Name: Shyam Narayan Solanke
UID: 121127761

Final Project Report

Research question: - Traffic data and analytics company INRIX estimates that traffic congestion cost U.S. commuters $305 billion in 2017 due to wasted fuel, lost time and the increased cost of transporting goods through congested areas. Traffic Forecasting can be used to choose from multiple routes to minimize the above-mentioned issues. In this project we would predict number of cars on a given day on a given junction.

Dataset: -

1. Timeseries Dataset
2. Number of Rows – 48121
3. Features – Time, date, junction name, number of cars, ID
4. To predict – number of cars
5. Source – Kaggle

Data File:

Data is a CSV file downloaded from Kaggle

Column Description:

1. Datetime column: This column has hourly Date and Time data. Each item in the column represents the date and the hour of the day number of Vehicle data was collected for that Junction. Datatype – String, Example – "01-11-2015 06:00:00",
2. ID: The ID feature is the unique identifier for each row in the dataset. Datatype – Integer, Example - 20151101001
3. Vehicles: Number of Vehicles, Datatype – Integer, Example - 42
4. Junction: Junction number at which the data was collected. There are in total 4 Junctions. Datatype – Integer, Example – 1

ML Methodology: -

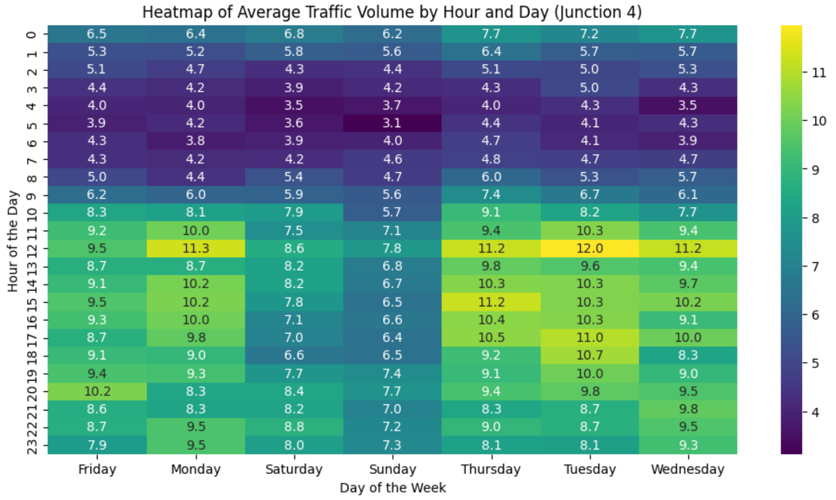
Data Preprocessing –

1. Checking for missing values
2. Converting Datetime String to Datetime object in python
3. Extracting features from the date such as Day of month, Day of week, Month, Year
4. Extracting features from the time such as hour of the day
5. Making a Holiday column to indicate if that day was holiday or not
6. Splitting the datasets based on the Junction to analyse each junction separately
7. Plotting partial autocorrelation function [PACF] graphs
8. Plotting for checking Trend and Seasonality

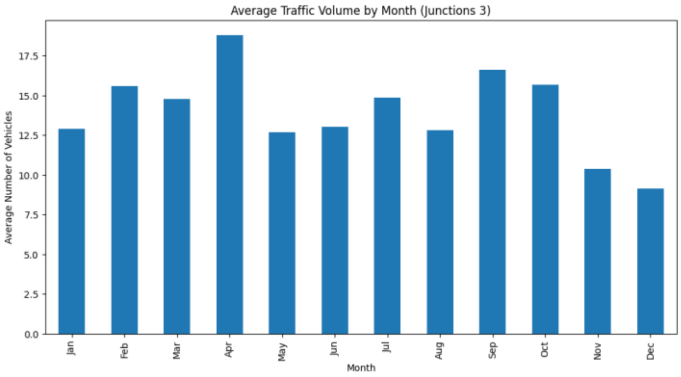
Graphs and Insights –

Common observations

1. There is less traffic During weekends
2. Very Early morning there is very less traffic

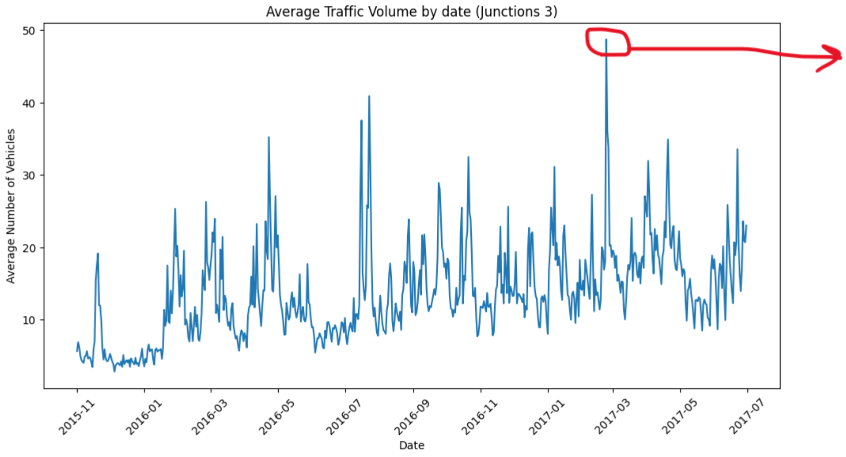


3. Usually, month of December and November has less traffic



Junction specific Observation:

4. We found outliers in Junction2 and Junction3



Sudden spike here

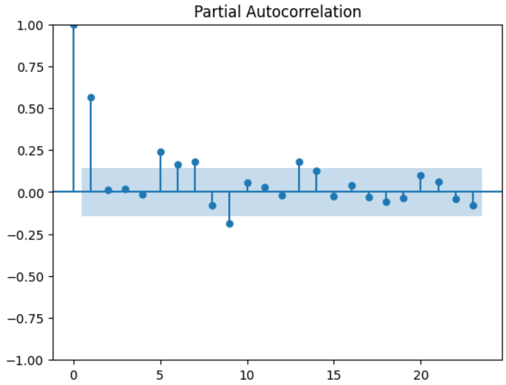
Checking PACF [Partial Autocorrelation Function], Trend and Seasonality

PACF - **Partial Autocorrelation Function (PACF)** measures the direct relationship between a time series and its lagged values, removing the influence of intermediate lags. The PACF plot helps determine how many lags are significant in explaining the time series behaviour. Significant spikes in the PACF plot suggest potential lags to include in the AR model.

1. In all four Junctions Partial Autocorrelation at 7, 8, 9 lag is high suggesting there is some seasonality or repeated pattern based on day of the week

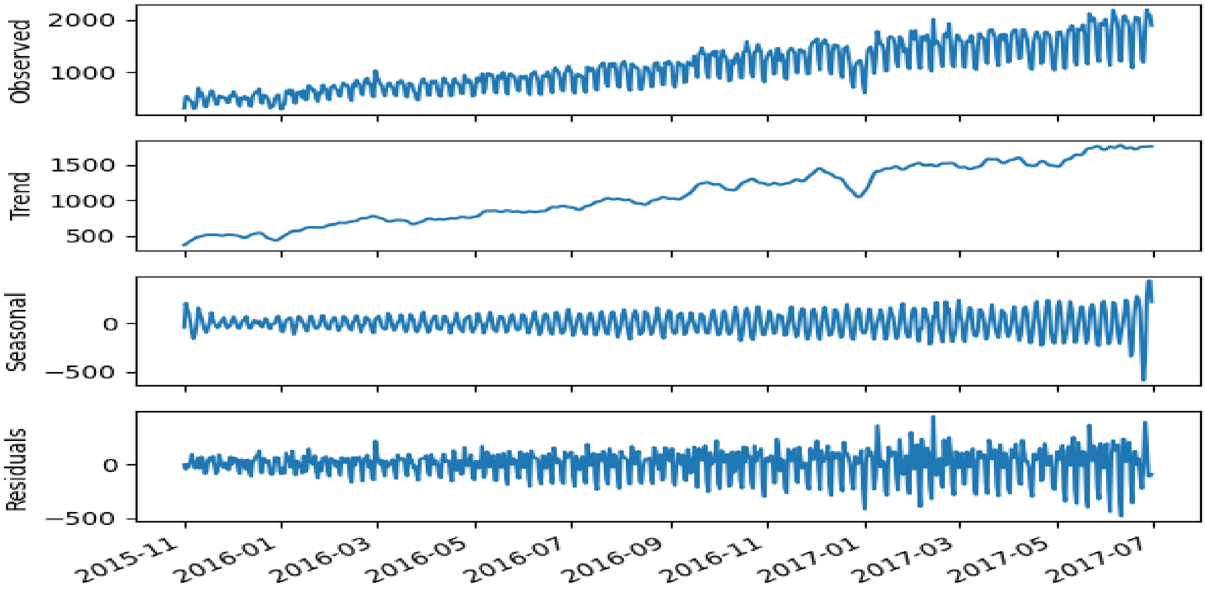
2. Also, we can see there is very high value of lag 1 suggesting the traffic a day before highly determines the traffic next day

Example PACF graph of Junction 4



Trend and Seasonality

If we see trends and Seasonality of our data we see there is a clear upward trend and some weekly seasonality as suspected for Junction 1

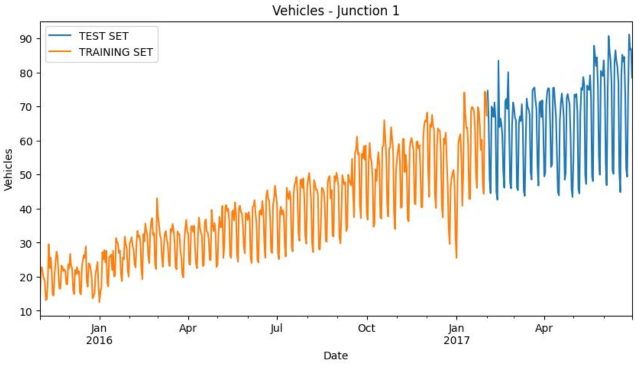


Models: - Here we have used SARIMA and LSTM model

SARIMA - SARIMA, or Seasonal ARIMA, is an extension of the ARIMA model to handle seasonal patterns in time series data. ARIMA consists of three components: AutoRegressive (AR), which models the relationship with past values; Integrated (I), differencing to ensure stationarity; and Moving Average (MA), which models past errors. SARIMA adds seasonal components: Seasonal AR (P), Seasonal Differencing (D), and Seasonal MA (Q), along with a seasonal period (m) to capture recurring cycles. Combining non- Seasonal- (p, d, q) and seasonal- (P, D, Q, m) parameters, SARIMA can effectively model trends and seasonal variations, thereby making it ideal for time series data such as monthly sales or temperatures.

Train Test Split

We split train and test data based on a date, dates after 1st Feb would be testing data and dates before that as training data



Experiment 1

1. First, we are creating a dataset which tells how many cars passed through Junction 1 daily
2. In First SARIMA model we will be only using date to train model

Setting up hyperparameters

a) In the previous PACF graphs we got to know that lag 1, 7, 8, 9 are affecting the time series the most so here in hyper parameter tuning we will tune model on parameter “p” from 0 to 11 and see which value gives best results

b) First order differencing is making model stable hence trying from d=0 to d=2

c) After iterating over various hyperparameters we finally got the optimal one - **Best parameters to train on are ARIMA order= (9, 1, 1), seasonal order= (0, 0, 2, 12)**

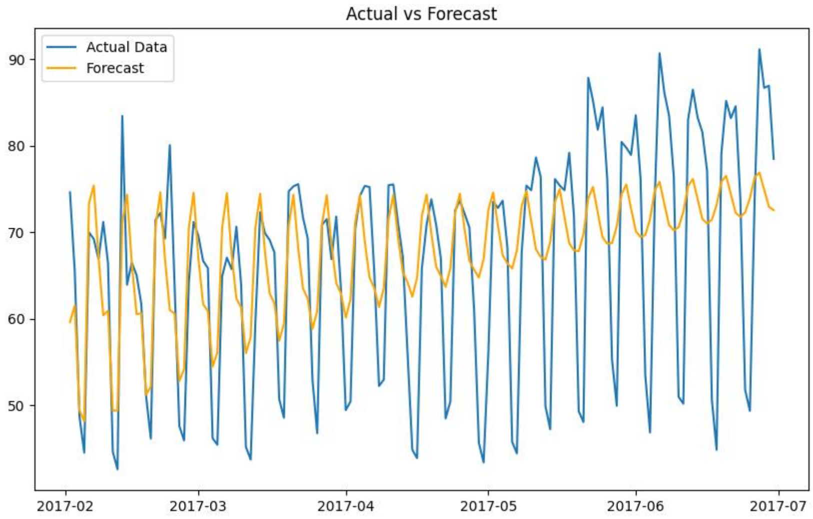
Results

a. RMSE: 10.55762769372565

b. MAE: 8.281720769268702

c. MAPE: 14.122977630913661%

Results



Experiment 2

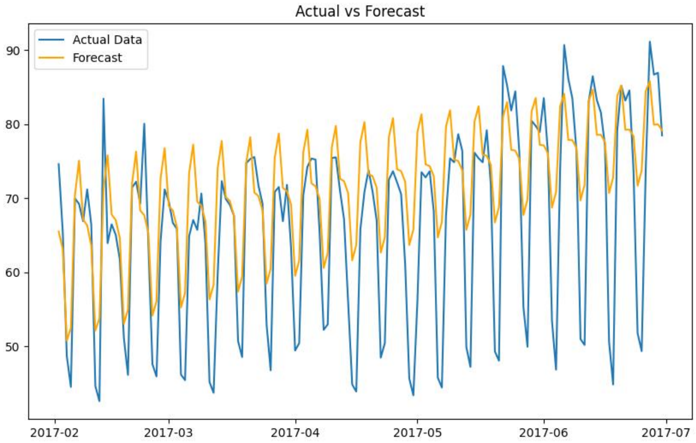
1. We will use the same hyperparameters we got from previous Experiment
2. This time we will also add other extracted features such as Day of the week, month etc... to predict the timeseries
3. This model is also called SARIMAX or SARIMA with external factors

Results

a) RMSE: 9.755773382121339
b) MAE: 7.31407756699199
c) MAPE: 13.090705828227527%

Observation:

We found out that adding external features helps the model to predict better



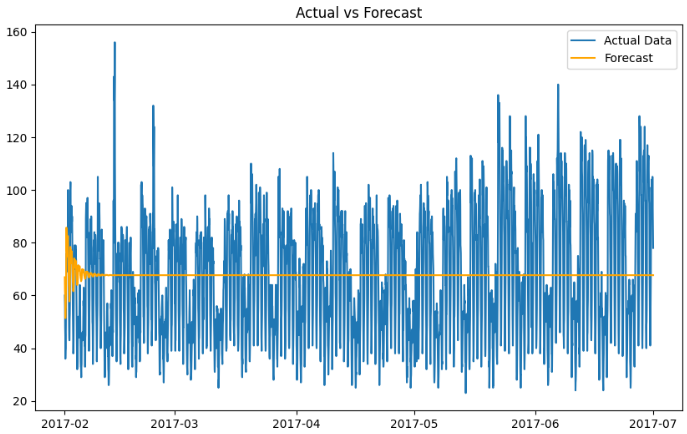
Experiment 3

1. We will keep everything same as Experiment 2 except we would increase the granularity of time series this time instead of predicting vehicles on each date we would predict number of vehicles on each hour

Results

a) RMSE: 23.13552641667907
b) MAE: 19.930847252413987
c) MAPE: 35.8408284781568%

Observation: The error increased a lot, so more the granularity more difficult it is for the model to predict in this case



Experiment 4

1. We will try to use LSTM model to predict Time series
2. LSTM works little different then the normal ARIMA and SARIMA here we will pass the extracted features from date as input instead of date
3. Our single input datapoint consists of 24 rows each having 5 columns with features like month number, day of the week, day of month, Vehicle count in those 24 days etc…
4. Our output is prediction of number of Vehicles on next day

LSTM

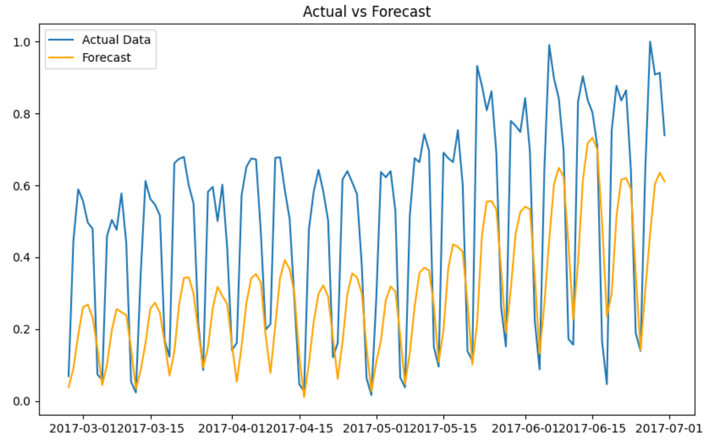
The LSTM model, which is a particular type of RNN, was designed to avoid these limitations of traditional RNNs, especially when learning those long-term dependencies. That it does by incorporating special memory cells, enabling the network to remember information for extended periods. Each of the memory cells consists of input, forget, and output gates, which control the flow of information into, within, and out of the cell. It regulates how much new information enters the cell state via the input gate, how much information to forget, and how much information to output using the output gate. This architecture prevents problems such as the vanishing gradient problem, hence making LSTMs powerful tools in handling sequential data like time series, natural language, and speech.

Results

a) Test MSE with dropout: 0.07849173318337026
b) Test MAE with dropout: 0.23631036898225163

Observations

LSTM does not produce as good results as SARIMA and ARIMA but again this is a basic model of LSTM and lot improvements can be done on this



Lesson Learned:

1. Adding more variables helps SARIMA model to perform better
2. Scaling the data is very important
3. Neural networks like LSTM takes much more time to train then models like SARIMA where SARIMA took 3 minutes LSTM with 100 epochs took 5 minutes
4. There is lot of things we can change working with LSTM to improve model like changing hyperparameters in layers, try giving different variations in input etc…