Springboard Capstone Project 2

Ford GoBike Bike Ride Forecasting

By David Tse

November 11th, 2018

Table of Contents

Table of Figures

Executive Summary

Forecasting overall bike rider demands to balance with bike station supply is one of many important metrics for a successful bike sharing system. 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). Both linear and nonlinear models were used to forecast bike rides: an Autoregressive Integrated Moving Average (ARIMA) model and a Long Short-Term Memory (LSTM) Recurrent Neural Network (RNN). The models were assessed using walk-forward validation (rolling forecast) that outputs the model's RMSE as a metric for evaluation. ARIMA outperformed the persistence model by ~20% for the daily dataset, and by ~60% for the hourly dataset, while the LSTM RNN was parity with the persistence model for the daily dataset, and outperformed the persistence model by ~30%. For the immediate future, it is recommended to use ARIMA models for univariate time series forecasts of aggregate bike rides, but an LSTM RNN would be beneficial at a later point when time series data is more abundant.

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. The demand forecasting problem is framed as a univariate 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). The problem may be of interest to:

- **Ford GoBike** may use this knowledge of demand to guide their supply-related decisions regarding the location and number of docking stations.
- **Bay Area Urban Planners** could develop improved transportation strategies for cities that minimize congestion during rush hour leading to improved work productivity.

2. Data Wrangling

a. Cleaning Steps

The data was downloaded as of November 5th, 2018 from Ford GoBike's public dataset. The .csv data contained a list of bike rides at a particular time and its associated data with 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 was required to train machine learning models.

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.

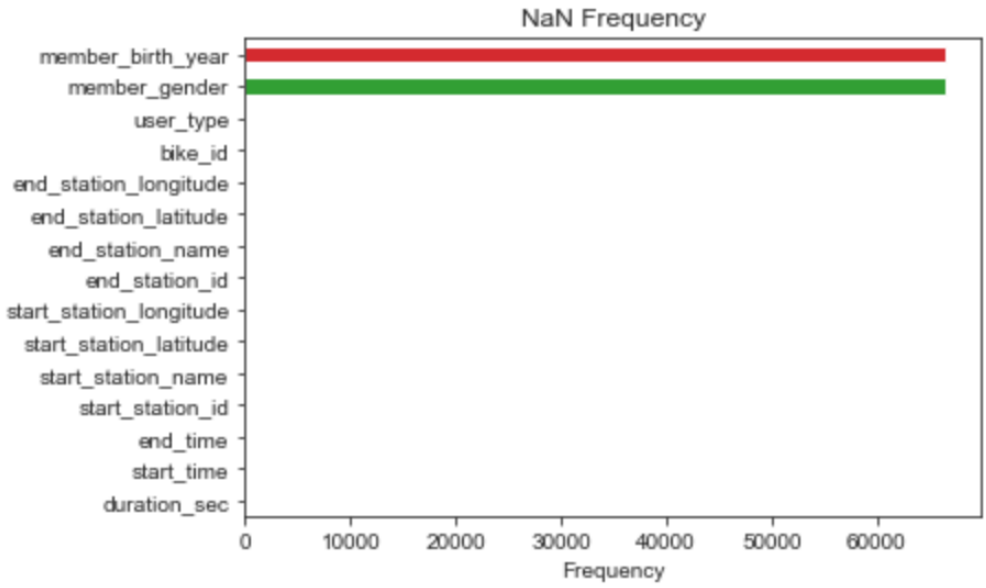

Figure 1. Missing Values

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.

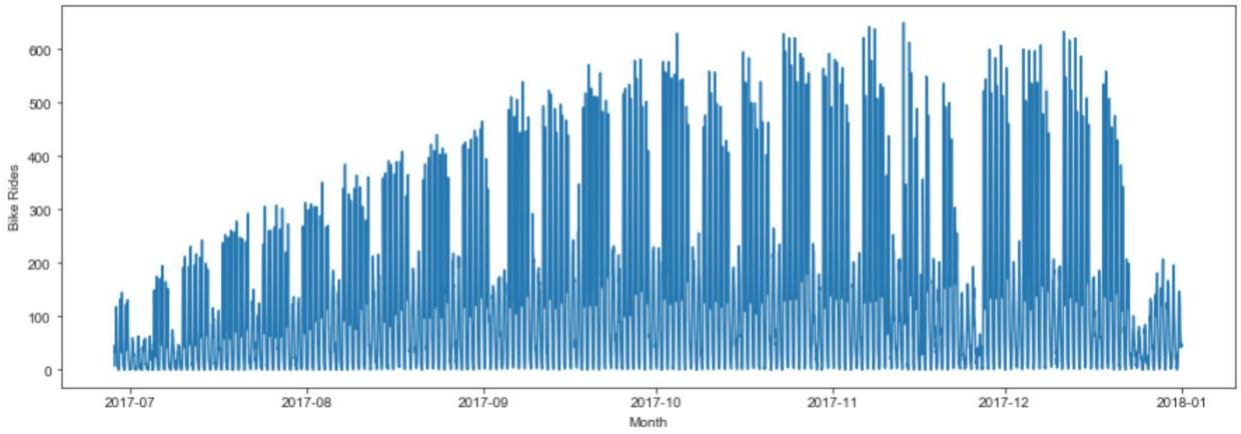

Figure 2. Bikes Rides by Hour

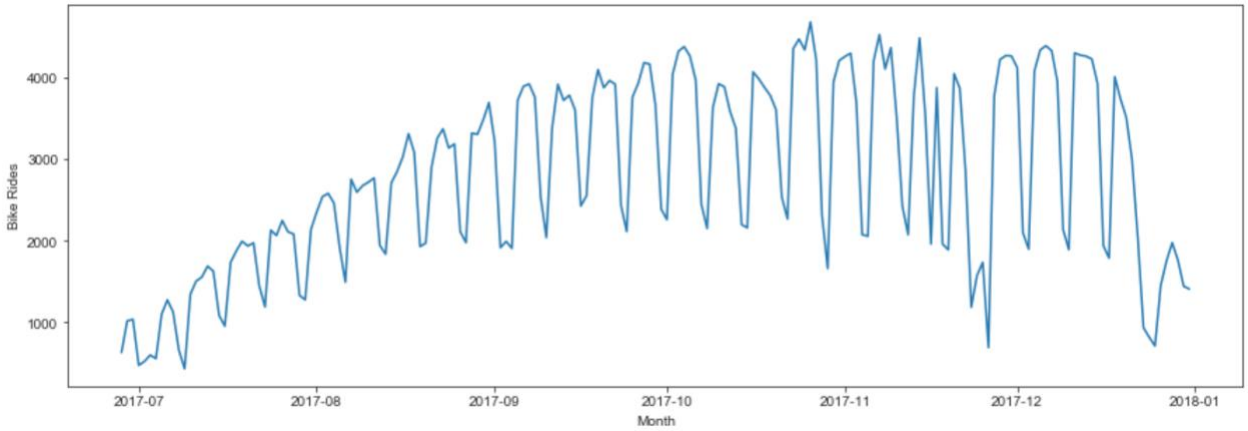

Figure 3. Bikes Rides by Day

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

- A general increasing trend until the last month, where there was a dip
- The spread seems to be increasing with time
- No outliers are present

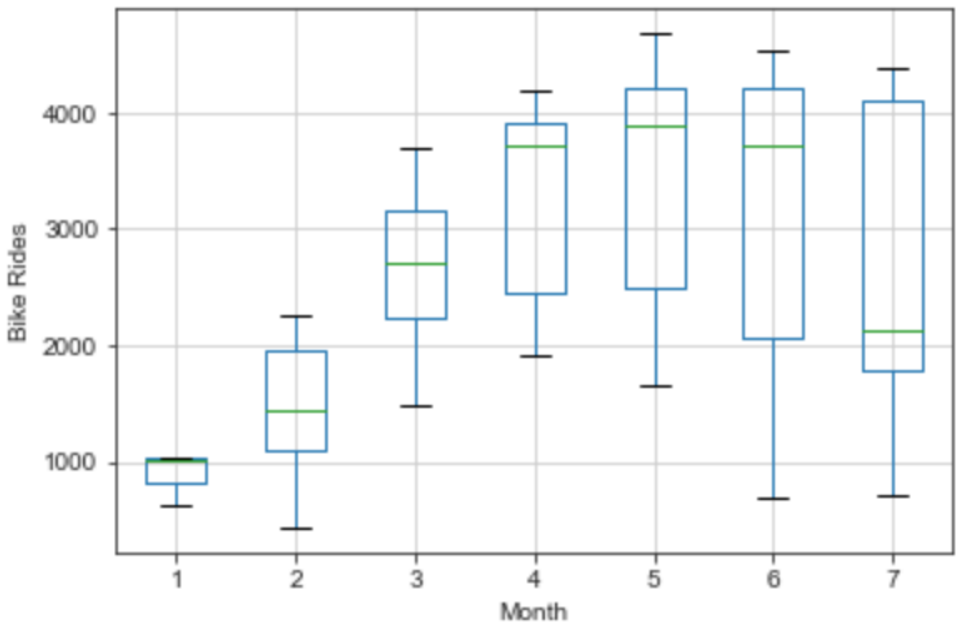

Figure 4. Bike Rides by Month

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?** It was observed that the daily distribution showed bimodal behavior, while the hourly distribution showed a right-skewed distribution with a large spike between 0-50 bike rides. The hourly distribution was then log transformed to obtain a distribution that is closer to a normal distribution.

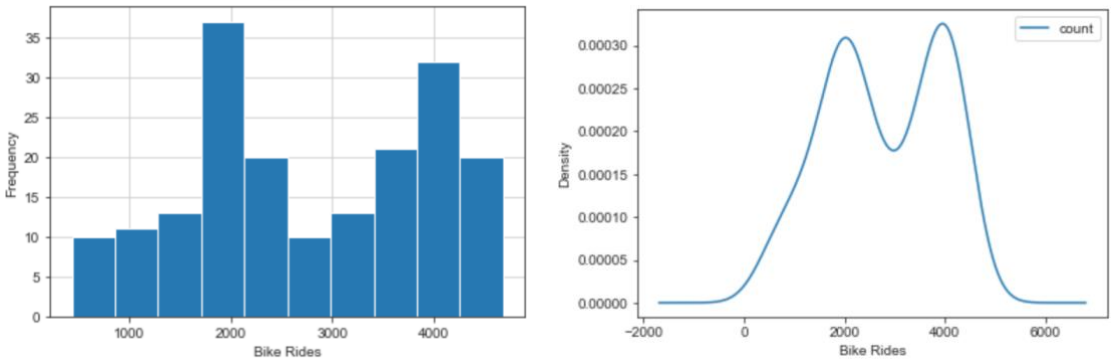

Figure 5. Daily Bike Ride Distribution

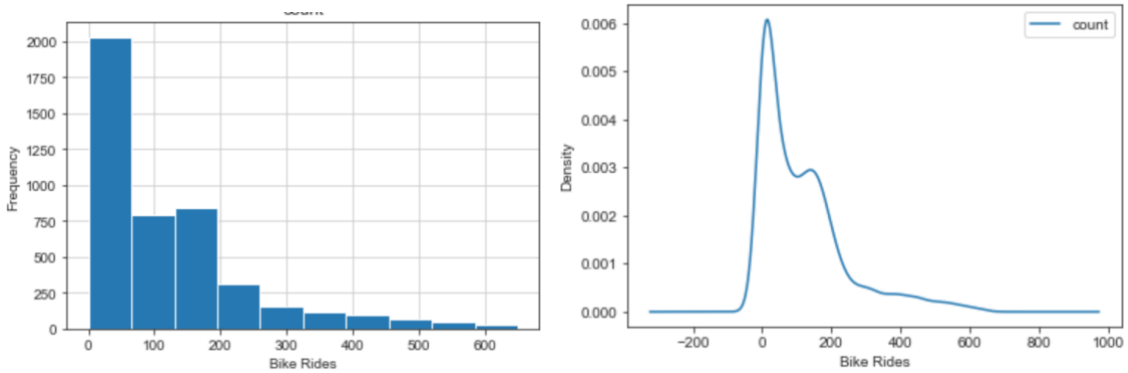

Figure 6. Hourly Bike Ride Distribution

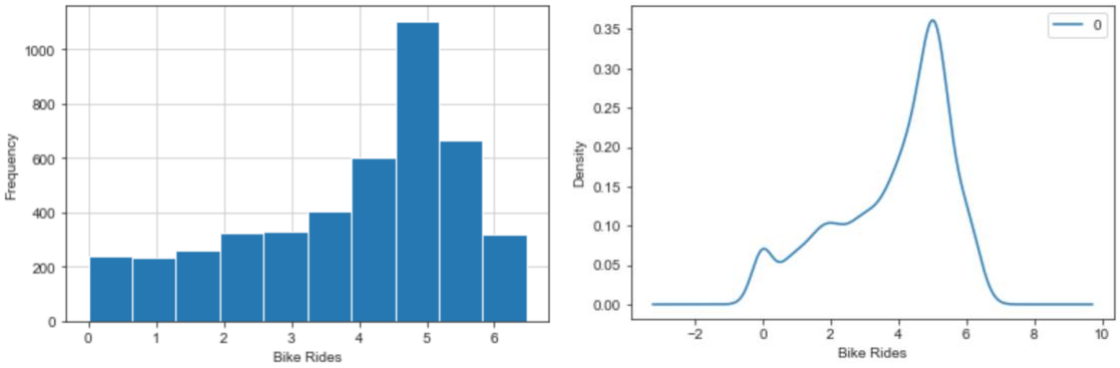

Figure 7. Log Transformed Hourly Bike Ride Distribution

- **How is the time series correlated with previous time lags?** Points tighter to 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.

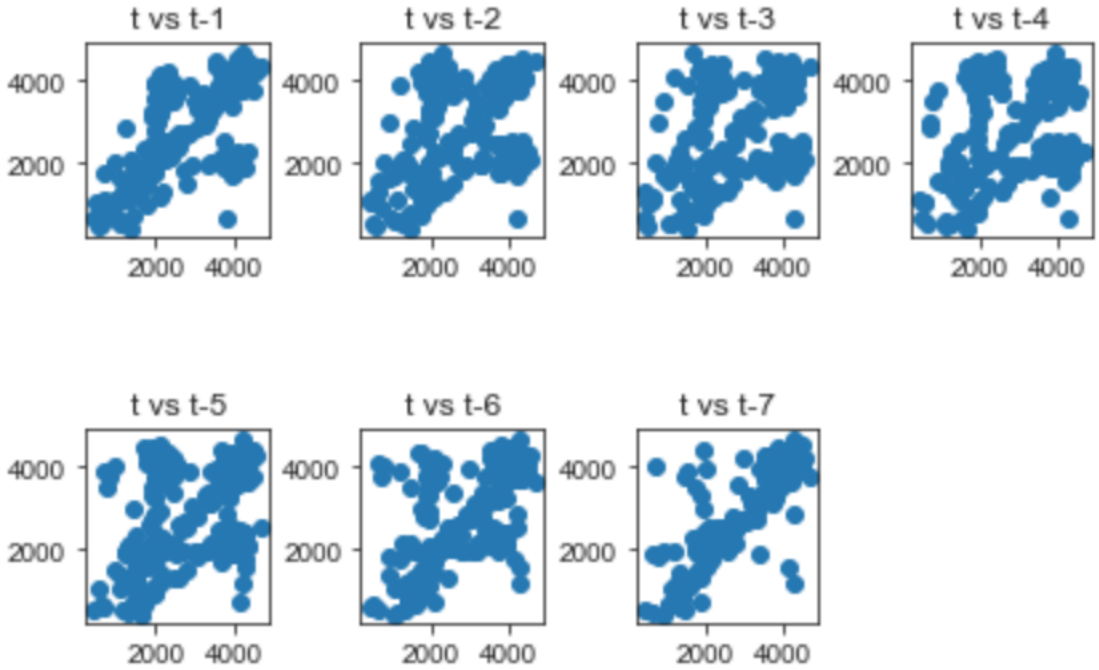

Figure 8. Autocorrelation Plots with Increasing Lag Values

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

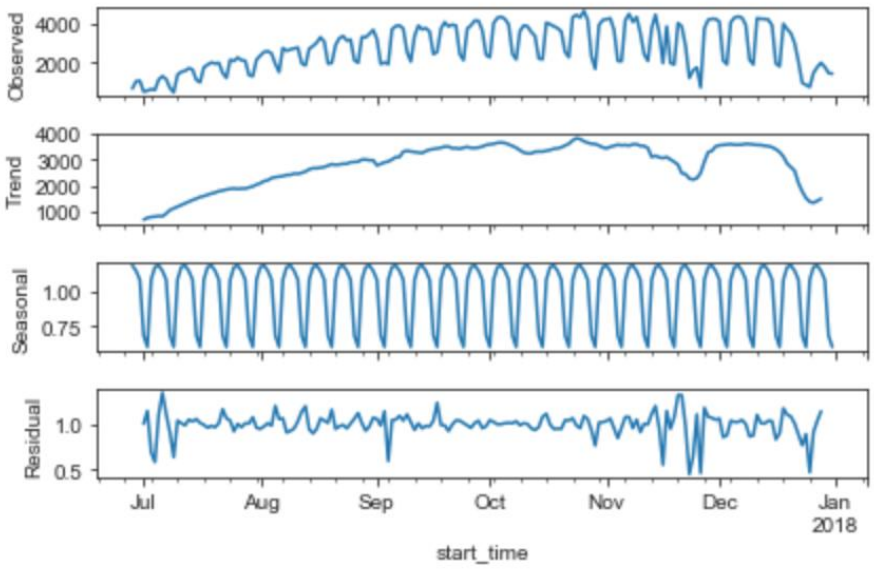

Figure 9. Time Series Decomposition

- **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 10. Start_time station location heatmap

b. Statistical Analyses

A stationary time series is one where the values are not a function of time. Stationarity is an important assumption needed for most linear time series forecasting methods like Autoregressive Integrated Moving Average (ARIMA) Models 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.

The daily dataset was differenced by 3 days. It was shown that the augmented Dickey-Fuller test statistic is -3.824, 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 = -3.824
- P-value = 0.056609
- Critical Values:
    - 1%: -3.469
    - 5%: -2.879
    - 10%: -2.576

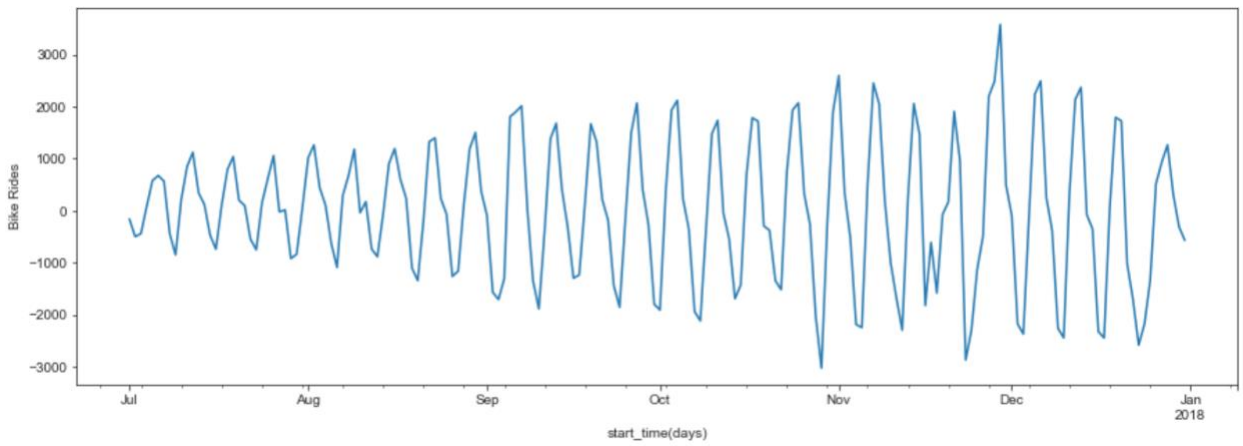

Figure 11. Two-Day Differencing of Daily Bike Rides

The hourly dataset was differenced by 24 hours. It was shown that the augmented Dickey-Fuller 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

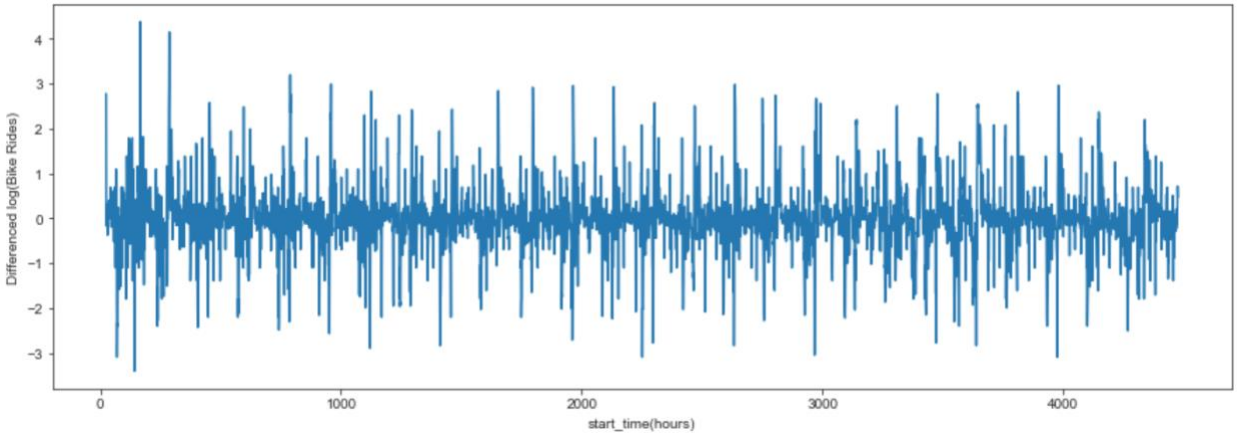

Figure 12. 24-Hour Differencing of Log Transformed Daily Bike Rides

Autocorrelation function (ACF) and partial autocorrelation function (PACF) were calculated using the hourly and daily dataset differenced and non-differenced. 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 61-day time step.

- 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 98-day time step.

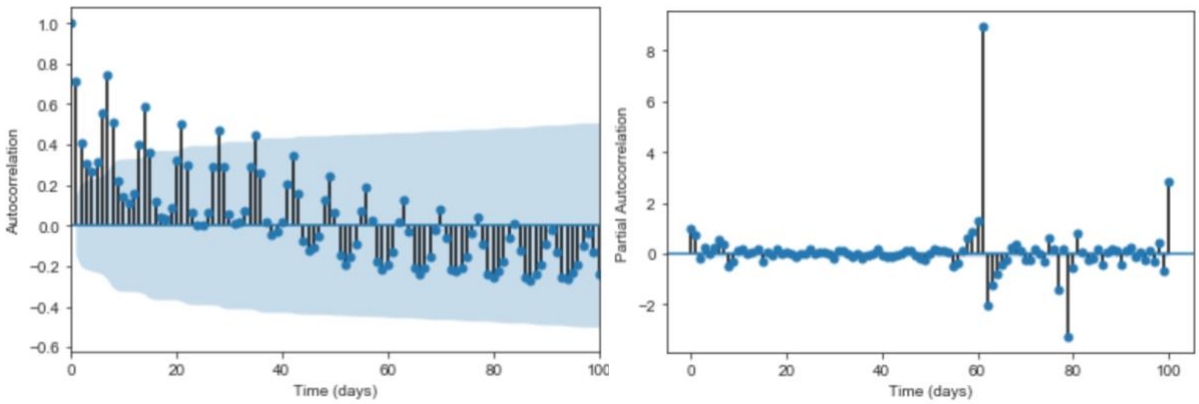

Figure 13. Daily ACF and PACF Plots

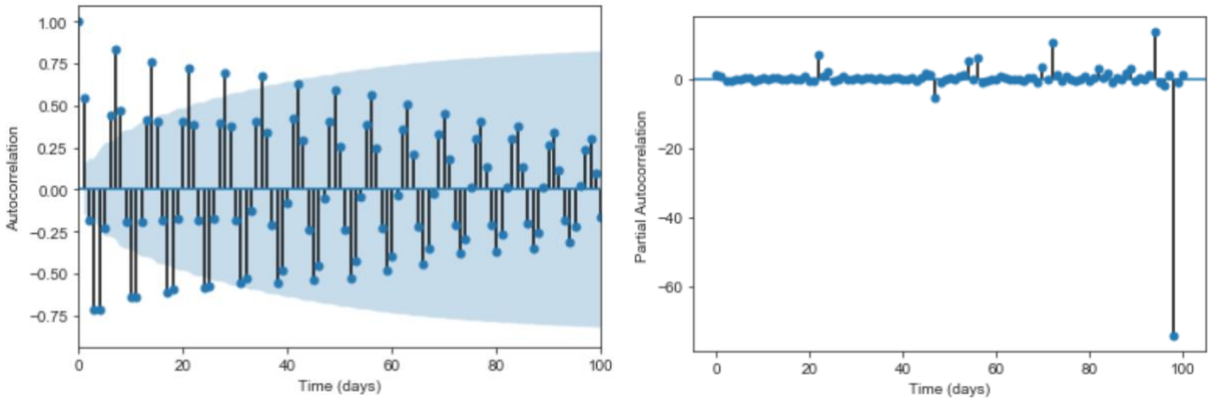

Figure 14. Daily ACF and PACF Plots w/ 3-Day Differencing

For the hourly dataset, 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.

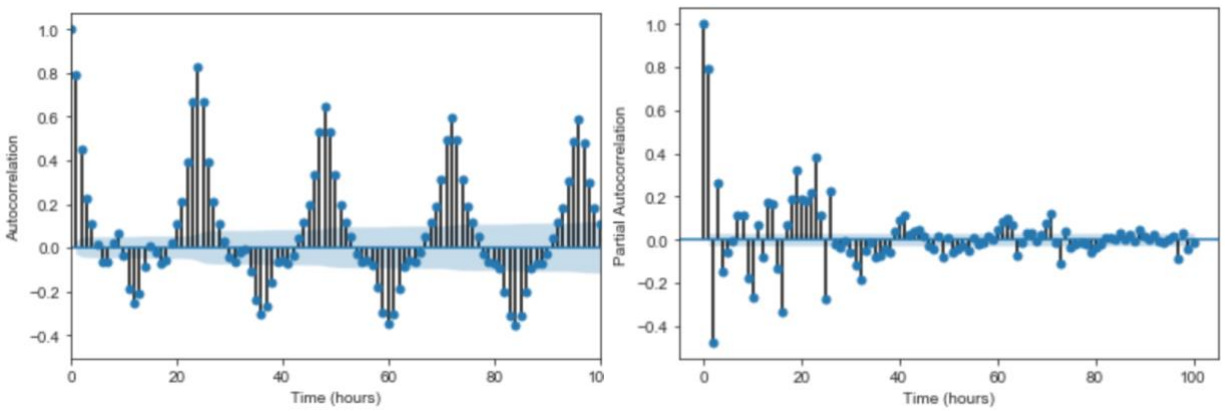

Figure 15. Hourly ACF and PACF Plots

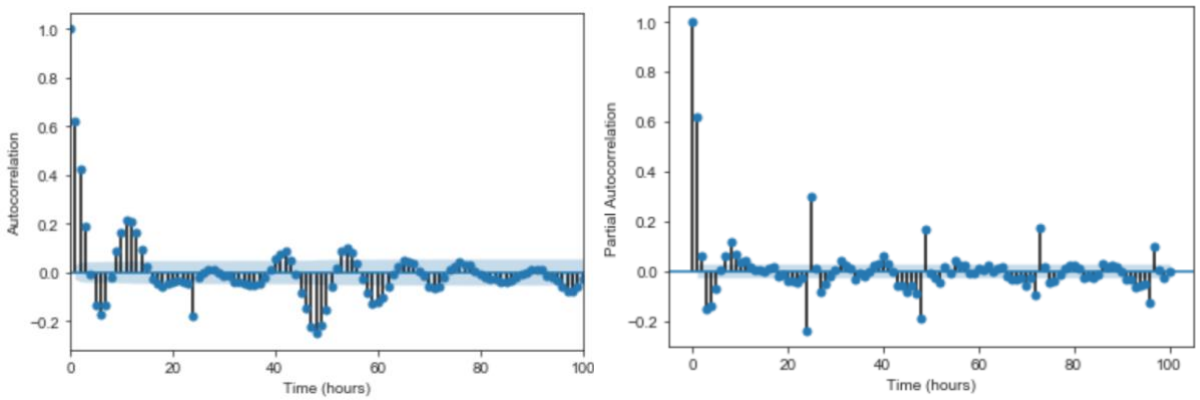

Figure 16. Hourly ACF and PACF Plots w/ 24-Hour Differencing

4. Machine Learning

The time series forecasting problem was framed as a supervised learning problem. Both linear and nonlinear models were used to develop predictions of both hourly and daily down-sampled 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 and amount of training data. 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.

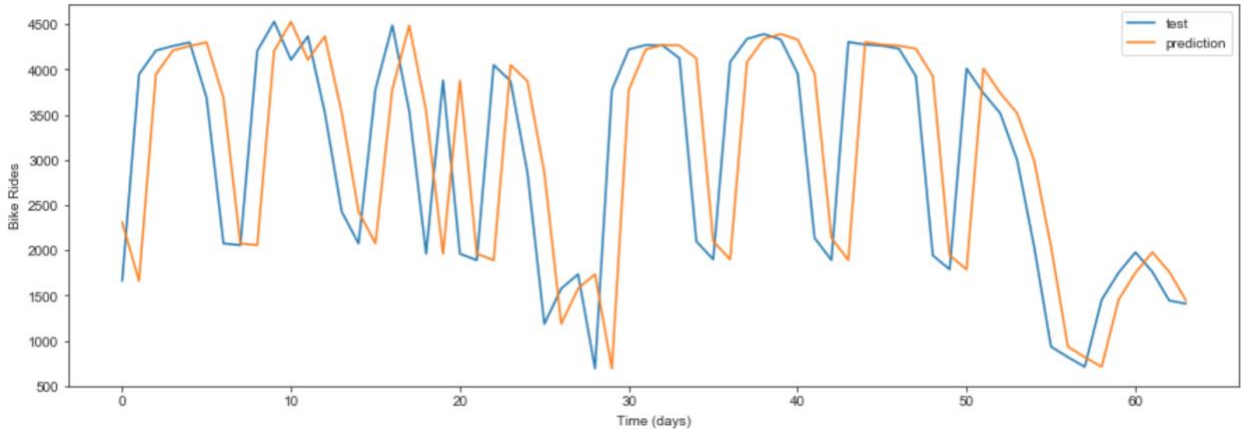

Figure 17. Persistence Daily Time Series Forecast

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

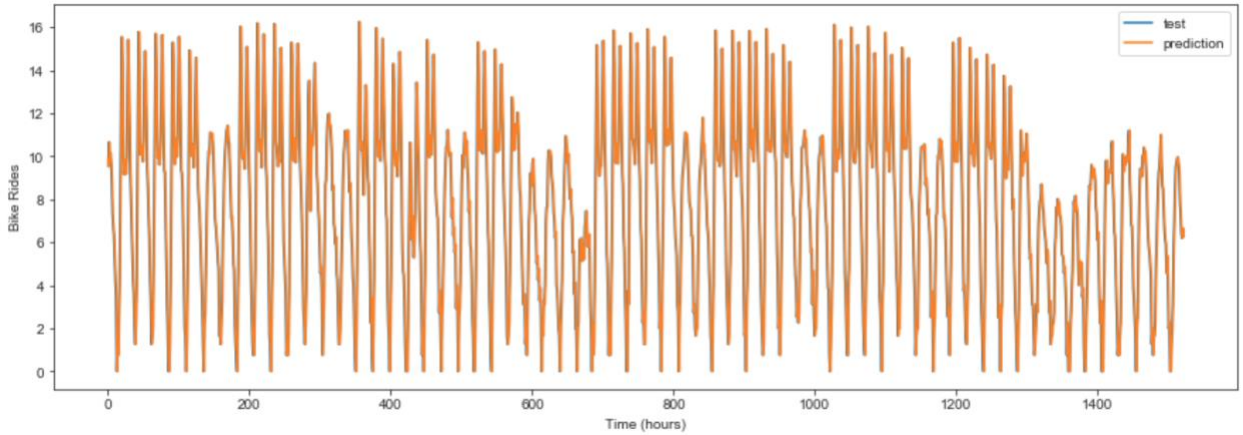

Figure 18. Persistence Hourly Time Series Forecast

b. Non-Seasonal ARIMA Model

A non-seasonal ARIMA model is 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 880. The residuals visually look normally distributed and centered at zero after removing a bias term. This observation was further supported using a Q-Q plot; residuals are centered, but there are deviations near the zero quartiles.

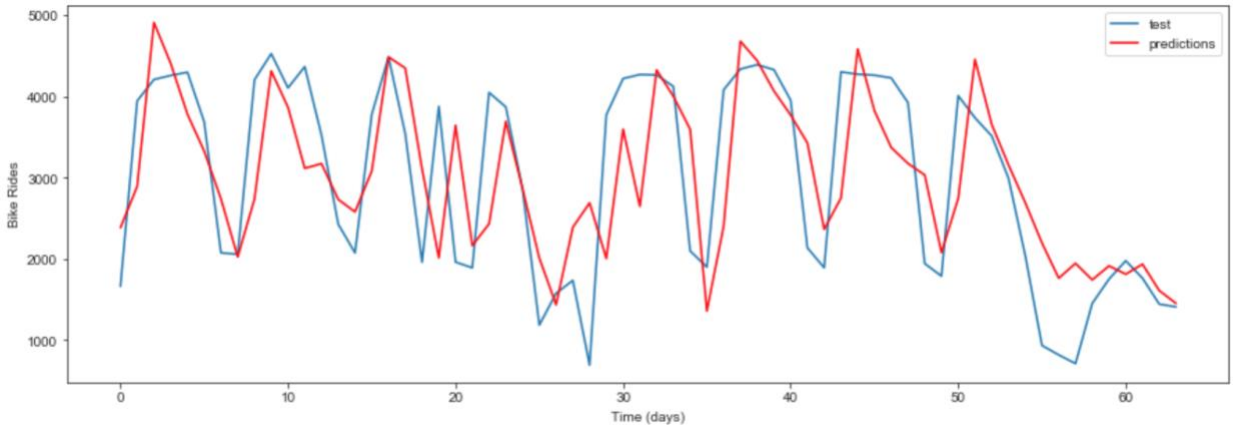

Figure 19. ARIMA Daily Time Series Forecast

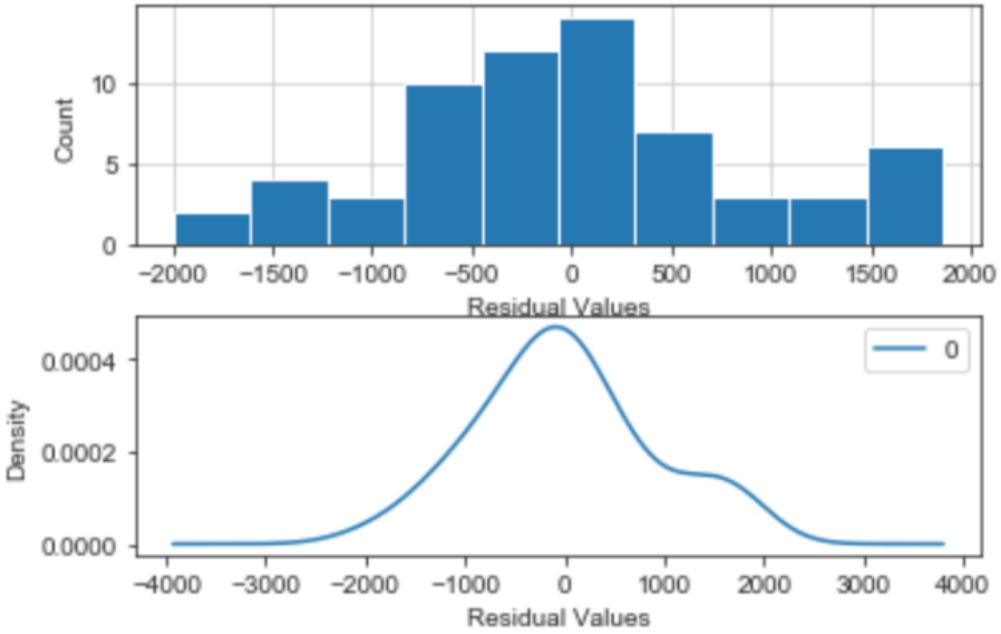

Figure 20. ARIMA Daily Time Series Residuals Distribution

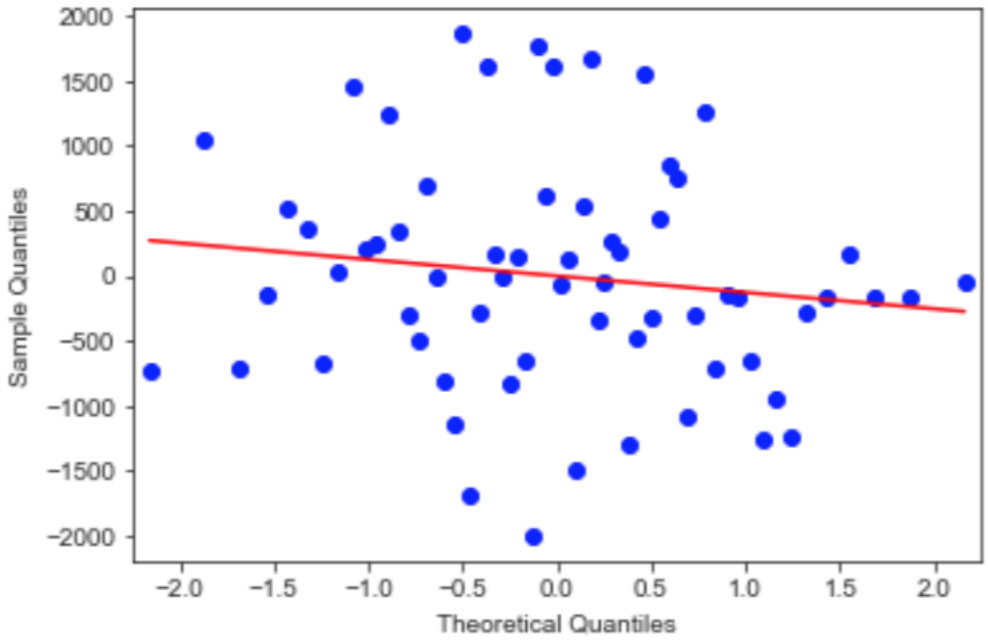

Figure 21. ARIMA Daily Time Series QQ Plot

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 was further supported using a Q-Q plot; residuals are centered, but there are deviations near the zero quartiles indicating residuals with high kurtosis.

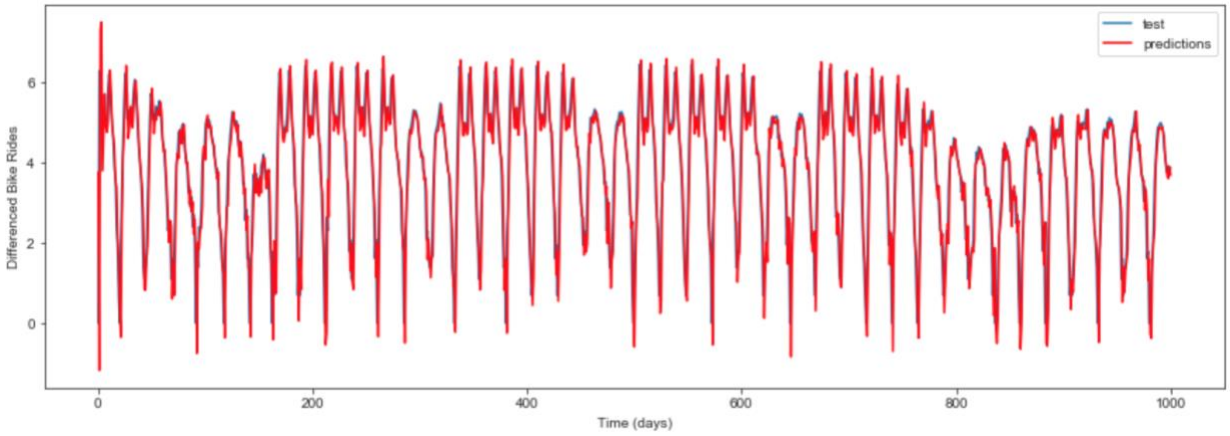

Figure 22. ARIMA Hourly Time Series Forecast

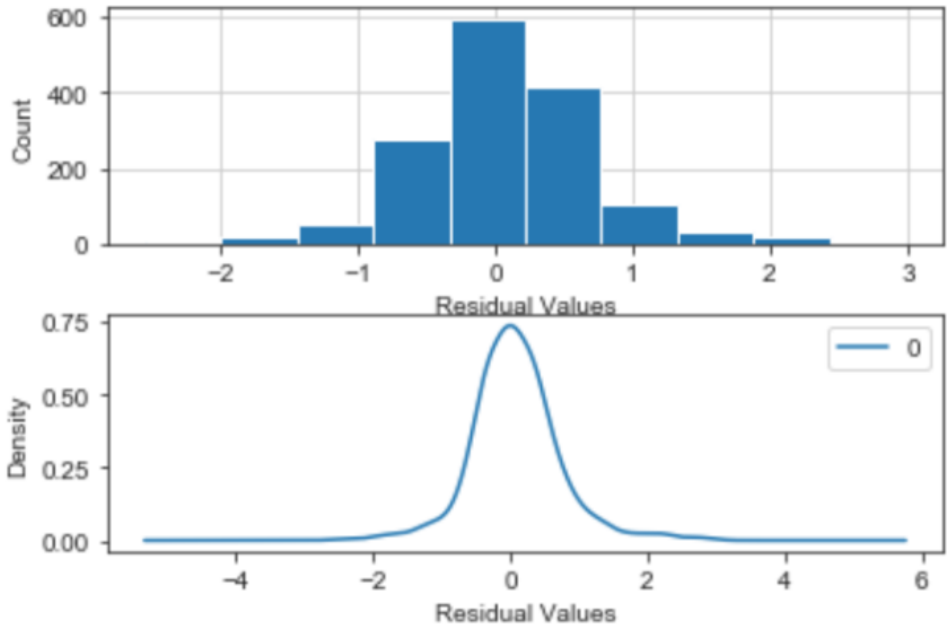

Figure 23. ARIMA Hourly Time Series Residuals Distribution

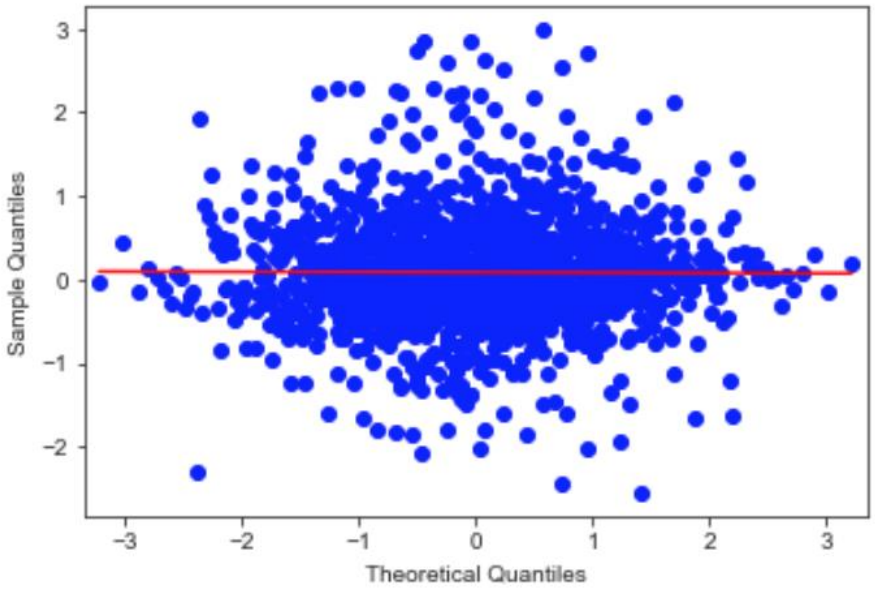

Figure 24. ARIMA Hourly Time Series QQ Plot

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 is skilled at 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.

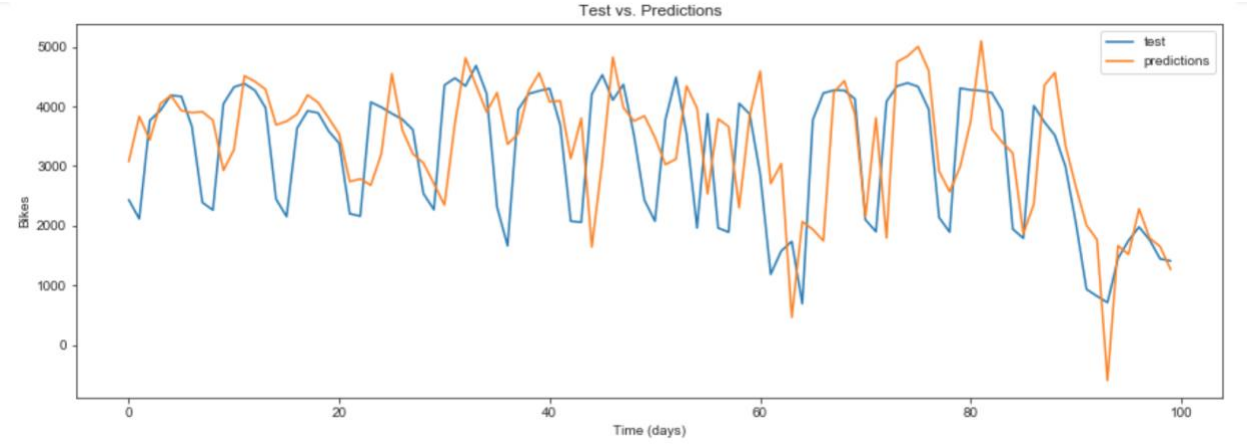

Figure 25. LSTM RNN Daily Time Series Forecast

It was shown from the model loss plot that the model suffers from overfitting. This was probably because the model was trained on 187 records for the daily dataset, which was not enough training data for a generalizable model even with hyperparameter tuning and experimentation with neural network model architecture.

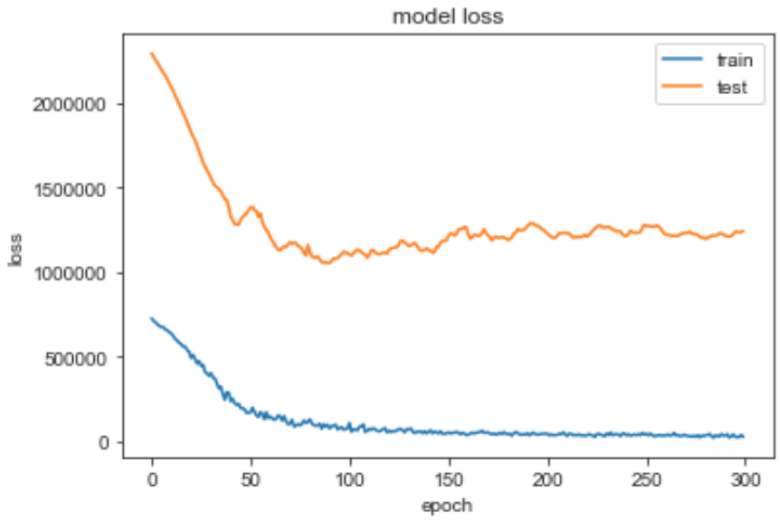

Figure 26. LSTM RNN Daily Train vs. Test Loss

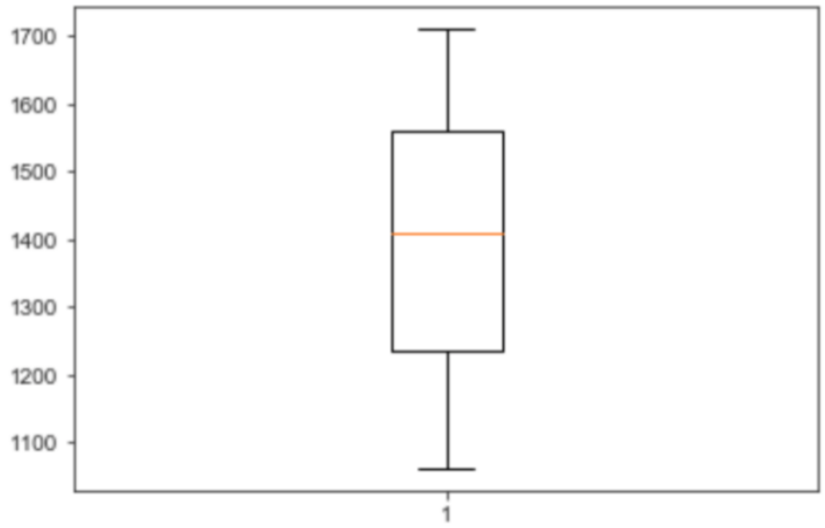

Figure 27. LSTM RNN Daily Time Series Loss

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.

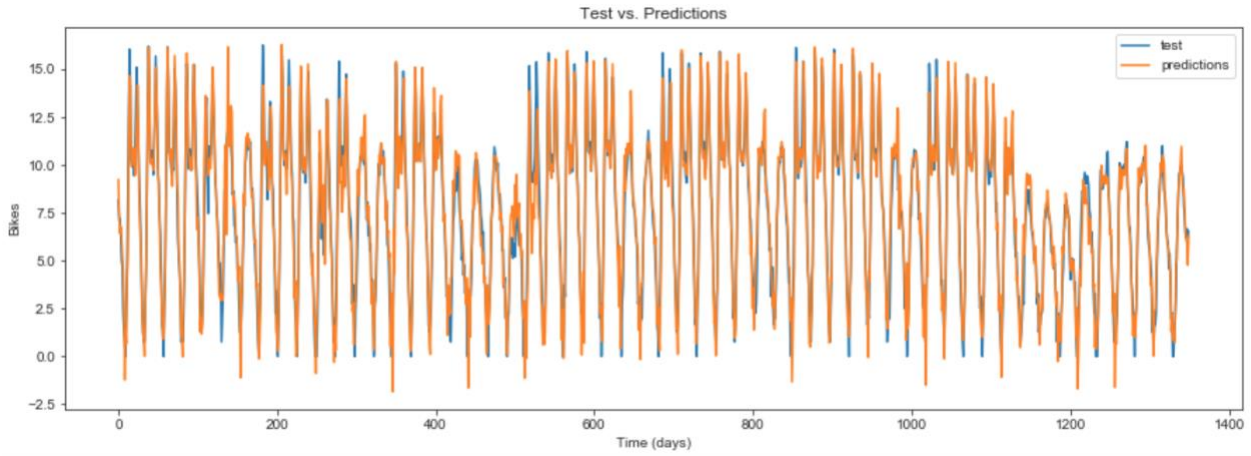

Figure 28. LSTM RNN Log Hourly Time Series Forecast

It was shown that both the test and the train losses were both low, with a slight divergence starting at around 30 epochs. Underfitting or overfitting was not an issue in this case, indicating that the model was properly tuned and had a minimum level of observations used for training.

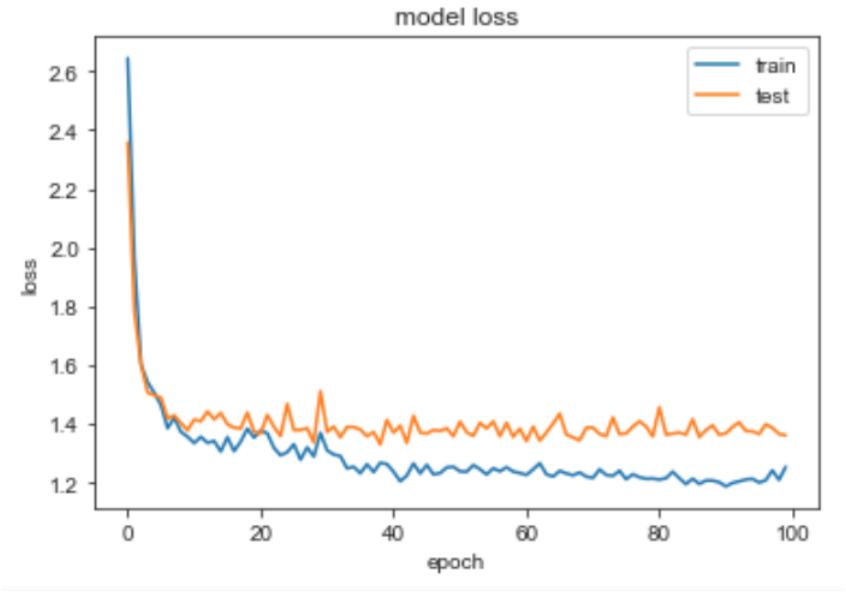

Figure 29. LSTM RNN Log Hourly Time Series Train vs. Test Loss

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.

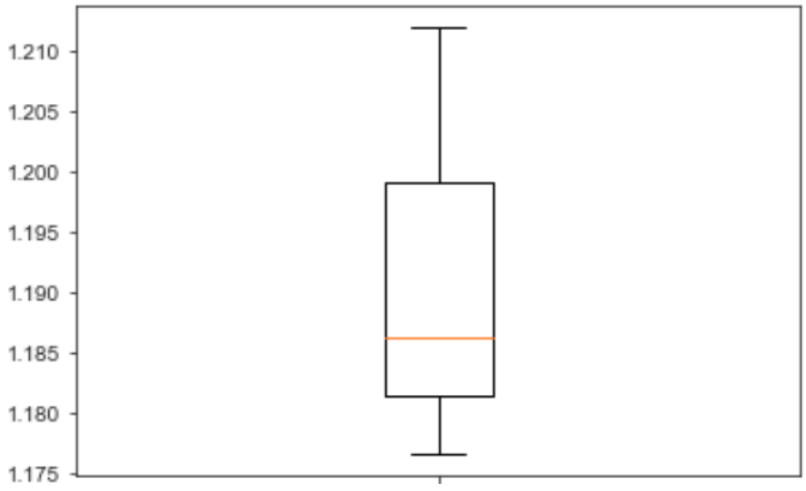

Figure 30. LSTM RNN Log Hourly Time Series Loss

5. Conclusions

a. Limitations

- The forecast is based on an aggregate view of bike rides and is not meant to be used for bike station rebalancing.
- The forecast is relevant for bikes that are docked at bike stations, but may not be relevant for dockless electric scooters and bikes.
- The dataset only provides a roughly a half year worth of data, LSTM RNN models may not be able to obtain trends with long time dependencies.

b. Future Work

- Investigate multivariate forecasting to determine how other time-dependent variables affect the output, e.g. weather, air quality.
- Use additional data from 2018 to further improve the LSTM model

c. Major Findings and Client Recommendations

- **Major Findings:**
    - The ARIMA models outperformed the persistence model and the LSTM model for both time the hourly and daily time series
    - The LSTM RNN models were able to outperform the persistence model at the hourly timestep, but was parity at the daily timestep.
- **Client Recommendations:**
    - Ford GoBike: Use the ARIMA model for time series forecasting, but as the company operates for a longer period of time consider LSTM RNNs as an alternative.
    - Bay Area Urban Planners:
        - Focus planning efforts near the downtown SF area, followed by Downtown Oakland, then Berkeley for docked bike stations
        - Use the ARIMA model for time series forecasting of new bike sharing companies like Ford GoBike, but as a company operates for a longer period of time consider LSTM RNNs as an alternative for demand forecasting.

d. Acknowledgments

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

Appendix

Summary Statistics:

	duration_sec	start_station_latitude	start_station_longitude	end_station_latitude	end_station_longitude	member_birth_year	member_age
count	519700	519700	519700	519700	519700	453159	453159
mean	1099.009521	37.771653	-122.363927	37.771844	-122.363236	1980.404787	37.595213
std	3444.146451	0.086305	0.105573	0.086224	0.105122	10.513488	10.513488
min	61	37.317298	-122.444293	37.317298	-122.444293	1886	19
25%	382	37.773492	-122.411726	37.77452	-122.410345	1974	30
50%	596	37.783521	-122.39887	37.78383	-122.398525	1983	35
75%	938	37.795392	-122.391034	37.795392	-122.391034	1988	44
max	86369	37.880222	-121.874119	37.880222	-121.874119	1999	132