Telecom Churn case Study

By

Manish Kumar

Mansi Kaparwan
Madhumita Tripathy

Business Problem Overview

- In the telecom industry, customers are able to choose from multiple service providers and actively switch from one operator to another. In this highly competitive market, the 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 now become even more important than customer acquisition.

For many incumbent operators, retaining high profitable customers is the number one business goal.

To reduce customer churn, telecom companies need to predict which customers are at high risk of churn.

In this project, you 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.

Definitions of Churn¶

There are various ways to define churn, such as:

Revenue-based churn: Customers who have not utilised any revenue-generating facilities such as mobile internet, outgoing calls, SMS etc. over a given period of time. One could also use aggregate metrics such as ‘customers who have generated less than INR 4 per month in total/average/median revenue’.

The main shortcoming of this definition is that there are customers who only receive calls/SMSes from their wage-earning counterparts, i.e. they don’t generate revenue but use the services. For example, many users in rural areas only receive calls from their wage-earning siblings in urban areas.

Understanding the Business Objective and the Data

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

Understanding Customer Behavior During Churn

Customers usually do not decide to switch to another competitor instantly, but rather over a period of time (this is especially applicable to high-value customers). In churn prediction, we assume that there are three phases of customer lifecycle :

The **‘good’ phase:** In this phase, the customer is happy with the service and behaves as usual.

The **‘action’ phase:** The customer experience starts to sore in this phase, for e.g. he/she gets a compelling offer from a competitor, faces unjust charges, becomes unhappy with service quality etc. In this phase, the customer usually shows different behaviour than the ‘good’ months. Also, it is crucial to identify high-churn-risk customers in this phase, since some corrective actions can be taken at this point (such as matching the competitor’s offer/improving the service quality etc.)

The ‘churn’ phase: In this phase, the customer is said to have churned. You define churn based on this phase. Also, it is important to note that at the time of prediction (i.e. the action months), this data is not available to you for prediction. Thus, after tagging churn as 1/0 based on this phase, you discard all data corresponding to this phase.

In this case, since you are working over a four-month window, the first two months are the ‘good’ phase, the third month is the ‘action’ phase, while the fourth month is the ‘churn’ phase.

Strategy

- ❑ Import data for Reading and Understanding
- ❑ Data Cleaning and Preparation for further analysis

a) Handling Missing Values

b) Outlier Treatment

c) Derive New Features

- ❑ Exploratory Data Analysis

a) Univariate Analysis

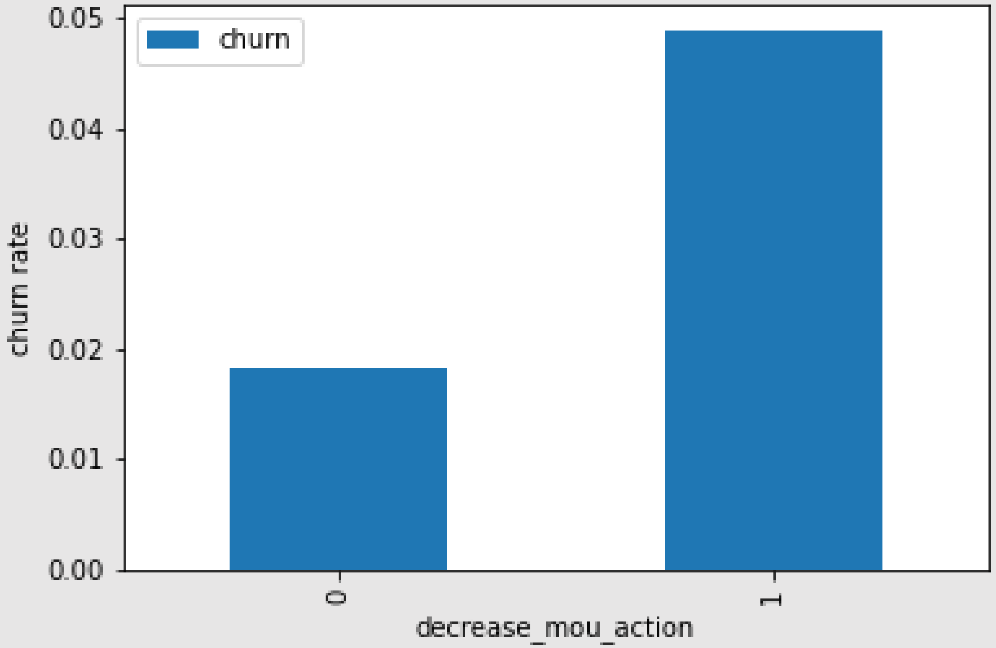
b) Bivariate Analysis

- ❑ Train Test split of data
- ❑ Performing Oversampling with SMOTE
- ❑ Feature Scaling
- ❑ PCA Test
- ❑ Model Building
- ❑ Feature Importance and Model Interpretation
- ❑ Conclusion

EDA (EXPLORATORY DATA ANALYSIS)

❖ Univariate Analysis

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



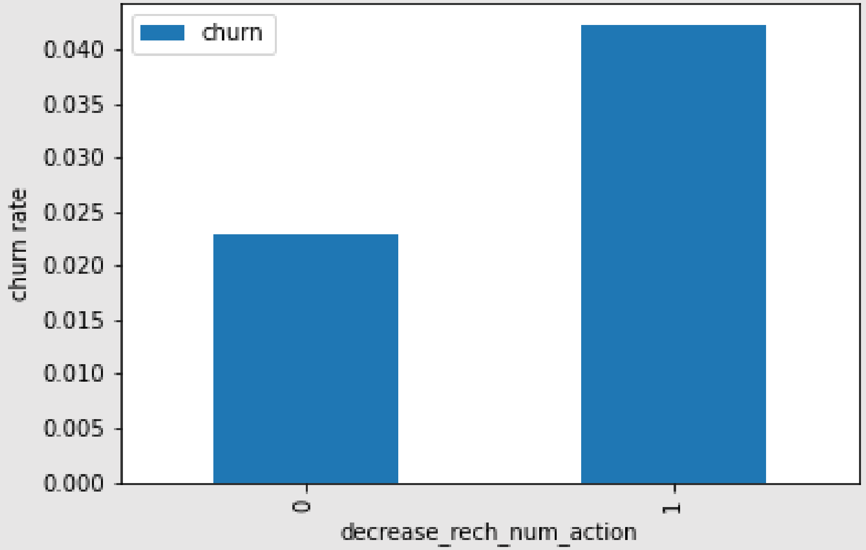
○ ***Analysis***

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.

EDA (EXPLORATORY DATA ANALYSIS)

❖ Univariate Analysis

Churn rate on the basis whether the customer decreased her/his number of recharge in action month



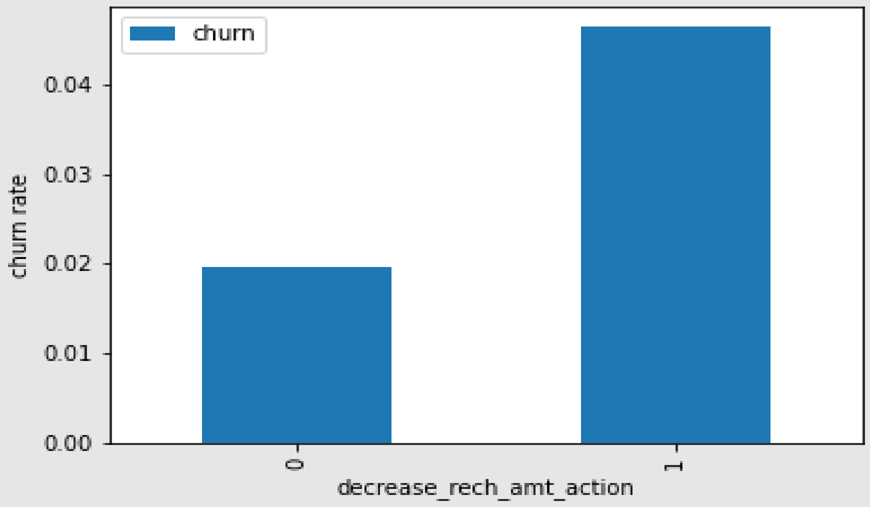
- ***Analysis***

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.

EDA (EXPLORATORY DATA ANALYSIS)

❖ Univariate Analysis

Churn rate on the basis whether the customer decreased her/his amount of recharge in action month



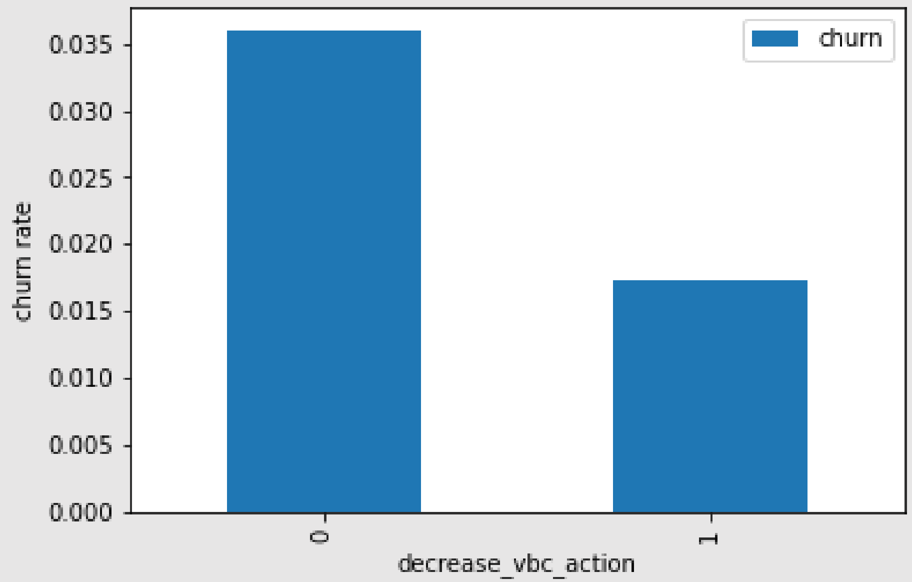
- ***Analysis***

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.

EDA (EXPLORATORY DATA ANALYSIS)

❖ Univariate Analysis

Churn rate on the basis whether the customer decreased her/his volume-based cost in action month

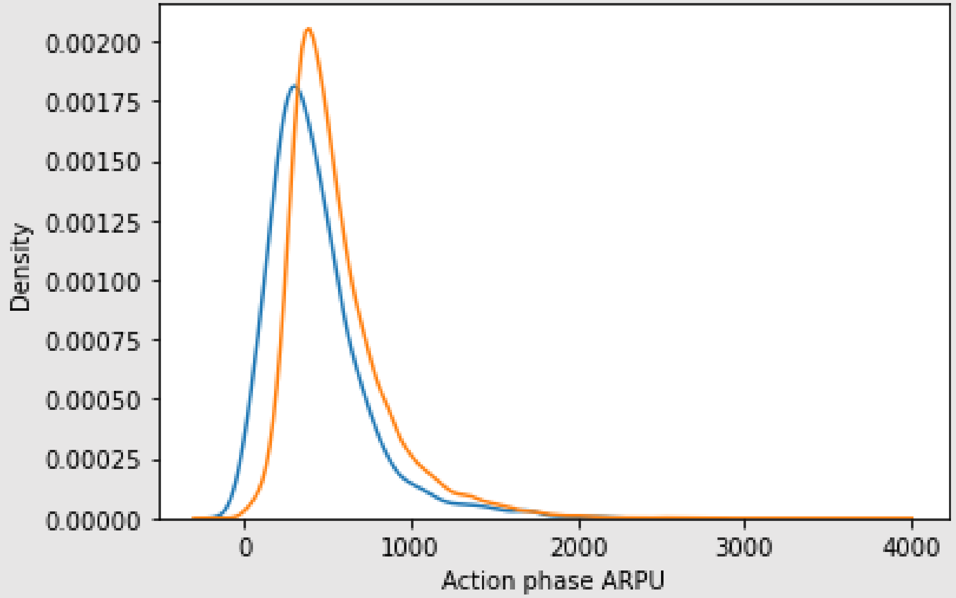


- ***Analysis***
- Here we see the expected result. The churn rate is more for the customers, whose volume based cost in action month is increased. That means the customers do not do the monthly recharge more when they are in the action phase.

EDA (EXPLORATORY DATA ANALYSIS)

❖ Univariate Analysis

Analysis of the average revenue per customer (churn and not churn) in the action phase

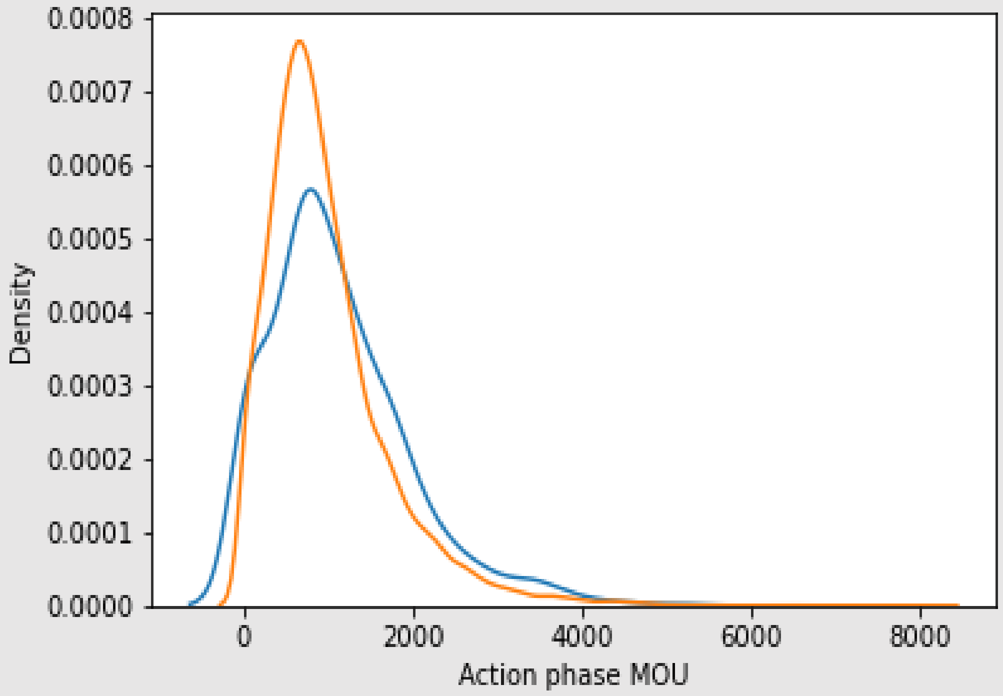


- Average revenue per user (ARPU) for the churned customers is mostly dense on the 0 to 900. The higher ARPU customers are less likely to be churned.
- ARPU for the not churned customers is mostly dense on the 0 to 1000.

EDA (EXPLORATORY DATA ANALYSIS)

❖ Univariate Analysis

Analysis of the minutes of usage MOU (churn and not churn) in the action phase.

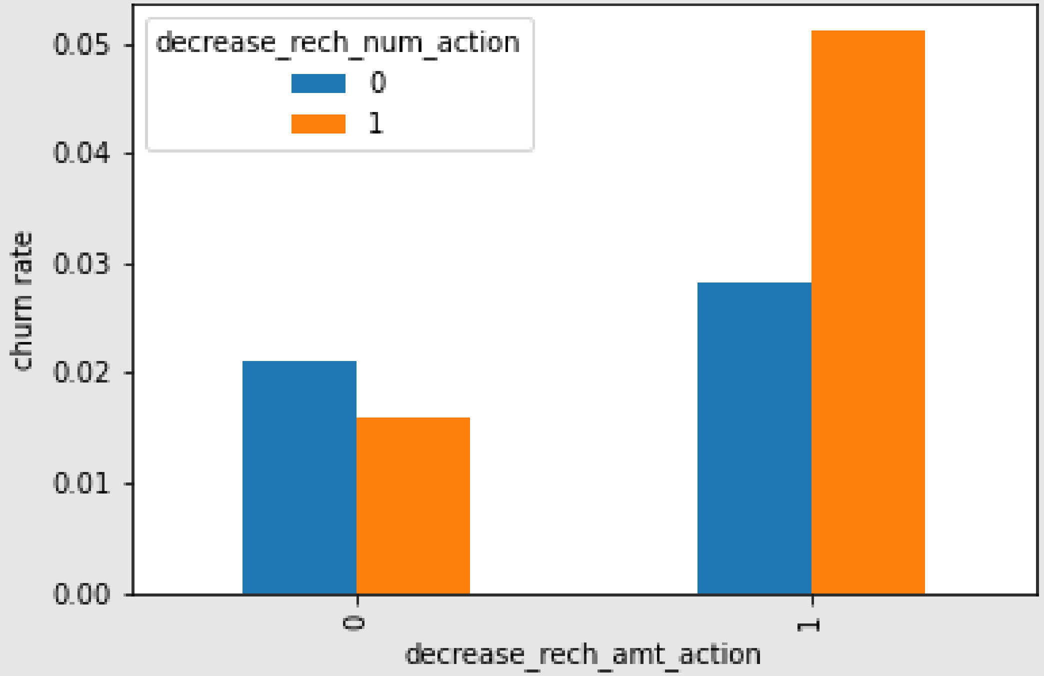


Minutes of usage(MOU) of the churn customers is mostly populated on the 0 to 2500 range. Higher the MOU, lesser the churn probability.

EDA (EXPLORATORY DATA ANALYSIS)

❖ Bivariate Analysis

Analysis of churn rate by the decreasing recharge amount and number of recharge in the action phase

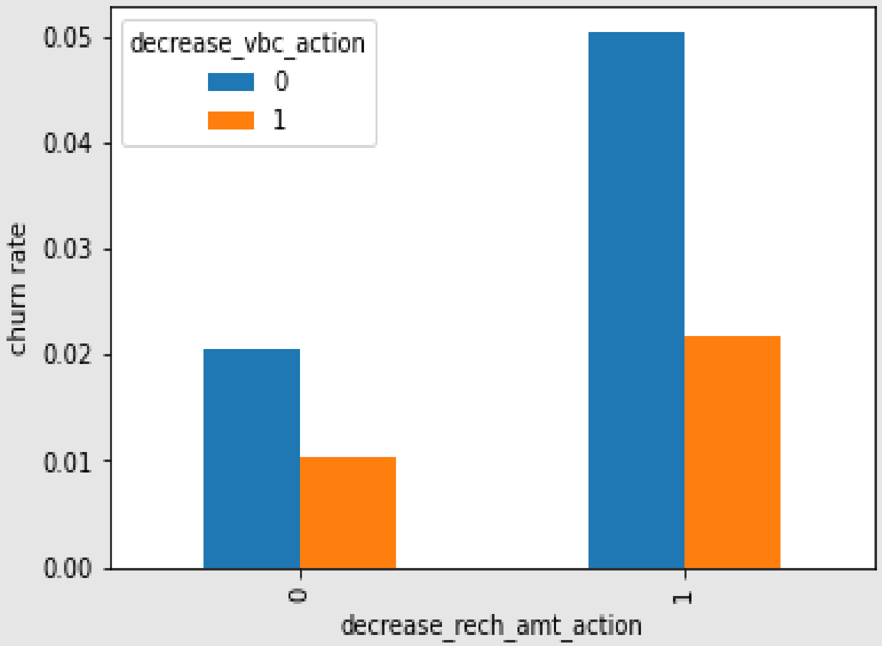


- ***Analysis***
- We can see from the above plot, that the churn rate is more for the customers, whose recharge amount as well as number of recharge have decreased in the action phase than the good phase.

EDA (EXPLORATORY DATA ANALYSIS)

❖ Bivariate Analysis

Analysis of churn rate by the decreasing recharge amount and volume based cost in the action phase



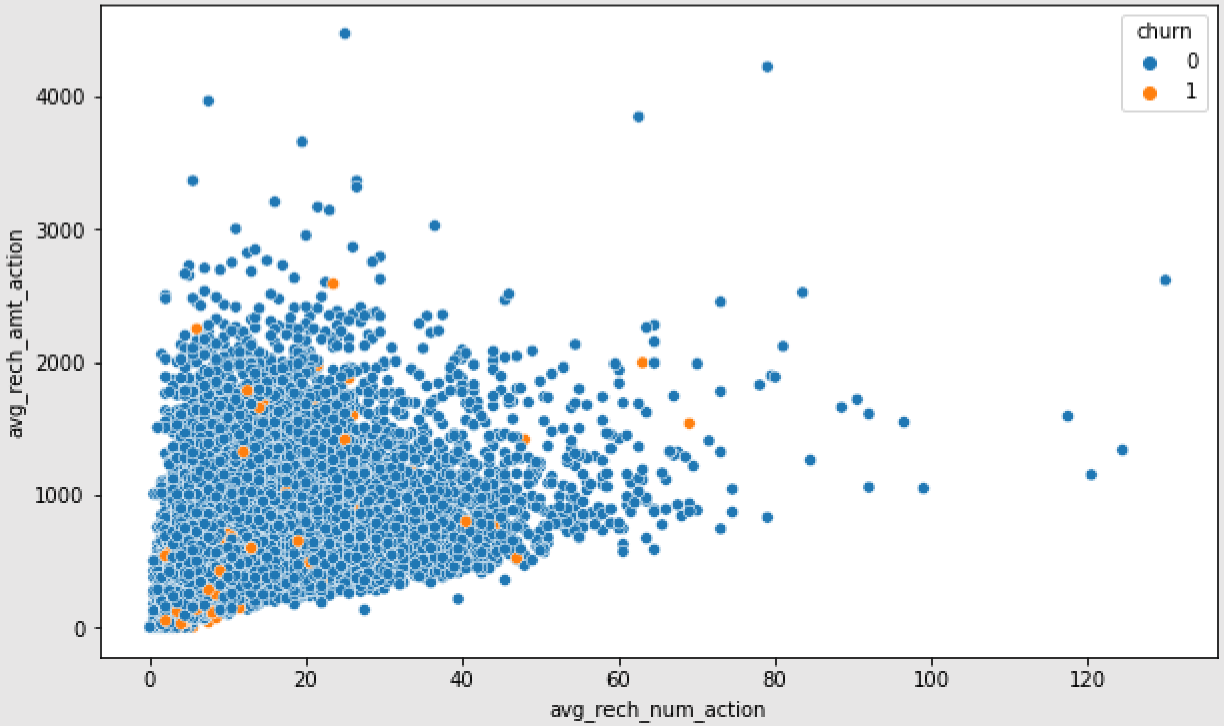
- ***Analysis***

Here, also we can see that the churn rate is more for the customers, whose recharge amount is decreased along with the volume-based cost is increased in the action month.

EDA (EXPLORATORY DATA ANALYSIS)

❖ Bivariate Analysis

Analysis of recharge amount and number of recharge in action month

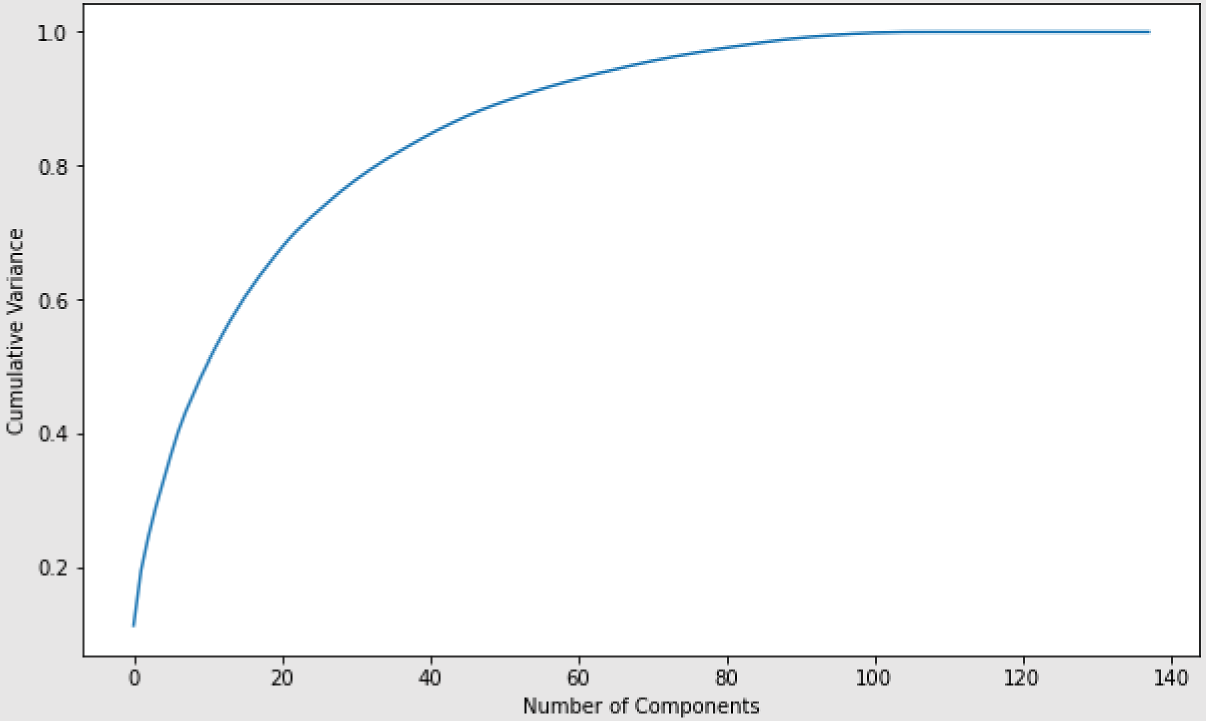


- ***Analysis***
- We can see from the above pattern that the recharge number and the recharge amount are mostly propotional. More the number of recharge, more the amount of the recharge.

PCA Testing

Plotting scree plot

Number of Components Vs Cumulative Variance



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

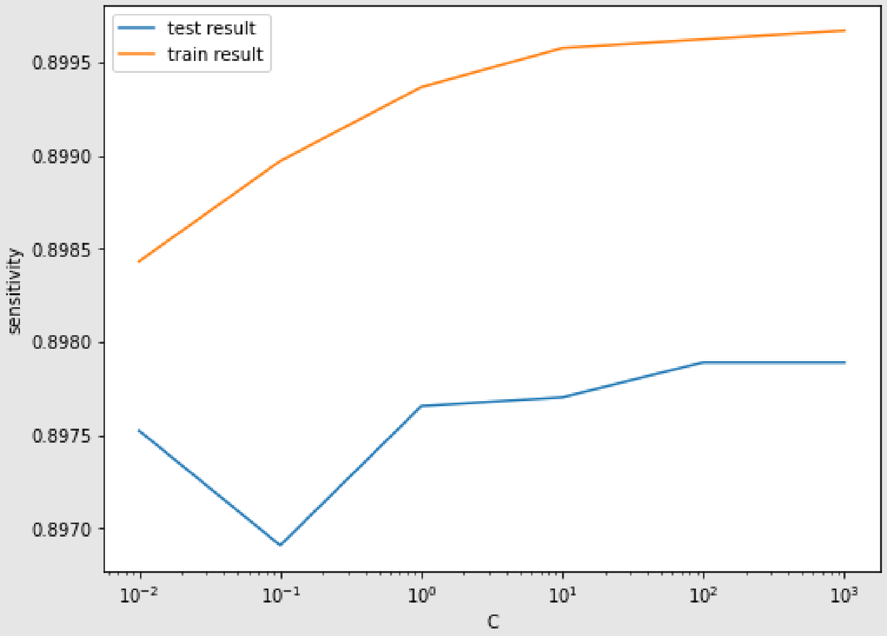
- **Emphasize Sensitivity/Recall than Accuracy**
- We are more focused on higher Sensitivity/Recall score than the accuracy.
- Because we need to care more about churn cases than the not churn cases. The main goal is to retain the customers, who have the possibility to churn. There should not be a problem, if we consider few not churn customers as churn customers and provide them some incentives for retaining them. Hence, the sensitivity score is more important here.

❖ Logistic regression with PCA

Tuning hyperparameter C

C is the the inverse of regularization strength in Logistic Regression. Higher values of C correspond to less regularization.

Plot of C versus train and validation scores



Here Best Sore with Best C is

The highest test sensitivity is 0.8978916608693863 at C = 100

Prediction on the train set

Confusion matrix

[[17908 3517]
 [2154 19271]]

Accuracy:- 0.8676546091015169

Sensitivity:- 0.899463243873979

Specificity:- 0.8358459743290548

Prediction on the test set

Confusion matrix

[[4452 896]
 [36 157]]

Accuracy:- 0.8317993142032124

Sensitivity:- 0.8134715025906736

Specificity:- 0.8324607329842932

Model summary

Train set

Accuracy = 0.86

Sensitivity = 0.89

Specificity = 0.83

Test set

Accuracy = 0.83

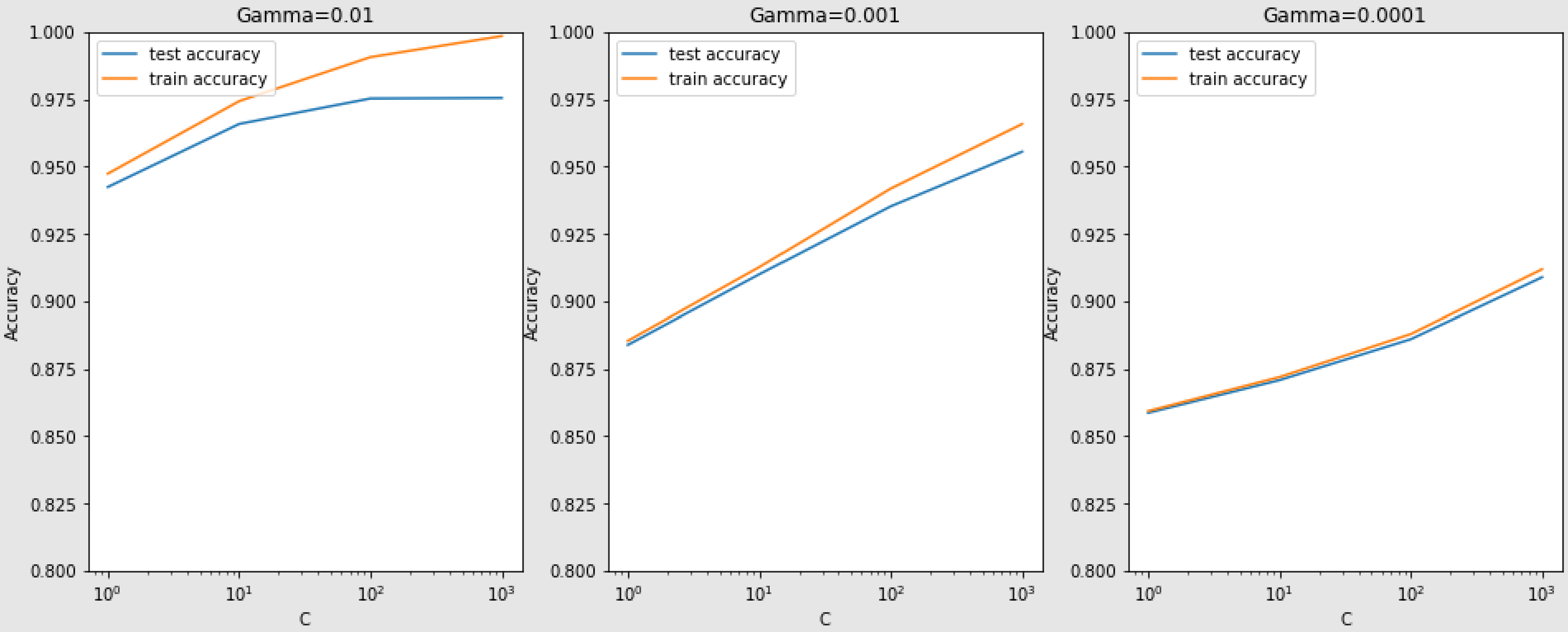
Sensitivity = 0.81

Specificity = 0.83

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

❖ Support Vector Machine(SVM) with PCA

Plotting the accuracy with various C and gamma values



Printing the best score

The best test score is 0.9754959911159373 corresponding to hyperparameters {'C': 1000, 'gamma': 0.01}

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=100 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=1000), 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.

Building the model with optimal hyperparameters

SVC(C=100, gamma=0.0001)

Prediction on the train set

Confusion matrix

[[18376 3049]

[1585 19840]]

Accuracy:- 0.891855309218203

Sensitivity:- 0.9260210035005835

Specificity:- 0.8576896149358226

Prediction on the test set

Confusion matrix

[[4557 791]

[36 157]]

Accuracy:- 0.8507489622811767

Sensitivity:- 0.8134715025906736

Specificity:- 0.8520942408376964

Model summary

Train set

Accuracy = 0.89

Sensitivity = 0.92

Specificity = 0.85

Test set

Accuracy = 0.85

Sensitivity = 0.81

Specificity = 0.85

❖ Decision tree with PCA

The optimal sensitivity score and hyperparameters

Best sensitivity:- 0.9004900816802801
DecisionTreeClassifier(max_depth=10, min_samples_leaf=50, min_samples_split=50)

Prediction on the train set
Confusion matrix

```
[[18913  2512]
 [ 1763 19662]]
```

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

Prediction on the test set
Confusion matrix

```
[[4632  716]
 [  58  135]]
```

Accuracy:- 0.8603140227395777
Sensitivity:- 0.6994818652849741
Specificity:- 0.8661181750186986

Model summary

Train set
Accuracy = 0.90
Sensitivity = 0.91
Specificity = 0.88

Test set

Accuracy = 0.86
Sensitivity = 0.70
Specificity = 0.87

We can see from the model performance that the Sensitivity has been decreased while evaluating the model on the test set. However, the accuracy and specificity is quite good in the test set.

❖ Random forest with PCA

The optimal accuracy score and hyperparameters

We can get accuracy of 0.8452042054330283 using {'max_depth': 5, 'max_features': 20, 'min_samples_leaf': 50, 'min_samples_split': 100, 'n_estimators': 100}

Prediction on the train set

Confusion matrix

```
[[17366  4059]
 [ 2434 18991]]
```

Accuracy:- 0.8484714119019837
Sensitivity:- 0.8863943990665111
Specificity:- 0.8105484247374563

Prediction on the test set

Confusion matrix

```
[[4293 1055]
 [  46  147]]
```

Accuracy:- 0.8012994044396319
Sensitivity:- 0.7616580310880829
Specificity:- 0.8027299925205684

Model summary

Train set
Accuracy = 0.84
Sensitivity = 0.88
Specificity = 0.80

Test set
Accuracy = 0.80
Sensitivity = 0.75
Specificity = 0.80

We can see from the model performance that the Sensitivity has been decreased while evaluating the model on the test set. However, the accuracy and specificity is quite good in the test set.

Final conclusion with PCA

After trying several models, we can see that for achieving the best sensitivity, which was our ultimate goal, the classic Logistic regression or the SVM models preforms well. For both the models the sensitivity was approx 81%. Also, we have good accuracy of approx. 85%.

❖ Model building Without PCA

Logistic regression with No PCA

Generalized Linear Model Regression Results

Dep. Variable:	churn	**No. Observations:**	42850
Model:	GLM	**Df Residuals:**	42720
Model Family:	Binomial	**Df Model:**	129
Link Function:	Logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	nan
Date:	Sat, 10 Jun 2023	**Deviance:**	23572.
Time:	00:10:12	**Pearson chi2:**	3.71e+05
No. Iterations:	100	**Pseudo R-squ. (CS):**	nan
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	-90.6375	4452.242	-0.020	0.984	-8816.872	8635.597
loc_og_t2o_mou	-3.08e-06	0.000	-0.022	0.982	-0.000	0.000
std_og_t2o_mou	-2.063e-06	0.000	-0.014	0.989	-0.000	0.000
loc_ic_t2o_mou	-9.631e-06	0.000	-0.021	0.983	-0.001	0.001
arpu_6	-0.0338	0.081	-0.418	0.676	-0.192	0.125
arpu_7	0.0855	0.086	0.995	0.320	-0.083	0.254
arpu_8	0.0909	0.110	0.828	0.408	-0.124	0.306
onnet_mou_6	15.5140	3.577	4.337	0.000	8.504	22.524
onnet_mou_7	-4.3249	1.811	-2.388	0.017	-7.875	-0.774
onnet_mou_8	2.3520	1.827	1.287	0.198	-1.229	5.933
offnet_mou_6	15.0883	3.365	4.484	0.000	8.494	21.683
offnet_mou_7	-1.7627	1.716	-1.027	0.304	-5.126	1.601
offnet_mou_8	-0.5503	1.885	-0.292	0.770	-4.244	3.144
roam_ic_mou_6	0.1622	0.036	4.487	0.000	0.091	0.233
roam_ic_mou_7	-0.0099	0.052	-0.189	0.850	-0.112	0.092
roam_ic_mou_8	0.2041	0.044	4.662	0.000	0.118	0.290
roam_og_mou_6	-5.1508	1.132	-4.549	0.000	-7.370	-2.932
roam_og_mou_7	0.8855	0.473	1.873	0.061	-0.041	1.812
roam_og_mou_8	0.0929	0.531	0.175	0.861	-0.948	1.134
loc_og_t2t_mou_6	-3303.0832	656.615	-5.030	0.000	-4590.024	-2016.142
loc_og_t2t_mou_7	-1474.6161	680.014	-2.169	0.030	-2807.419	-141.813
loc_og_t2t_mou_8	5516.0876	628.351	8.779	0.000	4284.542	6747.633
loc_og_t2m_mou_6	-3342.7075	664.372	-5.031	0.000	-4644.852	-2040.563
loc_og_t2m_mou_7	-1392.1067	641.322	-2.171	0.030	-2649.075	-135.139
loc_og_t2m_mou_8	5887.3427	670.474	8.781	0.000	4573.238	7201.447
loc_og_t2f_mou_6	-285.2471	56.730	-5.028	0.000	-396.436	-174.058
loc_og_t2f_mou_7	-123.0164	56.696	-2.170	0.030	-234.138	-11.895
loc_og_t2f_mou_8	487.3958	55.536	8.776	0.000	378.548	596.243
loc_og_t2c_mou_6	0.0433	0.022	1.971	0.049	0.000	0.086
loc_og_t2c_mou_7	0.0099	0.021	0.462	0.644	-0.032	0.052
loc_og_t2c_mou_8	0.0673	0.023	2.980	0.003	0.023	0.111
loc_og_mou_6	3756.6102	1269.528	2.959	0.003	1268.380	6244.840
loc_og_mou_7	5686.6260	1330.779	4.273	0.000	3078.348	8294.904
loc_og_mou_8	-265.7536	1351.526	-0.197	0.844	-2914.696	2383.189
std_og_t2t_mou_6	-1.309e+04	1867.608	-7.009	0.000	-1.68e+04	-9429.311
std_og_t2t_mou_7	-9674.3998	1822.532	-5.308	0.000	-1.32e+04	-6102.303
std_og_t2t_mou_8	5854.7918	1510.527	3.876	0.000	2894.213	8815.371
std_og_t2m_mou_6	-1.214e+04	1732.558	-7.009	0.000	-1.55e+04	-8748.260
std_og_t2m_mou_7	-9439.1294	1777.871	-5.309	0.000	-1.29e+04	-5954.566
std_og_t2m_mou_8	5966.0464	1538.574	3.878	0.000	2950.497	8981.596
std_og_t2f_mou_6	-255.4260	36.403	-7.017	0.000	-326.775	-184.077
std_og_t2f_mou_7	-213.6015	40.270	-5.304	0.000	-292.530	-134.673
std_og_t2f_mou_8	142.4566	36.769	3.874	0.000	70.391	214.523
std_og_t2c_mou_6	8.329e-06	0.000	0.020	0.984	-0.001	0.001
std_og_t2c_mou_7	-7.681e-06	0.000	-0.021	0.983	-0.001	0.001
std_og_t2c_mou_8	3.308e-06	0.000	0.021	0.983	-0.000	0.000
std_og_mou_6	1.446e+04	2967.405	4.873	0.000	8644.702	2.03e+04
std_og_mou_7	2.105e+04	3104.376	6.782	0.000	1.5e+04	2.71e+04
std_og_mou_8	7815.2301	2768.567	2.823	0.005	2388.938	1.32e+04

Model analysis

1.We can see that there are few features have positive coefficients and few have negative.

2.Many features have higher p-values and hence became insignificant in the model.

Coarse tuning (Auto+Manual)

We'll first eliminate a few features using Recursive Feature Elimination (RFE), and once we have reached a small set of variables to work with, we can then use manual feature elimination (i.e. manually eliminating features based on observing the p-values and VIFs).

❖ Without PCA

Feature Selection Using RFE

Model-1 with RFE selected columns

Generalized Linear Model Regression Results

Dep. Variable:	churn	**No. Observations:**	42850
Model:	GLM	**Df Residuals:**	42834
Model Family:	Binomial	**Df Model:**	15
Link Function:	Logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	nan
Date:	Sat, 10 Jun 2023	**Deviance:**	30008.
Time:	00:11:21	**Pearson chi2:**	4.49e+06
No. Iterations:	41	**Pseudo R-squ. (CS):**	nan
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	-53.0128	4235.111	-0.013	0.990	-8353.678	8247.653
offnet_mou_7	0.6096	0.026	23.449	0.000	0.559	0.661
offnet_mou_8	-3.2532	0.106	-30.548	0.000	-3.462	-3.045
roam_og_mou_8	1.2482	0.032	39.496	0.000	1.186	1.310
std_og_t2m_mou_8	2.4408	0.094	26.101	0.000	2.258	2.624
isd_og_mou_8	-1.0212	0.194	-5.271	0.000	-1.401	-0.641
og_others_7	-1.1915	0.862	-1.382	0.167	-2.881	0.498
og_others_8	-3780.7240	3.08e+05	-0.012	0.990	-6.08e+05	6.01e+05
loc_ic_t2f_mou_8	-0.7547	0.072	-10.487	0.000	-0.896	-0.614
loc_ic_mou_8	-1.9744	0.066	-30.078	0.000	-2.103	-1.846
std_ic_t2f_mou_8	-0.7922	0.075	-10.607	0.000	-0.939	-0.646
ic_others_8	-1.4913	0.132	-11.305	0.000	-1.750	-1.233
total_rech_num_8	-0.4840	0.018	-26.977	0.000	-0.519	-0.449
monthly_2g_8	-0.9031	0.043	-20.851	0.000	-0.988	-0.818
monthly_3g_8	-0.9871	0.043	-22.711	0.000	-1.072	-0.902
decrease_vbc_action	-1.3078	0.073	-17.956	0.000	-1.451	-1.165

Create a dataframe that will contain the names of all the feature variables and their respective VIFs

	Features	VIF
1	offnet_mou_8	7.45
3	std_og_t2m_mou_8	6.27
0	offnet_mou_7	1.92
8	loc_ic_mou_8	1.68
7	loc_ic_t2f_mou_8	1.21
11	total_rech_num_8	1.19
2	roam_og_mou_8	1.16
14	decrease_vbc_action	1.08
13	monthly_3g_8	1.06
6	og_others_8	1.05
12	monthly_2g_8	1.05
5	og_others_7	1.04
9	std_ic_t2f_mou_8	1.02
10	ic_others_8	1.02
4	isd_og_mou_8	1.01

Removing column og_others_8, which is insignificatnt as it has the highest p-value 0.99

Model-2

Generalized Linear Model Regression Results

Dep. Variable:	churn	**No. Observations:**	42850
Model:	GLM	**Df Residuals:**	42835
Model Family:	Binomial	**Df Model:**	14
Link Function:	Logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-15034.
Date:	Sat, 10 Jun 2023	**Deviance:**	30068.
Time:	00:11:22	**Pearson chi2:**	4.51e+06
No. Iterations:	11	**Pseudo R-squ. (CS):**	0.4957
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	-1.1052	0.031	-35.342	0.000	-1.167	-1.044
offnet_mou_7	0.6081	0.026	23.427	0.000	0.557	0.659
offnet_mou_8	-3.2557	0.106	-30.603	0.000	-3.464	-3.047
roam_og_mou_8	1.2491	0.031	39.747	0.000	1.188	1.311
std_og_t2m_mou_8	2.4428	0.093	26.146	0.000	2.260	2.626
isd_og_mou_8	-1.0982	0.196	-5.590	0.000	-1.483	-0.713
og_others_7	-1.8793	0.818	-2.299	0.022	-3.482	-0.277
loc_ic_t2f_mou_8	-0.7548	0.072	-10.491	0.000	-0.896	-0.614
loc_ic_mou_8	-1.9714	0.066	-30.058	0.000	-2.100	-1.843
std_ic_t2f_mou_8	-0.8020	0.075	-10.727	0.000	-0.949	-0.655
ic_others_8	-1.4871	0.132	-11.278	0.000	-1.746	-1.229
total_rech_num_8	-0.4864	0.018	-27.146	0.000	-0.522	-0.451
monthly_2g_8	-0.9066	0.043	-20.866	0.000	-0.992	-0.821
monthly_3g_8	-0.9862	0.043	-22.700	0.000	-1.071	-0.901
decrease_vbc_action	-1.3097	0.073	-17.994	0.000	-1.452	-1.167

Checking VIF for Model-2

	Features	VIF
1	offnet_mou_8	7.45
3	std_og_t2m_mou_8	6.27
0	offnet_mou_7	1.92
7	loc_ic_mou_8	1.68
6	loc_ic_t2f_mou_8	1.21
10	total_rech_num_8	1.19
2	roam_og_mou_8	1.16
13	decrease_vbc_action	1.08
12	monthly_3g_8	1.06
11	monthly_2g_8	1.05
8	std_ic_t2f_mou_8	1.02
4	isd_og_mou_8	1.01
9	ic_others_8	1.01
5	og_others_7	1.00

As we can see from the model summary that all the variables p-values are significant and offnet_mou_8 column has the highest VIF 7.45. Hence, deleting offnet_mou_8 column.

Model-3

Generalized Linear Model Regression Results

Dep. Variable:	churn	**No. Observations:**	42850
Model:	GLM	**Df Residuals:**	42836
Model Family:	Binomial	**Df Model:**	13
Link Function:	Logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-15720.
Date:	Sat, 10 Jun 2023	**Deviance:**	31440.
Time:	00:11:22	**Pearson chi2:**	3.92e+06
No. Iterations:	11	**Pseudo R-squ. (CS):**	0.4793
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	-1.2058	0.032	-37.536	0.000	-1.269	-1.143
offnet_mou_7	0.3665	0.022	16.456	0.000	0.323	0.410
roam_og_mou_8	0.7135	0.024	29.260	0.000	0.666	0.761
std_og_t2m_mou_8	-0.2474	0.022	-11.238	0.000	-0.291	-0.204
isd_og_mou_8	-1.3811	0.212	-6.511	0.000	-1.797	-0.965
og_others_7	-2.4711	0.872	-2.834	0.005	-4.180	-0.762
loc_ic_t2f_mou_8	-0.7102	0.075	-9.532	0.000	-0.856	-0.564
loc_ic_mou_8	-3.3287	0.057	-58.130	0.000	-3.441	-3.216
std_ic_t2f_mou_8	-0.9503	0.078	-12.181	0.000	-1.103	-0.797
ic_others_8	-1.5131	0.129	-11.771	0.000	-1.765	-1.261
total_rech_num_8	-0.5060	0.018	-28.808	0.000	-0.540	-0.472
monthly_2g_8	-0.9279	0.044	-21.027	0.000	-1.014	-0.841
monthly_3g_8	-1.0943	0.046	-23.615	0.000	-1.185	-1.004
decrease_vbc_action	-1.3293	0.072	-18.478	0.000	-1.470	-1.188

Checking VIF for Model-2

	Features	VIF
2	std_og_t2m_mou_8	1.87
0	offnet_mou_7	1.72
6	loc_ic_mou_8	1.33
5	loc_ic_t2f_mou_8	1.21
9	total_rech_num_8	1.17
12	decrease_vbc_action	1.07
1	roam_og_mou_8	1.06
11	monthly_3g_8	1.06
10	monthly_2g_8	1.05
7	std_ic_t2f_mou_8	1.02
3	isd_og_mou_8	1.01
8	ic_others_8	1.01
4	og_others_7	1.00

Now from the model summary and the VIF list we can see that all the variables are significant and there is no multicollinearity among the variables.

Hence, we can conclused that Model-3 log_no_pca_3 will be the final model.

Model performance on the train set

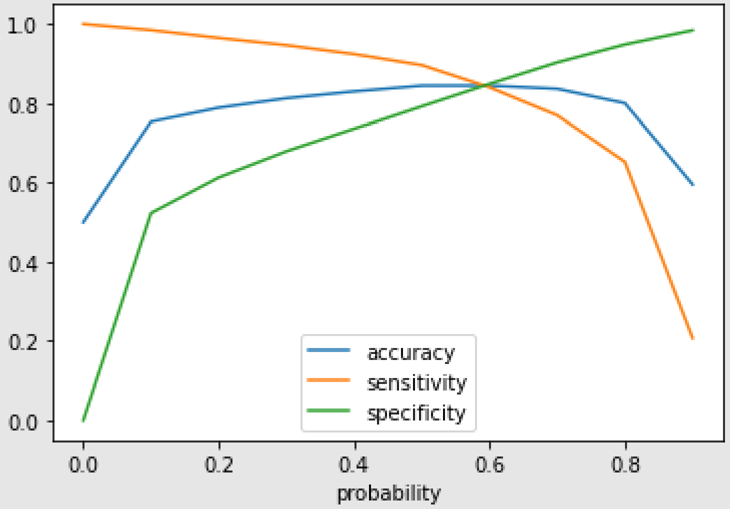
Creating a data frame with the actual churn and the predicted probabilities

	churn	churn_prob	CustID
0	0	2.687411e-01	0
1	0	7.047483e-02	1
2	0	8.024370e-02	2
3	0	3.439222e-03	3
4	0	5.253815e-19	4

Finding Optimal Probability Cutoff Point

	churn	churn_prob	CustID	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0	0	2.687411e-01	0	1	1	1	0	0	0	0	0	0	0
1	0	7.047483e-02	1	1	0	0	0	0	0	0	0	0	0
2	0	8.024370e-02	2	1	0	0	0	0	0	0	0	0	0
3	0	3.439222e-03	3	1	0	0	0	0	0	0	0	0	0
4	0	5.253815e-19	4	1	0	0	0	0	0	0	0	0	0

Plotting accuracy, sensitivity and specificity for different probabilities



<u>**Analysis of the above curve**</u>
Accuracy - Becomes stable **around 0.6**
Sensitivity - Decreases with the increased probability.
Specificity - Increases with the increasing probability.
At point 0.6 where the three parameters cut each other, we can see that there is a balance between sensitivity and specificity with a good accuracy.
Here we are intended to achieve 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 achieving higher sensitivity, which is our main goal

Model performance on the train set

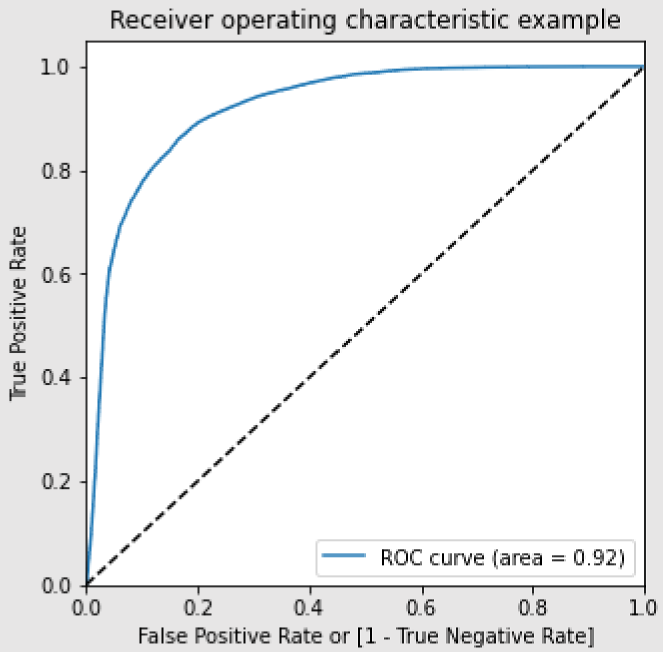
Metrics

Confusion metrics

[[16978 4447]
 [2232 19193]]

Accuracy:- 0.84413068884480747
Sensitivity:- 0.8958226371061844
Specificity:- 0.792438739789965
We have got good accuracy, sensitivity and specificity on the train set prediction.

Plotting the ROC Curve (Trade off between sensitivity & specificity)



Testing the model on the test set

Predictions on the test set with final mode

	CustID	churn	churn_prob
0	5704	0	0.034015
1	64892	0	0.000578
2	39613	0	0.513564
3	93118	0	0.020480
4	81235	0	0.034115

	CustID	churn	churn_prob	test_predicted
0	5704	0	0.034015	0
1	64892	0	0.000578	0
2	39613	0	0.513564	1
3	93118	0	0.020480	0
4	81235	0	0.034115	0

Metrics

Confusion matrix

[[4190 1158]
[34 159]]

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

Model summary

Train set
Accuracy = 0.84
Sensitivity = 0.81
Specificity = 0.83
Test set
Accuracy = 0.78
Sensitivity = 0.82
Specificity = 0.78
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 explains the important predictor variables as well as the significance of each variable. The model also helps 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.

Conclusion

Top predictors

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 probability.

- 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.

Conclusion

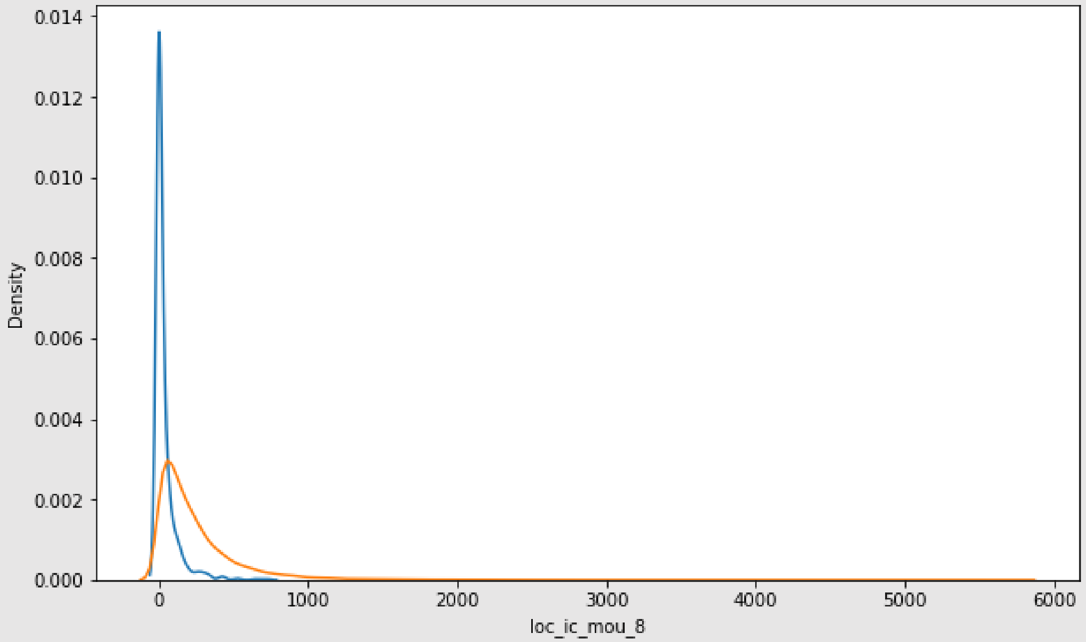
Recommendations to predict the churn customers and for better business:

- 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 in 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.
- Customers, who's 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.
- Customers 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.
- For the customers classified as a probable churn, provide the customers with attractive offers they cannot resist and retain them.
- Provide offers on long term plans so that the customer would be loyal.
- Provide the customers offers based on their usage and profile.

Conclusion

❖ **Plotting important predictors for churn and non churn customers**

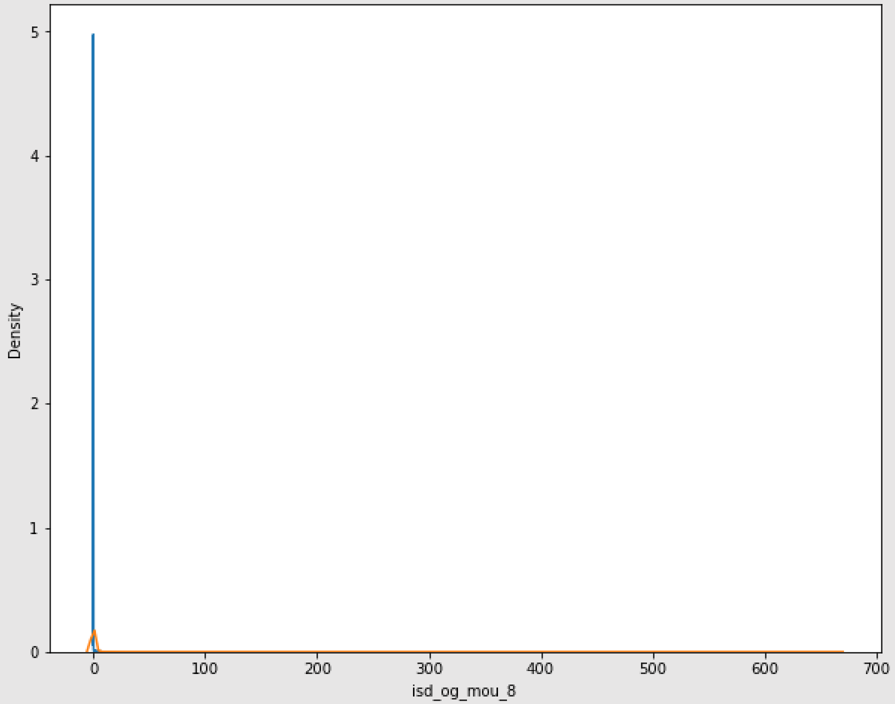
➢ **Plotting loc_ic_mou_8 predictor for churn and not churn customers**



We can see that for the churn customers the minutes of usage for the month of August is mostly populated on the lower side than the non churn customers.

Conclusion

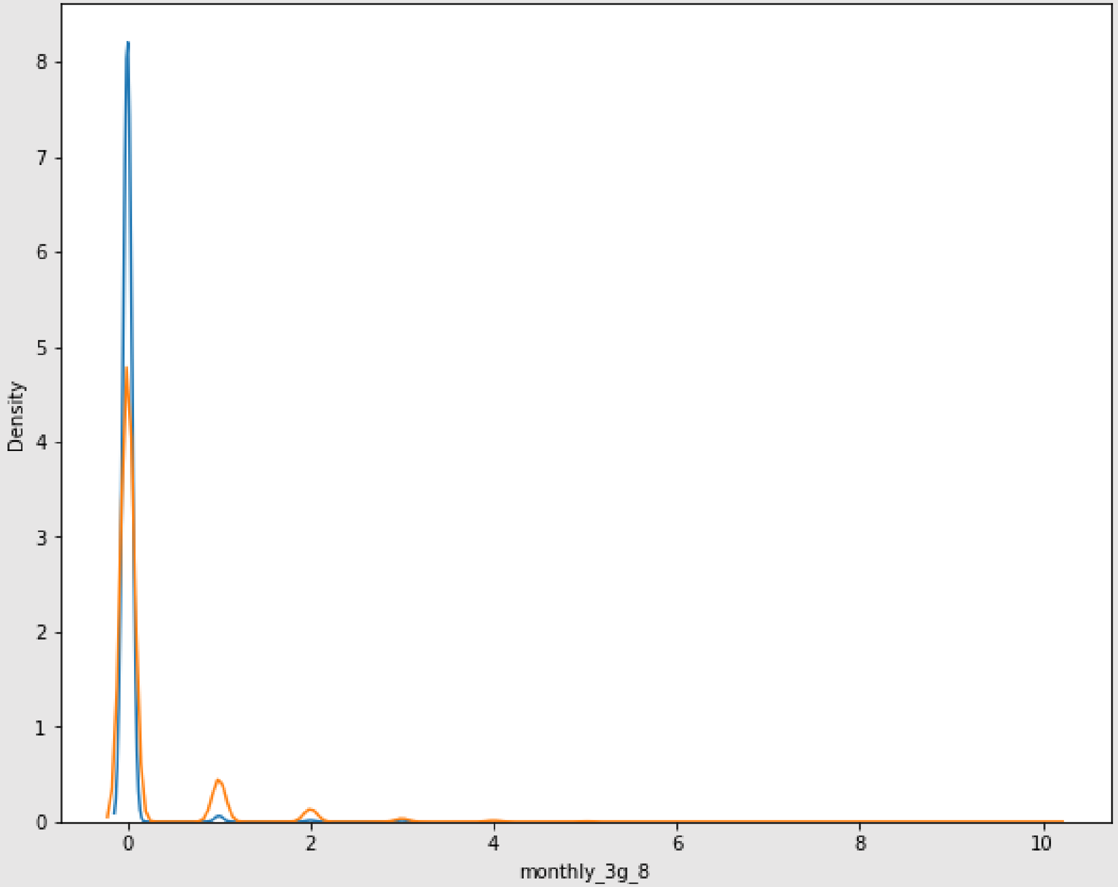
➢ **Plotting isd_og_mou_8 predictor for churn and not churn customers**



We can see that the ISD outgoing minutes of usage for the month of August for churn customers is densed approximately to zero. On the other hand for the non churn customers it is little more than the churn customers

Conclusion

➢ **Plotting monthly_3g_8 predictor for churn and not churn customers**



The number of monthly 3g data for August for the churn customers are very much around 1, whereas of non churn customers it spreaded across various numbers.

Similarly, we can plot each variables, which have higher coefficients, churn distribution.