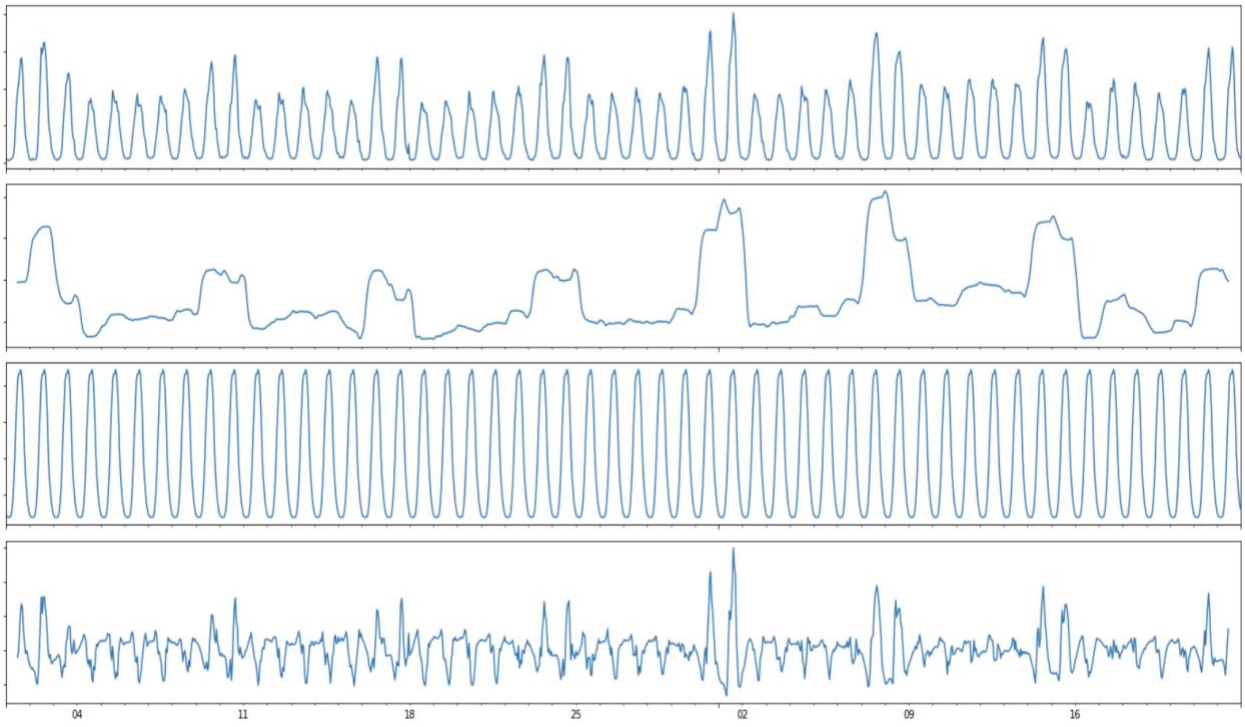
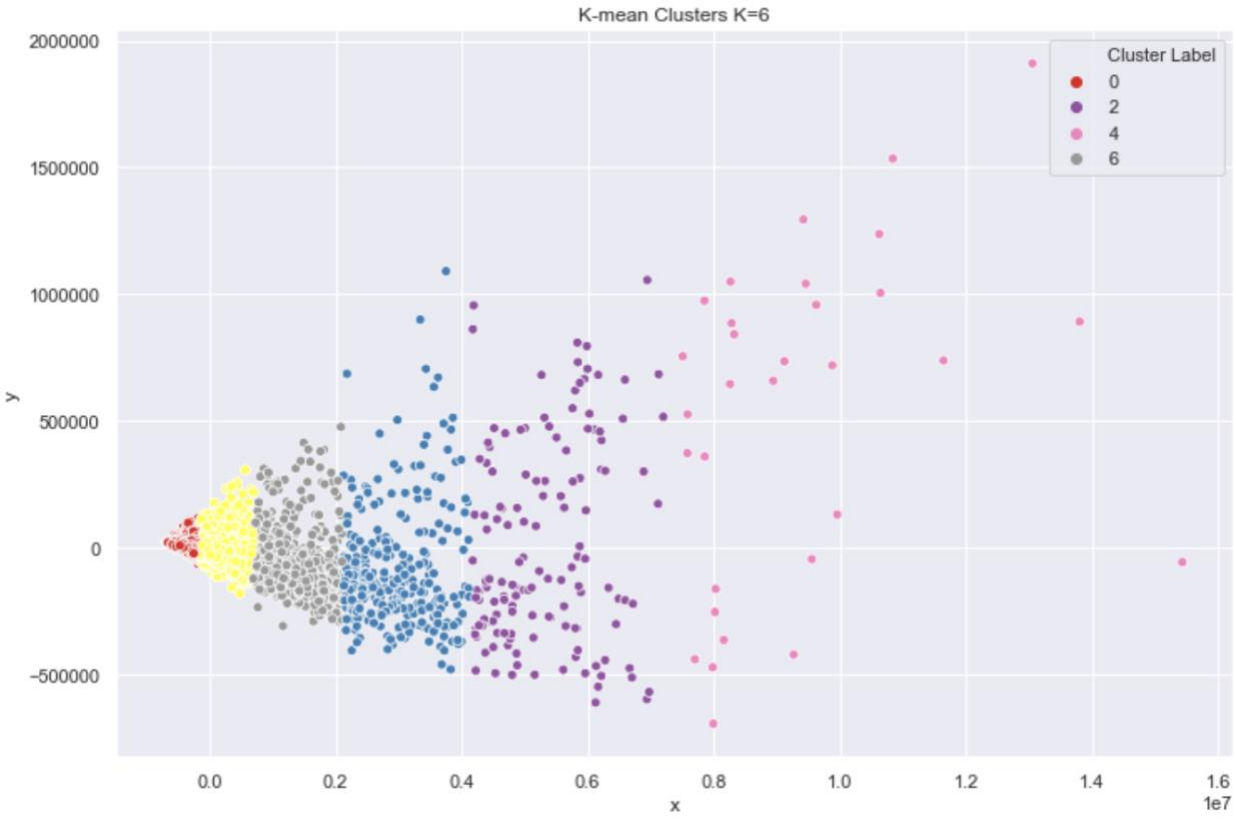
Modeling of Telecommunication CDRs in a Time Series dataset





By,
Aruna Subbiah

Table of Contents

1. CDR

The exponential increase in the use of internet services and mobile phones is generating large amount of data that can be used to provide useful insights about the network usage pattern. Call Detail Record [CDR] describes a specific instance of a telecommunication transaction that passes through a network element. Every time a user performs a telecom activity such as send/receive SMS and calls, a CDR is generated. It contains information about the caller/sender ID, location, time, data used, etc. Millions and millions of such records are generated and is mainly used for billing purposes by the telecom company.

Problem Statement

Analysis and modeling of this time series data helps to identify geographical boundaries of various usage patterns. This helps in decision making of resource allocation by telecommunication companies who own the network elements, inspecting quantitatively different aspects of human behavior such as socio-economic status of geographical regions and people's mobility. CDRs collected for a span of time can also be used in forecasting future volumes for a network.

Data Description

This dataset is a part of Telecom Italia Big Data Challenge which is an aggregation of telecommunications, weather, news, social networks and electricity data from the city of Milan and the Province of Trentino. This dataset has been released to the research teams under the Open Database License (ODbL) and is maintained by Harvard Dataverse.

For this project we will use telecommunication data from the city of Milan, Italy. It is available as .txt files with tab-delimited values (TSV) from the link, https://doi.org/10.7910/DVN/EGZHFV.
There are 62 files consisting of CDRs collected from Nov 1,2013 to Jan 1, 2014, one file for each day.

File size: ~ 200MB – 300MB with ~4 million rows each file

Files: sms-call-internet-mi-2013-11-01.txt to sms-call-internet-mi-2014-01-01.txt

There are 8 columns, with no headers. Each column represents,

- Grid id: identification string of a given square of Milan GRID. The geographical region of the city is spatially divided into 1000 square grids.
- Time Interval: start interval time expressed in milliseconds. The end interval time can be obtained by adding 600,000 milliseconds (10 min) to this value;
- SMS-in activity: activity proportional to the amount of received SMSs inside a given Square id and during a given Time interval. The SMSs are sent from the nation identified by the Country code;
- SMS-out activity: activity proportional to the amount of sent SMSs inside a given Square id during a given Time interval. The SMSs are received in the nation identified by the Country code;
- Call-in activity: activity proportional to the amount of received calls inside the Square id during a given Time interval. The calls are issued from the nation identified by the Country code;
- Call-out activity: activity proportional to the amount of issued calls inside a given Square id during a given Time interval. The calls are received in the nation identified by the Country code;
- Internet traffic activity: number of CDRs generated inside a given Square id during a given Time interval. The Internet traffic is initiated from the nation identified by the Country code;
- Country code: the phone country code of the nation.

Each observation holds information about the volume of telecommunication activities performed over the next 10 min.

	gridID	timeInterval	countryCode	smsIn	smsOut	callIn	callOut	internet
0	1	1383260400000	0	0.081363	NaN	NaN	NaN	NaN
1	1	1383260400000	39	0.141864	0.156787	0.160938	0.052275	11.028366
2	1	1383261000000	0	0.136588	NaN	NaN	0.027300	NaN
3	1	1383261000000	33	NaN	NaN	NaN	NaN	0.026137
4	1	1383261000000	39	0.278452	0.119926	0.188777	0.133637	11.100963
5	1	1383261600000	0	0.053438	NaN	NaN	NaN	NaN
6	1	1383261600000	39	0.330641	0.170952	0.134176	0.054601	10.892771
7	1	1383262200000	0	0.026137	NaN	NaN	NaN	NaN
8	1	1383262200000	39	0.681434	0.220815	0.027300	0.053438	8.622425
9	1	1383262800000	0	0.027300	NaN	NaN	NaN	NaN

Fig 1: Data as read from the input files

As the original bundle of dataset comes from various companies (telecommunications, weather, news, social networks and electricity) with different standards, in order to ease the comparisons of different geographical areas, the city of Milan's spatial distribution is aggregated in a grid with 10,000 square cells. The area is composed of a grid overlay of 10,000 squares with size of about 235×235 meters. This dataset provides the geographical reference of each square which composes the grids in the reference system: WGS 84—EPSG:4326. It is downloaded from the link, https://doi.org/10.7910/DVN/QJWLFU.

File type: geojson

File size: 3MB

File: milano-grid.geojson

Columns:

- *square id*: identification string of a given square of the Milan GRID;
- *Time Interval*: The cell geometry expressed as geoJSON and projected in WGS84 (EPSG:4326).

2. DATA WRANGLING

Reading data into Pandas DataFrame

Reading 20GB of data from 62 files iteratively and combining them into a single dataframe takes about 30 min and utilizes very high system memory. Alternatively, reading a file into a dataframe, perform operations like sampling [daily, hourly], grouping & indexing that reduces the number of rows and combining these individual data frames into a single dataframe results in a faster and efficient loading operation.

Handling Datetime column

Time Interval column is represented in milliseconds, as epoch/Unix timestamps. It is the number of milliseconds passed since 00:00:00 UTC Thursday, 1 January 1970. This column values are converted to pandas Datetime object which is passed to Pandas in built Datetime functions to convert to Milan's local time zone. This value is stored in a new column, startTime.

Removing unwanted columns

Time Interval column now has redundant values and Country code column will not be used in this project. They are both dropped from the data frame.

Resampling

Several rows of data, with 10 min time interval are aggregated into DataFrames with daily (24 hour) and hourly time intervals. They are grouped and indexed by Grid ID & startTime columns. Total volume of each activity over the 2 months for individual grids is calculated in another DataFrame. Because rows are aggregated into hourly, daily and total volumes there are no NaN values.

		smsIn	smsOut	callIn	callOut	internet
gridID	startTime					
1	2013-11-01 00:00:00	2.084285	1.104749	0.591930	0.429290	57.799009
	2013-11-01 01:00:00	1.163624	0.770031	0.190564	0.194139	44.046899
	2013-11-01 02:00:00	0.415579	0.300391	0.027925	0.135964	41.207149
	2013-11-01 03:00:00	1.152067	0.895724	0.001787	0.026137	33.022070
	2013-11-01 04:00:00	0.354453	0.511192	0.005362	0.026137	31.376930
...	...	...	...	...	...	...
2	2013-11-01 00:00:00	2.091501	1.087979	0.602031	0.438173	57.914858
	2013-11-01 01:00:00	1.178439	0.773207	0.192136	0.193979	44.151457
	2013-11-01 02:00:00	0.415258	0.302315	0.028278	0.137535	41.329761
	2013-11-01 03:00:00	1.151394	0.902170	0.000922	0.027356	33.078556
	2013-11-01 04:00:00	0.357948	0.520075	0.002765	0.027356	31.453361

Fig 2: DataFrame with hourly aggregation of telecommunication activities for each grid

gridID	startTime	smsIn	smsOut	callIn	callOut	internet
1	2013-11-01	78.709755	45.886570	41.108567	48.245378	1507.048349
	2013-11-02	86.415810	43.875946	47.891016	53.590637	1515.641856
	2013-11-03	77.728292	45.446780	36.145436	40.906425	1533.148425
	2013-11-04	104.793806	54.821018	67.898464	70.399418	1404.813593
	2013-11-05	97.425105	46.607029	68.735213	70.766221	1518.090111
	...	...	...	...	...	...
	2013-12-31	124.049269	85.569336	58.372156	63.266368	1376.737573
	2014-01-01	126.893711	96.486508	43.109098	54.512429	1532.564428
2	2013-11-01	79.846206	46.480586	41.741924	49.136913	1512.859757
	2013-11-02	87.738546	44.512066	48.636353	54.521711	1522.727906
	2013-11-03	78.740671	45.881772	36.713980	41.584801	1539.831167

Fig 3: DataFrame with daily aggregation of telecommunication activities for each grid

gridID	smsIn	smsOut	callIn	callOut	internet
1	6178.894730	3358.842325	3805.892719	3991.422048	92992.666580
2	6267.021008	3402.658923	3861.301592	4052.842143	93368.388389
3	6360.827944	3449.299959	3920.282146	4118.221405	93768.329391
4	5923.635378	3231.926757	3645.399918	3813.517635	91904.381588
5	5522.707656	3017.566898	3401.745307	3568.366951	83630.697355
6	6360.827944	3449.299959	3920.282146	4118.221405	93768.329391
7	6360.827944	3449.299959	3920.282146	4118.221405	93768.329391
8	6360.827944	3449.299959	3920.282146	4118.221405	93768.329391
9	6360.827944	3449.299959	3920.282146	4118.221405	93768.329391
10	4776.609226	2591.076508	2963.797077	3175.114591	56177.723211

Fig 4: DataFrame with total volume of telecommunication activities over the 2 months for each grid

3. EXPLORATORY DATA ANALYSIS

Visualization of Grids

milano-grid.geojson file is loaded using packages geopandas & geojsonio that shows the overlay of 10000 grids over the city of Milan's map.

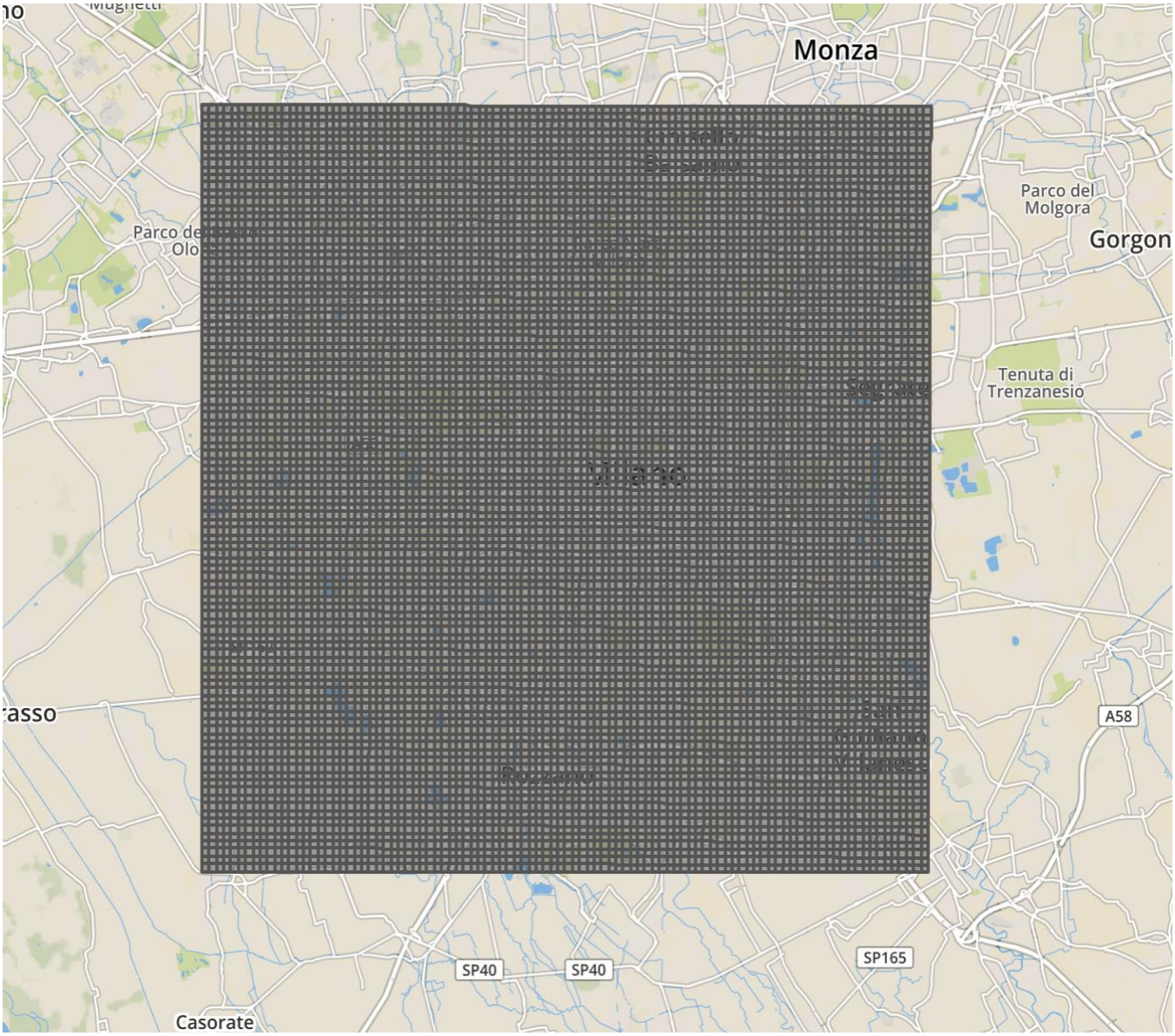


Fig 5: Spatial aggregation of 10,000 grids over the city of Milan

In the network, SMS-In and SMS-Out utilizes the control channel, Call-In and Call-Out utilizes transmitted over voice channel and the internet is transmitted over broadband frequencies. Thus, we will use SMS (sum of SMS-In & SMS-Out), Call (sum of Call-In & Call-Out) and Internet activity for the analysis.

Top 10 Grids

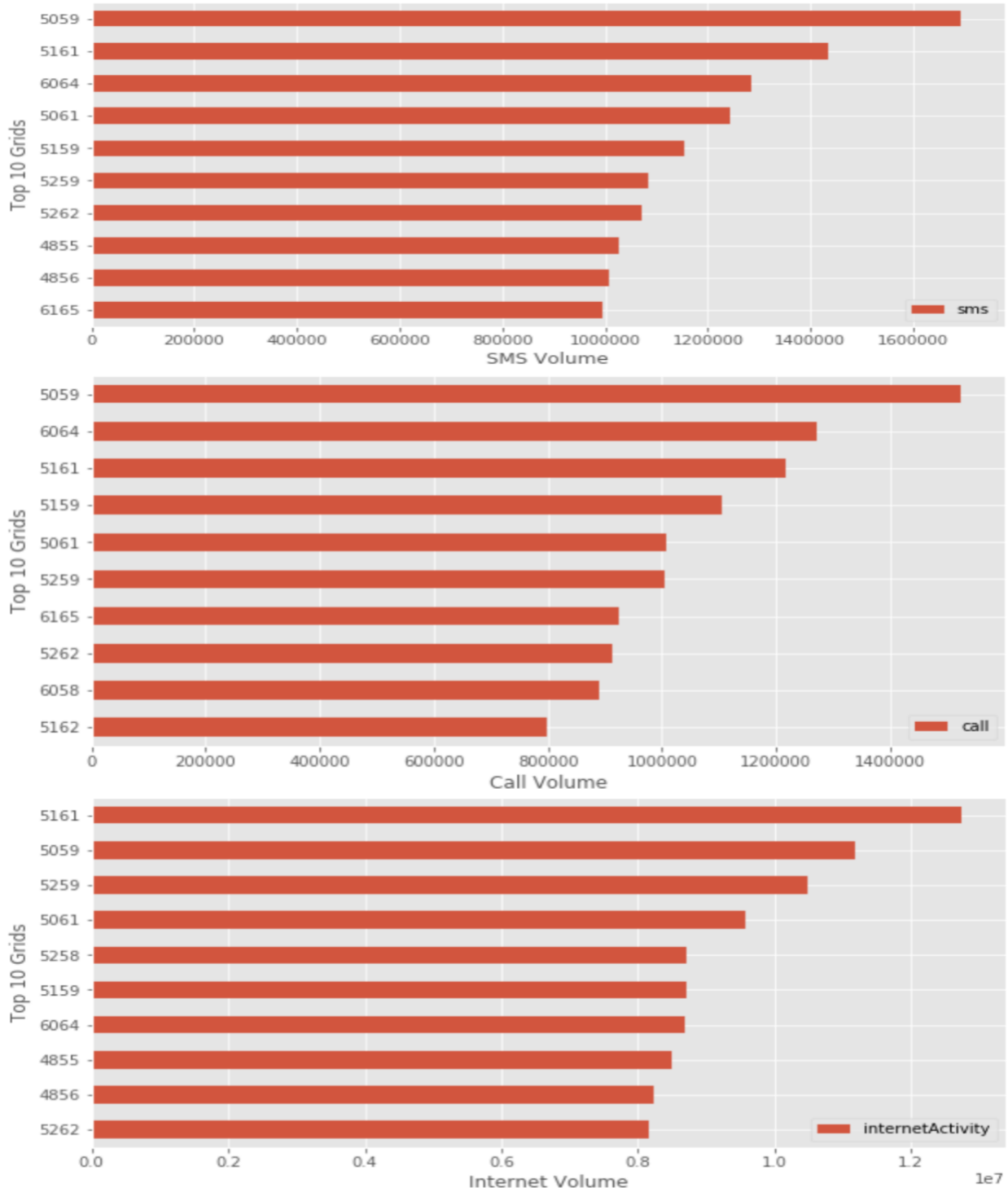


Fig 6: Horizontal bar plots showing top 10 grids with high total volumes in each telecommunication activity

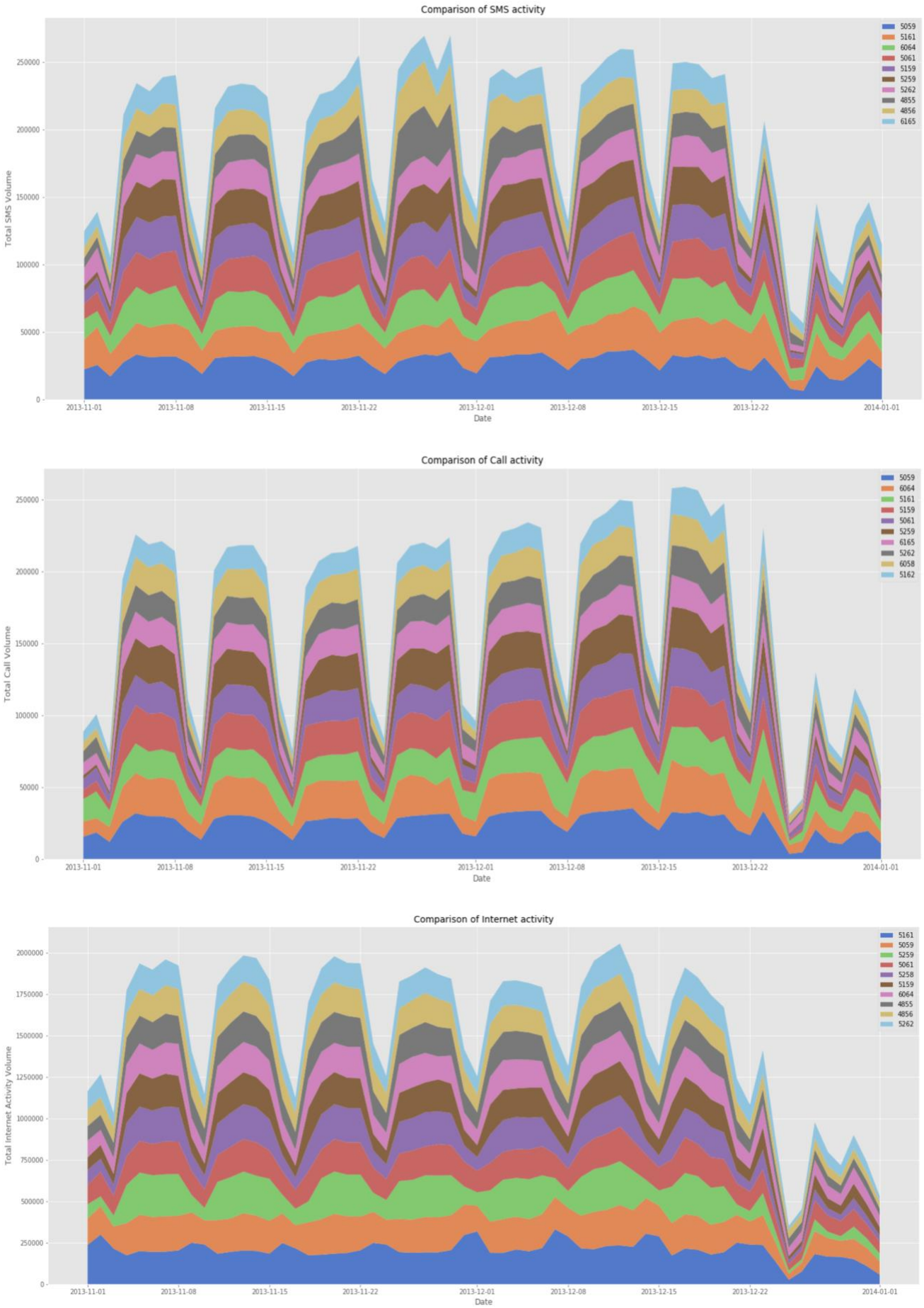


Fig 7: Stacked area plots showing comparison of top 10 grids daily pattern

Top 10 grids that experience high volumes for each of these activities for the 2 months are identified. From the above plots, we see that the first 4 grids have highly varying total volume and rest of the grids have almost same total volume for each activity. This is further verified by performing a set of one-way ANOVA tests [Fig 8].

P-value < 0.05 indicates mean values of the grids are not equal, P-value > 0.05 indicates means values of the grids are equal.

```
#Comparison of mean SMS values of top 4 grids
stats.f_oneway(daily5059.sms.to_list(), daily5161.sms.to_list(), daily6064.sms.to_list(), daily5061.sms.to_list())
```

```
F_onewayResult(statistic=15.6837152166561, pvalue=2.3305180034398526e-09)
```

```
#Comparison of mean SMS values of rest of the grids from the top 10 list
stats.f_oneway(daily5159.sms.to_list(), daily5259.sms.to_list(), daily5262.sms.to_list(), daily4855.sms.to_list(),
               daily4856.sms.to_list(), daily6165.sms.to_list())
```

```
F_onewayResult(statistic=1.0806731220881904, pvalue=0.37063204247098347)
```

```
#Comparison of mean Call values of top 4 grids
stats.f_oneway(daily5059.call.to_list(), daily6064.call.to_list(), daily5161.call.to_list(), daily5159.call.to_list())
```

```
F_onewayResult(statistic=8.245022535234758, pvalue=3.0159552209709387e-05)
```

```
#Comparison of mean Call values of rest of the grids from the top 10 list
stats.f_oneway(daily5061.call.to_list(), daily5259.call.to_list(), daily6165.call.to_list(), daily5262.call.to_list(),
               daily6058.call.to_list(), daily5162.call.to_list())
```

```
F_onewayResult(statistic=2.089333950123052, pvalue=0.06609465518655394)
```

```
#Comparison of mean Internet values of top 4 grids
stats.f_oneway(daily5161.internet.to_list(), daily5059.internet.to_list(), daily5259.internet.to_list(),
               daily5061.internet.to_list())
```

```
F_onewayResult(statistic=8.388137367527829, pvalue=2.5001501890290148e-05)
```

```
#Comparison of mean Internet values of rest of the grids from the top 10 list
stats.f_oneway(daily5258.internet.to_list(), daily5159.internet.to_list(), daily6064.internet.to_list(),
               daily4855.internet.to_list(), daily4856.internet.to_list(), daily5262.internet.to_list())
```

```
F_onewayResult(statistic=0.42971229199857125, pvalue=0.8278755021476163)
```

Fig 8: One-Way ANOVA tests verifying that, except the first 4 grids, rest in the top 10 list have similar mean volumes

Location of top 10 grids with highest volumes in the map shows that they are all from Duomo & Milano Centrale region.

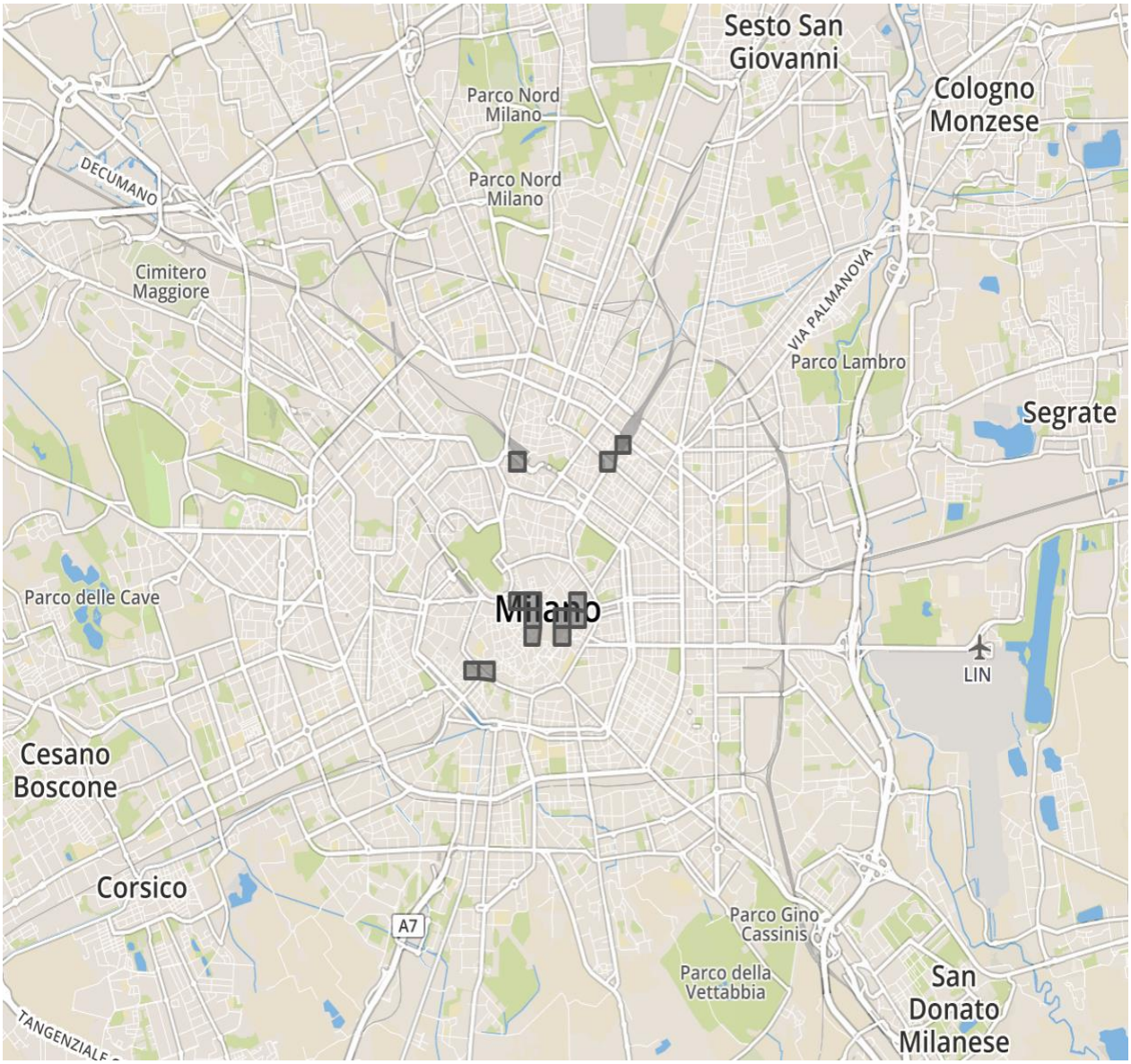


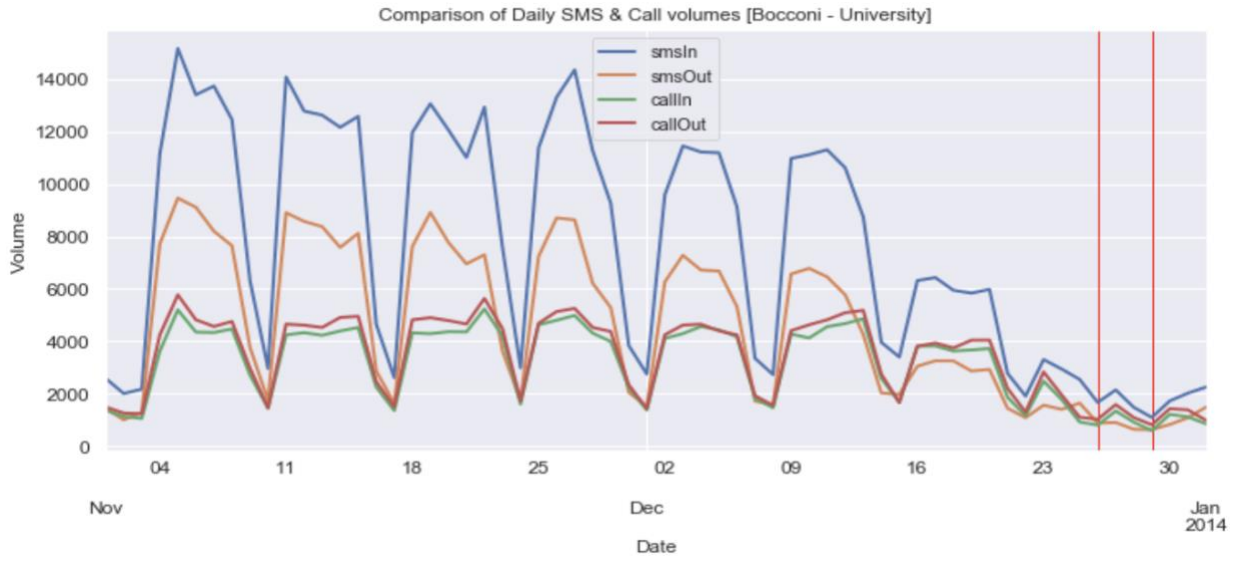
Fig 10: Location of top 10 high-volume grids in the city map

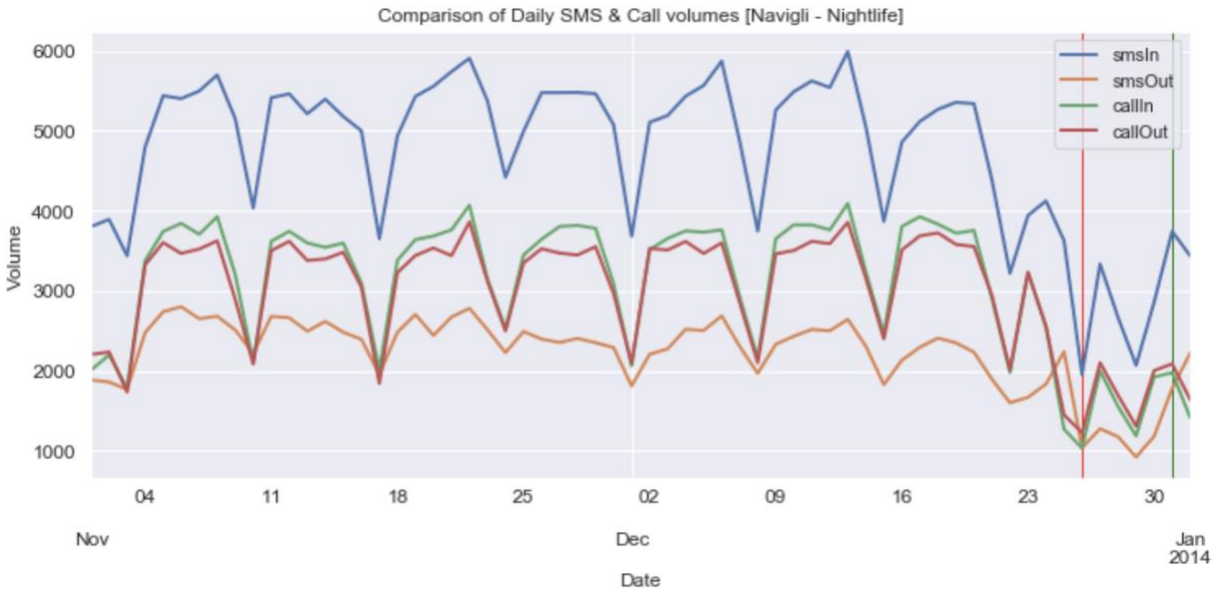
Grid's Telecommunication Activity Pattern

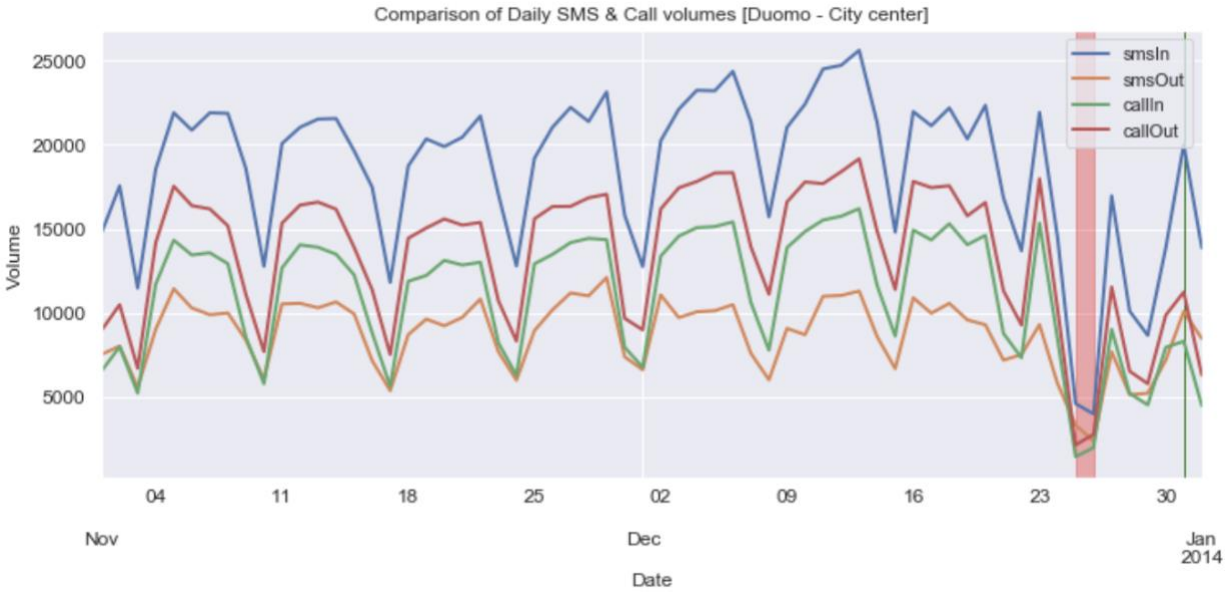
All the top contributing grids are from the city's center and mostly near transport hubs, they are expected to show similar behavior and most of these grids have approximately same mean values. In order to capture variations in the city's telecommunication activities, we will examine the following four grids that has markedly different behavioral signatures,

4459 - Bocconi, one of the most famous Universities in Milan
4456 - Navigli district, one of the most famous nightlife places in Milan
5060 - Duomo, the city center of Milan
5646 - Fiera, residential neighborhood of Milan

Daily Activity Plots







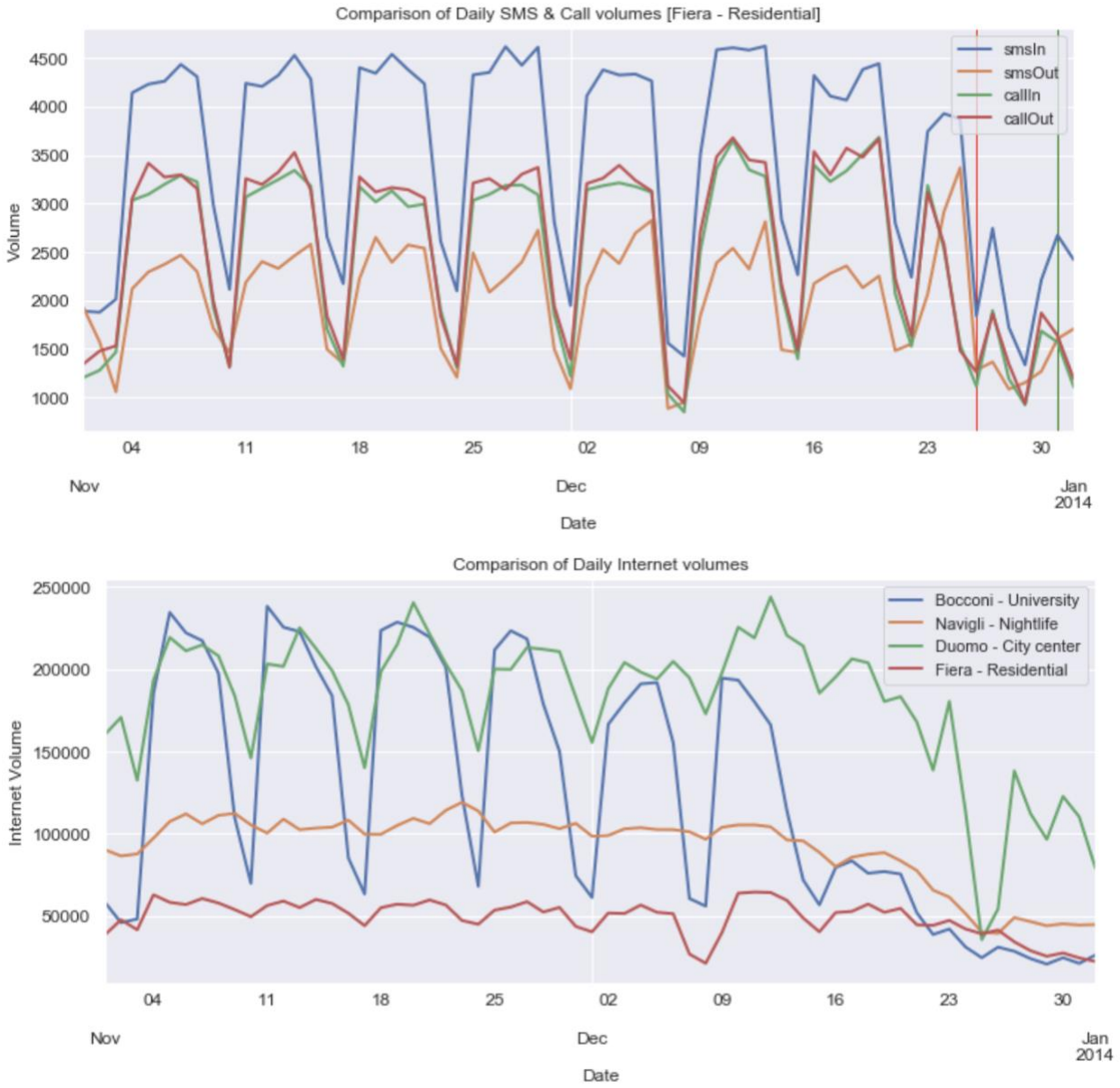


Fig 11: Time-series plot of daily telecommunication activities of the four grids

- All four grids have received high volumes of incoming SMS compared to other activities. Outgoing SMS has the least volume, almost equal amounts of Calls are made and received.
- Duomo has highest volume of all activities, followed by Navigli, then Bocconi and Fiera in the end. We can order the grids based on total volumes as,
  Duomo [city center] > Navigli [nightlife] > Bocconi [university] > Fiera [residential]
- All four grids exhibit seasonality in SMS & Call activities. In internet activity Navigli & Fiera doesn't show any seasonality. This may because of IoT, with many devices always being connected to the network.
- There is a drop in the volumes towards December end in all the plots due to holiday season.

Hourly Activity Plots

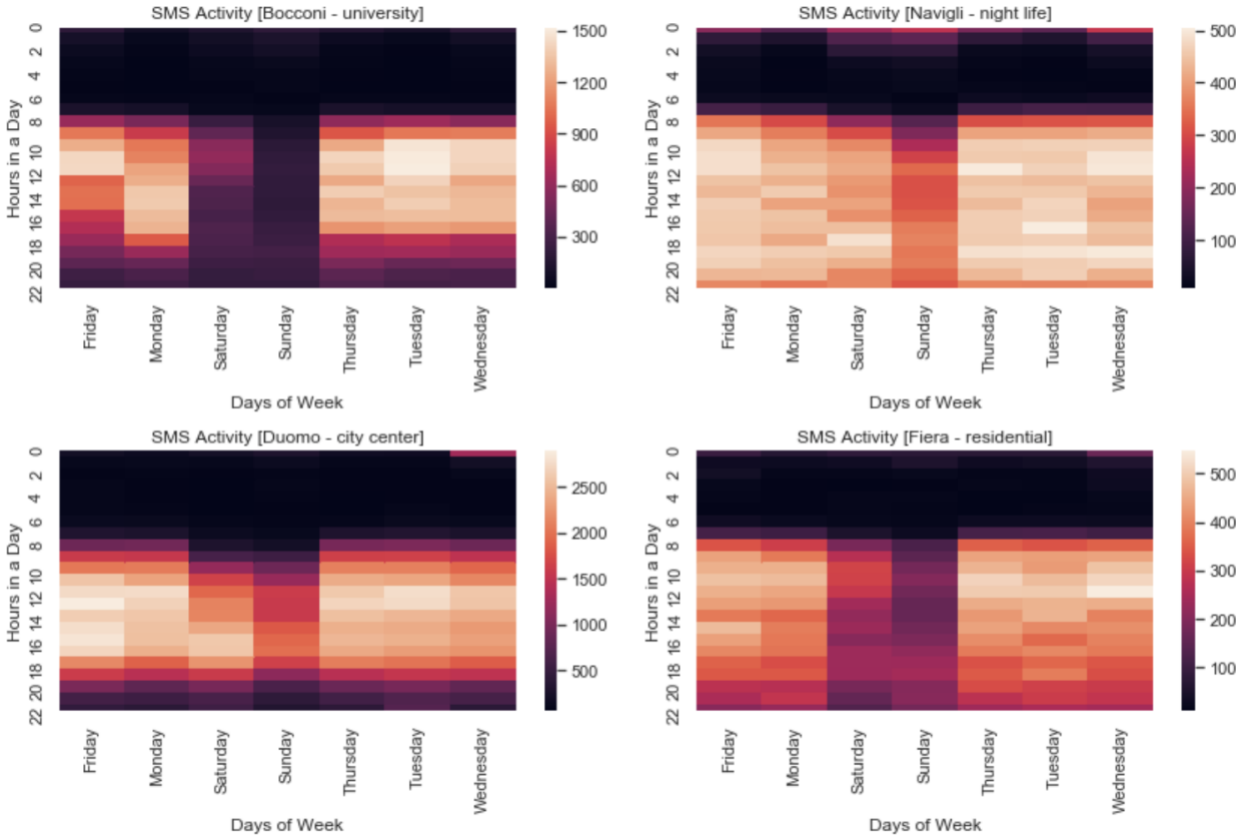


Fig 12: Heat map of hourly SMS activity

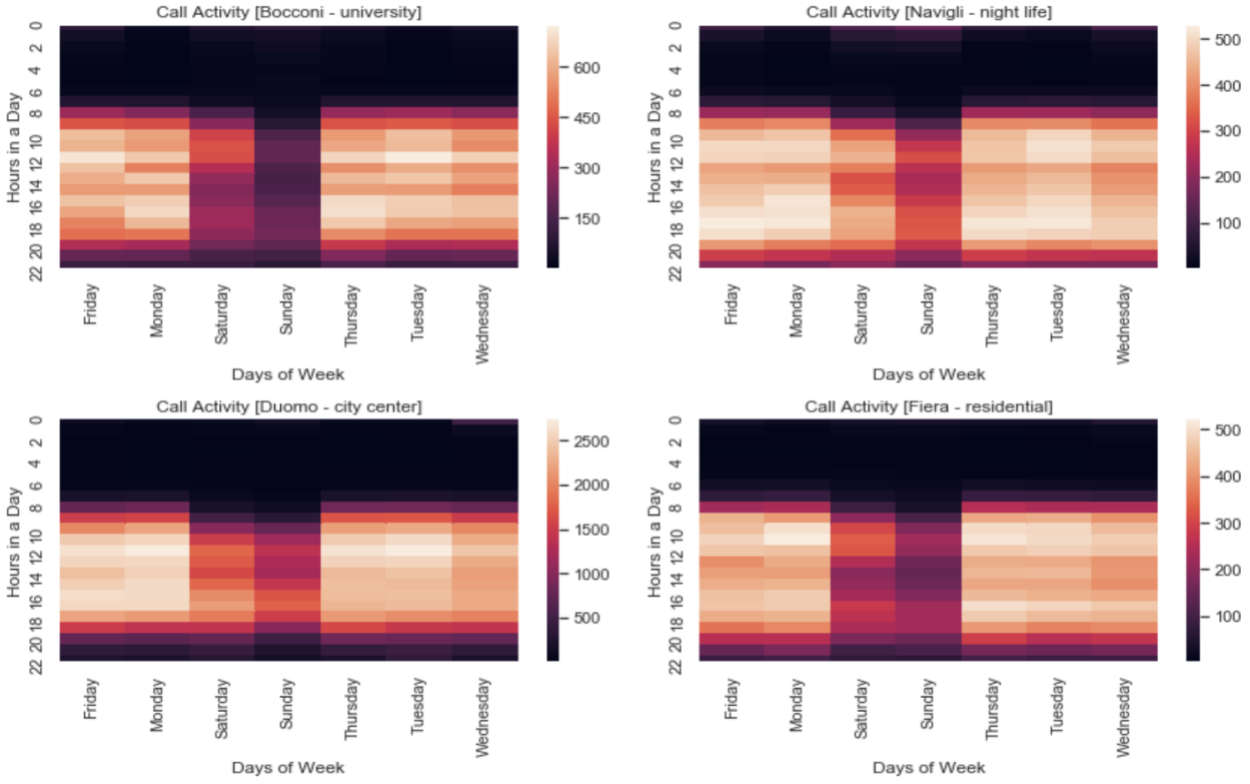


Fig 13: Heat map of hourly Call activity

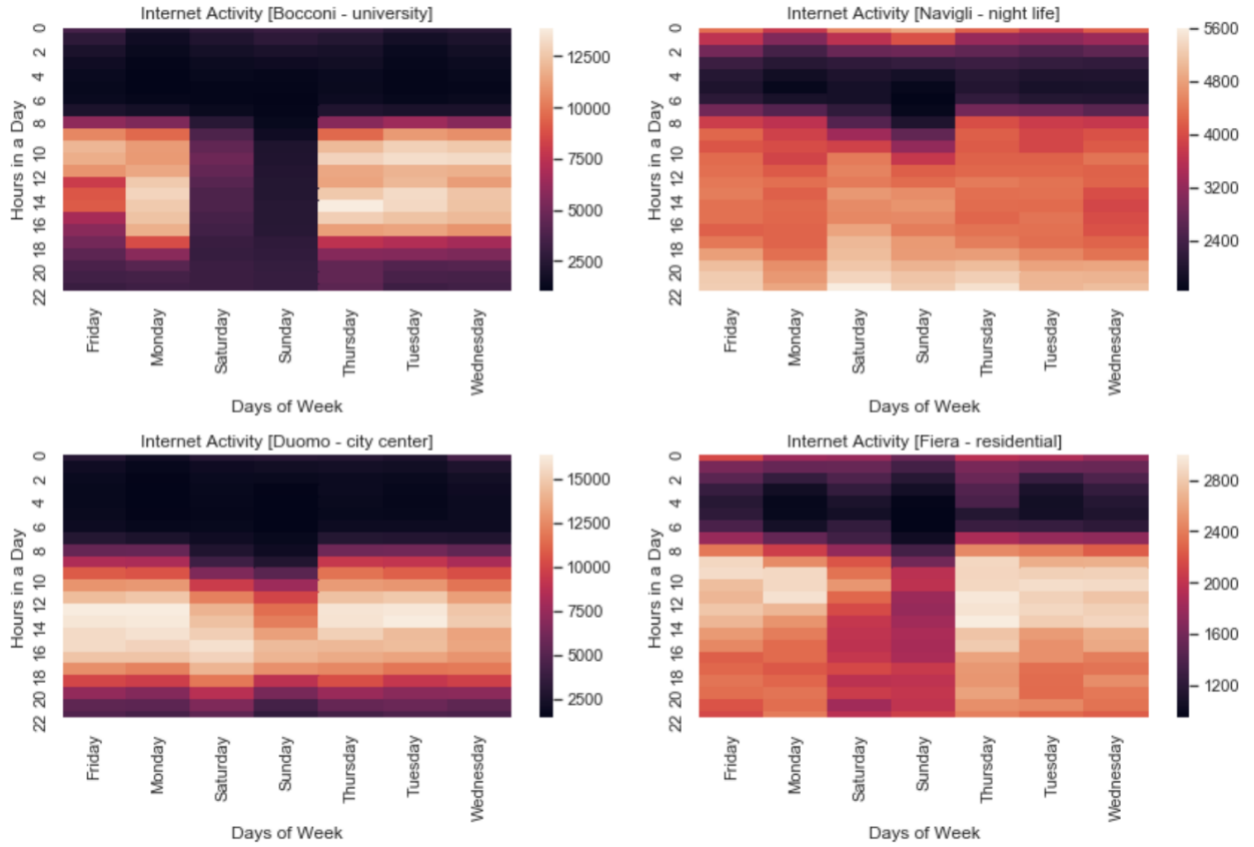


Fig 14: Heat map of hourly Internet activity

Heat maps shows significant differences in behavior of the four grids,

SMS Activity:

- In general, there is less SMS activities during the weekends (Saturday & Sunday).
- Navigli, night life region shows SMS activities until 2 am in the night on weekend.
- Fiera & Navigli regions are very active from 7am till 10pm on all days.
- Duomo, city center has very less SMS activity from 10pm until 7am in the morning on weekdays and 9 am on weekends.
- Bocconi, university shows less SMS activity compared to others.
- Navigli, Duomo & Fiera shows a bright region on Wednesday 12am, this must be due to New Eve falling on Tuesday.

Call Activity:

- Call activity has similar pattern as SMS, but lesser volumes.
- Surprisingly, there is no significant call activity on New Year eve. This shows how people are more connected via SMS and internet these days. Another possibility is that calls may have been made via internet.

Internet:

- Navigli & Fiera has internet activities almost all through the night. Even Bocconi, university shows some sparse activity after midnight on weekends.
- Duomo, city center although has the highest internet volumes, shows a steady pattern for all activities, 8am – 10pm on weekdays and 10am to 10pm on weekends.

Day of Week Plots

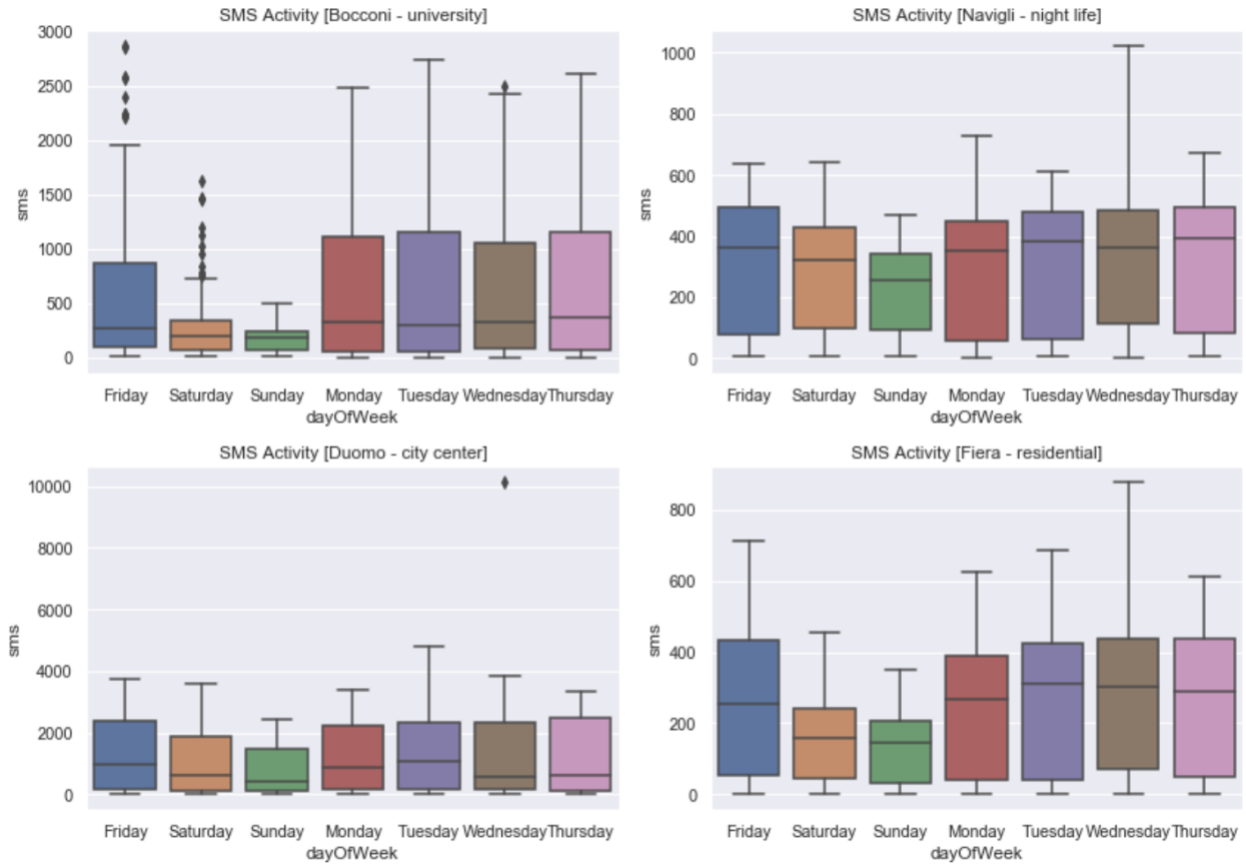


Fig 15: Box plot of SMS activity for each day of the week

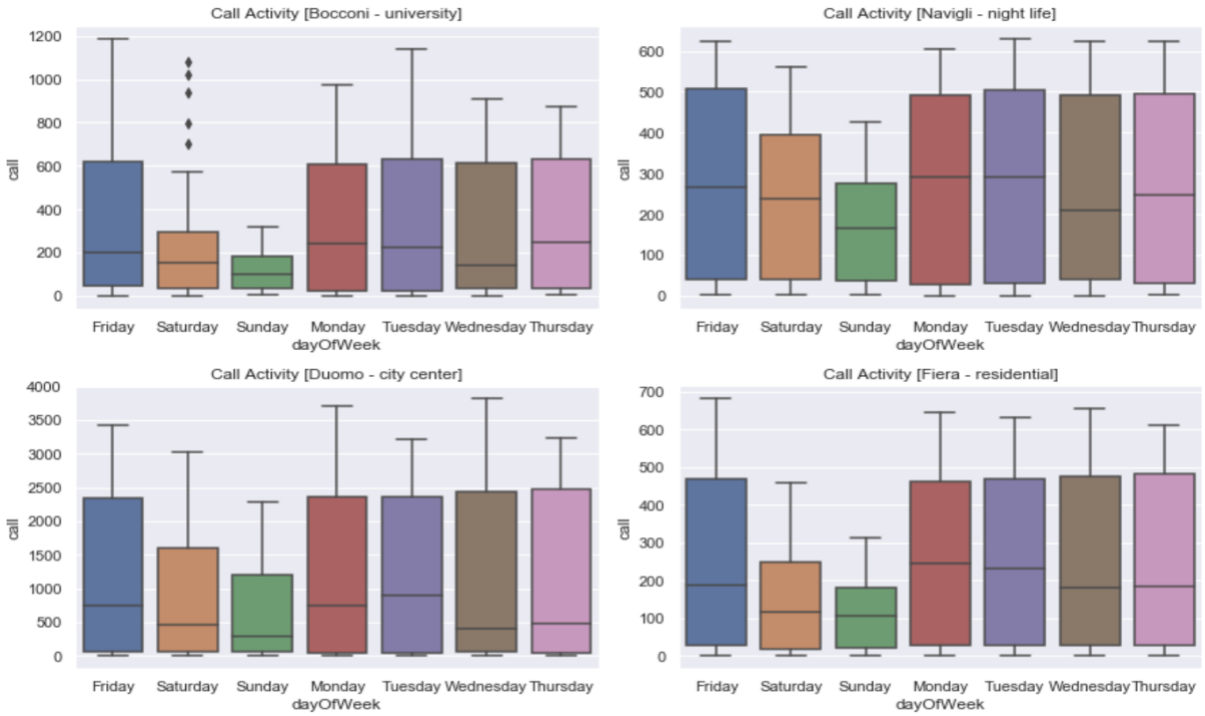


Fig 16: Box plot of Call activity for each day of the week

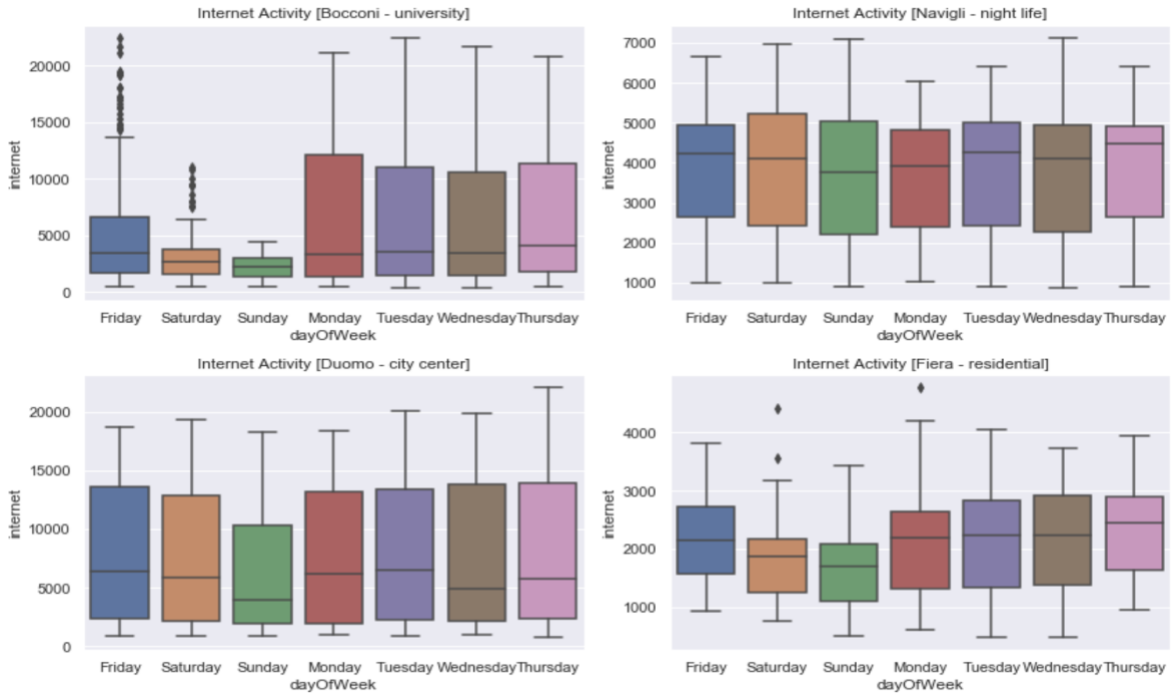


Fig 17: Box plot of Internet activity for each day of the week

SMS Activity:

- Bocconi has outliers on Friday & Saturday.
- Least volumes of SMS on Sunday for all four grids.
- Also, Bocconi shows significant difference between weekday and weekend volumes, followed by Fiera.

Call Activity:

- Again, Bocconi has some outliers on Saturday.
- Navigli & Fiera has almost equal call volumes on weekdays. They differ quite a lot on weekends.

Internet Activity:

- Navigli has approximately same median for all days.
- Bocconi also has same medians but the amount of activity varies for each day.

4. 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 data that helps Clustering algorithm to understand each grid better.

We have created a total of 101 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	Minimum volume of November month SMS, Calls & Internet

internetMin_Nov	
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

5. MACHINE LEARNING: 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 candidates for K, K=6 & K=7 beyond which the plot plateaus.

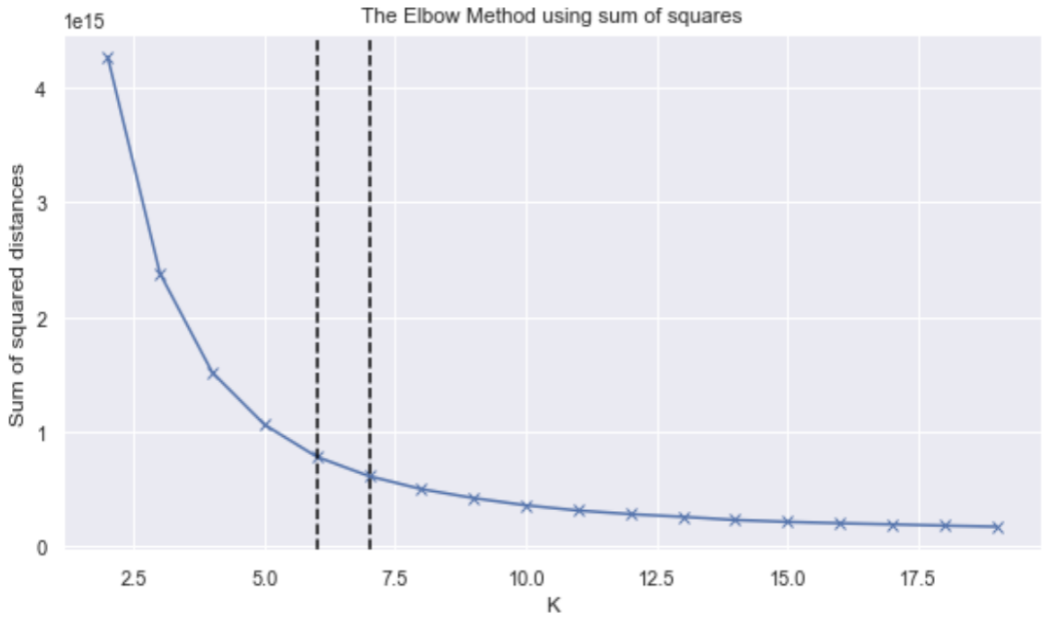


Fig 18: Elbow method to find the optimal K value

Kneed package

Visually inspecting the plot to identify the Knee/Elbow point could be confusing sometimes, as in our case. We will take help of Kneed package that mathematically computes the Knee/Elbow point.

For our dataset, Kneed package has returned K=6 as the knee point.

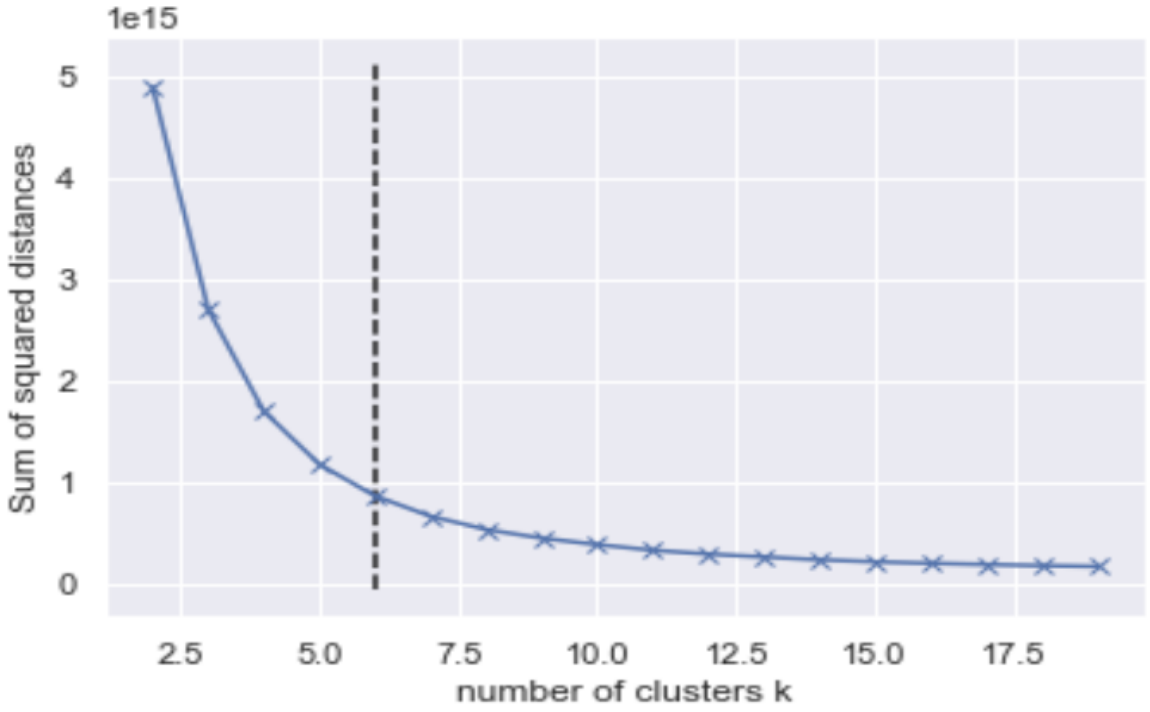


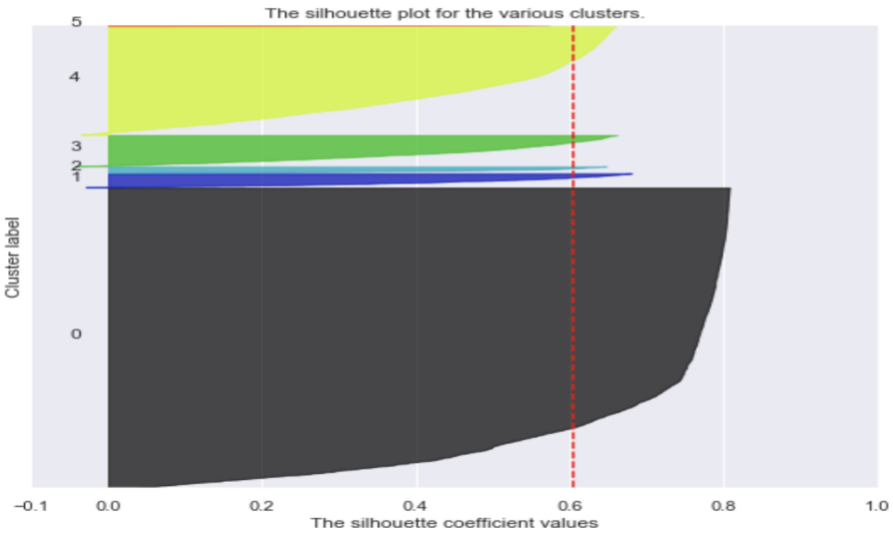
Fig 19: Kneed package output shows K=6

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 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 compared to K=7.

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




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

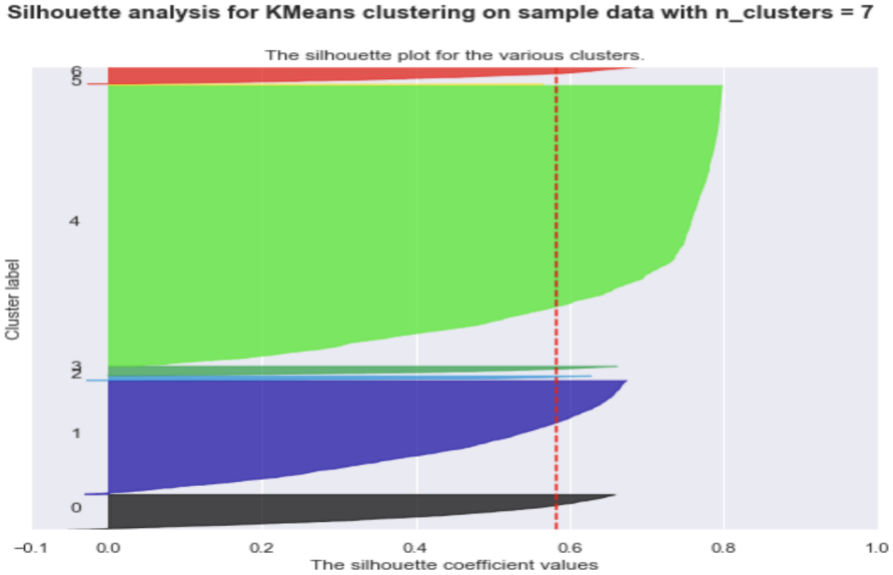
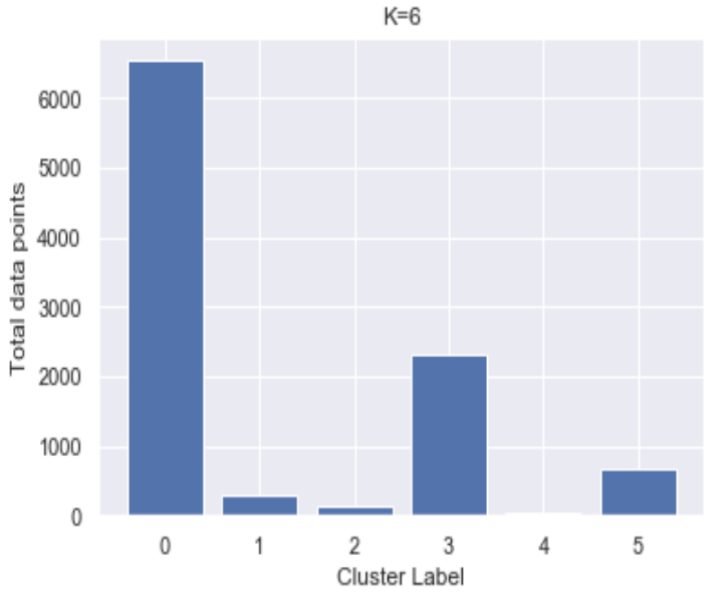


Fig 20: Plot of Silhouette coefficients of all the data points for K=6 & K=7

K-Means Clustering

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



Cluster Label	Number of Grids
0	6541
1	300
2	139
3	2324
4	31
5	665

Fig 21: Subgroups after K-Means Clustering with K=6

Visualization of K Clusters

Principal Component Analysis

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

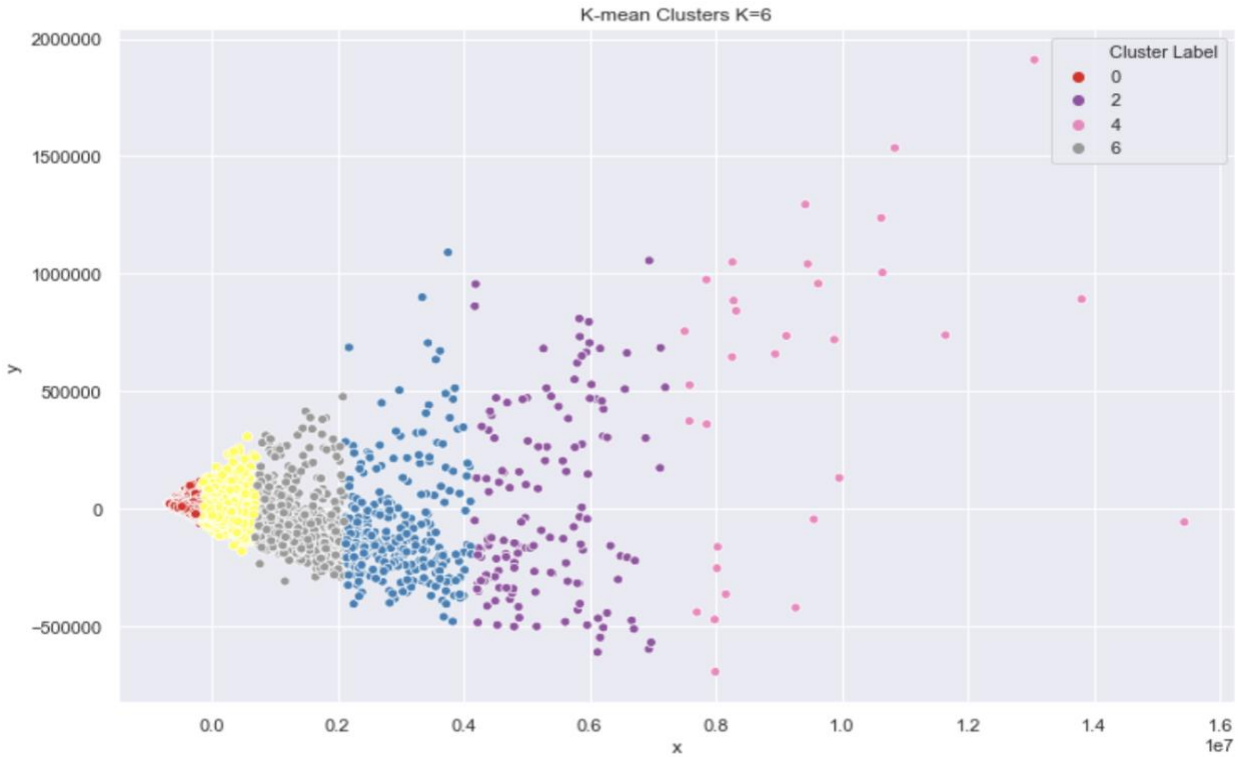


Fig 22: Visualization of the datapoints in its subgroups

GEOJSON

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. The subgroups are created based on the volumes of telecommunication activities in each grid. Notice how all our top 10 grids are in the red grids. They are also likely to be related to the population distribution of the region.

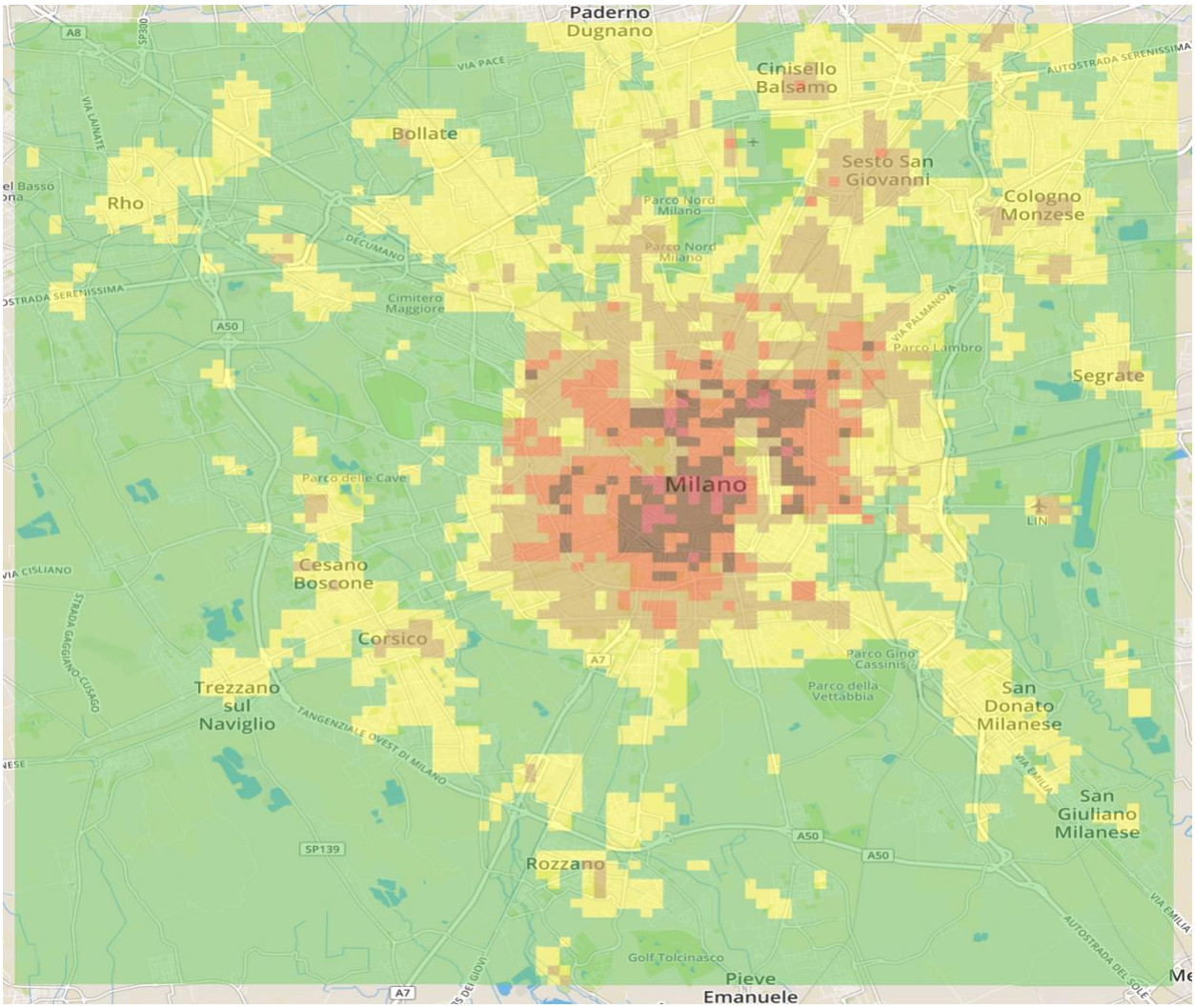


Fig 23: Visualization of 10000 grids clustered into 6 subgroups

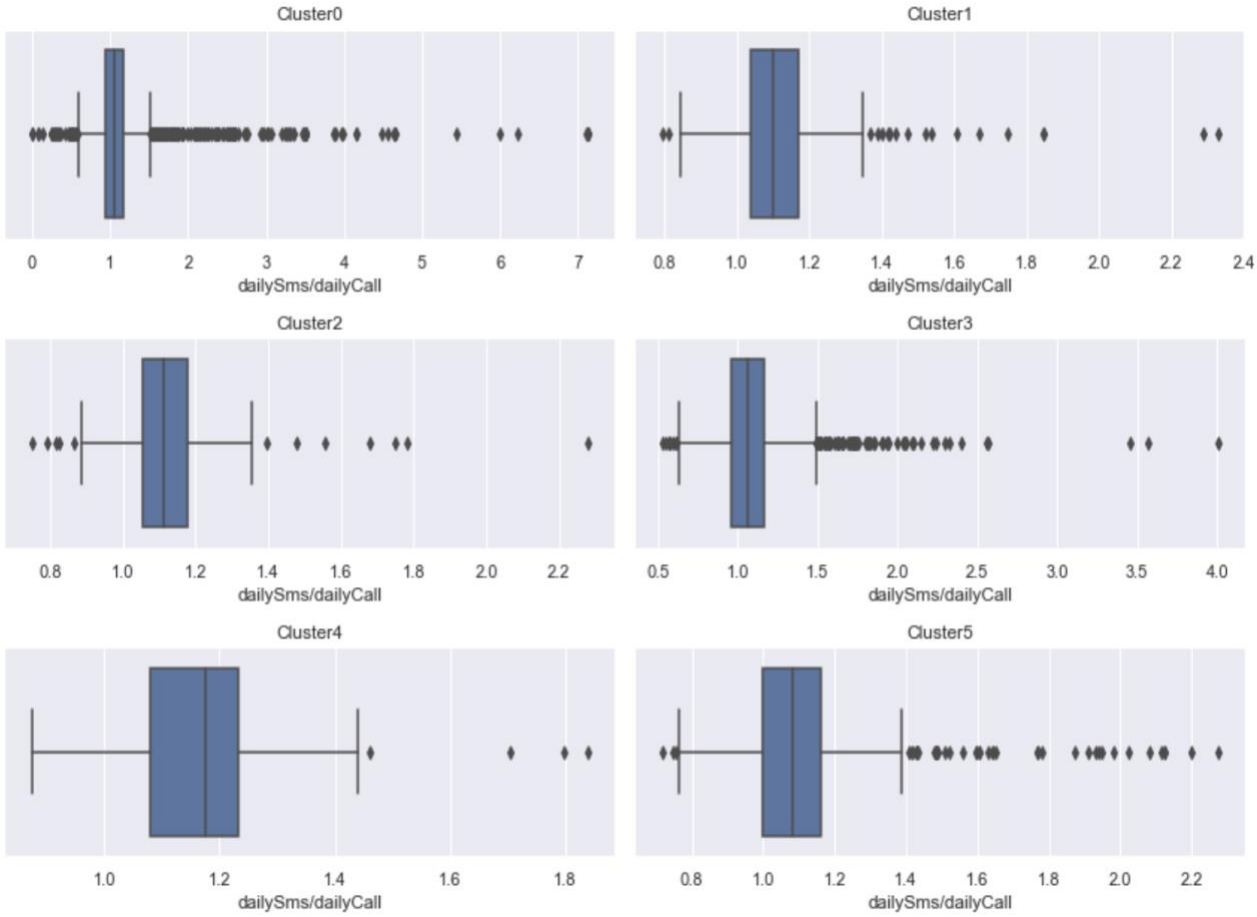
Rank	City	Population	Area (km²)	Density (inhabitants / km²)	Altitude (mslm)
1st	Milan	1336879	181.76	7355.2	122
2nd	Sesto San Giovanni	81750	11.74	6963.4	140
3rd	Cinisello Balsamo	74536	12.7	5869	154
4th	Legnano	59492	17.72	3357.3	199
5th	Rho	51033	22:32	2286.4	158
6th	Cologno Monzese	47880	8:46	5659.6	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 24: Largest municipalities by population of Milan [sourced from Wikipedia]

Analysis of K 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.

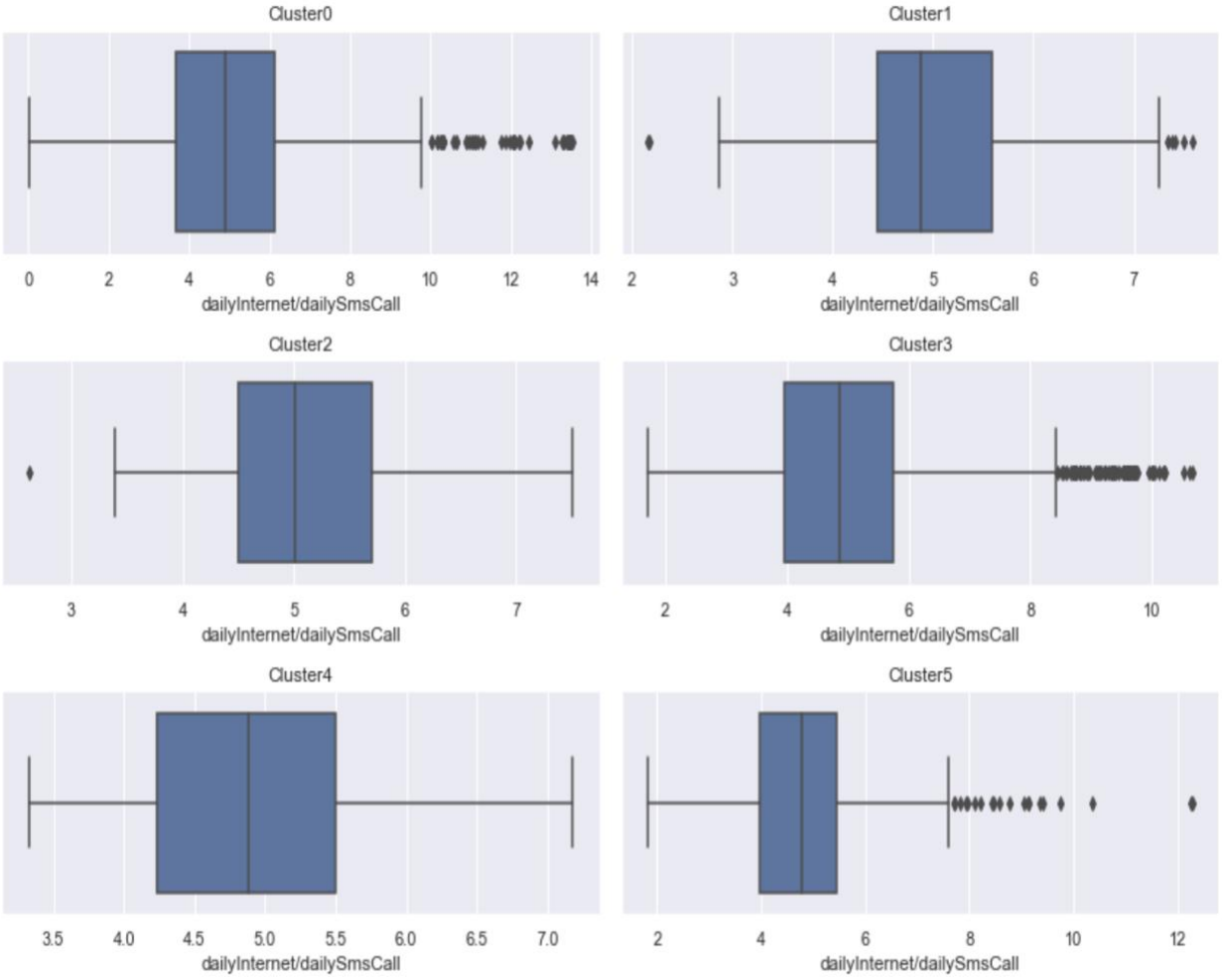
T-tests and one-way ANOVA tests can be performed on the subgroups for different dimensions in order to understand the similarities and dissimilarities in their behavioral patterns.




```
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=2.90980079630057, pvalue=0.012516645020791636)
```

Fig 25: Box-plot of 6 subgroups and its dailySms/dailyCall feature. One-Way ANOVA test shows different mean value for all the groups




```
#One-way ANOVA test shows mean of dailyInternet/dailySmsCall is same for all the groups
group0 = Cluster0['dailyInternet/dailySmsCall'].to_list()
group1 = Cluster1['dailyInternet/dailySmsCall'].to_list()
group2 = Cluster2['dailyInternet/dailySmsCall'].to_list()
group3 = Cluster3['dailyInternet/dailySmsCall'].to_list()
group4 = Cluster4['dailyInternet/dailySmsCall'].to_list()
group5 = Cluster5['dailyInternet/dailySmsCall'].to_list()
stats.f_oneway(group0,group1,group2, group3, group4, group5)
```

```
F_onewayResult(statistic=1.5325777809082444, pvalue=0.17592855661097262)
```

Fig 26: Box-plot of 6 subgroups and its dailyInternet/dailySmsCall feature. One-Way ANOVA test shows approximately same mean value all the groups

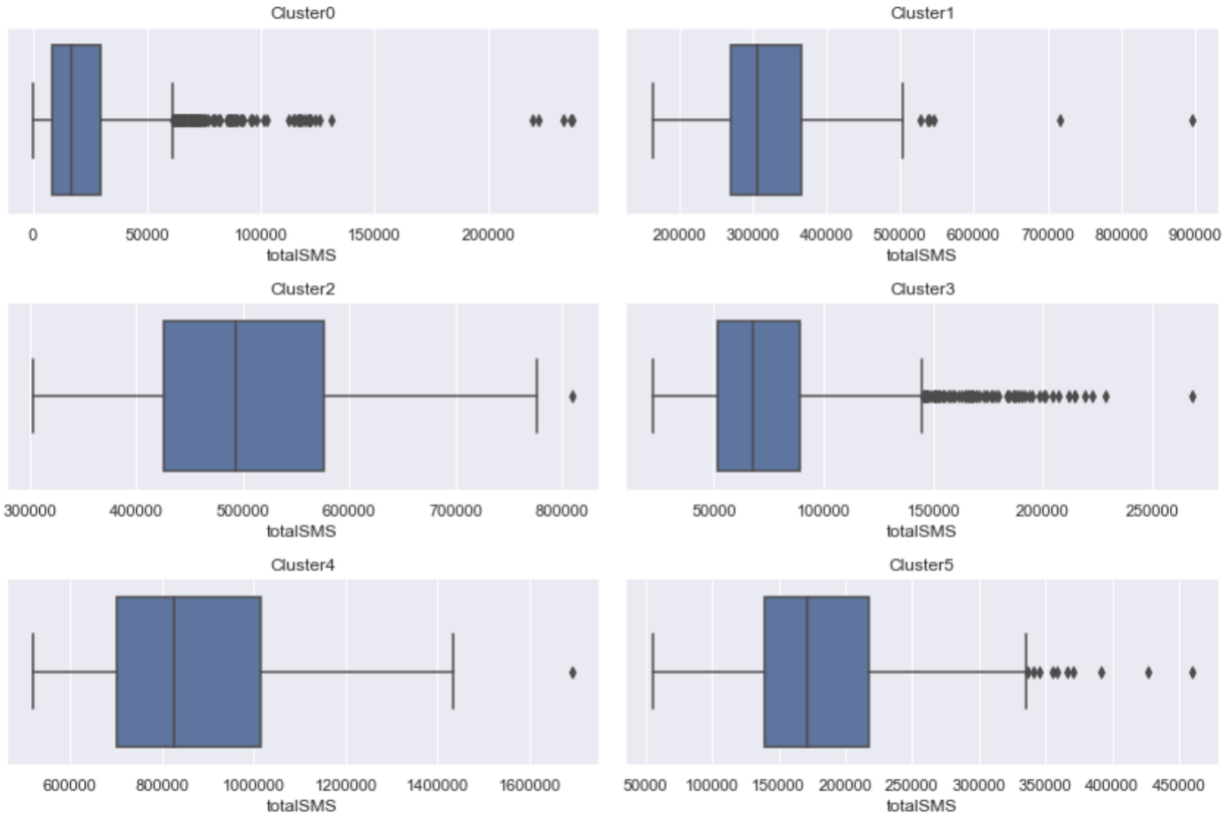


Fig 27: Box-plot of 6 subgroups and its totalSMS feature

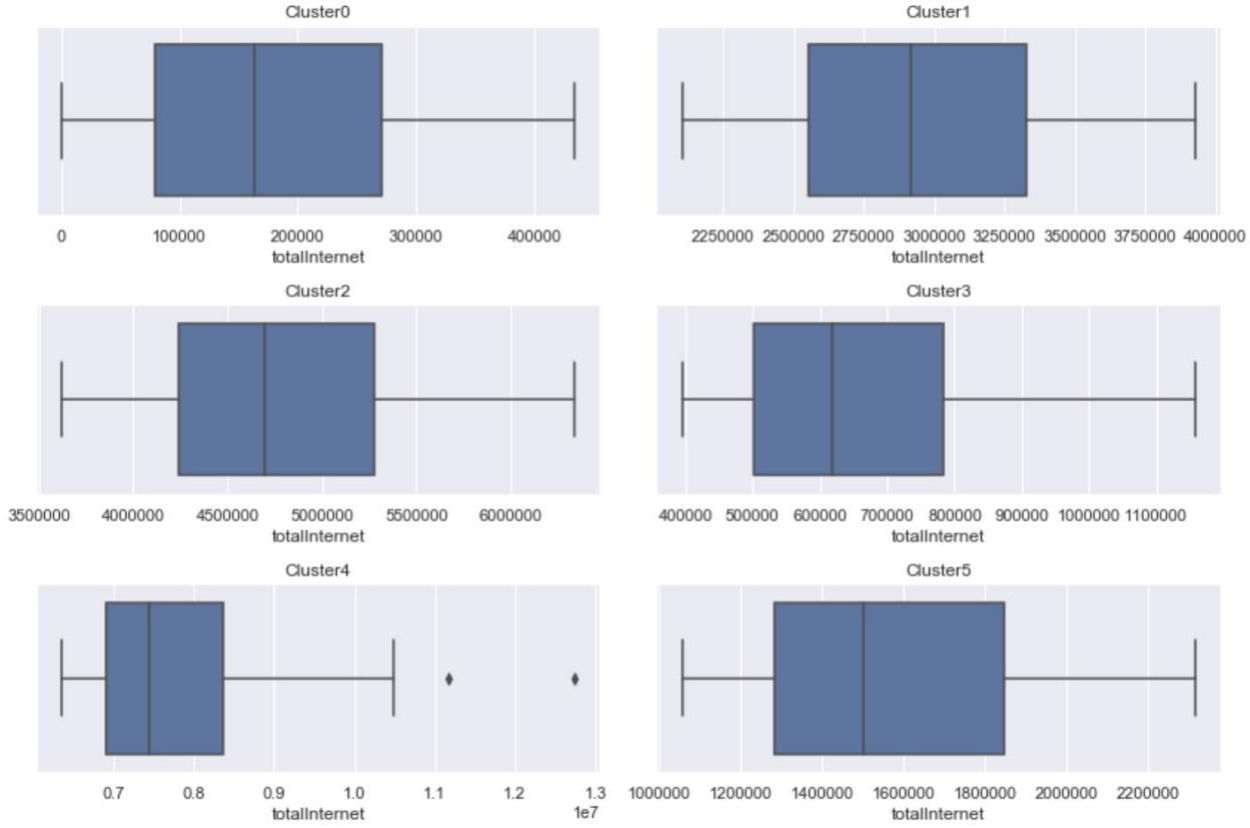


Fig 28: Box-plot of 6 subgroups and its totalCall feature.

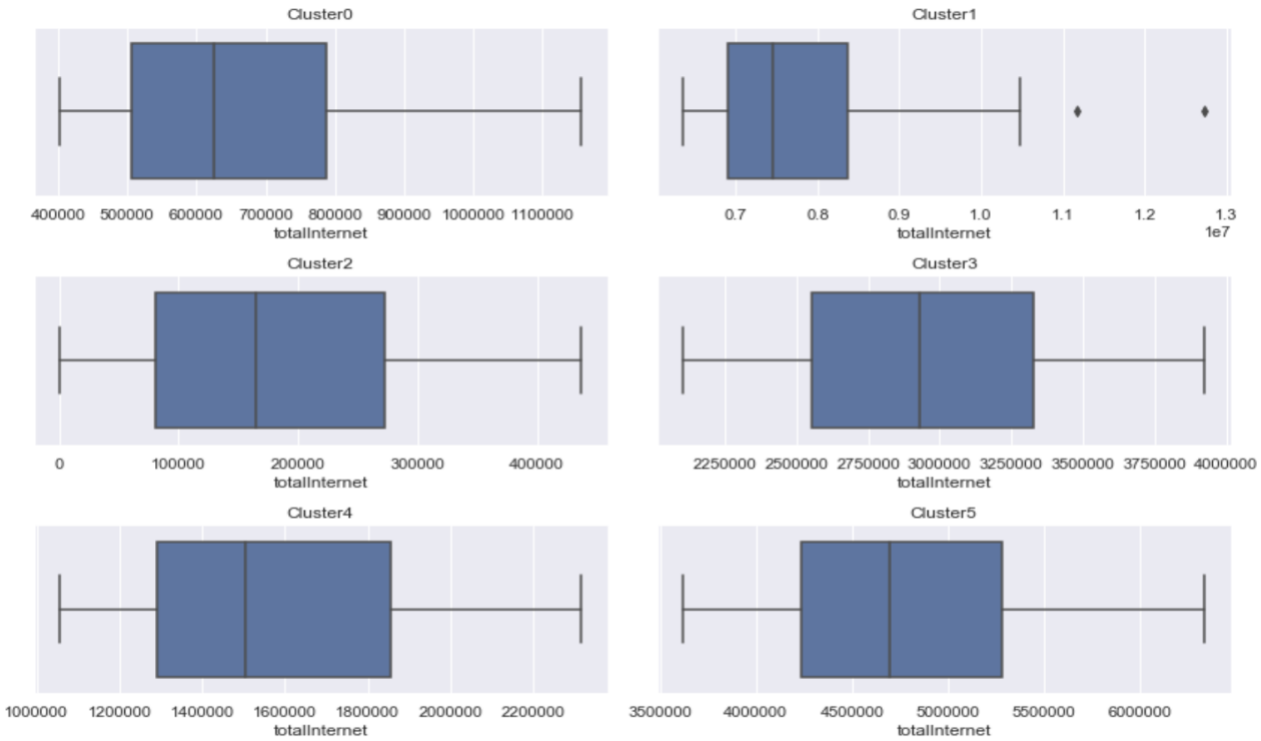


Fig 29: Box-plot of 6 subgroups and its totalInternet feature

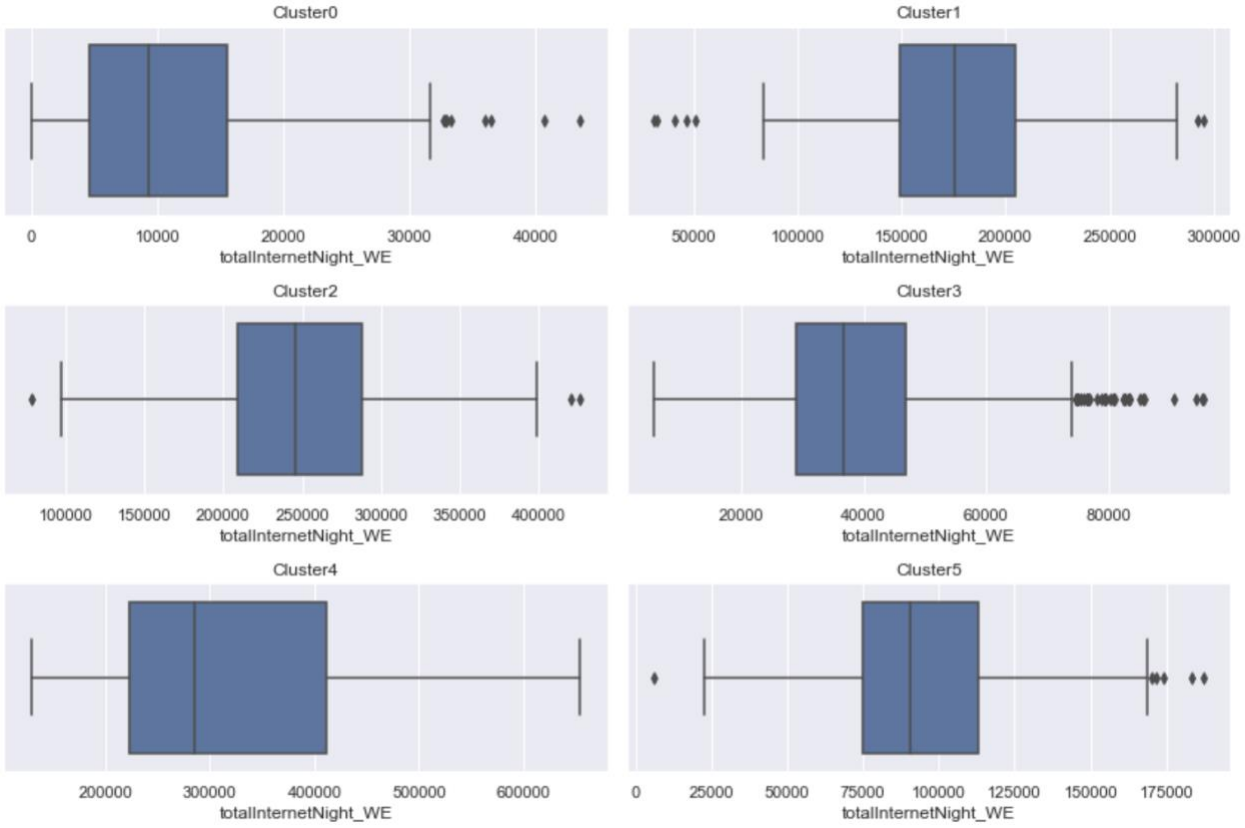


Fig 30: Box-plot of 6 subgroups and its totalInternetNight_WE feature

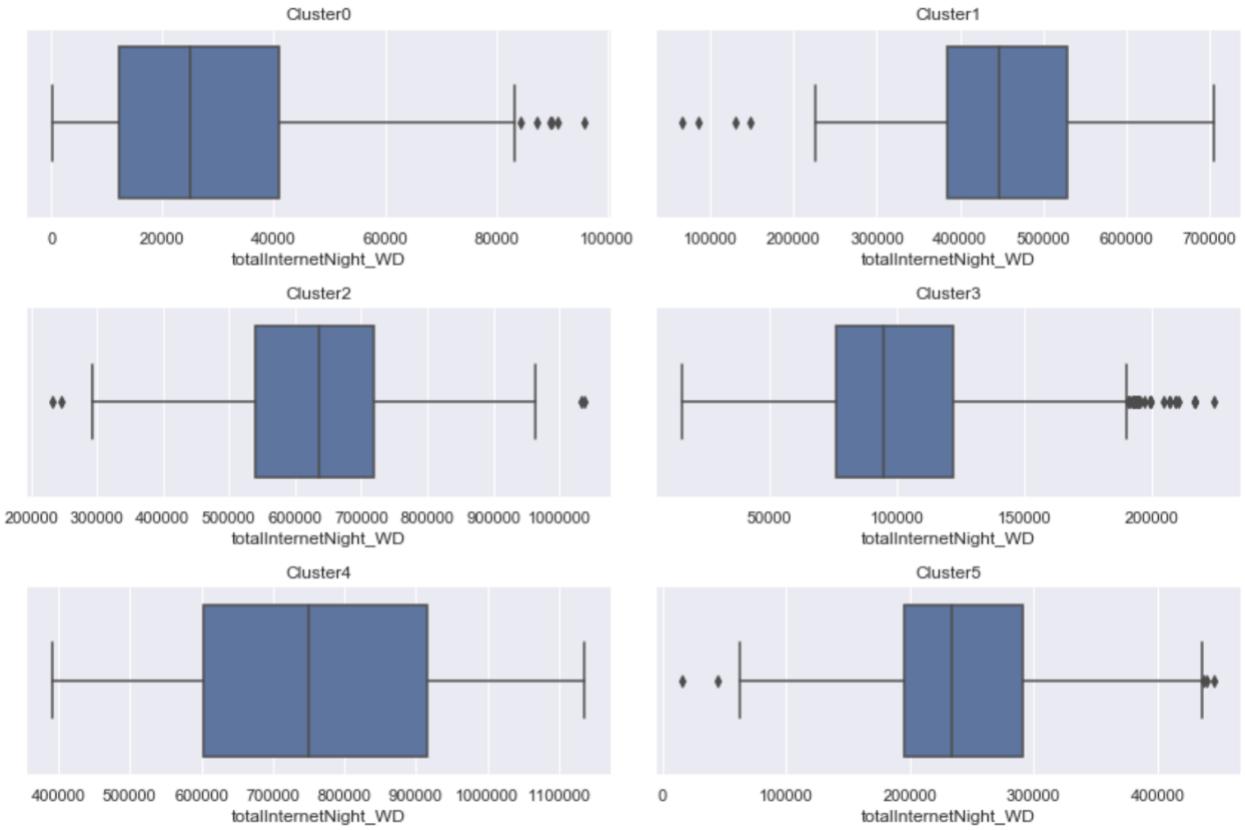


Fig 31: Box-plot of 6 subgroups and its totalInternetNight_WD feature

6. MACHINE LEARNING: TIME SERIES FORECASTING

Our dataset provides 62 days of historical data that is temporally aggregated every 10 min. Each observation holds information about the volume of telecommunication activities performed over the next 10 min. We have resampled and aggregated this data to daily frequency and hourly frequency.

SARIMA Model

We will be forecasting 7 days of daily Internet volumes for grid 5161, which is is very close to Duomo, city center and it generates highest volumes of Internet activity compared to other grids. Hourly volumes display multiple seasonality, 24 hours & 7 days and the model that we will be using does not support multiple seasonality.

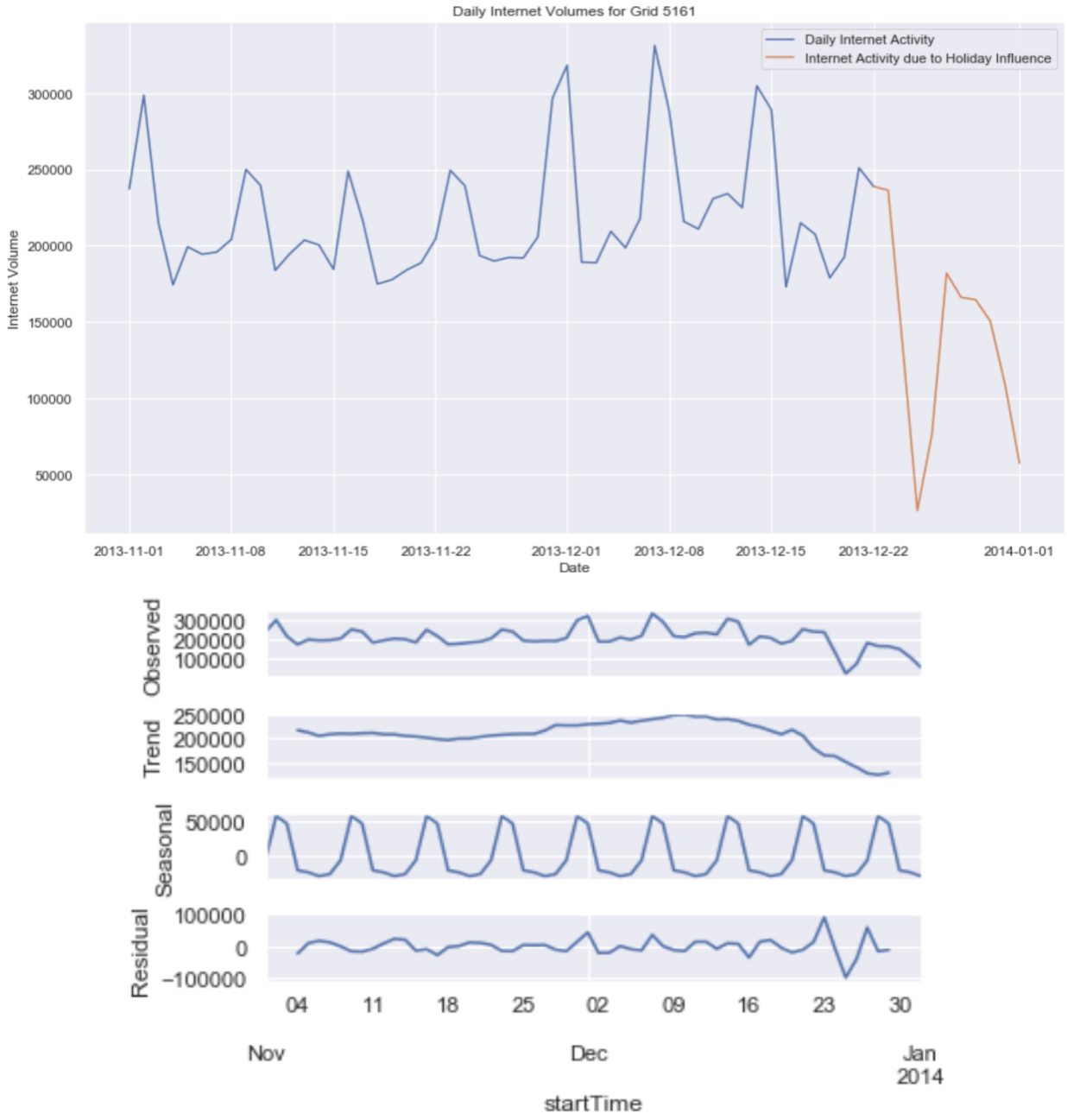


Fig 32: Time Series and time series decomposition plot of daily internet volumes for grid 5161

Time Series Decomposition

Plot of daily volumes and its time series decomposition in Fig 31 & Fig 32 shows seasonal effect for every 7 days. Volumes reaches new peaks during weekends and goes down during weekdays. In order to incorporate this seasonality, we will use ARIMA model with seasonal component, SARIMA [Seasonal Autoregressive Integrated Moving Average]. Also, the trend starts to drop after Dec 23rd due to the influence of holiday.

Hyperparameter Tuning

Augmented Dickey Fuller Test (ADF) is unit root test for stationarity.

```
from statsmodels.tsa.stattools import adfuller
results = adfuller(daily5161Internet.internet)
results
```

```
(-2.1600607165764596,
 0.22105898340125635,
 11,
 50,
 {'1%': -3.568485864, '5%': -2.92135992, '10%': -2.5986616},
 1193.4904300056205)
```

1st element of the result is the p-value which is > 0.05, hence our dataset is not stationary. A first order differencing will fix it.

```
daily5161Internet_diff = daily5161Internet.diff().fillna(0)
results = adfuller(daily5161Internet_diff.internet)
results
```

```
(-8.607627828769125,
 6.606378535556466e-14,
 4,
 57,
 {'1%': -3.5506699942762414,
  '5%': -2.913766394626147,
  '10%': -2.5946240473991997},
 1196.1746673623513)
```

1st element of the result is the p-value which is < 0.05, so we reject the null hypothesis of Augmented Dicky fuller test. Dataset is stationary now.

We will set d =1 and rest of the hyperparameters of SARIMA model is obtained using auto_arima function from package pmdarima.

The downward trend from Dec 23, 2013 to Jan 1, 2014 will result in skewed forecasting for the next 7 days, Fig 33. We should not be seeing volumes as low as forecasted and also the model mistakes the drop on Jan 1st as a weekend behavior. But we do not have data from previous years to see how the pattern is for the first week of January to understand the exact behavior. Hence, we will assume that Holiday effects will wear out during first week of January.

	internet	dayOfWeek
2014-01-02	83028.672698	Thursday
2014-01-03	145905.397964	Friday
2014-01-04	132180.349562	Saturday
2014-01-05	128249.066642	Sunday
2014-01-06	116618.428930	Monday
2014-01-07	88507.877210	Tuesday
2014-01-08	53729.928917	Wednesday

Fig 33: Skewed forecast for first week of January due to holiday influence

Our approach to this situation will be as follows,

1. Remove data that shows heavy holiday influence which is from Dec 23, 2013 to Jan 1, 2014. In our dataset these are outlier data points. [Fig 34]
2. Once we remove values from Dec 23, 2013 to Jan 1, 2014, it needs to be filled with some data before forecasting for first week of January. Hence, we will predict volumes for these days as though they were regular days with no holiday effect. [Fig 35]
3. Now with this new dataset we will forecast for the first week of January.[Fig 37]

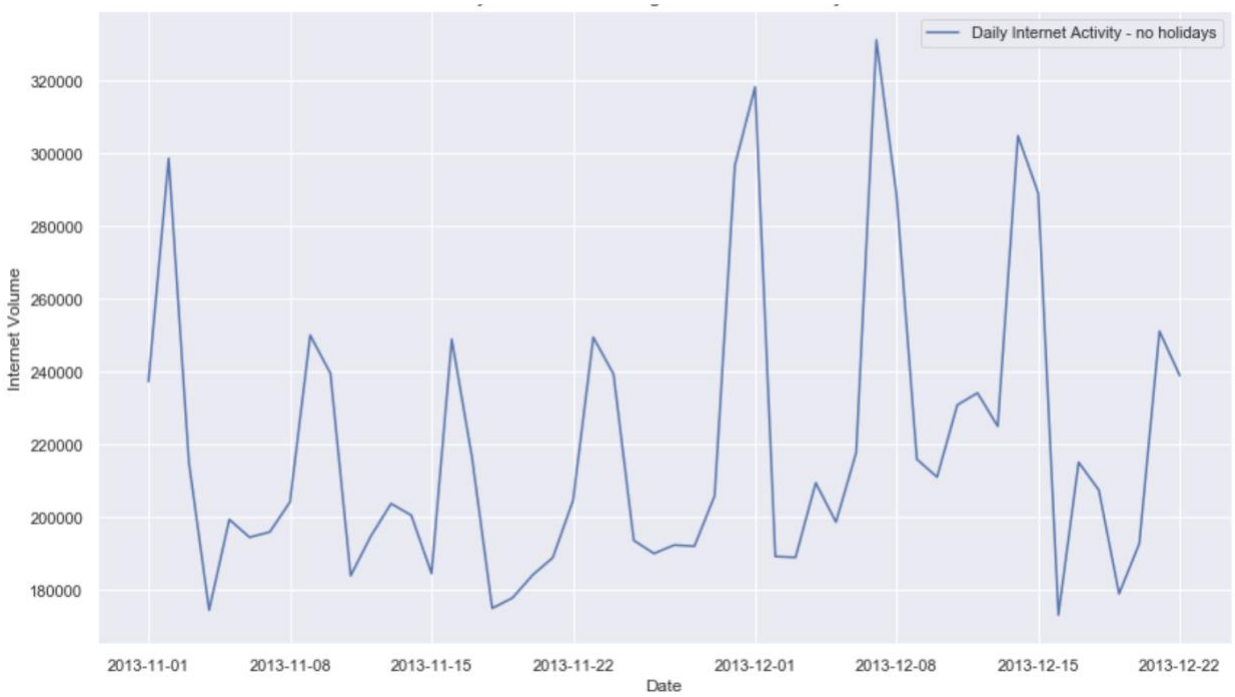
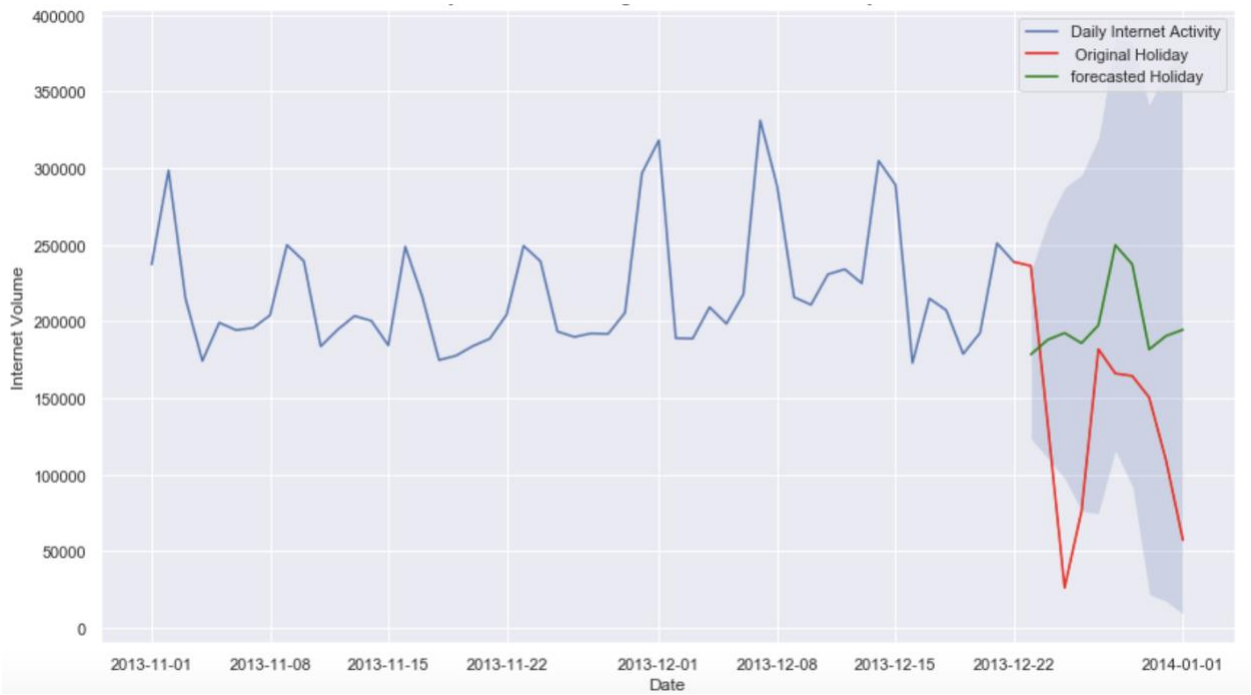


Fig 34: Time Series plot of daily internet volumes for grid 5161 with deleted holiday influence data



	internet	dayOfWeek
2013-12-23	178508.373305	Monday
2013-12-24	188075.521499	Tuesday
2013-12-25	192507.131925	Wednesday
2013-12-26	185892.844800	Thursday
2013-12-27	197541.708903	Friday
2013-12-28	249871.132468	Saturday
2013-12-29	237343.944442	Sunday
2013-12-30	181770.646094	Monday
2013-12-31	190579.683204	Tuesday
2014-01-01	194660.128067	Wednesday

Fig 35: Holiday data being replaced by predicted data

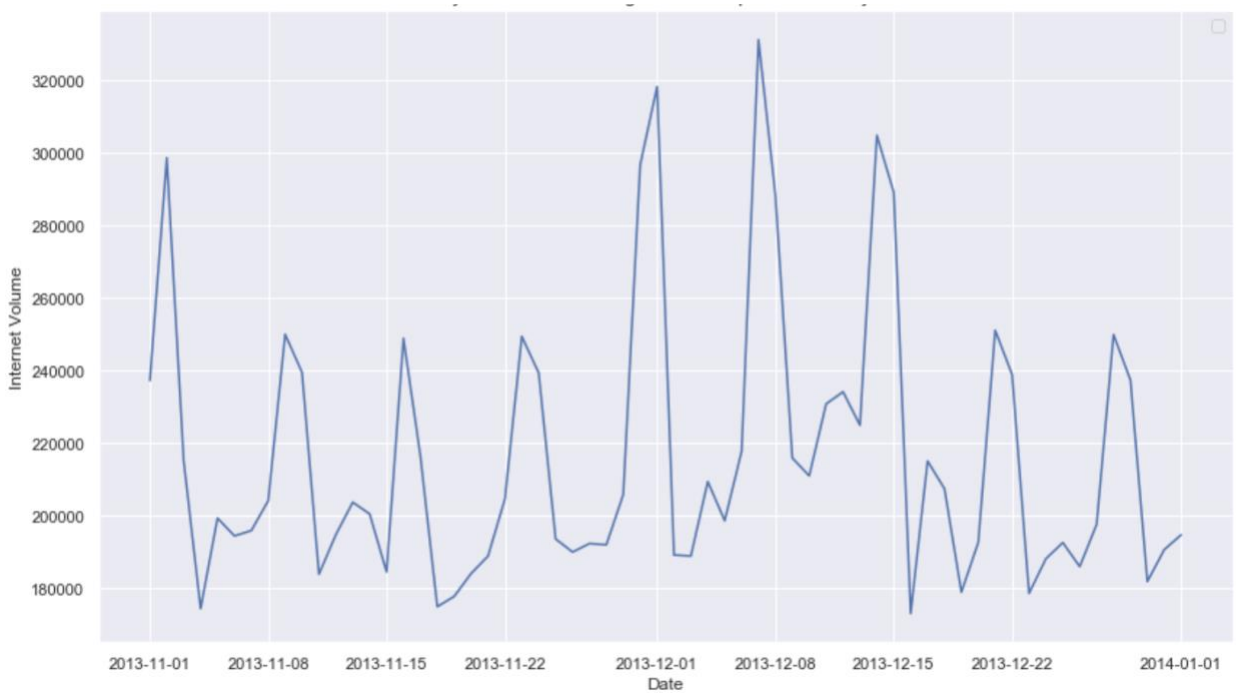
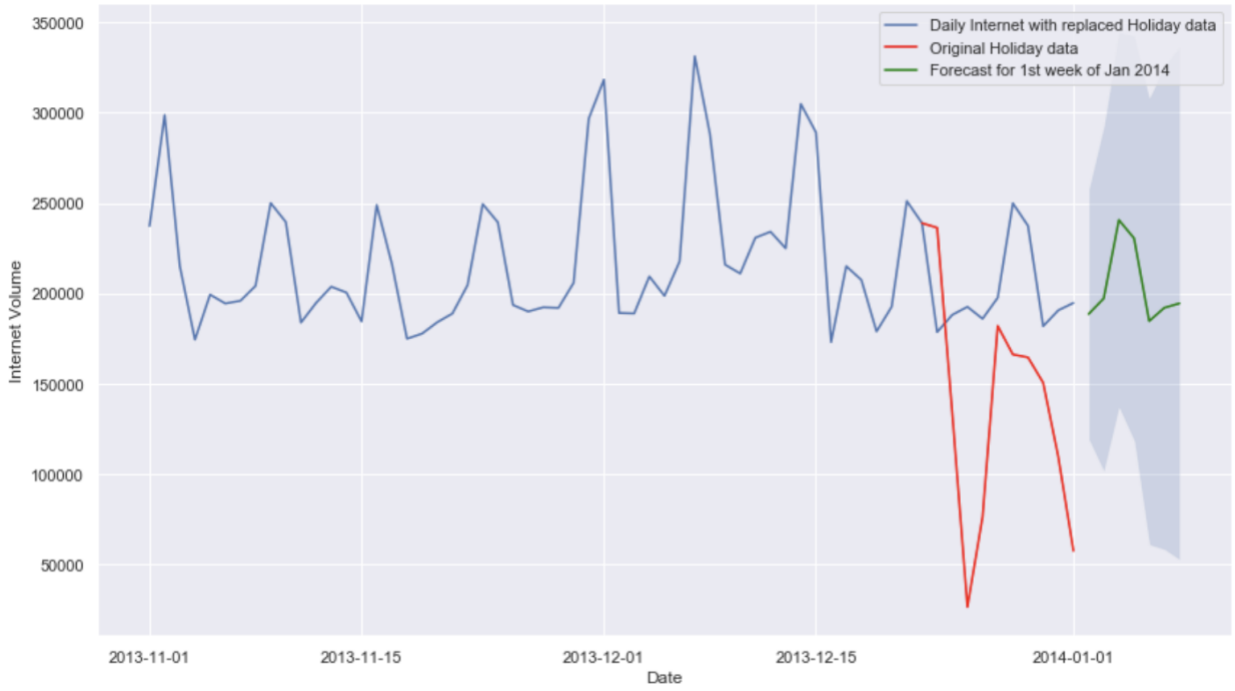


Fig 36: Time Series plot of daily internet volumes for grid 5161 with predicted holiday data



	internet	dayOfWeek
2014-01-02	188492.368833	Thursday
2014-01-03	197161.126424	Friday
2014-01-04	240587.960501	Saturday
2014-01-05	230495.113264	Sunday
2014-01-06	184610.400177	Monday
2014-01-07	191983.028152	Tuesday
2014-01-08	194410.997532	Wednesday

Fig 37: Time Series forecast of daily internet volumes for grid 5161 for first week of January,2014 with replaced holiday data

Model Evaluation

After identifying the parameter values using auto_arima(), we fit our model on the new dataset, and use it to evaluate its accuracy by predicting volumes for days whose volumes are known.

```
MAPE = 5.12%
Accuracy = 94.88%
The Root Mean Squared Error of our prediction is 15270.31
```

Fig 38: Model Evaluation

7. CITATIONS

- Barlacchi, G. et al. A multi-source dataset of urban life in the city of Milan and the Province of Trentino. Sci. Data2:150055 doi: 10.1038/sdata.2015.55 (2015).
- Telecom Italia, 2015, "Telecommunications - SMS, Call, Internet - MI", https://doi.org/10.7910/DVN/EGZHFV, Harvard Dataverse, V1
- Telecom Italia, 2015, "Milano Grid", https://doi.org/10.7910/DVN/QJWLFU, Harvard Dataverse, V1

8. REFERENCES

- How to use geojson - https://www.twilio.com/blog/2017/08/geospatial-analysis-python-geojson-geopandas.html
- Code for Silhouette Score Plot - https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html