Time Series Analysis & Modeling

DATS 6450, Section 1

Instructor: Dr. Reza Jafari

Term Project

Mariko McDougall

MM

05-05-2021

Table of Contents

Abstract

Introduction

Dataset

Stationarity

Time Series Decomposition

Feature Selection

Basic Models

Holt-Winters

Multiple Linear Regression

ARMA, ARIMA, SARIMA Models

Levenberg Marquardt Algorithm

Diagnostic Analysis

Final Model selection

h-step ahead Predictions

Summary and conclusion

References

Abstract

Much to commuter’s chagrin, traffic is an intensely time-sensitive constant in our daily lives. The volume of traffic can be largely predicted by the time – which can even be seen in expressions such as “rush hour”. In this analysis, we employ a suite of time-series tools, from feature selection to SARIMA models, to derive the best model for predicting hourly traffic density in vehicles/hour on the I-94W highway between Minneapolis and St-Paul.

Introduction

The dataset used for this project is the Metro Interstate Traffic Volume Data Set, detailing the hourly traffic along a the I-94 interstate highway between Minneapolis and St-Paul, in the westbound direction.

The dataset consists of the hourly traffic in number of vehicles, per the Minnesota Department of Transportation, the hourly temperature, amount of rain in mm, amount of snow in mm, a numeric percentage of cloud cover, a contextual description of the weather, in both short and detailed forms, and information if the date was a holiday.

The original dataset contained hourly traffic readings from October 2, 2012 to September 30th, 2018, encompassing 48,204 individual timepoints. However, due to a large gap in the data from August 8th, 2014 to May 11th, 2015 and slightly inconsistent data for the following week We have chosen to exclude the early subset of data from the analysis. Instead, the data used for this project spans from May 24th, 2015 to September 30th, 2018 and contains 28,676 datapoints.

Dataset

Several of the dataset’s original features, including the timestamps, required refinement prior to performing time series analysis.

The original weather descriptions, while likely to be informative for multivariate modeling, were originally provided as string descriptions, which are not ideal for time series models. These were converted into one hot encoding columns by creating a new binary feature column for each unique weather type and converting the string into a 1 in the column where it applied, and a 0 for all other weather encoding columns.

Additionally, to accommodate several weather conditions occurring within the same hour, the original dataset frequently represented the same hour multiple times each with a distinct weather value. The dataset was grouped by the timestamp, and all variables were aggregated such that the one hot encoding columns could contain the values from each of the duplicated times, and the rest of the data were kept at their original values.

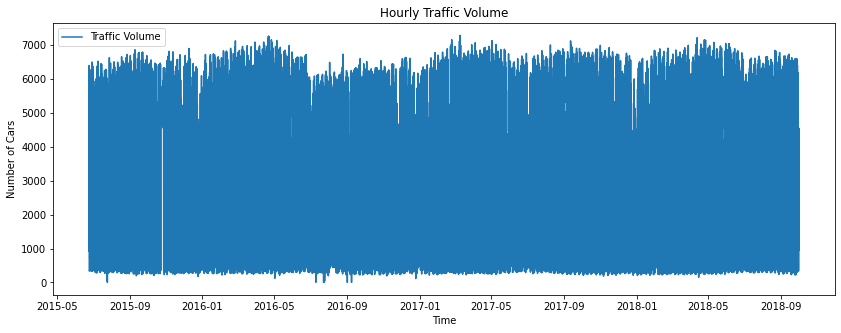
Several of the weather columns were grouped together into weather categories, to provide additional information to the model, with the expectation that a consolidated column may be more informative. An example of such a column would be the ‘Precipitation’ feature, which combined ‘Thunderstorm’, ‘Drizzle, ‘Rain’, and several other similar one hot encoding columns. The original binary columns were maintained in the dataset.

Similarly, the original ‘Holiday’ feature was a string column which explicitly listed which holiday was occurring, if any. While it is possible that the specific holiday may be relevant, it seemed unlikely that there would be a large impact given each annual holiday could occur at most four times in the dataset, given the timespan. Instead, this feature was converted into a binary holiday/not holiday classification.

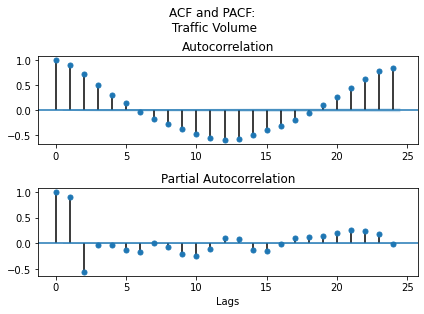
Finally, a boolean weekday/not weekday column was created using the timestamps of the hourly traffic report. This feature was created, as it seemed likely that traffic patterns would be strongly influenced by the day of the week, and thus if commuters would be largely present on the road.

After the feature creation was complete, the data were re-sampled to fill in any gaps in the timestamp data. In the case that any hourly datapoints were missing, the data from the previous hour was forward-filled into the missing values. If multiple hours in a row were missing, the data from the most recent point forward-filled until an existing timepoint was found.

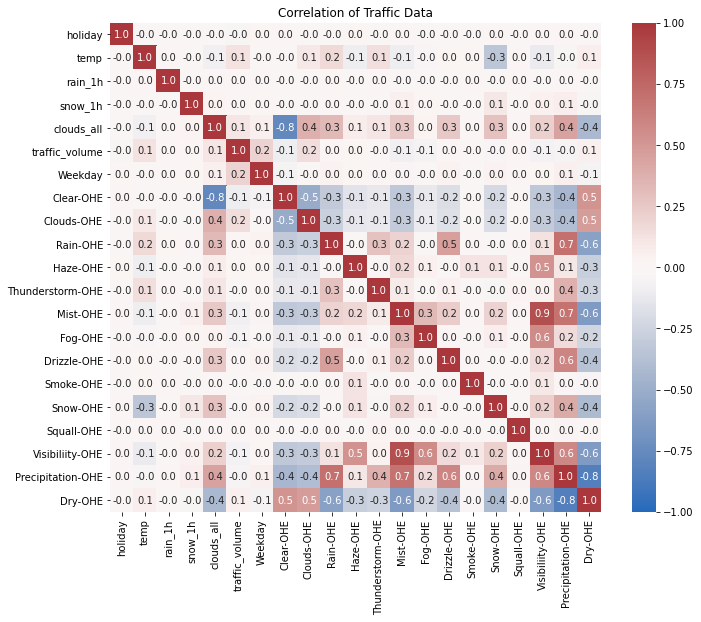
After Finalizing these features, our dataset contained 28,676 time points and 23 unique features.



Inspecting the traffic volume vs time, the volume of traffic does not have an obvious trend and varies constantly between 0 and 7000 vehicles per hour.



Looking at the ACF and PACF, it is apparent that



The correlation matrix for this dataset is understandably large, given the one hot encoding columns that were implemented to describe the weather. There is a large amount of correlation between the manufactured features that combine several weather conditions and the columns representing those same weather conditions, as expected.

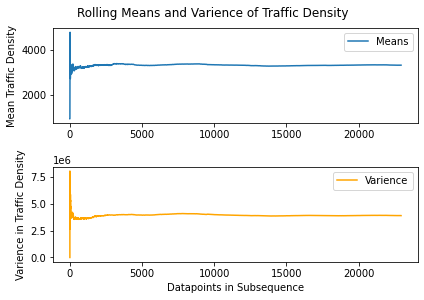
There is also strong negative correlation between the number of clouds and the clear sky columns, indicating that with higher cloud counts there is lower likelihood to report a clear sky.

As a final check, all columns were summed for null values, and none were found.

Stationarity

Prior to modeling, it was essential that the dataset be stationary, since that is an underlying assumption behind several models that we will use.

The data were split into a training (80%) and testing (20%) dataset, with Traffic Volume as the independent variable.



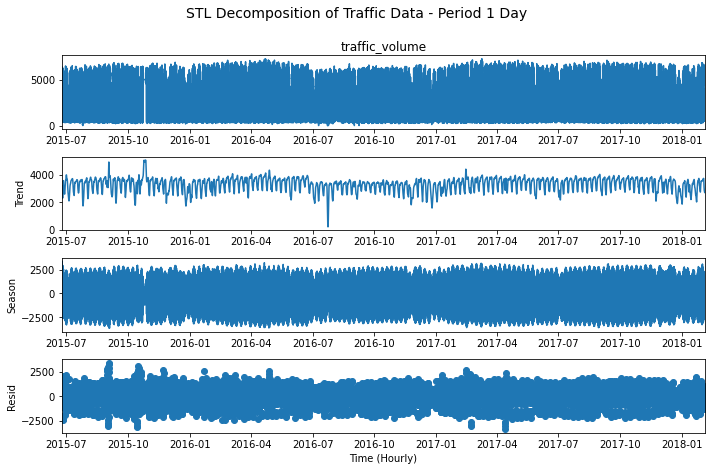
Examining the means and variance of the traffic volume, incrementally adding datapoints in the sequence, it appears that the data is stationary. This can be seen in the plots above, as the means and variance are remarkably stable over time after the initial few datapoints.

|  |  |
| --- | --- |
| ADF Test Results | |
| ADF Statistic | -18.245903 |
| p-value | 0.000000 |
| 95% Confidence | Significant |
| 99% Confidence | Significant |

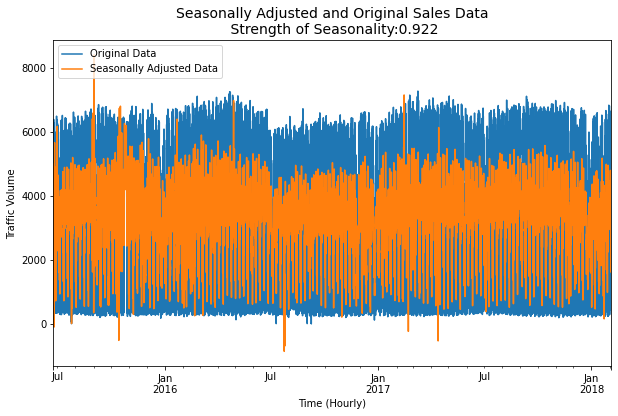
Performing an ADF test confirms the results anticipated from the incremental means and variance plots. At the 99% confidence level, the ADF test rejects the null hypothesis that the data is not stationary, and we accept the alternate hypothesis that the data is stationary.

Time Series Decomposition

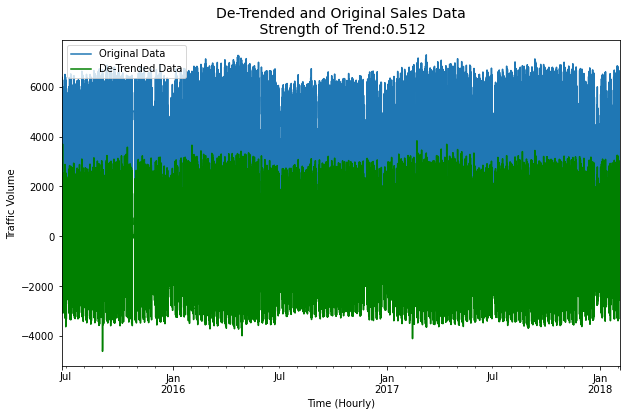
To quantify the effect of trend and seasonality of the data, we performed an STL decomposition (also known as a Seasonal and Trend decomposition using Loess). This analysis isolates trend and seasonality in the data, allowing their inspection, quantification, and potential removal from the data.



Looking at the results of the STL decomposition, it is clear that the data is stationary. Over the course of the data, the trend line remains at a consistent volume, with fluctuations that always remain in the same span. The seasonality is likewise consistent, with no shifting trend or variability.



The seasonally adjusted data appears similar to the original data, but with a reduced range of variability. This intuitively makes sense, as removing the seasonality from the data would reduce the impacts of daily traffic events, such as rush hour, or the reduced traffic in the middle of the night. The seasonally adjusted traffic reflects the mean amount of traffic one could reasonably expect without knowledge of the exact time.



Removing the effect of trend from the data, we see a marked effect on the de-trended plot. The de-trended data, in green, is shifted relative to the original data, in blue, such that it is centered on 0, rather than approximately 2500. Outside of this effect, there does not seem to be any other large impact following the removal of trend.

|  |  |
| --- | --- |
| Strength of Seasonality | Strength of Trend |
| 0.922 | 0.512 |

Feature Selection

To determine if feature selection was necessary, we first performed an SVD analysis, and examined the condition number of the full dataset. For the singular values, we fount that three of the values were approaching 0, indicating that at least one feature was highly correlated. For the condition number, we obtained a value of 1.18e19. As this value is greater than 100, it indicates that there is co-linearity, and since it is much greater than 1000, the co-linearity is severe.

As we anticipated when we examined the correlation plot, we need to perform feature selection to remove the correlated and colinear features.

To select which features needed to be removed, we performed a backward step-wise regression on the dataset. To do so, we took our training and testing data and performed a multiple linear regression using the OLS package from statsmodels. Using the summary output, we removed one feature at a time from the dataset, preferentially removing those with the largest p-values, and when all p-values were below 0.05, removing those with the largest standard error. For each feature removal, the adjusted r-squared value of the model was compared with the value before the feature was removed. In the case of a large drop in adjusted r-squared, the feature was returned to the dataset. Once no feature could be removed without a large decrease in adjusted r-squared, the feature selection was determined to be complete.

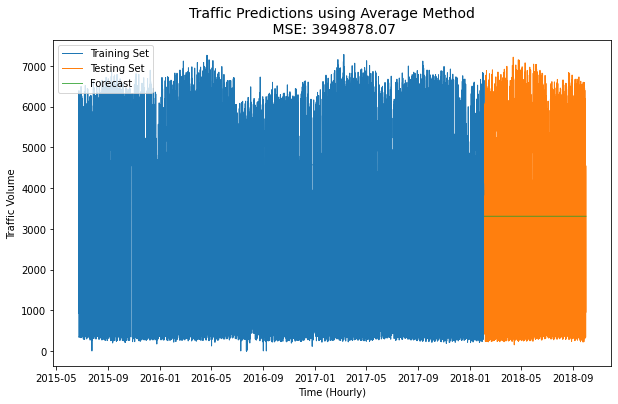
This feature selection reduced our number of features from 23 to 3. The remaining features predicting the amount of traffic were temperature, percent cloud cover and weekday.

A final SVD and Condition number analysis was performed for this reduced dataset. All singular values were much greater than 0, therefore no features are highly correlated. Since the condition number is <100, there is no co-linearity between features.

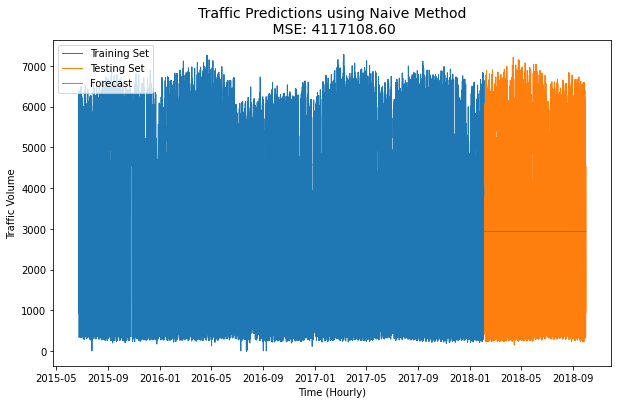
Basic Models

To form a baseline of comparison for our ARMA models, a suite of basic models were first run on the data. These models serve as a benchmark against which our more sophisticated models can be measured. If an ARMA model does the same or worse than a simple average, then we know for certain we have not found a final model.

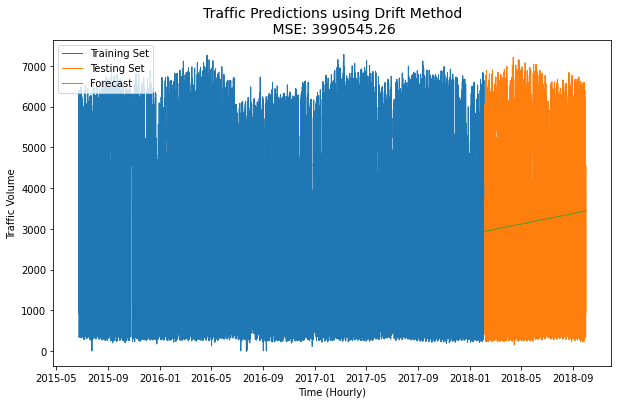
For each of the basic models we have done one-step ahead forecasting of the testing data, which was plotted for comparison.



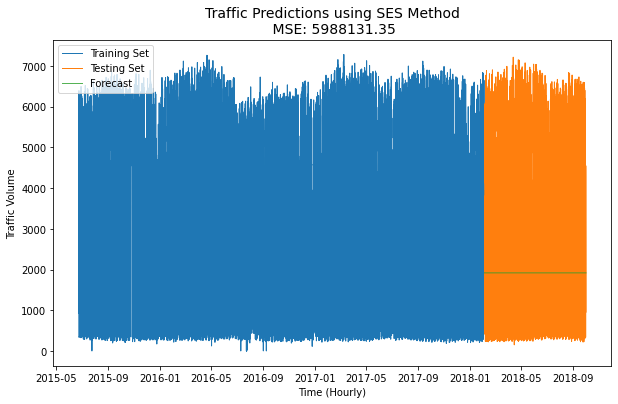
The Average basic model gives all datapoints equal weight, and in essence takes an average of all the datapoints prior to the one you are predicting. As seen above, all forecasted values are the same, and are equal to the mean value of the training set. The MSE of the forecast errors is 3,949,878, which indicates a poor predictive power.



The Naïve basic model applies zero weight to any datapoint beyond the most recent datapoint, which in the case of the testing set is the final datapoint of the training set. As with the average model, this single value is applied to all forecasted values. The MSE of the forecast errors is 4,117,108, which, as with the average method, indicates a poor predictive power.

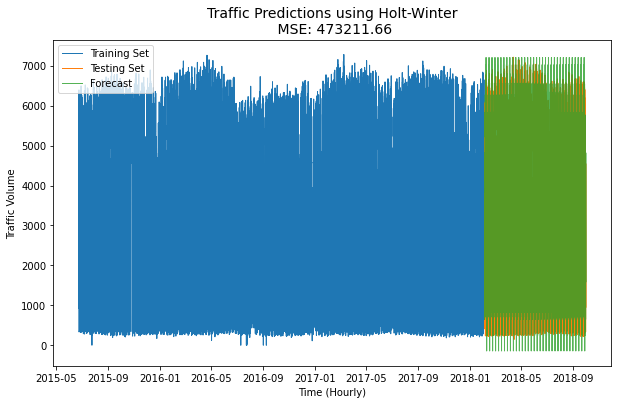


The Drift method is slightly more sophisticated than the average of the naïve method. It operates by applying weight to the first and last points, and extrapolating a slope from them, then extrapolating all forecasted values onto that slope. While the fit looks better, the MSE of the forecast errors is 3,990,545 which is slightly worse than the Average method, but better than the Naïve method.



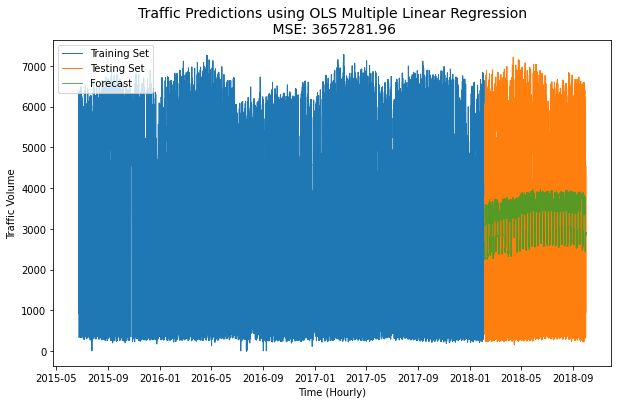
The Simple Exponential Smoothing (SES) method works as a compromise between Average and Naïve by putting a large amount of weight on the most recent point, but still applying a steadily decreasing weight to historical data. Despite the increase in complexity of the model, the SES method did not predict the data well, likely due to the 24h difference between the most recent timepoint and what would likely be the most informative timepoint for prediction. The MSE of the forecast errors is 5,988,131.

Holt-Winters



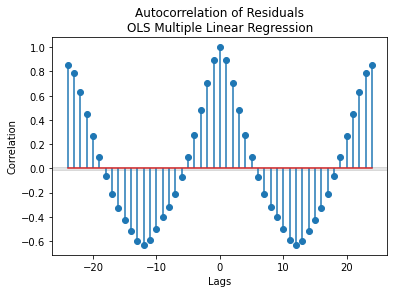
The Holt-Winters model is an extension of the SES model, which places weight on the most recent datapoint, and extends decreasing weights into previous datapoints. The Holt’s linear method builds upon this, and incorporates trend into the forecast, and the Holt-Winters method compounds on this by also capturing seasonality. As seen in the plot above, this method is very effective for capturing our dataset. It does not precisely predict the values, but its predictions do encompass the majority of the true values of the testing set. The MSE of this model is 473,211, which is an order of magnitude better than all of the previously shown basic models.

Multiple Linear Regression



The OLS multiple linear regression is a linear regressor that takes multiple features as input, in addition to the dependant predictor variable. For this analysis, we use not only the traffic density to make our predictions, as with the previous models, but also the features we selected during the feature selection step, which are temperature, percent cloud cover and weekday/not weekday.

|  |  |  |
| --- | --- | --- |
| Variance of the Residuals | Mean of the Residuals | RMSE of the Residuals |
| 1909.775 | -7.266 | 1909.6 |



Analyzing the residuals reveals that there is a large amount if information not captured by the model, confirming a visual inspection of the forecasting plot. The high amount of autocorrelation in the residuals shows that they are not white – and thus are not capturing all the data. The large mean of the residuals further suggests that the residuals are biased, which also indicates that the prediction is not appropriate.

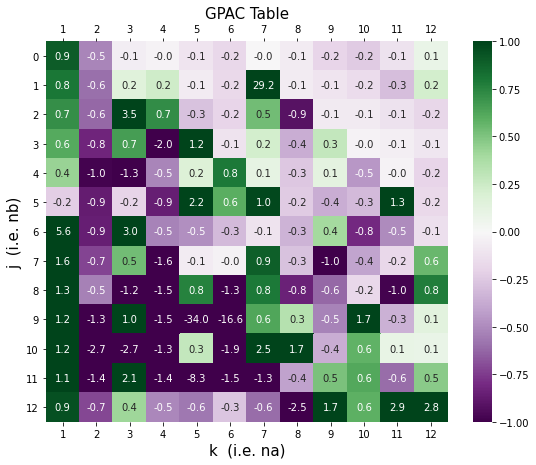
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| F-Test | F-Test Significance | Q-Value | Q Crit | Q Significance | AIC | BIC | R-squared | Adj. R-Squared |
| 2.345e+04 | Significant | 118942 | 32.6 | Not Sig | 4.117e+05 | 4.117e+05 | 0.754 | 0.754 |

Looking at the statistics of the model, we can further understand the poor fit. The r-squared and adjusted r-squared are both relatively high, which indicates a good of fit with adjusted r-squared representing a good fit without overcomplicating the model with extraneous features. However, AIC and BIC are also both very high, which indicates that the model may be overfitting, leading to poor results. Together, this suggests that while we have an appropriate number of features, the model may be putting too much weight on them, and is overfitting to the point of poor predictive power.

|  |  |  |  |
| --- | --- | --- | --- |
| Feature | t-Value | P>|t| | Significance |
| Temperature | 94.277 | 0.000 | Significant |
| % Cloud Cover | 14.162 | 0.000 | Significant |
| Weekday | 30.965 | 0.000 | Significant |

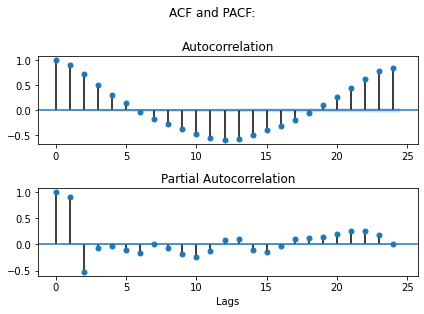
Individually, each feature has a significant t-test, as shown with the 0.000 p values. This indicates that our features are well-chosen, and significantly help the model.

ARMA, ARIMA, SARIMA Models



Looking at the GPAC table for our data, we can identify the likely ARMA model orders from the patterns present. As shown with the red outlines, there are two likely k values representing our na order, either 2 or 6. This can be seen by the columns of constant values at these indices. The j value is likely to be 0, as the row of 0s that intersects these values is located at j=0, representing an nb order of 0.

Therefore, an ARMA(2,0) or ARMA(6,0) are both candidates for potential models.



Examining the ACF and PACF plot, we can see that the PACF plot cuts off at lag=2, which supports our proposed ARMA(2,0) model. As there does not appear to be interference in the pattern, it is unlikely that we have a full ARMA model, and likely have a AR model. The ACF portion of the plot does not cut off nor trail off, and instead seems to fluctuate sinusoidally.

To address the seasonal component of the data, a SARIMA(2,0,0)12 model was also fit.

Levenberg Marquardt Algorithm

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Coefficient | | Confidence Interval | Standard Deviation |
| ARMA(2,0) |  | -1.374 | -1.365 to -1.384 | 0.757 |
|  |  | 0.521 | 0.529 to 0.513 | 0.606 |
|  |  |  |  |  |
| ARMA(6,0) |  | -1.321 | -1.310 to -1.331 | 0.757 |
|  |  | 0.431 | 0.447 to 0.414 | 1.212 |
|  |  | 0.091 | 0.113 to 0.068 | 1.818 |
|  |  | -0.044 | -0.022 to -0.066 | 1.666 |
|  |  | -0.123 | -0.101 to -0.145 | 1.666 |
|  |  | 0.172 | 0.188 to 0.157 | 1.212 |
|  |  |  |  |  |
| SARIMA(2,0,0)12 |  | -1.019 | -1.012 to -1.027 | 0.606 |
|  |  | 0.229 | 0.237 to 0.221 | 0.606 |
|  |  | 0.167 | -0.176 to -0.158 | 0.757 |
|  |  | -0.579 | 0.573 to 0.586 | 0.454 |

All three of our models were fit using a LM algorithm, and the coefficients of the parameters were obtained. The confidence intervals and standard deviation of that coefficient was calculated for each coefficient.

The confidence intervals for each coefficient was analyzed, and surprisingly given the number of coefficients, all were found to be significant, as none span across 0.

Of note, the first two coefficients of the ARMA(6,0) model are very similar to those of the ARMA(2,0) model, and the subsequent coefficients are very small. This suggests that the ARMA(6,0) model may be acting as effectively an ARMA(2,0) model with small adjustments.

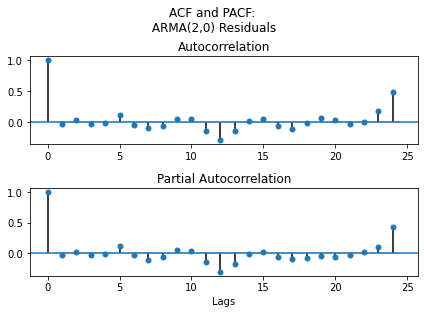
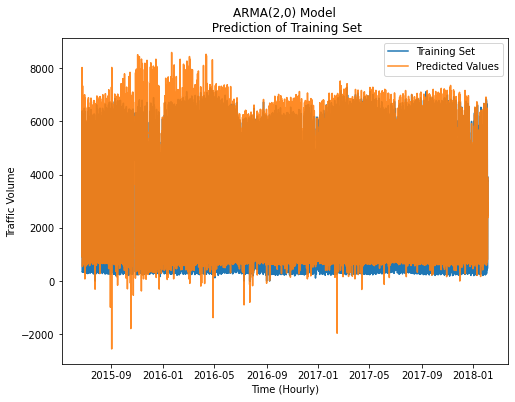
Diagnostic Analysis

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Model | Q-Score | Q Crit | Result | Mean of Residual | Bias? | Variance of Residuals | Variance of Forecast | Ratio |
| ARMA(2,0) | 5135 | 34 | Not White | -0.05 | No | 520002 | 352826 | 1.47 |
|  |  |  |  |  |  |  |  |  |
| ARMA(6,0) | 3585 | 29 | Not White | -0.07 | No | 496684 | 342821 | 1.45 |
|  |  |  |  |  |  |  |  |  |
| SARIMA(2,0,0)12 | 1914 | 34 | Not White | -0.08 | Minor | 344264 | 205222 | 1.68 |

To perform a diagnostic analysis for all three models, we first performed a one-step prediction for the training set. This was plotted versus the training set to allow for immediate visualization. Next, the residuals were calculated, and the ACF and PACF of the residuals were plotted for analysis.

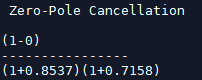
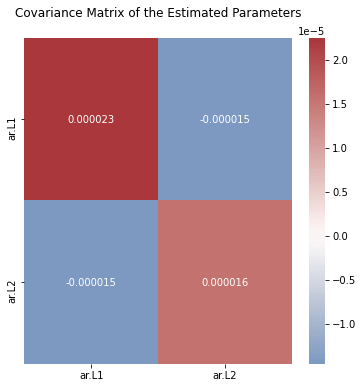
The Q-score was calculated on the residuals to determine if they were white. The means and variance of the residuals were calculated to check for bias and for comparison with the forecast variance to determine adaptability to new information respectively. These factors were tabulated for each model to aid model selection.

Below, we analyze the plots of the three models, and discuss the implications of each.

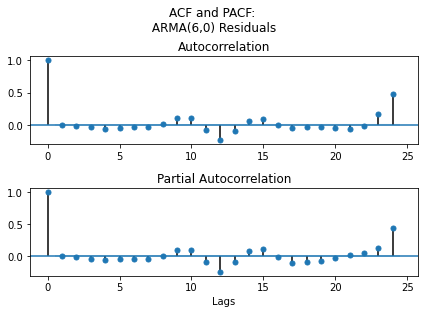
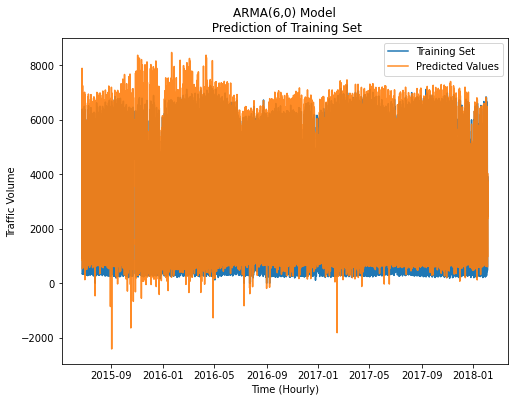


Examining the prediction of the training set, it is clear that some values cause substantial problems for the model’s predictions. In the first half of the training set, many values are overestimated, and periodically throughout the entire training set there are seemingly erratic negative predictions. Evidently negative traffic is not possible, and no such values appear in the training set.

The ACF/PACF plot of the residuals reveals that there is seasonal information available in the residuals that is not captured by the model, as seen at lag 12 and 24. These lags prompted the utilization of the SARIMA model mentioned earlier.

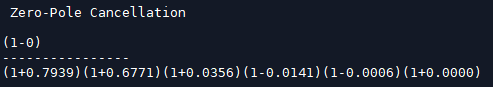
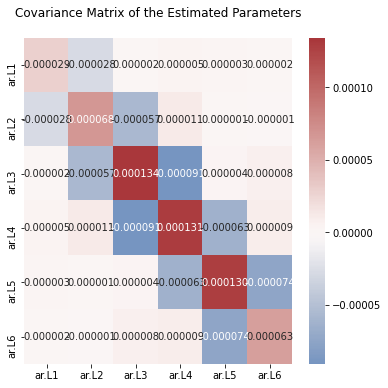


The covariance matrix shows that the two AR coefficients are in opposite directions, and are not very tightly correlated to each other. The zero-pole cancellations does not reveal any roots to cancel, largely because the model is an AR model, rather than a full ARMA model, lending less opportunities for cancellations.

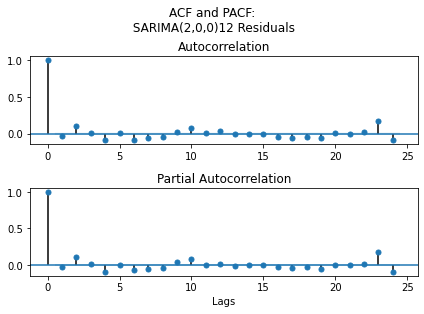
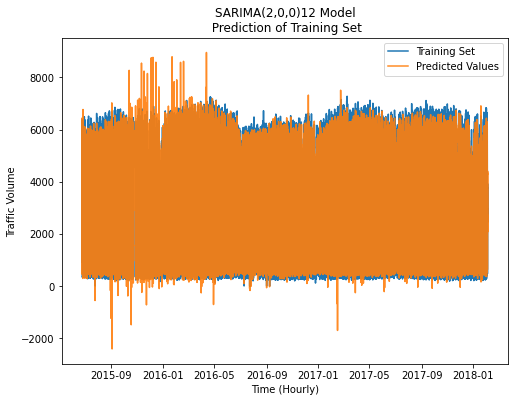


The prediction of the training set for the ARMA(6,0) model understandably looks very similar to the ARMA(2,0) prediction plot. As previously established, there is only slight variation in the first two coefficients between the models, and the remaining coefficients in the ARMA(6,0) model are very small. Nonetheless, there is a difference between the two. In the ARMA(6,0) plot, there are fewer fluctuations that spike to extreme values, and those that exist do so to similar or lesser extremes than the ARMA(2,0) model. Further, the ARMA(6,0) model seems to be more capable of predicting the low traffic conditions than the ARMA(2,0), though this difference is very slight.

The difference in the ACF/PACF plots is slightly more pronounced, with the lags between the significant 12, 24 lags being more consistently insignificant. However, these significant lags, indicating that seasonal information is not being captured.

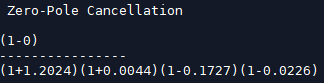
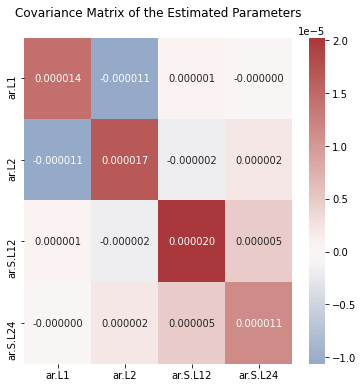


The zero-pole cancellation for this AR model surprisingly does exhibit cancellable roots, namely the 5th and 6th AR roots. If this model were to proceed as the selection for the final model, these would need to be cancelled out.



The SARIMA(2,0,0)12 model’s prediction of the training set is the most accurate of the three ARMA models tested. While, like the other two, it does have occasional extreme values, it has fewer of these than the other models. It fits the actual data much more closely, which is particularly noticeable in the first third, where it accurately predicts most data, where the other two models consistently overestimate. This model is also the best at predicting the lower range of the values as traffic approaches 0, and is consistently close to the upper range.

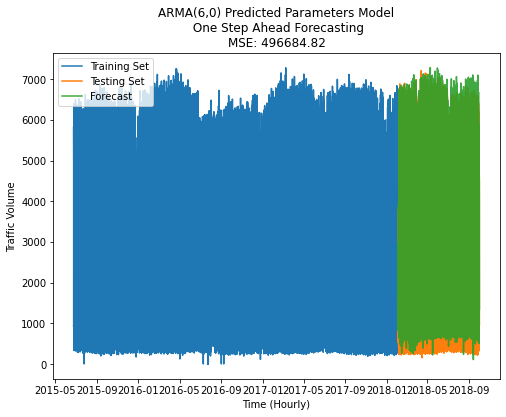
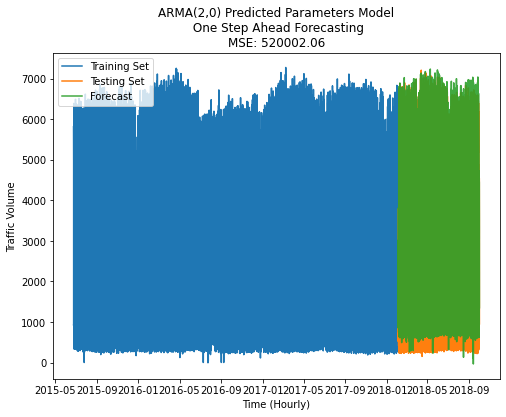
The ACF/PACF of the residuals are insignificant at all lags except lag=24, which is only barely significant. This indicates that there may be some amount of information left in the residuals that the model is not capturing, but not a large amount.

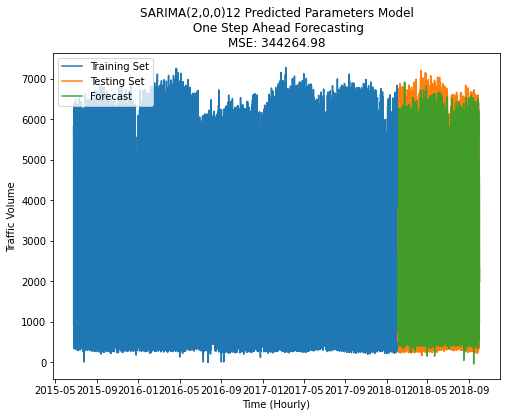


In the covariance matrix it is interesting to observe that the non-seasonal coefficients (ar.L1 and ar.L2) do not covary with the seasonal components (ar.S.L12 and ar.S.L24), but each set of components (seasonal or non-seasonal) do covary with eachother.

The zero-pole cancellation does not have any values that are close enough to cancel.

Final Model selection





To make an even comparison to the basic models, a one-step forecasting was performed for all the ARMA models for the length of the test set, and these were plotted for visual comparison. The MSE of the forecasting was tabulated and compared.

|  |  |
| --- | --- |
| Model | MSE – One Step Forecasting |
| Average | 3,949,878 |
| Naive | 4,117,109 |
| Drift | 3,990,545 |
| SES | 5,988,131 |
| Holt- Winter | 473,212 |
| ARMA(2,0) | 520,002 |
| ARMA(6,0) | 496,684 |
| SARIMA(2,0,0)12 | 344,265 |

Of the basic models, SES has the highest MSE of the forecasting, indicating the worst fit. Next is Naïve, followed by the Average method, hen the Drift method. Of the simple models, Holt-Winter predictions did shockingly well, with a better fir than the two ARMA models. This is likely due to the seasonality of the data, which is better represented in the Holt-Winter model than a non-seasonal ARMA.

The SARIMA model performed the best of all models, a full order of magnitude better than the best basic model, and significantly better than any of the ARMAs or the Holt-Winter model.

We will select the SARIMA model as our final model for predicting traffic density.

Forecast Function

h-step ahead Predictions

Summary and conclusion

References

Guide to Resampling in Pandas [https://kanoki.org/2020/04/14/resample-and-interpolate-time-series-data/]