PARKING AVAILABILITY PREDICTION SYSTEM

A DIRECT READINGS IN

Computer Science (Data Science)

Presented to the Faculty of the
University of Missouri-Kansas City
on partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

By

MASTAN KRISHNA NEERAJ PADARTHI

Student ID: 16274175

Kansas City, Missouri

2019

➔ Abstract

Parking is one of the most important aspect for the commuter on a daily basis. Parking could be a problem if there are many vehicles to be parked but there is no space to accommodate to all the vehicles. Due to the vast usage of cars and motorcycles in the Unites States, we experience parking spots availability issues mainly in populated areas like Downtowns, Shopping Districts, Food districts, etc. Now as a team we have developed a 'Parking Availability Prediction System' which aims at predicting the availability of number of parking spots at a given location, date and time. This model is based on the analyzing the historical parking data and predicting the number of available parking spots through time series forecasting methods.

➔ Contributions

1. Engaged in data cleaning process.
2. Performed the preprocessing of data on the Kansas City and Melbourne datasets.
3. Merging the Kansas dataset with the weather data.
4. Did exploratory data analysis to understand and see the correlations between the features.
5. Involved in time series model selection and training.
6. Performed multiple POC'S and Implemented the time series models.
    a. Prophet
    b. LSTM – Neural Networks
    c. ARIMA/SARIMA
7. Helped in documenting and making of review paper and thesis document.

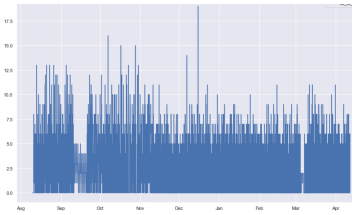
➔ Datasets

- **Kansas City**

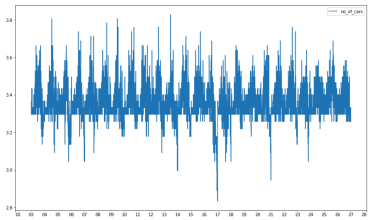
Parking events is given by the XAQT company from Kansas City, Missouri USA. Kansas City data is collected by the sensors which are placed across the parking plots of the main street. XAQT company's data is from August 2018 to April 2019 which consists of mainly 9 features and 274366 records. Fig(a) describes about the distribution of parking events in Kansas City.

- **Melbourne City**

It has a total of around 36 million parking events throughout the year of 2017. Since it is very hard to process such a huge dataset, we have narrowed the data set down to one area if a street. The area of banks in Queen Street, Melbourne contains around 780,000 parking events in 2017. Fig(b) describes about the distribution of parking events in Melbourne City.



Fig(a)



Fig(b)

➔ **Data preprocessing**

a. **Identifying and handling missing values** – In the given data, end date is null for around 60K records. As it is parking events, having ending date as null which wouldn't be used for meaning statistical analysis. So, we have dropped the records where the end date is null. As it a time series analysis, In the given dataset there are few missing days which we have handled using Facebook Prophet algorithm and we would be discussing this in later sections. While merging with weather data, we have observed that few of the records doesn't have the temperature and summary attributes. So, we have replaced the null values with the mean and mode respectively.

b. **Removing Features** – In the given dataset, columns such as 'sensorID', 'geometry', 'sessionType', 'UUID', 'longitude', 'latitude' doesn't bring any predicting power while forecasting the data. So, we can remove these columns.

c. **Data Formatting** – In the given dataset, time analysis feature 'startTime' and 'endTime' are of object type which couldn't be used for meaningful statistical analysis. So, we need to convert them into 'dateTime' features.

```
# Before
df.dtypes
```

```
asset_id      object
start_date    object
end_date      object
dtype: object
```

```
df['start_date'] = pd.to_datetime(df.start_date,format='%Y/%m/%d').astype('datetime64[s]')
df['end_date'] = pd.to_datetime(df.end_date,format='%Y/%m/%d').astype('datetime64[s]')
```

```
# After
df.dtypes
```

```
asset_id              object
start_date    datetime64[ns]
end_date      datetime64[ns]
dtype: object
```

d. **Binning** - It means that grouping the values together into bins so that it would be useful in a meaningful statistical analysis, basically it gives a better understanding of the data distribution. In the given dataset temperature column is an attribute ranging from -5 to 97. Using binning, I have categorized the temperature column into 11 bins each with 10-degree Fahrenheit.

e. Pre-processing the data

We have identified that there are mainly three stages in pre-processing the entire dataset into the data which the model requires

- **Stage 1** - Converting the dataset into 5 Min bins.
- **Stage 2** - 288 bins per day.
- **Stage 3** - Merging with the weather data.

Stage 1 - Converting the dataset into 5 Min bins

Approach followed over here is creating 2 new attributes by performing the (5 Mins) Floor to the start date and (5 Mins) Ceil to the End date. Then subtracted the above two columns to get the difference amount and divided it with the 5 to get the number of slots it has been parked. Now converting the differenced amount to the list and finally exploding the data into different records.

Basically, dataset consists of 2 main scenarios which are: -

Scenario 1: Car entered and exited the parking slot within the 5 mins time.
Before (Data Set Data)

UUID	Asset ID	Start Time	End Time
SENSITY-kc-9-CDDB22AD-31F6-4A11-BC94-CCC6602575A1	SENSITY-kc-9	8/10/2018 23:37:39	8/10/2018 23:37:39

After (Data Frame Data)

Start Time	End Time	Asset ID	No_of_Cars
8/10/2018 23:35:39	8/10/2018 23:40:39	SENSITY-kc-9	1

Scenario 2: Car entered and parked for more than 5 mins and then exited the.
parking slot.
Before (Data Set Data)

UUID	Asset ID	Start Time	End Time
SENSITY-kc-9-CDDB22AD-31F6-4A11-BC94-CCC6602575A1	SENSITY-kc-9	8/10/2018 23:37:39	8/10/2018 23:46:39

After (Data Frame Data)

Start Time	End Time	Asset ID	No_of_Cars
8/10/2018 23:35:39	8/10/2018 23:40:39	SENSITY-kc-9	1
8/10/2018 23:40:39	8/10/2018 23:45:39	SENSITY-kc-9	1
8/10/2018 23:45:39	8/10/2018 23:50:39	SENSITY-kc-9	1

Stage 2 – 288 bins per day.

The approach followed over here is taking the first and end dates of the data set then creating a new data frame consisting of all the dates within the range of first and end dates. Now splitting each day into 5 min bins which would create 288 bins per day. Finally, merging this new data frame with our data set.

start_time	
2018-09-21	288
2018-09-22	288
2018-09-23	288
2018-09-24	288
2018-09-25	288
2018-09-26	288
2018-09-27	288
2018-09-28	288

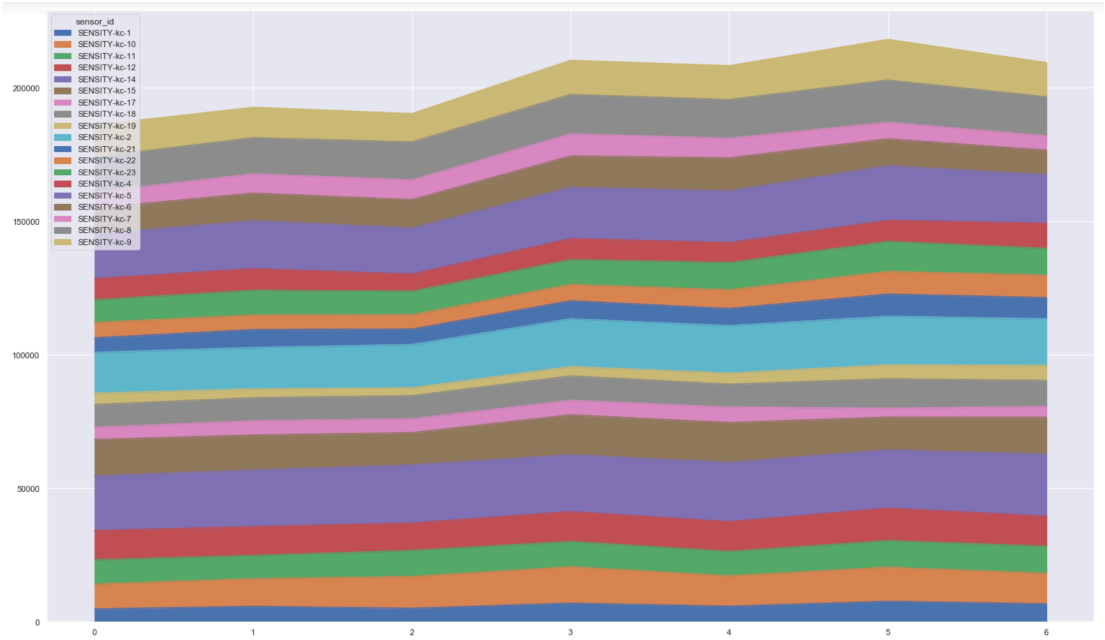
Stage 3 – Merging with Weather Data

Using the 'forecastio' library in Python, we have collected the few important attributes which are summary, time and temperature. Summary column which describes the brief description of the temperature for an hour such as Overcast, Heavy Rain, Snow, etc. Temperature columns consists the temperature recorded for that hour. The main challenge over here is the library returns the data on an hourly basis, but we would be requiring the weather data for every 5 mins. So, we would be splitting an hourly data into 5 mins bins which would finally create 288 bins per day then we would merge it with our data set.

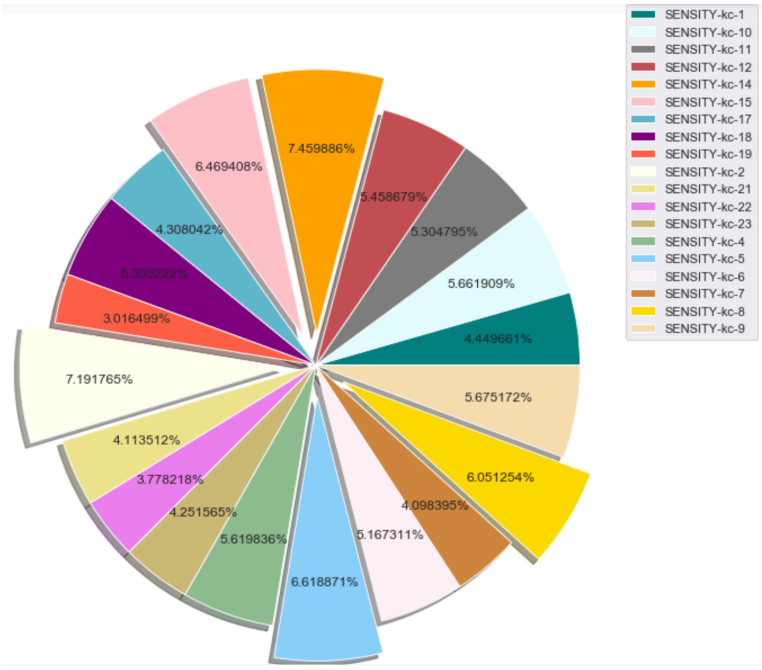
start_time	no_of_cars	summary	temperature
2019-04-11 20:00:00	2.0	Clear	46.468514
2019-04-11 20:05:00	5.0	Clear	46.468514
2019-04-11 20:10:00	4.0	Clear	46.468514
2019-04-11 20:15:00	4.0	Clear	46.468514
2019-04-11 20:20:00	5.0	Clear	46.468514
2019-04-11 20:25:00	2.0	Clear	46.468514
2019-04-11 20:30:00	0.0	Clear	46.468514
2019-04-11 20:35:00	0.0	Clear	46.468514
2019-04-11 20:40:00	0.0	Clear	46.468514
2019-04-11 20:45:00	3.0	Clear	46.468514
2019-04-11 20:50:00	6.0	Clear	46.468514
2019-04-11 20:55:00	3.0	Clear	46.468514

➔ Exploratory Data Analysis

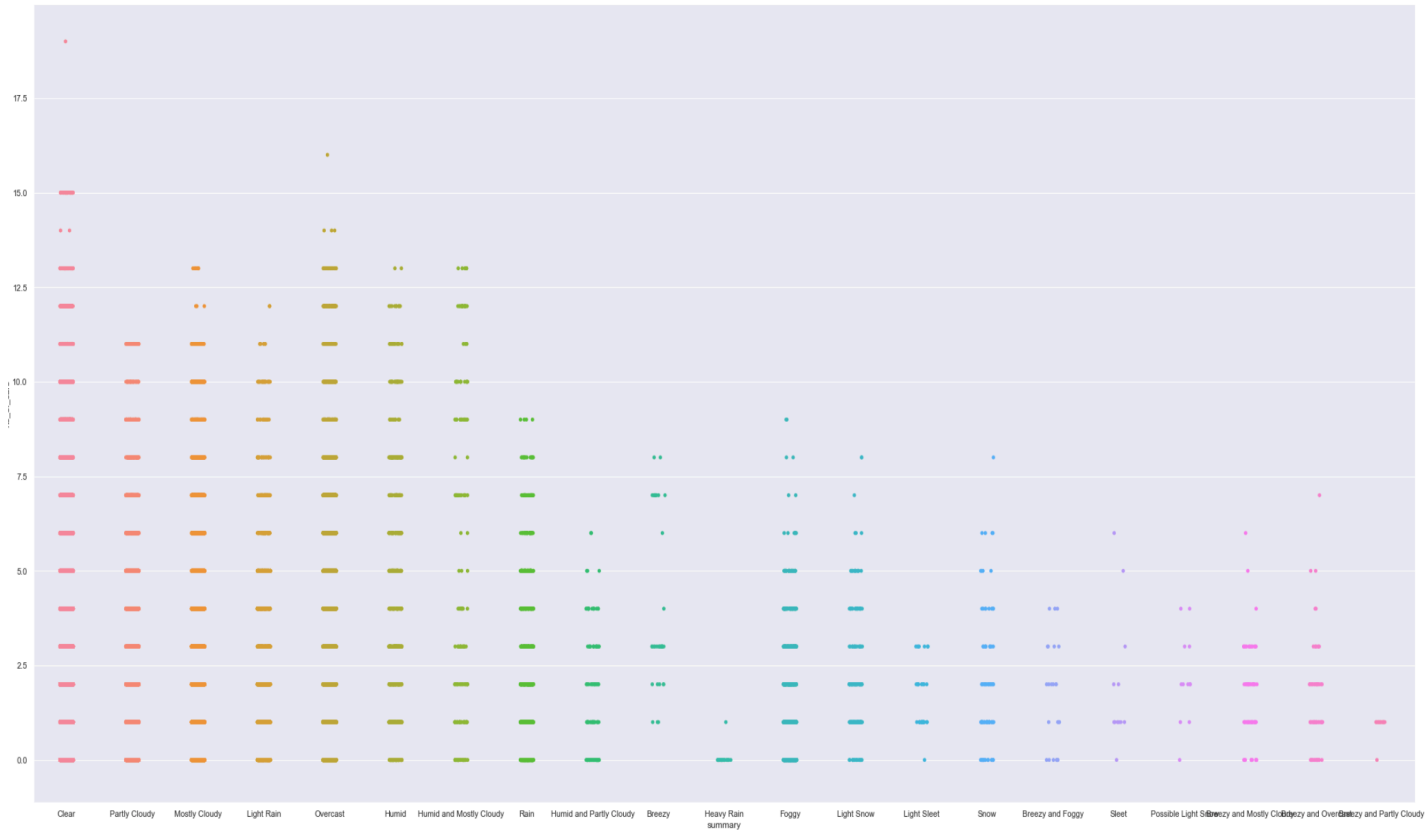
Below graph describes the trend within the sensors Wednesday and Friday has the maximum number of cars parked.



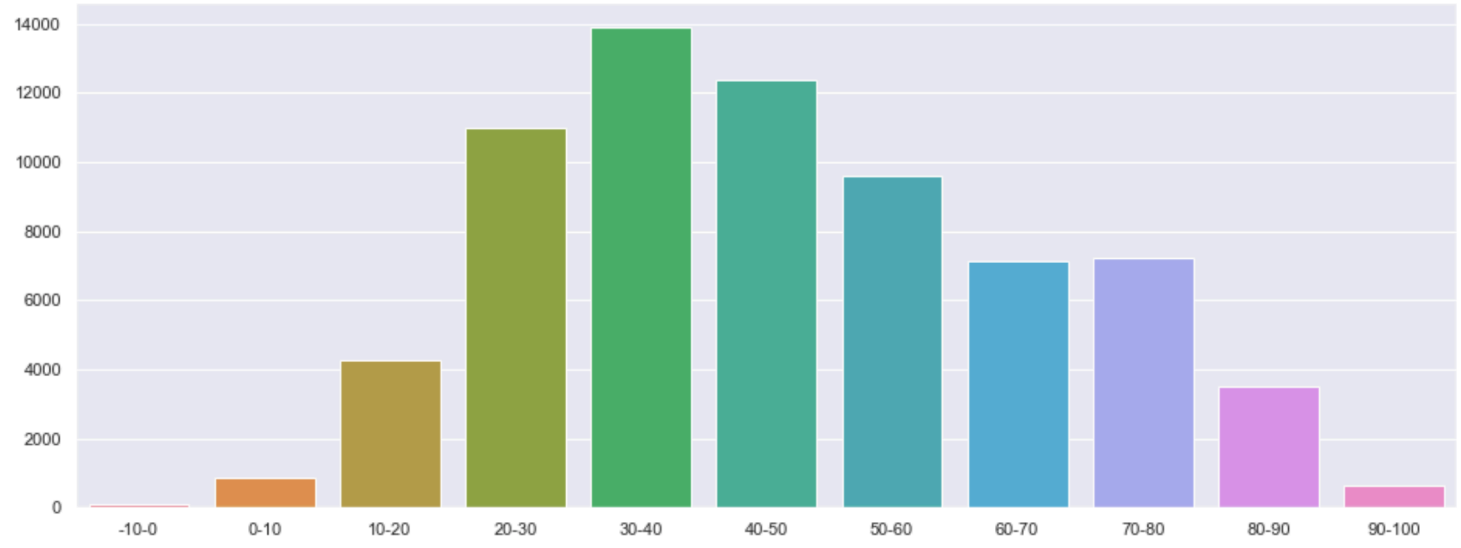
The following pie chart describes the distribution of parking events within the sensors. 14, 2 and sensor 5 has the maximum number of cars parked.



Below graph describes the different climate summaries to the number of cars parked in the Kansas City. Clear has the maximum number of cars parked ranging from 0 to 18 parking events whereas the heavy rain has the minimum number of cars parked.



The following graph describes the temperature bin data to the number of instances the car parked. We can clearly see that in range from 30-40degree Fahrenheit the maximum number of cars parked.



The Analysis of Variance (ANOVA) is a statistical method used to test whether there are significant differences between the means of two or more groups. If our temperature variable is strongly correlated with the number of cars, then ANOVA will return a sizeable F-test score and a small p-value. So, we can clearly see that for the 'Heavy Rain' attribute has the maximum F and small P values.

```
from scipy import stats
f_val, p_val = stats.f_oneway(grouped_test.get_group('Clear')['no_of_cars'], grouped_test.get_group('Sleet')['no_of_
print( "ANOVA results: F=", f_val, ", P =", p_val )
```

```
ANOVA results: F= 0.12013116339950254 , P = 0.7288943700689673
```

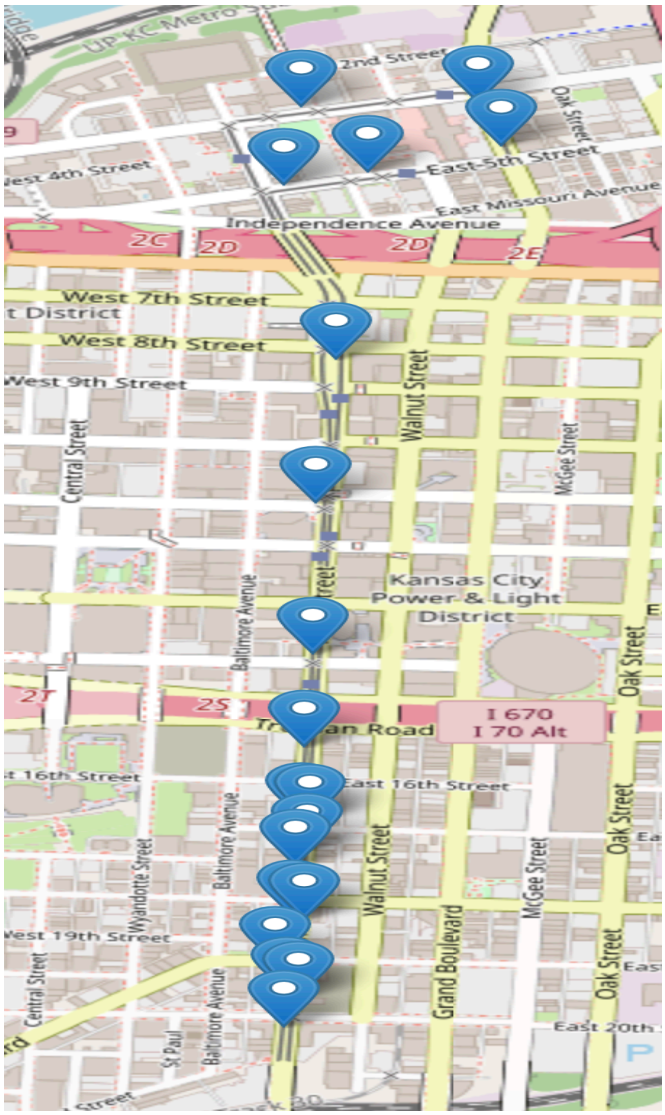
```
_val, p_val = stats.f_oneway(grouped_test.get_group('Clear')['no_of_cars'], grouped_test.get_group('Heavy Rain')['no_
print( "ANOVA results: F=", f_val, ", P =", p_val )
```

```
ANOVA results: F= 19.528405683332014 , P = 9.941370854575151e-06
```

```
f_val, p_val = stats.f_oneway(grouped_test.get_group('Clear')['no_of_cars'], grouped_test.get_group('Rain')['no_of_ca
print( "ANOVA results: F=", f_val, ", P =", p_val )
```

```
ANOVA results: F= 0.2725425905219645 , P = 0.6016345052485266
```

Below screenshot describes the lon, lat locations of the various sensors on the main street.



➔ Machine Learning Models

- **Prophet**

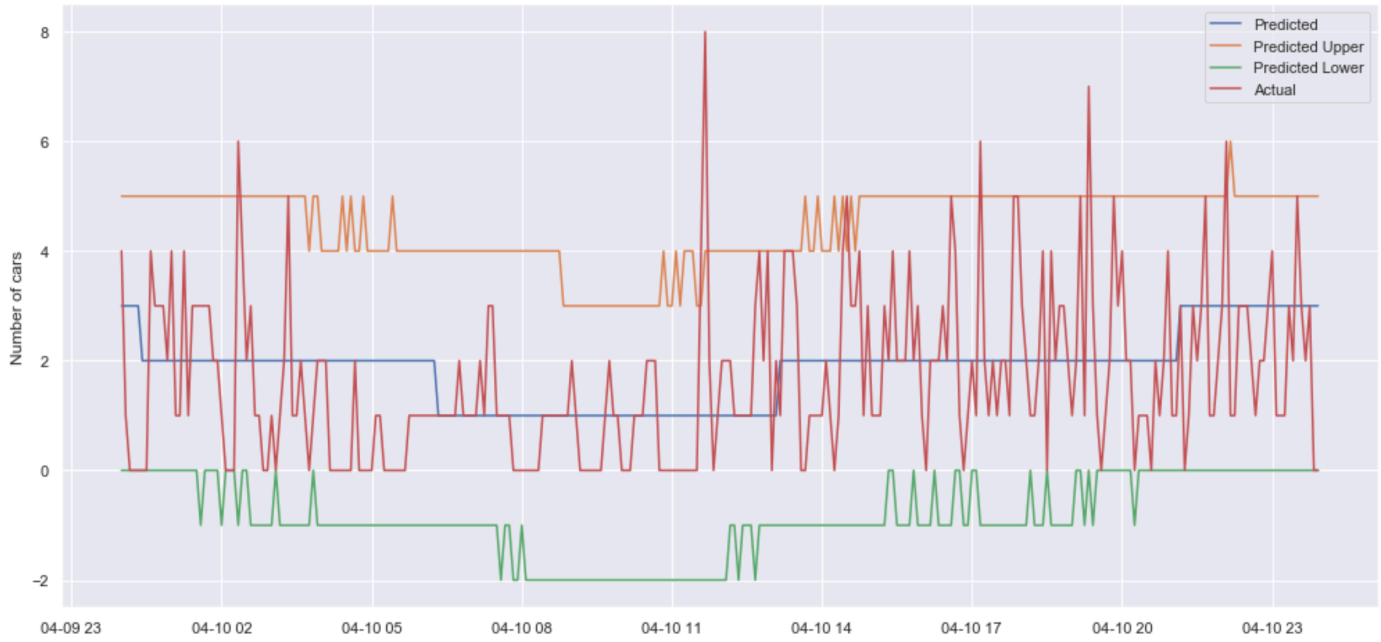
Prophet is an open source library published by Facebook, input to Prophet is always a data frame with two columns: ds and y. The ds (date stamp) column represents the date format and the y column must be numeric and represents the measurement we wish to forecast.

	ds	y
70267	2019-04-10 23:35:00	3.0
70268	2019-04-10 23:40:00	2.0
70269	2019-04-10 23:45:00	3.0
70270	2019-04-10 23:50:00	0.0
70271	2019-04-10 23:55:00	0.0

Below output compares the actual and predicted values. Along with the forecasted(yaht) value it also displays the lower and upper bounds.

ds	yhat	yhat_lower	yhat_upper	y
2019-04-10 14:00:00	1.818518	-1.071034	4.499924	1.0
2019-04-10 14:05:00	1.842319	-0.866567	4.290819	2.0
2019-04-10 14:10:00	1.864958	-0.846350	4.417714	1.0
2019-04-10 14:15:00	1.886405	-0.681292	4.555213	0.0
2019-04-10 14:20:00	1.906633	-0.650619	4.475064	1.0
2019-04-10 14:25:00	1.925620	-0.902873	4.655103	4.0
2019-04-10 14:30:00	1.943352	-0.854728	4.439061	5.0
2019-04-10 14:35:00	1.959819	-0.738868	4.617780	3.0
2019-04-10 14:40:00	1.975018	-0.695933	4.493368	3.0
2019-04-10 14:45:00	1.988951	-0.610201	4.722962	4.0
2019-04-10 14:50:00	2.001626	-0.745592	4.634566	1.0

Below graph compares the actual values to the prophet forecasted values.



- **LSTM – Neural Networks**

The Long Short-Term Memory network, or LSTM network, is a recurrent neural network that is trained using Backpropagation Through Time and overcomes the vanishing gradient problem. Hyperparameters tuned using cross validation for this model are Learn Rate, Optimizer, Number of Neurons, Activation Function, epochs.

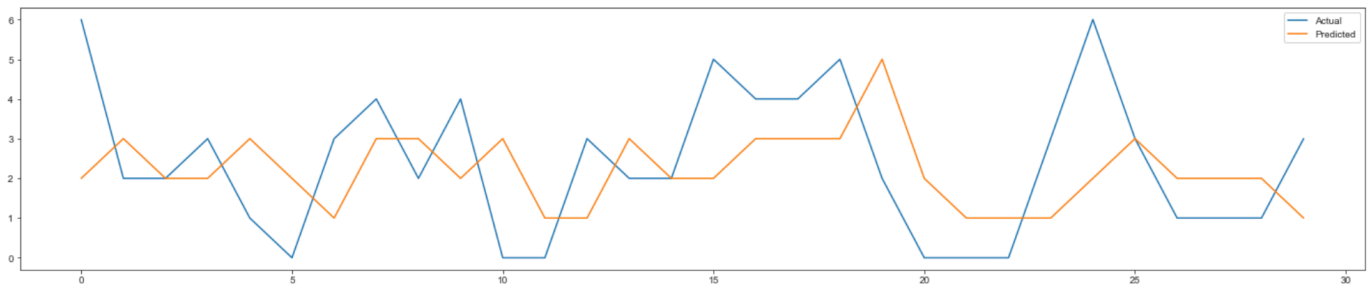
Approach

We need input variable(s) and an output variable. We can transform the data and use previous time steps as input variables and use the next time step as the output variable.
For example: If the dataset looked like
1
2
3

We will be restructuring it into a supervised learning problem by writing it in the following format
1, 2
2, 3
where the first column is the input and the second column are the output.

start_time	no_of_cars	(t-1)	(t-2)	(t-3)
2019-04-11 20:35:00	0	3.0	6.0	3.0
2019-04-11 20:40:00	0	6.0	3.0	1.0
2019-04-11 20:45:00	3	3.0	1.0	1.0
2019-04-11 20:50:00	6	1.0	1.0	1.0
2019-04-11 20:55:00	3	1.0	1.0	3.0

Below graph compares the actual values to the predicted values.

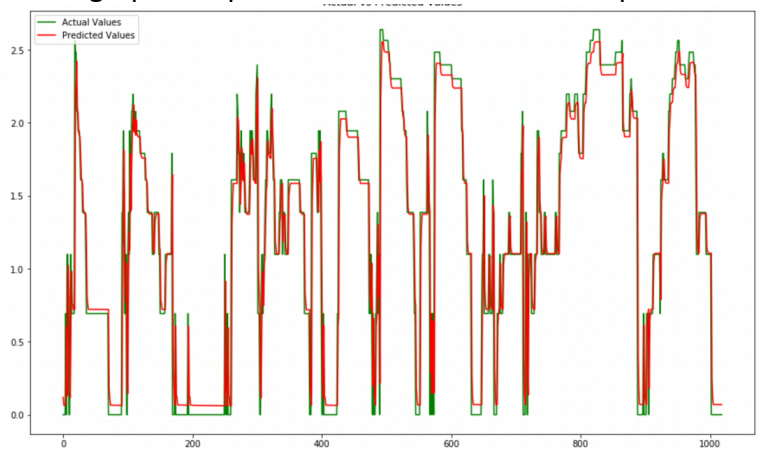


- **ARIMA/SARIMA**

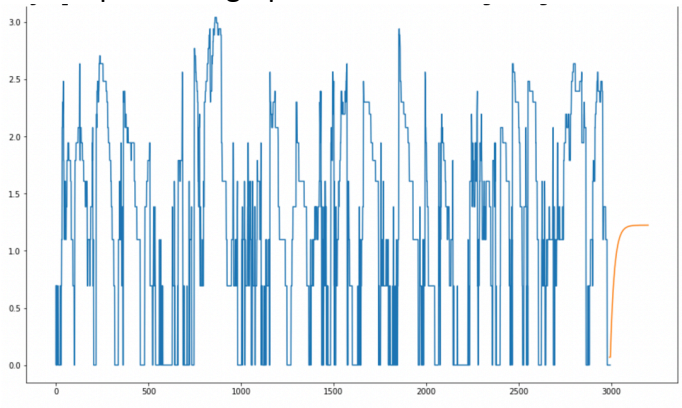
ARIMA

Auto Regressive Integrated Moving Average is a class of statistical models for analyzing and forecasting time series data. It is a combination of 2 main models which are Auto Regressive (pattern of growth/decline in the data is accounted), Moving Average (noise between consecutive time points is accounted)

Below graph compares the actual values to the predicted values.

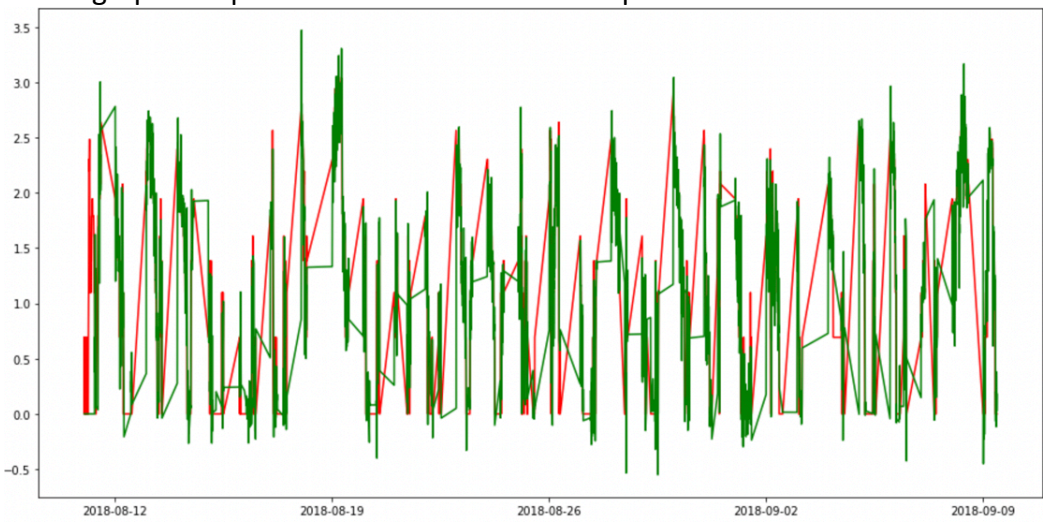


Below graph describes the forecasted values using ARIMA model which clearly says that after a certain point the graph has been constant.

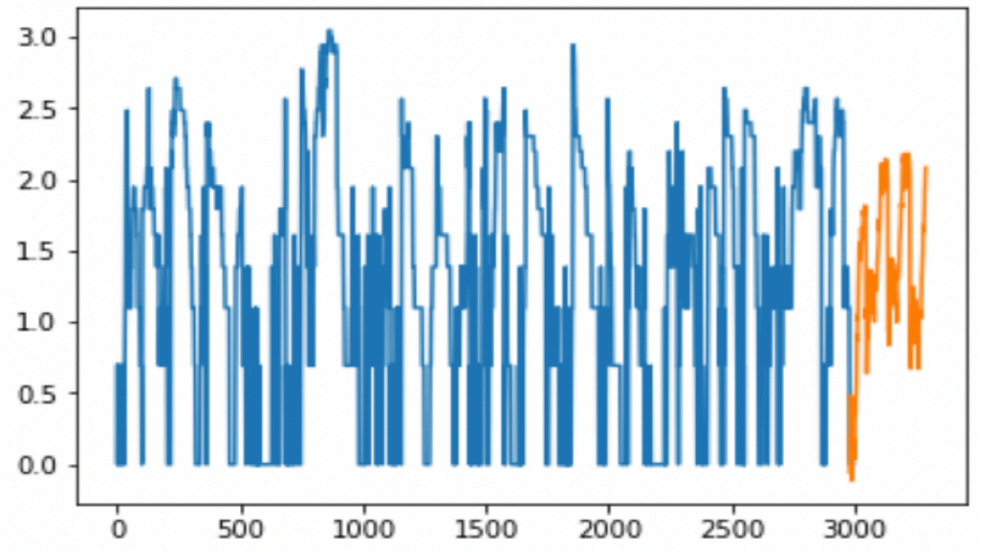


Seasonal Autoregressive Integrated Moving Average, SARIMA or Seasonal ARIMA, is an extension of ARIMA that explicitly supports univariate time series data with a seasonal component. It has three main hyperparameters which are autoregression (AR), differencing (I) and moving average (MA) for the seasonal component of the series, as well as an additional parameter for the period of the seasonality.

Below graph compares the actual values to the predicted values.



Below graph describes the forecasted values using SARIMA model which clearly forecasts the seasonality within the data.



➔ Documentation

Provided the content for the pre-processing of the data which included 5 mins bins, 288 bins per day and merging with weather data. Also, provided the clustering, EDA visualizations and machine learning models observations. Helped in writing the draft version of the paper and thesis document in every section and reviewing the paper after every version to ensure that the structure is covered in IEEE format. Formatted the content into IEEE format paper. Adding the proper references which need to be given some proof i.e evidence of the statement which is written in the paper.