Kapil Bastola

4/23/2017

Assignment 13

## Abstract

Predicting future traffic is an integral part of a smart city. Using dataset from a smart city at Aarhus, Denmark, we try to predict future traffic flow from historical datasets. Traffic congestion, in form of average speed of traffic, prediction models were created using supervised learning and time series analysis. A seasonal and non-seasonal auto regressive integrated moving average (ARIMA) models were built along with a decision trees. Accuracy of the predictions shows that seasonal ARIMA model is the best model to be used for prediction of traffic flow form historical data.

## Work by Competitor

The competitor article chosen to compare against is the article titled “Short-Term Trafﬁc Flow Forecasting: An Experimental Comparison of Time-Series Analysis and Supervised Learning” by Lippi, M., Bertini, M., & Frasconi, P published in *IEEE Transactions on Intelligent Transportation Systems.* The article we have selected presents an experimental view of the different statistical & machine-learning approaches to short-term traffic-flow forecast. They follow the approach of seasonal ARIMA (SARIMA) and have proposed 2 new Support Vector Regressor (SVR) models using a seasonal kernel to determine the similarity with the time-series examples. The results that they present confirm that seasonality is the key feature to achieve a high-accuracy. The competitor’s article compares the accuracy using metric called mean absolute percentage error or MAPE. Though, the more accurate models usually require high computational resources – both while the training and prediction phase. Therefore, the seasonal kernel approach may be a reasonable compromise between the forecast accuracy and computational complexity. The SARIMA version that doesn’t include a Kalman filter and the ANNs performed worst than SVR with an RBF kernel. This in-turn is less accurate than seasonal kernel variant. Furthermore, another important direction of the research paper that has been indicated by the experimental results presented in this paper consists of investigating the covariate shift in traffic.

The competitor has presented an extensive experimental review of many statistical and machine-learning approaches to short-term traffic flow forecasting. Following the approach in SARIMA, they proposed two new SVR models: employing a seasonal kernel to measure similarity between time-series examples. They presented results confirm that seasonality is a key feature in achieving high accuracy; however, the most accurate models often require high computational resources both during the training phase and at prediction time. For this reason, they presented that seasonal kernel approach might be a reasonable compromise between forecasting accuracy and computational complexity issues. In particular, while SARIMA employed in combination with the Kalman filter ends up being the best model on average, the competitors proposed approach is particularly competitive when considering predictions during highly congested periods. The SARIMA version that does not include a Kalman filter and the ANNs perform consistently worse than SVR with an RBF kernel, which, in turn, is less accurate than the seasonal kernel variant. The competitor’s result show that the accuracy of the Seasonal Mean (SM) predictor starts degrading when the temporal distance between training and test set grows too much, and for the other predictors, no further improvement is observed when using larger training sets including past months, which are too distant from prediction time.

## Contribution

There are three important numerical values present in the dataset. One is the time taken to travel between two points, next is the average time it takes the vehicles to travel from one point to another and the third is the number of vehicles that travel between the two points for each reading.

First thing WEwould like to do with the dataset is to look at the descriptive statistics of the different fields in the dataset. In particular, for the numerical fields like Average time taken, Average speed, or Number of vehicles, we will look at the histograms of each of these field to see the distribution of values. Datasets consists of 9 different columns with a mix of numerical, categorical and date time variables. With SQL and R in zeppelin, we will look at the summary of each variable. For categorical variables we will look at the number of factors or levels in the variables. For the numerical variables we will look at the distribution and variation of each variable along with measures of central tendencies like mean and median. For the numerical variables we will also look at any correlation that might exist between two variables. This will help me in sorting out the variables. we will look at histograms for distribution and scatterplots to evaluate correlations between numerical variables.

In order to perform supervised learning algorithms, we need to train the dataset using labels. In this dataset there are no labels present. Average speed column in the dataset can be used to label the dataset as speed represents how fast the cars are going and this tells us how congested the road is. If we want to predict traffic congestion, velocity of cars in the road can be considered the label we want to predict.

Since the data we have is time based, we can conduct time-series analysis. In order to do the time series analysis, first we will plot the entire time series to see if there is any trend in the dataset and to check for any seasonality in the dataset. When we look at the time stamp field in the datasets, there are data points for every five seconds. We can use the velocity data to predict future velocity or number of vehicles at a given time using time series models like Auto-Regressive Integrated Moving Average (ARIMA) model. we will use the auto correlation function (ACF) and partial auto correlation function (PACF) to determine which order of auto regression and moving average is required for the model. By predicting velocity at a certain time we can make suggestions as to what time are busy and when to be careful when using the roads.

After building the models we can look at the accuracy of the different models created to find the model that performs the best.

## Data

The dataset consists of data from traffic in a city called Aarhus in Denmark. This dataset is collection of traffic data between two points for certain duration of time in CSV format for different durations. A CSV metadata file is also available that provides additional information regarding the different two points. Upon further analysis it was found that the dataset has 9 columns.

The **status** column has “OK” in all of the first 15 lines that we pulled. we wanted to see if there are any other values for this field. we looked at all the distinct values in the status column and saw that “OK” is the only value in the whole column.

**avgMeasuredTime** column has numerical values. The two sensors on two points of the road measure how long it took a vehicle in seconds to reach the second point from the first point. This field gives us the mean of total time taken in seconds by different vehicles to reach from the first point to the second point for each reading.

**avgSpeed** column also consists of numerical values. This column provides the average speed of vehicles between the two points in kilometer per hour (kmh).

**extID** column consists of 3-4 digit numerical values. Initially we was not certain what this column represented so we looked at all the distinct values in the column. we saw that they are sequential numbers and there are total of 449 distinct values in the table. Since we had 449 total files in our dataset, extID is a unique identifier for each file.

**medianMeasuredTime** column also consists of numerical values and has similar values as the avgMeasuredTime in the first 15 data points we looked at. This column gives us the median of total time taken in seconds by different vehicles to travel between the first and the second point for each reading.

**TIMESTAMP** column consists of date and time values and gives us the date and time of each reading.

**vehicleCount** column consists of numerical values. For each reading there are multiple vehicles passing between the two points. This column gives us the number of vehicles that travel between the two points during the readings.

**\_id** column consists of numerical values as well. In the first 15 rows, there are 6 digit numerical values that are all different. we wanted to count the distinct number of values in the \_id column and saw that the number of distinct values in this column is equal to the count of rows in the table. This suggests that \_id is the unique identifier for each row of data.

**REPORT\_ID** column consists of numerical values as well. In the first 15 rows, all the values were same in this column. Hence, we looked at the count of distinct values and found that there are 449 total unique values in the column. This suggests that this is an identifier for each file as well. In the metadata provided in the website we see that there is a column with same name and the values as the report\_id in the dataset. Hence, report\_id can be used to join the data set with the metadata file to obtain more information on each of the reading that took place.

The metadata file is a single .csv file with more information on the data streams. It has 449 rows of data implying that each row corresponds to each file in the dataset. The metadata file has information on where exactly the two points were. It contains information like street, city, latitude, longitude, postal code, and country for the two points. Apart from this it also contains ext\_id and REPORTID. These columns were present in the data set as well. Upon further look, ext\_id in the metadata file did not match the ext\_id present in the datasets but the REPORTID were same in the both file. So we can use the REPORTID in the metadata file with the REPORTID in the dataset to join the two tables if the need arises.

## Method

The flowchart outlines below shows the high level method going to be used in this capstone project.

**Data Retrieval, Combining** **and** **Labeling**

After retrieval and combining of the dataset, in order to perform supervised learning algorithms, we need to train the dataset using labels. In this dataset there are no labels present. Average speed column in the dataset can be used to label the dataset as speed represents how fast the cars are going and this tells us how congested the road is. If we want to predict traffic congestion, velocity of cars in the road can be considered the label we want to predict.

**Machine Learning Algorithms/ Predictive Models**

Machine learning algorithms can be generalized into two different types, supervised learning and unsupervised learning. In supervised learning, we train a model for each input with a corresponding target and later predict target for any new input. If the targets are in distinct classes we call it a classification model and if the target is continuous we call it a regression model. Where as in unsupervised learning there are no targets. We evaluate the relationship between different inputs and their structure in unsupervised learning. One of the most important unsupervised learning methods is clustering; where we group input data based on the inherent structure of those inputs and build a model to place a new input data in one of the groups created.



Figure : This image outlines the process flow of the method. We use supervised learning and time series analysis to predict traffic congestion

For the purpose of the study we will look into following machine learning algorithms:

**Decision Tree**

Decision tree is an algorithm used for building classification models whose output looks like a tree structure. Decision tree consists of root node, test node and decision nodes (leaf node). A decision tree is a flowchart-like structure in which each internal node represents a test on an attribute, each branch represents the outcome of the test and each leaf node represents a class label, which is the decision taken after computing all attributes. The paths from root to leaf represent classification rules. In decision analysis a decision tree and the closely related influence diagram are used as a visual and analytical decision support tool, where the expected values of competing alternatives are calculated. A decision tree classifies data by following the route from root node to the decision node according the set attributes or criteria. Once the decision tree is build we start at the root node to apply the testing scenario and follow the branch that fits the scenario, ending up at one of the leaf nodes which are one of the classes of the classification model. We can use Rpart package in R through zeppelin to perform decision trees algorithms.

**Time Series Analysis: Autoregressive Integrated Moving Average (ARIMA)**

Since the dataset we are looking at is time based we can conduct time series analysis to predict future points in the series. We can also look into seasonality as we may guess that the traffic data changes during the course of the day and the week. We can use the velocity data to predict future velocity or number of vehicles at a given time using time series models like Auto-Regressive Integrated Moving Average (ARIMA) model. The auto correlation function (ACF) and partial auto correlation function (PACF) can be used to determine which order of auto regression and moving average is required for the model. By predicting velocity at a certain time we can make suggestions as to what time are busy and when to be careful when using the roads. If the ACF shows any significance at any of the lags we use that order of lag for the moving average and if PACF shows any significant spikes at any of the lags we use that order of lags for the auto regression. We can also look at the ACF and PACF plot to determine if there is any seasonality in the dataset and whether seasonality should be considered or not. After we set the order for the model we can generate the model using R through Zeppelin. We can then predict next series of data points based on the model. In order to predict the accuracy of the model we can use mean absolute percentage error (MAPE).

## Results

After downloading and loading the data into zeppelin, first step taken was to understand the data and descriptive statistics behind it. Looking at the different variables present in the dataset average speed is chosen as the variable to mimic traffic flow since higher average speed means better flow of traffic and lower average speed means traffic flow is not very good. Time of day is one of the biggest factor in determining traffic flow as more vehicle are in the roads during day hours, especially during commuting hours, compared to night time. When we look at the hourly average speed in our dataset to verify the previous statement we can confirm that the average speed is considerably lower between 6 am and 2 pm. We also looked at vehicle count by hour and found that most vehicles are in the roads between 6 am and 2 pm. Both these results show that we are in right path to use time of day and vehicle count to predict average speed in the dataset. Scatter plot of average speed versus vehicle count shows a negative correlation between the two variables, as expected.

After the initial descriptive analysis we start out with a simple decision tree to predict average speed based on two variables hour of the day and vehicle count.

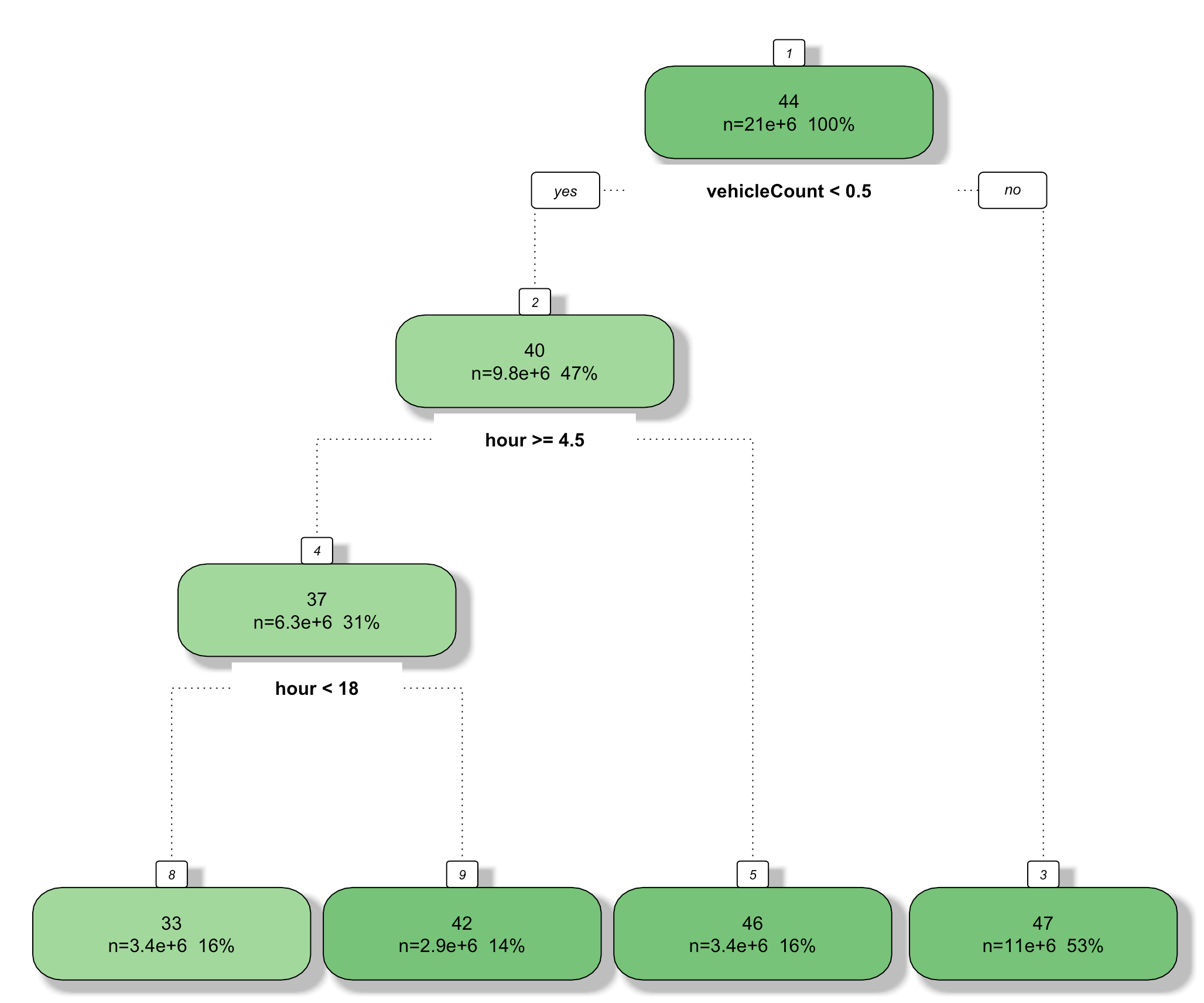


Figure : Decision tree representation for traffic flow data showing different nodes and features

We got a decision tree as shown in the picture above. First node of the decision tree is whether the vehicle count is higher than 0.5 or not. If it is higher than 0.5, average speed is 47. If the vehicle count is lower than 0.5 we encounter another node, which examines whether time of day is greater than 4 am or not. If the time of day is less than 4 am, average speed is 46. If the time of day is greater than 5 am we encounter another node that examines whether time of day is after 6 pm or not. If time of day is after 6 pm the predicted average speed is 42 kph and if it is before 6 pm and after 5 am, the predicted average speed is 33 kph. We looked at the error of this model and saw that the average MSE is 478.

Next, we wanted to do time series analysis on the average speed. We first aggregated the 20 million rows data into hourly data for the time series analysis. After sorting the data by day and hour, we have an ordered tie series data that can be used for time series analysis. Using R and forecast library, we use auto.arima function to run an auto regressive integrated moving average (ARIMA) model to create a fit that can be used to predict future data. In order to use the ARIMA model we need to provide which level we want to set the model at. We can utilize just the auto-regression (AR) part of the model by setting to a numerical value or we can just set the moving average (MA) portion of the model.

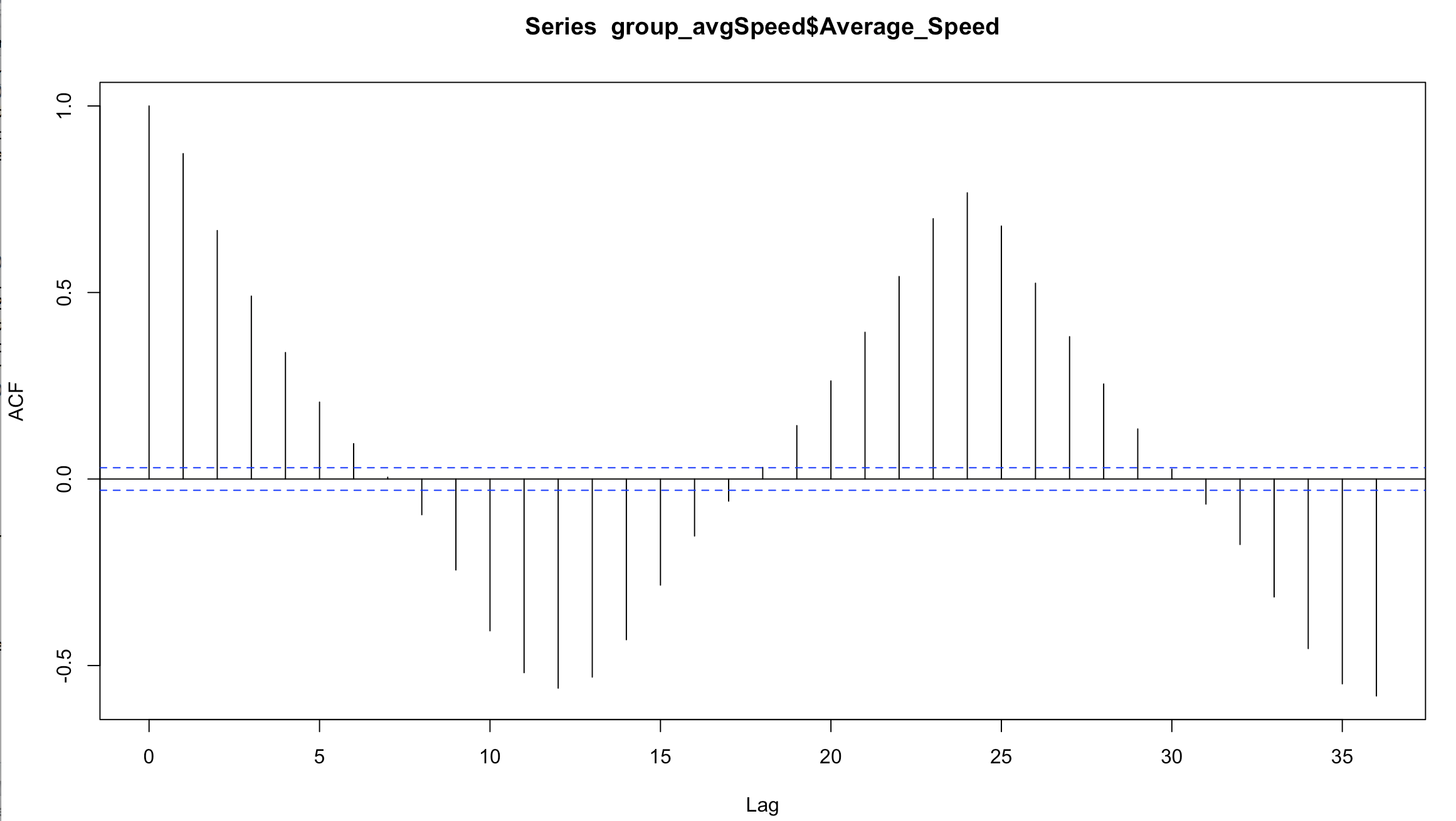


Figure : Auto Correlation Function (ACF) of the hourly data. This shows significant moving average lag at lag 7 and significant seasonality at lag 18.

In order to determine which part of the model to choose and at what orders, we look at the auto correlation function and partial auto correlation function. Auto correlation function (ACF) gives us the correlation between a time series and the lags of itself while the partial auto correlation function (PACF) gives us the correlation of a time series with its lag that is not explained by correlation at lower order lags. If the ACF shows any significance at any of the lags we use that order of lag for the moving average and if PACF shows any significant spikes at any of the lags we use that order of lags for the auto regression. After we set the order for the model we can generate the model using R. We can then predict next series of data points based on the model. In order to predict the accuracy of the model we can use mean absolute percentage error (MAPE). Looking at the value of MAPE will tell us whether our model is highly accurate, good, reasonable or inaccurate.

Auto.arima function in the forecast library selected an order of 4 for AR, 1 for I, and an order of 4 for MA. We got a Mean Absolute Percentage Error (MAPE) of 2.52. We also predicted the last 12 values of the time series. When comparing the actual versus the predicted and the variation in the actual speed is not as prevalent in the predicted speed. Also looking at the Ljung-Box test for significance, we see that there are first few lags that are not significant.

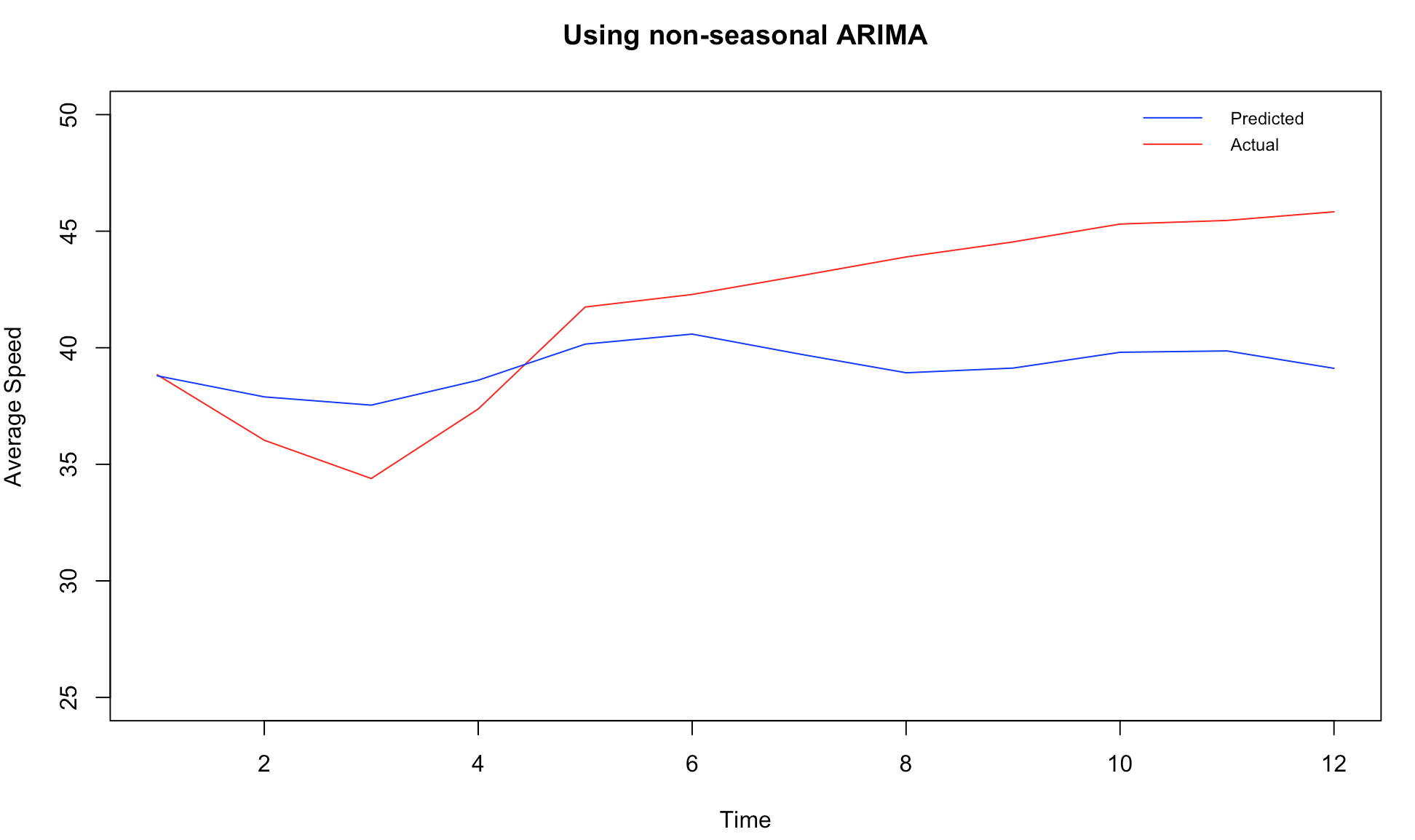


Figure : Actual versus predicted result from non-seasonal ARIMA. This shows how predicted values do not follow actual values after 5 or so points

We wanted to pick our own orders for AR and MA by looking at the ACF and PACF. Looking at the ACF and PACF we see that there is some seasonality in the data and decided to add a seasonality component to the ARIMA and performed an ARIMA order of (0,2,7) with a seasonality of (0,0,18). Using these orders, we reran the ARIMA fit and predicted the last 12 hours of average speed of the time series again. For this model we got an MAPE of 2.19, which is an improvement from the previous time series model. For the 12 hours of predicted average speed versus the actual average speed we see that the variation in the actual values is represented in the predicted values as well as seen in the figure below. When looking at the Ljung-Box test for significance, we see that all the lags has p-values less than or close to 0.05 meaning a statistically significant result. This tells us that the seasonal ARIMA is a better model compared to the non-seasonal one.

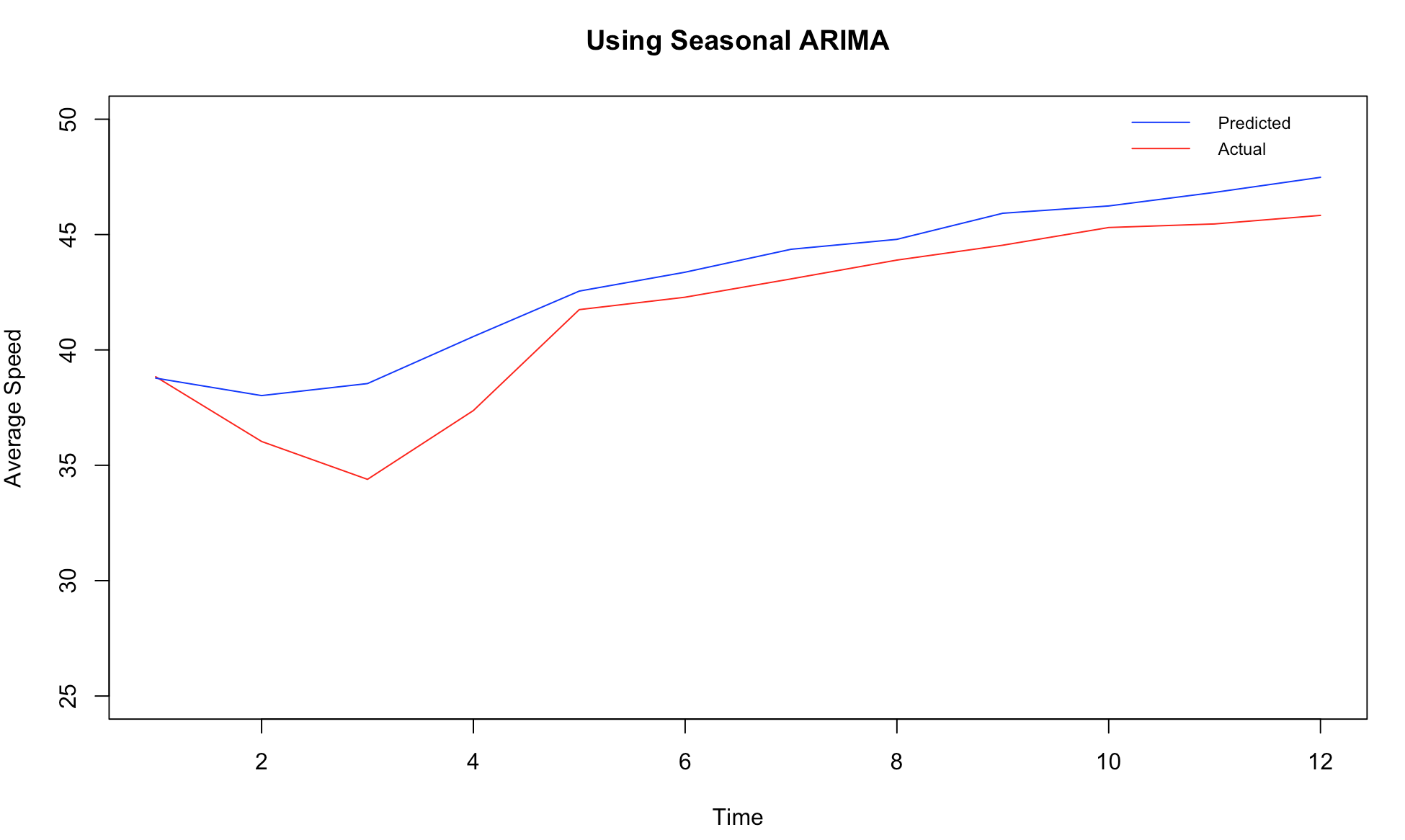


Figure : Actual versus predicted result from seasonal ARIMA. This shows how predicted model closely follows actual result

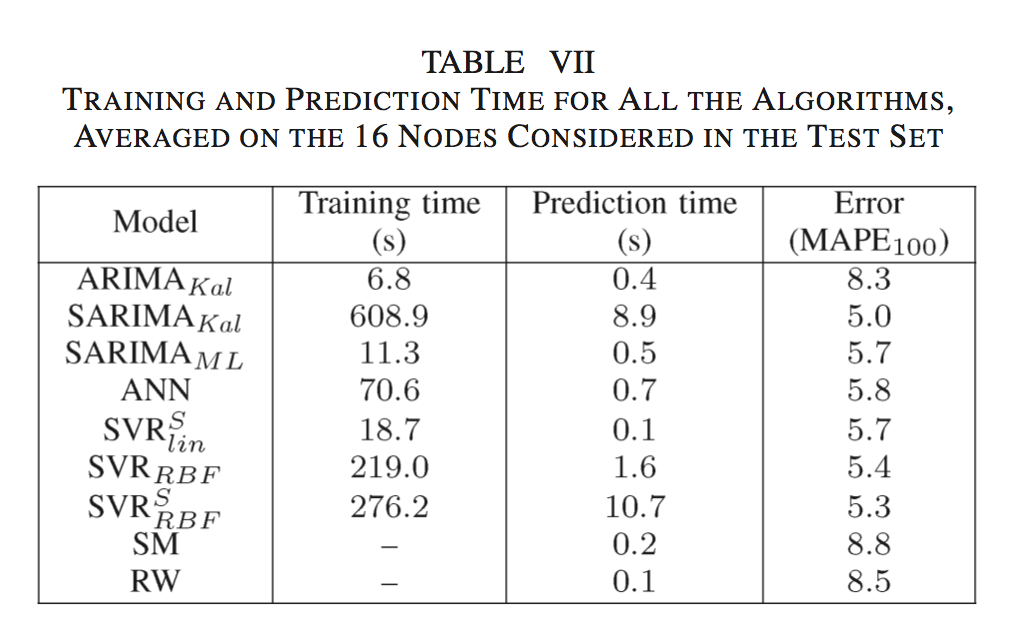
## Discussions

The competitor article we chose to compare against is the article titled “Short-Term Trafﬁc Flow Forecasting: An Experimental Comparison of Time-Series Analysis and Supervised Learning” by Lippi, M., Bertini, M., & Frasconi, P published in *IEEE Transactions on Intelligent Transportation Systems.* In this article the authors first reviewed existing approaches to traffic flow forecasting the common view of probabilistic graphical models, presenting an extensive experimental comparison, which proposes a common baseline for their performance analysis and provides the infrastructure to operate on a publicly available data set. Then the authors provide two new support vector regression models, which are speciﬁcally devised to beneﬁt from typical trafﬁc ﬂow seasonality and are shown to represent an interesting compromise between prediction accuracy and computational efﬁciency.

For the result the competitor did eight different models, which were either supervised learning or time series analysis. The set of tested competitors consists of the following algorithms:

* + RW, which is a simple baseline that predicts trafﬁc in the future as equal to current conditions
  + SM, which predicts for a given time of the day the average in the training set;
  + ARIMA model with Kalman ﬁlter
  + SARIMA model with maximum-likelihood ﬁtting
  + SARIMA model with Kalman ﬁlter
  + ANNs
  + Support Vector Regressor (SVR) with RBF kernel
  + SVR with RBF kernel multiplied by a seasonal kernel
  + SVR with linear seasonal kernel

The authors found that the Seasonal ARIMA (SARIMA) model coupled with a Kalman ﬁlter is the most accurate model. However, when forecasting during the most congested periods the seasonal support vector regressor (SVR) was found to be highly competitive method as well.

we did three different models with one supervised learning model with the decision tree and two time-series analysis model with one seasonal ARIMA and one non-seasonal ARIMA. For the time series analysis and the supervised learning algorithms, the competitor use Mean Absolute Percentage Error (MAPE) to evaluate the accuracy of their models while computing the time taken to run the model. The time taken to run the model between the competitor and our dataset cannot be compared since we are not sure how may data points there were in the competitor’s dataset but we can look at MAPE to evaluate how our model stands with respect to the competitor’s model. The competitor, in their article, presented a table with the MAPE for each model as shown in the figure below:

The table from the competitor article shows that the lowest mean absolute percentage error they got was for a Seasonal ARIMA or SARIMA model with MAPE of 5.0. Both of our time series models performed better than the best Seasonal ARIMA from the competitor.

First we let R select the best order for the auto regressive integrated moving average (ARIMA) model and it picked an order of (4, 1, 4) with no seasonality. We got a Mean Absolute Percentage Error of 2.52. We also predicted the last 12 values of the time series. When comparing the actual versus the predicted and the variation in the actual speed is not as prevalent in the predicted speed. Also looking at the Ljung-Box test for significance, we see that there are first few lags that are not significant. We wanted to pick our own orders for AR and MA by looking at the ACF and PACF. Looking at the ACF and PACF we see that there is some seasonality in the data and decided to add a seasonality component to the ARIMA and performed an ARIMA order of (0, 2, 7) with a seasonality of (0, 0, 18). Using these orders, we reran the ARIMA fit and predicted the last 12 hours of average speed of the time series again. For this model we got an MAPE of 2.19, which is an improvement from the previous time series model. Both MAPE of models that we created were better than the best MAPE of the competitor’s model.

We also performed a supervised learning model with a decision tree to predict average speed based on two variables hour of the day and vehicle count. A decision tree is a flowchart-like structure in which each internal node represents a test on an attribute, each branch represents the outcome of the test and each leaf node represents a class label, which is the decision taken after computing all attributes.

First node of the decision tree resulted is whether the vehicle count is higher than 0.5 or not. If it is higher than 0.5, average speed is 47 kph. If the vehicle count is lower than 0.5 we encounter another node, which examines whether time of day is greater than 4 am or not. If the time of day is less than 4 am, average speed is 46. If the time of day is greater than 5 am we encounter another node that examines whether time of day is after 6 pm or not. If time of day is after 6 pm the predicted average speed is 42 kph and if it is before 6 pm and after 5 am, the predicted average speed is 33 kph. We looked at the error of this model and saw that the average MSE is 478. This is a high error and hence we decided to focus mainly on the time-series analysis, as we do not have many variables we can use.

## Conclusion

As the competitor we also found that Seasonal Auto Regressive Integrated Moving Average (SARIMA) model is best model to predict traffic flow as we observed the least amount of mean absolute percentage error using this model. Using supervised learning models like the decision trees was not the best way to approach the dataset we had because of the lack of variables and we observed a high error in the decision tree’s result. The seasonal ARIMA and ARIMA model that we built had a much lower MAPE compared to the competitor. Since the dataset the competitor’s article used and we used are different with different variables and attributes, it may be difficult to compare the two results but only looking at how the model we built predicted the results compared to the actual values, we can say that seasonal auto regressive integrated moving average (SARIMA) is the best predictor of traffic flow.

## Reference

1. Lippi, M., Bertini, M., & Frasconi, P. (2013). Short-Term Traffic Flow Forecasting: An Experimental Comparison of Time-Series Analysis and Supervised Learning. *IEEE Transactions on Intelligent Transportation Systems*, 14(2), 871-882. doi:10.1109/tits.2013.2247040
2. Steinley, D., & Brusco, M. J. (2011). Choosing the number of clusters in Κ-means clustering. *Psychological Methods*, *16*(3), 285-297. doi:10.1037/a0023346
3. Guha, S., Rastogi, R., & Shim, K. (2001). Cure: an efficient clustering algorithm for large databases. *Information Systems*, 26(1), 35-58. doi:10.1016/s0306-4379(01)00008-4
4. Kumar, S. V., & Vanajakshi, L. (2015). Short-term traffic flow prediction using seasonal ARIMA model with limited input data.*European Transport Research Review, 7*(3), 1-9. doi:http://dx.doi.org/10.1007/s12544-015-0170-8
5. Smith, B., & Demetsky, M. (n.d.). Short-term traffic flow prediction models-a comparison of neural network and nonparametric regression approaches. *Proceedings of IEEE International Conference on Systems, Man and Cybernetics.* doi:10.1109/icsmc.1994.400094
6. Zhao, Z., Chen, W., Yue, H., & Liu, Z. (2016). A novel short-term traffic forecast model based on travel distance estimation and ARIMA. *2016 Chinese Control and Decision Conference (CCDC)*. doi:10.1109/ccdc.2016.7532126
7. Business, F. S. (n.d.). Identifying the orders of AR and MA terms in an ARIMA model. Retrieved February 07, 2017, from http://people.duke.edu/~rnau/411arim3.htm
8. Li, R., Jiang, C., Zhu, F., & Chen, X. (2016). Traffic flow data forecasting based on interval type-2 fuzzy sets theory. (2016). *IEEE/CAA Journal of Automatica Sinica*, 3(2), 141-148. doi:10.1109/jas.2016.7451101
9. Hou, Y., Edara, P., & Sun, C. (2015). Traffic Flow Forecasting for Urban Work Zones. *IEEE Transactions on Intelligent Transportation Systems*, 16(4), 1761-1770. doi:10.1109/tits.2014.2371993
10. Abadi, A., Rajabioun, T., & Ioannou, P. A. (2014). Traffic Flow Prediction for Road Transportation Networks With Limited Traffic Data. *IEEE Transactions on Intelligent Transportation Systems,* 1-10. doi:10.1109/tits.2014.2337238
11. Hou, Z., & Li, X. (2016). Repeatability and Similarity of Freeway Traffic Flow and Long-Term Prediction Under Big Data. IEEE Transactions on Intelligent Transportation Systems, 17(6), 1786-1796. doi:10.1109/tits.2015.2511156
12. Lv, Y., Duan, Y., Kang, W., Li, Z., & Wang, F. (2014). Traffic Flow Prediction With Big Data: A Deep Learning Approach. *IEEE Transactions on Intelligent Transportation Systems,*1-9. doi:10.1109/tits.2014.2345663
13. Koesdwiady, A., Soua, R., & Karray, F. (2016). Improving Traffic Flow Prediction With Weather Information in Connected Cars: A Deep Learning Approach. *IEEE Transactions on Vehicular Technology,*65(12), 9508-9517. doi:10.1109/tvt.2016.2585575
14. Xia, D., Li, H., Wang, B., Li, Y., & Zhang, Z. (2016). A Map Reduce-Based Nearest Neighbor Approach for Big-Data-Driven Traffic Flow Prediction. *IEEE Access*, 4, 2920-2934. doi:10.1109/access.2016.2570021
15. Yu, Y., Li, J., Guan, H., Wang, C., & Wen, C. (2016). Bag of Contextual-Visual Words for Road Scene Object Detection From Mobile Laser Scanning Data. *IEEE Transactions on Intelligent Transportation Systems,*17(12), 3391-3406. doi:10.1109/tits.2016.2550798
16. Xu, Y., Kong, Q., Klette, R., & Liu, Y. (2014). Accurate and Interpretable Bayesian MARS for Traffic Flow Prediction. *IEEE Transactions on Intelligent Transportation Systems,* 15(6), 2457-2469. doi:10.1109/tits.2014.231579
17. Chan, K. Y., Dillon, T. S., & Chang, E. (2013). An Intelligent Particle Swarm Optimization for Short-Term Traffic Flow Forecasting Using on-Road Sensor Systems. *IEEE Transactions on Industrial Electronics,* 60(10), 4714-4725. doi:10.1109/tie.2012.2213556
18. Chan, K. Y., & Dillon, T. S. (2013). On-Road Sensor Configuration Design for Traffic Flow Prediction Using Fuzzy Neural Networks and Taguchi Method. *IEEE Transactions on Instrumentation and Measurement*, 62(1), 50-59. doi:10.1109/tim.2012.2212506
19. Zhao, J., & Sun, S. (2016). High-Order Gaussian Process Dynamical Models for Traffic Flow Prediction. *IEEE Transactions on Intelligent Transportation Systems*,17(7), 2014-2019. doi:10.1109/tits.2016.2515105
20. Mo, H., Hao, X., Zheng, H., Liu, Z., & Wen, D. (2016). Linguistic Dynamic Analysis of Traffic Flow Based on Social Media—A Case Study. *IEEE Transactions on Intelligent Transportation Systems*, 17(9), 2668-2676. doi:10.1109/tits.2016.2530698