Telecom Churn Case Study

Rishi, Sai, Aijazuddin

Introduction & Problem statement

- 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.**
- Analyze 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.

Mandatory Data Preparation steps

Filter high-value customers

- Churn only for high-value customers need to be predicted. Define high-value customers as follows: Those who have recharged with an amount more than or equal to X, where X is the **70th percentile** of the average recharge amount in the first two months (the good phase).

Tag churners and remove attributes of the churn phase

- Now tag the churned customers (churn=1, else 0) based on the fourth month as follows: Those who have not made any calls (either incoming or outgoing) AND have not used mobile internet even once in the churn phase. The attributes you need to use to tag churners are:
  - total_ic_mou_9
  - total_og_mou_9
  - vol_2g_mb_9
  - vol_3g_mb_9
- After tagging churners, **remove all the attributes corresponding to the churn phase** (all attributes having '_9', etc. in their names).

Data Preparation

Handle Missing Data

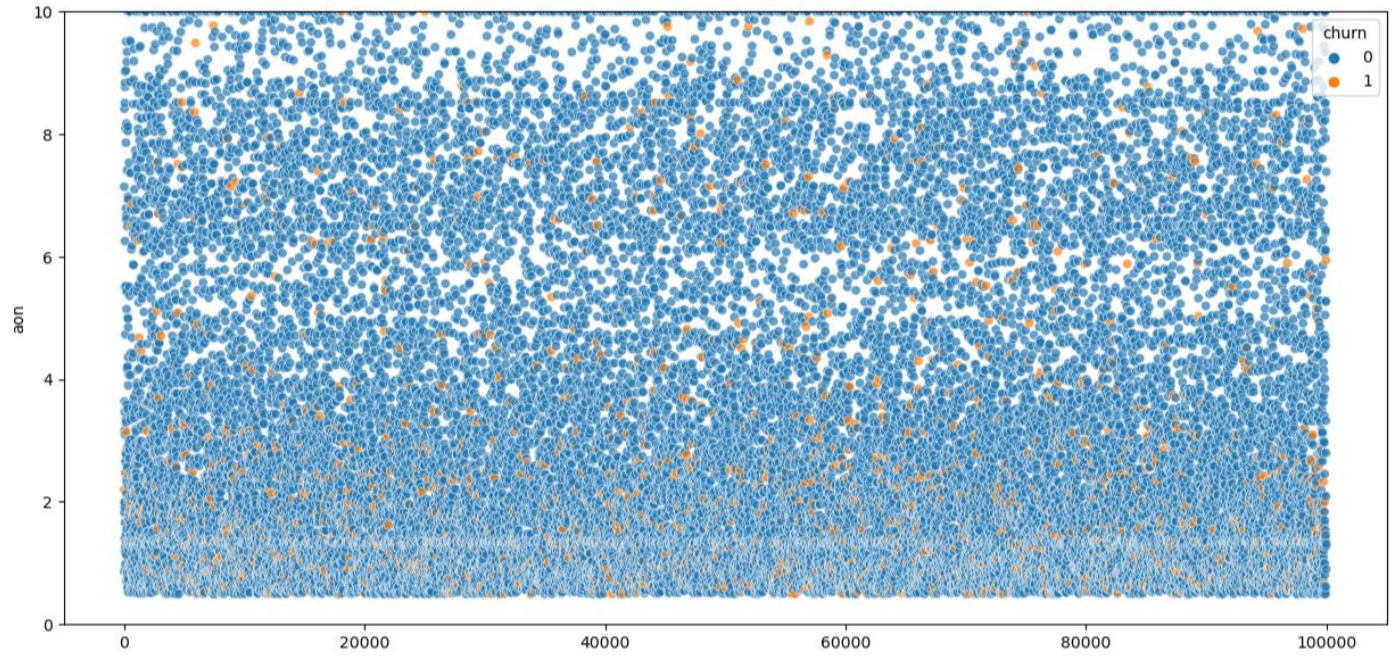
- Drop all columns with more than 40% missing data
- Drop all unnecessary columns like the dates which are unnecessary
- Drop all rows with columns having missing data less than 5%
- Drop unnecessary columns like circle_id
- Drop all columns with no variance at all i.e. min and max values as 0
- Drop all columns with low variance i.e. 75% percentile is 0 for all the months i.e. 6,7,8
- Drop the columns that are highly correlated with a threshold of 0.8

Outlier Treatment

- Several columns have a huge difference between the 75% percentile and max values and hence the outlier treatment is done by capping the columns with 90% percentile + 1.5 times the IQR and 10% percentile -1.5 times the IQR

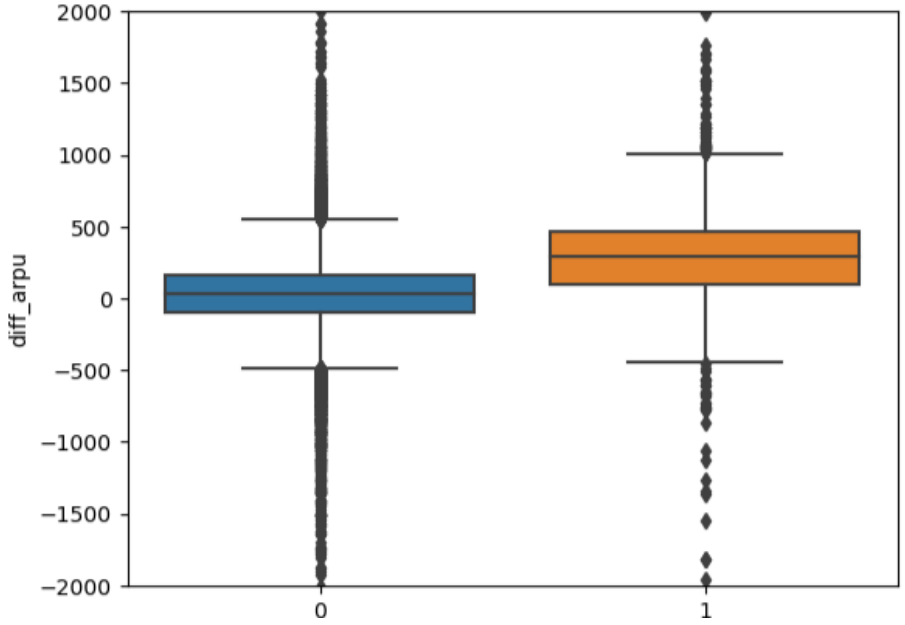
EDA

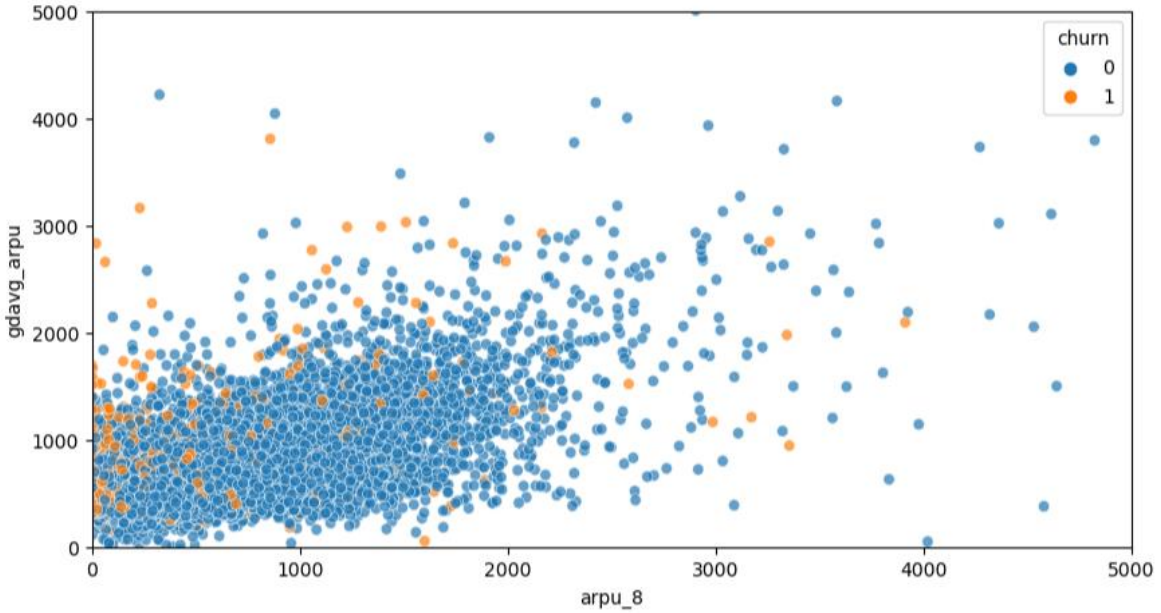
Most of the churners are of the age with the operator less than 3 years



EDA

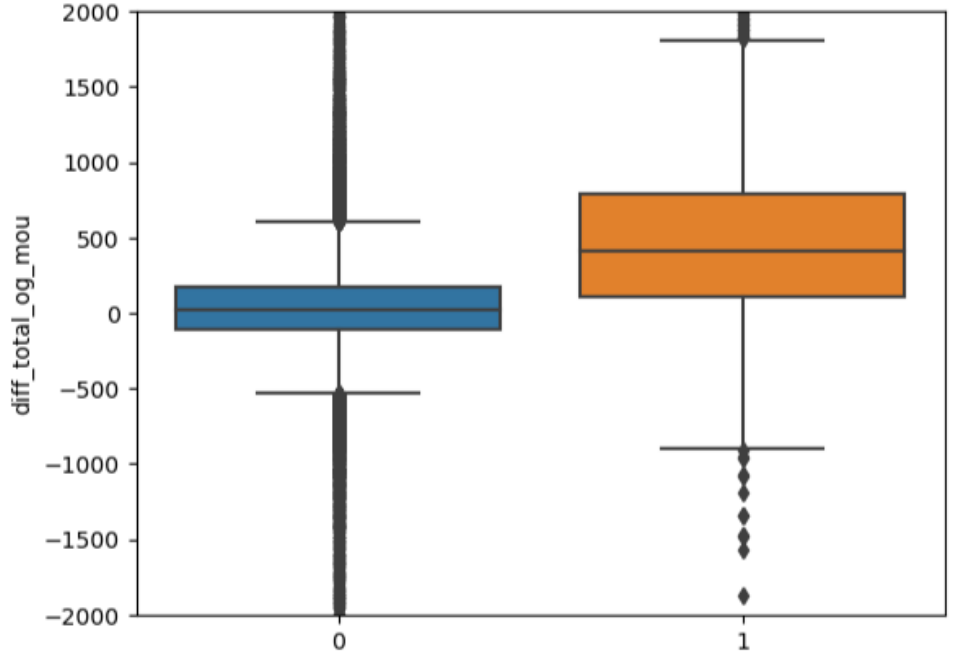
Most of the churners have a high difference in arpu of the action phase vs the average arpu of good phase

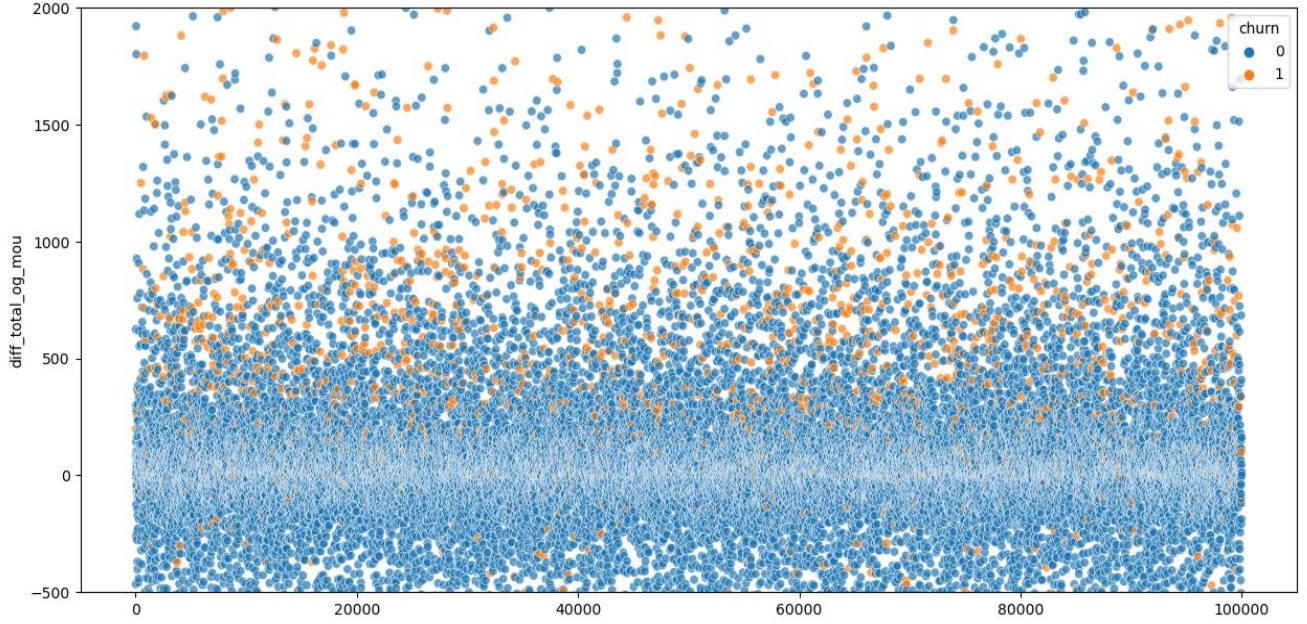




EDA

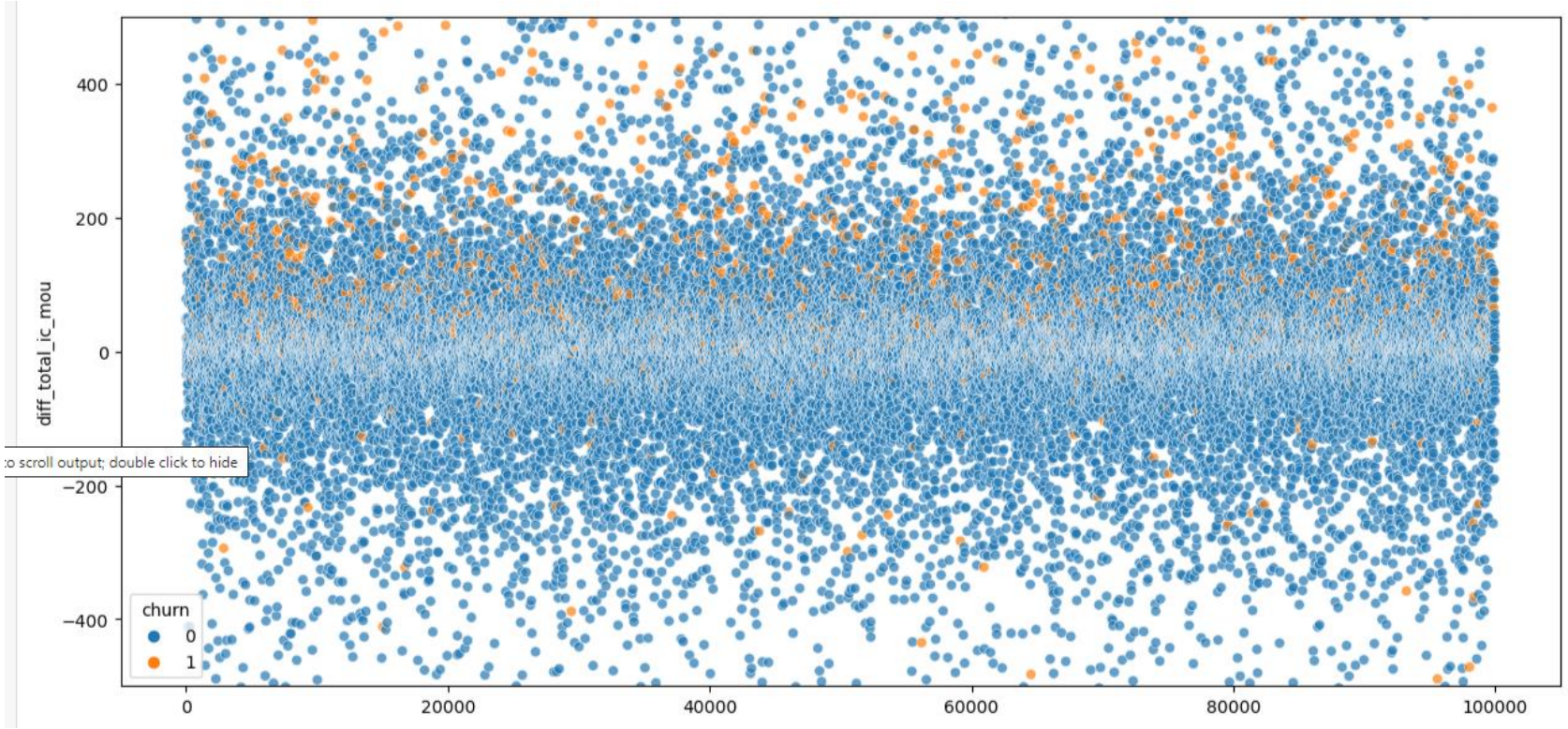
Most of the churners have a high difference in total outgoing mou of the action phase vs the average total outgoing mou of good phase





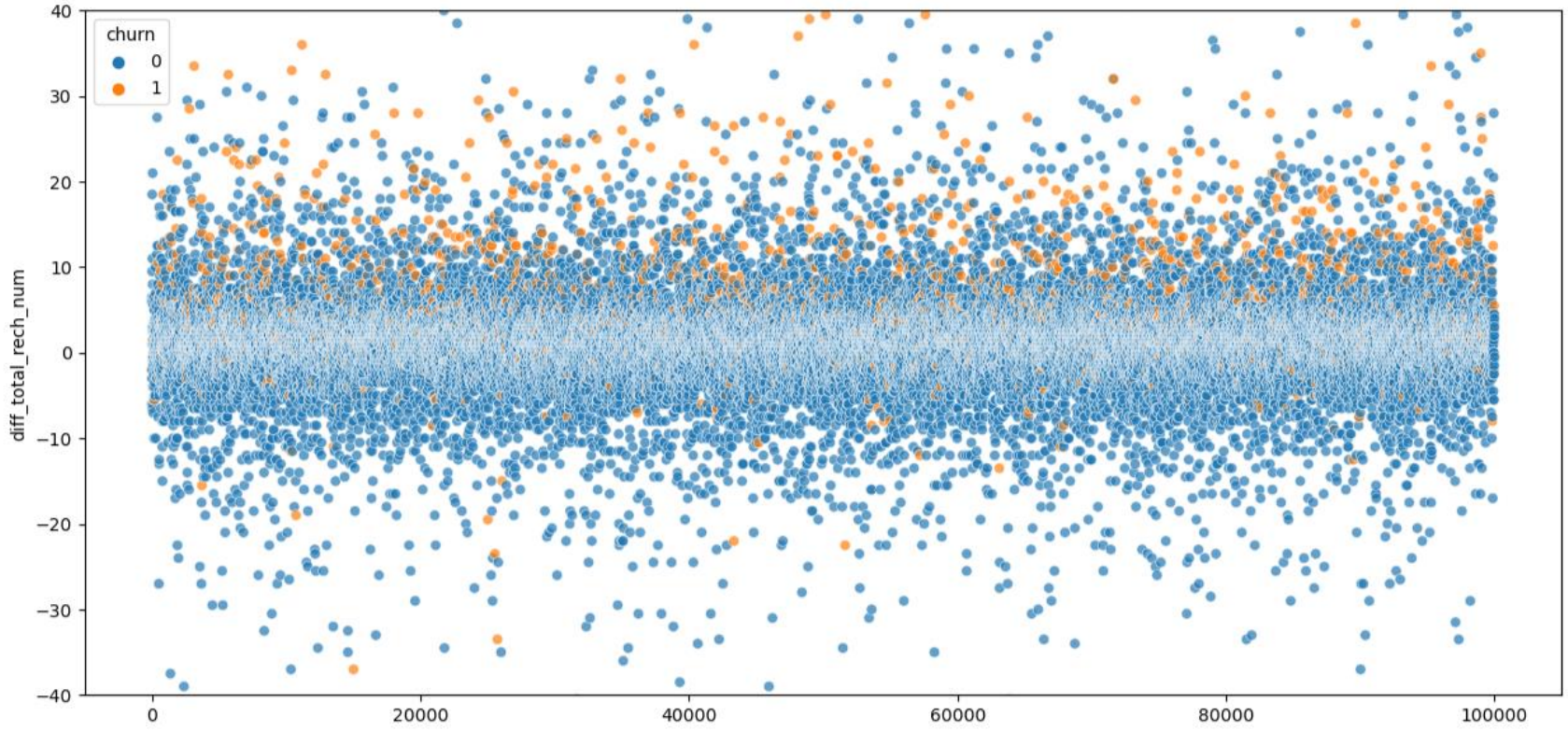
EDA

Most of the churners have a high difference in total ingoing mou of the action phase vs the average total ingoing mou of good phase



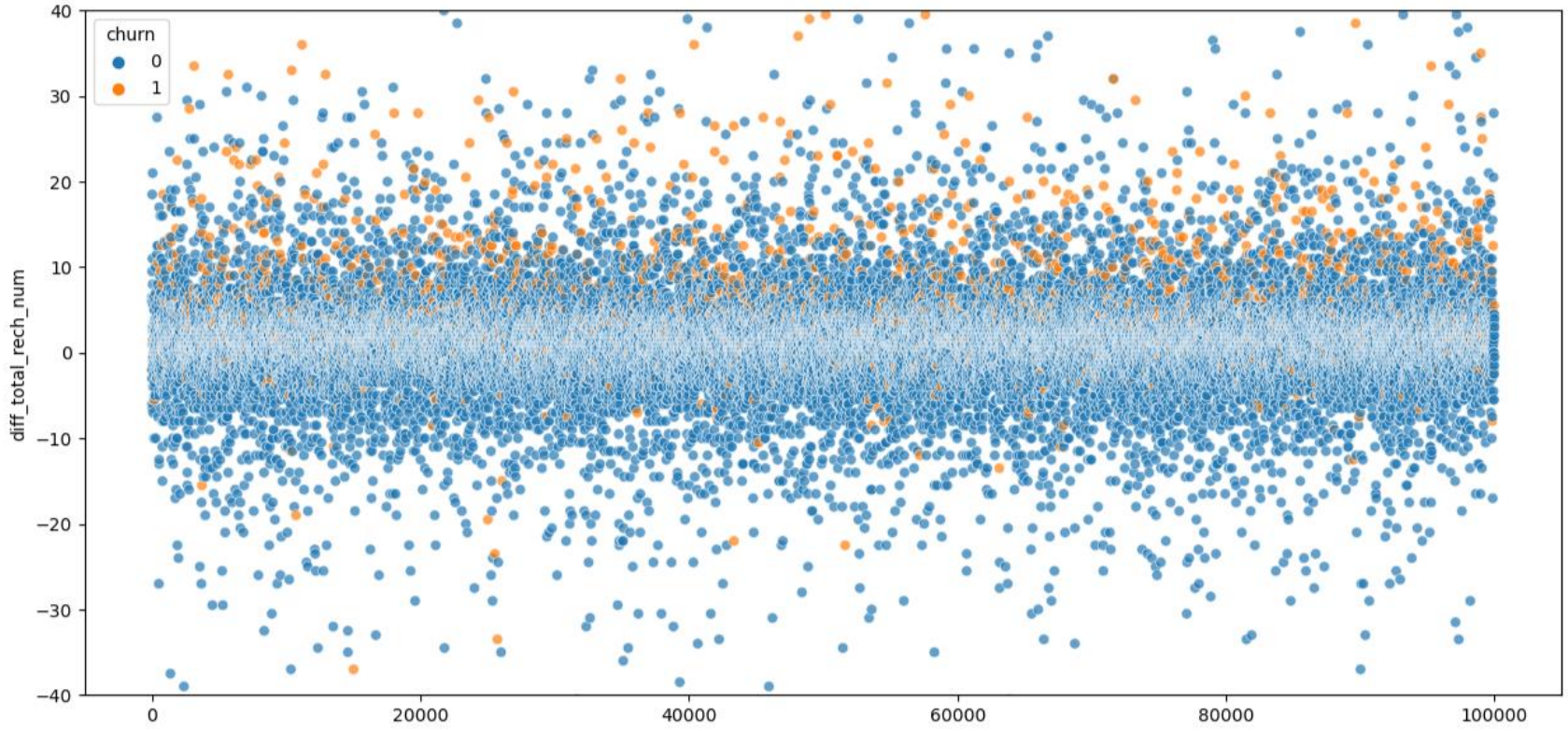
EDA

Most of the churners have a high difference in total recharge number of the action phase vs the average total recharge number of good phase



EDA

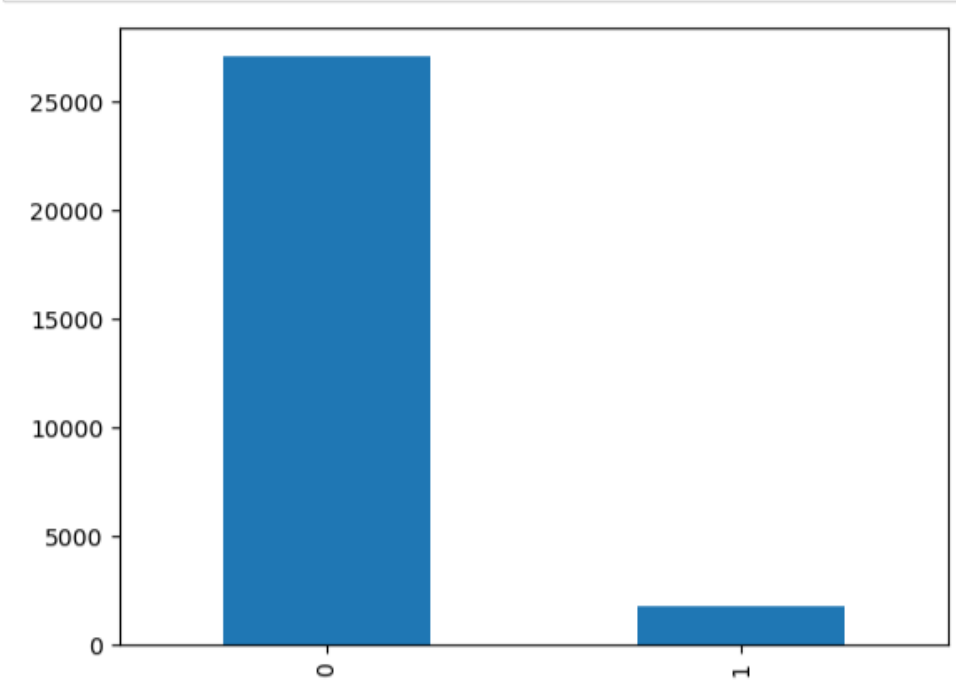
Most of the churners have a high difference in total recharge amt of the action phase vs the average total recharge amt of good phase



Class Imbalance

Identify Class imbalance

- The data has a high-class imbalance. This has been handled using oversampling technique SMOTE



Scaling

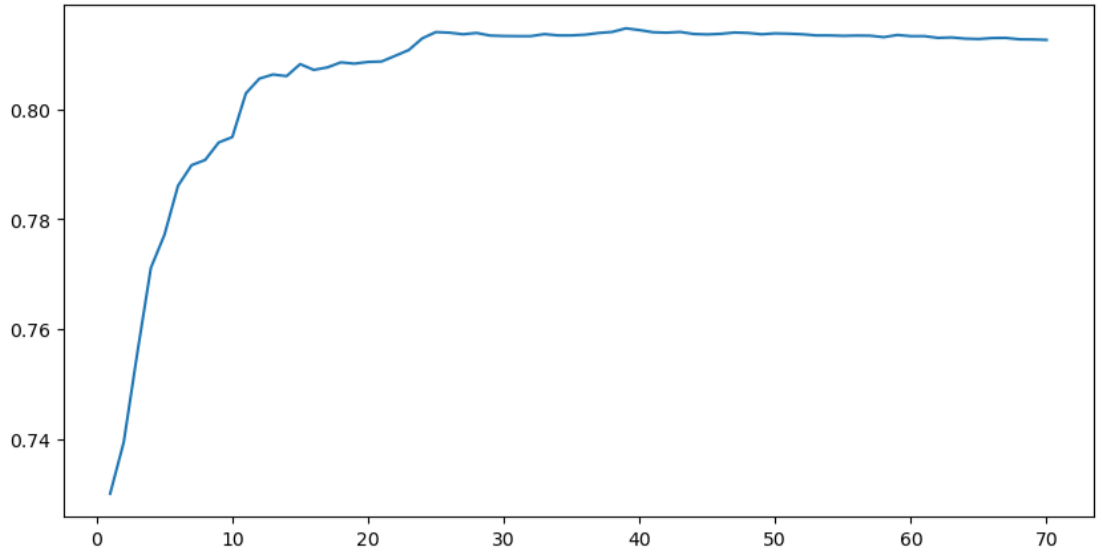
Scaling

- To scale the parameters Min Max scaling technique is used.

Model Building – Logistics Regression

Preparation for Building Logistic Regression Model

- To find out the optimal number of parameters RFECV technique is used. Accuracy remains stable after 30 parameters. 30 parameters will be selected for RFE



Model Building – Logistics Regression

Preparation for Building Logistic Regression Model

- To find out the 30 parameters, RFE technique is used.

```
In [338]: X_train.columns[rfe.support_]

Out[338]: Index(['onnet_mou_6', 'onnet_mou_7', 'offnet_mou_6', 'offnet_mou_7',
                 'loc_og_t2t_mou_7', 'loc_og_t2t_mou_8', 'loc_og_t2m_mou_6',
                 'loc_og_t2m_mou_7', 'loc_og_t2m_mou_8', 'loc_og_mou_6', 'loc_og_mou_7',
                 'total_og_mou_6', 'total_og_mou_7', 'total_og_mou_8',
                 'loc_ic_t2f_mou_7', 'loc_ic_mou_6', 'loc_ic_mou_7', 'loc_ic_mou_8',
                 'std_ic_mou_6', 'std_ic_mou_8', 'ic_others_8', 'total_rech_num_7',
                 'total_rech_num_8', 'total_rech_amt_7', 'total_rech_amt_8',
                 'last_day_rch_amt_8', 'vol_2g_mb_8', 'vol_3g_mb_7', 'vol_3g_mb_8',
                 'aon'],
                dtype='object')
```

Model Building – Logistics Regression(Sklearn)

Building Logistic Regression Model

- Model is built using Regressor from sklearn package

Train Data Metrics

```
In [345]: print(classification_report(y_train, y_train_pred))
```

	precision	recall	f1-score	support
0	0.83	0.79	0.81	18937
1	0.80	0.84	0.82	18937
accuracy			0.81	37874
macro avg	0.81	0.81	0.81	37874
weighted avg	0.81	0.81	0.81	37874

Test Data Metrics

```
In [661]: # Accuracy with precision and recall values
          print(classification_report(y_test, y_test_pred))
```

	precision	recall	f1-score	support
0	0.98	0.79	0.88	8117
1	0.20	0.79	0.32	542
accuracy			0.79	8659
macro avg	0.59	0.79	0.60	8659
weighted avg	0.93	0.79	0.84	8659

Model Building – Logistics Regression(Statsmodels)

Building Logistic Regression Model

- Model is built using Regressor GLM from statsmodel
- Several times the model is re-built by dropping parameters having a high VIF score i.e. multicollinearity and/or significance of the parameter using the p-value. The final model summary is on next slide
- Using the various curves a threshold of 0.53 is determined to predict the churners

Model Building – Logistics Regression(Statsmodels)

Generalized Linear Model Regression Results

Dep. Variable:	churn	**No. Observations:**	37874
Model:	GLM	**Df Residuals:**	37855
Model Family:	Binomial	**Df Model:**	18
Link Function:	Logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-16990.
Date:	Sun, 09 Jul 2023	**Deviance:**	33980.
Time:	16:35:37	**Pearson chi2:**	5.53e+04
No. Iterations:	6	**Pseudo R-squ. (CS):**	0.3868
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	1.2139	0.035	34.310	0.000	1.145	1.283
onnet_mou_6	0.9912	0.101	9.771	0.000	0.792	1.190
onnet_mou_7	2.1501	0.120	17.915	0.000	1.915	2.385
offnet_mou_7	3.5307	0.100	35.234	0.000	3.334	3.727
loc_og_t2t_mou_8	-1.9999	0.120	-16.660	0.000	-2.235	-1.765
loc_og_t2m_mou_6	-0.6085	0.140	-4.333	0.000	-0.884	-0.333
loc_og_t2m_mou_8	-3.6734	0.194	-18.938	0.000	-4.054	-3.293
total_og_mou_8	-5.4989	0.163	-33.780	0.000	-5.818	-5.180
loc_ic_t2f_mou_7	-1.2710	0.096	-13.225	0.000	-1.459	-1.083
loc_ic_mou_6	0.2948	0.118	2.507	0.012	0.064	0.525
std_ic_mou_6	1.1919	0.093	12.783	0.000	1.009	1.375
std_ic_mou_8	-2.2877	0.110	-20.836	0.000	-2.503	-2.072
ic_others_8	-0.7093	0.072	-9.872	0.000	-0.850	-0.569
total_rech_num_8	-1.3991	0.108	-12.928	0.000	-1.611	-1.187
last_day_rch_amt_8	-2.2632	0.102	-22.114	0.000	-2.464	-2.063
vol_2g_mb_8	-1.6457	0.079	-20.835	0.000	-1.801	-1.491
vol_3g_mb_7	1.6215	0.110	14.767	0.000	1.406	1.837
vol_3g_mb_8	-3.7100	0.155	-23.994	0.000	-4.013	-3.407
aon	-1.1194	0.069	-16.167	0.000	-1.255	-0.984

Model Building – Logistics Regression(Statsmodels)

- Model evaluation

Train Data Metrics

	precision	recall	f1-score	support
0	0.83	0.79	0.81	18937
1	0.80	0.84	0.82	18937
accuracy			0.81	37874
macro avg	0.81	0.81	0.81	37874
weighted avg	0.81	0.81	0.81	37874

Test Data Metrics

	precision	recall	f1-score	support
0	0.98	0.80	0.88	8117
1	0.20	0.76	0.32	542
accuracy			0.80	8659
macro avg	0.59	0.78	0.60	8659
weighted avg	0.93	0.80	0.85	8659

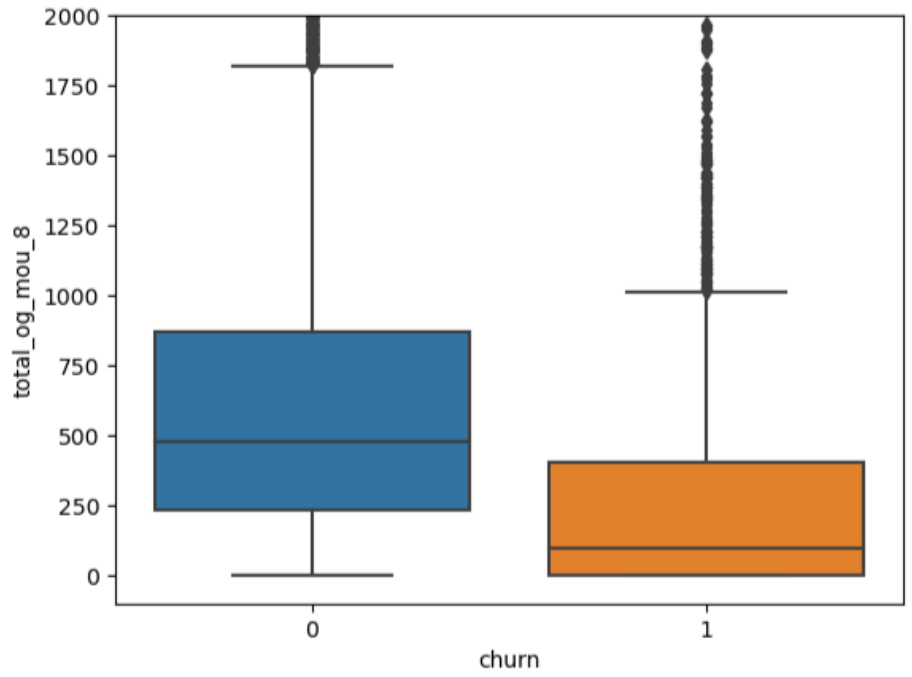
Top parameters affecting Churn

- The top parameters are
  - total_og_mou_8
  - loc_og_t2m_mou_8
  - std_ic_mou_8
  - last_day_rch_amt_8

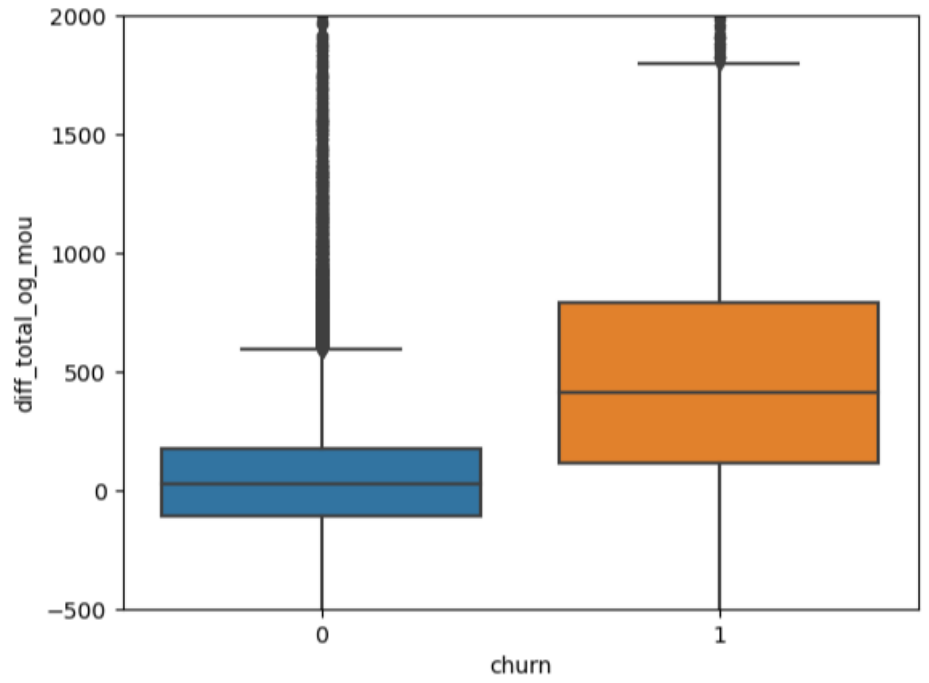
Top parameters affecting Churn

- The top parameters are
  - total_og_mou_8

Actual Action phase Vs Churn



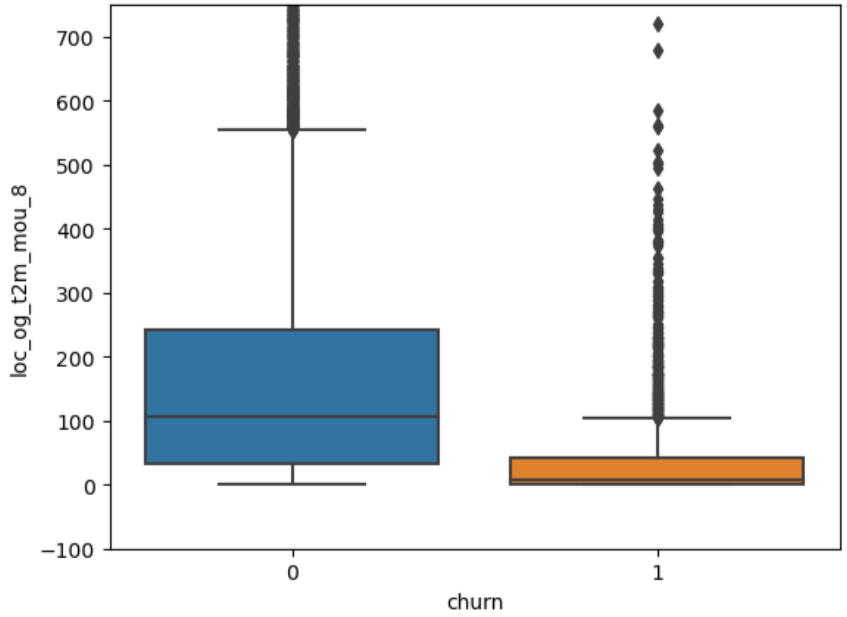
Avg of Good phase-Action phase Vs Churn



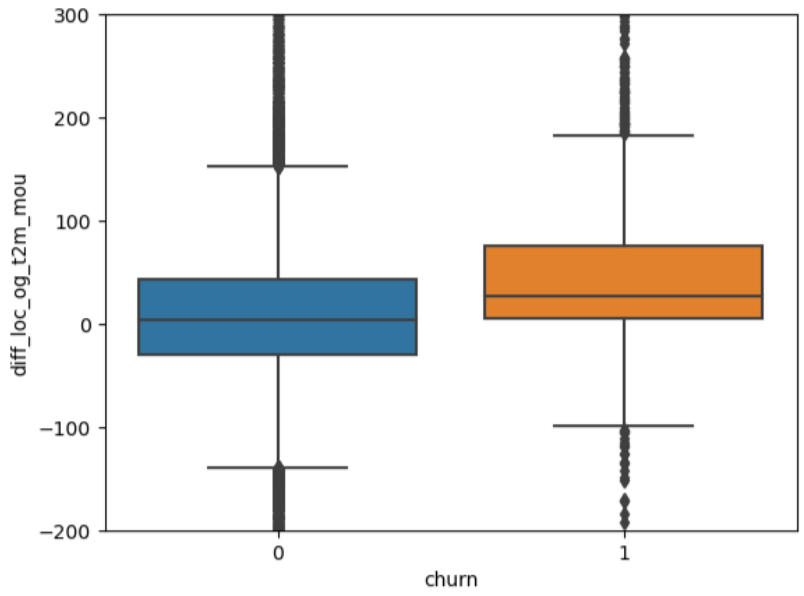
Top parameters affecting Churn

- The top parameters are
  - loc_og_t2m_mou_8

Actual Action phase Vs Churn



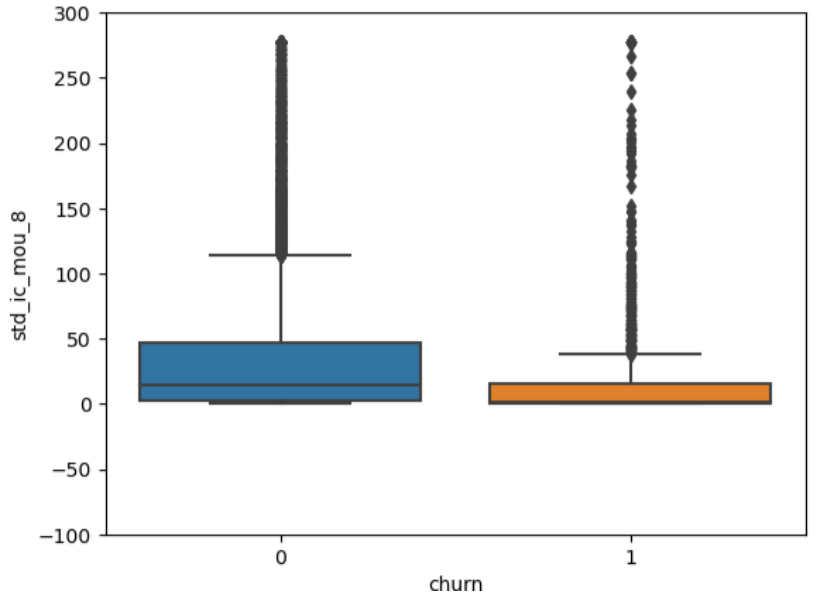
Avg of Good phase-Action phase Vs Churn



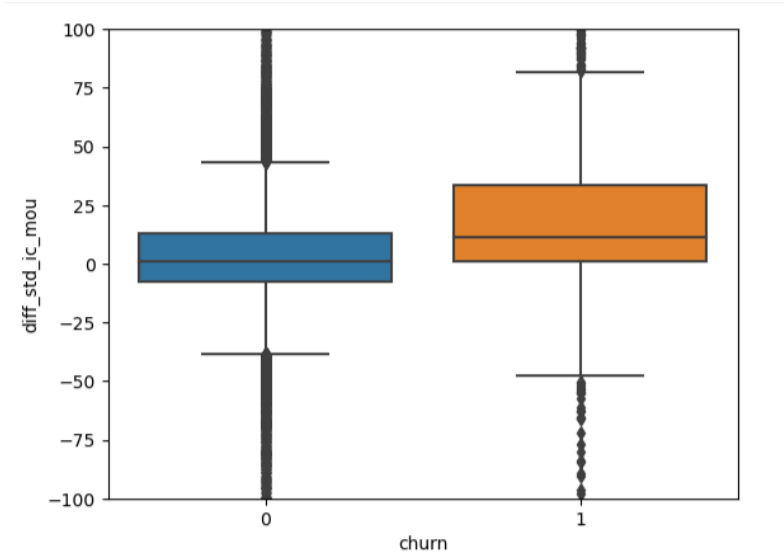
Top parameters affecting Churn

- The top parameters are
  - std_ic_mou_8

Actual Action phase Vs Churn



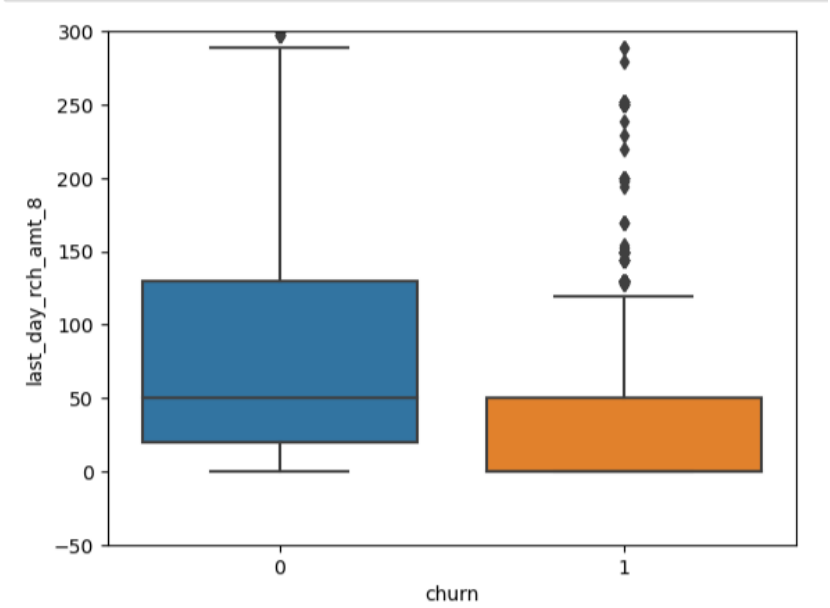
Avg of Good phase-Action phase Vs Churn



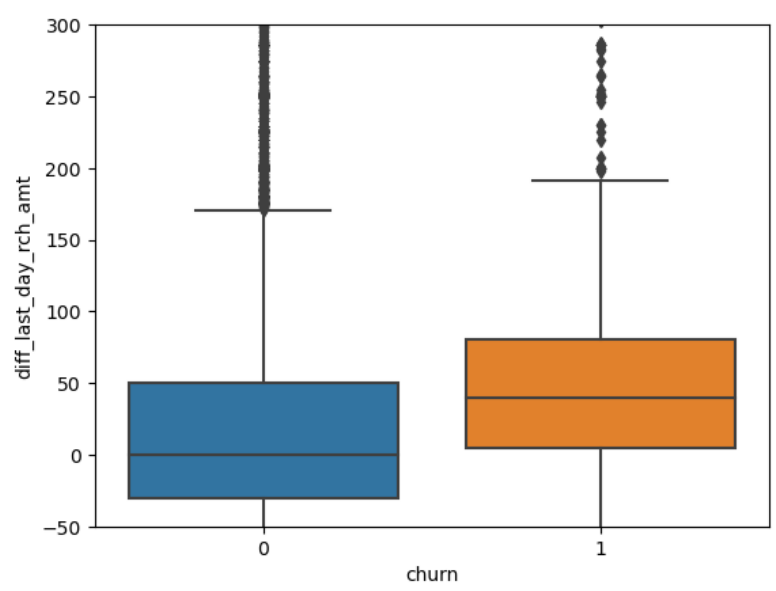
Top parameters affecting Churn

- The top parameters are
  - last_day_rch_amt_8

Actual Action phase Vs Churn



Avg of Good phase-Action phase Vs Churn



Model Building – Random forests

Building Random forests model

- Built an initial model of Random forests which gave an accuracy of 100% probably this is overfitting but gives an accuracy of 93%
- Subsequently using a Randomized CV method to get the best model i.e. hyperparameter tuning
- The best model parameters are

```
RandomForestClassifier
RandomForestClassifier(max_depth=14, max_features=17, min_samples_leaf=20,
                       n_estimators=90, n_jobs=-1, random_state=42)
```

Model Building – Random Forests(after HPT)

- Model evaluation

Train Data Metrics

	precision	recall	f1-score	support
0	0.96	0.94	0.95	18937
1	0.95	0.96	0.95	18937
accuracy			0.95	37874
macro avg	0.95	0.95	0.95	37874
weighted avg	0.95	0.95	0.95	37874

Test Data Metrics

	precision	recall	f1-score	support
0	0.97	0.92	0.95	8117
1	0.34	0.61	0.44	542
accuracy			0.90	8659
macro avg	0.66	0.76	0.69	8659
weighted avg	0.93	0.90	0.91	8659

Model Building – Decision Tree

Building Decision Tree model

- Built an initial model of Decision Trees which gave an accuracy of 91% but gives an accuracy of 83%
- Subsequently using a Grid Search CV method to get the best model i.e. hyperparameter tuning
- The best model parameters are

```
DecisionTreeClassifier
DecisionTreeClassifier(max_depth=20, min_samples_leaf=5, random_state=42)
```

Model Building – Decision Trees(after HPT)

- Model evaluation

Train Data Metrics

	precision	recall	f1-score	support
0	0.97	0.96	0.96	18937
1	0.96	0.97	0.96	18937
accuracy			0.96	37874
macro avg	0.96	0.96	0.96	37874
weighted avg	0.96	0.96	0.96	37874

Test Data Metrics

	precision	recall	f1-score	support
0	0.96	0.88	0.92	8117
1	0.20	0.45	0.27	542
accuracy			0.85	8659
macro avg	0.58	0.66	0.60	8659
weighted avg	0.91	0.85	0.88	8659

Model Selection (Metrics based on Test Data)

Logistics Regression - Sklearn

	precision	recall	f1-score	support
0	0.98	0.79	0.88	8117
1	0.20	0.79	0.32	542
accuracy			0.79	8659
macro avg	0.59	0.79	0.60	8659
weighted avg	0.93	0.79	0.84	8659

Logistics Regression -GLM

	precision	recall	f1-score	support
0	0.98	0.80	0.88	8117
1	0.20	0.76	0.32	542
accuracy			0.80	8659
macro avg	0.59	0.78	0.60	8659
weighted avg	0.93	0.80	0.85	8659

Random Forests

	precision	recall	f1-score	support
0	0.97	0.92	0.95	8117
1	0.34	0.61	0.44	542
accuracy			0.90	8659
macro avg	0.66	0.76	0.69	8659
weighted avg	0.93	0.90	0.91	8659

Decision Tree

	precision	recall	f1-score	support
0	0.96	0.88	0.92	8117
1	0.20	0.45	0.27	542
accuracy			0.85	8659
macro avg	0.58	0.66	0.60	8659
weighted avg	0.91	0.85	0.88	8659

Summary

- Both Logistic Regression and Random Forests have performed well.
- If recall is more important than precision then Logistics Regression should be selected
- If more balance is needed on recall and precision then Random forest should be selected.

Recommendations to Business

- Logistics Regression should be selected as it has a better recall i.e. identifies most churners with additional non churners as well.
- Once churners are identified they should be given extra MBs, calling minutes/texts for keeping the customer. The customers who were not going to churn and get the freebies will also become less likely to churn in future.
- The Business should concentrate on parameters like reduction of outgoing calls, std incoming calls, local outgoing calls to other operator mobile, recharge amount on last day to monitor the customers and proactively offer them promotions.