Capstone Project 1 – In Depth Analysis

FEATURE ENGINEERING

Our original dataset is a time series data with 5 features; smsIn, smsOut, callIn, callOut and internet volumes. For clustering our 10,000 grids into different groups, we will convert this time series data into grid-wise data. Creating a total of smsIn, smsOut, callIn, callout and internet volumes for each grid will have very minimal information about the behavior patterns of a grid's telecommunication activities. Hence, we will perform Feature Engineering which is nothing but extracting more information from the existing time series data that helps Clustering algorithm to understand each grid better.

We have created a total of 83 features that are indexed by grid id.

Features	*Description*
Weekend Hourly:	

`hourlysmsMax_WE`
`hourlycallMax_WE`
`hourlyinternetMax_WE`
	Maximum hourly volume of SMS sent and received, call made and received, internet accessed during weekend

`hourlysmsMin_WE`
`hourlycallMin_WE`
`hourlyinternetMin_WE`
	Minimum hourly volume of SMS sent and received, call made and received, internet accessed during weekend

`hourlysmsAvg_WE`
`hourlycallAvg_WE`
`hourlyinternetAvg_WE`
	Average hourly volume of SMS sent and received, call made and received, internet accessed during weekend

Weekend Daily:	

`smsMax_WE`
`callMax_WE`
`internetMax_WE`
	Maximum daily volume of SMS, Call & Internet during weekend

`smsMin_WE`
`callMin_WE`
`internetMin_WE`
	Minimum daily volume of SMS, Call & Internet during weekend

`smsAvg_WE`
`callAvg_WE`
`internetAvg_WE`
	Average daily volume of SMS, Call & Internet during weekend

`totalSmsDay_WE`
`totalCallDay_WE`
`totalInternetDay_WE`
	Total SMS, Calls & Internet from 8AM till 10PM on weekends

`totalSmsNight_WE`
`totalCallNight_WE`
`totalInternetNight_WE`
	Total SMS, Calls & Internet from midnight till 8AM on weekends

WeekDay Hourly:	

`hourlysmsMax_WD`
`hourlycallMax_WD`
`hourlyinternetMax_WD`
	Maximum hourly volume of SMS sent and received, call made and received, internet accessed during weekday

hourlysmsMin_WD hourlycallMin_WD hourlyinternetMin_WD	Minimum hourly volume of SMS sent and received, call made and received, internet accessed during weekday
hourlysmsAvg_WD hourlycallAvg_WD hourlyinternetAvg_WD	Average hourly volume of SMS sent and received, call made and received, internet accessed during weekday
WeekDay Daily:	
smsMax_WD callMax_WD internetMax_WD	Maximum daily volume of SMS, Call & Internet during weekday
smsMin_WD callMin_WD internetMin_WD	Minimum daily volume of SMS, Call & Internet during weekday
smsAvg_WD callAvg_WD internetAvg_WD	Average daily volume of SMS, Call & Internet during weekday
totalSmsDay_WD totalCallDay_WD totalInternetDay_WD	Total SMS, Calls & Internet from 8AM till 10PM on weekdays
totalSmsNight_WD totalCallNight_WD totalInternetNight_WD'	Total SMS, Calls & Internet from midnight till 8AM on weekdays
Daily:	
dailySmsIn/dailySmsOut	Ratio of SMS received to SMS sent daily
dailyCallIn/dailyCallOut	Ratio of Calls received to Calls made daily
dailySms/dailyCall	Ratio of daily SMS to daily Call volumes
dailyInternet/dailySmsCall	Ratio of daily Internet to daily SMS & Call volumes
'totalSmsDay_WD', 'totalCallDay_WD', 'totalInternetDay_WD'	Total SMS, Calls & Internet from Midnight to 8AM
Weekly:	
smsAvgdiff_weekly callAvgdiff_weekly internetAvgdiff_weekly	Average of difference in the volume of SMS, Calls & Internet from one week to another
smsMax_weekly callMax_weekly internetMax_weekly	Maximum volume of weekly SMS, Calls & Internet
smsMin_weekly callMin_weekly internetMin_weekly	Minimum volume of weekly SMS, Calls & Internet
smsAvg_weekly callAvg_weekly internetAvg_weekly	Average volume of weekly SMS, Calls & Internet
Monthly:	

monthlyAvg_sms monthlyAvg_call monthlyAvg_internet	Average volume of monthly SMS, Calls & Internet
smsAvg_Nov callAvg_Nov internetAvg_Nov	Average volume of November month SMS, Calls & Internet
smsAvg_Dec callAvg_Dec internetAvg_Dec	Average volume of December SMS, Calls & Internet
smsMax_Nov callMax_Nov internetMax_Nov	Maximum volume of November month SMS, Calls & Internet
smsMax_Dec callMax_Dec internetMax_Dec	Maximum volume of December month SMS, Calls & Internet
smsMin_Nov callMin_Nov internetMin_Nov	Minimum volume of November month SMS, Calls & Internet
smsMin_Dec callMin_Dec internetMin_Dec	Minimum volume of December month SMS, Calls & Internet
Christmas & New Year	
totalSms_xMas totalCall_xMas totalInternet_xMas	Total SMS, Calls & Internet volumes on Christmas day
totalSms_NewYear totalCall_NewYear totalInternet_NewYear	Total SMS, Calls & Internet volumes on New Year day
totalSms_NewYearEve totalCall_NewYearEve totalInternet_NewYearEve	Total SMS, Calls & Internet volumes on New Year Eve [Dec 31st 6Pm to 1AM]
Totals	
totalSmsIn totalSmsOut totalCallIn totalCallOut totalSMS totalCall totalInternet	Grid-wise total SMS-In, SMS-Out, Call-In, Call-Out, SMS, Calls & Internet

K-MEANS CLUSTERING

We will apply K-Means algorithm from Sci-kit learn package for clustering the grids. K-Means iteratively partitions the dataset into K subgroups, such that each data point belongs to only one group (no overlapping). Data points are assigned to a cluster such that its sum of the squared distance from the cluster's centroid is at the minimum.

There are few steps to follow in order to prepare the dataset for K-Means model,

1. Remove all NAN values from the dataset after creating new features.
2. Standardization of the data: Since clustering algorithms use distance-based measurements to determine the similarity between data points, it's recommended to

standardize the data to have a mean of zero and a standard deviation of one since almost always the features in any dataset would have different units of measurements.
But our dataset has features with same unit of measurement, which is volumes of telecommunication activities, thus, we do not do any standardization.
3. All column values are converted to NumPy array, which is the input format for Sci-kit learn K-Means algorithm.
4. Number of subgroups must be pre-determined from the dataset.

Finding the Optimal K value

Elbow Method:

It is a plot of sum of squared distance (SSE) between data points and their assigned clusters' centroids for a range of K values. We pick K at the spot where SSE starts to flatten out and forming an elbow.
From our plot we have two optimal candidates for K, K=6 & K=7 beyond which the plot plateaus.

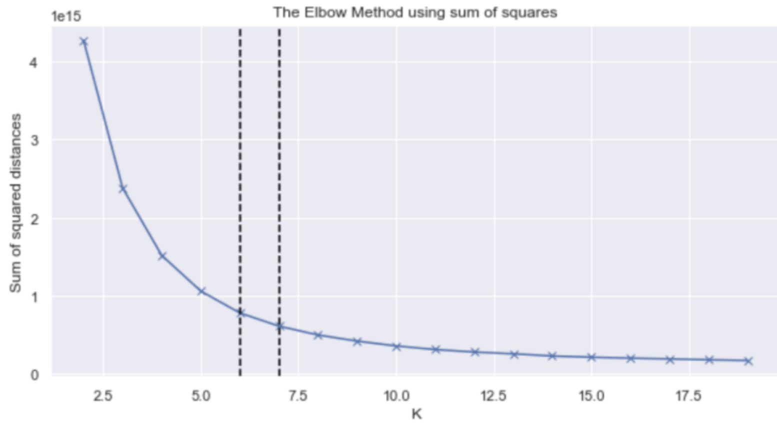


Fig 1: Elbow method to find the optimal K value

Kneed package:

Visually inspecting the plot to identify the Knee/Elbow point could be confusing as in our case. We will take help of Kneed package the mathematically computes the Knee/Elbow point.
For our dataset, Kneed package has returned K=6 as the knee point.

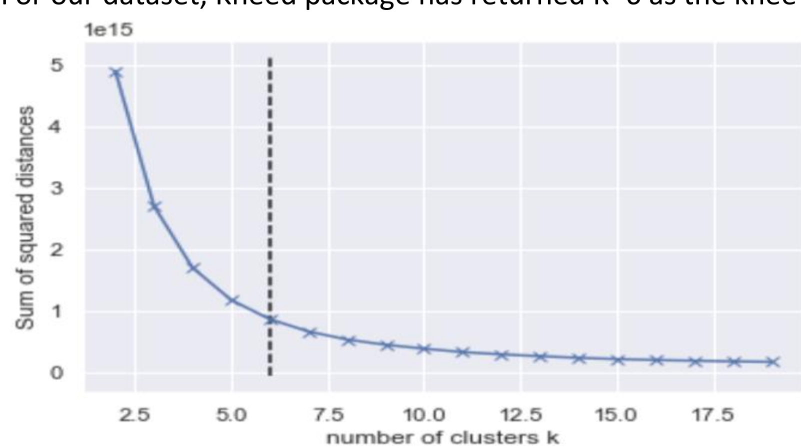


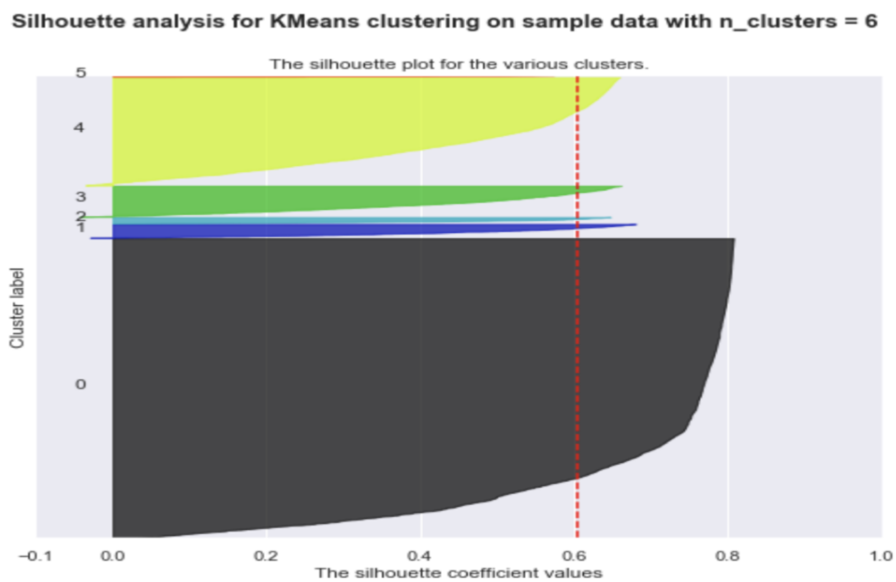
Fig 2: Kneed package output shows K=6

Silhouette coefficient plot:

The silhouette plot displays a measure of how close each point in one cluster is to points in the neighboring clusters and thus provides a way to assess parameters like number of clusters visually. Silhouette coefficients (as these values are referred to as) near +1 indicate that the sample is far away from the neighboring clusters. A value of 0 indicates that the sample is on or very close to the decision boundary between two neighboring clusters and negative values indicate that those samples might have been assigned to the wrong cluster.

For our dataset we will pick K=6, as Silhouette coefficient plot shows a smaller number of datapoints assigned to the wrong cluster for K=7.

```
For n_clusters = 6 The average silhouette_score is : 0.6046179303624023
```




```
For n_clusters = 7 The average silhouette_score is : 0.5824554704630547
```

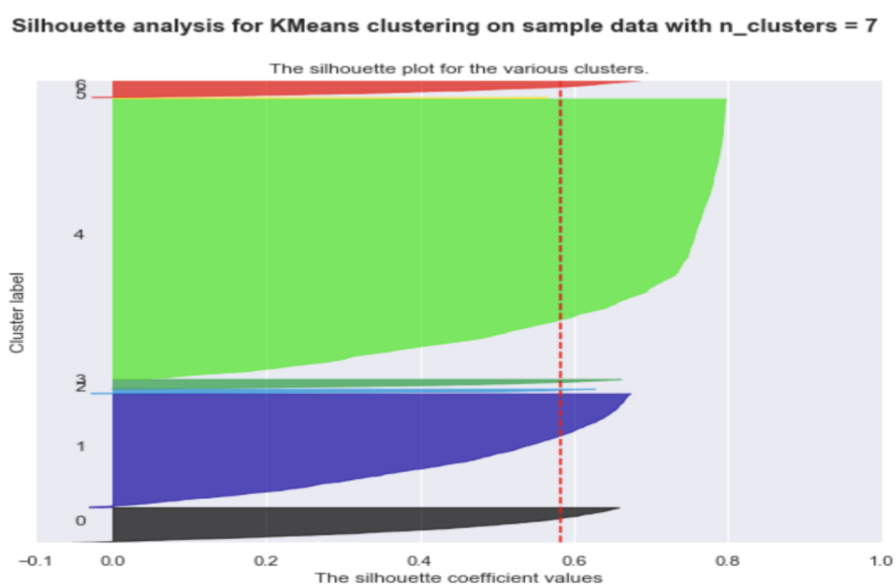
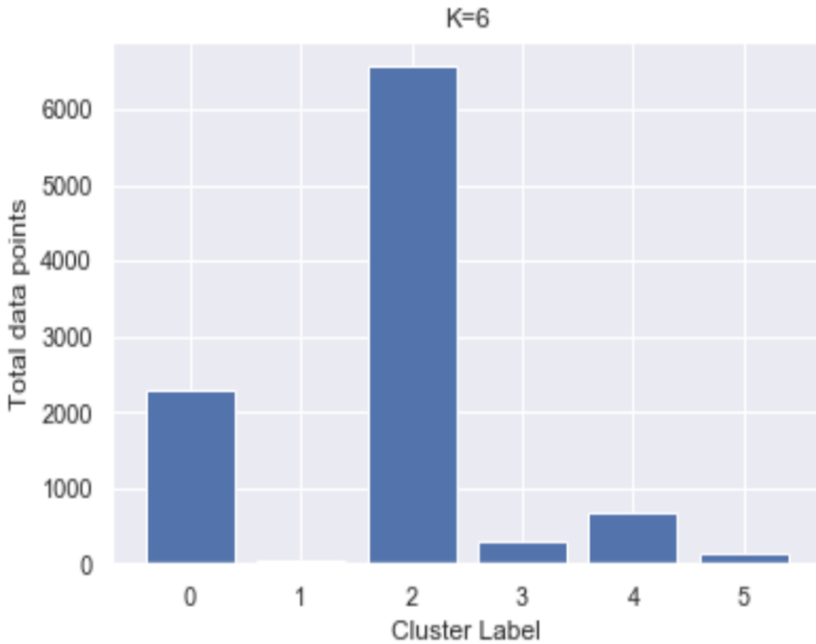


Fig 3: Plot of Silhouette coefficients of all the data points for K=6 & K=7

K-Means Clustering with K = 6

Applying the model with K=6, results in 6 subgroups with distribution of grids as shown below,



Cluster Label	Number of Grids
0	2306
1	31
2	6566
3	298
4	660
5	139

Fig 4: Subgroups after K-Means Clustering with K=6

Principal Component Analysis

Displaying the clusters in subgroups in 92 dimensions is not possible. We will reduce the dimensions to 2 dimensions using PCA in order to visualize the datapoints in subgroups.

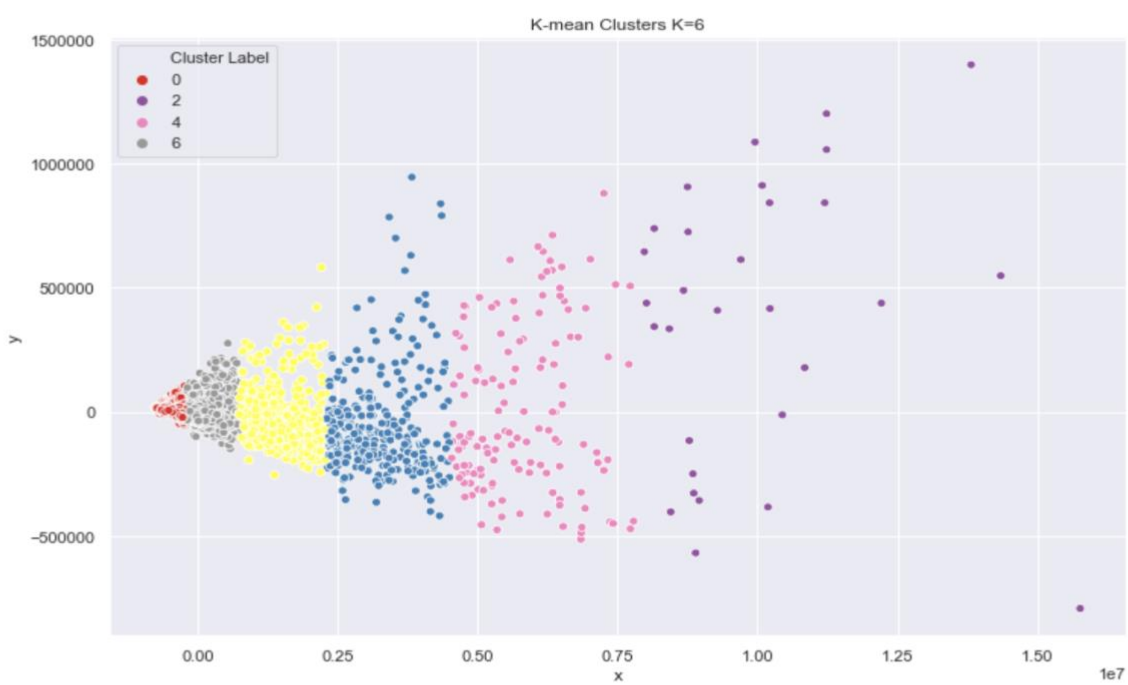


Fig 5: Visualization of the datapoints in its subgroups

Subgroup labels and grid id are extracted and converted to geojson format with color properties added for each subgroup. This geojson is then displayed on the map. It appears that the subgroups created by the volumes of telecommunication activities is closely related to the population distribution of the region.

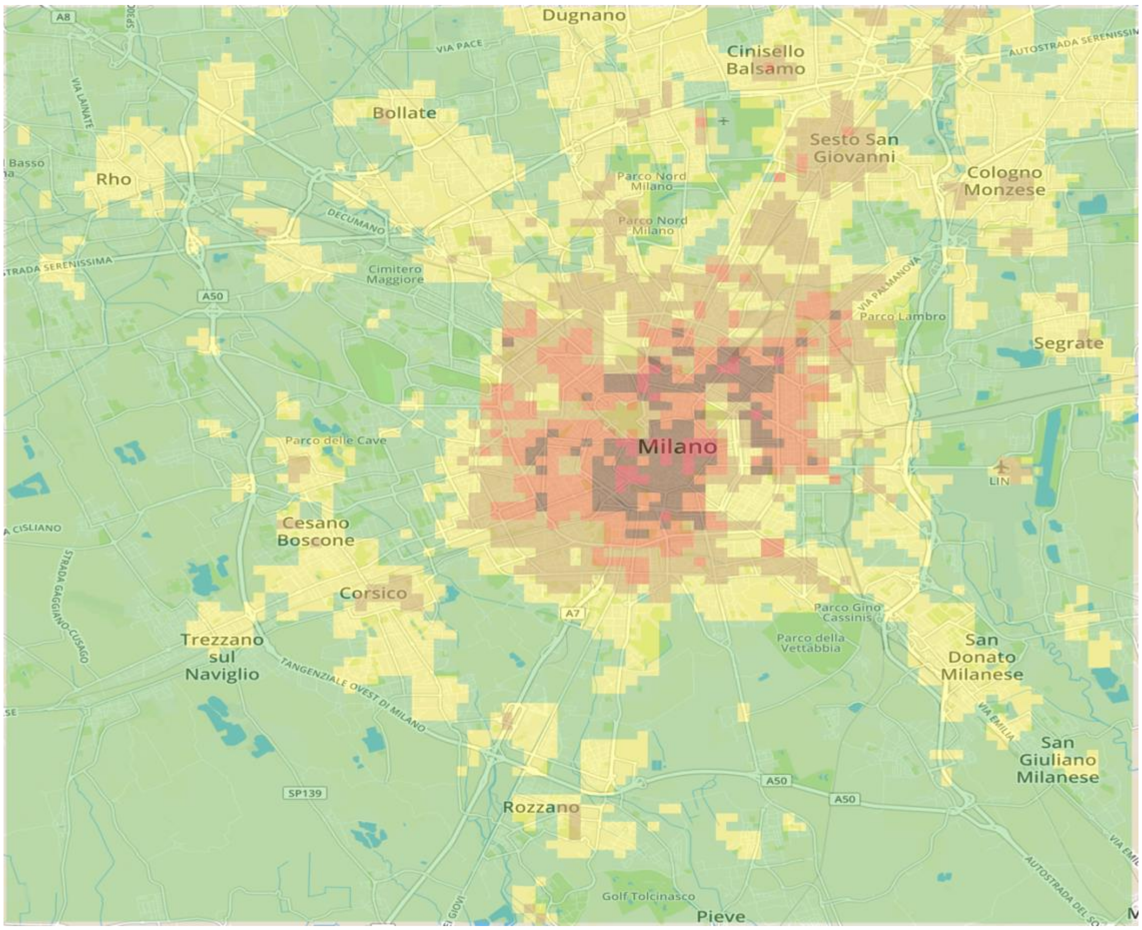


Fig 6: Visualization of 10000 grids clustered into 6 subgroups

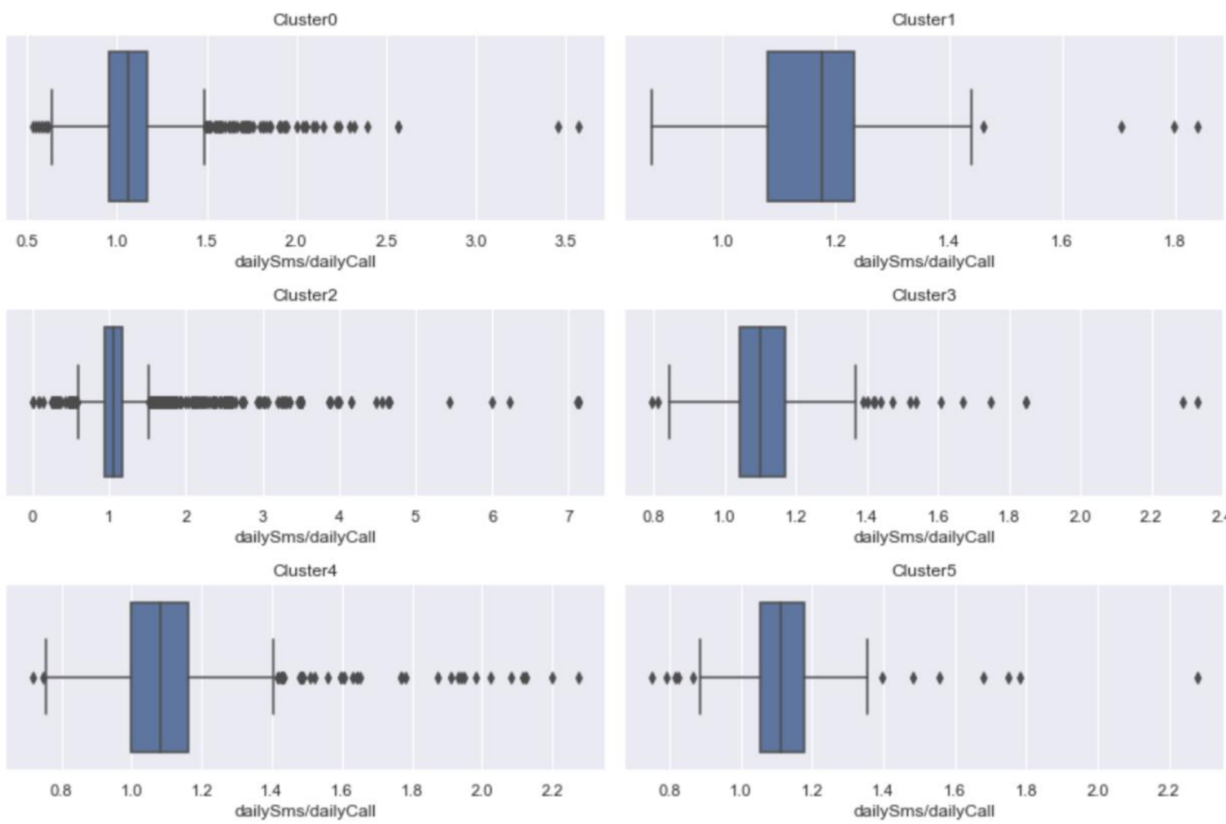
Rank	City	Population	Area (km^2)	Density (inhabitants / km^2)	Altitude (mslm)
1st	Milan	1336879	181.76	7355.2	122
2nd	Sesto San Giovanni	81750	11.74	6963.4	140
3rd	Cinisello Balsamo	74536	12.7	5869	154
4th	Legnano	59492	17.72	3357.3	199
5th	Rho	51033	22:32	2286.4	158
6th	Cologno Monzese	47880	8:46	5659.6	134
7th	Paderno Dugnano	47750	14.1	3386.5	163
8th	Rozzano	41581	13:01	3196.1	103
9th	San Giuliano Milanese	37235	30.71	1212.5	98
10th	Pioltello	36756	13.1	2805.8	156

Fig 7: Largest municipalities by population of Milan [sourced from Wikipedia]

ANALYSIS OF THE CLUSTERS

Clusters are plotted against different dimensions. Subgroups from the clustering model shows defined volume ranges for each dimension. Presence of outliers indicates wrong assignment of the data points to a subgroup with respect to that dimension alone. In general, this clustering model has done well for Internet volumes. In case of SMS & Calls, Cluster1, Cluster5 & Cluster3 are clearly formed with few outliers only.

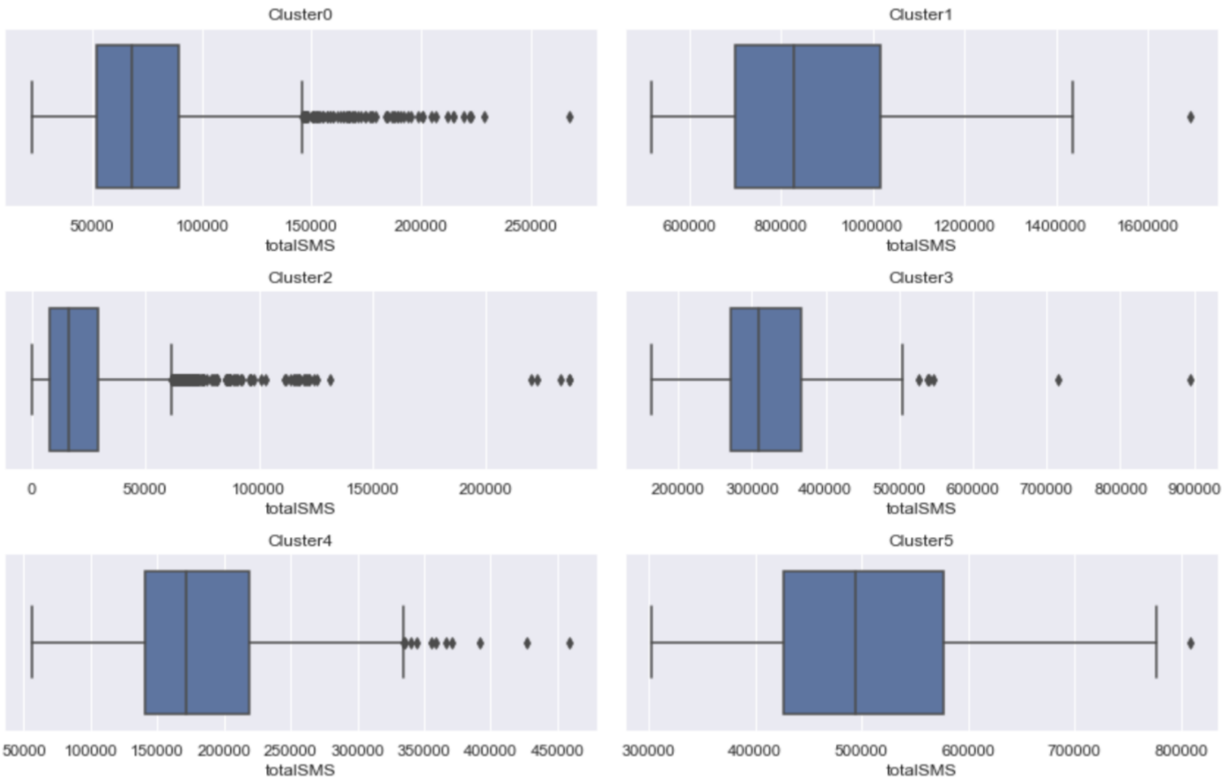
T-tests and one-way ANOVA tests can be performed on the subgroups for different dimensions in order to understand the similarities and dissimilarities in their behavioral patterns.




```
group0 = Cluster0['dailySms/dailyCall'].to_list()
group1 = Cluster1['dailySms/dailyCall'].to_list()
group2 = Cluster2['dailySms/dailyCall'].to_list()
group3 = Cluster3['dailySms/dailyCall'].to_list()
group4 = Cluster4['dailySms/dailyCall'].to_list()
group5 = Cluster5['dailySms/dailyCall'].to_list()
stats.f_oneway(group0, group1, group2, group3, group4, group5)
```

```
F_onewayResult(statistic=3.047848879647677, pvalue=0.009423790504088599)
```

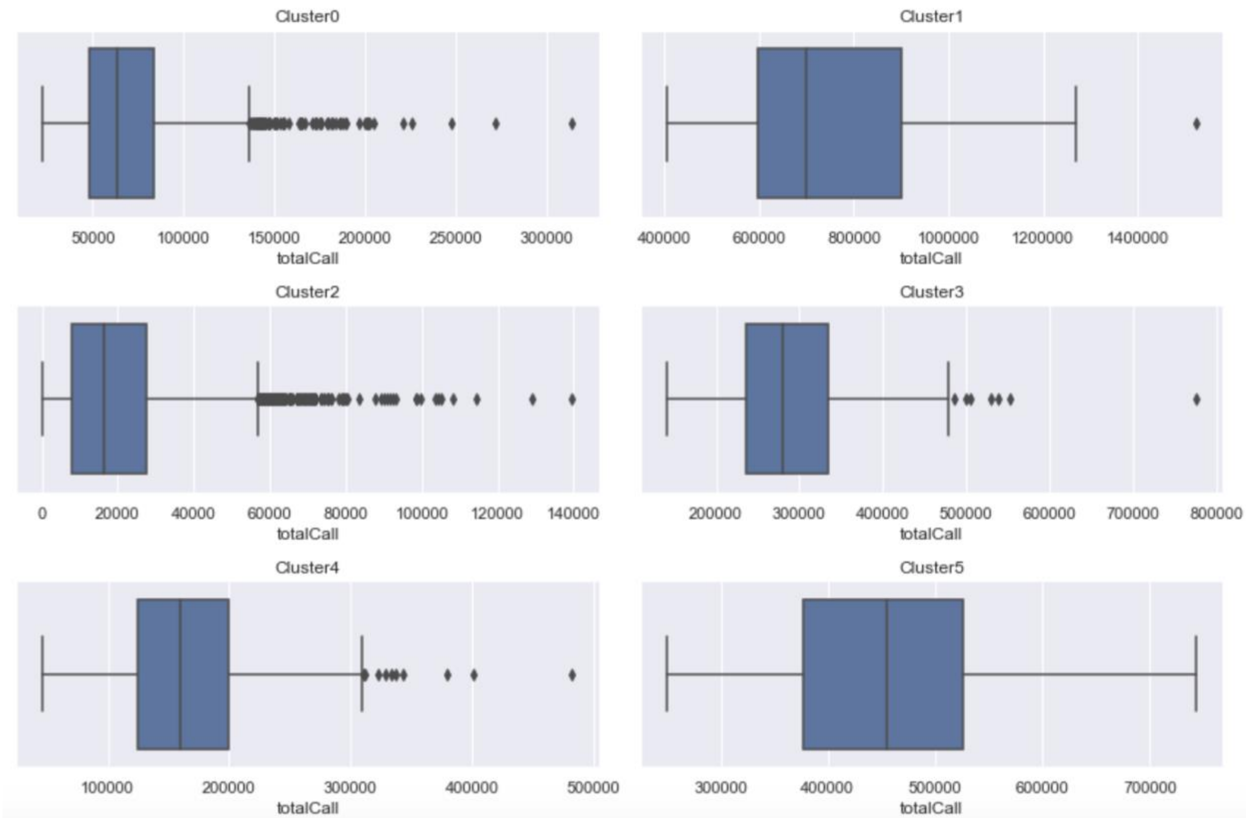
Fig 8: Box-plot of 6 subgroups and its dailySms/dailyCall feature. One-Way ANOVA test shows approximately same mean value for all the groups




```
group0 = Cluster1['totalSMS'].to_list()
group1 = Cluster5['totalSMS'].to_list()
stats.ttest_ind(group0, group1)
```

```
Ttest_indResult(statistic=12.831446000132484, pvalue=1.0429195023851667e-26)
```

Fig 9: Box-plot of 6 subgroups and its totalSMS feature. T-test shows approximately same mean value for subgroup 1&5




```
group0 = Cluster1['totalCall'].to_list()
group1 = Cluster5['totalCall'].to_list()
stats.ttest_ind(group0, group1)
```

```
Ttest_indResult(statistic=10.807200843138878, pvalue=5.293920323077129e-21)
```

Fig 10: Box-plot of 6 subgroups and its totalCall feature. T-test shows approximately same mean value for subgroup 1&5

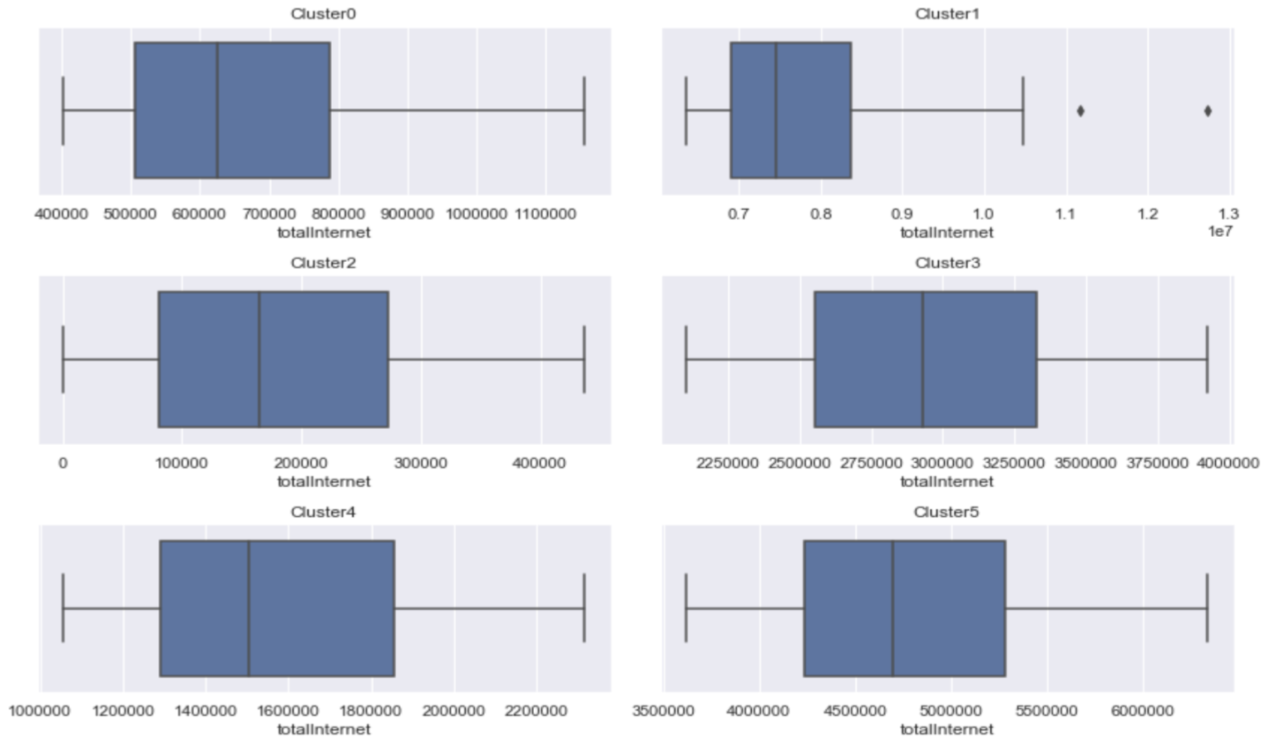


Fig 11: Box-plot of 6 subgroups and its totalInternet feature

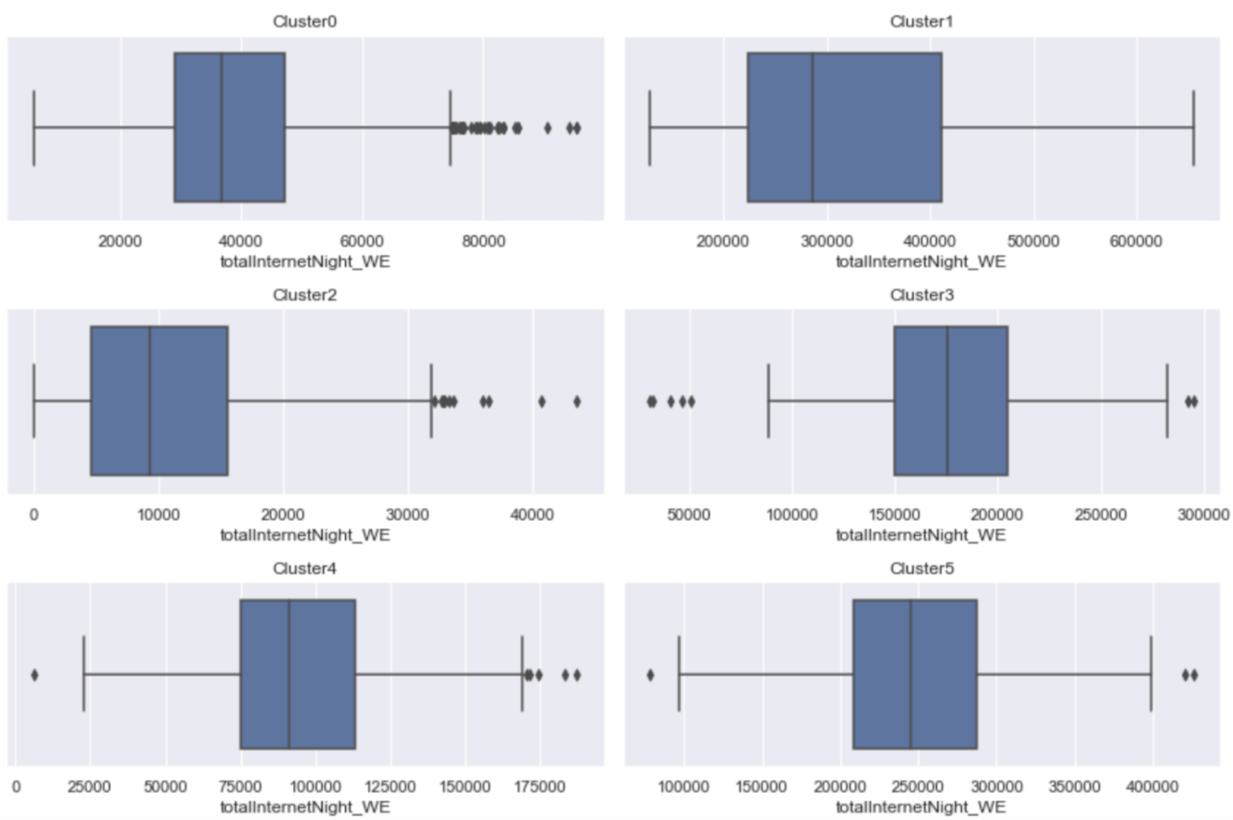


Fig 12: Box-plot of 6 subgroups and its totalInternetNight_WE feature

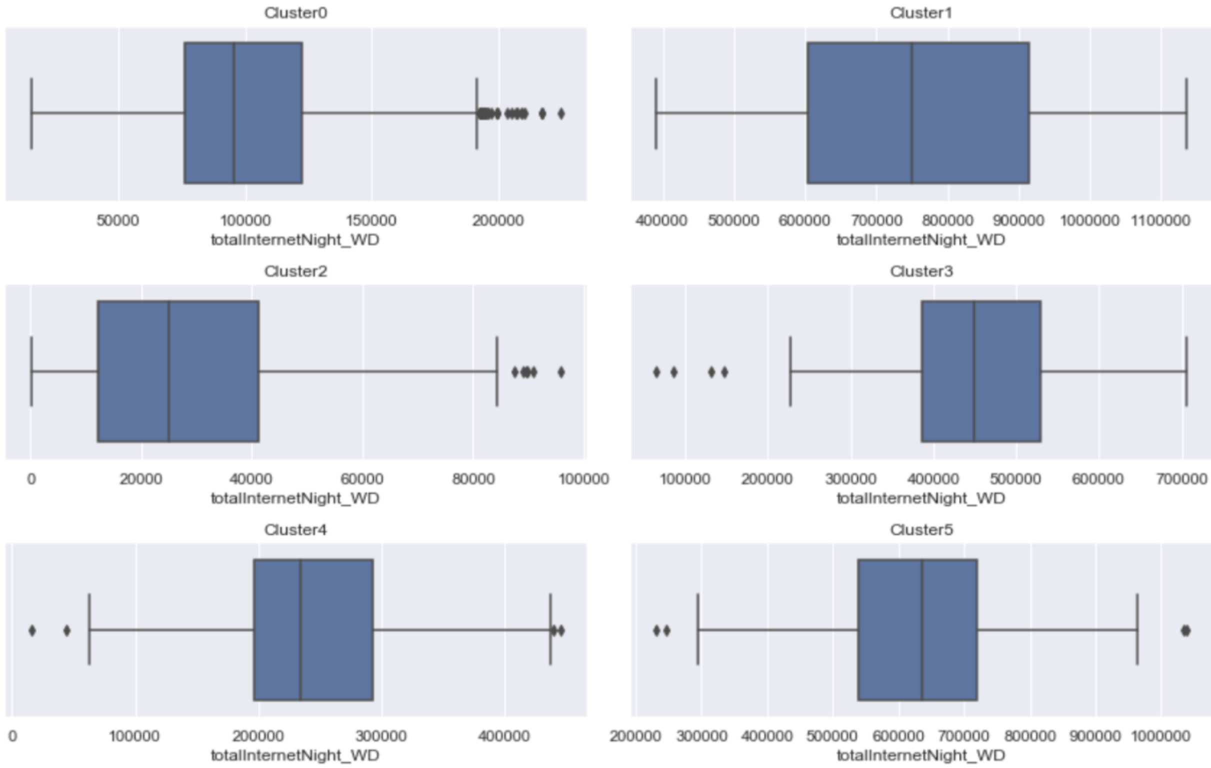


Fig 13: Box-plot of 6 subgroups and its totalInternetNight_WD feature

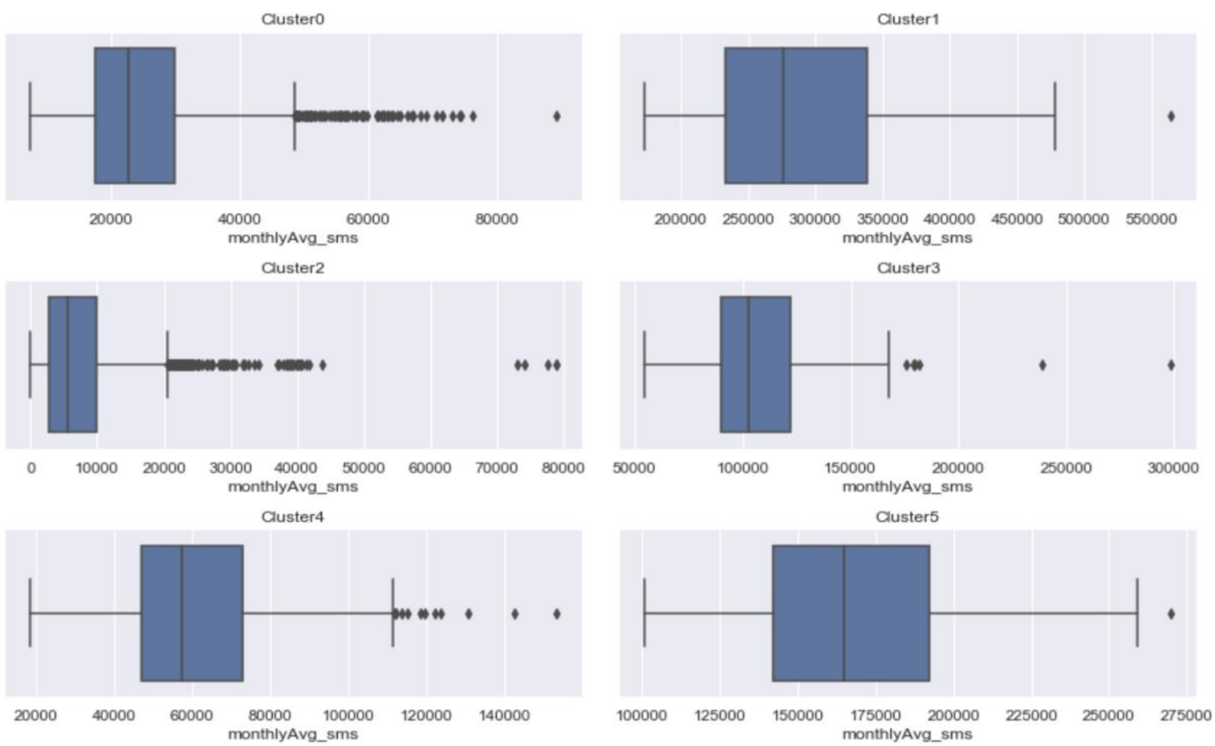


Fig 14: Box-plot of 6 subgroups and its monthlyAvg_sms feature

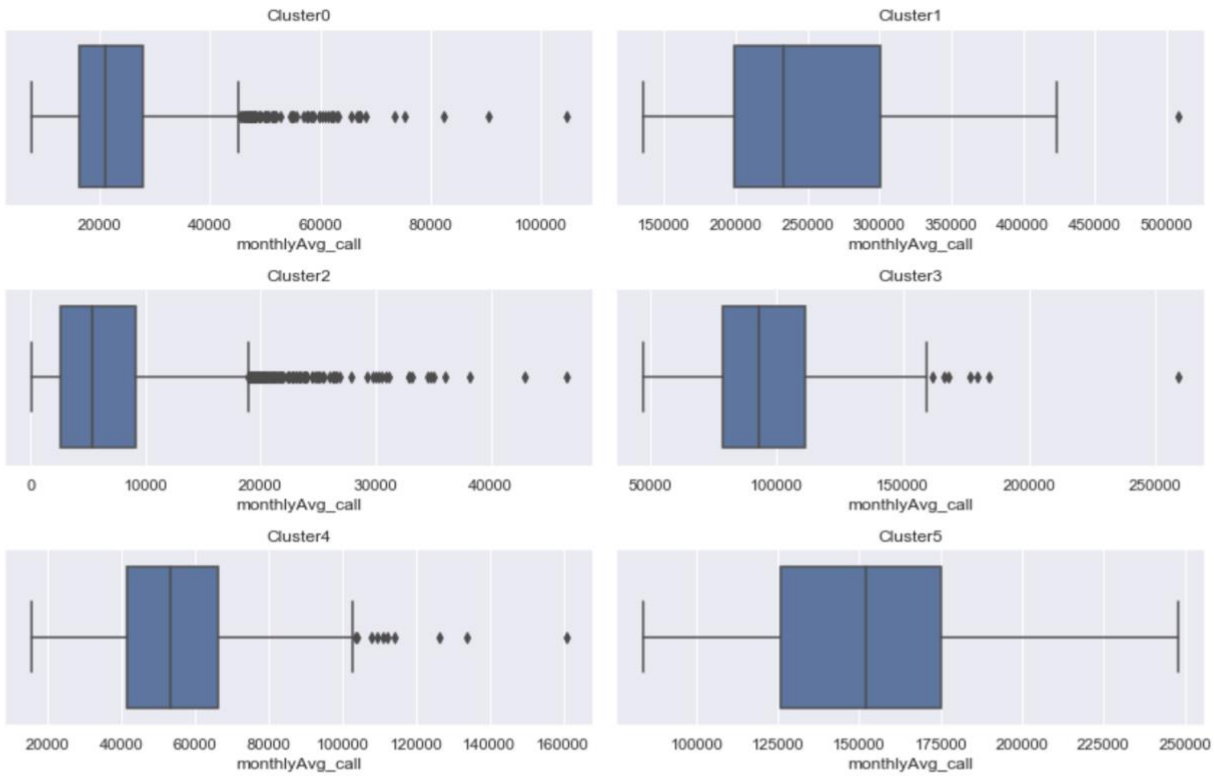


Fig 15: Box-plot of 6 subgroups and its monthlyAvg_call feature

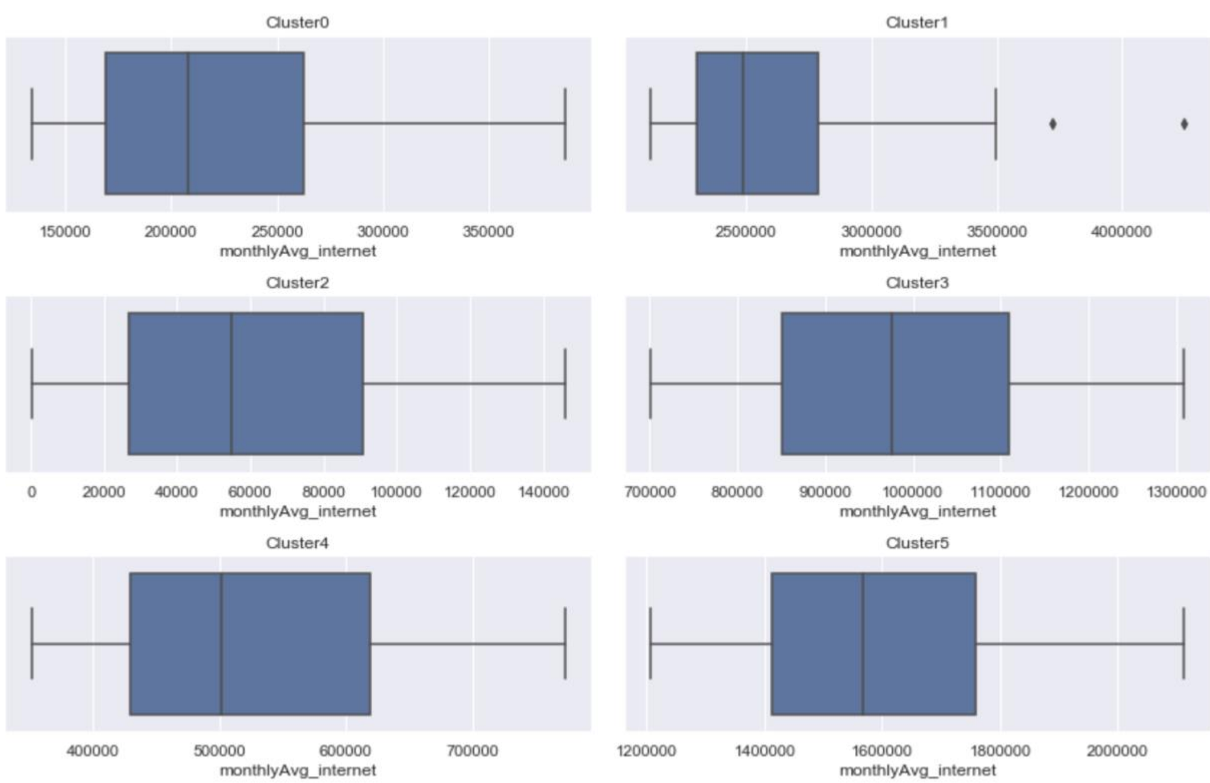


Fig 16: Box-plot of 6 subgroups and its monthlyAvg_internet feature