PGPDSBA SEPT 19,GROUP 9

Business Report On Telecom Churn Prediction

Final Report

Prepared by

Mr.Jaydeep Bhaskar Ashtekar

Under the Guidance of

Mr.Suryanarayana Reddy

25/10/2020

Abstract: This report contains introduction to business problem, data problem, eda, business insights from EDA. It is basically a foundation and feature engineering to build further models to predict the churn. Important variables (continuous as well as categorical) are filtered using LR, p-value and VIF then for remaining variables EDA is done and some important insights are understood. This report also contains performances of different models for prediction of customers churns in telecom company. From their performances best models are suggested for identifying churns. Different combinations are tried for model tuning. This report also gives recommendations to the company to take proactive action for customers who are about to churn.

Content

1. INTRODUCTION OF THE BUSINESS PROBLEM

Nowadays telecom industries are trying to give best service in fewer prices. If we compare situations few years before and now, then one can notice that telecom industries have become service and customer centric. They are trying hard to provide best service under to their customers.

However, it also leads to the tough competition amongst various telecom service providers. Governing bodies of telecom industries has also given high liberty to customers and restrictions to operator. Under all these circumstances service provider has to compromise somewhere. Due to this, some criteria in certain demographical condition could not meet which causes different service issues like call drops, network issues etc. This leads customer to change the service provider known as **CUSTOMER CHURN.**

Service provider never wants that his customer should churn. If more customers churn out then it leads to loss to company which causes adverse effects on company. Thus, all telecom service providers are really very careful as far as customer satisfaction is concerned. Every service provider would like to add new customers without losing old customers.

Unfortunately, it's very difficult to find which customers are going to churn and which remains. But, we as data analysts can predict the customer churn up to certain accuracy and confidence.

1.1 Defining problem statement

The senior management in a telecom provider organization is worried about the rising customer attrition levels. Additionally, a recent independent survey has suggested that the industry as a whole will face increasing churn rates and decreasing ARPU (average revenue per unit).

The effort to retain customers so far has been very reactive. Only when the customer calls to close their account is when the company takes action. That has not proved to be a great strategy so far. The management team is keen to take more proactive measures on this front. You as a data scientist are tasked to derive insights, predict the potential behavior of customers, and then recommend steps to reduce churn.

1.2 Need of study/Project

- Primary focus in this project we need to predict which person is going to close the account. So that telecom service provider representative will call that customer and take preventive actions.

- Along with this, service provider want to know that why customer want to close account, why customer is unhappy by service etc. technically speaking, Which factors from data set provided causes the customer to churn?

- Whether customers are unhappy with “cost and billing” and “network and service quality”?

- Customer behavior directly impacts the revenue of company. If customers are retained then revenue will be generated. We will also come to know that w If more customers stop or change service then bad publicity will also cause the satisfactory customers to churn out.

1.3 Understanding business/social opportunity

After overall analysis, we can get an idea of business strategy, advertisement strategy, marketing strategy etc. The objective of this study is to give answers to following question:

1. What are the top five factors driving the likelihood of churn?

2. Validation of independent survey's findings for the telecom industry.

a) Whether **“cost and billing”** and **“network and service quality”** are important factors influencing churn behavior.

b) Are data usage connectivity issues turning out to be costly?

In other words, is it leading to churn?

3. Would you recommend rate plan migration as a proactive retention strategy?

4. What would be your recommendation on how to use this churn model for prioritization of customers for a proactive retention campaign in the future?

5. What would be the target segments for proactive retention campaigns?

Falling ARPU forecast is also a concern and therefore, the service provider would like to save their high revenue customers besides managing churn. Given a budget constraint of a contact list of 20% of the subscriber pool, which subscribers should be prioritized if “revenue saves” is also a priority besides controlling churn. In other words, controlling churn is the primary objective and revenue saves is the secondary objective.

1.4 Who will be benefitted by this study?

After this study, **company** will get a certain model that predicts whether customer will churn or not. Secondly, company will get an idea and opportunity for improvements in fields where they are lacking.

Customers are also being benefitted by getting a service they are paying for. This leads to customer satisfaction.

If published, this study helps to **students/professionals** to understand churn behaviors of customers.

Government/governing body of telecommunications will also get benefitted by this study to decide some guidelines for users and company both. Obviously this study has to be done for other companies too.

2. EDA AND BUSINESS IMPLICATIONS

2.1 Visual inspection of data

a. Number of observations and number of features

Visual Data inspection includes some of the following things:

1. Number of observations/rows: **26518**

2. Number of attributes/variables/features/columns: **81**

b. Descriptive detail

1. Categorical Variables: Data set has 21 categorical variables and description is as below

	crclscod	asl_flag	prizm_social_one	area	refurb_new	hnd_webcap	marital	ethnic	dwlltype	dwllsize	mailordr	occu1
count	26518	26518	24637	26513	26518	24136	26050	26050	18137	16428	9508	7030
unique	49	2	5	19	2	3	5	17	2	15	1	21
top	AA	N	S	NEW YORK	N	WCMB	U	N	S	A	B	1
freq	9602	22535	8475	3055	22863	20660	9821	8912	12968	12521	9508	2723

Data set has 21 categorical variables and description is as above:

- **prizm_social_one, area, hnd_webcap, marital, ethnic, car_buy** has **missing** values.
  Eg. asl_flag has **26518** observations (**count**), having two unique subcategories. Out of which **N** has observed maximum time (**top**), and its frequency (**freq**) observation is **22535**. (Same can be read for others too).
- Maximum customers are from New York City Area.

2. Continuous Variables: 61 continuous variables are present. **Mean shows location of centre of data and standard deviation (std) shows spread of data.**

Minimum, 25th, 50th, 75th and maximum values show the spread from mean.

	totmrc_Mean	mou_Range	change_mou	income	eqpdays	custcare_Mean	rev_Mean	comp_vce_Mean
count	26518	26518	26518	26518	26518	26518	26518	26518
mean	47.18425	378.4551	-10.1776	5.829889	376.4501	1.903424	59.3342	112.5869
std	24.23797	428.4653	254.4267	1.894268	251.9672	5.488562	44.60341	121.0656

min	-26.915	0	-2785	1	-5	0	-2.52	0
25%	30	115	-82.25	5	202	0	33.78875	30.66667
50%	44.99	245	-4.75	6	326	0	48.8	78
75%	59.99	484	64	7	510	1.666667	71.855	155
max	399.99	6233	3046.75	9	1812	365.6667	926.0775	1812.667

1. **mou_Range** has mean of 378.4551 with standard deviation of 428.4653.It has minimum value 0. Maximum value is 6233.So one can say data has long tail on right tail so data is left skewed and has outliers.

	age1	age2	models	hnd_price	drop_vce_Mean	adjmou	churn
count	26518	26518	26518	26518	26518	26518	26518
mean	31.24836	20.69014	1.569462	104.9029	6.083666	7707.227	0.239988
std	21.94682	23.969	0.910224	60.03146	9.095215	9060.873	0.427084
min	0	0	1	9.989998	0	0	0
25%	0	0	1	59.98999	0.666667	2454	0
50%	36	0	1	99.98999	3	5105	0
75%	48	42	2	149.99	7.666667	9739	0
max	94	99	15	499.99	195.3333	174383.4	1

Eg: age1 and age2 are seems to be normally distributed without much outlier.

- mou_Range has minimum zero which means these have not used service.
- drop_vce_Mean is zero means they have observed call drop very less frequently.

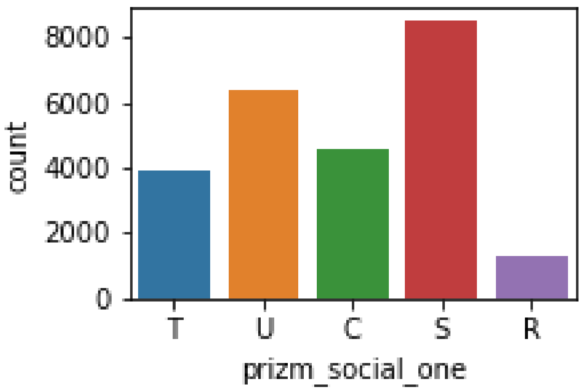
2.2 Understanding of attributes: Attributes related to**:**

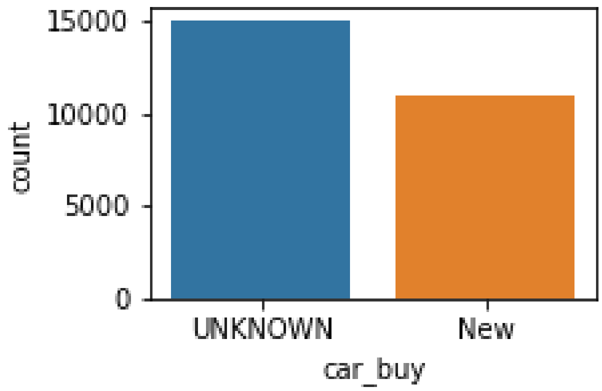
cost and billing	Network and service quality
totmrc_Mean,rev_Range,adjqty,ovrrev_Mean,rev_Mean ,datovr_Mean, datovr_Range, adjmou, totrev, adjrev, avgrev.	mou_Mean,mou_Range,change_mou,drop_blk_Mean,drop_vce_Range,owylis_vce_Range,mou_opkv_Range,months,totcalls,eqpdays,custcare_Mean,callwait_Mean,iwylis_vce_Mean,callwait_Range,ccrndmou_Range,ovrmou_Mean,comp_vce_Mean,plcd_vce_Mean,avg3mou,avgmou,avg3qty,avgqty,avg6mou,avg6qty,opk_dat_Mean,retdays,roam_Mean,recv_sms_Mean,blck_dat_Mean,mou_pead_Mean,da_Mean,da_Range,drop_dat_Mean,drop_vce_Mean,comp_dat_Mean,plcd_dat_Mean

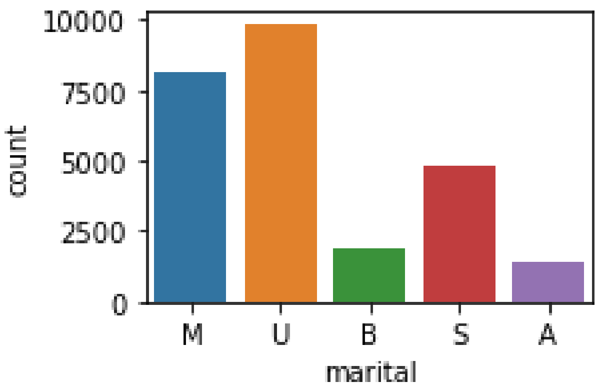
2.3 Univariate Analysis

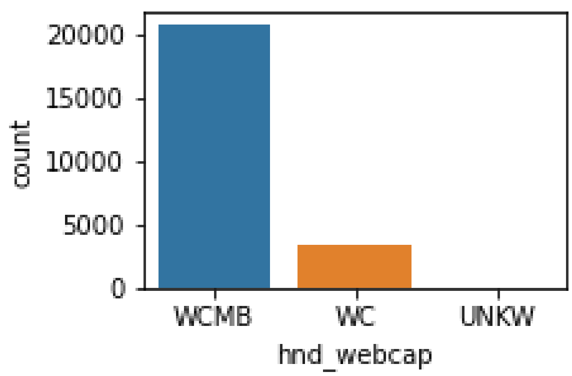
It basically includes count plot for categorical variables and kde plot for continuous variables.

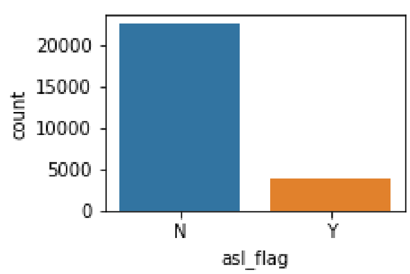
A. Count plots:

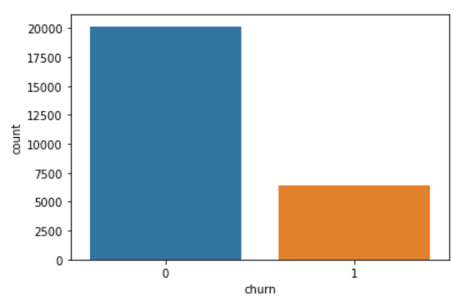






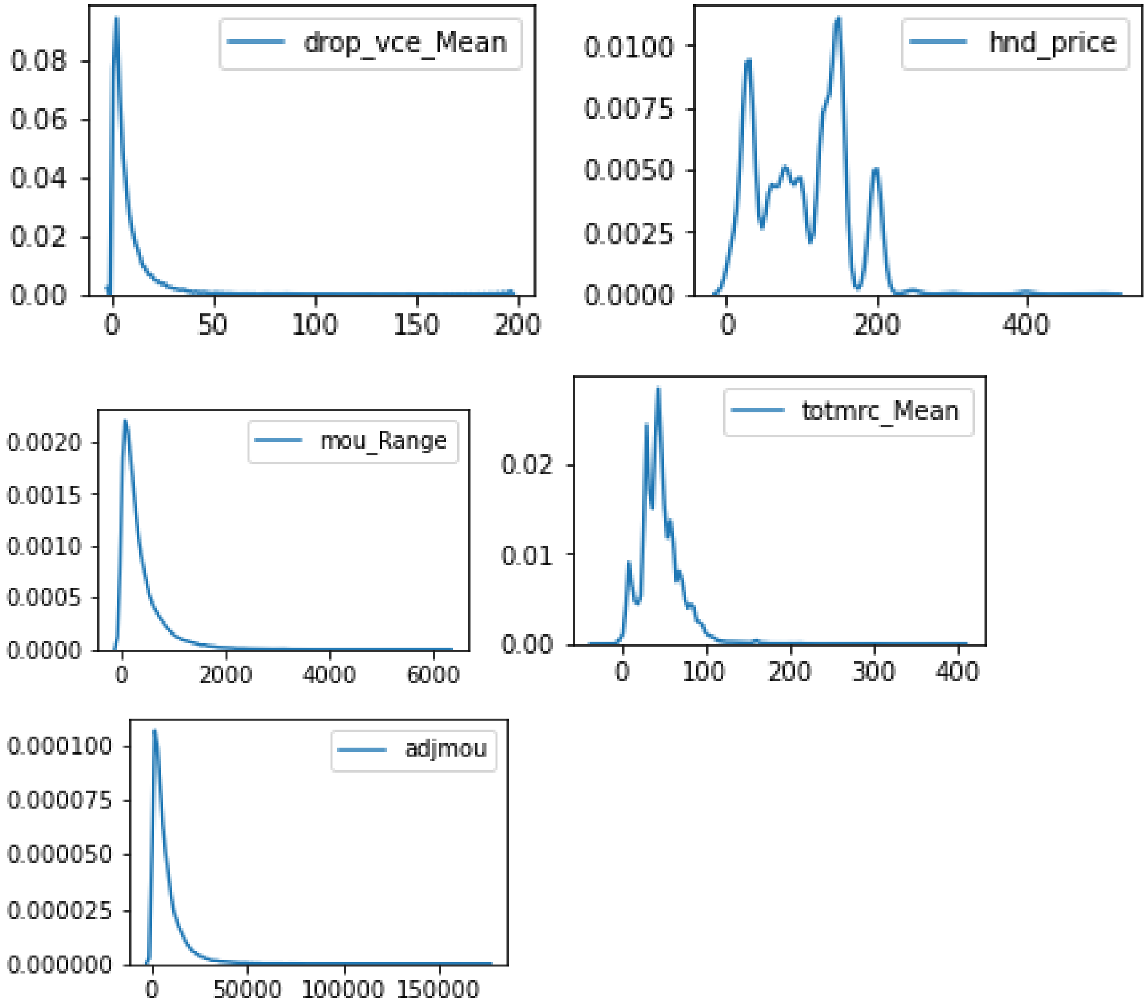






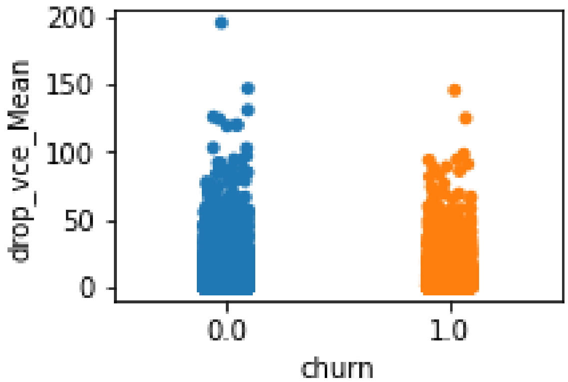
Above count plots are self explanatory.

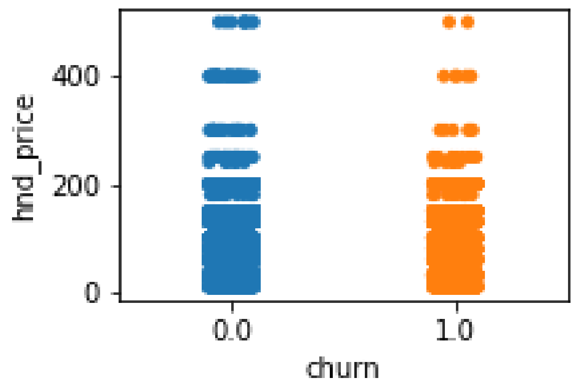
B.kde plots: It shows distribution of continuous variables.

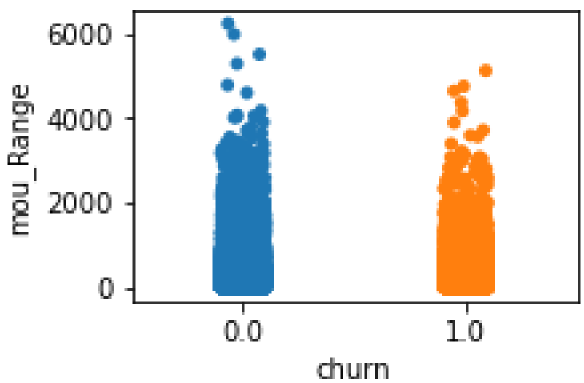


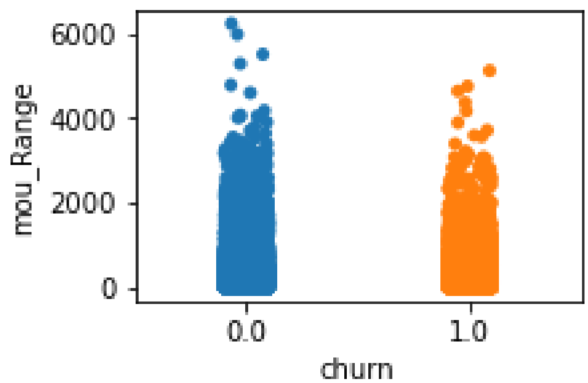
C. Strip plot: It is a plot of churn (As type Categorical) and other continuous variables.

- Income is multimodal with no outliers.
- drop_vce_mean has single mode and it is left skewed but **hnd_price** is left skewed but with multimodal.
- eqpdays is approximately normally distributed
- Custcare mean has single mode with large value of outliers.









Strip plot gives a distribution of continuous variables in terms of categorical variable.In above plots if we consider.

2.4 Bivariate analysis:

A.Pairplot:Bivariate analysis is relation between one continuous variable with other continuous variable. Correlation of churn with other variables:

It can be observed that **eqpdays**, **income, mou_range** are positively correlated with very less correlation factors and **other are negatively correlated**. Same can be observed by the slope of line in above graphs.

```
eqpdays          0.10
income           0.02
mou_Range        0.01
rev_Mean        -0.00
drop_vce_Mean   -0.00
adjmou          -0.01
change_mou      -0.03
custcare Mean   -0.03
```

```
custcare_Mean    -0.03
age2             -0.03
comp_vce_Mean    -0.04
age1             -0.04
models           -0.04
totmrc_Mean      -0.05
hnd_price        -0.08
```

2.5 Data Balance

So data doesn't seem to be unbalanced because ratio is 76:24.Data is unbalanced when ratio is crudely said to be less than 85:15.But still we can apply synthetic techniques to balance data and to improve model performance.

```
0.0    76%
1.0    24%...........target
```

2.6 Insights using Clustering

Clustering is done on dataset using K Means clustering algorithm. Irrespective of inertia value in clustering, 5 clusters are made which telecom company want.

The analysis of cluster is as follows:

	rev_Mean	count	mean_rev
Clus_kmeans			
2	348816.85	3847	90.672432
0	347145.56	7352	47.21784
1	306638.32	3541	86.596532
4	276894.89	6202	44.646065
3	201938.77	5576	36.215705

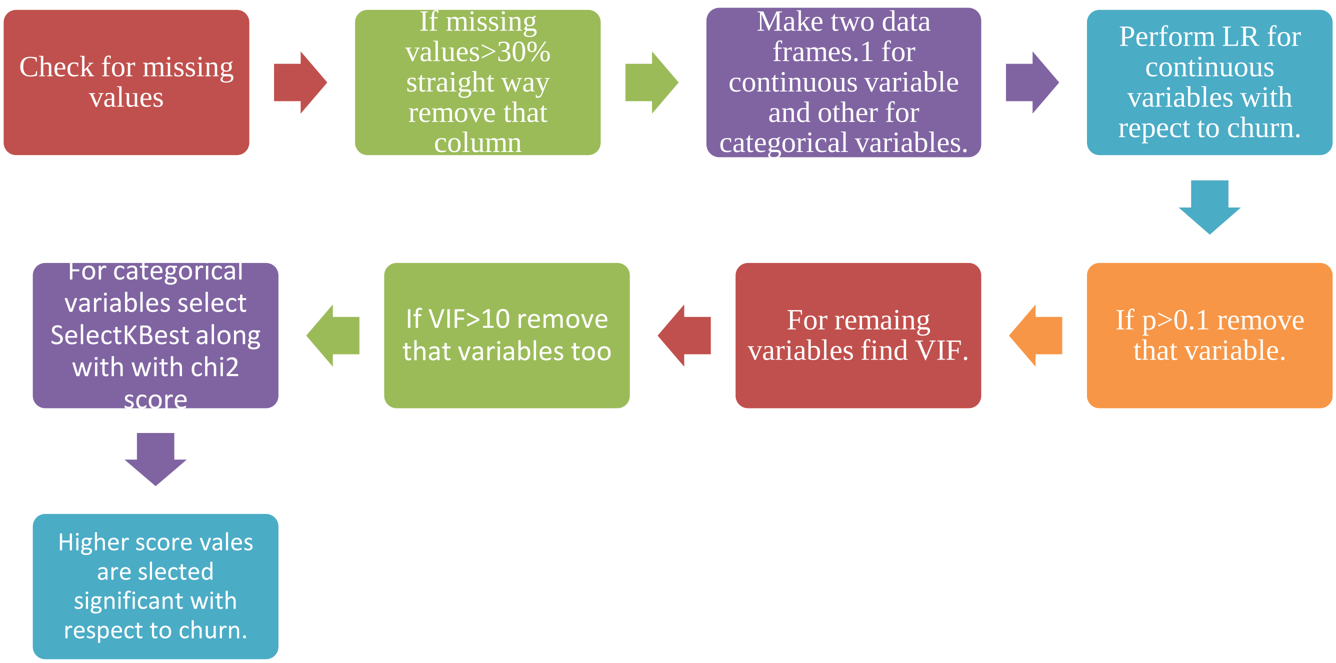
Cluster 2 will give maximum revenue which has 3847 customers

3. DATA CLEANING AND PRE-PROCESSING

3.1 Approach

'Customer ID' and 'csa' are removed from data set. csa is removed due to 690 levels.

Method:



After following above method 81 variables are reduced to the 22 variables.

totmrc_Mean	custcare_Mean	models	hnd_webcap
mou_Range	rev_Mean	hnd_price	marital
change_mou	comp_vce_Mean	drop_vce_Mean	ethnic
income	age1	adjmou	churn
eqpdays	age2	crclscod	area
		asl_flag	refurb_new

3.2 Missing Value treatment

Percentage of missing values: Missing values for continuous variables are treated with **median** because of presence of the outliers.

- Missing values for categorical values are treated with **mode.**

3.3 Outliers Treatment

Percentage of outliers present in the dataset is as follows:

rev_Mean	5.939362
comp_vce_Mean	4.894788
age1	0.000000
age2	0.000000
models	4.182065
hnd_price	0.339392
drop_vce_Mean	7.549589
adjmou	6.256128

totmrc_Mean	2.043895
mou_Range	6.708651
change_mou	14.043291
income	3.952033
eqpdays	2.749076
custcare_Mean	13.455012

% of the outliers is less so outliers are treated by capping technique here.

3.4 Variable Transformations

Whenever variables are in different scale it is recommended to scale the data. Some models have capability to adjust themselves without transformation. Here **standard scalar** is used to scale data.

Following insights can be summarized:

- Many continuous variables are correlated to each other so some of them must be removed to reduce multicolinearity. After removing insignificant variables using logistic regression 81 variables were reduced to 21.
- Missing values were appropriately imputed by median and mode.
- Outliers were present in all variables except age1 and age2.So outliers are treated by capping.

4. MODEL BUILDING

In continuation with previous report, 21 important variables are selected to predict the churn. Those are as follows:

Sr.No.	Variable	Type
1	totmrc_Mean	Continuous Variables
2	mou_Range	
3	change_mou	
4	income	
5	eqpdays	
6	custcare_Mean	
7	rev_Mean	
8	comp_vce_Mean	
9	age1	
10	age2	
11	models	
12	hnd_price	
13	drop_vce_Mean	
14	adjmou	
15	crclscod	Categorical Variables
16	asl_flag	
17	area	
18	refurb_new	
19	hnd_webcap	
20	marital	
21	ethnic	
22	churn	Integer(Dependent variable)

4.1 Build various models

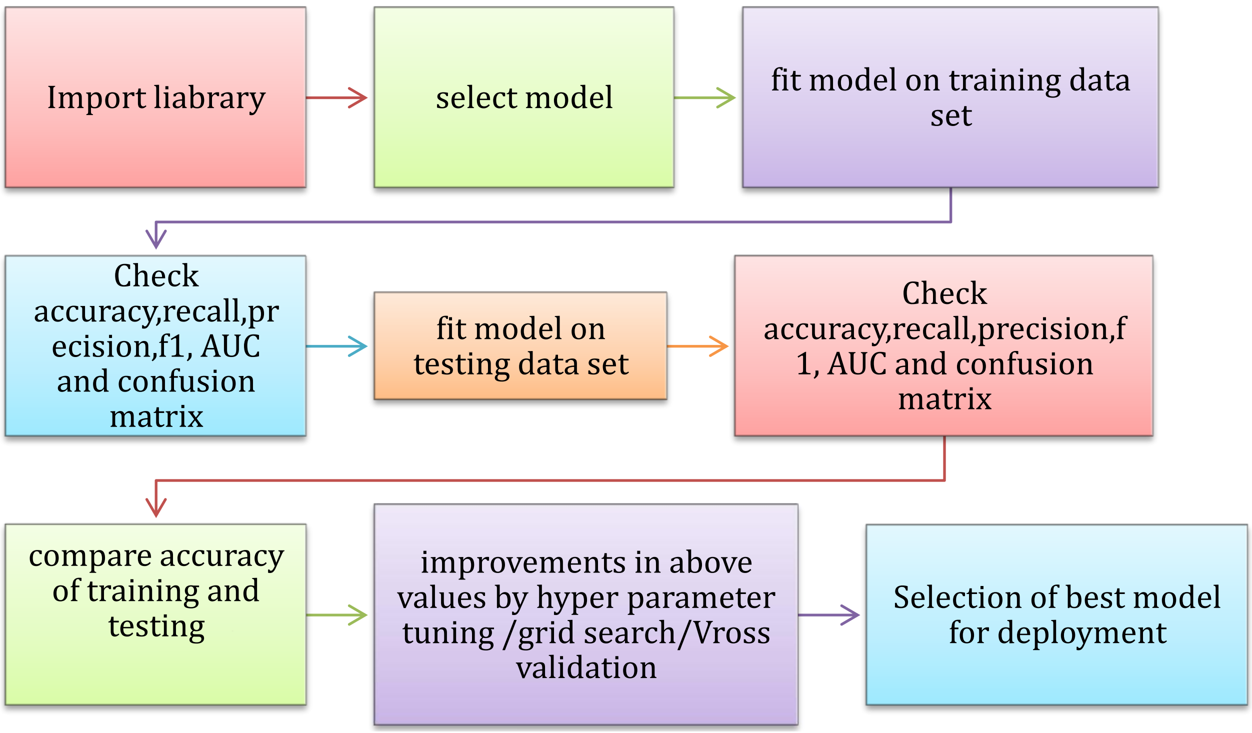
1. Divide data in training and testing randomly in 70:30 proportion as follows:

```
from sklearn.linear_model import LogisticRegression
import scipy.stats as stats
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.metrics import confusion_matrix
from sklearn.preprocessing import scale
from sklearn import preprocessing
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
from sklearn import metrics
```

```
X=df_final.drop('churn',axis=1)
y=df_final['churn']
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3 , random_state=1)
```

Using above variables churn is predicted. The flow of model is as follows:



1. Dummy Classifier:

It is used only as a simple baseline for the other classifiers i.e. any other classifier is expected to perform better on the given dataset.

```
#Dummy Classifier
from sklearn.dummy import DummyClassifier
clf = DummyClassifier(strategy= 'most_frequent').fit(X_train,y_train)
y_pred = clf.predict(X_test)

#Distribution of y test
print('y actual : \n' +  str(y_test.value_counts()))

#Distribution of y predicted
print('y predicted : \n' + str(pd.Series(y_pred).value_counts()))

y actual :
0.0    6015
1.0    1941
Name: churn, dtype: int64

<IPython.core.display.Javascript object>

y predicted :
0.0    7956
dtype: int64
```

```
# Model Evaluation metrics
from sklearn.metrics import accuracy_score,recall_score,precision_score,f1_score
print('Accuracy Score : ' + str(accuracy_score(y_test,y_pred)))
print('Precision Score : ' + str(precision_score(y_test,y_pred)))
print('Recall Score : ' + str(recall_score(y_test,y_pred)))
print('F1 Score : ' + str(f1_score(y_test,y_pred)))

#Dummy Classifier Confusion matrix
from sklearn.metrics import confusion_matrix
print('Confusion Matrix : \n' + str(confusion_matrix(y_test,y_pred)))
```

```
Accuracy Score : 0.7560331825037707
Precision Score : 0.0
Recall Score : 0.0
F1 Score : 0.0
Confusion Matrix :
[[6015    0]
 [1941    0]]
```

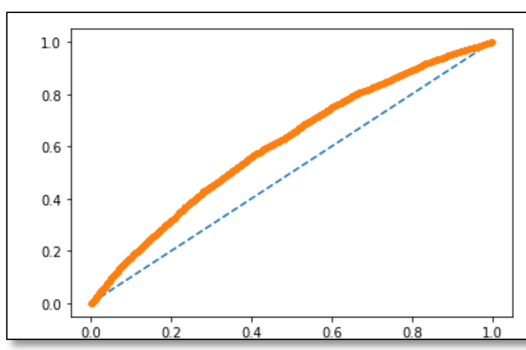
So it is expected that all other classifier should give accuracy more that 0.756 and at least should predict some "1"

2. Logistic Regression

```
model = LogisticRegression(max_iter=1000,)
model.fit(X_train, y_train)

LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=1000,
                   multi_class='auto', n_jobs=None, penalty='l2',
                   random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
                   warm_start=False)

y_train_predict = model.predict(X_train)
model_score = model.score(X_train, y_train)

print("Accuracy \n\n",model_score)
print("\nConfusion matrix \n\n",metrics.confusion_matrix(y_train, y_train_predict))
print("\nclassification report \n",metrics.classification_report(y_train, y_train_predict,digits=3))
```

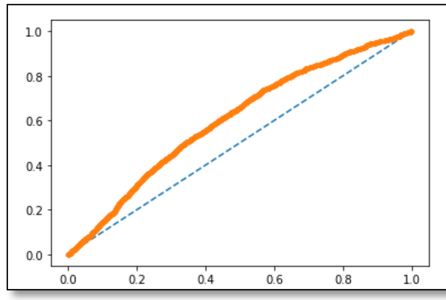
LR on train data (AUC=0.608)

```
Accuracy

 0.7609093847645728

Confusion matrix

 [[14116    23]
 [ 4415     8]]

classification report
              precision    recall  f1-score   support

         0.0      0.762     0.998     0.864     14139
         1.0      0.258     0.002     0.004      4423

    accuracy                          0.761     18562
   macro avg      0.510     0.500     0.434     18562
weighted avg      0.642     0.761     0.659     18562
```



LR on test data (AUC=0.606)

```
Accuracy

 0.7551533433886375

Confusion matrix

 [[6005    10]
 [1938     3]]

classification report
               precision    recall  f1-score   support

         0.0       0.756     0.998     0.860      6015
         1.0       0.231     0.002     0.003      1941

    accuracy                           0.755      7956
   macro avg       0.493     0.500     0.432      7956
weighted avg       0.628     0.755     0.651      7956
```



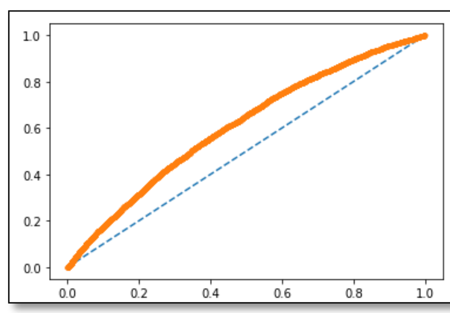
2. Linear Discriminate Analysis(LDA)

```
model = LinearDiscriminantAnalysis()
model.fit(X_train, y_train)
```

```
LinearDiscriminantAnalysis(n_components=None, priors=None, shrinkage=None,
                           solver='svd', store_covariance=False, tol=0.0001)
```

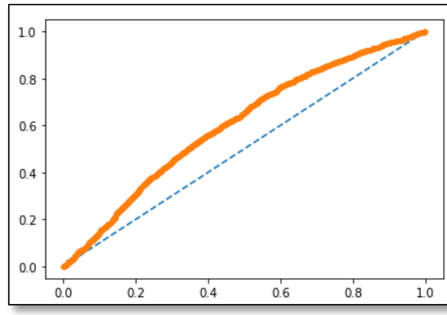
LDA(Train)[AUC:0.608]

```
Accuracy

 0.7608555112595625

Confusion matrix

 [[14114    25]
 [ 4414     9]]

classification report
               precision    recall  f1-score   support

         0.0       0.762     0.998     0.864     14139
         1.0       0.265     0.002     0.004      4423

    accuracy                           0.761     18562
   macro avg       0.513     0.500     0.434     18562
weighted avg       0.643     0.761     0.659     18562
```



LDA (Test)[AUC:0.606]

```
Accuracy

 0.7547762694821518

Confusion matrix

 [[6002   13]
 [1938    3]]

classification report
              precision    recall  f1-score   support

         0.0      0.756     0.998     0.860      6015
         1.0      0.188     0.002     0.003      1941

    accuracy                          0.755      7956
   macro avg      0.472     0.500     0.432      7956
weighted avg      0.617     0.755     0.651      7956
```

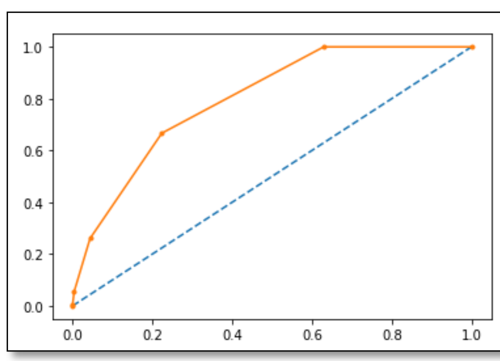


3. KNN Model:

```
from sklearn.neighbors import KNeighborsClassifier

model=KNeighborsClassifier()
model.fit(X_train,y_train)

KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=None, n_neighbors=5, p=2,
                     weights='uniform')
```

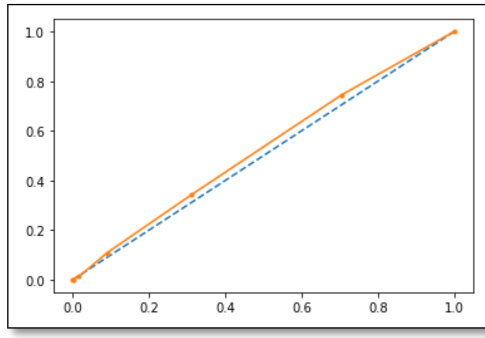
KNN (Train)[AUC:0.798]

```
Accuracy

 0.7903781920051719

Confusion matrix

 [[13520   619]
 [ 3272  1151]]

classification report
              precision    recall  f1-score   support

         0.0      0.805     0.956     0.874     14139
         1.0      0.650     0.260     0.372      4423

    accuracy                          0.790     18562
   macro avg      0.728     0.608     0.623     18562
weighted avg      0.768     0.790     0.754     18562
```



KNN (Test)[AUC:0.526]

```
Accuracy

 0.7146807440925088

Confusion matrix

 [[5477  538]
 [1732  209]]

classification report
               precision    recall  f1-score   support

         0.0       0.760     0.911     0.828      6015
         1.0       0.280     0.108     0.156      1941

    accuracy                           0.715      7956
   macro avg       0.520     0.509     0.492      7956
weighted avg       0.643     0.715     0.664      7956
```



4.2 Testing of predictive model against the test set using various appropriate performance metrics

All such models are made of which summary is as follows. Summary contains only test data:

	1			0				
	Precision	Recall	f1	Precision	Recall	f1	Acc	AUC
logistic regression	0.231	0.002	0.003	0.756	0.998	0.86	0.755	0.606
LDA	0.188	0.002	0.003	0.756	0.998	0.86	0.755	0.606
KNN	0.28	0.108	0.156	0.76	0.911	0.828	0.7146	0.526
Naïve_Bayes	0.329	0.197	0.246	0.771	0.871	0.818	0.7062	0.591
SVM	0	0	0	0.756	1	0.861	0.756	0.537
CART	0.401	0.057	0.099	0.762	0.973	0.854	0.749	0.624
RF	0.528	0.035	0.065	0.761	0.99	0.86	0.7569	0.626
ANN	0.313	0.081	0.129	0.761	0.942	0.842	0.7322	0.586

4.3 Interpretation of Models:

Class "1" is important class of the churn prediction because it actually gives the churn. So aim of our model is to precisely predict class 1.

Following observation can be drawn from above models:

1. Accuracy wise Random forest is the best model. But accuracy is not the sufficient criteria.
2. Maximum precision is obtained by RF model.
3. Maximum Recall is obtained by Naïve Bayes model.

4.4. Model Tuning

To improve the performance of model let's try model tuning by various methods and check whether performance improves or not.

Ensemble modeling and model tuning

1. RF hyper parameter tuning:

```
from sklearn.ensemble import RandomForestClassifier

from sklearn.model_selection import GridSearchCV

param_grid = {
    'max_depth': [7,9,11],
    'max_features': [5,6,7],
    'min_samples_leaf': [5,10,15],
    'min_samples_split': [50,75,100],
    'oob_score':[True],
    'n_estimators': [100]
}

model = RandomForestClassifier()

grid_search = GridSearchCV(estimator = model, param_grid = param_grid, cv = 3)
grid_search.fit(X_train, y_train)
grid_search.best_params_
best_grid = grid_search.best_estimator_
best_grid
model=best_grid
model.fit(X_train, y_train)
```

```
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=9, max_features=7,
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=5, min_samples_split=50,
                       min_weight_fraction_leaf=0.0, n_estimators=100,
                       n_jobs=None, oob_score=True, random_state=None,
                       verbose=0, warm_start=False)
```

```
Accuracy

 0.756913021618904

Confusion matrix

 [[5991   24]
 [1910   31]]

classification report
               precision    recall  f1-score   support

         0.0      0.758     0.996     0.861      6015
         1.0      0.564     0.016     0.031      1941

    accuracy                          0.757      7956
   macro avg      0.661     0.506     0.446      7956
weighted avg      0.711     0.757     0.659      7956

AUC: 0.645
```

2. Tuned cart

```
param_grid = {
    'criterion':['gini','entropy'],
    'splitter': ["best"],
    'max_depth':[7,8,8,10],
    'min_samples_split':[10,15,20],
    'min_samples_leaf':[8,9,10],
    'random_state':[10],
    'max_leaf_nodes':[1,2,3,4,5,6],
    'min_impurity_split':[1e-7]
}
#criterion = 'gini', max_depth = 7,min_samples_leaf=10,min_samples_split=30
model = tree.DecisionTreeClassifier( )
grid_search = GridSearchCV(estimator = model, param_grid = param_grid, cv = 3)
grid_search.fit(X_train, y_train)
grid_search.best_params_
best_grid = grid_search.best_estimator_
best_grid
model=best_grid
model.fit(X_train, y_train)
```

```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
                       max_depth=7, max_features=None, max_leaf_nodes=4,
                       min_impurity_decrease=0.0, min_impurity_split=1e-07,
                       min_samples_leaf=8, min_samples_split=10,
                       min_weight_fraction_leaf=0.0, presort='deprecated',
                       random_state=10, splitter='best')
```

```
**************************FOR TESTING DATASET**********
Accuracy

 0.7560331825037707

Confusion matrix

 [[5971   44]
 [1897   44]]

classification report
              precision    recall  f1-score   support

         0.0      0.759     0.993     0.860      6015
         1.0      0.500     0.023     0.043      1941

    accuracy                          0.756      7956
   macro avg      0.629     0.508     0.452      7956
weighted avg      0.696     0.756     0.661      7956

AUC: 0.589
```

3. ANN Tuned

```
from sklearn.neural_network import MLPClassifier
param_grid = {
    'hidden_layer_sizes': [100,50,200],
    'max_iter': [7000,5000,2500],
    'solver': ['adam','sgd'],
    'tol': [0.01]
}

model = MLPClassifier()

grid_search = GridSearchCV(estimator = model, param_grid = param_grid, cv = 10)

grid_search.fit(X_train, y_train)
grid_search.best_params_
best_grid = grid_search.best_estimator_
best_grid
model=best_grid
model.fit(X_train, y_train)
```

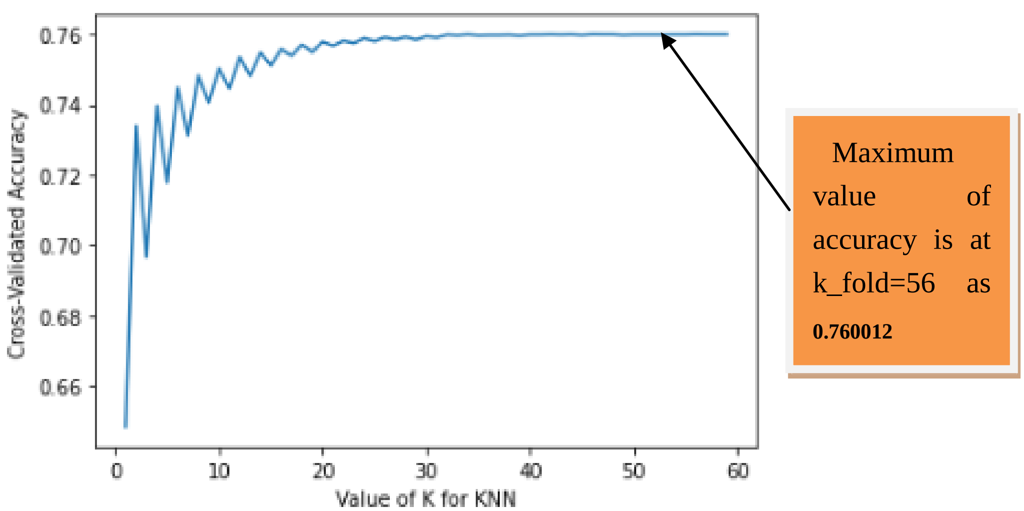
```
MLPClassifier(activation='relu', alpha=0.0001, batch_size='auto', beta_1=0.9,
              beta_2=0.999, early_stopping=False, epsilon=1e-08,
              hidden_layer_sizes=200, learning_rate='constant',
              learning_rate_init=0.001, max_fun=15000, max_iter=2500,
              momentum=0.9, n_iter_no_change=10, nesterovs_momentum=True,
              power_t=0.5, random_state=None, shuffle=True, solver='sgd',
              tol=0.01, validation_fraction=0.1, verbose=False,
              warm_start=False)
```

```
*************************FOR TESTING DATASET***********
Accuracy

 0.7515082956259427

Confusion matrix

 [[5963   52]
 [1925   16]]

classification report
              precision    recall  f1-score   support

         0.0      0.756     0.991     0.858      6015
         1.0      0.235     0.008     0.016      1941

    accuracy                          0.752      7956
   macro avg      0.496     0.500     0.437      7956
weighted avg      0.629     0.752     0.652      7956

AUC: 0.578
```

4. KNN with varying K-folds

```
k_range = range(1, 60)
k_scores = []
for k in k_range:
    model = KNeighborsClassifier(n_neighbors=k)
    scores = cross_val_score(model, X, y, cv=5, scoring='accuracy')
    k_scores.append(scores.mean())
# plot to see clearly
plt.plot(k_range, k_scores)
plt.xlabel('Value of K for KNN')
plt.ylabel('Cross-Validated Accuracy')
plt.show()
```

```
<IPython.core.display.Javascript object>

<IPython.core.display.Javascript object>

<IPython.core.display.Javascript object>

<IPython.core.display.Javascript object>
```



k_range	k_scores
47	0.759974
48	0.759974
49	0.759786
50	0.759899
51	0.759899
52	0.759899
53	0.759899
54	0.759937
55	0.759937
56	0.760012
57	0.759974
58	0.759974
59	0.759974

Here maximum accuracy of the KNN is aobserved to be 0.76 at k_fold=56.So considering 56 as value of the k_folds we will find model performance.

```
*************************FOR TESTING DATASET*********
Accuracy

 0.7559074912016088

Confusion matrix

 [[6014    1]
 [1941    0]]

classification report
               precision    recall  f1-score   support

         0.0       0.756     1.000     0.861      6015
         1.0       0.000     0.000     0.000      1941

    accuracy                           0.756      7956
   macro avg       0.378     0.500     0.430      7956
weighted avg       0.572     0.756     0.651      7956

AUC: 0.556
```

5. MODEL VALIDATION

5.1 Performance comparison of simple and tuned models

	1			0				
	Precision	Recall	f1	Precision	Recall	f1	Acc	AUC
ANN	0.313	0.081	0.129	0.761	0.942	0.842	0.7322	0.586
ANN tuned	0.235	0.008	0.016	0.756	0.991	0.858	0.751	0.578
CART	0.401	0.057	0.099	0.762	0.973	0.854	0.749	0.624
CART Tuned	0.5	0.023	0.043	0.759	0.993	0.86	0.756	0.589
KNN	0.28	0.108	0.156	0.76	0.911	0.828	0.7146	0.526
KNN with Kfolds	0	0	0	0.756	1	0.861	0.7559	0.556
KNN_tuning	0	0	0	0.756	1	0.861	0.7559	0.573
RF	0.528	0.035	0.065	0.761	0.99	0.86	0.7569	0.626
RF tuned	0.564	0.016	0.031	0.758	0.996	0.861	0.757	0.645

5.2 Inferences of all models and best three models

1. ANN: After tuning accuracy has increased from 0.731 to 0.751,recall 0 increased from 0.942 to 0.991 but precision, recall and f1 decreased after tuning.
2. CART: precision of 1 has increased from 0.4 to 0.5 accuracy has also increased.
3. KNN: After tuning only accuracy has increased but other parameters decreased.

5.3 Interpretation of the most optimum model and its implication on the business

1. Criteria for selection of the model:

Model should be capable of predicting class “1”, so a criterion is **precision.**
Precision =True class 1/Predicted class 1= True class 1 / (True class 1+ False class 1).If customer is of class 0 but if predicted churn will not affect much because he is not going to churn. But if customer is about to churn and not able to predict then there will be loss to company.

			1			0		
	Precision	Recall	f1	Precision	Recall	f1	Acc	AUC
ANN	0.313	0.081	0.129	0.761	0.942	0.842	0.7322	0.586
ANN tuned	0.235	0.008	0.016	0.756	0.991	0.858	0.751	0.578
CART	0.401	0.057	0.099	0.762	0.973	0.854	0.749	0.624
CART Tuned	0.5	0.023	0.043	0.759	0.993	0.86	0.756	0.589
KNN	0.28	0.108	0.156	0.76	0.911	0.828	0.7146	0.526
KNN with Kfolds	0	0	0	0.756	1	0.861	0.7559	0.556
KNN_tuning	0	0	0	0.756	1	0.861	0.7559	0.573

LDA	0.188	0.002	0.003	0.756	0.998	0.86	0.755	0.606
logistic regression	0.231	0.002	0.003	0.756	0.998	0.86	0.755	0.606
Naïve_Bayes	0.329	0.197	0.246	0.771	0.871	0.818	0.7062	0.591
RF	0.528	0.035	0.065	0.761	0.99	0.86	0.7569	0.626
RF tuned	0.564	0.016	0.031	0.758	0.996	0.861	0.757	0.645
SVM	0	0	0	0.756	1	0.861	0.756	0.537

So best model can be considered as:

1. CART: Because it has good precision and good accuracy
2. RF: Because it has second highest precision and highest accuracy.
3. Naïve Bayes: This model has less precision (0.329) as compared to others but it has captured recall, f1 very well than other models.

6. FINAL INTERPRETATION / RECOMMENDATION

6.1 Important factors which affect business

So, it can be observed that following factors actually impacting the churn. Those are divided in two categories **"service quality"** and **"cost and billing".**

1	mou_Range	Range of number of minutes of use	service quality
2	change_mou	Percentage change in monthly minutes of use vs previous three month average	service quality
3	rev_Mean	Mean monthly revenue (charge amount	cost and billing
4	adjmou	Billing adjusted total minutes of use over the life of the customer	cost and billing
5	totmrc_Mean	Mean total monthly recurring charge	Cost and billing

So these five factors on which company have to focus.

1.Range of number of minutes of use :Good service network and data quality should be provided so that number of minutes usage will increase and which leads to increase in revenue

2. Percentage change in monthly minutes of use vs previous three month average

Case I: Percentage change in monthly minutes<previous three month average

It shows that customer usage is reducing. It concludes that customer is relying on some other service provider.

Case I: Percentage change in monthly minutes>=previous three month average

It shows that customer usage is increasing or same. It concludes that customer is still firm on the service provider.

3. Mean monthly revenue

It is directly related to the costing. If it is reducing means customer is not using the service like before or rate tariff plan has increased.

4. Billing adjusted total minutes of use over the life of the customer:

It should be more and shows the customer loyalty to service provider.

5. Mean total monthly recurring charge

Recurring charges should be less.

6.2 Customer segmentation:

Clustering is done on dataset using K Means clustering algorithm. Irrespective of inertia value in clustering, 5 clusters are made which telecom company want.

The analysis of cluster is as follows:

	rev_Mean	count	mean_rev
Clus_kmeans			
2	348816.85	3847	90.672432
0	347145.56	7352	47.21784
1	306638.32	3541	86.596532
4	276894.89	6202	44.646065
3	201938.77	5576	36.215705

Cluster 1 and Cluster 2 will give maximum revenue which has 3847 customers.

So using this **7388** customer should be found and some discount should be provided so that they will not churn. For other clusters some promotional activity is suggested.