BA - Telecom churn case study

Nithin Vijayabaskaran
Jitender Pal Singh
Hiral Shah

Problem statement:-

- To reduce customer churn, telecom companies need to predict which customers are at high risk of churn. In this project, we will analyse customer-level data of a leading telecom firm, build predictive models to identify customers at high risk of churn and identify the main indicators of churn.
- Retaining high profitable customers is the main business goal here.
- Steps:- Reading, understanding and visualising the data Preparing the data for modelling Building the model Evaluate the model

Steps:-

1. Reading, understanding and visualising the data
2. Preparing the data for modelling
3. Building the model
4. Evaluate the model

Analysis approach

- Telecommunications industry experiences an average of 15 - 25% annual churn rate. Given the fact that it costs 5-10 times more to acquire a new customer than to retain an existing one, customer retention has become even more important than customer acquisition.
- Here we are given with 4 months of data related to customer usage. In this case study, we analyse customer-level data of a leading telecom firm, build predictive models to identify customers at high risk of churn and identify the main indicators of churn.
- Churn is predicted using two approaches. Usage based churn and Revenue based churn. Usage based churn:
- Customers who have zero usage, either incoming or outgoing - in terms of calls, internet etc. over a period of time.
- This case study only considers usage-based churn.
- In the Indian and the southeast Asian market, approximately 80% of revenue comes from the top 20% customers (called high-value customers). Thus, if we can reduce churn of the high-value customers, we will be able to reduce significant revenue leakage. Hence, this case study focuses on high value customers only.
- The dataset contains customer-level information for a span of four consecutive months - June, July, August and September. The months are encoded as 6, 7, 8 and 9, respectively.
- The **business objective** is to predict the churn in the last (i.e. the ninth) month using the data (features) from the first three months.
- This is a classification problem, where we need to predict whether the customers is about to churn or not. We have carried out Baseline Logistic Regression, then Logistic Regression with PCA, PCA + Random Forest.

Reading and understanding the data

In [3]:
```
df = pd.read_csv(r'C:\Users\Nithin.v\Downloads\BA-Case-Study-main\telecom_churn_data.csv')
df.head()
```

Out[3]:

	mobile_number	circle_id	loc_og_t2o_mou	std_og_t2o_mou	loc_ic_t2o_mou	last_date_of_month_6	last_date_of_month_7	last_date_of_month_8	last_date_of
0	7000842753	109	0.0	0.0	0.0	6/30/2014	7/31/2014	8/31/2014	
1	7001865778	109	0.0	0.0	0.0	6/30/2014	7/31/2014	8/31/2014	
2	7001625959	109	0.0	0.0	0.0	6/30/2014	7/31/2014	8/31/2014	
3	7001204172	109	0.0	0.0	0.0	6/30/2014	7/31/2014	8/31/2014	
4	7000142493	109	0.0	0.0	0.0	6/30/2014	7/31/2014	8/31/2014	

In [4]:
```
df.shape
```

Out[4]:
```
(99999, 226)
```

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99999 entries, 0 to 99998
Columns: 226 entries, mobile_number to sep_vbc_3g
dtypes: float64(179), int64(35), object(12)
memory usage: 172.4+ MB
```

Handling the missing values

Identifying the columns which has missing values and handling it in approporitate way.

In [7]:
```
df_missing_columns = (round(((df.isnull().sum()/len(df.index))*100),2).to_frame('null')).sort_values('null', ascending=False)
df_missing_columns
```

Out[7]:

	null
arpu_3g_6	74.85
night_pck_user_6	74.85
total_rech_data_6	74.85
arpu_2g_6	74.85
max_rech_data_6	74.85
...	...
max_rech_amt_7	0.00
max_rech_amt_6	0.00
total_rech_amt_9	0.00
total_rech_amt_8	0.00
sep_vbc_3g	0.00

226 rows × 1 columns

In [8]:
```
# List the columns having more than 30% missing values
col_list_missing_30 = list(df_missing_columns.index[df_missing_columns['null'] > 30])
```

In [9]:
```
# Delete the columns having more than 30% missing values
df = df.drop(col_list_missing_30, axis=1)
```

In [10]:
```
df.shape
```

Out[10]:
```
(99999, 186)
```

Data handling

- Deleting the date columns as the date columns are not required in our analysis
- Filter high-value customers
- Handling missing values in the rows
- Deleting all the attributes corresponding to the churn phase
- Churn percentage came as 3.39.

There is very little percentage of churn rate. We will take care of the class imbalance later.

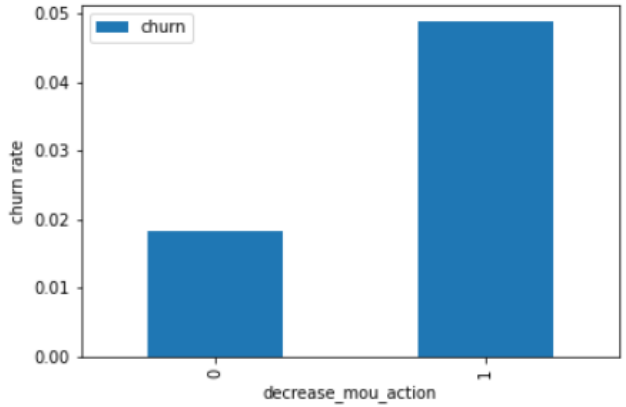
Outlier treatment

- In the filtered dataset except mobile_number and churn columns all the columns are numeric types. Hence, converting mobile_number and churn datatype to object.
- Drive new features
- Deriving new column decrease_arpu_action
- Deriving new column decrease_vbc_action

EDA

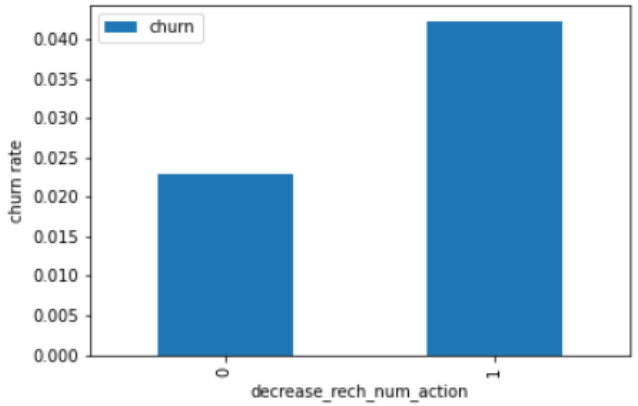
Univariate analysis:

Churn rate on the basis whether the customer decreased her/his MOU in action month.



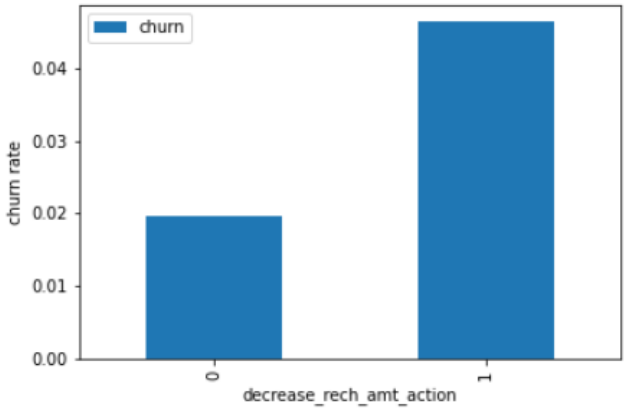

```
Analysis

We can see that the churn rate is more for the customers, whose minutes of usage(mou) decreased in the action phase than the
good phase.
```



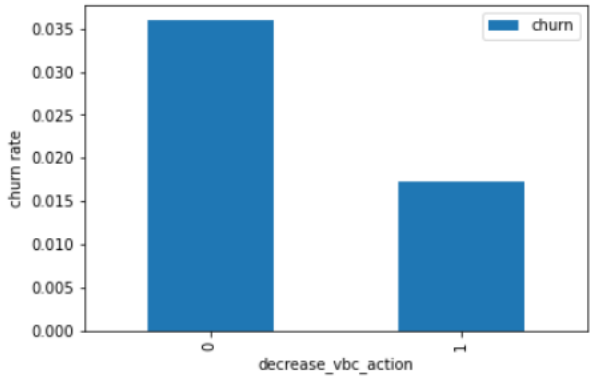
Analysis

As expected, the churn rate is more for the customers, whose number of recharge in the action phase is lesser than the number in good phase.



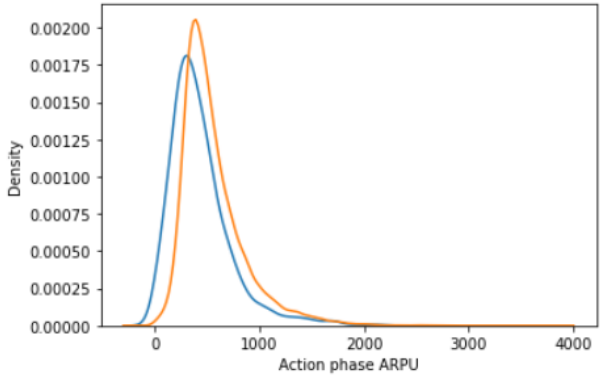
Analysis

Here also we see the same behaviour. The churn rate is more for the customers, whose amount of recharge in the action phase is lesser than the amount in good phase.



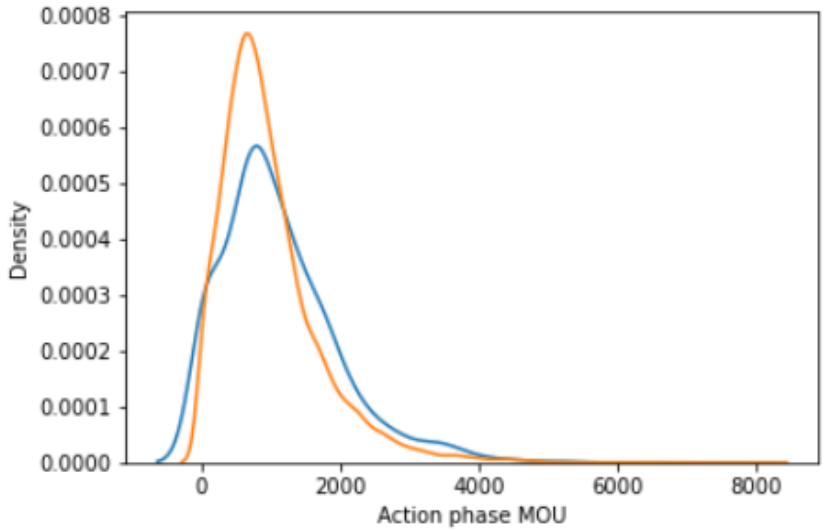
Analysis

Here we see the expected result. The churn rate is more for the customers, whose volume based cost in action month is increased. That means the customers do not do the monthly recharge more when they are in the action phase.



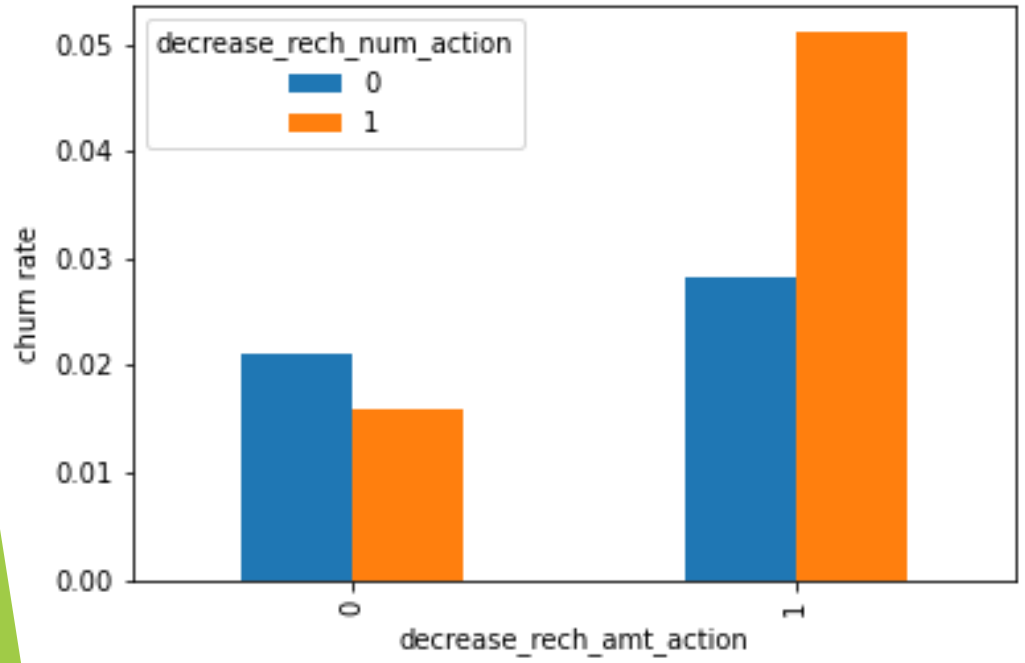
Average revenue per user (ARPU) for the churned customers is mostly densed on the 0 to 900. The higher ARPU customers are less likely to be churned.

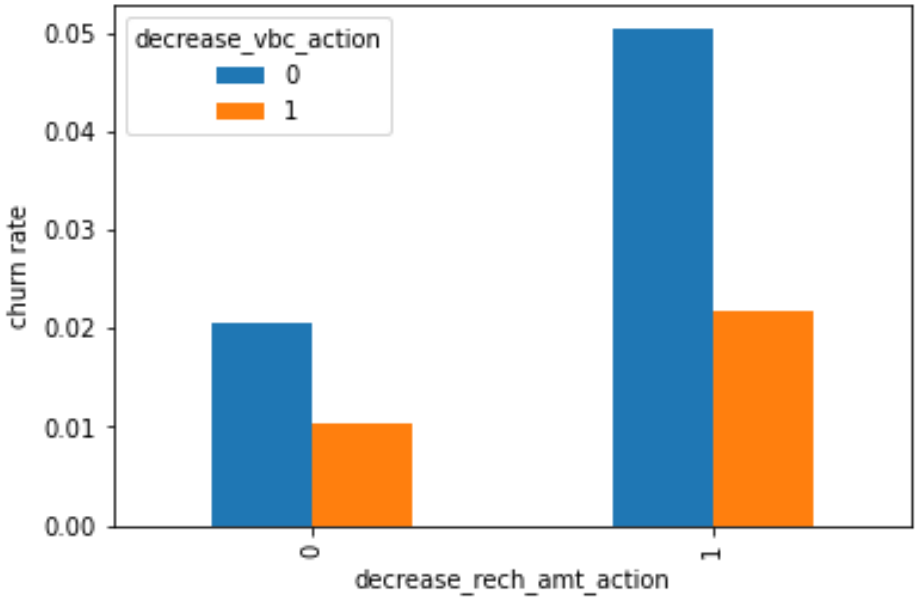
ARPU for the not churned customers is mostly densed on the 0 to 1000.

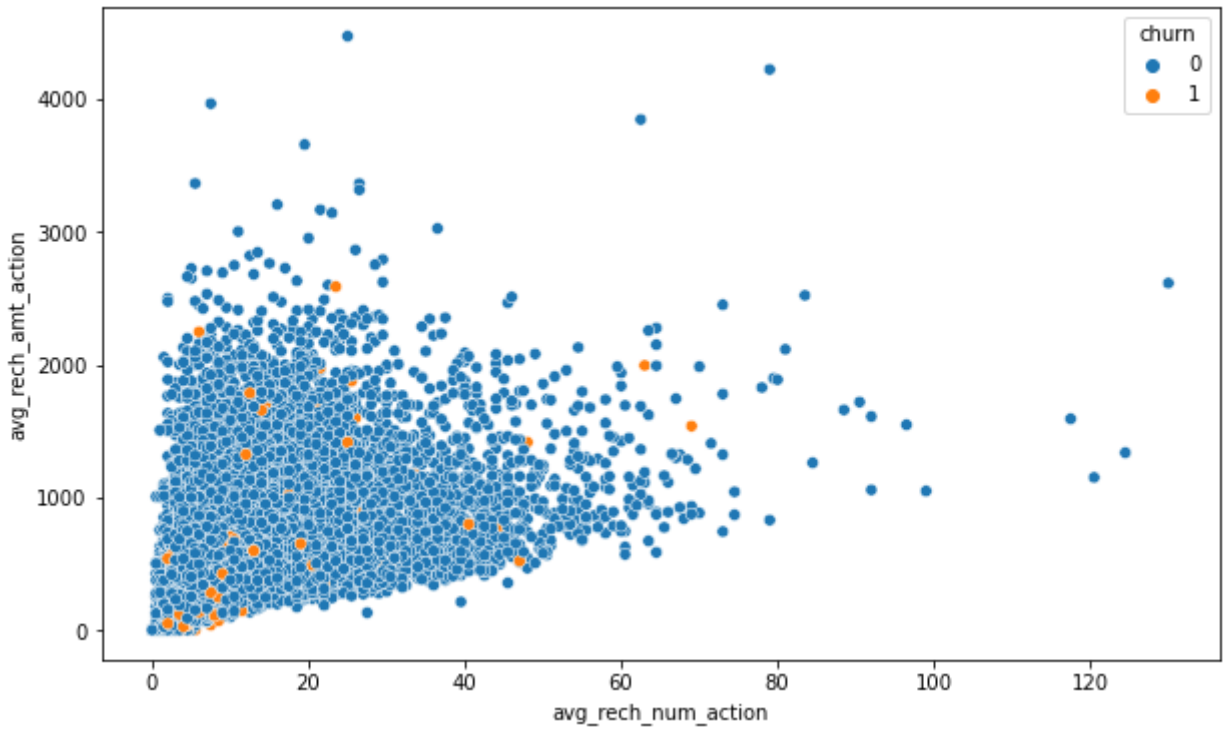


Minutes of usage(MOU) of the churn customers is mostly populated on the 0 to 2500 range. Higher the MOU, lesser the churn probability.

Bivariate analysis



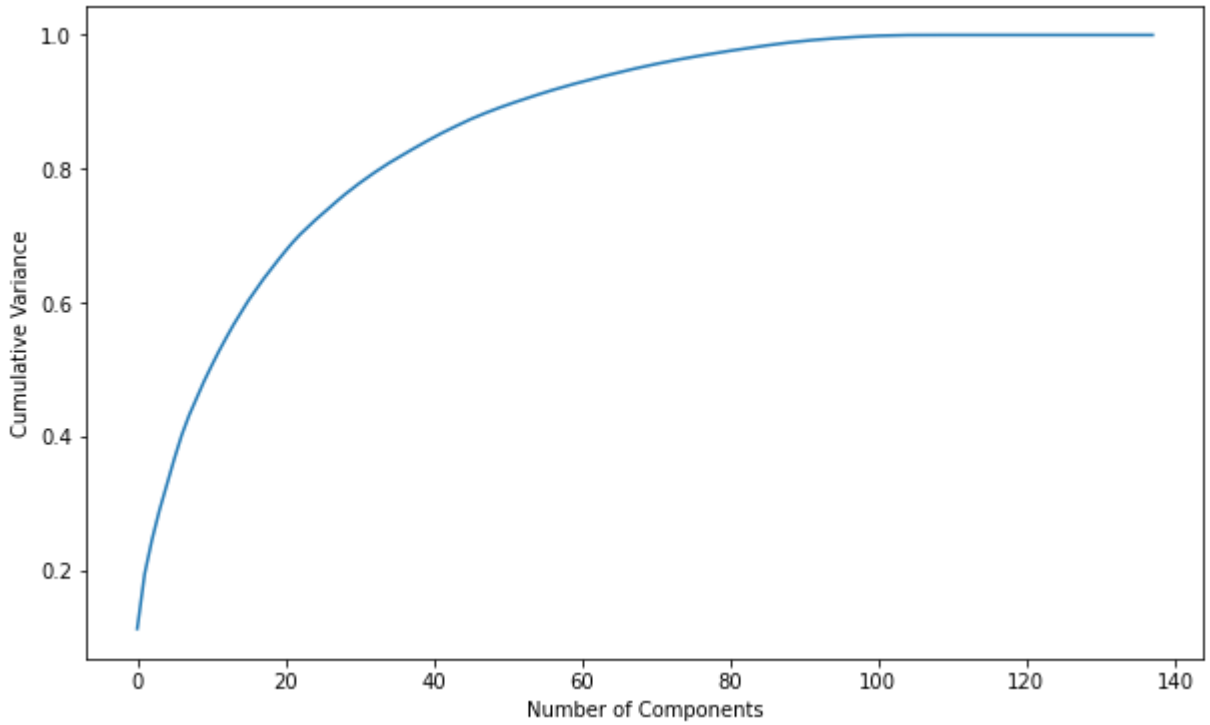




Analysis

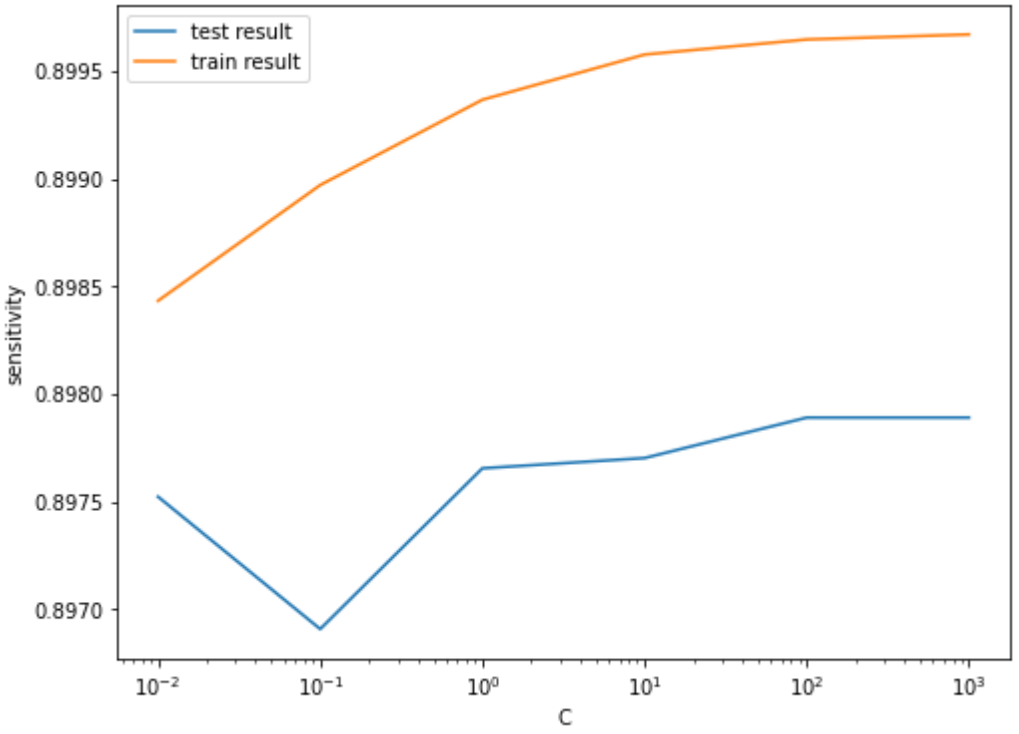
We can see from the above pattern that the recharge number and the recharge amount are mostly propotional. More the number of recharge, more the amount of the recharge.

Model with PCA

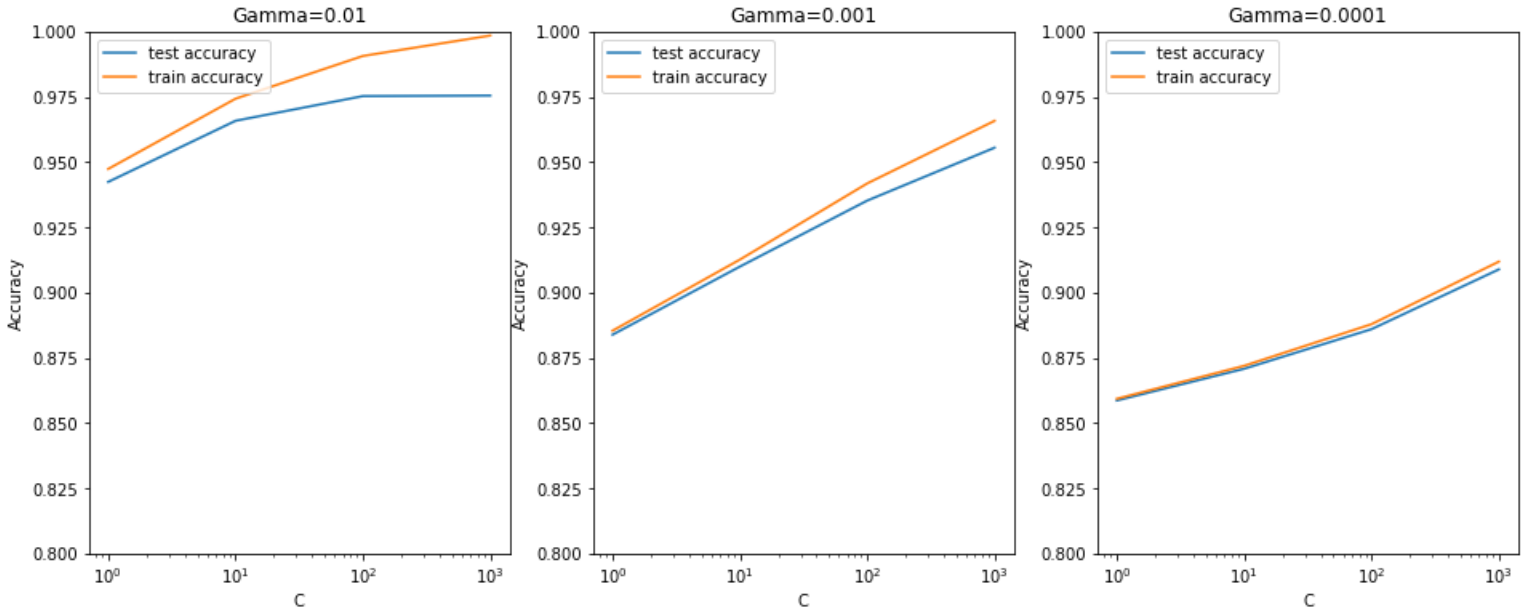


We can see that 60 components explain amost more than 90% variance of the data. So, we will perform PCA with 60 components.

Plot of C versus train and validation scores



Converting C to numeric type for plotting on x-axis




```
From the above plot, we can see that higher value of gamma leads to overfitting the model. With the lowest value of gamma
(0.0001) we have train and test accuracy almost same.

Also, at C=1 we have a good accuracy and the train and test scores are comparable.

Though sklearn suggests the optimal scores mentioned above (gamma=0.01, C=1), one could argue that it is better to choose a
simpler, more non-linear model with gamma=0.0001. This is because the optimal values mentioned here are calculated based on the
average test accuracy (but not considering subjective parameters such as model complexity).

We can achieve comparable average test accuracy (~90%) with gamma=0.0001 as well, though we'll have to increase the cost C for
that. So to achieve high accuracy, there's a tradeoff between:

High gamma (i.e. high non-linearity) and average value of C
Low gamma (i.e. less non-linearity) and high value of C
We argue that the model will be simpler if it has as less non-linearity as possible, so we choose gamma=0.0001 and a high C=100.
```

Decision tree with PCA

```
Best sensitivity:- 0.9007234539089849
DecisionTreeClassifier(max_depth=10, min_samples_leaf=50, min_samples_split=50)
```

Model with optimal hyperparameters

```
TRAIN DATA:

Accuracy:- 0.9002333722287048
Sensitivity:- 0.9177129521586931
Specificity:- 0.8827537922987164
```

```
TEST DATA:
Accuracy:- 0.8603140227395777
Sensitivity:- 0.6994818652849741
Specificity:- 0.8661181750186986
```

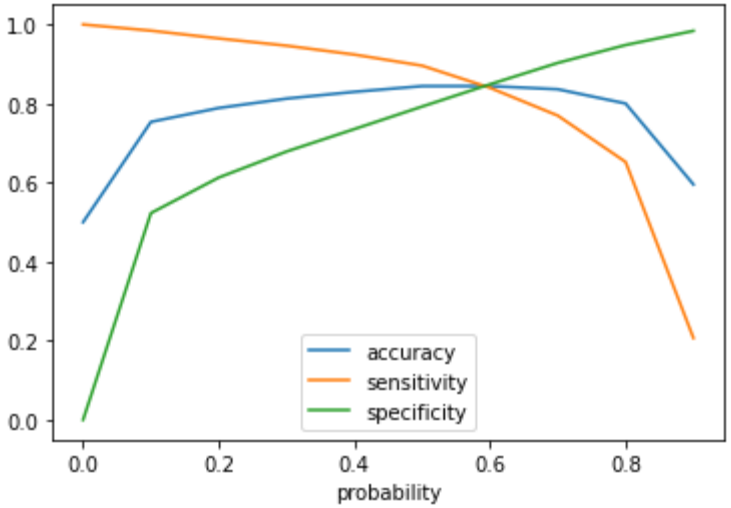
Random forest with PCA

```
Accuracy:- 0.7987727846959033
Sensitivity:- 0.7512953367875648
Specificity:- 0.800486163051608
```

Final conclusion with PCA

After trying several models we can see that for acheiving the best sensitivity, which was our ultimate goal, the classic Logistic regression or the SVM models preforms well. We have good accuracy of apporx 85%.

Without PCA



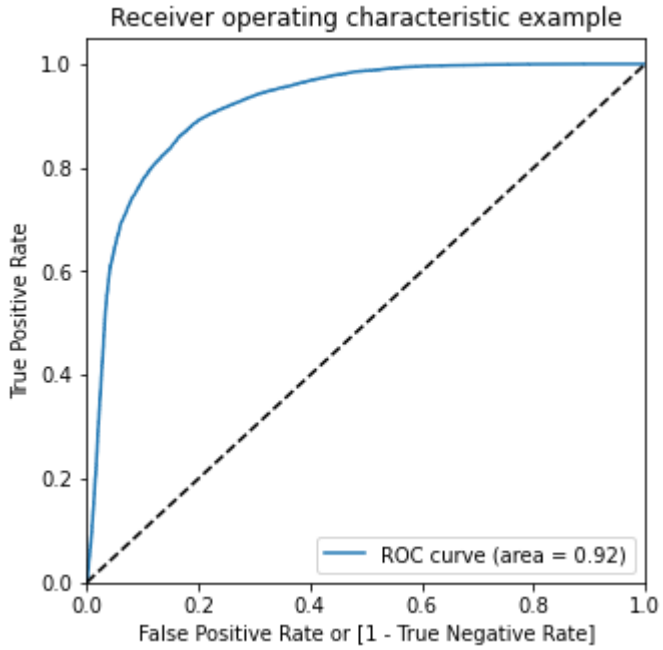

```
Analysis of the above curve
Accuracy - Becomes stable around 0.6

Sensitivity - Decreases with the increased probablity.

Specificity - Increases with the increasing probablity.

At point 0.6 where the three parameters cut each other, we can see that there is a balance bethween sensitivity and specificity
with a good accuracy.

Here we are intended to acheive better sensitivity than accuracy and specificity. Though as per the above curve, we should take
0.6 as the optimum probability cutoff, we are taking *0.5* for acheiving higher sensitivity, which is our main goal.
```



We can see the area of the ROC curve is closer to 1, whic is the Gini of the model.

Predications on final model

Accuracy:- 0.7848763761053962
Sensitivity:- 0.8238341968911918
Specificity:- 0.7834704562453254

Overall, the model is performing well in the test set, what it had learnt from the train set.

Final conclusion with no PCA
We can see that the logistic model with no PCA has good sensitivity and accuracy, which are comparable to the models with PCA. So, we can go for the more simplistic model such as logistic regression with PCA as it expliains the important predictor variables as well as the significance of each variable. The model also hels us to identify the variables which should be act upon for making the decision of the to be churned customers. Hence, the model is more relevant in terms of explaining to the business.

Business recomendation

Top predictors
Below are few top variables selected in the logistic regression model.

Variables	Coefficients
loc_ic_mou_8	-3.3287
og_others_7	-2.4711
ic_others_8	-1.5131
isd_og_mou_8	-1.3811
decrease_vbc_action	-1.3293
monthly_3g_8	-1.0943
std_ic_t2f_mou_8	-0.9503
monthly_2g_8	-0.9279
loc_ic_t2f_mou_8	-0.7102
roam_og_mou_8	0.7135

We can see most of the top variables have negative coefficients. That means, the variables are inversely correlated with the churn probablity.

E.g.:-

If the local incoming minutes of usage (loc_ic_mou_8) is lesser in the month of August than any other month, then there is a higher chance that the customer is likely to churn.

Recomendations

Target the customers, whose minutes of usage of the incoming local calls and outgoing ISD calls are less in the action phase (mostly in the month of August).
Target the customers, whose outgoing others charge in July and incoming others on August are less.
Also, the customers having value based cost in the action phase increased are more likely to churn than the other customers.
Hence, these customers may be a good target to provide offer.
Cutomers, whose monthly 3G recharge in August is more, are likely to be churned.
Customers having decreasing STD incoming minutes of usage for operators T to fixed lines of T for the month of August are more likely to churn.
Cutomers decreasing monthly 2g usage for August are most probable to churn.
Customers having decreasing incoming minutes of usage for operators T to fixed lines of T for August are more likely to churn.
roam_og_mou_8 variables have positive coefficients (0.7135). That means for the customers, whose roaming outgoing minutes of usage is increasing are more likely to churn.