Jio Internship Report

Name: Ananya
Team: Network parameters and other insights
Duration: Mid-may to mid-July

I was part of the 5G team. I got to learn about the following things here: Multivariate Time series analysis, clustering and forecasting.

The first thing I did was **applied a VAR model to the data of multiple Jio Centres and forecasted the per cent customer churn rate in a year.**

New things I encountered during the study:
1. Multivariate time series
2. Grangers Causality test
3. working with uneven time series
4. Resampling and interpolating time series

Multivariate time series is the time series data where we can observe multiple dependent variables determining an outcome variable and each dependent variable affecting each other too with the time, hence first forecasting of those dependent variable values for future and then determining the dependent variable value.

In easy words, a Multivariate Time Series **consists of more than one time-dependent variable** and each variable depends not only on its past values but also has some dependency on other variables.

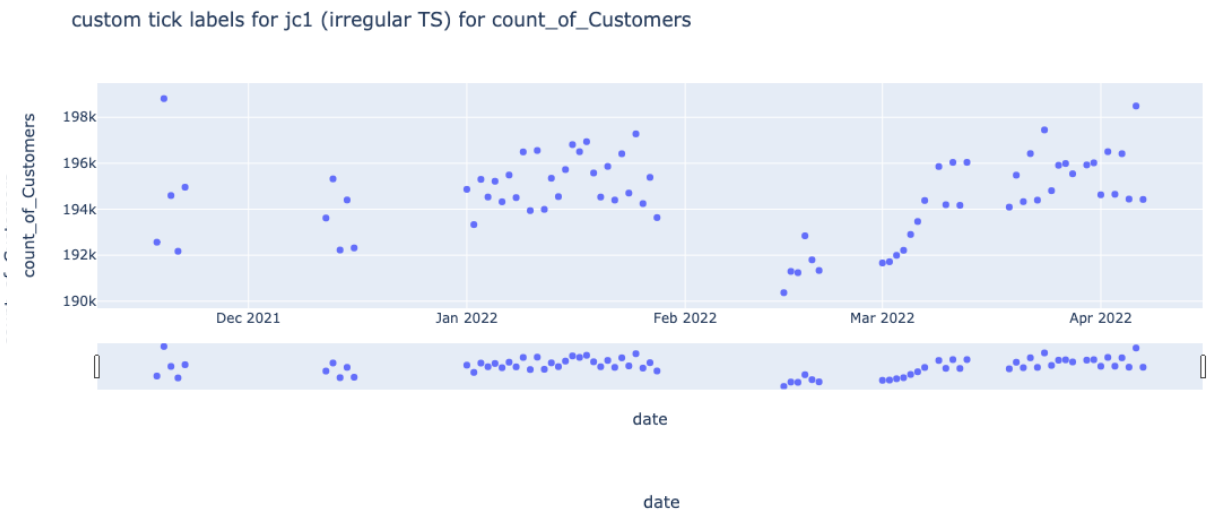
date	count_of_Customers	churn_customers	coverage_percent	high_speed_percent	voice_percent	sum_bad_coverage_duration	sum_coverage_duration	sum_highspeed_duration	sum_bad_highspeed_duration	sum-voice_duration	sum_voice_bad_duration
2021-11-18	192566.0	234.0	7.0	8.0	12.0	389387437.0	5.206397e+09	6.947939e+09	538529923.0	725922041.0	86054800.0
2021-11-19	198813.0	130.0	7.0	9.0	11.0	364685110.0	5.341813e+09	7.113288e+09	623412799.0	720366911.0	82334387.0
2021-11-20	194597.0	270.0	7.0	8.0	11.0	397396552.0	5.444283e+09	7.257249e+09	592900678.0	680063171.0	77947235.0
2021-11-21	192172.0	258.0	7.0	9.0	12.0	391782219.0	5.260777e+09	6.969109e+09	625604793.0	726734948.0	85405824.0
2021-11-22	194959.0	358.0	7.0	9.0	11.0	390147382.0	5.357297e+09	7.151672e+09	617374433.0	719014096.0	82094715.0
...	...	...	...	...	...	...	...	...	...	...	...
2022-04-03	194651.0	230.0	7.0	8.0	12.0	391392660.0	5.410592e+09	7.133312e+09	559852651.0	724127101.0	83901400.0
2022-04-04	196416.0	222.0	7.0	8.0	11.0	361311732.0	5.417333e+09	7.146420e+09	578235264.0	756033813.0	85662559.0
2022-04-05	194442.0	198.0	7.0	8.0	12.0	386516640.0	5.364516e+09	7.122623e+09	544331062.0	716713005.0	82457073.0
2022-04-06	198491.0	298.0	7.0	9.0	11.0	357792407.0	5.363675e+09	7.098185e+09	636399688.0	744284987.0	83704474.0
2022-04-07	194426.0	230.0	7.0	8.0	12.0	399509883.0	5.405632e+09	7.177638e+09	556974659.0	700787689.0	80918103.0

75 rows × 11 columns

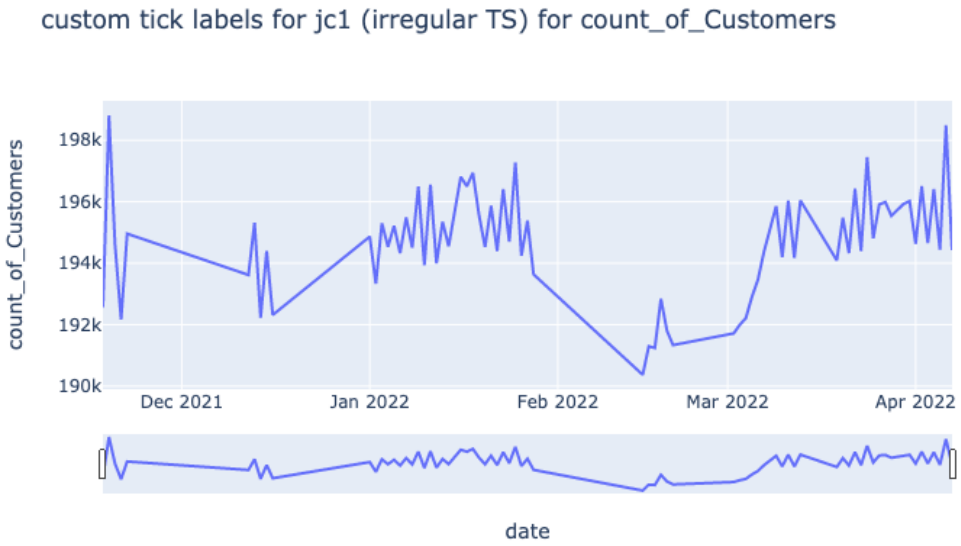
pandas dataframe

Here the data is unevenly spaced.

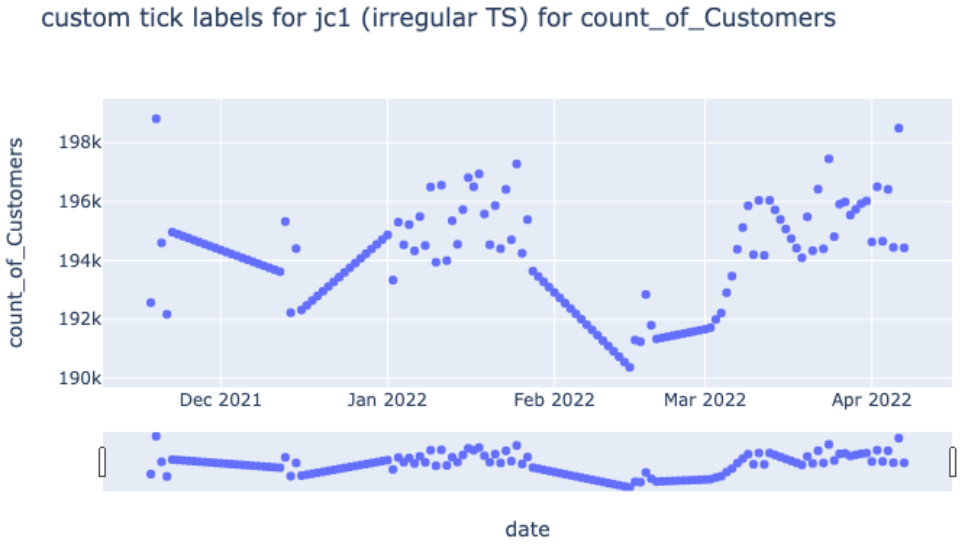
I plotted line and scatter plots using the Plotly library. In the scatter plot, it is visible that data for some dates are missing. Hence first have to make the time series even. Hence applied resampling (in particular upsampling) day-wise followed by linear interpolation. Linear interpolation is a method of curve fitting using linear polynomials to construct new data points within the range of a discrete set of known data points.



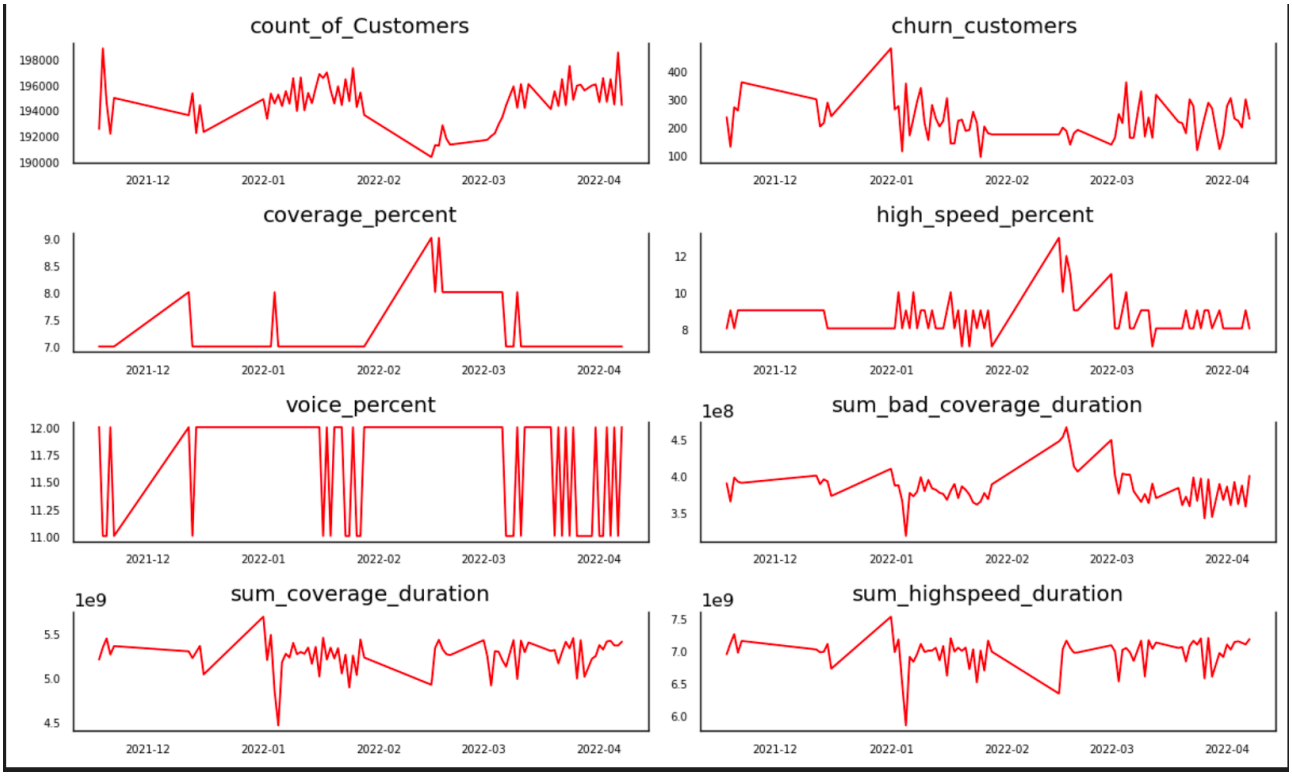
Before interpolation: Scatter Plot



After interpolation: line plot



After interpolation: scatter plot



After interpolation: The line plot for all parameters

In order to apply VAR on Time series, first, we have to check if our time series is stationary or not. A stationary time series is **one whose properties do not depend on the time at which the series is observed**. Thus, time series with trends, or with seasonality, are not stationary — the trend and seasonality will affect the value of the time series at different times.

We can check if a time series is stationary or not with the help of Augmented Dickey Fuller test (ADF Test), a statistical test. In this test we check for the p-value, if p-value > 0.5, then our time series is non-stationary and if p-value < 0.5, then our time series is stationary. By the first differencing method, I made the time series stationary. The first difference of a time series is **the series of changes from one period to the next**. If Y_t denotes the value of the time series Y at period t, then the first difference of Y at period t is equal to $Y_t - Y_{t-1}$.

The Granger causality test is **a statistical hypothesis test for determining whether one time series is useful in forecasting another**. If the probability value is less than any α level, then the hypothesis would be rejected at that level.

Then I applied the VAR model on the data and predicted for the value of the independent parameters count_of_customers, coverage_percent, high_speed_percentage

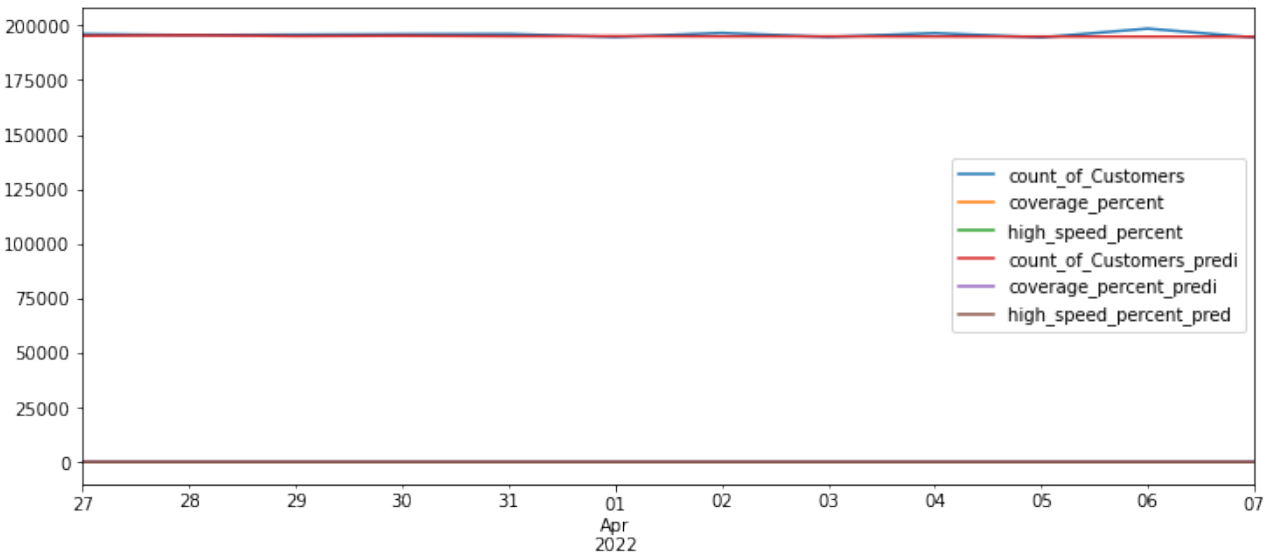
$$Y_{1,t} = \alpha_1 + \beta_{11,1} Y_{1,t-1} + \beta_{12,1} Y_{2,t-1} + \epsilon_{1,t}$$
$$Y_{2,t} = \alpha_2 + \beta_{21,1} Y_{1,t-1} + \beta_{22,1} Y_{2,t-1} + \epsilon_{2,t}$$

VAR model

	count_of_Customers_predi	coverage_percent_predi	high_speed_percent_pred
2022-03-27	195207.730742	7.046022	8.215505
2022-03-28	195449.574571	7.016602	8.533211
2022-03-29	195049.586170	7.018631	8.193923
2022-03-30	195214.637343	7.019251	8.348519
2022-03-31	194999.100650	7.022842	8.214755
2022-04-01	195051.756659	7.026797	8.286936
2022-04-02	194947.037782	7.033825	8.239367
2022-04-03	194949.480604	7.039707	8.277546
2022-04-04	194889.482224	7.047380	8.265215
2022-04-05	194875.270809	7.054480	8.289396
2022-04-06	194834.816335	7.062228	8.291880
2022-04-07	194814.947715	7.069684	8.309908

predicted values

The error I got from using the VAR model is around 0.6%.



Graph for predicted and actual value comparison

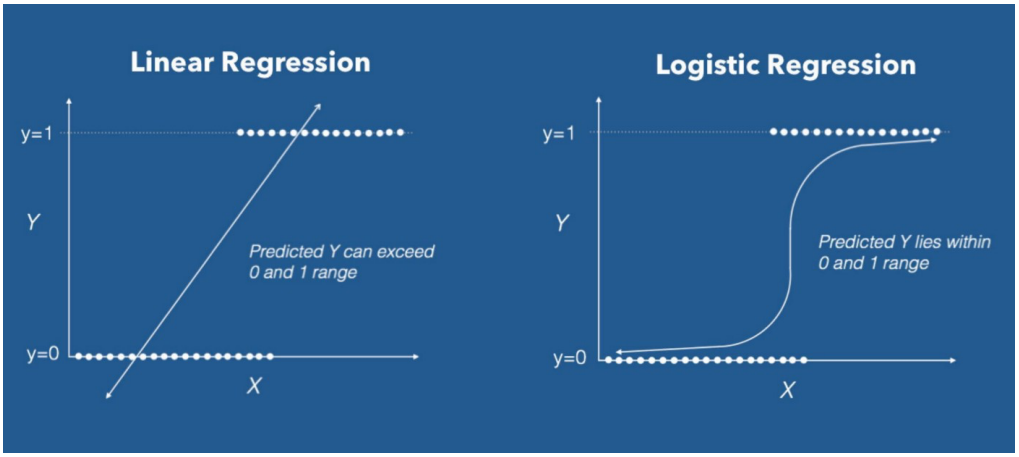
The second thing that I did was **clustering of data.** I was having health card data. It contained multiple columns. We had to cluster the data of customers. The basis of our clustering was the tendency of a particular cluster to convert into a 5G device per month.

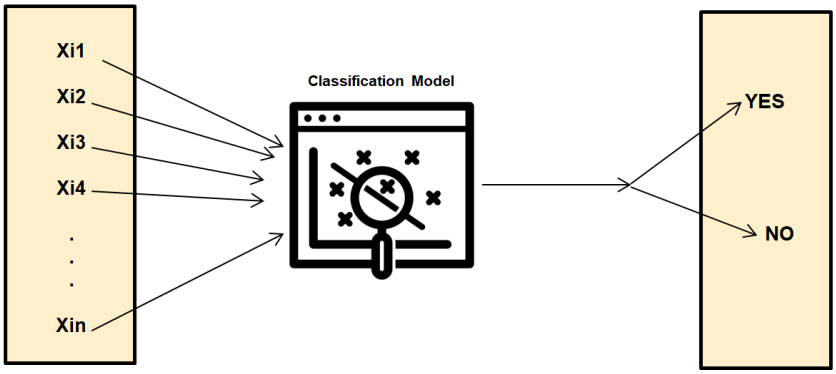
Since there were many attributes in the data, we first decided on the parameters that affect customers' conversion to a 5G device. And by heuristics and statistics, these parameters were found: price of device, age of device, data consumption of the user, TAC code of the device and IMSI number.

Firstly by going with normal clustering, The results were not so accurate since all the parameters didn't have the same contribution for a customer's conversion to a 5G device.

In order to find the significance level of the parameter, I first applied logistic regression. I first wrote a query to find if the customer converted to 5G since we knew the TAC number of the devices in the month of January and February, if the TAC number was changed to a 5G TAC

number, we drew the conclusion that user has converted to a 5G device. And now the data was of binary classification type with parameters such as price of device, age of device, data consumption of the user being the independent variable and the user converted to 5G device is a dependent variable. In the end, by applying the logistic regression I got the coefficients of the function which I thought was the level of significance for the parameters.





After some research, I found out about feature engineering which helped to know about the significance level of each parameter.

And then apply the clustering algorithm on the data.

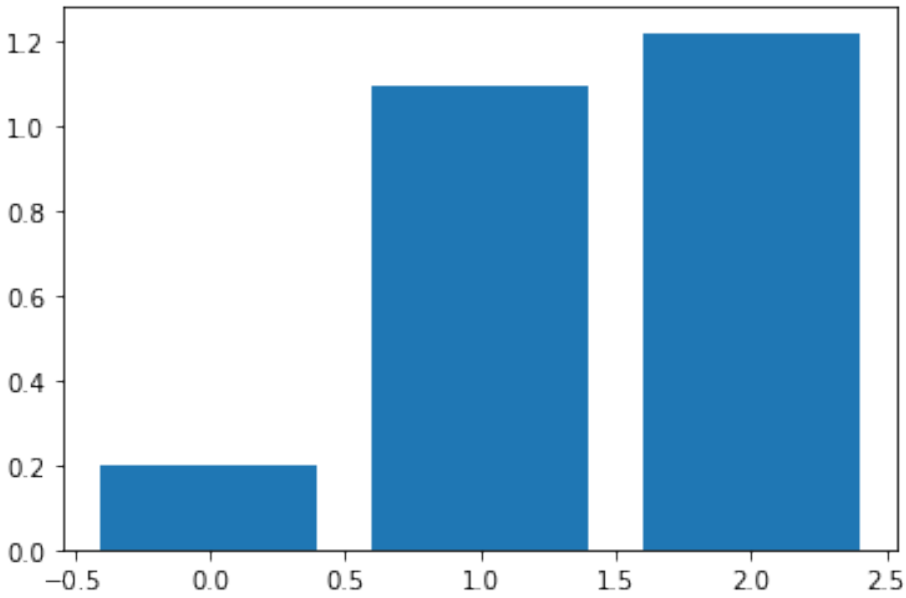
Since I was not provided with the data, I made a dummy dataset for me having 40 rows.

	msisdn	price	Age	data	changedto_5g_device
0	1	7000	1	0.91	0
1	2	50000	2	1.01	0
2	3	15000	5	1.30	1
3	4	10000	2	0.80	0
4	5	20000	1	2.70	0
5	6	30000	2	0.60	0
6	7	40000	4	0.50	0
7	8	7000	8	0.90	1
8	9	10000	2	1.80	1
9	10	7000	4	0.50	1
10	11	50000	4	1.60	1
11	12	15000	6	0.30	0
12	13	10000	9	0.40	0
13	14	20000	4	0.90	1
14	15	30000	5	1.40	1
15	16	40000	3	2.00	1
16	17	7000	2	1.60	0
17	18	20000	4	1.90	1

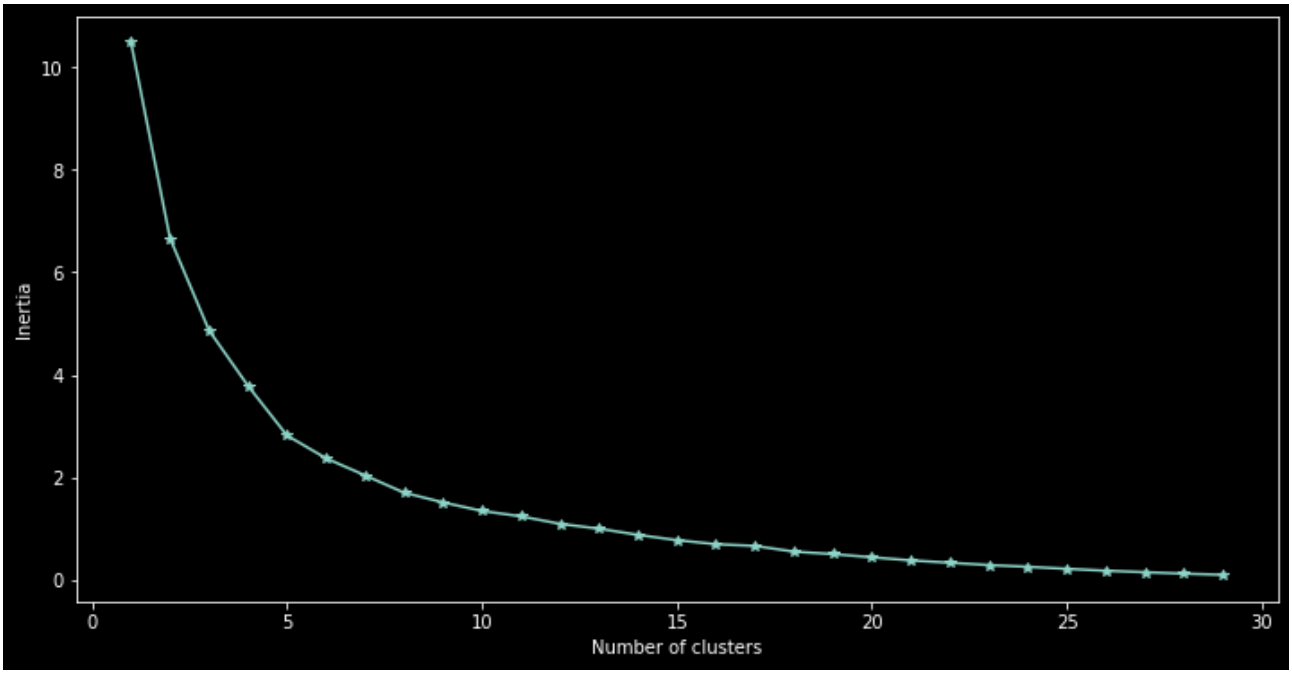
Firstly I normalised the data column-wise and not row-wise. And then found the score of the independent parameters. In the case of dummy-dataset, the following were the results:

	msisdn	price	Age	data	changedto_5g_device
0	1	0.000000	0.000	0.260000	0
1	2	1.000000	0.125	0.288571	0
2	3	0.186047	0.500	0.371429	1
3	4	0.069767	0.125	0.228571	0
4	5	0.302326	0.000	0.771429	0
5	6	0.534884	0.125	0.171429	0
6	7	0.767442	0.375	0.142857	0
7	8	0.000000	0.875	0.257143	1
8	9	0.069767	0.125	0.514286	1
9	10	0.000000	0.375	0.142857	1
10	11	1.000000	0.375	0.457143	1
11	12	0.186047	0.625	0.085714	0
12	13	0.069767	1.000	0.114286	0
13	14	0.302326	0.375	0.257143	1
14	15	0.534884	0.500	0.400000	1
15	16	0.767442	0.250	0.571429	1
16	17	0.000000	0.125	0.457143	0
17	18	0.302326	0.375	0.542857	1

```
Feature: 0, Score: 0.20128
Feature: 1, Score: 1.09347
Feature: 2, Score: 1.22012
```

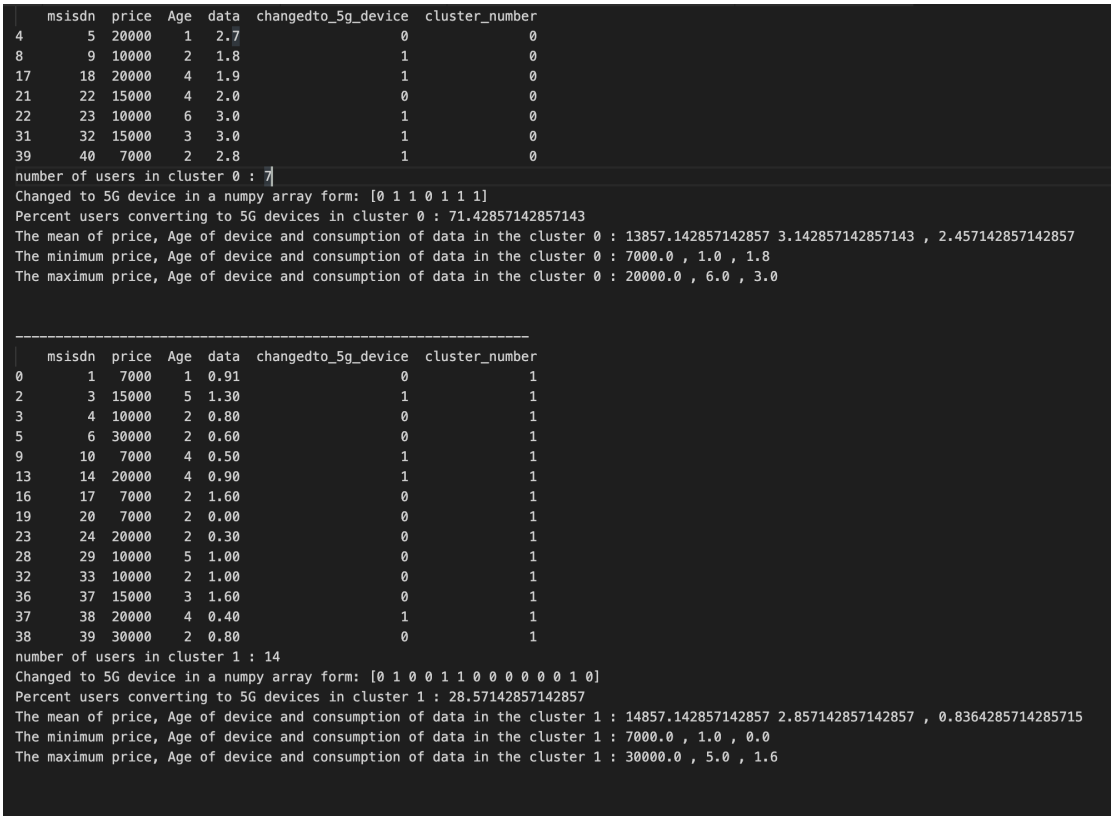


Then instead of using the traditional euclidian distance, we went for the weighted euclidian distance. And in order to modify this part of the code, the whole algorithm was written from scratch. Then by SSE, I came to know about the elbow point, i.e. the number of clusters. Here number of clusters were 5.



	msisdn	price	Age	data	changedto_5g_device	cluster_number
0	1	7000	1	0.91	0	1
1	2	50000	2	1.01	0	2
2	3	15000	5	1.30	1	1
3	4	10000	2	0.80	0	1
4	5	20000	1	2.70	0	0
5	6	30000	2	0.60	0	1
6	7	40000	4	0.50	0	4
7	8	7000	8	0.90	1	3
8	9	10000	2	1.80	1	0
9	10	7000	4	0.50	1	1
10	11	50000	4	1.60	1	2
11	12	15000	6	0.30	0	3
12	13	10000	9	0.40	0	3
13	14	20000	4	0.90	1	1
14	15	30000	5	1.40	1	4
15	16	40000	3	2.00	1	2
16	17	7000	2	1.60	0	1
17	18	20000	4	1.90	1	0

And the following were my conclusions for each cluster.



```
   msisdn  price  Age  data  changedto_5g_device  cluster_number
4       5  20000    1   2.7                    0               0
8       9  10000    2   1.8                    1               0
17     18  20000    4   1.9                    1               0
21     22  15000    4   2.0                    0               0
22     23  10000    6   3.0                    1               0
31     32  15000    3   3.0                    1               0
39     40   7000    2   2.8                    1               0
number of users in cluster 0 : 7
Changed to 5G device in a numpy array form: [0 1 1 0 1 1 1]
Percent users converting to 5G devices in cluster 0 : 71.42857142857143
The mean of price, Age of device and consumption of data in the cluster 0 : 13857.142857142857 3.142857142857143 , 2.457142857142857
The minimum price, Age of device and consumption of data in the cluster 0 : 7000.0 , 1.0 , 1.8
The maximum price, Age of device and consumption of data in the cluster 0 : 20000.0 , 6.0 , 3.0


------------------------------------------------------------
   msisdn  price  Age  data  changedto_5g_device  cluster_number
0       1   7000    1  0.91                    0               1
2       3  15000    5  1.30                    1               1
3       4  10000    2  0.80                    0               1
5       6  30000    2  0.60                    0               1
9      10   7000    4  0.50                    1               1
13     14  20000    4  0.90                    1               1
16     17   7000    2  1.60                    0               1
19     20   7000    2  0.00                    0               1
23     24  20000    2  0.30                    0               1
28     29  10000    5  1.00                    0               1
32     33  10000    2  1.00                    0               1
36     37  15000    3  1.60                    0               1
37     38  20000    4  0.40                    1               1
38     39  30000    2  0.80                    0               1
number of users in cluster 1 : 14
Changed to 5G device in a numpy array form: [0 1 0 0 1 1 0 0 0 0 0 0 1 0]
Percent users converting to 5G devices in cluster 1 : 28.57142857142857
The mean of price, Age of device and consumption of data in the cluster 1 : 14857.142857142857 2.857142857142857 , 0.8364285714285715
The minimum price, Age of device and consumption of data in the cluster 1 : 7000.0 , 1.0 , 0.0
The maximum price, Age of device and consumption of data in the cluster 1 : 30000.0 , 5.0 , 1.6
```