****

**Final Teleprompter**

**Vehicle Traffic, Provided by City of Aarhus in Denmark**

**Presented by:** Amit Pandey

**Guided by:** Professor Sylvain Jaume

**Course:** DS670 - Capstone

**Novel Contribution:**

We did the time-series analysis using ARIMA (Autoregressive integrated moving average) model. ARIMA is a generalization of an autoregressive moving average (ARMA) model, both of these models are fitted to time series data either to better understand the data or to predict future points in the series (forecasting). ARIMA models are applied in some cases where data show evidence of non-stationarity, where an initial differencing step can be applied one or more times to eliminate the non-stationarity.

We predicted the next 5 hours (data points) for the time-series using ARIMA and compared them with the actual.

**Description of Contribution:**

Conduct supervised learning as well as time series analyses to predict future traffic congestion based on historical data

Cross validate the models created using accuracy metrics and visualize results to find the model that performs the best

We did the time-series analysis using ARIMA (Autoregressive integrated moving average) model. ARIMA is a generalization of an autoregressive moving average (ARMA) model, both of these models are fitted to time series data either to better understand the data or to predict future points in the series (forecasting). ARIMA models are applied in some cases where data show evidence of non-stationarity, where an initial differencing step can be applied one or more times to eliminate the non-stationarity.

**Contribution of Competitor’s Article**

Competitor’s Article: “Short-Term Trafﬁc Flow Forecasting: An Experimental Comparison of Time-Series Analysis and Supervised Learning”

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.

The paper we’ve selected presents an experimental view of the different statistical & machine-learning approach to short-term traffic-flow forecast. They follow the approach of SARIMA and have proposed 2 new 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. 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 have 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

**Data Source and Content**

The data which has been used on this paper is belongs to a smart city on EU which called CityPulse. This 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.

Below were the fields in the dataset

* status,
* avgMeasuredTime,
* avgSpeed,
* extID,
* medianMeasuredTime,
* TIMESTAMP,
* vehicleCount,
* \_id, and
* REPORT\_ID.

First type is metadata. There is a separate metadata file for each individual city.

Second type of data is street traffic data. Each file represents individual street traffic information. Each line represents traffic information for every five

The dataset assigned for the capstone project is the “Vehicle Traffic, Provided by City of Aarhus in Denmark” from CityPulse dataset collection. A collection of datasets of vehicle traffic, observed between two points for a set duration of time over a period of 6 months (449 observation points in total). The data is available in raw (CSV) and semantically annotated format using the citypulse information model. This proposal looks at how the Vehicle Traffic data collected can be used in making Aarhus a smart city in terms of traffic guidance and management.

A collection of datasets of vehicle traffic in the city called Aarhus in Denmark, observed between two points for a set duration of time over a period certain months (449 observation points in total). The data is available in raw (CSV) and semantically annotated format using the Citypulse information model. There are total four (4) different datasets over different durations available:

1st data set: February 2014 to June 2014

2nd data set: August 2014 to September 2014

3rd data set: October 2014 to November 2014

4th data set: July 2015 to October 2015

The data is available in raw (CSV) and semantically annotated format using the citypulse information model. For this project, I’m going to use the “.csv” format only as it’s easy to manipulate and analyze the data. The data analysis will be done in Apache Zeppelin environment that is being taught in the class. To test out the data, I took the first dataset from February 2014 to June 2014 and saved all the files in a folder “traffic\_feb\_june” in my Downloads directory. Using Zeppelin Spark, I was able to consolidate all the files in one table called roadtraffic. Using SQL table, I ran a few queries to look into the data set and get a summary. I have only looked at the first dataset (Feb 2014 – June 2014) for now to do the analysis.

**Method**

We loaded the data into Zeppelin to clean and analyze it with the idea of stacking the data into the framework and clean it/investigate it. The idea was to invest as much energy as it could to set up the data into a strong base of a domain that could be valuable and simpler to do the model. This way it would be easier to analyze it simply. Thus, in view of that, the data initially has been stacked into Spark, which did not work out as efficient language for the data and the purpose of our model. We, therefore, changed it over the data outline into a SQL as it simple to deal with excessively numerous records. This turned out to be great to just analyze the data. However, we still had to deal with a prediction and analysis part. In this way, we inferred that SQL won't work, therefore we have attempted to stack the data into a PySpark, which is a competitor language amongst Spark and Python, and that where the PySpark originated from. This was a decent decision notwithstanding, because of the absence of data and information in how to deal with the programming part on the grounds that the linguistic structure must conform to both language Spark and PySpark, and because of the long run time that we have confronted on doing the stacking and the analysis, then we have chosen to do only python for loading, clean, analysis and prediction the data and the model.

1. **Collect and download the datasets into one folder.**

Based on the data provided for Aarhus, we will download and collect all the datasets in one folder. As the data is available in the raw (CSV) and semantically annotated format using the Citypulse information model, we will use the raw (CSV) format to process and analyze the ata.

1. **Load data and combine all data sets in Zeppelin.**

Using Zippelin, we will load and combine all the files (449) in the datasets to analyze the information. We have utilized numerous languages to the analysis. We have utilized: Spark, SQL PySpark and Python. The languages that we used to fit to the data were both PySpark and Python, in any case we have chosen to run with Python as it was anything but difficult to deal with a sentence structure for one language as opposed to taking care of the structure for two languages. What's more, the outcome for that part, the run time stacking into PySpark took twofold the run time stacking into Python. As Python was more proficiency in this part, why have chosen to finish with python.

Due to the vast data set we have chosen to take a major measure of the data to test and fabricate the model on it, as taking every one of the data won't be conceivable because of the capacities that we have. It is more than 29 million records and that without mapping and other data.

1. **Clean the data.**

We’ll did an analysis to see if the data needs to be cleaned and if there are any errors that may corrupt our analysis. We also wanted to make sure that the null values are set to zero and take care of missing values. We checked for common errors like: missing values, corrupted values, data range errors, etc. We looked through the file rows and columns and sample test values to see if the values make reasonable sense.

We have utilized the cleaning techniques to clean the data of N/A by either supplanting or barring from the example or notwithstanding expelling from the data if the record has an excessive number of missing qualities. What's more, it can't withdraw or supplanted. Likewise, we have examined the data to ensure that the data is not tainted for instance if the data has an immense out of the sudden spike on the chart that implies the data is ruined. Or, on the other hand if the data has some wrong esteem, for example, a period of 99 that implies it is ruined as there are no 99 clock time and the time goes just to 24 hours.

1. **Handling the outliers:**

In order to handle the outliers, what we have done here is we have rejected the anomaly from the data, as this will influence the data as we didn't evacuate it or erase it, as this can be utilized on further analysis. For instance, utilizing the outliers to anticipate the possibility of having an accident on this street. Nonetheless, on this case we truly need to ensure that the outliers are a result of an accident not due to absent or undermined values, or might be a result of street work in the city which cannot be utilized at this stage, with the goal that we have chosen to avoid the outliers from the data and account for the typical data to improve a creation.

1. **Split Datasets into Training (80%) and Testing Data (20%).**

As we are going to do some prediction, it we split our combined dataset into two different datasets: Training (80%) and Testing (20%). The Training dataset was used to build and test different statistical mode and the Testing dataset was used to evaluate (cross-validate) our model and assumption. We did the splitting and create the Test partition to provide us with a fair assessment of our prediction model build.

There was no single method for selecting the extent of training and testing data. Some people use 90/10 and some prefer 80/20. In any case, doing as such can cause bias the classification results. For selecting the right split, we used “N-Fold cross validation” and/or “K-fold cross validation. This will take out any bias out of our assumptions.

1. **Mapping the data:**

We have mapped the data into the Meta Data to have more data about the record that the framework caught and comprehend the data

1. **Final discovery:**

We did the time-series analysis using ARIMA (Autoregressive integrated moving average) model. ARIMA is a generalization of an autoregressive moving average (ARMA) model, both of these models are fitted to time series data either to better understand the data or to predict future points in the series (forecasting). ARIMA models are applied in some cases where data show evidence of non-stationarity, where an initial differencing step can be applied one or more times to eliminate the non-stationarity.

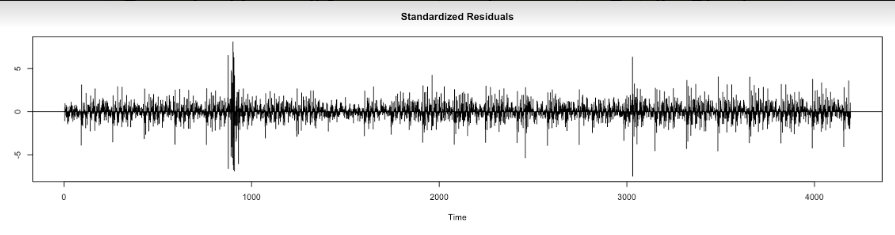
Using Spark, load and combine all the datasets

Using SQL and R, look at the descriptive statistics of the dataset. Split the dataset into training and test data sets for cross validation and evaluation of the models to be created

* Build different supervised learning models based on training datasets using different packages in R
  + Decision Trees
  + Time Series Analysis – Order (4,1,4)
  + Seasonal Time Series Analysis – Order (0, 2, 7) & Seasonality (0, 0, 18)
* Evaluate the models with different performance metrics to pick the best model that can be used for prediction
* Select the final model

**Quantitative Results**

We predicted the next 5 hours (data points) for the time-series using ARIMA and compared them with the actual. Here’s the result of the model:



**Chart 1**:For the existing data, this is the residuals of the predicted fits.

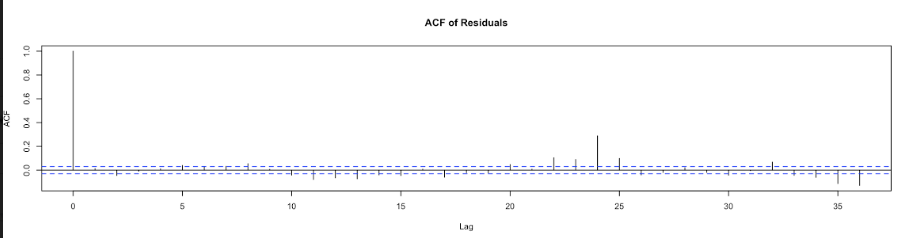
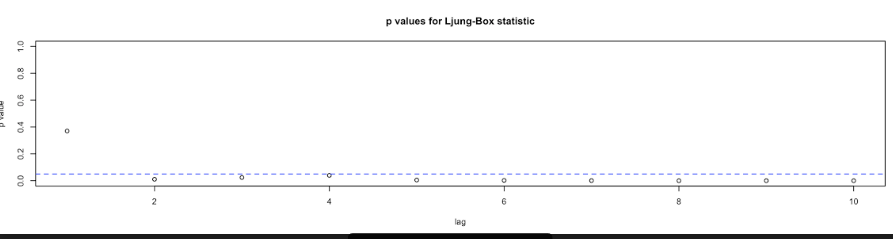


Chart 2: Autocorrelation function (ACF) of residuals shows the correlation of the residuals (as a time series) with its own lags.



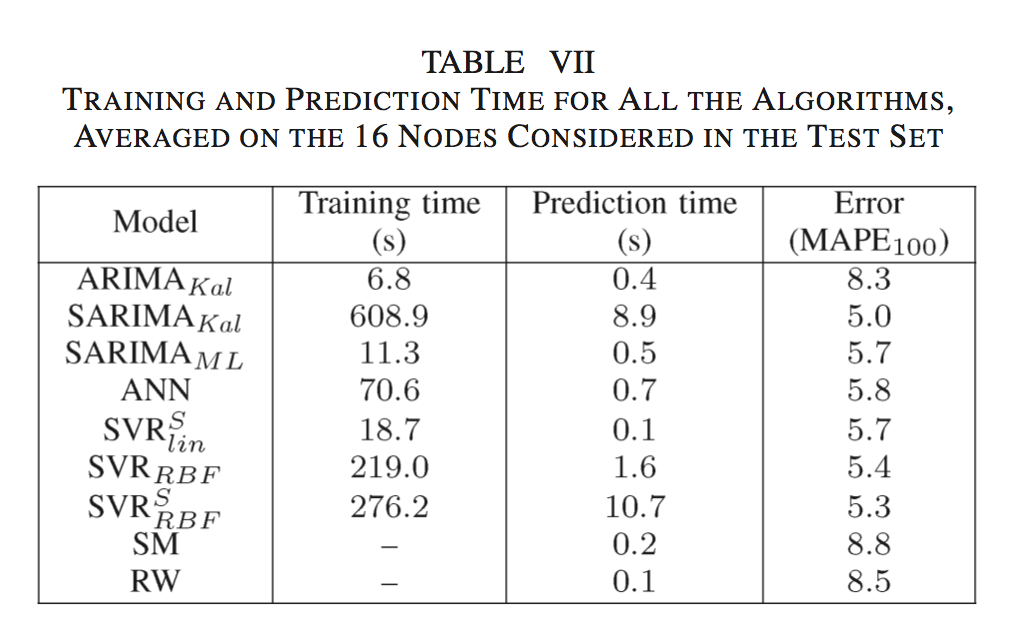
**Chart 3:** p-value showing the fitting of the model

Even though the competitor have used seven different models, their MAPE is coming out to be much higher, as shown in table 1:

The mean absolute percentage error (MAPE), also known as mean absolute percentage deviation (MAPD), is a measure of prediction accuracy of a forecasting method in statistics, for example in trend estimation.

**Comparison With Your Competitor**

**Competitor’s Results**

* The set of tested models by competitors consists of the following algorithms:
* 
* The table from the competitor article shows that the lowest mean absolute percentage error they got as for a Seasonal ARIMA or SARIMA model with MAPE of 5.0.

**Your results:**

* First we performed supervised learning with decision tree with two variables. The errors were high.
* Next we let forecast package in R select the best order for ARIMA model and it selected order of (4, 1, 4) with no seasonality. We got a Mean Absolute Percentage Error of 2.52.
* 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). For this model we got an MAPE of 2.19
* Both MAPE of models that we created were better than the best MAPE of the competitor’s model.

**Performance on Big Data: Time Measurements**

|  |  |  |
| --- | --- | --- |
| **#** | **Operation** | **Time Measured** |
| **1** | Load raw dataset in Spark | 52 sec for 20 million records |
| **2** | Load dataset in R | 1 min 31 sec |
| **3** | Aggregate (Group by) data by hour | 4 min 18 sec |
| **4** | Look at descriptive statistics | 4 seconds |
| **5** | Decision Tree (20 million Records) | 3 min 5 sec |
| **6** | Time Series Non-Seasonal | 32 sec |
| **7** | Time Series Seasonal | 4 min 1 sec |
| **8** | Count using SQL | 18 sec |
| **9** | Plot non-seasonal TS result | <1 sec |
| **10** | Plot seasonal TS result | 2 sec |

**Conclusion**

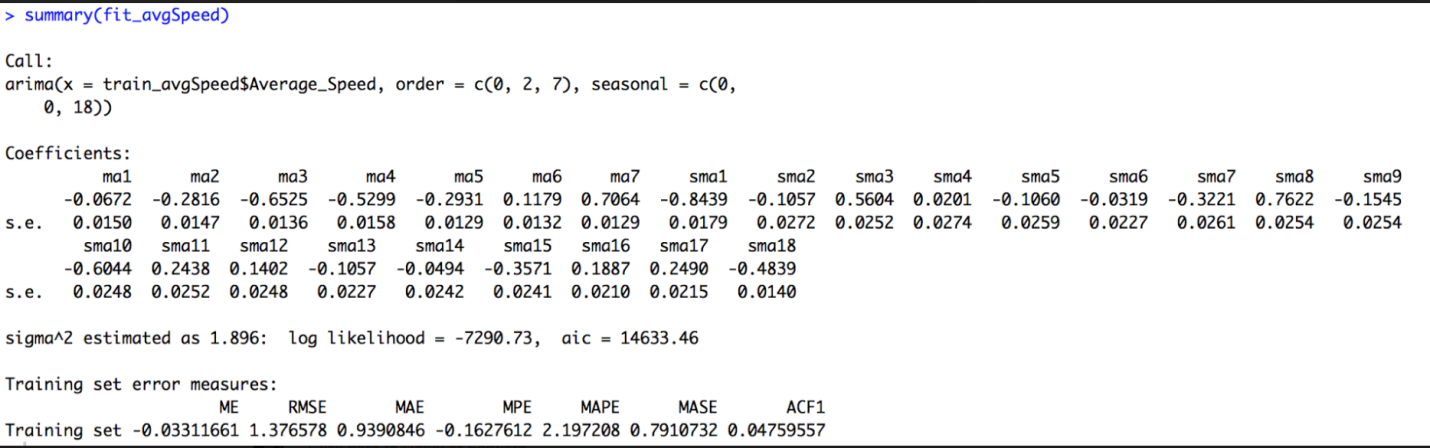
As the competitor we also found that SARIMA model is best model to predict traffic flow as we observed the least amount of MAPE using this model.

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, 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 SARIMA is the best predictor of traffic flow.

The model which has been used here is a time series model (ARIMA), and this model has been used to predict the traffic based on the speed as measurement. The higher speed means the lower traffic and the lower speed means the higher traffic. The result was focusing on predicting the traffic on daily and hourly basis. We were able to predict the traffic based on historical data quite accurately.

The conclusion for this paper has built a new model that produces and predicts the traffic on a giving road based on historical data. The model which has been used is a timer series model. And the model was able to predict the needed information based on the historical data. Even though the competitor have used seven different models, their MAPE is coming out to be much higher, as shown before in “work by competitor”.



When compared to our model, we are getting a much lower MAPE, which shows that our model is better fitting than the competitor.