Applied Data Science Project Report

Analysis of Ridership of MTA Subway Turnstile Data with Data Science

REPORT BY:

VAIDEHI VIDHYADHAR THETE

NYU CENTER FOR URBAN SCIENCE AND PROGRESS

APPLIED DATA SCIENCE

MS URBAN INFORMATICS 2018-19

vvt221@nyu.edu

ACKNOWLEDGEMENTS

I would like to take this opportunity to thank Professor TImothy Savage for giving me an opportunity to work on this project and also imparting the necessary knowledge and resources about the various Applied Data Science techniques without which the success of this project would not have been possible. I would also like to thank Teaching Assistant Alex Shannon for helping clear doubts and providing new perspective on the workings of the underlying algorithms.

DECLARATION

The following project submission is mostly my own work. It adheres to New York University's guidelines on plagiarism and Student Code of Conduct. All the materials used and referenced have been duly acknowledged in the bibliography and references.

Table of Contents:

PROJECT STATEMENT:

Analysis and Prediction of Ridership based on MTA weekly turnstile data for Grand Central Station.

Algorithms Used:

1. ARMA time series model.
2. Extracting external trends using Poisson Regression and using time series analysis on the residuals.
3. Agglomerative hierarchical clustering to identify similar turnstiles.

DATA:

Data was obtained from the MTA website which provides a week-to-week update about the number of turnstiles entries and exits for all of its stations.

Weather data was obtained through an external website which made api calls to weatherUnderground.com to retrieve historical weather data.

DATA PREPROCESSING:

Preprocessing Turnstile Data:

A data collection script was run to collect turnstile files from the time period: **December 30th 2016 to December 7 2018**

1. In its original form, the timestile data entries are cumulative values which are collected every 4 hours (0300,0700,1100 and so on). However, during the summer months when daylight saving time is in effect the hours are increased by 1 hour (0400,0800,1200 and so on). In order to maintain uniformity, the data is preprocessed so that the 3-7-11-15-19-2 notation is maintained.

2. Once all the datasets are merged along the row axis, we aggregate this data at the individual level. Each turnstile is uniquely identified by:

   Turnstile ID = C/A + UNIT + SCP + Station + Linename
   Where **C/A** = Control Area (A002)
   UNIT = Remote Unit for a station (R051)
   SCP = Subunit Channel Position represents an specific address for a device

STATION = Represents the station name the device is located at
LINE NAME = Represents all train lines that can be boarded at this station

3. After this, the data is sorted by the machine identifier and then by date.
4. Following this, the absolute entries of each turnstile is obtained by subtracting the data of the later timestamp from that of the earlier timestamp.
5. Negative values obtained after this are filtered out . Also some outrageously high values (may be due to a glitch) have also not been taken into account. Assuming an entry rate of 1 entry/second, the maximum entries in a 4 hour period is 14400
6. Also , the recorded entries used for the analysis are of type “REGULAR” and not “RECOVR AUD”
7. The hour of the data collected was extracted from the date
8. The entries were then aggregated at the station level and then at the date level

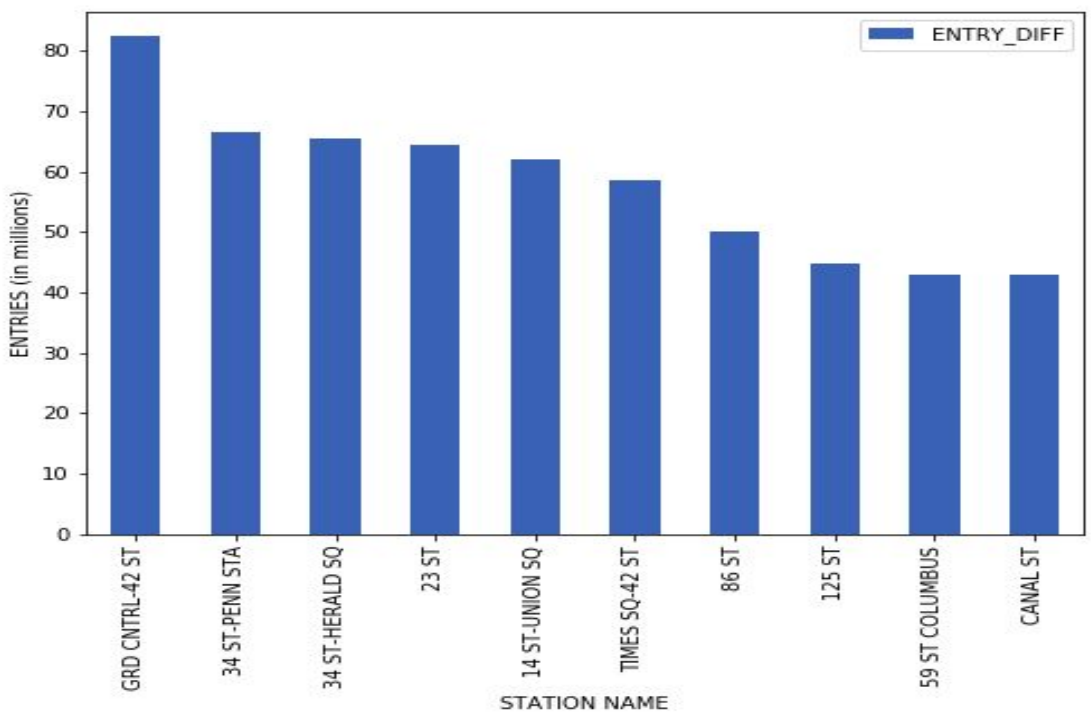


FIGURE : Bar plot showing the stations with the highest Entries received during the range of the dataset.

Since Grand-Central Station received the highest entries, I will be basing my analysis on the entries received at this station in it’s turnstiles.

Preprocessing Weather Data:

1. Download historical weather data from weather underground from dates 31 December 2016 to 7 December 2018 at Central Park weather station.
2. Replace the null values in precipitation with 0.0
3. Since the data is available on a hourly basis, it is binned for every 4 hours such that for continuous values such as temperature, precipitation , wind speed etc , the mean value at every four hours is used.

Feature Selection of Weather Data :

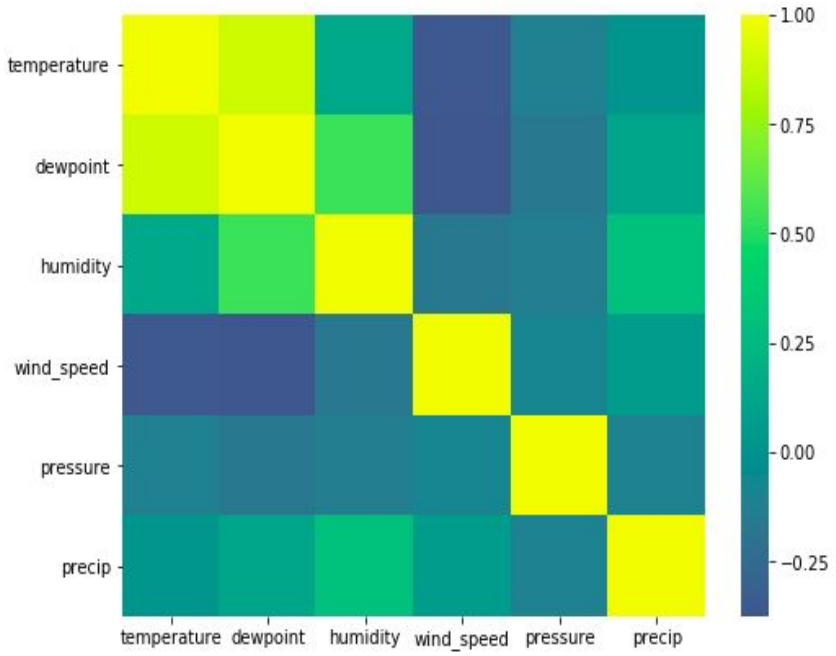


FIGURE : Heatmap showing the covariance between the various weather features

To avoid collinearity, I standardised the weather parameters and plotted the heatmap to check if the values are highly correlated. From the heatmap above, it can be seen that:

1. Temperature and dewpoint are highly correlated.
2. Temperature and wind_speed are highly correlated inversely.

Based on the heatmap, I chose temperature and precipitation in my analysis of impact on ridership due to external factors.

Time Series Forecasting using ARMA model

I employed the time series forecasting algorithm to predict the number of entries at Grand Central Station.

Checking the stationarity of the time series:
A TS is said to be stationary if its **statistical properties** such as mean, variance remain **constant over time**. But why is it important? Most of the TS models work on the assumption that the TS is stationary. Intuitively, we can say that if a TS has a particular behaviour over time, there is a very high probability that it will follow the same in the future.

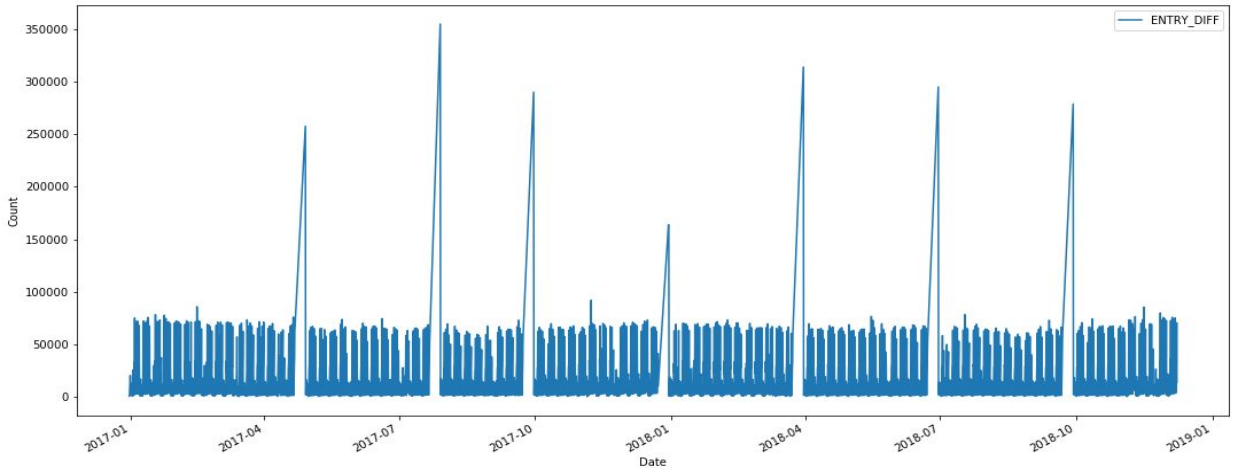


FIGURE : The plot shows the entries at the Grand Central Station over the course of 2 years.

The above plot clearly exhibits seasonality and naturally because subway stations receive high volume of commuters only during certain times of the day such as early mornings and evenings. This clearly indicates that the time series is not stationary.

AD - Fuller test to check for stationarity:
Null Hypothesis : The time series is not stationary
Alternate Hypothesis: The time series is stationary.
Significance Level alpha = 0.05

```
from statsmodels.tsa.stattools import adfuller
result = adfuller(pd.Series(gcounts.ENTRY_DIFF.values))
print('ADF Statistic: %f' % result[0])
print('p-value: %f' % result[1])
print('Critical Values:')
for key, value in result[4].items():
    print('\t%s: %.3f' % (key, value))
```

```
ADF Statistic: -20.352706
p-value: 0.000000
Critical Values:
        5%: -2.862
        10%: -2.567
        1%: -3.432
```

The low ADF statistic thus makes us reject the null hypothesis that the time series is non-stationary. However when we observe the following graph which decomposes the time series into the following components such as :

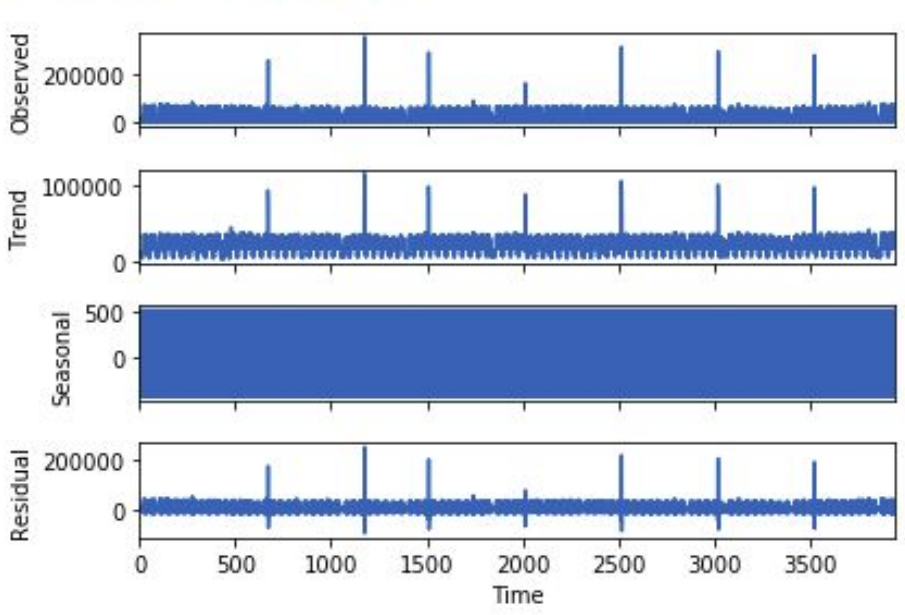


FIGURE : Plot showing the decomposition of a time series into its components namely trend, Residual and seasonal.

While there may be no identifiable trend, which makes the time series stationary, there is definitely a component of seasonality or periodicity which needs to be addressed.

To do this, the time series is differenced by its periodicity. Turnstile data is collected every 4 hours over the course of the day

Therefore, the periodicity of the time series is 6 hours which is exactly the number of intervals in which the turnstile entries are collected. So we take the difference the time series and take its logarithm .

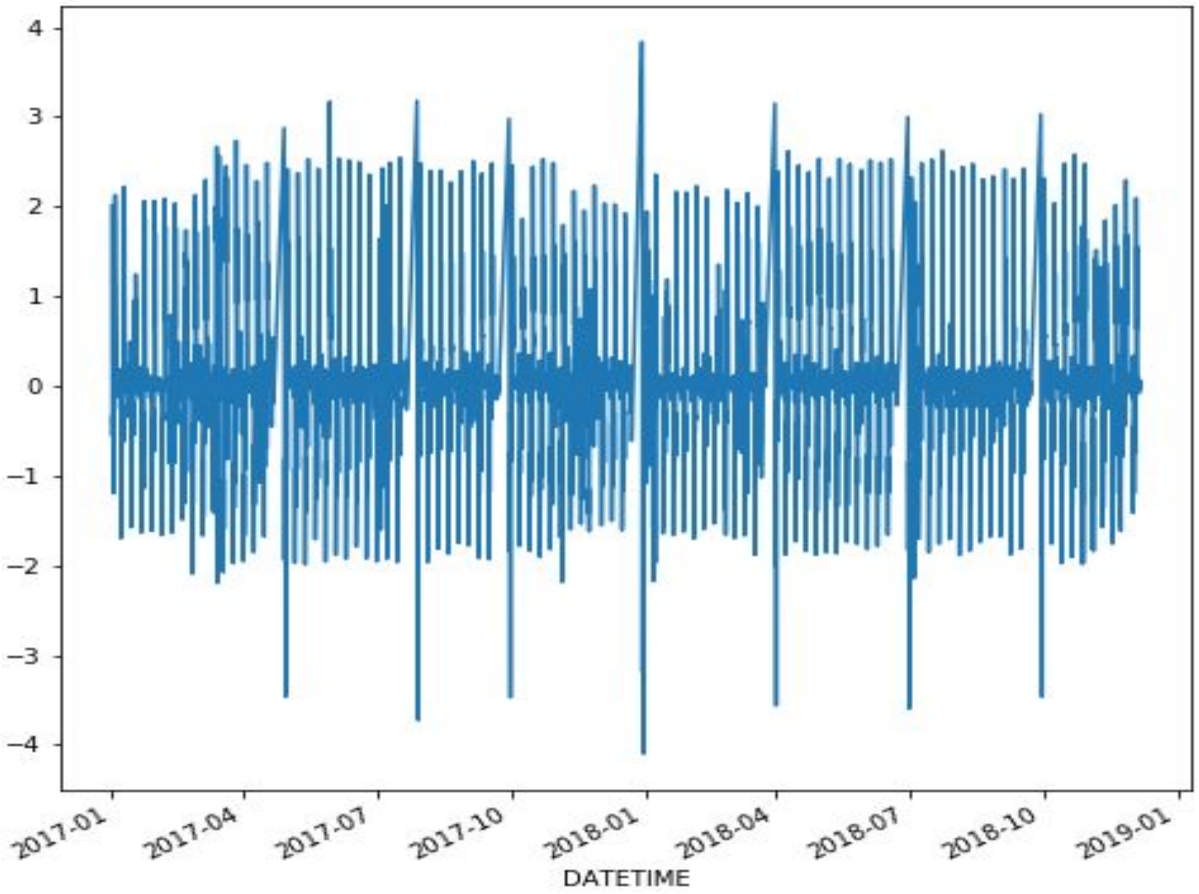


FIGURE : Differenced time series

Selection of AR and MA models for time series prediction:

The next first step is to select the lag values for the Autoregression (AR) and Moving Average (MA) parameters, p and q respectively.

This is done by reviewing Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots.

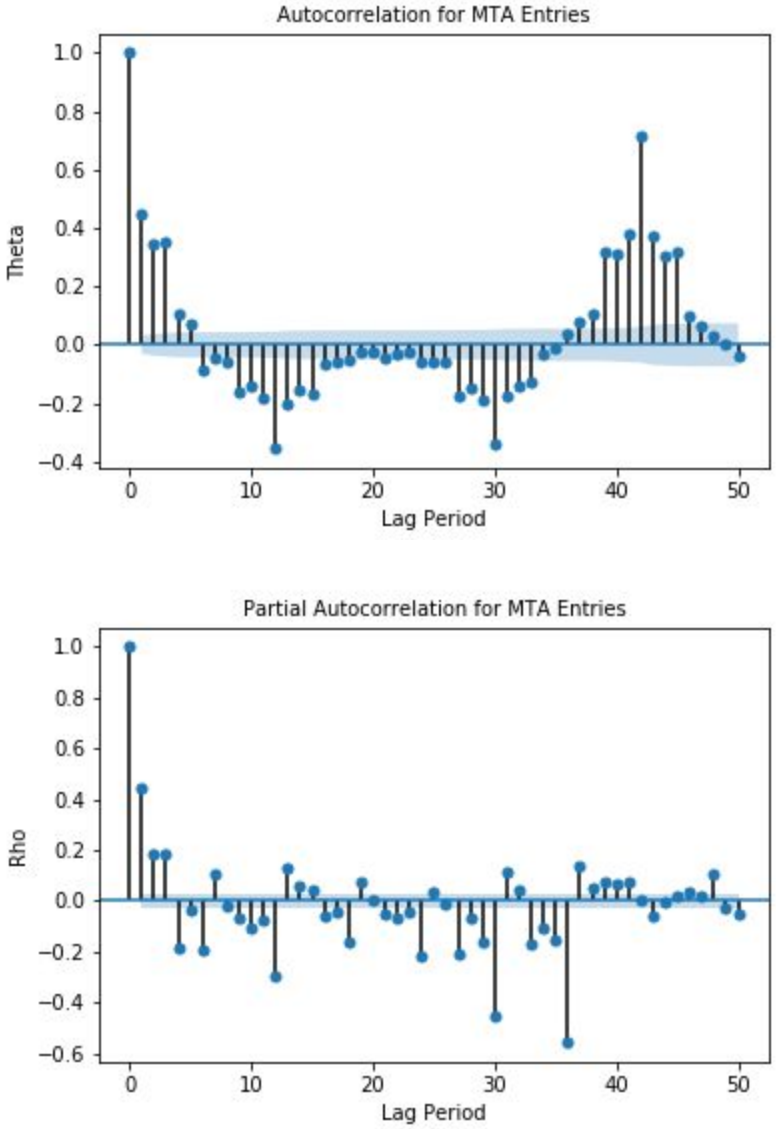


FIGURE : Autocorrelation and Partial Autocorrelation Plots

Following are some of the observations from the plots:

1. The partial autocorrelation shows significant lag time period t = 7(AR)
2. The autocorrelation shows significant lag at time period t=5 (MA)

p is the auto-regressive part of the model. It allows us to incorporate the effect of past values into our model.

q is the moving average part of the model. This allows us to set the error of our model as a linear combination of the error values observed at previous time points in the past.

Using 7 and 5 as the input to the ARMA model , the following results are observed:

ARMA Model Results

Dep. Variable:	ENTRY_DIFF	No. Observations:	3933
Model:	ARMA(7, 5)	Log Likelihood	-3248.532
Method:	css-mle	S.D. of innovations	0.552
Date:	Sun, 16 Dec 2018	AIC	6525.065
Time:	22:55:41	BIC	6612.945
Sample:	0	HQIC	6556.243

	coef	std err	z	P>|z|	[0.025	0.975]
const	7.399e-05	0.000	0.643	0.520	-0.000	0.000
ar.L1.ENTRY_DIFF	-0.1587	0.117	-1.362	0.173	-0.387	0.070
ar.L2.ENTRY_DIFF	0.0141	0.043	0.330	0.741	-0.070	0.098
ar.L3.ENTRY_DIFF	1.2012	0.035	34.158	0.000	1.132	1.270
ar.L4.ENTRY_DIFF	0.1681	0.131	1.278	0.201	-0.090	0.426
ar.L5.ENTRY_DIFF	-0.0178	0.033	-0.540	0.589	-0.082	0.047
ar.L6.ENTRY_DIFF	-0.5423	0.028	-19.166	0.000	-0.598	-0.487
ar.L7.ENTRY_DIFF	-0.0645	0.057	-1.126	0.260	-0.177	0.048
ma.L1.ENTRY_DIFF	0.5027	0.116	4.323	0.000	0.275	0.731
ma.L2.ENTRY_DIFF	0.2796	0.082	3.401	0.001	0.118	0.441
ma.L3.ENTRY_DIFF	-0.9808	0.003	-328.166	0.000	-0.987	-0.975
ma.L4.ENTRY_DIFF	-0.5154	0.113	-4.556	0.000	-0.737	-0.294
ma.L5.ENTRY_DIFF	-0.2840	0.084	-3.390	0.001	-0.448	-0.120

Roots

	Real	Imaginary	Modulus	Frequency
AR.1	1.0861	-0.2303j	1.1103	-0.0333
AR.2	1.0861	+0.2303j	1.1103	0.0333
AR.3	-0.3589	-1.0457j	1.1056	-0.3026
AR.4	-0.3589	+1.0457j	1.1056	0.3026
AR.5	-0.7464	-0.8199j	1.1088	-0.3675
AR.6	-0.7464	+0.8199j	1.1088	0.3675
AR.7	-8.3692	-0.0000j	8.3692	-0.5000
MA.1	1.0004	-0.0000j	1.0004	-0.0000
MA.2	-0.4989	-0.8816j	1.0130	-0.3320
MA.3	-0.4989	+0.8816j	1.0130	0.3320
MA.4	-0.9088	-1.6139j	1.8522	-0.3316
MA.5	-0.9088	+1.6139j	1.8522	0.3316

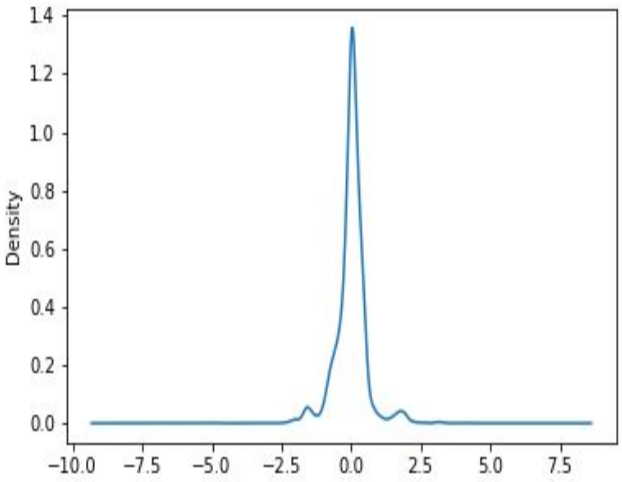
FIGURE: Summary of the ARMA(7,5) Time Series

Interpretation:

1. All the MA parameters are significant (low p-values) . They are very well within the estimated confidence intervals.
2. Out of the 7 AR components, only 2 AR parameters are significant.
3. For the rest of the components, we cannot reject the null of zero or no effect.

Residuals:

Ideally, the residuals should be randomly distributed. That is, the model should be able to successfully extract all the components of the time series which are necessary to forecast the future values. Also the residuals should have a gaussian distribution.



	0
count	3933.000000
mean	0.006689
std	0.553115
min	-4.834413
25%	-0.182211
50%	0.036296
75%	0.227369
max	4.136353

FIGURE: KDE Plot and Summary Statistics of ARMA(7,5) model.

Looking at the autocorrelation and the partial autocorrelation plots of the residuals, there is still some autocorrelation in the 6th and the 9th interval which the model has not been able to extract.

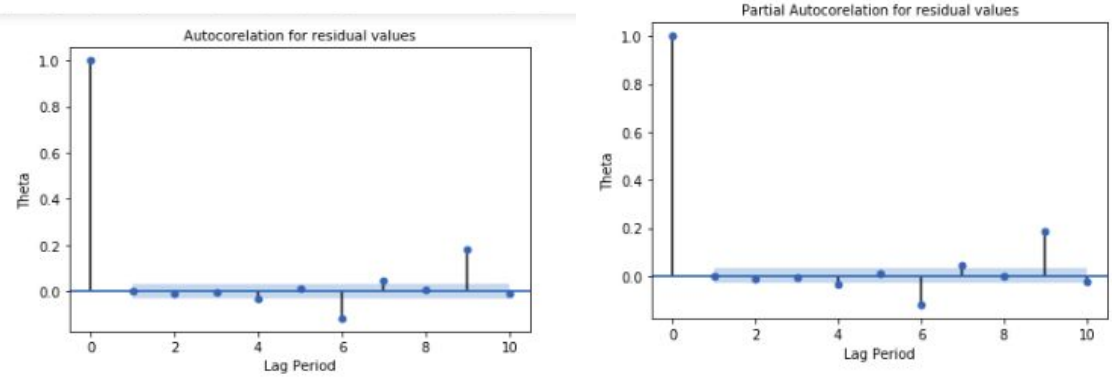


FIGURE : ACF and PACF Plots for ARMA(7,5) model

Time Series Forecast:

The ARMA(7,5) has an RMSE error of 0.3059 on the differenced time series.

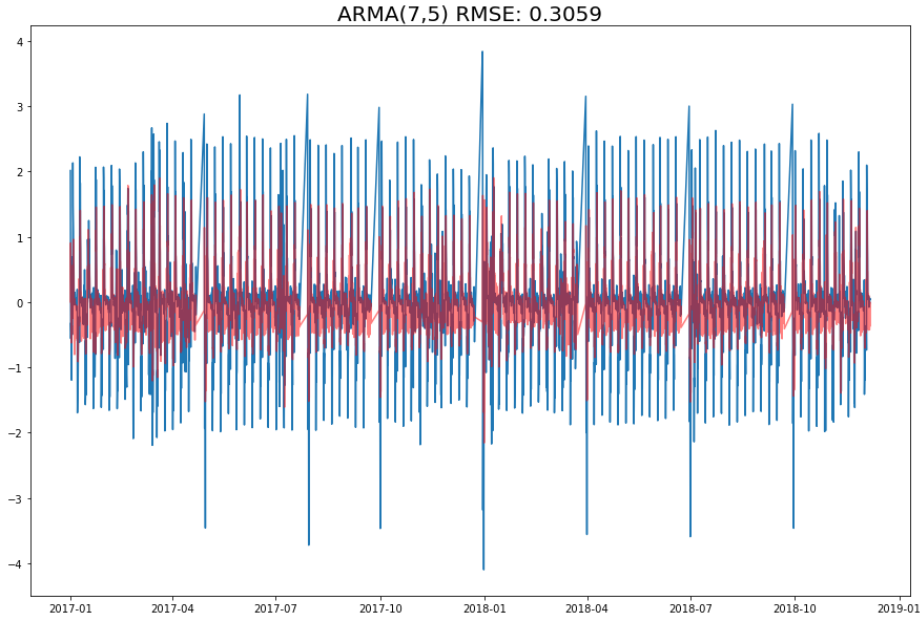


FIGURE: Plot showing observed vs predicted values of ARMA(7,5) and station data for differenced time series

The predicted values of the model are then inverted and differenced to obtain the actual values.

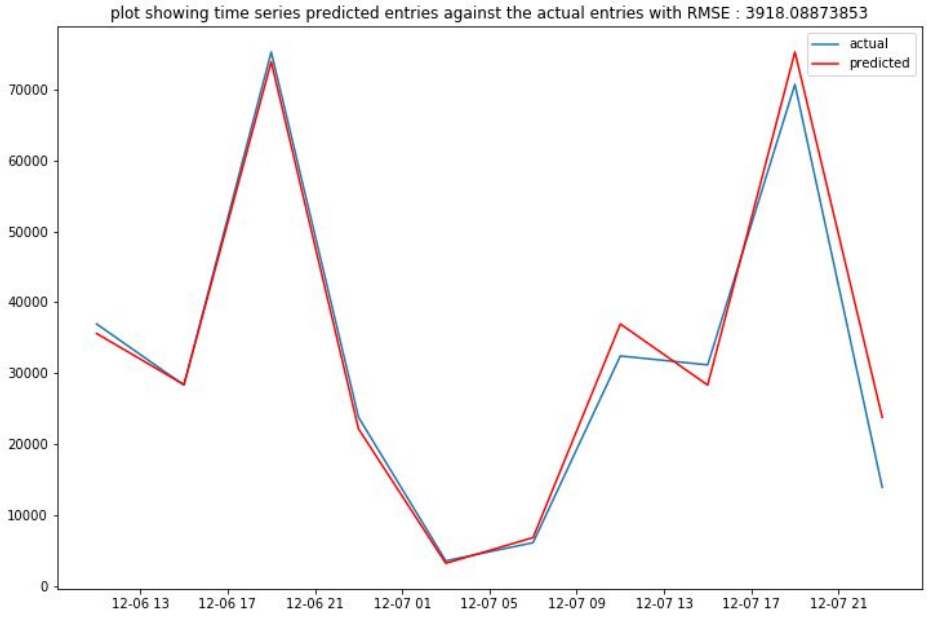


FIGURE: Plot showing observed vs predicted values of ARMA(7,5) and station data

When we test this model on the hold out set of the last 10 entries, a root mean square error of 3918 is obtained. This means that our models results are off by a margin of about 4000 entries over the course of 10 days.

Time Series Forecast After Removal of Impact of External Factors

The aim of this segment of analysis is to assess the impact of external factors such as weather, time of the day, weekday/ weekend. If there is indeed an effect, then we try to remove/ minimize the effect of these factors.

Preprocessing has already been explained in the previous section.
Independent Variables:

Independent Variable	**Type**
Temperature	Continuous Variable
Precipitation	Continuous Variable
Hour of the Day	Discrete/ Categorical [3,7,11,15,19,23]
Day of the Week	Discrete/Categorical[0-7] 0 -> Monday

TABLE : Parameter Names and Types

Algorithms:

1. Poisson regression model since entries are discrete values.
2. ARMA time series model on the residuals obtained from the above model

Initital Exploratory Analysis:

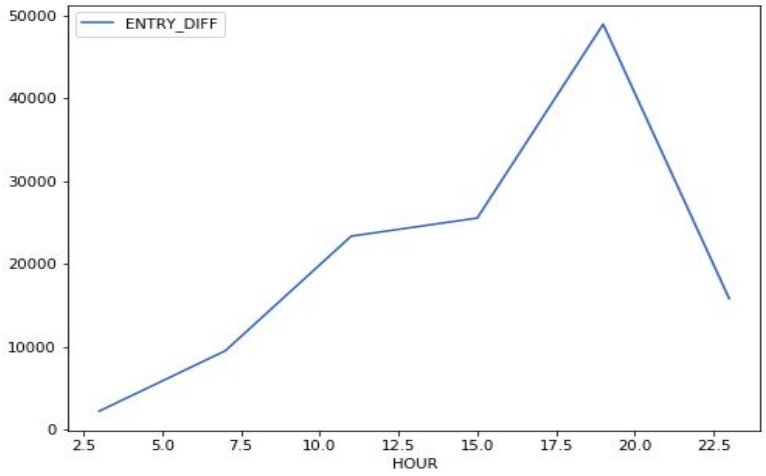


FIGURE: Plot showing the average ridership throughout the day

Ridership steadily climbs as morning gives way to afternoon. It reaches a peak period during the evening at 1900 hours and then goes down drastically at night. The hour of the day, thus, is a possible predictor of the number of entries at the station

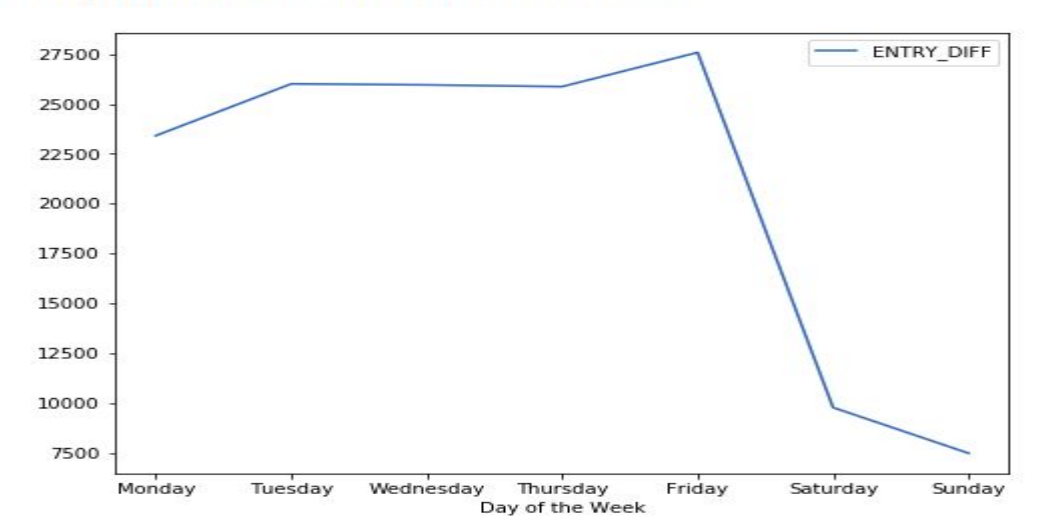


FIGURE : Mean Ridership for Each Day of the Week

The number of entries are high during the working days. Understandably so because people use subways to commute to their place of work . Ridership reaches its peak on Friday and drastically drops during the weekends. So the day of the week could be a highly probable predictor of the number of entries.

Effect of Temperature and Precipitation (Visual Analysis)

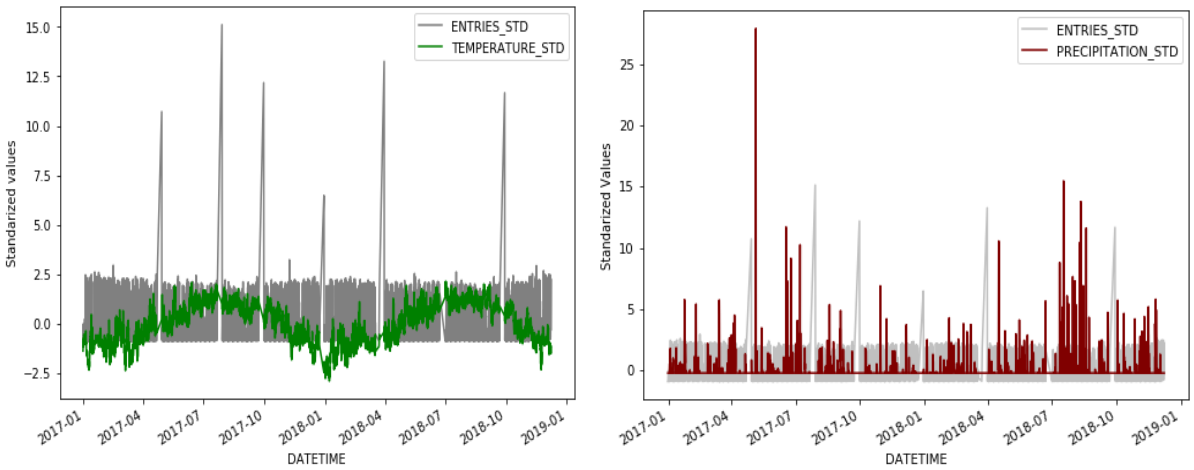


FIGURE : Temperature and Ridership (LEFT) Precipitation and Ridership(RIGHT) (Standardized)

ALGORITHM 1: POISSON REGRESSION MODEL:

Because the entries are discrete values, I will be making use of the Poisson Regression model. The parameters are then used to predict the ridership with the following results using the following formula:

formula='ENTRY_DIFF ~ C(HOUR) + C(week_no) + temperature + precip'

```
Optimization terminated successfully.
         Current function value: 1742.748359
         Iterations 7
                          Poisson Regression Results
==============================================================================
Dep. Variable:              ENTRY_DIFF   No. Observations:                 3909
Model:                         Poisson   Df Residuals:                     3895
Method:                            MLE   Df Model:                           13
Date:                 Tue, 18 Dec 2018   Pseudo R-squ.:                 0.8047
Time:                         11:44:05   Log-Likelihood:           -6.8124e+06
converged:                        True   LL-Null:                  -3.4877e+07
                                         LLR p-value:                    0.000
==============================================================================
                     coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept          7.8144      0.001   8887.818      0.000       7.813       7.816
C(HOUR)[T.7]       1.4508      0.001   1570.477      0.000       1.449       1.453
C(HOUR)[T.11]      2.3518      0.001   2698.926      0.000       2.350       2.354
C(HOUR)[T.15]      2.4405      0.001   2811.134      0.000       2.439       2.442
C(HOUR)[T.19]      3.0933      0.001   3637.024      0.000       3.092       3.095
C(HOUR)[T.23]      1.9383      0.001   2179.470      0.000       1.937       1.940
C(week_no)[T.1]    0.1057      0.000    276.867      0.000       0.105       0.106
C(week_no)[T.2]    0.1081      0.000    282.756      0.000       0.107       0.109
C(week_no)[T.3]    0.1012      0.000    264.905      0.000       0.100       0.102
C(week_no)[T.4]    0.1480      0.000    392.386      0.000       0.147       0.149
C(week_no)[T.5]   -0.8715      0.001  -1708.784      0.000      -0.873      -0.871
C(week_no)[T.6]   -1.1414      0.001  -2028.074      0.000      -1.142      -1.140
temperature        0.0006   1.14e-05     51.338      0.000       0.001       0.001
precip            -0.0032      0.000    -19.657      0.000      -0.004      -0.003
==============================================================================
```

FIGURE: Results of the Poisson Regression Model

Interpretation:

1. Categorical parameter coefficients are calculated relative to a particular value amongst themselves. So, at 0700 hours ridership increases by a factor of 1.45 relative to the ridership at 0300 hours. Similarly, ridership at 1900 hours increases by a factor of 3.09 of that at 0300 hours.
2. Likewise, ridership on Tuesday [T.1] (Monday =0) increases by a factor of 0.1057 relative to Monday and drastically decreases on Sunday by a factor of -1.1414.
3. Almost all the parameters show significance with a p-value of 0.000

4. The confidence intervals exclude zero, whereby we can reject the null of 0 or no effect.
5. Temperature causes a teeny - tiny increase in ridership by a factor of 0.0006. Confidence intervals of temperature are the same
6. Precipitation causes a drop in ridership by a factor of -0.0032.

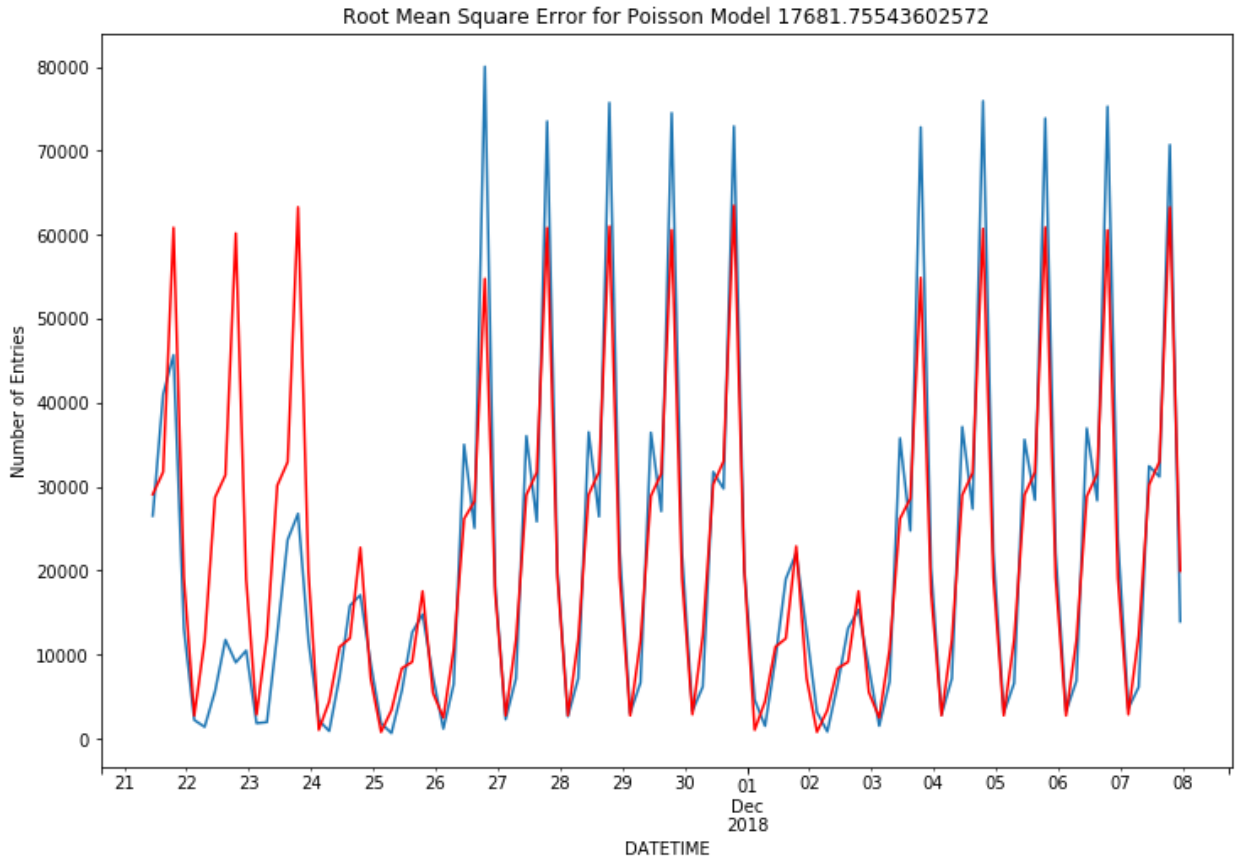


FIGURE : Plot showing the observed entries and the predicted entries(last 10 values) of the Poisson model

The figure below shows the plot of the residuals of the above Poisson model. These residuals are independent of the effect of the external factors such as day of the week, hour of the day , temperature and precipitation. However, there is still a periodicity which the Poisson model has not been able to extract.

This is where we turn to time series forecasting models to extract the remaining information of ridership from the periodic looking residuals.

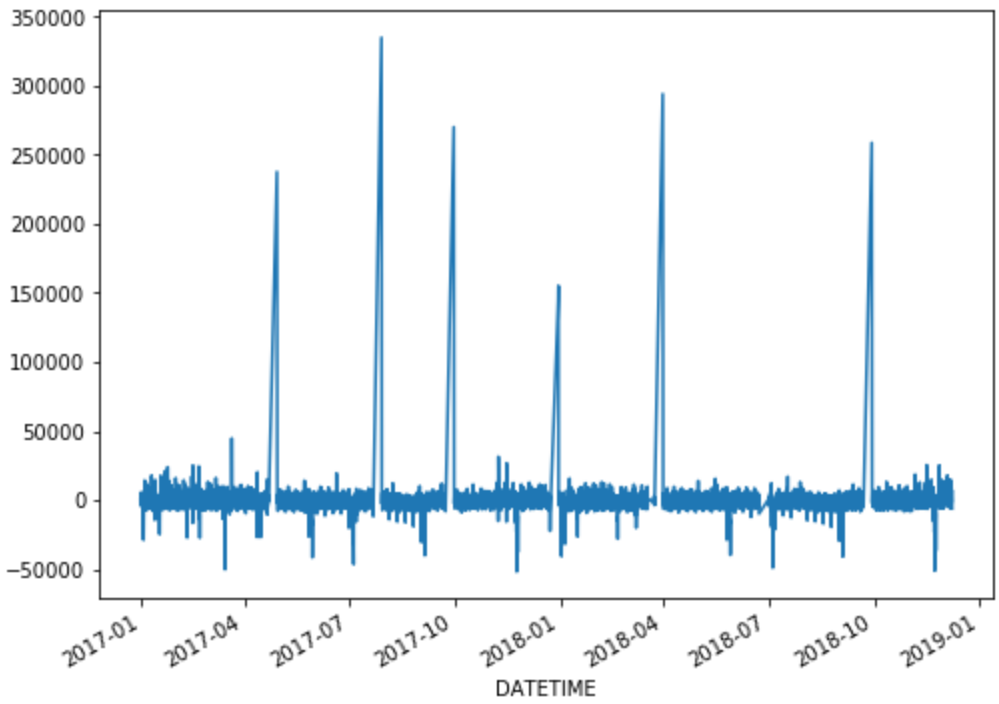


FIGURE: Residuals of the Poisson Regression Model

ALGORITHM 2: TIME SERIES FORECASTING

We first address the seasonality of the residuals.

To do this, the time series is differenced by its periodicity. Turnstile data is collected every 4 hours over the course of the day

Therefore, the periodicity of the time series is 6 hours which is exactly the number of intervals in which the turnstile entries are collected. So we take the difference the time series and take its logarithm.

Selection of AR and MA models for time series prediction:

The next first step is to select the lag values for the Autoregression (AR) and Moving Average (MA) parameters, p and q respectively.

This is done by reviewing Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots.

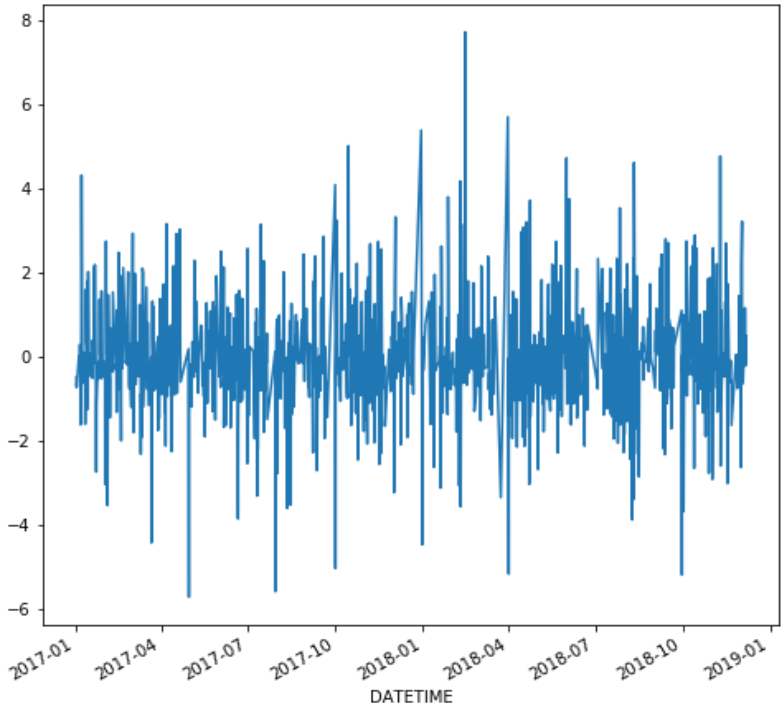


FIGURE : Plot showing the differenced time series of the residuals

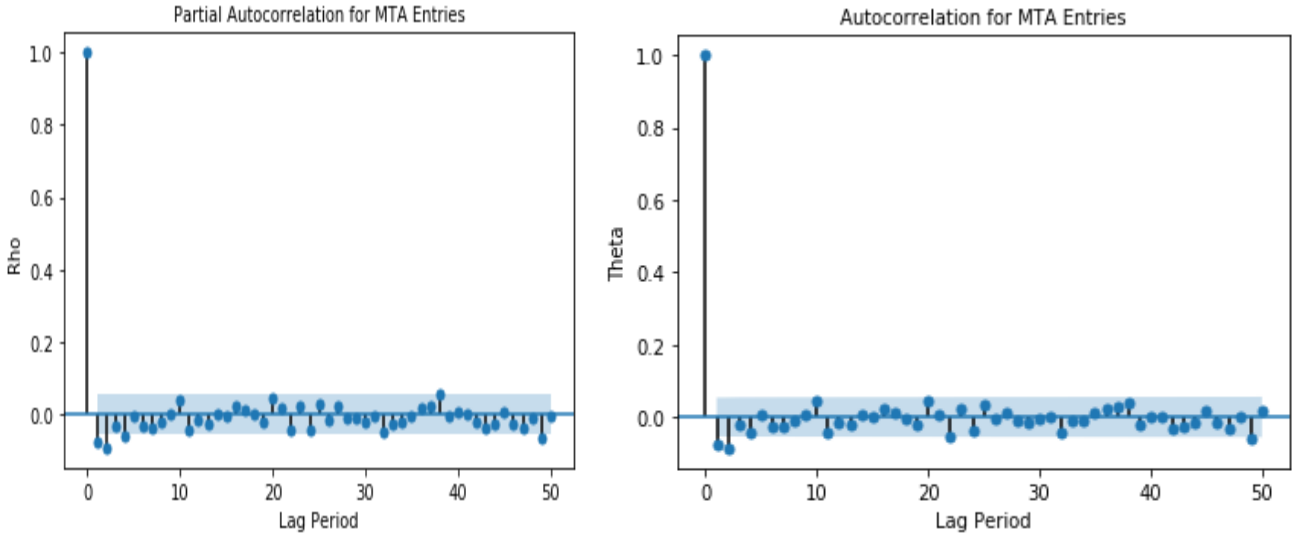


FIGURE : Plot showing the ACF and the PACF plots of the residuals of the Poisson Model

From the ACF and PACF, plots it is observed that there is no autocorrelation between the lagged values.

So starting from ARMA(0,0) I experimented with various AR, MA parameters whose RMSE values I have included in the table below:

ARMA(p,q)	Root Mean Square Error
(1,1)	1.4338
(1,2)	1.4333
(6,4)	1.4273
(7,5)	1.412
(8,8)	1.408
(9,8)	1.403

TABLE : RMSE values of various ARMA models for fitting on Poisson Residuals

I have selected ARMA(9,8) as the parameters of the model.

ARMA Model Results

Dep. Variable:		resid	No. Observations:			1340
Model:		ARMA(9, 8)	Log Likelihood			-2128.537
Method:		css-mle	S.D. of innovations			1.181
Date:		Tue, 18 Dec 2018	AIC			4293.073
Time:		12:49:45	BIC			4386.681
Sample:		0	HQIC			4328.142

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1.resid	-1.6201	0.109	-14.903	0.000	-1.833	-1.407
ar.L2.resid	-1.8478	0.264	-7.003	0.000	-2.365	-1.331
ar.L3.resid	-1.7825	0.400	-4.456	0.000	-2.566	-0.999
ar.L4.resid	-1.3985	0.484	-2.889	0.004	-2.347	-0.450
ar.L5.resid	-0.8202	0.494	-1.661	0.097	-1.788	0.148
ar.L6.resid	-0.1974	0.429	-0.460	0.645	-1.037	0.643
ar.L7.resid	0.3785	0.307	1.232	0.218	-0.223	0.980
ar.L8.resid	0.5440	0.155	3.508	0.000	0.240	0.848
ar.L9.resid	-0.0441	0.034	-1.287	0.198	-0.111	0.023
ma.L1.resid	1.5290	0.106	14.416	0.000	1.321	1.737
ma.L2.resid	1.6181	0.240	6.731	0.000	1.147	2.089
ma.L3.resid	1.4614	0.348	4.204	0.000	0.780	2.143
ma.L4.resid	1.0064	0.411	2.448	0.015	0.201	1.812
ma.L5.resid	0.4169	0.410	1.016	0.310	-0.387	1.221
ma.L6.resid	-0.1785	0.347	-0.514	0.607	-0.859	0.502
ma.L7.resid	-0.6851	0.238	-2.874	0.004	-1.152	-0.218
ma.L8.resid	-0.7236	0.105	-6.879	0.000	-0.930	-0.517

Roots

	Real	Imaginary	Modulus	Frequency
AR.1	-1.0041	-0.0000j	1.0041	-0.5000
AR.2	-0.8110	-0.5937j	1.0051	-0.3994
AR.3	-0.8110	+0.5937j	1.0051	0.3994
AR.4	-0.2556	-0.9702j	1.0033	-0.2910
AR.5	-0.2556	+0.9702j	1.0033	0.2910
AR.6	0.4166	-0.9205j	1.0104	-0.1824
AR.7	0.4166	+0.9205j	1.0104	0.1824
AR.8	1.6784	-0.0000j	1.6784	-0.0000
AR.9	12.9500	-0.0000j	12.9500	-0.0000
MA.1	-1.0001	-0.0000j	1.0001	-0.5000
MA.2	-0.8161	-0.5942j	1.0095	-0.3998
MA.3	-0.8161	+0.5942j	1.0095	0.3998
MA.4	-0.2514	-0.9689j	1.0010	-0.2904
MA.5	-0.2514	+0.9689j	1.0010	0.2904
MA.6	0.4176	-0.9087j	1.0000	-0.1814
MA.7	0.4176	+0.9087j	1.0000	0.1814
MA.8	1.3531	-0.0000j	1.3531	-0.0000

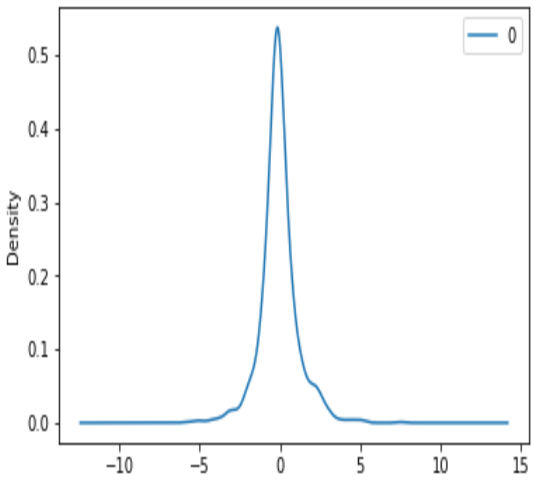
FIGURE: SUMMARY OF ARMA(9,8) MODEL FOR POISSON RESIDUAL MODELLING

Interpretation:

1. All AR parameters except 3 are significant with a low p value.
2. All MA parameters except 2 are significant with low p values
3. For the non significant components we cannot reject the null of zero or no effect.

Residuals:

Ideally, the residuals should be randomly distributed. That is, the model should be able to successfully extract all the components of the time series which are necessary to forecast the future values. Also the residuals should have a gaussian distribution.




```
pd.DataFrame(results_AR1.resid).describe()
```

	0
count	1340.000000
mean	-0.048927
std	1.183995
min	-5.778522
25%	-0.555051
50%	-0.124668
75%	0.407175
max	7.541832

FIGURE: Kernel Density Plot and Summary Statistics of the residuals of the residuals of the Poisson Model

Looking at the autocorrelation and the partial autocorrelation plots of the residuals, any autocorrelation that was present has been completely removed.

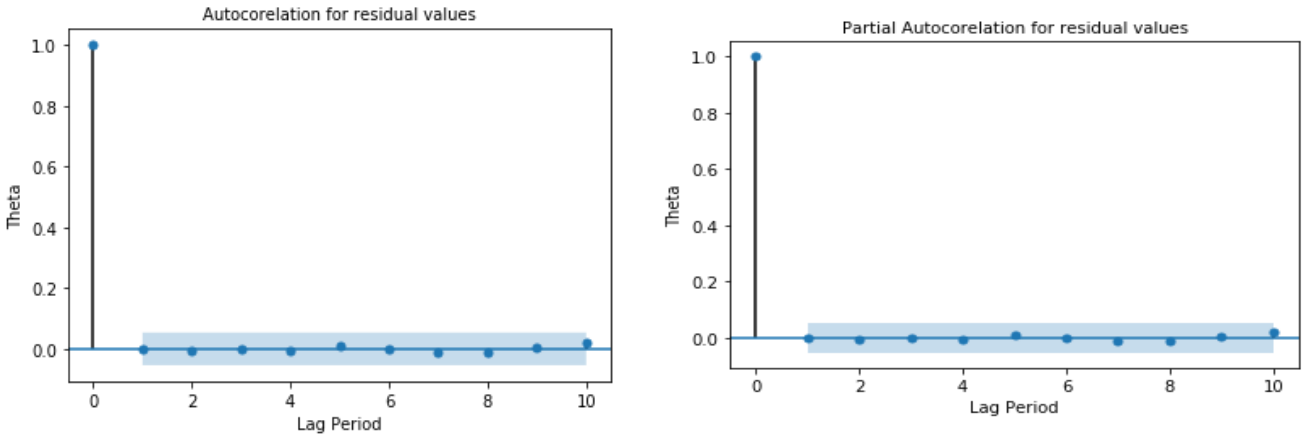


FIGURE: ACF AND PACF PLOTS OF THE RESIDUALS

Time Series Forecast:

The ARMA(9,8) has an RMSE error of 1.403 on the differenced time series.

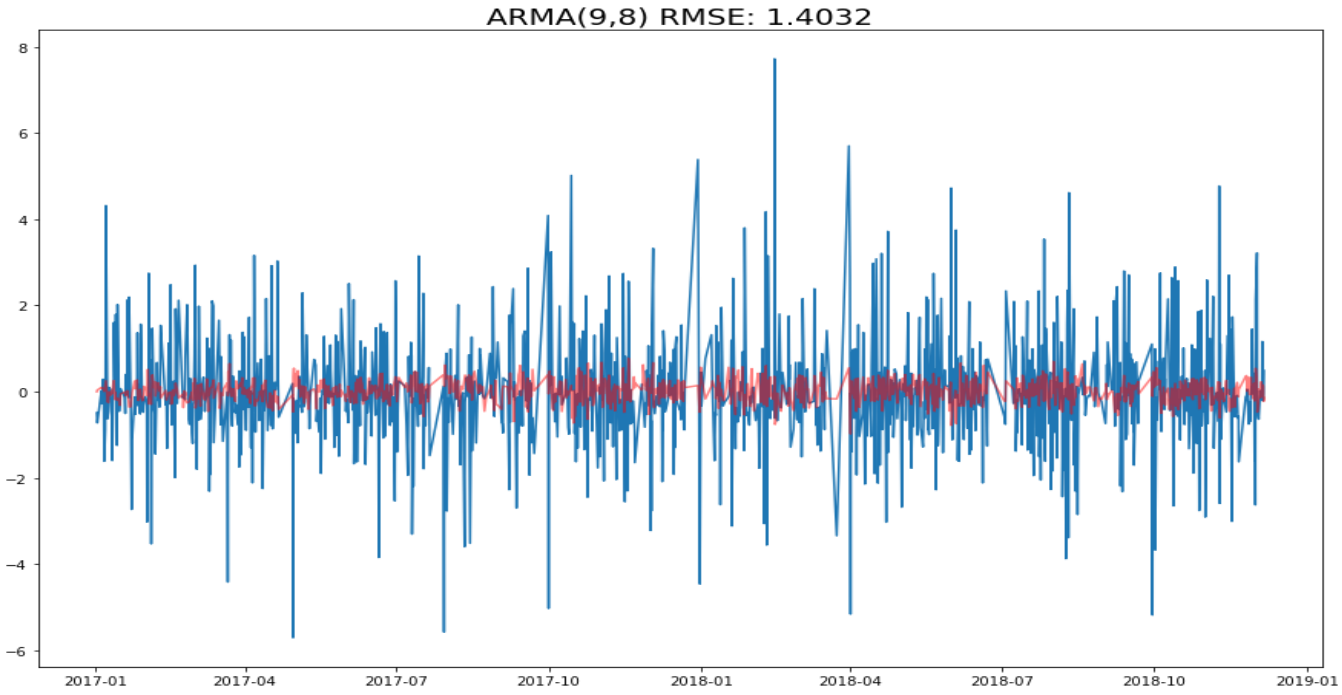


FIGURE: Plot showing observed vs predicted values of ARMA(9,8) and residuals data for differenced time series

The predicted values of the model are then inverted and differenced to obtain the actual values.

FIGURE: Plot showing observed vs predicted values of ARMA(9,8) and station data

When we test this model on the hold out set of the last 10 entries, a root mean square error of 4642.415 is obtained. This means that our models results are off by a margin of about 4650 entries over the course of 10 days.

Clustering Time Series:

The rationale behind performing clustering of time series per turnstile is to identify if some entries (or exits) are more frequently in comparison to others. This information could be used to identify pockets of crowded areas and potentially be used for advertising or entertainment purposes. It could also be used by authorities to make appropriate provisions for better crowd control and management in these areas in case of a disaster or an emergency.

Data Preprocessing:

1. The turnstile data aggregated earlier at the station level is broken down at individual turnstile level for each timestamp.
2. A transpose of the dataframe of the form (Timestamp x Entries) is resulting in (Entries x Timestamp) where Entries is the number of entries at each turnstile. The dataframe is then indexed at row level with the turnstile identifier name.
3. These entries are then standardised at row level to preserve the inherent characteristics of the time series.

DATETIME	2016-12-31 03:00:00	2016-12-31 07:00:00	2016-12-31 11:00:00	2016-12-31 15:00:00	2016-12-31 19:00:00	2016-12-31 23:00:00	2017-01-01 03:00:00	2017-01-01 07:00:00	2017-01-01 11:00:00	2017-01-01 15:00:00	...	2018-12-06 11:00:00	2018-12-06 15:00:00	2018-12-06 19:00:00	2018- 23:00
R236-R045-00-00-00-GRD CNTRL-42 ST-4567S	-0.495850	-0.483274	-0.335961	-0.215595	-0.174275	-0.373687	-0.443751	-0.474292	-0.456327	-0.377280	...	-0.237153	0.282037	1.611450	0.114
R236-R045-00-00-01-GRD CNTRL-42 ST-4567S	-0.488709	-0.458435	-0.320039	-0.194617	-0.179480	-0.384912	-0.432486	-0.480059	-0.449785	-0.402212	...	-0.062709	0.272470	1.861863	-0.097
R236-R045-00-00-02-GRD CNTRL-42 ST-4567S	-0.500474	-0.483920	-0.290003	-0.105546	-0.159937	-0.384597	-0.443718	-0.476826	-0.455542	-0.346760	...	0.256275	0.149857	1.720111	-0.200
R236-R045-00-00-03-GRD CNTRL-42 ST-4567S	-0.532473	-0.507321	-0.336289	-0.064651	-0.104893	-0.396654	-0.411745	-0.522412	-0.462048	-0.245743	...	1.228149	-0.125015	1.208028	-0.361
R236-R045-00-00-04-GRD CNTRL-42 ST-4567S	-0.527676	-0.510681	-0.319492	-0.162291	-0.141048	-0.289751	-0.387470	-0.497935	-0.417211	-0.285502	...	1.418209	-0.119805	1.328987	-0.357

FIGURE: DATAFRAME SNAPSHOT FOR TURNSTILE TIME SERIES CLUSTERING

I retrieved all the turnstile ids at Grand Central Station (68 turnstiles) and did the some slicing and extraction to identify the group of turnstiles.

With the assumption that turnstiles located together will have a similar time series pattern, I set the number of clusters to 14 as an input to hierarchical agglomerative clustering with the

distance measure as Euclidean distance. These are the some examples of the various turnstiles in each turnstile groups:

Set of Turnstiles under R237B-R047-01-00 R237B-R047-01-00-00 R237B-R047-01-00-01 **Total number of turnstiles : 4**	**Set of Turnstiles under R241A-R048-00-00** R241A-R048-00-00-00 R241A-R048-00-00-01 **Total number of turnstiles : 5**
Set of Turnstiles under R236-R045-00-00 R236-R045-00-00-00 R236-R045-00-00-01 **Total number of turnstiles : 6**	**Set of Turnstiles under R236-R045-00-03** R236-R045-00-03-00 R236-R045-00-03-01 **Total number of turnstiles : 4**
Set of Turnstiles under R238A-R046-02-00 R238A-R046-02-00-00 R238A-R046-02-00-01 **Total number of turnstiles : 5**	**Set of Turnstiles under R238-R046-00-00** R238-R046-00-00-00 R238-R046-00-00-01 **Total number of turnstiles : 10**
Set of Turnstiles under R236-R045-00-06 R236-R045-00-06-00 R236-R045-00-06-01 **Total number of turnstiles : 2**	**Set of Turnstiles under R238-R046-00-06** R238-R046-00-06-00 R238-R046-00-06-01 **Total number of turnstiles : 5**
Set of Turnstiles under R240-R047-00-00 R240-R047-00-00-00 R240-R047-00-00-01 **Total number of turnstiles : 2**	**Set of Turnstiles under R240-R047-00-03** R240-R047-00-03-00 R240-R047-00-03-01 **Total number of turnstiles : 9**
Set of Turnstiles under R238A-R046-02-03 R238A-R046-02-03-00 R238A-R046-02-03-01 **Total number of turnstiles : 3**	**Set of Turnstiles under R238-R046-00-03** R238-R046-00-03-00 R238-R046-00-03-01 **Total number of turnstiles : 5**
Set of Turnstiles under R238-R046-00-05 R238-R046-00-05-00 **Total number of turnstiles : 1**	**Set of Turnstiles under R237-R046-01-00** R237-R046-01-00-00 R237-R046-01-00-01 **Total number of turnstiles : 7**

TABLE : Turnstile Groups and some examples of their corresponding turnstiles

These are the number of turnstiles per cluster identified by the clustering algorithm as compared against the actual number of turnstiles per group.

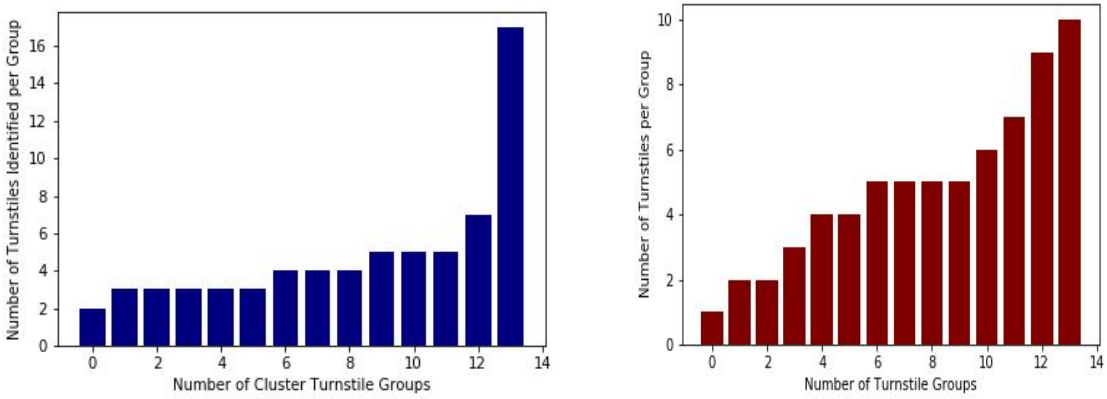


FIGURE : Bar Charts Showing the difference between the number of observed and actual cluster and their distribution in various cluster groups.

Clearly, the clusters dont mirror the actual turnstile groups.From the figure generated below, the distance between the turnstiles belonging to the same group are closer. They are often interspersed with a turnstile from a different group at different intervals, which is merged at a greater distance. The cluster number labels are pre appended to the turnstile group labels to distinguish between the different clusters formed.

While the clusters did not group the turnstiles exactly according to their location , it does generate clusters within the turnstiles which are located nearer to each other.

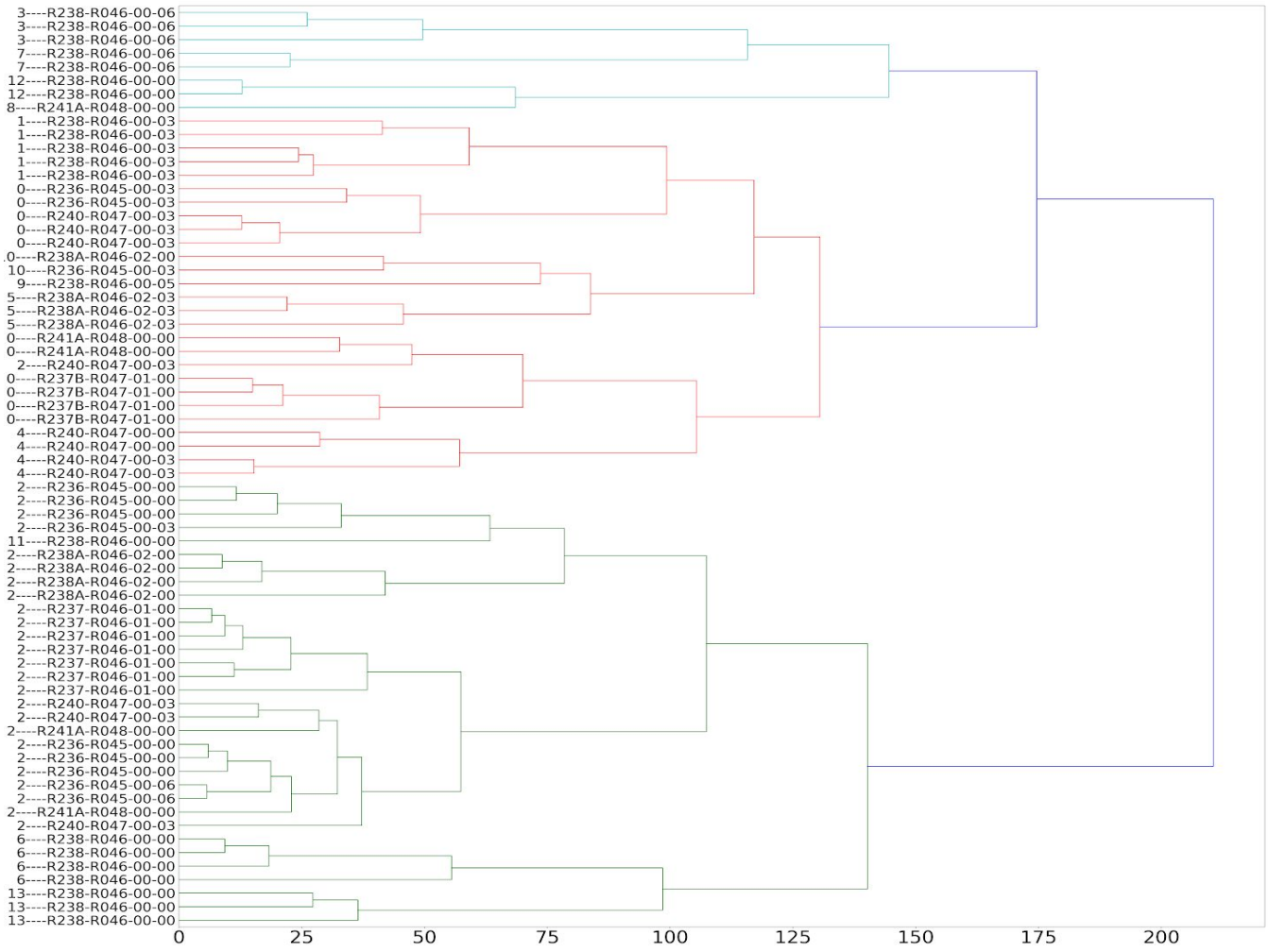
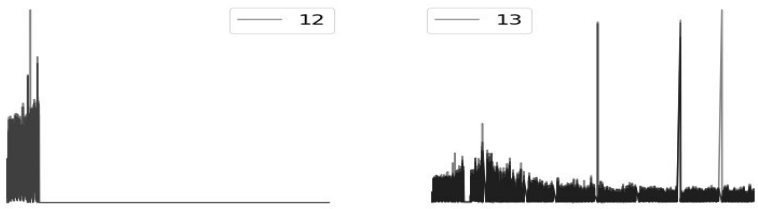


FIGURE : Plot displaying turnstile clusters using agglomerative clustering. The cluster labels are of the form [cluster-label-number]---<turnstile- group> where each leaf represents a unique turnstile id.

If the clusters are not formed by virtue of their location , then are they formed on the basis of the number of entries over the course of time? Here are the clusters generated:



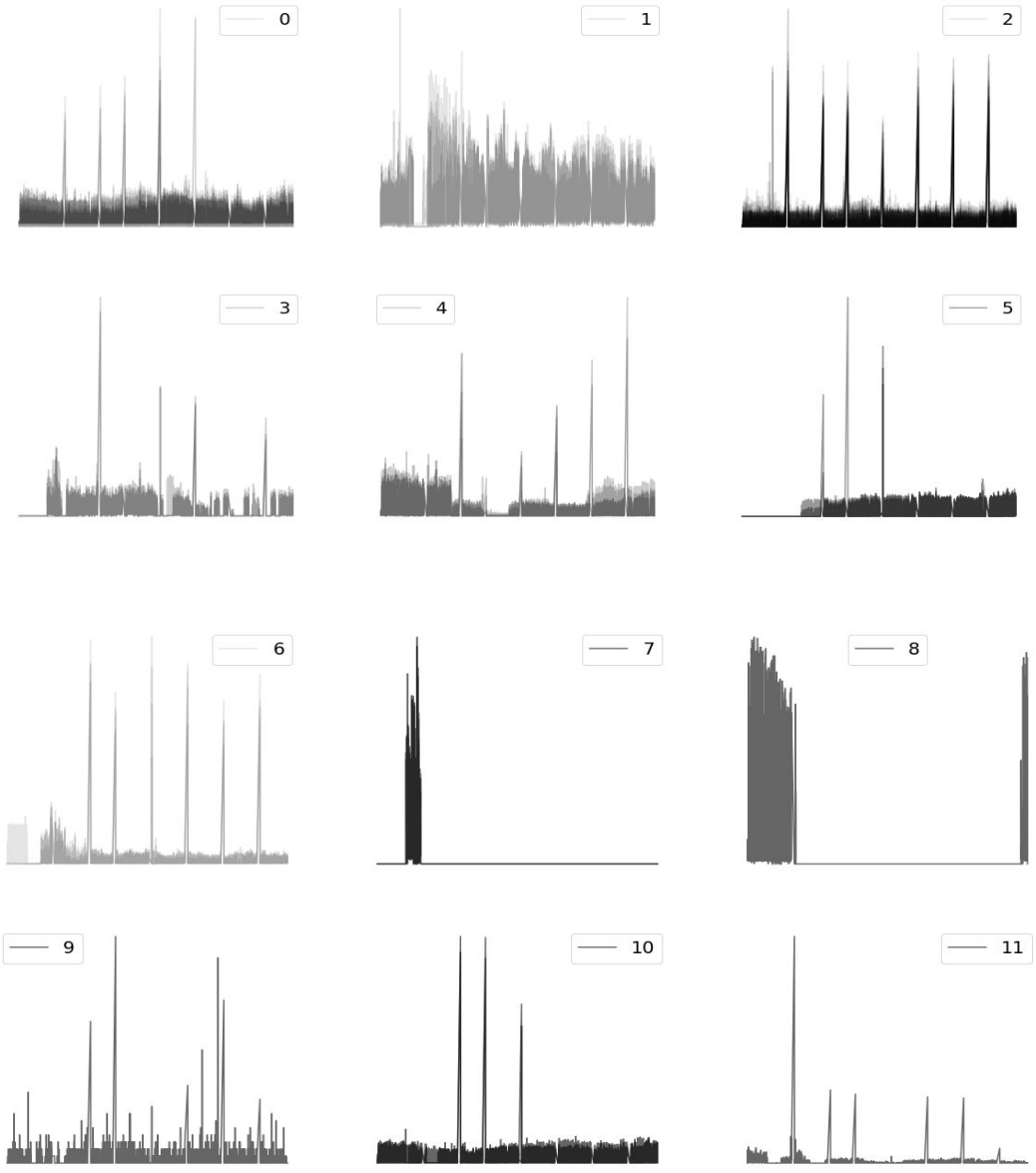


FIGURE 9: Clusters generated using Agglomerative Hierarchical Clustering.

The clusters generated do indicate that the turnstiles having similar time series tend to cluster together more than turnstiles that are merely located together. The clustering algorithm very effectively teases out the different time series of the various turnstiles as can be seen in cluster number 6 , 7 and 8.

Limitations of Analysis:

1. Lack of 4-hourly weather data so aggregation was required which may have led to less accurate values.
2. Turnstile data dirty and lot of preprocessing required.
3. Could not completely isolate correlation and autocorrelation from the time series residuals.
4. Better refinement of clustering techniques required.
5. RMSE errors are of the order of about ~ 4500 which requires further inspection.

Conclusion:

The analysis attempts to explore the ridership habits of the commuters of the Grand Central Station.

The first algorithm, ie , the ARMA time series model relies solely on its previous values to predict the new values.

The second algorithm tries to eliminate the effect of other external factors on the ridership habits of the commuters and then seeks to forecast ridership values on the denatured model. The time of the day and the day of the week are significant predictors of ridership.

The clustering algorithm tries to identify if turnstiles located at the same location of the station have similar entries over the course of time. While this holds true for a subset of turnstiles , it is not completely true for the entire set.

REFERENCES AND BIBLIOGRAPHY

1. http://web.mta.info/developers/turnstile.html
2. http://piratefsh.github.io/projects/2015/10/03/mta-subway-turnstile-data.html
3. https://bigdatasubwayanalysis.wordpress.com/
4. https://github.com/fedhere/PUI2018_fb55/tree/master/HW12_fb55
5. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html
6. https://machinelearningmastery.com/time-series-forecast-study-python-monthly-sales-french-champagne/
7. https://people.duke.edu/~rnau/411arim3.htm
8. https://jakevdp.github.io/blog/2014/06/10/is-seattle-really-seeing-an-uptick-in-cycling/