**Time Series Analysis of TTC Delays**

**Don Nguyen**

**12/11/2019**

**Table of Contents**

[**Problem Statement** 3](#_Toc26927578)

[**Data Cleaning and Wrangling** 3](#_Toc26927579)

[**Initial findings from exploratory analysis** 4](#_Toc26927580)

[**Modeling** 6](#_Toc26927581)

[**Discussion** 7](#_Toc26927582)

[**Conclusion** 8](#_Toc26927583)

# **Problem Statement**

As the city of Toronto is growing, so is the population which will affect how people get around the city. One of the best ways to get around the city is the public transportation system, especially the subways. Unfortunately, subway delays are common and can ruin anyone’s day. The goal of this project would be to forecast the pattern of delays over the last 5 years and develop insight on how to respond.

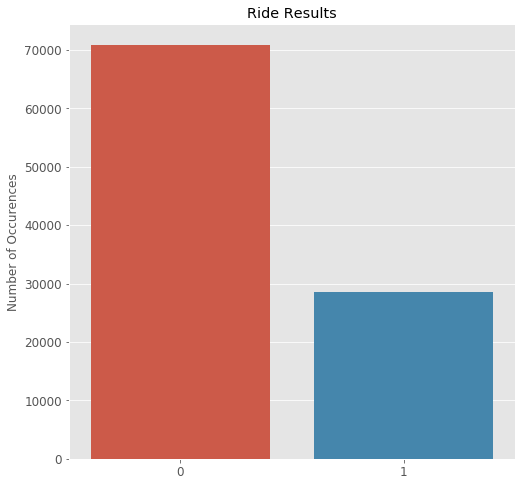
The client in this scenario would be the local government and Toronto Transit Commission (TTC). It would be in their best interest to identify when to address inefficiencies in the public transportation system to retain ridership numbers and reduce the number of cars on the road. This could lead to overcrowded highways and congestion on roads which could also lead to accidents.

# **Data Cleaning and Wrangling**

For this analysis, data from Toronto Open Data was collected in the form of multiple .csv files spanning over 5 years. During the process of reading in the data, it was found that there was very little missing data but there were inconsistencies with several of the values of features. In the case of missing data, it was viable to drop the rows due to missing data in so many other features.

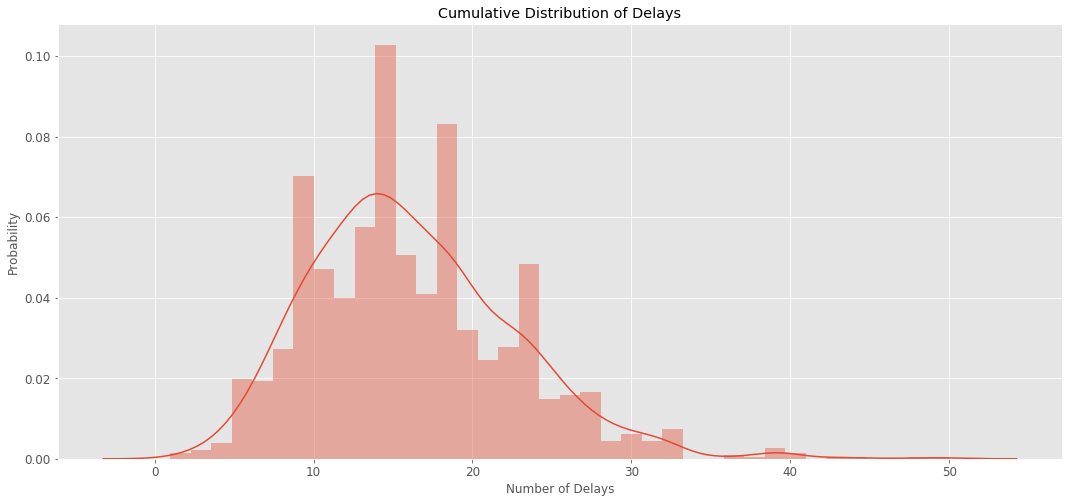
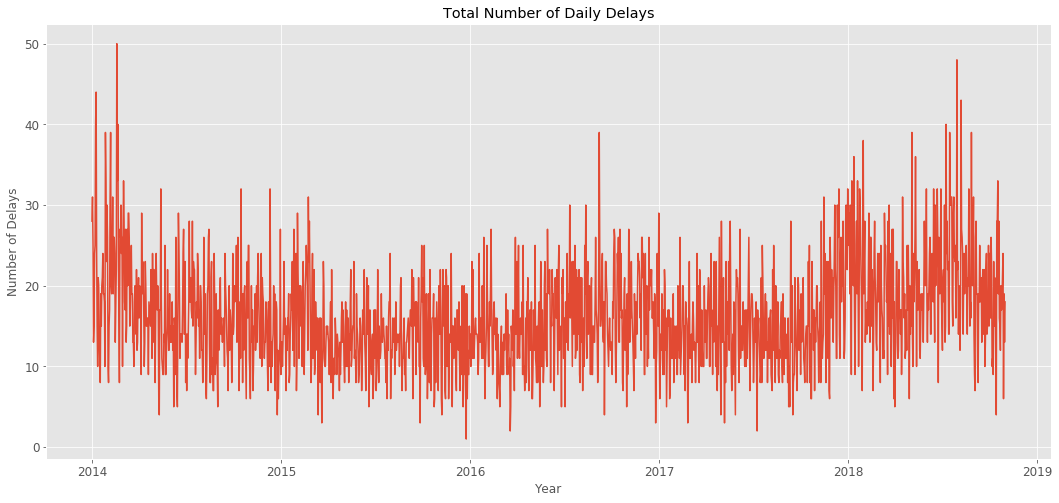
The features of interest for time series forecasting were limited to the date and number of delays in a day so not much cleaning and feature selection was required.

# **Initial findings from exploratory analysis**

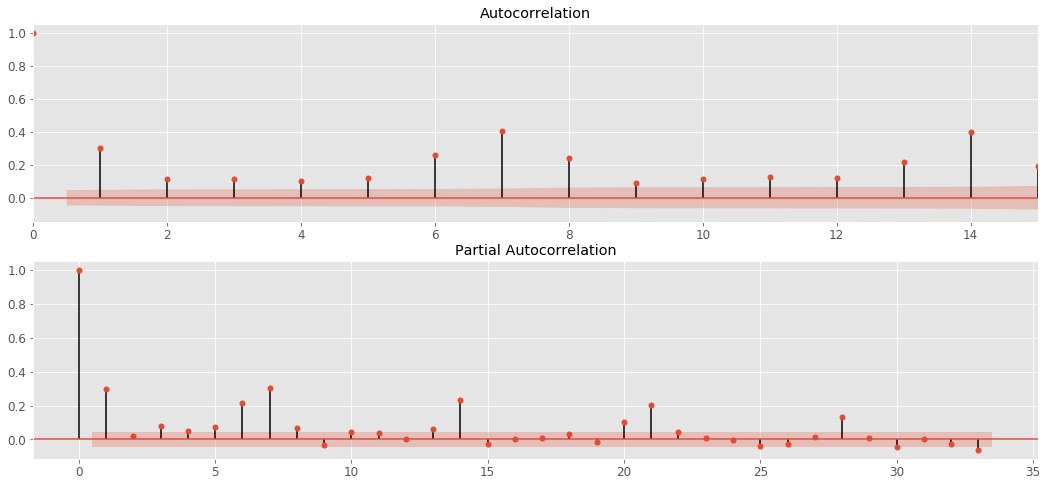


Initial visualizations of delay data reveals that the response variable is balanced enough and that delays make up about thirty percent of total rides according to figure 1. This looks great in comparison to New York’s subway trains which saw a 92% delay rate during morning rush hours in 2018 according to NYpost.

*Figure 1. A bar plot showing the ratio of delays (1) and non-delays (0)*

It is also observed that the response variable has a normal distribution with a mean around 15 in figure 2, which will allow for statistical tests to be performed later on. Figure 3 shows a plot of the number of daily delays occurring over the 5 years. With the help of the Dickey-Fuller test, it is confirmed that the data is stationary which is important for any model to work in the proceeding steps. It is worth mentioning that a threshold value of 5% was chosen when performing this test and the p-value came close to exceeding that threshold. This would mean that the null hypothesis that the data is not stationary would not be rejected. To be safe, a one order differencing value was considered in the modeling phase.

*Figure 2. A plot showing the distribution of delays that happen over 5 years*



*Figure 4. Autocorrelation and Partial Autocorrelation plots revealing that the clear existence of autocorrelation and seasonality*

*Figure 3. A plot showing the daily delay amount over about 5 years*

After plotting both the Autocorrelation, and partial autocorrelation functions, it is clear to see that there is a correlation between the current day and any future date within 14 days. It also shows the existence of some seasonality every 7 days which corresponds to the average work week for people commuting to work using the subway. In order for accurate forecasting to happen, the correlation will have to be removed.

# **Modeling**

In this phase the ARIMA and Facebook’s Prophet models will be compared to see which one has the best performance. The success metric will be the root mean squared error for all three models for consistency purposes. Additionally, a baseline model using a basic walk-forward method will be added with the goal of outperforming it with the two models. The walk-forward method is a forecasting technique that involves repeatedly making predictions one day away by taking in any previous predictions to train the model. The baseline and ARIMA method will be using this technique for their predictions.

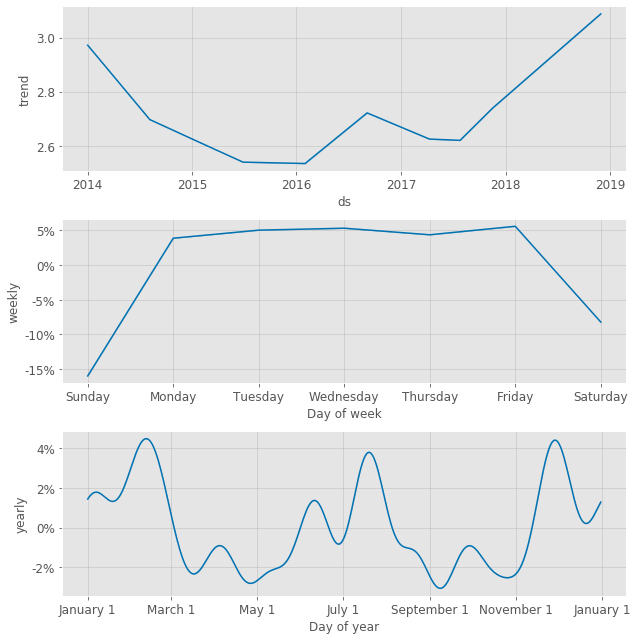
Using hyper parameter tuning the parameter values p, d, and q for the ARIMA model were optimized to find the best performing model. Unfortunately, with the size of the dataset and the number of combinations of parameter values, only 20% of the models were tested in the span of optimizing for 2 consecutive days. As a compromise, the current best performing model was taken as the final result for the ARIMA model. The corresponding parameter values for p, d, and q were 2, 1, and 5 respectively and had an RMSE score of 5.96. This is a vast improvement from the baseline model which gives some perspective in terms of the model’s performance.

The prophet model was tuned as well but was able to finish within a couple of hours for a large parameter set. It was also able to take in a holiday parameter which accepted a dataframe of dates. Since there are a large variety of special occasions like festivals, cultural and religious celebrations it would make it hard to account for all of them. Instead, only statutory holidays were accounted for during the 5 years of data. The final optimized model resulted in an RMSE score of 5.39 which makes it the overall winner amongst all the models.

|  |  |
| --- | --- |
| Model | RMSE Score |
| Baseline | 8.51 |
| ARIMA | 5.96 |
| Prophet | 5.39 |

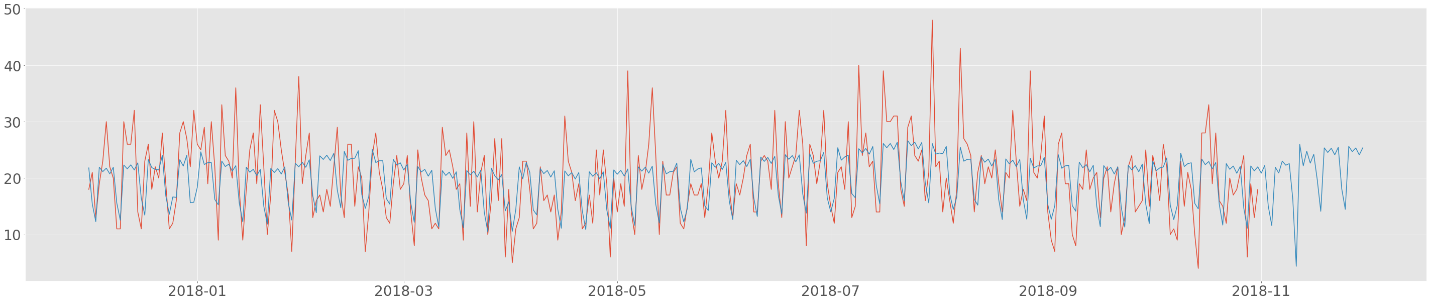
# **Discussion**

The prophet model is able to produce a plot of trends and seasonality as one of the outputs of the model. The weekly seasonality observed in figure 5 agrees with what was deduced from the autocorrelation plot in figure 4. There also seems to be a pattern in the yearly seasonality of peaks during the winter and summer and dips during the spring and fall seasons. Finally, the overall trend of delays seems to be increasing over the past 3 years which is worrying.



*Figure 5. Component output from fb prophet model.*

Figure 6 shows the actual and predicted data plotted on top of each other for 2018. It would imply that the prophet model seems to do a poor job predicting the peaks but does an acceptable job with the dips. This could be attributed to the holiday parameter that was defined with only statutory holidays. In reality there are many events that are not holidays and could cause a major delay on any given day. This includes big sporting events, natural disasters, or power outages.

****

*Figure 6. Actual (red) and predicted (blue) data for 2018 using the prophet model.*

In fact, the second and third worst predictions were days where there a flood+fire (2018-08-07) and wind storm (2018-05-04) occurred.

# **Conclusion**

Facebook’s prophet model outperformed the ARIMA model in this study with an RMSE score of 5.39. It was determined that there is a strong seasonality that clearly effects the amount of delays in a given day. The plots to focus on are the yearly seasonality, and the trend.

In the perspective of the Toronto Transit Commission, the time of year where there is a high seasonality effect is where they should be proactively preparing for a higher delay rate. This could be in the form of more first responders to resolve delays quicker, or more security to prevent conflicts on the subway. One radical suggestion would be to consider AI assisted subway trains that could solve problems related to congestion when a delay is resolved. This has proved to work successfully in places like Japan where trains are rarely late to arrive at their platforms.

According to the trend, the delay rate has continued to rise for the past 3 years with no signs of plateauing of dipping. This is worrying because the population in Toronto continues to increase as well so if the problem with delays are not addressed soon, it might get out of control.