Churn Prediction in Telecom Domain

Presented By:

M.V.S.VIDHYA SAGAR

Mentors:

Gautham & Ashish Poigal

ABSTRACT

Nowadays, the problems that are facing in most of the companies is customer churn. This has a major impact in telecom industry. In telecom industry, the subscribers are moving from a specific service or a service provider to another in a given period of time. So there is a loss of revenue due to fierce competition and loss of potential customers. The main loss is not only due to revenue generated by these customers but also the resources that are invested for the subscribers.

WHY DO CUSTOMERS CHURN ?

Due to lack of customer satisfaction like

1. Telecom service options i.e.; voice(prepaid,postpaid), data (DSL,3G,4G), voice+data.
2. Plan Costs
3. Demography

In this project, I have implemented and analyzed churn prediction in telecom for Konnect India Cellular Services which is a leading telecom services provider in India. It has a vast subscriber base, with most of them being pre-paid subscribers. I used datasets to find whether customers will churn or not.

Churns can be reduced by analyzing the past history of the potential customers systematically. In the past few years, the fast-emerging requirements from both academia and industry has helped R programming language to emerge as one of the necessary tool for visualization, computational statistics and data science.

PURPOSE

The purpose of this project is to build the global model which will reduce the overall misclassification error in predicting whether customer will churn or not. So that, companies can understand the customers who are going to churn and give them offers to avoid the churning from their network .

Customer churn is the focal concern of most of the companies which are active in industries with low switching cost. Among all the industries which suffers from this issue is telecommunications. Telecommunications industry can be considered in the top of the list with approximate annual churn rate of 30%.

Consequently, in order to tackle this problem we must recognize the churners before they going to churn. so that we can develop a model which predicts the future churners. This model has to be able to recognize the customers which tend to churn in future.

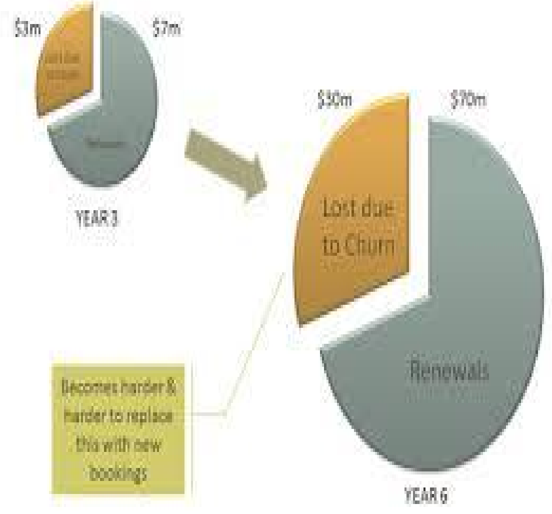
Old loyal customer are important to the telecom industry because they tend to give more revenue to the companies comparing to the new customers. So predicting the customers who are churning in advance help the telecom industry to achieve more revenue and profits.

Data Science predictive models are widely used in most of the major telecom providers like Airtel, idea, Vodaphone etc. These models are lot helpful to these companies to prevent loss of revenue from customers churning from their network.

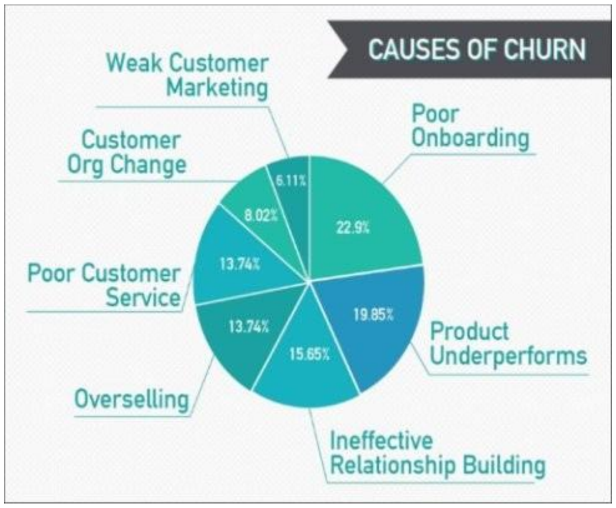
STATE OF THE ART

Acquisition and retention of new clients are one of the most significant concerns of businesses. While recipient companies concentrate on acquiring new customers, mature ones try to focus on retention of the existing ones in order to provide themselves with the opportunity of cross – selling. one of the most significant ways of increasing customers' value is to keep them for longer period of time.

This has a major impact in telecom industry. In telecom industry, the subscribers are moving from a specific service or a service provider to another in a given period of time. So there is a loss of revenue due to fierce competition and loss of potential customers. The main loss is not only due to revenue generated by these customers but also the resuorces that are invested for the subscribers. In research shows today that the companies have an average churn of 1.9 to 2 percent month on month and annualized churn ranging from 10 to 60 percent.



The customers churn due to the lack of customer satisfaction like- service options i.e.; voice(prepaid and postpaid), data plan costs and also demography.



Regarding this, examining the existing statistics concerning churn magnitude and its costs in this realm would be beneficial for gaining an appropriate picture of the importance of this area of research

- SAS (2000) reported that the telecommunications sector endures an annual rate of churn, ranging from 25 per cent to 30 percent this churn rate could still continue to increase in correlation with the growth of the market.
- The ratio (customer acquisition costs/ customer retention or satisfaction costs) would be equal to eight for the wireless companies (SAS Institute, 2000).
- Revenue in the telecom industry is calculated by ARPU(Average revenue per user) . It is the measure for the telecom companies to know how much revenue they are generating for particular financial year.
- Average revenue per user(ARPU) is calculated by dividing the total revenue by the number of the customers
- Old customer is always a desirable option to the company because the tends to produce higher profits
- Attracting new customers costs almost 5-6 times more than the retentive the old customers

- Attracting new customers comprises of new recruits of man power, cost of publicity & more discounts
- Return of investment on old customers will be more comparing to attracting new customers
- So by predicting how many customers will churn in advance help the telecom companies to retrain their customers
- From the rapidly growing market of telecom in past decade it has reached to the situation where it is saturated and there is fierce competition as new competitors are entering into the telecom market ex: In recent times reliance jio .

- Focus shifted from building a large customer base into keeping customers 'in house'.

- There it is important to identify the customers who are churning from the network and prevent them churning by giving additional offers.

The telecom industries uses multiple models to predict the churn. By using these models they are trying to reduce the churn. Many models are built using logistic regression, svm, decision trees and random forests etc.

They take the following attributes as the input data:

- Customer Demographics (age,gender,martial status,location etc.)
- Call Statistics(length of calls like local,national,international)
- Billing Information(what the customer paid for)
- Voice and Data Product(broadband services,special tariff plans etc.)
- Complaints and Disputes(customer satisfaction issues and remedial steps taken) Credit History

on the output:

Target/Response considered for the model

- The value '1' indicates the churn customers.
- The value '0' indicates the active customers.

DIFFERENT MACHINE LEARNING MODELS USED IN THE PROJECT

1.Logistic Regression
2. Decision Trees
3.Support Vector Machines
4.Random Forest

1. Logistic Regression:

Logistic regression is a statistical method for analyzing a dataset in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable (in which there are only two possible outcomes).

In logistic regression, the dependent variable is binary or dichotomous, i.e. it only contains data coded as 1 (TRUE, success, pregnant, etc.) or 0 (FALSE, failure, non-pregnant, etc.).

The goal of logistic regression is to find the best fitting (yet biologically reasonable) model to describe the relationship between the dichotomous characteristic of interest (dependent variable = response or outcome variable) and a set of independent (predictor or explanatory) variables. Logistic regression generates the coefficients (and its standard errors and significance levels) of a formula to predict a logit transformation of the probability of presence of the characteristic of interest:

The goal of logistic regression is to find the best fitting (yet biologically reasonable) model to describe the relationship between the dichotomous characteristic of interest (dependent variable = response or outcome variable) and a set of independent (predictor or explanatory) variables. Logistic regression generates the coefficients (and its standard errors and significance levels) of a formula to predict a logit transformation of the probability of presence of the characteristic of interest:
formula

$$logit(p) = b_0 + b_1X_1 + b_2X_2 + b_3X_3 + \ldots + b_kX_k$$

where p is the probability of presence of the characteristic of interest. The logit transformation is defined as the logged odds:

$$odds = \frac{p}{1-p} = \frac{probability\ of\ presence\ of\ characteristic}{probability\ of\ absence\ of\ characteristic}$$

and

$$logit(p) = \ln\left(\frac{p}{1-p}\right)$$

2.Decision Trees:

Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

For instance, in the example below, decision trees learn from data to approximate a sine curve with a set of if-then-else decision rules. The deeper the tree, the more complex the decision rules and the fitter the model.

Some advantages of decision trees are:

- Simple to understand and to interpret. Trees can be visualized.
- Requires little data preparation. Other techniques often require data normalization, dummy variables need to be created and blank values to be removed. Note however that this module does not support missing values.
- Uses a white model. If a given situation is observable in a model, the explanation for the condition is easily explained by Boolean logic. By contrast, in a black box model (eg. In an artificial neural network), results may be more difficult to interpret.
- Possible to validate a model using statistical tests. That makes it possible to account for the reliability of the model.

The disadvantages of decision trees include:

- Decision-tree learners can create over-complex trees that do not generalise the data well. This is called over fitting. Mechanisms such as pruning (not currently supported), setting the minimum number of samples required at a leaf node or setting the maximum depth of the tree are necessary to avoid this problem.
- Decision trees can be unstable because small variations in the data might result in a completely different tree being generated. This problem is mitigated by using decision trees within an ensemble.
- There are concepts that are hard to learn because decision trees do not express them easily, such as XOR, parity or multiplexer problems.
- Decision tree learners create biased trees if some classes dominate. It is therefore recommended to balance the dataset prior to fitting with the decision tree.

3.Support Vector Machines:

In machine learning, support vector machines (SVMs, also support vector networks are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

Linear SVM

we are given a training data set of n points of the form (x1,y1)..........(xn,yn) where the **yi** are either 1 or −1, each indicating the class to which the point **xi** belongs. Each **xi** is a **p**-dimensional real vector. We want to find the "maximum-margin hyper plane" that divides the group of points for which from the group of points for which which is defined so that the distance between the hyper plane and the nearest point from either group is maximized.

Nonlinear Classification (Kernal Functions)

The original maximum-margin hyper plane algorithm proposed by Vapnik in 1963 constructed a linear classifier. However, in 1992, Bernhard E. Boser, Isabelle M. Guyon and Vladimir N. Vapnik suggested a way to create nonlinear classifiers by applying the kernel trick (originally proposed by Aizerman et al. to maximum-margin hyper planes. The resulting algorithm is formally similar, except that every dot product is replaced by a nonlinear kernel function. This allows the algorithm to fit the maximum-margin hyper plane in a transformed feature space. The transformation may be nonlinear and the transformed space high dimensional; although the classifier is a hyper plane in the transformed feature space, it may be nonlinear in the original input space.

Some common kernels include

- Polynomial kernel
- Radial basis kernel
- Hyperbolic tangent kernel

5.Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of over fitting to their training set.

The first algorithm for random decision forests was created by Tin Kam Ho using the random subspace method which, in Ho's formulation, is a way to implement the "stochastic discrimination" approach to classification proposed by Eugene Kleinberg.

The first algorithm for random decision forests was created by Tin Kam Ho using the random subspace method which, in Ho's formulation, is a way to implement the "stochastic discrimination" approach to classification proposed by Eugene Kleinberg.

The training algorithm for random forests applies the general technique of bootstrap aggregating, or bagging, to tree learners. Given a training set X = x1, ..., xn with responses Y = y1, ..., yn, bagging repeatedly (B times) selects a random sample with replacement of the training set and fits trees to these samples:

For b = 1, ..., B:

1. Sample, with replacement, n training examples from X, Y; call these Xb, Yb.
2. Train a decision or regression tree fb on Xb, Yb.

Variable Importance:

Random forests can be used to rank the importance of variables in a regression or classification problem in a natural way. The following technique was described in Breiman's original paper and is implemented in the R package random Forest

To measure the importance of the j-th feature after training, the values of the j -th feature are permuted among the training data and the out-of-bag error is again computed on this perturbed data set. The importance score for the j-th feature is computed by averaging the difference in out-of-bag error before and after the permutation over all trees. The score is normalized by the standard deviation of these differences.

Features which produce large values for this score are ranked as more important than features which produce small values.

METHOD

A. Problem Statement:

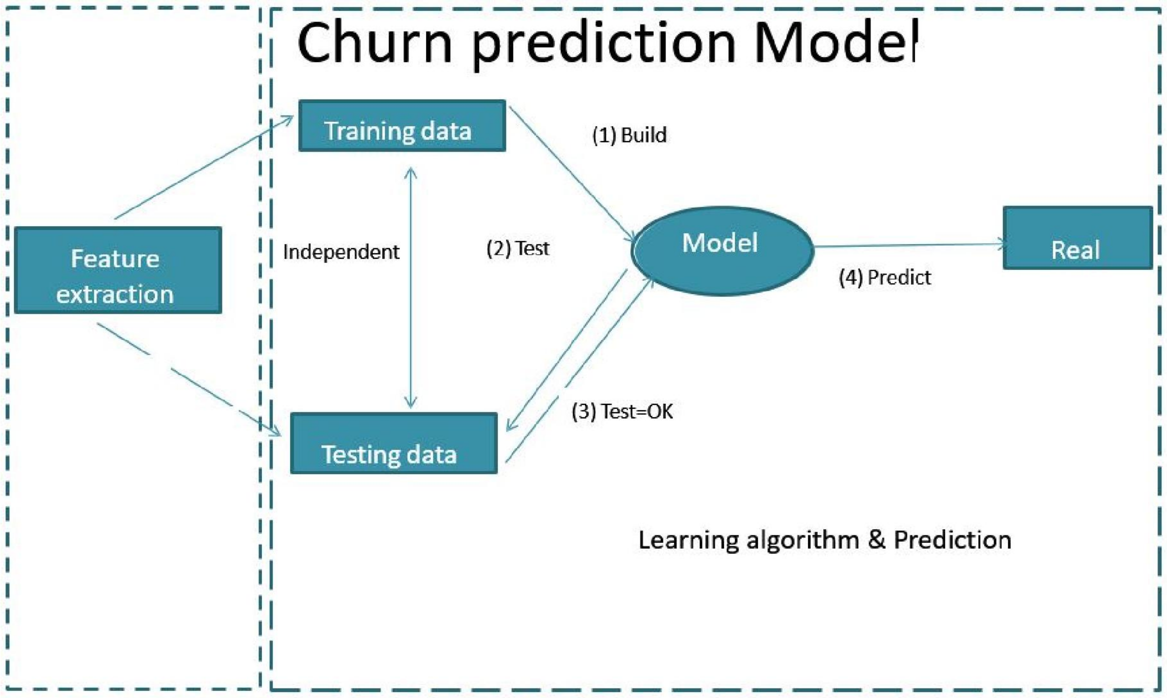
Konnect India Cellular Services is a leading telecom services provider in India. It has a vast subscriber base, with most of them being pre-paid subscribers. Our main goal is to predict churn for this telecom major. Churn with respect to the Telecom industry, is defined as the percentage of subscribers moving from a specific service or a service provider to another in a given period of time. Given the data of users' call and data usage patterns and demographics, the task is to.

• Build a global model that predicts churn of the entire subscriber base, where overall misclassification rate is minimized.

B. Approach:

- Standardize the data using "range"(min- max) method.
- Remove outliers in the data by using cooks distance.
- Remove multicollinearity variables from the data.
- Check whether there is a class imbalance or not.
- Check for the missing values, if there impute them by central imputation or knn imputation.
- Select the important features from the dataset.

- Divide the dataset into training and testing with same important features selected earlier.
- Build the model on the training dataset and test the accuracy's on the test dataset.
- Tune the model on the train dataset and again check error metrics on the test dataset until we get optimal tuning parameter.

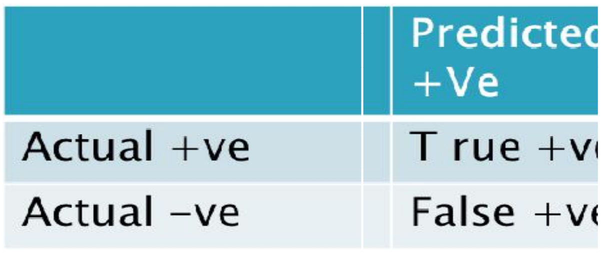


- In same way build multiple models and select the best model which gives good Accuracy and reduces the misclassification error.

Important Error Metrics to be considered in churn prediction in telecom domain are

1.Accuracy

2.Recall



	Predicte +Ve
Actual +ve	T rue +v
Actual –ve	False +v

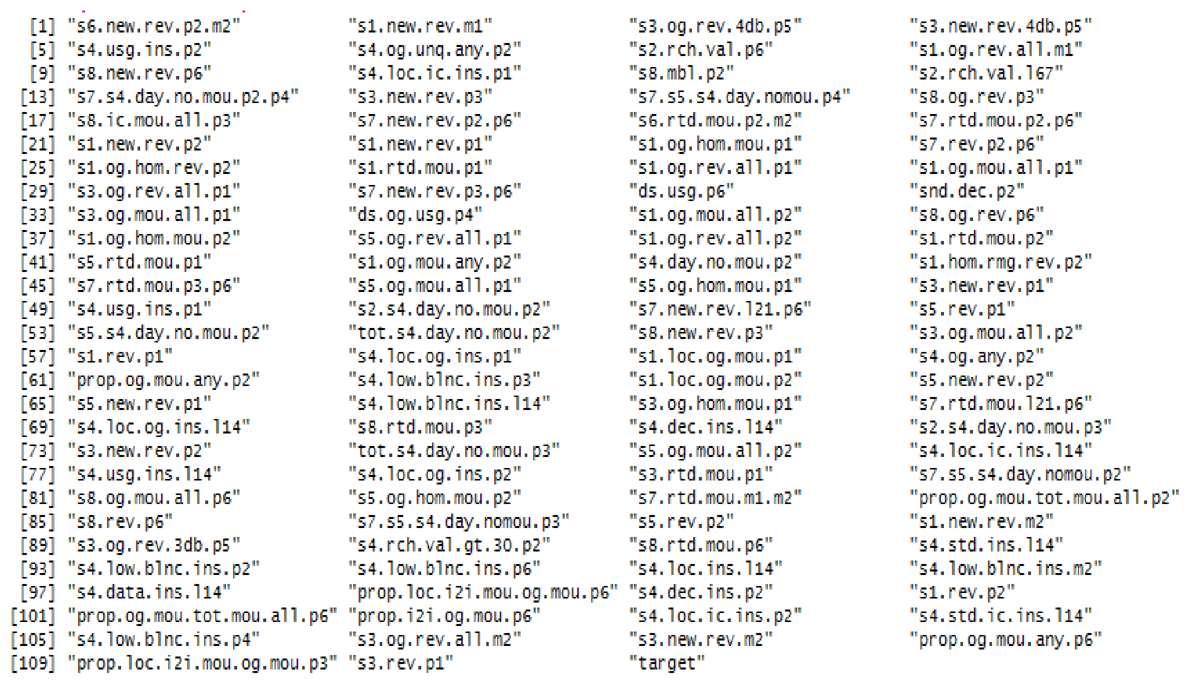
Accuracy: it is predicting correctly customers those who are going to churn as churners and those who are not going to churn as non-churners

Recall: Customer who Churns (+ve) should not be predicted as non churner(-ve)

DATA

There are total 111 attributes in the dataset out of which only one attribute is the target variable i.e., it indicates whether customer will churn or not.

C. Attributes in the Dataset



```
  [1] "s6.new.rev.p2.m2"           "s1.new.rev.m1"              "s3.og.rev.4db.p5"         "s3.new.rev.4db.p5"
  [5] "s4.usg.ins.p2"              "s4.og.unq.any.p2"           "s2.rch.val.p6"            "s1.og.rev.all.m1"
  [9] "s8.new.rev.p6"              "s4.loc.ic.ins.p1"           "s8.mbl.p2"                "s2.rch.val.l67"
 [13] "s7.s4.day.no.mou.p2.p4"     "s3.new.rev.p3"              "s7.s5.s4.day.nomou.p4"    "s8.og.rev.p3"
 [17] "s8.ic.mou.all.p3"           "s7.new.rev.p2.p6"           "s6.rtd.mou.p2.m2"         "s7.rtd.mou.p2.p6"
 [21] "s1.new.rev.p2"              "s1.new.rev.p1"              "s1.og.hom.mou.p1"         "s7.rev.p2.p6"
 [25] "s1.og.hom.rev.p2"           "s1.rtd.mou.p1"              "s1.og.rev.all.p1"         "s1.og.mou.all.p1"
 [29] "s3.og.rev.all.p1"           "s7.new.rev.p3.p6"           "ds.usg.p6"                "snd.dec.p2"
 [33] "s3.og.mou.all.p1"           "ds.og.usg.p4"               "s1.og.mou.all.p2"         "s8.og.rev.p6"
 [37] "s1.og.hom.mou.p2"           "s5.og.rev.all.p1"           "s1.og.rev.all.p2"         "s1.rtd.mou.p2"
 [41] "s5.rtd.mou.p1"              "s1.og.mou.any.p2"           "s4.day.no.mou.p2"         "s1.hom.rmg.rev.p2"
 [45] "s7.rtd.mou.p3.p6"           "s5.og.mou.all.p1"           "s5.og.hom.mou.p1"         "s3.new.rev.p1"
 [49] "s4.usg.ins.p1"              "s2.s4.day.no.mou.p2"        "s7.new.rev.l21.p6"        "s5.rev.p1"
 [53] "s5.s4.day.no.mou.p2"        "tot.s4.day.no.mou.p2"       "s8.new.rev.p3"            "s3.og.mou.all.p2"
 [57] "s1.rev.p1"                  "s4.loc.og.ins.p1"           "s1.loc.og.mou.p1"         "s4.og.any.p2"
 [61] "prop.og.mou.any.p2"         "s4.low.blnc.ins.p3"         "s1.loc.og.mou.p2"         "s5.new.rev.p2"
 [65] "s5.new.rev.p1"              "s4.low.blnc.ins.l14"        "s3.og.hom.mou.p1"         "s7.rtd.mou.l21.p6"
 [69] "s4.loc.og.ins.l14"          "s8.rtd.mou.p3"              "s4.dec.ins.l14"           "s2.s4.day.no.mou.p3"
 [73] "s3.new.rev.p2"              "tot.s4.day.no.mou.p3"       "s5.og.mou.all.p2"         "s4.loc.ic.ins.l14"
 [77] "s4.usg.ins.l14"             "s4.loc.og.ins.p2"           "s3.rtd.mou.p1"            "s7.s5.s4.day.nomou.p2"
 [81] "s8.og.mou.all.p6"           "s5.og.hom.mou.p2"           "s7.rtd.mou.m1.m2"         "prop.og.mou.tot.mou.all.p2"
 [85] "s8.rev.p6"                  "s7.s5.s4.day.nomou.p3"      "s5.rev.p2"                "s1.new.rev.m2"
 [89] "s3.og.rev.3db.p5"           "s4.rch.val.gt.30.p2"        "s8.rtd.mou.p6"            "s4.std.ins.l14"
 [93] "s4.low.blnc.ins.p2"         "s4.low.blnc.ins.p6"         "s4.loc.ins.l14"           "s4.low.blnc.ins.m2"
 [97] "s4.data.ins.l14"            "prop.loc.i2i.mou.og.mou.p6" "s4.dec.ins.p2"            "s1.rev.p2"
[101] "prop.og.mou.tot.mou.all.p6" "prop.i2i.og.mou.p6"         "s4.loc.ic.ins.p2"         "s4.std.ic.ins.l14"
[105] "s4.low.blnc.ins.p4"         "s3.og.rev.all.m2"           "s3.new.rev.m2"            "prop.og.mou.any.p6"
[109] "prop.loc.i2i.mou.og.mou.p3" "s3.rev.p1"                  "target"
```

We are having only very minimal information about the variables i.e.,

Rev:	Revenue
ic :	Incoming calls
Og :	Outgoing calls
Rmg:	Roaming
Usg :	Usage
Mbl :	Mobile
Rch:	Recharge
Mou:	Minutes of usage
Loc :	Location

D. VISUALIZING THE DATA

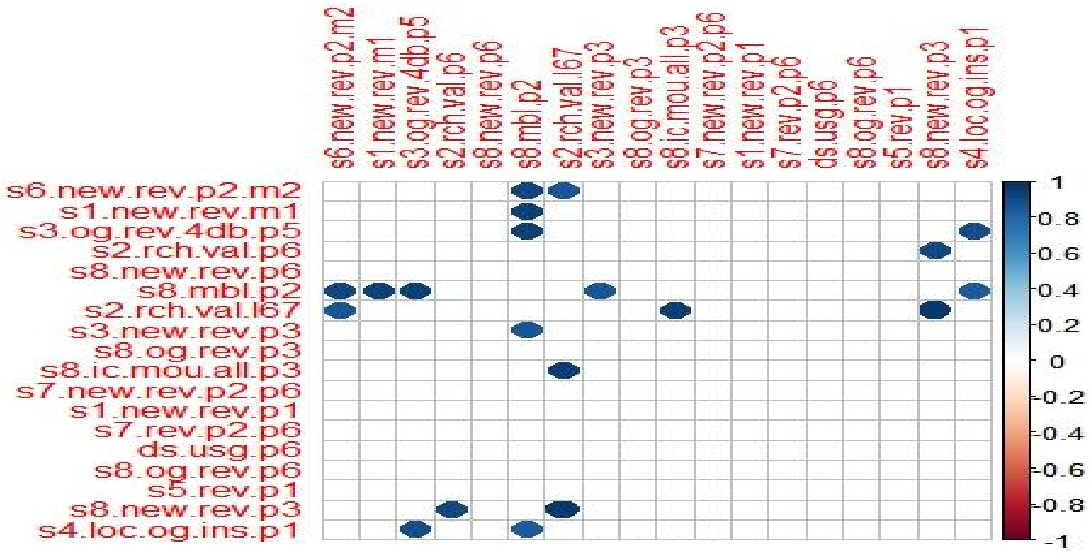


fig-1

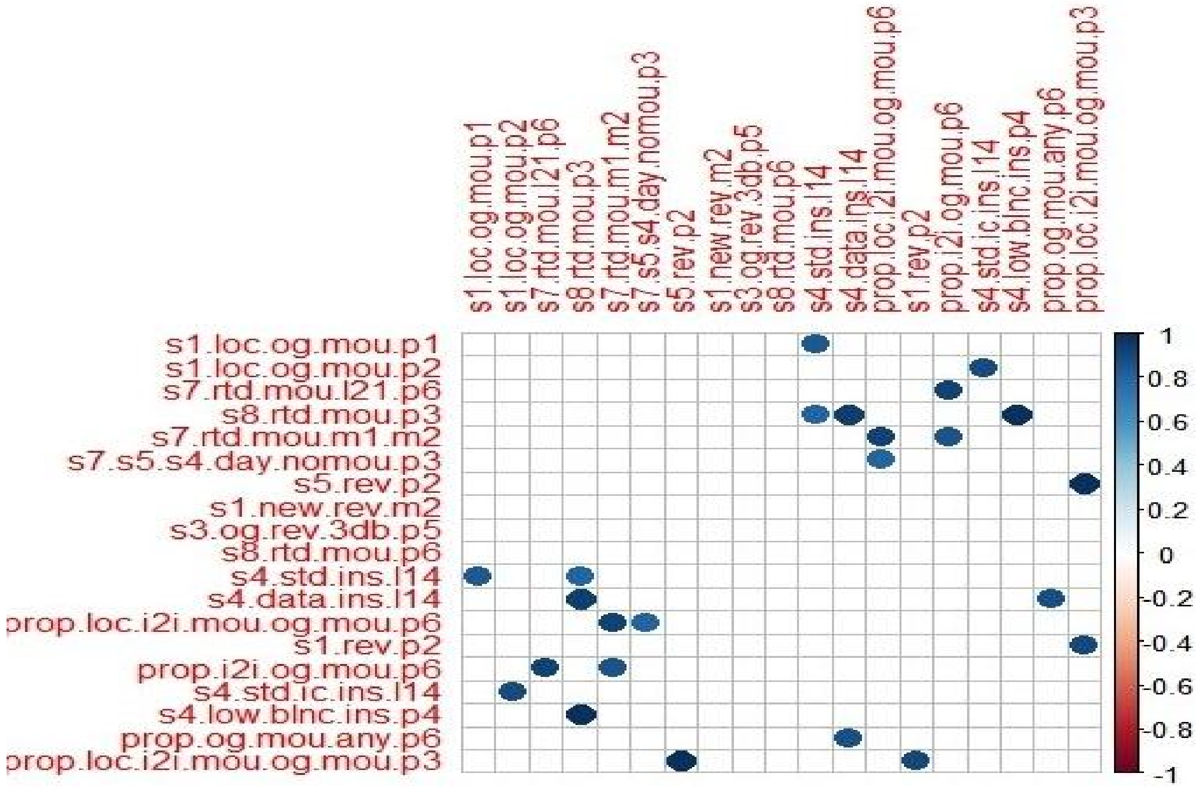


fig-2

The above are the correlation plots which shows how predictor variables are correlated among themselves and the correlation is above 80%.

From the plots it is clear that, the correlation is quite high between the predictor variables.

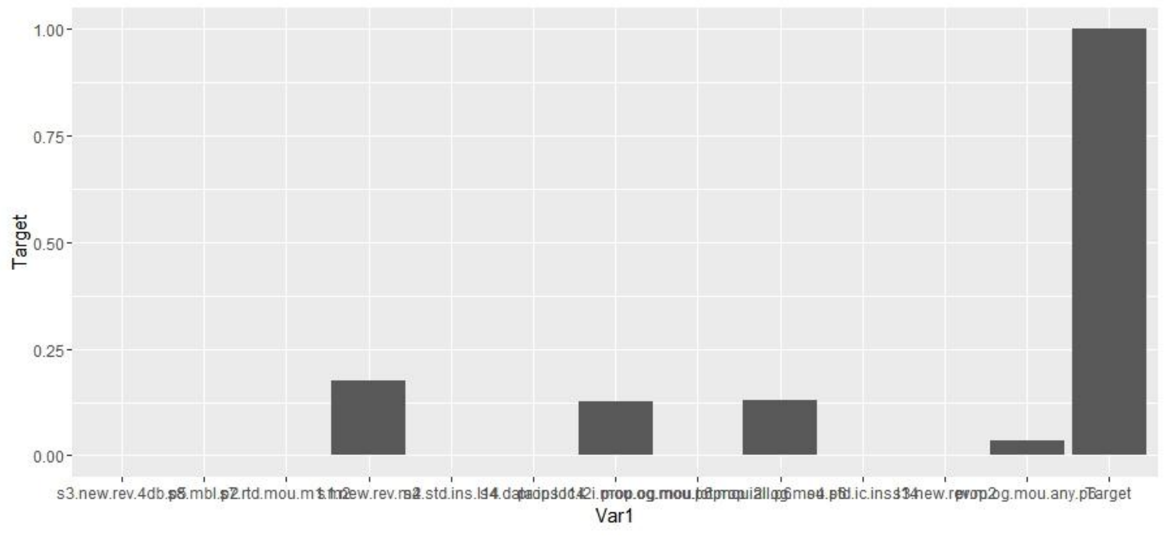


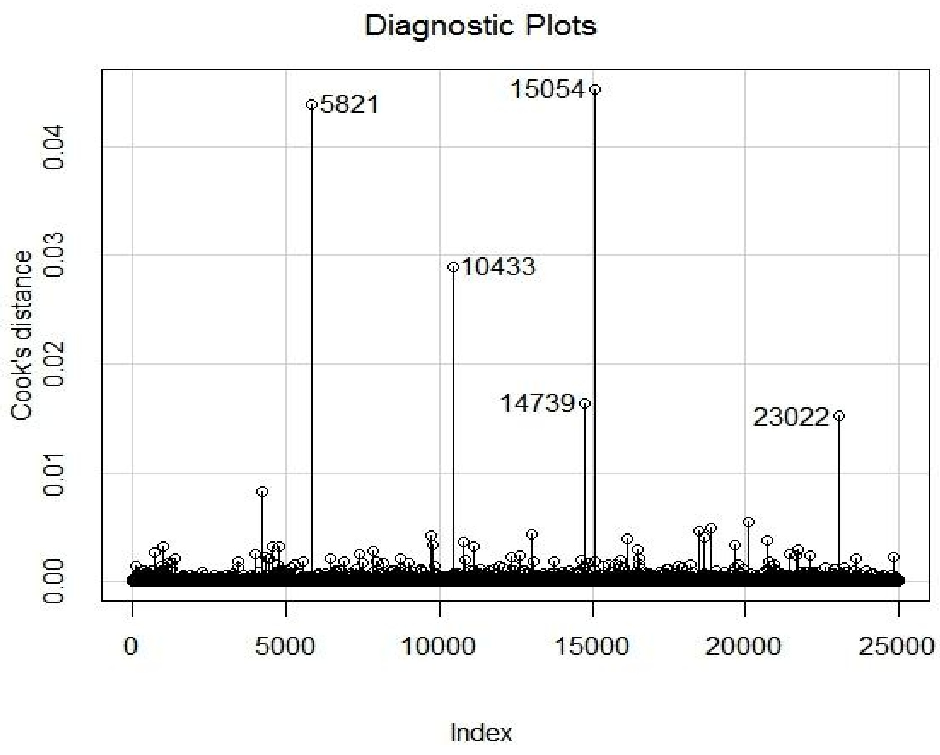
fig-3

The above plot shows how predictors are correlated to the target variables. It’s been observed that there is very less correlation between the predictor variables and the target variable.

So therefore, the insights I got from the dataset by visualizing is predictors are highly correlated among themselves and least correlated to the target variable.

E. Removing outliers:

I have identified few outliers in the data by using cooks distance.



From the above figure I have identified there are only 5 outliers in the dataset.

F. Important Feature Selection:

Techniques for Feature Selection can be divided in two approaches: feature ranking and subset selection.

Feature Ranking Approach:

In the ranking approach, features are ranked by some criteria and those which are above a defined threshold are selected. A general algorithm can be considered for such approach where you just need to decide which one if the best ranking criteria to be used. In the F-Selector Package, such criteria is represented by a set of functions that calculate weights to your features according to a model..

Pseudo code

```
for each feature F_i
   wf_i = getFeatureWeight(F_i)
   add wf_i to weight_list
sort weight_list
choose top-k features
```

we can use random forest algorithm to rank the features and select the attributes which are more important .

Random forest for feature selection:

Variable importance can be calculated in the random forest by two methods

1. Mean Decrees in Gini Coefficient of a variable

2. Mean Decrease In Accuracy of variable

The mean decrease in accuracy of a variable is determined during the out of bag error calculation phase. The more the accuracy of the random forest decreases due to the exclusion (or permutation) of a single variable, the more important that variable is deemed, and therefore variables with a large mean decrease in accuracy are more important for classification of the data. The mean decrease in Gini coefficient is a measure of how each variable contributes to the homogeneity of the nodes and leaves in the resulting random forest. Each time a particular variable is used to split a node, the Gini coefficient for the child nodes are calculated and compared to that of the original node. The Gini coefficient is a measure of homogeneity from 0 (homogeneous) to 1 (heterogeneous). The changes in Gini are summed for each variable and normalized at the end of the calculation. Variables that result in nodes with higher purity have a higher decrease in Gini coefficient.

RESULTS

I have applied 4 different classification algorithms to choose the best model which is giving good recall and accuracy.

	Random Forest	Logistic Regression	Decision Trees	SVM (linear)	SVM (radial)
Accuracy	83.83	73.5	80.98	73.50	81.06
Recall	82.31	70.4	78.83	68.72	75.65

- Its been observed that Random Forests gives 83.3 accuracy and recall of 82.31 with tuning parameters of ntree=300.
- Logistic Regression gives accuracy of 73.5 and recall of 70.4 by keeping the threshold at 0.5
- Decision Trees (C5.0) gives accuracy of 80.98 and recall of 78.83 on test data.
- SVM with linear kernel gives accuracy of 73.5 and recall of 68.72 with a tuning cost parameter of 0.1.
- SVM with radial kernel gives accuracy of 81.06 and recall of 75.65 with a tuning cost parameter of 0.5 and gamma of 0.2.

ANALYSIS

By analyzing the results it is clear that only Random Forests gives good recall and accuracy among all other classifiers. so as per our business scenario recall is more important that precision i.e., Customers who are churning should not be predicted as non-churners .

So I finally conclude that Random Forest is appropriate for this churn prediction in the telecom domain.

Calculation of error metrics for Random Forests:

Actual/Predicted	Not Churn	Churn
Not Churn	3233	557
Churn	658	3051

Recall : 3051/(3051+658) = 82.31

Accuracy : (3233+3051)/(3233+3051+658+557) = 83.83

APPENDICES

Source code:

```
#clearing the directory
  rm(list=ls(all=T))

#setting the working directory
setwd("C:\\Users\\mvsvs\\Desktop\\insofe material\\PROJECTS\\churn data\\data")

#reading the data
churn_data<-read.csv("Data.csv", header=T, sep=',',na.strings = c("?","NA","NULL"))

#checking missing values
sum(is.na(churn_data))
#zero missing values

#understanding of data
summary(churn_data)
dim(churn_data) #give the dimensions of the data
str(churn_data) #shows the structure of the data
nrow(churn_data) #25000 rows
ncol(churn_data) #111 attributes

#quantiles for more understanding of data
quantiles <-data.frame(apply(churn_data,2,function(x) {quantile(x)}))
quantiles


##checking unique values
apply(churn_data,2,function(x){length(unique(x))})
churn_data$target

#type convertions
churn_data$target<-as.factor(churn_data$target)
churn_data$target

#moving target attribute to the end
churn_data = churn_data[,c(setdiff(names(churn_data),"target"),"target")]

#visualization of data
#plotting histograms for first fifty attribute
par("mar")
par(mar=c(1,1,1,1))
par(mfrow=c(5,10))
for(i in 1:50) {
  hist(churn_data[,i], main=names(churn_data)[i])
}
#plotting histograms for remaining attributes
par(mfrow=c(5,13))
for(i in 51:111) {
  hist(churn_data[,i], main=names(churn_data)[i])
}

#corrgram
install.packages("corrgram")
library(corrgram)

corrgram(churn_data, order=NULL, panel=panel.shade, text.panel=panel.txt,
         main="Correlogram")


#boxplot
boxplot(churn_data)
#ploting outliers
LogReg <- glm(target ~., data=churn_data,family=binomial)
summary(LogReg)
#install.packages("car")
install.packages("pbkrtest")
install.packages("lme4")
install.packages("car")
library(car)
outliers<-influenceIndexPlot(LogReg,vars = c("cook"),id.n = 5)
#removing outliers from the data
churn_data1 <- churn_data[-c(5821,10433,14739,15054,23022),]
names(churn_data)
boxplot(churn_data1)

##removing outliers
install.packages("outliers")
library(outliers)
churn_data2 <- churn_data[,c(1:110)]
churn_data2 = rm.outlier(churn_data2,fill = TRUE)
boxplot(churn_data2)
names(churn_data2)
churn_data2$target = churn_data$target
names(churn_data2)
```

```r
#looking at points which are outside the boxplot
outliers = apply(churn_data2[,c(1:110)],2,function (x) boxplot(x)$out)
outlier_data = lapply(churn_data2[,c(1:110)],function (x) boxplot(x)$out)
lapply(outlier_data,length)
# some columns have many outliers
# we cannot consider them as outliers
# the data is skewed

#standardizing the data Using range method
library(vegan)
standardized_data <- churn_data2[,-c(111)] # removing target attribute
standardized_data <- decostand(standardized_data,"range")
standardized_data

#attaching the target attribute
standardized_data$target<-churn_data2$target
names(standardized_data)


#checking multicollinearity
library("VIF")
names(standardized_data)
churnfinal<-data.frame(subset(standardized_data[-c(111)]))
xyz = vif(as.numeric(standardized_data$target),churnfinal)
str(xyz)
y=data.frame(xyz$modelmatrix)
View(y)
data_vif = data.frame(y,standardized_data$target)
dim(data_vif)
names(data_vif)
--------------------------------------------------------------------------------
# checking important attributes by using random forests
install.packages("randomForest")
library(randomForest)
data_rf <- randomForest(standardized_data.target ~., data=data_vif, keep.forest=TRUE,
ntree=36)
summary(data_rf)
print(data_rf)
names(data_rf)
data_rf$predicted
data_rf$fitted
data_rf$importance
round(importance(data_rf), 2)
```

```
# Extract and store important variables obtained from the random forest model
data_rf <- data.frame(data_rf$importance)
data_rf <- data.frame(row.names(data_rf),data_rf[,1])
colnames(data_rf) = c('Attributes','Importance')
data_rf1 <- data_rf[order(data_rf$Importance , decreasing = TRUE),]
data_rf1
dim(data_rf1)
data_rf <- data_rf1[1:20,]
names(data_rf)
data_rf$Attributes
#selecting the attributes with highest mean decrease genefrom random forest
data_rf$Attributes%in%names(data_vif)

#subsetting data with selected variables
data_newrf <- churn_data1[data_rf$Attributes]
data_newrf$target <- churn_data1$target
names(data_newrf)

# Checking class imbalance
#install.packages("ROSE")
library(ROSE)
data_new1 <- ROSE(target ~ ., data = data_newrf, seed = 1)$data
table(data_new1$target)
str(data_new1)
names(data_new1)


# splitting of data into train and test
rows<-seq(1,nrow(data_new1),1)
set.seed(1129)
trainrows<-sample(rows,0.7*nrow(data_new1))
Train<-data_new1[trainrows,]
Test<-data_new1[-trainrows,]
names(Train)

#---------------------model building---------------------------------------
## 1) random forest
library(randomForest)
random_forest <- randomForest(target ~ ., data=Train, keep.forest=TRUE, ntree=300)

# View results and understand important attributes
print(random_forest)
random_forest$predicted
random_forest$importance # gives 1st column (accuracy will reduce if imp var are
removed)
```

```
# Predict on Train data
pred_model_train <-predict(random_forest,Train[,-c(21)],type="response",
norm.votes=TRUE)
result_train <- table("actual _values"= Train$target,pred_model_train);result_train
# or table(trainR$target, predict(hepatitis_rf, trainR, type="response",
norm.votes=TRUE))

# Prediction Test Data
pred_model_test <-predict(random_forest,Test[,-c(21)],type="response",
norm.votes=TRUE)
result_test <- table("actual _values"= Test$target,pred_model_test);result_test

# Accuracy, Precision and Recall on test
test_accuracy <- sum(diag(result_test))/sum(result_test)*100;test_accuracy
test_recall <- ((result_test[2,2])/(result_test[2,2]+result_test[2,1])*100);test_recall
test_precision <-((result_test[2,2])/(result_test[2,2]+result_test[1,2])*100);test_precision

--------------------------------------------
## 2)logistic regression
Log12 = glm(target~., data = Train, family=binomial)
summary(Log12)

#Accuracy on the training set
predicttrain = predict(Log12, type="response", newdata=Train)

# Confusion matrix with threshold of 0.5
metrix=table(Train$target, predicttrain > 0.5)

# accuracy,recall and precision on train data
accuracy =(metrix[1,1]+metrix[2,2])/(length(predicttrain));accuracy
Recall = metrix[2,2]/(metrix[2,2]+metrix[2,1]);Recall
Precision = metrix[2,2]/(metrix[2,2]+metrix[1,2]);Precision


# Predictions on the test set
predictTest = predict(Log12, type="response", newdata=Test)

# Confusion matrix with threshold of 0.5
metrix1= table(Test$target, predictTest > 0.5)

# accuracy,recall and precision on test
accuracy =(metrix1[1,1]+metrix1[2,2])/(length(predictTest));accuracy
Recall= metrix1[2,2]/(metrix1[2,2]+metrix1[2,1]);Recall
Precision = metrix1[2,2]/(metrix1[2,2]+metrix1[1,2]);Precision
```

```
-----------------------------------------------------

##3) Decision trees
###building classification model using c50
library(C50)
C50_model =C5.0(target~.,data=Train,rules=T )
summary(C50_model)
#building confusion matrix for C5.0
conf.train=table(Train$target, predict(C50_model, newdata=Train, type="class"))
conf.test=table(Test$target, predict(C50_model, newdata=Test, type="class"))

#compute error matrix on train data
accuracy <- (sum(diag(conf.train))/sum(conf.train))*100;accuracy
recall<-(conf.train[2,2]/(conf.train[2,1]+conf.train[2,2]))*100;recall
precision<-(conf.train[2,2])/(conf.train[1,2]+conf.train[2,2])*100;precision

#compute error matrix on test data
accuracy <- (sum(diag(conf.test))/sum(conf.test))*100;accuracy
recall<-(conf.test[2,2]/(conf.test[2,1]+conf.test[2,2]))*100;recall
precision<-(conf.test[2,2])/(conf.test[1,2]+conf.test[2,2])*100;precision

------------------------------------------------------------------------------

## 4) building SVM Model
install.packages("e1071")
library(e1071)
x = subset(Train, select = -target)
y  = as.factor(Train$target)

#tunning svm using grid search
tuned.svm<- tune.svm(target~.,data = data_new1,cost =
c(0.1,0.2,0.3,0.4,0.5),kernel="radial",
                gamma = c(0.0001,0.001,0.1,0.2,1), tunecontrol=tune.control(sampling = "fix"))

Churn_svm <- svm(x,y, method = "C-classification", kernel = "linear", cost = 0.1)
Churn_svm1 <- svm(x,y, method = "C-classification", kernel = "radial", cost = 0.5,gamma
= 0.2)

#linear kernel
svmpredict.train <- predict(Churn_svm,x)
conf.test <- table(y,svmpredict.train)
conf.test
###building confusion matrix for SVM
a = subset(Test, select = -target)
b1 = as.factor(Test$target)
```

```
svmpredict.test <- predict(Churn_svm,a)
conf.test <- table(b1,svmpredict.test)
# accuracy on test data
accuracy <- (sum(diag(conf.test))/sum(conf.test))*100;accuracy
#recall on test data
recall=(conf.test[2,2]/(conf.test[2,1]+conf.test[2,2]))*100;recall
# precision on test data
precision=(conf.test[2,2])/(conf.test[1,2]+conf.test[2,2])*100;precision

# radial kernel
svmpredict.train <- predict(Churn_svm1,x)
conf.test <- table(y,svmpredict.train)
conf.test
###building confusion matrix for SVM
a = subset(Test, select = -target)
b1 = as.factor(Test$target)

svmpredict.test <- predict(Churn_svm1,a)
conf.test <- table(b1,svmpredict.test)
# accuracy on test data
accuracy <- (sum(diag(conf.test))/sum(conf.test))*100;accuracy
#recall on test data
recall=(conf.test[2,2]/(conf.test[2,1]+conf.test[2,2]))*100;recall
# precision on test data
precision=(conf.test[2,2])/(conf.test[1,2]+conf.test[2,2])*100;precision
```

CONCLUSION

Further improvements can be done to this project by clustering the customer data and building multiple models on it and choosing the appropriate model for each cluster.

The proposed research has used Machine Learning techniques and R package to predict the results of churn customers on the benchmark Churn dataset available. It has evaluated, the number of churns by using the different classification techniques. The R tool has represented the large dataset churn in form of graphs which depicts the outcomes vividly and in a unique pattern visualization manner. The Churn Factor is used in many functions to depict the various areas or scenarios when the churn rate is high. The graphs are made taking churn factors as the deciding parameters. Graphs represent the different ways of observing the number of churners from the dataset. Once the root area is recognized the steps can be taken by Telecom Company to improve their services and retain their old customers from churning

REFERENCES

- Predicting Customer Churn in Mobile Telephony Industry Using probabilistic Classifiers in Data Mining‖ ,Clement Kirui1, Li Hong2, Wilson Cheruiyot3 and Hillary Kirui4.
- Comparing Clustering Techniques for Telecom Churn Management - Proceedings of the 5th WSEAS International Conference on Telecommunications and Informatics, Istanbul, Turkey, May 27-29, 2006.
- Using Data Mining for Mobile Communication Clustering and Characterization - A. Bascacov, C. Cernazanu and M. Marcu, Politehnica University of Timisoara/Computer and Software Engineering Department, Timisoara, Romania.
- Clustering Telecom Customers using Emergent Self Organizing Maps for Business Profitability - Kanchangauri Tulankar, Dr. Manali Kshirsagar, Rakhi Wajgi, Dept. of CSE, Yeshwantrao Chavan College of Engineering, Nagpur, Maharashtra, India.
- Customer Churn Prediction in Telecommunication Industries using Data Mining Technique - International Journal, Kiran Dahiya, KanikaTalwar, MRCE Faridabad, India.
- Han and Kamber. Data Mining: Concepts and Techniques. Second Morgan Kaufman Publisher, 2006.
- A Proposed Churn Prediction Model‖, Essam Shaaban, Yehia Helmy, Ayman Khedr, Mona Nasr,Department of Information Systems, MUST University, 6 October, Cairo, Egypt and Department of Information Systems, Helwan University, Cairo, Egypt.
- Data Mining Applications in Customer Churn Management‖, Sahand KhakAbi, Industrial Engineering Department, Iran University of Science and Technology, Tehran, Iran, Mohammad R. Gholamian, Morteza Namvar.
- -- (IJACSA) International Journal of Advanced Computer Science and Applications, ―Churn Prediction in Telecommunication Using Data Mining Technology‖, Rahul J. Jadhav of Bharati Vidyapeeth Deemed University,Pune, Yashwantrao Mohite of Institute of Management, Karad and Usharani T. Pawar2, Shivaji University Kolhapur.