Springboard Capstone Project 2: Milestone Report 2

Ford GoBike Bike Ride Forecasting

By David Tse

November 11th, 2018

1. Problem Statement

Forecasting overall bike rider demands to balance with bike station supply is one of many important metrics for a successful bike sharing system. Bike sharing systems are a means of renting bicycles where the process of obtaining membership, rental, and bike return is automated via a network of kiosk locations throughout a city. Using these systems, people are able to rent a bike from one location and return it to a different place on an as-needed basis. Currently, there are over 500 bike-sharing programs around the world. The forecasting problem is framed as a time series forecasting problem where the dependent variable is the number of bike rides based on start time, and the independent variable is time (hours or days).

- Bike Sharing Companies may use this information as a proxy to guide their decisions regarding the location and number of docking stations.
- Urban Planners could develop improved transportation strategies for cities that minimize congestion during rush hour leading to improved work productivity.

https://www.kaggle.com/c/bike-sharing-demand

https://s3.amazonaws.com/fordgobike-data/index.html

2. Data Wrangling

a. Cleaning Steps

The data was downloaded as of November 5th, 2018 from Ford GoBike's public dataset. The .csv data contains a list of bike rides at a particular time and its associated data. The raw dataset contains 15 unique columns and nearly 519,700 rows of data.

Each trip is anonymized and includes:

- Trip Duration (seconds)
- Start Time and Date
- End Time and Date
- Start Station ID
- Start Station Name
- Start Station Latitude
- Start Station Longitude
- End Station ID
- End Station Name
- End Station Latitude

- End Station Longitude
- Bike ID
- User Type (Subscriber or Customer – “Subscriber” = Member or “Customer” = Casual)
- Member Year of Birth
- Member Gender

The data was then imported into a Pandas DataFrame for ease of data manipulations. Feature names were adjusted to be short yet meaningful, free of spaces via replacement using underscores and converted to lowercase. The raw data contained bike rides for both 2017 and 2018, however, only data for 2017 was used unless additional data is required to train machine learning models.

https://s3.amazonaws.com/fordgobike-data/index.html

b. Missing Values

Missing values were primarily found in a member’s birth year and gender. These features were not particularly relevant in the scope of the problem; no further work cleaning was performed on these features.

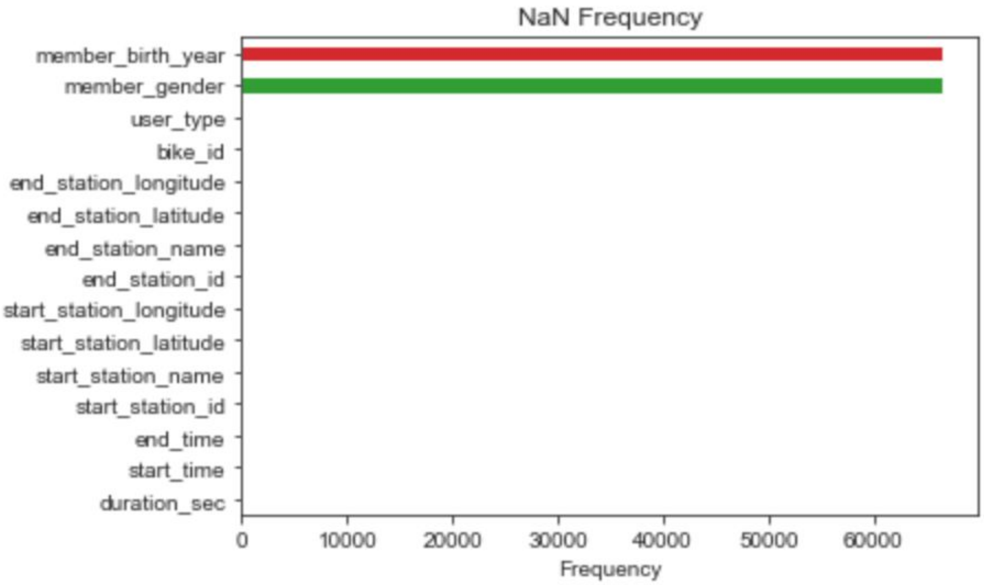

Figure X. Missing Values

c. Outliers

There do not appear to be outliers based on the time series plots and box and whiskers plots. Although there does seem to be a significant fluctuation downward near December 2017.

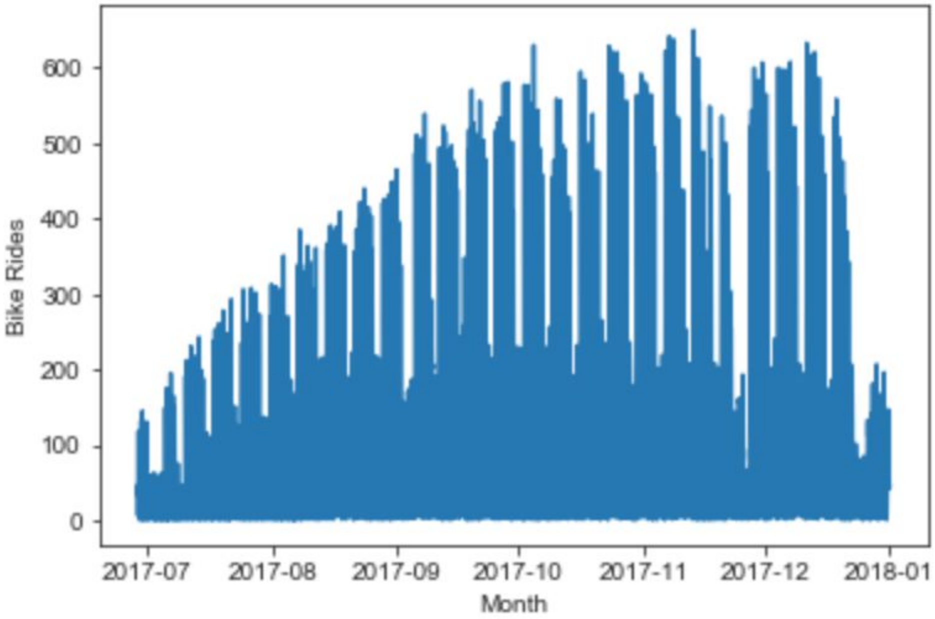

Figure X. Bikes Rides by Hour

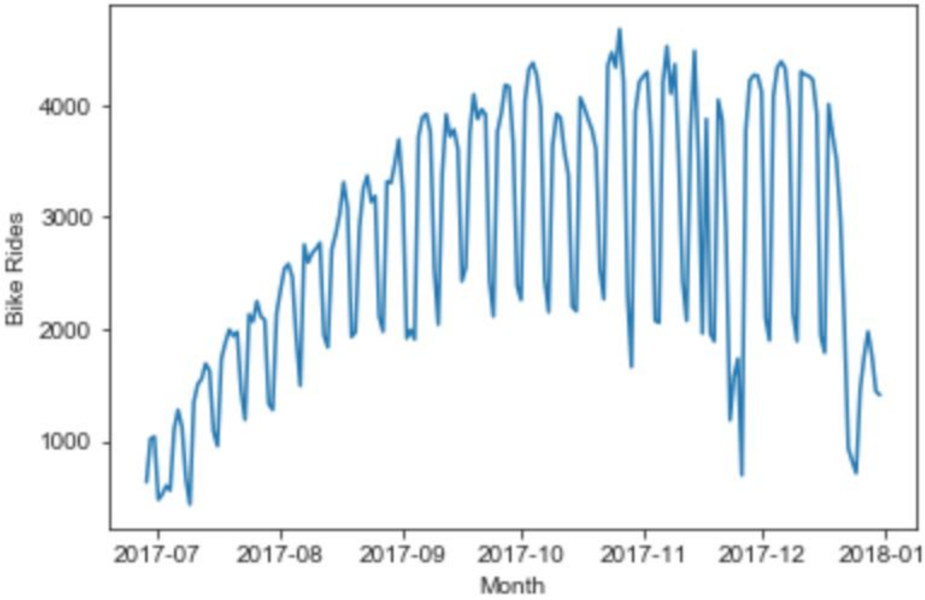

Figure X. Bikes Rides by Day

The bike rides were plotted with box and whiskers plots, and the following was observed:

- General increasing trend until the last month, where there was a dip

- The spread seems to be increasing with time
- No outliers are present

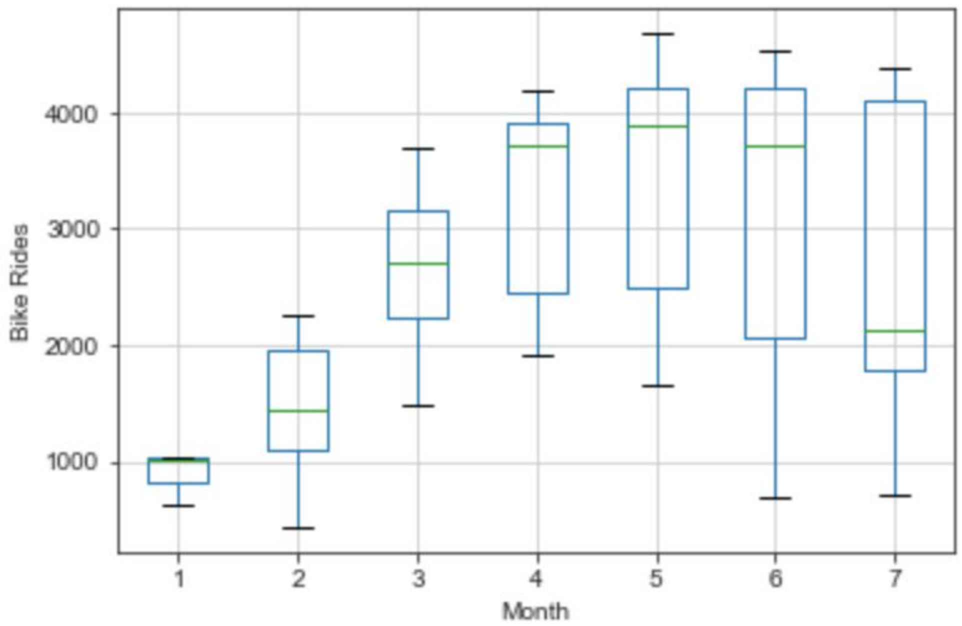

Figure X. Bike Rides by Month

3. Exploratory Data Analysis

a. Initial Trends and Questions Explored

The data was initially explored for any potential outliers and trends by creating and answering questions, including:

- **What is the distribution of bike rides?**

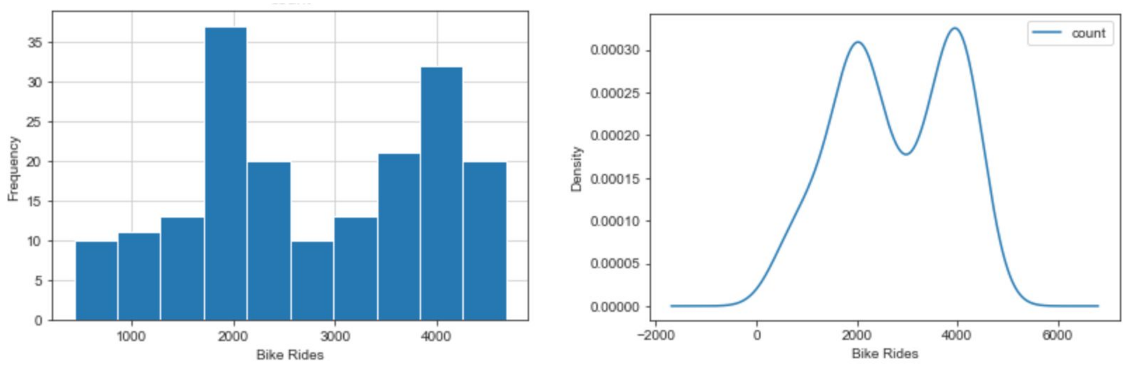

Figure X. Daily Bike Ride Distribution

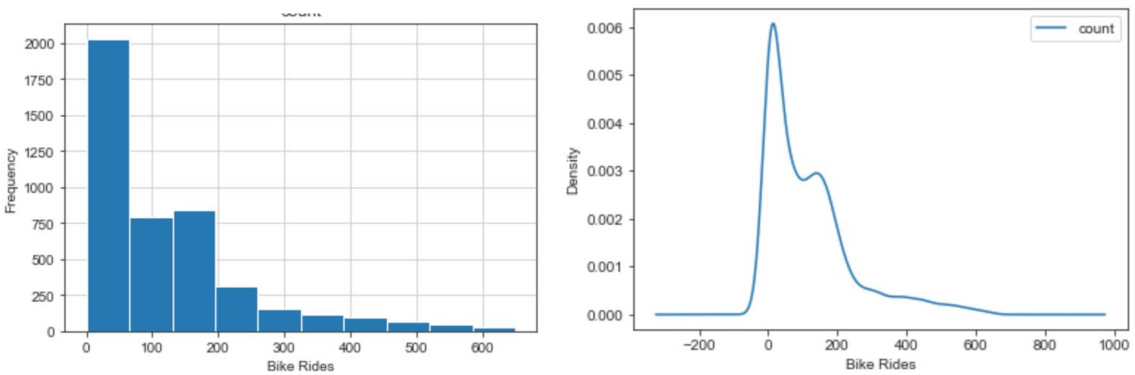

Figure X. Hourly Bike Ride Distribution

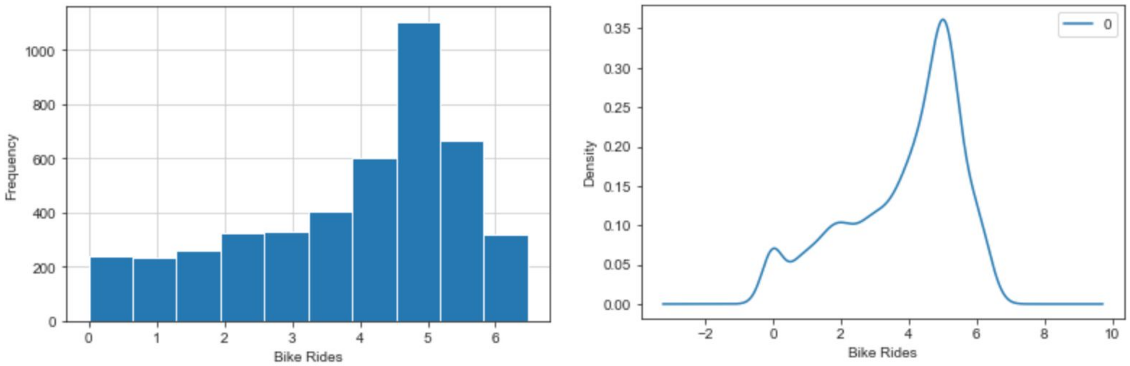

Figure X. Log Transformed Hourly Bike Ride Distribution

- **How is the time series correlated with previous time lags?** More points tighter into the diagonal line suggests a stronger relationship and more spread from the line suggests a weaker relationship. The plots generally show a positive correlation with each value in the last week.

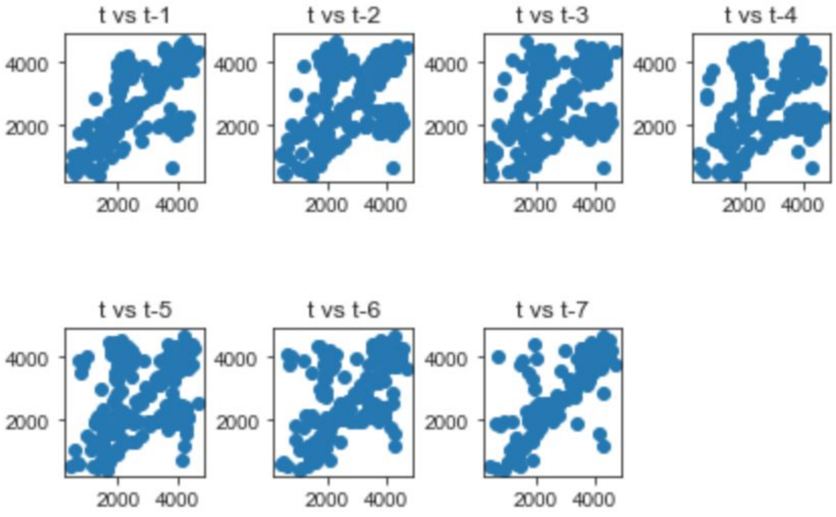

Figure X. Autocorrelation Plots with Increasing Lag Values

- **What does the time series decomposition look like?** Weekly Seasonality and a positive trend exist in the data.

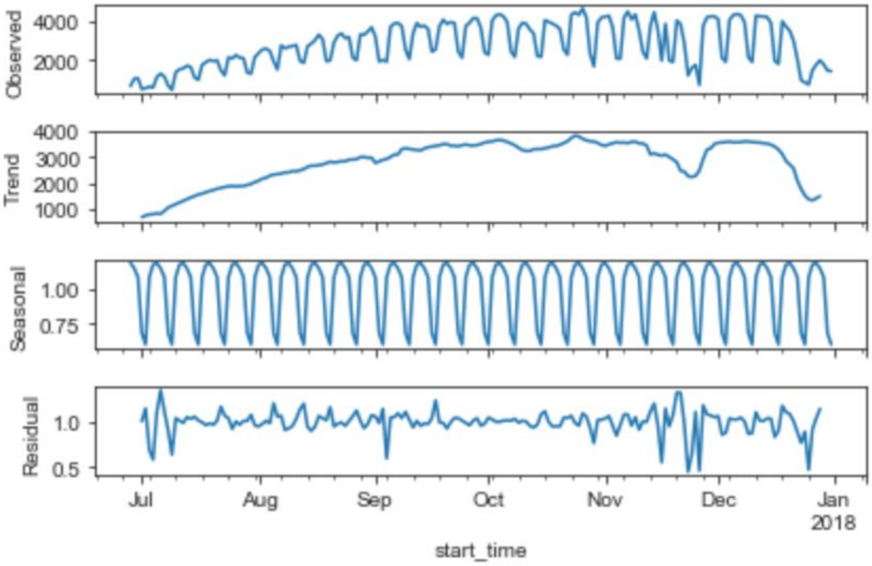

Figure X. Time Series Decomposition

- **Where are Ford GoBikes used most often in the Bay Area?** Downtown San Francisco, Oakland, and Berkeley see the most usage in that order.

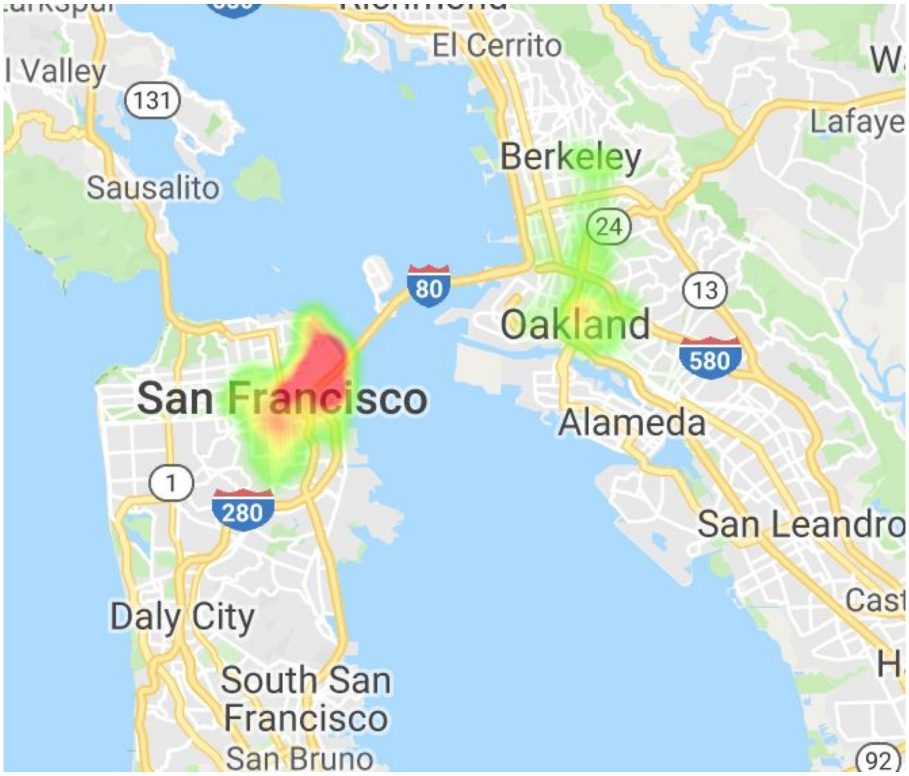

Figure X. Start_time station location heatmap

b. Statistical Analyses

Stationarity is one assumption to demonstrate for linear time series forecasting methods like Autoregressive Integrated Moving Average Models (ARIMA) and leads to better performance for some nonlinear forecasting methods like Long Short-Term Memory (LSTM) Recurrent Neural Networks (RNN). Stationarity could be achieved using differencing and assessed using the augmented Dickey-Fuller test, a type of statistical test called a unit root test. The intuition behind a unit root test is that it determines how strongly a time series is defined by a trend.

For the daily dataset, It was shown that the test statistic is -2.811, which is more extreme than the critical value at ~5%, therefore we reject the null hypothesis. The resulting time series is stationary and does not have a time-dependent structure.

- ADF Statistic = -2.811794
- P-value = 0.056609
- Critical Values:
  - 1%: -3.469
  - 5%: -2.879
  - 10%: -2.576

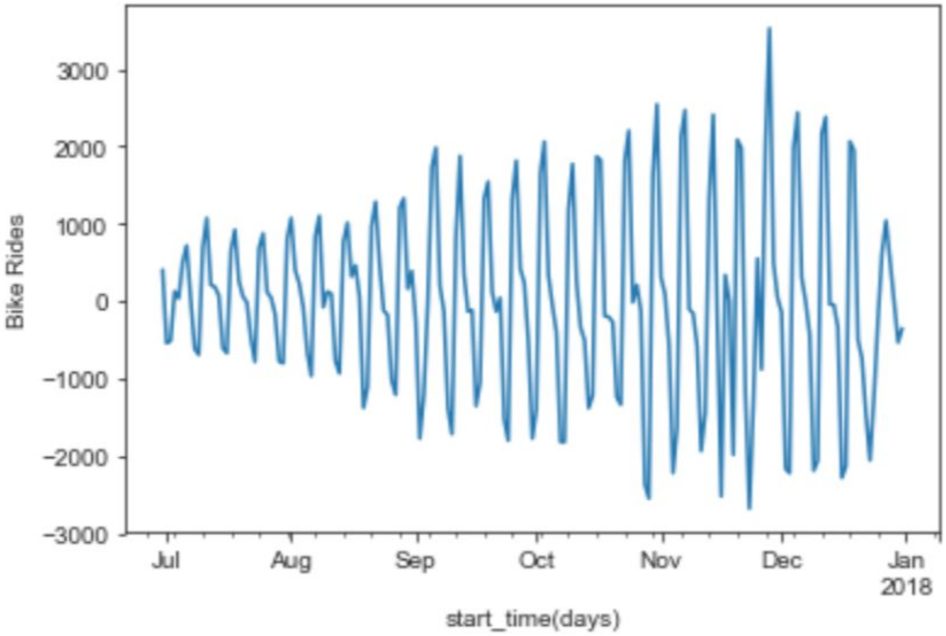

Figure X. Two-Day Differencing of Daily Bike Rides

For the hourly dataset, It was shown that the test statistic is -11.742686, which is more extreme than the critical value at 5%, therefore we reject the null hypothesis. The resulting time series is stationary and does not have a time-dependent structure.

- ADF Statistic = -11.742686
- P-value = 0.000000
- Critical Values:
  - 1%: -3.432
  - 5%: -2.862
  - 10%: -2.567

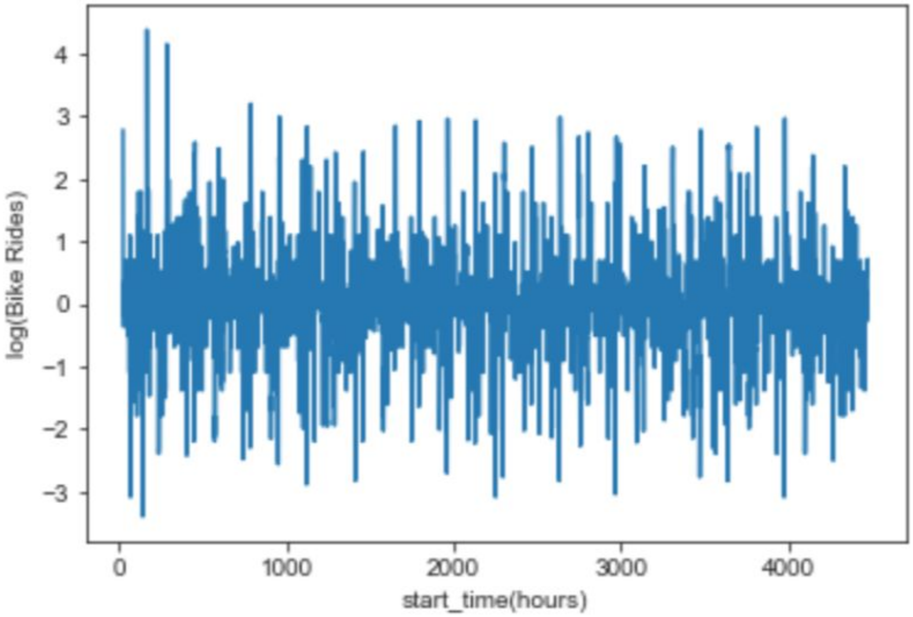

Figure X. 24-Hour Differencing of Daily Bike Rides

Autocorrelation function (ACF) and partial autocorrelation function (PACF) were calculated using the differenced hourly and daily dataset. Autocorrelation is the correlation from an observation to an observation at a prior time step with intermediate time steps considered, whereas partial autocorrelation does not include intermediate time steps in the calculation. All plots were differenced to ensure statistically significant stationarity.

- The model is AR if the ACF trails off after a lag and has a hard cut-off in the PACF after a lag. This lag is taken as the value for p.
- The model is MA if the PACF trails off after a lag and has a hard cut-off in the ACF after the lag. This lag value is taken as the value for q.
- The model is a mix of AR and MA if both the ACF and PACF trail off.

The following was observed for the daily dataset:

- Raw Data:
  - The ACF shows significant lags to nearly 30-day time steps.
  - The PACF shows significant lags to a 60-day time step.
  - A noticeable spike was observed at day 61
- Differenced Data:
  - The two-day differenced ACF shows significant lags to nearly 50-day time steps.
  - The two-day differenced PACF shows significant lags to a 60-day time step.
  - A noticeable spike was observed at day 60

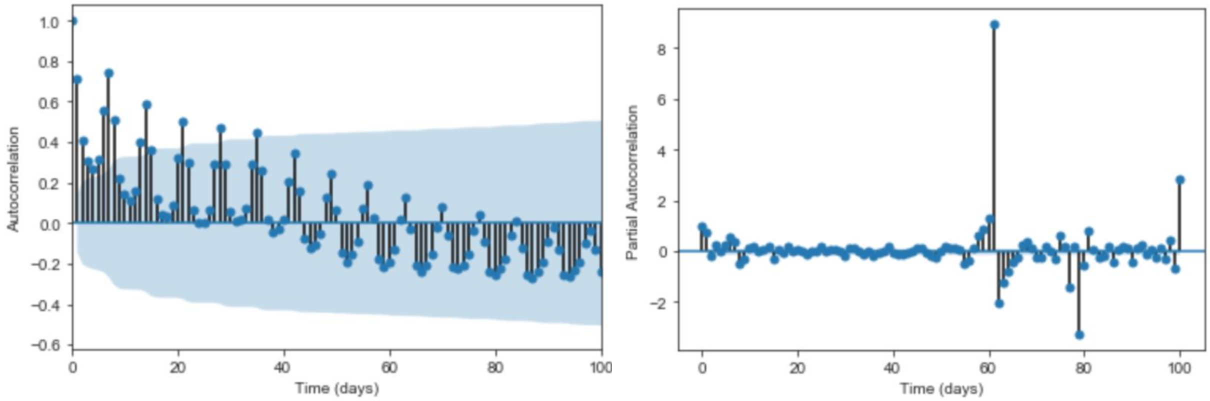

Figure X. Daily ACF and PACF Plots

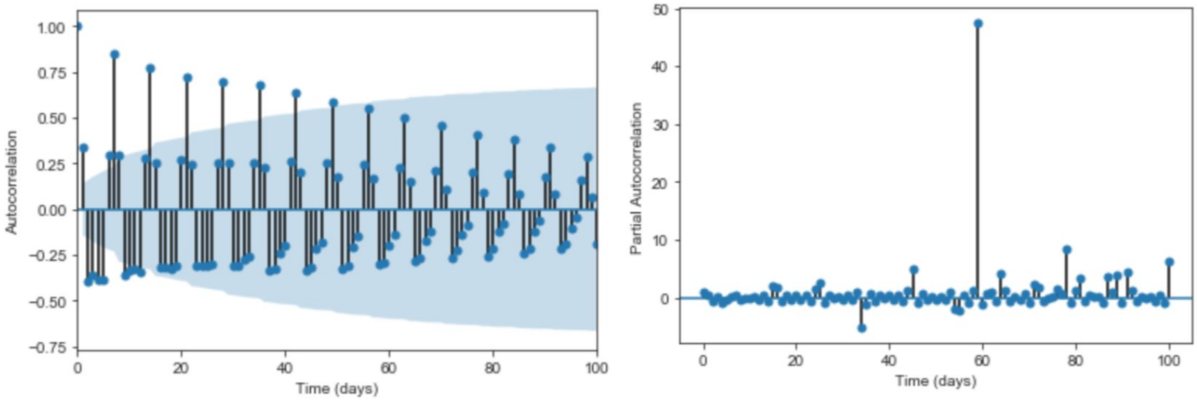

Figure X. Daily ACF and PACF Plots w/ 2-Day Differencing

A seasonal component of 24-hours was observed based on the plots below and multiples of 24-hour seasonality are viable as well, e.g. 48-hour, 72-hour, etc.

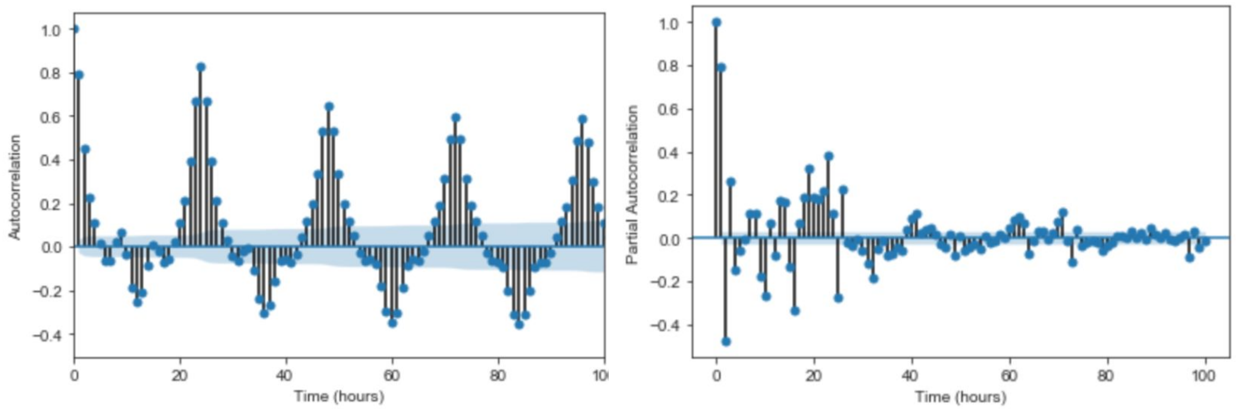

Figure X. Hourly ACF and PACF Plots

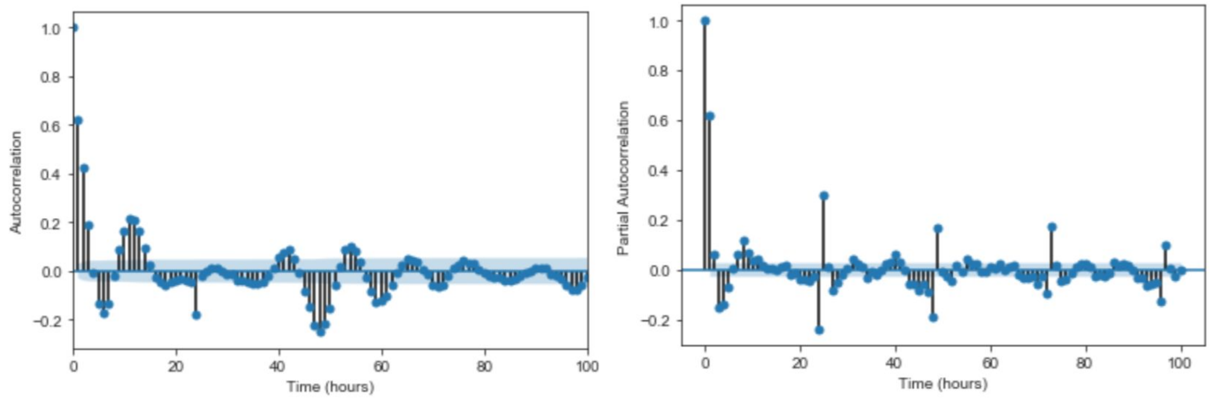

Figure X. Hourly ACF and PACF Plots w/ 24-Hour differencing

4. Machine Learning

The time series forecasting problem is framed as a supervised learning problem. Both linear and nonlinear models were used to develop predictions of both hourly and daily downsampled data. Both daily and hourly data were used for these forecasts to determine the advantages and disadvantages of the chosen forecasting models at differing time steps. A non-seasonal Autoregressive Integrated Moving Average (ARIMA) model was chosen as the linear forecasting model, while a Long Short-Term Memory Recurrent Neural Network (LSTM RNNs) model was chosen as the nonlinear forecasting model. These models were compared to a persistence model, or naive model, for a baseline comparison.

The models were assessed using walk-forward validation (rolling forecast) that outputs the model RMSE. The benefit of RMSE is that it penalizes large errors and the scores are in the same units as the forecast values. The walk-forward validation method was used because, in practice, time-dependent models are typically retrained as new data becomes available. This would give the model the best opportunity to make good forecasts at each time step.

a. Persistence Model

The persistence algorithm uses the value at the current time step (t) to predict the expected

outcome at the next time step (t+1). In this way, the method exhibits the following qualities:

- Simple: A method that requires little or no training or intelligence.
- Fast: A method that is fast to implement and computationally trivial to make a prediction.
- Repeatable: A method that is deterministic, meaning that it produces an expected output given the same input.

The daily persistence model resulted in an RMSE value of 1118.

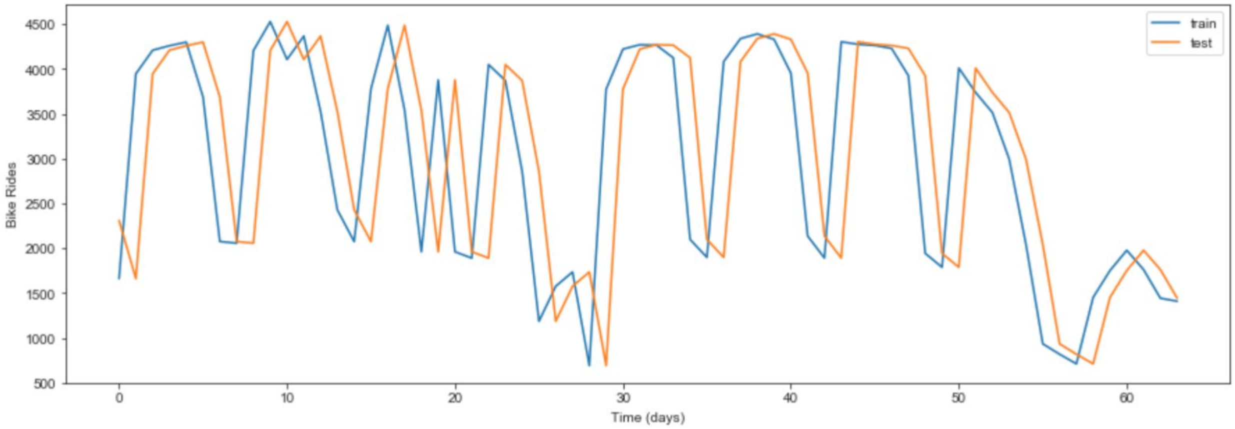

Figure X. Daily Time Series Forecast

The log hourly persistence model resulted in an RMSE value of 1.766.

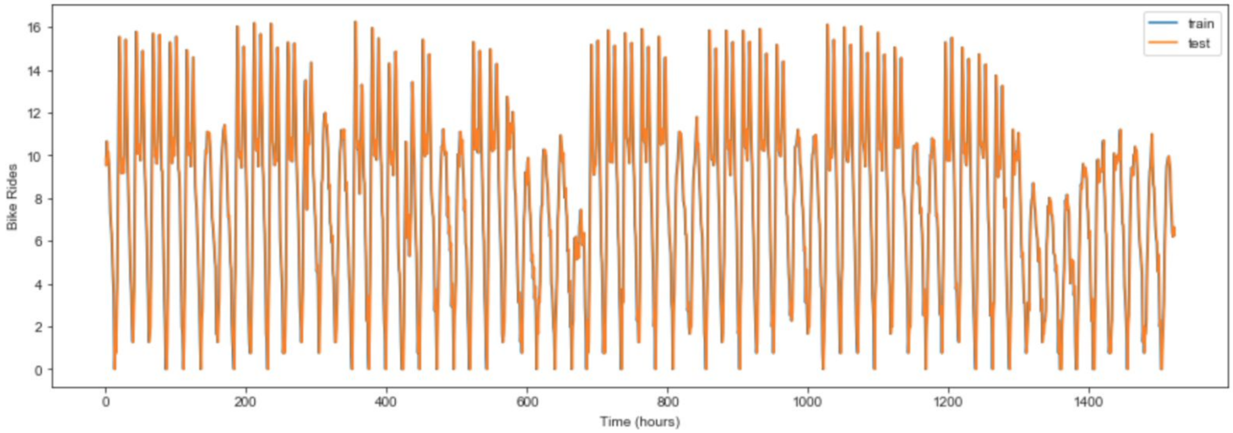

Figure X. Hourly Time Series Forecast

b. Non-Seasonal ARIMA Model

A non-seasonal ARIMA model is based on a linear combination of autoregression terms, moving average terms with differencing (for stationarity). The model assumes stationarity and normality of residuals, which must be checked for model validity.

The daily forecast model was shown to outperform the persistence model RMSE of 1118 with an RMSE of 777. The residuals visually look normally distributed and centered at zero after removing a bias term. This observation is further confirmed using a Q-Q plot; residuals are centered, but there are deviations near the zero quartiles indicating residuals with high peakedness.

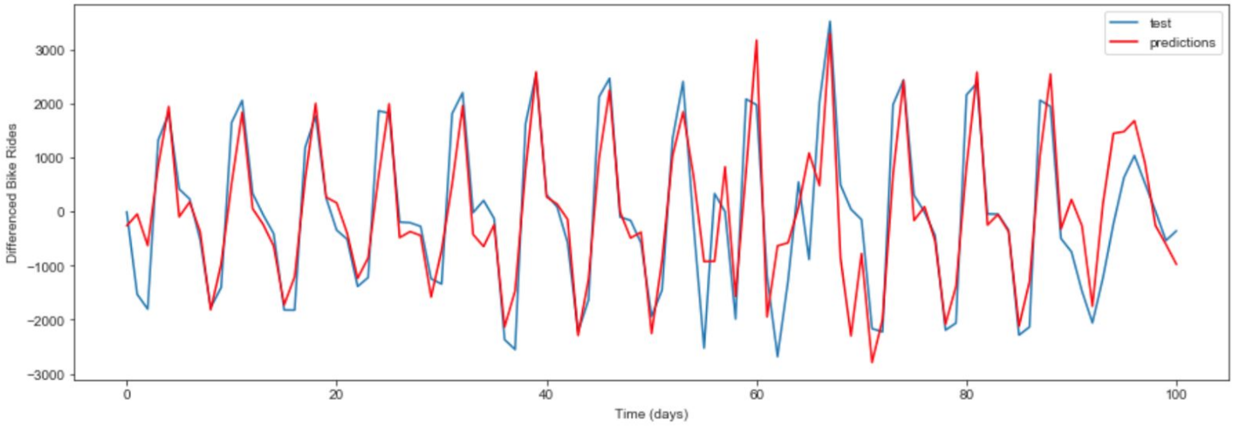

Figure X. ARIMA Daily Time Series Forecast

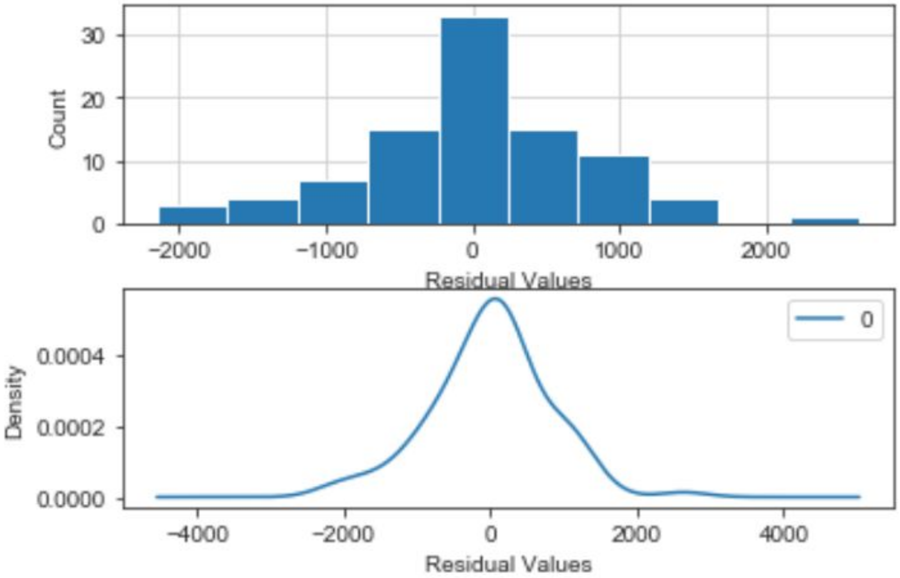

Figure X. ARIMA Daily Time Series Residuals Distribution

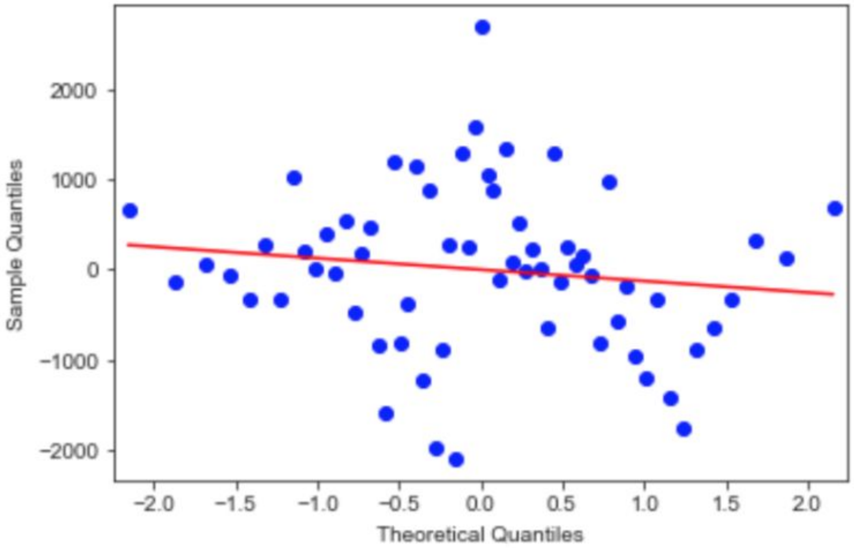

Figure X. ARIMA Daily Time Series QQ Plot

The daily forecast model was shown to outperform the persistence model RMSE of 1.766 with an RMSE of 0.657. The residuals visually look normally distributed and centered at zero after removing a bias term. This observation is further confirmed using a Q-Q plot; residuals are centered, but there are deviations near the zero quartiles indicating residuals with high peakedness.

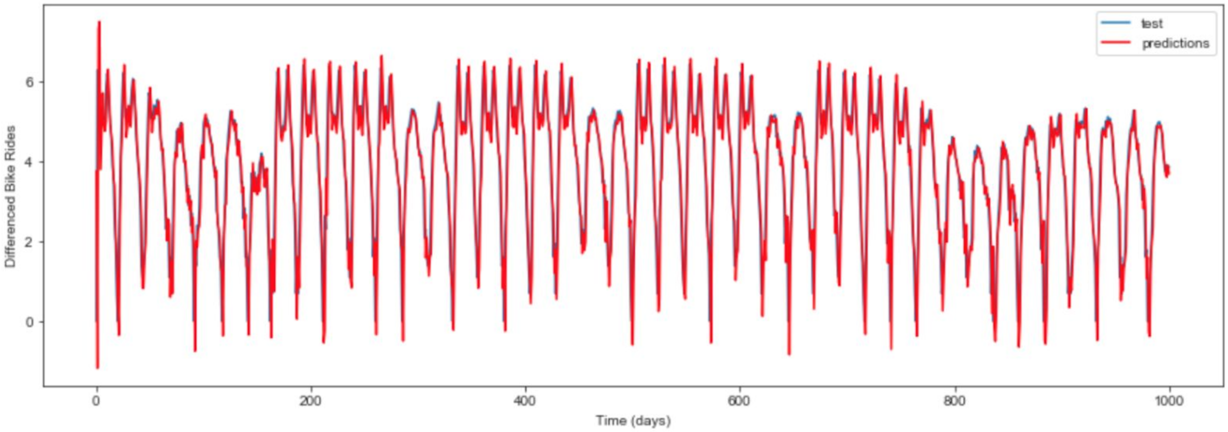

Figure X. ARIMA Hourly Time Series Forecast

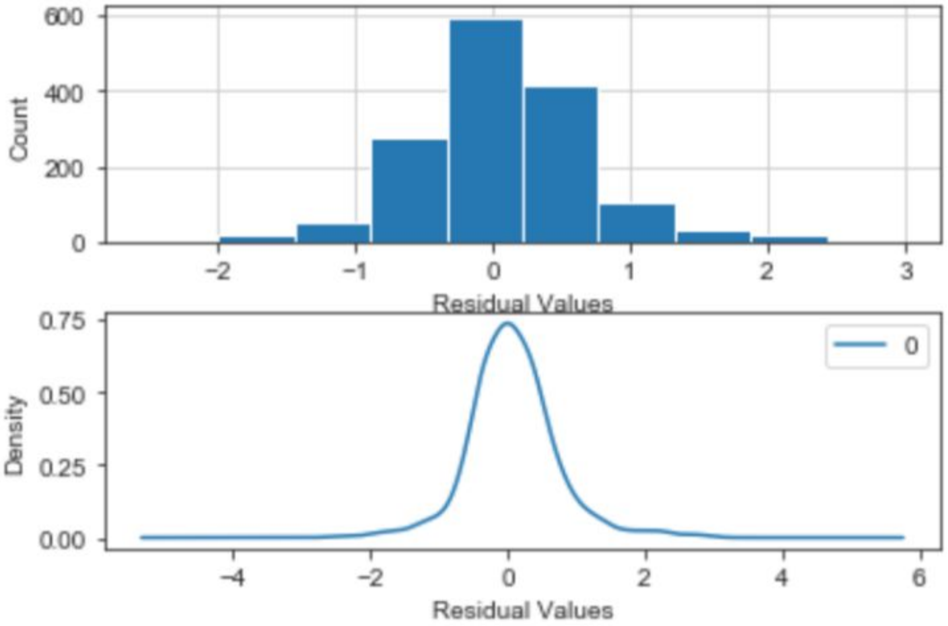

Figure X. ARIMA Hourly Time Series Residuals Distribution

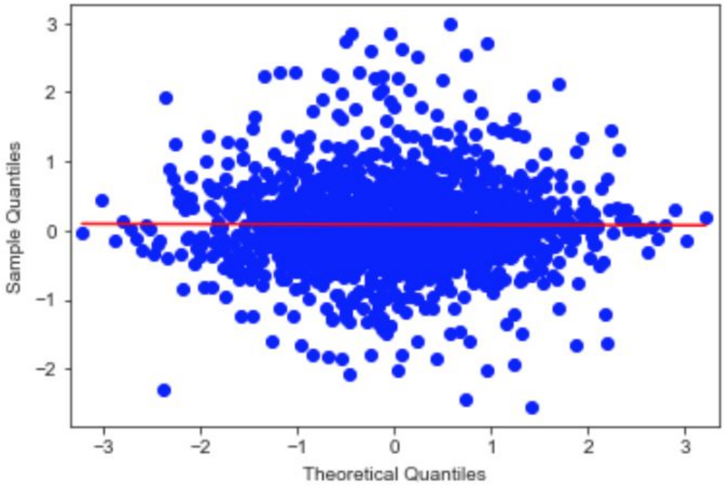

Figure X. ARIMA Hourly Time Series QQ Plot

c. LSTM RNN Model

An LSTM RNN is a type of deep neural network model that is better at retaining long-term temporal “memory” over a traditional RNN, which has better capabilities with short-term “memory.” LSTM RNNs accomplish this by including a “memory cell” where a set of gates is used to control when information enters the memory, outputs, and forgotten. Although not required, the hourly and daily data was differenced to ensure stationarity. LSTM typically performs better if the data is stationary.

The daily forecast model underperformed with an RMSE of 1393.481 +/- 264.903 compared to the persistence model's RMSE value of 1118.

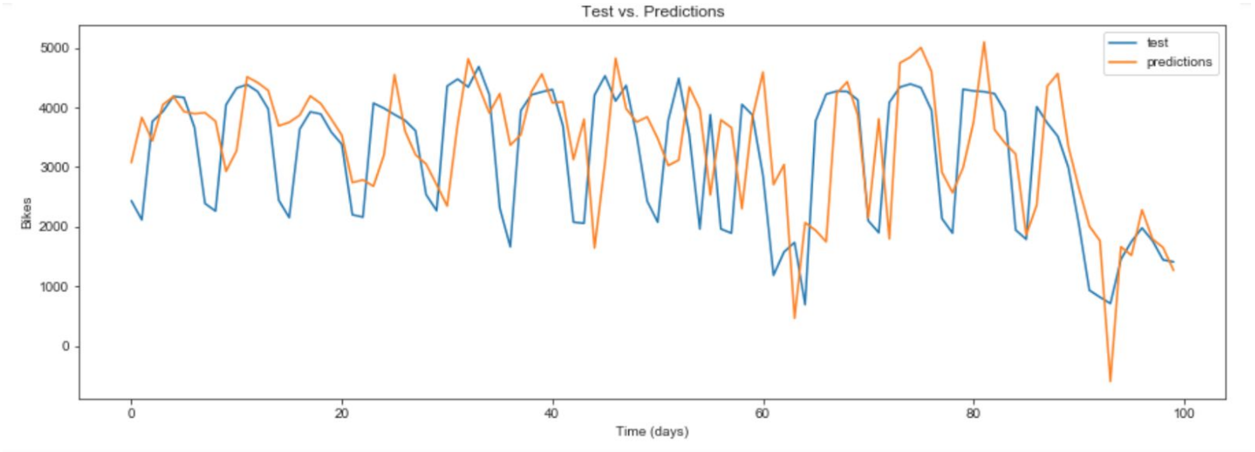

Figure X. LSTM RNN Daily Time Series Forecast

It was shown from the model loss plot that the model suffers from overfitting. This is likely because the model is only using 187 records for training when a daily timestep is used, which is not enough training data for a generalizable model even with hyperparameter tuning and experimentation with neural network model architecture.

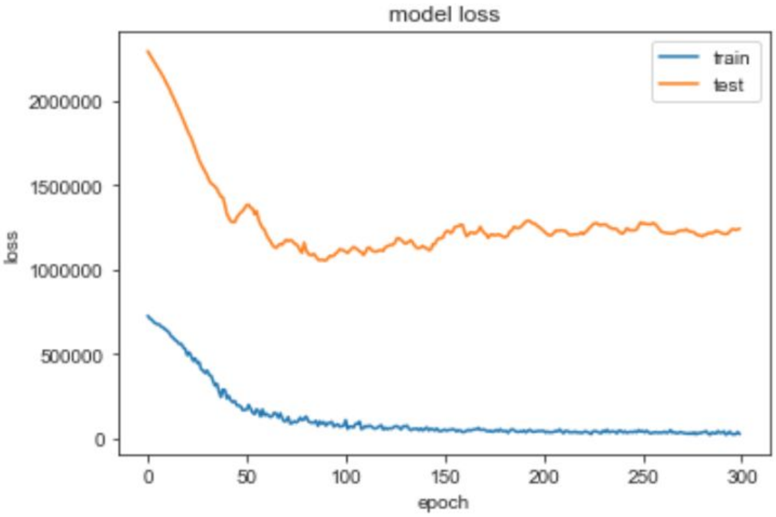

Figure X. LSTM RNN Daily Train vs. Test Loss

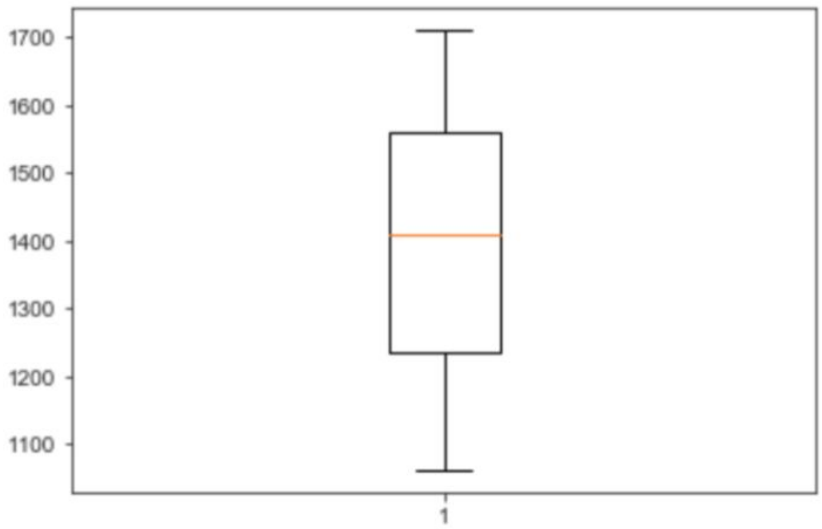

Figure X. LSTM RNN Daily Time Series Loss

The daily forecast model outperformed the persistence model with an RMSE of 1.192 +/- 0.015 compared to the persistence model's RMSE value of 1.766.

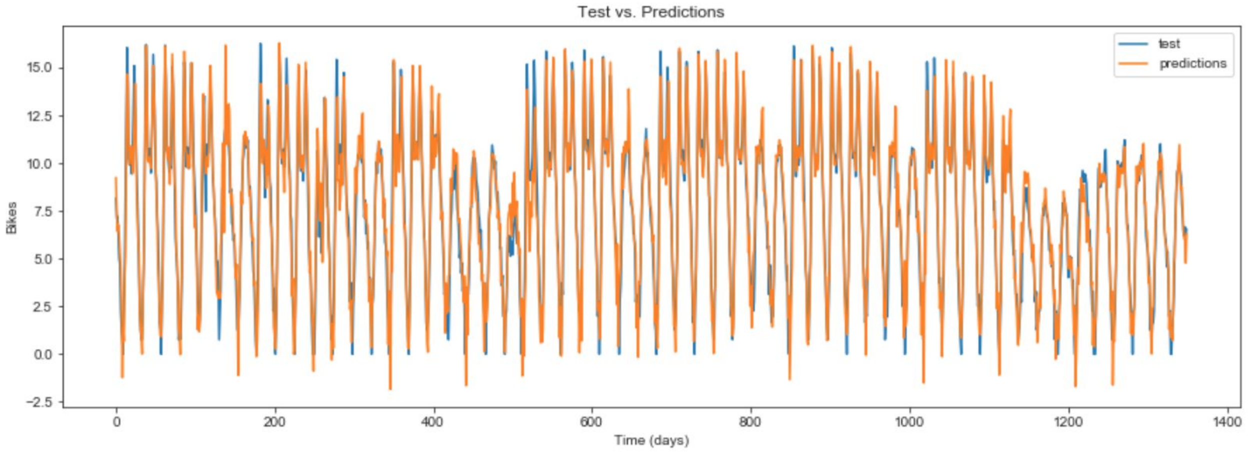

Figure X. LSTM RNN Log Hourly Time Series Forecast

It was shown that both the test and the train losses were both low, with a slight divergence starting at around 30 epochs. The increase in data from Underfitting or overfitting was not an issue in this case, indicating that the model was properly tuned.

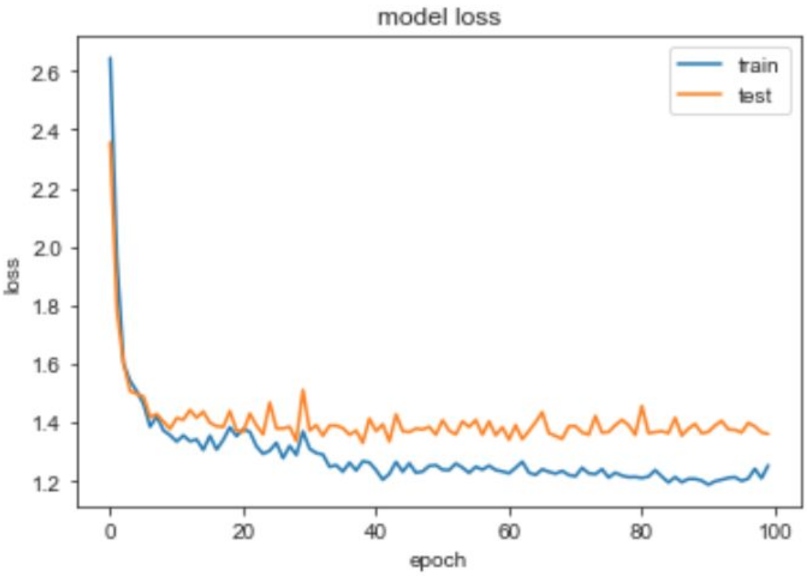

Figure X. LSTM RNN Log Hourly Time Series Train vs. Test Loss

Since LSTM RNNs are inherently non-deterministic, i.e. stochastic, the model was run at least ten times and the RMSE values were summarized using a box and whiskers plot.

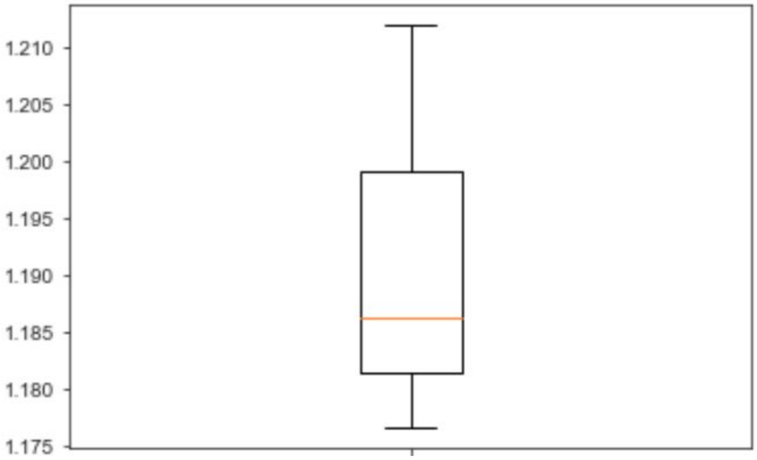

Figure X. LSTM RNN Log Hourly Time Series Loss

5. Conclusions

a. Limitations

b. Future Work

c. Major Findings and Client Recommendations

d. Acknowledgments

The author would like to thanks Springboard and especially his mentor for the advice and support throughout the capstone project.