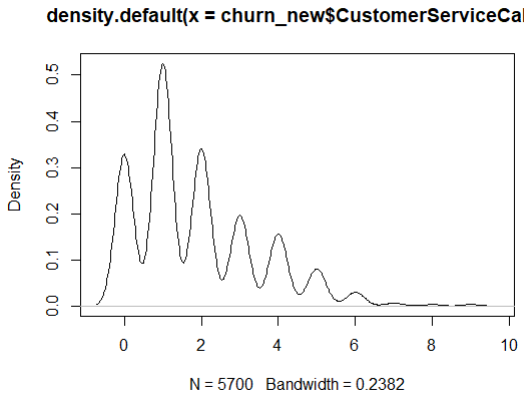
Skewness:

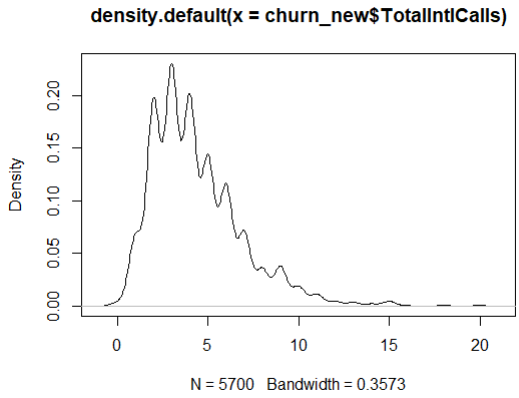
| Predictor | Before | After |
|---|---|---|
| Customer Service Calls | 1.09087 | -0.1385 |
| TotalIntCalls | 1.32088 | -0.1993 |
| Total Revenue | 1.4365 | -0.5247 |

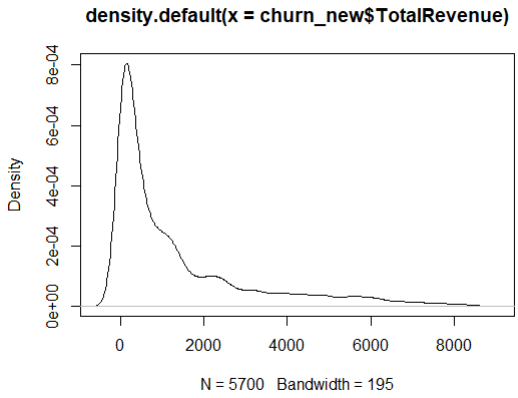
Kurtosis:

| Predictor | Before | After |
|---|---|---|
| Customer Service Calls | 6.07717 | 3.43339 |
| TotalIntCalls | 4.72652 | 2.70232 |
| Total Revenue | 4.20932 | 2.66059 |

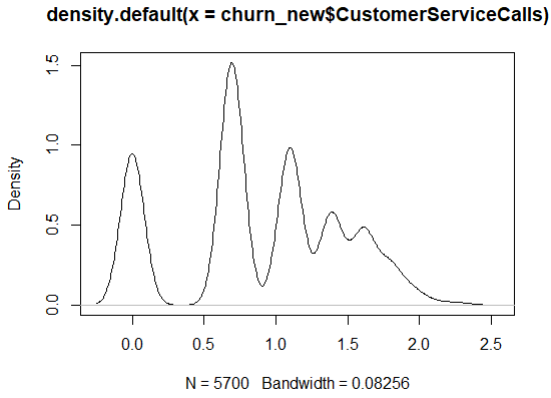
Density plots before transformation:

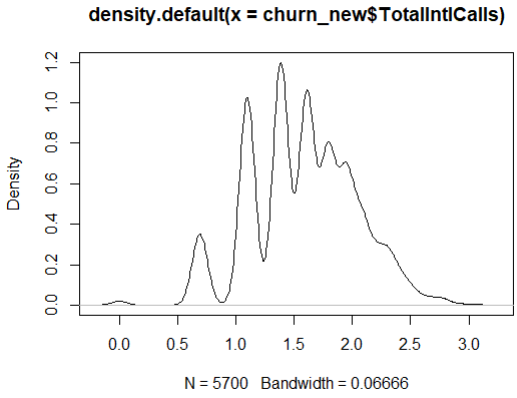


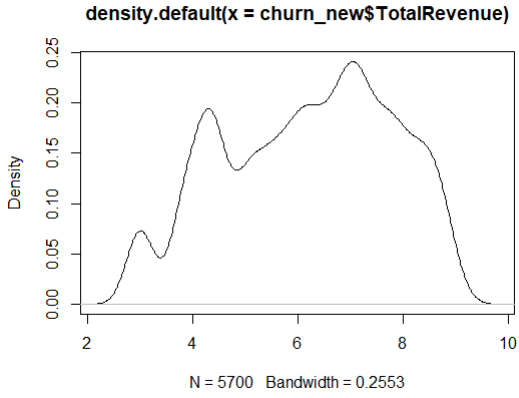






Density plots after transformation:




## Standardizing the Data

Scaling was used for all the continuous variables so that all have them have a comparable impact on the outcome variable.

Scaling also helps in visualizing and analyzing the distribution of all the variables and have a better comparison

- **Scale**: The scale transform calculates the standard deviation for an attribute and divides each value by SD. It was applied using **Z-score**.
- **Center**: The center transform calculates the mean for an attribute and subtracts it from each value.

- **Standardize**: Scaling and centering help to standardize the data; mean = 0 and SD =1.

## Results of Normalization and Standardization

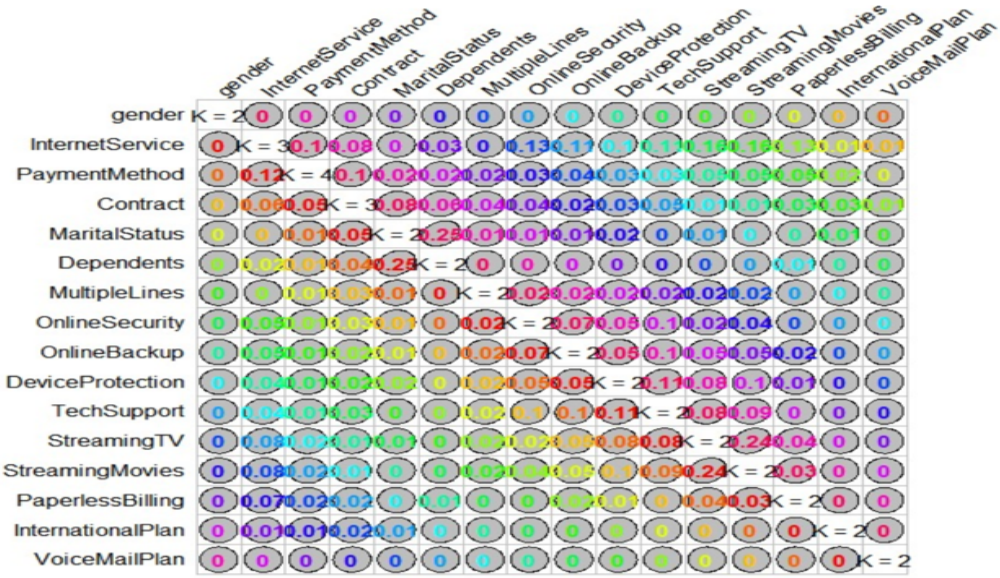
- **Skewness Reduced:**
  Log transformations helped reduce the skewness of the data and have a normal peakedness for the data.

- **Pattern Identification:**
  Log transformations made the data more pictorially insightful.

- **Easier Statistical Analysis**
  Having done the Log Transformations, and Scaling and Normalization, the analysis of the data was easier and insights could be drawn easily.

## Correlation in Data

**Collinearity Check for Categorical Variables**
The amount of collinearity among the Categorical variables was checked to see if there were any redundant variables that do not increase any value to the prediction contribution.
The following table was found.



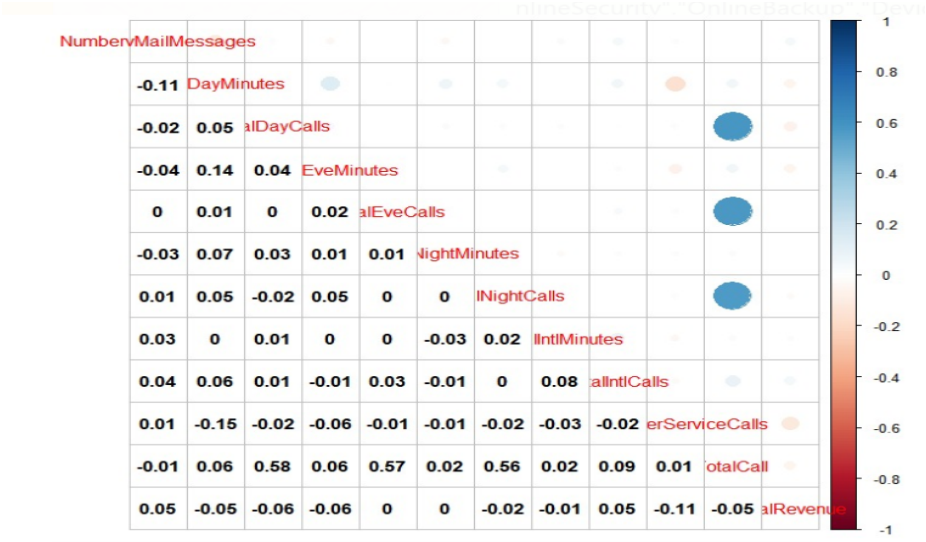

As we didn't find any red values in the correlation table, it indicates that there was no predictor pair which had a very strong correlation, i.e. an increase or decrease in one of them, causing a subsequent proportional increase or decrease in the other predictor value.

**Collinearity Check for Continuous Variables**

Similar to Categorical variables, we also carried out a collinearity test for the continuous

variables available in the data frame to check if we had a strongly correlated predictor pair, where a change in one of the variables, led to a subsequent proportional increase in the second one. One out of such variables, if found could be removed from the data.

The following table was obtained in the collinearity check for continuous variables:




Similar to the observation in Categorical variables, we found that there was not very strong correlation found between any of the continuous predictor pair. The maximum correlation which was found between a pair was 0.57, and looking at the data distribution, 0.57 was considered to be a fair amount of correlation and we did not consider it as a measure to remove one variable, as there would have been a considerable fraction of data lost with that variable.

This was also justified as after preprocessing, we only had 31 variables, except for the Churn variable to be predicted. All the variables had a contribution towards the prediction, therefore we decided to move forward with all the remaining predictors.

## Data Balancing

The dataset that we extracted, on being analyzed we found that had an imbalance in the number of “Yes” and “No” as an outcome. This was also justified as the number of customers retained by the company are sought to be lower than the number of customers not retained. So, we had a dataset with following distribution:

- Total Observations = 3333
- Churn - Yes  Observations = 483
- Churn - No Observations = 2850

To avoid our prediction to be biased toward the ‘No’ observations, we decided to oversample the data. We created dummy observations having ‘Yes’ as the churn value to make sure that the predictions are not biased using ovun sampling. Equal number of Yes and No outcome observations were created.

**After oversampling**

- Total Observations = 5700
- Churn - Yes  Observations = 2850
- Churn - No Observations = 2850

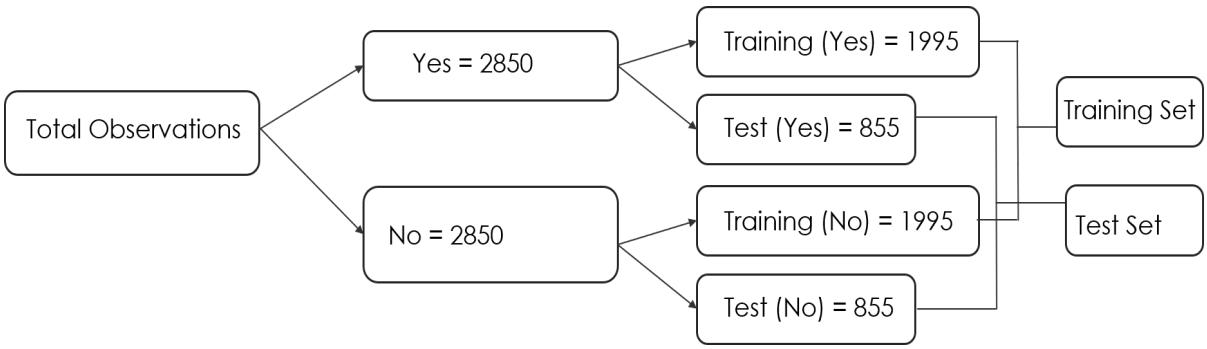
# Modeling

To finally build models that would predict whether a customer observed in future, for whom the given parameters are known, would be retained by a customer churn or not, and what are the essential parameters that might affect the churn. This model could be used by the company to reduce the number of customers abandoning the service provider.

## Training - Test Split

The dataset was divided into training and test set. This is done so that the performance of the prediction of the model can be analysed over data that is unseen by the model and what percent accuracy it gives for such observations.

Training- test split was done in a ratio of 70:30 where we trained the model using 70% of the available data and checked its performance on the unseen 30% of the data. We find the outcome of churn for this 30% data and then verify what percent of predictions made by the model match the actual data. And that is how we determined model accuracy.




## K-fold Validation

We used k-fold cross validation to train our models, keeping in mind the small size of the data. K- fold is a process where one out of the k folds is used as Validation and the rest k-1 folds are used for training. However, in this process, each observation is seen by the model at least once. Therefore, the final models were tested on the test data set.

# Models

We used the following prediction methods to build prediction models and later compared the performance of these.

- Logistic Regression
- Logistic with LASSO (Least absolute shrinkage and selection operator)
- Decision Tree
- Random Forest
- Boosted Tree
- Support vector machine
- Neural Network

## 1. Logistic Regression

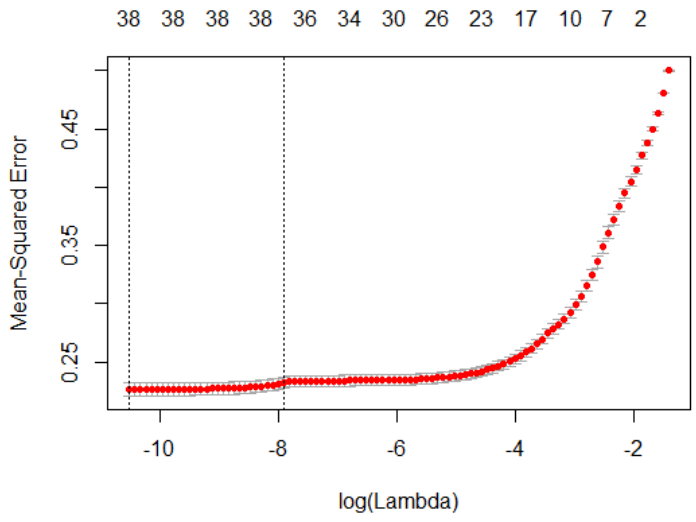
We first built a logistic regression model using all the variables and used 10 fold cross validation to train the model. This model gave us an accuracy of 85.11%.

Next, we used LASSO for variable selection and used 10 fold cross validation to obtain the tuning parameter lambda. We obtained the following 2 values of lambda:

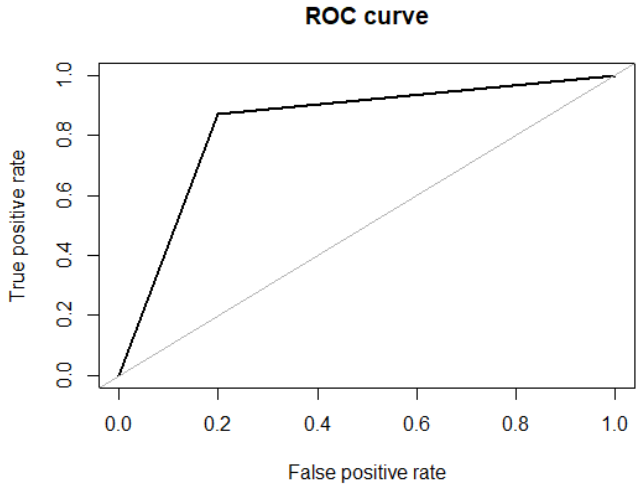
| Tuning parameter Lambda | Value |
|---|---|
| Lambda Min | 2.78047e-05 |
| Lambda 1se | 0.0002593076 |

Lambda min is the minimum value of lambda for which MSE is minimum (Most accurate model). However, we want to balance accuracy and simplicity this gives the best value of lambda for which the model is the simplest but also within one standard error of the optimal value of lambda. This balances bias with variance so we have used lambda 1se in our model.

The graph below shows the variation of MSE with log(lambda).



Following is the ROC for the obtained model:

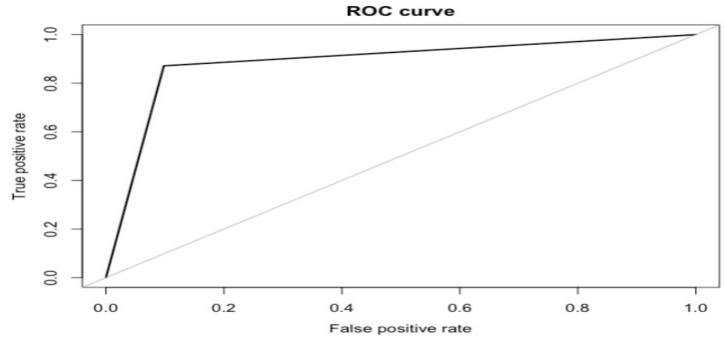



The Logistic Regression gave a misclassification rate of 14.88% and Area Under the Curve of 0.83110

## 2. Decision Tree

We have built a decision tree model using rpart function on the training data. We have then done pruning using the prune function where we have passed the training model and complexity parameter(cp) corresponding to the minimum value of cross-validation error (xerror). More levels in the tree means that it has lower classification error on the training. However, we run the risk of overfitting in this case and the cross-validation error grows as the tree gets more levels (at least, after the 'optimal' level). So, we find a minimum value of cross-validation error to pass for the pruning.

We have then used a pruned tree for prediction on the test data set.




ROC for test data

AUC for test data= 0.8865

Confusion Matrix

| Predicted | Actual 0 | 1 |
|---|---|---|
| 0 | 771 | 110 |
| 1 | 84 | 745 |

## 3. Random Forest

In Random forest there are two kind of parameters which control the bias-variance trade-off in the model and model accuracy.

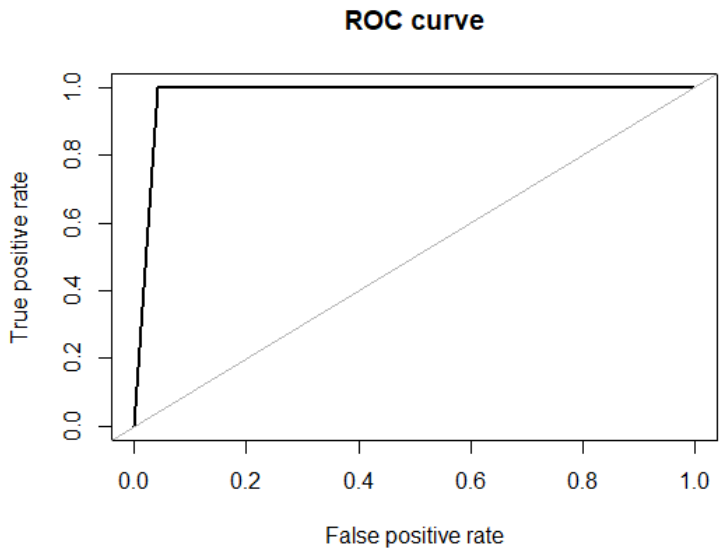
**Tuning parameters :**

- **mtry :** Number of variables randomly sampled as candidates at each split
- **ntree :** Number of trees to grow

First, we created a baseline model where we set the value of mtry to the square root of total number of predictors and ntree to 500. Next, we created a model where we do not set the tuning parameter mtry as above and only set ntree = 500. mtry is determined by the model using 10-fold repeated Cross Validation. The best model with highest accuracy was found for mtry = 20. But since, the accuracy for greater value of mtry does not show significant improvement from the lower value of mtry, we have considered low value of mtry for prediction.

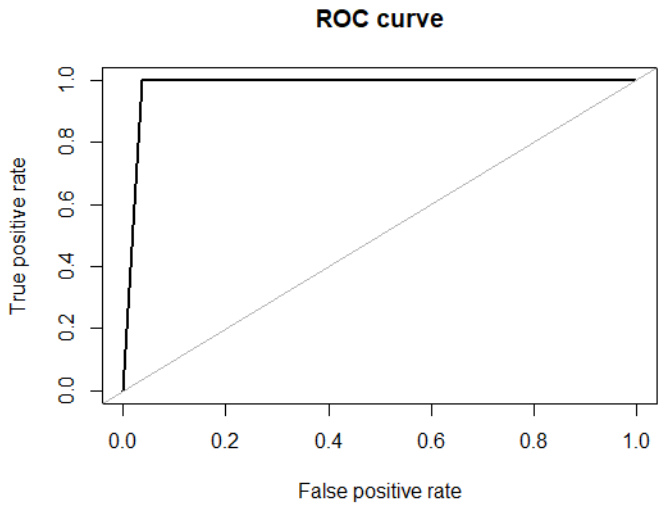
Baseline model

| mtry | ntree | AUC |
|---|---|---|
| Sqrt(no. of predictors) = 5 | 500 | 0.9777 |



Repeated 10 fold CV

| mtry | ntree | AUC |
|---|---|---|
| 20 | 500 | 0.9813 |



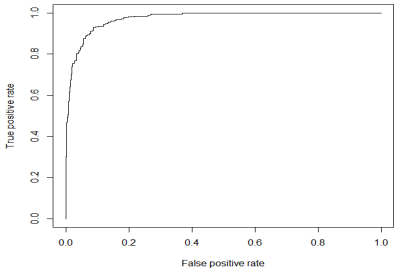

## 4. Boosted Tree

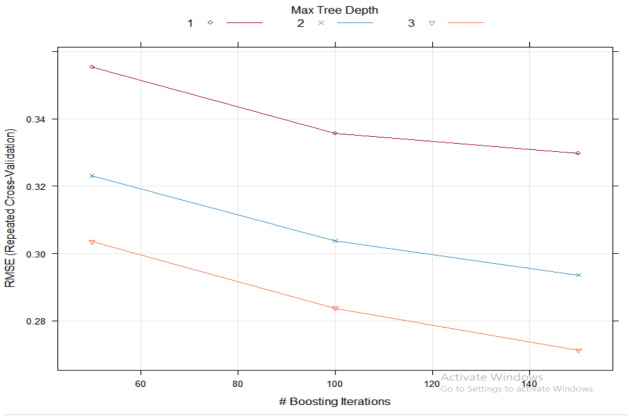
We have built the gradient boosting model on the training data. We initially passed number of trees in the model to be 1000 but model was over fitted in this case with an AUC of around 98% so we set the numbers of trees 100 for our baseline model and we got AUC as 92%. Then we have done tuning using cross validation and passed four hyper parameters number of trees, interaction depth, shrinkage and minimum number of training set samples in a node to commence splitting. RMSE was used to select the optimal model using the smallest value.

The final values used for the hyperparameters were
n.trees = 150,
interaction.depth = 3,
shrinkage(learning rate) = 0.1
minimum number of training set samples in a node to commence splitting )= 10.

AUC before tuning: 0.9226155

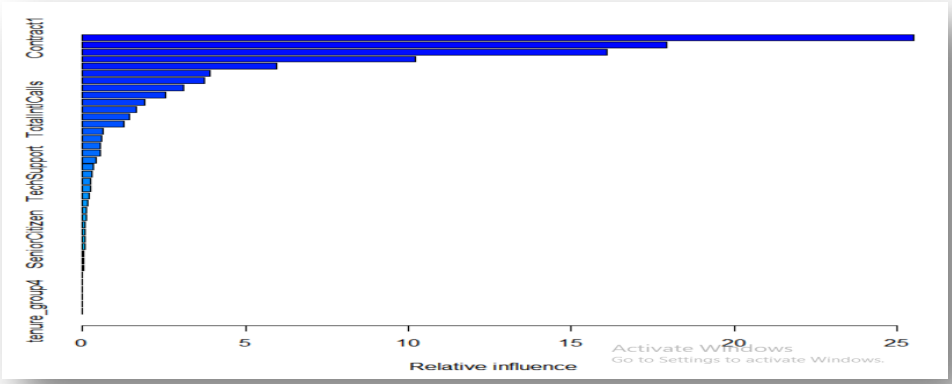
AUC after tuning: 0.9735248




## Relative influence curve as per Boosting:

Most important factor to predict churn is contract.



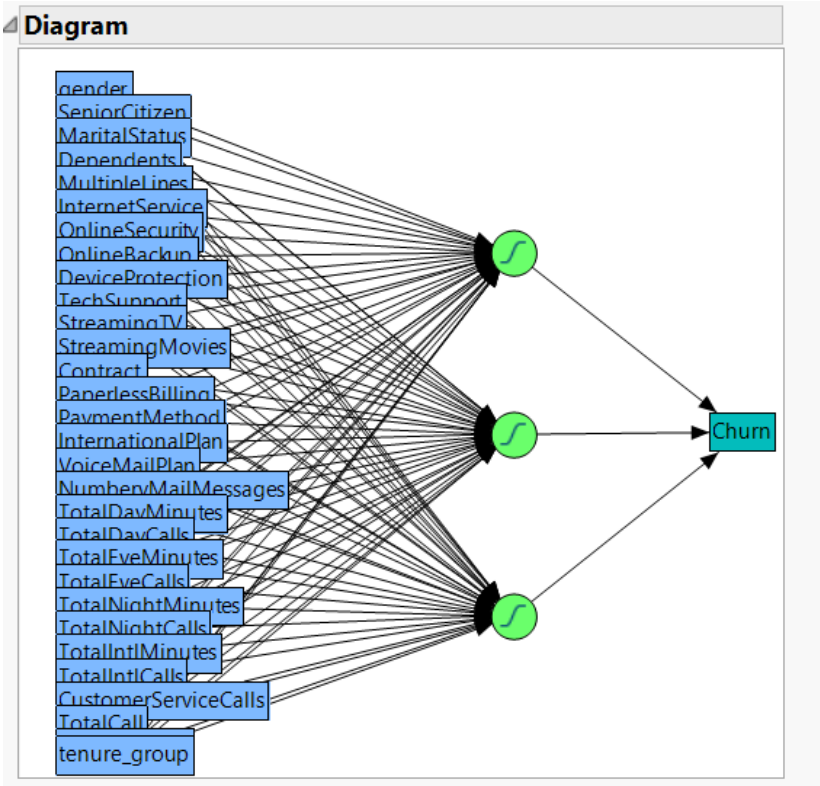

## 5. Support Vector Machine

We have used run Support Vector Machine on training for linear, Gaussian and polynomial three times without k fold, with k fold and with grid search. The SVM Gaussian optimum model is coming at cost function C=1 and smoothing parameter (sigma) = 0.03606054. The SVM Polynomial optimum model is coming at degree = 3, scale = 0.1 and C = 0.25

SVM Summary:

| | SVM - Linear | SVM - Polynomial | SVM - Gaussian |
|---|---|---|---|
| Without K Fold | 0.8251462 | 0.905848 | 0.9070175 |
| With K Fold | 0.8416011 | 0.9070015 | 0.9080166 |
| Grid Search | | 0.9 | 0.9070175 |

## 6. Neural network




R Square for multiple models:

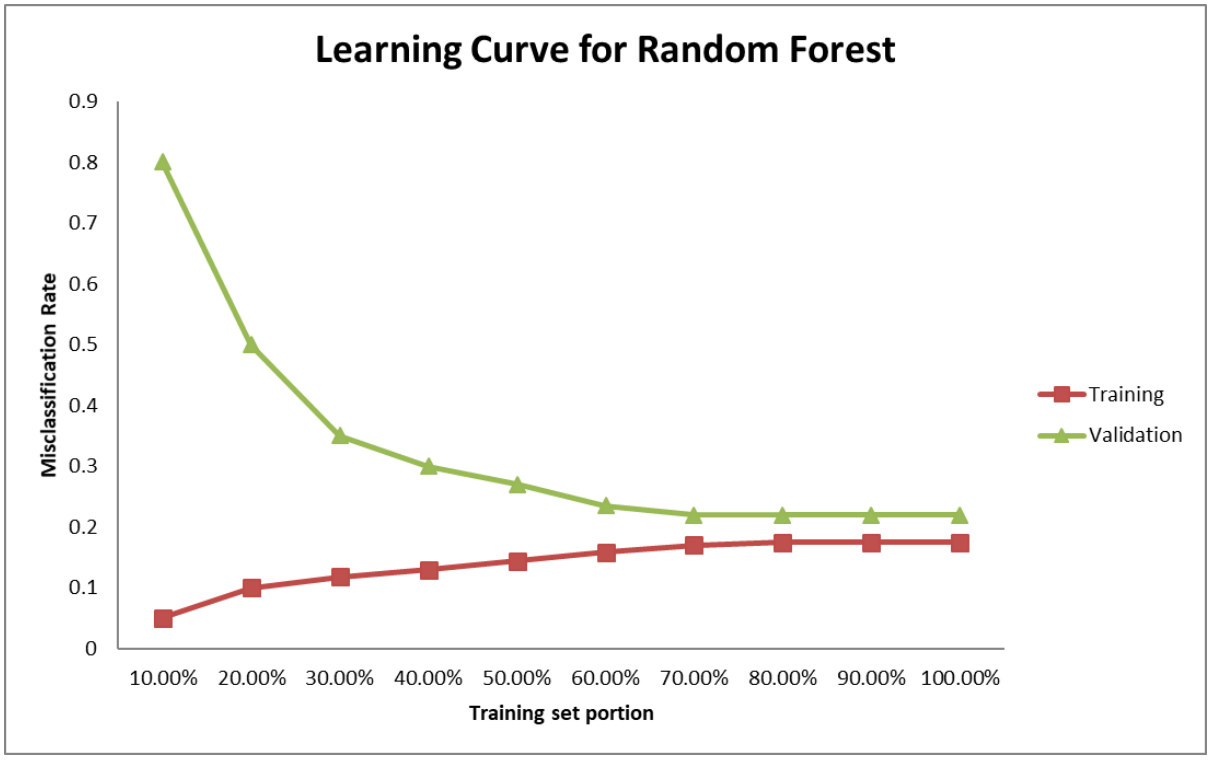
| | 5 trees | | 25 trees | |
|---|---|---|---|---|
| | Training | Validation | Training | Validation |
| **Tan** | 0.5911 | 0.5544 | 0.7581 | 0.734 |
| **Linear** | 0.5017 | 0.517 | 0.5571 | 0.5411 |
| **Gausian** | 0.791 | 0.7238 | 0.8176 | 0.6623 |
| **Complex** | 0.6066 | 0.5477 | | |
| **With k fold** | | | | |
| **Tan** | 0.6188 | 0.6091 | 0.719 | 0.698 |
| **Linear** | 0.5027 | 0.5275 | 0.507 | 0.5171 |
| **Gausian** | 0.8077 | 0.8325 | 0.8928 | 0.693 |
| **Complex** | 0.7011 | 0.7111 | | |

## Model Comparison

| Machine Learning Algorithm | AUC |
|---|---|
| Logistic Regression | 0.8310 |
| Decision Tree | 0.8865 |
| Random Forest | 0.9777 |
| Boosted Tree | 0.9735 |
| Support Vector Machine | 0.9070 |
| Neural Network | 0.8928 |

As we observe, the best Model is Random Forest which gives an R- square of 97.77

## Learning Curve




The graph shows the Learning Curve for our best model. The Train Error increases with increase in the training data size while the Test Error decreases gradually with the increase in training data size.
The crossover point shows that the data beyond 80% of the dataset would not lead to any significant impact on the model.

## Recommendations

1. This model can be used by Marketing department to put up lucrative offers to customers of different company that can be easily persuaded to subscribe to their company's services.

2. It can also be used for internal training of the customer service department, so that the customers that are more probable to churn, can be handled cautiously.

3. Another use can be identifying the customers that have high probability of churn in

the close future, giving them incentives to convince them to stay with the company.