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:-

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

```
# Importing the libraries
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns

import warnings
warnings.filterwarnings('ignore')
pd.set_option('display.max_columns', 500)
```

Reading and understanding the data

```
# Reading the dataset
df = pd.read csv('telecom churn data.csv')
df.head()
   mobile number circle id loc og t2o mou
                                               std og t2o mou
loc_ic_t2o mou
      7000842753
                         109
                                          0.0
                                                           0.0
0
0.0
      7001865778
                         109
                                          0.0
                                                           0.0
1
0.0
      7001625959
                                          0.0
                                                           0.0
2
                         109
0.0
3
      7001204172
                         109
                                          0.0
                                                           0.0
0.0
      7000142493
                         109
                                          0.0
                                                           0.0
0.0
```

last_0 0 1 2 3	date_of_mon 6/30/ 6/30/ 6/30/ 6/30/ 6/30/	2014 2014 2014 2014	ast_date	_of_month 7/31/20 7/31/20 7/31/20 7/31/20 7/31/20		date_of_mon 8/31/2 8/31/2 8/31/2 8/31/2 8/31/2	20 1 4 2014 2014 2014
last_ onnet_m 0	date_of_mon ou_6 \ 9/30/	_	arpu_6 197.385	arpu_7 214.816	arpu_8 213.803	. –	
NaN 1	9/30/	2014	34.047	355.074	268.321	. 86.285	
24.11 2 11.54	9/30/	2014	167.690	189.058	210.226	290.714	
3 99.91	9/30/	2014	221.338	251.102	508.054	389.500	
4 50.31	9/30/	2014	261.636	309.876	238.174	163.426	
	t_mou_7 on mou_7 \ NaN	net_mo	u_8 onn	et_mou_9 NaN	offnet_	_mou_6 NaN	NaN
1	78.68	/	. 68	18.34		15.74	99.84
2	55.24	37	. 26	74.81	1	.43.33	220.59
3	54.39	310	.98	241.71	1	.23.31	109.01
4	149.44	83	.89	58.78		76.96	91.88
	et_mou_8 c _mou_8 \	ffnet_	mou_9 r	oam_ic_mo	u_6 roa	m_ic_mou_7	
0 0.00	0.00		NaN		NaN	NaN	
1 0.00	304.76		53.76		0.0	0.00	
2	208.36	1	18.91		0.0	0.00	
3 44.38	71.68	1	13.54		0.0	54.86	
4 0.00	124.26		45.81		0.0	0.00	
	_ic_mou_9 _mou_9 \	roam_o	g_mou_6	roam_og_	mou_7 r	oam_og_mou_8	3
0 NaN	NaN		NaN		NaN	0.00	9

1	0.00	0.0	0.00	0.00
0.00 2	38.49	0.0	0.00	0.00
70.94 3	0.00	0.0	28.09	39.04
0.00				
4 0.00	0.00	0.0	0.00	0.00
		loc_og_t2t_mou_7	loc_og_t2t_mou_8	
loc_og_ 0	_t2t_mou_9 \ NaN	NaN	0.00	
NaN 1	23.88	74.56	7.68	
18.34	23.00	74.30		
2 14.39	7.19	28.74	13.58	
3	73.68	34.81	10.61	
15.49 4	50.31	149.44	83.89	
58.78				
		loc_og_t2m_mou_7	loc_og_t2m_mou_8	
0	_t2m_mou_9 \ NaN	NaN	0.00	
NaN 1	11.51	75.94	291.86	
53.76				
2 28.16	29.34	16.86	38.46	
3	107.43	83.21	22.46	
65.46 4	67.64	91.88	124.26	
37.89				
		loc_og_t2f_mou_7	loc_og_t2f_mou_8	
0	_t2f_mou_9 \ NaN	NaN	0.00	
NaN 1	0.00	0.00	0.00	
0.00				
2 22.24	24.11	21.79	15.61	
3	1.91	0.65	4.91	
2.06 4	0.00	0.00	0.00	
1.93				
	_og_t2c_mou_6 _t2c_mou_9 \	loc_og_t2c_mou_7	loc_og_t2c_mou_8	

0	NaN		NaN		0.00
NaN 1	0.0		2.91		0.00
0.00 2	0.0		135.54		45.76
0.48 3	0.0		0.00		0.00
0.00					
4 0.00	0.0		0.00		0.00
loc o	g mou 6 loc	c og mou 7	loc og	mou 8	loc_og_mou_9
std_og_t	2t_mou_6 \			_	
0 NaN	NaN	NaN		0.00	NaN
nan 1	35.39	150.51	2	99.54	72.11
0.23	55.55		_		,
2	60.66	67.41		67.66	64.81
4.34	102 02	110 60		27 00	02.02
3 26.23	183.03	118.68		37.99	83.03
4	117.96	241.33	2	08.16	98.61
0.00					
c+d o	g_t2t_mou_7	ctd og ti)+ maii Q	c+d o	a +2+ mau 0
	9_c2c_mou_7 2m_mou_6 \	3 tu_0g_t2	c_iiiou_o	3 tu_0	g_tzt_iiiou_9
0	NaN		0.00		NaN
NaN	4 11		0.00		0.00
1 0.00	4.11		0.00		0.00
2	26.49		22.58		8.76
41.81					
3	14.89		289.58		226.21
2.99 4	0.00		0.00		0.00
9.31	0.00		0.00		0.00
	. +2 7) O	-4-1 -	. +2 0
	g_t2m_mou_7 2f_mou_6 \	sta_og_tz	zm_mou_8	sta_o	g_tzm_mou_9
0 0	NaN		0.00		NaN
NaN					
1	0.46		0.13		0.00
0.00 2	67.41		75.53		9.28
1.48	07.41		75.55		9.20
3	1.73		6.53		9.99
0.00	2 22		0.00		0.00
4 0.00	0.00		0.00		0.00
0.00					

std	_og_t2f_mo	u_7	std_og_t2	f_mou_8	std_o	g_t2f_m	ou_9	
	_t2c_mou_6	_ NaN		0.00	_		NaN	
0 NaN		IVAIV		0.00			IVAIV	
1 0.0	0	.00		0.00			0.0	
2	14	.76		22.83			0.0	
0.0	0	.00		0.00			0.0	
0.0 4 0.0	0	.00		0.00			0.0	
	_og_t2c_mo	u_7	std_og_t2	c_mou_8	std_o	g_t2c_m	ou_9	std_og_mou_6
0		NaN		0.0			NaN	NaN
1		0.0		0.0			0.0	0.23
2		0.0		0.0			0.0	47.64
3		0.0		0.0			0.0	29.23
4		0.0		0.0			0.0	9.31
std	_og_mou_7	std	oa mou 8	std oa	mall 9	isd on	mou 6	3
isd_og_		J.Cu_	0.00	Jtu_og_	NaN	134_09	ou NaN	
NaN								
1 0.0	4.58		0.13		0.00		0.0	Ð
2	108.68		120.94		18.04		0.0	9
0.0 3	16.63		296.11	2	36.21		0.0	Ð
0.0 4	0.00		0.00		0.00		0.0	Ð
0.0								
-	_og_mou_8	isd_	_og_mou_9	spl_og_	mou_6	spl_og	_mou_7	7
0 0 0.00	_mou_8 \ 0.0		NaN		NaN		Nal	V
1	0.0		0.0		4.68		23.43	3
12.76 2	0.0		0.0		46.56		236.84	1
96.84 3	0.0		0.0		10.96		0.00	9
18.09 4	0.0		0.0		0.00		0.00	9

```
0.00
                                og others 7
                                              og others 8
                                                            og others 9 \
   spl og mou 9
                  og others 6
0
                           NaN
                                         NaN
                                                       0.0
             NaN
                                                                     NaN
                          0.00
1
           0.00
                                         0.0
                                                       0.0
                                                                     0.0
2
           42.08
                          0.45
                                         0.0
                                                       0.0
                                                                     0.0
3
                          0.00
                                         0.0
                                                       0.0
                                                                     0.0
          43.29
4
                          0.00
                                         0.0
           5.98
                                                       0.0
                                                                     0.0
                    total og mou 7 total og mou 8
                                                       total og mou 9 \
   total og mou 6
0
                               0.00
                                                0.00
                                                                 0.00
              0.00
1
            40.31
                             178.53
                                              312.44
                                                                72.11
2
                                                               124.94
            155.33
                             412.94
                                              285.46
3
            223.23
                             135.31
                                              352.21
                                                               362.54
4
                                              208.16
            127.28
                             241.33
                                                               104.59
   loc_ic_t2t_mou_6 loc_ic_t2t_mou_7 loc_ic_t2t_mou_8
loc ic t2t mou 9 \
                 NaN
                                    NaN
                                                       0.16
NaN
                1.61
                                  29.91
                                                      29.23
116.09
              115.69
                                  71.11
                                                      67.46
148.23
               62.08
                                  19.98
                                                       8.04
3
41.73
              105.68
                                  88.49
                                                     233.81
154.56
   loc_ic_t2m_mou_6 loc_ic_t2m_mou_7 loc_ic_t2m_mou_8
loc ic t2m mou 9 ∖
                                    NaN
                                                       4.13
                 NaN
NaN
                                  65.38
               17.48
                                                     375.58
1
56.93
               14.38
                                  15.44
                                                      38.89
38.98
              113.96
                                  64.51
                                                      20.28
52.86
              106.84
                                 109.54
                                                     104.13
48.24
   loc_ic_t2f_mou_6 loc_ic_t2f_mou_7 loc_ic_t2f_mou_8
loc ic t2f mou 9 \
                 NaN
                                    NaN
                                                       1.15
NaN
                                   8.93
                0.00
                                                       3.61
1
0.00
               99.48
                                 122.29
                                                      49.63
158.19
```

3 65.59	57.43	2	7.09	19.84
	1 50		0 00	0.00
4 0.00	1.50		0.00	0.00
0.00				
loc i	.c mou 6 loc	ic mou 7 lo	c ic mou	8 loc ic mou 9
	:2 T mou 6 ∖			
0	_ NaN	NaN	5.4	4 NaN
NaN				
1	19.09	104.23	408.4	3 173.03
0.00				
2	229.56	208.86	155.9	9 345.41
72.41	222 40	111 50	40 1	0 100 10
3 43.48	233.48	111.59	48.1	8 160.19
43.40	214.03	198.04	337.9	4 202.81
0.00	214.03	190.04	337.9	4 202.01
0.00				
std i	c t2t mou 7	std ic t2t m	ou 8 std	ic t2t mou 9
	:2m_mou_6 \		_	
0	NaN		0.00	NaN
NaN				
1	0.00		2.35	0.00
5.90	71 20	2	0.60	40.44
2	71.29	2	8.69	49.44
45.18 3	66.44		0.00	120 04
1.33	00.44		0.00	129.84
4	0.00		0.86	2.31
1.93	0100		0.00	2.51
		std_ic_t2m_m	ou_8 std	_ic_t2m_mou_9
	2f_mou_6 \		0.00	
0 N = N	NaN		0.00	NaN
NaN 1	0.00	1	2 40	15 01
1 0.00	0.00	1	2.49	15.01
2	177.01	16	7.09	118.18
21.73	177.01	10	7.03	110.10
3	38.56		4.94	13.98
1.18				
4	0.25		0.00	0.00
0.00				
	.c_t2f_mou_7	std_ic_t2f_m	ou_8 std	_ic_t2f_mou_9
	:2o_mou_6 \		0.00	N = N
0 NaN	NaN		0.00	NaN
Nan 1	0.00		0.00	0.00
0.0	0.00		0.00	0.00
5.5				

2	58.34	43.2	23	3.86	
0.0					
3 0.0	0.00	0.0	90	0.00	
4	0.00	0.0	90	0.00	
0.0					
	ic_t2o_mou_7	std_ic_t2o_mou_	_8 std_ic_t2o	_mou_9 std_	ic_mou_6
0	NaN	0	. 0	NaN	NaN
1	0.0	0	. 0	0.0	5.90
2	0.0	0	. 0	0.0	139.33
3	0.0	0	. 0	0.0	45.99
4	0.0	0	. 0	0.0	1.93
std_i	ic_mou_7 std_	ic_mou_8 std_:	ic_mou_9 tota	l_ic_mou_6	
0	mou_7 \ NaN	0.00	NaN	0.00	
0.00	0.00	14 04	15 01	26 02	
1 104.23	0.00	14.84	15.01	26.83	
2 519.53	306.66	239.03	171.49	370.04	
3	105.01	4.94	143.83	280.08	
216.61 4	0.25	0.86	2.31	216.44	
198.29	0123	0.00	2.31	210111	
		tal_ic_mou_9 :	spl_ic_mou_6	spl_ic_mou_7	
Spt_1c_π 0	nou_8 \ 5.44	0.00	NaN	NaN	
0.0 1	423.28	188.04	0.00	0.0	
0.0	423.20	100.04	0.00	0.0	
2 0.0	395.03	517.74	0.21	0.0	
3	53.13	305.38	0.59	0.0	
0.0 4	338.81	205.31	0.00	0.0	
0.0	330.01	203.31	0.00	0.0	
spl i	ic mou 9 isd	ic_mou_6 isd_:	ic mou 7 isd	ic mou 8	
isd_ic_n	nou_9 \				
0 NaN	NaN	NaN	NaN	0.0	

1	0.00	1.83	0.00	0.0	
0.00 2	0.45	0.00	0.85	0.0	
0.01					
3 0.00	0.55	0.00	0.00	0.0	
4	0.18	0.00	0.00	0.0	
0.00	0.120				
ic ot	hors 6 is a	others 7 is a	others_8 ic_	others O	
	ch_num_6 \	others_/ ic_	others_o ic_	others_9	
0	NaN	NaN	0.0	NaN	
4					
1	0.00	0.00	0.0	0.00	
4					
2	0.93	3.14	0.0	0.36	
5	0.00	0.00	0.0	0.00	
3	0.00	0.00	0.0	0.80	
10	0.40	0.00	0.0	0.00	
4 5	0.48	0.00	0.0	0.00	
3					
total	rech num 7	total rech i	num_8 total_	rech num 9	
	ch_amt_6 \				
0	3		2	6	
362					
1	9		11	5	
74	1		2	7	
2 168	4		2	7	
3	11		18	14	
230	11		10		
4	6		3	4	
196					
_					
		total_rech_a	amt_8 total_	rech_amt_9	
max_rech			252	0	
0 252	252		252	Θ	
1	384		283	121	
44	304		203	121	
2	315		116	358	
86					
3	310		601	410	
60					
4	350		287	200	
56					
may r	ech amt 7 r	max_rech_amt_8	8 may recha	mt_9 date_of_last_r	ech 6
\ \	ecii_aiiit_/ I	iiax_i ecii_aiiit_	o max_recii_a	mr_9 date_01_tast_1	ecii_0
1					

1 154 65 50 6/29/2014 2 200 86 100 6/17/2014 3 50 50 50 50 6/28/2014 4 110 110 50 6/26/2014	0 252	252	0	6/21/2014
3 50 50 50 6/28/2014 4 110 110 50 6/26/2014 date_of_last_rech_7 date_of_last_rech_8 date_of_last_rech_9 \ 0 7/16/2014 8/8/2014 9/28/2014 1 7/31/2014 8/28/2014 9/30/2014 2 7/24/2014 8/31/2014 9/30/2014 3 7/31/2014 8/31/2014 9/30/2014 4 7/28/2014 8/9/2014 9/28/2014 last_day_rch_amt_6 last_day_rch_amt_7 last_day_rch_amt_8 \ 0 252 252 252 1 44 23 30 2 0 200 86 3 30 50 50 4 50 110 110 last_day_rch_amt_9 date_of_last_rech_data_6 date_of_last_rech_data_7 \ 0 6/21/2014 7/16/2014 NaN NaN NaN NaN NaN date_of_last_rech_data_8 date_of_last_rech_data_9 total_rech_data_6 \ 0 8/8/2014 NaN 1.0 1 8/10/2014 NaN NaN NaN date_of_last_rech_data_8 date_of_last_rech_data_9 total_rech_data_6 \ 0 8/8/2014 NaN NaN NaN 1.0 1 8/10/2014 NaN	1 154	65	50	6/29/2014
4 110 110 50 6/26/2014 date_of_last_rech_8 date_of_last_rech_9 0 7/16/2014 8/8/2014 9/28/2014 1 7/31/2014 8/28/2014 9/30/2014 2 7/24/2014 8/14/2014 9/29/2014 3 7/31/2014 8/31/2014 9/30/2014 4 7/28/2014 8/9/2014 9/28/2014 1 1ast_day_rch_amt_6 1ast_day_rch_amt_7 1ast_day_rch_amt_8 252 252 1 44 23 30 2 0 200 86 3 30 50 50 4 50 110 110 1 1 0 6/21/2014 7/16/2014 0 NaN 1 0 NaN NaN 3 30 NaN NaN NaN NaN NaN 4 50 6/4/2014 NaN 1 0 8/8/2014 NaN 1 0 8/8/2014 NaN 1 0 1 8/10/2014 NaN NaN NaN 2 NaN 9/17/2014 NaN 3 NaN	2 200	86	100	6/17/2014
date_of_last_rech_7 date_of_last_rech_8 date_of_last_rech_9	3 50	50	50	6/28/2014
0	4 110	110	50	6/26/2014
0	0 7/16/2014 1 7/31/2014 2 7/24/2014 3 7/31/2014	8/8/2014 8/28/2014 8/14/2014 8/31/2014		9/28/2014 9/30/2014 9/29/2014 9/30/2014
date_of_last_rech_data_7 \	0 252 1 44 2 0 3 30	252 23 200 50	last_day_	- 252 30 86 50
7/16/2014 1 0 NaN 7/25/2014 2 0 NaN NaN 3 30 NaN NaN 4 50 6/4/2014 NaN date_of_last_rech_data_8 date_of_last_rech_data_9 total_rech_data_6 0 8/8/2014 NaN 1.0 1 8/10/2014 NaN	date_of_last_rech_dat	a_7 \	_	
7/25/2014 2		6/21/	′2014	
2 0 NaN NaN 3 30 NaN NaN 4 50 6/4/2014 NaN date_of_last_rech_data_8 date_of_last_rech_data_9 total_rech_data_6 0 8/8/2014 NaN 1.0 1 8/10/2014 NaN NaN NaN NaN NaN NaN NaN			NaN	
3 30 NaN NaN 4 50 6/4/2014 NaN date_of_last_rech_data_8 date_of_last_rech_data_9 total_rech_data_6 0 8/8/2014 NaN 1.0 1 8/10/2014 NaN NaN 2 NaN 9/17/2014 NaN 3 NaN NaN NaN	2 0		NaN	
NaN 4 50 6/4/2014 NaN date_of_last_rech_data_8 date_of_last_rech_data_9 total_rech_data_6 0 8/8/2014 NaN 1.0 1 8/10/2014 NaN NaN 2 NaN 9/17/2014 NaN 3 NaN NaN NaN			NaN	
NaN date_of_last_rech_data_8 date_of_last_rech_data_9 total_rech_data_6 0 8/8/2014 NaN 1.0 1 8/10/2014 NaN NaN 2 NaN 9/17/2014 NaN 3 NaN NaN NaN	NaN			
\(\) 8/8/2014 NaN 1.0 1 8/10/2014 NaN NaN 2 NaN 9/17/2014 NaN 3 NaN NaN NaN		0/4/	72014	
1 8/10/2014 NaN NaN 2 NaN 9/17/2014 NaN 3 NaN NaN NaN		ata_8 date_of_last_re	ech_data_9	total_rech_data_6
2 NaN 9/17/2014 NaN 3 NaN NaN	0 8/8	/2014	NaN	1.0
3 NaN NaN NaN	1 8/10	/2014	NaN	NaN
	2	NaN	9/17/2014	NaN
4 NaN NaN 1.0	3	NaN	NaN	NaN
	4	NaN	NaN	1.0

total_rech_d max rech data 6		total_rech_dat	a_8 total_rech_d	ata_9
0	1.0		1.0	NaN
252.0 1	1.0		2.0	NaN
NaN 2	NaN		NaN	1.0
NaN				
3 NaN	NaN		NaN	NaN
4	NaN		NaN	NaN
56.0				
max_rech_dat	:a_7 m	nax_rech_data_8	max_rech_data_9	count_rech_2g_6
	52.0	252.0	NaN	0.0
1 15	54.0	25.0	NaN	NaN
2	NaN	NaN	46.0	NaN
3	NaN	NaN	NaN	NaN
4	NaN	NaN	NaN	1.0
count_rech_2	2g_7 c	count_rech_2g_8	count_rech_2g_9	count_rech_3g_6
0	0.0	0.0	NaN	1.0
1	1.0	2.0	NaN	NaN
2	NaN	NaN	1.0	NaN
3	NaN	NaN	NaN	NaN
4	NaN	NaN	NaN	0.0
count_rech_3 av_rech_amt_dat			count_rech_3g_9	
0	1.0	1.0	NaN	
252.0 1	0.0	0.0	NaN	
NaN				
2 NaN	NaN	NaN	0.0	
3 NaN	NaN	NaN	NaN	
4	NaN	NaN	NaN	
56.0				

		ata_7 av_r	ech_amt_data_8	av_rech_amt_o	data_9
0 _		252.0	252.0		NaN
30.1 1		154.0	50.0		NaN
0.00					
2 0.00		NaN	NaN		46.0
3 0.00		NaN	NaN		NaN
4		NaN	NaN		NaN
0.00					
0 1 2 3 4	ol_2g_mb_7 1.32 108.07 0.00 0.00	5.75	0.0 0.0 0.0	vol_3g_mb_6 vol_3g_mb_6 vol_3g_mb_6 vol_3g_mb_6 vol_90 vol	vol_3g_mb_7 \ 150.76 0.00 0.00 0.00 0.00
		vol_3g_mb_9	arpu_3g_6 a	rpu_3g_7 arpu	_3g_8
0	_3g_9 \ 109.61	0.00	212.17	212.17 23	12.17
NaN 1	0.00	0.00	NaN	0.00	0.00
NaN 2	0.00	8.42	NaN	NaN	NaN
2.84					
3 NaN	0.00	0.00	NaN	NaN	NaN
4 NaN	0.00	0.00	0.00	NaN	NaN
a 0 1 2 3 4	rpu_2g_6 ar 212.17 NaN NaN NaN 0.00	pu_2g_7 ar 212.17 28.61 NaN NaN NaN	pu_2g_8 arpu_2 212.17 7.60 NaN NaN NaN	2g_9 night_pcl NaN NaN 0.0 NaN NaN	<_user_6 \
n	ight_pck_use	r_7 night_	pck_user_8 ni	ght_pck_user_9	monthly_2g_6
ò		0.0	0.0	NaN	Θ
1		0.0	0.0	NaN	0
2		NaN	NaN	0.0	0
3		NaN	NaN	NaN	0
J			Hall	IVAIN	J

4		NaN	NaN		NaN	Θ
\	monthly_2g_7	monthly_2g_8	3 monthly_	_2g_9 s	achet_2g_6	sachet_2g_7
ò	0	(9	0	0	0
1	1	(9	0	0	0
2	0	(9	0	0	0
3	0	(9	0	0	0
4	0	(9	0	1	0
\	sachet_2g_8	sachet_2g_9	monthly_3g	g_6 mon	thly_3g_7	monthly_3g_8
0	0	0		1	1	1
1	2	0		0	0	0
2	0	1		0	0	0
3	0	0		0	0	0
4	0	0		0	Θ	0
0 1 2 3 4	monthly_3g_9 0 0 0 0	sachet_3g_6 0 0 0 0 0	sachet_3o	g_7 sac 0 0 0 0 0 0	het_3g_8 s 0 0 0 0 0	achet_3g_9 \ 0 0 0 0 0
ju	fb_user_6 fb l_vbc_3g \	o_user_7 fb_u				vbc_3g
0 0.	1.0	1.0	1.0	NaN	968	30.4
1 0.	NaN	1.0	1.0	NaN	1006	0.0
2	NaN	NaN	NaN	1.0	1103	0.0
3 0.	NaN	NaN	NaN	NaN	2491	0.0
0. 4 0.	0.0	NaN	NaN	NaN	1526	0.0
	jun_vbc_3g s	sep_vbc_3g				

```
0
       101.20
                      3.58
1
         0.00
                      0.00
2
         4.17
                      0.00
3
         0.00
                      0.00
4
         0.00
                      0.00
df.shape
(99999, 226)
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(\overline{3}5), object(12)
memory usage: 172.4+ MB
df.describe()
       mobile number
                       circle id
                                   loc_og_t2o_mou
                                                    std_og_t2o_mou \
        9.999900e+04
                         99999.0
                                           98981.0
                                                            98981.0
count
                            109.0
                                               0.0
                                                                0.0
mean
        7.001207e+09
                              0.0
                                               0.0
                                                                0.0
std
        6.956694e+05
        7.000000e+09
                            109.0
                                               0.0
                                                                0.0
min
25%
        7.000606e+09
                            109.0
                                               0.0
                                                                0.0
        7.001205e+09
                            109.0
                                               0.0
50%
                                                                0.0
75%
        7.001812e+09
                            109.0
                                               0.0
                                                                0.0
        7.002411e+09
                            109.0
                                               0.0
                                                                0.0
max
       loc ic t2o mou
                               arpu_6
                                              arpu_7
                                                             arpu 8
arpu 9
                                                      99999.000000
                        99999.000000
                                       99999,000000
count
               98981.0
99999.000000
                   0.0
                          282.987358
                                         278.536648
                                                        279.154731
mean
261.645069
                          328.439770
                                         338.156291
                                                        344.474791
std
                   0.0
341.998630
                       -2258.709000
                                       -2014.045000
                                                       -945.808000
min
                   0.0
1899.505000
25%
                   0.0
                           93.411500
                                           86.980500
                                                         84.126000
62.685000
50%
                   0.0
                          197.704000
                                         191.640000
                                                        192.080000
176.849000
75%
                   0.0
                          371.060000
                                         365.344500
                                                        369.370500
353.466500
                        27731.088000
                                       35145.834000
                                                      33543.624000
                   0.0
max
38805.617000
        onnet mou 6
                       onnet mou 7
                                      onnet mou 8
                                                     onnet mou 9
offnet mou 6 \
```

count 9	6062.000000	96140.000000	94621.000000	92254.000000	
mean 197.9355	132.395875	133.670805	133.018098	130.302327	
std	297.207406	308.794148	308.951589	308.477668	
316.8516 min	0.000000	0.000000	0.000000	0.000000	
0.000000 25%	7.380000	6.660000	6.460000	5.330000	
34.73000 50% 96.31000	34.310000	32.330000	32.360000	29.840000	
75% 231.8600	118.740000	115.595000	115.860000	112.130000	
	7376.710000	8157.780000	10752.560000	10427.460000	
	ffnet_mou_7	offnet_mou_8	offnet_mou_9	roam_ic_mou_6	
count 9	mou_7 \ 6140.000000	94621.000000	92254.000000	96062.000000	
96140.00 mean	197.045133	196.574803	190.337222	9.950013	
7.149898 std	325.862803	327.170662	319.396092	72.825411	
73.44794	0.000000	0.000000	0.000000	0.00000	
0.000000 25%	32.190000	31.630000	27.130000	0.000000	
0.000000 50%	91.735000	92.140000	87.290000	0.000000	
0.000000 75%	226.815000	228.260000	220.505000	0.000000	
0.000000 max 15371.04	9667.130000 0000	14007.340000	10310.760000	13724.380000	
count mean std min 25% 50% 75% max r loc_og_t	oam_ic_mou_8 94621.000000 7.292981 68.402466 0.000000 0.000000 0.000000 13095.360000 oam_og_mou_8 2t_mou_7 94621.000000	roam_ic_mou_9 92254.000000 6.343843 57.137533 0.000000 0.0000000 0.0000000 8464.0300000 roam_og_mou_9	96062.0000 1 13.9113 7 71.4431 9 0.0000 9 0.0000 9 0.0000 9 0.0000 9 3775.1100 9 loc_og_t2t_	00 96140.000000 37 9.818732 96 58.455762 00 0.000000 00 0.000000 00 0.000000 00 0.000000 00 2812.040000	\
96140.00		92234.000000	y 90002.0	00000	

mean 9 46.473010	.971890	8.555519	47.100763
	.713221	58.438186	150.856393
min 0	.000000	0.000000	0.000000
	.000000	0.000000	1.660000
	.000000	0.000000	11.910000
	.000000	0.00000	40.960000
39.910000 max 5337 7400.660000	.040000	4428.460000	6431.330000
		loc_og_t2t_mou_9	loc_og_t2m_mou_6
	621.000000	92254.000000	96062.000000
96140.000000 mean 91.397131	45.887806	44.584446	93.342088
	151.184830	147.995390	162.780544
min 0.000000	0.000000	0.000000	0.000000
25% 10.025000	1.600000	1.360000	9.880000
50% 40.430000	11.730000	11.260000	41.030000
75% 107.560000	40.110000	39.280000	110.390000
	752.560000	10389.240000	4729.740000
loc_og loc_og_t2f_mo	_t2m_mou_8	loc_og_t2m_mou_9	loc_og_t2f_mou_6
	621.000000	92254.000000	96062.000000
mean 3.792985	91.755128	90.463192	3.751013
	156.537048	158.681454	14.230438
min 0.000000	0.000000	0.000000	0.000000
25% 0.000000	9.810000	8.810000	0.000000
50% 0.000000	40.360000	39.120000	0.000000
	109.090000	106.810000	2.080000

2 0000	.00						
2.0900 max 1196.4	4961.3	30000	4429	.880000	1466	.03000	0
100 00	loc_og_t2f_		loc_og_t2	f_mou_9	loc_og_t2	c_mou_	6
count	_t2c_mou_7 94621.0 000000		92254	.000000	96062	.00000	0
mean 1.3685	3.6	77991	3	.655123	1	. 12305	6
std	13.2	70996	13	.457549	5	. 44894	6
7.5334 min	0.0	00000	0	.000000	0	.00000	0
0.0000 25%	0.0	00000	0	.000000	0	.00000	0
0.0000 50%	0.0	00000	0	.000000	0	.00000	0
0.0000 75%	2.0	40000	1	.940000	0	.00000	0
0.0000 max 916.24	928.4	90000	927	.410000	342	.86000	0
	loc_og_t2c_	mou_8	loc_og_t2	c_mou_9	loc_og_mo	u_6 l	oc_og_mou_7
count	94621.0	00000	92254	.000000	96062.000	000 9	6140.000000
mean	1.4	33821	1	.232726	144.201	175	141.670476
std	6.7	83335	5	.619021	251.751	489	248.731086
min	0.0	00000	0	.000000	0.000	000	0.000000
25%	0.0	00000	0	.000000	17.110	000	17.480000
50%	0.0	00000	0	.000000	65.110	000	63.685000
75%	0.0	00000	0	.000000	168.270	900	164.382500
max	502.0	90000	339	.840000	10643.380	000	7674.780000
	loc_og_mou_	8 loc	_og_mou_9	std_og_	t2t_mou_6	std_o	g_t2t_mou_7
count	94621.00000	0 922	54.000000	960	62.000000	9	6140.000000
mean	141.32820	9 1	38.709970		79.829870		83.299598
std	245.91431	.1 2	45.934517	2	52.476533		263.631042
min	0.00000	0	0.000000		0.000000		0.000000

25%	1	7.110000	15	.560000		0.000000		0.000000
50%	6.	3.730000	61	. 840000		0.000000		0.000000
75%		6.110000		. 225000		30.807500		31.132500
75%								
max	11039	9.910000	11099	.260000	73	66.580000	8	133.660000
	- 4 -1		0 -		0			
std og		og_t2t_mou mou_7 \	_8 S	td_og_t2t_	mou_9	std_og_t2	m_mou_6	
count		$946\overline{2}1.0000$	00	92254.0	00000	96062	2.000000	
96140.0 mean		ິນ 83.2826	73	82.3	42919	87	7.299624	
90.8041	L37	265 4060	00	267 1	84991	25.5	617050	
std 269.347	7911	265.4860	90	207.1	84991	255	5.617850	
min	10	0.0000	00	0.0	00000	0	0.000000	
0.00000 25%	טע	0.0000	00	0.0	00000	0	0.000000	
0.00000 50%	00	0.0000	00	0.0	00000		3.950000	
3.63500	00	0.0000	00	0.0	00000		.930000	
75% 54.0400	000	30.5800	00	28.2	30000	53	3.290000	
max	000	8014.4300	00	9382.5	80000	8314	1.760000	
9284.74	10000							
		og_t2m_mou	_8 s	td_og_t2m_	mou_9	std_og_t2	2f_mou_6	
std_og_ count		nou_7 \ 94621.0000	00	92254.0	00000	96062	2.000000	
96140.0		9						
mean 1.11501	LO	89.8383	90	86.2	76622		1.129011	
std	0.0	271.7577	83	261.4	07396	7	7.984970	
8.59940 min	96	0.0000	00	0.0	00000	0	0.000000	
0.00000	00	0 0000	00	0.0	00000		000000	
25% 0.00000	00	0.0000	00	0.0	00000	ť	0.000000	
50%	10	3.3100	00	2.5	00000	0	0.000000	
0.00000 75%	טע	52.4900	00	48.5	60000	(0.00000	
0.00000		13950.0400	ൈ	10223.4	30000	629	3.560000	
max 544.630		13930.0400	00	10223.4	20000	020		
	std	og t2f mou	8 s	td og t2f	mou 9	std og t2	2c mou 6	
std_og_		mou_7 \						

count	94621.000000	92254	1.000000	9606	52 0
96140.0				3000	
mean 0.0	1.067792]	1.042362		0.0
std	7.905971	3	3.261770		0.0
0.0 min	0.000000	(0.000000		0.0
0.0 25%	0.000000	(0.000000		0.0
0.0					
50% 0.0	0.000000	(0.00000		0.0
75%	0.00000	(0.00000		0.0
0.0 max	516.910000	000	3.490000		0.0
0.0	310.910000	000	1490000		0.0
	d_og_t2c_mou_8	std_og_t2	2c_mou_9	std_og_mou_6	std_og_mou_7
\ count	94621.0		92254.0	96062.000000	96140.000000
	0.0		0.0	168.261218	175.221436
mean	0.0		0.0	100.201210	1/3.221430
std	0.0		0.0	389.948499	408.922934
min	0.0		0.0	0.000000	0.000000
25%	0.0		0.0	0.000000	0.000000
50%	0.0		0.0	11.640000	11.090000
75%	0.0		0.0	144.837500	150.615000
max	0.0		0.0	8432.990000	10936.730000
st isd og mo		d_og_mou_9	isd_og_	mou_6 isd_og_	_mou_7
count 94	$6\overline{2}1.000000$ 92	254.000000	96062.0	00000 96140.0	000000
		169.664466	0.7	98277 0.7	776572
0.791247 std	411.633049	405.138658	25.7	65248 25.6	603052
25.544471 min	0.000000	0.000000	0.0	00000 0.0	00000
0.000000					
25% 0.000000	0.000000	0.000000	0.0	00000 0.0	00000
50% 0.000000	10.410000	8.410000	0.0	00000 0.0	000000
3100000					

75% 1.000000	47.940000	142.105000	0.000000	0.00000
max 139	80.060000	11495.310000	5900.660000	5490.280000
5681.54000	Θ			
isd spl og mou	_og_mou_9	spl_og_mou_6	spl_og_mou_7	spl_og_mou_8
count 922	54.000000	96062.000000	96140.000000	94621.000000
92254.0000 mean	0.723892	3.916811	4.978279	5.053769
4.412767 std	21.310751	14.936449	20.661570	17.855111
16.328227 min	0.000000	0.000000	0.000000	0.000000
0.000000				
25% 0.000000	0.000000	0.000000	0.00000	0.000000
50% 0.000000	0.000000	0.00000	0.00000	0.00000
75%	0.000000	2.430000	3.710000	3.990000
3.230000 max 42	44.530000	1023.210000	2372.510000	1390.880000
1635.71000				
	_others_6	og_others_7	og_others_8	og_others_9
total_og_m count 960	ou_6 \ 62.000000	96140.000000	94621.000000	92254.000000
99999.0000	00			
mean 305.133424	0.454157	0.030235	0.033372	0.047456
std 463.419481	4.125911	2.161717	2.323464	3.635466
min	0.000000	0.00000	0.000000	0.000000
0.000000 25%	0.000000	0.00000	0.000000	0.000000
44.740000 50%	0.000000	0.000000	0.000000	0.000000
145.140000				
75% 372.860000	0.000000	0.000000	0.00000	0.000000
max 8 10674.0300		370.130000	394.930000	787.790000
		7 +o+ol on mo	0 +-+-1	
loc_ic_t2t			u_8 total_og_	mou_9
count 9 96062.0000	9999.00000 00	99999.000	99999.0	00000
mean	310.23117	304.119	513 289.2	79198
47.922365 std	480.031178	3 478.150	031 468.9	80002

140.258485 min	0.000000	0.000000	0.000000
0.000000 25%	43.010000	38.580000	25.510000
2.990000 50% 15.690000	141.530000	138.610000	125.460000
75% 46.840000	378.570000	369.900000	353.480000
	1365.310000)	14043.060000	11517.730000
loc_ loc ic t2m		loc_ic_t2t_mou_8	loc_ic_t2t_mou_9
count 96062.00000	$961\overline{40.000000}$	94621.000000	92254.000000
mean	47.990520	47.211362	46.281794
107.475650 std	145.795055	137.239552	140.130610
171.713903 min	0.000000	0.000000	0.000000
0.000000 25%	3.230000	3.280000	3.290000
17.290000 50%	15.740000	16.030000	15.660000
56.490000 75%	45.810000	46.290000	45.180000
132.387500 max	9324.660000	10696.230000	10598.830000
4693.860000			
loc_ loc_ic_t2f_		loc_ic_t2m_mou_8	loc_ic_t2m_mou_9
count 96062.00000	96140.000000 90	94621.000000	92254.000000
mean 12.084305	107.120493	108.460515	106.155471
std 40.140895	169.423620	169.723759	165.492803
min 0.000000	0.000000	0.000000	0.000000
25% 0.000000	18.590000	18.930000	18.560000
50%	57.080000	58.240000	56.610000
0.880000 75%	130.960000	133.930000	130.490000
8.140000 max	4455.830000	6274.190000	5463.780000
1872.340000)		

	oc_ic_t2f_m	ou_7 lo	c_ic_t21	_mou_8	loc_ic_t2	2f_mou_9		
loc_ic_m count 96062.00	96140.00	0000	94621.	000000	92254	1.000000		
mean	12.59	9697	11.	751834	12	2.173105		
167.4910 std	42.97	7442	39.	125379	43	3.840776		
254.12402 min	29 0.00	0000	0.	000000	(0.00000		
0.000000 25%	0.00	0000	0.	000000	(0.00000		
30.39000	9							
50% 92.16000	0.93 9	0000	Θ.	930000	(0.960000		
75% 208.0750	8.28	2500	8.	110000	8	3.140000		
max	1983.01	0000	2433.	060000	4318	3.280000		
7454.630	900							
count 9 mean std min 25% 50% 75%	oc_ic_mou_7 6140.000000 167.719540 256.242707 0.000000 32.460000 92.550000 205.837500 9669.910000	94621. 167. 250. 0. 32. 93. 207.	000000 432575	$ 922\overline{5}4.\overline{0} \\ 164.6 \\ 249.8 \\ 0.0 \\ 32.2 \\ 91.6 $	000 0 0 19293 45070 00000 90000 40000 37500	d_ic_t2t_md 96062.000 9.575 54.330 0.000 0.000 4.060 5459.560	993 607 000 000 000	
					std ic t2			
std_ic_t	td_ic_t2t_m 2m_mou_6 \	_						
count 96062.00	96140.00 9000	0000	94621.	000000	92254	1.000000		
mean	10.01	1904	9.	883921	Ć	9.432479		
20.72224 std	57.41	1971	55.	073186	53	3.376273		
80.79341 min	4 0.00	0000	0.	000000	(0.00000		
0.000000 25%	0.00			000000		0.000000		
0.000000								
50% 2.030000	0.00	0000	0.	000000	(0.00000		
75%	4.23	0000	4.	080000	3	3.510000		
15.03000 max 5647.160	5800.93	0000	4309.	290000	3819	9.830000		
	td_ic_t2m_m 2f_mou_6 \		d_ic_t2m	n_mou_8	std_ic_t2	2m_mou_9		

count 96062.0000	96140.000000	94621.000000	92254.000000
mean	21.656415	21.183211	19.620913
2.156397	06 [21202	02 (02565	74 012050
std 16.495594	86.521393	83.683565	74.913050
min	0.000000	0.000000	0.00000
0.000000	0.00000	0.00000	0.00000
25% 0.000000	0.000000	0.000000	0.000000
50%	2.040000	2.030000	1.740000
0.000000	15 740000	15 260000	14 260000
75% 0.000000	15.740000	15.360000	14.260000
max	6141.880000	5645.860000	5689.760000
1351.110000	9		
	_ic_t2f_mou_7	std_ic_t2f_mou_8	std_ic_t2f_mou_9
std_ic_t2o_ count	_mou_6 \ 96140.000000	94621.000000	92254.000000
96062.0			
mean 0.0	2.216923	2.085004	2.173419
std	16.454061	15.812580	15.978601
0.0	0.00000	2 22222	0.00000
min 0.0	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000
0.0	0.000000	0.00000	0.00000
50% 0.0	0.000000	0.000000	0.000000
75%	0.000000	0.000000	0.000000
0.0	1126 00000	1204 00000	1421 060000
max 0.0	1136.080000	1394.890000	1431.960000
	÷- +2 7	atal da 10 a man 0	-1-d -1-a -1-a -1-a -1-a -1-a -1-a -1-a
std ic mou		std_ic_t2o_mou_8	sta_1c_t2o_mou_9
count	96140.0	94621.0	92254.0
96062.00000		0.0	0.0
mean 32.457179	0.0	0.0	0.0
std	0.0	0.0	0.0
106.283386	0.0	0.0	0.0
min 0.000000	0.0	0.0	0.0
25%	0.0	0.0	0.0
0.000000 50%	0.0	0.0	0.0
50 0	0.0	0.0	0.0

Г 0000	00			
5.8900 75%	90	0.0	0.0	0.0
26.930	000	0.0	0.0	0.0
max		0.0	0.0	0.0
5712.1	10000			
	std ic mou 7	ctd ic mou 9	std ic mou 9	total_ic_mou_6 \
count	96140.000000			
mean	33.887833			
std	113.720168	110.127008	101.982303	291.651671
min	0.00000			
25%	0.000000			
50% 75%	5.960000 28.310000			
max	6745.760000			
max	07 13170000	33371110000	33301000000	77101110000
	total_ic_mou			
count	99999.0000			000000 96062.000000
mean std	202.8530 298.1249			214260 0.061557 823024 0.160920
min	0.0000			000000 0.000000
25%	41.1900			370000 0.000000
50%	116.3400	00 114.66		890000 0.000000
75%	250.6600			320000 0.000000
max	9699.0100	00 10830.38	0000 10796.	590000 19.760000
	spl_ic_mou_7	spl_ic_mou_8	spl_ic_mou_9	isd_ic_mou_6
	_mou_7 \	0.4621 000000	02254 000000	00000 000000
96140.	96140.000000	94621.000000	92254.000000	96062.000000
mean	0.033585	0.040361	0.163137	7.460608
8.3349				
std	0.155725	0.146147	0.527860	59.722948
65.219	829 0.000000	0 00000	0 00000	0.00000
min 0.0000		0.00000	0.00000	0.00000
25%	0.00000	0.00000	0.000000	0.00000
0.0000		0.00000	0.00000	0.00000
50% 0.0000	0.000000	0.000000	0.000000	0.000000
75%	0.000000	0.00000	0.060000	0.000000
0.0000		0.00000	0.00000	0.00000
max	21.330000	16.860000	62.380000	6789.410000
5289.5	40000			
	isd ic mou 8	isd ic mou 9	ic others 6	ic others 7
ic_oth	ers_8 \			
count		92254.000000	96062.000000	96140.000000
94621.		0.00000	0.054656	1 012060
mean	8.442001	8.063003	0.854656	1.012960

0.970800 std	63.813098	63.505379	11.955164	12.673099
13.284348				
min 0.000000	0.000000	0.000000	0.000000	0.00000
25% 0.000000	0.000000	0.000000	0.000000	0.000000
50%	0.000000	0.00000	0.000000	0.000000
0.000000 75%	0.000000	0.00000	0.000000	0.00000
0.000000 max 41 2327.51000	127.010000 00	5057.740000	1362.940000	1495.940000
		total_rech_num	_6 total_rec	h_num_7
	n_num_8 \ 254.000000	99999.0000	99999	.000000
mean	1.017162	7.5588	96 7	.700367
7.212912 std	12.381172	7.0784	05 7	. 070422
7.203753 min	0.000000	0.0000	90 0	.000000
0.000000 25%	0.000000	3.0000	00 3	. 000000
3.000000 50%	0.000000	6.0000	99 6	. 000000
5.000000				
75% 9.000000	0.000000	9.0000		.000000
max 10 196.000000	005.230000)	307.0000	00 138	.000000
	al_rech_num	_9 total_rech	_amt_6 total	_rech_amt_7
count	n_amt_8 \ 99999.0000	99999.	000000 9	9999.000000
99999.0000 mean	6.8930	19 327.	514615	322.962970
324.157122 std	2 7.0962	61 398.	019701	408.114237
416.540455 min	0.0000	00 0.	000000	0.000000
0.000000 25%	3.0000		000000	100.000000
90.000000				
50% 225.000000			000000	220.000000
75% 434.500000	9.0000)	00 437.	500000	428.000000

max 45320.000000	131.000000	35190.000000	40335.000000	
total max_rech_amt		nax_rech_amt_6 ma	x_rech_amt_7	
count 9	9999.000000	99999.000000	99999.000000	
mean 107.728207	222 24525	104.637486	104.752398	
std 126.902505	404.588583	120.614894	124.523970	
min 0.000000	0.000000	0.000000	0.000000	
25%	52.000000	30.000000	30.000000	
30.000000 50%	200.000000	110.000000	110.000000	
98.000000 75%	415.000000	120.000000	128.000000	
	7235.000000	4010.000000	4010.000000	
4449.000000				
count 999 mean 1 std 1 min 25% 50% 75% 1 max 33 last_ count mean	99.000000 01.943889 .25.375109 0.000000 28.000000 61.000000 .44.000000 .99.000000 day_rch_amt_8 99999.000000 62.641716	99999.000000 63.156252 97.356649 0.000000 0.000000 30.000000 110.000000 4010.000000 last_day_rch_amt 99999.0000 43.9012	0.000000 0.000000 30.000000 110.000000 4010.000000 25153.000000 2.463802	5 \ 9
std min 25% 50% 75% max	104.431816 0.000000 0.000000 30.000000 130.000000 4449.000000	90.8097 0.0000 0.0000 0.0000 50.0000 3399.0000	$egin{array}{cccccccccccccccccccccccccccccccccccc$)))
	_rech_data_7 25571.000000 2.666419 3.031593 1.000000 1.000000 1.000000 3.000000	total_rech_data_8 26339.000000 2.651999 3.074987 1.000000 1.000000 3.000000	$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	\

max	54.00000	0 60.000	0000 84.0	00000
	_rech_data_6	max_rech_data_7	max_rech_data_8	
	$251\overline{5}3.000000$	25571.000000	26339.000000	
25922.00000 mean	126.393392	126.729459	125.717301	
124.94144 std	108.477235	109.765267	109.437851	
111.36376 min	1.000000	1.000000	1.000000	
1.00000 25%	25.000000	25.000000	25.000000	
25.00000				
50% 145.00000	145.000000	145.000000	145.000000	
75% 179.00000	177.000000	177.000000	179.000000	
max 1555.00000	1555.000000	1555.000000	1555.000000	
	nt rech 2g 6	count rech 2a 7	count rech 2g 8	
count_rech_		25571.000000	26339.000000	
25922.00000	00			
mean 1.781807	1.864668	2.044699	2.016288	
std 2.214701	2.570254	2.768332	2.720132	
min 0.000000	0.000000	0.000000	0.000000	
25% 1.000000	1.000000	1.000000	1.000000	
50% 1.000000	1.000000	1.000000	1.000000	
75% 2.000000	2.000000	2.000000	2.000000	
max	42.000000	48.000000	44.000000	
40.000000				
count_rech_		count_rech_3g_7	count_rech_3g_8	
count 25922.00000	25153.000000 00	25571.000000	26339.000000	
mean 0.659363	0.599133	0.621720	0.635711	
std 1.411513	1.274428	1.394524	1.422827	
min	0.000000	0.00000	0.000000	
0.000000				

25% 0.000000	0.00000	00	0.6	00000		0.0000	00	
50%	0.0000	00	0.0	900000		0.0000	00	
0.000000	1 00000	١٥	1 (200000		1 0000	0.0	
75% 1.000000	1.00000	90	1.0	00000		1.0000	00	
max	29.00000	00	35.0	00000		45.0000	00	
49.000000								
av_re count mean	ch_amt_da 25153.00 192.60	0000		25571.0			_amt_data_8 6339.000000 197.526489	\
std min	192.64 1.00	16318 00000		196.7 0.5	91224 00000		191.301305 0.500000	
25% 50%	82.00 154.00				00000		87.000000 154.000000	
75%	252.00	0000		252.0	00000		252.000000	
max	7546.00	00000		4365.0	00000		4076.000000	
av_re	ech_amt_da 25922.00		vol_2g		vol_2 99999	2g_mb_7 .000000	vol_2g_mb_8 99999.000000	
mean std	192.73 188.40			904956 356445		. 229937	50.170154 212.347892	
min		00000		00000		.000000	0.000000	
25%	69.00			00000		.000000	0.000000	
50% 75%	164.00 252.00			00000		.000000	0.000000 0.000000	
max	4061.00		10285.9			550000	11117.610000	
vol	_2g_mb_9	vol :	3g mb 6	vol	3g mb 7	7 vol	3g_mb_8	
vol_3g_mb_9	\	_		_				
count 99999 99999.000000		99999	.000000	99999	.000000	99999	.000000	
mean 44	.719701	121	.396219	128	.995847	7 135	.410689	
136.056613 std 198	3.653570	544	.247227	541	.494013	3 558	.775335	
577.394194								
min 0 0.000000	0.000000	0	.000000	0	.000000	9 0	.000000	
	0.000000	0	.000000	0	.000000	9 0	.000000	
0.000000 50% 0	0.00000	O.	.000000	G	.00000	a 6	.000000	
0.000000	.000000	U	.000000	C	. 00000	9 6	.000000	
75% 0 0.000000	0.000000	0	.000000	0	.000000	9 0	.000000	
		45735	.400000	28144	.120000	30036	.060000	
arpu_2g_6 \	rpu_3g_6	ar	pu_3g_7	ar	pu_3g_8	3 ar	pu_3g_9	

count 25153.0 25153.000000	000000 25	571.000000	26339.000000	25922.000000	
	555057	89.384120	91.173849	100.264116	
	124653	195.893924	188.180936	216.291992	
	320000	-26.040000	-24.490000	-71.090000 -	
	000000	0.000000	0.000000	0.000000	
	180000	0.420000	0.880000	2.605000	
	70000	119.560000	122.070000	140.010000	
max 6362.2 6433.760000	280000 4	980.900000	3716.900000	13884.310000	
count 25571.0 mean 85.9 std 176.3 min -15.4 25% 0.0	u_2g_7 000000 26 014450 079871 180000 000000	arpu_2g_8 339.000000 86.599478 168.247852 -55.830000 0.000000 9.270000	arpu_2g_9 25922.000000 93.712026 171.384224 -45.740000 0.000000 14.800000	night_pck_user_6 25153.000000 0.025086 0.156391 0.000000 0.000000	\
	70000	122.070000 3483.170000	140.010000 3467.170000	0.000000 1.000000	
	ock_user_7			pck user 9	
monthly_2g_6	571.000000			25922.000000	
99999.000000 mean	0.023034	· 0	.020844	0.015971	
0.079641 std	0.150014		. 142863	0.125366	
0.295058					
min 0.000000	0.000000	0	.000000	0.000000	
25% 0.000000	0.000000	0	.000000	0.000000	
50%	0.000000	0	.000000	0.000000	
0.000000 75%	0.00000	0	.000000	0.00000	
0.000000 max 4.000000	1.000000) 1	.000000	1.000000	
monthly		onthly_2g_8	monthly_2g_9	sachet_2g_6	
sachet_2g_7 \count 99999.0		999.000000	99999.000000	99999.000000	

mean 0.439634	0.083221	0.081001	0.068781	0.389384
std	0.304395	0.299568	0.278120	1.497320
1.636230 min	0.000000	0.000000	0.000000	0.000000
0.000000 25%	0.000000	0.000000	0.000000	0.00000
0.000000 50%	0.000000	0.000000	0.000000	0.00000
0.000000 75%	0.000000	0.000000	0.000000	0.00000
0.000000 max	5.000000	5.000000	4.000000	42.000000
48.000000	3.000000	3.00000	4100000	42.00000
	achet_2g_8	sachet_2g_9	monthly_3g_6	monthly_3g_7
monthly_3 count 99 99999.000	$9\overline{9}9.000000$	99999.000000	99999.000000	99999.000000
mean	0.450075	0.393104	0.075921	0.078581
0.082941 std	1.630263	1.347140	0.363371	0.387231
0.384947 min	0.000000	0.000000	0.000000	0.00000
0.000000 25%	0.000000	0.000000	0.000000	0.00000
0.000000 50%	0.000000	0.000000	0.000000	0.00000
0.000000 75%	0.000000	0.000000	0.000000	0.000000
0.000000 max	44.000000	40.000000	14.000000	16.000000
16.000000	44.000000	40.000000	14.000000	10.00000
mor sachet 3g	nthly_3g_9	sachet_3g_6	sachet_3g_7	sachet_3g_8
count 99	999.000000	99999.000000	99999.000000	99999.000000
99999.0000 mean	0.086341	0.074781	0.080401	0.084501
0.084581 std	0.384978	0.568344	0.628334	0.660234
0.650457 min	0.000000	0.000000	0.000000	0.00000
0.000000 25%	0.000000	0.000000	0.000000	0.000000
0.000000 50%	0.000000	0.000000	0.000000	0.00000
0.000000 75%	0.000000	0.000000	0.000000	0.000000
150	0.00000	0.00000	0.00000	0.00000

### ##################################						
#9.000000 fb_user_6	0.000000					
fb_user_6 fb_user_7 fb_user_8 fb_user_9 aon \ count 25153.000000 25571.000000 26339.000000 25922.000000 aog 9999.000000 aean 0.914404 0.908764 0.890808 0.860968 1219.854749 at 0.279772 0.287950 0.311885 0.345987 254.733842 ain 0.000000 0.000000 0.000000 0.000000 0.000000	max		29.000000	35.000000	41.000000	
aon \ count 25153.000000 25571.000000 26339.000000 25922.0000000 nean	49.00000	U				
count 25153.000000 25571.000000 26339.000000 25922.000000 dean 0.914404 0.908764 0.890808 0.860968 1219.854749 0.287950 0.311885 0.345987 054.733842 0.000000 0.000000 0.000000 0.000000 180.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1807.500000 1.000000 1.000000 1.000000 1.000000 18337.000000 1.000000 1.000000 1.000000 1.000000 1807.500000 1.000000 1.000000 1.000000 1.000000 1807.500000 1.000000 1.000000 1.000000 1.000000 1807.500000 1.000000 1.000000 1.000000 1.000000 1807.500000 1.000000 1.000000 1.000000 1.000000 1807.500000 1.0000000 1.000000 1.000000 1.000000 1807.500000 1.0000000 1.0000000		fb_user_6	fb_user_7	fb_user_8	fb_user_9	
99999.000000 mean	aon \	F1F2 000000	25571 000000	26220 000000	25022 000000	
mean 0.914404 0.908764 0.890808 0.860968 1219.854749 15td 0.279772 0.287950 0.311885 0.345987 12954.733842 110 0.000000 0.000000 0.000000 0.000000 125% 1.000000 1.000000 1.000000 1.000000 1363.0000000 1363.0000000 1363.0000000 1363.0000000 1363.0000000 1363.0000000000 1363.0000000 1363.0000000 1363.0000000 1363.000000000000000 1363.00000000000000000000000000000000000			255/1.000000	26339.000000	25922.000000	
6td 0.279772 0.287950 0.311885 0.345987 054.733842 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 1.0000000 1.000000 1.000000 1.000000 1.000000 1.0000000 <td< td=""><td>mean</td><td></td><td>0.908764</td><td>0.890808</td><td>0.860968</td></td<>	mean		0.908764	0.890808	0.860968	
0.000000						
min 0.000000 0.000000 0.000000 0.000000 0.000000	std		0.287950	0.311885	0.345987	
180.000000 25%			0.00000	0.00000	0.00000	
467.000000 50%			0.00000	0.00000	0.00000	
1.000000 1.0000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.0000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.0000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.0000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.0000000 1.000000 1.000000 1.0000000 1.000000 1.0000000 1.000000 1.00000000	25%		1.000000	1.000000	1.000000	
363.000000 75% 1.000000 1.000000 1.000000 1.000000 1807.500000 max 1.000000 1.000000 1.000000 1.000000 4337.000000 aug_vbc_3g jul_vbc_3g jun_vbc_3g sep_vbc_3g count 99999.000000 99999.000000 99999.000000 99999.000000 mean 68.170248 66.839062 60.021204 3.299373 std 267.580450 271.201856 253.938223 32.408353 min 0.000000 0.000000 0.000000 0.000000 25% 0.000000 0.000000 0.000000 0.000000 25% 0.000000 0.000000 0.000000 0.000000 25% 0.000000 0.000000 0.000000 0.0000000 25% 0.000000 0.000000 0.000000 0.000000000			1 000000	1 000000	1 000000	
1807.500000 max			11000000	11000000	1100000	
max 1.000000 1.000000 1.000000 4337.000000 aug_vbc_3g jul_vbc_3g jun_vbc_3g sep_vbc_3g count 99999.000000 99999.000000 99999.000000 99999.000000 99999.000000 nean 68.170248 66.839062 60.021204 3.299373 std 267.580450 271.201856 253.938223 32.408353 nin 0.000000 0.000000 0.000000 0.000000 25% 0.000000 0.000000 0.000000 0.000000 75% 0.000000 0.000000 0.000000 0.000000	75%		1.000000	1.000000	1.000000	
4337.000000 aug_vbc_3g jul_vbc_3g jun_vbc_3g sep_vbc_3g count 99999.000000 99999.000000 99999.000000 99999.000000 99999.000000 99999.000000 99999.000000 99999.000000 99999.000000 99999.000000 99999.000000 99999.000000 99999.000000 99999.000000 99999.000000 99999.000000 99999.000000 99999.000000 9999999.0000000 9999999.0000000 <th cols<="" td=""><td></td><td></td><td>1 000000</td><td>1 000000</td><td>1 000000</td></th>	<td></td> <td></td> <td>1 000000</td> <td>1 000000</td> <td>1 000000</td>			1 000000	1 000000	1 000000
aug_vbc_3g jul_vbc_3g jun_vbc_3g sep_vbc_3g count 99999.000000 99999.000000 99999.000000 99999.000000 mean 68.170248 66.839062 60.021204 3.299373 std 267.580450 271.201856 253.938223 32.408353 min 0.000000 0.000000 0.000000 0.000000 25% 0.000000 0.000000 0.000000 0.000000 50% 0.000000 0.000000 0.000000 0.000000 75% 0.000000 0.000000 0.000000 0.000000			1.000000	1.000000	1.000000	
count 99999.00000 99999.00000 99999.00000 99999.00000 mean 68.170248 66.839062 60.021204 3.299373 std 267.580450 271.201856 253.938223 32.408353 min 0.000000 0.000000 0.000000 0.000000 25% 0.000000 0.000000 0.000000 0.000000 50% 0.000000 0.000000 0.000000 0.000000 75% 0.000000 0.000000 0.000000 0.000000						
nean 68.170248 66.839062 60.021204 3.299373 std 267.580450 271.201856 253.938223 32.408353 nin 0.000000 0.000000 0.000000 0.000000 25% 0.000000 0.000000 0.000000 0.000000 50% 0.000000 0.000000 0.000000 0.000000 75% 0.000000 0.000000 0.000000 0.000000	count 0					
std 267.580450 271.201856 253.938223 32.408353 nin 0.000000 0.000000 0.000000 0.000000 25% 0.000000 0.000000 0.000000 0.000000 50% 0.000000 0.000000 0.000000 0.000000 75% 0.000000 0.000000 0.000000 0.000000	mean					
25% 0.000000 0.000000 0.000000 0.000000 50% 0.000000 0.000000 0.000000 0.000000 75% 0.000000 0.000000 0.000000 0.000000	std	267.580450	271.201856	253.938223	32.408353	
50% 0.000000 0.000000 0.000000 0.000000 75% 0.000000 0.000000 0.000000 0.000000	min					
75% 0.000000 0.000000 0.000000 0.000000	25% 50%					
max 12916.220000 9165.600000 11166.210000 2618.570000	75%	0.000000	0.000000	0.000000	0.00000	
	max 1	2916.220000	9165.600000	11166.210000	2618.570000	

Handling missing values

Handling missing values in columns

```
# Cheking percent of missing values in columns
df_missing_columns =
(round(((df.isnull().sum()/len(df.index))*100),2).to_frame('null')).so
rt_values('null', ascending=False)
df_missing_columns
                            null
arpu 3g 6
                           74.85
night_pck_user_6
                           74.85
total_rech_data_6
                           74.85
arpu \overline{2}g 6
                           74.85
max_rech_data_6
                           74.85
```

```
fb user 6
                            74.85
av rech amt data 6
                            74.85
date of last rech data 6
                           74.85
count rech 2g 6
                            74.85
count rech 3g 6
                            74.85
date_of_last_rech_data_7
                           74.43
                            74.43
total rech data 7
                            74.43
fb user 7
max rech data 7
                            74.43
night pck user 7
                            74.43
                            74.43
count_rech_2g_7
av_rech_amt_data_7
                            74.43
arpu_2g_7
                            74.43
count_rech_3g_7
                            74.43
arpu_3g_7
                            74.43
total rech data 9
                            74.08
count_rech_3g_9
                            74.08
fb user 9
                            74.08
max rech data 9
                            74.08
                            74.08
arpu 3g 9
date of last rech data 9
                           74.08
night pck user 9
                            74.08
arpu 2g 9
                            74.08
count rech 2g 9
                            74.08
av rech amt data 9
                            74.08
. . .
circle id
                             0.00
                             0.00
total og mou 8
vol 3g mb 7
                             0.00
                             0.00
total_og_mou_7
total og mou 6
                             0.00
                             0.00
arpu 9
arpu 8
                             0.00
arpu 7
                             0.00
                             0.00
arpu 6
last date of month 6
                             0.00
total rech num 8
                             0.00
total rech num 9
                             0.00
total_rech_amt_6
                             0.00
total rech amt 7
                             0.00
vol_3g_mb_6
                             0.00
                             0.00
vol 2g mb 9
                             0.00
vol_2g_mb_8
vol 2g mb 7
                             0.00
vol_2g_mb_6
                             0.00
last_day_rch_amt_9
                             0.00
                             0.00
last_day_rch_amt_8
last day rch amt 7
                             0.00
last day rch amt 6
                             0.00
```

```
0.00
max rech amt 9
                           0.00
max rech amt 8
max rech amt 7
                           0.00
                           0.00
max rech amt 6
total rech amt 9
                           0.00
total_rech_amt_8
                           0.00
                           0.00
sep vbc 3q
[226 rows x 1 columns]
# List the columns having more than 30% missing values
col list missing 30 =
list(df missing columns.index[df missing columns['null'] > 30])
# Delete the columns having more than 30% missing values
df = df.drop(col list missing 30, axis=1)
df.shape
(99999, 186)
```

Deleting the date columns as the date columns are not required in our analysis

```
# List the date columns
date_cols = [k for k in df.columns.to_list() if 'date' in k]
print(date_cols)

['last_date_of_month_6', 'last_date_of_month_7',
'last_date_of_month_8', 'last_date_of_month_9', 'date_of_last_rech_6',
'date_of_last_rech_7', 'date_of_last_rech_8', 'date_of_last_rech_9']

# Dropping date columns
df = df.drop(date_cols, axis=1)
```

Dropping circle_id column as this column has only one unique value. Hence there will be no impact of this column on the data analysis.

```
# Drop circle_id column
df = df.drop('circle_id', axis=1)
df.shape
(99999, 177)
```

Filter high-value customers

Creating column avg_rech_amt_6_7 by summing up total recharge amount of month 6 and 7. Then taking the average of the sum.

```
df['avg_rech_amt_6_7'] = (df['total_rech_amt_6'] +
df['total_rech_amt_7'])/2
```

Finding the 70th percentile of the avg_rech_amt_6_7

```
X = df['avg_rech_amt_6_7'].quantile(0.7)
X
368.5
```

Filter the customers, who have recharged more than or equal to X.

<pre>df = df df.head</pre>		/g_rech_am	nt_6_7'] >	= X]		
		nber loc_	_og_t2o_mo	u std_og_t2c	_mou loc_i	_t2o_mou
arpu_6 7	7000701	1601	Θ.	0	0.0	0.0
1069.18 8	7001524	1846	0.	0	0.0	0.0
378.721 13	l 7002191	1713	0.	Θ	0.0	0.0
492.846 16	5 7000875	5565	Θ.	0	0.0	0.0
430.975			0.		0.0	0.0
690.008		77)	0.	O .	0.0	0.0
	arpu_7	arpu_8	arpu_9	onnet_mou_6	onnet_mou_7	onnet_mou_8
7 134	19.850	3171.480	500.000	57.84	54.68	52.29
8 49	92.223	137.362	166.787	413.69	351.03	35.08
13 26)5.671	593.260	322.732	501.76	108.39	534.24
16 29	99.869	187.894	206.490	50.51	74.01	70.61
17 1	18.980	25.499	257.583	1185.91	9.28	7.79
	net_mou_ _mou_9		:_mou_6 o	ffnet_mou_7	offnet_mou_8	3
7 NaN	Na	aN	453.43	567.16	325.91	
8	33.4	16	94.66	80.63	136.48	3
13 214.06	244.8	31	413.31	119.28	482.46	5
16	31.3	34	296.29	229.74	162.76	i .
224.39 17 87.89	558.5	51	61.64	0.00	5.54	

roam i	c mou 6 ro	oam ic mou 7	roam	ic mou 8	roam ic m	ou 9
roam_og_mou	ı_6		_			_
7 23.74	16.23	33.49		31.64		NaN
8	0.00	0.00		0.00		0.00
0.00 13	23.53	144.24		72.11	13	6.78
7.98 16	0.00	2 02		0.00		0 00
0.00	0.00	2.83		0.00		0.00
17	0.00	4.76		4.81		0.00
0.00						
	g_mou_7 ro 12.59	oam_og_mou_8	roam_	og_mou_9	loc_og_t2	t_mou_6 \ 51.39
7 8	0.00	38.06 0.00		NaN 0.00		297.13
13	35.26	1.44		12.78		49.63
16	17.74	0.00		0.00		42.61
17	8.46	13.34		17.98		38.99
loc og	+2+ mou 7	loc og t2t	mou 8	loc on t	2+ mau 0	
loc og t2m		100_09_121_	_iiiou_o	toc_og_t	2 t_1110u_9	
7	31.38		40.28		NaN	
308.63						
8	217.59		12.49		26.13	
80.96	6 10		26 01		6 14	
13 151.13	6.19		36.01		6.14	
16	65.16		67.38		26.88	
273.29						
17	0.00		0.00		36.41	
58.54						
loc_og_	_t2m_mou_7	loc_og_t2m_	mou_8	loc_og_t	2m_mou_9	
loc_og_t2f_ 7	_mou_6 \ 447.38	1	.62.28		NaN	
62.13	117130	-	.02120		IVGIV	
8	70.58		50.54		34.58	
0.00 13	47.28	7	294.46		108.24	
4.54	47.20	2	.94.40		100.24	
16	145.99	1	28.28		201.49	
0.00	0.00		0.00		0.20	
17 0.00	0.00		0.00		9.38	
3.00						
		loc_og_t2f_	_mou_8	loc_og_t	2f_mou_9	
loc_og_t2c_ 7	_mou_6 \ 55.14		53.23		NaN	
1	55.14		JJ. ZJ		ivaiv	

0.0 8	0.00	0.00	0.00	
0.0 13	0.00	23.51	5.29	
0.0		25.51		
16 0.0	4.48	10.26	4.66	
17	0.00	0.00	0.00	
0.0				
	_og_t2c_mou_7	loc_og_t2c_mou_8	loc_og_t2c_mou_9	loc_og_mou_6
7	0.0	0.00	NaN	422.16
8	0.0	7.15	0.0	378.09
13	0.0	0.49	0.0	205.31
16	0.0	0.00	0.0	315.91
17	0.0	0.00	0.0	97.54
				37.13 .
loc_ 7 8 13 16	_og_mou_7 loc_ 533.91 288.18 53.48	255.79 63.04	NaN 60.71 1	$4.\overline{30}$ 16.56
17	215.64 0.00	205.93 2	33.04	46.41 7.89 46.91
17	0.00	205.93 2	33.04 45.79 11	7.89
17 std_ std_og_1	0.00 _og_t2t_mou_7 t2m_mou_6 \	205.93 2 0.00 std_og_t2t_mou_8	33.04 45.79 11 std_og_t2t_mou_9	7.89
std_og_1 7 49.89	0.00 _og_t2t_mou_7	205.93 2	33.04 45.79 11	7.89
std_og_t std_og_t 7 49.89	0.00 _og_t2t_mou_7 t2m_mou_6 \	205.93 2 0.00 std_og_t2t_mou_8	33.04 45.79 11 std_og_t2t_mou_9	7.89
std_og_t 7 49.89 8 13.69	0.00 _og_t2t_mou_7 t2m_mou_6 \ 23.29	205.93 2 0.00 std_og_t2t_mou_8 12.01	33.04 45.79 11 std_og_t2t_mou_9 NaN	7.89
std_og_t std_og_t 7 49.89 8 13.69	0.00 _og_t2t_mou_7 t2m_mou_6 \	205.93 2 0.00 std_og_t2t_mou_8 12.01 22.58	33.04 45.79 11 std_og_t2t_mou_9 NaN 7.33	7.89
std_og_t 7 49.89 8 13.69 13 255.36 16 22.99	0.00 _og_t2t_mou_7 t2m_mou_6 \	205.93 2 0.00 2 std_og_t2t_mou_8 12.01 22.58 498.23 3.23	33.04 45.79 11 std_og_t2t_mou_9 NaN 7.33 230.38 4.46	7.89
std_og_f 7 49.89 8 13.69 13 255.36	0.00 _og_t2t_mou_7 t2m_mou_6 \	205.93 2 0.00 2 std_og_t2t_mou_8 12.01 22.58 498.23	33.04 45.79 11 std_og_t2t_mou_9 NaN 7.33 230.38	7.89
std_std_og_t 7 49.89 8 13.69 13 255.36 16 22.99 17 1.55	0.00 _og_t2t_mou_7 t2m_mou_6 \	205.93 2 0.00 2 std_og_t2t_mou_8 12.01 22.58 498.23 3.23 0.00	33.04 45.79 11 std_og_t2t_mou_9 NaN 7.33 230.38 4.46 504.11	7.89
std_std_og_t 7 49.89 8 13.69 13 255.36 16 22.99 17 1.55 std_og_t 7	0.00 _og_t2t_mou_7 t2m_mou_6 \	205.93 2 0.00 2 std_og_t2t_mou_8 12.01 22.58 498.23 3.23 0.00	33.04 45.79 11 std_og_t2t_mou_9 NaN 7.33 230.38 4.46 504.11	7.89
std_std_og_t 7 49.89 8 13.69 13 255.36 16 22.99 17 1.55	0.00 _og_t2t_mou_7 t2m_mou_6 \	205.93 2 0.00 2 std_og_t2t_mou_8 12.01 22.58 498.23 3.23 0.00 std_og_t2m_mou_8	33.04 45.79 11 std_og_t2t_mou_9 NaN 7.33 230.38 4.46 504.11 std_og_t2m_mou_9	7.89
std_og_f 7 49.89 8 13.69 13 255.36 16 22.99 17 1.55 std_og_f 7 6.66	0.00 _og_t2t_mou_7 t2m_mou_6 \	205.93 2 0.00 2 std_og_t2t_mou_8 12.01 22.58 498.23 3.23 0.00 std_og_t2m_mou_8 49.14	33.04 45.79 11 std_og_t2t_mou_9 NaN 7.33 230.38 4.46 504.11 std_og_t2m_mou_9 NaN	7.89

0.00						
16	64.51		18.29		13.79	
0.00 17	0.00		0.00		78.51	
0.00						
	og_t2f_mou_7 2c mou 6 \	std_og_t2	f_mou_8	std_o	g_t2f_mou_9	
7	20.08		16.68		NaN	
0.0 8	0.00		0.00		0.00	
0.0 13 0.0	0.00		0.00		0.00	
16	0.00		0.00		4.43	
0.0 17	0.00		0.00		0.00	
0.0						
	og_t2c_mou_7	std_og_t2	c_mou_8	std_o	g_t2c_mou_9	std_og_mou_6
7	0.0		0.0		NaN	60.86
8	0.0		0.0		0.0	130.26
13	0.0		0.0		0.0	701.78
16	0.0		0.0		0.0	30.89
17	0.0		0.0		0.0	1148.46
std_d isd og mo		d_og_mou_8	std_og_	mou_9	isd_og_mou_	6
7	75.14	77.84		NaN	0.	9
0.18 8 0.00	143.48	98.28		81.46	0.	9
13	138.93	655.18	3	26.39	0.0	9
0.00 16	67.09	21.53		22.69	0.0	9
0.00 17	0.81	0.00	Б	02 62	0.	0
0.00	0.01	0.00	J	82.63	0.	U
		d_og_mou_9	spl_og_	mou_6	spl_og_mou_	7
spl_og_mo	10.01	NaN		4.50	0.0	9
6.50 8	0.00	0.0		0.00	0.0	9
J	3.00	0.0		0.00	010	

10.23					
13 4.78	1.29	0.0	0.00	0.00	
16	0.00	0.0	0.00	3.26	
5.91 17	0.00	0.0	2.58	0.00	
0.00					
spl_o og others		others_6 og_oth	ers_7 og_othe	ers_8	
7	NaN	0.00	0.0	0.0	NaN
8	0.00	0.00	0.0	0.0	0.0
13	0.00	0.00	0.0	0.0	0.0
16	0.00	0.00	0.0	0.0	0.0
17	2.64	0.93	0.0	0.0	0.0
7	_og_mou_6 to 487.53	otal_og_mou_7 to 609.24	otal_og_mou_8 350.16	total_o	g_mou_9 \ 0.00
8 13	508.36 907.09	431.66 192.41	171.56 1015.26		142.18 446.09
16	346.81	286.01	233.38		255.74
17	1249.53	0.81	0.00		631.08
	c_t2t_mou_6 t mou 9 \	loc_ic_t2t_mou_	7 loc_ic_t2t_	_mou_8	
7 – –	58.14	32.2	5	27.31	
NaN 8	23.84	9.8	4	0.31	
4.03 13	67.88	7.5	3	52.58	
24.98 16	41.33	71.4		28.89	
50.23					
17 40.91	34.54	0.0	9	0.00	
loc i	c t2m mou 6	loc ic t2m mou	7 loc ic +2m	mou 8	
loc_ic_t2	m_mou_9 \				
7 NaN	217.56	221.4	9 .	121.19	
8 17.34	57.58	13.9	3	15.48	
13	142.88	18.5	3 :	195.18	
104.79 16	226.81	149.6	9	150.16	
				-	

172.86 17 43.86	47.41	2.31	0.00	
loc_i		loc_ic_t2f_mou_7	loc_ic_t2f_mou_8	
7 – –	?f_mou_9 \ 152.16	101.46	39.53	
NaN 8 0.00	0.00	0.00	0.00	
13 8.51	4.81	0.00	7.49	
16 65.21	8.71	8.68	32.71	
17 0.71	0.00	0.00	0.00	
		_ic_mou_7 loc_ic_	_mou_8 loc_ic_mou_9	
7 36.89	2t_mou_6 \ 427.88	355.23 1	.88.04 NaN	
8	81.43	23.83	15.79 21.38	
13 115.68	215.58	26.11 2	255.26 138.29	
16 68.79	276.86	229.83 2	211.78 288.31	
17 8.63	81.96	2.31	0.00 85.49	
	.c_t2t_mou_7 ?m mou 6 \	std_ic_t2t_mou_8	std_ic_t2t_mou_9	
7 91.44	11.83	30.39	NaN	
8 22.43	0.58	0.10	0.00	
13 308.13	38.29	154.58	62.39	
16 18.68	78.64	6.33	16.66	
17 1.28	0.00	0.00	0.00	
	.c_t2m_mou_7 ?f mou 6 \	std_ic_t2m_mou_8	std_ic_t2m_mou_9	
7 52.19	<u>1</u> 26.99	141.33	NaN	
8	4.08	0.65	13.53	
13	29.79	317.91	151.51	

0.00					
16 0.51	73.08	73.9	3	29.58	
17	0.00	0.0	0	1.63	
0.00					
	c_t2f_mou_7 o mou 6 \	std_ic_t2f_mou_	8 std_ic_t2	f_mou_9	
7 – –	34.24	22.2	1	NaN	
0.0 8	0.00	0.0	0	0.0	
0.0 13 0.0	0.00	1.9	1	0.0	
16	0.00	2.1	.8	0.0	
0.0 17	0.00	0.0	10	0.0	
0.0				2.2	
	c_t2o_mou_7	std_ic_t2o_mou_	8 std_ic_t2	o_mou_9	std_ic_mou_6
7	0.0	Θ.	0	NaN	180.54
8	0.0	0.	0	0.0	22.43
13	0.0	Θ.	0	0.0	423.81
16	0.0	Θ.	0	0.0	87.99
17	0.0	0.	0	0.0	9.91
std_i total ic		_ic_mou_8 std_i	.c_mou_9 tot	al_ic_mou	ı_6
7 – –	173.08	193.94	NaN	626.	46
558.04 8	4.66	0.75	13.53	103.	86
28.49 13	68.09	474.41	213.91	968.	61
172.58 16	151.73	82.44	46.24	364.	86
381.56				01	00
17 2.31	0.00	0.00	1.63	91.	00
		otal_ic_mou_9 s	pl_ic_mou_6	spl_ic_m	nou_7
spl_ic_mo 7	u_8 \ 428.74	0.00	0.21		0.0
0.0 8	16.54	34.91	0.00		0.0
5	10.54	57.51	0.00		310

0.0						
13	1144.53	631	. 86	0.	. 45	0.0
0.0	204 46	224	r.c	^	00	0 0
16 0.0	294.46	334	. 50	0.	. 00	0.0
17	0.00	87	. 13	O	. 00	0.0
0.0	0.00	07.	. 13	0.	. 00	0.0
		isd_ic_mou_6	isd_ic_m	nou_7	isd_ic_mou_8	
isd_ic_mo	_					
7	NaN	2.06	1	4.53	31.59	
NaN	0.0	0.00		0 00	0.00	
8 0.00	0.0	0.00		0.00	0.00	
13	0.0	245.28	6	52.11	393.39	
259.33	0.0	2 13 1 2 0		, _ , _ ,	333133	
16	0.0	0.00		0.00	0.23	
0.00						
17	0.0	0.00		0.00	0.00	
0.00						
ic ot	hors 6 i	s others 7 is	o thors	0 ic	athers 0	
total rec		c_others_7 io	_others_	0 IC_	_0111613_9	
7	15.74	15.19	15.1	4	NaN	
5		20.20				
8	0.00	0.00	0.0	0	0.00	
19						
13	83.48	16.24	21.4	14	20.31	
6	0 00	0.00	0 0	10	0.00	
16 10	0.00	0.00	0.0	טט	0.00	
17	0.00	0.00	0.0	00	0.00	
19	0.00	0100	010	, 0	0.00	
		_7 total_rech	n_num_8	total_	_rech_num_9	
total_rec	h_amt_6	\	_			
7		5	7		3	
1580 8		21	14		15	
437		21	74		15	
13		4	11		7	
507						
16		6	2		1	
570						
17		2	4		10	
816						
total	rech amt	7 total rech	n amt 8	total	rech amt 9	
max rech		_, :0:4:_1:661	ae	cocac_	0 0.11_diii 05	
7		90	3638		Θ	

16 0.00) (Ð. O	0		0	0
17 0.00) (9.0	0		0	0
monthly_2g_ sachet 2g 9 \	9 sachet_2	2g_6 sa	chet_2g_7	sache	et_2g_8	
7 – 3–	0	0	0		0	0
8	0	0	1		3	0
13	0	0	0		3	0
16	0	0	0		0	0
17	0	0	0		0	0
monthly 2a	6 man+hlv	2a 7 m	on+hlv 2a	0 mar	+hlv 2~ 0	
monthly_3g_ sachet_3g_6 \			onthly_3g_			
7 0	0	0		0	0	
8 0	0	0		0	0	
13 0	0	0		0	0	
16	0	0		0	Θ	
0 17	0	0		0	0	
0						
<pre>sachet_3g_7 jul_vbc_3g \</pre>	sachet_3o	g_8 sac	het_3g_9	aon	aug_vbc_3g	
7 0 19.38		0	0	802	57.74	
8 910.65		0	0	315	21.03	
13 6		0	0	2607	0.00	
0.00 16 0		0	0	511	0.00	
2.45 17 6		0	0	667	0.00	
0.00						
jun_vbc_3g 7 18.74	sep_vbc_3g 0.0		ech_amt_6_ 1185.			
8 122.16 13 0.00	0.0	9	519. 380.	0		
13 0.00	0.0	,	200.	U		

```
16 21.89 0.0 459.0
17 0.00 0.0 408.0
df.shape
(30011, 178)
```

We can see that we have around ~30K rows after filtering

Handling missing values in rows

```
# Count the rows having more than 50% missing values
df missing rows 50 = df[(df.isnull().sum(axis=1)) >
(len(df.columns)//2)]
df missing rows 50.shape
(114, 178)
# Deleting the rows having more than 50% missing values
df = df.drop(df missing rows 50.index)
df.shape
(29897, 178)
# Checking the missing values in columns again
df missing columns =
(round(((df.isnull().sum()/len(df.index))*100),2).to frame('null')).so
rt values('null', ascending=False)
df missing columns
                     null
loc ic mou 9
                     5.32
og others \overline{9}
                     5.32
loc og t2t mou 9
                     5.32
loc ic_t2t_mou_9
                     5.32
loc og t2m mou 9
                     5.32
loc og t2f mou 9
                     5.32
loc og t2c mou 9
                     5.32
std ic t2m mou 9
                     5.32
loc og mou 9
                     5.32
                     5.32
std og t2t mou 9
                     5.32
roam og mou 9
                     5.32
std_ic_t2o_mou_9
std og t2m mou 9
                     5.32
std og t2f mou 9
                     5.32
spl og mou 9
                     5.32
std_og_t2c_mou_9
                     5.32
                     5.32
std og mou 9
isd og mou 9
                     5.32
std_ic_t2t_mou_9
                     5.32
std ic mou 9
                     5.32
```

```
5.32
onnet mou 9
spl ic mou 9
                     5.32
ic others 9
                     5.32
isd ic mou 9
                     5.32
loc ic t2f mou 9
                     5.32
offnet mou 9
                     5.32
loc ic t2m mou 9
                     5.32
std ic t2f mou 9
                     5.32
roam ic mou 9
                     5.32
loc og t2t mou 8
                     2.76
total_ic_mou_8
                     0.00
std_og_t2o_mou
                     0.00
loc_ic_t2o_mou
                     0.00
arpu 6
                     0.00
                     0.00
arpu 7
arpu 8
                     0.00
arpu 9
                     0.00
total og mou 6
                     0.00
total_og_mou_7
                     0.00
total og mou 8
                     0.00
total og mou 9
                     0.00
loc og t2o mou
                     0.00
total ic mou 6
                     0.00
total ic mou 7
                     0.00
total ic mou 9
                     0.00
last_day_rch_amt_7
                     0.00
total rech num 6
                     0.00
total rech num 7
                     0.00
total_rech_num_8
                     0.00
total rech num 9
                     0.00
total_rech_amt_6
                     0.00
total rech amt 7
                     0.00
total rech amt 8
                     0.00
total rech amt 9
                     0.00
max rech amt 6
                     0.00
max rech amt 7
                     0.00
max rech amt 8
                     0.00
max rech amt 9
                     0.00
last day rch amt 6
                     0.00
avg_rech_amt_6_7
                     0.00
[178 rows \times 1 columns]
```

Looks like MOU for all the types of calls for the month of September (9) have missing values together for any particular record.

Lets check the records for the MOU for Sep(9), in which these coulmns have missing values together.

```
# Listing the columns of MOU Sep(9)
print(((df missing columns[df missing columns['null'] ==
5.32]).index).to list())
['loc ic mou 9', 'og others 9', 'loc og t2t mou 9',
'loc_ic_t2t_mou_9', 'loc_og_t2m_mou_9', 'loc_og_t2f_mou_9',
'loc_og_t2c_mou_9', 'std_ic_t2m_mou_9', 'loc_og_mou_9',
'std_og_t2t_mou_9', 'roam_og_mou_9', 'std_ic_t2o_mou_9',
'std_og_t2t_mou_9', 'std_og_t2f_mou_9', 'spl_og_mou_9',
'std_og_t2c_mou_9', 'std_og_mou_9', 'isd_og_mou_9',
'std_ic_t2t_mou_9', 'std_ic_mou_9', 'onnet_mou_9', 'spl_ic_mou_9',
'ic_others_9', 'isd_ic_mou_9', 'loc_ic_t2f_mou_9', 'offnet_mou_9',
'loc ic t2m mou 9', 'std ic t2f mou 9', 'roam ic mou 9']
# Creating a dataframe with the condition, in which MOU for Sep(9) are
null
df null mou 9 = df[(df['loc og t2m mou 9'].isnull()) &
(df['loc_ic_t2f_mou_9'].isnull()) & (df['roam_og_mou_9'].isnull()) &
(df['std ic t2m mou 9'].isnull()) &
  (df['loc og t2t mou 9'].isnull()) &
(df['std ic t2t mou 9'].isnull()) & (df['loc og t2f mou 9'].isnull())
& (df['loc ic mou 9'].isnull()) &
  (df['loc_og_t2c_mou_9'].isnull()) & (df['loc_og_mou_9'].isnull()) &
(df['std_og_t2t_mou_9'].isnull()) & (df['roam_ic_mou_9'].isnull()) &
  (df['loc ic t2m mou 9'].isnull()) &
(df['std og t2m mou 9'].isnull()) & (df['loc ic t2t mou 9'].isnull())
& (df['std og t2f mou 9'].isnull()) &
  (df['std og t2c mou 9'].isnull()) & (df['og others 9'].isnull()) &
(df['std\ og\ mou\ 9'].isnull())\ \&\ (df['spl\ og\ mou\ 9'].isnull())\ \&
  (df['std ic t2f mou 9'].isnull()) & (df['isd og mou 9'].isnull()) &
(df['std ic mou 9'].isnull()) & (df['offnet mou 9'].isnull()) &
  (df['isd ic mou 9'].isnull()) & (df['ic others 9'].isnull()) &
(df['std ic t2o mou 9'].isnull()) & (df['onnet mou 9'].isnull()) &
  (df['spl ic mou 9'].isnull())]
df null mou 9.head()
     mobile number loc og t2o mou std og t2o mou loc ic t2o mou
arpu_6
         7000701601
                                     0.0
                                                        0.0
                                                                           0.0
1069.180
97
         7000589828
                                     0.0
                                                        0.0
                                                                           0.0
374.863
                                                                           0.0
         7001300706
                                     0.0
                                                        0.0
111
596.301
143
         7000106299
                                     0.0
                                                        0.0
                                                                           0.0
695,609
188
         7000340381
                                     0.0
                                                        0.0
                                                                           0.0
734.641
```

7
97
111
143
188 183.668 0.000 0.0 0.0 4.38 0.98 NaN onnet_mou_9 offnet_mou_6 offnet_mou_7 offnet_mou_8 offnet_mou_9 \ 7 NaN
onnet_mou_9 offnet_mou_6 offnet_mou_7 offnet_mou_8 offnet_mou_9 \
offnet_mou_9 \ NaN 453.43 567.16 325.91 NaN 74.54 43.66 31.86 NaN 111 NaN 45.51 12.34 NaN NaN 143 NaN 13.91 1.89 NaN NaN 188 NaN 105.16 39.39 NaN NaN 16.23 33.49 31.64 NaN 97 0.00 0.00 6.16 NaN 111 0.00 0.00 NaN NaN 123 0.00 0.00 NaN NaN 124 0.00 0.00 NaN NaN 125 0.00 0.00 NaN NaN 128 0.00 0.00 NaN NaN 125 38.06 NaN 127 38.06 NaN 128 0.00 0.00 NaN 129 38.06 NaN 120 0.00 NaN <
NaN 97 NaN 74.54 43.66 31.86 NaN 111 NaN 45.51 12.34 NaN 143 NaN 13.91 1.89 NaN NaN 188 NaN 105.16 39.39 NaN NaN 16.23 33.49 31.64 NaN 97 0.00 0.00 6.16 NaN 111 0.00 0.00 NaN NaN 143 0.00 8.94 NaN NaN 188 0.00 0.00 NaN NaN 197 0.00 0.00 23.91 NaN 111 0.00 0.00 NaN NaN 123 NaN NaN
97
111 NaN 45.51 12.34 NaN 143 NaN 13.91 1.89 NaN NaN 188 NaN 105.16 39.39 NaN 188 NaN 105.16 39.39 NaN NaN 16.23 33.49 31.64 NaN 97 0.00 0.00 6.16 NaN 111 0.00 0.00 NaN NaN 143 0.00 8.94 NaN NaN 188 0.00 0.00 NaN NaN 97 0.00 0.00 23.91 NaN 97 0.00 0.00 23.91 NaN 111 0.00 0.00 NaN NaN 123 0.00 8.53 NaN NaN 143 0.00 8.53 NaN NaN 143 0.00 0.00 NaN NaN 143 0.00 0.00 NaN NaN 143 0.00 0.00 NaN NaN 143
143 NaN 13.91 1.89 NaN 188 NaN 105.16 39.39 NaN roam_ic_mou_6 roam_ic_mou_7 roam_ic_mou_8 roam_ic_mou_9 \ 7 16.23 33.49 31.64 NaN 97 0.00 0.00 6.16 NaN 111 0.00 0.00 NaN NaN 143 0.00 8.94 NaN NaN 188 0.00 0.00 NaN NaN 97 0.00 0.00 23.91 NaN 111 0.00 0.00 23.91 NaN 97 0.00 0.00 NaN NaN 111 0.00 0.00 NaN NaN 111 0.00 0.00 NaN NaN 12.59 38.06 NaN 111 0.00 0.00 NaN NaN 143 0.00 8.53 NaN NaN 188 0.00 0.00 NaN NaN 188 0.00
NaN 188 NaN 105.16 39.39 NaN roam_ic_mou_6 roam_ic_mou_7 roam_ic_mou_8 roam_ic_mou_9 \tag{7.000} 7 16.23 33.49 31.64 NaN 97 0.00 0.00 6.16 NaN 111 0.00 0.00 NaN NaN 143 0.00 8.94 NaN NaN 188 0.00 0.00 NaN NaN 97 23.74 12.59 38.06 NaN 97 0.00 0.00 23.91 NaN 111 0.00 0.00 NaN NaN 123.74 12.59 38.06 NaN 97 0.00 0.00 NaN NaN 111 0.00 0.00 NaN NaN 143 0.00 8.53 NaN NaN 188 0.00 0.00 NaN NaN 188 0.00 0.00 NaN NaN </td
NaN roam_ic_mou_6 roam_ic_mou_8 roam_ic_mou_9 \ 7 16.23 33.49 31.64 NaN 97 0.00 0.00 6.16 NaN 111 0.00 0.00 NaN NaN 143 0.00 8.94 NaN NaN 188 0.00 0.00 NaN NaN 97 0.00 0.00 23.91 NaN 97 0.00 0.00 NaN NaN 111 0.00 0.00 NaN NaN 111 0.00 0.00 NaN NaN 111 0.00 0.00 NaN NaN 188 0.00 8.53 NaN NaN 188 0.00 0.00 NaN NaN 10c_og_t2t_mou_9 \ 7 51.39 31.38 40.28 NaN 97 2.83 16.19 9.73 NaN
7
7 23.74 12.59 38.06 NaN 97 0.00 0.00 23.91 NaN 111 0.00 0.00 NaN NaN 143 0.00 8.53 NaN NaN 188 0.00 0.00 NaN NaN 10c_og_t2t_mou_6 loc_og_t2t_mou_7 loc_og_t2t_mou_8 loc_og_t2t_mou_9 \ 7 51.39 31.38 40.28 NaN 97 2.83 16.19 9.73 NaN
loc_og_t2t_mou_9 \ 7
97 2.83 16.19 9.73 NaN
NaN
111 55.19 3.26 NaN NaN
143 18.89 6.83 NaN

188 NaN	4.38	0.98	NaN	
		loc_og_t2m_mou_7	loc_og_t2m_mou_8	
7 _	og_t2m_mou_9 \ 308.63	447.38	162.28	
NaN 97	16.99	23.14	17.79	
NaN 111	43.83	12.34	NaN	
NaN 143	8.58	1.56	NaN	
NaN 188 NaN	99.81	38.98	NaN	
loc	loc_og_t2f_mou_6 og t2f mou 9 \	loc_og_t2f_mou_7	loc_og_t2f_mou_8	
7 NaN	62.13	55.14	53.23	
97 NaN	3.54	1.46	1.83	
111 NaN	0.00	0.00	NaN	
143	0.00	0.00	NaN	
NaN 188 NaN	5.34	0.41	NaN	
110.11	loc og t2c mou 6	loc og t2c mou 7	loc og t2c mou 8	
loc_ 7	og_t2c_mou_9 \ 0.00	0.0	0.0	
NaN 97	0.40	0.0	0.0	
NaN 111	0.00	0.0	NaN	
NaN 143	2.09	0.0	NaN	
NaN 188	0.00	0.0	NaN	
NaN	0.00	0.0	Naiv	
std	<pre>loc_og_mou_6 loc og t2t mou 6 \</pre>	_og_mou_7 loc_og_	_mou_8 loc_og_mou_9)
7 4.30	422.16	533.91 2	255.79 NaN	I
97 430.	23.38	40.81	29.36 NaN	J
111 0.00	99.03	15.61	NaN NaN	J

143	27.48	8.39	NaN N	aN
1307.01	100 54	40. 20	A1 A1 A1	
188	109.54	40.38	NaN N	aN
0.00				
c+d	og +2+ mou 7	std og t2t mou 8	std og t2t man 0	
	2m mou 6 \	3 tu_0g_tz t_iii0u_0	3 tu_0g_tzt_iii0u_9	
7	23.29	12.01	NaN	
49.89	23123	12.01	Nun	
97	399.46	191.31	NaN	
53.59				
111	0.00	NaN	NaN	
0.00				
143	13.58	NaN	NaN	
1.95				
188	0.00	NaN	NaN	
0.00				
		std_og_t2m_mou_8	std_og_t2m_mou_9	
	2f_mou_6 \	40 14	N - N	
7 6.66	31.76	49.14	NaN	
97	13.81	8.33	NaN	
0.00	13.01	0.55	IVAIV	
111	0.00	NaN	NaN	
1.30	0.00	Han	Nan	
143	0.00	NaN	NaN	
0.00				
188	0.00	NaN	NaN	
0.00				
		std_og_t2f_mou_8	std_og_t2f_mou_9	
	2c_mou_6 \	10.00	AI - AI	
7	20.08	16.68	NaN	
0.0 97	0.00	0.00	NaN	
0.0	0.00	0.00	INdiv	
111	0.00	NaN	NaN	
0.0	0.00	IVAIV	Nan	
143	0.00	NaN	NaN	
0.0	3.30		710.11	
188	0.00	NaN	NaN	
0.0				
_		std_og_t2c_mou_8	std_og_t2c_mou_9	
std_og_mc		2	a,	
7	0.0	0.0	NaN	
60.86	0.0	0.0	NI - NI	
97 484.36	0.0	0.0	NaN	
404.30				

111						
143			0.0	NaN	NaN	
1308.96 188			0 0	NaN	NaN	
Std_og_mou_7 Std_og_mou_8 Std_og_mou_9 isd_og_mou_6		. 96	0.0	Nan	Nan	
std_og_mou_7 std_og_mou_9 isd_og_mou_6 isd_og_mou_7 7 75.14 77.84 NaN 0.0 0.18 97 413.28 199.64 NaN 0.0 0.00 111 0.00 NaN NaN 0.0 111 0.00 NaN NaN 0.0 0.00 13.58 NaN NaN 0.0 0.00 188 0.00 NaN NaN 0.0 0.00 188 0.00 NaN NaN 0.0 0.00 188 0.00 NaN 0.00 0.0 0.00 188 0.00 NaN 0.00 0.00 0.00 10.01 NaN 4.50 0.00 0.00 0.00 NaN 2.54 11.81 2.01 111 NaN NaN 0.38 2.71 NaN 143 NaN NaN 0.00 0.00 NaN NaN <td></td> <td></td> <td>0.0</td> <td>NaN</td> <td>NaN</td> <td></td>			0.0	NaN	NaN	
isd_og_mou_7 7 75.14 77.84 NaN 0.0 0.18 97 413.28 199.64 NaN 0.0 0.00 111 0.00 NaN NaN 0.0 0.00 143 13.58 NaN NaN 0.0 188 0.00 NaN NaN 0.0 isd_og_mou_8 isd_og_mou_9 spl_og_mou_6 spl_og_mou_7 spl_og_mou_8 7 10.01 NaN 4.50 0.00 6.50 97 0.00 NaN 2.54 11.81 2.01 111 NaN NaN 0.38 2.71 NaN 143 NaN NaN 0.00 0.00 NaN NaN 0.00 0.00 0.00 NaN NaN 0.00 0.00 0.00 NaN 0.00 0.0 0.0 NaN 188 NaN 0.00 0.0 0.0 NaN 97 NaN	0.00					
isd_og_mou_7 7 75.14 77.84 NaN 0.0 0.18 97 413.28 199.64 NaN 0.0 0.00 111 0.00 NaN NaN 0.0 0.00 143 13.58 NaN NaN 0.0 188 0.00 NaN NaN 0.0 isd_og_mou_8 isd_og_mou_9 spl_og_mou_6 spl_og_mou_7 spl_og_mou_8 7 10.01 NaN 4.50 0.00 6.50 97 0.00 NaN 2.54 11.81 2.01 111 NaN NaN 0.38 2.71 NaN 143 NaN NaN 0.00 0.00 NaN NaN 0.00 0.00 0.00 NaN NaN 0.00 0.00 0.00 NaN 0.00 0.0 0.0 NaN 188 NaN 0.00 0.0 0.0 NaN 97 NaN		std og mou 7	std og mou 8	std og mou 9	isd og mou 6	ì
0.18 97	isd_c		5 ta_5 g5 a_5	5 ta_0g0a_5		
97		75.14	77.84	NaN	0.0)
0.00 111		412 20	100 64	NoN	0 0	
111		413.28	199.04	Nan	0.0	
0.00 143		0.00	NaN	NaN	0.0)
0.00 188						
188 0.00 NaN NaN 0.0 isd_og_mou_8 isd_og_mou_9 spl_og_mou_6 spl_og_mou_7 spl_og_mou_8 \ 0.00 0.00 0.00 6.50 0.00 NaN 2.54 11.81 2.01 111 NaN NaN 0.38 2.71 NaN 143 NaN NaN 3.38 0.00 NaN 188 NaN NaN 0.00 0.00 NaN 0.00 0.0 0.0 NaN 97 NaN 0.86 0.0 0.0 NaN 111 NaN 1.29 0.0 NaN NaN 111 NaN 1.20 0.0 NaN NaN 188 NaN 1.20 0.0 NaN NaN 188 NaN 0.00 0.0 NaN NaN		13.58	NaN	NaN	0.0	
0.00		0 00	NaN	MaN	0 6	
isd_og_mou_8 isd_og_mou_9 spl_og_mou_6 spl_og_mou_7 spl_og_mou_8 \ 7		0.00	IValv	IVAIN	0.0	
spl_og_mou_8 NaN 4.50 0.00 6.50 0.00 NaN 2.54 11.81 2.01 111 NaN NaN 0.38 2.71 NaN 143 NaN NaN 3.38 0.00 NaN 188 NaN NaN 0.00 0.00 NaN 0.00 0.0 0.0 NaN 97 NaN 0.86 0.0 0.0 NaN 111 NaN 1.29 0.0 NaN NaN 111 NaN 1.20 0.0 NaN NaN 143 NaN 1.20 0.0 NaN NaN 188 NaN 0.00 0.0 NaN NaN 188 NaN 0.00 NaN NaN 143 NaN 1.20 0.0 NaN NaN 188 NaN 0.00 NaN NaN NaN 188 NaN 0.00 NaN						
7	7		isd_og_mou_9	spl_og_mou_6	spl_og_mou_7	
6.50 97			NaN	4 50	0.06	
97		10.01	IVAIV	4.50	0.00	4
111 NaN NaN 0.38 2.71 NaN 143 NaN NaN 3.38 0.00 NaN 188 NaN NaN 0.00 0.00 0.00 NaN 0.00 0.0 0.0 0.0 NaN 97 NaN 0.86 0.0 0.0 NaN 111 NaN 1.29 0.0 NaN NaN 143 NaN 1.20 0.0 NaN NaN 188 NaN 0.00 0.0 NaN NaN 188 NaN 0.00 0.0 NaN NaN 7 487.53 609.24 350.16 0.0 97 511.16 465.91 231.03 0.0	97	0.00	NaN	2.54	11.81	
NaN 143 NaN NaN 3.38 0.00 NaN NaN 0.00 0.00 0.00 NaN 0.00 0.0 0.0 0.0 7 NaN 0.86 0.0 0.0 NaN 97 NaN 1.29 0.0 NaN NaN 111 NaN 1.29 0.0 NaN NaN 143 NaN 1.20 0.0 NaN NaN 188 NaN 0.00 0.0 NaN NaN 188 NaN 0.00 0.0 NaN NaN 7 487.53 609.24 350.16 0.0 97 511.16 465.91 231.03 0.0			N. N.	0.20	2 71	
143 NaN NaN 3.38 0.00 NaN NaN 0.00 0.00 0.00 NaN 0.00 0.0 0.0 0.0 NaN 97 NaN 0.86 0.0 0.0 NaN NaN 111 NaN 1.29 0.0 NaN NaN 143 NaN 1.20 0.0 NaN NaN 188 NaN 0.00 0.0 NaN NaN 188 NaN 0.00 0.0 NaN NaN 188 NaN 0.00 0.0 NaN NaN 189 100 0.0 0.0 0.0 0.0 0.0 0.0 189 100 0.0		NaN	NaN	0.38	2./1	
NaN 188 NaN NaN NaN 0.00 0.00 spl_og_mou_9 og_others_6 og_others_7 og_others_8 og_others_9 7 NaN 0.00 0.0 0.0 NaN 97 NaN 0.86 0.0 0.0 NaN 111 NaN 1.29 0.0 NaN NaN 143 NaN 1.20 0.0 NaN NaN 188 NaN 0.00 0.0 NaN NaN total_og_mou_6 total_og_mou_7 total_og_mou_8 total_og_mou_9 \ 487.53 609.24 350.16 0.0 97 511.16 465.91 231.03 0.0		NaN	NaN	3.38	0.00)
NaN spl_og_mou_9 og_others_6 og_others_7 og_others_8 og_others_9 7 NaN 0.00 0.0 0.0 NaN 97 NaN 0.86 0.0 0.0 NaN 111 NaN 1.29 0.0 NaN NaN 143 NaN 1.20 0.0 NaN NaN 188 NaN 0.00 0.0 NaN NaN 188 NaN 0.00 0.0 NaN NaN 7 487.53 609.24 350.16 0.0 97 511.16 465.91 231.03 0.0	NaN	-				
spl_og_mou_9 og_others_6 og_others_7 og_others_8 og_others_9 7 NaN 0.00 0.0 0.0 NaN 97 NaN 0.86 0.0 0.0 NaN 111 NaN 1.29 0.0 NaN NaN 143 NaN 1.20 0.0 NaN NaN 188 NaN 0.00 0.0 NaN NaN 188 NaN 0.00 0.0 NaN NaN 189 0.00 0.00 0.00 0.00 0.00 180 0.00 0.00 0.00 0.00 0.00 0.00 180 0.00		NaN	NaN	0.00	0.00)
NaN 0.00 0.0 0.0 NaN 97 NaN 0.86 0.0 0.0 NaN 111 NaN 1.29 0.0 NaN NaN 143 NaN 1.20 0.0 NaN NaN 188 NaN 0.00 0.0 NaN NaN 487.53 609.24 350.16 0.0 97 511.16 465.91 231.03 0.0	NaN					
NaN 0.00 0.0 0.0 NaN 97 NaN 0.86 0.0 0.0 NaN 111 NaN 1.29 0.0 NaN NaN 143 NaN 1.20 0.0 NaN NaN 188 NaN 0.00 0.0 NaN NaN 487.53 609.24 350.16 0.0 97 511.16 465.91 231.03 0.0		spl og mou 9	og others 6	og others 7	og others 8 c	g others 9
97 NaN 0.86 0.0 0.0 NaN 111 NaN 1.29 0.0 NaN NaN 143 NaN 1.20 0.0 NaN NaN 188 NaN 0.00 0.0 NaN NaN total_og_mou_6 total_og_mou_7 total_og_mou_8 total_og_mou_9 \ 7 487.53 609.24 350.16 0.0 97 511.16 465.91 231.03 0.0	\		3_1 1 2		_	3_ _ -
111 NaN 1.29 0.0 NaN NaN 143 NaN 1.20 0.0 NaN NaN 188 NaN 0.00 0.0 NaN NaN total_og_mou_6 total_og_mou_7 total_og_mou_8 total_og_mou_9 \ 487.53 609.24 350.16 0.0 97 511.16 465.91 231.03 0.0	7	NaN	0.00	0.0	0.0	NaN
111 NaN 1.29 0.0 NaN NaN 143 NaN 1.20 0.0 NaN NaN 188 NaN 0.00 0.0 NaN NaN total_og_mou_6 total_og_mou_7 total_og_mou_8 total_og_mou_9 \ 487.53 609.24 350.16 0.0 97 511.16 465.91 231.03 0.0	97	NaN	0.86	0.0	0.0	NaN
143 NaN 1.20 0.0 NaN NaN 188 NaN 0.00 0.0 NaN NaN total_og_mou_6 total_og_mou_7 total_og_mou_8 total_og_mou_9 \ 7 487.53 609.24 350.16 0.0 97 511.16 465.91 231.03 0.0						
188 NaN 0.00 0.0 NaN NaN total_og_mou_6 total_og_mou_7 total_og_mou_8 total_og_mou_9 \ 487.53 609.24 350.16 0.0 97 511.16 465.91 231.03 0.0	111	NaN	1.29	0.0	NaN	NaN
total_og_mou_6 total_og_mou_7 total_og_mou_8 total_og_mou_9 \ 7	143	NaN	1.20	0.0	NaN	NaN
total_og_mou_6 total_og_mou_7 total_og_mou_8 total_og_mou_9 \ 7	188	NaN	0.00	0.0	NaN	NaN
7 $\overline{487.53}$ $\overline{609.24}$ $\overline{350.16}$ 0.0 97 511.16 465.91 231.03 0.0						
7 $\overline{487.53}$ $\overline{609.24}$ $\overline{350.16}$ 0.0 97 511.16 465.91 231.03 0.0		total on mou	6 total on mo	yı 7 total oq	mou 8 total	og mou 0 \
97 511.16 465.91 231.03 0.0	7					
111 102.01 18.33 0.00 0.0	97	511.1	.6 465	5.91		
	111	102.0	18	3.33	0.00	0.0

143 188	1341.03 109.54	21.98 40.38	0.00 0.00	0.0 0.0
	_ic_t2t_mou_9 \	loc_ic_t2t_mou_7		
7 NaN	58.14	32.26	27.31	
97	11.61	32.89	4.46	
NaN 111	50.01	16.66	NaN	
NaN 143	30.19	7.06	NaN	
NaN	50.19	7.00	ivaiv	
188 NaN	21.18	13.44	NaN	
	loc_ic_t2m_mou_6	loc_ic_t2m mou 7	loc_ic_t2m_mou_8	
loc_ 7	_ic_t2m_mou_9 \ 217.56	221.49	121.19	
, NaN	217.30	221.49	121.19	
97	16.94	26.94	26.63	
NaN 111	160.68	58.53	NaN	
NaN				
143 NaN	27.98	1.35	NaN	
188	217.03	56.63	NaN	
NaN				
		loc_ic_t2f_mou_7	loc_ic_t2f_mou_8	
10c_ 7	_ic_t2f_mou_9 \ 152.16	101.46	39.53	
NaN	2.22	0.60	0.00	
97 NaN	0.98	0.63	0.00	
111	5.06	0.40	NaN	
NaN				
143 NaN	10.13	0.00	NaN	
188	18.28	2.94	NaN	
NaN				
	_ic_t2t_mou_6 \		_mou_8 loc_ic_mou_	
7	427.88	355.23	L88.04 Na	V
36.8 97	29.54	60.48	31.09 Na	V
0.49				
111	215.76	75.59	NaN Na	N
0.00)			

143	68.31	8.41	NaN	NaN
25.56	252 42			
188	256.49	73.03	NaN	NaN
0.00				
c+d i	c +2+ may 7	c+d ic +2+ may 0	c+d ic +2+ man	0
		std_ic_t2t_mou_8	Std_IC_tZt_IIIOU	_9
7	n_mou_6 \	30.30	N	- M
-	11.83	30.39	IV	aN
91.44	1 26	1.06	N	- N
97	1.36	1.06	IV	aN
0.00	0.00	NI - NI		- NI
111	0.00	NaN	N	aN
0.00	0.00			
143	0.00	NaN	N	aN
0.00	0.00			
188	0.00	NaN	N	aN
0.00				
c+d ÷	c +2m may 7	ctd ic t2m may 0	ctd ic +2m may	0
Stu_1	.c_cziii_iiiou_/	std_ic_t2m_mou_8	Stu_IC_tZIII_IIIOU	_9
7 Sta_1c_t21	_mou_6 \ 126.99	141.33	NI NI	aN
	120.99	141.33	IV	dIV
52.19	1 16	0.00	N	n NI
97	4.16	0.00	IV	aN
0.00	0.00	N - N	M	- NI
111	0.00	NaN	IN	aN
1.13	0.00			
143	0.00	NaN	N	aN
0.00	0.00	NI - NI		- NI
188	0.00	NaN	N	aN
0.00				
c+d i	c +2f mou 7	std ic t2f mou 8	std ic t2f mou	0
_	mou 6 \	3 td_1c_t21_110d_0	3 td_1ttz1_iiiod	_9
7	34.24	22.21	N	aN
0.0	J4.24	22.21	IV	GIV.
97	0.00	0.00	N	aN
0.0	0.00	0.00	IV	GIV.
111	0.00	NaN	N	aN
0.0	0.00	INDIN	IV	alv
143	0.00	NaN	M	aN
0.0	0.00	INDIN	IV	an
188	0.00	NaN	N	aN
0.0	0.00	INDIN	IV	alv
0.0				
std i	c t2o mou 7	std ic t2o mou 8	std ic t2o mou	9
std ic mou		5 ca_10_c20_iii0a_0	5 tu_1t_ t20_1110u	
7	0.0	0.0	N	aN
•	0.0	0.0	IX	GIV
180 54				
180.54 97	A A	0.0	N	aN
180.54 97 0.49	0.0	0.0	N	aN

111	0.0		NaN	NaN	
1.13	010		Hait	nan	
143	0.0)	NaN	NaN	
25.56					
188	0.0)	NaN	NaN	
0.00					
		d_ic_mou_8	std_ic_mou_9	9 total_ic_mou	_6
	_/ \ 73.08	193.94	Naf	N 626.	46
558.04		1.00			0.4
97	5.53	1.06	Nal	N 32.	04
67.84 111	0.00	NaN	Nal	V 217.	0.4
75.59	0.00	Ivaiv	Ivai	V 217.	04
143	0.00	NaN	Nal	N 93.	88
8.41					
188	0.00	NaN	Nal	V 256.	49
73.03					
4.4.1 <i>:</i>	a ma 0	4-4-1 :- w-	01		7
rotat_1 spl ic mou 8		total_1c_mo	u_9 spt_1c_r	mou_6 spl_ic_m	ou_/
7	428.74		0.0	0.21	0.0
0.0	720177		0.0	0.21	0.0
97	32.16		0.0	0.63	0.0
0.0					
111	0.00		0.0	0.00	0.0
NaN					
143	0.00		0.0	0.00	0.0
NaN	0 00		0 0	0.00	0 0
188 NaN	0.00		0.0	0.00	0.0
IVAIV					
spl_ic_	mou_9 is	d_ic_mou_6	isd_ic_mou_7	7 isd_ic_mou_8	
isd_ic_mou_9	_				
7	NaN	2.06	14.53	31.59	
NaN	M - M	0.00	0.00	0.00	
97 NaN	NaN	0.00	0.00	0.00	
NaN 111	NaN	0.00	0.00	9 NaN	
NaN	IVAIV	0.00	0.00) INGIN	
143	NaN	0.00	0.00	9 NaN	
NaN		0.00	0.00	, itali	
188	NaN	0.00	0.00	9 NaN	
NaN					
				·	
ic_othe		_otners_/ 1	c_others_8 :	c_otners_9	
total_rech_n 7 1	um_6 \ 5.74	15.19	15.14	NaN	
5	J 1 7	13.19	13.14	IVAIV	
9					

97	1.36	1.83	0.00	NaN
14 111	0.15	0.00	NaN	NaN
L2	0.15	0.00	Nan	IVAIV
143	0.00	0.00	NaN	NaN
31	0.00	0.00	N = N	N - N
188 6	0.00	0.00	NaN	NaN
to	tal_rech_num_7 to	tal_rech_num	n_8 total_r	ech_num_9
total_r 7	rech_amt_6 \ 5		7	3
, 1580	J		,	3
97	17		14	3
432				
111	8		5	2
704	6		4	2
143 796	0		4	Z
188	1		0	0
864	_			-
4.	.t.1		. 0 +.+.1	
	otal_rech_amt_7 to ch amt 6 \	tal_recn_amt	_8 total_r	ecn_amt_9
7	790	36	538	0
1580	, • •	-		•
97	328	2	206	0
36	170		^	0
111 154	178		0	Θ
143	40		0	0
90	10		· ·	Ü
188	120		0	0
252				
	nx_rech_amt_7 max_	rech_amt_8	max_rech_am	t_9
	y_rch_amt_6 \		_	
7	790	1580		0
0 97	44	36		0
30	77	30		J
111	50	0		0
154				
143	30	0		0
10 188	120	0		Θ
252	120	U		U
	ist_day_rch_amt_7	last_day_rch	n_amt_8 las	t_day_rch_amt_9
νοι_2g_	_mb_6 \			

7		0	779		Θ
0.00 97		20	Θ		0
0.00		30	0		0
111 284.	50				
143		0	0		0
188		120	0		0
58.4	4				
vol		vol_2g_mb_8	vol_2g_mb_9	vol_3g_mb_6	
7	3g_mb_7 \ 0.0	0.0	0.0	0.0	0.0
97	0.0	0.0	0.0	0.0	0.0
111	0.0	0.0	0.0	0.0	0.0
143	0.0	0.0	0.0	0.0	0.0
188	0.0	0.0	0.0	1522.4	0.0
	vol 3g mb 8	vol 3g mb 9	monthly 2g 6	monthly_2g_7	
mont 7	hly_2g_8 \ 0.0	0.0	0	0	
0					
97 0	0.0	0.0	0	0	
111	0.0	0.0	1	0	
0 143	0.0	0.0	0	0	
0 188	0.0	0.0	0	Θ	
0			_	-	
	monthly_2g_9	sachet_2g_6	sachet_2g_7	sachet_2g_8	sachet_2g_9
7	Θ	0	0	0	0
97	0	0	0	0	0
111	0	0	0	0	0
143	0	0	0	0	0
188	Θ	0	0	Θ	Θ
	monthly 2a 6	monthly 3g	7 monthly 2a	Q monthly 2a	0
	monthity_by_b	monthity_3g_	monthity_3g_	_8 monthly_3g_	_9

```
sachet 3g 6 \
7
                 0
                                0
                                               0
                                                              0
0
97
                 0
                                0
                                                              0
0
                 0
                                0
                                                              0
111
1
143
                 0
                                0
                                               0
                                                              0
0
188
                 2
                                0
                                                              0
     sachet 3g 7 sachet 3g 8 sachet 3g 9
                                             aon
                                                    aug vbc 3g
jul vbc 3g
                              0
                                           0
                                               802
                                                         57.74
19.38
                0
                              0
                                           0
                                               502
                                                          0.00
97
0.00
111
                0
                              0
                                           0
                                               332
                                                          0.00
0.00
                0
                              0
                                                          0.00
143
                                           0
                                               264
0.00
                0
                              0
                                                          0.00
188
                                           0
                                             244
831.48
     jun vbc 3g
                  sep vbc 3g
                              avg rech amt 6 7
7
                         0.0
          18.74
                                         1185.0
97
           0.00
                         0.0
                                          380.0
111
           0.00
                         0.0
                                          441.0
143
           0.00
                         0.0
                                          418.0
188
        1223.04
                         0.0
                                          492.0
df null mou 9.shape
(1590, 178)
# Deleting the records for which MOU for Sep(9) are null
df = df.drop(df_null_mou 9.index)
# Again Cheking percent of missing values in columns
df missing columns =
(round(((df.isnull().sum()/len(df.index))*100),2).to frame('null')).so
rt_values('null', ascending=False)
df_missing_columns
                   null
isd og mou 8
                   0.55
roam ic mou 8
                   0.55
loc og mou 8
                   0.55
std ic t2o mou 8
                   0.55
roam_og_mou_8
                   0.55
```

```
loc_ic_t2f_mou_8
                   0.55
loc og t2t mou 8
                   0.55
std ic t2f mou 8
                   0.55
std og t2m mou 8
                   0.55
loc og t2m mou 8
                   0.55
std_og_t2t_mou_8
                   0.55
std ic t2m mou 8
                   0.55
loc og t2f mou 8
                   0.55
spl og mou 8
                   0.55
loc ic mou 8
                   0.55
loc_og_t2c_mou_8
                   0.55
std_ic_t2t_mou_8
                   0.55
loc_ic_t2m_mou_8
                   0.55
                   0.55
std og t2f mou 8
spl ic mou 8
                   0.55
std ic mou 8
                   0.55
                   0.55
offnet mou 8
ic others 8
                   0.55
og others 8
                   0.55
loc ic t2t mou 8
                   0.55
onnet mou 8
                   0.55
isd ic mou 8
                   0.55
std og t2c mou 8
                   0.55
std og mou 8
                   0.55
isd og mou 6
                   0.50
                    . . .
arpu 9
                   0.00
                   0.00
arpu 8
arpu 7
                   0.00
                   0.00
arpu 6
loc ic t2o mou
                   0.00
std_og_t2o_mou
                   0.00
std og mou 9
                   0.00
spl og mou 9
                   0.00
isd ic mou 9
                   0.00
og others 9
                   0.00
spl ic mou 9
                   0.00
total ic mou 9
                   0.00
total_ic_mou_8
                   0.00
total ic mou 7
                   0.00
total_ic_mou_6
                   0.00
std ic mou 9
                   0.00
std_ic_t2o_mou_9
                   0.00
std_ic_t2f_mou_9
                   0.00
std_ic_t2m_mou_9
                   0.00
std_ic_t2t_mou_9
                   0.00
                   0.00
loc ic mou 9
loc_ic_t2f_mou_9
                   0.00
loc og t2o mou
                   0.00
```

```
loc_ic_t2m_mou_9 0.00
loc_ic_t2t_mou_9 0.00
total_og_mou_9 0.00
total_og_mou_8 0.00
total_og_mou_7 0.00
total_og_mou_6 0.00
avg_rech_amt_6_7 0.00

[178 rows x 1 columns]
```

Looks like MOU for all the types of calls for the month of Aug (8) have missing values together for any particular record.

Lets check the records for the MOU for Aug(8), in which these coulmns have missing values together.

```
# Listing the columns of MOU Aug(8)
print(((df missing columns[df missing columns['null'] ==
0.55]).index).to list())
['isd_og_mou_8', 'roam_ic_mou_8', 'loc_og_mou_8', 'std_ic_t2o_mou_8',
'roam_og_mou_8', 'loc_ic_t2f_mou_8', 'loc_og_t2t_mou_8',
'std_ic_t2f_mou_8', 'std_og_t2m_mou_8', 'loc_og_t2m_mou_8',
'std_og_t2t_mou_8', 'std_ic_t2m_mou_8', 'loc_og_t2f_mou_8',
'spl_og_mou_8', 'loc_ic_mou_8', 'loc_og_t2c_mou_8',
'std ic t2t mou 8', 'loc ic t2m mou 8', 'std og t2f mou 8',
'spl_ic_mou_8', 'std_ic_mou_8', 'offnet_mou_8', 'ic_others_8',
'og_others_8', 'loc_ic_t2t_mou_8', 'onnet_mou_8', 'isd_ic_mou_8',
'std og t2c mou 8', 'std og mou 8']
# Creating a dataframe with the condition, in which MOU for Aug(8) are
null
df null mou 8 = df[(df['loc og t2m mou 8'].isnull()) &
(df['loc\ ic\ t2f\ mou\ 8'].isnull()) \& (df['roam\ og\ mou\ 8'].isnull()) &
(df['std ic t2m mou 8'].isnull()) &
  (df['loc og t2t mou 8'].isnull()) &
(df['std ic t2t mou 8'].isnull()) & (df['loc og t2f mou 8'].isnull())
& (df['loc ic mou 8'].isnull()) &
  (df['loc_og_t2c_mou_8'].isnull()) & (df['loc_og_mou_8'].isnull()) &
(df['std_og_t2t_mou_8'].isnull()) & (df['roam_ic_mou_8'].isnull()) &
  (df['loc_ic_t2m_mou_8'].isnull()) &
(df['std og t2m mou 8'].isnull()) & (df['loc ic t2t mou 8'].isnull())
& (df['std og t2f mou 8'].isnull()) &
  (df['std og t2c mou 8'].isnull()) & (df['og others 8'].isnull()) &
(df['std og mou 8'].isnull()) & (df['spl og mou 8'].isnull()) &
  (df['std ic t2f mou 8'].isnull()) & (df['isd og mou 8'].isnull()) &
(df['std ic mou 8'].isnull()) & (df['offnet mou 8'].isnull()) &
  (df['isd ic mou 8'].isnull()) & (df['ic others 8'].isnull()) &
(df['std ic t2o mou 8'].isnull()) & (df['onnet mou 8'].isnull()) &
  (df['spl ic mou 8'].isnull())]
```

df_nu	ll_mou_8.h	ead()					
2 KDII		mber l	.oc_og_t2o_	_mou std_og	_t2o_mou	loc_ic_t2	o_mou
arpu_ 375 580.4	700225	2754		0.0	0.0		0.0
578	700024	8548		0.0	0.0		0.0
569.6 788 532.7	700063	6808		0.0	0.0		0.0
1802 810.4	700051	6213		0.0	0.0		0.0
4837 649.1	700219	2662		0.0	0.0		0.0
\	arpu_7	arpu_8	arpu_9	onnet_mou_6	onnet_m	ou_7 onne	t_mou_8
375	111.878	0.0	378.881	249.43	3	9.64	NaN
578	237.289	0.0	4.440	718.01	21	2.73	NaN
788	546.756	0.0	269.274	1173.39	89	1.83	NaN
1802	0.000	0.0	0.000	91.33		NaN	NaN
4837	149.572	0.0	0.250	1354.24	8	5.13	NaN
	onnet mou	9 off	net mou 6	offnet mou	7 offne	t mou 8	
offne 375	t_mou_ $\overline{9}$ \\\\\\\\		62.24	37.2		 NaN	
144.5 578	3 0.	00	487.06	139.	71	NaN	
1.26 788	149.	34	61.59	137.	14	NaN	
428.3 1802	6 0.	00	1371.04	Na	aN	NaN	
0.00 4837	0.		50.63	37.		NaN	
0.00							
375 578 788 1802 4837		ou_6 r 5.49 0.00 0.00 1.21 0.00	1.		_mou_8 r NaN NaN NaN NaN NaN	0. 0.	00 43 00
	roam_og_m	ou_6 r	oam_og_mou	ı_7 roam_og	_mou_8 r	oam_og_mou	_9 \

375 578 788 1802 4837	312.59 0.00 0.00 11.23 0.00	78.58 6.30 14.43 NaN 44.78	NaN NaN NaN NaN NaN	0.00 1.26 0.00 3.91 0.43
loc_og loc_og_t2t_m		loc_og_t2t_mou_7	loc_og_t2t_mou_8	
375	0.00	0.00	NaN	
11.54 578 0.00	11.28	27.89	NaN	
788	31.06	27.49	NaN	
7.39				
1802	17.86	NaN	NaN	
0.00 4837	6.71	1.35	NaN	
0.00	0.71	1.35	ivaiv	
loc_og loc og t2m m		loc_og_t2m_mou_7	loc_og_t2m_mou_8	
375 25.31	0.00	0.00	NaN	
578	42.24	46.94	NaN	
0.00 788	24 66	60.06	NaN	
34.23	34.66	60.86	INdiv	
1802	84.51	NaN	NaN	
0.00 4837	15.18	15.76	NaN	
0.00	13110	13170	Nan	
loc_og loc og t2f m		loc_og_t2f_mou_7	loc_og_t2f_mou_8	
375	0.0	0.0	NaN	
0.0	0.0	0.0	N = N	
578 0.0	0.0	0.0	NaN	
788	0.0	0.0	NaN	
0.0				
1802 0.0	0.0	NaN	NaN	
4837 0.0	0.0	0.0	NaN	
		loc_og_t2c_mou_7	loc_og_t2c_mou_8	
loc_og_t2c_m 375 0.41	ou_9 \ 0.00	0.0	NaN	
578	2.33	0.0	NaN	

0.00				
788	0.00	0.0	NaN	
5.58				
1802	10.29	NaN	NaN	
0.00				
4837	0.00	0.0	NaN	
0.00	0.00	0.0	Ivalv	
0.00				
	loc og mou 6 loc	a. mau 7 1.aa.	mou 9 loc og mou	0 \
275			_mou_8 loc_og_mou_	
375	0.00	0.00	NaN 36.8	
578	53.53	74.84	NaN 0.0	
788	65.73	88.36	NaN 41.6	
1802	102.38	NaN	NaN 0.0	
4837	21.89	17.11	NaN 0.0	00
		std_og_t2t_mou_7	std_og_t2t_mou_8	
std_o	g_t2t_mou_9 \			
375	0.00	0.00	NaN	
233.5	1			
578	706.73	178.53	NaN	
0.00				
788	1142.33	854.08	NaN	
141.9		031100	nan-	
1802	73.46	NaN	NaN	
0.00	75.40	IVAIV	Ivaiv	
4837	1247 52	48.48	NaN	
	1347.53	40.40	Ivaiv	
0.00				
	std og t2m mou 6	s+d og +2m mou 7	std og tom mou 0	
c+d o		std_og_t2m_mou_7	stu_og_tzIII_IIIou_o	
	g_t2m_mou_9 \	0.00	NI - NI	
375	0.00	0.00	NaN	
118.79		00.76		
578	442.48	92.76	NaN	
0.00				
788	26.93	67.24	NaN	
388.5	4			
1802	1207.86	NaN	NaN	
0.00				
4837	35.44	11.88	NaN	
0 00				
0.00				
0.00				
0.00	std og t2f mou 6	std og t2f mou 7	std oa t2f mou 8	
	std_og_t2f_mou_6 a t2f mou 9 \	std_og_t2f_mou_7	std_og_t2f_mou_8	
std_o	g_t2f_mou_9 \			
std_o		std_og_t2f_mou_7	std_og_t2f_mou_8 NaN	
std_og 375 0.0	g_t2f_mou_9 \	0.0	NaN	
std_og 375 0.0 578	g_t2f_mou_9 \			
std_og 375 0.0 578 0.0	g_t2f_mou_9 \	0.0	NaN	
std_og 375 0.0 578 0.0 788	g_t2f_mou_9 \	0.0	NaN	
std_og 375 0.0 578 0.0 788 0.0	g_t2f_mou_9 \	0.0 0.0 0.0	NaN NaN NaN	
std_og 375 0.0 578 0.0 788	g_t2f_mou_9 \	0.0	NaN	

0.0				
4837		0.0	0.0	NaN
0.0				
		6 13		
مامله	std_og_t2c_mc		c_mou_7 std_c	og_t2c_mou_8
	g_t2c_mou_9 \		0 0	NaN
375 0.0		0.0	0.0	NaN
578		0.0	0.0	NaN
0.0		0.0	0.0	IVAIV
788		0.0	0.0	NaN
0.0		0.0	0.0	11011
1802		0.0	NaN	NaN
0.0				
4837		0.0	0.0	NaN
0.0				
م امما	std_og_mou_6	std_og_mou_/	std_og_mou_8	std_og_mou_9
15a_0 375	g_mou_6 \ 0.00	0.00	NaN	352.31
0.0	0.00	0.00	Ivaiv	332.31
578	1149.21	271.29	NaN	0.00
9.0	1143.21	2/11/23	Nun	0.00
788	1169.26	921.33	NaN	530.49
0.0				2221.12
L802	1281.33	NaN	NaN	0.00
9.0				
4837	1382.98	60.36	NaN	0.00
0.0				
	icd on mou 7	icd og mau 0	icd og mou O	col og mou 6
snl o		isd_og_mou_8	150_09_11100_9	Spt_og_iiiou_o
375	9_11104_/ (NaN	0.0	0.00
0.00	0.0	Nan	0.0	0.00
578	0.0	NaN	0.0	2.58
1.21				30
788	0.0	NaN	0.0	0.00
4.85				
1802	NaN	NaN	0.0	91.94
NaN				
4837	0.0	NaN	0.0	0.00
0.00				
	spl og mou 8	spl og mou 9	og others 6	og others 7
na nti	hers 8 \	3pt_og_mou_9	og_others_o	og_others_/
375	NaN	4.78	0.00	0.0
NaN	Hall	1170	3100	010
578	NaN	0.00	1.55	0.0
NaN				
788	NaN	5.58	0.00	0.0

NaN					
1802	NaN	0.00	1.53	NaN	
NaN					
4837	NaN	0.00	0.00	0.0	
NaN					
			_		
275			og_mou_7	total_og_ı	
375	0.0	0.00	0.00		0.0
578	0.0	1206.88	347.36		0.0
788 1802	0.0 0.0	1234.99 1477.19	1014.54 0.00		0.0 0.0
4837	0.0	1404.88	77.48		0.0
4037	0.0	1404.00	//.40		0.0
	total og mou 9 lo	c ic t2t mou 6 l	oc ic t2t	mou 7	
loc i	t2t mou 8 \				
375	<u>3</u> 93.96	0.00		0.00	
NaN					
578	0.00	48.01		63.39	
NaN					
788	577.71	54.19		52.64	
NaN	0.00	17.60		N. N.	
1802	0.00	17.68		NaN	
NaN	0.00	104 46		2 15	
4837 NaN	0.00	104.46		3.15	
IVAIV					
	loc_ic_t2t_mou_9	loc_ic_t2m_mou_6	loc_ic_t	2m_mou_7	
loc_i	c_t2m_mou_8 \				
375	6.74	0.00		0.00	
NaN					
578	0.00	83.09		64.31	
NaN	12 51	F4 C0		107.00	
788	12.51	54.69		187.96	
NaN 1802	0.00	39.46		NaN	
NaN	0.00	39.40		Ivaiv	
4837	0.00	162.01		17.94	
NaN	0100	102101		1,131	
		loc_ic_t2f_mou_6	loc_ic_t	2f_mou_7	
_	c_t2f_mou_8 \				
375	38.53	0.00		0.00	
NaN	2 22	0.00		0.00	
578	0.00	0.00		0.00	
NaN 788	81.83	1.16		2.01	
NaN	01.03	1.10		2.01	
1802	0.00	0.70		NaN	
NaN	0100	0.70		Nan	
4837	0.00	0.00		0.00	
	0.00	3.00			

NaN								
375 578 788	loc_ic_t2f_m	ou_9 0.0 0.0 0.0	loc_ic_mou_ 0.0 131.1 110.0	0 1	_ic_mou_7 0.00 127.71 242.63	loc_ic_	mou_8 NaN NaN NaN	\
1802 4837		0.0	57.8 266.4		NaN 21.09		NaN NaN	
std i	loc_ic_mou_9 c t2t mou 8		_ic_t2t_mou_	6 std ₋	_ic_t2t_m	ou_7		
375 NaN	45.28		0.0	0	(9.00		
578 NaN	0.00		24.9	8	46	6.43		
788	94.34		14.5	5	ļ	5.48		
NaN 1802	0.00		1.8	8		NaN		
NaN 4837	0.00		35.1	1	3:	1.96		
NaN		0		6				
std i	std_ic_t2t_m c t2m mou 8	ou_9 \	std_ic_t2m_	mou_6	std_ic_t	2m_mou_/		
375 NaN		8.31		0.00		0.00		
578		0.00		1.63		16.69		
NaN 788	2	5.61		11.49		62.19		
NaN 1802		0.00		11.98		NaN		
NaN 4837		0.00		48.48		0.00		
NaN								
std i	std_ic_t2m_m c_t2f_mou_8	ou_9 \	std_ic_t2f_	mou_6	std_ic_t2	2f_mou_7		
375 NaN		7.31		0.00		0.0		
578		0.00		0.00		0.0		
NaN 788	1	3.93		0.00		0.0		
NaN 1802		0.00		0.00		NaN		
NaN 4837		0.00		0.28		0.0		
NaN								
std_i	std_ic_t2f_m c_t2o_mou_8		std_ic_t2o_	mou_6	std_ic_t2	2o_mou_7		

375	0.0	0.0		0.0
NaN	0.0	0 0	•	0.0
578 NaN	0.0	0.0		0.0
788	0.0	0.0		0.0
NaN				
1802	0.0	0.0		NaN
NaN 4837	0.0	0.0		0.0
NaN	010	0.0		010
	- L. L. L. D		4 4	ald 'a man 0)
375	std_ic_t2o_mou_9 std 0.0	l_ic_mou_6 st 0.00	d_ic_mou_7 0.00	std_ic_mou_8 \ NaN
578	0.0	26.61	63.13	NaN
788	0.0	26.04	67.68	NaN
1802	0.0	13.86	NaN	NaN
4837	0.0	83.88	31.96	NaN
	std_ic_mou_9 total_i	.c_mou_6 tota	l_ic_mou 7	total_ic_mou_8 \
375	35.63	$\overline{}$ 0. $\overline{0}$ 0	$-\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ $	0.0
578	0.00	157.73	190.84	0.0
788 1802	39.54 0.00	140.74 71.71	310.31 0.00	0.0 0.0
4837	0.00	350.36	53.06	0.0
ال المما		.c_mou_6 spl_	_ic_mou_7 sp	l_ic_mou_8
375	c_mou_9 \ 80.91	0.00	0.0	NaN
0.00	00.91	0.00	0.0	IVAIN
578	0.00	0.00	0.0	NaN
0.00	704.74	0.70	0.0	
788 0.25	134.14	0.73	0.0	NaN
1802	0.00	0.00	NaN	NaN
0.00	0.00	0.00		
4837	0.00	0.00	0.0	NaN
0.00				
	isd ic mou 6 isd ic	mou 7 isd ic	mou 8 isd	ic mou 9
	hers_6 \			
375	0.0	0.0	NaN	0.0
0.00 578	0.0	0.0	NaN	0.0
0.00	0.0	0.0	Ivalv	0.0
788	0.0	0.0	NaN	0.0
3.89				
1802	0.0	NaN	NaN	0.0
0.00 4837	0.0	0.0	NaN	0.0
0.00	0.0	0.0	IVAIV	0.0

	ic others 7 ic ot	hers 8 ic others	s 9 total rech nu	m 6 \
375			0.0	$\overline{1}$ 7
578	0.0	NaN 0	0.0	19
788	0.0	NaN 6	0.0	10
1802	NaN	NaN 6	0.0	21
4837	0.0	NaN 6	0.0	11
total	<pre>total_rech_num_7 rech amt 6 \</pre>	total_rech_num_8	total_rech_num_9	
375	6	3	11	
700	10	0		
578	10	0	4	
717	7	1	-	
788 714	7	4	5	
1802	3	Θ	0	
955	3	U	U	
4837	6	3	4	
666	0	3	4	
000				
	total_rech_amt_7	total rech amt 8	total rech amt 9	
max r	ech amt 6 \	cocac_reen_ame_o	cocac_reen_ame_s	
375	130	0	440	
80	130			
578	220	0	0	
110				
788	494	0	336	
128				
1802	Θ	0	0	
110				
4837	176	0	0	
110				
1	max_rech_amt_7 max	x_rech_amt_8 max	c_rech_amt_9	
	day_rch_amt_6 \	0	Γ0	
375	50	0	50	
30	E0	0	0	
578	50	0	0	
27 788	120	0	130	
128	128	U	130	
1802	0	0	0	
30	U	U	U	
4837	110	0	0	
20	110	U	U	
20				
	last_day_rch_amt_7	last dav rch am	nt 8 last dav rch	amt 9
vol 2	g_mb_6 \			
375	9_ 0		0	30

0.0 578 0.0 0.0 788 0 0.0 1802 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0						
0.0 788 0.0 1802 0.0 4837 0.0 0.0 vol_2g_mb_7 vol_2g_mb_8 vol_2g_mb_9 vol_3g_mb_6 vol_3g_mb_7 375 0.0 0.0 800 0.0 0.0 0.0 vol_2g_mb_7 vol_2g_mb_8 vol_2g_mb_9 vol_3g_mb_6 vol_3g_mb_7 375 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.			20	0		0
788			30	U		U
0.0 1802 0.0 4837 0.0 vol_2g_mb_7 vol_2g_mb_8 vol_2g_mb_9 vol_3g_mb_6 vol_3g_mb_7 \ \begin{array}{cccccccccccccccccccccccccccccccccccc			Θ	A		130
1802			O .	O .		150
4837			0	0		0
0.0 vol_2g_mb_7	0.0					
vol_2g_mb_7 vol_2g_mb_8 vol_2g_mb_9 vol_3g_mb_6 vol_3g_mb_7 375 0.0 0.0 0.0 0.0 0.0 578 0.0 0.0 0.0 0.0 0.0 788 0.0 0.0 0.0 0.0 0.0 1802 0.0 0.0 0.0 0.0 0.0 4837 0.0 0.0 0.0 0.0 0.0 vol_3g_mb_8 vol_3g_mb_9 monthly_2g_6 monthly_2g_7 monthly_2g_7 monthly_2g_8 0.0 0.0 0.0 0.0 0 0.0 0.0 0.0 0.0 788 0.0 0.0 0.0 0.0 1802 0.0 0.0 0.0 0.0 4837 0.0 0.0 0.0 0.0 578 0 0 0 0.0 578 0 0 0 0 0 578 0 0 0 0 <td></td> <td></td> <td>0</td> <td>0</td> <td></td> <td>0</td>			0	0		0
375	0.0					
375		vol 2a mb 7	vol 2a mh 8	vol 2a mh 0	vol 3a mh 6 v	/ol 3a mh 7
375 0.0 0.0 0.0 0.0 0.0 0.0 578 0.0 0.0 0.0 0.0 0.0 0.0 788 0.0 0.0 0.0 0.0 0.0 0.0 1802 0.0 0.0 0.0 0.0 0.0 0.0 4837 0.0 0.0 0.0 0.0 0.0 0.0 758 0.0 <td< td=""><td>\</td><td>VOC_29_IIID_7</td><td>VO C_29_IIID_0</td><td>VO C_29_IIID_5</td><td>VOC_59_IIIb_0</td><td>/oc_5g_iiib_/</td></td<>	\	VOC_29_IIID_7	VO C_29_IIID_0	VO C_29_IIID_5	VOC_59_IIIb_0	/oc_5g_iiib_/
578 0.0 0.0 0.0 0.0 0.0 788 0.0 0.0 0.0 0.0 0.0 1802 0.0 0.0 0.0 0.0 0.0 4837 0.0 0.0 0.0 0.0 0.0 monthly_2g_8 0.0 0.0 0 0 578 0.0 0.0 0 0 788 0.0 0.0 0 0 8837 0.0 0.0 0 0 8375 0 0 0 0 8837 0.0 0.0 0 0 8837 0 0 0 0 9 0 0 0 0 578 0 0 0 0 0 578 0 0 0 0 0 578 0 0 0 0 0 578 0 0 0 0 0 578 0 0 0 0 0 <t< td=""><td></td><td>0.0</td><td>0.0</td><td>0.0</td><td>0.0</td><td>0.0</td></t<>		0.0	0.0	0.0	0.0	0.0
788						
1802	578	0.0	0.0	0.0	0.0	0.0
1802	700	0.0	0.0	0.0	0.0	0.0
4837 0.0 0.0 0.0 0.0 0.0 vol_3g_mb_8 vol_3g_mb_9 monthly_2g_6 monthly_2g_7 monthly_2g_8 0.0 0.0 0 0 578 0.0 0.0 0 0 0 788 0.0 0.0 0 0 0 1802 0.0 0.0 0 0 0 4837 0.0 0.0 0 0 0 578 0 0 0 0 0 375 0 0 0 0 0 578 0 0 0 0 0 788 0 0 0 0 0 788 0 0 0 0 0 1802 0 0 0 0 0 4837 0 0 0 0 0	700	0.0	0.0	0.0	0.0	0.0
vol_3g_mb_8 vol_3g_mb_9 monthly_2g_6 monthly_2g_7 375 0.0 0.0 0 0 578 0.0 0.0 0 0 788 0.0 0.0 0 0 1802 0.0 0.0 0 0 4837 0.0 0.0 0 0 monthly_2g_9 sachet_2g_6 sachet_2g_7 sachet_2g_8 sachet_2g_9 375 0 0 0 0 0 578 0 0 0 0 0 788 0 0 0 0 0 1802 0 0 0 0 0 4837 0 0 0 0 0	1802	0.0	0.0	0.0	0.0	0.0
vol_3g_mb_8 vol_3g_mb_9 monthly_2g_6 monthly_2g_7 375 0.0 0.0 0 0 578 0.0 0.0 0 0 788 0.0 0.0 0 0 1802 0.0 0.0 0 0 4837 0.0 0.0 0 0 monthly_2g_9 sachet_2g_6 sachet_2g_7 sachet_2g_8 sachet_2g_9 375 0 0 0 0 0 578 0 0 0 0 0 788 0 0 0 0 0 1802 0 0 0 0 0 4837 0 0 0 0 0						
monthly_2g_8 0.0 0.0 0 0 0 578 0.0 0.0 0 0 0 788 0.0 0.0 0 0 0 1802 0.0 0.0 0 0 0 4837 0.0 0.0 0 0 0 monthly_2g_9 sachet_2g_6 sachet_2g_7 sachet_2g_8 sachet_2g_9 375 0 0 0 0 0 578 0 0 0 0 0 788 0 0 0 0 0 1802 0 0 0 0 0 4837 0 0 0 0 0	4837	0.0	0.0	0.0	0.0	0.0
monthly_2g_8 0.0 0.0 0 0 0 578 0.0 0.0 0 0 0 788 0.0 0.0 0 0 0 1802 0.0 0.0 0 0 0 4837 0.0 0.0 0 0 0 monthly_2g_9 sachet_2g_6 sachet_2g_7 sachet_2g_8 sachet_2g_9 375 0 0 0 0 0 578 0 0 0 0 0 788 0 0 0 0 0 1802 0 0 0 0 0 4837 0 0 0 0 0						
monthly_2g_8 0.0 0.0 0 0 0 578 0.0 0.0 0 0 0 788 0.0 0.0 0 0 0 1802 0.0 0.0 0 0 0 4837 0.0 0.0 0 0 0 monthly_2g_9 sachet_2g_6 sachet_2g_7 sachet_2g_8 sachet_2g_9 375 0 0 0 0 0 578 0 0 0 0 0 788 0 0 0 0 0 1802 0 0 0 0 0 4837 0 0 0 0 0		vol 3a mb 8	vol 3a mb 9	monthly 2a 6	monthly 2a 7	
0 578	month1		_ 3	7_ 3_	7_ 3_	
578 0.0 0.0 0 0 788 0.0 0.0 0 0 1802 0.0 0.0 0 0 4837 0.0 0.0 0 0 monthly_2g_9 sachet_2g_6 sachet_2g_7 sachet_2g_8 sachet_2g_9 375 0 0 0 0 0 578 0 0 0 0 0 788 0 0 0 0 0 1802 0 0 0 0 0 4837 0 0 0 0 0		0.0	0.0	0	0	
0 788		0.0	0.0		•	
788		0.0	0.0	0	Θ	
0		0.0	0.0	0	۵	
1802 0.0 0.0 0 0 4837 0.0 0.0 0 0 monthly_2g_9 sachet_2g_6 sachet_2g_7 sachet_2g_8 sachet_2g_9 375 0 0 0 0 0 578 0 0 0 0 0 788 0 0 0 0 0 1802 0 0 0 0 0 4837 0 0 0 0 0		0.0	0.0	U	U	
0		0.0	0.0	0	0	
0 monthly_2g_9 sachet_2g_6 sachet_2g_7 sachet_2g_8 sachet_2g_9 375 0 0 0 0 0 578 0 0 0 0 0 788 0 0 0 0 0 1802 0 0 0 0 0 4837 0 0 0 0 0						
monthly_2g_9 sachet_2g_6 sachet_2g_7 sachet_2g_8 sachet_2g_9 375 0 0 0 0 0 578 0 0 0 0 0 0 788 0 0 0 0 0 0 1802 0 0 0 0 0 0 4837 0 0 0 0 0 0		0.0	0.0	0	0	
375 0 0 0 0 0 0 0 0 578 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0					
375 0 0 0 0 0 0 0 0 578 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		monthly 2a 0	sachet 2m 6	sachet 2m 7	sachet 2m Q	sachet 2m 0
375 0 0 0 0 0 578 0 0 0 0 0 788 0 0 0 0 0 1802 0 0 0 0 0 4837 0 0 0 0 0	\	morrerrey_2g_9	sachet_2g_0	Sachet_2g_7	3achet_2g_0	sachet_2g_9
578 0 0 0 0 0 788 0 0 0 0 0 1802 0 0 0 0 0 4837 0 0 0 0 0		0	0	0	0	Θ
788 0 0 0 0 0 1802 0 0 0 0 0 4837 0 0 0 0 0 0						
1802 0 0 0 0 0 4837 0 0 0 0 0	578	0	0	0	0	0
1802 0 0 0 0 0 4837 0 0 0 0 0	788	A	Θ	Θ	A	Q
4837 0 0 0 0 0	700	U	U	0	U	O
	1802	0	0	0	0	0
			_	_		
monthly 3g 6 monthly 3g 7 monthly 3g 8 monthly 3g 9	4837	Θ	0	0	Θ	0
monthly 3g 6 monthly 3g 7 monthly 3g 8 monthly 3g 9						
		monthly 3g 6	monthly 3g 7	7 monthly 3a	8 monthly 3a	9
sachet_3g_6 \	sachet		,	,,	_ , ,	

```
375
                  0
                                 0
                                               0
                                                              0
0
578
                                 0
                                               0
                                                              0
0
                                 0
788
                                               0
                                                              0
1802
                                 0
                                               0
                                                              0
0
4837
                                 0
                                               0
                                                              0
      sachet 3g 7 sachet 3g 8 sachet 3g 9
                                                aon aug vbc 3g
jul vbc 3g
375
                 0
                              0
                                                1102
                                                             0.0
0.0
578
                 0
                              0
                                            0
                                                274
                                                             0.0
0.0
                 0
                              0
                                                             0.0
788
                                                936
0.0
1802
                 0
                              0
                                            0
                                                755
                                                             0.0
0.0
4837
                 0
                              0
                                                             0.0
                                                520
0.0
      jun vbc 3g sep vbc 3g avg rech amt 6 7
375
             0.0
                          0.0
                                           415.0
                          0.0
578
              0.0
                                           468.5
788
                          0.0
                                           604.0
              0.0
1802
              0.0
                          0.0
                                           477.5
4837
             0.0
                          0.0
                                           421.0
# Deleting the records for which MOU for Aug(8) are null
df = df.drop(df_null_mou_8.index)
# Again cheking percent of missing values in columns
df missing columns =
(round(((df.isnull().sum()/len(df.index))*100),2).to frame('null')).so
rt_values('null', ascending=False)
df missing columns
                   null
roam ic mou 6
                   0.44
spl og mou 6
                   0.44
og others 6
                   0.44
loc ic t2t mou 6
                   0.44
loc_og_t2m_mou_6
                   0.44
loc og t2c mou 6
                   0.44
loc_ic_t2m_mou_6
                   0.44
isd_og_mou_6
                   0.44
loc og t2t mou 6
                   0.44
```

```
std og t2m mou 6
                   0.44
loc ic t2f mou 6
                   0.44
ic others 6
                   0.44
roam og mou 6
                   0.44
loc ic mou 6
                   0.44
std_og_mou_6
                   0.44
loc og t2f mou 6
                   0.44
isd ic mou 6
                   0.44
std ic t2t mou 6
                   0.44
std ic mou 6
                   0.44
std_og_t2t_mou_6
                   0.44
std_ic_t2o_mou_6
                   0.44
std_og_t2f_mou_6
                   0.44
                   0.44
std_ic_t2f_mou_6
spl_ic_mou_6
                   0.44
onnet mou 6
                   0.44
                   0.44
std og t2c mou 6
std ic t2m mou 6
                   0.44
offnet mou 6
                   0.44
loc og mou 6
                   0.44
std og t2f mou 7
                   0.16
                    . . .
loc ic t2m mou 8
                   0.00
std ic t2o mou 8
                   0.00
std ic t2f mou 9
                   0.00
std ic t2f mou 8
                   0.00
std_ic_t2m_mou_9
                   0.00
std ic t2m mou 8
                   0.00
std ic t2t mou 9
                   0.00
std_ic_t2t_mou 8
                   0.00
loc ic mou 9
                   0.00
loc_ic_mou_8
                   0.00
loc ic t2f mou 9
                   0.00
loc ic t2f mou 8
                   0.00
loc og t2o mou
                   0.00
loc ic t2m mou 9
                   0.00
loc ic t2t mou 9
                   0.00
std og t2c mou 9
                   0.00
loc_ic_t2t_mou_8
                   0.00
total og mou 9
                   0.00
total_og_mou_8
                   0.00
                   0.00
total og mou 7
                   0.00
total_og_mou_6
og others 9
                   0.00
og_others_8
                   0.00
spl_og_mou_9
                   0.00
                   0.00
spl og mou 8
isd og mou 9
                   0.00
isd og mou 8
                   0.00
```

```
std_og_mou_9     0.00
std_og_mou_8     0.00
avg_rech_amt_6_7     0.00
[178 rows x 1 columns]
```

Looks like MOU for all the types of calls for the month of Jun (6) have missing values together for any particular record.

Lets check the records for the MOU for Jun(6), in which these coulmns have missing values together.

```
# Listing the columns of MOU Jun(6)
print(((df missing columns[df missing columns['null'] ==
0.44]).index).to list())
['roam ic mou 6', 'spl og mou 6', 'og others 6', 'loc ic t2t mou 6',
'loc_og_t2m_mou_6', 'loc_og_t2c_mou_6', 'loc_ic_t2m_mou_6',
'isd_og_mou_6', 'loc_og_t2t_mou_6', 'std_og_t2m_mou_6',
'loc ic t2f mou 6', 'ic others_6', 'roam_og_mou_6', 'loc_ic_mou_6',
'std_og_mou_6', 'loc_og_t2f_mou_6', 'isd_ic_mou_6',
'std_og_mou_6', 'std_ic_mou_6', 'isd_ic_mou_6',
'std_ic_t2t_mou_6', 'std_ic_mou_6', 'std_og_t2t_mou_6',
'std_ic_t2o_mou_6', 'std_og_t2f_mou_6', 'std_ic_t2f_mou_6',
'spl_ic_mou_6', 'onnet_mou_6', 'std_og_t2c_mou_6', 'std_ic_t2m_mou_6',
'offnet_mou_6', 'loc_og_mou_6']
# Creating a dataframe with the condition, in which MOU for Jun(6) are
null
df null mou 6 = df[(df['loc og t2m mou 6'].isnull()) &
(df['loc ic t2f mou 6'].isnull()) & (df['roam og mou 6'].isnull()) &
(df['std ic t2m mou 6'].isnull()) &
  (df['loc og t2t mou 6'].isnull()) &
(df['std ic t2t mou 6'].isnull()) & (df['loc og t2f mou 6'].isnull())
& (df['loc ic mou 6'].isnull()) &
  (df['loc_og_t2c_mou_6'].isnull()) & (df['loc og mou 6'].isnull()) &
(df['std og t2t mou 6'].isnull()) & (df['roam ic mou 6'].isnull()) &
  (df['loc ic t2m mou 6'].isnull()) &
(df['std og t2m mou 6'].isnull()) & (df['loc ic t2t mou 6'].isnull())
& (df['std_og_t2f_mou_6'].isnull()) &
  (df['std og t2c mou 6'].isnull()) & (df['og others 6'].isnull()) &
(df['std_og_mou_6'].isnull()) & (df['spl_og_mou_6'].isnull()) &
  (df['std ic t2f mou 6'].isnull()) & (df['isd og mou 6'].isnull()) &
(df['std ic mou 6'].isnull()) & (df['offnet mou 6'].isnull()) &
  (df['isd ic mou 6'].isnull()) & (df['ic others 6'].isnull()) &
(df['std ic t2o mou 6'].isnull()) & (df['onnet mou 6'].isnull()) &
  (df['spl ic mou 6'].isnull())]
df null mou 6.head()
```

arpu_6	mobile_n	umber lo	c_og_t2o_mou	u std_og_t2	2o_mou loc_i	c_t2o_mou
77 30.000	70013	28263	0.0)	0.0	0.0
364 0.000	70021	68045	0.0)	0.0	0.0
423	70006	35248	0.0)	0.0	0.0
213.80 934	70021	52278	0.0)	0.0	0.0
48.000 1187	70004	86275	0.0)	0.0	0.0
0.000	arpu 7	arpu 8	arpu 9 or	nnet mou 6	onnet mou 7	onnet mou 8
\ 77	82.378	674.950	158.710	NaN	34.23	149.69
364	792.112	989.368	923.040	NaN	433.49	198.96
423	304.194	149.710	329.643	NaN	0.00	0.00
934	764.152	500.030	194.400	NaN	14.24	17.48
1187	757.170	995.719	0.000	NaN	1366.71	2268.91
	annot ma	u O offn	o+ mou 6 o+	ffnot mou 7	offnot mou	0
		_		ffnet_mou_7		•
77 57.68		.31	NaN	39.44	179.1	
364 828.29	571	.99	NaN	845.11	923.5	8
423 0.34	0	.00	NaN	10.03	1.4	5
934 43.64	7	.69	NaN	16.99	76.8	6
1187 0.00	0	.00	NaN	7.78	36.1	3
77 364 423 934 1187	roam_ic_	mou_6 ro NaN NaN NaN NaN NaN	am_ic_mou_7 0.0 0.0 0.0 0.0 0.0) () 8	ou_8 roam_ic 0.00 0.00 0.00 0.81 3.81	_mou_9 \ 0.0 0.0 0.0 0.0 0.0 0.0
77 364 423 934	roam_og_	mou_6 ro NaN NaN NaN NaN	am_og_mou_7 0.0 0.0 0.0 0.0	(i	ou_8 roam_og).00).00).00).00 L.56	_mou_9 \ 0.00 0.00 0.00 0.00

1187	NaN	0.0	25.23	0.21
loc og	t2t mou 6	loc og t2t mou 7	loc og t2t mou 8	
loc_og_t2t_m			140.60	
77 6.31	NaN	34.23	149.69	
364	NaN	28.78	7.46	
64.73	NI - NI	0.00	0.00	
423 0.00	NaN	0.00	0.00	
934	NaN	0.08	17.48	
7.69		4 76	46.10	
1187 0.00	NaN	4.76	46.18	
0.00				
	_t2m_mou_6	loc_og_t2m_mou_7	loc_og_t2m_mou_8	
loc_og_t2m_m 77	NaN	32.18	101.63	
29.41		32.120	101.05	
364	NaN	78.78	584.76	
490.71 423	NaN	0.00	0.58	
0.33				
934	NaN	16.99	63.23	
39.99 1187	NaN	7.78	31.29	
0.00	11011	7.70	31.23	
		loc_og_t2f_mou_7	loc_og_t2f_mou_8	
loc_og_t2f_m 77		0.91	20.06	
28.26	NaN	0.91	29.86	
364	NaN	21.58	9.43	
0.00 423	NaN	0.00	0.00	
0.00	NaN	0.00	0.00	
934	NaN	0.00	12.08	
3.65 1187	NaN	0.00	0.00	
0.00	IVAIN	0.00	0.00	
100.00	t2c mou 6	loc og +2c may 7	loc og t2c mou 8	
loc og t2c m		toc_og_tzc_iiiou_/	toc_og_tzc_iiiou_o	
77	NaN	0.0	3.9	
0.00 364	NaN	0.0	0.0	
2.78	IValV	0.0	0.0	
423	NaN	0.0	0.0	
0.00	N - N	0.0	0.0	
934	NaN	0.0	0.0	

0.00		0.0	0.0	
1187 0.00	NaN	0.0	0.0	
loc_og_m	ou_6 loc NaN	_og_mou_7 loc_og_ 67.33 2	mou_8 loc_og_mou_ 81.19 63.9	
364	NaN		01.66 555.4	
423	NaN	0.00	0.58 0.3	33
934	NaN	17.08	92.79 51.3	34
1187	NaN	12.54	77.48 0.0	90
std_og_t std og t2t mou		std_og_t2t_mou_7	std_og_t2t_mou_8	
77	_ `NaN	0.00	0.00	
0.00		404 71	101 40	
364	NaN	404.71	191.49	
507.26	NI - NI	0 00	0.00	
423 0.00	NaN	0.00	0.00	
934	NaN	14.16	0.00	
0.00	Ivaiv	14.10	0.00	
1187	NaN	1361.94	2202.03	
0.00	Nan	1301134	2202103	
0.00				
		std_og_t2m_mou_7	std_og_t2m_mou_8	
std_og_t2m_mou		0.00	0.00	
77	NaN	0.00	0.00	
0.00	NI - NI	722 01	221 41	
364 302.91	NaN	722.01	321.41	
423	NaN	0.00	0.25	
0.00	Nan	0.00	0.123	
934	NaN	0.00	0.00	
0.00				
1187	NaN	0.00	1.13	
0.00				
	2f_mou_6	std_og_t2f_mou_7	std_og_t2f_mou_8	
std_og_t2f_mou				
77	NaN	6.35	40.09	
0.0	NI - NI	0.00	0.00	
364	NaN	0.00	0.00	
0.0	NaN	0.00	0.61	
423 0.0	NaN	0.00	0.01	
934	NaN	0.00	0.00	
0.0	IVAIV	0.00	0.00	
1187	NaN	0.00	0.00	
0.0		0.00	0.00	

	_og_t2c_mou_ c mou 9 \	_6 std_og_t2	c_mou_7 std_o	g_t2c_mou_8
77		aN	0.0	0.0
9.0 364	Na	aN	0.0	0.0
9.0	IVC	aiv	0.0	0.0
123	Na	aN	0.0	0.0
9.0 934	Na	aN	0.0	0.0
0.0	IVC	21 4	0.0	0.0
187	Na	aN	0.0	0.0
0.0				
std_ sd og mo		std_og_mou_7	std_og_mou_8	std_og_mou_9
7	_ NaN	6.35	40.09	0.00
laN 64	NaN	1126.73	512.91	810.18
laN	IVAIV	1120.75	312.31	010.10
23	NaN	0.00	0.86	0.00
IaN 934	NaN	14.16	0.00	0.00
aN	IVAIV	14.10	0.00	0.00
187	NaN	1361.94	2203.16	0.00
laN				
		isd_og_mou_8	isd_og_mou_9	spl_og_mou_6
spl_og_mo: 7	u_7 \ 2.93	28.04	3.25	NaN
.00				
64	0.00	0.00	0.00	NaN
5.14 23	10.03	0.00	0.01	NaN
. 00				
34	20.13	8.41	0.00	NaN
.00 .187	0.00	0.00	0.00	NaN
.34	0.00	2.00	3.30	
spl	_og_mou_8 s	spl_og_mou_9	og_others_6	og_others_7
g_others		0.00	NI - NI	0.0
7 .0	7.58	0.00	NaN	0.0
64	13.84	37.74	NaN	0.0
. 0	0.00	0.00	NI - NI	0.0
23 .0	0.00	0.00	NaN	0.0
34	0.00	0.00	NaN	0.0
.0	1 70			2 2
.187	1.78	0.00	NaN	0.0

0.0			
77 364 423 934 1187	og_others_9 total_o 0.0 0.0 0.0 0.0 0.0		.61 356.93 .03 1128.43 .03 1.45 .38 101.21
		ic_t2t_mou_6 loc_ic_	_t2t_mou_7
loc_io 77	c_t2t_mou_8 \ 67.24	NaN	79.46
191.24 364	4 1403.38	NaN	7.41
10.23			
423 0.00	0.34	NaN	0.00
934	51.34	NaN	0.39
20.09 1187 56.38	0.00	NaN	19.34
	loc ic t2t mou 9 lo	c ic t2m mou 6 loc :	ic t2m mou 7
loc_io	c_t2m_mou_8 \ 5.26		43.31
94.18			
364 93.48	17.46	NaN	69.39
423	0.00	NaN	0.00
0.00 934	12.19	NaN	4.53
51.16 1187	0.00	NaN	28.19
16.31	0.00	Nuiv	20113
100 4		c_ic_t2f_mou_6 loc_:	ic_t2f_mou_7
77 _	c_t2f_mou_8 \ 16.39	NaN	2.03
0.00 364	44.89	NaN	0.00
0.83			
423 0.00	0.00	NaN	0.00
934 17.08	59.83	NaN	7.80
1187 0.00	0.00	NaN	0.00
77	loc_ic_t2f_mou_9 lo 15.78		ou_7 loc_ic_mou_8 \ 4.81 285.43

364	0.00	NaN	76.81	104.54
423	0.00	NaN	0.00	0.00
934	5.13	NaN	12.73	88.34
1187	0.00	NaN	47.54	72.69
loc_ic_n	nou_9 std	_ic_t2t_mou_6 std	_ic_t2t_mou_7	
std_ic_t2t_mou	ı_8 \			
77 3	37.44	NaN	8.00	
0.00				
	52.36	NaN	5.81	
10.09				
423	0.00	NaN	0.00	
0.00				
	77.16	NaN	0.00	
0.00	0.00		105 44	
1187	0.00	NaN	125.44	
149.81				
عاد المعاد	- O+ O	a+d +a +2m ma 6	a+d ia +7m ma 7	,
		std_ic_t2m_mou_6	Std_1C_t2m_mou_/	
std_ic_t2m_mou		NaN	0.00	
77 0.00	0.00	INdiv	0.00	
364	22.36	NaN	37.94	r
86.63	22.30	ivaiv	37.94	
423	0.00	NaN	0.00	
0.00	0.00	INGIN	0.00	
934	0.00	NaN	0.00	
0.00	0.00	Hait	0100	
1187	0.00	NaN	9.84	I
17.06				
std ic t	2m mou 9	std ic t2f mou 6	std ic t2f mou 7	
std_ic_t2f_mou	ı_8_ \			
77	0.00	NaN	0.0	
0.00				
364	34.49	NaN	0.0	
0.00				
423	0.00	NaN	0.0	
0.36				
934	0.00	NaN	0.0	
0.00	0.00		0.0	
1187	0.00	NaN	0.0	
0.00				
c+d +- +	-2.f ma 0	c+d ic +20 may 6	c+d ic +2c ma:: 7	
std_ic_t2o_mou	2f_mou_9	std_ic_t2o_mou_6	std_ic_t2o_mou_7	
77	15.93	NaN	0.0	
0.0	13.93	INCIN	0.0	
364	0.00	NaN	0.0	
0.0	0.00	INCIN	0.0	
010				

423	0.00	N	NaN	0.0
0.0				
934	0.00	N	NaN	0.0
0.0				
1187	0.00	N	NaN	0.0
0.0				
			std_ic_mou_7	std_ic_mou_8 \
77	0.0	NaN	8.00	0.00
364	0.0	NaN	43.76	96.73
423	0.0	NaN	0.00	0.36
934	0.0	NaN	0.00	0.00
1187	0.0	NaN	135.29	166.88
				1
				total_ic_mou_8 \
77	15.93	0.0	135.38	289.33
364	56.86	0.0	185.14	219.59
423	0.00	0.0	8.31	0.36
934	0.00	0.0	14.69	100.94
1187	0.00	0.0	182.84	239.58
				1
, .		ic_mou_6 sp	ol_ic_mou_7 s	pl_1c_mou_8
	c_mou_9 \		0.0	0.0
77	53.38	NaN	0.0	0.0
0.0	120 10	N. N.	0.0	0.0
364	129.19	NaN	0.0	0.0
0.0	0.00	NI - NI	0.0	0.0
423	0.00	NaN	0.0	0.0
0.0	70.00	N - N	0.0	0.0
934	78.99	NaN	0.0	0.0
0.0	0.00	N - N	0 0	0.0
1187	0.00	NaN	0.0	0.0
0.0				
	isd_ic_mou_6 isd_ic	mou 7 icd	ic mou Q icd	ic mou 0
ic ot	hers 6 \	_1110u_/ 15u_	_1C_1110U_0 15U	_1C_IIIOU_9
77	NaN	2.56	0.50	0.00
NaN	IVAIN	2.50	0.50	0.00
364	NaN	64.56	18.31	9.96
NaN	IVAIN	04.50	10.31	9.90
423	NaN	8.31	0.00	0.00
NaN	IVAIV	0.31	0.00	0.00
934	NaN	1.96	12.59	1.83
NaN	IVAIV	1.90	14.33	1.05
1187	NaN	0.00	0.00	0.00
NaN	IVAIV	0.00	0.00	0.00
IVAIV				
	ic others 7 ic othe	rs 8 ic oth	ners 9 total	rech num 6 \
77		3.39 ic_oti	0.0	4
364		0.00	0.0	4
JU7	0.0	0.00	0.0	Т

423 934	0.0 0.0	0.00 0.00	0.0	4 3 2	}
1187	0.0	0.00	0.0	2	
total	<pre>total_rech_num_7 rech amt 6 \</pre>	total_rech_num	_8 tot	al_rech_num_9	
77	5		3	3	
0	12		2.4	20	
364 0	12		24	20	
423	4		3	3	
252	•		3	J	
934	4		9	4	
0					
1187	20		24	6	
0					
	total rech amt 7	total rech amt	8 tot	al rech amt 9	
max r	ech amt 6 \	cotat_roon_amt	_0 :0:	a c_r cen_ame_s	
77	1154	7.	50	0	
0					
364	970	11	04	1214	
0	F01		^	202	
423 252	591		0	382	
934	1302	1	50	108	
0	1501	_		100	
1187	883	11	60	0	
0					
	max_rech_amt_7 ma	ov roch amt 0	may roc	h am+ 0	
last	day_rch_amt_6 \	ax_recii_aiiit_o	iliax_i ec	II_dIIIL_9	
77	1000	750		0	
0					
364	154	154		250	
0	220	0		252	
423 252	339	0		252	
934	550	150		54	
0	330	150		31	
1187	150	250		0	
0					
		7 7		1	
vol 2	last_day_rch_amt_7	/ last_day_rch	_amt_8	last_day_rch_am	it_9
77	g_mb_6 \ ()	750		0
0.0		,	, 50		J
364	50)	50		0
0.0					
423	(9	0		0

3.3 934		0	150		Θ
0.0		U	130		U
1187		30	0		0
0.0					
\	vol_2g_mb_7	vol_2g_mb_8	vol_2g_mb_9	vol_3g_mb_6	vol_3g_mb_7
77	96.48	0.00	0.00	0.00	0.00
364	565.78	2108.66	0.00	0.00	0.00
423	38.45	0.00	4.52	669.36	837.18
934	0.31	38.77	78.66	0.00	1045.79
1187	0.00	0.00	0.00	0.00	0.00
	vol_3g_mb_8	vol_3g_mb_9	monthly_2g_6	monthly_2g_7	
month 77	ly_2g_8 \ 0.00	0.00	Θ	1	
0					
364 1	0.00	0.00	0	1	
423	0.00	423.59	0	Θ	
0 934	245.91	471.48	0	0	
0 1187	0.00	0.00	0	0	
0	0.00	0.00	0	U	
	monthly_2g_9	sachet_2g_6	sachet_2g_7	sachet_2g_8	sachet_2g_9
\ 77	0	0	0	0	0
		0	О	U	
364	0	0	0	2	Θ
423	0	0	0	0	0
934	0	0	0	0	0
1187	0	0	0	0	0
sachet	monthly_3g_6 t_3g_6 \	monthly_3g_7	monthly_3g_	_8 monthly_3g	_9
77	0	()	0	0
0 364	0	()	0	0
0	0				J

```
423
                                1
                                               0
                                                              1
0
934
                                1
                                               1
                                                              0
0
1187
                                0
                                               0
                                                              0
      sachet 3g 7 sachet 3g 8 sachet 3g 9
                                                aon aug vbc 3g
jul_vbc_3g \
77
                0
                              0
                                            0
                                               1894
                                                            0.00
0.00
364
                 0
                              1
                                            0
                                                424
                                                            0.00
0.00
                              0
423
                 0
                                                945
                                                           73.55
266.94
934
                0
                              2
                                            1
                                                490
                                                          188.83
215.00
                0
                              0
1187
                                            0
                                                737
                                                            0.00
0.00
      jun_vbc_3g
                  sep_vbc_3g
                               avg rech amt 6 7
77
            0.00
                                           577.0
                         0.00
364
            0.00
                         0.00
                                           485.0
423
           63.04
                         0.00
                                           421.5
934
            0.00
                        24.18
                                           651.0
1187
            0.00
                         0.00
                                           441.5
# Deleting the records for which MOU for Jun(6) are null
df = df.drop(df null mou 6.index)
# Again cheking percent of missing values in columns
df missing columns =
(round(((df.isnull().sum()/len(df.index))*100),2).to_frame('null')).so
rt values('null', ascending=False)
df missing columns
                   null
loc ic t2f mou 7
                   0.12
isd ic mou 7
                   0.12
loc og t2f mou 7
                   0.12
loc og t2c mou 7
                   0.12
loc og mou 7
                   0.12
std og t2t mou 7
                   0.12
std og t2f mou 7
                   0.12
std_og_t2c_mou_7
                   0.12
std og mou 7
                   0.12
ic others 7
                   0.12
isd og mou 7
                   0.12
spl_og_mou_7
                   0.12
loc og t2t mou 7
                   0.12
```

```
og others 7
                   0.12
spl ic mou 7
                   0.12
loc ic t2t mou 7
                   0.12
std ic mou 7
                   0.12
loc ic t2m mou 7
                   0.12
std_ic_t2o_mou_7
                   0.12
std ic t2f mou 7
                   0.12
loc ic mou 7
                   0.12
std ic t2t mou 7
                   0.12
loc og t2m mou 7
                   0.12
std_og_t2m_mou_7
                   0.12
std_ic_t2m_mou_7
                   0.12
roam ic mou 7
                   0.12
onnet mou 7
                   0.12
roam og mou 7
                   0.12
offnet mou 7
                   0.12
isd_ic_mou_8
                   0.00
loc ic t2m mou 8
                   0.00
loc_ic_t2f_mou_6
                   0.00
std og t2f mou 6
                   0.00
loc og t2o mou
                   0.00
loc ic t2f mou 8
                   0.00
loc ic t2f mou 9
                   0.00
loc ic mou 6
                   0.00
loc ic mou 8
                   0.00
loc_ic_mou_9
                   0.00
std ic t2t mou 6
                   0.00
total og mou 7
                   0.00
                   0.00
total_og_mou_6
og others 9
                   0.00
og others 8
                   0.00
std og t2f mou 8
                   0.00
std og t2f mou 9
                   0.00
std_og_t2c mou 6
                   0.00
std og t2c mou 8
                   0.00
std og t2c mou 9
                   0.00
std og mou 6
                   0.00
std_og_mou_8
                   0.00
std og mou 9
                   0.00
isd og mou 6
                   0.00
                   0.00
isd og mou 8
                   0.00
isd_og_mou_9
spl og mou 6
                   0.00
spl_og_mou_8
                   0.00
spl_og_mou_9
                   0.00
og_others 6
                   0.00
avg rech amt 6 7
                   0.00
```

[178 rows x 1 columns]

Looks like MOU for all the types of calls for the month of July (7) have missing values together for any particular record.

Lets check the records for the MOU for Jul(7), in which these coulmns have missing values together.

```
# Listing the columns of MOU Jul(7)
print(((df missing columns[df missing columns['null'] ==
0.12]).index).to list())
['loc_ic_t2f_mou_7', 'isd_ic_mou_7', 'loc_og_t2f_mou_7',
'loc_og_t2c_mou_7', 'loc_og_mou_7', 'std_og_t2t_mou_7',
'std_og_t2f_mou_7', 'std_og_t2c_mou_7', 'std_og_mou_7',
                                                                 'ic others 7',
                  'spl_og_mou_7', 'loc_og_t2t_mou_7', 'og_others_7',
'isd og mou 7',
'spl_ic_mou_7', 'loc_ic_t2t_mou_7', 'std_ic_mou_7',
'loc_ic_t2m_mou_7', 'std_ic_t2o_mou_7', 'std_ic_t2f_mou_7',
'loc_ic_mou_7', 'std_ic_t2t_mou_7', 'loc_og_t2m_mou_7',
'std_og_t2m_mou_7', 'std_ic_t2m_mou_7', 'roam_ic_mou_7',
'onnet mou 7', 'roam og mou 7', 'offnet mou 7']
# Creating a dataframe with the condition, in which MOU for Jul(7) are
null
df null mou 7 = df[(df['loc og t2m mou 7'].isnull()) &
(df['loc_ic_t2f_mou_7'].isnull()) & (df['roam_og_mou_7'].isnull()) &
(df['std ic t2m mou 7'].isnull()) &
  (df['loc og t2t mou 7'].isnull()) &
(df['std ic t2t mou 7'].isnull()) & (df['loc og t2f mou 7'].isnull())
& (df['loc ic mou 7'].isnull()) &
  (df['loc og t2c mou 7'].isnull()) & (df['loc og mou 7'].isnull()) &
(df['std og t2t mou 7'].isnull()) & (df['roam ic mou 7'].isnull()) &
  (df['loc ic t2m mou 7'].isnull()) &
(df['std og t2m mou 7'].isnull()) & (df['loc_ic_t2t_mou_7'].isnull())
& (df['std_og_t2f_mou_7'].isnull()) &
  (df['std og t2c mou 7'].isnull()) & (df['og others 7'].isnull()) &
(df['std og mou 7'].isnull()) & (df['spl og mou 7'].isnull()) &
  (df['std\_ic\_t2f\_mou\_7'].isnull()) & (df['isd\_og\_mou\_7'].isnull()) &
(df['std ic mou 7'].isnull()) & (df['offnet mou 7'].isnull()) &
  (df['isd_ic_mou_7'].isnull()) & (df['ic_others_7'].isnull()) &
(df['std ic t2o mou 7'].isnull()) & (df['onnet mou 7'].isnull()) &
  (df['spl ic mou 7'].isnull())]
df null mou 7.head()
        mobile number loc og t2o mou std og t2o mou
loc_ic_t2o mou
5616
           7001238202
                                      0.0
                                                        0.0
                                                                           0.0
```

9451	700147	7649	0	. 0	0.0	0.0
9955	700165	8068	0	. 0	0.0	0.0
10724	700139	1499	0	. 0	0.0	0.0
12107	700013	1738	0	. 0	0.0	0.0
\	arpu_6	arpu_7	arpu_8	arpu_9	onnet_mou_6	onnet_mou_7
5616	760.815	531.088	992.818	1144.676	324.91	. NaN
9451	1129.566	0.000	128.252	802.648	11.89	NaN
9955	925.028	189.000	789.761	445.707	46.39	NaN
10724	894.818	85.000	207.040	363.314	117.21	. NaN
12107	1803.475	0.000	0.600	25.243	1742.61	. NaN
offnet	onnet_mou mou 8 \	_8 onnet	_mou_9 o	ffnet_mou_	6 offnet_mc	ou_7
5616 399.64	386.	13 1	180.29	350.29	9	NaN
9451	1.4	46	33.89	259.1	8	NaN
26.21 9955	43.3	39	56.61	333.7	8	NaN
196.53 10724	97.0	01	35.43	119.79	9	NaN
12.79 12107	0.0	00	0.00	278.7	9	NaN
14.29	011		0.00	27017	3	Tion to the tion t
5616 9451 9955 10724 12107		.76 .18	_ic_mou_6 463.63 9.98 0.00 0.00 0.00	roam_ic_ı	mou_7 roam_ NaN NaN NaN NaN NaN	ic_mou_8 \ 221.46 1.73 0.00 0.00 0.00
5616 9451 9955 10724 12107	roam_ic_m	0.0 0.0 0.0 0.0 0.0	m_og_mou_0 505.73 5.60 0.00 0.00	1 5 9 9	NaN NaN NaN NaN NaN	n_og_mou_8 \ 175.93 2.46 0.00 0.00 0.00
loc_og_	roam_og_m _t2t_mou_8		_og_t2t_m	ou_b loc_	og_t2t_mou_7	

5616	0.0	145.91	NaN
243.43 9451	0.0	6.73	NaN
1.46 9955	0.0	46.39	NaN
43.39 10724	0.0	115.08	NaN
97.01 12107	0.0	96.08	NaN
0.00		_	
loc_og_	_t2m_mou_8 \	loc_og_t2m_mou_6	
5616 184.78	1108.38	0.85	NaN
9451 20.54	20.84	171.46	NaN
9955 163.68	56.61	227.91	NaN
10724 6.59	34.98	86.39	NaN
12107	0.00	64.98	NaN
0.86	loc og +2m mou 0	loc on +2f mou 6	lee og +2f mau 7
	_t2f_mou_8 \	loc_og_t2f_mou_6	
5616 7.94	300.19	1.13	NaN
9451 0.00	148.88	0.00	NaN
9955 28.96	121.54	104.69	NaN
10724 6.19	55.44	17.18	NaN
12107 0.00	0.00	0.00	NaN
0.00	loc og t2f mou 9	loc og t2c mou 6	loc og t2c mou 7
loc_og_ 5616	_t2c_mou_8 \ 67.11	0.00	NaN
12.51			
9451 0.00	0.00	0.00	NaN
9955 0.00	21.04	0.00	NaN
10724 0.00	28.08	0.00	NaN
12107 13.43	0.00	50.03	NaN

5616 9451 9955 10724 12107	loc_og_t2c_mou_9 18.89 0.00 0.00 0.05 4.50	loc_og_mou_6 loc 147.89 178.19 379.01 218.66 161.06	NaN NaN NaN	_mou_8 \ 436.16 22.01 236.04 109.81 0.86
	1 0 11			
		_og_t2t_mou_6 std	_og_t2t_mou_7	
	_t2t_mou_8 \	0.00	N - N	
5616	1475.69	0.96	NaN	
17.06	160 73	F 16	NI - NI	
9451	169.73	5.16	NaN	
0.00	100 21	0.00	NI - NI	
9955	199.21	0.00	NaN	
0.00	110 51	2 12	N - N	
10724	118.51	2.13	NaN	
0.00	0.00	1040 50	N - N	
12107	0.00	1646.53	NaN	
0.00				
	s+d og +2+ mou 0	c+d og +2m mau 6	s+d og +2m mau 7	
c+d oa		std_og_t2m_mou_6	Stu_og_tzIII_IIIou_/	
5616	_t2m_mou_8 \ 69.51	15.91	NaN	
144.04	09.31	13.91	Ivaiv	
9451	13.05	0.00	NaN	
0.00	15.05	0.00	Ivaiv	
9955	0.00	0.00	NaN	
0.00	0.00	0.00	IVAIN	
10724	0.45	2.43	NaN	
0.00	0.43	2173	IVAIV	
12107	0.00	140.16	NaN	
0.00	0.00	110110	itait	
0.00				
	std og t2m mou 9	std og t2f mou 6	std og t2f mou 7	
std og	_t2f_mou_8 \		_ 3	
5616	490.61	0.00	NaN	
0.0				
9451	0.00	0.00	NaN	
0.0				
9955	1.26	1.16	NaN	
2.9				
10724	7.18	6.09	NaN	
0.0				
12107	0.00	1.26	NaN	
0.0				
	std_og_t2f_mou_9	std_og_t2c_mou_6	std_og_t2c_mou_7	
	_t2c_mou_8 \			
5616	13.33	0.0	NaN	
0.0				

9451	0	.00	0.0	NaN	
0.0					
9955	0	.00	0.0	NaN	
0.0					
10724	1	. 28	0.0	NaN	
0.0					
12107	0	.00	0.0	NaN	
0.0	O	.00	0.0	Nuiv	
0.0					
	std og t2c mo	u 9 std og mo	ou 6 std og m	ou 7 std og mou	ı 8 \
5616	3 tu_0g_t2t_iii0		5.88	NaN 161.	
9451			5.16		00
9955			. 16		90
10724			0.66		00
12107		0.0 1787	7.96	NaN 0.	00
		isd_og_mou_6	isd_og_mou_7	isd_og_mou_8	
	_mou_9 \				
5616	573.46	0.00	NaN	0.00	
0.00					
9451	13.05	74.91	NaN	4.74	
92.29					
9955	1.26	53.14	NaN	31.06	
33.69		33.1.		0_100	
10724	8.91	16.86	NaN	6.21	
2.18	0.31	10.00	Nan	0.21	
12107	0.00	0.00	NaN	0.00	
	0.00	0.00	Ivaiv	0.00	
0.00					
	col og mau 6	col og mou 7	cnl og mou 0	cnl og mou O	
og oth		Spt_og_iiiou_/	Spt_og_illou_o	spl_og_mou_9	
og_oth		N - N	12 56	10.00	
5616	4.71	NaN	12.56	18.89	
0.00	- 10		0.00	1 00	
9451	7.13	NaN	0.00	1.08	
0.00					
9955	0.00	NaN	0.00	0.00	
0.00					
10724	0.00	NaN	0.00	0.05	
0.00					
12107	72.61	NaN	13.43	4.50	
1.76					
	og others 7	og_others_8 c	a others 9 t	otal og mou 6	
total	og mou 7 \	- 3_	3_ · · · · _ ·	5	
5616	NaN	0.0	0.0	169.49	
0.0	Han	010	010	203113	
9451	NaN	0.0	0.0	265.41	
0.0	IVAIV	0.0	0.0	203.41	
9955	NaN	0.0	0.0	122 22	
	IValv	0.0	0.0	433.33	
0.0					

10724	NaN	0.0	0.0 24	6.19
0.0				
12107	NaN	0.0	0.0 202	3.41
0.0				
0.0				
	+-+-1 0 +	-a+al ag mau 0 la	o i o +2+ mou 6	
		otal_og_mou_9 lo	C_IC_LZL_IIIOU_6	
	_t2t_mou_7 \			
5616	609.84	2068.06	78.76	
NaN				
9451	26.76	276.16	17.24	
NaN				
9955	270.01	234.18	80.98	
NaN	270.01	254110	00.50	
	116 02	120 66	007.04	
10724	116.03	129.66	887.04	
NaN				
12107	14.29	4.50	65.76	
NaN				
	loc ic t2t mou 8	loc ic t2t mou 9	locic t2m mo	u 6
loc ic	t2m mou 7 \			
5616	233.66	558.84	1	. 36
NaN	255.00	330.04	_	
	0.60	36 60	120	00
9451	0.60	36.69	130	.09
NaN				
9955	32.69	112.14	201	. 38
NaN				
10724	200.51	408.66	104	. 18
NaN				
12107	1.73	5.88	92	. 18
NaN		2.00		0
itait				
	loc ic t2m mou 8	loc ic t2m mou 9	locic +2f mo	6
loc ic	t2f mou 7 \	toc_1c_tzm_mou_3	, , , , , , , , , , , , , , , , , , , ,	u_0
		75 21	C	61
5616	11.53	75.31	. 0	.61
NaN				
9451	16.54	110.19) 25	. 46
NaN				
9955	169.24	155.58	3 41	. 68
NaN				
10724	22.24	76.39	16	.74
NaN		, 5155		.,.
12107	5.59	2.75		.00
	5.59	2.75		.00
NaN				
	<pre>loc_ic_t2f_mou_8</pre>	loc_ic_t2f_mou_9	loc_ic_mou_6	loc_ic_mou_7
\				
5616	0.00	31.81	. 86.74	NaN
9451	8.76	40.24	172.81	NaN

9955	25.68	12.33	324.04	NaN
10724	1.61	28.18	1007.98	NaN
12107	0.00	0.00	157.94	NaN
\	loc_ic_mou_8 loc	_ic_mou_9 std_ic_	_t2t_mou_6 std_ic	_t2t_mou_7
5616	245.19	665.98	0.00	NaN
9451	25.91	187.14	1.50	NaN
9955	227.63	280.06	0.00	NaN
10724	224.38	513.24	0.00	NaN
12107	7.33	8.63	103.66	NaN
		std_ic_t2t_mou_9	std_ic_t2m_mou_6	
5616	_t2m_mou_7 \ 12.13	42.39	21.76	
NaN	12.13	12133	211,70	
9451	0.00	0.00	0.41	
NaN 9955	0.00	0.00	0.98	
NaN				
10724 NaN	0.00	0.00	5.94	
12107	0.00	0.00	3.01	
NaN				
		std_ic_t2m_mou_9	std_ic_t2f_mou_6	
$561\overline{6}$	_t2f_mou_7 \ 110.99	263.98	0.0	
NaN 9451	0.00	12.29	0.0	
NaN	0.00	12.23	0.0	
9955	2.13	2.58	0.0	
NaN 10724	0.00	4.88	0.0	
NaN				
12107 NaN	0.00	0.00	0.0	
	std ic t2f mou 8	std ic t2f mou 9	std ic t2o mou 6	
std_ic	_t2o_mou_7 \	3 tu_1t_ t21_mou_9	3 td_10_ t20_1110d_0	
561 6	0.00	6.43	0.0	
NaN				

9451 NaN	0.00		4.48	0.0	
9955	0.23		0.00	0.0	
NaN 10724	10.03		1.26	0.0	
NaN 12107	0.00		0.00	0.0	
NaN					
\	std_ic_t2o_mou_8	std_ic_t2o_r	mou_9 std_	_ic_mou_6 sto	l_ic_mou_7
5616	0.0		0.0	21.76	NaN
9451	0.0		0.0	1.91	NaN
9955	0.0		0.0	0.98	NaN
10724	0.0		0.0	5.94	NaN
12107	0.0		0.0	106.68	NaN
				6	
5616 9451 9955 10724 12107	std_ic_mou_8 std 123.13 0.00 2.36 10.03 0.00	I_ic_mou_9 to 312.81 16.78 2.58 6.14 0.00	otal_ic_mou 189. 217. 332. 1140. 265.	81 33 33 54	mou_7 \ 0.0 0.0 0.0 0.0 0.0 0.0
5616 9451 9955 10724 12107	total_ic_mou_8 t 397.13 43.44 506.94 342.78 7.33	cotal_ic_mou_9 1020.16 307.43 526.54 642.33 8.63	6	nou_6 spl_ic_ 0.00 0.00 0.00 0.14 0.00	mou_7 \ NaN NaN NaN NaN NaN NaN NaN
icd ic	spl_ic_mou_8 spl	_ic_mou_9 i	sd_ic_mou_6	isd_ic_mou_	_7
isd_ic_ 5616	0.00	0.13	81.29) Na	N
28.79 9451	0.00	0.00	42.59	Na Na	N
17.53 9955	0.00	0.00	7.29) Na	N
173.61 10724	0.08	0.09	126.13	Na Na	N
106.53 12107 0.00	0.00	0.00	0.00) Na	ıN
	isd_ic_mou_9 ic_	others_6 ic	_others_7	ic_others_8	

ic_others_9					
5616	41.23	0.00	NaN	0.00	
0.00 9451	103.49	0.00	NaN	0.00	
0.00	105.49	0.00	Ivaiv	0.00	
9955	229.44	0.00	NaN	103.33	
14.45					
10724	116.83	0.33	NaN	1.74	
5.99 12107	0.00	0.40	NaN	0.00	
0.00	0.00	0.40	Ivaiv	0.00	
		total_rech_n	um_7 total	_rech_num_8	
total_rech_	num_9 \ 5		7	0	
5616 13	5		1	9	
9451	14		4	4	
9					
9955	6		1	4	
3	0		2	2	
10724 5	8		3	3	
12107	17		2	1	
2					
	1				
		total_rech_a	mt_7 total	_rech_amt_8	
total_rech_	$amt_9 \setminus$	total_rech_a	_		
		total_rech_a	mt_7 total 780	_rech_amt_8 904	
total_rech_ 5616 1591 9451	$amt_9 \setminus$	total_rech_a	_		
total_rech_ 5616 1591 9451 991	amt_9 \ 776	total_rech_a	780 0	904	
total_rech_ 5616 1591 9451 991 9955	amt_9 \ 776	total_rech_a	780	904	
total_rech_ 5616 1591 9451 991 9955 912	amt_9 \ 776 1206 1385	total_rech_a	780 0 0	904 223 835	
total_rech_ 5616 1591 9451 991 9955	amt_9 \ 776 1206 1385 1020	total_rech_a	780 0 0	904 223 835 360	
total_rech_ 5616 1591 9451 991 9955 912 10724 480 12107	amt_9 \ 776 1206 1385	total_rech_a	780 0 0	904 223 835	
total_rech_ 5616 1591 9451 991 9955 912 10724 480	amt_9 \ 776 1206 1385 1020	total_rech_a	780 0 0	904 223 835 360	
total_rech_ 5616 1591 9451 991 9955 912 10724 480 12107 30	amt_9 \ 776 1206 1385 1020 1990		780 0 0 0	904 223 835 360	amt 9
total_rech_ 5616 1591 9451 991 9955 912 10724 480 12107 30	amt_9 \ 776 1206 1385 1020 1990	total_rech_a nax_rech_amt_7	780 0 0 0	904 223 835 360	_amt_9
total_rech_ 5616 1591 9451 991 9955 912 10724 480 12107 30	amt_9 \ 776 1206 1385 1020 1990		780 0 0 0 0 max_rech_	904 223 835 360	_amt_9 289
total_rech_ 5616 1591 9451 991 9955 912 10724 480 12107 30 max_\ 5616	amt_9 \ 776 1206 1385 1020 1990 rech_amt_6 m	nax_rech_amt_7 330	780 0 0 0 0 max_rech_	904 223 835 360 0 amt_8 max_rech_	289
total_rech_ 5616 1591 9451 991 9955 912 10724 480 12107 30 max_ \	amt_9 \ 776 1206 1385 1020 1990 rech_amt_6 m	nax_rech_amt_7	780 0 0 0 0 max_rech_	904 223 835 360 0 amt_8 max_rech	_
total_rech_ 5616 1591 9451 991 9955 912 10724 480 12107 30 max_\ 5616	amt_9 \ 776 1206 1385 1020 1990 rech_amt_6 m	nax_rech_amt_7 330	780 0 0 0 0 max_rech_	904 223 835 360 0 amt_8 max_rech_	289
total_rech_ 5616 1591 9451 991 9955 912 10724 480 12107 30 max_\ 5616 9451	amt_9 \ 776 1206 1385 1020 1990 rech_amt_6 m 250 250	nax_rech_amt_7 330 0	780 0 0 0 0 max_rech_	904 223 835 360 0 amt_8 max_rech 200 130 300	289 130 479
total_rech_ 5616 1591 9451 991 9955 912 10724 480 12107 30 max_\ 5616	amt_9 \ 776 1206 1385 1020 1990 rech_amt_6 m 250	nax_rech_amt_7 330 0	780 0 0 0 0 max_rech_	904 223 835 360 0 amt_8 max_rech 200 130	289 130
total_rech_ 5616 1591 9451 991 9955 912 10724 480 12107 30 max_\ 5616 9451 9955 10724	amt_9 \ 776 1206 1385 1020 1990 rech_amt_6 m 250 250 350 500	nax_rech_amt_7 330 0 0	780 0 0 0 0 max_rech_	904 223 835 360 0 amt_8 max_rech 200 130 300 130	289 130 479 150
total_rech_ 5616 1591 9451 991 9955 912 10724 480 12107 30 max_\ 5616 9451	amt_9 \ 776 1206 1385 1020 1990 rech_amt_6 m 250 250	nax_rech_amt_7 330 0	780 0 0 0 0 max_rech_	904 223 835 360 0 amt_8 max_rech 200 130 300	289 130 479

5616 9451 9955 10724 12107	last_day_rch	_amt_6 la: 250 250 250 250 500 110	st_day_rch	n_amt_7 la 0 0 0 0 0	ast_day_rch	_amt_8 \ 130 130 300 130 0
5616	last_day_rch _mb_9 \	_amt_9 vo [·] 250	0.00		b_7 vol_2g 9.0	_mb_8 11.26
83.32 9451 431.85		130	321.86		9.0	0.00
9955 0.00		479	0.00		9.0	0.00
10724 0.00 12107		0 30	0.00		9.0	0.00
0.00	vol 3g mb 6	vol_3g_mb	7 vol 30	ımb 8 vo	1 3a mh 0	
5616	y_2g_6 \ 0.0		_/ VOC_5g .0	79.94	668.4	
0 9451 1	0.0	0	. 0	0.00	0.0	
9955 0	0.0		. 0	0.00	0.0	
10724 0 12107	0.0		. 0	0.00	0.0	
0						G
sachet _s	monthly_2g_7 _2g_7 \ _0	monthly_2	2g_8 011t	:hly_2g_9 1	sachet_2g_	0
0 9451 0	0		0	1		1
9955 0	0		0	0		0
10724 0 12107	9		0	9		0
0		an ab at 2				
monthly 5616	sachet_2g_8 y_3g_8 \ 0	sachet_2g	_9 monthl 0	.y_3g_6 m	onthly_3g_7 0	

```
0
9451
                               2
                                              0
                                                            0
0
9955
                               0
                                                            0
                               0
                                                            0
10724
0
12107
                               0
                                              0
                                                            0
       monthly_3g_9 sachet_3g_6 sachet_3g_7 sachet_3g_8
sachet 3g 9
              aon \
5616
                  0
                                0
                                              0
                                                           0
0
    576
                                                           0
9451
                   0
                                0
0 672
9955
                   0
                                0
                                                           0
0 3107
10724
                   0
                                0
                                                           0
0 2664
                   0
                                0
                                                           0
12107
  219
       aug vbc 3g jul vbc 3g jun vbc 3g
                                             sep vbc 3g
avg rech amt 6 7
            63.38
5616
                           0.0
                                       0.0
                                                 163.39
778.0
9451
             0.00
                           0.0
                                       0.0
                                                   0.00
603.0
9955
             0.00
                           0.0
                                       0.0
                                                   0.00
692.5
10724
             0.00
                           0.0
                                       0.0
                                                   0.00
510.0
             0.00
12107
                           0.0
                                       0.0
                                                   0.00
995.0
# Deleting the records for which MOU for Jul(7) are null
df = df.drop(df_null_mou_7.index)
# Again cheking percent of missing values in columns
df missing columns =
(round(((df.isnull().sum()/len(df.index))*100),2).to frame('null')).so
rt values('null', ascending=False)
df missing columns
                   null
mobile number
                   0.0
total_rech_num_7
                   0.0
std_ic_mou_7
                   0.0
std ic mou 8
                    0.0
```

```
std ic mou 9
                    0.0
total ic mou 6
                    0.0
total ic mou 7
                    0.0
total ic mou 8
                    0.0
total ic mou 9
                    0.0
spl_ic_mou_6
                    0.0
spl ic mou 7
                    0.0
spl ic mou 8
                    0.0
spl_ic_mou 9
                    0.0
isd ic mou 6
                    0.0
isd_ic_mou 7
                    0.0
isd_ic_mou_8
                    0.0
isd ic mou 9
                    0.0
ic others 6
                    0.0
ic others 7
                    0.0
ic others 8
                    0.0
ic others 9
                    0.0
std ic mou 6
                    0.0
std ic t2o mou 9
                    0.0
std_ic_t2o_mou_8
                    0.0
std ic t2t mou 9
                    0.0
loc ic t2f mou 9
                    0.0
                    0.0
loc ic mou 6
loc_ic_mou_7
                    0.0
loc_ic_mou 8
                    0.0
loc_ic_mou_9
                    0.0
                    . . .
loc ic t2t mou 6
                    0.0
loc_ic_t2t_mou_7
                    0.0
                    0.0
loc_ic_t2t_mou_8
loc ic t2t mou 9
                    0.0
loc_ic_t2m_mou_6
                    0.0
loc ic t2m mou 7
                    0.0
loc ic t2m mou 8
                    0.0
                    0.0
spl og mou 6
isd og mou 8
                    0.0
std og t2t mou 8
                    0.0
std og t2f mou 9
                    0.0
std_og_t2t_mou_9
                    0.0
std og t2m mou 6
                    0.0
std_og_t2m_mou_7
                    0.0
std og t2m mou 8
                    0.0
std_og_t2m_mou_9
                    0.0
std_og_t2f_mou_6
                    0.0
std_og_t2f_mou_7
                    0.0
std_og_t2f_mou_8
                    0.0
                    0.0
std og t2c mou 6
isd og mou 7
                    0.0
std og t2c mou 7
                    0.0
```

```
0.0
std og t2c mou 8
std og t2c mou 9
                    0.0
std og mou 6
                    0.0
std og mou 7
                    0.0
std og mou 8
                    0.0
std og mou 9
                    0.0
isd og mou 6
                    0.0
avg rech amt 6 7
                    0.0
[178 rows x 1 columns]
```

We can see there are no more missing values in any columns.

```
df.shape
(27991, 178)
# Checking percentage of rows we have lost while handling the missing values
round((1- (len(df.index)/30011)),2)
0.07
```

We can see that we have lost almost 7% records. But we have enough number of records to do our analysis.

Tag churners

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.

```
df['churn'] = np.where((df['total ic mou 9']==0) &
(df['total og mou 9']==0) & (df['vol 2g mb 9']==0) &
(df['vol 3g mb 9']==0), 1, 0)
df.head()
    mobile number loc og t2o mou std og t2o mou loc ic t2o mou
arpu 6 \
       7001524846
                               0.0
                                               0.0
                                                                0.0
378.721
13
       7002191713
                               0.0
                                               0.0
                                                                0.0
492.846
                                                                0.0
       7000875565
                               0.0
                                               0.0
16
430.975
                               0.0
                                                                0.0
17
       7000187447
                                               0.0
690,008
                               0.0
                                               0.0
                                                                0.0
21
       7002124215
514.453
```

	arpu_7		arpu_9	onnet_mou_6	onnet_mou_7	
onne 8	et_mou_8 492.223		166.787	413.69	351.03	35.08
13	205.671	593.260	322.732	501.76	108.39	534.24
16	299.869	187.894	206.490	50.51	74.01	70.61
17	18.980	25.499	257.583	1185.91	9.28	7.79
21	597.753	637.760	578.596	102.41	132.11	85.14
offi	onnet_m net_mou_		et_mou_6	offnet_mou_7	offnet_mou_8	
8 108	_ 3	3.46	94.66	80.63	136.48	
13	24	4.81	413.31	119.28	482.46	
214 16	3	1.34	296.29	229.74	162.76	
224 17	55	8.51	61.64	0.00	5.54	
87.8 21		1.63	757.93	896.68	983.39	
869	. 89					
500	roam_ic		am_ic_mou_	_7 roam_ic_m	ou_8 roam_ic_	mou_9
8	m_og_mou	0.00	0.0	00	0.00	0.00
0.00 13	9	23.53	144.2	24 7:	2.11 1	36.78
7.98 16	3	0.00	2.8	33	0.00	0.00
0.00 17	9	0.00	4.7		4.81	0.00
0.0	9					
21 0.0	9	0.00	0.6	90	9.00	0.00
	roam_og		oam_og_mou_			
8 13		0.00 35.26	0.0 1.4		0.00 2.78	297.13 49.63
16 17		17.74 8.46	0.0 13.3		0.00 7.98	42.61 38.99
21		0.00	0.6		0.00	4.48
100			loc_og_t2	2t_mou_8 loc	_og_t2t_mou_9	
8	_09_t2111_	mou_6 \ 217.59		12.49	26.13	

80.96				
13	6.19	36.01	6.14	
151.13 16	65.16	67.38	26.88	
273.29	05.10	07.50	20.00	
17	0.00	0.00	36.41	
58.54 21	6 16	22.24	20.00	
91.81	6.16	23.34	29.98	
31101				
loc_og_ loc_og_t2f_		loc_og_t2m_mou_8	loc_og_t2m_mou_9	
8	70.58	50.54	34.58	
0.00	47 20	204 46	100 24	
13 4.54	47.28	294.46	108.24	
16	145.99	128.28	201.49	
0.00				
17	0.00	0.00	9.38	
0.00 21	87.93	104.81	107.54	
0.75	07.00	101101	207.13	
1	+24 7	1 +24 0	1 +24 0	
loc_og_ loc og t2c		loc_og_t2f_mou_8	loc_og_tzt_mou_9	
8	0.00	0.00	0.00	
0.0				
13	0.00	23.51	5.29	
0.0 16	4.48	10.26	4.66	
0.0		20120	1100	
17	0.00	0.00	0.00	
0.0 21	0.00	1.58	0.00	
0.0	0.00	1.50	0.00	
				_
	_t2c_mou_7	loc_og_t2c_mou_8	loc_og_t2c_mou_9	loc_og_mou_6
8	0.0	7.15	0.0	378.09
13	0.0	0.49	0.0	205.31
16	0.0	0.00	0.0	315.91
17	0.0	0.00	0.0	97.54
21	0.0	0.00	0.0	97.04
loc_og_	_mou_7 loc	_og_mou_8 loc_og_	mou_9 std_og_t2t_	mou_6 \

8 13 16 17 21	288.18 53.48 215.64 0.00 94.09	353.99 1 205.93 2 0.00	.19.69 4 33.04	.16.56 446.41 7.89 .46.91 97.93
std_og_t2 8 13.69 13 255.36 16 22.99	eg_t2t_mou_7 em_mou_6 \ 133.43 85.98 2.58	22.58 498.23 3.23	std_og_t2t_mou_9 7.33 230.38 4.46	
17 1.55 21 665.36	0.81 125.94	0.00 61.79	504.11 131.64	
	og_t2m_mou_7 2f_mou_6 \ 10.04	std_og_t2m_mou_8 75.69	std_og_t2m_mou_9 74.13	
13 0.0 16 0.0	52.94 64.51	156.94 18.29	96.01 13.79	
17 0.0 21 0.0	0.00 808.74	0.00 876.99	78.51 762.34	
std_og_t2 8 0.0	og_t2f_mou_7 c_mou_6 \ 0.0	0.0		
13 0.0 16 0.0 17	0.0 0.0 0.0	0.0 0.0 0.0	0.00 4.43 0.00	
0.0 21 0.0	0.0	0.0	0.00	
std_c 8	0.0 0.0	std_og_t2c_mou_8 0.0 0.0	std_og_t2c_mou_9 0.0 0.0	std_og_mou_6 130.26 701.78

16	0.0	0.0	0.0	30.89
17	0.0	0.0	0.0	1148.46
21	0.0	0.0	0.0	763.29
std_og_mou_7 isd_og_mou_7 \	std_og_mou_8	std_og_mou_9	isd_og_mou_6	
8 143.48 0.0	98.28	81.46	0.0	
13 138.93 0.0	655.18	326.39	0.0	
16 67.09	21.53	22.69	0.0	
0.0 17 0.81	0.00	582.63	0.0	
0.0 21 934.69	938.79	893.99	0.0	
0.0				
isd_og_mou_8 spl og mou 8 \	isd_og_mou_9	spl_og_mou_6	spl_og_mou_7	
8 0.00 10.23	0.0	0.00	0.00	
13 1.29	0.0	0.00	0.00	
4.78 16 0.00	0.0	0.00	3.26	
5.91 17 0.00	0.0	2.58	0.00	
0.00 21 0.00	0.0	0.00	0.00	
0.00				
<pre>spl_og_mou_9 og_others_9 \</pre>	og_others_6	og_others_7 o	g_others_8	
8 0.00	0.00	0.0	0.0	0.0
13 0.00	0.00	0.0	0.0	0.0
16 0.00	0.00	0.0	0.0	0.0
17 2.64	0.93	0.0	0.0	0.0
21 0.00	0.00	0.0	0.0	0.0
total_og_mou 8 508.			mou_8 total_o 71.56	g_mou_9 \ 142.18
13 907.			15.26	446.09

16 17 21	346.81 1249.53 860.34	286.01 0.81 1028.79	233.38 0.00 1068.54	255.74 631.08 1031.53
	ic_t2t_mou_6 2t_mou_9 \	loc_ic_t2t_mou_7	loc_ic_t2t_mou_8	3
8	23.84	9.84	0.33	L
13 24.98	67.88	7.58	52.58	3
16 50.23	41.33	71.44	28.89)
17 40.91	34.54	0.00	0.00)
21 17.99	2.48	10.19	19.54	1
	ic_t2m_mou_6 2m mou 9 \	loc_ic_t2m_mou_7	loc_ic_t2m_mou_8	3
8 17.34	57.58	13.98	15.48	3
17.34 13 104.79	142.88	18.53	195.18	3
16	226.81	149.69	150.16	5
172.86 17	47.41	2.31	0.00)
43.86 21 113.46	118.23	74.63	129.16	5
	ic t2f mou 6	loc ic t2f mou 7	loc ic t2f mou 8	3
_	2f_mou_9 \ 0.00	0.00	0.00	
0.00				
13 8.51	4.81	0.00	7.49	
16 65.21	8.71	8.68	32.73	
17 0.71	0.00	0.00	0.00)
21 8.41	4.61	2.84	10.39)
		_ic_mou_7 loc_ic_	_mou_8 loc_ic_mou	_ 9
std_ic_t2	2t_mou_6 \ 81.43	23.83	15.79 21	. 38
0.00 13	215.58	26.11 2	255.26 138	. 29
115.68 16	276.86	229.83 2	211.78 288	.31

68.79				
17	81.96	2.31	0.00 85.4	19
8.63				
21	125.33	87.68 1	59.11 139.8	38
14.06				
std	ic t2t mou 7	std ic t2t mou 8	std ic t2t mou 9	
	2m mou 6 \	5 tu_10_ t2 t5u_5	5 tu_10_t1 t6u_5	
3 3	0.58	0.10	0.00	
22.43	0.50	0.10	0.00	
	20.20	154 50	62.20	
13	38.29	154.58	62.39	
308.13				
16	78.64	6.33	16.66	
L8.68				
17	0.00	0.00	0.00	
L.28				
21	5.98	0.18	16.74	
67.69				
std	ic t2m mou 7	std ic t2m mou 8	std ic t2m mou 9	
	2f_mou_6 \			
8	4.08	0.65	13.53	
0.00	1100	0103	13.33	
13	29.79	317.91	151.51	
9.00	29.79	517.91	151.51	
	72 00	72 02	20 50	
16	73.08	73.93	29.58	
0.51	0.00	0.00	1 62	
17	0.00	0.00	1.63	
0.00			0= 00	
21	38.23	101.74	95.98	
0.00				
		std_ic_t2f_mou_8	std_ic_t2f_mou_9	
	2o_mou_6 \			
	0.0	0.00	0.0	
9.0				
13	0.0	1.91	0.0	
0.0				
16	0.0	2.18	0.0	
0.0				
17	0.0	0.00	0.0	
9.0			2.10	
21	0.0	0.00	0.0	
9.0	0.0	0.00	0.0	
0.0				
c+d	ic +20 may 7	std ic t2o mou 8	std_ic_t2o_mou_9	std ic mou
	10_120_1110u_/	Jtd_It_tZU_IIIUd_0	3 td_10_t20_iii0d_9	3 ca_rc_iiiou
\ 8	0.0	0.0	0.0	22.
J	0.0	0.0	0.0	22.
13	0.0	0.0	0.0	423.
13	0.0	0.0	0.0	423.

16	0.0		0.0	0.0	87.99
17	0.0		0.0	0.0	9.91
21	0.0		0.0	0.0	81.76
		_ic_mou_8 st	d_ic_mou_9	total_ic_mou_0	6
totat_1c_ 8	_mou_7 \ 4.66	0.75	13.53	103.80	6
28.49 13	68.09	474.41	213.91	968.6	1
172.58					
16 381.56	151.73	82.44	46.24	364.80	0
17	0.00	0.00	1.63	91.88	8
2.31 21	44.21	101.93	112.73	207.09	9
131.89					
		otal_ic_mou_9	spl_ic_mou	u_6 spl_ic_mo	u_7
spl_ic_mo	ou_8 \ 16.54	34.91	Θ	. 00	0.0
0.0	10.54	54.51	0	.00	5.0
13	1144.53	631.86	Θ	. 45	0.0
0.0 16	294.46	334.56	Θ	. 00	0.0
0.0	231110	331130	0		
17	0.00	87.13	0	. 00	0.0
0.0 21	261.04	252.61	A	. 00	0.0
0.0	201.04	232.01	0	.00	5.0
spl_i isd ic mo		_ic_mou_6 is	d_ic_mou_7	isd_ic_mou_8	
8 0.00	0.0	0.00	0.00	0.00	
13	0.0	245.28	62.11	393.39	
259.33	0.0	0.00	0.00	0.22	
16 0.00	0.0	0.00	0.00	0.23	
17	0.0	0.00	0.00	0.00	
0.00 21	0.0	0.00	0.00	0.00	
0.00	0.0	0.00	0.00	0.00	
ic_ot total red		thers_7 ic_o	tners_8 ic_	_others_9	
8	0.00	0.00	0.00	0.00	

19					
13	83.48	16.24	21.44	20.31	
6					
16 10	0.00	0.00	0.00	0.00	
17	0.00	0.00	0.00	0.00	
19	0.00	0.00			
21	0.00	0.00	0.00	0.00	
22					
	total rech num 7	total rech n	um 8 tota	al rech num	9
	al_rech_amt_6 \		_		
8	21		14	1	5
437 13	4		11		7
507					,
16	6		2		1
570	2		1	1	0
17 816	2		4	1	0
21	26		27	1	7
600					
max_	total_rech_amt_7 _rech_amt_6 \			al_rech_amt_	
8	601		120	18	6
90 13	253		717	35	3
110	233		, 1,	33	3
16	348		160	22	0
110 17	0		30	33	5
110	U		30	33	J
21	680		718	68	0
50					
	max_rech_amt_7	max rech amt 8	max rec	h amt 9 las	t dav rch amt 6
8					
8	154	30		36	50
13	110	130		130	110
16	110	130		220	100
17	0	30		130	30
21	50	50		50	30
2.1	30	30		30	30
	last_day_rch_amt	_7 last_day_r	ch_amt_8	last_day_rc	h_amt_9

vol 2g mb 6 \						
8	0	10		Θ		
0.0	50	0		0		
0.0 16	100	130		220		
0.0 17	0	Θ		0		
0.0	20	50		30		
0.0						
vol_2g_mb_7 8 356.0 13 0.0 16 0.0 17 0.0 21 0.0	vol_2g_mb_8 0.03 0.02 0.00 0.00	vol_2g_mb_9 0.0 0.0 0.0 0.0 0.0	vol_3g_mb_6 v 0.0 0.0 0.0 0.0 0.0	ol_3g_mb_7 \ 750.95 0.00 0.00 0.00 0.00		
	vol_3g_mb_9	monthly_2g_6	monthly_2g_7	monthly_2g_8		
8 11.94	0.0	0	1	0		
13 0.00	0.0	0	0	0		
16 0.00	0.0	0	0	0		
17 0.00	0.0	0	0	0		
21 0.00	0.0	Θ	0	0		
<pre>monthly_2g_9 sachet_2g_6 sachet_2g_7 sachet_2g_8 sachet_2g_9 \</pre>						
8 0	0	1	3	0		
13 0		0	3	0		
16 0	0	0	0	Θ		
17 0	0	0	0	Θ		
21 0		0	0	0		
monthly_3g_6 sachet_3g_6 \ 8 0			_8 monthly_3g_ 0	9		
0						
13 0 0	()	0	0		

16	0	0		0	0
0					
17	0	0		0	0
0					
21	0	0		0	0
0					
		chet_3g_8 sach	et_3g_9	aon	aug_vbc_3g
jul_vbc_ 8	3g \ 0	0	Θ	315	21.03
910.65	U	ð	U	213	21.03
13	Θ	0	0	2607	0.00
0.00	O	U	O .	2007	0.00
16	Θ	0	0	511	0.00
2.45		· ·	· ·	011	0.00
17	0	0	Θ	667	0.00
0.00					
21	0	0	0	720	0.00
0.00					
jun_	vbc_3g sep	_vbc_3g avg_re	ch_amt_6	_7 ch	urn
	122.16	0.0	519		0
13	0.00	0.0	380		0
16	21.89	0.0	459		0
17	0.00	0.0	408		0
21	0.00	0.0	640	. 0	0

Deleting all the attributes corresponding to the churn phase

```
# List the columns for churn month(9)
col_9 = [col for col in df.columns.to_list() if '_9' in col]
print(col_9)

['arpu_9', 'onnet_mou_9', 'offnet_mou_9', 'roam_ic_mou_9',
'roam_og_mou_9', 'loc_og_t2t_mou_9', 'loc_og_t2m_mou_9',
'loc_og_t2f_mou_9', 'loc_og_t2c_mou_9', 'loc_og_mou_9',
'std_og_t2t_mou_9', 'std_og_t2m_mou_9', 'std_og_t2f_mou_9',
'std_og_t2c_mou_9', 'std_og_mou_9', 'isd_og_mou_9', 'spl_og_mou_9',
'og_others_9', 'total_og_mou_9', 'loc_ic_t2t_mou_9',
'loc_ic_t2m_mou_9', 'loc_ic_t2f_mou_9', 'loc_ic_t2f_mou_9',
'std_ic_t2t_mou_9', 'std_ic_t2m_mou_9', 'std_ic_t2f_mou_9',
'std_ic_t2o_mou_9', 'std_ic_mou_9', 'total_ic_mou_9', 'spl_ic_mou_9',
'isd_ic_mou_9', 'ic_others_9', 'total_rech_num_9', 'total_rech_amt_9',
'max_rech_amt_9', 'last_day_rch_amt_9', 'vol_2g_mb_9', 'vol_3g_mb_9',
'monthly_2g_9', 'sachet_2g_9', 'monthly_3g_9', 'sachet_3g_9']

# Deleting the churn month columns
df = df.drop(col_9, axis=1)
```

```
# Dropping sep_vbc_3g column
df = df.drop('sep_vbc_3g', axis=1)
```

Checking churn percentage

```
round(100*(df['churn'].mean()),2)
3.39
```

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

Outliers 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.

```
df['mobile number'] = df['mobile number'].astype(object)
df['churn'] = df['churn'].astype(object)
df.info()
<class 'pandas.core.frame.DataFrame'>
Int64Index: 27991 entries, 8 to 99997
Columns: 136 entries, mobile number to churn
dtypes: float64(109), int64(\overline{25}), object(2)
memory usage: 29.3+ MB
# List only the numeric columns
numeric cols = df.select dtypes(exclude=['object']).columns
print(numeric cols)
Index(['loc_og_t2o_mou', 'std_og_t2o_mou', 'loc_ic_t2o_mou', 'arpu_6',
       'arpu 7', 'arpu 8', 'onnet mou 6', 'onnet mou 7',
'onnet mou 8^{-},
       'offnet mou 6',
       'monthly_3g_7', 'monthly_3g_8', 'sachet_3g_6', 'sachet_3g_7',
       'sachet_3g_8', 'aon', 'aug_vbc_3g', 'jul_vbc_3g', 'jun_vbc_3g',
       'avg_rech_amt_6_7'],
      dtype='object', length=134)
# Removing outliers below 10th and above 90th percentile
for col in numeric cols:
    q1 = df[col].quantile(0.10)
    q3 = df[col].quantile(0.90)
    iqr = q3-q1
    range low = q1-1.5*iqr
    range high = q3+1.5*igr
    # Assigning the filtered dataset into data
    data = df.loc[(df[col] > range low) & (df[col] < range high)]</pre>
```

```
data.shape
(27705, 136)
```

Derive new features

```
# List the columns of total mou, rech_num and rech_amt
[total for total in data.columns.to_list() if 'total' in total]

['total_og_mou_6',
   'total_og_mou_8',
   'total_ic_mou_6',
   'total_ic_mou_7',
   'total_ic_mou_8',
   'total_rech_num_6',
   'total_rech_num_7',
   'total_rech_amt_6',
   'total_rech_amt_6',
   'total_rech_amt_7',
   'total_rech_amt_8']
```

Deriving new column decrease mou_action

This column indicates whether the minutes of usage of the customer has decreased in the action phase than the good phase.

```
# Total mou at good phase incoming and outgoing
data['total mou good'] = (data['total og mou 6'] +
data['total ic mou 6'])
# Avg. mou at action phase
# We are taking average because there are two months(7 and 8) in
action phase
data['avg mou action'] = (data['total og mou 7'] +
data['total_og_mou_8'] + data['total_ic_mou_7'] +
data['total ic mou 8'])/2
# Difference avg mou good and avg mou action
data['diff mou'] = data['avg mou action'] - data['total mou good']
# Checking whether the mou has decreased in action phase
data['decrease mou action'] = np.where((data['diff mou'] < 0), 1, 0)</pre>
data.head()
   mobile_number loc_og_t2o_mou std_og_t2o_mou loc_ic_t2o_mou
arpu_6 \
      7001524846
                             0.0
                                             0.0
                                                              0.0
378.721
```

13 700219 492.846	91713	0.0	0.0	0.0		
16 700087	75565	0.0	0.0	0.0		
430.975 17 700018	37447	0.0	0.0	0.0		
690.008 21 700212 514.453	24215	0.0	0.0	0.0		
arpu_7 offnet mou 0		onnet_mou_6 o	onnet_mou_7 onn	et_mou_8		
	137.362	413.69	351.03	35.08		
13 205.671 413.31	593.260	501.76	108.39	534.24		
16 299.869 296.29	187.894	50.51	74.01	70.61		
17 18.980	25.499	1185.91	9.28	7.79		
61.64 21 597.753 757.93	637.760	102.41	132.11	85.14		
		et_mou_8 roar	m_ic_mou_6 roam	_ic_mou_7		
	_8 \ 30.63	136.48	0.00	0.00		
	19.28	482.46	23.53	144.24		
	29.74	162.76	0.00	2.83		
0.00 17	0.00	5.54	0.00	4.76		
4.81 21 89 0.00	96.68	983.39	0.00	0.00		
roam_og_				oc_og_t2t_mou_6 \		
8 13	0.00 7.98	0.00 35.26	0.00 1.44	297.13 49.63		
16 17	0.00 0.00	17.74 8.46	0.00 13.34	42.61 38.99		
21	0.00	0.00	0.00	4.48		
<pre>loc_og_t2t_mou_7 loc_og_t2t_mou_8 loc_og_t2m_mou_6 loc_og_t2m_mou_7 \</pre>						
8 70.58	217.59	12	. 49	80.96		
13 47.28	6.19	36	.01 1	51.13		
16 145.99	65.16	67	.38 2	73.29		

17	0.00	0.00	58.54	
0.00 21	6.16	23.34	91.81	
87.93				
		loc_og_t2f_mou_6	loc_og_t2f_mou_7	
loc_og_t2f _. 8	_mou_8 \ 50.54	0.00	0.00	
0.00 13	294.46	4.54	0.00	
23.51				
16 10.26	128.28	0.00	4.48	
17	0.00	0.00	0.00	
0.00 21	104.81	0.75	0.00	
1.58	101101	0173	0100	
	_t2c_mou_6	loc_og_t2c_mou_7	loc_og_t2c_mou_8	loc_og_mou_6
8	0.0	0.0	7.15	378.09
13	0.0	0.0	0.49	205.31
16	0.0	0.0	0.00	315.91
17	0.0	0.0	0.00	97.54
21	0.0	0.0	0.00	97.04
100.00	may 7 las	og mau O std og	+2+ may 6 s+d as	+2+ may 7 \
8 13	_mou_7 loc_ 288.18 53.48 215.64 0.00 94.09	_og_mou_8	t2t_mou_6 std_og_ 116.56 446.41 7.89 1146.91 97.93	t2t_mou_7 \ 133.43 85.98 2.58 0.81 125.94
std_og std og t2m	_t2t_mou_8	std_og_t2m_mou_6	std_og_t2m_mou_7	
8	22.58	13.69	10.04	
75.69 13	498.23	255.36	52.94	
156.94 16	3.23	22.99	64.51	
18.29				
17 0.00	0.00	1.55	0.00	
21 876.99	61.79	665.36	808.74	
2,0.55				

	_og_t2f_mou_6	std_og_t2f_mc	ou_7 std_c	og_t2f_mou_8	
8	t2c_mou_6 \ 0.0		0.0	0.0	
0.0 13	0.0		0.0	0.0	
0.0 16	0.0		0.0	0.0	
0.0 17	0.0		0.0	0.0	
0.0 21	0.0		0.0	0.0	
0.0					
std_ 8 13 16 17 21	_og_t2c_mou_7 0.0 0.0 0.0 0.0 0.0	std_og_t2c_m	ou_8 std_c 0.0 0.0 0.0 0.0 0.0	og_mou_6 std_o 130.26 701.78 30.89 1148.46 763.29	g_mou_7 \ 143.48 138.93 67.09 0.81 934.69
	_og_mou_8 isd _. nou 6 \	_og_mou_6 iso	d_og_mou_7	isd_og_mou_8	
8 0.00	98.28	0.0	0.0	0.00	
13 0.00	655.18	0.0	0.0	1.29	
16 0.00	21.53	0.0	0.0	0.00	
17 2.58	0.00	0.0	0.0	0.00	
21	938.79	0.0	0.0	0.00	
	_og_mou_7 spl	_og_mou_8 og_	_others_6	og_others_7 o	g_others_8
8	0.00	10.23	0.00	0.0	0.0
13	0.00	4.78	0.00	0.0	0.0
16	3.26	5.91	0.00	0.0	0.0
17	0.00	0.00	0.93	0.0	0.0
21	0.00	0.00	0.00	0.0	0.0
		otal_og_mou_7	total_og_	_mou_8	
8	t2t_mou_6 \ 508.36	431.66	1	171.56	23.84

13	907.09	192.41	1015.26	67.88
16	346.81	286.01	233.38	41.33
17	1249.53	0.81	0.00	34.54
21	860.34	1028.79	1068.54	2.48
	c_t2t_mou_7 m_mou_7 \	loc_ic_t2t_mou_8	loc_ic_t2m_mou_6 57.58 142.88 226.81 47.41 118.23	
	c_t2m_mou_8 f_mou_8 \	loc_ic_t2f_mou_6	loc_ic_t2f_mou_7	
8 13 16 17 21 std_i	c_mou_6 loc_ 81.43 215.58 276.86 81.96 125.33 c_t2t_mou_7 m_mou_7 \ 0.58	26.11 2 229.83 2 2.31 87.68 1	15.79 55.26 11.78 0.00 59.11	_mou_6 \ 0.00 115.68 68.79 8.63 14.06
16	78.64	6.33	18.68	

73.08							
17	0.00		0.00		1.28		
0.00							
21	5.98		0.18		67.69		
38.23							
			_				
		std_ic_t2f_	mou_6	std_ic_	_t2f_mou_7		
std_ic_t2f							
8	0.65		0.00		0.0		
0.00							
13	317.91		0.00		0.0		
1.91							
16	73.93		0.51		0.0		
2.18							
17	0.00		0.00		0.0		
0.00							
21	101.74		0.00		0.0		
0.00							
std ic	t2o mou 6	std_ic_t2o_	mou 7	std ic	t2o mou 8	std i	c mou 6
			_			_	
8	0.0		0.0		0.0		22.43
13	0.0		0.0		0.0		423.81
16	0.0		0.0		0.0		87.99
17	0.0		0.0		0.0		9.91
21	0.0		0.0		0.0		81.76
		_ic_mou_8 t	otal_i				\
8	4.66	0.75		103.86		28.49	
13	68.09	474.41		968.61	1	72.58	
16	151.73	82.44		364.86	3	81.56	
17	0.00	0.00		91.88		2.31	
21	44.21	101.93		207.09	1	31.89	
total_	ic_mou_8 s	pl_ic_mou_6	spl_i	c_mou_7	spl_ic_mo	u_8	
isd_ic_mou	_6 \						
8	16.54	0.00		0.0		0.0	
0.00							
13	1144.53	0.45		0.0		0.0	
245.28							
16	294.46	0.00		0.0		0.0	
0.00							
17	0.00	0.00		0.0		0.0	
0.00							
21	261.04	0.00		0.0		0.0	

0.0	0							
	isd_ic_mou_7	isd_ic_	mou_8	ic_othe	rs_6	ic_others_	_7 ic_c	thers_8
8	0.00		0.00		0.00	0.0	90	0.00
13	62.11	3	93.39	8	3.48	16.2	24	21.44
16	0.00		0.23		0.00	0.0	90	0.00
17	0.00		0.00		0.00	0.0	90	0.00
21	0.00		0.00		0.00	0.0	00	0.00
8 437 13 507 16 570 17 816 21 600 \ 8	total_rech_am	19 6 10 19 22		21 4 6 2 26 120 717 160		rech_amt_6 90 110 110	14 11 2 4 27	ech_amt_7 154 110 110 0
17				30		110		
21		680		718		50		50
8 13 16 17 21	13 13	0 0 0 0	day_rch	n_amt_6 50 110 100 30 30	last _.	_day_rch_an	nt_7 \ 0 50 100 0 20	
vol	last_day_rch_ _3g_mb_6 \	amt_8 v	ol_2g_m	nb_6 vo		mb_7 vol_2 56.0	2g_mb_8 0.03	

0.0					
13		Θ	0.0	0.0	0.02
0.0 16		130	0.0	0.0	0.00
0.0 17		0	0.0	0.0	0.00
0.0 21		50	0.0	0.0	0.00
0.0		30	0.0	0.0	0.00
\	vol_3g_mb_7	vol_3g_mb_8	monthly_2g_6	monthly_2g_7	monthly_2g_8
8	750.95	11.94	Θ	1	. 0
13	0.00	0.00	0	e	0
16	0.00	0.00	Θ	6	0
17	0.00	0.00	0	6	0
21	0.00	0.00	Θ	6	0
	sachet_2g_6	sachet_2g_7	sachet_2g_8	monthly_3g_6	monthly_3g_7
8	0	1	3	Θ	0
13	0	0	3	0	0
16	0	0	0	0	0
17	0	0	0	0	Θ
21	Θ	0	0	0	Θ
	· ·		· ·	· ·	· ·
2110		sachet_3g_6	sachet_3g_7	sachet_3g_8	aon
8	_vbc_3g \ 6) (9 0	Θ	315
21.6 13	93 6) (9 0	Θ	2607
0.00 16) () (9 0	0	511
0.00)				
17 0.00			0		667
21 0.00	(e) ()	(9 0	0	720
8		jun_vbc_3g a 122.16	avg_rech_amt_6 519	_7 churn tota .0 0	al_mou_good \ 612.22
J	910.03	122.10	519	.0	014.44

13 16 17 21	0.00 2.45 0.00 0.00	0.00 21.89 0.00 0.00	380.0 459.0 408.0 640.0	0 0 0	1875.70 711.67 1341.41 1067.43
8 13 16 17 21	avg_mou_action 324.125 1262.390 597.705 1.560 1245.130		decrease_mou_acti	on 1 1 1 1 0	

Deriving new column decrease_rech_num_action

This column indicates whether the number of recharge of the customer has decreased in the action phase than the good phase.

```
# Avg rech number at action phase
data['avg_rech_num_action'] = (data['total_rech_num 7'] +
data['total_rech_num_8'])/2
# Difference total rech num 6 and avg rech action
data['diff rech num'] = data['avg rech num action'] -
data['total rech num 6']
# Checking if rech num has decreased in action phase
data['decrease rech num action'] = np.where((data['diff rech num'] <</pre>
0), 1, 0)
data.head()
   mobile number loc og t2o mou std og t2o mou loc ic t2o mou
arpu 6 \
      7001524846
                                              0.0
                                                              0.0
                             0.0
378.721
      7002191713
                             0.0
                                              0.0
                                                              0.0
13
492.846
16
      7000875565
                             0.0
                                              0.0
                                                              0.0
430.975
17
      7000187447
                             0.0
                                              0.0
                                                              0.0
690.008
21
      7002124215
                             0.0
                                              0.0
                                                              0.0
514.453
              arpu_8 onnet_mou_6 onnet_mou_7 onnet_mou_8
     arpu 7
offnet mou 6 \
    492.223 137.362
                           413.69
                                         351.03
                                                       35.08
94.66
13 205.671 593.260
                           501.76
                                         108.39
                                                      534.24
413.31
```

16 299.86 296.29	59 187.894	50.51	74.01	70.61	
17 18.98	30 25.499	1185.91	9.28	7.79	
61.64 21 597.75 757.93	637.760	102.41	132.11	85.14	
		net_mou_8 roam	_ic_mou_6 ro	am_ic_mou_7	
roam_ic_mc 8 0.00	ou_8 \ 80.63	136.48	0.00	0.00	
13	119.28	482.46	23.53	144.24	
72.11 16	229.74	162.76	0.00	2.83	
0.00 17	0.00	5.54	0.00	4.76	
4.81 21	896.68	983.39	0.00	0.00	
0.00					
roam_c 8 13 16 17 21	og_mou_6 ro 0.00 7.98 0.00 0.00 0.00	am_og_mou_7 roa 0.00 35.26 17.74 8.46 0.00	am_og_mou_8 0.00 1.44 0.00 13.34 0.00	loc_og_t2t_mou_6 297.13 49.63 42.61 38.99 4.48	
		loc_og_t2t_mou	_8 loc_og_t2	m_mou_6	
8	n_mou_7 \ 217.59	12.4	49	80.96	
70.58 13	6.19	36.0	91	151.13	
47.28 16	65.16	67.3	38	273.29	
145.99 17	0.00	0.0	90	58.54	
0.00 21	6.16	23.3	34	91.81	
87.93					
loc_og_t2f	g_t2m_mou_8 f_mou_8 \	loc_og_t2f_mou			
8 0.00	50.54	0.0	90	0.00	
13 23.51	294.46	4.	54	0.00	
16 10.26	128.28	0.0	90	4.48	
17 0.00	0.00	0.0	90	0.00	
0.00					

21 10 1.58	4.81		0.75		0.00	
loc_og_t2c_m	ou_6 ⁻	loc_og_t2c_	_mou_7	loc_og_t2d	c_mou_8	loc_og_mou_6
8	0.0		0.0		7.15	378.09
13	0.0		0.0		0.49	205.31
16	0.0		0.0		0.00	315.91
17	0.0		0.0		0.00	97.54
21	0.0		0.0		0.00	97.04
loc_og_mou_7 8 288.18 13 53.48 16 215.64 17 0.00 21 94.09		og_mou_8 63.04 353.99 205.93 0.00 129.74	std_og_	t2t_mou_6 116.56 446.41 7.89 1146.91 97.93	std_og_	t2t_mou_7 \ 133.43 85.98 2.58 0.81 125.94
		std_og_t2m __	_mou_6 13.69	std_og_t2n	m_mou_7 10.04	
75.69 13 49	8.23	2	255.36		52.94	
156.94 16 18.29	3.23		22.99		64.51	
17	0.00		1.55		0.00	
0.00 21 6 876.99	1.79	(665.36		808.74	
std_og_t2f_m std og t2c mou 6		std_og_t2f __	_mou_7	std_og_t2	f_mou_8	
8 0.0	0.0		0.0		0.0	
13	0.0		0.0		0.0	
0.0 16	0.0		0.0		0.0	
0.0 17	0.0		0.0		0.0	
0.0 21 0.0	0.0		0.0		0.0	
std_og_t2c_m	ou_7 :	std_og_t2c __	_mou_8	std_og_mou	u_6 std	_og_mou_7 \

8 13 16 17 21	0.0 0.0 0.0 0.0 0.0	0.0 0.0 0.0 0.0 0.0	130.26 701.78 30.89 1148.46 763.29	143.48 138.93 67.09 0.81 934.69
_std_og_mou_		isd_og_mou_7	isd_og_mou_	8
spl_og_mou_6 \ 8 98.2 0.00		0.0	0.0	9
13 655.1	8 0.0	0.0	1.2	9
0.00 16 21.5	0.0	0.0	0.0	9
0.00 17 0.0	0.0	0.0	0.0	9
2.58 21 938.7 0.00	9 0.0	0.0	0.0	9
	7 spl_og_mou_8	og_others_6	og_others_7	og_others_8
8 0.0	0 10.23	0.00	0.0	0.0
13 0.0	0 4.78	0.00	0.0	0.0
16 3.2	6 5.91	0.00	0.0	0.0
17 0.0	0.00	0.93	0.0	0.0
21 0.0	0.00	0.00	0.0	0.0
total_og_mo loc_ic_t2t_mou_	6 \	ou_7 total_og		
8 508			171.56	23.84
13 907			015.26	67.88
16 346			233.38	41.33
17 1249		0.81	0.00	34.54
21 860	.34 1028	8.79 1	068.54	2.48
loc_ic_t2m_mou_ 8 13.98	9.84	0.31	57.58	
13 18.53	7.58	52.58	142.88	

16	71.44	28.89	226.81	
149.69 17	0.00	0.00	47.41	
2.31				
21 74.63	10.19	19.54	118.23	
74.03				
	c_t2m_mou_8 lo	oc_ic_t2f_mou_6 loc_i	ic_t2f_mou_7	
8	15.48	0.00	0.00	
0.00		4 01	0.00	
13 7.49	195.18	4.81	0.00	
16	150.16	8.71	8.68	
32.71 17	0.00	0.00	0.00	
0.00				
21 10.39	129.16	4.61	2.84	
10.39				
loc_i 8	c_mou_6 loc_io 81.43	_mou_7 loc_ic_mou_8 23.83 15.79	std_ic_t2t_mou_6 \ 0.00	
13	215.58	26.11 255.26	115.68	
16	276.86	229.83 211.78	68.79	
			0.00	
17	81.96	2.31 0.00	8.63 14.06	
17 21	81.96 125.33	2.31 0.00 87.68 159.11	14.06	
17 21 std_i	81.96 125.33 c_t2t_mou_7 st	2.31 0.00	14.06	
17 21 std_i std_ic_t2 8	81.96 125.33	2.31 0.00 87.68 159.11	14.06	
17 21 std_i std_ic_t2 8 4.08	81.96 125.33 c_t2t_mou_7 st m_mou_7 \ 0.58	2.31 0.00 87.68 159.11 d_ic_t2t_mou_8 std_i 0.10	14.06 ic_t2m_mou_6 22.43	
17 21 std_i std_ic_t2 8 4.08 13 29.79	81.96 125.33 c_t2t_mou_7 st m_mou_7 \ 0.58 38.29	2.31 0.00 87.68 159.11 d_ic_t2t_mou_8 std_i 0.10 154.58	14.06 ic_t2m_mou_6 22.43 308.13	
17 21 std_ic_t2 8 4.08 13 29.79	81.96 125.33 c_t2t_mou_7 st m_mou_7 \ 0.58	2.31 0.00 87.68 159.11 d_ic_t2t_mou_8 std_i 0.10	14.06 ic_t2m_mou_6 22.43	
17 21 std_i std_ic_t2 8 4.08 13 29.79	81.96 125.33 c_t2t_mou_7 st m_mou_7 \ 0.58 38.29	2.31 0.00 87.68 159.11 d_ic_t2t_mou_8 std_i 0.10 154.58	14.06 ic_t2m_mou_6 22.43 308.13	
17 21 std_ic_t2 8 4.08 13 29.79 16 73.08 17 0.00	81.96 125.33 c_t2t_mou_7 st m_mou_7 \ 0.58 38.29 78.64 0.00	2.31 0.00 87.68 159.11 d_ic_t2t_mou_8 std_s 0.10 154.58 6.33 0.00	14.06 ic_t2m_mou_6 22.43 308.13 18.68 1.28	
17 21 std_ic_t2 8 4.08 13 29.79 16 73.08	81.96 125.33 c_t2t_mou_7 st m_mou_7 \ 0.58 38.29 78.64	2.31 0.00 87.68 159.11 d_ic_t2t_mou_8 std_i 0.10 154.58 6.33	14.06 ic_t2m_mou_6 22.43 308.13 18.68	
std_i std_ic_t2 std_ic_t2 8 4.08 13 29.79 16 73.08 17 0.00 21 38.23	81.96 125.33 c_t2t_mou_7 st m_mou_7 \ 0.58 38.29 78.64 0.00 5.98	2.31 0.00 87.68 159.11 d_ic_t2t_mou_8 std_i 0.10 154.58 6.33 0.00 0.18	14.06 ic_t2m_mou_6 22.43 308.13 18.68 1.28 67.69	
17 21 std_ic_t2 8 4.08 13 29.79 16 73.08 17 0.00 21 38.23 std_i	81.96 125.33 c_t2t_mou_7 st m_mou_7 \ 0.58 38.29 78.64 0.00 5.98 c_t2m_mou_8 st	2.31 0.00 87.68 159.11 d_ic_t2t_mou_8 std_i 0.10 154.58 6.33 0.00 0.18	14.06 ic_t2m_mou_6 22.43 308.13 18.68 1.28	
17 21 std_ic_t2 8 4.08 13 29.79 16 73.08 17 0.00 21 38.23 std_ic_t2 8	81.96 125.33 c_t2t_mou_7 st m_mou_7 \ 0.58 38.29 78.64 0.00 5.98	2.31 0.00 87.68 159.11 d_ic_t2t_mou_8 std_i 0.10 154.58 6.33 0.00 0.18	14.06 ic_t2m_mou_6 22.43 308.13 18.68 1.28 67.69	
17 21 std_ic_t2 8 4.08 13 29.79 16 73.08 17 0.00 21 38.23 std_ic_t2 8 0.00	81.96 125.33 c_t2t_mou_7 st m_mou_7 \ 0.58 38.29 78.64 0.00 5.98 c_t2m_mou_8 st f_mou_8 \ 0.65	2.31 0.00 87.68 159.11 d_ic_t2t_mou_8 std_i 0.10 154.58 6.33 0.00 0.18 d_ic_t2f_mou_6 std_i 0.00	14.06 ic_t2m_mou_6 22.43 308.13 18.68 1.28 67.69 ic_t2f_mou_7 0.0	
std_i std_ic_t2 std_ic_t2 8 4.08 13 29.79 16 73.08 17 0.00 21 38.23 std_ic_t2 8 0.00 13 1.91	81.96 125.33 c_t2t_mou_7 st m_mou_7 \ 0.58 38.29 78.64 0.00 5.98 c_t2m_mou_8 st f_mou_8 \ 0.65 317.91	2.31 0.00 87.68 159.11 d_ic_t2t_mou_8 std_i 0.10 154.58 6.33 0.00 0.18 d_ic_t2f_mou_6 std_i 0.00 0.00	14.06 ic_t2m_mou_6 22.43 308.13 18.68 1.28 67.69 ic_t2f_mou_7 0.0 0.0	
17 21 std_i std_ic_t2 8 4.08 13 29.79 16 73.08 17 0.00 21 38.23 std_i std_ic_t2 8 0.00 13 1.91 16	81.96 125.33 c_t2t_mou_7 st m_mou_7 \ 0.58 38.29 78.64 0.00 5.98 c_t2m_mou_8 st f_mou_8 \ 0.65	2.31 0.00 87.68 159.11 d_ic_t2t_mou_8 std_i 0.10 154.58 6.33 0.00 0.18 d_ic_t2f_mou_6 std_i 0.00	14.06 ic_t2m_mou_6 22.43 308.13 18.68 1.28 67.69 ic_t2f_mou_7 0.0	
std_i std_ic_t2 std_ic_t2 8 4.08 13 29.79 16 73.08 17 0.00 21 38.23 std_ic_t2 8 0.00 13 1.91	81.96 125.33 c_t2t_mou_7 st m_mou_7 \ 0.58 38.29 78.64 0.00 5.98 c_t2m_mou_8 st f_mou_8 \ 0.65 317.91	2.31 0.00 87.68 159.11 d_ic_t2t_mou_8 std_i 0.10 154.58 6.33 0.00 0.18 d_ic_t2f_mou_6 std_i 0.00 0.00	14.06 ic_t2m_mou_6 22.43 308.13 18.68 1.28 67.69 ic_t2f_mou_7 0.0 0.0	

21 0.00	101.74	0	. 00	0.0	
	ic_t2o_mou_6	std ic t2o mou	u 7 std ic	t2o mou 8	std ic mou 6
8	0.0		 0.0	0.0	22.43
13	0.0	(0.0	0.0	423.81
16	0.0	(9.0	0.0	87.99
17	0.0	(0.0	0.0	9.91
21	0.0	(0.0	0.0	81.76
std_: 8 13 16 17 21	ic_mou_7 std_ 4.66 68.09 151.73 0.00 44.21	ic_mou_8 tota 0.75 474.41 82.44 0.00 101.93	al_ic_mou_6 103.86 968.61 364.86 91.88 207.09	<u>-</u> 17 38	nou_7 \ 28.49 72.58 31.56 2.31 31.89
tota isd_ic_mo 8 0.00		l_ic_mou_6 s _l 0.00	pl_ic_mou_7 0.0	· 	u_8 0.0
13 245.28	1144.53	0.45	0.0	(0.0
16 0.00	294.46	0.00	0.0	(0.0
17	0.00	0.00	0.0	(0.0
0.00 21 0.00	261.04	0.00	0.0	(0.0
_	ic_mou_7 isd_	ic_mou_8 ic_o	others_6 i	c_others_7	ic_others_8
8	0.00	0.00	0.00	0.00	0.00
13	62.11	393.39	83.48	16.24	21.44
16	0.00	0.23	0.00	0.00	0.00
17	0.00	0.00	0.00	0.00	0.00
21	0.00	0.00	0.00	0.00	0.00
tota total_re		total_rech_num	n_7 total_	rech_num_8	

8 437	19	9	21	1	.4
13		6	4	1	1
507 16	10	9	6		2
570 17	19	9	2		4
816 21	22	2	26	2	77
600					
\	total_rech_amt_	7 total_r	ech_amt_8	max_rech_amt_6	max_rech_amt_7
8	60:	1	120	90	154
13	253	3	717	110	110
16	348	8	160	110	110
17	(9	30	110	Θ
21	680	9	718	50	50
8 13 16 17 21	max_rech_amt_8 30 130 130 30 50	last_day_	rch_amt_6 50 110 100 30 30	last_day_rch_am	ot_7 \ 0 50 100 0 20
vol	last_day_rch_am 3g mb 6 \	t_8 vol_2	g_mb_6 vol	2g_mb_7 vol_2	g_mb_8
8	_39	10	0.0	356.0	0.03
13		0	0.0	0.0	0.02
16		130	0.0	0.0	0.00
0.0		0	0.0	0.0	0.00
0.0 21 0.0		50	0.0	0.0	0.00
	vol_3g_mb_7 vo	l_3g_mb_8	monthly_2g	y_6 monthly_2g_	7 monthly_2g_8
8	750.95	11.94		0	1 0
13	0.00	0.00		0	0 0

16	0.00	0.00	0		9 0
17	0.00	0.00	0		9 0
21	0.00	0.00	0		9 0
	sachet 2g 6	sachet 2g 7	sachet 2g 8	monthly 3g 6	monthly_3g_7
8	0	1	3		- 0
13	0	0	3	0	0
16	0	0	0	0	Θ
17	0	0	0	0	0
21	0	0	0	0	Θ
	monthly_3g_8 vbc 3g \	sachet_3g_6	sachet_3g_7	sachet_3g_8	aon
8 21.0	0	0	0	Θ	315
13	0	0	0	0	2607
0.00 16	0	0	0	0	511
0.00 17	0	0	0	0	667
0.00 21	Θ	Θ	0	0	720
0.00					
8 13 16 17 21	jul_vbc_3g j 910.65 0.00 2.45 0.00 0.00	122.16	519.	0 0 0 0 0 0 0 0	al_mou_good \ 612.22 1875.70 711.67 1341.41 1067.43
	avg_mou_actio	on diff_mou	decrease_mou_	action avg_	rech_num_action
8	324.12	25 -288.095		1	17.5
13	1262.39	00 -613.310		1	7.5
16	597.70)5 -113.965		1	4.0
17	1.56	60 -1339.850		1	3.0
21	1245.13	30 177.700		0	26.5

		<pre>decrease_rech_num_action</pre>
8	-1.5	1
13	1.5	Θ
16	-6.0	1
17	-16.0	1
21	4.5	0

Deriving new column decrease_rech_amt_action

This column indicates whether the amount of recharge of the customer has decreased in the action phase than the good phase.

```
# Avg rech_amt in action phase
data['avg rech amt action'] = (data['total rech amt 7'] +
data['total rech amt 8'])/2
# Difference of action phase rech amt and good phase rech amt
data['diff rech amt'] = data['avg rech amt action'] -
data['total_rech_amt_6']
# Checking if rech amt has decreased in action phase
data['decrease rech amt action'] = np.where((data['diff rech amt'] <</pre>
0), 1, 0)
data.head()
   mobile number loc og t2o mou std og t2o mou loc ic t2o mou
arpu_6 \
      7001524846
                              0.0
                                               0.0
                                                               0.0
378.721
      7002191713
                              0.0
                                              0.0
                                                               0.0
13
492.846
                                                               0.0
      7000875565
                              0.0
                                               0.0
16
430.975
                              0.0
                                               0.0
                                                               0.0
17
      7000187447
690,008
                              0.0
                                              0.0
                                                               0.0
21
     7002124215
514.453
                       onnet mou 6
                                    onnet mou 7 onnet mou 8
     arpu 7
              arpu 8
offnet mou 6
8 	 49\overline{2.223} 	 137.362
                            413.69
                                         351.03
                                                        35.08
94.66
13 205.671 593.260
                            501.76
                                         108.39
                                                       534.24
413.31
16 299.869 187.894
                             50.51
                                          74.01
                                                        70.61
296.29
     18.980
              25.499
                           1185.91
                                                         7.79
17
                                           9.28
```

61.64 21 597.7 757.93	53 637.760	102.41	132.11	85.14	ļ
		net_mou_8 roa	am_ic_mou_6	roam_ic_mou_	7
roam_ic_m	80.63	136.48	0.00	0.0	00
0.00 13	119.28	482.46	23.53	144.2	24
72.11 16	229.74	162.76	0.00	2.8	33
0.00 17	0.00	5.54	0.00	4.7	′ 6
4.81 21 0.00	896.68	983.39	0.00	0.6	00
roam_ 8 13 16 17 21	og_mou_6 roa 0.00 7.98 0.00 0.00 0.00	am_og_mou_7 r 0.00 35.26 17.74 8.46 0.00	roam_og_mou_8 0.00 1.44 0.00 13.34 0.00	 - -	297.13 49.63 42.61 38.99 4.48
	g_t2t_mou_7 m mou 7 \	loc_og_t2t_mc	ou_8 loc_og_	t2m_mou_6	
8 70.58	217.59	12	2.49	80.96	
13	6.19	36	5.01	151.13	
47.28 16	65.16	67	7.38	273.29	
145.99 17	0.00	6	0.00	58.54	
0.00 21	6.16	23	3.34	91.81	
87.93					
loc_og_t2	g_t2m_mou_8 f_mou_8 \				
8 0.00	50.54		0.00	0.00	
13 23.51	294.46	4	1.54	0.00	
16 10.26	128.28	0	0.00	4.48	
17	0.00	6	0.00	0.00	
0.00 21	104.81	6	0.75	0.00	
1.58					

	_	_	_	
\	loc_og_t2c_mou_6	loc_og_t2c_mou_7	loc_og_t2c_mou_8	loc_og_mou_6
8	0.0	0.0	7.15	378.09
13	0.0	0.0	0.49	205.31
16	0.0	0.0	0.00	315.91
17	0.0	0.0	0.00	97.54
21	0.0	0.0	0.00	97.04
8 13 16 17 21	loc_og_mou_7 loc 288.18 53.48 215.64 0.00 94.09	_og_mou_8 std_og_ 63.04 353.99 205.93 0.00 129.74	t2t_mou_6 std_og_ 116.56 446.41 7.89 1146.91 97.93	t2t_mou_7 \ 133.43 85.98 2.58 0.81 125.94
	std_og_t2t_mou_8	std_og_t2m_mou_6	std_og_t2m_mou_7	
std_ 8	_og_t2m_mou_8 \ 22.58	13.69	10.04	
75.6 13	69 498.23	255.36	52.94	
156	. 94			
16 18.2	3.23	22.99	64.51	
17	0.00	1.55	0.00	
0.00	61.79	665.36	808.74	
876	. 99			
c+d	std_og_t2f_mou_6 _og_t2c_mou_6 \	std_og_t2f_mou_7	std_og_t2f_mou_8	
8	0.0	0.0	0.0	
0.0	0.0	0.0	0.0	
0.0				
16 0.0	0.0	0.0	0.0	
17	0.0	0.0	0.0	
0.0 21	0.0	0.0	0.0	
0.0				
8 13 16	std_og_t2c_mou_7 0.0 0.0 0.0	std_og_t2c_mou_8 0.0 0.0 0.0	std_og_mou_6 std 130.26 701.78 30.89	_og_mou_7 \ 143.48 138.93 67.09

17 21	0.0 0.0		0.0 0.0	1148.46 763.29	0.81 934.69
	_og_mou_8 is mou_6 \ 98.28	d_og_mou_6 i 0.0	.sd_og_mou_7 0.0	isd_og_mou_8	
0.00 13	655.18	0.0	0.0	1.29	
0.00 16 0.00	21.53	0.0	0.0	0.00	
17 2.58 21	0.00 938.79	0.0	0.0	0.00	
_	_og_mou_7 sp	l_og_mou_8 o	og_others_6	og_others_7	og_others_8
8	0.00	10.23	0.00	0.0	0.0
13	0.00	4.78	0.00	0.0	0.0
16	3.26	5.91	0.00	0.0	0.0
17	0.00	0.00	0.93	0.0	0.0
21	0.00	0.00	0.00	0.0	0.0
	al_og_mou_6 t2t_mou_6 \ 508.36	total_og_mou_ 431.6		_mou_8 171.56	23.84
13	907.09	192.4		015.26	67.88
16	346.81	286.0		233.38	41.33
17	1249.53	0.8	31	0.00	34.54
21	860.34	1028.7	79 10	068.54	2.48
loc_ic_ 8 13.98	_ic_t2t_mou_7 t2m_mou_7 \ 9.84		0.31	ic_t2m_mou_6 57.58	
13 18.53	7.58		52.58	142.88	
16 149.69 17	71.44		0.00	226.81 47.41	
_,	0.00		3.00	1,111	

2.31			
21	10.19	19.54	118.23
74.63			
	c_t2m_mou_8 f mou 8 \	loc_ic_t2f_mou_6	loc_ic_t2f_mou_7
8	15.48	0.00	0.00
0.00 13	195.18	4.81	0.00
7.49	133.10	1101	0.00
16	150.16	8.71	8.68
32.71 17	0.00	0.00	0.00
0.00	0.00	0.00	0.00
21	129.16	4.61	2.84
10.39			
loc i	c mou 6 loc	ic mou 7 loc ic m	nou 8 std ic t2t mou 6 \
8	81.43		.5.79 0.00
13	215.58		55.26 115.68
16	276.86		1.78 68.79
17	81.96		0.00 8.63
21	125.33		59.11 14.06
		std_ic_t2t_mou_8	std_1c_t2m_mou_6
8	m_mou_7 \ 0.58	0.10	22.43
4.08			
13	38.29	154.58	308.13
29.79 16	78.64	6.33	18.68
73.08	70.04	0.55	10.00
17	0.00	0.00	1.28
0.00			
21	5.98	0.18	67.69
38.23			
std i	c t2m mou 8	std ic t2f mou 6	std ic t2f mou 7
	f_mou_8 \	3td_1tt21_iii0d_0	3td_1tt21_mod_7
8	0.65	0.00	0.0
0.00			
13	317.91	0.00	0.0
1.91	72.02	0.51	0.0
16 2.18	73.93	0.51	0.0
17	0.00	0.00	0.0
0.00	0.00	0.00	3.0
21	101.74	0.00	0.0
0.00			

	.c_t2o_mou_6	std_ic_t2o_mc	ou_7 std_i	ic_t2o_mou_8	std_ic_mou_6
8	0.0		0.0	0.0	22.43
13	0.0		0.0	0.0	423.81
16	0.0		0.0	0.0	87.99
17	0.0		0.0	0.0	9.91
21	0.0		0.0	0.0	81.76
std_i 8 13 16 17 21	.c_mou_7 std_ 4.66 68.09 151.73 0.00 44.21	ic_mou_8 tot 0.75 474.41 82.44 0.00 101.93	al_ic_mou_ 103.8 968.6 364.8 91.8 207.6	51 1 36 3 38	mou_7 \ 28.49 72.58 81.56 2.31 31.89
total isd_ic_mo 8	ic_mou_8 sp ou_6 \ 16.54	ol_ic_mou_6 s 0.00	spl_ic_mou_ 0.		u_8 0.0
0.00 13	1144.53	0.45	0.		0.0
245.28 16 0.00	294.46	0.00	0.	. 0	0.0
17	0.00	0.00	0.	. 0	0.0
0.00 21 0.00	261.04	0.00	0.	. 0	0.0
isd_i	.c_mou_7 isd_	_ic_mou_8 ic_	_others_6	ic_others_7	ic_others_8
8	0.00	0.00	0.00	0.00	0.00
13	62.11	393.39	83.48	16.24	21.44
16	0.00	0.23	0.00	0.00	0.00
17	0.00	0.00	0.00	0.00	0.00
21	0.00	0.00	0.00	0.00	0.00
total_red		total_rech_nu	_		
8 437	19		21	14	
13	6		4	11	

F 0.7					
507 16		10	6		2
570		10	2		1
17 816		19	2	2	1
21		22	26	27	7
600					
\	total_rech_amt	_7 total_r	ech_amt_8 m	ax_rech_amt_6	max_rech_amt_7
8	60	91	120	90	154
13	2!	53	717	110	110
16	34	48	160	110	110
17		0	30	110	0
21	68	30	718	50	50
	max_rech_amt_8	last_day_		ast_day_rch_amt	_7 \
8 13	30 130		50 110		0 50
16	130		100	-	L00
17 21	30 50		30 30		0 20
		m+ 0 vol 2		2g mb 7 vol 2g	
vol	_3g_mb_6 \		g_iiib_b vot_	2g_mb_7 vol_2	J_IIIU_0
8 0.0		10	0.0	356.0	0.03
13		0	0.0	0.0	0.02
0.0 16		130	0.0	0.0	0.00
0.0					
17 0.0		0	0.0	0.0	0.00
21 0.0		50	0.0	0.0	0.00
0.0					
\	vol_3g_mb_7 vol_	ol_3g_mb_8	monthly_2g_	6 monthly_2g_7	7 monthly_2g_8
8	750.95	11.94		0	L 0
13	0.00	0.00		0 0	0
16	0.00	0.00		0 (9
17					

21	0.00	0.00	Θ	0	Θ
sache	et_2g_6 sa	chet_2g_7 sac	het_2g_8 monthl	y_3g_6 month	Ly_3g_7
8	0	1	3	0	0
13	0	0	3	0	0
16	0	0	0	0	0
17	0	0	0	0	0
21	0	0	0	0	0
month aug_vbc_3		achet_3g_6 sa	chet_3g_7 sache	et_3g_8 aon	
8 21.03	0	0	0	0 315	
13	0	0	0	0 2607	
0.00 16	0	0	0	0 511	
0.00 17	0	0	Θ	0 667	
0.00 21	0	0	0	0 720	
0.00	O .	U	U	0 720	
	/bc_3g jun 910.65 0.00 2.45 0.00 0.00	_vbc_3g avg_r 122.16 0.00 21.89 0.00 0.00	ech_amt_6_7 chur 519.0 380.0 459.0 408.0 640.0	0 612 0 1875	2.22 5.70 1.67 1.41
	nou_action	diff_mou dec	rease_mou_action	n avg_rech_nur	m_action
8	324.125	-288.095	1	L	17.5
13	1262.390	-613.310	1	L	7.5
16	597.705	-113.965]	L	4.0
17	1.560	-1339.850	1		3.0
21	1245.130	177.700	()	26.5
			num_action avg_		

8	-1.5	1	360.5	
13	1.5	0	485.0	
16	-6.0	1	254.0	
17	-16.0	1	15.0	
21	4.5	0	699.0	
8 13 16 17 21	diff_rech_amt -76.5 -22.0 -316.0 -801.0 99.0	decrease_rech_amt_action 1 1 1 1 0		

Deriving new column decrease_arpu_action

This column indicates whether the average revenue per customer has decreased in the action phase than the good phase.

```
# ARUP in action phase
data['avg_arpu_action'] = (data['arpu_7'] + data['arpu_8'])/2
# Difference of good and action phase ARPU
data['diff arpu'] = data['avg arpu action'] - data['arpu 6']
# Checking whether the arpu has decreased on the action month
data['decrease arpu action'] = np.where(data['diff arpu'] < 0, 1, 0)</pre>
data.head()
   mobile number loc og t2o mou std og t2o mou loc ic t2o mou
arpu 6 \
      7001524846
                             0.0
                                              0.0
                                                              0.0
8
378.721
      7002191713
                             0.0
                                              0.0
                                                              0.0
492.846
16
      7000875565
                             0.0
                                              0.0
                                                              0.0
430.975
17
     7000187447
                             0.0
                                              0.0
                                                              0.0
690,008
                             0.0
                                                              0.0
21
     7002124215
                                              0.0
514,453
              arpu 8 onnet mou 6 onnet mou 7 onnet mou 8
     arpu 7
offnet mou 6
   492.223 137.362
                           413.69
                                         351.03
                                                       35.08
94.66
                           501.76
13 205.671 593.260
                                         108.39
                                                      534.24
413.31
16 299.869
            187.894
                            50.51
                                          74.01
                                                       70.61
296.29
```

17 18.980	25.499	1185.91	9.28	7.79	
61.64	. 627 760	102 41	122 11	OF 14	
757.93	3 637.760	102.41	132.11	85.14	
737.93					
offnet	mou 7 off	net mou 8 roam	n ic mou 6	roam ic mou 7	
roam_ic_mo					
8	80.63	136.48	0.00	0.00	
0.00	110 00	400 46	22.52	144.04	
13 72.11	119.28	482.46	23.53	144.24	
	229.74	162.76	0.00	2.83	
0.00	229.74	102.70	0.00	2.03	
17	0.00	5.54	0.00	4.76	
4.81					
	396.68	983.39	0.00	0.00	
0.00					
		7	0	1	C
8 roam_og	g_mou_6 ro 0.00	am_og_mou_7 rc 0.00	oam_og_mou_8 0.00		97.13
13	7.98	35.26	1.44		49.63
16	0.00	17.74	0.00		42.61
17	0.00	8.46	13.34		38.99
21	0.00	0.00	0.00		4.48
1	121 7	1	0 1	12	
loc_og_ loc og t2m		loc_og_t2t_mou	ı_8 toc_og_	t2m_mou_6	
8	217.59	12.	40	80.96	
70.58	217.55	12 1	73	00.50	
13	6.19	36.	01	151.13	
47.28					
16	65.16	67.	38	273.29	
145.99	0.00	0	0.0	FO F4	
17	0.00	0.	00	58.54	
0.00 21	6.16	23.	3/1	91.81	
87.93	0.10	25.	34	31.01	
07.135					
loc_og_	_t2m_mou_8	loc_og_t2f_mou	_6 loc_og_ [.]	t2f_mou_7	
loc_og_t2f					
8	50.54	0.	00	0.00	
0.00	204 46	1	Γ 4	0.00	
13 23.51	294.46	4.	54	0.00	
16	128.28	О	.00	4.48	
10.26	120120	0.			
17	0.00	0.	00	0.00	
0.00					
21	104.81	0.	75	0.00	
1.58					

loc_og_t2c_mou_0	loc_og_t2c_mou_7	loc_og_t2c_mou_8	loc_og_mou_6
8 0.0	0.0	7.15	378.09
13 0.0	0.0	0.49	205.31
16 0.0	0.0	0.00	315.91
17 0.0	0.0	0.00	97.54
21 0.0	0.0	0.00	97.04
	2		
loc_og_mou_7 loc_os_mou_7 loc_o	oc_og_mou_8 std_og_ 63.04 353.99 205.93 0.00 129.74	_t2t_mou_6 std_og_ 116.56 446.41 7.89 1146.91 97.93	t2t_mou_7 \ 133.43 85.98 2.58 0.81 125.94
std_og_t2t_mou_8		std_og_t2m_mou_7	
std_og_t2m_mou_8 \ 8		10.04	
75.69 13 498.23	255.36	52.94	
156.94 16 3.23	22.99	64.51	
18.29 17 0.00	1.55	0.00	
0.00 21 61.79	665.36	808.74	
876.99			
<pre>std_og_t2f_mou_6 std_og_t2c_mou_6 \</pre>	std_og_t2f_mou_7	std_og_t2f_mou_8	
8 0.0	0.0	0.0	
13 0.0 0.0	0.0	0.0	
16 0.0	0.0	0.0	
0.0 17 0.0	0.0	0.0	
0.0 21 0.0	0.0	0.0	
	/ c+d on +2 0	a+d aa ma 6	l og mou 7
std_og_t2c_mou_3 8 0.0 13 0.0	0.0	std_og_mou_6 std 130.26 701.78	l_og_mou_7 \ 143.48 138.93

16 17 21		0.0 0.0 0.0	e).0).0).0	30.89 1148.46 763.29		67.09 0.81 934.69
		isd_og_mou	_6 isd_	og_mou_7	isd_og_m	ou_8	
spl_og_mou 8	98.28	0	.0	0.0		0.00	
0.00 13	655.18	0	.0	0.0		1.29	
0.00 16	21.53	0	.0	0.0		0.00	
0.00 17	0.00	0	.0	0.0		0.00	
2.58 21 0.00	938.79	0	.0	0.0		0.00	
	n mou 7	spl_og_mou	9 00 0	thors 6	og othors	7 00	othors 9
\				_	_	_	
8	0.00	10.		0.00		0.0	0.0
13	0.00		78	0.00		0.0	0.0
16	3.26	5.	91	0.00	6	0.0	0.0
17	0.00	0.	00	0.93	e	0.0	0.0
21	0.00	0.	00	0.00	6	0.0	0.0
total	oa mou	6 total og	mou 7	total og	mou 8		
loc_ic_t2t		_	431.66		 171.56		23.84
13	907.0		192.41		915.26		67.88
16	346.8		286.01	•	233.38		41.33
17	1249.5	3	0.81		0.00		34.54
21	860.3	4 1	.028.79	10	968.54		2.48
loc_ic loc_ic_t2r 8				_8 loc_: 31	ic_t2m_mou 57.	_	
13.98							
13 18.53		.58	52.		142.		
16 149.69	71	. 44	28.	89	226.	81	

17	0.00	0.00	47.41
2.31 21	10.19	19.54	118.23
74.63			
loc_i	.c_t2m_mou_8 l	oc_ic_t2f_mou_6 loc_	ic_t2f_mou_7
loc_ic_t2	f_mou_8 \		
8 0.00	15.48	0.00	0.00
13	195.18	4.81	0.00
7.49 16	150.16	8.71	8.68
32.71			
17 0.00	0.00	0.00	0.00
21	129.16	4.61	2.84
10.39			
loc_i 8 13 16 17 21	.c_mou_6 loc_i 81.43 215.58 276.86 81.96 125.33	c_mou_7 loc_ic_mou_8 23.83 15.79 26.11 255.26 229.83 211.78 2.31 0.00 87.68 159.11	$ \begin{array}{cccccccccccccccccccccccccccccccccccc$
		td_ic_t2t_mou_8 std_	ic_t2m_mou_6
std_ic_t2 8	c_t2t_mou_7 s m_mou_7 \ 0.58	td_ic_t2t_mou_8 std_ 0.10	ic_t2m_mou_6 22.43
std_ic_t2 8 4.08 13	m_mou_7 \		
std_ic_t2 8 4.08 13 29.79	m_mou_7 \ 0.58 38.29	0.10 154.58	22.43
std_ic_t2 8 4.08 13 29.79 16 73.08	m_mou_7 \ 0.58 38.29 78.64	0.10	22.43
std_ic_t2 8 4.08 13 29.79 16 73.08	m_mou_7 \ 0.58 38.29	0.10 154.58	22.43
std_ic_t2 8 4.08 13 29.79 16 73.08 17 0.00	m_mou_7 \ 0.58 38.29 78.64	0.10 154.58 6.33	22.43 308.13 18.68
std_ic_t2 8 4.08 13 29.79 16 73.08 17 0.00	m_mou_7 \ 0.58 38.29 78.64 0.00	0.10 154.58 6.33 0.00	22.43 308.13 18.68 1.28
std_ic_t2 8 4.08 13 29.79 16 73.08 17 0.00 21 38.23	m_mou_7 \	0.10 154.58 6.33 0.00 0.18	22.43 308.13 18.68 1.28
std_ic_t2 8 4.08 13 29.79 16 73.08 17 0.00 21 38.23 std_ic_t2 8	m_mou_7 \ 0.58 38.29 78.64 0.00 5.98	0.10 154.58 6.33 0.00 0.18	22.43 308.13 18.68 1.28 67.69
std_ic_t2 8 4.08 13 29.79 16 73.08 17 0.00 21 38.23 std_ic_t2 8 0.00 13	m_mou_7 \	0.10 154.58 6.33 0.00 0.18 td_ic_t2f_mou_6 std_	22.43 308.13 18.68 1.28 67.69 ic_t2f_mou_7
std_ic_t2 8 4.08 13 29.79 16 73.08 17 0.00 21 38.23 std_ic_t2 8 0.00 13 1.91 16	m_mou_7 \	0.10 154.58 6.33 0.00 0.18 td_ic_t2f_mou_6 std_ 0.00	22.43 308.13 18.68 1.28 67.69 ic_t2f_mou_7 0.0
std_ic_t2 8 4.08 13 29.79 16 73.08 17 0.00 21 38.23 std_ic_t2 8 0.00 13 1.91 16 2.18	m_mou_7 \	0.10 154.58 6.33 0.00 0.18 td_ic_t2f_mou_6 std_ 0.00 0.00	22.43 308.13 18.68 1.28 67.69 ic_t2f_mou_7 0.0 0.0 0.0
std_ic_t2 8 4.08 13 29.79 16 73.08 17 0.00 21 38.23 std_ic_t2 8 0.00 13 1.91 16 2.18 17 0.00	m_mou_7 \	0.10 154.58 6.33 0.00 0.18 td_ic_t2f_mou_6 std_ 0.00 0.00 0.51 0.00	22.43 308.13 18.68 1.28 67.69 ic_t2f_mou_7 0.0 0.0 0.0 0.0
std_ic_t2 8 4.08 13 29.79 16 73.08 17 0.00 21 38.23 std_ic_t2 8 0.00 13 1.91 16 2.18 17	m_mou_7 \	0.10 154.58 6.33 0.00 0.18 td_ic_t2f_mou_6 std_ 0.00 0.00	22.43 308.13 18.68 1.28 67.69 ic_t2f_mou_7 0.0 0.0 0.0

	_t2o_mou_6	std_ic_t2o_mou	_7 std_ic_	t2o_mou_8 sto	d_ic_mou_6
8	0.0	0	. 0	0.0	22.43
13	0.0	0	. 0	0.0	423.81
16	0.0	Θ	. 0	0.0	87.99
17	0.0	Θ	. 0	0.0	9.91
21	0.0	Θ	. 0	0.0	81.76
8 13	_mou_7 std_	ic_mou_8 tota 0.75 474.41 82.44 0.00 101.93	l_ic_mou_6 103.86 968.61 364.86 91.88 207.09	total_ic_mou_ 28.4 172.5 381.5 2.3 131.8	19 58 56 31
total_ isd ic mou		ol_ic_mou_6 sp	l_ic_mou_7	spl_ic_mou_8	
8 0.00	16.54	0.00	0.0	0.0	
13	1144.53	0.45	0.0	0.0	
245.28 16	294.46	0.00	0.0	0.0	
0.00 17	0.00	0.00	0.0	0.0	
0.00					
21 0.00	261.04	0.00	0.0	0.0	
	_mou_7 isd_	_ic_mou_8 ic_o	thers_6 ic	_others_7 ic_	_others_8
8	0.00	0.00	0.00	0.00	0.00
13	62.11	393.39	83.48	16.24	21.44
16	0.00	0.23	0.00	0.00	0.00
17	0.00	0.00	0.00	0.00	0.00
21	0.00	0.00	0.00	0.00	0.00
total_ total_rech 8 437	rech_num_6 _amt_6 \ 19	total_rech_num_	_7 total_r 21	ech_num_8 14	

13		6	4		11	
507 16	1	LO	6		2	
570 17		19	2		4	
816 21 600	2	22	26		27	
	total_rech_amt_	_7 total_r	ech_amt_8	max_rech_a	mt_6 max_re	ech_amt_7
8	60	91	120		90	154
13	25	53	717		110	110
16	34	18	160		110	110
17		0	30		110	0
21	68	30	718		50	50
8 13 16 17 21	max_rech_amt_8 30 130 130 30 50	tast_day_	110 100 30 30	tast_day_r	ch_amt_7 \ 0 50 100 0 20	
wol	last_day_rch_am	nt_8 vol_2	g_mb_6 vol	_2g_mb_7	vol_2g_mb_8	
8	_3g_mb_6 \	10	0.0	356.0	0.03	
0.0		Θ	0.0	0.0	0.02	
0.0 16		130	0.0	0.0	0.00	
0.0 17		0	0.0	0.0	0.00	
0.0 21 0.0		50	0.0	0.0	0.00	
	vol_3g_mb_7 vo	ol_3g_mb_8	monthly_2g	_6 monthly	y_2g_7 mont	thly_2g_8
8	750.95	11.94		0	1	0
13	0.00	0.00		0	0	Θ
16	0.00	0.00		0	0	0

17		0.00		0.00		Θ		0		0
21		0.00		0.00		0		0		0
	sachet	2a 6	sachet	2a 7	sachet	2a 8	monthly	3a 6	monthly_	3a 7
\ 8	_	0	_	1		33		0		0
13		0		0		3		0		0
16		0		0		0		0		0
17		0		0		0		Θ		0
21		0		0		0		0		0
	monthly _vbc_3g	_\	sachet				sachet_		aon	
8 21.	03	0		()	0		0	315	
13	9	0		6)	0		0	2607	
16 0.0	9	0		6)	0		0	511	
17 0.0		0		6)	0		0	667	
21		0		6)	0		0	720	
0.0		- 3a -	jun_vbc_	3a a	ova rech	amt 6	_7 churn	tota]	l_mou_god	od \
8 13	910	0.65 0.00	122		9	519 380	.0 0	1014	612.2 1875.7	22
16 17	2	2.45	21	89		459			711.6	57
21		0.00		00		640			1067.4	
	avg_mou	u_actio	on dif	_mou	decrea	se_mou	_action	avg_re	ech_num_a	action
8		324.12	25 -288	3.095			1			17.5
13	1	1262.39	90 -613	3.310			1			7.5
16		597.70	95 - 113	3.965			1			4.0
17		1.56	60 -1339	850			1			3.0
21	1	1245.13	30 177	7.700			0			26.5

diff 8 13 16 17 21	rech_num -1.5 1.5 -6.0 -16.0 4.5	decrease_rech_nu	m_action 1 0 1 1 0	avg_rech_amt_action 360.5 485.0 254.0 15.0 699.0	
diff diff arp		decrease_rech_am	nt_action	avg_arpu_action	
8	-76.5		1	314.7925 -	
63.9285 13 93.3805	-22.0		1	399.4655 -	
16 187.0935	-316.0		1	243.8815 -	
17	-801.0		1	22.2395 -	
667.7685 21 103.3035	99.0		0	617.7565	
decr 8 13 16 17 21	ease_arpu_	action 1 1 1 1 0			

Deriving new column decrease vbc action

This column indicates whether the volume based cost of the customer has decreased in the action phase than the good phase.

```
# VBC in action phase
data['avg_vbc_3g_action'] = (data['jul_vbc_3g'] +
data['aug vbc 3g'])/2
# Difference of good and action phase VBC
data['diff_vbc'] = data['avg_vbc_3g_action'] - data['jun_vbc_3g']
# Checking whether the VBC has decreased on the action month
data['decrease vbc action'] = np.where(data['diff vbc'] < 0 , 1, 0)</pre>
data.head()
   mobile_number loc_og_t2o_mou std_og_t2o_mou loc_ic_t2o_mou
arpu_6 \
     7001524846
                             0.0
                                             0.0
                                                              0.0
378.721
     7002191713
                             0.0
                                             0.0
                                                              0.0
13
492.846
```

16 7000875565 430.975	0.0	0.0	0.0
17 7000187447 690.008	0.0	0.0	0.0
21 7002124215 514.453	0.0	0.0	0.0
arpu_7 arpu_offnet mou 6 \	_8 onnet_mou_6	onnet_mou_7 onnet	_mou_8
8 492.223 137.30 94.66	62 413.69	351.03	35.08
13 205.671 593.20 413.31	50 501.76	108.39	534.24
16 299.869 187.89 296.29	94 50.51	74.01	70.61
17 18.980 25.49 61.64	99 1185.91	9.28	7.79
21 597.753 637.76 757.93	102.41	132.11	85.14
	offnet mou 8 roa	am ic mou 6 roam i	c mou 7
roam_ic_mou_8 \ 8 80.63	136.48	0.00	0.00
0.00 13 119.28	482.46	23.53	144.24
72.11 16 229.74	162.76	0.00	2.83
0.00 17 0.00	5.54	0.00	4.76
4.81 21 896.68 0.00	983.39	0.00	0.00
roam_og_mou_6 8 0.00 13 7.98 16 0.00 17 0.00 21 0.00	roam_og_mou_7 0.00 35.26 17.74 8.46 0.00	roam_og_mou_8 loc_ 0.00 1.44 0.00 13.34 0.00	
loc_og_t2t_mou loc og t2m mou 7		ou_8 loc_og_t2m_mo	u_6
8 217.5 70.58		2.49 80	.96
13 6.3 47.28	19 36	5.01 151	. 13
16 65.1 145.99	16 67	7.38 273	.29
17 0.0 0.00	90	9.00 58	.54

21 87.93	6.16	23.34	91.81	
loc_og_t2f_m			loc_og_t2f_mou_7	
8 0.00	50.54	0.00	0.00	
13 23.51	294.46	4.54	0.00	
16 10.26	128.28	0.00	4.48	
17	0.00	0.00	0.00	
0.00 21 1.58	104.81	0.75	0.00	
	2	1 +2 7	1 +2 0	1
\	2c_mou_6	loc_og_t2c_mou_/	loc_og_t2c_mou_8	loc_og_mou_6
8	0.0	0.0	7.15	378.09
13	0.0	0.0	0.49	205.31
16	0.0	0.0	0.00	315.91
17	0.0	0.0	0.00	97.54
21	0.0	0.0	0.00	97.04
8 28 13 5 16 21 17	ou_7 loc_ 8.18 3.48 5.64 0.00 4.09	og_mou_8 std_og_ 63.04 353.99 205.93 0.00 129.74	t2t_mou_6 std_og_ 116.56 446.41 7.89 1146.91 97.93	t2t_mou_7 \ 133.43 85.98 2.58 0.81 125.94
std_og_t std og t2m m		std_og_t2m_mou_6	std_og_t2m_mou_7	
8 75.69	22.58	13.69	10.04	
13 156.94	498.23	255.36	52.94	
16	3.23	22.99	64.51	
18.29 17	0.00	1.55	0.00	
0.00 21 876.99	61.79	665.36	808.74	
std_og_t	2f_mou_6	std_og_t2f_mou_7	std_og_t2f_mou_8	

	t2c_mou_6 \	0	0	0.0	
8 0.0	0.0	υ.	. 0	0.0	
13	0.0	0	. 0	0.0	
0.0 16	0.0	۵	۵	0.0	
0.0	0.0	U .	. 0	0.0	
17	0.0	0	. 0	0.0	
0.0 21	0.0	۵	. 0	0.0	
0.0	0.0	0.	. 0	0.0	
					_ ,
std _.	_og_t2c_mou_7	std_og_t2c_mou_ ຄ		ou_6 std_og_ 0.26 1	mou_7 \ 43.48
13	0.0				38.93
16	0.0	0	.0 30		67.09
17	0.0		.0 1148		0.81
21	0.0	0.	.0 763	3.29 9	34.69
std	_og_mou_8 isd_o	og mou 6 isd o	og mou 7 iso	d og mou 8	
	mou_6 \				
8	98.28	0.0	0.0	0.00	
0.00 13	655.18	0.0	0.0	1.29	
0.00	033110	010	0.0	1123	
16	21.53	0.0	0.0	0.00	
0.00 17	0.00	0.0	0.0	0.00	
2.58				0.00	
21	938.79	0.0	0.0	0.00	
0.00					
spl _.	_og_mou_7 spl_c	og_mou_8 og_ot	thers_6 og_c	others_7 og_	others_8
8	0.00	10.23	0.00	0.0	0.0
13	0.00	4.78	0.00	0.0	0.0
16	3.26	5.91	0.00	0.0	0.0
17	0.00	0.00	0.93	0.0	0.0
21	0.00	0.00	0.00	0.0	0.0
		al_og_mou_7 1	total_og_mou_	_8	
	t2t_mou_6 \	121 66	171 [56	22 04
8	508.36	431.66	171.5	00	23.84
13	907.09	192.41	1015.2	26	67.88

16	346.81	286.01	233.38	41.33
17	1249.53	0.81	0.00	34.54
21	860.34	1028.79	1068.54	2.48
		loc_ic_t2t_mou_8	loc_ic_t2m_mou_6	
8	2m_mou_7 \ 9.84	0.31	57.58	
13.98 13	7.58	52.58	142.88	
18.53				
16 149.69	71.44	28.89	226.81	
17 2.31	0.00	0.00	47.41	
21	10.19	19.54	118.23	
74.63				
	ic_t2m_mou_8 2f_mou_8 \	loc_ic_t2f_mou_6	loc_ic_t2f_mou_7	
8	15.48	0.00	0.00	
0.00 13	195.18	4.81	0.00	
7.49				
16 32.71	150.16	8.71	8.68	
17	0.00	0.00	0.00	
0.00 21	129.16	4.61	2.84	
10.39				
_			mou_8 std_ic_t2t	_
8 13	81.43 215.58	23.83 26.11 2	15.79 255.26	0.00 115.68
16	276.86		211.78 0.00	68.79
17 21	81.96 125.33	2.31 87.68 1	.59.11	8.63 14.06
std i	ic t2t mou 7	std ic t2t mou 8	std ic t2m mou 6	
std_ic_t2	2m_mou_7 \			
8 4.08	0.58	0.10	22.43	
13 29.79	38.29	154.58	308.13	
16	78.64	6.33	18.68	
73.08 17	0.00	0.00	1.28	

0.00					
21	5.98	0.1	8	67.69	
38.23	3.30	011	o .	0,103	
		std_ic_t2f_mou_	6 std_ic_	_t2f_mou_7	
std_1c_t2 8	f_mou_8 \ 0.65	0.0	A	0.0	
0.00	0.05	0.0	O	0.0	
13	317.91	0.0	0	0.0	
1.91					
16	73.93	0.5	1	0.0	
2.18	0.00	0.0	0	0.0	
17 0.00	0.00	0.0	U	0.0	
21	101.74	0.0	Θ	0.0	
0.00	101171	0.10	•	0.0	
_	c_t2o_mou_6	std_ic_t2o_mou_	7 std_ic_	_t2o_mou_8	std_ic_mou_6
8	0.0	0	0	0.0	22 42
0	0.0	0.	U	0.0	22.43
13	0.0	0.	0	0.0	423.81
16	0.0	0.	0	0.0	87.99
17	0.0	0.	Θ	0.0	9.91
21	0.0	0.	Θ	0.0	81.76
ctd i	c mou 7 std	ic mou 0 total	ic mou 6	total_ic_	mou 7 \
8	4.66	_ic_mod_8 totat 0.75	103.86		28.49
13	68.09	474.41	968.61		72.58
16	151.73	82.44	364.86		81.56
17	0.00	0.00	91.88		2.31
21	44.21	101.93	207.09	1	31.89
		pl_ic_mou_6 spl	_ic_mou_7	spl_ic_mo	u_8
isd_ic_mo		0.00	0.0		0 0
8 0.00	16.54	0.00	0.0		0.0
13	1144.53	0.45	0.0		0.0
245.28	1111133	01 15	0.0		0.10
16	294.46	0.00	0.0	1	0.0
0.00					
17	0.00	0.00	0.0		0.0
0.00	261 04	0.00	0.0		0 0
21 0.00	261.04	0.00	0.0		0.0
0.00					

	isd_ic_mou_7	isd_ic_	mou_8 i	c_others_6	ic_others_7	ic_others_8
8	0.00		0.00	0.00	0.00	0.00
13	62.11	3	93.39	83.48	16.24	21.44
16	0.00		0.23	0.00	0.00	0.00
17	0.00		0.00	0.00	0.00	0.00
21	0.00		0.00	0.00	0.00	0.00
8 437 13 507 16 570 17 816 21		m_6 tot 19 6 10 19 22	al_rech_i	num_7 tota 21 4 6 2 26	al_rech_num_8 14 11 2 4	
600		t_7 tot	al_rech_a	amt_8 max	_rech_amt_6	max_rech_amt_7
8		601		120	90	154
13						
		253		717	110	110
16		253 348		717 160	110 110	110 110
16 17						
		348		160	110	110
17		348 0 680 8 last_ 0 0 0	day_rch_a	160 30 718	110 110 50 t_day_rch_amt_	110 0 50
17 21 8 13 16 17 21	max_rech_amt_8 30 130 130 30 50 last_day_rch_8 3g_mb_6 \	348 0 680 8 last_ 0 0 0 0	ol_2g_mb __	160 30 718 amt_6 last 50 110 100 30 30 30 _6 vol_2g	110 110 50 t_day_rch_amt_	110 0 50 -7 0 50 00 0

0 0					
0.0 16 0.0	1	30 0	.0 0.	0	0.00
17		0 0	.0 0.	0	0.00
0.0 21 0.0		50 0	.0 0.	0	0.00
vol_3	g_mb_7 vol	_3g_mb_8 mon	thly_2g_6 mon	thly_2g_7	monthly_2g_8
	750.95	11.94	0	1	0
13	0.00	0.00	0	0	0
16	0.00	0.00	0	0	0
17	0.00	0.00	0	0	Θ
21	0.00	0.00	0	0	0
sache	et_2g_6 sac	het_2g_7 sac	het_2g_8 mont	hly_3g_6	monthly_3g_7
8	0	1	3	Θ	0
13	0	0	3	0	Θ
16	0	0	0	0	Θ
17	0	0	0	0	0
21	0	0	0	0	0
month		chet_3g_6 sa	chet_3g_7 sac	het_3g_8	aon
8 21.03	0	0	0	0	315
13	0	0	0	0	2607
0.00 16	0	0	0	0	511
0.00 17	0	0	Θ	0	667
0.00 21 0.00	0	0	0	0	720
jul_v	bc_3g jun_ 10.65 0.00 2.45	vbc_3g avg_r 122.16 0.00 21.89	ech_amt_6_7 ch 519.0 380.0 459.0	urn total 0 0 0	_mou_good \ 612.22 1875.70 711.67

17 21	0.00 0.00	0.00 0.00	408.0 640.0		1341.4 1067.4	
	avg_mou_action	diff_mou	decrease_mou_a	action avg_	_rech_num_a	ction
8	324.125	-288.095		1		17.5
13	1262.390	-613.310		1		7.5
16	597.705	-113.965		1		4.0
17	1.560	-1339.850		1		3.0
21	1245.130	177.700		0		26.5
				·		
8 13 16 17 21	diff_rech_num -1.5 1.5 -6.0 -16.0 4.5	decrease_r	ech_num_action 1 0 1 1 0	avg_rech_a	amt_action 360.5 485.0 254.0 15.0 699.0	\
	diff_rech_amt	decrease_r	ech_amt_action	avg_arpu_a	action	
dif	f_arpu \ -76.5		1	314	1.7925 -	
63.9	9285 -22.0		1	300	9.4655 -	
93.3	3805					
16 187	-316.0 .0935		1	243	3.8815 -	
17 667	-801.0 .7685		1	22	2.2395 -	
21	99.0		0	617	7.7565	
103	. 3035					
dec	_decrease_arpu_ rease_vbc_actio		_vbc_3g_action	diff_vbc		
8		1	465.840	343.680		
0 13		1	0.000	0.000		
0 16		1	1.225	-20.665		
1						
17 0		1	0.000	0.000		
21		0	0.000	0.000		
0						

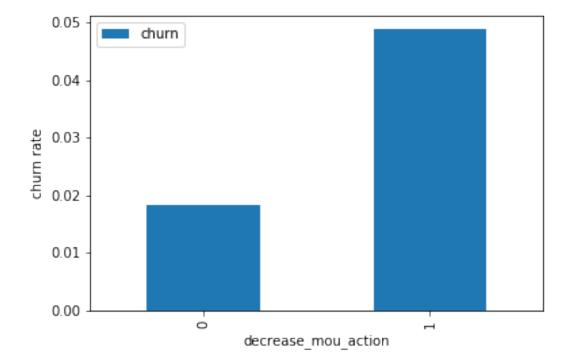
EDA

Univariate analysis

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

```
# Converting churn column to int in order to do aggfunc in the pivot
table
data['churn'] = data['churn'].astype('int64')

data.pivot_table(values='churn', index='decrease_mou_action',
aggfunc='mean').plot.bar()
plt.ylabel('churn rate')
plt.show()
```

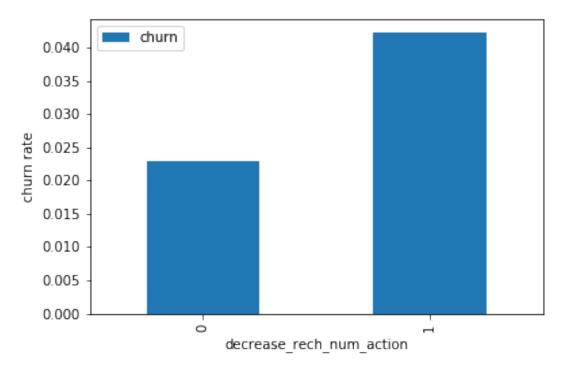


Analysis

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.

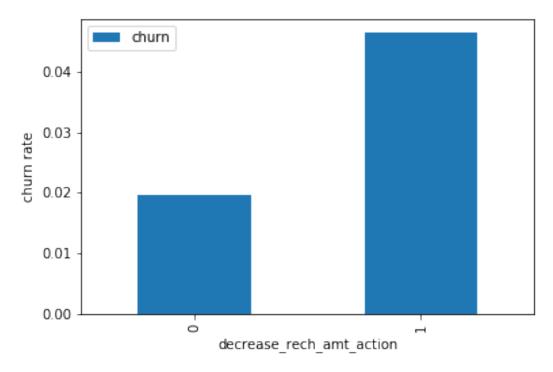
Churn rate on the basis whether the customer decreased her/his number of recharge in action month

```
data.pivot_table(values='churn', index='decrease_rech_num_action',
   aggfunc='mean').plot.bar()
plt.ylabel('churn rate')
plt.show()
```



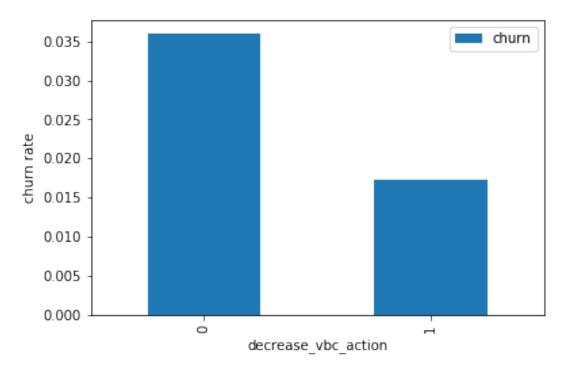
As expected, the churn rate is more for the customers, whose number of recharge in the action phase is lesser than the number in good phase.

```
Churn rate on the basis whether the customer decreased her/his amount of recharge in action month data.pivot_table(values='churn', index='decrease_rech_amt_action', aggfunc='mean').plot.bar() plt.ylabel('churn rate') plt.show()
```



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.

```
Churn rate on the basis whether the customer decreased her/his volume based cost in action month data.pivot_table(values='churn', index='decrease_vbc_action', aggfunc='mean').plot.bar() plt.ylabel('churn rate') plt.show()
```



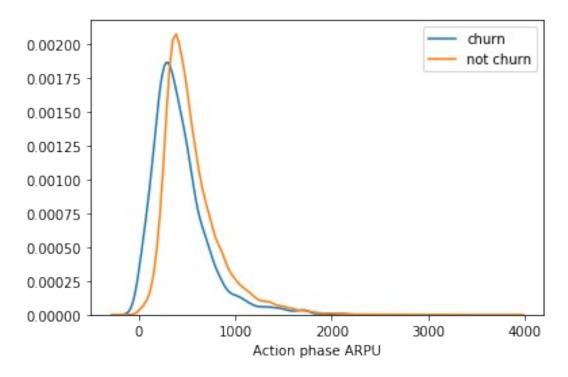
Here we see the expected result. The churn rate is more for the customers, whose volume based cost in action month is increased. That means the customers do not do the monthly recharge more when they are in the action phase.

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

```
# Creating churn dataframe
data_churn = data[data['churn'] == 1]
# Creating not churn dataframe
data_non_churn = data[data['churn'] == 0]

# Distribution plot
ax =
sns.distplot(data_churn['avg_arpu_action'],label='churn',hist=False)
ax = sns.distplot(data_non_churn['avg_arpu_action'],label='not
churn',hist=False)
ax.set(xlabel='Action phase ARPU')

[Text(0.5, 0, 'Action phase ARPU')]
```

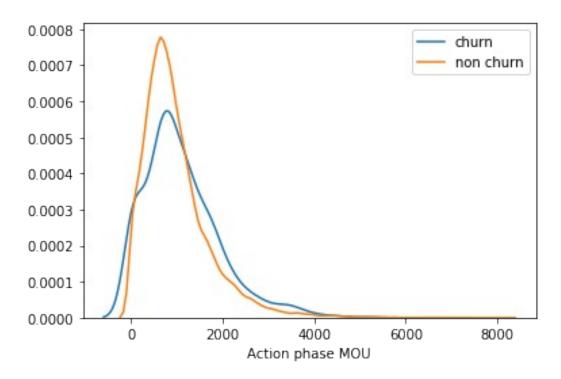


Average revenue per user (ARPU) for the churned customers is mostly densed on the 0 to 900. The higher ARPU customers are less likely to be churned.

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

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

```
# Distribution plot
ax =
sns.distplot(data_churn['total_mou_good'],label='churn',hist=False)
ax = sns.distplot(data_non_churn['total_mou_good'],label='non
churn',hist=False)
ax.set(xlabel='Action phase MOU')
[Text(0.5, 0, 'Action phase MOU')]
```

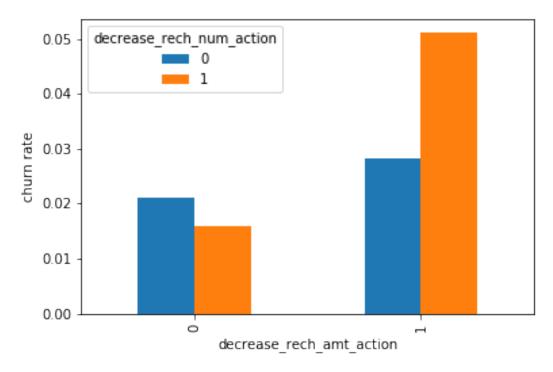


Minutes of usage(MOU) of the churn customers is mostly populated on the 0 to 2500 range. Higher the MOU, lesser the churn probability.

Bivariate analysis

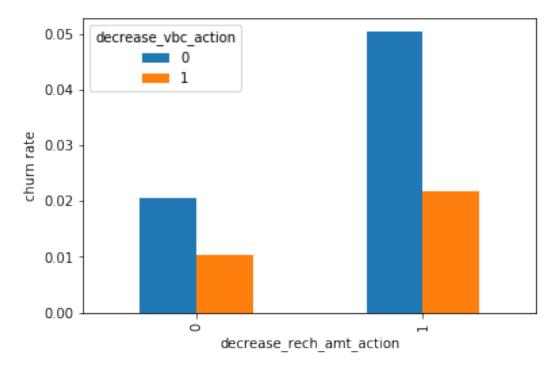
Analysis of churn rate by the decreasing recharge amount and number of recharge in the action phase

```
data.pivot_table(values='churn', index='decrease_rech_amt_action',
columns='decrease_rech_num_action', aggfunc='mean').plot.bar()
plt.ylabel('churn rate')
plt.show()
```



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.

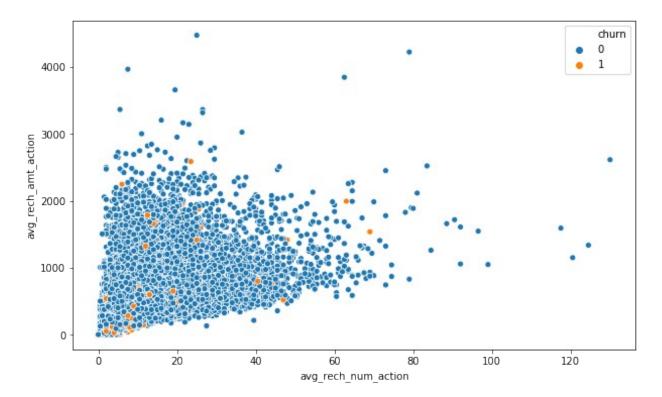
```
Analysis of churn rate by the decreasing recharge amount and volume based cost in the action phase data.pivot_table(values='churn', index='decrease_rech_amt_action', columns='decrease_vbc_action', aggfunc='mean').plot.bar() plt.ylabel('churn rate') plt.show()
```



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.

Analysis of recharge amount and number of recharge in action month

```
plt.figure(figsize=(10,6))
ax = sns.scatterplot('avg_rech_num_action','avg_rech_amt_action',
hue='churn', data=data)
```



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

Dropping few derived columns, which are not required in further analysis

```
data =
data.drop(['total_mou_good','avg_mou_action','diff_mou','avg_rech_num_
action','diff_rech_num','avg_rech_amt_action',

'diff_rech_amt','avg_arpu_action','diff_arpu','avg_vbc_3g_action','dif
f_vbc','avg_rech_amt_6_7'], axis=1)
```

Train-Test Split

```
# Import library
from sklearn.model_selection import train_test_split

# Putting feature variables into X
X = data.drop(['mobile_number','churn'], axis=1)

# Putting target variable to y
y = data['churn']

# Splitting data into train and test set 80:20
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.8, test_size=0.2, random_state=100)
```

Dealing with data imbalance

We are creating synthetic samples by doing upsampling using SMOTE(Synthetic Minority Oversampling Technique).

```
# Imporing SMOTE
from imblearn.over_sampling import SMOTE

# Instantiate SMOTE
sm = SMOTE(random_state=27)

# Fittign SMOTE to the train set
X_train, y_train = sm.fit_sample(X_train, y_train)
```

Feature Scaling

```
# Standardization method
from sklearn.preprocessing import StandardScaler
# Instantiate the Scaler
scaler = StandardScaler()
# List of the numeric columns
cols scale = X train.columns.to list()
# Removing the derived binary columns
cols scale.remove('decrease mou action')
cols scale.remove('decrease rech num action')
cols scale.remove('decrease rech amt action')
cols scale.remove('decrease arpu action')
cols scale.remove('decrease vbc action')
# Fit the data into scaler and transform
X train[cols scale] = scaler.fit transform(X train[cols scale])
X train.head()
   loc og t2o mou
                   std og t2o mou loc ic t2o mou
                                                     arpu 6
                                                            arpu 7
                                                   0.140777 -0.522792
0
              0.0
                              0.0
                                              0.0
                                              0.0 -1.427243 4.428047
              0.0
                              0.0
              0.0
2
                              0.0
                                              0.0 -0.222751 0.543206
3
              0.0
                              0.0
                                              0.0 -0.911173 0.842273
              0.0
                              0.0
                                              0.0 0.271356 0.247684
             onnet mou 6 onnet mou 7 onnet mou 8 offnet mou 6 \
     arpu 8
0 -0.276289
                0.106540
                            -0.662084
                                         -0.465777
                                                       -0.211202
```

```
3.254270
                -0.658491
                                            -0.004450
                                                           -0.776075
                              -0.236590
  0.809117
                -0.601239
                              -0.599206
                                            -0.331043
                                                           -0.363395
   0.731302
                -0.702232
                              -0.650471
                                            -0.458464
                                                           -0.789784
4 1.256421
                -0.356392
                              -0.180394
                                             0.114727
                                                            0.899204
   offnet mou 7 offnet mou 8
                                 roam ic mou 6
                                                roam ic mou 7
roam ic mou 8 \
      -0.636415
                      0.317224
                                     -0.254996
                                                     -0.001208
0.235211
       2.523985
                      2.732154
                                     -0.254996
                                                     -0.253231
0.304660
                                     -0.254996
      -0.495976
                     -0.028236
                                                     -0.253231
0.304660
      -0.654483
                     -0.519047
                                     -0.254996
                                                     -0.253231
0.304660
       0.904465
                      1.255807
                                     -0.231882
                                                     -0.253231
0.304660
   roam og mou 6
                   roam og mou 7
                                   roam og mou 8
                                                   loc og t2t mou 6
0
       -0.300833
                        -0.374857
                                       -0.412810
                                                           -0.263308
1
       -0.300833
                        -0.374857
                                        -0.431026
                                                           -0.201396
2
       -0.300833
                        -0.374857
                                        -0.431026
                                                            0.077694
3
       -0.300833
                       -0.374857
                                        -0.431026
                                                           -0.192289
       -0.202644
                       -0.374857
                                       -0.431026
                                                            0.128384
   loc_og_t2t_mou_7
                     loc_og_t2t_mou_8 loc_og_t2m_mou_6
loc og t2m mou 7 \setminus
          -0.311548
                              -0.251411
                                                  0.485770
0.190660
           0.270791
                               0.198344
                                                 -0.529474
1.106670
          -0.095916
                               0.228431
2
                                                  0.605362
0.258376
          -0.181513
                              -0.064925
                                                 -0.371787
0.205099
           0.784682
                               1.062326
                                                  1.423002
0.996094
                     loc og t2f mou_6 loc_og_t2f_mou_7
   loc og t2m mou 8
loc og t2f mou 8 \
          -0.3\overline{9}9182
                              -0.256866
                                                 -0.267401
0.244832
           0.288951
                              -0.276320
                                                 -0.267401
0.244832
           0.908270
                               1.475098
                                                  0.451689
0.131562
          -0.251524
                              -0.157090
                                                  0.216496
0.244832
           1.845573
                               0.780430
                                                  1.055332
0.519904
```

```
loc og t2c mou 6 loc og t2c mou 7 loc og t2c mou 8 loc og mou 6
/
0
           -0.191587
                              -0.267368
                                                  -0.244432
                                                                  0.129144
1
           -0.191587
                              -0.267368
                                                  -0.244432
                                                                 -0.477059
2
           -0.191587
                              -0.267368
                                                  -0.244432
                                                                  0.512549
3
            1.002136
                               2.438345
                                                   2.557369
                                                                 -0.364845
            1.811266
                              -0.267368
                                                   0.843143
                                                                  1.025297
   loc og mou 7
                 loc og mou 8
                                 std og t2t mou 6
                                                     std og t2t mou 7 \
      -0.335468
                                          0.254982
                                                             -0.528622
0
                      -0.418749
1
       0.843930
                      0.290569
                                         -0.570615
                                                             -0.320253
2
       0.121104
                      0.710496
                                         -0.618738
                                                             -0.551860
3
      -0.233086
                      -0.212616
                                         -0.619956
                                                             -0.570510
       1.183543
                      1.843624
                                         -0.414192
                                                             -0.474892
                      std og t2m mou 6 std og t2m mou 7
   std og t2t mou 8
std og t2m mou 8 \
           -0.3\overline{3}8018
                              -0.342394
                                                  -0.504282
0.650664
           -0.041333
                              -0.512504
                                                   2.294191
1
3.087483
           -0.420186
                              -0.617043
                                                  -0.571393
0.416795
           -0.420186
                              -0.621707
                                                  -0.578677
0.406309
           -0.327001
                               0.357250
                                                   0.585026
0.555984
                      std og t2f mou 7 std og t2f mou 8
   std og t2f mou 6
std og t2c mou 6 \
           -0.143576
                              -0.139257
                                                  -0.119299
0.0
1
           -0.143576
                              -0.139257
                                                  -0.119299
0.0
2
           -0.143576
                              -0.139257
                                                  -0.067469
0.0
           -0.143576
                              -0.139257
                                                  -0.119299
3
0.0
           -0.143576
                              -0.139257
                                                  -0.119299
4
0.0
                                                         std_og_mou_7
   std og t2c mou 7
                      std og t2c mou 8
                                          std og mou 6
0
                 0.0
                                     0.0
                                             -0.048161
                                                             -0.731560
                 0.0
1
                                     0.0
                                             -0.771902
                                                              1.368343
```

```
2
                 0.0
                                    0.0
                                                            -0.794939
                                             -0.878705
3
                 0.0
                                    0.0
                                             -0.882786
                                                            -0.813361
                 0.0
                                    0.0
                                             -0.062970
                                                             0.066271
                  isd og mou 6
                                 isd og mou 7
                                                isd_og_mou_8
   std og mou 8
spl og mou 6 \
       0.214243
                     -0.080803
                                    -0.092449
                                                   -0.061631
0
0.347585
       2.063999
                     -0.080803
                                    -0.092449
                                                    -0.061631
0.347585
      -0.563412
                     -0.080803
                                    -0.031701
                                                    -0.061631
0.347585
      -0.557142
                     -0.080803
                                    -0.092449
                                                    -0.061631
0.299948
                     -0.080803
                                    -0.092449
       0.157380
                                                    0.132387
0.906003
   spl og mou 7 spl og mou 8
                                 og others 6 og others 7
og others 8 \
      -0.363159
                     -0.017165
                                   -0.346191
                                                 -0.015583
                                                               -0.013735
      -0.363159
                     -0.290355
                                   -0.346191
                                                 -0.015583
                                                               -0.013735
      -0.203140
                      0.151727
                                   -0.346191
                                                 -0.015583
                                                               -0.013735
3
       0.639325
                      0.743145
                                   -0.244436
                                                 -0.015583
                                                               -0.013735
      -0.251495
                      0.107282
                                   -0.346191
                                                 -0.015583
                                                               -0.013735
   total og mou 6
                    total og mou 7
                                     total og mou 8
                                                       loc ic t2t mou 6 \
0
         -0.000389
                          -0.860412
                                           -0.011382
                                                              -0.203981
1
         -0.970285
                          1.670188
                                            1.938953
                                                              -0.410762
2
         -0.637091
                          -0.716013
                                           -0.155609
                                                              -0.073331
3
         -1.012362
                          -0.864684
                                           -0.569774
                                                              -0.262725
         0.411184
                           0.572929
                                            1.014123
                                                              -0.238919
   loc ic t2t mou 7
                     loc ic t2t mou 8 loc ic t2m mou 6
loc_ic_t2m_mou_7 \
           -0.266718
                              -0.242771
                                                 -0.380593
0.272733
                               0.156537
           0.193158
                                                 -0.481723
1
0.744741
           -0.082299
                               0.189717
                                                  0.211940
0.166326
           -0.287643
                              -0.150724
                                                  0.157353
0.540086
           0.483606
                               0.750395
                                                 -0.281606
0.384609
```

```
loc ic t2f mou 6 loc ic t2f mou 7
   loc_ic_t2m_mou_8
loc ic t2f mou 8 \
          -0.437571
                              -0.290528
                                                 -0.270877
0.150060
           0.256589
                              -0.290528
                                                 -0.270877
0.257696
2
                               0.223523
                                                 -0.117519
           0.542595
0.167136
          -0.095861
                              -0.290528
                                                 -0.268736
0.250781
           0.578588
                               0.220005
                                                  0.304016
0.113318
                                loc_ic_mou 8
                                                std ic_t2t_mou_6 \
   loc ic mou 6
                 loc ic mou 7
0
      -0.409101
                     -0.363983
                                    -0.440411
                                                       -0.175106
1
      -0.583307
                      0.570197
                                     0.219470
                                                       -0.215496
2
       0.142990
                                     0.490068
                      0.054813
                                                       -0.215496
3
      -0.061177
                      0.184367
                                    -0.172152
                                                       -0.215496
      -0.286414
                      0.556564
                                     0.777995
                                                       -0.215496
   std ic t2t mou 7
                      std ic t2t mou 8 std ic t2m mou 6
std ic t2m mou 7 \
          -0.159825
                              0.078711
                                                 -0.164347
0.367474
          -0.200464
                              -0.112725
                                                 -0.355157
0.100763
          -0.200464
                              -0.187265
                                                 -0.361304
0.256979
                              -0.187265
                                                 -0.361304
          -0.200464
0.343715
          -0.200464
                              0.108490
                                                 -0.066471
1.099027
   std ic t2m mou 8
                      std ic t2f mou 6
                                        std ic t2f mou 7
std ic t2f mou 8 \
          -0.117454
                              -0.135479
                                                 -0.137327
0.110642
                              -0.135479
                                                 -0.137327
          -0.034777
0.110642
          -0.217027
                              -0.135479
                                                 -0.137327
0.109904
                              -0.135479
          -0.229338
                                                 -0.137327
0.110642
           0.087246
                               0.078665
                                                 -0.072592
0.078362
   std ic t2o mou 6
                      std_ic_t2o_mou_7 std_ic_t2o_mou_8
                                                            std ic mou 6
/
                                                       0.0
0
                 0.0
                                    0.0
                                                                -0.234904
```

```
1
                0.0
                                   0.0
                                                      0.0
                                                               -0.386264
2
                0.0
                                   0.0
                                                      0.0
                                                               -0.390310
                0.0
                                   0.0
3
                                                      0.0
                                                               -0.390310
                0.0
                                   0.0
                                                      0.0
                                                               -0.171917
   std ic mou 7 std ic mou 8 total ic mou 6 total ic mou 7
total_ic_mou_8 \
       0.121\overline{3}32
                     -0.064154
                                      -0.475564
                                                      -0.287010
0.420829
      -0.078694
                     -0.096335
                                      -0.688082
                                                       0.417278
0.125859
      -0.312284
                     -0.272135
                                      -0.073424
                                                      -0.106086
0.290685
      -0.368919
                     -0.281605
                                      -0.248992
                                                       -0.029566
0.273184
       0.581141
                      0.130311
                                      -0.108798
                                                       2.215081
2.220000
   spl ic mou 6 spl ic mou 7 spl ic mou 8 isd ic mou 6
isd ic mou 7 ∖
      -0.366516
                     -0.089786
                                   -0.192624
                                                  -0.151655
0.153778
      -0.366516
                     -0.089786
                                   -0.192624
                                                  -0.151655
0.153778
      -0.366516
                     -0.089786
                                   -0.192624
                                                  -0.132791
0.099984
      -0.366516
                      0.968997
                                   -0.192624
                                                  -0.151655
0.153778
    -0.366516
                                   -0.192624
                     -0.089786
                                                   1.103151
4.262216
   isd ic mou 8
                 ic others 6 ic others 7 ic others 8
total rech num 6
     -0.126576
                    -0.099745
                                 -0.121704
                                               -0.081491
0.192736
                    -0.099745
                                 -0.121704
                                               -0.081491
      -0.126576
0.738325
      -0.126576
                    -0.099745
                                 -0.121704
                                               -0.081491
0.738325
      -0.126576
                    -0.099745
                                 -0.121704
                                               -0.081491
0.272794
       4.608310
                     0.746998
                                 26.877658
                                               25.149134
0.738325
   total_rech_num_7 total_rech_num_8 total_rech_amt_6
total rech amt 7 \
```

```
-0.444988
                               0.305289
                                                   0.044172
0.726027
1
           4.142873
                               2.933529
                                                  -1.364090
4.102682
           -0.327351
                              -0.195328
                                                  -0.110493
0.728436
                               0.555597
                                                  -0.946224
3
           0.378474
0.693869
           -0.209713
                               0.555597
                                                  -0.368267
0.175368
   total rech amt 8
                      max rech amt 6
                                        max rech amt 7
                                                         max rech amt 8 \
0
           -0.235478
                             0.054992
                                              0.023937
                                                                0.029739
1
            3.350107
                            -0.748908
                                              -0.386255
                                                               -0.054702
2
            0.772451
                             0.039532
                                               0.023937
                                                                0.198621
3
            0.803137
                             0.054992
                                              0.768359
                                                                0.966264
            1,225665
                            -0.207821
                                              -0.158371
                                                                0.029739
   last day rch amt 6 last day rch amt 7 last day rch amt 8
vol 2g mb 6
              0.601511
                                   -0.811577
                                                         -0.626096
0.094017
             -0.405085
                                   -0.350629
                                                         -0.066907
1
0.245535
              0.272431
                                    0.386888
                                                         0.565618
0.077862
             -0.598662
                                   -0.535008
                                                         -0.351085
0.059437
                                    0.386888
                                                         0.565618
              0.272431
0.245535
                 vol 2g mb 8
                                             vol 3g mb 7
   vol 2g mb 7
                               vol 3g mb 6
                                                           vol 3g mb 8 \
                    \overline{1.750783}
0
      0.696113
                                  0.510634
                                                 1.202971
                                                              -0.241652
1
     -0.235847
                   -0.207939
                                 -0.262491
                                                -0.274601
                                                              -0.249913
2
     -0.034247
                    0.104903
                                  0.950720
                                                 1.994409
                                                               1.671342
3
                                                 5.767468
     -0.040569
                   -0.027155
                                  2.610032
                                                               4.000137
                   -0.207939
                                 -0.262491
     -0.235847
                                                -0.274601
                                                              -0.249913
   monthly 2g 6
                  monthly_2g_7
                                 monthly_2g_8
                                                sachet 2g 6 sachet 2g 7
/
                      3.104207
                                     -0.232664
                                                                  2.358097
0
       3.236849
                                                    4.023237
                                     -0.232664
      -0.246650
                      -0.251375
                                                   -0.255793
                                                                 -0.269796
                                     -0.232664
      -0.246650
                      3.104207
                                                    0.457379
                                                                  1.044151
                     -0.251375
                                     -0.232664
                                                   -0.255793
       3.236849
                                                                 -0.269796
                                                   -0.255793
      -0.246650
                     -0.251375
                                     -0.232664
                                                                 -0.269796
```

```
sachet 2g 8
                monthly 3g 6 monthly 3g 7 monthly 3g 8 sachet 3g 6
0
      2.447476
                    -0.224183
                                   -0.221779
                                                  -0.216364
                                                               -0.141182
     -0.268245
                    -0.224183
                                   -0.221779
                                                 -0.216364
                                                               -0.141182
      1.089616
                    -0.224183
                                   -0.221779
                                                 -0.216364
                                                                1.315163
     -0.268245
                     2.171393
                                    9.083717
                                                  4.618685
                                                               -0.141182
                    -0.224183
     -0.268245
                                   -0.221779
                                                  -0.216364
                                                               -0.141182
                sachet 3g 8
                                    aon aug vbc 3g jul vbc 3g
   sachet 3g 7
jun_vbc_3g \
                   -0.113882 -0.361238
     -0.136208
                                          -0.236209
                                                       -0.265392
0.110582
     -0.136208
                   -0.113882 -0.790173
                                          -0.255884
                                                       -0.265392
0.259366
                    2.526725 1.571302
                                           3.307334
                                                        2.691063
      2.575301
1.700218
     -0.136208
                   -0.113882 -0.951024
                                          -0.255884
                                                       -0.265392
0.259366
                   -0.113882 -0.519757
                                          -0.255884
                                                       -0.265392
     -0.136208
0.259366
   decrease mou action
                         decrease rech num action
decrease_rech_amt_action
                                                 1
1
1
                      0
                                                 0
0
2
                                                 0
0
3
                                                 0
0
4
                      0
                                                 0
0
   decrease arpu action
                          decrease vbc action
0
                       1
                                             1
1
                       0
                                             0
2
                       0
                                             0
3
                       0
                                             0
```

Scaling the test set

We don't fit scaler on the test set. We only transform the test set.

```
# Transform the test set
X test[cols scale] = scaler.transform(X test[cols scale])
X test.head()
       loc og t2o mou std og t2o mou loc ic t2o mou
                                                         arpu 6
arpu 7
5704
                  0.0
                                   0.0
                                                    0.0 0.244310 -
0.268832
                  0.0
                                   0.0
                                                    0.0 0.048359 -
64892
0.779609
                  0.0
                                   0.0
                                                    0.0 0.545470
39613
0.184388
                                   0.0
                  0.0
                                                    0.0 0.641508
93118
0.816632
                                   0.0
81235
                  0.0
                                                    0.0 3.878627
0.911619
                 onnet mou 6 onnet mou 7 onnet mou 8
         arpu 8
offnet mou 6 \
5704 1.005890
                    -0.725286
                                 -0.690223
                                               -0.476634
                                                              0.483540
64892 -0.157969
                   -0.734066
                                 -0.698072
                                               -0.502219
                                                             -0.358555
39613 1.403349
                   -0.537110
                                 -0.521615
                                               -0.206890
                                                              0.694901
93118 -0.211023
                    -0.058843
                                               -0.155872
                                  0.029897
                                                             -0.148197
81235 2.745295
                    4.117829
                                  1.452446
                                               2.809582
                                                             -0.002634
       offnet mou 7
                     offnet mou 8
                                    roam ic mou 6
                                                    roam ic mou 7 \
5704
           0.307300
                          2.323745
                                        -0.077655
                                                        -0.\overline{253231}
64892
          -0.577717
                         -0.256061
                                         0.022864
                                                        -0.253231
39613
           0.435043
                          1.465067
                                        -0.254996
                                                        -0.253231
          -0.143451
93118
                         -0.410827
                                        -0.254996
                                                        -0.253231
81235
          -0.290323
                          0.029332
                                        -0.254996
                                                        -0.253231
       roam ic mou 8
                       roam og mou 6
                                      roam og mou 7
                                                      roam og mou 8 \
5704
           -0.304660
                            0.215992
                                           -0.374857
                                                          -0.431026
64892
           -0.304660
                           -0.120122
                                           -0.374857
                                                          -0.431026
39613
           -0.304660
                           -0.300833
                                           -0.374857
                                                          -0.431026
                                           -0.374857
93118
           -0.304660
                           -0.300833
                                                          -0.431026
81235
           -0.003778
                           -0.300833
                                          -0.374857
                                                          1.456232
       loc og t2t mou 6 loc og t2t mou 7 loc og t2t mou 8
loc og t2m mou 6 \
5704
              -0.278217
                                 -0.282623
                                                    -0.106758
0.028192
                                 -0.302589
64892
              -0.278380
                                                    -0.174571
0.300150
```

39613	0.254268	0.146234	0.514266	
2.795255 93118	0.871759	1.002772	0.222587	
0.871444 81235	2.888120	0.289221	1.362336	
0.767176				
	c_og_t2m_mou_7 f mou 7 \	loc_og_t2m_mou_8	loc_og_t2f_mou_6	
5704	0.006336	0.034141	-0.087435	-
0.267401 64892	-0.204014	-0.295881	-0.261886	-
0.267401 39613	2.186811	3.743713	0.011714	-
0.076422 93118	0.713384	-0.116066	1.669630	
1.311405 81235	0.540001	0.742988	0.005438	_
0.267401	0.540001	0.742300	0.005450	_
		loc_og_t2c_mou_6	loc_og_t2c_mou_7	
loc_og_t20 5704	c_mou_8 \ -0.244832	0.037799	-0.267368	-
0.244432 64892	-0.244832	-0.191587	-0.267368	_
0.244432 39613	1.174644	-0.191587	-0.267368	
0.244432				
93118 0.244432	0.642996	-0.191587	-0.267368	-
81235 0.244432	-0.244832	-0.191587	-0.267368	-
100	coa mou 6 loc	og mou 7 loc og r	mou 8 std og t2t mou 6	\
5704	-0.161248 -0.379084	-0.195270 -0.05	55078 -0.610819 06653 -0.619956	,
64892 39613	1.932970	1.438327 2.76	63270 -0.619956	
93118 81235	1.189608 2.297311		93205 -0.358067 78083 3.237049	
sto	d og t2t mou 7	std og t2t mou 8	std og t2m mou 6	
	n_mou_7 \ -0.570510	-0.420186	0.346789	
0.369671				
64892 0.437192	-0.570510	-0.415897	-0.231854	-
39613 0.343132	-0.570510	-0.420186	-0.394991	-
93118 0.406765	-0.342850	-0.221957	-0.507976	-
0.400/03				

81235 0.435005	1.462701	L 2	.078469	-0.263825	-
std std og t2f	_og_t2m_mou_8 _mou_8 \	3 std_og_t2	f_mou_6 std_	_og_t2f_mou_7	
5704	2.702104	- 0	. 143576	-0.139257	-
0.119299 64892 0.104326	-0.040526	- 0	. 143576	-0.139257	-
39613	-0.177784	1	. 244575	-0.139257	-
0.119299 93118	-0.346116	5 -0	. 143576	-0.139257	-
0.119299 81235 0.119299	-0.400887	- 0	. 143576	-0.139257	-
std std og mou		std_og_t2	c_mou_7 std_	_og_t2c_mou_8	
5704	0.6)	0.0	0.0	-
0.214836 64892	0.0)	0.0	0.0	-
0.616620					
39613 0.708089	0.0)	0.0	0.0	-
93118	0.0)	0.0	0.0	-
0.612400 81235 2.200139	0.6)	0.0	0.0	
	og mou 7 st	d og mou 8	isd og mou 6	isd og mou 7	
isd_og_mou	_8 \				
5704 0.061631	-0.152215	1.550482	-0.080803	-0.092449	-
64892	-0.714724	-0.306010	-0.080803	-0.092449	-
0.061631 39613	-0.649149	-0.402193	-0.080803	3 -0.092449	-
0.061631	0 520705	0 204271	0 00000	0 002440	
93118 0.061631	-0.530795	-0.384371	-0.080803	-0.092449	-
81235 0.061631	0.739892	1.109859	-0.080803	-0.092449	-
		ol_og_mou_7	spl_og_mou_8	3 og_others_6	
og_others_ 5704	1.055196	0.774917	0.757966	0.315218	-
0.015583 64892	-0.327156	-0.363159	-0.290355	-0.346191	-
0.015583 39613 0.015583	-0.347585	-0.363159	-0.290355	-0.346191	-
0.013303					

```
93118
          -0.278869
                         -0.293867
                                        -0.231687
                                                      -0.025662
0.015583
81235
           0.049230
                         -0.363159
                                        -0.290355
                                                      -0.346191
0.015583
       og others 8
                     total og mou 6
                                      total og mou 7
                                                       total og mou 8 \
         -0.013735
5704
                          -0.254350
                                           -0.209855
                                                             1.354152
64892
         -0.013735
                          -0.775847
                                           -0.845314
                                                            -0.422452
         -0.013735
39613
                           0.155438
                                           -0.005238
                                                             0.943279
93118
         -0.013735
                          -0.077192
                                           -0.008896
                                                            -0.300672
81235
         -0.013735
                           3.147976
                                            0.919824
                                                             1.568307
       loc_ic_t2t_mou_6 loc_ic_t2t_mou_7 loc_ic_t2t_mou_8
loc ic t2m mou 6 \
5704
              -0.356975
                                  -0.095026
                                                      0.281846
0.089162
               -0.107944
                                  -0.347607
                                                     -0.187444
64892
0.377903
39613
               0.075275
                                  -0.307553
                                                     -0.130965
0.113096
93118
               2.234897
                                   1.680142
                                                      1.178931
0.874402
81235
               0.238993
                                   0.429108
                                                      0.113725
1.182362
       loc ic t2m mou 7 loc ic t2m mou 8 loc ic t2f mou 6
loc ic t2f mou 7 \
5704
              -0.112790
                                   0.515971
                                                     -0.290528
0.270877
               0.199498
                                   0.240935
64892
                                                     -0.275866
0.257495
39613
               -0.328115
                                   0.017490
                                                     -0.104613
0.247057
93118
               0.796818
                                   0.524427
                                                     -0.043033
0.080535
81235
               0.691388
                                   0.502602
                                                     -0.182615
0.060511
       loc ic t2f mou 8
                          loc ic mou 6
                                        loc ic mou 7
                                                        loc ic mou 8 \
5704
               -0.194257
                             -0.156095
                                            -0.166424
                                                            0.468259
64892
               -0.235146
                                            -0.078726
                              0.172870
                                                            0.045944
39613
               -0.125105
                             -0.056143
                                            -0.419352
                                                           -0.067426
               0.274772
93118
                              1.723994
                                             1.417516
                                                            0.968096
81235
              -0.257696
                              0.923322
                                             0.685320
                                                            0.369639
       std ic t2t mou 6
                          std ic t2t mou 7 std ic t2t mou 8
std ic t2m mou 6 \
5704
              -0.215496
                                  -0.200464
                                                     -0.187265
0.113370
64892
               -0.215496
                                  -0.152024
                                                      0.151031
```

```
2.985768
                                 -0.200464
39613
              -0.215496
                                                   -0.187265
0.190865
93118
              -0.141573
                                 -0.121450
                                                   -0.104402
0.219794
81235
               3.676291
                                  2.238646
                                                    4.587345
0.261982
       std ic t2f mou_7 \
5704
              -0.185210
                                 -0.166335
                                                   -0.135479
0.137327
64892
               2.167834
                                  2.203886
                                                    0.947838
2.883004
              -0.281392
                                 -0.252874
39613
                                                    0.271115
0.137327
93118
              -0.280534
                                 -0.238753
                                                   -0.135479
0.137327
81235
              -0.280902
                                 -0.232114
                                                   -0.135479
0.137327
       std ic t2f mou 8 std ic t2o mou 6 std ic t2o mou 7
std ic t2o mou 8 \
5704
              -0.110642
                                       0.0
                                                         0.0
0.0
64892
               1.082447
                                       0.0
                                                         0.0
0.0
                                       0.0
                                                         0.0
39613
              -0.110642
0.0
93118
              -0.110642
                                       0.0
                                                         0.0
0.0
81235
              -0.110642
                                       0.0
                                                         0.0
0.0
       std ic mou 6
                     std ic mou 7
                                    std ic mou 8
                                                  total ic mou 6 \
                                       -0.\overline{2}336\overline{1}0
5704
          -0.077912
                         -0.265421
                                                        -0.194148
64892
           1.935296
                         1.671948
                                        1.888954
                                                        1.033317
39613
          -0.231969
                         -0.328225
                                       -0.299535
                                                        -0.176324
93118
          -0.250056
                         -0.277357
                                       -0.247586
                                                        1.336498
                         1.225035
81235
           2.151705
                                        2.089855
                                                        1.680641
       total ic mou 7
                       total ic mou 8
                                        spl ic mou 6
                                                      spl ic mou 7 \
5704
            -0.204469
                             0.286255
                                           -0.366516
                                                          -0.089786
64892
             0.758037
                             0.716003
                                           -0.366516
                                                          -0.089786
39613
            -0.522456
                             -0.189128
                                           -0.366516
                                                          -0.089786
93118
             1.048813
                             0.704124
                                           -0.366516
                                                          -0.089786
81235
             1.061830
                             1.050754
                                           -0.366516
                                                          -0.089786
       spl_ic_mou_8 isd_ic_mou_6 isd_ic_mou_7 isd_ic_mou_8
ic others 6 \
```

```
5704
          -0.192624
                         -0.151655
                                         0.285066
                                                       -0.126576
0.099745
64892
          -0.192624
                          0.312051
                                        -0.021464
                                                       -0.126576
0.050008
39613
          -0.192624
                         -0.151655
                                        -0.153778
                                                       -0.126576
0.099745
93118
                         -0.047189
                                        -0.153778
                                                       -0.126576
          -0.192624
0.099745
                         -0.151655
81235
          -0.192624
                                        -0.153778
                                                       -0.126576
0.099745
                                  total rech num 6
       ic others 7
                     ic others 8
                                                      total rech num 7 \
5704
         -0.121704
                       -0.081491
                                          -0.156412
                                                              0.260837
64892
          4.574367
                        0.366640
                                          -0.040029
                                                             -0.680263
39613
         -0.121704
                       -0.011433
                                          -0.738325
                                                             -1.033175
93118
         -0.121704
                       -0.081491
                                          -0.738325
                                                              0.025562
81235
         -0.121704
                       -0.081491
                                           1.472945
                                                              0.025562
       total rech num 8 total rech amt 6 total rech amt 7
total rech amt 8 \
5704
               1.306523
                                   0.087587
                                                     -0.236774
0.817300
64892
               0.054980
                                   0.060452
                                                     -0.688801
0.051360
39613
               -0.695945
                                   0.421336
                                                     -0.518626
1.256352
93118
                                   1.126824
                                                      0.638030
               -0.445637
0.511656
                                   3.764262
81235
               0.555597
                                                      0.688551
3.201396
       max rech amt 6
                        max rech amt 7 max rech amt 8
last day rch amt 6 \
5704
             0.054992
                             -0.173563
                                               0.029739
0.175643
64892
             0.054992
                              0.358167
                                               0.551737
0.598662
39613
             2.412584
                              2.340760
                                               2.555285
3.553548
93118
             1.183544
                              0.753166
                                               0.950911
1.530677
                              0.008745
81235
             0.812513
                                               1.718554
0.175643
       last_day_rch_amt_7 last_day_rch_amt_8 vol_2g_mb_6
vol 2g mb 7
5704
                  0.368450
                                      -0.351085
                                                    3.313695
2.175444
64892
                 -0.350629
                                      -0.626096
                                                    3.855666
6.784445
```

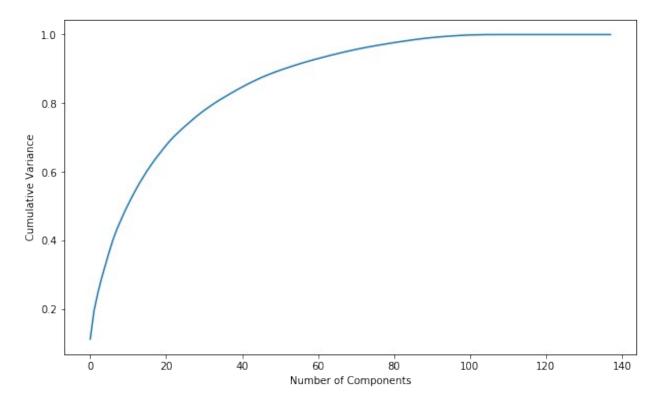
39613	3.4	419926	3.040717	-0.245535	-
0.235847 93118	1.4	493164	-0.626096	-0.245535	-
0.235847 81235 2.782323	-0.3	350629	1.207310	5.471850	
vo 2 monthly	l_2g_mb_8	vol_3g_mb_6 v	ol_3g_mb_7 vo	ol_3g_mb_8	
5704	-0.098306	-0.262491	-0.063995	0.506232	
3.236849 64892 6.720348	4.402555	1.716008	0.276844	0.288491	
39613 0.246650	-0.207939	-0.262491	-0.274601	-0.249913	-
93118 0.246650	-0.207939	-0.262491	-0.274601	-0.249913	-
81235 0.246650	1.989567	-0.262491	-0.274601	-0.249913	-
	onthly 2g 7	monthly 2g 8	sachet 2g 6	sachet_2g_7	
sachet_2g	_8 \	morrerrey_2g_o	Sacric t_2g_0	sachet_zg_/	
5704	-0.251375	-0.232664	0.457379	2.358097	
2.447476 64892 2.447476	3.104207	3.421905	-0.255793	-0.269796	
39613 0.268245	-0.251375	-0.232664	-0.255793	-0.269796	-
93118 0.268245	-0.251375	-0.232664	-0.255793	-0.269796	-
81235 0.410685	-0.251375	-0.232664	1.883722	1.044151	
	nthly_3g_6	monthly_3g_7	monthly_3g_8	sachet_3g_6	
sachet_3g 5704 1 210546	-0.224183	-0.221779	-0.216364	1.315163	
1.219546 64892	-0.224183	-0.221779	-0.216364	-0.141182	
0.136208 39613 0.136208	-0.224183	-0.221779	-0.216364	-0.141182	
93118 0.136208	-0.224183	-0.221779	-0.216364	-0.141182	
81235 2.575301	2.171393	-0.221779	2.201160	2.771508	
sa 5704 64892 39613	chet_3g_8 2.526725 -0.113882 -0.113882	0.225051 0. 0.622516 2.	018023 0.19 423668 2.35	oc_3g jun_vbc 94794 -0.259 57564 5.861 55392 -0.259	366 151

```
93118
         -0.113882 1.742643
                                -0.255884
                                            -0.265392
                                                        -0.259366
          1.206422 -0.244679
                                -0.255884
                                            -0.265392
81235
                                                        -0.259366
       decrease mou action decrease rech num action \
5704
64892
                         1
                                                    1
                         1
                                                    1
39613
                         1
93118
                                                    0
                         1
81235
       decrease_rech_amt_action decrease arpu action
decrease vbc action
                               1
                                                     1
5704
0
64892
                                                     1
39613
                                                     0
                                                     1
93118
                                                     1
81235
0
```

Model with PCA

```
#Import PCA
from sklearn.decomposition import PCA
# Instantiate PCA
pca = PCA(random state=42)
# Fit train set on PCA
pca.fit(X train)
PCA(copy=True, iterated power='auto', n components=None,
random state=42,
    svd solver='auto', tol=0.0, whiten=False)
# Principal components
pca.components_
array([[-7.50315936e-20, 4.16333634e-17, 1.11022302e-16, ...,
        -2.59799614e-02, -2.57740516e-02,
                                          1.40032998e-02],
       [-1.61507486e-19, -5.55111512e-17,
                                           0.00000000e+00, ...,
        -1.16737642e-02, -9.94022864e-03, -1.42598315e-02],
       [ 1.91332162e-19, -2.77555756e-17, 0.00000000e+00, ...,
        -4.18532955e-02, -4.28357226e-02, 2.46812846e-02],
       [-0.00000000e+00, -3.78694731e-02, -3.56427844e-02, ...,
```

```
1.23056947e-16, -4.06575815e-17, -0.00000000e+00],
       [ 0.00000000e+00, 2.32804774e-01,
                                           3.95374959e-02, ...,
         6.41847686e-17, 3.12250226e-17, 8.32667268e-17],
       [ 9.99999199e-01, -3.85782335e-04,
                                          1.19512948e-03, ...,
         1.35525272e-20, 3.11708125e-19, -1.99086624e-17]])
# Cumuliative varinace of the PCs
variance cumu = np.cumsum(pca.explained variance ratio )
print(variance cumu)
[0.11213256 0.19426234 0.24575583 0.28953571 0.32841891 0.36623473
 0.40173361 0.43144425 0.45702167 0.48194328 0.50480575 0.52673812
 0.54724457 0.5670202 0.58530008 0.60304258 0.6190213
                                                         0.63473458
 0.64927873  0.66341423  0.67712828  0.69025011  0.7020618
                                                         0.71278516
 0.72309435 0.73290234 0.74255604 0.75209676 0.76151565 0.77010093
 0.77861315 0.7866115 0.79429496 0.80173555 0.80878909 0.81538157
 0.82193734 0.8283476 0.83472622 0.84089758 0.84687761 0.85280024
 0.85840083 0.86374029 0.86901646 0.87418749 0.87891437 0.88341796
            0.89186057 0.89588256 0.89966074 0.90339384 0.90704071
 0.91060084 0.91411689 0.91752343 0.92076319 0.92395413 0.92705111
 0.93001239 0.93296077 0.93580029 0.93862291 0.94138851 0.9441162
 0.94678675 0.94937767 0.95188405 0.95433786 0.95665036 0.95893735
 0.96116409 0.96323063 0.96526039 0.967203
                                             0.96912626 0.97100138
 0.97284931 0.9746657 0.97639261 0.97806622 0.97972617 0.98133794
 0.98290963 0.98446566 0.98601222 0.98753485 0.98877905 0.98998795
 0.99114751 0.99224606 0.99321228 0.99407803 0.9949224
                                                         0.99573799
 0.99652652 0.99717502 0.99776401 0.99831985 0.99880793 0.99912289
 0.99942656 0.99969174 0.99985313 0.99994737 0.99998103 0.99999839
                                             1.
 0.99999963 0.99999989 1.
                                  1.
                                                         1.
 1.
            1.
                       1.
                                  1.
                                             1.
                                                         1.
 1.
                                  1.
                                             1.
                                                         1.
            1.
                       1.
 1.
            1.
                       1.
                                  1.
                                              1.
                                                         1.
            1.
                       1.
                                  1.
                                              1.
 1.
                                                         1.
# Plotting scree plot
fig = plt.figure(figsize = (10,6))
plt.plot(variance cumu)
plt.xlabel('Number of Components')
plt.ylabel('Cumulative Variance')
Text(0, 0.5, 'Cumulative Variance')
```



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

Performing PCA with 60 components

```
# Importing incremental PCA
from sklearn.decomposition import IncrementalPCA

# Instantiate PCA with 60 components
pca_final = IncrementalPCA(n_components=60)

# Fit and transform the X_train
X_train_pca = pca_final.fit_transform(X_train)
```

Applying transformation on the test set

We are only doing Transform in the test set not the Fit-Transform. Because the Fitting is already done on the train set. So, we just have to do the transformation with the already fitted data on the train set.

```
X_test_pca = pca_final.transform(X_test)
```

Emphasize Sensitivity/Recall than Accuracy

We are more focused on higher Sensitivity/Recall score than the accuracy.

Beacuse we need to care more about churn cases than the not churn cases. The main goal is to reatin the customers, who have the possiblity 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

```
# Importing scikit logistic regression module
from sklearn.linear_model import LogisticRegression
# Impoting metrics
from sklearn import metrics
from sklearn.metrics import confusion_matrix
```

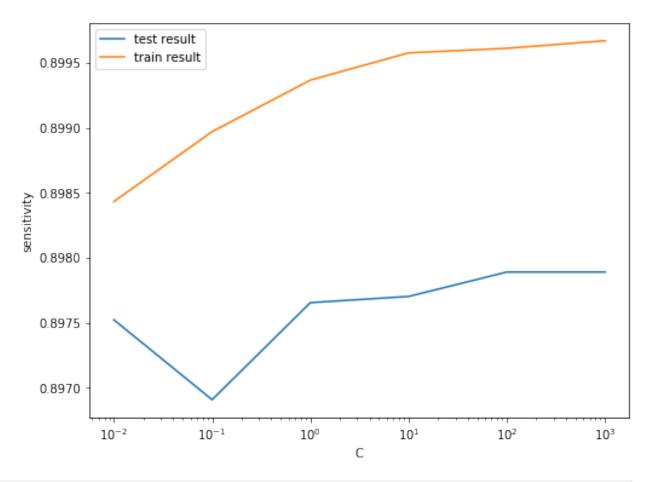
Tuning hyperparameter C

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

```
# Importing libraries for cross validation
from sklearn.model selection import KFold
from sklearn.model selection import cross val score
from sklearn.model selection import GridSearchCV
# Creating KFold object with 5 splits
folds = KFold(n splits=5, shuffle=True, random state=4)
# Specify params
params = \{"C": [0.01, 0.1, 1, 10, 100, 1000]\}
# Specifing score as recall as we are more focused on acheiving the
higher sensitivity than the accuracy
model cv = GridSearchCV(estimator = LogisticRegression(),
                        param_grid = params,
                        scoring= 'recall',
                        cv = folds,
                        verbose = 1,
                        return train score=True)
# Fit the model
model cv.fit(X train pca, y train)
Fitting 5 folds for each of 6 candidates, totalling 30 fits
[Parallel(n jobs=1)]: Using backend SequentialBackend with 1
concurrent workers.
[Parallel(n jobs=1)]: Done 30 out of 30 | elapsed: 21.6s finished
GridSearchCV(cv=KFold(n splits=5, random state=4, shuffle=True),
             error score=nan,
             estimator=LogisticRegression(C=1.0, class weight=None,
dual=False,
                                          fit intercept=True,
```

```
intercept scaling=1,
l1 ratio=None,
                                           max iter=100,
multi class='auto',
                                           n jobs=None, penalty='l2',
                                           random state=None,
solver='lbfgs',
                                           tol=0.0001, verbose=0,
                                           warm start=False),
             iid='deprecated', n jobs=None,
             param_grid={'C': [0.01, 0.1, 1, 10, 100, 1000]},
             pre_dispatch='2*n_jobs', refit=True,
return train score=True,
             scoring='recall', verbose=1)
# results of grid search CV
cv results = pd.DataFrame(model cv.cv results )
cv results
   mean_fit_time std_fit time
                                 mean score time
                                                  std score time
param C \
        0.478627
                      0.060932
                                        0.007600
                                                    1.200167e-03
0.01
                      0.021868
                                        0.006801
                                                    3.999949e-04
1
        0.731842
0.1
2
                      0.008100
                                                    6.325605e-04
        0.743043
                                        0.007000
1
3
        0.754643
                      0.024106
                                        0.007200
                                                    1.469782e-03
10
4
        0.720841
                      0.015716
                                        0.007000
                                                    1.784161e-07
100
                                                    4.899208e-04
        0.719441
                      0.008778
                                        0.006600
1000
                split0 test score split1 test score
        params
split2 test score \
                          0.900071
0 {'C': 0.01}
                                             0.897759
0.895814
   {'C': 0.1}
                          0.898177
                                             0.896359
0.894651
      {'C': 1}
                         0.898650
                                             0.898693
0.895581
     {'C': 10}
                          0.898887
                                             0.898459
0.896744
    {'C': 100}
                          0.899597
                                             0.898226
0.896977
5 {'C': 1000}
                                             0.898226
                         0.899597
0.896977
   split3 test score
                      split4_test_score mean_test_score
```

```
std test score \
            0.906425
                                                  0.897524
                                0.887552
0.006134
            0.905959
                                0.889403
                                                  0.896910
1
0.005390
            0.905028
                                0.890329
                                                  0.897656
0.004783
            0.904562
                                0.889866
                                                  0.897704
0.004719
            0.904330
                                0.890329
                                                  0.897892
0.004528
            0.904330
                                0.890329
                                                  0.897892
0.004528
                     split0 train_score
                                          split1 train_score \
   rank test score
0
                 5
                               0.901116
                                                    0.898256
                 6
1
                               0.901174
                                                    0.898431
2
                 4
                               0.901988
                                                    0.898606
3
                  3
                               0.902511
                                                    0.898956
4
                  1
                               0.902628
                                                    0.898722
5
                  1
                               0.902628
                                                    0.898839
   split2 train score
                        split3 train score
                                             split4 train score \
0
             0.899387
                                  0.895440
                                                       0.897971
1
             0.899270
                                  0.896725
                                                       0.899257
2
             0.898861
                                  0.898184
                                                       0.899199
3
             0.898394
                                  0.898476
                                                       0.899550
4
             0.898569
                                  0.898593
                                                       0.899550
5
             0.898686
                                  0.898593
                                                       0.899608
                      std_train_score
   mean train score
                             0.001861
0
           0.898434
1
           0.898971
                             0.001440
2
           0.899368
                             0.001351
3
           0.899577
                             0.001524
4
                             0.001550
           0.899612
5
           0.899671
                             0.001521
# plot of C versus train and validation scores
plt.figure(figsize=(8, 6))
plt.plot(cv_results['param_C'], cv_results['mean_test_score'])
plt.plot(cv_results['param_C'], cv_results['mean_train_score'])
plt.xlabel('C')
plt.ylabel('sensitivity')
plt.legend(['test result', 'train result'], loc='upper left')
plt.xscale('log')
```



```
# Best score with best C
best_score = model_cv.best_score_
best_C = model_cv.best_params_['C']

print(" The highest test sensitivity is {0} at C = {1}".format(best_score, best_C))

The highest test sensitivity is 0.8978916608693863 at C = 100
```

Logistic regression with optimal C

```
# Instantiate the model with best C
logistic_pca = LogisticRegression(C=best_C)
# Fit the model on the train set
log_pca_model = logistic_pca.fit(X_train_pca, y_train)
```

Prediction on the train set

```
# Predictions on the train set
y_train_pred = log_pca_model.predict(X_train_pca)
```

```
# Confusion matrix
confusion = metrics.confusion matrix(y train, y train pred)
print(confusion)
[[17908 3517]
[ 2154 19271]]
TP = confusion[1,1] # true positive
TN = confusion[0,0] # true negatives
FP = confusion[0,1] # false positives
FN = confusion[1,0] # false negatives
# Accuracy
print("Accuracy:-",metrics.accuracy_score(y_train, y_train_pred))
# Sensitivity
print("Sensitivity:-",TP / float(TP+FN))
# Specificity
print("Specificity:-", TN / float(TN+FP))
Accuracy: - 0.8676546091015169
Sensitivity:- 0.899463243873979
Specificity: - 0.8358459743290548
```

Prediction on the test set

```
# Prediction on the test set
y_test_pred = log_pca_model.predict(X_test_pca)
# Confusion matrix
confusion = metrics.confusion matrix(y test, y test pred)
print(confusion)
[[4452 896]
[ 36 157]]
TP = confusion[1,1] # true positive
TN = confusion[0,0] # true negatives
FP = confusion[0,1] # false positives
FN = confusion[1,0] # false negatives
# Accuracy
print("Accuracy:-",metrics.accuracy_score(y_test, y_test_pred))
# Sensitivity
print("Sensitivity:-",TP / float(TP+FN))
# Specificity
print("Specificity:-", TN / float(TN+FP))
```

```
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

```
# Importing SVC
from sklearn.svm import SVC
```

Hyperparameter tuning

C:- Regularization parameter.

gamma:- Handles non linear classifications.

```
[Parallel(n jobs=1)]: Using backend SequentialBackend with 1
concurrent workers.
[Parallel(n jobs=1)]: Done 36 out of 36 | elapsed: 95.4min finished
GridSearchCV(cv=3, error_score=nan,
             estimator=SVC(C=1.0, break ties=False, cache size=200,
                           class weight=None, coef0=0.0,
                           decision function shape='ovr', degree=3,
                           gamma='scale', kernel='rbf', max_iter=-1,
                           probability=False, random state=None,
shrinking=True,
                           tol=0.001, verbose=False),
             iid='deprecated', n_jobs=None,
             param_grid=[{'C': [1, 10, 100, 1000],
                           'gamma': [0.01, 0.001, 0.0001]}],
             pre dispatch='2*n jobs', refit=True,
return train score=True,
             scoring='accuracy', verbose=1)
# cv results
cv results = pd.DataFrame(model cv.cv results )
cv results
    mean fit time std fit time mean score time std score time
param C \
0
        42.032071
                       0.126973
                                        12.777397
                                                         0.027355
1
1
        53.846413
                       0.407173
                                        17.470999
                                                         0.045588
1
2
        67.914884
                       0.491901
                                        22.857974
                                                         0.176785
1
3
        39.426255
                       0.483549
                                         8.479152
                                                         0.292587
10
4
        47.964410
                       0.411481
                                        14.921520
                                                         0.169904
10
5
        57.650298
                       0.850001
                                        19.027755
                                                         0.122431
10
6
        57.674299
                       6.235520
                                       670.884914
                                                       941.456985
100
        56,490898
                       0.578504
                                        11.192307
                                                         0.035219
100
8
        54.180099
                       0.697826
                                        16.342601
                                                         0.187994
100
9
        93.312670
                       2.057615
                                         3.813552
                                                         0.053278
1000
10
       127.868314
                       3.560431
                                         8.287141
                                                         0.121411
1000
        79.973241
                       1.123954
                                        14.635170
                                                         0.255735
11
1000
```

```
params
                                                    split0 test score
   param gamma
                        {'C': 1, 'gamma': 0.01}
0
           0.01
                                                              0.944903
1
          0.001
                       {'C': 1, 'gamma': 0.001}
                                                              0.883366
2
                     {'C': 1, 'gamma': 0.0001}
         0.0001
                                                              0.858513
3
           0.01
                      {'C': 10, 'gamma': 0.01}
                                                              0.967096
                     {'C': 10, 'gamma': 0.001}
4
          0.001
                                                              0.910459
                    {'C': 10, 'gamma': 0.0001}
5
         0.0001
                                                              0.870414
                    {'C': 100, 'gamma': 0.01}
{'C': 100, 'gamma': 0.001}
6
           0.01
                                                              0.973397
7
          0.001
                                                              0.935662
                   {'C': 100, 'gamma': 0.0001}
{'C': 1000, 'gamma': 0.01}
8
         0.0001
                                                              0.884906
9
           0.01
                                                              0.972277
                   {'C': 1000, 'gamma': 0.001}
10
          0.001
                                                              0.955965
                  {'C': 1000, 'gamma': 0.0001}
11
         0.0001
                                                              0.908499
    split1 test score split2 test score mean test score
std test score
                                    0.940699
                                                       0.942427
               0.941679
0.001796
                                    0.884058
                                                       0.883897
1
              0.884268
0.000385
               0.858433
                                    0.859133
                                                       0.858693
0.000313
3
               0.965413
                                    0.965063
                                                       0.965858
0.000887
              0.911433
                                    0.908073
                                                       0.909988
0.001412
                                    0.872786
                                                       0.870852
               0.869355
0.001434
                                    0.975775
                                                       0.975286
               0.976686
0.001387
7
               0.935518
                                    0.934608
                                                       0.935263
0.000467
              0.886438
                                    0.886718
                                                       0.886021
8
0.000797
               0.977876
                                    0.976336
                                                       0.975496
0.002362
10
               0.955121
                                    0.955472
                                                       0.955519
0.000346
                                    0.907302
11
               0.910943
                                                       0.908915
0.001515
                        split0_train_score
                                              split1 train score
     rank test score
0
                    5
                                   0.947210
                                                          0.947247
1
                   10
                                   0.883813
                                                          0.886757
2
                   12
                                   0.858993
                                                          0.859908
3
                    3
                                   0.975040
                                                          0.973536
4
                    7
                                   0.913709
                                                          0.911891
5
                   11
                                   0.871421
                                                          0.873875
6
                    2
                                                          0.990444
                                   0.991318
```

```
7
                   6
                                 0.941819
                                                      0.942066
8
                   9
                                 0.886368
                                                      0.888893
9
                   1
                                 0.998425
                                                      0.998495
10
                   4
                                 0.965623
                                                      0.965345
                   8
11
                                 0.912868
                                                      0.910806
    split2_train_score
                         mean_train_score std_train_score
0
               0.947702
                                  0.947386
                                                    0.000224
1
                                                    0.001218
               0.885707
                                  0.885426
2
               0.859173
                                  0.859358
                                                    0.000396
3
               0.974306
                                  0.974294
                                                    0.000614
4
               0.912381
                                  0.912660
                                                    0.000768
5
               0.870690
                                  0.871995
                                                    0.001362
6
               0.990198
                                  0.990653
                                                    0.000481
7
               0.941681
                                  0.941855
                                                    0.000159
8
                                  0.887935
                                                    0.001117
               0.888543
9
               0.998495
                                  0.998471
                                                    0.000033
10
               0.966465
                                  0.965811
                                                    0.000476
11
               0.911996
                                  0.911890
                                                    0.000845
```

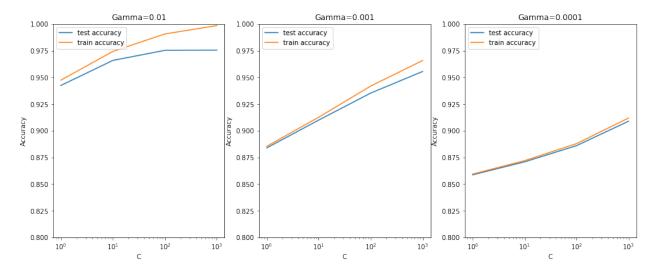
Plotting the accuracy with various C and gamma values

```
# converting C to numeric type for plotting on x-axis
cv results['param C'] = cv results['param C'].astype('int')
# # plotting
plt.figure(figsize=(16,6))
# subplot 1/3
plt.subplot(131)
gamma_01 = cv_results[cv_results['param gamma']==0.01]
plt.plot(gamma_01["param_C"], gamma_01["mean_test_score"])
plt.plot(gamma 01["param C"], gamma 01["mean train score"])
plt.xlabel('C')
plt.ylabel('Accuracy')
plt.title("Gamma=0.01")
plt.ylim([0.80, 1])
plt.legend(['test accuracy', 'train accuracy'], loc='upper left')
plt.xscale('log')
# subplot 2/3
plt.subplot(132)
gamma 001 = cv results[cv results['param gamma']==0.001]
plt.plot(gamma 001["param C"], gamma 001["mean test score"])
plt.plot(gamma 001["param C"], gamma 001["mean train score"])
plt.xlabel('C')
plt.ylabel('Accuracy')
plt.title("Gamma=0.001")
```

```
plt.ylim([0.80, 1])
plt.legend(['test accuracy', 'train accuracy'], loc='upper left')
plt.xscale('log')

# subplot 3/3
plt.subplot(133)
gamma_0001 = cv_results[cv_results['param_gamma']==0.0001]

plt.plot(gamma_0001["param_C"], gamma_0001["mean_test_score"])
plt.plot(gamma_0001["param_C"], gamma_0001["mean_train_score"])
plt.xlabel('C')
plt.xlabel('Accuracy')
plt.title("Gamma=0.0001")
plt.ylim([0.80, 1])
plt.legend(['test accuracy', 'train accuracy'], loc='upper left')
plt.xscale('log')
```



```
# Printing the best score
best_score = model_cv.best_score_
best_hyperparams = model_cv.best_params_

print("The best test score is {0} corresponding to hyperparameters
{1}".format(best_score, best_hyperparams))

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.

Build the model with optimal hyperparameters

```
# Building the model with optimal hyperparameters
svm_pca_model = SVC(C=100, gamma=0.0001, kernel="rbf")
svm_pca_model.fit(X_train_pca, y_train)
SVC(C=100, break_ties=False, cache_size=200, class_weight=None,
coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma=0.0001,
kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

Prediction on the train set

```
# Specificity
print("Specificity:-", TN / float(TN+FP))

Accuracy:- 0.891855309218203
Sensitivity:- 0.9260210035005835
Specificity:- 0.8576896149358226
```

Prediction on the test set

```
# Prediction on the test set
y test pred = svm pca model.predict(X test pca)
# Confusion matrix
confusion = metrics.confusion matrix(y test, y test pred)
print(confusion)
[[4557 791]
[ 36 15711
TP = confusion[1,1] # true positive
TN = confusion[0,0] # true negatives
FP = confusion[0,1] # false positives
FN = confusion[1,0] # false negatives
# Accuracy
print("Accuracy:-",metrics.accuracy_score(y_test, y_test_pred))
# Sensitivity
print("Sensitivity:-",TP / float(TP+FN))
# Specificity
print("Specificity:-", TN / float(TN+FP))
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

```
# Importing decision tree classifier
from sklearn.tree import DecisionTreeClassifier
```

```
Hyperparameter tuning
# Create the parameter grid
param grid = {
    'max depth': range(5, 15, 5),
    'min samples leaf': range(50, 150, 50),
    'min samples split': range(50, 150, 50),
}
# Instantiate the grid search model
dtree = DecisionTreeClassifier()
grid search = GridSearchCV(estimator = dtree,
                           param grid = param grid,
                           scoring= 'recall',
                           cv = 5,
                           verbose = 1)
# Fit the grid search to the data
grid search.fit(X train pca,y train)
Fitting 5 folds for each of 8 candidates, totalling 40 fits
[Parallel(n jobs=1)]: Using backend SequentialBackend with 1
concurrent workers.
[Parallel(n jobs=1)]: Done 40 out of 40 | elapsed: 1.7min finished
GridSearchCV(cv=5, error score=nan,
             estimator=DecisionTreeClassifier(ccp alpha=0.0,
class weight=None,
                                               criterion='gini',
max depth=None,
                                               max features=None,
                                               max leaf nodes=None,
min impurity decrease=0.0,
                                               min impurity split=None,
                                               min samples leaf=1,
                                               min samples split=2,
min weight fraction leaf=0.0,
                                               presort='deprecated',
                                               random state=None,
                                               splitter='best'),
             iid='deprecated', n jobs=None,
```

```
param grid={'max depth': range(5, 15, 5),
                          'min samples leaf': range(50, 150, 50),
                          'min samples split': range(50, 150, 50)},
             pre dispatch='2*n jobs', refit=True,
return train score=False,
             scoring='recall', verbose=1)
# cv results
cv results = pd.DataFrame(grid search.cv results )
cv results
   mean fit time
                  std fit time
                                 mean score time
                                                   std score time
0
        1.948911
                       0.023829
                                         0.008601
                                                           0.00049
1
                       0.010277
                                         0.008400
                                                           0.00049
        1.941111
2
        1.925310
                       0.003188
                                         0.008400
                                                           0.00049
3
        1.925510
                       0.002871
                                         0.008600
                                                           0.00049
4
        3.343991
                       0.015459
                                         0.008601
                                                           0.00049
5
        3.370193
                       0.083993
                                         0.008601
                                                           0.00049
6
        3.199783
                       0.044874
                                         0.008801
                                                          0.00040
7
                       0.025967
                                         0.008801
        3.186582
                                                          0.00040
  param max depth param min samples leaf param min samples split
0
                5
                                        50
                                                                 50
                5
1
                                       50
                                                                100
                5
2
                                       100
                                                                 50
                5
3
                                       100
                                                                100
4
               10
                                        50
                                                                 50
5
                                       50
               10
                                                                100
6
               10
                                       100
                                                                 50
7
               10
                                       100
                                                                100
                                                params
split0 test score
0 {'max depth': 5, 'min samples leaf': 50, 'min ...
0.862310
1 {'max depth': 5, 'min samples leaf': 50, 'min ...
0.862310
 {'max_depth': 5, 'min_samples_leaf': 100, 'min...
0.858110
   {'max_depth': 5, 'min_samples_leaf': 100, 'min...
0.858110
4 {'max depth': 10, 'min samples leaf': 50, 'min...
0.886114
   {'max depth': 10, 'min samples leaf': 50, 'min...
0.886114
  {'max depth': 10, 'min samples leaf': 100, 'mi...
0.889615
   {'max depth': 10, 'min samples leaf': 100, 'mi...
0.889615
```

```
split1 test score split2 test score split3 test score
split4 test score \
            0.855776
                                0.878413
                                                    0.875379
0.855309
            0.855776
                                0.878413
                                                    0.875379
0.855309
            0.855309
                                                    0.869078
                                0.875846
0.849008
                                0.875846
            0.855309
                                                    0.869078
0.849008
            0.894516
                                0.903851
                                                    0.905484
0.913652
            0.894516
                                0.903851
                                                    0.905484
0.912485
            0.869778
                                0.875613
                                                    0.891949
0.884247
            0.871179
                                0.875613
                                                    0.891949
0.883781
   mean test score
                    std test score
                                     rank test score
0
          0.865438
                           0.009725
                                                    5
                                                    5
1
          0.865438
                           0.009725
                                                    7
2
          0.861470
                           0.009686
                                                    7
3
          0.861470
                           0.009686
4
                                                    1
          0.900723
                           0.009503
5
                                                    2
          0.900490
                           0.009192
6
          0.882240
                                                    4
                           0.008389
7
          0.882427
                           0.007964
                                                    3
# Printing the optimal sensitivity score and hyperparameters
print("Best sensitivity:-", grid_search.best_score_)
print(grid search.best estimator )
Best sensitivity:- 0.9007234539089849
DecisionTreeClassifier(ccp alpha=0.0, class weight=None,
criterion='gini',
                        max depth=10, max features=None,
max leaf nodes=None,
                        min impurity decrease=0.0,
min impurity split=None,
                        min samples leaf=50, min samples split=50,
                        min_weight_fraction_leaf=0.0,
presort='deprecated',
                        random state=None, splitter='best')
```

Model with optimal hyperparameters

Prediction on the train set

```
# Predictions on the train set
y train pred = dt pca model.predict(X train pca)
# Confusion matrix
confusion = metrics.confusion matrix(y train, y train pred)
print(confusion)
[[18913 2512]
[ 1763 19662]]
TP = confusion[1,1] # true positive
TN = confusion[0,0] # true negatives
FP = confusion[0,1] # false positives
FN = confusion[1,0] # false negatives
# Accuracy
print("Accuracy:-",metrics.accuracy_score(y_train, y train pred))
# Sensitivity
print("Sensitivity:-",TP / float(TP+FN))
# Specificity
print("Specificity:-", TN / float(TN+FP))
Accuracy: - 0.9002333722287048
Sensitivity: - 0.9177129521586931
Specificity: - 0.8827537922987164
```

Prediction on the test set

```
# Prediction on the test set
y_test_pred = dt_pca_model.predict(X_test_pca)
```

```
# Confusion matrix
confusion = metrics.confusion matrix(y test, y test pred)
print(confusion)
[[4632 716]
[ 58 135]]
TP = confusion[1,1] # true positive
TN = confusion[0,0] # true negatives
FP = confusion[0,1] # false positives
FN = confusion[1,0] # false negatives
# Accuracy
print("Accuracy:-",metrics.accuracy_score(y_test, y_test_pred))
# Sensitivity
print("Sensitivity:-",TP / float(TP+FN))
# Specificity
print("Specificity:-", TN / float(TN+FP))
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 Sesitivity 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

```
# Importing random forest classifier
from sklearn.ensemble import RandomForestClassifier
```

```
Hyperparameter tuning
```

```
param_grid = {
    'max_depth': range(5,10,5),
    'min_samples_leaf': range(50, 150, 50),
    'min_samples_split': range(50, 150, 50),
```

```
'n estimators': [100,200,300],
    'max features': [10, 20]
}
# Create a based model
rf = RandomForestClassifier()
# Instantiate the grid search model
grid search = GridSearchCV(estimator = rf,
                           param grid = param grid,
                           cv = 3,
                           n jobs = -1,
                           verbose = 1,
                           return_train_score=True)
# Fit the model
grid search.fit(X train pca, y train)
Fitting 3 folds for each of 24 candidates, totalling 72 fits
[Parallel(n jobs=-1)]: Using backend LokyBackend with 4 concurrent
workers.
[Parallel(n_jobs=-1)]: Done 42 tasks
                                           | elapsed: 13.6min
[Parallel(n jobs=-1)]: Done 72 out of 72 | elapsed: 132.9min
finished
GridSearchCV(cv=3, error score=nan,
             estimator=RandomForestClassifier(bootstrap=True,
ccp_alpha=0.0,
                                               class weight=None,
                                               criterion='gini',
max depth=None,
                                               max features='auto'.
                                               max leaf nodes=None,
                                               max samples=None,
min impurity decrease=0.0,
                                               min_impurity_split=None,
                                               min samples leaf=1,
                                               min samples split=2,
min_weight_fraction_leaf=0.0,
                                               n estimators=100,
n jobs=None,
                                               oob score=False,
                                               random state=None,
verbose=0,
                                              warm start=False),
             iid='deprecated', n jobs=-1,
             param grid={'max depth': range(5, 10, 5), 'max features':
[10, 20],
                         'min samples leaf': range(50, 150, 50),
```

Model with optimal hyperparameters

```
# model with the best hyperparameters
rfc model = RandomForestClassifier(bootstrap=True,
                             max depth=5,
                             min_samples_leaf=50,
                             min samples split=100,
                             max features=20,
                             n estimators=300)
# Fit the model
rfc model.fit(X train pca, y train)
RandomForestClassifier(bootstrap=True, ccp alpha=0.0,
class weight=None,
                       criterion='gini', max depth=5, max features=20,
                       max leaf nodes=None, max samples=None,
                       min impurity decrease=0.0,
min impurity split=None,
                       min_samples_leaf=50, min_samples_split=100,
                       min weight fraction leaf=0.0, n estimators=300,
                       n jobs=None, oob score=False,
random state=None,
                       verbose=0, warm start=False)
```

Prediction on the train set

```
# Predictions on the train set
y_train_pred = rfc_model.predict(X_train_pca)
# Confusion matrix
confusion = metrics.confusion_matrix(y_train, y_train_pred)
print(confusion)

[[17363     4062]
      [ 2419     19006]]
```

```
TP = confusion[1,1] # true positive
TN = confusion[0,0] # true negatives
FP = confusion[0,1] # false positives
FN = confusion[1,0] # false negatives

# Accuracy
print("Accuracy:-",metrics.accuracy_score(y_train, y_train_pred))

# Sensitivity
print("Sensitivity:-",TP / float(TP+FN))

# Specificity
print("Specificity:-", TN / float(TN+FP))

Accuracy:- 0.8487514585764294
Sensitivity:- 0.8870945157526254
Specificity:- 0.8104084014002334
```

Prediction on the test set

```
# Prediction on the test set
y_test_pred = rfc_model.predict(X_test_pca)
# Confusion matrix
confusion = metrics.confusion matrix(y test, y test pred)
print(confusion)
[[4294 1054]
[ 47 146]]
TP = confusion[1,1] # true positive
TN = confusion[0,0] # true negatives
FP = confusion[0,1] # false positives
FN = confusion[1,0] # false negatives
# Accuracy
print("Accuracy:-",metrics.accuracy_score(y_test, y_test_pred))
# Sensitivity
print("Sensitivity:-",TP / float(TP+FN))
# Specificity
print("Specificity:-", TN / float(TN+FP))
Accuracy: - 0.8012994044396319
Sensitivity: - 0.7564766839378239
Specificity: - 0.8029169783096485
```

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 Sesitivity 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 acheiving 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 apporx 85%.

Without PCA

Logistic regression with No PCA

```
##### Importing stats model
import statsmodels.api as sm
# Instantiate the model
# Adding the constant to X train
log_no_pca = sm.GLM(y_train,(sm.add_constant(X_train)),
family=sm.families.Binomial())
# Fit the model
log no pca = log no pca.fit().summary()
# Summary
log_no_pca
<class 'statsmodels.iolib.summary.Summary'>
                 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, 16 May 2020 Deviance:

nan

Time: 17:56:38 Pearson chi2:

3.70e+05

No. Iterations: 100

Covariance Type: nonrobust

covariance i	уре.	Holli obus			
	0.975]	coef	std err	z	P> z
const -8799.145 loc_og_t2o_m -8.39e-05	8652.765 ou 8.53e-05	-73.1903 7.163e-07	4452.100 4.32e-05	-0.016 0.017	0.987 0.987
std_og_t2o_m -2.75e-05 loc_ic_t2o_m	2.81e-05	2.572e-07 1.441e-06	1.42e-05 6.09e-05	0.018 0.024	0.986 0.981
-0.000 arpu_6 -0.192	0.000	-0.0338	0.081	-0.418	0.676
arpu_7 -0.083 arpu_8 -0.124	0.254 0.306	0.0855	0.086 0.110	0.994 0.828	0.320 0.407
onnet_mou_6	2.516	15.5129 -4.3251	3.573 1.817	4.342	0.000 0.017
-7.886 onnet_mou_8 -1.225	-0.764 5.930	2.3528	1.825	1.289	0.197
offnet_mou_7	1.674	15.0874 -1.7629	3.361 1.721	4.489	0.000 0.306
-5.136 offnet_mou_8 -4.240	3.141	-0.5496	1.883	-0.292 4.473	0.770
<pre>roam_ic_mou_ 0.091 roam_ic_mou0.112</pre>	0.233	0.1622	0.036 0.052	-0.189	0.000 0.850
roam_ic_mou_	8 0.290	0.2041 -5.1505	0.044 1.131	4.663 -4.554	0.000 0.000
_ 3					

-7.367 -2.934	0 0055	0 474	1 067	0.000
roam_og_mou_7 -0.044 1.815	0.8855	0.474	1.867	0.062
roam_og_mou_8	0.0927	0.531	0.175	0.861
-0.948 1.133 loc og t2t mou 6	-3302.8216	656.377	-5.032	0.000
$-45\overline{8}9.\overline{2}97 - 2\overline{0}16.346$				
loc_og_t2t_mou_7 -2806.968 -142.267	-1474.6175	679.783	-2.169	0.030
loc_og_t2t_mou_8	5516.1251	628.160	8.781	0.000
4284.953 6747.297 loc_og_t2m_mou_6	-3342.4429	664.131	-5.033	0.000
$-46\overline{4}4.\overline{1}16 - 2\overline{0}40.770$	1202 1070	641 104	2 171	0.020
loc_og_t2m_mou_7 -2648.649 -135.567	-1392.1079	641.104	-2.171	0.030
loc_og_t2m_mou_8 4573.677 7201.089	5887.3829	670.271	8.784	0.000
loc_og_t2f_mou_6	-285.2245	56.710	-5.030	0.000
-396.373 -174.076 loc og t2f mou 7	- 123 . 0165	56.677	-2.171	0.030
$-23\overline{4}.\overline{10}1$ $-\overline{1}1.933$				
loc_og_t2f_mou_8 378.584 596.214	487.3991	55.519	8.779	0.000
loc_og_t2c_mou_6	0.0433	0.022	1.982	0.048
0.000	0.0099	0.021	0.463	0.643
-0.032 0.052				
loc_og_t2c_mou_8 0.023	0.0673	0.023	2.988	0.003
loc_og_mou_6	3756.2132	1269.036	2.960	0.003
1268.947 6243.479 loc og mou 7	5686.5575	1330.512	4.274	0.000
3078.802 8294.313				
loc_og_mou_8 -2913.835 2382.258	-265.7885	1351.069	-0.197	0.844
std_og_t2t_mou_6	-1.309e+04	1867.084	-7.011	0.000
-1.67e+04 -9430.445 std_og_t2t_mou_7	-9674.1364	1822.022	-5.310	0.000
$-1.\overline{3}2e+04$ $-6\overline{1}03.040$	5054 0112	1510 000	2 077	0.000
std_og_t2t_mou_8 2895.093 8814.530	5854.8113	1510.088	3.877	0.000
std_og_t2m_mou_6	-1.214e+04	1732.071	-7.011	0.000
-1.55e+04 -8749.312 std_og_t2m_mou_7	-9438.8725	1777.374	-5.311	0.000
-1.29e+04 -5955.284 std og t2m mou 8	5966.0664	1538.126	3.879	0.000
2951.394 8980.739	3900.0004	1330.120	5.079	0.000
std_og_t2f_mou_6 -326.757 -184.099	-255.4281	36.393	-7.019	0.000
- 320.737 - 104.099				

std_og_t2f_mou_7 -292.502 -134.689	-213.5957	40.259	-5.306	0.000
std_og_t2f_mou_8	142.4571	36.758	3.876	0.000
70.412 214.502 std_og_t2c_mou_6	-3.536e-06	0.000	-0.018	0.986
-0.000 0.000 std_og_t2c_mou_7	-2.353e-06	0.000	-0.014	0.989
-0.000 0.000 std_og_t2c_mou_8	-2.486e-06	0.000	-0.017	0.986
-0.000 0.000 std og mou 6	1.446e+04	2966.600	4.875	0.000
8646.469 2.03e+04	111100101	23001000	11075	0.000
std_og_mou_7 1.5e+04	2.105e+04	3103.817	6.783	0.000
std_og_mou_8 2390.393 1.32e+04	7815.2524	2767.836	2.824	0.005
isd_og_mou_6	-51.5636	29.686	-1.737	0.082
-109.747 6.620 isd_og_mou_7	94.2299	27.788	3.391	0.001
39.767 148.693 isd_og_mou_8	320.5622	34.133	9.392	0.000
253.663 387.462 spl_og_mou_6	-83.1552	47.782	-1.740	0.082
-176.805 10.495 spl_og_mou_7	229.8600	67.719	3.394	0.001
97.134 362.586				
spl_og_mou_8 414.206 632.102	523.1539	55.587	9.411	0.000
og_others_6 -21.496 1.312	-10.0916	5.818	-1.734	0.083
og_others_7 6.034 25.255	15.6443	4.903	3.191	0.001
og_others_8	-5276.7831	3.24e+05	-0.016	0.987
-6.41e+05 6.3e+05 total_og_mou_6	3406.4402	1971.608	1.728	0.084
-457.840 7270.720 total og mou 7	-7829.5225	2307.277	-3.393	0.001
$-1.24\overline{e}+0\overline{4}$ -3307.343		23071277	3.333	0.001
total_og_mou_8 -2.29e+04 -1.5e+04	-1.894e+04	2011.752	-9.413	0.000
loc_ic_t2t_mou_6 -1259.382 315.443	-471.9694	401.748	-1.175	0.240
loc_ic_t2t_mou_7	2043.4379	441.193	4.632	0.000
1178.715 2908.161 loc_ic_t2t_mou_8	6411.9241	417.700	15.351	0.000
5593.247 7230.601 loc_ic_t2m_mou_6	-662.3378	563.889	-1.175	0.240
-1767.540 442.865 loc_ic_t2m_mou_7	2751.5586	593.997	4.632	0.000

1587.346 3915.772 loc ic t2m mou 8	9239.7305	601.928	15.350	0.000
8059.973 1.04e+04	3233.7303	001.520	13.330	0.000
loc_ic_t2f_mou_6 -349.055 87.312	-130.8717	111.320	-1.176	0.240
loc_ic_t2f_mou_7	595.9951	128.711	4.630	0.000
343.725 848.265 loc_ic_t2f_mou_8 1531.426 1979.826	1755.6262	114.390	15.348	0.000
loc_ic_mou_6	-1472.7581	1056.667	-1.394	0.163
-3543.788 598.272 loc_ic_mou_7	-2703.9336	1115.928	-2.423	0.015
-4891.113 -516.754 loc_ic_mou_8	3460.7823	1136.224	3.046	0.002
1233.824 5687.740 std_ic_t2t_mou_6	-2047.5989	316.623	-6.467	0.000
-2668.169 -1427.028 std_ic_t2t_mou_7	-414.5246	317.544	-1.305	0.192
-1036.900 207.851 std_ic_t2t_mou_8	-551.5337	227.041	-2.429	0.015
-996.526 -106.541 std ic t2m mou 6	-2117.6774	327.457	-6.467	0.000
-2759.481 -1475.874 std ic t2m mou 7	-425.3418	325.663	-1.306	0.192
-1063.630 212.946 std ic t2m mou 8	-844.5218	347.926	-2.427	0.015
$-15\overline{2}6.\overline{4}45 - \overline{1}62.599$				
std_ic_t2f_mou_6 -475.242 -254.134	-364.6880	56.406	-6.465	0.000
std_ic_t2f_mou_7 -199.347	-79.5903	61.102	-1.303	0.193
std_ic_t2f_mou_8 -250.338 -27.314	-138.8260	56.895	-2.440	0.015
std_ic_t2o_mou_6 -5.67e-05 5.55e-05	-5.826e-07	2.86e-05	-0.020	0.984
std_ic_t2o_mou_7 -0.000 0.000	1.092e-06	7.95e-05	0.014	0.989
std_ic_t2o_mou_8 -0.000 0.000	1.37e-06	8.38e-05	0.016	0.987
std_ic_mou_6	1980.5760	602.710	3.286	0.001
799.287 3161.865 std_ic_mou_7	1297.3541	611.724	2.121	0.034
98.398 2496.310 std_ic_mou_8	8343.2863	569.153	14.659	0.000
7227.768 9458.805 total_ic_mou_6	2863.2928	942.847	3.037	0.002
1015.348 4711.238 total_ic_mou_7 -3515.061 437.167	-1538.9468	1008.240	-1.526	0.127

total_ic_mou_8	-1.982e+04	1035.030	-19.153	0.000
-2.19e+04 -1.78e+04 spl_ic_mou_6	-1.5369	0.562	-2.733	0.006
-2.639 -0.435 spl_ic_mou_7	0.5833	0.518	1.127	0.260
-0.431 1.598 spl_ic_mou_8	5.2078	0.295	17.671	0.000
4.630 5.785				
isd_ic_mou_6 -796.259 -171.704	-483.9815	159.328	-3.038	0.002
isd_ic_mou_7 -77.816 626.482	274.3329	179.671	1.527	0.127
isd_ic_mou_8	3507.8222	183.152	19.153	0.000
3148.851 3866.794 ic others 6	-81.0624	26.650	-3.042	0.002
-133.295 -28.829 ic others 7	42.4590	27.835	1.525	0.127
$-1\overline{2}.096$ 97.014				
ic_others_8 493.723 606.506	550.1144	28.772	19.120	0.000
total_rech_num_6	0.0224	0.035	0.638	0.523
total_rech_num_7	0.0726	0.040	1.804	0.071
-0.006	-0.6403	0.041	-15.682	0.000
-0.720 -0.560				
total_rech_amt_6 0.452 0.774	0.6131	0.082	7.465	0.000
total_rech_amt_7 -0.375 -0.060	-0.2171	0.080	-2.701	0.007
total_rech_amt_8	0.2182	0.114	1.913	0.056
-0.005 0.442 max_rech_amt_6	-0.2237	0.037	-6.053	0.000
-0.296 -0.151 max_rech_amt_7	-0.0587	0.036	-1.645	0.100
-0.129 0.011				
max_rech_amt_8 0.062	0.1475	0.043	3.398	0.001
last_day_rch_amt_6 -0.233 -0.119	-0.1756	0.029	-6.043	0.000
last_day_rch_amt_7	0.0028	0.029	0.098	0.922
-0.054 0.059 last_day_rch_amt_8	-0.5102	0.033	-15.449	0.000
-0.575 -0.445 vol 2g mb 6	0.1398	0.030	4.724	0.000
$0.0\overline{8}2^{\circ}$ 0.198				
vol_2g_mb_7 -0.033 0.093	0.0299	0.032	0.927	0.354
vol_2g_mb_8	0.0882	0.034	2.580	0.010

0.021 0.155				
vol_3g_mb_6	0.3625	0.044	8.158	0.000
0.275 0.450	0.3023	0.0	0.150	0.000
vol_3g_mb_7	0.4089	0.056	7.289	0.000
0.299 0.519				
vol_3g_mb_8	-0.1838	0.068	-2.700	0.007
-0.317 -0.050 monthly_2g_6	-0.6068	0.045	-13.514	0.000
-0.695 -0.519	-0.0000	0.045	-13.314	0.000
monthly_2g_7	-0.4095	0.042	-9.834	0.000
-0.491 -0.328				
monthly_2g_8	-0.6419	0.059	-10.961	0.000
-0.757 -0.527	0.0000	0 001	0 770	0 430
sachet_2g_6	-0.0239	0.031	-0.773	0.439
-0.085 0.037 sachet_2g_7	-0.2143	0.033	-6.464	0.000
-0.279 -0.149	-0.2143	0.055	-0.404	0.000
sachet_2g_8	-0.2391	0.032	-7.513	0.000
-0.301 -0.177				
monthly_3g_6	-0.3220	0.046	-6.989	0.000
-0.412 -0.232				
monthly_3g_7	-0.5808	0.052	-11.100	0.000
-0.683 -0.478 monthly_3g_8	-0.8649	0.078	-11.083	0.000
-1.018 -0.712	-0.0049	0.076	-11.005	0.000
sachet_3g_6	-0.0281	0.032	-0.871	0.384
-0.091 0.035				
sachet_3g_7	-0.0829	0.042	-1.964	0.050
-0.166 -0.000				
sachet_3g_8	-0.1553	0.048	-3.222	0.001
-0.250 -0.061 aon	-0.1564	0.022	-7.269	0.000
-0.199 -0.114	-0.1504	0.022	-7.209	0.000
aug_vbc_3g	-0.1965	0.057	-3.441	0.001
-0.308 -0.085				
jul_vbc_3g	-0.0522	0.047	-1.118	0.264
-0.144 0.039	0. 2264	0.047	F 007	0.000
jun_vbc_3g	0.2364	0.047	5.007	0.000
0.144 0.329 decrease mou action	-0.4989	0.053	-9.461	0.000
-0.602 -0.396	-0.4909	0.055	-9.401	0.000
decrease rech num action	-1.0229	0.048	-21.429	0.000
-1.116 -0.929				
<pre>decrease_rech_amt_action</pre>	-0.3065	0.065	-4.720	0.000
-0.434 -0.179	0 1-0-	0.00=	0 =01	0.007
decrease_arpu_action	-0.1797	0.067	-2.701	0.007
-0.310 -0.049 decrease vbc action	-1.7537	0.130	-13.538	0.000
-2.008 -1.500	-1./33/	0.130	- 13,330	0.000
21000 -11500				

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).

Feature Selection Using RFE

```
# Importing logistic regression from sklearn
from sklearn.linear_model import LogisticRegression
# Intantiate the logistic regression
logreg = LogisticRegression()
```

RFE with 15 columns

```
# Importing RFE
from sklearn.feature selection import RFE
# Intantiate RFE with 15 columns
rfe = RFE(logreg, 15)
# Fit the rfe model with train set
rfe = rfe.fit(X train, y train)
# RFE selected columns
rfe cols = X train.columns[rfe.support ]
print(rfe cols)
Index(['offnet mou 7', 'offnet mou 8', 'roam og mou 8',
'std og t2m mou 8',
       isd og mou 8', 'og others 7', 'og others 8',
'loc_ic_t2f_mou_8',
       'loc_ic_mou_8', 'std_ic_t2f_mou_8', 'ic_others_8',
'total rech_num_8',
       'monthly_2g_8', 'monthly_3g_8', 'decrease_vbc_action'],
      dtype='object')
```

Model-1 with RFE selected columns

```
# Adding constant to X_train
X_train_sm_1 = sm.add_constant(X_train[rfe_cols])
```

```
#Instantiate the model
log_no_pca_1 = sm.GLM(y_train, X_train_sm_1,
family=sm.families.Binomial())
# Fit the model
log_no_pca_1 = log_no_pca_1.fit()
log no pca 1.summary()
<class 'statsmodels.iolib.summary.Summary'>
                Generalized Linear Model Regression Results
                               churn No. Observations:
Dep. Variable:
42850
                                       Df Residuals:
Model:
                                 GLM
42834
Model Family:
                            Binomial Df Model:
15
Link Function:
                               logit Scale:
1.0000
Method:
                                IRLS Log-Likelihood:
nan
Date:
                    Sat, 16 May 2020
                                       Deviance:
nan
Time:
                            18:04:18 Pearson chi2:
4.49e+06
No. Iterations:
                                 100
Covariance Type:
                           nonrobust
                         coef std err
                                                         P>|z|
           0.9751
                      -58.6610
                                4419.624
                                             -0.013
                                                         0.989 -
const
8720.965 8603.643
offnet mou 7
                       0.6096
                                   0.026
                                             23,449
                                                         0.000
0.559
           0.661
                                            -30.548
offnet mou 8
                       -3.2532
                                   0.106
                                                         0.000
           -3.045
3.462
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
                                                         0.000
                                             26.101
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	-1.1313	0.002	-1.502	0.107	
og others 8	-4191.9652	3.22e+05	-0.013	0.990	_
6.35e+05 6.27e+0			0.025	0.000	
loc ic t2f mou 8	-0.7547	0.072	-10.487	0.000	-
$0.8\overline{9}6$ $-0.\overline{6}14$					
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	1 4010	0 100	11 205	0.000	
ic_others_8	-1.4913	0.132	-11.305	0.000	-
1.750 -1.233	-0.4840	0.018	-26.977	0.000	
total_rech_num_8 0.519 -0.449	-0.4040	0.010	-20.977	0.000	-
monthly 2g 8	-0.9031	0.043	-20.851	0.000	_
0.988 -0.818	0.5051	0.043	20.031	0.000	
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	-
$1.451 - \overline{1.165}$					

Checking VIFs

```
# Check for the VIF values of the feature variables.
from statsmodels.stats.outliers influence import
variance inflation factor
# Create a dataframe that will contain the names of all the feature
variables and their respective VIFs
vif = pd.DataFrame()
vif['Features'] = X train[rfe cols].columns
vif['VIF'] = [variance_inflation_factor(X_train[rfe_cols].values, i)
for i in range(X_train[rfe_cols].shape[1])]
vif['VIF'] = round(vif['VIF'], 2)
vif = vif.sort_values(by = "VIF", ascending = False)
vif
               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
       loc_ic_t2f_mou_8
7
                        1.21
11
       total rech num 8 1.19
2
          roam og mou 8 1.16
    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

```
# Removing og_others_8 column
log_cols = rfe_cols.to_list()
log_cols.remove('og_others_8')
print(log_cols)

['offnet_mou_7', 'offnet_mou_8', 'roam_og_mou_8', 'std_og_t2m_mou_8',
'isd_og_mou_8', 'og_others_7', 'loc_ic_t2f_mou_8', 'loc_ic_mou_8',
'std_ic_t2f_mou_8', 'ic_others_8', 'total_rech_num_8', 'monthly_2g_8',
'monthly_3g_8', 'decrease_vbc_action']
```

Model-2

Building the model after removing og_others_8 variable.

```
# Adding constant to X train
X train sm 2 = sm.add constant(X train[log cols])
#Instantiate the model
log_no_pca_2 = sm.GLM(y_train, X_train_sm_2,
family=sm.families.Binomial())
# Fit the model
log_no_pca_2 = log_no_pca_2.fit()
log no pca 2.summary()
<class 'statsmodels.iolib.summary.Summary'>
                 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, 16 May 2020	Deviance:
30068.		
Time:	18:06:36	Pearson chi2:

4.51e+06

0.00

No. Iterations: 11

Covariance Type: nonrobust

	========	:=======			=======	====
[0.025	0.975]	coef	std err	Z	P> z	
const		-1.1052	0.031	-35.342	0.000	_
1.167	-1.044	111032	0.031	331312	01000	
offnet mo		0.6081	0.026	23.427	0.000	
0.557	0.659					
offnet_mo	u_8	-3.2557	0.106	-30.603	0.000	-
3.464	-3.047					
roam_og_m		1.2491	0.031	39.747	0.000	
1.188						
std_og_t2		2.4428	0.093	26.146	0.000	
2.260		1 0000	0 100	F F00	0.000	
isd_og_mo		-1.0982	0.196	-5.590	0.000	-
1.483		-1.8793	0.818	-2.299	0.022	
og_others 3.482		-1.0/93	0.010	-2.299	0.022	_
loc ic t2	*	-0.7548	0.072	-10.491	0.000	_
0.896		-0.7540	0.072	-10.431	0.000	
loc_ic_mo		-1.9714	0.066	-30.058	0.000	-
2.100			0.000		0.000	
std ic t2	f mou 8	-0.8020	0.075	-10.727	0.000	-
$0.9\overline{4}9^{-}$						
ic_others	_8	-1.4871	0.132	-11.278	0.000	-
1.746	-1.229					
total_rec		-0.4864	0.018	-27.146	0.000	-
0.522	-0.451					
monthly_2		-0.9066	0.043	-20.866	0.000	-
0.992	-0.821	0.0000	0.042	22.700	0.000	
monthly_3		-0.9862	0.043	-22.700	0.000	-
1.071	-0.901	1 2007	0.072	-17.994	0.000	
1.452	vbc_action -1.167	-1.3097	0.073	-17.994	0.000	-
		========	=======		=======	====

Checking VIF for Model-2

```
# Create a dataframe that will contain the names of all the feature
variables and their respective VIFs
vif = pd.DataFrame()
vif['Features'] = X_train[log cols].columns
vif['VIF'] = [variance inflation factor(X train[log cols].values, i)
for i in range(X train[log cols].shape[1])]
vif['VIF'] = round(vif['VIF'], 2)
vif = vif.sort values(by = "VIF", ascending = False)
vif
               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
           monthly 3g 8
                        1.06
12
11
           monthly 2g 8 1.05
       std ic t2f mou 8 1.02
8
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.

```
# Removing offnet_mou_8 column
log_cols.remove('offnet_mou_8')
```

Model-3

Model after removing offnet_mou_8 column.

```
# Adding constant to X_train
X_train_sm_3 = sm.add_constant(X_train[log_cols])
#Instantiate the model
log_no_pca_3 = sm.GLM(y_train, X_train_sm_3,
family=sm.families.Binomial())
# Fit the model
log_no_pca_3 = log_no_pca_3.fit()
log_no_pca_3.summary()
```

```
<class 'statsmodels.iolib.summary.Summary'>
                  Generalized Linear Model Regression Results
Dep. Variable:
                                   churn
                                           No. Observations:
42850
                                           Df Residuals:
Model:
                                     GLM
42836
Model Family:
                               Binomial
                                           Df Model:
13
Link Function:
                                   logit
                                           Scale:
1.0000
Method:
                                           Log-Likelihood:
                                    IRLS
-15720.
                       Sat, 16 May 2020
                                           Deviance:
Date:
31440.
Time:
                               18:07:30
                                           Pearson chi2:
3.92e+06
No. Iterations:
                                      11
Covariance Type:
                              nonrobust
                            coef std err
                                                               P>|z|
[0.025]
             0.975]
                                       0.032
                                                 -37.536
                                                               0.000
const
                         -1.2058
1.269
            -1.143
offnet mou 7
                          0.3665
                                       0.022
                                                  16.456
                                                               0.000
0.323
             0.410
                          0.7135
                                       0.024
                                                  29.260
                                                               0.000
roam_og_mou_8
             0.761
0.666
std_og_t2m_mou_8
                         -0.2474
                                       0.022
                                                 -11.238
                                                               0.000
            -0.204
0.291
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
                                                               0.000
                                                  -9.532
0.856
            -0.\overline{5}64
loc ic mou 8
                         -3.3287
                                       0.057
                                                 -58.130
                                                               0.000
3.441
            -3.216
                                       0.078
                                                 -12.181
                                                               0.000
std ic t2f mou 8
                         -0.9503
            -0.\overline{7}97
1.103
ic others 8
                         -1.5131
                                       0.129
                                                 -11.771
                                                               0.000
1.765
            -1.261
```

total_rech_num_8 0.540 -0.472	-0.5060	0.018	-28.808	0.000	-
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			========	========	====

VIF Model-3

```
vif = pd.DataFrame()
vif['Features'] = X train[log cols].columns
vif['VIF'] = [variance inflation factor(X train[log cols].values, i)
for i in range(X_train[log_cols].shape[1])]
vif['VIF'] = round(vif['VIF'], 2)
vif = vif.sort values(by = "VIF", ascending = False)
vif
               Features
                        VIF
       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
       std_ic_t2f_mou 8
7
                        1.02
3
           isd og mou 8
                         1.01
8
            ic others 8 1.01
            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

```
# Getting the predicted value on the train set
y_train_pred_no_pca = log_no_pca_3.predict(X_train_sm_3)
y_train_pred_no_pca.head()

0     2.687411e-01
1     7.047483e-02
```

```
2 8.024370e-02
3 3.439222e-03
4 5.253815e-19
dtype: float64
```

Creating a dataframe with the actual churn and the predicted probabilities

```
y train pred final = pd.DataFrame({'churn':y train.values,
'churn prob':y train pred no pca.values})
#Assigning Customer ID for each record for better readblity
#CustID is the index of each record.
y train pred final['CustID'] = y train pred final.index
y train pred final.head()
            churn prob
                        CustID
   churn
0
         2.687411e-01
       0
1
                             1
       0 7.047483e-02
2
                             2
       0 8.024370e-02
3
        3.439222e-03
                             3
4
       0 5.253815e-19
                             4
```

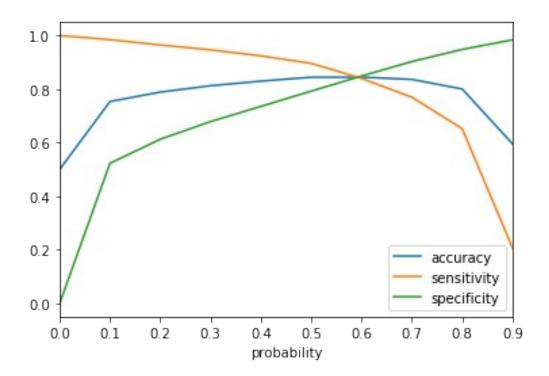
Finding Optimal Probablity Cutoff Point

```
# Creating columns for different probablity cutoffs
prob_cutoff = [float(p/10) for p in range(10)]
for i in prob cutoff:
    y_train_pred_final[i] =
y_{train\_pred\_final['churn\_prob'].map(lambda x : 1 if x > i else 0)
y train pred final.head()
   churn churn prob CustID 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7
0.8
0
       0 2.687411e-01
0
       0 7.047483e-02
1
                             1
                                                                      0
0
2
       0 8.024370e-02
0
3
       0 3.439222e-03
                                       0
                                            0
                                                       0
                                                            0
                                                                      0
0
4
       0 5.253815e-19
                                       0
0
   0.9
0
     0
1
     0
2
     0
```

```
3 0
4 0
```

Now let's calculate the accuracy sensitivity and specificity for various probability cutoffs.

```
# Creating a dataframe
cutoff df = pd.DataFrame(columns=['probability', 'accuracy',
'sensitivity', 'specificity'])
for i in prob cutoff:
    cm1 = metrics.confusion matrix(y train pred final['churn'],
y train pred final[i] )
    total1=sum(sum(cm1))
    accuracy = (cm1[0,0]+cm1[1,1])/total1
    speci = cm1[0,0]/(cm1[0,0]+cm1[0,1])
    sensi = cm1[1,1]/(cm1[1,0]+cm1[1,1])
    cutoff df.loc[i] =[ i ,accuracy,sensi,speci]
print(cutoff df)
     probability accuracy
                                         specificity
                            sensitivity
0.0
             0.0 0.500000
                               1.000000
                                            0.000000
0.1
             0.1 0.753629
                               0.984411
                                            0.522847
0.2
             0.2
                 0.788751
                               0.964714
                                            0.612789
0.3
             0.3 0.812509
                               0.946371
                                            0.678646
0.4
             0.4 0.829638
                               0.923874
                                            0.735403
0.5
             0.5 0.844131
                               0.895823
                                            0.792439
0.6
             0.6 0.844271
                               0.839860
                                            0.848681
                               0.769522
0.7
             0.7 0.836173
                                            0.902824
0.8
             0.8 0.800163
                               0.652275
                                            0.948051
             0.9 0.595426
                               0.207001
                                            0.983851
0.9
# Plotting accuracy, sensitivity and specificity for different
probabilities.
cutoff df.plot('probability',
['accuracy','sensitivity','specificity'])
plt.show()
```



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.

```
# Creating a column with name "predicted", which is the predicted
value for 0.5 cutoff
y train pred final['predicted'] =
y_train_pred_final['churn_prob'].map(lambda x: 1 if x > 0.5 else 0)
y train pred final.head()
   churn
          churn prob
                        CustID
                                           0.2
                                                0.3
                                                     0.4
                                                           0.5
                                                                0.6
                                                                     0.7
                                 0.0
                                      0.1
0.8
         2.687411e-01
0
0
1
         7.047483e-02
                              1
                                                                       0
0
2
          8.024370e-02
       0
                                                                       0
0
3
         3.439222e-03
                                                                       0
```

```
0
      0 5.253815e-19
                         4 1 0 0 0 0 0 0
4
                                                            0
0
  0.9
      predicted
0
    0
1
             0
    0
2
             0
    0
3
    0
             0
4
             0
    0
```

Metrics

```
# Confusion metrics
confusion = metrics.confusion matrix(y train pred final['churn'],
y train pred final['predicted'])
print(confusion)
[[16978 4447]
[ 2232 19193]]
TP = confusion[1,1] # true positive
TN = confusion[0,0] # true negatives
FP = confusion[0,1] # false positives
FN = confusion[1,0] # false negatives
# Accuracy
print("Accuracy:-",metrics.accuracy_score(y_train_pred_final['churn'],
y_train_pred_final['predicted']))
# Sensitivity
print("Sensitivity:-",TP / float(TP+FN))
# Specificity
print("Specificity:-", TN / float(TN+FP))
Accuracy: - 0.8441306884480747
Sensitivity: - 0.8958226371061844
Specificity: - 0.792438739789965
```

We have got good accuracy, sensitivity and specificity on the train set prediction.

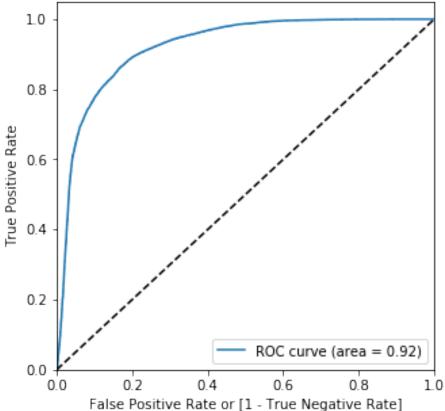
Plotting the ROC Curve (Trade off between sensitivity & specificity)

```
plt.plot( fpr, tpr, label='ROC curve (area = %0.2f)' % auc_score )
  plt.plot([0, 1], [0, 1], 'k--')
  plt.xlim([0.0, 1.0])
  plt.ylim([0.0, 1.05])
  plt.xlabel('False Positive Rate or [1 - True Negative Rate]')
  plt.ylabel('True Positive Rate')
  plt.title('Receiver operating characteristic example')
  plt.legend(loc="lower right")
  plt.show()

return None

draw_roc(y_train_pred_final['churn'],
  y_train_pred_final['churn_prob'])
```





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

Testing the model on the test set

```
# Taking a copy of the test set
X_test_log = X_test.copy()
```

```
# Taking only the columns, which are selected in the train set after
removing insignificant and multicollinear variables
X_test_log = X_test_log[log_cols]

# Adding constant on the test set
X_test_sm = sm.add_constant(X_test_log)
```

Predictions on the test set with final model

```
# Predict on the test set
y test pred = log no pca 3.predict(X test sm)
y test pred.head()
5704
         0.034015
64892
         0.000578
39613
         0.513564
93118
         0.020480
81235
         0.034115
dtype: float64
# Converting y test pred to a dataframe because y test pred is an
array
y pred 1 = pd.DataFrame(y test pred)
y pred 1.head()
5704
       0.034015
64892 0.000578
39613 0.513564
93118 0.020480
81235 0.034115
# Convetting y test to a dataframe
y test df = pd.DataFrame(y test)
y_test_df.head()
       churn
5704
64892
           0
39613
           0
           0
93118
81235
           0
# Putting index to Customer ID
y_test_df['CustID'] = y_test_df.index
# Removing index form the both dataframes for merging them side by
y pred 1.reset index(drop=True, inplace=True)
y test df.reset index(drop=True, inplace=True)
```

```
# Appending y pred 1 and y test df
y test pred final = pd.concat([y test df, y pred 1], axis=1)
y test pred final.head()
   churn CustID
0
            5704 0.034015
       0
1
       0
           64892 0.000578
2
       0
           39613 0.513564
3
       0
           93118 0.020480
       0
           81235 0.034115
# Renaming the '0' column as churn probablity
y test pred final = y test pred final.rename(columns={0:'churn prob'})
# Rearranging the columns
y test pred final =
y test pred final.reindex axis(['CustID','churn','churn prob'],
axis=1)
y_test_pred_final.head()
   CustID churn churn prob
0
     5704
               0
                    0.034015
1
    64892
               0
                    0.000578
2
               0
                    0.513564
    39613
3
    93118
               0
                    0.020480
4 81235
               0
                    0.034115
# In the test set using probablity cutoff 0.5, what we got in the
train set
y test pred final['test predicted'] =
y test pred final['churn prob'].map(lambda x: 1 if x > 0.5 else 0)
y test pred final.head()
   CustID churn churn prob
                              test predicted
0
     5704
               0
                    0.034015
                                           0
               0
                                           0
1
    64892
                    0.000578
                                           1
    39613
               0
                    0.513564
3
    93118
               0
                    0.020480
                                           0
    81235
               0
                    0.034115
                                           0
```

Metrics

```
# Confusion matrix
confusion = metrics.confusion_matrix(y_test_pred_final['churn'],
y_test_pred_final['test_predicted'])
print(confusion)

[[4190 1158]
       [ 34 159]]
```

```
TP = confusion[1,1] # true positive
TN = confusion[0,0] # true negatives
FP = confusion[0,1] # false positives
FN = confusion[1,0] # false negatives

# Accuracy
print("Accuracy:-",metrics.accuracy_score(y_test_pred_final['churn'],
y_test_pred_final['test_predicted']))

# Sensitivity
print("Sensitivity:-",TP / float(TP+FN))

# Specificity
print("Specificity:-", TN / float(TN+FP))

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 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

Variables	Coefficients
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.:-

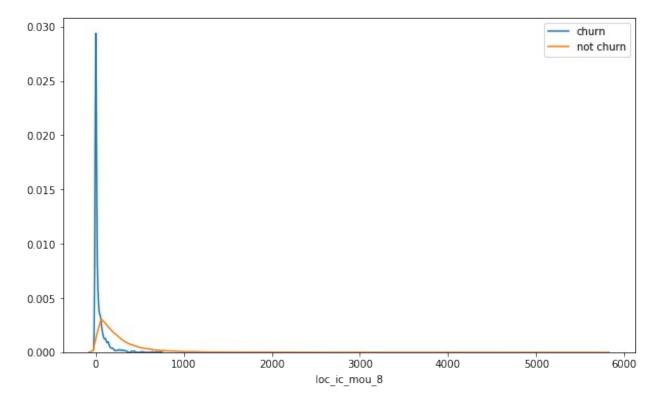
If the local incoming minutes of usage (loc_ic_mou_8) is lesser in the month of August than any other month, then there is a higher chance that the customer is likely to churn.

Recomendations

- 1. 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).
- 2. Target the customers, whose outgoing others charge in July and incoming others on August are less.
- 3. 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.
- 4. Cutomers, whose monthly 3G recharge in August is more, are likely to be churned.
- 5. 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.
- 6. Cutomers decreasing monthly 2g usage for August are most probable to churn.
- 7. Customers having decreasing incoming minutes of usage for operators T to fixed lines of T for August are more likely to churn.
- 8. 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.

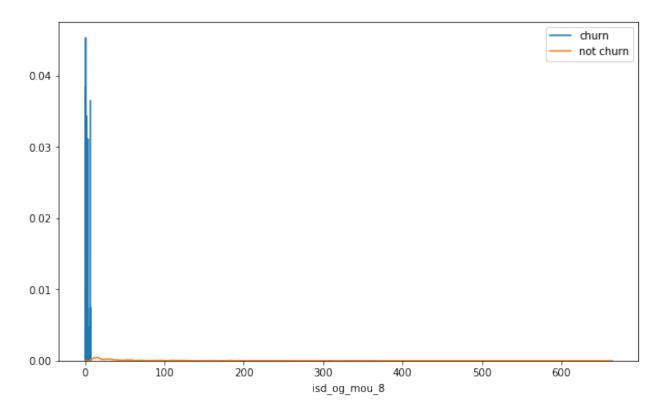
Plots of important predictors for churn and non churn customers

```
# Plotting loc_ic_mou_8 predictor for churn and not churn customers
fig = plt.figure(figsize=(10,6))
sns.distplot(data_churn['loc_ic_mou_8'],label='churn',hist=False)
sns.distplot(data_non_churn['loc_ic_mou_8'],label='not
churn',hist=False)
plt.show()
```



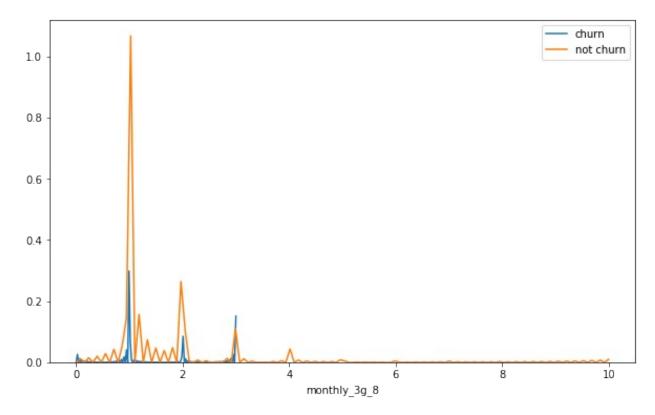
We can see that for the churn customers the minutes of usage for the month of August is mostly populated on the lower side than the non churn customers.

```
# Plotting isd_og_mou_8 predictor for churn and not churn customers
fig = plt.figure(figsize=(10,6))
sns.distplot(data_churn['isd_og_mou_8'],label='churn',hist=False)
sns.distplot(data_non_churn['isd_og_mou_8'],label='not
churn',hist=False)
plt.show()
```



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 onther hand for the non churn customers it is little more than the churn customers.

```
# Plotting monthly_3g_8 predictor for churn and not churn customers
fig = plt.figure(figsize=(10,6))
sns.distplot(data_churn['monthly_3g_8'],label='churn',hist=False)
sns.distplot(data_non_churn['monthly_3g_8'],label='not
churn',hist=False)
plt.show()
```



The number of mothly 3g data for August for the churn customers are very much populated aroud 1, whereas of non churn customers it spreaded accross various numbers.

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