PREDICTING TRANSIT RIDERSHIP FOR A CITY

CAPSTONE PROJECT BY SIRI SURABATHULA
MENTOR - SRDJAN SANTIC

SPRINGBOARD DATA SCIENCE CAREER TRACK

PROBLEM

The success of transit systems depends on their ability to scale and their responsiveness to regional variations in demand. Short-term and long-term prediction models are vital to solving problems of urban mobility.

This project aims at predicting transit ridership at different stations across a large city over the short-term using weather, traffic, taxi/cab and gas price data. The geospatial aspect of transit, taxi/cab and traffic data is taken into account while building the model for prediction.

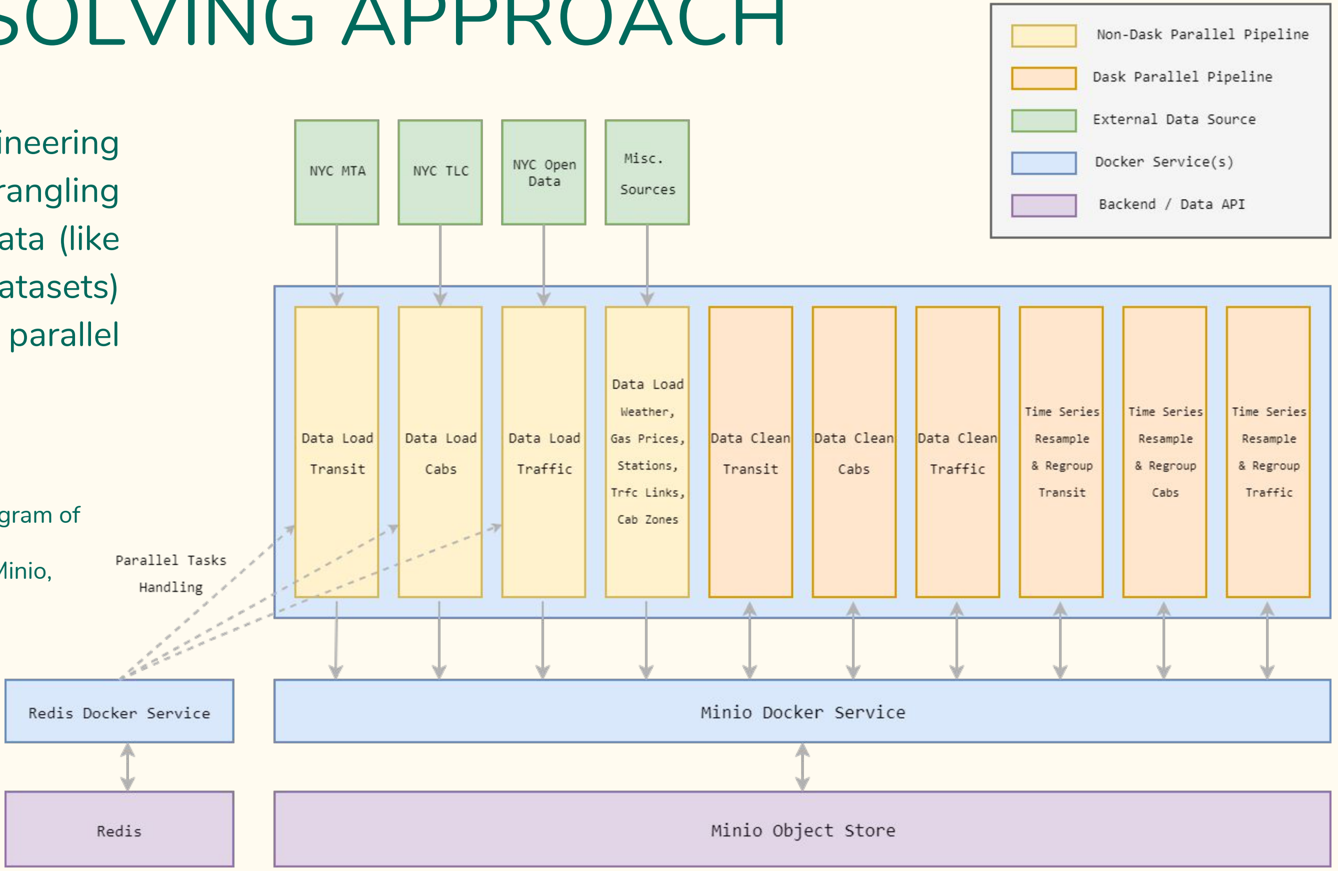
Prospective clients for the model are city transportation authorities (NYC DOT, NYC MTA) and city planning authorities (City of New York). Urban planners could use ridership predictions to efficiently allocate resources and modulate the frequency of trains and buses along specific routes.

PROBLEM SOLVING APPROACH

The project focuses on engineering data cleaning and wrangling components. Most of the data (like the NYC cabs and traffic datasets) were massive and required parallel processing techniques.

Right: The figure shows a high-level diagram of the data pipelines.
These pipelines were built using Dask, Minio, Redis and Docker*

**Dask* - parallel computing python library.
Minio - object-storage with S3-like-API.
Redis - lightweight in-memory data store.
Docker - container platform with environment and platform virtualization.



DATA

- **Transit** - ridership (turnstile entry and exit counts) of all subway stations in NYC. This includes station name, date, time, entry count and exit count.
- **Transit Stations** - description of all subway stations in NYC. This includes station name and geographic coordinates of the station.
- **Cabs** - details of all recorded taxi/cab trips in NYC. This includes drop-off time, drop-off location coordinates, pick-up time, pick-up location coordinates and number of passengers for each ride.
- **Cab Zones** - geospatial details* of all cab zones in NYC.
- **Traffic** - average motor-vehicle speed recorded at various locations on NYC streets and roadways.
- **Traffic Links** - sets of geographic coordinates representing a stretch of roadway or street over which the average speed of traffic was recorded.
- **Weather** - daily average temperature, rainfall and snowfall data for NYC.
- **Gas** - weekly retail gasoline and diesel prices for NYC.

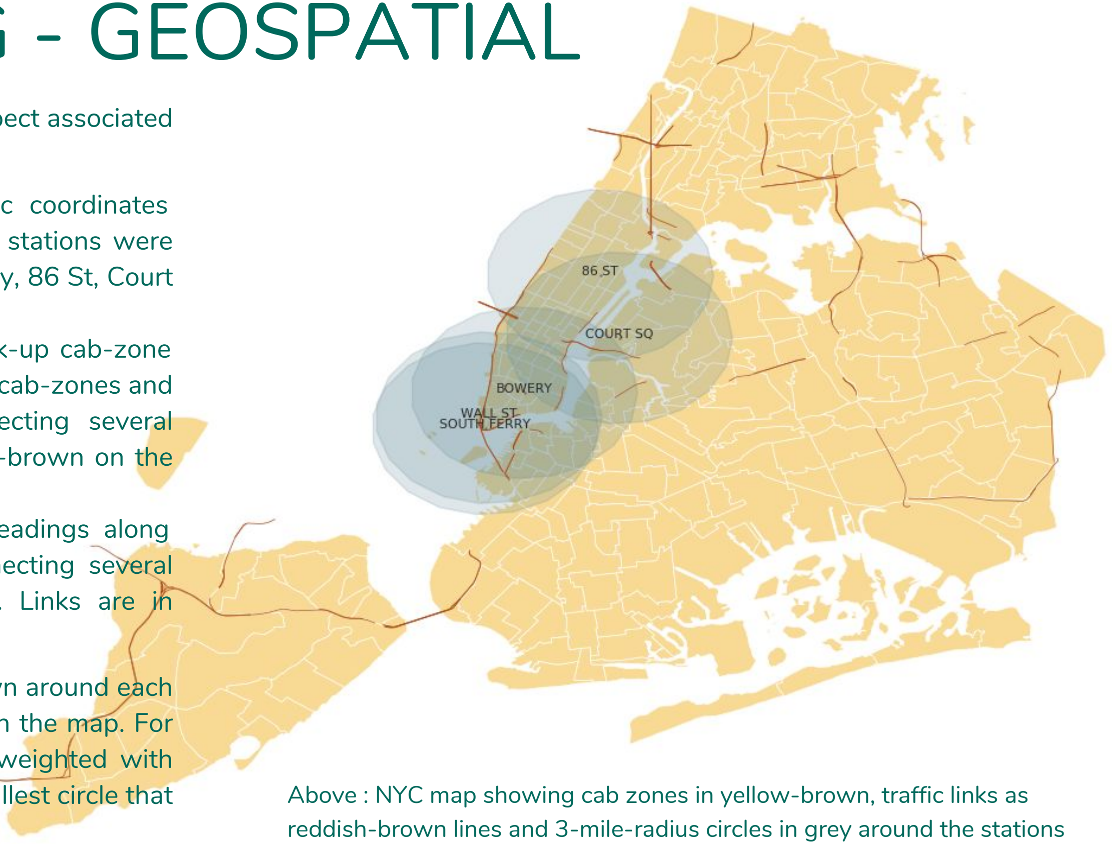
*these zones are geographic polygons which are stored in a geospatial format called GIS Shapefile

PREPROCESSING - GEOSPATIAL

The transit, cabs and traffic data have a geospatial aspect associated with them.

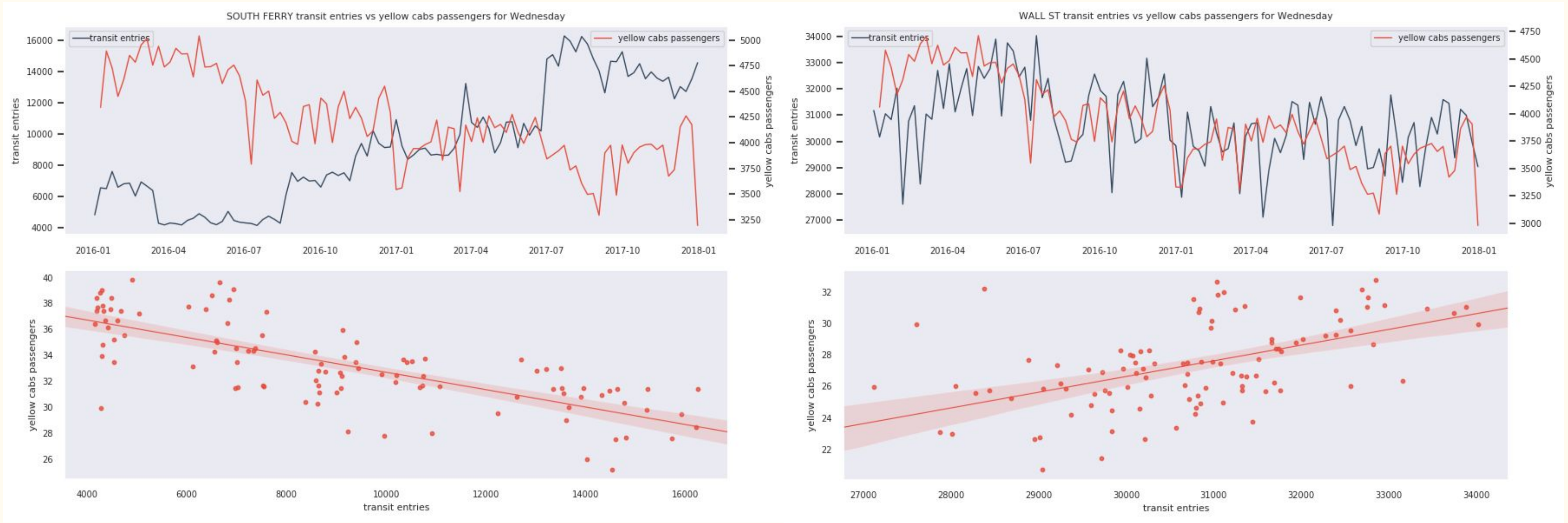
- **Transit** data - Each station has geographic coordinates (latitude and longitude) associated with it. 5 stations were chosen for study (Wall St, South Ferry, Bowery, 86 St, Court Sq. These are located on the map)
- **Cab** data - Each ride has a drop-off and pick-up cab-zone associated with it (NYC is divided into several cab-zones and each zone is a polygon formed by connecting several geographic coordinates). Zones are in yellow-brown on the map.
- **Traffic** data is comprised of traffic speed readings along specific 'links' (links are paths or lines connecting several geographic coordinates along NYC streets). Links are in reddish-brown on the map.

Circles of radius half to nine-and-half miles were drawn around each station. Three-mile radius circles are shown in grey on the map. For each station, all cab rides and traffic speeds were weighted with weights inversely proportional to the radius of the smallest circle that the cab(traffic) reading's zone(link) intersects with.



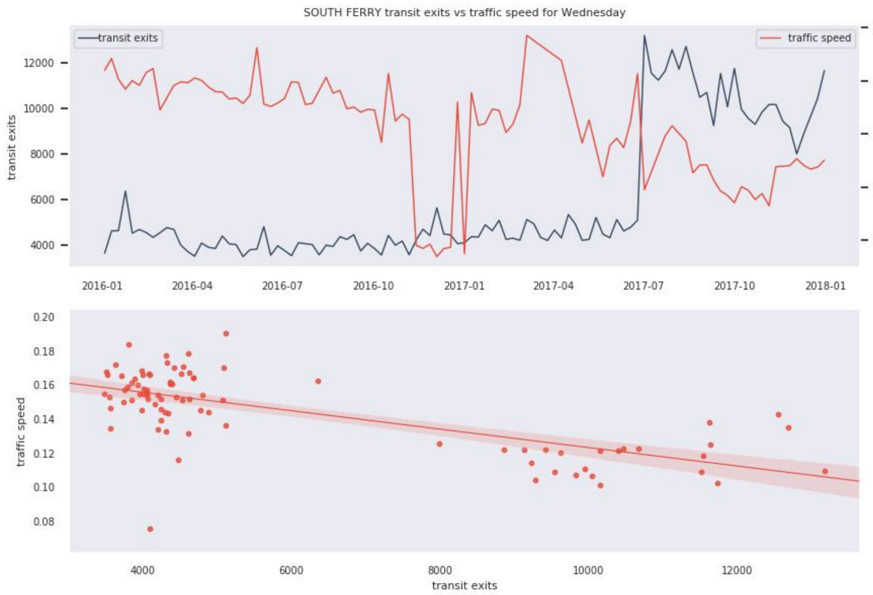
Above : NYC map showing cab zones in yellow-brown, traffic links as reddish-brown lines and 3-mile-radius circles in grey around the stations Wall St, South Ferry, Bowery, 86 St and Court Square.

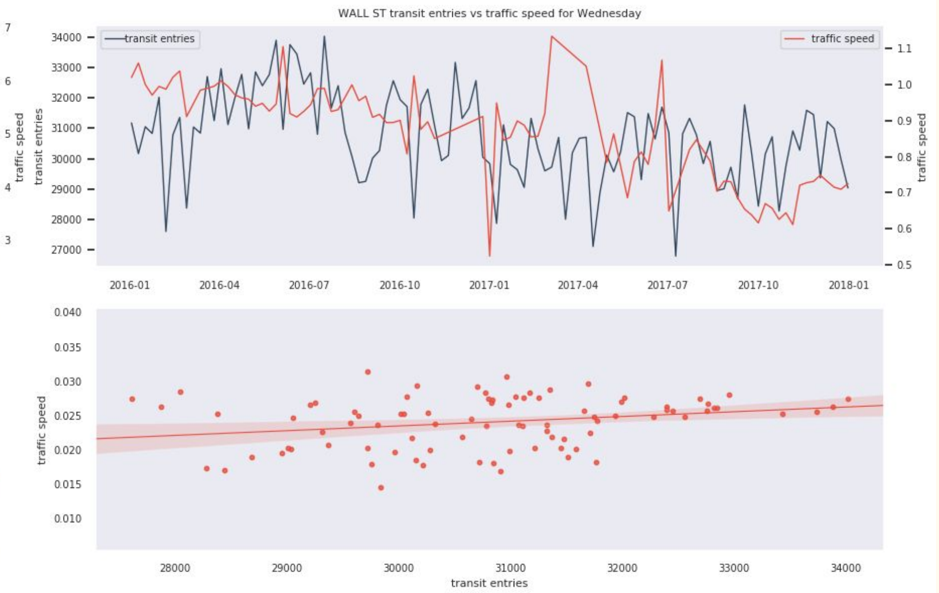
ANALYSIS - TRANSIT vs. CABS



For most stations (refer to plots on left for South Ferry), there is a negative correlation between the number of people entering (or exiting) the station and the number of cab rides in the vicinity of the station. For Wall Street (refer to plots on right for Wall St) and Court Square stations there is a positive correlation between station entries and cab rides in the vicinity of the station.

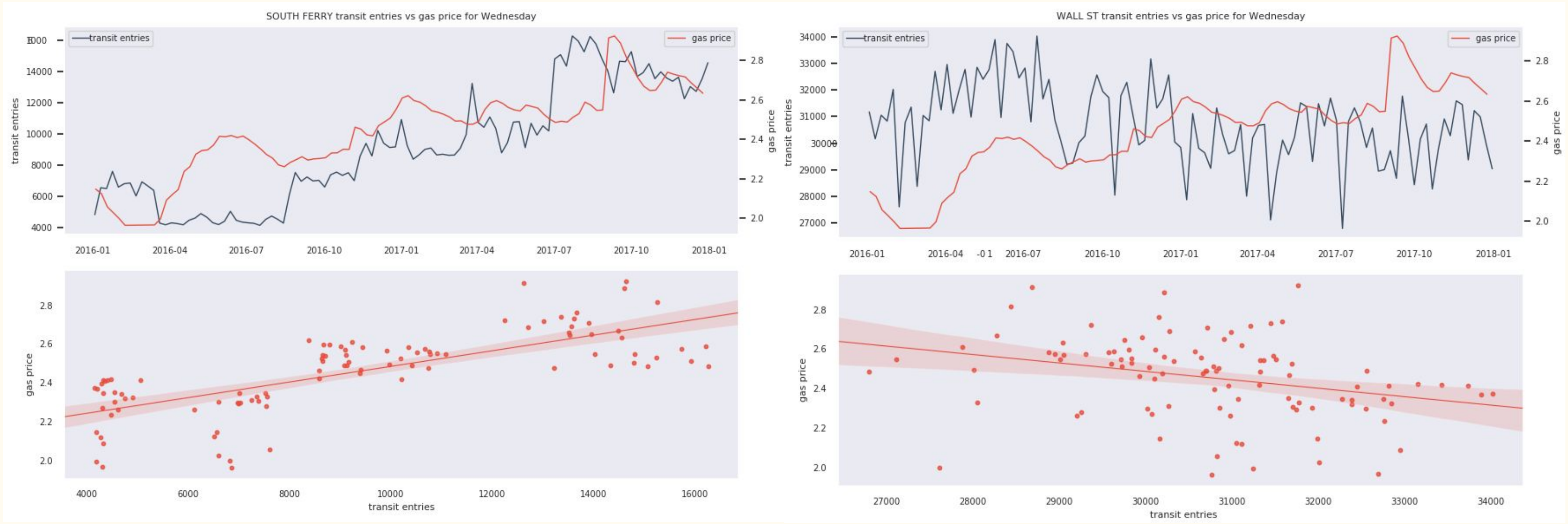
ANALYSIS - TRANSIT vs. TRAFFIC





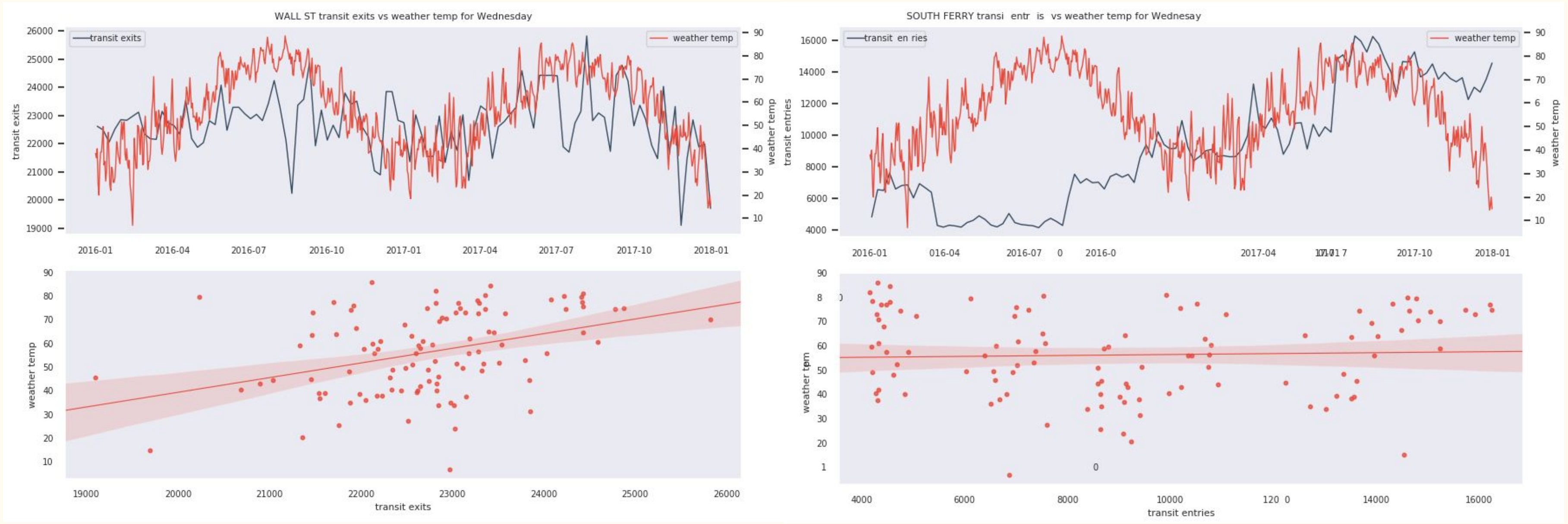
For most stations (refer to plots on left for South Ferry), there is a negative correlation between between the number of people entering (or exiting) the station and the traffic speed in the vicinity of the station. For the Wall Street (refer to plots on right for Wall St) and Court Square stations there is a slight positive correlation between station entries and traffic speed in the vicinity of the station.

ANALYSIS - TRANSIT vs. GAS



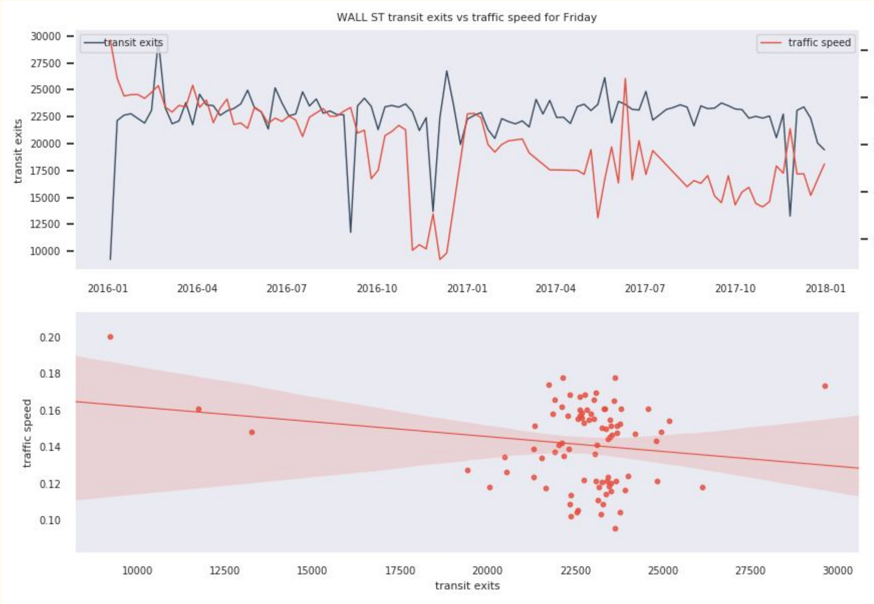
For most stations (refer to plots on left for South Ferry), there is a positive correlation between between the number of people entering (or exiting) the station and gas price. For the Wall Street (refer to plots on right for Wall St) and Court Square stations there is a negative correlation between station entries and gas price.

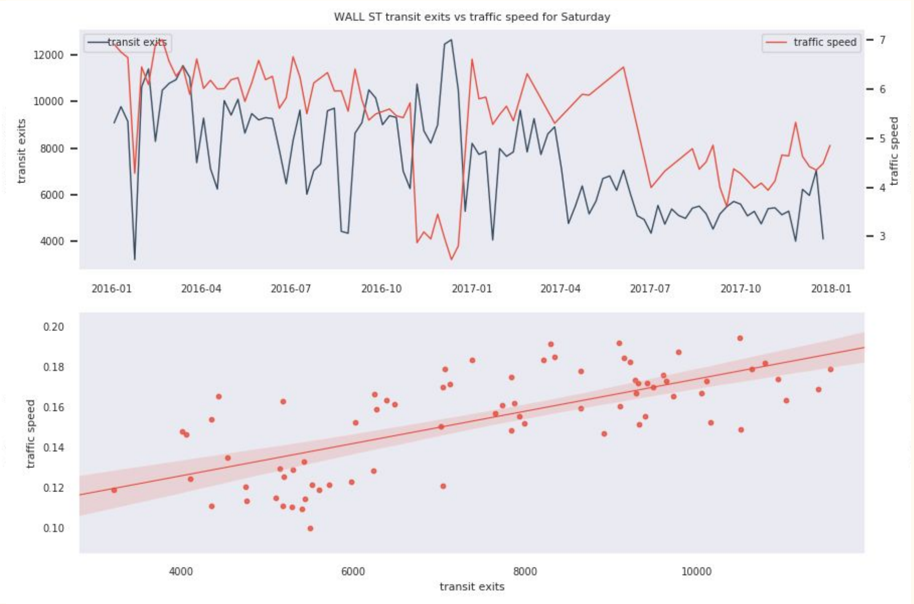
ANALYSIS - TRANSIT vs. WEATHER



For some stations (refer to plots on left for Wall St), there is a positive correlation between the number of people entering (or exiting) the station and mean daily temperature, while for some others (refer to plots on right for South Ferry) there does not seem to be much correlation between weather and transit usage.

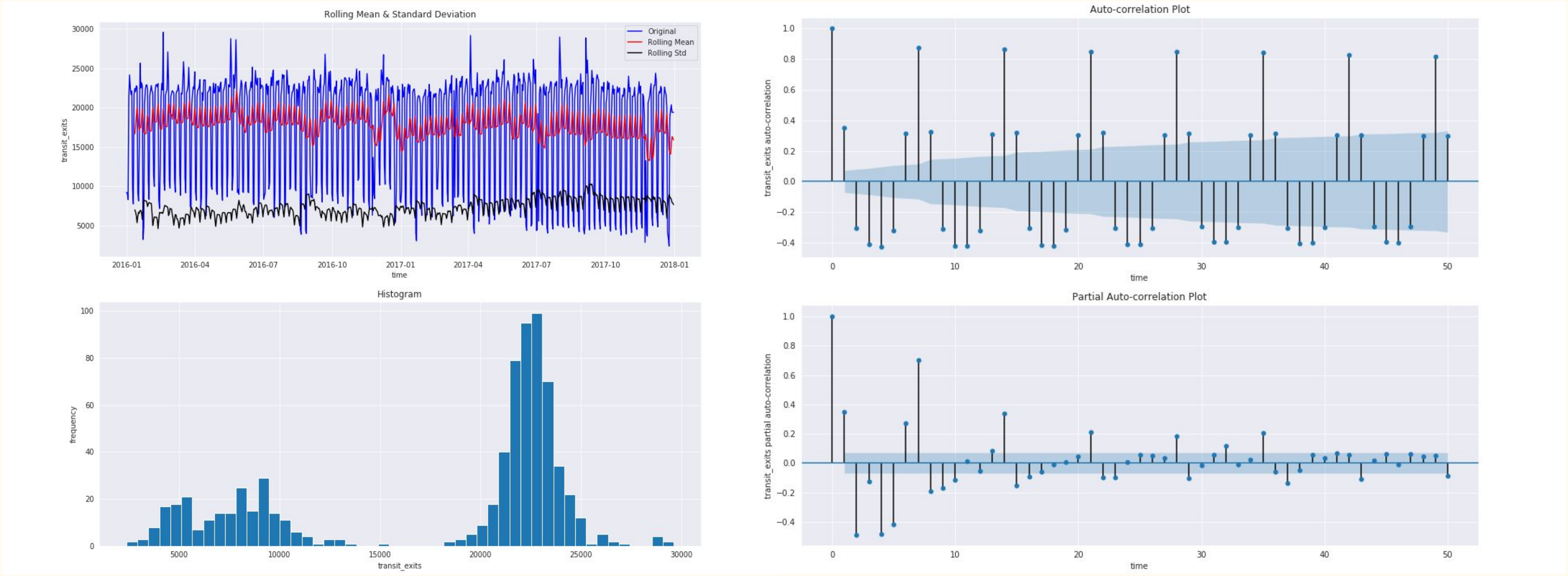
ANALYSIS - WEEKDAY vs. WEEKEND





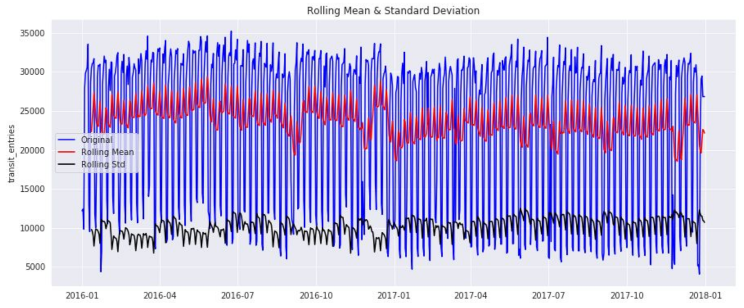
Different correlation is observed between transit riders and traffic on weekdays and weekends (Wall St ridership has a much stronger correlation with traffic on Saturday than on Friday)

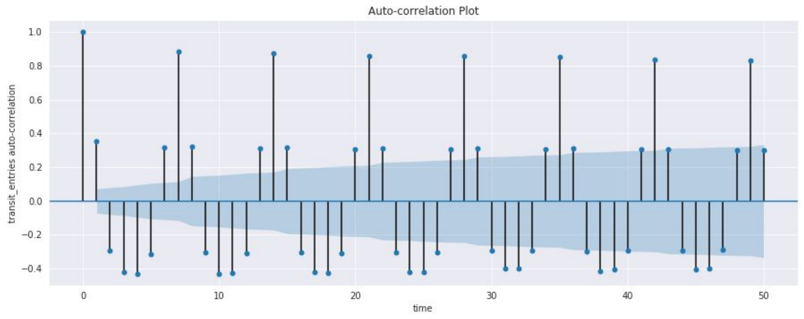
TIME SERIES ANALYSIS - TRANSIT EXITS

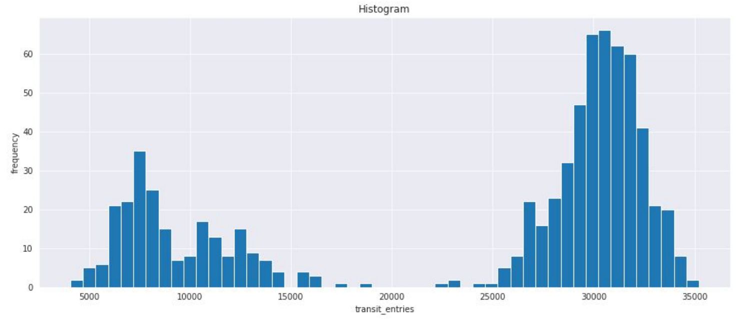


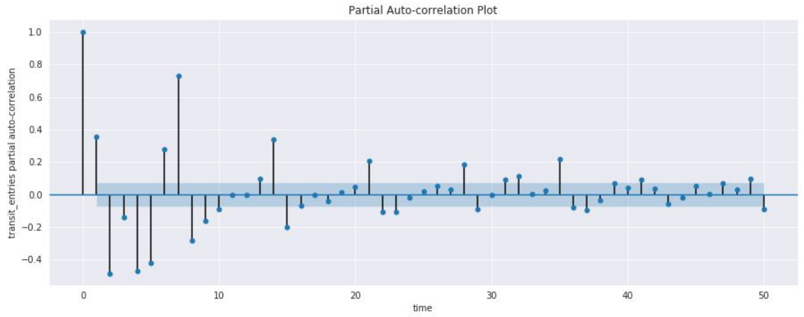
The time series plot (above left) indicates that the data does not have a trend. The Histogram (below left) and the Autocorrelation and Partial Autocorrelation plots (right) indicate that the data is non-stationary and has a lag of 7 days.

TIME SERIES ANALYSIS - TRANSIT ENTRIES



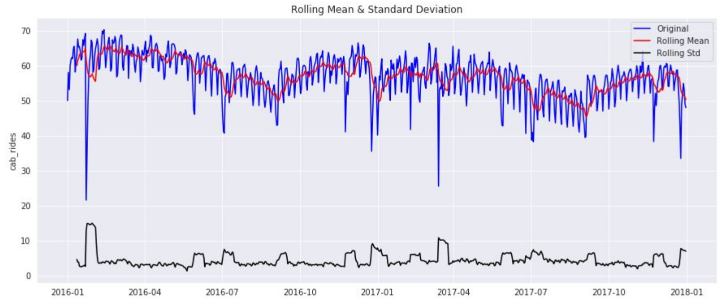


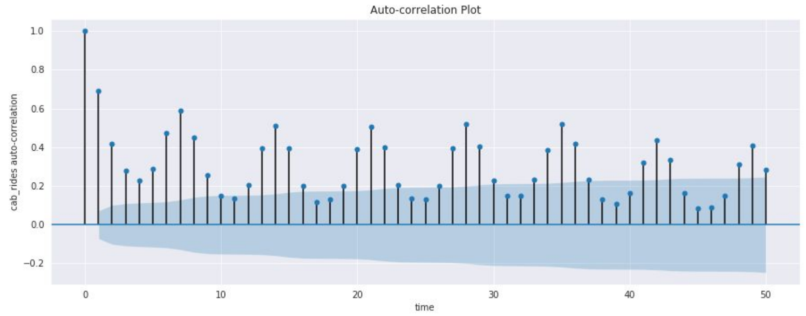


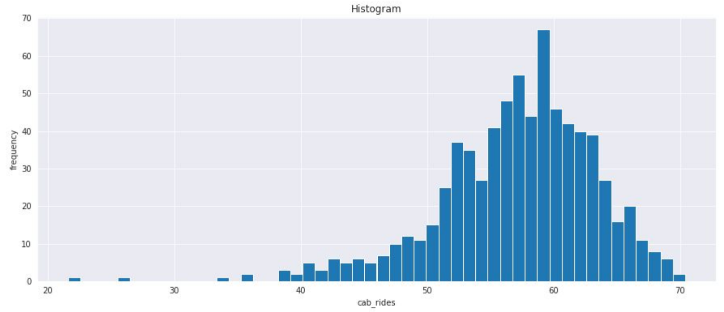


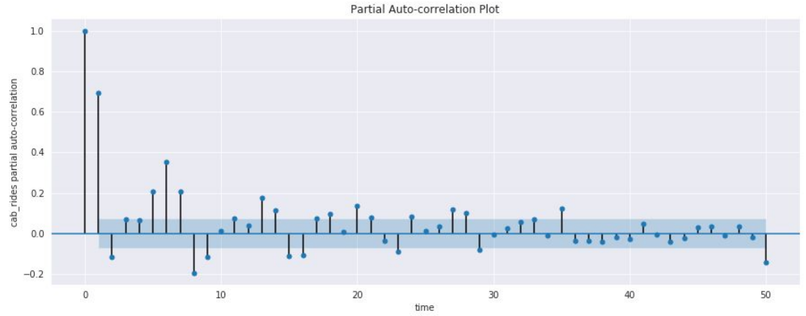
The time series plot (above left) indicates that the data does not have a trend. The Histogram (below left) and the Autocorrelation and Partial Autocorrelation plots (right) indicate that the data is non-stationary and has a lag of 7 days.

TIME SERIES ANALYSIS - CABS



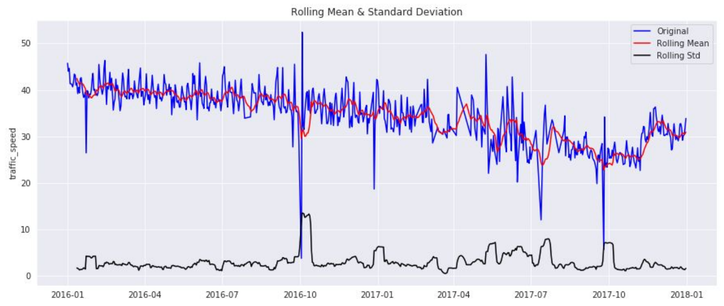


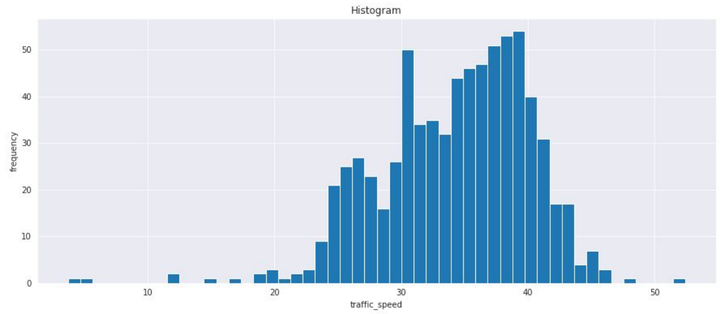


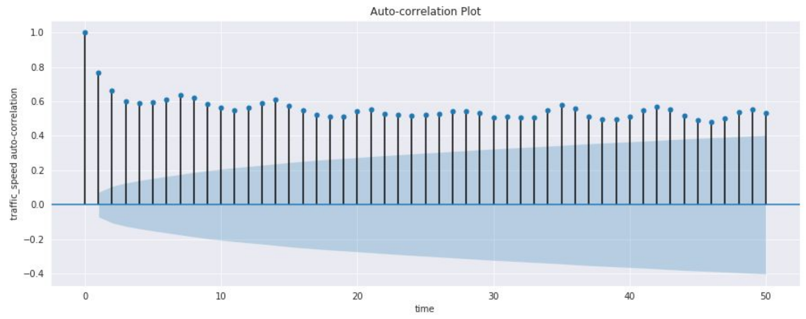


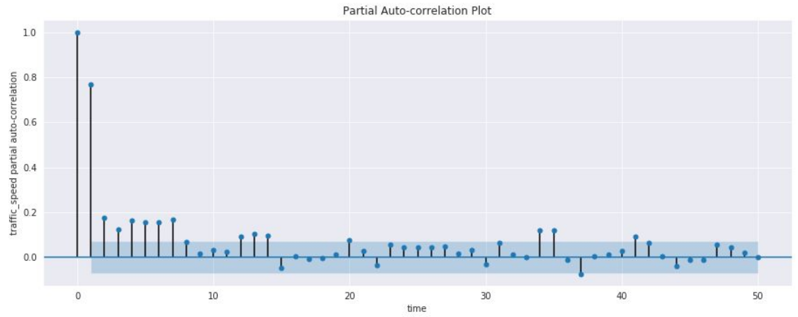
The time series plot (above left) indicates that the data has a varying trend, where it decreases from January to September of 2016; after which it rises, then drops abruptly in January 2017 and again rises until March 2017. It exhibits a downward trend again until September 2017, after which it rises once again. It appears that the trend itself has a seasonality of 1 year. The Autocorrelation and Partial Autocorrelation plots (right) indicate that the data is non-stationary with lags of 7 days and 1 day.

TIME SERIES ANALYSIS - TRAFFIC



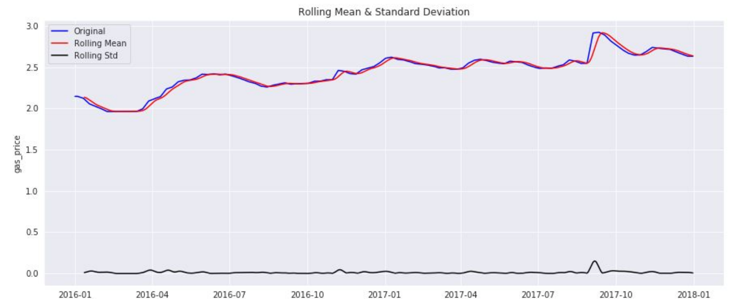


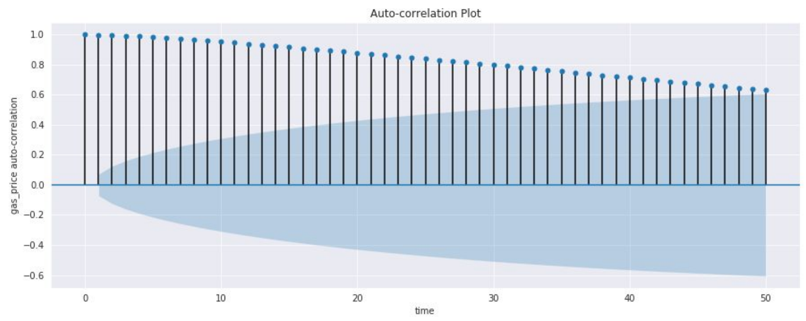


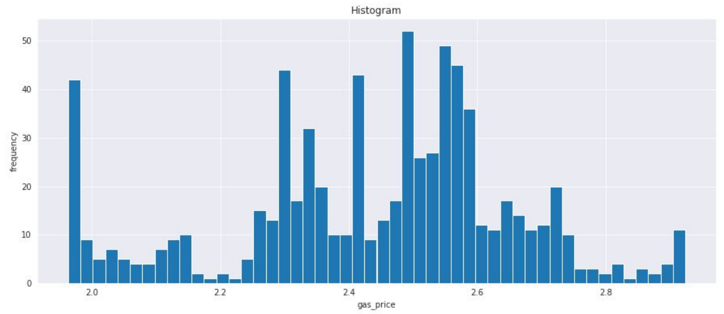


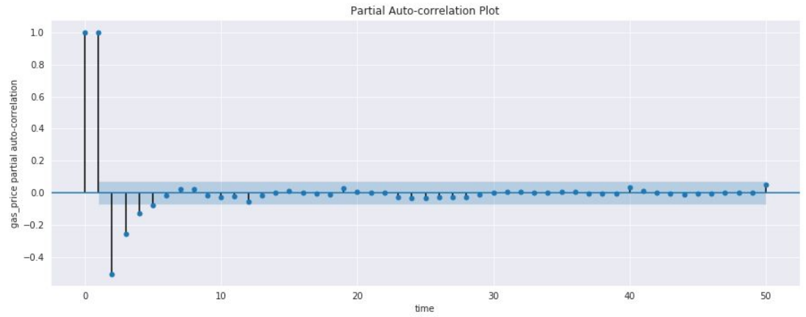
The time series plot (above left) indicates that the data has a general downward trend from January 2016 to October 2017, after which it begins to exhibit an upward trend. The Autocorrelation and Partial Autocorrelation plots (right) indicate that the data is non-stationary with a lag of 1 day.

TIME SERIES ANALYSIS - GAS



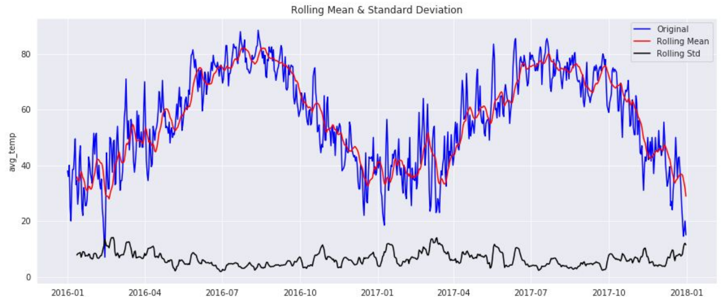


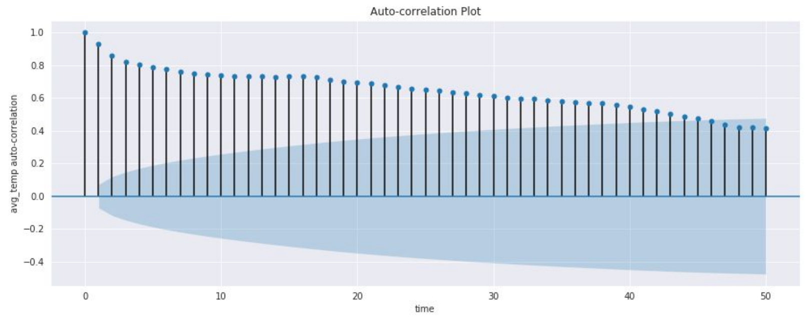


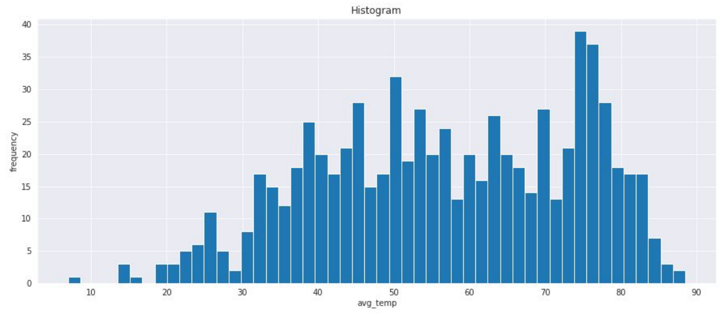


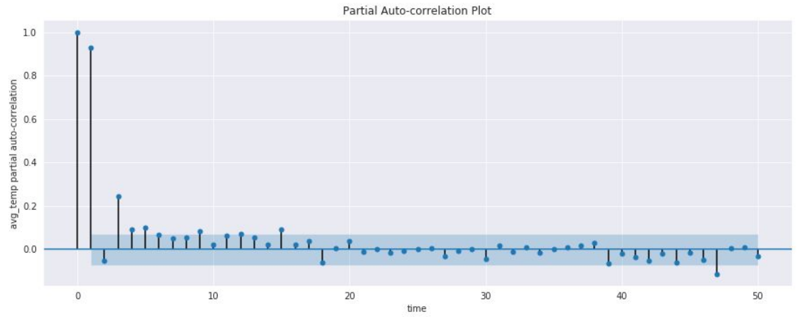
The time series plot (above left) shows a gently rising trend. The Autocorrelation and Partial Autocorrelation plots (right) indicate that the data has significant autocorrelation at lag 1.

TIME SERIES ANALYSIS - WEATHER



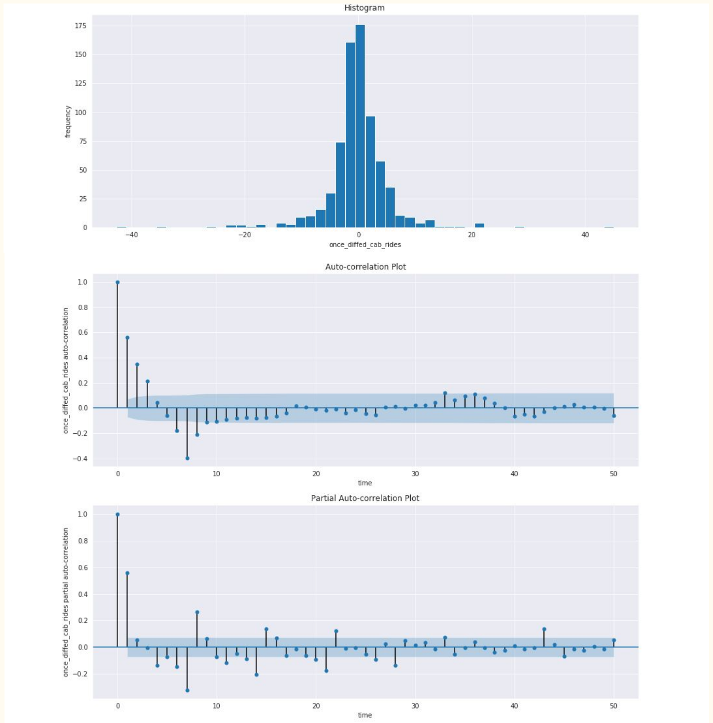


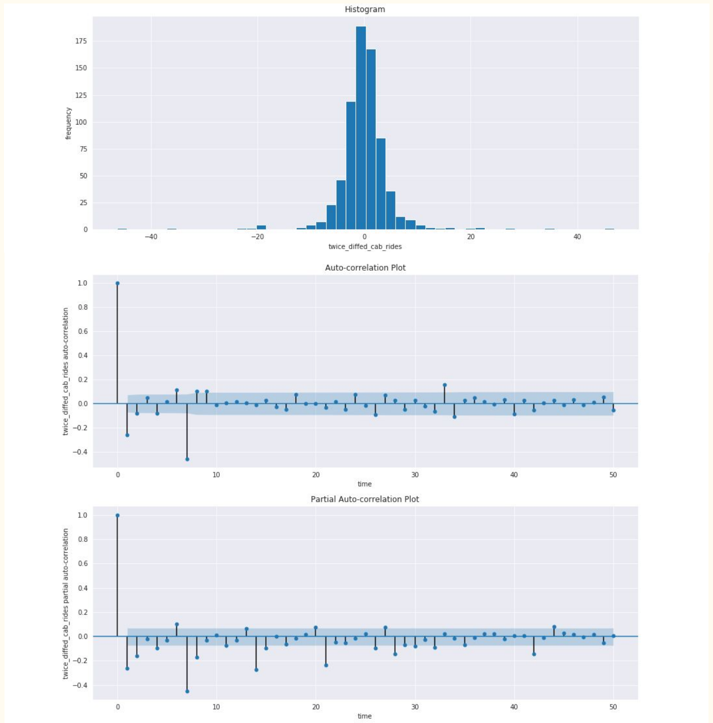




The time series plot (above left) shows a local level with significant variance and annual seasonality. The Autocorrelation and Partial Autocorrelation plots (right) indicate that the data has significant autocorrelation at lag 1 and has a seasonality of 365 days (annual)

SEASONALITY & TREND REMOVAL - CABS



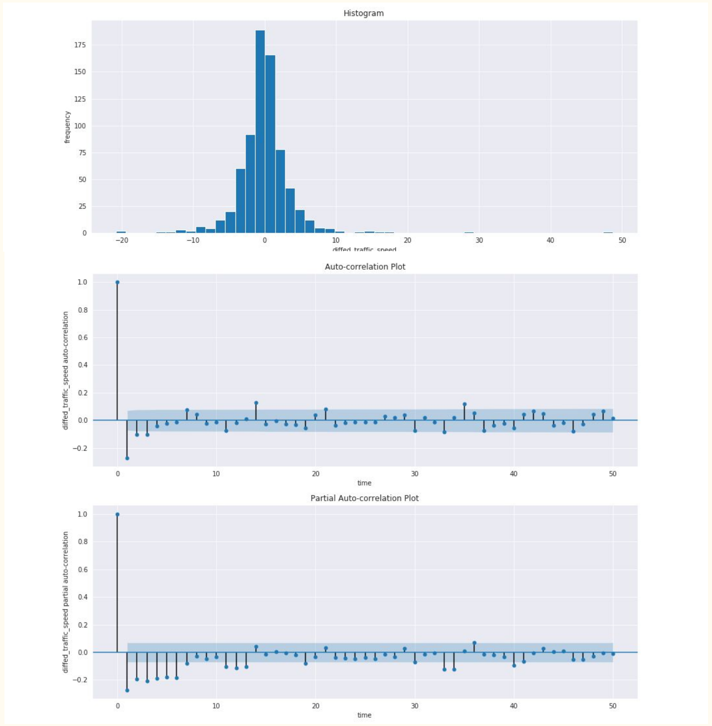


Difference smoothing was carried out for lags 7 and 1. The resulting double differenced data has an approximate normal distribution. On the left are the histogram and acf plots of once differenced cab rides (lag 7). The standard deviation has fallen from 6.27 to 5.59 , but there is still a significant auto-correlation at lag 1. On the right are the histogram and acf plots of twice differenced cab rides(lags 7 and 1). The standard deviation has fallen from 5.59 to 5.22 and there is little residual auto-correlation in the twice differenced data.

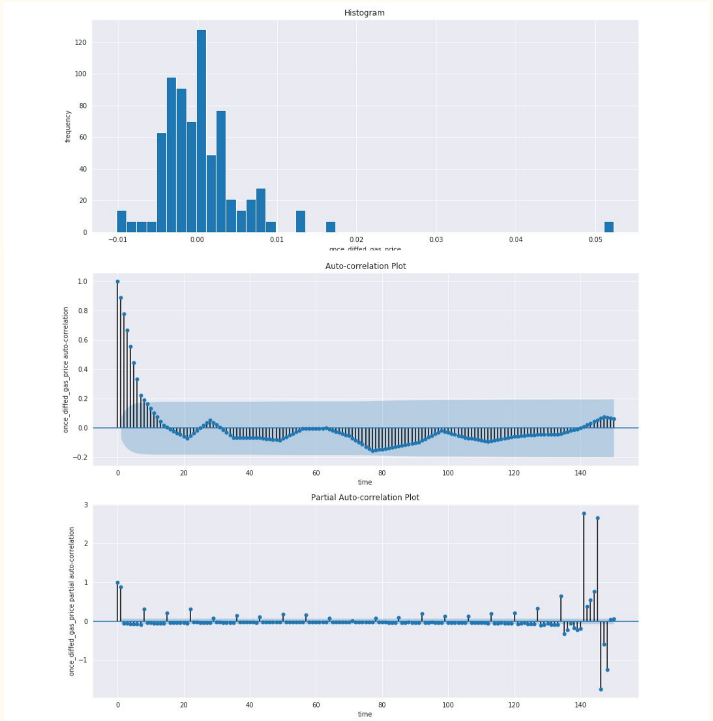
SEASONALITY & TREND REMOVAL - TRAFFIC

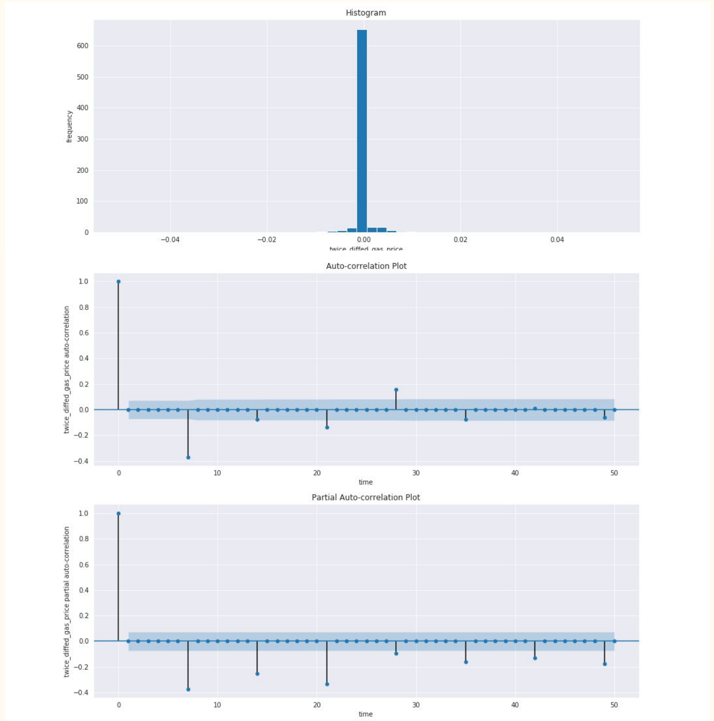
Difference smoothing was carried out for lag 1. The resulting single differenced data has an approximate normal distribution.

On the right are the histogram and acf plots of the once differenced traffic speed data (lag 1). The standard deviation has fallen from 5.89 to 3.99 after the first difference. There is little residual auto-correlation in the differenced data.



SEASONALITY & TREND REMOVAL - GAS



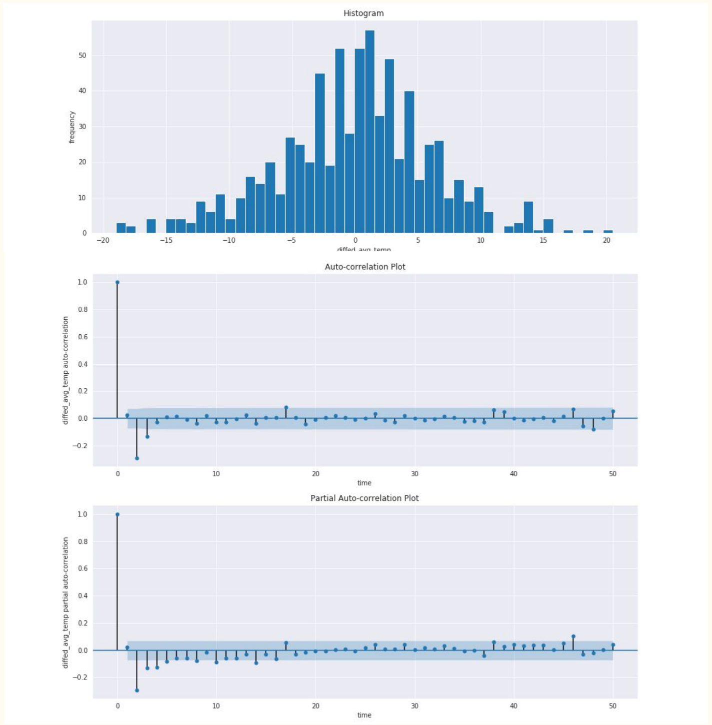


Difference smoothing was carried out twice using lag 1 each time. The resulting double differenced data has an approximate normal distribution. On the left are the histogram and acf plots of the once differenced gas price (lag 1). The standard deviation has fallen from 0.22 to 0.007 after the first difference. There is still a significant auto-correlation at lag 1. On the right are the histogram and acf plots of the twice differenced gas price (lag 1 both times). The standard deviation has fallen from 0.007 to 0.003 and there is little residual auto-correlation in the twice differenced data.

SEASONALITY, TREND REMOVAL - WEATHER

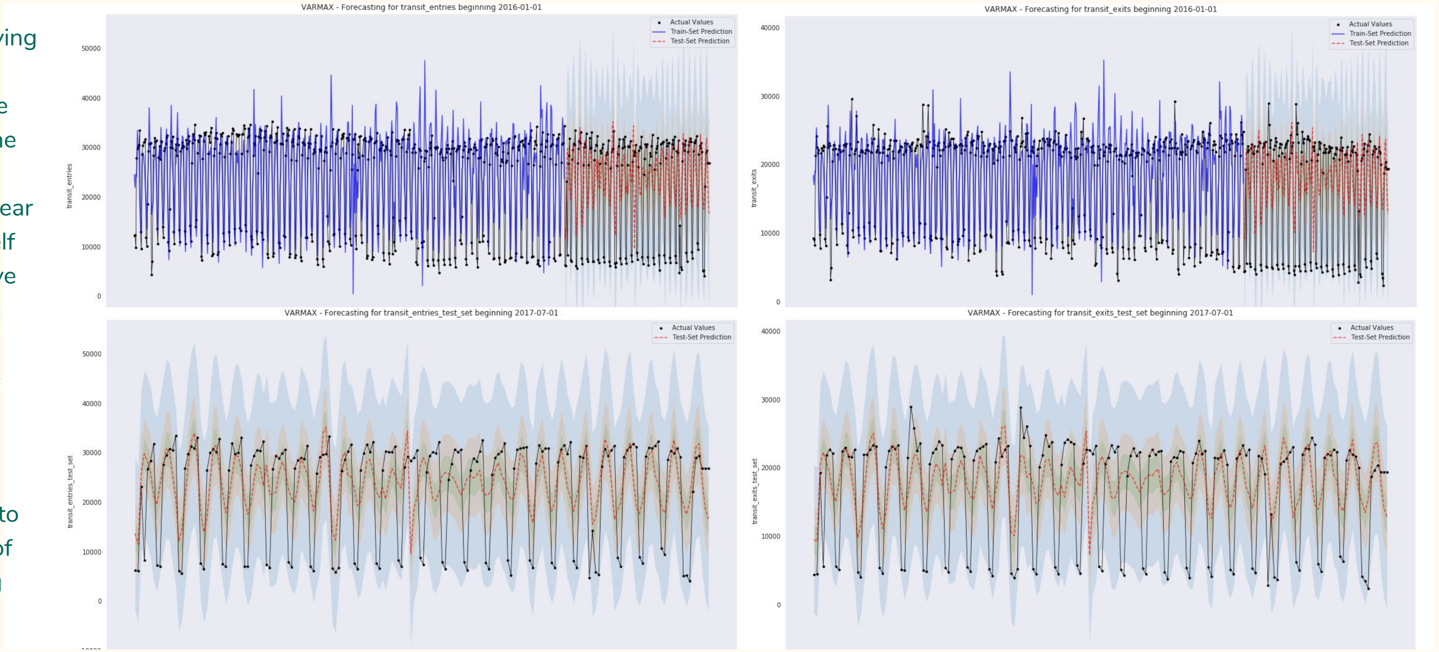
Difference smoothing was carried out for lag 1. The resulting single differenced data has an approximate normal distribution.

On the right are the histogram and acf plots of the once differenced avg. temperature (lag 1). The standard deviation has fallen from 16.98 to 6.14 after the first difference. There is little residual auto-correlation in the differenced data.



TIME SERIES MODELING - VARMAX

Vector Autoregressive Moving Average with eXogenous regressors(VARMAX) is the multivariate extension of the ARMA* model where each variable is modeled as a linear function of past lags of itself (endogenous autoregressive or AR** components), past lags of the other variables (exogenous autoregressive components) and moving average or MA*** components. 5-fold cross-validation was used to find the optimum number of autoregressive and moving average components.

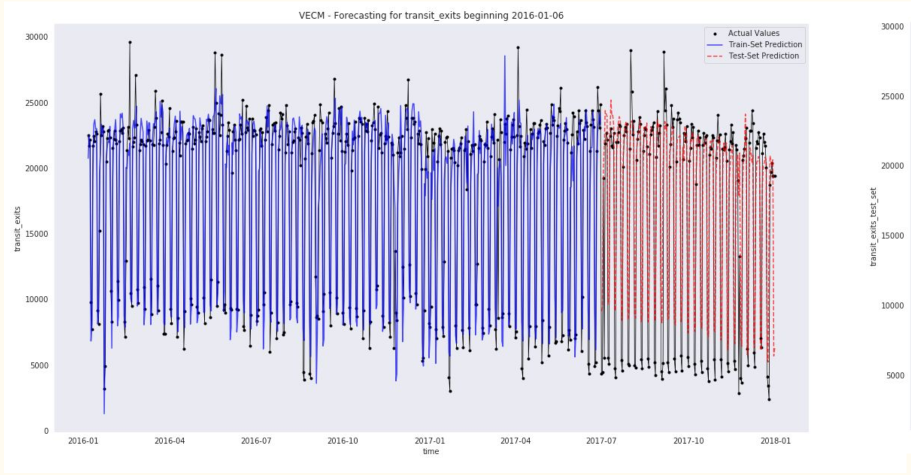


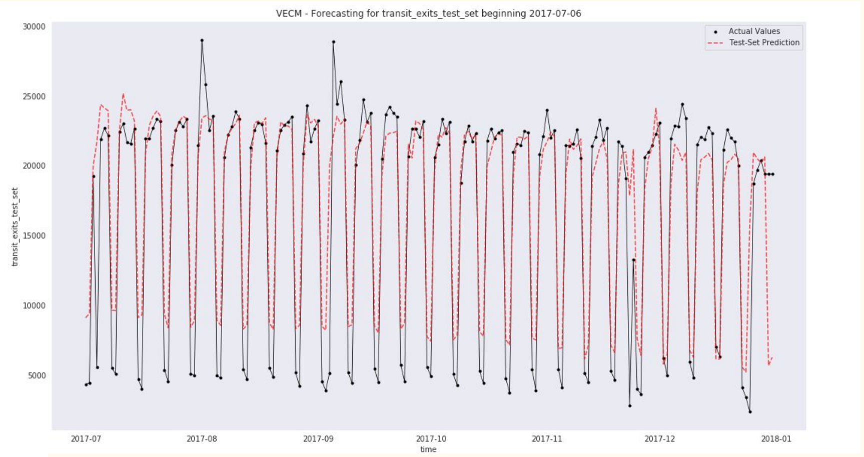
The forecasted values for transit entry and exit counts for Wall St. station were computed using the tuned VARMAX model for the period 1st July to 31st December 2017 . On the top are Actual vs Predicted Value plots for test and train data (train prediction in blue solid, test prediction in red dashed) for Entries(left) and Exits(right). On the bottom are Actual vs Predicted Value plots for test data only (test prediction in red dashed).

*autoregressive–moving-average (ARMA) models provide a model for a stationary stochastic process in terms of two polynomials, one for the autoregression (AR) and the second for the moving average (MA)
**an autoregressive model learns from a series of timed steps and takes measurements from previous actions as inputs for a regression model, in order to predict the value of the next time step
***a moving-average model is conceptually a linear regression of the current value of the series against current and previous (observed) white noise error terms or random shocks

TIME SERIES MODELING - VECM

Vector Error Correction Model (VECM) is used for data that has a long-run stochastic trend, also known as cointegration*. 5-fold cross-validation was used to find the optimum values for the number of lagged differences and the cointegration rank**





The forecasted values for transit entry and exit counts for Wall St. station were computed using the tuned VECM model for the period 1st July to 31st December 2017 . On the left are Actual vs Predicted Value plots for test and train data (train prediction in blue solid, test prediction in red dashed) for Exits. On the right are Actual vs Predicted Value plots for test data only (test prediction in red dashed). The predictions seem to be superior to those of VARMAX.

*If two integrated variables share a common stochastic trend such that a linear combination of these variables is stationary, they are called cointegrated. If there are several linearly independent cointegrating variables in a dataset, then linear combinations of these variables are also cointegrating variables.
**this is the rank of the error-correction matrix (error-correction brings the dependent variable or output back to the long-term equilibrium).

CONCLUSION & FURTHER EXPLORATION

The model evaluation metrics computed on test data for predicting transit exits are,

● Accuracy score (RMSE*) using VARMAX	6773.6
● Accuracy score (RMSE) using VECM	3439.4

The test data prediction plots and the test accuracy scores indicate that VECM performs significantly better than VARMAX.

Further exploration of this time-series forecasting problem include:

- improving the VECM model
  - running the model on undifferenced (non-stationary) inputs (the current version uses stationarized inputs)
  - further tuning of VECM hyperparameters
- exploring other methods like Dynamic Linear Regression, GAS Regression and Beta-t-EGARCH Regression.

*Root Mean Square Error is the standard deviation of the residuals (prediction errors)

REFERENCES

Dask and Geopandas

1. http://matthewrocklin.com/blog/work/2017/01/12/dask-dataframes
2. https://towardsdatascience.com/geospatial-operations-at-scale-with-dask-and-geopandas-4d92d00eb7e8
3. http://matthewrocklin.com/blog/work/2017/09/21/accelerating-geopandas-1
4. https://nbviewer.jupyter.org/urls/gist.githubusercontent.com/mrocklin/ba6d3e2376e478c344af7e874e6fcbb1/raw/e0db89644f78f4371ee30fbdd517ce9bd6032a5e/nyc-taxi-geospatial.ipynb
5. https://github.com/mrocklin/dask-geopandas

Setup with Docker, Redis, Minio

1. https://www.dataquest.io/blog/install-and-configure-docker-swarm-on-ubuntu/
2. https://medium.com/@james.p.edwards42/redis-swarm-12039b486b89
3. https://blog.garage-coding.com/2016/02/05/bash-fifo-jobqueue.html
4. https://testdriven.io/asynchronous-tasks-with-flask-and-redis-queue

REFERENCES

Time Series Analysis

1. https://people.duke.edu/~rnau/411arim3.htm
2. http://www-personal.umich.edu/~lkilian/SVARch03.pdf
3. https://pyflux.readthedocs.io/en/latest/
4. https://otexts.org/fpp2/stationarity.html
5. https://www.youtube.com/watch?v=_vQ0W_qXMxk
6. https://theintelligenceofinformation.wordpress.com/2018/06/18/time-series-analysis-in-python-and-r/
7. https://mxbu.github.io/logbook/2017/01/04/forecasting-swiss-gdp-growth-with-statsmodels/
8. https://machinelearningmastery.com/make-sample-forecasts-arima-python/
9. https://machinelearningmastery.com/time-series-seasonality-with-python/