Problem Statement

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

Business Objective

•The Business objective here is to reduce customer churn, Retain high profitable customer and to identify which customers are at high risk of churn so that corrective action can be taken place

Understanding and Defining Churn

- Postpaid churn - In the postpaid model, when customers want to switch to another operator, they usually inform the existing operator to terminate the services, and you directly know that this is an instance of churn.
- Prepaid Churn - In Prepaid Churn Customers who want to switch to another network can simply stop using the services without any notice, and it is hard to know whether someone has actually churned or is simply not using the services temporarily (e.g. someone may be on a trip abroad for a month or two and then intend to resume using the services again).

Types of Churn

- **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.

- **Usage based Churn** - Customers who have not done any usage, either incoming or outgoing - in terms of calls, internet etc. over a period of time.

A potential shortcoming of this definition is that when the customer has stopped using the services for a while, it may be too late to take any corrective actions to retain them. For e.g., if you define churn based on a ‘two-months zero usage’ period, predicting churn could be useless since by that time the customer would have already switched to another operator.

* **In this project we will be targeting on Usage Based Churn.**

High Value Churn :

In the **Indian and Southeast Asian markets**, approximately **80%** of revenue comes from the top **20%** of customers (called high-value customers). Thus, if we can reduce the churn of high-value customers, we will be able to reduce significant revenue leakage.

Understanding Data

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. To do this task well, understanding the typical customer behaviour during churn will be helpful.

There are three phases of the customer lifecycle:-
Good Phase - In this phase, the customer is happy with the service and behaves as usual.
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. The corrective action has to taken in this phase only
Churn Phase - In this phase, the customer is said to have churned

Importing Data and checking type of data

- Importing the data set and checking the data shape : Data shape was found (99999 , 226)
- Importing the description of the data

```
# Descriptive analysis of the dataset
inp0.describe()
```

	mobile_number	circle_id	loc_og_t2o_mou	std_og_t2o_mou	loc_ic_t2o_mou	arpu_6	arpu_7	arpu_8	arpu_9	onnet_mou_6	on
count	9.999900e+04	99999.0	98981.0	98981.0	98981.0	99999.000000	99999.000000	99999.000000	99999.000000	96062.000000	96
mean	7.001207e+09	109.0	0.0	0.0	0.0	282.987358	278.536648	279.154731	261.645069	132.395875	
std	6.956694e+05	0.0	0.0	0.0	0.0	328.439770	338.156291	344.474791	341.998630	297.207406	
min	7.000000e+09	109.0	0.0	0.0	0.0	-2258.709000	-2014.045000	-945.808000	-1899.505000	0.000000	
25%	7.000606e+09	109.0	0.0	0.0	0.0	93.411500	86.980500	84.126000	62.685000	7.380000	
50%	7.001205e+09	109.0	0.0	0.0	0.0	197.704000	191.640000	192.080000	176.849000	34.310000	
75%	7.001812e+09	109.0	0.0	0.0	0.0	371.060000	365.344500	369.370500	353.466500	118.740000	
max	7.002411e+09	109.0	0.0	0.0	0.0	27731.088000	35145.834000	33543.624000	38805.617000	7376.710000	8

- Checking the type of data by having checking info of the data type.

Data Cleaning

- Checking the missing values and dropping the missing values of data.
- After cleaning of the data the shape of the variable is (99999 , 153)

High Value Customers

High-value customers are 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).

In [172]:
```
# Once checking the dataset
inp0.head()
```

Out[172]:

	arpu_6	arpu_7	arpu_8	arpu_9	onnet_mou_6	onnet_mou_7	onnet_mou_8	onnet_mou_9	offnet_mou_6	offnet_mou_7	offnet_mou_8	offnet_mou_9	roa
0	197.385	214.816	213.803	21.100	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
1	34.047	355.074	268.321	86.285	24.11	78.68	7.68	18.34	15.74	99.84	304.76	53.76	
2	167.690	189.058	210.226	290.714	11.54	55.24	37.26	74.81	143.33	220.59	208.36	118.91	
3	221.338	251.102	508.054	389.500	99.91	54.39	310.98	241.71	123.31	109.01	71.68	113.54	
4	261.636	309.876	238.174	163.426	50.31	149.44	83.89	58.78	76.96	91.88	124.26	45.81	

In [173]:
```
# Creating the column of Total recharge amount injune & July (good_phase) = Total Data recharge amount * Average recharge amount+
inp0["Average_Amount"]=(inp0["total_rech_amt_6"]+inp0["total_rech_amt_7"])/2
```

In [174]:
```
inp0["Average_Amount"].quantile([0.7])
```

Out[174]:
```
0.7    368.5
Name: Average_Amount, dtype: float64
```

In [175]:
```
# Subsetting the data set to filtering out high value customer having Recharge amount more than 368.5
inp0=inp0[inp0.Average_Amount>=368.5]
```

In [176]:
```
# Dropping the column used to filter high value customer
inp0.drop(["total_rech_amt_6","total_rech_amt_7"], axis=1,inplace=True)
```

After same checking the shape of the data : shape comes out to be (30011 , 152)

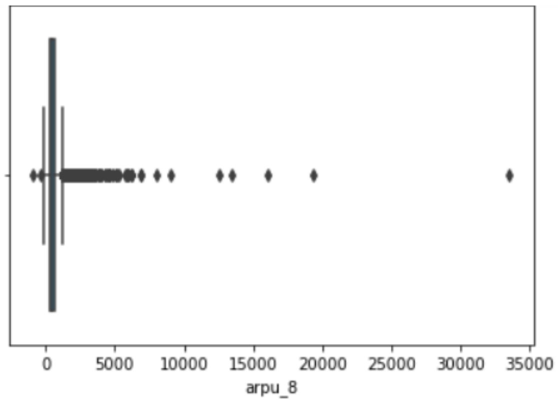
- Deriving feature Engineering by taking the average of good Phase and Later checking the shape (30011, 80).
- Checking the head :

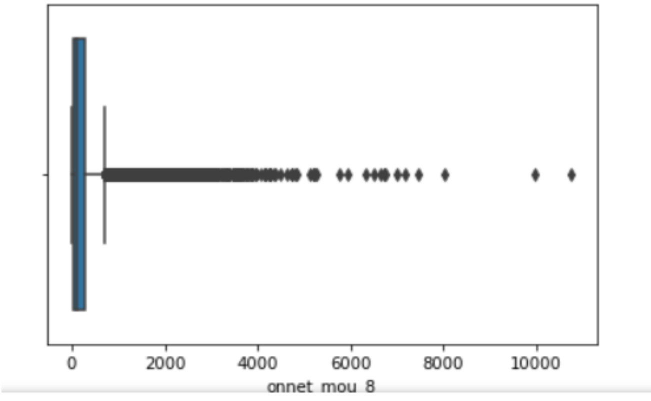
In [189]:
```
# Checking the new df
inp1.head().reset_index()
```

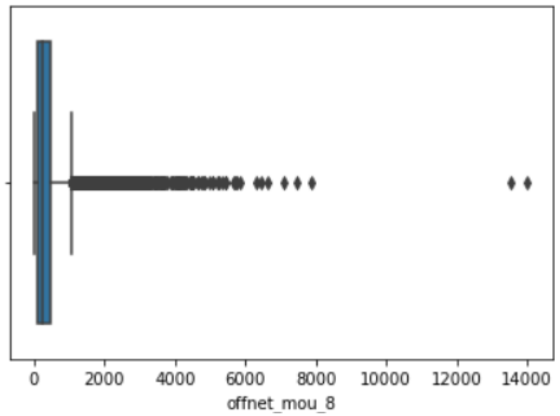
Out[189]:

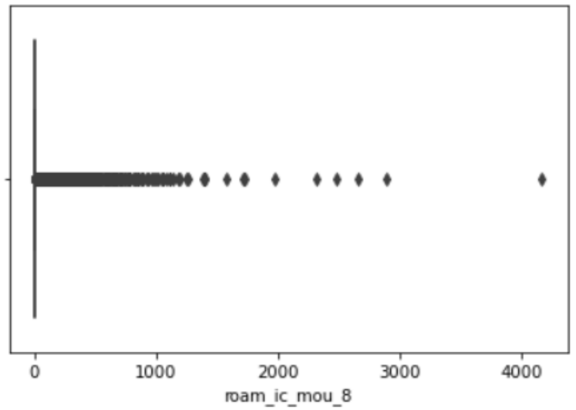
	index	arpu_8	onnet_mou_8	offnet_mou_8	roam_ic_mou_8	roam_og_mou_8	loc_og_t2t_mou_8	loc_og_t2m_mou_8	loc_og_t2f_mou_8	loc_og_t2c_mou_8
0	7	3171.480	52.29	325.91	31.64	38.06	40.28	162.28	53.23	0.00
1	8	137.362	35.08	136.48	0.00	0.00	12.49	50.54	0.00	7.15
2	13	593.260	534.24	482.46	72.11	1.44	36.01	294.46	23.51	0.49
3	16	187.894	70.61	162.76	0.00	0.00	67.38	128.28	10.26	0.00
4	17	25.499	7.79	5.54	4.81	13.34	0.00	0.00	0.00	0.00

Doing the EDA analysis

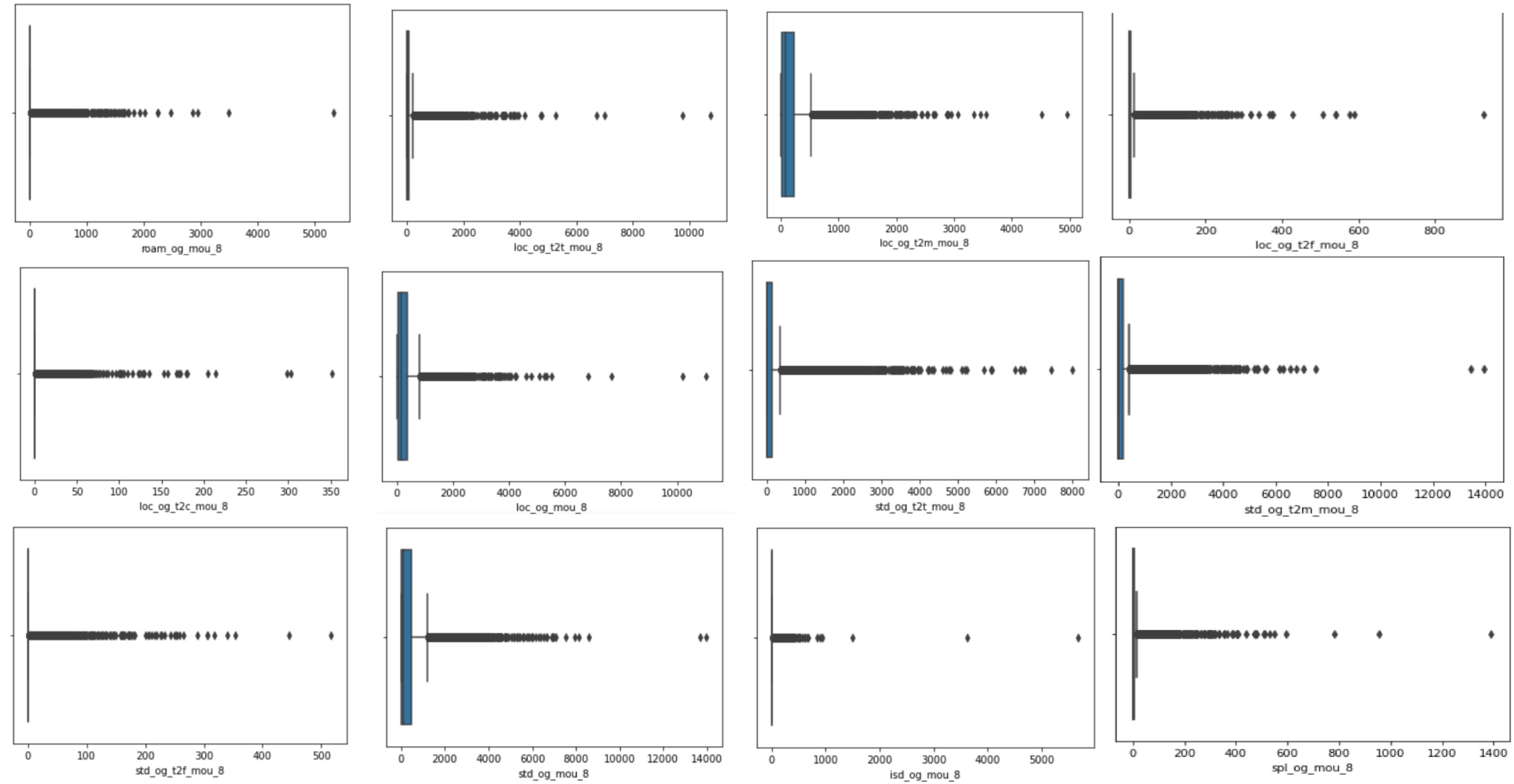




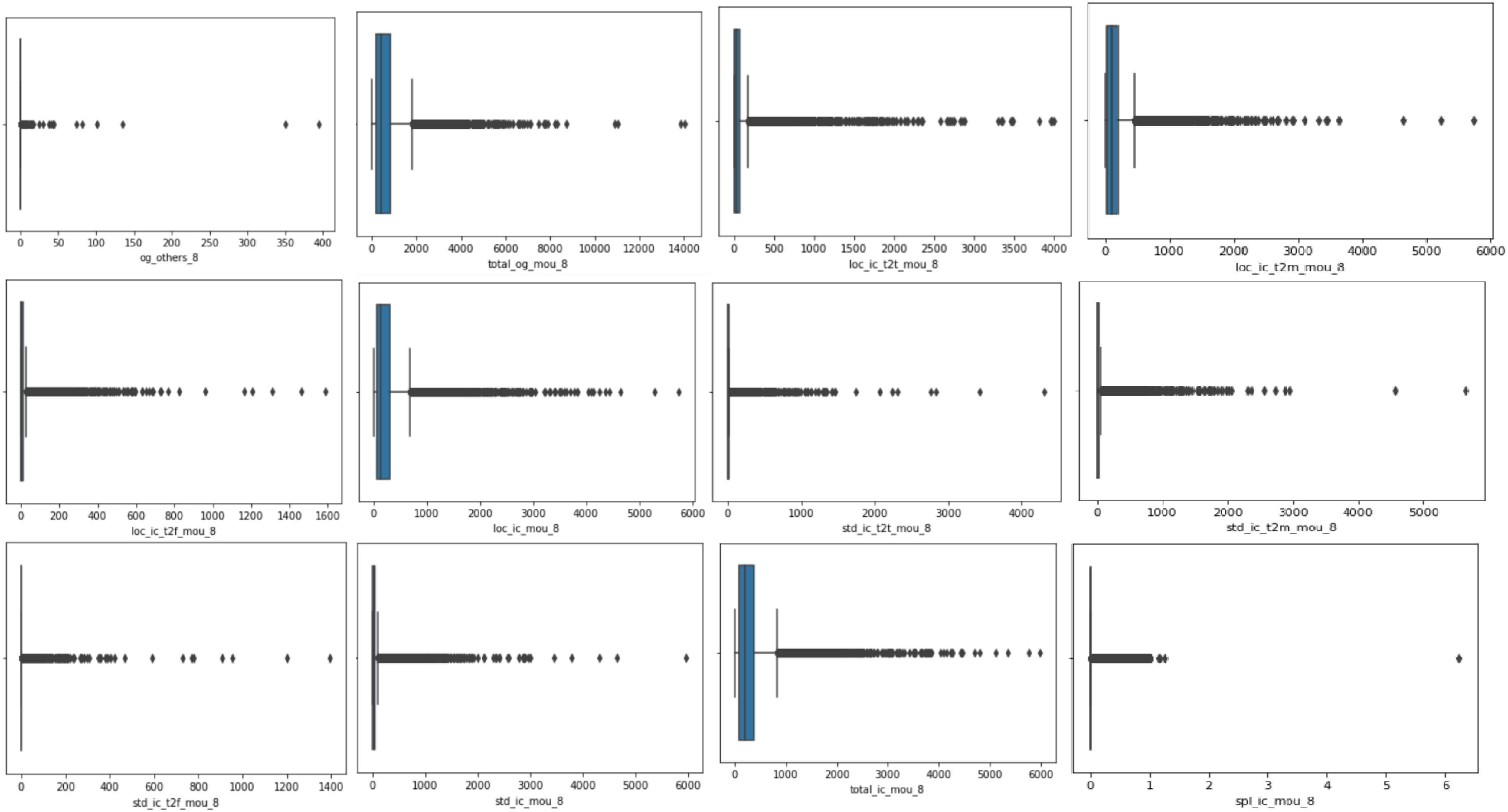




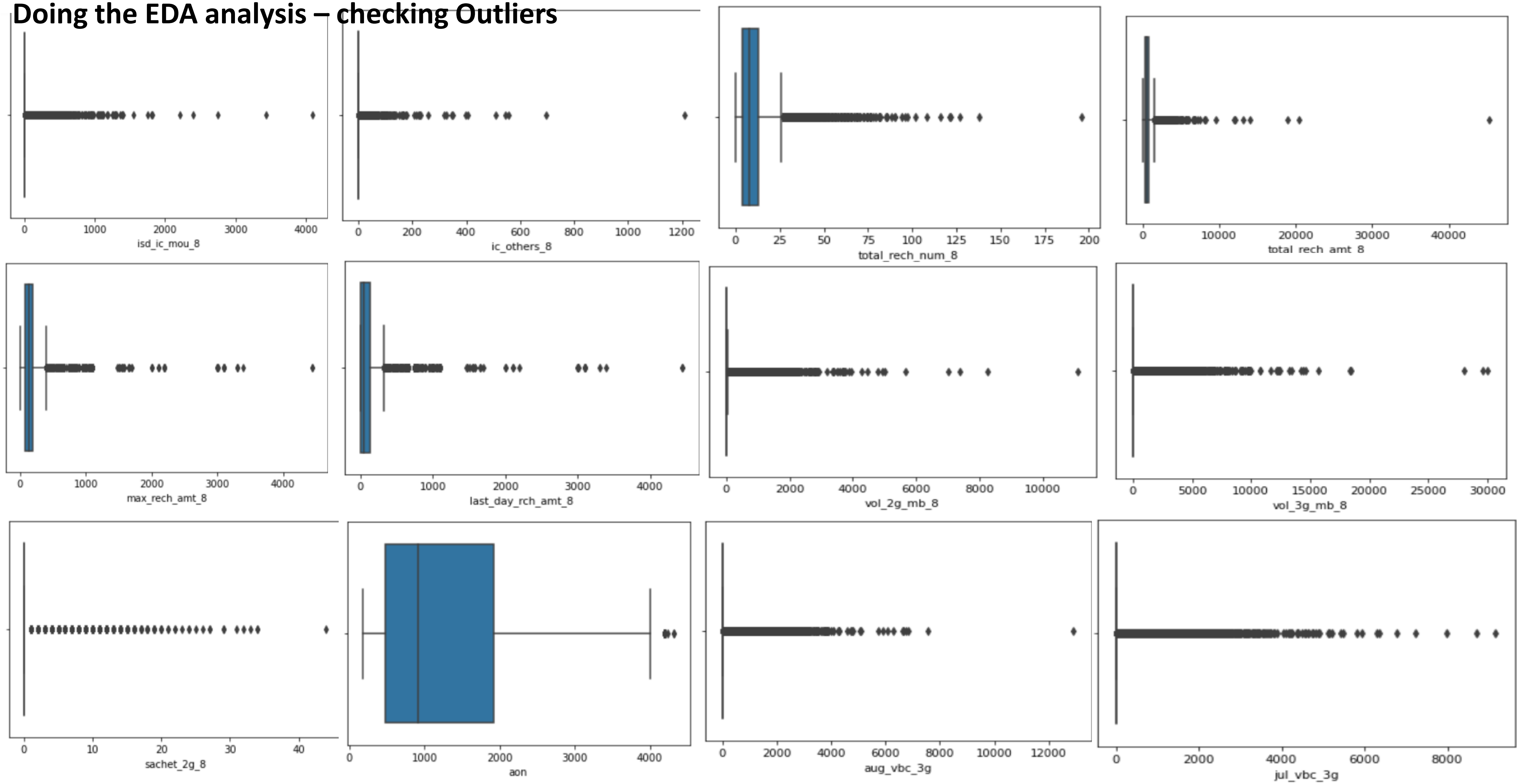
Doing the EDA analysis – checking Outliers



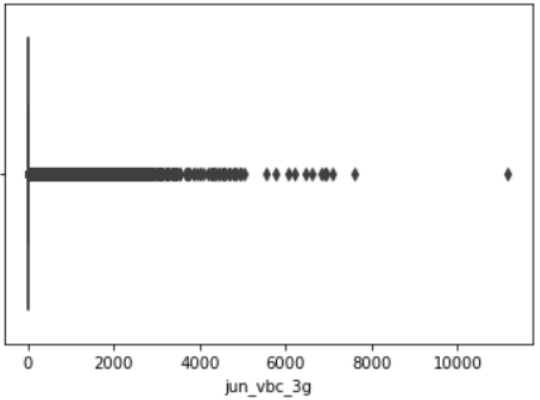
Doing the EDA analysis – checking Outliers

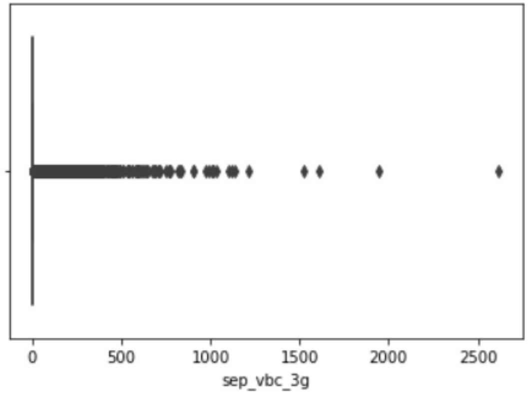


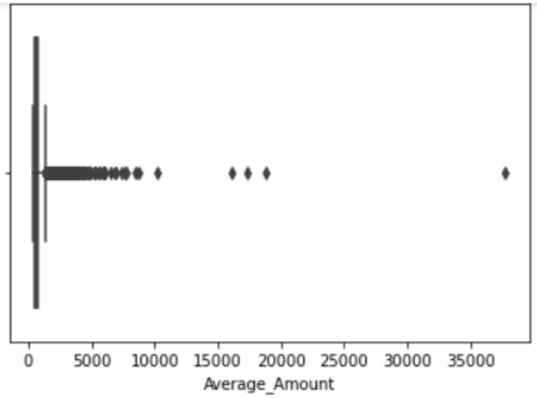
Doing the EDA analysis – checking Outliers

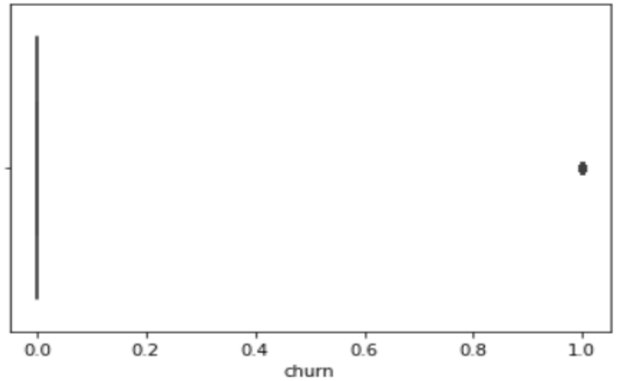


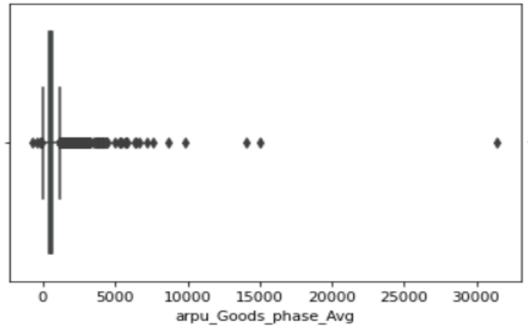
Doing the EDA analysis – checking Outliers

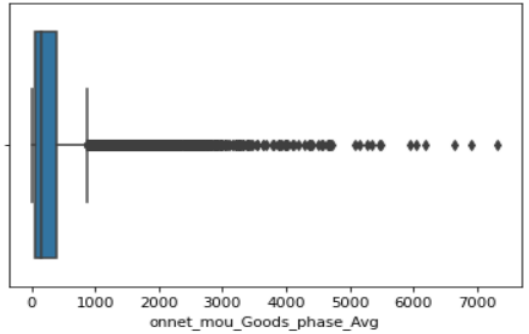


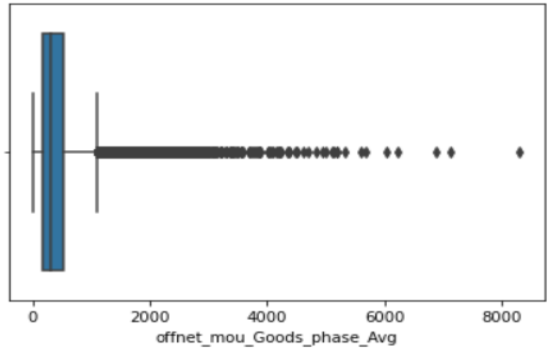


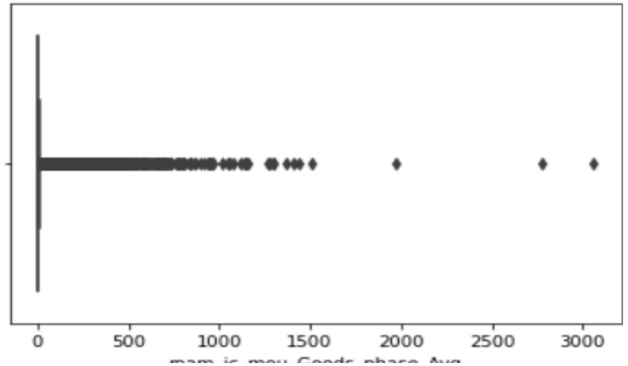


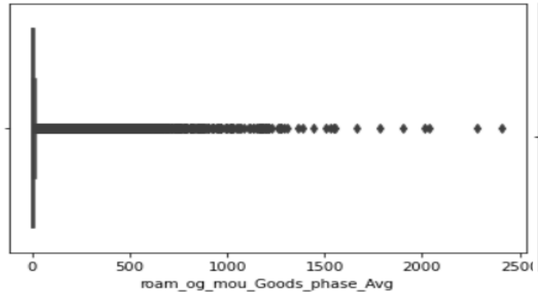


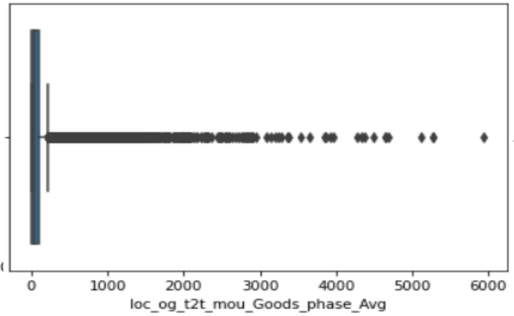


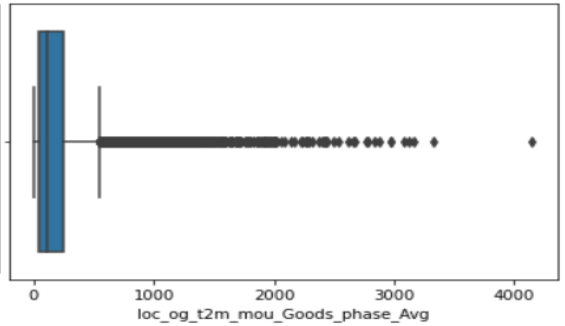


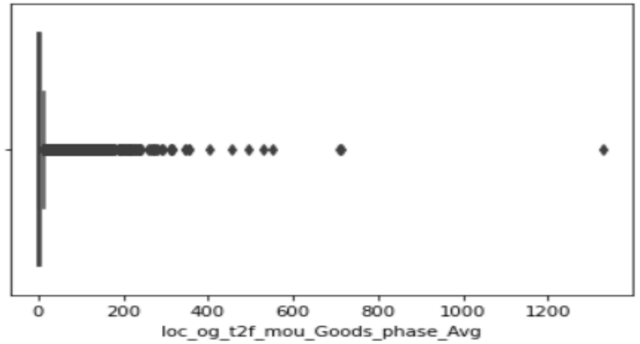




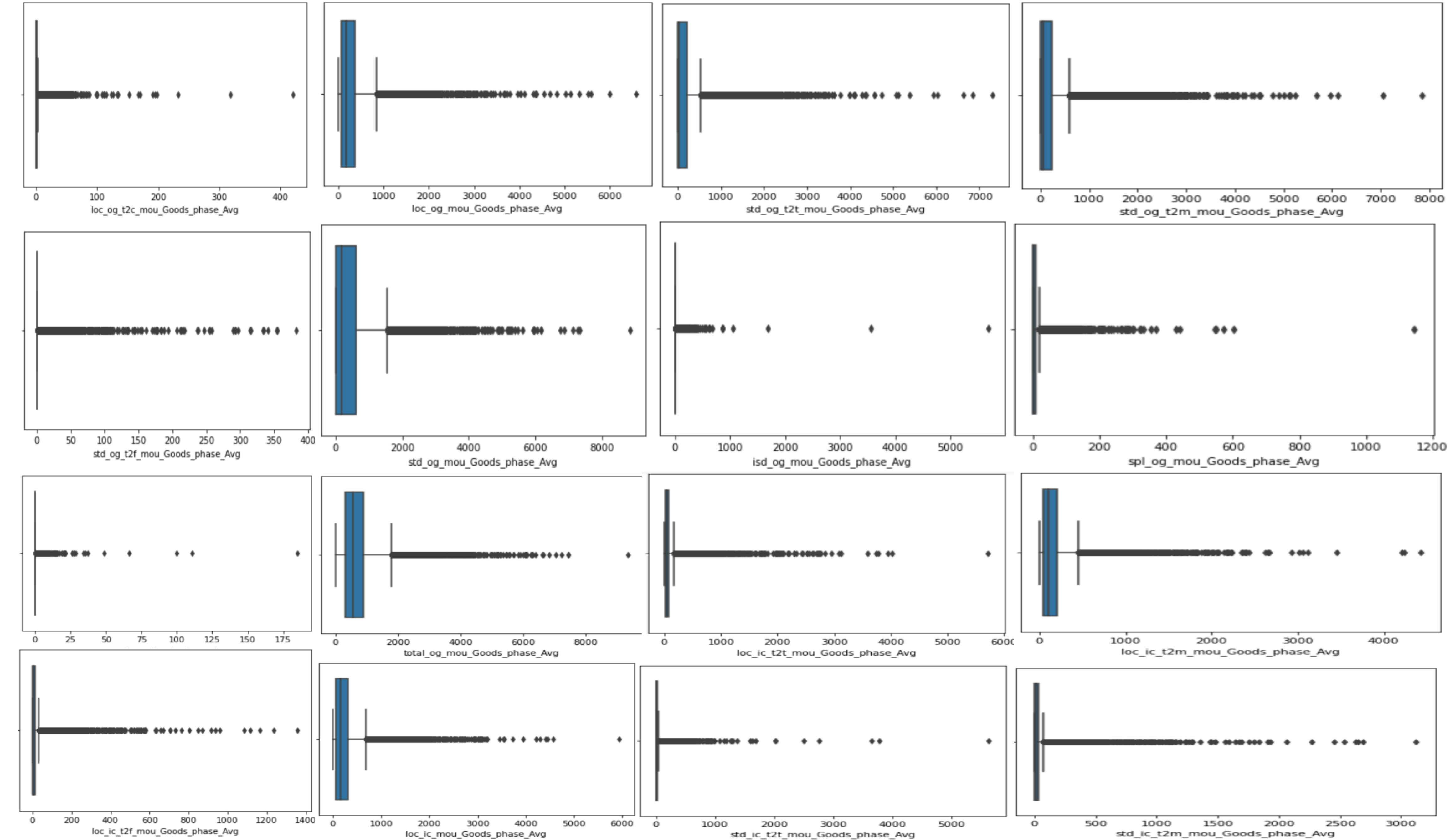




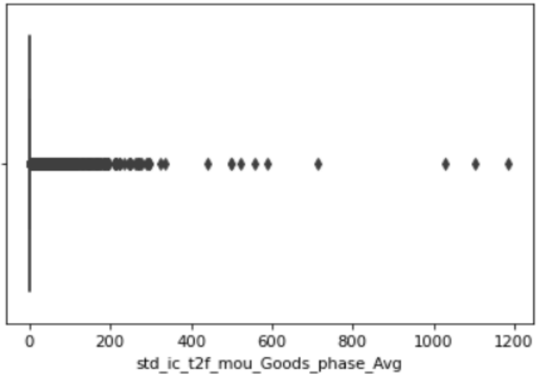


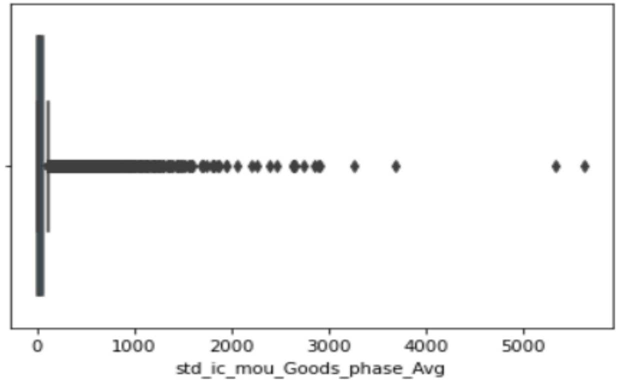


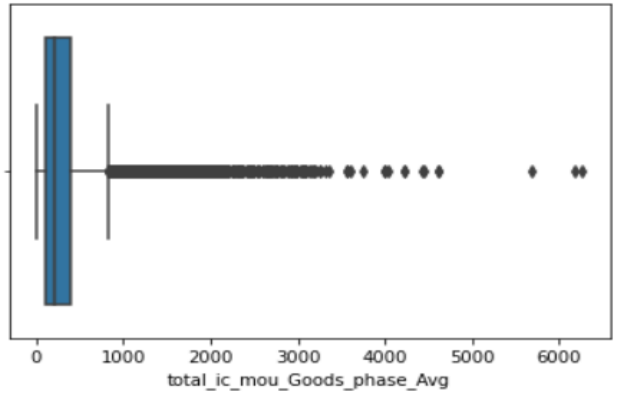
Doing the EDA analysis – checking Outliers

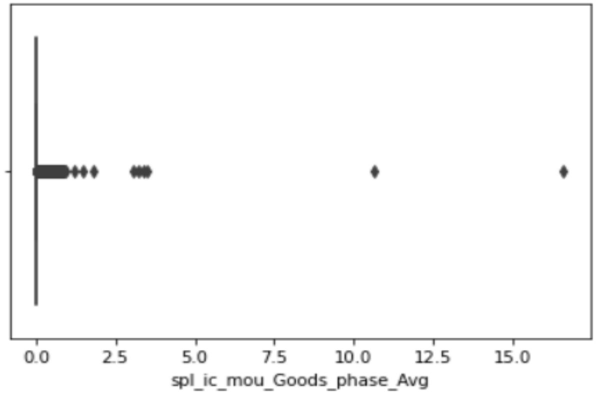


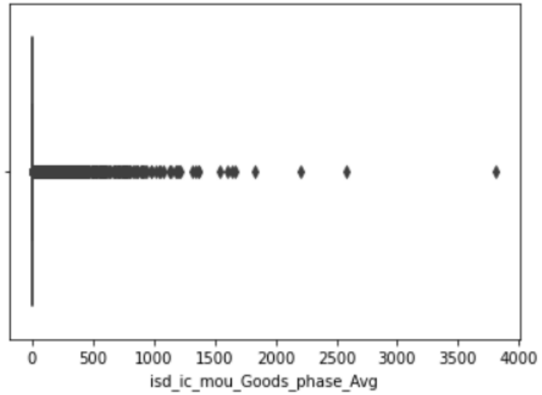
Doing the EDA analysis – checking Outliers

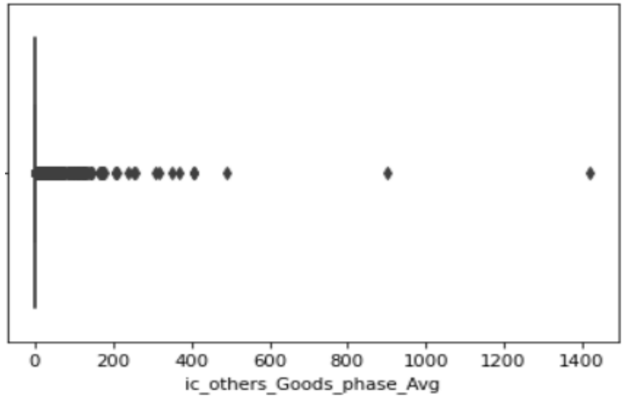


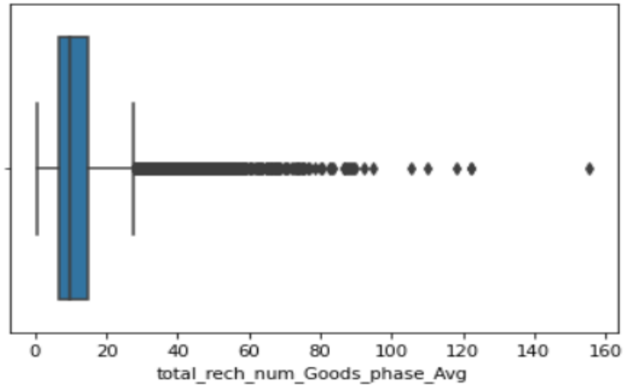


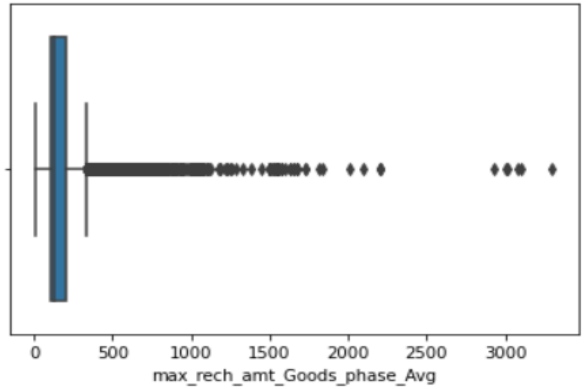


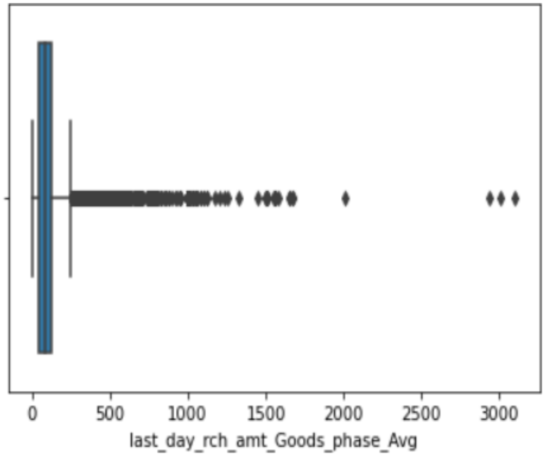


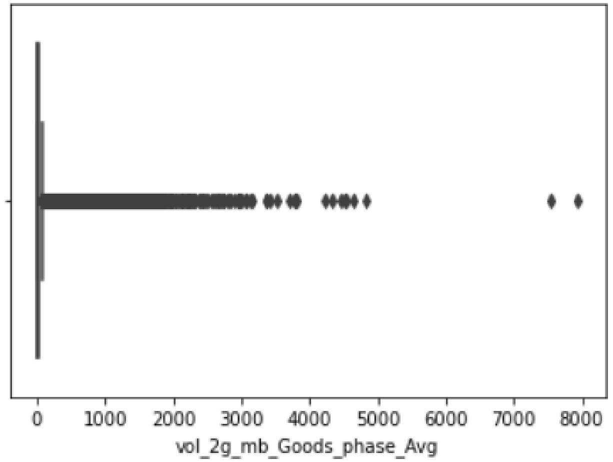


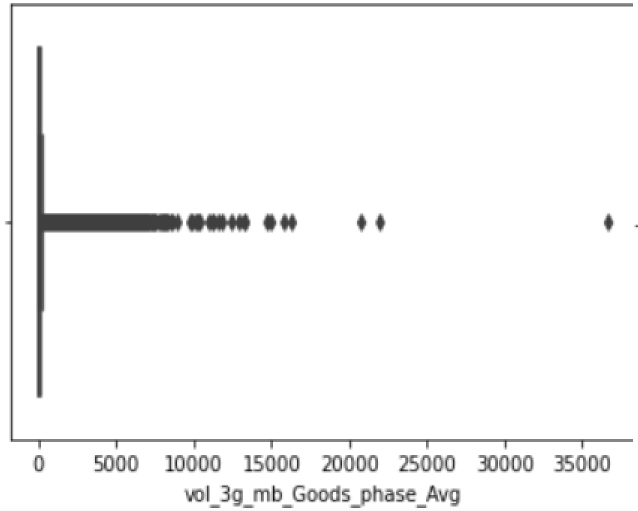


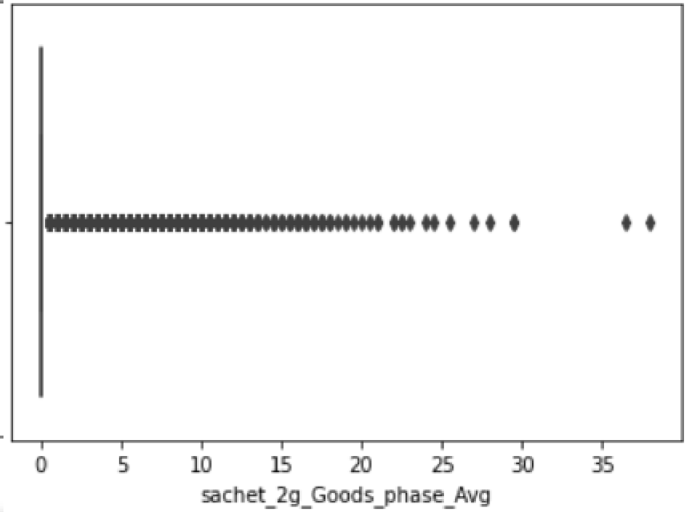




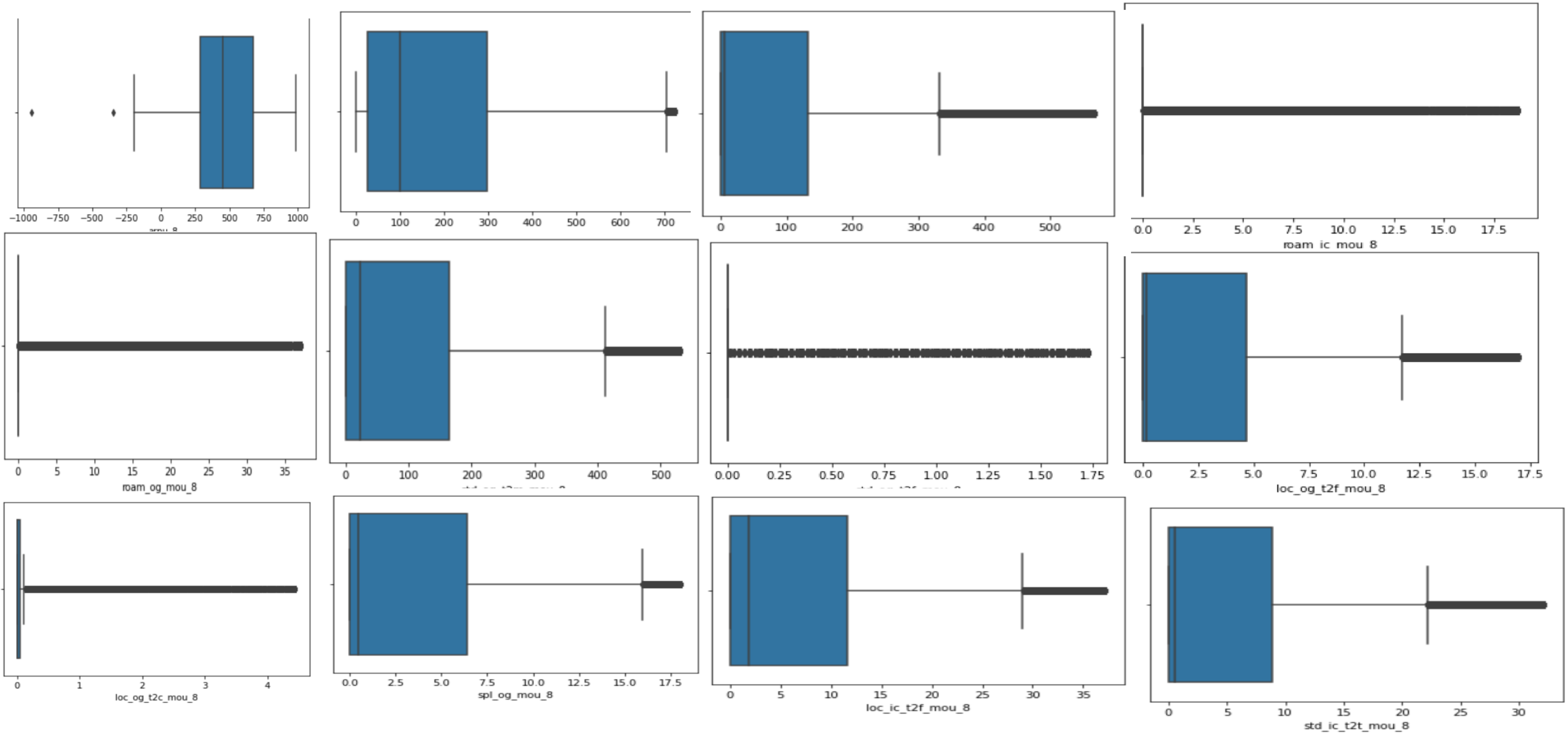


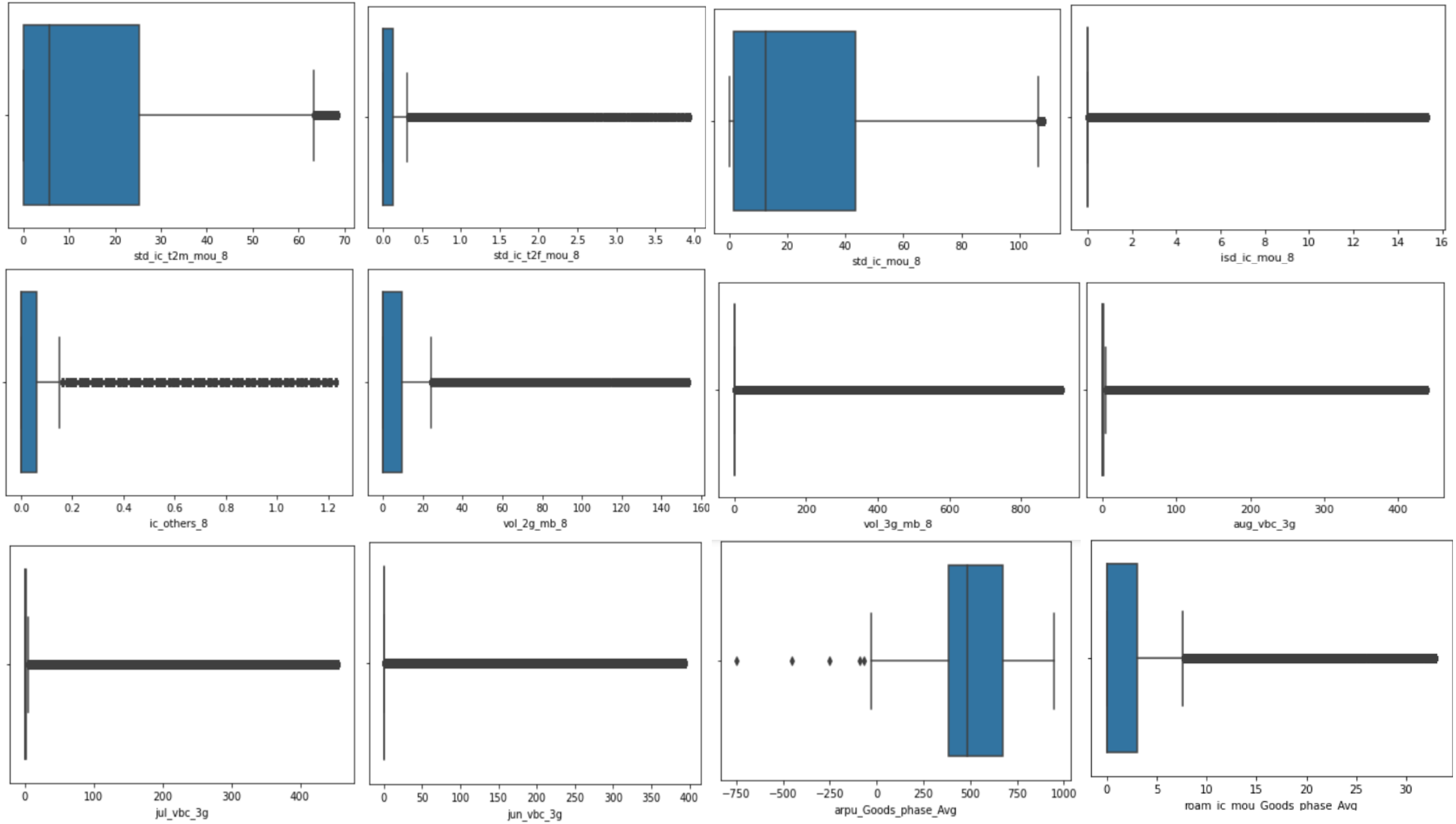


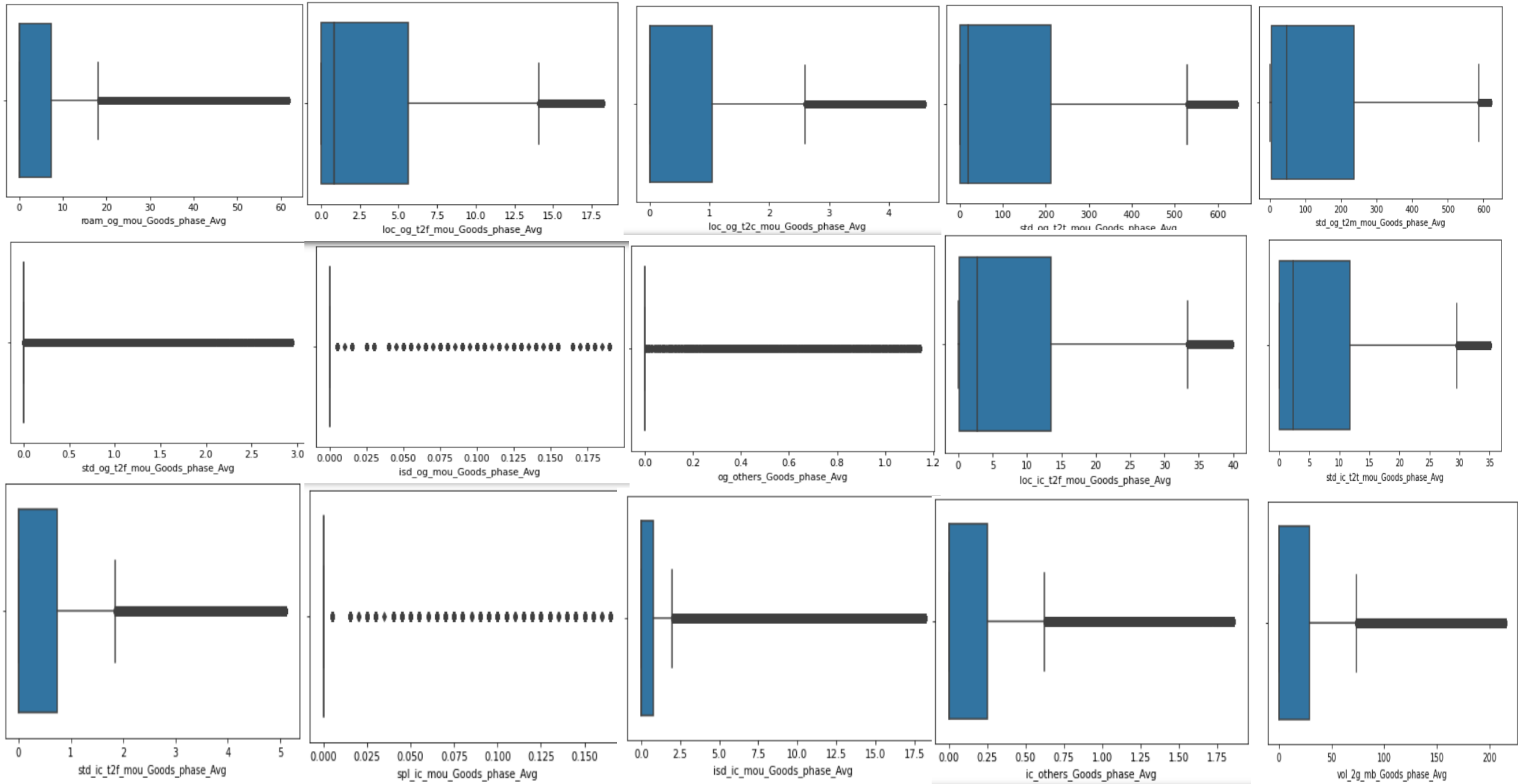


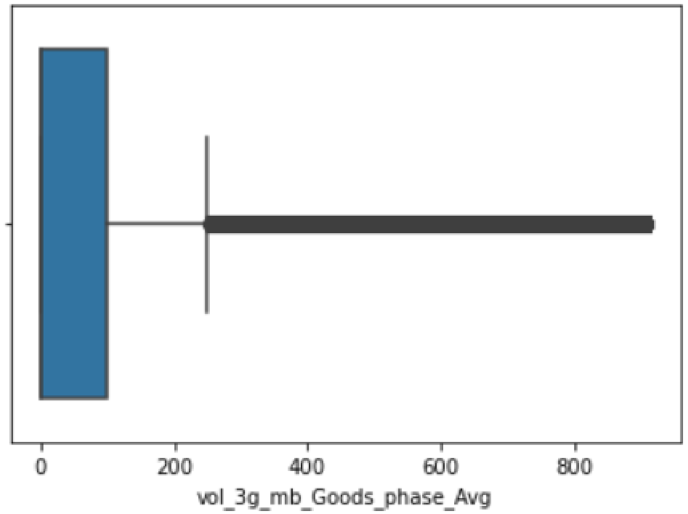


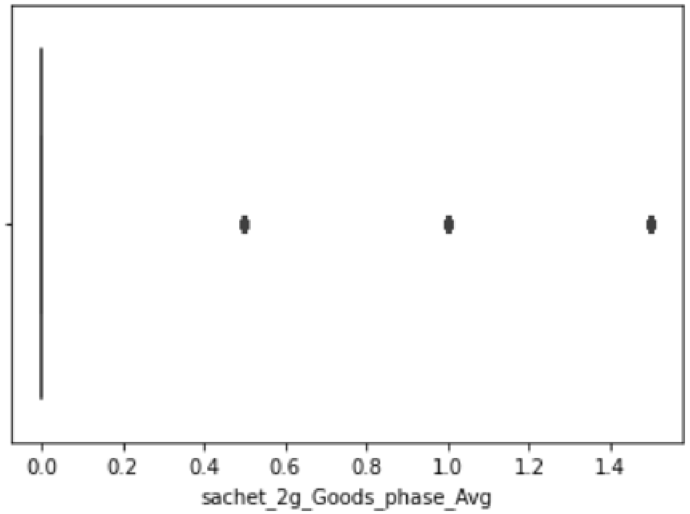
So here we can observe that the outlier present in data.Now Changing the quantile range to 90% to cap the outlier




std_ic_t2m_mou_8
std_ic_t2f_mou_8
std_ic_mou_8
isd_ic_mou_8
ic_others_8
vol_2g_mb_8
vol_3g_mb_8
aug_vbc_3g
jul_vbc_3g
jun_vbc_3g
arpu_Goods_phase_Avg
roam_ic_mou_Goods_phase_Avg

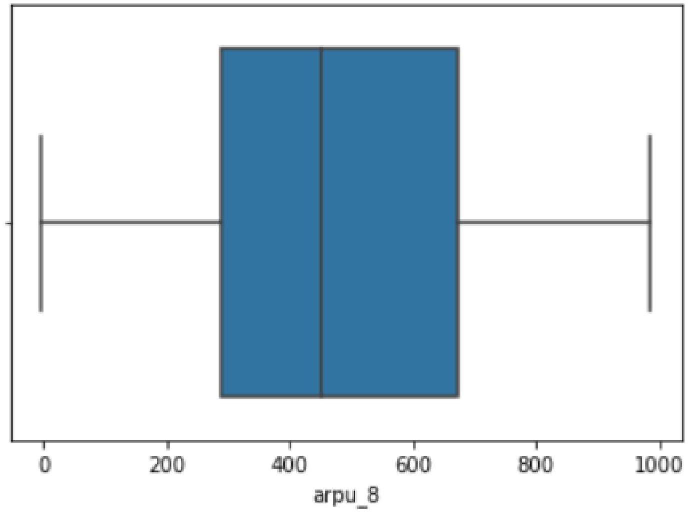

roam_og_mou_Goods_phase_Avg
loc_og_t2f_mou_Goods_phase_Avg
loc_og_t2c_mou_Goods_phase_Avg
std_og_t2t_mou_Goods_phase_Avg
std_og_t2m_mou_Goods_phase_Avg
std_og_t2f_mou_Goods_phase_Avg
isd_og_mou_Goods_phase_Avg
og_others_Goods_phase_Avg
loc_ic_t2f_mou_Goods_phase_Avg
std_ic_t2t_mou_Goods_phase_Avg
std_ic_t2f_mou_Goods_phase_Avg
spl_ic_mou_Goods_phase_Avg
isd_ic_mou_Goods_phase_Avg
ic_others_Goods_phase_Avg
vol_2g_mb_Goods_phase_Avg

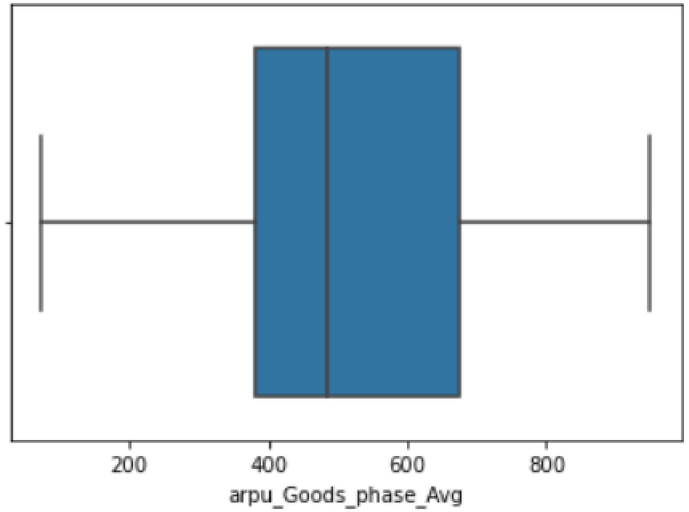




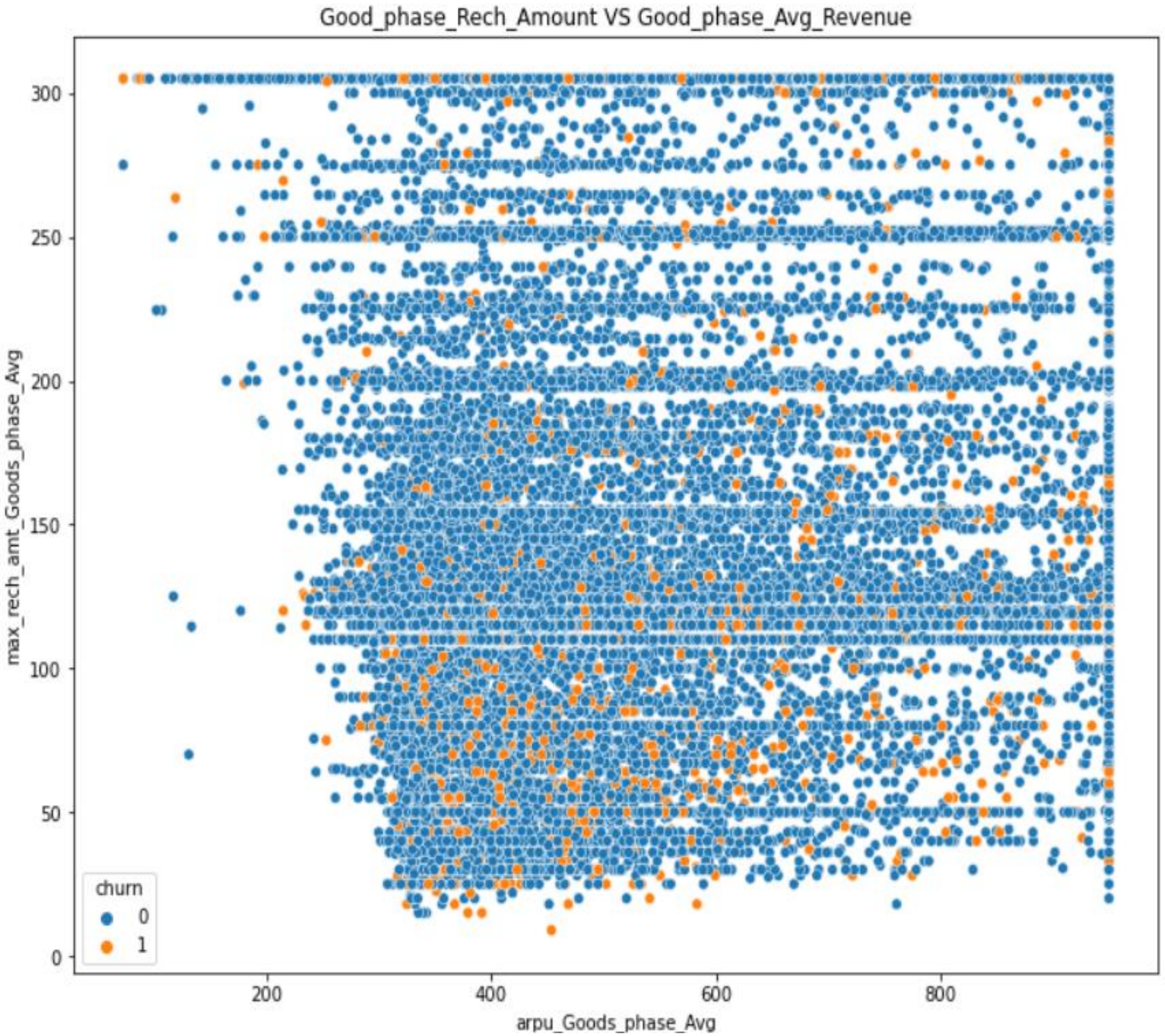
Dropping the Outliers : final shape comes out to be (30011, 63).
Two outliers present in lower side in IQR range.
Caping them to 1 Percentile.

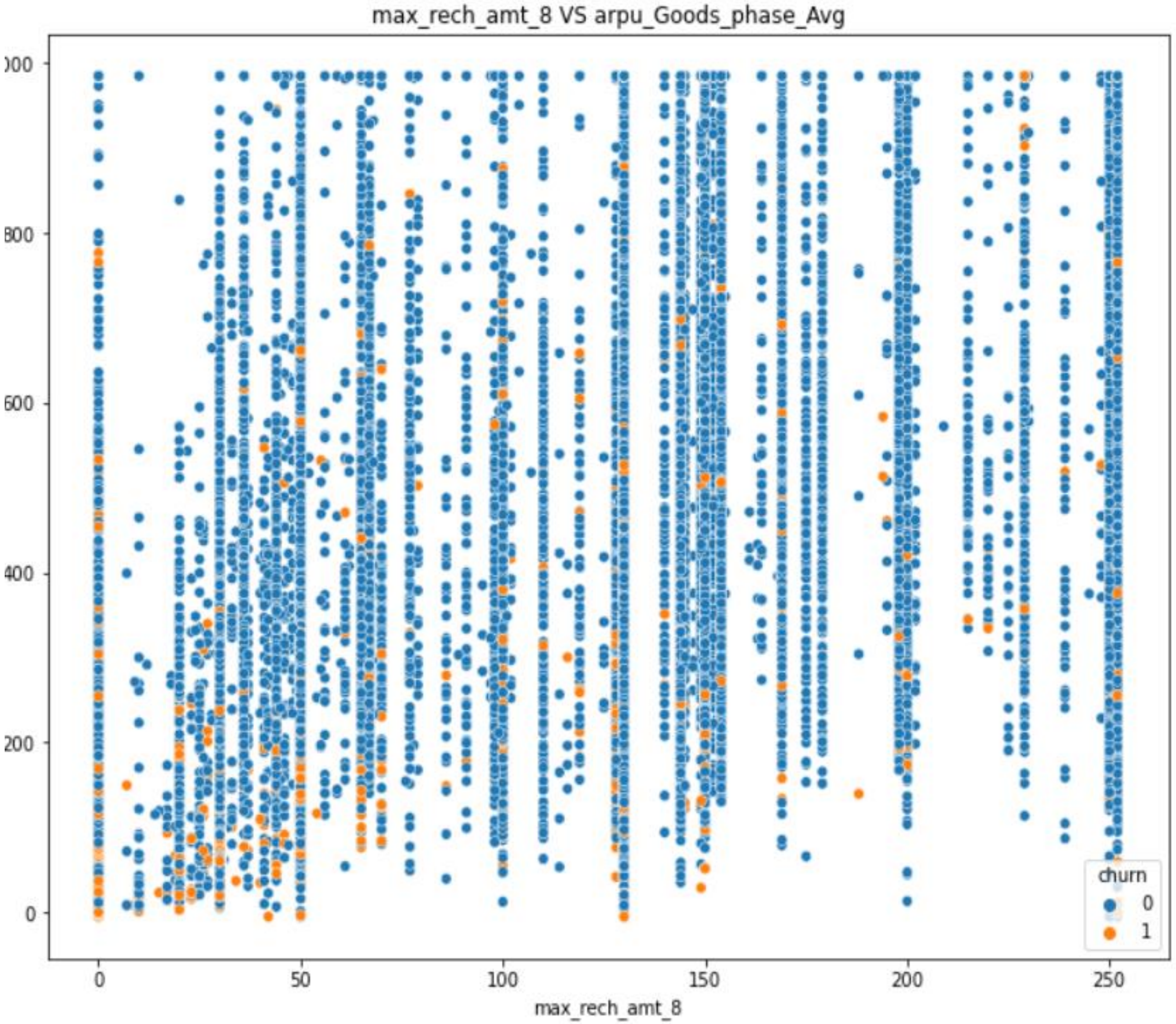
	arpu_8	arpu_Goods_phase_Avg
count	30011.000000	30011.000000
mean	486.823367	545.980785
std	277.996871	205.787143
min	-945.808000	-749.783000
25%	289.609500	381.272250
50%	452.091000	485.602500
75%	671.150000	674.492000
max	985.202000	949.430500





Bivariate Analysis

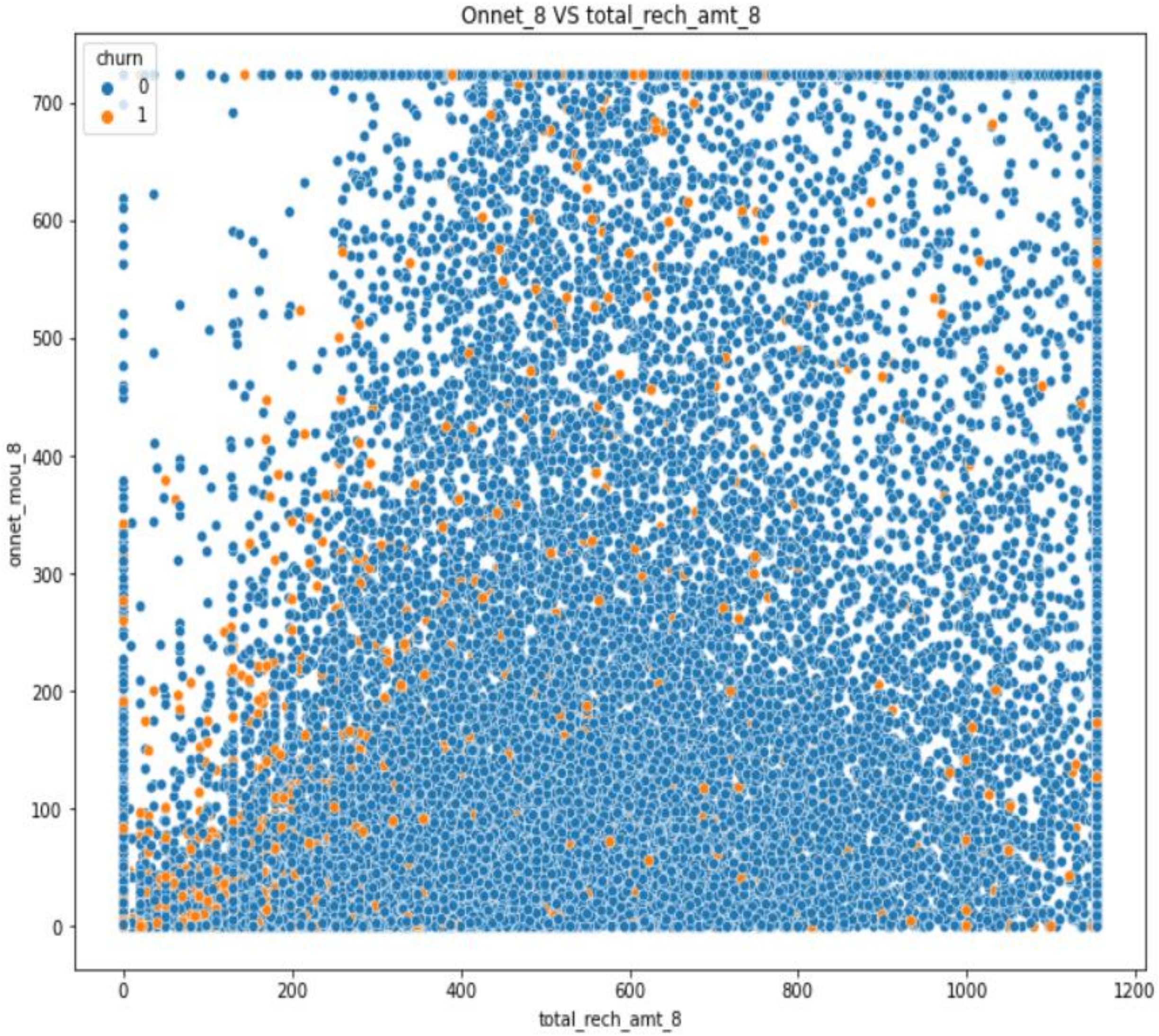


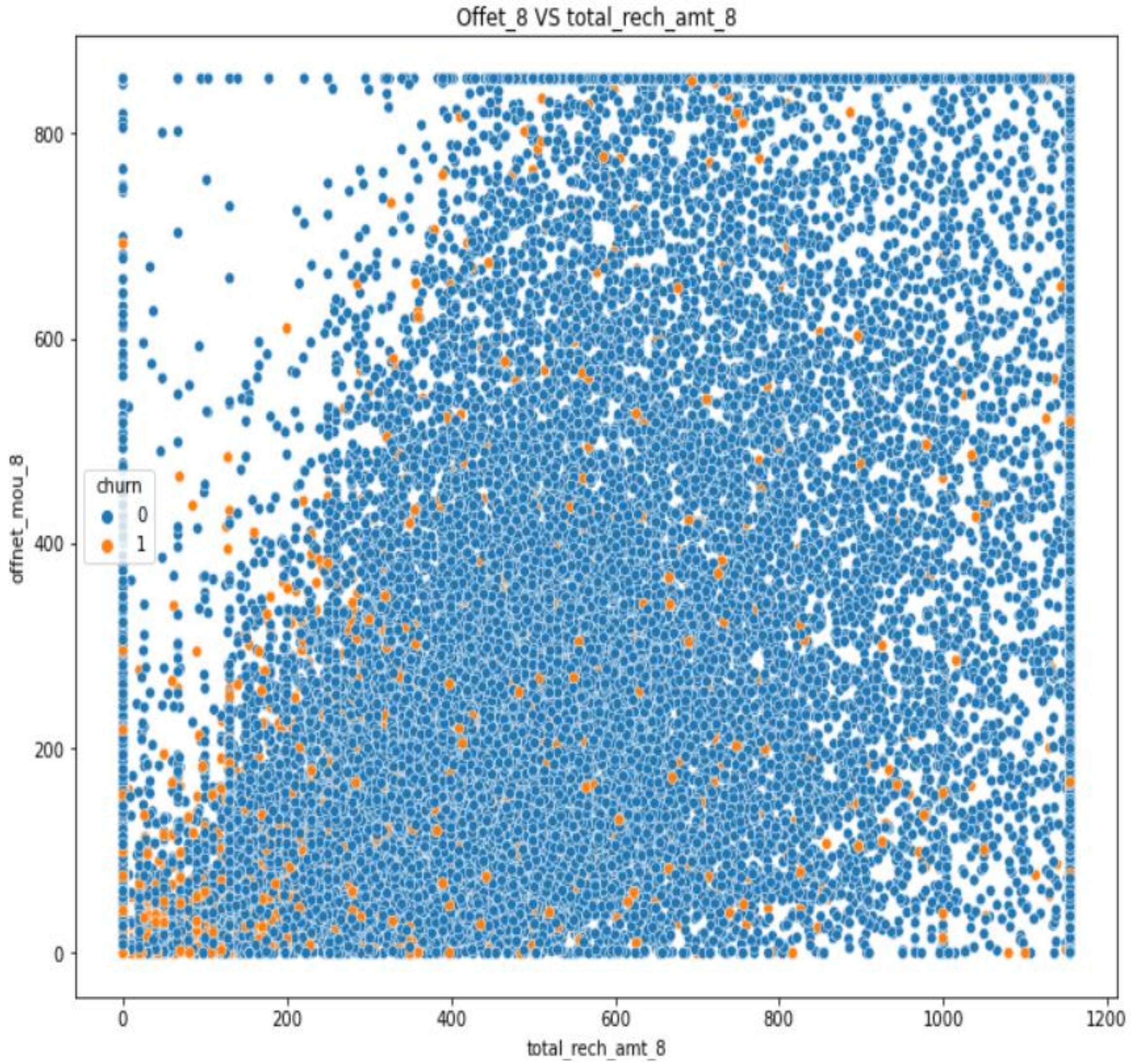


Insights:

Here we can observe that the customer having recharge amount less than 50 in action phase are more liley to churn

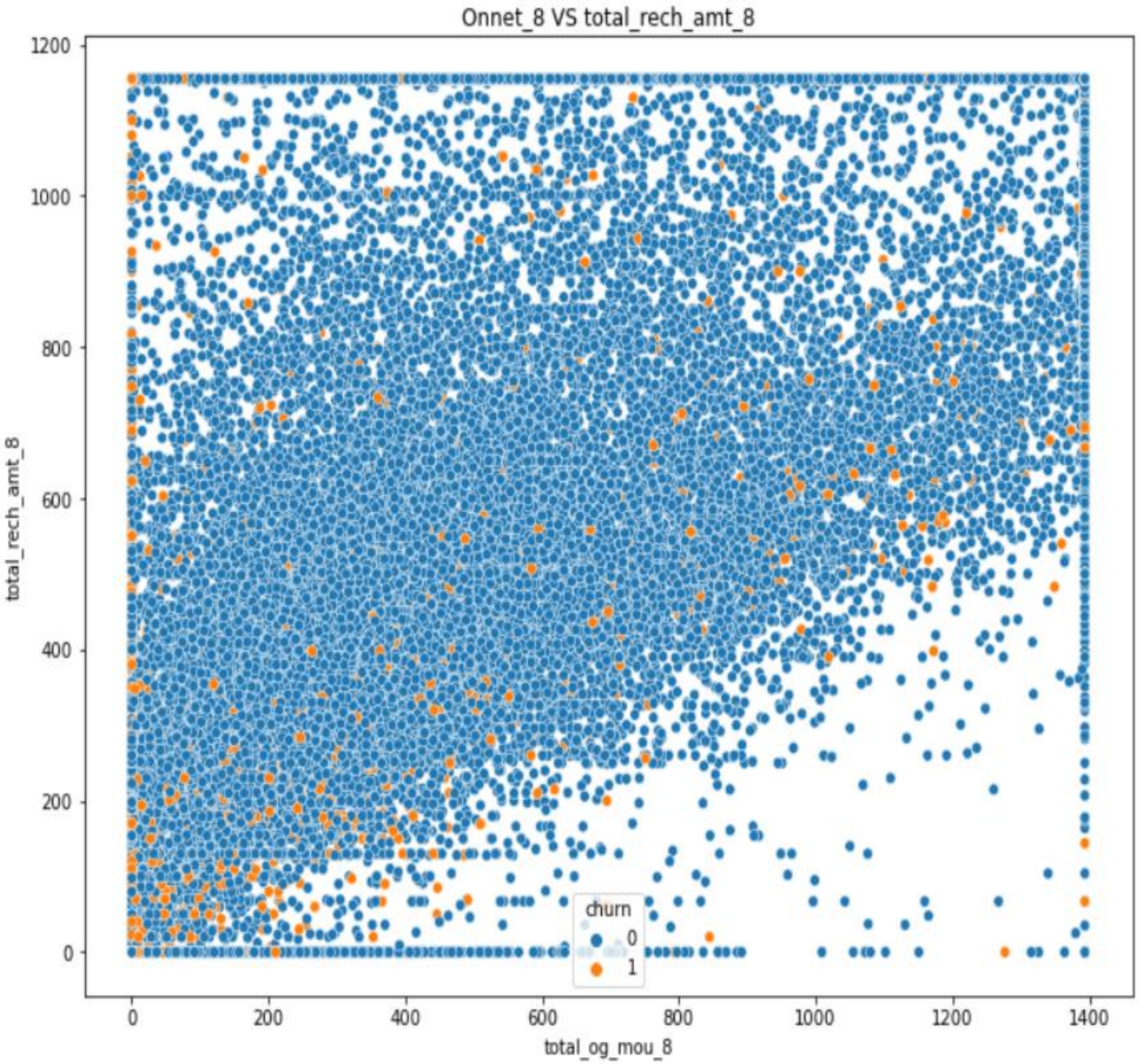
Also those who have recharge amount more than 150 in goods phase are generting most of the revenue and having very less churn rate .

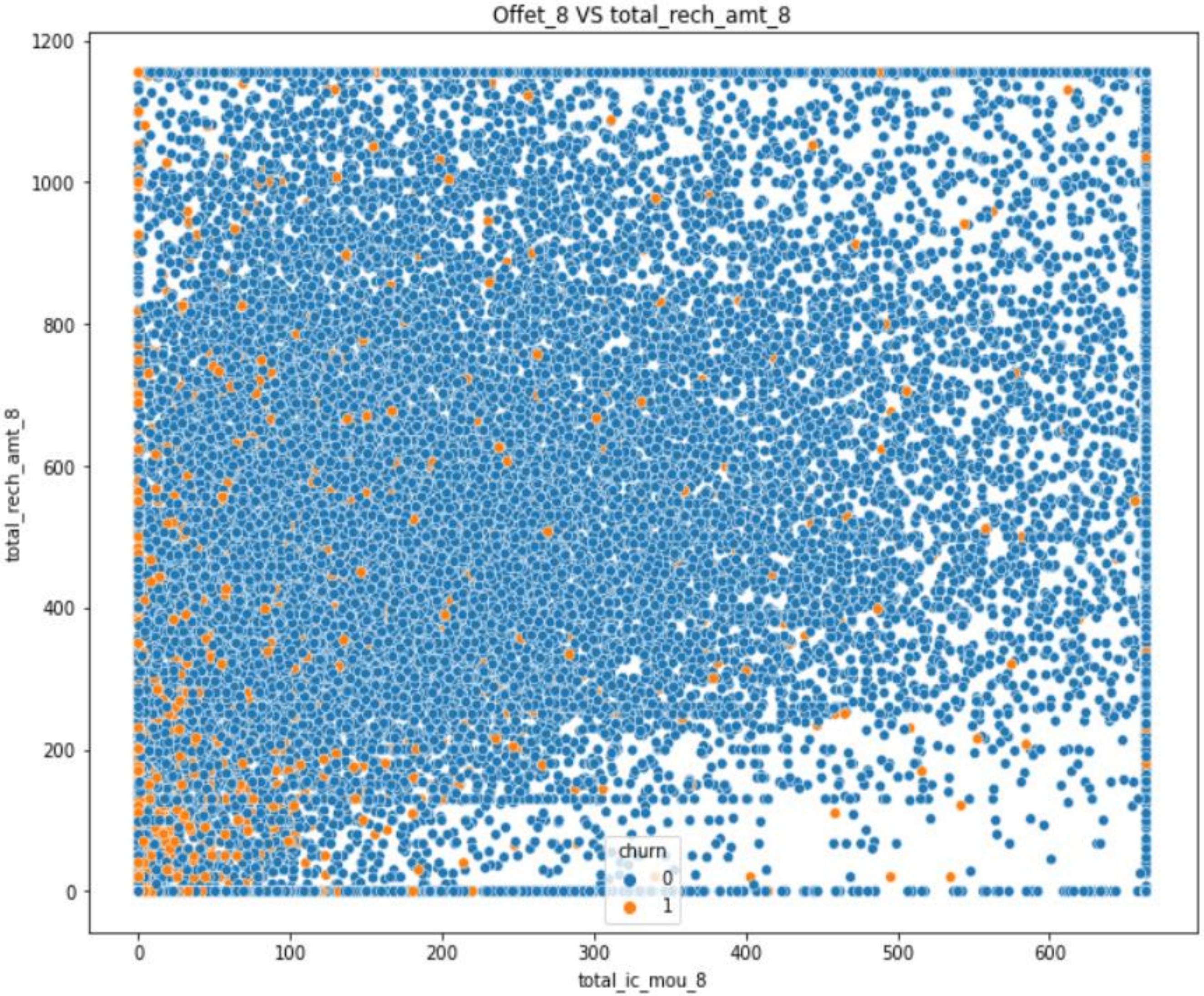




Insights:

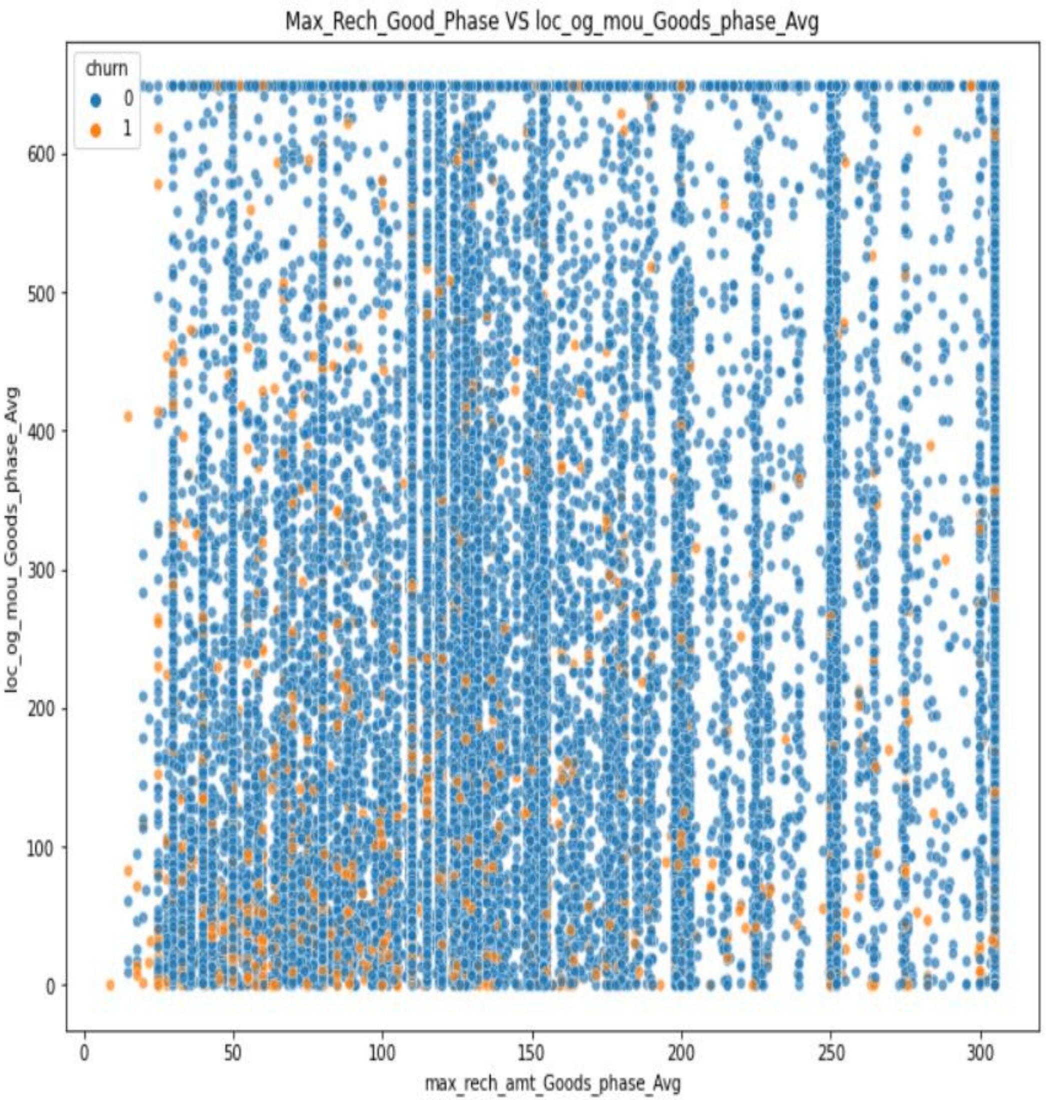
We can observe here that customer whose number of call less than 200 are most likey to churn

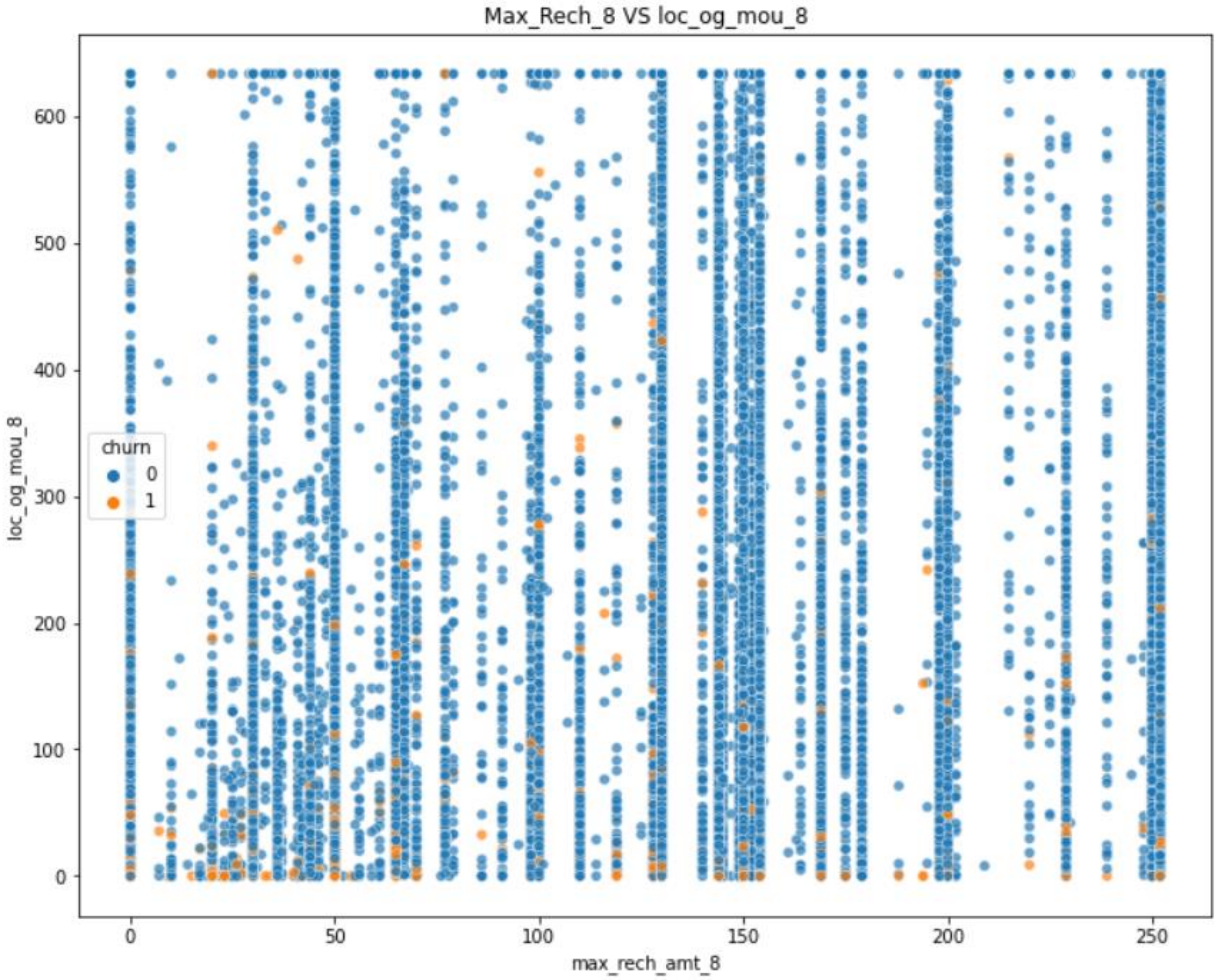




Insights:

Here we can observe that the customer having Incoming calls less than 75 and outgoing call less than 200 are more likely to churn





Insights:

From the graph we can observe that the customer who recharge amount is less than 100 and having local outgoing calls minutes less than 200 are more likely to churn .

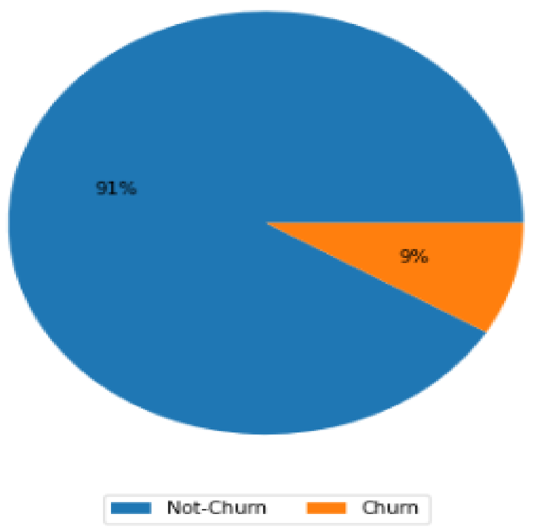
We can also observe that the customer who has recharge amount less than 100 in good phase might doing recharge less than 50 in action phase are more likely that they have found some alternate option and those customer are more likely to churn.

Multivariate Analysis :

Insights:

Here we can observe that some variable his highly correlated with the others variable means the multicollinearity present in data.

Checking Data Imbalance : The data is highly skewed towards zero's



- So here we can observe that the data is high imbalance
- To handle the imbalance we will use SMOTE technique in modelling phase

Model Building : Logistic Regression : Model 1 summary :

Generalized Linear Model Regression Results

Dep. Variable:	churn	No. Observations:	38385
Model:	GLM	Df Residuals:	38369
Model Family:	Binomial	Df Model:	15
Link Function:	Logit	Scale:	1.0000
Method:	IRLS	Log-Likelihood:	-15176.
Date:	Tue, 09 May 2023	Deviance:	30352.
Time:	10:41:22	Pearson chi2:	4.64e+04
No. Iterations:	6	Pseudo R-squ. (CS):	0.4487
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	1.3488	0.050	27.227	0.000	1.252	1.446
onnet_mou_8	2.8613	0.178	16.088	0.000	2.513	3.210
loc_og_t2t_mou_8	-1.7132	0.106	-16.143	0.000	-1.921	-1.505
std_og_t2t_mou_8	-2.7803	0.173	-16.079	0.000	-3.119	-2.441
std_og_mou_8	1.9542	0.176	11.108	0.000	1.609	2.299
total_og_mou_8	-3.2227	0.201	-16.052	0.000	-3.616	-2.829
loc_ic_t2m_mou_8	-2.7910	0.120	-23.328	0.000	-3.026	-2.557
loc_ic_t2f_mou_8	-0.9367	0.073	-12.766	0.000	-1.080	-0.793
total_rech_num_8	-3.5347	0.107	-33.154	0.000	-3.744	-3.326
total_rech_amt_8	1.8223	0.122	14.953	0.000	1.583	2.061
max_rech_amt_8	-1.2025	0.080	-15.061	0.000	-1.359	-1.046
last_day_rch_amt_8	-2.4284	0.066	-36.730	0.000	-2.558	-2.299
vol_2g_mb_8	-1.4494	0.067	-21.523	0.000	-1.581	-1.317
arpu_Goods_phase_Avg	1.5331	0.087	17.662	0.000	1.363	1.703
loc_ic_t2m_mou_Goods_phase_Avg	1.1105	0.094	11.811	0.000	0.926	1.295
total_rech_num_Goods_phase_Avg	1.1553	0.093	12.397	0.000	0.973	1.338

- Insights:- All the variable in Model 1 have P-value less than 0.05 which is good . Once check VIF for model 1 for checking multicollinearity in model 1

Checking VIF factor of Model 1 :

	Feature	VIF
5	total_og_mou_8	12.11
1	onnet_mou_8	11.92
3	std_og_t2t_mou_8	11.77
0	const	11.61
4	std_og_mou_8	10.42
9	total_rech_amt_8	7.19
6	loc_ic_t2m_mou_8	4.16
8	total_rech_num_8	3.87
10	max_rech_amt_8	3.49
2	loc_og_t2t_mou_8	3.44
14	loc_ic_t2m_mou_Goods_phase_Avg	2.56
15	total_rech_num_Goods_phase_Avg	2.20
11	last_day_rch_amt_8	2.05
13	arpu_Goods_phase_Avg	1.63
7	loc_ic_t2f_mou_8	1.50
12	vol_2g_mb_8	1.17

- Insights:- So here we can observe in model 1 we have total_og_mou_8 variable having VIF value more than 5 so now dropping the total_og_mou_8 variable to remove the multicollinearity .

Model Building : Logistic Regression : Model 2 summary :

Generalized Linear Model Regression Results

Dep. Variable:	churn	No. Observations:	38385
Model:	GLM	Df Residuals:	38370
Model Family:	Binomial	Df Model:	14
Link Function:	Logit	Scale:	1.0000
Method:	IRLS	Log-Likelihood:	-15323.
Date:	Tue, 09 May 2023	Deviance:	30646.
Time:	10:41:23	Pearson chi2:	4.47e+04
No. Iterations:	6	Pseudo R-squ. (CS):	0.4445
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	1.3758	0.049	27.921	0.000	1.279	1.472
onnet_mou_8	1.9468	0.156	12.505	0.000	1.642	2.252
loc_og_t2t_mou_8	-2.3356	0.099	-23.618	0.000	-2.529	-2.142
std_og_t2t_mou_8	-1.9126	0.154	-12.391	0.000	-2.215	-1.610
std_og_mou_8	-0.5721	0.075	-7.599	0.000	-0.720	-0.425
loc_ic_t2m_mou_8	-3.3349	0.116	-28.750	0.000	-3.562	-3.108
loc_ic_t2f_mou_8	-1.0037	0.073	-13.750	0.000	-1.147	-0.861
total_rech_num_8	-3.4982	0.106	-32.951	0.000	-3.706	-3.290
total_rech_amt_8	1.4232	0.118	12.106	0.000	1.193	1.654
max_rech_amt_8	-1.0664	0.079	-13.535	0.000	-1.221	-0.912
last_day_rch_amt_8	-2.3917	0.065	-36.684	0.000	-2.519	-2.264
vol_2g_mb_8	-1.3820	0.066	-20.817	0.000	-1.512	-1.252
arpu_Goods_phase_Avg	1.4592	0.086	16.989	0.000	1.291	1.628
loc_ic_t2m_mou_Goods_phase_Avg	1.0795	0.094	11.521	0.000	0.896	1.263
total_rech_num_Goods_phase_Avg	1.1929	0.093	12.836	0.000	1.011	1.375

- Insights:- All the variable in model 2 having P-value less than 0.05 which is good.

Checking VIF factor of Model 2 :

	Feature	VIF
0	const	11.58
3	std_og_t2t_mou_8	10.73
1	onnet_mou_8	10.72
8	total_rech_amt_8	6.86
7	total_rech_num_8	3.87
5	loc_ic_t2m_mou_8	3.79
4	std_og_mou_8	3.47
9	max_rech_amt_8	3.45
2	loc_og_t2t_mou_8	3.18
13	loc_ic_t2m_mou_Goods_phase_Avg	2.56
14	total_rech_num_Goods_phase_Avg	2.20
10	last_day_rch_amt_8	2.05
12	arpu_Goods_phase_Avg	1.63
6	loc_ic_t2f_mou_8	1.49
11	vol_2g_mb_8	1.17

- Insights:- Here we can observe in Model 2 the std_og_t2t_mou_8 varible having VIF>5 so better is to drop this variable to reduce the multicollinearity.

Model Building : Logistic Regression : Model 3 summary :

Generalized Linear Model Regression Results

Dep. Variable:	churn	No. Observations:	38385
Model:	GLM	Df Residuals:	38371
Model Family:	Binomial	Df Model:	13
Link Function:	Logit	Scale:	1.0000
Method:	IRLS	Log-Likelihood:	-15399.
Date:	Tue, 09 May 2023	Deviance:	30798.
Time:	10:41:23	Pearson chi2:	4.35e+04
No. Iterations:	6	Pseudo R-squ. (CS):	0.4423
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	1.3773	0.049	27.989	0.000	1.281	1.474
onnet_mou_8	0.1977	0.068	2.897	0.004	0.064	0.331
loc_og_t2t_mou_8	-1.6248	0.077	-21.055	0.000	-1.776	-1.474
std_og_mou_8	-0.9864	0.068	-14.497	0.000	-1.120	-0.853
loc_ic_t2m_mou_8	-3.5265	0.116	-30.531	0.000	-3.753	-3.300
loc_ic_t2f_mou_8	-1.0346	0.073	-14.165	0.000	-1.178	-0.891
total_rech_num_8	-3.4855	0.106	-32.976	0.000	-3.693	-3.278
total_rech_amt_8	1.6537	0.116	14.273	0.000	1.427	1.881
max_rech_amt_8	-1.1186	0.079	-14.245	0.000	-1.272	-0.965
last_day_rch_amt_8	-2.3818	0.065	-36.624	0.000	-2.509	-2.254
vol_2g_mb_8	-1.3951	0.066	-21.114	0.000	-1.525	-1.266
arpu_Goods_phase_Avg	1.4901	0.086	17.380	0.000	1.322	1.658
loc_ic_t2m_mou_Goods_phase_Avg	1.0905	0.094	11.643	0.000	0.907	1.274
total_rech_num_Goods_phase_Avg	1.1722	0.093	12.636	0.000	0.990	1.354

- Insights:- Here we can observe that all the variable in Model 3 having P-value less than 0.05 which is good.

Checking VIF factor of Model 3 :

	Feature	VIF
0	const	11.58
7	total_rech_amt_8	6.74
6	total_rech_num_8	3.87
4	loc_ic_t2m_mou_8	3.73
8	max_rech_amt_8	3.45
3	std_og_mou_8	2.72
1	onnet_mou_8	2.58
12	loc_ic_t2m_mou_Goods_phase_Avg	2.56
13	total_rech_num_Goods_phase_Avg	2.20
9	last_day_rch_amt_8	2.05
2	loc_og_t2t_mou_8	2.03
11	arpu_Goods_phase_Avg	1.62
5	loc_ic_t2f_mou_8	1.49
10	vol_2g_mb_8	1.17

- Insights:- Here in Model 4 we observe that the total_rech_amt_8 variable having VIF>5 so dropping these variable to reduce multcollinearity

Model Building : Logistic Regression : Model 4 summary :

Generalized Linear Model Regression Results

Dep. Variable:	churn	No. Observations:	38385
Model:	GLM	Df Residuals:	38372
Model Family:	Binomial	Df Model:	12
Link Function:	Logit	Scale:	1.0000
Method:	IRLS	Log-Likelihood:	-15502.
Date:	Tue, 09 May 2023	Deviance:	31003.
Time:	10:41:24	Pearson chi2:	4.39e+04
No. Iterations:	6	Pseudo R-squ. (CS):	0.4393
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	1.1216	0.046	24.634	0.000	1.032	1.211
onnet_mou_8	0.2316	0.068	3.385	0.001	0.098	0.366
loc_og_t2t_mou_8	-1.5271	0.077	-19.929	0.000	-1.677	-1.377
std_og_mou_8	-0.7962	0.067	-11.890	0.000	-0.927	-0.665
loc_ic_t2m_mou_8	-3.3757	0.114	-29.556	0.000	-3.600	-3.152
loc_ic_t2f_mou_8	-1.0399	0.073	-14.266	0.000	-1.183	-0.897
total_rech_num_8	-2.5985	0.084	-31.074	0.000	-2.762	-2.435
max_rech_amt_8	-0.5308	0.067	-7.924	0.000	-0.662	-0.400
last_day_rch_amt_8	-2.0994	0.061	-34.291	0.000	-2.219	-1.979
vol_2g_mb_8	-1.3922	0.066	-21.110	0.000	-1.521	-1.263
arpu_Goods_phase_Avg	2.0189	0.078	25.746	0.000	1.865	2.173
loc_ic_t2m_mou_Goods_phase_Avg	1.0017	0.093	10.824	0.000	0.820	1.183
total_rech_num_Goods_phase_Avg	0.9027	0.090	10.063	0.000	0.727	1.078

- Insights: Here in Model 4 all the variable having p-values less than 0.05 , so now looking to VIF value

Checking VIF factor of Model 4 :

	Feature	VIF
0	const	10.28
4	loc_ic_t2m_mou_8	3.69
3	std_og_mou_8	2.59
1	onnet_mou_8	2.58
7	max_rech_amt_8	2.55
11	loc_ic_t2m_mou_Goods_phase_Avg	2.55
6	total_rech_num_8	2.52
12	total_rech_num_Goods_phase_Avg	2.13
2	loc_og_t2t_mou_8	2.00
8	last_day_rch_amt_8	1.87
5	loc_ic_t2f_mou_8	1.49
10	arpu_Goods_phase_Avg	1.36
9	vol_2g_mb_8	1.17

- So Now we have model 4 which have P-value less than 0.05 and VIF less than 5 ,So finalizing the model 4 as our final model.

Building Prediction :

	Churn	Churn_probaility
24443	0	0.038109
42841	1	0.939149
41118	1	0.800832
2254	0	0.265182
29470	0	0.798344

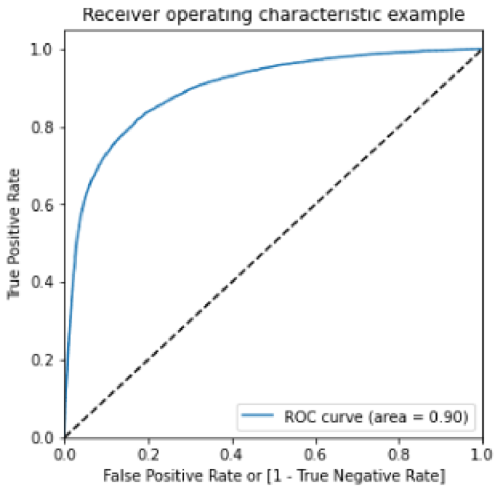
	Churn	Churn_probaility	Churn_Predict
0	0	0.038109	0
1	1	0.939149	1
2	1	0.800832	1
3	0	0.265182	0
4	0	0.798344	1

Model Evaluation :

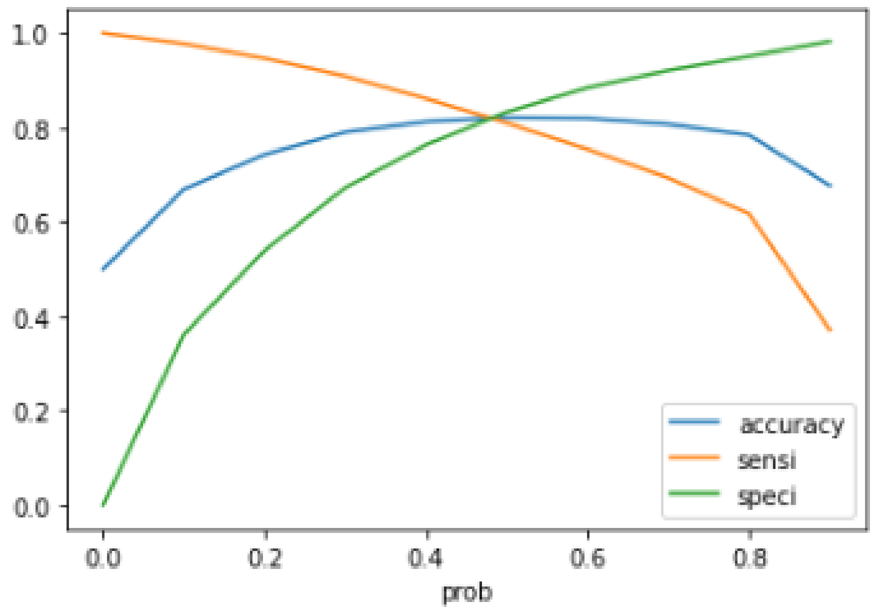
Model have 82.08 % accuracy, 80.93 % sensitivity , 83,24 % specificity, 16.75 % false positive rate, 82.82 %positive prediction value, 81.38 % negative prediction value.

Finding the Optimum cut off point :

Insights:- Area under curve is 0.90 which is very good.



Plot for accuracy , sensitivity and specificity for various Probabilities :From the curve , 0.5 is the optimum point to take it as a cutoff probability



So we have Scores on Train data,Accuracy Score- 82.08%,Sensitivity - 80.95%,Specificity- 83.24%
Prediction on test data : We have Scores on Test data
Accuracy Score- 82.28%
Sensitivity - 80.95%
Specificity- 83.61%

```python
# Top 10 predictors

lr.params.sort_values(ascending=False)[0:11]
```

```
arpu_Goods_phase_Avg             2.018857
const                            1.121627
loc_ic_t2m_mou_Goods_phase_Avg   1.001658
total_rech_num_Goods_phase_Avg   0.902676
onnet_mou_8                      0.231646
max_rech_amt_8                  -0.530813
std_og_mou_8                    -0.796151
loc_ic_t2f_mou_8                -1.039935
vol_2g_mb_8                     -1.392162
loc_og_t2t_mou_8                -1.527121
last_day_rch_amt_8              -2.099426
dtype: float64
```

- So by using Logistic regression model we achieve accuracy score of 82.18% on test data and 82.08% on train data

Decision Tree : Building Prediction :

```
Classification report on train data
              precision    recall  f1-score   support

           0       0.98      0.94      0.96     19207
           1       0.94      0.98      0.96     19178

    accuracy                           0.96     38385
   macro avg       0.96      0.96      0.96     38385
weighted avg       0.96      0.96      0.96     38385
```

```
Classification report on test data
              precision    recall  f1-score   support

           0       0.92      0.86      0.89      8211
           1       0.87      0.93      0.90      8240

    accuracy                           0.89     16451
   macro avg       0.89      0.89      0.89     16451
weighted avg       0.89      0.89      0.89     16451
```

- So we have accuracy score on Decision tree model
- On Train data- 96%
- On Test data- 89%

Hyper Parameter Tuning :

```
Fitting 4 folds for each of 10 candidates, totalling 40 fits

RandomizedSearchCV(cv=4, estimator=DecisionTreeClassifier(random_state=42),
                   n_jobs=-1,
                   param_distributions={'max_depth': [5, 10, 20, 30, 40, 50,
                                                      100],
                                        'min_samples_leaf': [5, 10, 20, 50, 100,
                                                             250, 500, 800,
                                                             1000],
                                        'min_samples_split': [1, 5, 10, 25, 50,
                                                              100]},
                   scoring='accuracy', verbose=1)
```

```
# Getting the GridSearch_CV best score
grid_search.best_score_
```

```
0.8781034316771651
```

```
# Getting the best estimator which the grid search has found out
dt=grid_search.best_estimator_
dt
```

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

Building Predictions:

```
Classification report on test data
              precision    recall  f1-score   support

           0       0.90      0.88      0.89      8211
           1       0.88      0.90      0.89      8240

    accuracy                           0.89     16451
   macro avg       0.89      0.89      0.89     16451
weighted avg       0.89      0.89      0.89     16451
```

- So we get accuracy score on Decision tree Hyper-paramater model

- On Train data- 92%
- On Test data- 88%

- So by using Decision Tree model we achieve accuracy score of 88% on test data and 92% on train data

Random Forest:

```
Classification Report on Train data
              precision    recall  f1-score   support

           0       1.00      0.97      0.98     19207
           1       0.97      1.00      0.98     19178

    accuracy                           0.98     38385
   macro avg       0.98      0.98      0.98     38385
weighted avg       0.98      0.98      0.98     38385
```

```
Classification report on test data
              precision    recall  f1-score   support

           0       0.95      0.94      0.95      8211
           1       0.95      0.95      0.95      8240

    accuracy                           0.95     16451
   macro avg       0.95      0.95      0.95     16451
weighted avg       0.95      0.95      0.95     16451
```

- So we have accuracy score on Random forest model
- On Train data- 98%
- On Test data- 95%

So we have accuracy score on Random forest model

- On Train data- 98%
- On Test data- 95%

Hyper Parameter tuning:

```
Fitting 4 folds for each of 10 candidates, totalling 40 fits

RandomizedSearchCV(cv=4,
                   estimator=RandomForestClassifier(n_jobs=-1, random_state=42),
                   n_jobs=-1,
                   param_distributions={'max_depth': [5, 10, 20, 30, 40, 50,
                                                      100],
                                        'min_samples_leaf': [5, 10, 20, 50, 100,
                                                             250, 500],
                                        'n_estimators': [5, 10, 20, 50]},
                   scoring='accuracy', verbose=1)
```

```
Classification Report on Train data
              precision    recall  f1-score   support

           0       0.98      0.97      0.98     19207
           1       0.97      0.98      0.98     19178

    accuracy                           0.98     38385
   macro avg       0.98      0.98      0.98     38385
weighted avg       0.98      0.98      0.98     38385
```

```
Classification report on test data
              precision    recall  f1-score   support

           0       0.94      0.95      0.94      8211
           1       0.95      0.94      0.94      8240

    accuracy                           0.94     16451
   macro avg       0.94      0.94      0.94     16451
weighted avg       0.94      0.94      0.94     16451
```

- So we have accuracy score on Random forest model with Hyper-parameter tuning
- On Train data- 97%
- On Test data- 93%

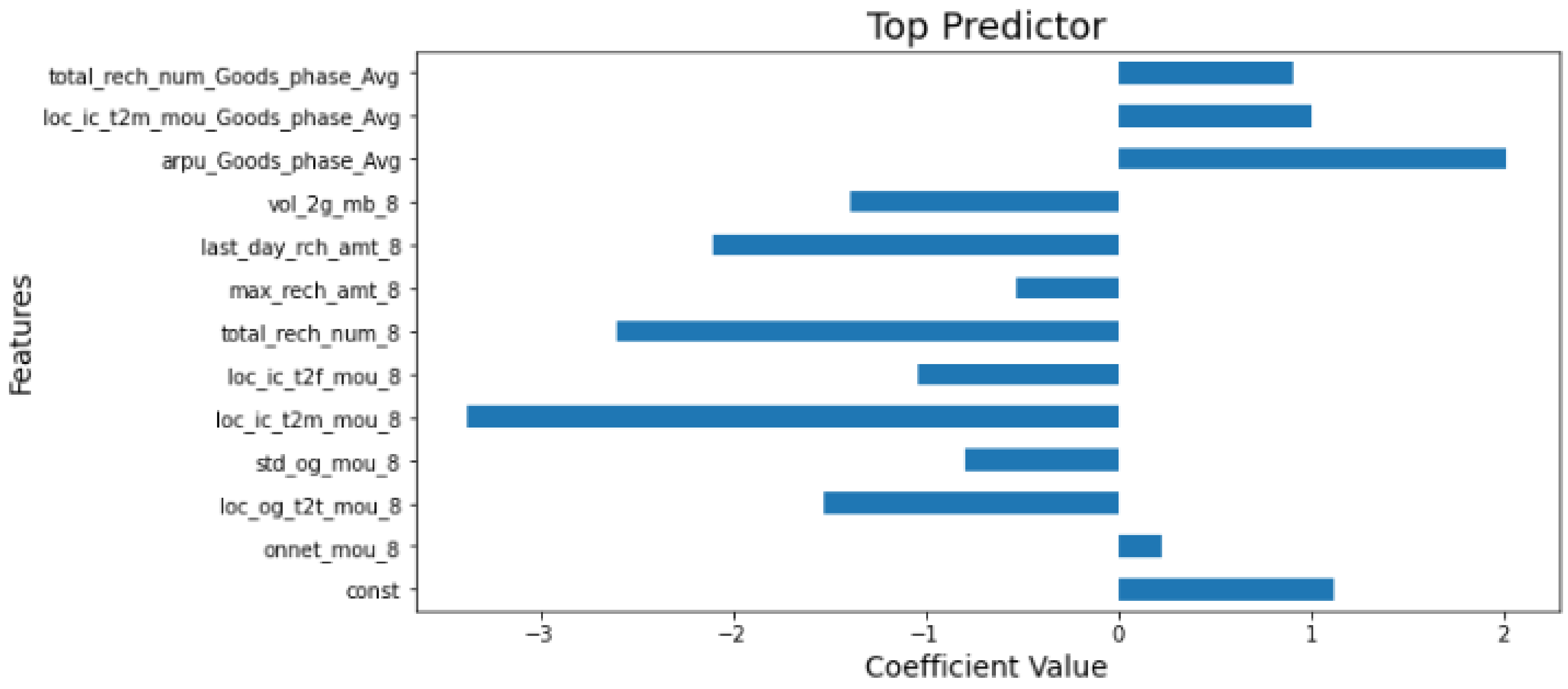
- So by using Random Forest model we achieve accuracy score of 95% on test data and 98% on train data whcih is bit higher than what the Random forest Model afer Hyper-Parameter models gives .
- S0 Now finalizing Random Forest Model(without hyper-parameter tuning) as the final model.

Conclusion

- As per out Bussniess Problem we have to retain the high value customer and for that we need model having high Recall value i.e(True Positive rate) so we have build the multiple model to find out the best fit model having high accuracy and high recall rate
- We have compare various mode and found Random Forest model havig high accuracy rate of 95% (test data) and high recall rate of 94% and 95% respectively .

```
Classification Report on Train data
              precision    recall  f1-score   support

           0       1.00      0.97      0.98     19207
           1       0.97      1.00      0.98     19178

    accuracy                           0.98     38385
   macro avg       0.98      0.98      0.98     38385
weighted avg       0.98      0.98      0.98     38385

Classification report on test data
              precision    recall  f1-score   support

           0       0.95      0.94      0.95      8211
           1       0.95      0.95      0.95      8240

    accuracy                           0.95     16451
   macro avg       0.95      0.95      0.95     16451
weighted avg       0.95      0.95      0.95     16451
```

Business recommendation :



- Insights
- The Company should take a look on customer those having rech amount less than 100 and simultaneously having local outgoing call less than 200 MOU are more likely that those customer are doing recharge less than 100 in next month (action phase) and simultaneously local outgoing call less than 100 MOU are more likely to churn so the company should look at these customers and should provide discounts on recharge or provide some extra benefits on existing recharge or should launch new plans for those customers.
- The company should focus on STD rate and due to high rate the customer may churn .So the company should provide discounts on STD calls .