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

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.

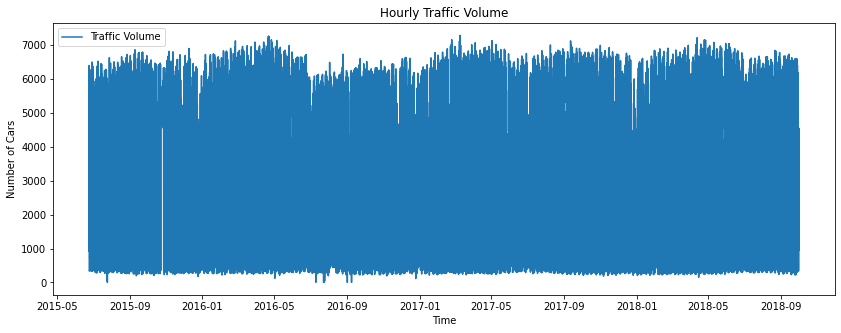
Several of the weather columns were grouped together into weather categories, to provide additional information to the model, with the expectation that a consolidated column may be more informative. An example of such a column would be the ‘Precipitation’ feature, which combined ‘Thunderstorm’, ‘Drizzle,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

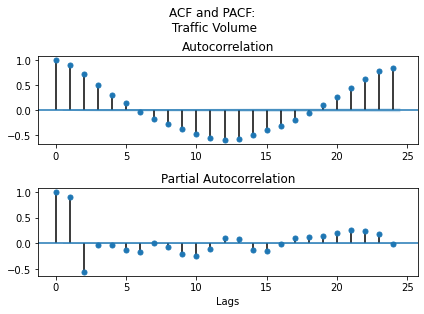
Finally, a boolean weekday/not weekday column was created using the timestamps of the hourly traffic report. This feature was created, as it seemed likely that traffic patterns would be strongly influenced by the day of the week, and thus if commuters would be largely present on the road.

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.

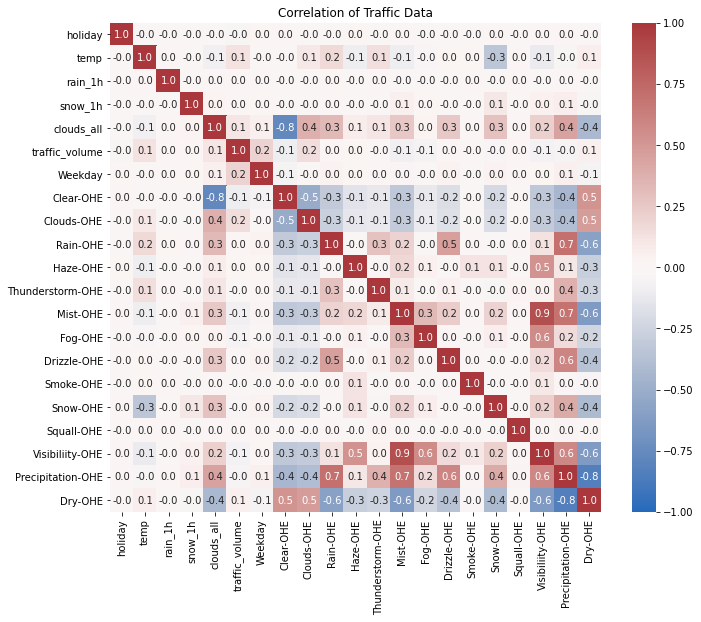
After Finalizing these features, our dataset contained 28,676 time points and 23 unique features.



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.

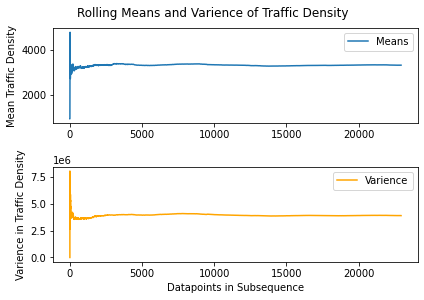
There is also strong negative correlation between the number of clouds and the clear sky columns, indicating that with higher cloud counts there is lower likelihood to report a clear sky.

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.

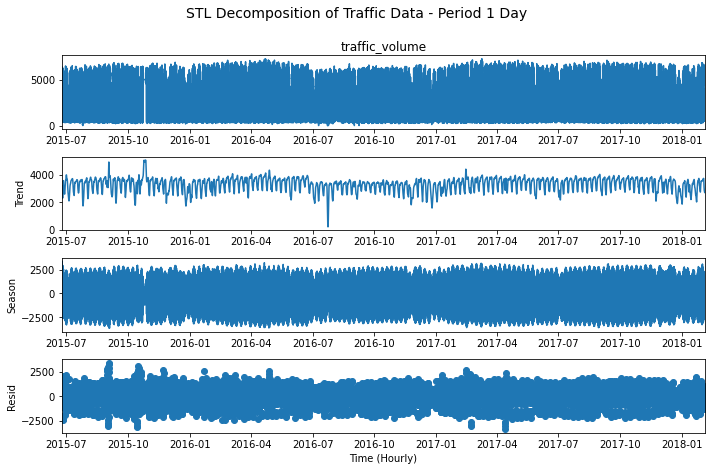


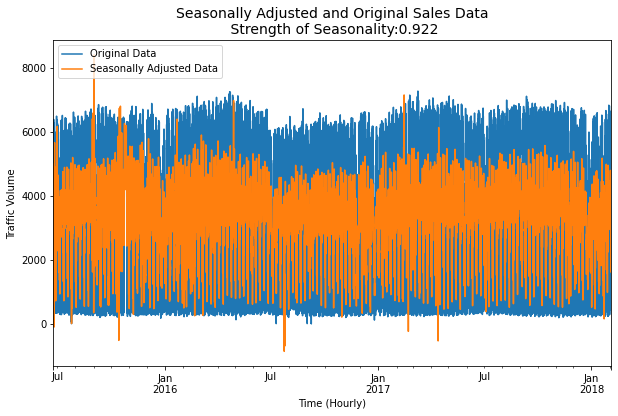
Examining the means and variance of the traffic volume, incrementally adding datapoints in the sequence, it appears that the data is stationary. This can be seen in the plots above, as the means and variance are remarkably stable over time after the initial few datapoints.

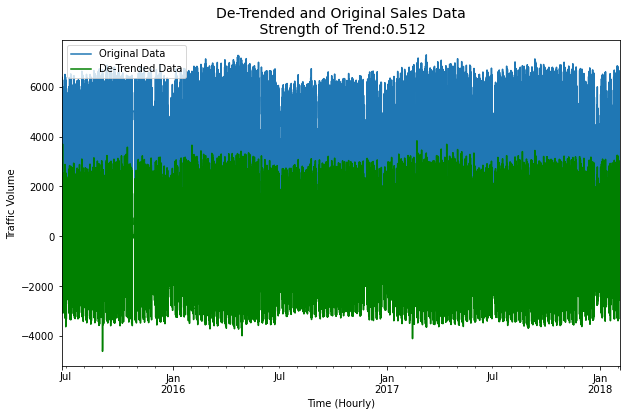
|  |  |
| --- | --- |
| ADF Test Results | |
| ADF Statistic | -18.245903 |
| p-value | 0.000000 |
| 95% Confidence | Significant |
| 99% Confidence | Significant |

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







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, prefentially 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 feaure 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 cloudcover 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

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

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