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 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 time series analysis process

*Dataset*

Before making predictions, the dataset is cleaned and processed to optimize features for predictions, and remove possible issues that may arise.

Some steps required for this may include re-sampling the data to ensure that there are no gaps in the time-series data, or merging data if multiple datapoints exist for the same point in time.

Feature engineering may also be required to allow use of multiple linear regressions, such as transforming string variables to one-hot encoding columns, or booleans as appropriate. Extraneous features may be removed here if they will interfere with the feature selection process, or they may be removed at that time.

*Stationarity*

Many tools and models used in time series analysis rely on the assumption of stationarity to function correctly. Stationarity is established when the dependant variable is stable through time, and the means and variability of the data is constant through time. To test for this, one plots a rolling means and variability through time, adding one datapoint at a time and visually assess if the rolling values change with the addition of sequential datapoints. Additionally, an Augmented Dickey–Fuller test (ADF) test is performed on the data to quantitatively test for stationarity. Finally, an Autocorrelation plot is created to test for seasonality, which appears as an oscillatory pattern in the autocorrelation.

If at any of these steps non-stationarity, the dependant variable may be transformed to induce stationarity. These transformations may include one or more differencings, seasonal differencings, or exponential transformations, as needed.

*Time Series Decomposition*

Time Series Decomposition is performed to quantify the effect of trend and seasonality of the data. In this analysis, we used a Seasonal and Trend decomposition using Loess (STL decomposition) to separate trend and seasonality from our data, which we then used to create a seasonally adjusted and a de-trended dataset, and to calculate the strength of trend and seasonality in our data.

*Feature Selection*

While theoretically the addition of supplemental data to a model via additional features should improve model performance, this is not always the case. Unhelpful features, which do not provide information that are used to predict the dependant variable increase processing times, and can cause poor fits of the model. Colinear features, which are pairs or groups of features which share information, can cause high computation costs and poor model performance if multiple of these features are left in the final dataset. Collinearity can be detected in multiple ways, including SVD analysis, and Condition Number.

During feature selection the data is fit to a multiple linear regressor, and features are sequentially removed in order of highest p-value, followed by highest standard error. After each removal, the AIC, BIC and adjusted r-squared are compared to the value prior to removal. Increases in adjusted r-squared, or very small decreases are desirable, as are decreases in AIC and BIC. When the removal of any feature results in a sharp decrease in adjusted r-squared, the feature selection is complete.

*Basic Models*

Basic models are computationally inexpensive models that serve as a baseline for comparison with more complex models developed later on. These simple models are typically relatively simple processes with minimal tuning required to achieve a complete output.

In this analysis, we employ the Average method, the Naïve method, the Drift method, and the SES method of forecasting. The Average method gives all datapoints equal weight and is equivalent to taking the average of all preceding datapoints. The Naïve method applies 0 weight to any datapoint beyond the most recent point. The Drift method applies weight to the first and last points, and extrapolated a slope from them. Finally, the SES method, or the Simple Exponential Smoothing method, works as a compromise between Average and Naïve by putting a large amount of weight on the most recent point, but still applying decreasing weights to historical data.

*Holt-Winters*

The Holt-Winters model is an extension of the SES model, which also incorporates trend and seasonality into the forecast. In non-stationary data, the Holt-Winters model tends to perform better than the SES variant.

*Multiple Linear Regression*

Multiple linear regression is a linear regressor that takes multiple features as input, in addition to the dependant predictor variable. In many cases this can provide a much more accurate prediction of future values, if the independent variables are accurate predictors of the dependant variable. For this analysis, we use an ordinary least squares (OLS) multiple linear regressor, and input the traffic density as well as the features we select during the feature selection step to make our predictions.

*ARMA, ARIMA, SARIMA Models*

ARMA models (Auto-Regressive Moving Average models) are effective linear models which combine Auto-Regressive (AR) and Moving Average (MA) models into a single predictive package. ARMA models require a minimal number of parameters to predict future values – unlike a multiple linear regression they only require past values of the dependant variable to produce forecasts. However, ARMA models are sensitive to trend and seasonality.

ARIMA models are ARMA model that have been extended to accommodate trend in the data, and similarly SARIMA models accommodate both trend and seasonality.

The order of the ARMA model can be determined in several ways, including the use of a GPAC table, or analysis of a ACF/PACF plot. A GPAC table can be interpreted by scanning the table to find columns of constants that adjoin rows of 0s. The column of constants at column k indicates an AR order of k, and the row of 0s at row j represents an MA order of J.

An ACF/PACF plot can be used to determine the orders of AR or MA models by assessing if there is a cutting-off pattern of significance in one plot and a trailing off in the other. If this is the case, then the number of significant lags before the cut off represents the order of the AR if the cut-off occurred in the PACF plot, or the MA if in the ACF plot, and the opposite ARMA component is 0. If neither cuts off, or the plots are chaotic, it is likely that the data is best represented by an ARMA model, and the GPAC table should be consulted.

*Levenberg Marquardt Algorithm*

The Levenberg Marquardt Algorithm (LM) is a fast-converging maximum likelihood estimation algorithm which is used to estimate the coefficients of the ARMA models, given the orders determined using the GPAC table, or ACF/PACF plot.

These coefficients must be checked for significance using their confidence intervals, which are determined to be insignificant if they span across 0.

*Diagnostic Analysis*

Diagnostic analysis is a broad term for a suite of tests that are performed on the model to determine the quality of the prediction and the goodness of fit of the model. This includes determining if the roots of the coefficients are as simplified as they can be, using a zero-pole cancellation, performing a chi-squared test on the residuals to assess if they are capturing all information (white), and checking for bias in the prediction using the mean value of the residuals. Further, assessments of the variability of the residual errors vs forecast error can show if the model adapts well to new data. Additional plots, such as ACF/PACF plots of the residuals can also help illuminate what information is not being captured by the model, if any.

*Final Model selection*

Final model selection is performed by assessing all compiled metrics relating to the models tested, from the basic models to the multiple linear regressions and the ARMA models. Using the quantitative measurements taken in previous steps, the model that fits the data the best is determined and selected for further processing.

*Forecasting and h-step Ahead Predictions*

Using the coefficients from the final model, the forecasting function is created. The forecasting function is a general form equation that takes past values of the dependant variable and makes predictions about future values. It can be transformed to make predictions h steps in the future, where h is any integer greater than 0.

This forecasting function is then used to make a forecast about the future values of the testing set.

Dataset

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.

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.

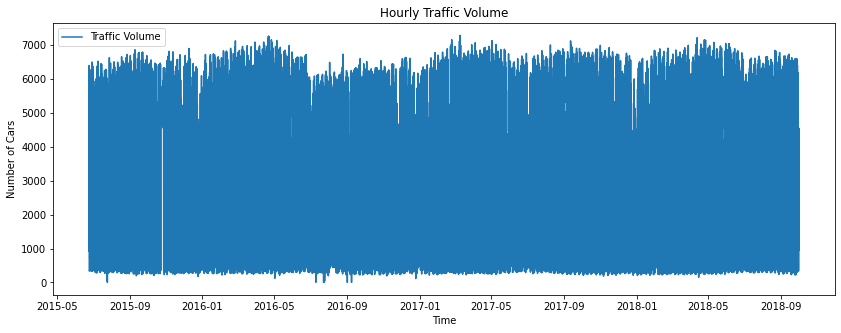
Several of the weather columns were grouped together into weather categories, to provide additional information to the model, with the expectation that a consolidated column may be more informative. An example of such a column would be the ‘Precipitation’ feature, which combined ‘Thunderstorm’, ‘Drizzle, ‘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.

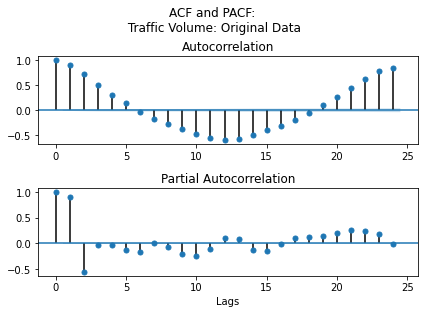
Finally, a boolean weekday/not weekday column was created using the timestamps of the hourly traffic report. This feature was created, as it seemed likely that traffic patterns would be strongly influenced by the day of the week, and thus if commuters would be largely present on the road.

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.

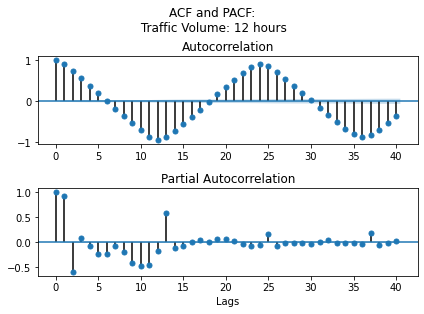


Inspecting the traffic volume vs time, the volume of traffic does not have an obvious trend and varies constantly between 0 and 7000 vehicles per hour.

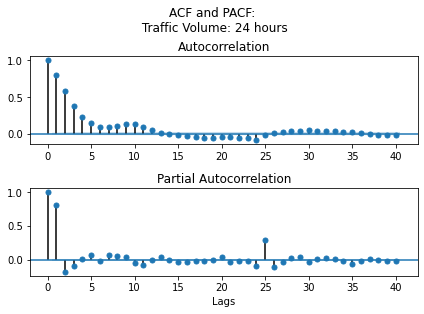


Looking at the ACF and PACF, it is apparent that there is a large degree of seasonality in the data, which is shown by the sinusoidal ACF plot. This intuitively makes sense, as traffic patterns tend to repeat on a daily basis – Rush hour is at 6pm each day, and there is very little traffic at 3am.

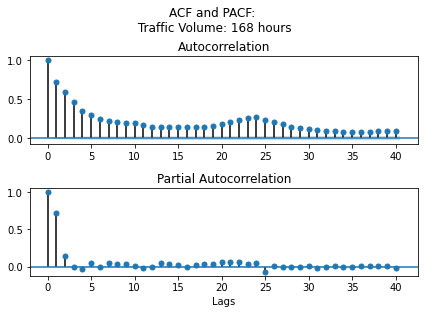
To make the data stationary, we performed a seasonal differencing. To select the interval to difference, several seasons were tested and visualized with an ACF/PACF plot.



The first season tested was a 12 hour difference, as it was thought that a 12 hour difference would typically be the opposite traffic pattern. However, the 12 hour differencing did not reduce the oscillatory seasonality in the autocorrelation, and rendered the PACF chaotic. The 12 hour differencing data was not stationary.

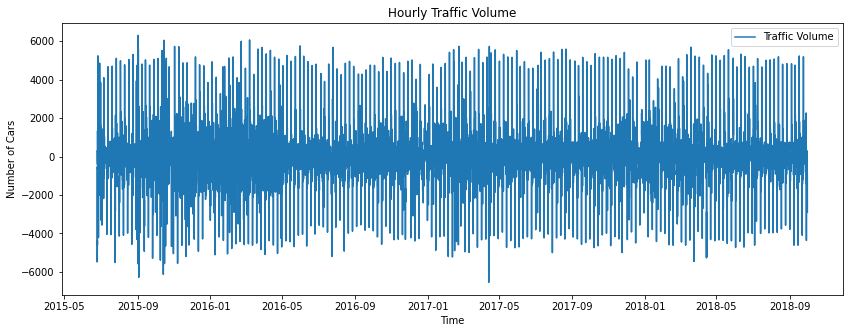


A 24 hour seasonal differencing was then tested, with the theory that traffic patterns are likely seasonal based around individual days, with similar traffic patterns at similar times regardless of day of the week. This seasonal differencing has led to an acf that trails off quickly and remains non-significant without oscillations, indicating a lack of seasonality. The PACF cuts off after 1 significant lag, with another significant at lag=25.

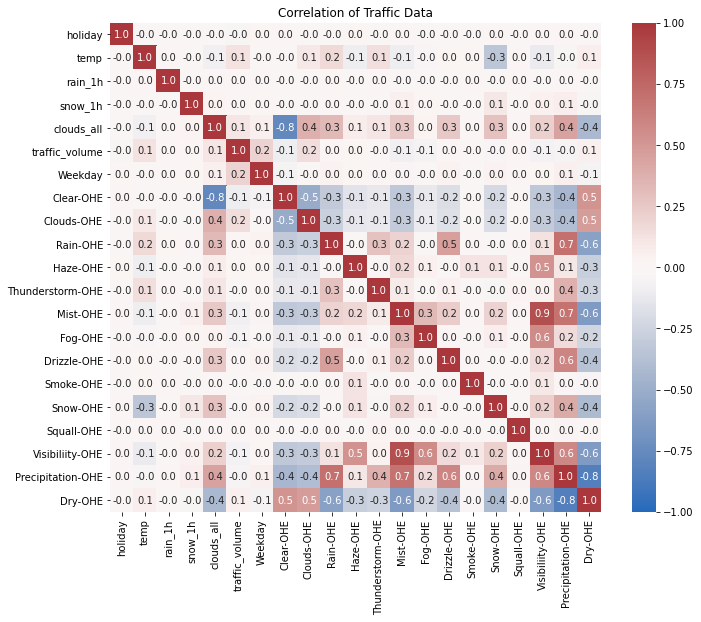


A 168 hour seasonal differencing was the final differencing option tested, representing one week of time between each time point. The theory underlying this was that the day of week may play a large role in the traffic pattern – eg, comparing Saturday traffic to Monday traffic may not be ideal. Examining the ACF plot, the autocorrelation does decrease over time, however duration where it is significant, combined with the slight peak at 24 hours was concerning.

For this analysis, the 24 hour seasonal differencing was used to make the traffic density data stationary. The traffic data was differenced, and recombined with the other features with a 24 index set-back, such that the first 24 hour period of the non-target features were removed.



After differencing, the traffic density now has a range between 6000 and -6000, with an approximate mean of 0.



The correlation matrix for this dataset is extensive, largely driven by the addition of the one hot encoding columns 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.

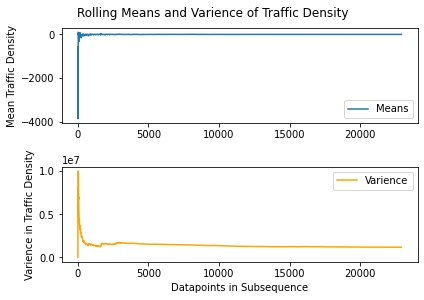
As a final check, all columns were summed for null values, and none were found.

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.

During the description of the dataset, a strong seasonality was observed in the ACF plot, and the data was seasonally differenced with an interval of 24 hours to address this. This seasonally differenced data is henceforth the dataset that will be used for the remainder of the analysis.

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



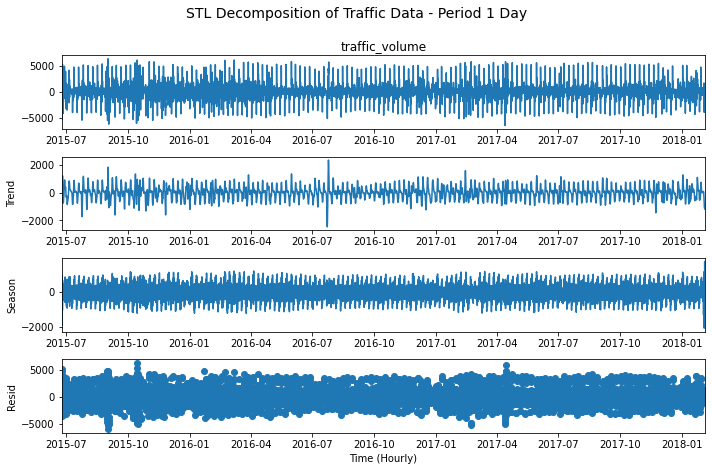
Examining the means and variance of the traffic volume, incrementally adding datapoints in the sequence, it appears that the data is stationary. This can be seen in the plots above, as the means and variance are remarkably stable over time after the initial few datapoints.

|  |  |
| --- | --- |
| ADF Test Results | |
| ADF Statistic | -33.889877 |
| p-value | 0.000000 |
| 95% Confidence | Significant |
| 99% Confidence | Significant |

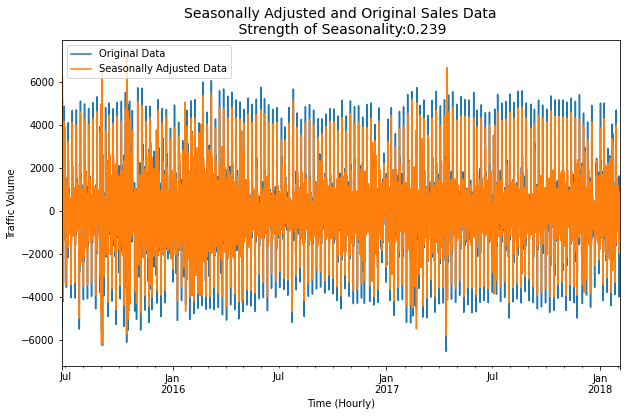
Performing an ADF test confirms the results anticipated from the incremental means and variance plots. At the 99% confidence level, the ADF test rejects the null hypothesis that the data is not stationary, and we accept the alternate hypothesis that the data is stationary.

Time Series Decomposition

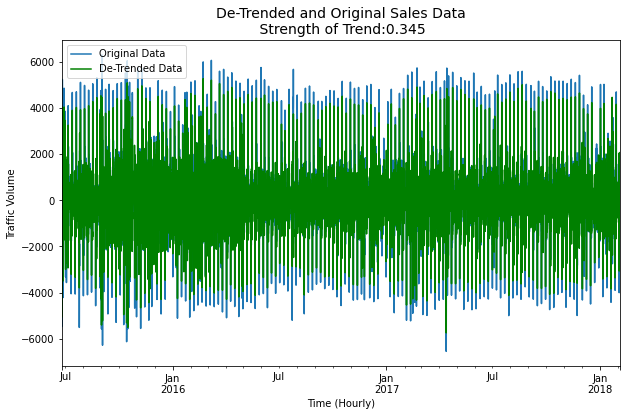
To quantify the effect of trend and seasonality of the data, we performed an STL decomposition (also known as a Seasonal and Trend decomposition using Loess). This analysis isolates trend and seasonality in the data, allowing their inspection, quantification, and potential removal from the data.



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 nearly identical to the original data, but with a reduced range of variability. This is understandable, as we have previously made efforts to remove seasonality from our dataset, and therefore seasonally adjusting the data should not produce much change.



Removing the effect of trend from the data, we see a similarly small effect on the de-trended plot. The de-trended data, in green, is slightly less variable compared to the original data, in blue. As our data was stationary, it is logical that removing the trend had no large impact, as we did not have a significant trend to remove.

|  |  |
| --- | --- |
| Strength of Seasonality | Strength of Trend |
| 0. 239 | **0.345** |

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 2.52e+18. 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.

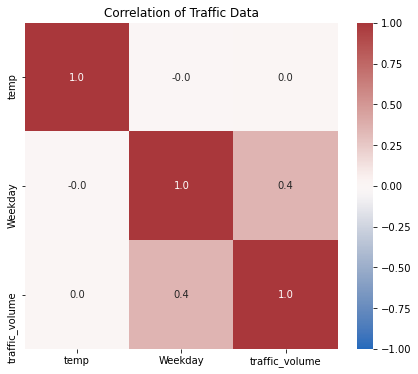
As we anticipated when we examined the correlation plot, we need to perform feature selection to remove the correlated and colinear features.

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 2. The remaining features predicting the amount of traffic were temperature, 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 minor co-linearity between features, but it is <1000 and so it is not severe.

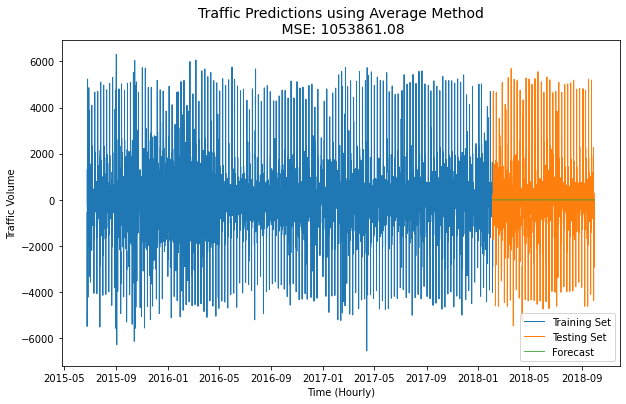
|  |  |  |
| --- | --- | --- |
|  | Condition Number | Singular Values |
| Before Feature Selection | 2.52e+18 | 1.82e+05 4.84e+04 9.83e+03 6.62e+03 1.71e+02 8.45e+01 7.14e+01 6.63e+01 4.981e+01 4.59e+01 3.38e+01 3.04e+01  2.82e+01 2.63e+01 6.07e+00 4.95e+00 1.40e+00 1.08e+00 9.13e-14 2.57e-14 1.16e-14 |
| After Feature Selection | 625.48 | 47905.77 76.60 |



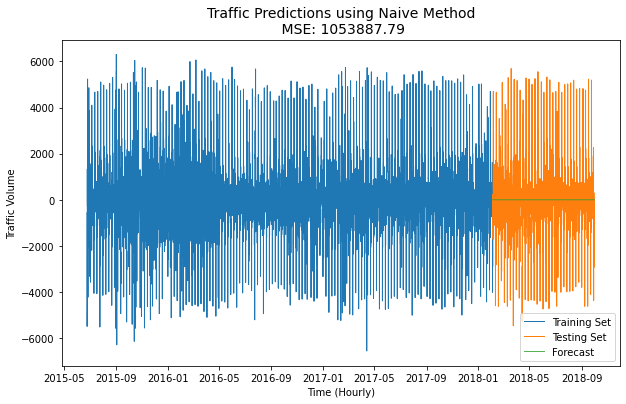
Basic Models

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.

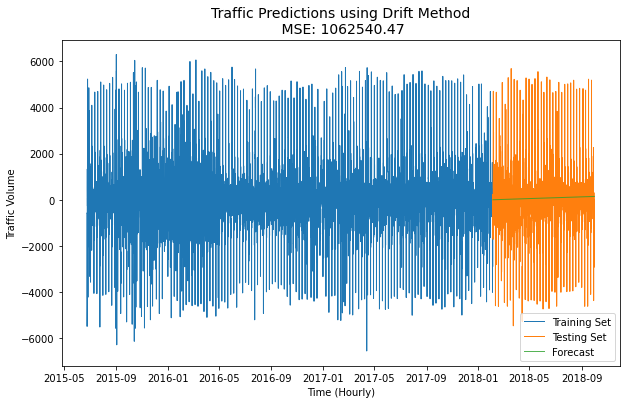
For each of the basic models we have done one-step ahead forecasting of the testing data, which was plotted for comparison.



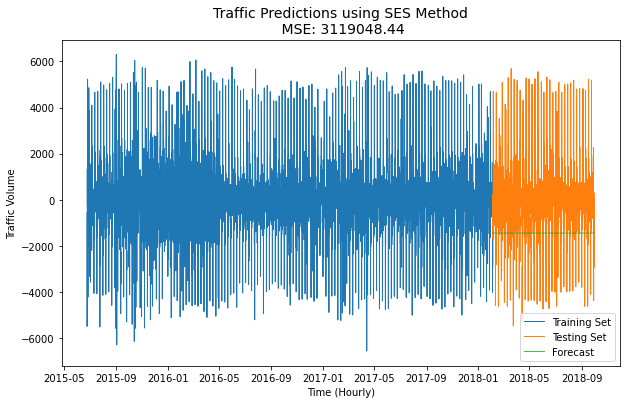
The Average basic model gives all datapoints equal weight, and in essence takes an average of all the datapoints prior to the one you are predicting. As seen above, all forecasted values are the same, and are equal to the mean value of the training set. The MSE of the forecast errors is 1,053,861 which indicates a poor predictive power. The Q-score calculated on the residuals of 29,200 indicates that there is a large amount of information not captured by the model.



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 1,053,887, which, as with the average method, indicates a poor predictive power. In this case, it appears that the final training datapoint was very near the average value. The Q-score calculated on the residuals of 2,161 indicates that there is a large amount of information not captured by the model.

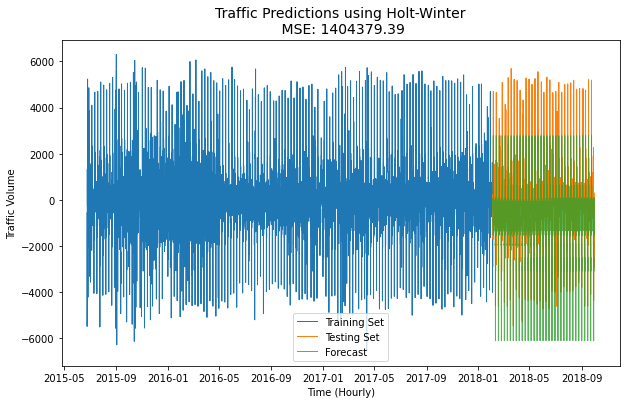


The Drift method is slightly more sophisticated than the average of the naïve method. It operates by applying weight to the first and last points, and extrapolating a slope from them, then extrapolating all forecasted values onto that slope. While the fit looks slightly better, in this case the MSE of the forecast errors is 1,062,540 which is slightly worse than the Average method and the Naïve method. The Q-score calculated on the residuals of 2,146 indicates that there is a large amount of information not captured by the model.



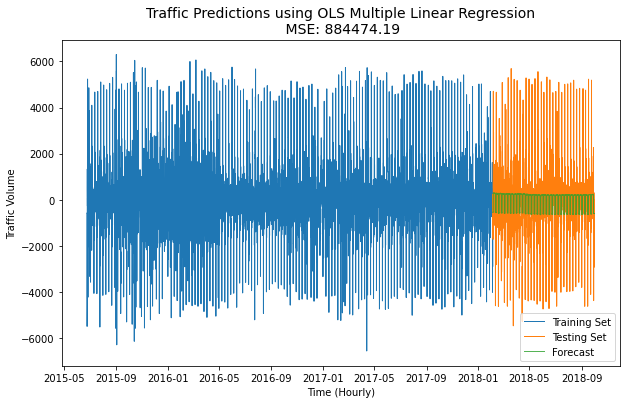
The Simple Exponential Smoothing (SES) method works as a compromise between Average and Naïve by putting a large amount of weight on the most recent point, but still applying a steadily decreasing weight to historical data. Despite the increase in complexity of the model, the SES method did not predict the data well, possibly due to a fluctuation in the data near the end of the training set. The MSE of the forecast errors is 3,119,048. The Q-score of 10,514 indicates that there is a large amount of information not captured by the model.

Holt-Winters



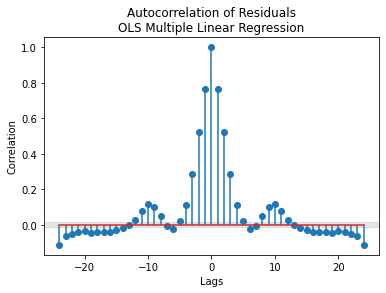
The Holt-Winters model is an extension of the SES model, which places weight on the most recent datapoint, and extends decreasing weights into previous datapoints. The Holt’s linear method builds upon this, and incorporates trend into the forecast, and the Holt-Winters method compounds on this by also capturing seasonality. As seen in the plot above, this method is only moderately effective for capturing our dataset. It does not precisely predict the values, and seems to have a bias in its predictions. This may be due to the lack of trend and seasonality in our differenced dataset. The MSE of this model is 1,404,379, which is slightly worse than most basic models, but significantly better than the SES model it is based off. The Q-score of 9,218 indicates that there is a large amount of information not captured by the model.

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 and weekday/not weekday.

|  |  |  |
| --- | --- | --- |
| Variance of the Residuals | Mean of the Residuals | RMSE of the Residuals |
| 1023.591 | - 1.600 | 1023.5 |



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 |
| 1531.0 | Significant | 22974 | 32.6 | Not Sig | 3.828e+05 | 3.828e+05 | 0.118 | 0.118 |

Looking at the statistics of the model, we can further understand the poor fit. The r-squared and adjusted r-squared are both low, which indicates a poor overall fit, with adjusted r-squared representing the fit taking into account the number of features. Even with all features in the dataset before feature selection, the maximum R-squared value was only 0.122, which is a poor fit. As the r-squared and adjusted r-squared are the same value, we can conclude that the number of features we retained is appropriate, despite the poor overall fit.

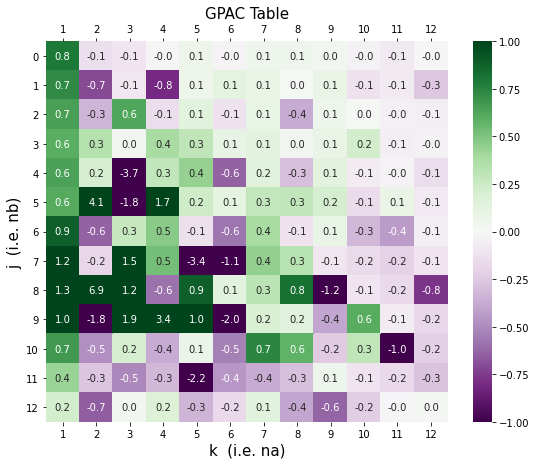
Interestingly, before the data was seasonally differenced, an r-squared value of 0.788 was achieved. This indicates that the removal of the seasonality, or differencing without altering the other features, may have negatively impacted the ability of the multiple linear regression to make predictions.

AIC and BIC are also both very high, which indicates that the model may be overfitting, further 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 | -46.836 | 0.000 | Significant |
| Weekday | 55.340 | 0.000 | Significant |

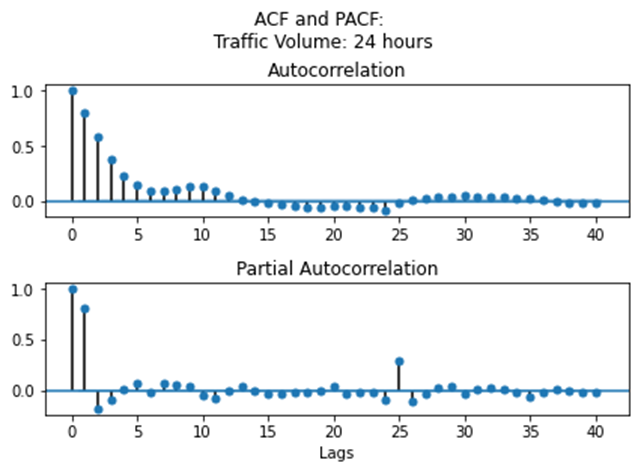
Individually, each feature has a significant t-test, as shown with the 0.000 p values. This indicates that our features are well-chosen, and significantly help the model.

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 ARMA models representing our data. The first, and strongest candidate appears at j=0, k=1, represented by a row of 0s and a column of constants. This represents an ARMA(1,0) model. The next candidate appears at j=3, k=5, representing an ARMA(5,3) model. This second model is slightly less distinct than the ARMA(1,0) model, but warrants investigation.

Therefore, an ARMA(1,0) or ARMA(5,3) are both candidates for potential models based on inspection of the GPAC table.



Examining the ACF and PACF plot, we can see that the PACF plot cuts off at lag=1, which supports our proposed ARMA(1,0) model. There is a small, significant lag at lag = 25, which may be an indicator of some remaining seasonality in the data.

The ACF plot trails off, which indicates that there in not likely to be an MA component. 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.

To address the seasonal component of the data, a SARIMA(1,0,0)24 model was also fit.

Levenberg Marquardt Algorithm

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Coefficient | | Confidence Interval | Significant | Standard Deviation |
| ARMA(1,0) |  | -0.795 | -0.790 to -0.799 | Yes | 0.303 |
|  |  |  |  |  |  |
| ARMA(5,3) |  | -0.158 | 0.872 to -1.188 | No | 79.483 |
|  |  | -0.422 | -0.295 to -0.548 | Yes | 9.841 |
|  |  | -0.068 | 0.388 to -0.523 | No | 35.124 |
|  |  | 0.164 | 0.393 to -0.065 | No | 17.7 |
|  |  | -0.039 | 0.133 to -0.211 | No | 13.323 |
|  |  | 0.737 | -0.292 to 1.766 | No | 79.483 |
|  |  | 0.190 | -0.820 to 1.199 | No | 77.969 |
|  |  | -0.026 | -0.436 to 0.384 | No | 31.642 |
|  |  |  |  |  |  |
| SARIMA(1,0,0)24 |  | -0.813 | -0.808 to -0.817 | Yes | 0.303 |
|  |  | 0.262 | 0.270 to 0.255 | Yes | 0.606 |

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 checked for significance by determining if the interval spanned across 0. For the ARMA(1,0) and SARIMA(1,0,0)24 all coefficients were significant. For the ARMA(5,3) model only one value was found to be significant, and the remaining 7 were insignificant.

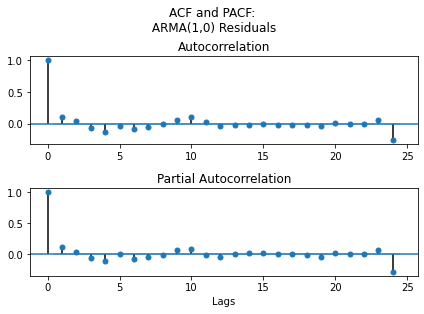
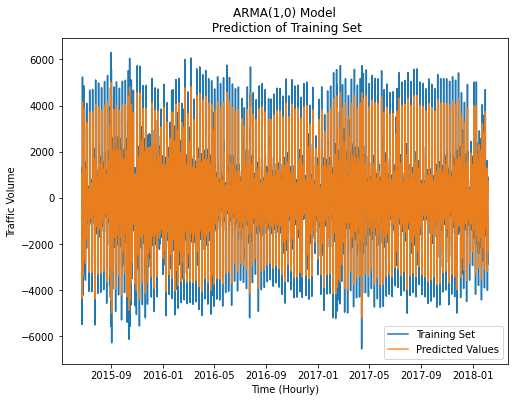
Diagnostic Analysis

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Model | Q-Score | Q Crit | Result | Mean of Residual | Bias? | Variance of the Error | Variance of Residuals | Variance of Forecast | Ratio |
| ARMA(1,0) | 1633.7 | 35 | Not White | -0.00 | No | 1.632 | 408347 | 279306 | 1.46 |
|  |  |  |  |  |  |  |  |  |  |
| ARMA(5,3) | 590.628 | 26 | Not White | -0.01 | No | 1.924 | 422198 | 250644 | 1.68 |
|  |  |  |  |  |  |  |  |  |  |
| SARIMA(1,0,0)24 | 2428.50 | 35 | Not White | 0.06 | No | 1.725 | 408347 | 276437 | 1.48 |

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.

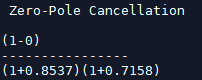
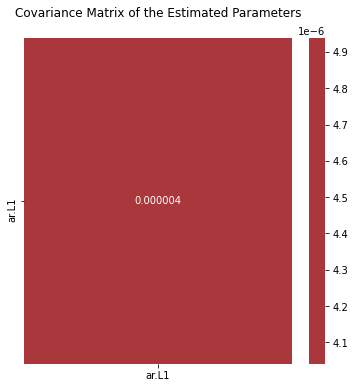
The Q-score was calculated on the residuals to determine if they were white. The means and variance of the residuals were calculated to check for bias and for comparison with the forecast variance to determine adaptability to new information respectively. These factors were tabulated for each model to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

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

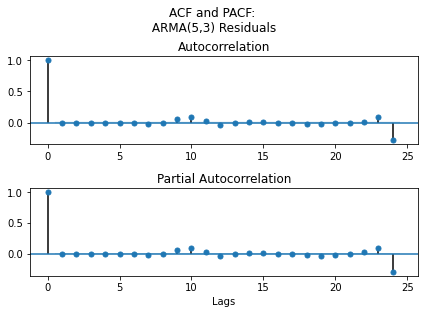
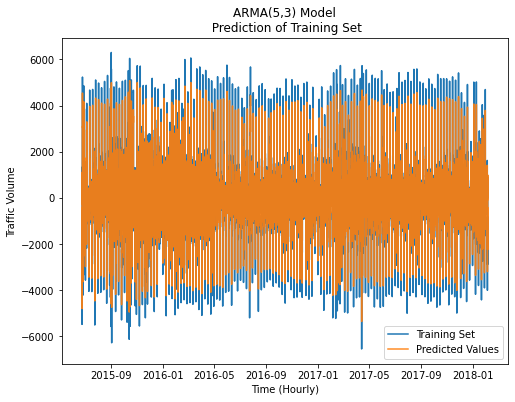


Examining the prediction of the training set, it seems that the model is doing a fairly accurate job in estimating the values of the training set. While it does not appear to be capable of estimating the extreme values, the majority of the data is being predicted correctly.

The ACF/PACF plot of the residuals reveals that there is very little data remaining in the residuals. With the exception of the single significant lag at lag=24, the rest of the ACF of the residuals appears to be non-significant.

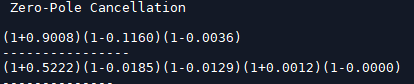
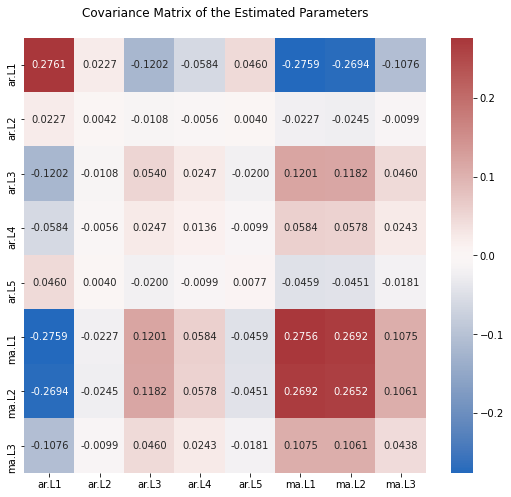


The covariance matrix of only one parameter shows the covariance only with itself, and is not very useful. The zero-pole cancellations does not reveal any roots to cancel, largely because the model is an AR model, rather than a full ARMA model, lending less opportunities for cancellations.

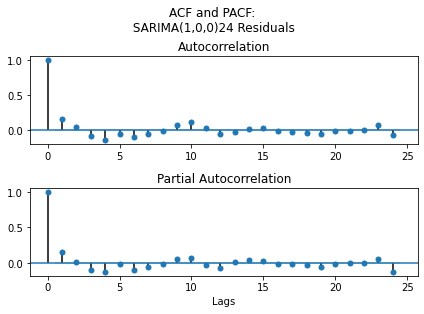
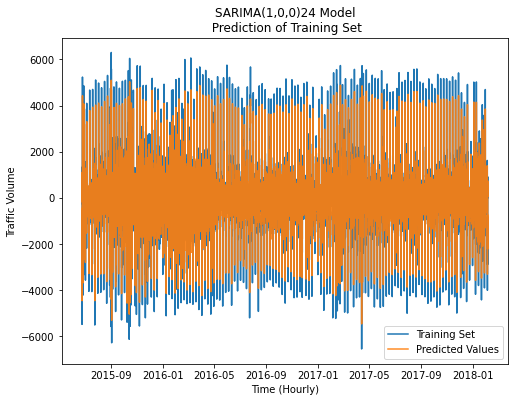


The prediction of the training set for the ARMA(5,3) model understandably looks very similar to the ARMA(1,0) prediction plot. As previously established, only one of the coefficients of the ARMA(5,3) models is significant, which will likely lead to it resembling and ARMA(1,0) model. Nonetheless, there is a difference between the two. The ARMA(5,3) model appears to be slightly better at predicting the extreme values compared to the ARMA(1,0) model.

The difference in the ACF/PACF plots is slightly more pronounced, with the lags between the significant 10 and 24 lags being more consistently insignificant. The lag at 24 may indicate that there is still seasonal information remaining in the residuals.

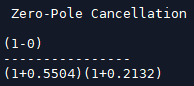
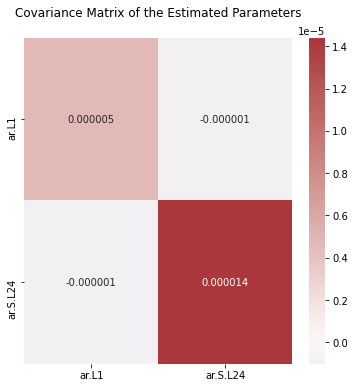


The zero-pole cancellation for this AR model may exhibit cancellable roots, as the 1-0.0036 zero and the 1+0.0012 pole are similar enough to cancel. If this model were to proceed as the selection for the final model, these would need to be cancelled out.



The SARIMA(1,0,0)24 model’s prediction of the training set superficially resembles those of the other two ARMA models tested. Like the other two, it has difficulty predicting the extreme values of the dataset, but consistently predicts the majority of the data.

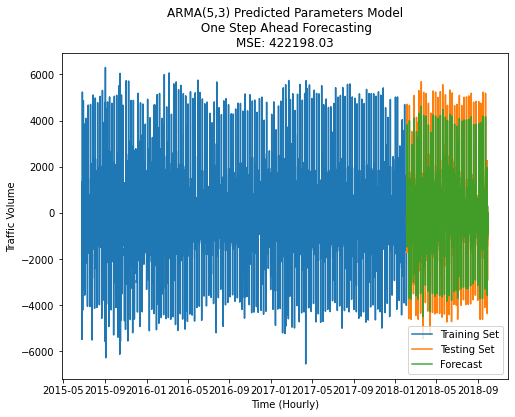
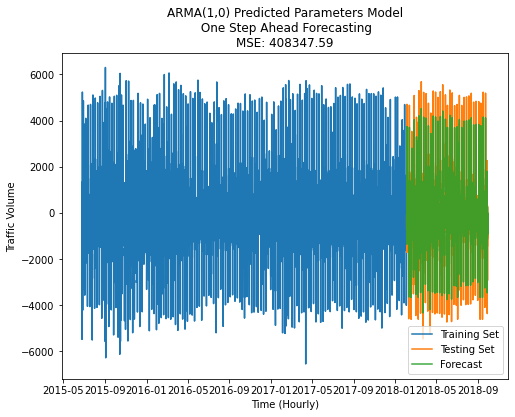
The ACF/PACF of the residuals appear noisier than the other two models, with some oscillatory behaviour in the ACF plot. The lag at 24 does not appear to be significant in the residuals, indicating that seasonality was captured by the model.

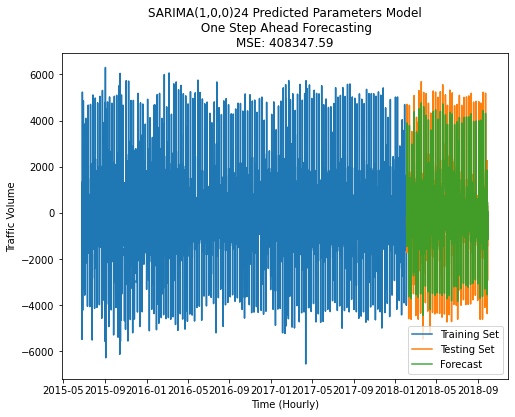


In the covariance matrix it is interesting to observe that the non-seasonal coefficient ar.L1 does not covary at all with the seasonal component ar.S.L24. This may be due to their seasonality, or perhaps due to the large lag difference between them.

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

Final Model selection





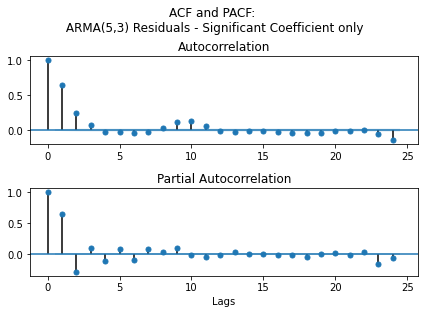
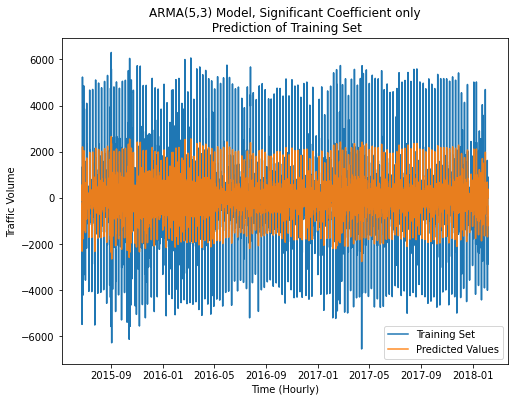
To make an even comparison to the basic models, a one-step forecasting was performed for all the ARMA models for the length of the test set, and these were plotted for visual comparison. The MSE of the forecasting was tabulated and compared.

|  |  |  |
| --- | --- | --- |
| Model | Q-Score | MSE – One Step Forecasting |
| Average | 29,200 | 1,053,861 |
| Naive | 2,161 | 1,053,887 |
| Drift | 2,146 | 1,062,540 |
| SES | 10,514 | 3,119,048 |
| Holt- Winter | 9,218 | 1,404,379 |
| OLS | 22,974 | 884,474 |
| ARMA(1,0) | 1,633 | 408,347 |
| ARMA(5,3) | 590 | 422,198 |
| SARIMA(1,0,0)24 | 2,428 | 408,347 |

Of the basic models, SES has the highest MSE of the forecasting, indicating the worst fit. This is followed by the Holt-Winter model, and the Drift model. The Average and Naïve model performed nearly identically. The OLS model performed the best of the simple models, with a MSE of 884,474.

Of the ARMA models, the ARMA(1,0) model performed the best, with an MSE of 408,347, compared to the ARMA(5,3)’s similar MSE of 422,198.

To select our final model, we must also take into account the Q-scores which indicate the whiteness of the residuals, and to an extent the amount of information that is not being captured by the model. The SARIMA model, despite capturing the seasonal component, had the highest Q-score of the three ARMA models, indicating the most information loss. While it is tempting to see the lower Q-score of the ARMA(5,3) and select it as the final model, recall that only one coefficient of the 8 was significant. If we perform a manual one-step prediction of the model using only the significant coefficient, we obtain a dramatically different result.



The Q-score obtained from this model is 12,014, an order of magnitude higher than the ARMA(1,0) model.

Therefore, with the lowest MSE of the forecast errors, and the smallest Q-score of the residuals, we select ARMA(1,0) as our final model.

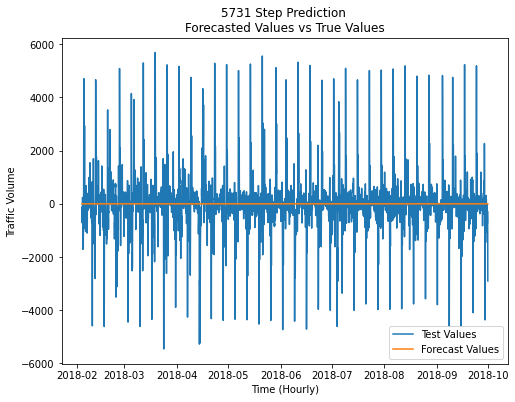
Forecast Function

The Forecast function for the ARMA(1,0) model can be written as:

From this, we can derive the 1-step prediction as:

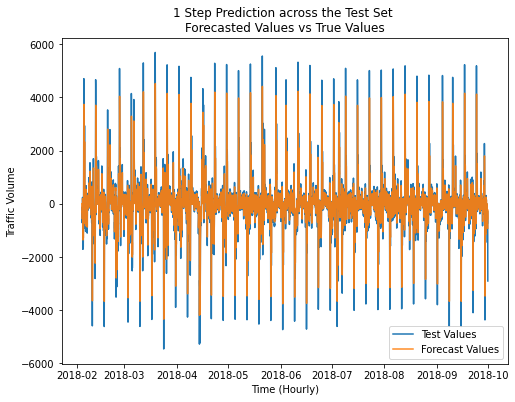
And the h-step prediction as:

h-step ahead Predictions



Performing a forecast function for the full test set, functionally a 5731-step prediction, yields a prediction that rapidly trends to 0. This shows that a simple ARMA(1,0) model is not appropriate for distant approximations.

However, using the same h-step prediction function to iteratively perform one-step predictions over the test set yields the following:



The one-step ahead prediction over the testing set produces a much more accurate model than the 5731-step prediction.

Summary and conclusion

References

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