A
Project Report
On
"Predictive Analytics for Traffic"

(CE447 - Software Project Major)

Prepared by
Dwijesh Shah (16CE109)

Under the Supervision of
Asst. Prof. Aayushi Chaudhari

Submitted to

Charotar University of Science & Technology (CHARUSAT)
for the Partial Fulfillment of the Requirements for the
Degree of Bachelor of Technology (B.Tech.)
in U & P U. Patel Department of Computer Engineering (CE)
for B. Tech Semester 8

Submitted at

CHARUSAT
CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY

Accredited with Grade A by NAAC

U & P U. PATEL DEPARTMENT OF COMPUTER ENGINEERING
Chandubhai S. Patel Institute of Technology (CSPIT)
Faculty of Technology & Engineering (FTE), CHARUSAT
At: Changa, Dist: Anand, Pin: 388421.
April-May, 2020

A
Project Report
On
"Predictive Analytics for Traffic"

(CE447 - Software Project Major)

Prepared by

Dwijesh Shah (16CE109)

Under the Supervision of

Asst. Prof. Aayushi Chaudhari

Submitted to

Charotar University of Science & Technology (CHARUSAT)
for the Partial Fulfillment of the Requirements for the
Degree of Bachelor of Technology (B.Tech.)
in U & P U. Patel Department of Computer Engineering (CE)
for B. Tech Semester 8

Submitted at

CHARUSAT
CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY

Accredited with Grade A by NAAC

cspit
Chandubhai S Patel
Institute of Technology

U & P U. PATEL DEPARTMENT OF COMPUTER ENGINEERING
Chandubhai S. Patel Institute of Technology (CSPIT)
Faculty of Technology & Engineering (FTE), CHARUSAT
At: Changa, Dist: Anand, Pin: 388421.
April-May, 2020

DECLARATION BY THE CANDIDATE

I hereby declare that the project report entitled “**Predictive Analytics for Traffic**” submitted by me to Chandubhai S. Patel Institute of Technology, Changa in partial fulfilment of the requirements for the award of the degree of **B.Tech Computer Engineering**, from U & P U. Patel Department of Computer Engineering, CSPIT, FTE, is a record of bonafide CE447 Software Project Major (project work) carried out by me under the guidance of **Asst. Prof. Aayushi Chaudhari**. I further declare that the work carried out and documented in this project report has not been submitted anywhere else either in part or in full and it is the original work, for the award of any other degree or diploma in this institute or any other institute or university.

D. J. Shah

(Dwijesh Shah - 16CE109)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

AChaudhari

Mrs. Aayushi Chaudhari
Assistant Professor
U & P U. Patel Department of Computer Engineering,
Chandubhai S Patel Institute of Technology (CSPIT)
Faculty of Technology (FTE)
Charotar University of Science and Technology (CHARUSAT) - Changa.

AMNEX

To whomsoever it may Concern

Date: 21st April 2020

This is to certify that Dwijesh Shah has done his Internship with our organization from **16th December 2019 to 20th April 2020.**

He has worked on the project titled “Predictive Analytics for Traffic”.

We wish the best for future endeavors.

For Amnex Infotechnologies (P) Ltd.,

Prachi Sharma
Senior Associate – HR

Note: This is an electronic document. No signature required.

अमृतं तु विद्या

CHARUSAT

CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY

Accredited with Grade A by NAAC

CERTIFICATE

This is to certify that the report entitled "**Predictive Analytics for Traffic**" is a bonafied work carried out by **Dwijesh Shah (16CE109)** under the guidance and supervision of **Asst. Prof. Aayushi Chaudhari** & **Mr. Nilesh Bhavsar** for the subject **Software Project Major (CE447)** of 8th Semester of Bachelor of Technology in **Computer Engineering** at Chandubhai S. Patel Institute of Technology (CSPIT), Faculty of Technology & Engineering (FTE) – CHARUSAT, Gujarat.

To the best of my knowledge and belief, this work embodies the work of candidate himself, has duly been completed, and fulfills the requirement of the ordinance relating to the B.Tech. Degree of the University and is up to the standard in respect of content, presentation and language for being referred by the examiner(s).

Under the supervision of,

AChaudhari

Miss. Aayushi Chaudhari
Assistant Professor
U & P U. Patel Dept. of Computer Engineering
CSPIT, FTE, CHARUSAT, Changa, Gujarat

Nilesh Bhavsar

Mr. Nilesh Bhavsar
Senior Consultant at Embedded Systems
Machine Learning
Amnex Infotechnologies Pvt. Ltd.

Dr. Ritesh Patel
Head - U & P U. Patel Department of Computer Engineering,
CSPIT, FTE, CHARUSAT, Changa, Gujarat.

Chandubhai S. Patel Institute of Technology (CSPIT)

Faculty of Technology & Engineering (FTE), CHARUSAT

At: Changa, Ta. Petlad, Dist. Anand, Pin: 388421. Gujarat.

ACKNOWLEGEMENT

It has been an enriching experience for me to undergo my summer internship at **Amnex Infotechnologies Pvt. Ltd.**, Ahmedabad which would not have been possible without the goodwill and support of the people around.

As a student of Chandubhai S. Patel Institute of Technology, I am highly thankful to **Dr. A. D. Patel (Principle of CSPIT, CHARUSAT University, Changa)** who allowed me for this internship at **Amnex Infotechnologies Pvt. Ltd.**

I would like to give my heartily thanks to **Mr. Nilesh Bhavsar (Senior Consultant - Embedded)** and **Mr. Dhaval Panchal (Associate Machine Learning Engineer)** who guided me and support throughout the project. Without them I would not be able to complete this project successfully. I am very thankful to the whole staff who helped me at every step whenever needed.

I sincerely thanks to my internal guide **Asst. Prof. Aayushi Chaudhari**, for providing help in preparing internship project report and for guiding me throughout the Summer Training.

Lastly, I convey my regards to the whole staff, which made my internship project at **Amnex Infotechnologies Pvt. Ltd.**, a memorial part of my life.

Thanks and Regards,
Dwijesh Jigarbhai Shah
(16CE109)

ABSTRACT

The internship project "**Predictive Analytics for Traffic**", which is a solution developed for the traffic operators to predict the traffic related various parameters for the upcoming days. The project is based on research and development.

The main aim of this project is to develop an advance system to measure the traffic and other relevant factors in the upcoming future occurrences or events. Currently, the government has decided to implement this approach under ‘Smart City Mission’ to solve the issue of traffic.

Plenty of innovations and advancements are occurring in the field of Transportation in current days with the emergence of Machine Learning, IOT and other latest technologies. To provide better understanding of Predictive Analytics for Traffic various graphs are produced on the basis of various features such as PCU counts, arm wise data, junction wise data and various reports are also generated to get better understanding of the prediction. On the basis of graphs traffic operator can get to know about the PCU count, with the help of which traffic light time for red light and green light is managed.

The purpose of the internship is to work with the eminent technologies in the field of Predictive Analytics by applying various statistical techniques and machine learning models. The scope of the project is limited to the predictive analytics of PCU count which can be enhanced for other factors such as traffic congestion at various arms, traffic congestion at various junction, traffic congestion by vehicle classification and many more.

Table of Contents

ACKNOWLEDGEMENT…………………………………………………..i

ABSTRACT……………………………………………………………..ii

List of Figures...iv

List of Tables vi

Abbreviations...vii

CHAPTER 1 INTRODUCTION 1

1.1 Introduction 1

1.2 Research Definition...1

1.3 Problem Description...2

1.4 Motivation 2

1.5 Scope and Objective...3

1.6 Planning...3

CHAPTER 2 Literature Review & Comparative Study 4

2.1 Evolution of the topic 4

2.2 Background Study (Theoretical and Mathematical Background):..................5

2.3 Review Previous Research Findings 7

2.4 Comparative Study 8

2.5 Open Issues (Research Gaps) 9

CHAPTER 3 Experimentation / Simulations / Lab Set Up......................10

CHAPTER 4 Proposed Hypothesis / Model / Algorithm.........................17

4.1 Algorithm 17

4.2 Implementation Details (Tools and Technologies used)..............................18

4.3 Complexity Analysis 35

CHAPTER 5 Results and Discussion on Results 37

CHAPTER 6 Conclusion 47

CHAPTER 7 Future Extensions 49

Bibliography……………………………………………………….50

List of Figures

Figure 2.1 Evolution of Analytics....4
Figure 2.2 Decision Tree Information Gain....6
Figure 3.1 Master Table Internal Logic....15
Figure 4.1 Train Dataset and Test Dataset Generation Algorithm....17
Figure 4.2 Decision Tree Algorithm Implementation....20
Figure 4.3 XGBoost Algorithm Implementation....21
Figure 4.4 Time Series Decomposition....22
Figure 4.5 Time Series Differencing and ACF....25
Figure 4.6 Original Time Series and ACF....25
Figure 4.7 1st Differencing of Time Series and ACF....26
Figure 4.8 Original Time Series and PACF....26
Figure 4.9 1st Differencing of Time Series and PACF....27
Figure 4.10 2nd Differencing of Time Series and PACF....27
Figure 4.11 ARIMA result summary....27
Figure 4.12 Forecasting by ARIMA....28
Figure 4.13 Forecasting of Time Series using Naïve Approach....29
Figure 4.14 Forecasting of Time Series using Simple Average method....29
Figure 4.15 Forecasting using Moving Average of 5 window size....30
Figure 4.16 Forecasting of Time Series using Moving Average of 10 window size...30
Figure 4.17 Forecasting of Time Series using Moving Average of 60 window size...30
Figure 4.18 Forecasting of Time Series using Simple Exponential Smoothing....31
Figure 4.19 Forecasting of Time Series using Holt Linear’s Approach....31
Figure 4.20 Forecasting of Time Series using Holt Winters Approach....31
Figure 4.21 Forecasting of Time Series using SARIMA....32
Figure 4.22 Time Series Plotting....34
Figure 4.23 Forecasting of Time Series using FBProphet....35
Figure 5.1 Decision Tree Result....38
Figure 5.2 XGBoost Result....39
Figure 5.4 ARIMA 1-minute Results....40
Figure 5.5 SARIMA 1-minute Error....41
Figure 5.6 SARIMA 5-minute Error....42
Figure 5.7 SARIMA 5-minutes Result....42

Figure 5.8 SARIMA 15-minute Result 43
Figure 5.9 SARIMA 30-minute Result 43
Figure 5.10 FBProphet Seasonality 44
Figure 5.11 FBProphet Result... 45

List of Tables

Table 5-1 Comparison of the results of Algorithms on various data 45

Abbreviations

PCU	Passenger Car Unit
JSON	Java Script Object Notation
ARIMA	Auto Regressive Integrated Moving Average
SARIMA	Seasonal Auto Regressive Integrated Moving Average
ACF	Auto Correlation Function
PACF	Partial Auto Correlation Function
FBProphet	Facebook Prophet

CHAPTER 1 INTRODUCTION

1.1 Introduction

- Crashes and traffic congestion are among the most critical challenging issues in traffic engineering.
- Road capacities and road accidents have great impacts on traffic congestion.
- An accurate prediction of traffic flow is one of the important steps in Intelligent Transportation Systems (ITS).
- ITS is termed as the application of advanced sensors, computer, electronics, telecommunication technologies and management strategies in an integrated / combined way to improve the safety and efficiency of the transportation system.
- ITS have enabled the engineers to get access to real time data but real time data not only can be helpful for the road users to decide their routes for travelling, but also can give the chance to the engineer to manage the routes more effectively.
- Insufficient capacity or density and unrestrained demand are interconnected but signal delays are hard coded and do not depend on the amount of traffic density.
- Therefore, there is a necessity to optimize the traffic control system and make it more dynamic so as to accommodate the varying traffic density.

1.2 Research Definition

- Research definition of the internship project is ‘**Predictive Analytics for Traffic**’.
- As the project definition consists of two main keywords, ‘Predictive’ and ‘Analytics’ where, ‘Predictive’ stands for the prediction of the unknown values or features on the basis of historical data and ‘Analytics’ stands for the discovery and interpretation of hidden patterns of data using various statistical modelling approaches.
- In this research project the prediction of various traffic relevant features will be provided on the basis of historical data.
- Currently Government of India is developing most of the cities in the ‘Smart City’ under ‘Smart Cities Mission’.
- ‘Predictive Analytics for Traffic’ is one of the solutions to solve the issue of traffic, which in current situation becomes very hectic to handle. So, Government of India has decided to make it autonomous.
- Under this mission most of the cities of India have either cameras or controllers installed at every cross roads to control the flow of traffic.

- The field for research for us is to predict traffic flow which we collect through cameras and controllers installed on the field and the future occurrences of traffic flow will be provided on the basis of historical data.

1.3 Problem Description

- India has one of the largest road networks across the world, which is spanned over a total of 5.5 million kilometres.
- Indian roads carry almost 90 % of the country’s passenger traffic.
- Indian road network transports 64.5 % of all goods in the country.
- Road transportation has gradually increased over the years with the improvements in the connectivity between cities, towns and villages in the country.
- With the increase in the number of vehicles there is a need of proper and advance traffic handling solution.
- Existing solution of traffic handling systems were based on real data, which are enlisted as below,
    1. Pre - Timed Traffic Control System
    2. Vehicle Actuated Control System
    3. Adaptive Traffic Control System
- The main issue with all of the above enlisted solutions is that, they all are real time which are fully dependent on the cameras and traffic controllers installed at the cross roads.
- If any of the utilities will be failed working due to any hardware issue or any other malfunctioning then it won’t be able to provide effective solution to manage the traffic.
- Therefore, there is a need for ‘Predictive Analytics for Traffic’ which is based on the historical data i.e. one step further advanced solution to solve the issue.

1.4 Motivation

- The motivation behind this research-based internship project in the sector of traffic is to undertake one of the challenges posed by the Government of India and IRC (Indian Road Congress) by providing an appropriate and effective solution of the problem.
- The ‘Smart City Mission’ also motivated to perform research in this area. The opportunity to learn about the real-life application of computer engineering to solve the problems faced by the citizens of India motivated me to work in this research-based project.

1.5 Scope and Objective

- The business scope for the predictive analytics for traffic provided by the client is to give the prediction of traffic flow at particular time intervals.
- The research-based project will generate different types of reports in various formats which makes analysis of data easier.
- The project will perform prediction on the data at the end of the day and generate the results.
- The project will be in fully automated mode i.e. it won't require any developer to run the programs to perform various tasks.
- The objective of the research-based project is to provide a proper and efficient solution to solve the issue of traffic raised by the Government of India under 'Smart Cities Mission'.
- Another objective is to reduce the failures of traffic handling at the junction. If the camera of the junction fails to capture the number of vehicles at particular arm and unable to decide the green time for that particular arm so by using predictive analytics, we can predict the number of vehicles to define green light and red-light time for that arm.

1.6 Planning

- The planning of the project is carried out in different phases which are enlisted as below,
  1. Data Collection – This phase includes the collection of data coming from the camera.
  2. Data Storage – This phase consists of the storage of the data of camera into database.
  3. Data Cleaning – This phase performs various data cleaning tasks such as, handling of missing data, checking of particular time formats etc.
  4. Data Separation – This phase performs data separation tasks on the basis of various factors of the data coming from the camera.
  5. Data Processing – This phase consists of processing of data to perform logic when the camera is off.
  6. Data Modelling – This phase performs various statistical modelling and machine learning techniques to provide the prediction of the data.
  7. Data Visualization – This phase will provide various graphical visualizations to have a better understanding of the prediction.

CHAPTER 2 Literature Review & Comparative Study

2.1 Evolution of the topic

- Predictive analytics has been around for over 75 years, but just recently hit mainstream status.
- It is currently being utilized across industries and functional areas, such as: insurance underwriting, fraud detection, risk management, direct marketing, upsell and cross-sell, customer retention, collections, at-risk patient determination, and so much more.
- Adoption is being driven by two key factors:
    1) The explosion of data – both structured and unstructured – inside and outside corporate walls.
    2) A plethora of new technologies that make it easy and affordable to store, clean, combine, explore, and visualize the data in real-time.

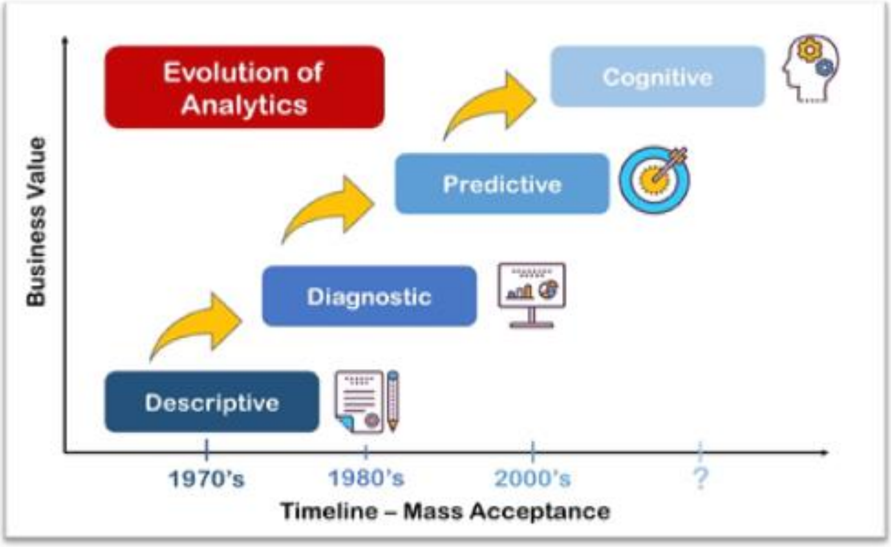


Figure 2.1 Evolution of Analytics

- Descriptive: the most basic form of analytics that illustrates historical trends.
- Diagnostic: the next level up which allows users to dig deeper and uncover root causes.
- Predictive: the development of mathematical algorithms using weightings and scores to make predictions about the future; utilizes techniques from data mining, statistics, modeling, machine learning, and artificial intelligence.
- Prescriptive: the application of recommendations based on predictive insights.

2.2 Background Study (Theoretical and Mathematical Background):

Background study for predictive analytics for traffic requires some amount of statistical knowledge and machine learning skills which is described as below,

1. Time Series:

- Time Series forecasting models are used to make predictions of future values based on previously observed / historical values.
- Main two goals of the timeseries are the identification of the phenomenon represented by the sequence of observations, and the forecasting of future values in the time series variable.
- The pattern of observed time series data is identified, described and integrated / combined with other data.
- The identified pattern is further inferred to predict future events.
- Time Series predictive models are used to make forecasts where the temporal dimensions are critical to the analysis.
- Typical application scenarios of time series are demand prediction of a product during a particular month / period, estimation of inventory costs, forecast of train passengers for the next financial year, and so on.

2. Decision Tree Algorithm:

- A decision tree is a supervised machine learning model used to predict a target by learning decision rules from features. As the name suggests, we can think of this model as breaking down our data by making a decision based on asking a series of questions.
- Based on the features of training set, the decision tree model learns a series of questions to infer the class labels of the samples.
- Decision tree algorithm is a CART (Classification and Regression Trees) algorithm means it is used in both classification and regression problems.
- A decision tree is constructed by recursive partitioning starting from the root node (known as the first parent), each node can be split into left and right child nodes. These nodes can then be further split and they themselves become parent nodes of their resulting children nodes.
- So, how do we know what the optimal splitting point is at each node?
- Starting from the root, the data is split on the feature that results in the largest Information Gain (IG) (explained in more detail below). In a repetitive process, we then

repeat this splitting procedure at each child node until the leaves are pure i.e. samples at each node belongs to the same class classified by the decision tree.

- This result in a very deep tree with many nodes, which can easily lead to overfitting problems. Thus, we want to prune the tree by setting a limit for the maximal depth of the tree.
- In order to split the nodes to perform prediction and classification at the most informative features, we need to define an objective function that we want to optimize via the tree learning algorithm.

$$IG(D_p, f) = I(D_p) - \left(\frac{N_{left}}{N_p} I(D_{left}) + \frac{N_{right}}{N_p} I(D_{right})\right)$$

Figure 2.2 Decision Tree Information Gain

- Main objective of the function is to maximize the information gain at each split. Here, f is the feature to perform the split, Dp, Dleft, and Dright are the datasets of the parent and child nodes, I is the impurity measure, Np is the total number of samples at the parent node, and Nleft and Nright are the number of samples in the child nodes.
- Information gain is explained as the difference between the impurity of the parent node and the sum of the child node impurities the lower the impurity of the child nodes, the larger the information gain.

3. **XGBoost Algorithm:**

- Generating a large amount of data has become a need to develop more advanced and sophisticated machine learning techniques.
- There are three types of boosting Adaptive Boosting, Gradient Boosting & XGBoost.
- Adaptive Boosting is implemented by combining several weak learners into one strong learner. Adaptive boosting starts by assigning equal weight edge to all of the data points and on the basis of that a decision stump is drawn out for a unique input feature, so the next step is the results that you get from the first decision stump which are analyzed.
- If any observations are misclassified, then they are assigned higher weights this correctly. After that new decision stump is drawn by considering the representations of higher pressures as more significant.

- So whichever data point was misclassified they are given a higher weight it in the next step you'll draw another decision stump that tries to classify the data points by giving more importance to the data points with more upper weight age.
- Adaptive Boosting will keep looping until all the observations will fall into the right class. The end goal of adaptive boosting is to make sure that all your data points are classified into the correct courses.
- Gradient boosting is also based upon the sequential and symbol learning model. The base learners are generated sequentially such that the present learner is always more effective than the previous one. The overall model improves sequentially with each iteration now.
- The difference in this boosting is that the weights for misclassified outcomes are not incremented. Instead, in gradient increasing we try to optimize the loss function of the previous learner by adding a new adaptive model that combines weak learners.
- This happens to reduce loss function. The main idea here is to overcome the errors in the previous learner's prediction.
- Gradient Boosting has three main components. The loss function is the one that needs to be optimized (Reduce the error) you have to keep adding a model that will regularize the loss function from the previous learner. Just like adaptive boosting gradient boosting can also be used for both classification and regression tasks.
- XGBoost performs missing data handling by filling them with appropriate values.
- XGBoost is the advance version of gradient boosting.
- The main purpose behind the usage of XGBoost algorithm is to increase speed and to increase the efficiency of your competitions.
- XGBoost was introduced because the gradient boosting algorithm required prolonged rate to perform computation of the output as it performs sequential analysis of the data set due to which it takes a longer time.
- XGBoost focuses on your speed and your model efficiency. To do this, XGBoost has a couple of features.
- It supports parallelization by creating decision trees. There's no sequential modelling in computing methods for evaluating any large and any complex modules.

2.3 Review Previous Research Findings

- By reviewing various research findings on the topic of 'Prediction Analytics for Traffic' it is found out that the researchers have lots of amount of data.

- The researchers have carried out forecasting of traffic flow on 10-minutes, 30-minutes, 60-minutes and monthly time intervals data.
- It is found that to perform short term time series forecasting lots of data is needed to get more accurate results.
- By reviewing various time series forecasting based research papers it is found that first of all researchers find out which type of patterns exist in the time series i.e. trend, seasonality or cyclical.
- To find out the existing patterns in the time series research findings consists decomposition of time series and graphical representations to get the idea of the exhibiting patterns in the timeseries.
- It is found out that to apply any time series-based forecasting models the time series is considered as stationary which is created after detrending or deseasonalising by removing trend and seasonality from the time series respectively.
- On the basis of the exhibiting patterns in the timeseries data it is found out that ARIMA, SARIMA and FBProphet are used to provide more appropriate forecasting result.
- It is found out that various time series forecasting models use GridSearch approach to hyper tune the parameters on the basis of Akaike Information Criterion (AIC).

2.4 Comparative Study

- Existing research findings in the field of ‘Predictive Analytics for Traffic’ when performs time series forecasting at that time the researchers divide the data in pre-defined time intervals such as 10-minutes, 15-minutes, 30-minutes and hourly.
- Comparatively to perform time series forecasting for every minute and every five minutes we have divided the whole data in the intervals of 1-minute and 5-minutes respectively.
- To perform time series forecasting on 10-minutes interval data researchers take seasonality period as 24*6 where, 24 is the total number of hours in the day and 6 represents that there exists 6 datapoints in one hour for 10-minutes intervals.
- Comparatively to apply various time series forecasting models and approaches we have applied seasonality period of 24*60 = 1440 for 1-minute interval of timeseries data, where 24 is the total number of hours in the day and 60 represents the number of datapoints per hour for 1-minute intervals.
- Comparatively to apply various time series forecasting models and approaches on five minutes intervals of time series data we have considered seasonality period of 24*12 =

288 for 5-minute interval of timeseries data, where 24 is the total number of hours in the day and 12 represents the number of datapoints per hour for 5-minute intervals.

2.5 Open Issues (Research Gaps)

- There are not any specific available models available regarding short term time series prediction.
- According to our research we have to perform the forecast for the short time span so, this is one of the major research gaps which with we have to deal.
- Some of the prediction models fail to provide forecast for the short time intervals because most of the models fail to perform forecast due to the limitations of the models.

CHAPTER 3 Experimentation / Simulations / Lab Set Up

Research-based project ‘Predictive Analytics for Traffic’ is set up on the server having required configuration. For the lab set up we have installed MySQL and Python and other required utilities on the server. During lab set up I have carried out collection of camera data, storage of data in database, cleaning of data is performed. While in simulations data separation and data processing phases are carried out. The data generated from the above phases is used in various forecasting models to perform experiments to carry out prediction. All the phases are described in brief as below,

1. Data Collection

- Data is one of the main asset of data analytics. For the research-based project ‘Predictive Analytics for Traffic Flow’ cameras offered by china based Dahua company are installed at ‘Ch – 5 Circle, Gandhinagar’.
- The firmware of the Dahua camera is designed in such a way, that there are two processes are running in the background of the camera – one for image processing which detects the number and the types of the vehicles and other various factors and the other process sends the data detected by the camera on the URL.
- The data comes in the streaming form i.e. at particular interval camera sent the data on the allocated URL.
- For, the project the time interval for the streaming of the Dahua cameras were set to 1 second.
- The cameras were sending the data at the interval of 1 second whether it detects any vehicle or not. If none of the vehicles will be detected by the camera then camera will send ‘ ’ value for those particular fields.
- The data streaming on the URL is the combination of JSON and Text formatted data. The JSON formatted data contains the necessary fields while the Text formatted data consists the details of the camera.

2. Data Storage

- The data storage of the camera data is carried out using Python Script and the data is stored in the MySQL database.
- MySQL is one of the most popular open source relational Structured Query Language database management system. MySQL is fast and easy-to-use Relational Data Base Management System (RDBMS) used for small as well as big applications.

- MySQL uses a standard form of the well-known SQL data language, and it also works well even with the large data sets too.
- MySQL is customizable. Default file size limit for a table is 4GB in MySQL which can be increased up to 8 million terabytes if the operating system is capable for that. The fields which are stored in the database are as follows,
    1. OccuredTime – It shows the time at which the data came from camera. There is an UTC field which is converted in the human readable format.
    2. OccuredDate – It stores the date at which the data came from the camera.
    3. Day – It contains the day name of the week.
    4. SystemTime – It stores the system time of the system in which the database is implemented i.e. server.
    5. Event – Event represents the junction name. Junction is the combination of 4 or more than 4 cross roads where, the traffic cameras installed.
    6. Camera_IP – It is the manually added field to differentiate cameras.
    7. MachineName – It is the unique name for each machine i.e. camera installed at each arm.
    8. Arm – It is the arm number i.e. 1,2,3,4 for particular junction having 4 cross roads where cameras are installed.
    9. VehicleType1 – It represents the vehicle detected at the lane-1 by the camera for particular arm.
    10. VehicleType2 – It represents the vehicle detected at the lane-2 by the camera for particular arm.
    11. VehicleType3 – It represents the vehicle detected at the lane-3 by the camera for particular arm.
    12. CountLane1 – It shows the number of vehicles detected at the lane-1 for particular arm.
    13. CountLane2 – It shows the number of vehicles detected at the lane-2 for particular arm.
    14. CountLane3 – It shows the number of vehicles detected at the lane-3 for particular arm.
    15. TotalCount – It represents the total number of vehicles detected at particular second by specific arm.

16. PCU – PCU stands for Passenger Car Unit, which is a metric used in the Transportation Engineering. It is a unique value defined for each specific types of vehicles. In ‘Predictive Analytics for Traffic’ it is used to count the time for which traffic signal will show red light and green light. The algorithm for deciding the red light and green light time is developed as an internal logic by taking the reference of IRC standards.
17. Speed1 – It stores the speed of the vehicles detected by the camera for lane1 of specific arm.
18. Speed2 – It contains the speed of the vehicles detected by the camera for lane2 of specific arm.
19. Speed3 – It stores the speed of the vehicles detected by the camera for lane3 of specific arm.
20. QueueLength – It contains BackofQueue detected by camera for particular camera when the traffic signal shows the red light.

- Python is one of the interpreted, high level general-purpose programming language. It consists of lots of libraries which provides different types of functionalities.
- The research project has used some of the popular libraries to perform specific tasks. The libraries used for the Data Collection from the camera and the Data Storage purpose are enlisted as below,

  1. requests –
     - requests module provides functionalities to send HTTP requests using Python.
     - A session is created using requests.Session() to persist certain parameters across requests. It also persists cookies across all requests made form the Session instance, and it will use urllib3’s connection pooling to store them.
     - Connection pooling will result in significant performance increase if several requests are made to the same host, the underlying TCP connection will be reused.
     - To check whether the successful connection is established or not with the provided URL of the camera API the concept of Exceptional-Handling is used.
     - When the successful connection is established with the URL the response returned by the camera will be ‘< Response: 200 >’, if the response

returned by the URL is something else then it represents that there is some issue while establishing connection.

- The data coming from the camera is in the streaming form i.e. at every second data coming from the camera appends at the end of the previous data at the particular allocated URL.
- To establish connection with the URL and to access the streaming data coming from the camera is handled by the requests.get() request.
- For authentication purpose authentication parameters are passed in it. As the data hitting on the URL contains text and JSON formatted data but the JSON formatted data consists of the necessary fields.
- So, there is a need to consider only JSON formatted data and ignore text data which is done through iter_lines() and strip() methods, iter_lines() iterates through each and every line of the response and strip() method strips out the text formatted data.
- The JSON formatted data is stored in the temporary JSON file which is parsed later on and after that parsed data is stored in the MySQL database.

2. json –
    - json module is used to load the data from the JSON file.
    - After loading the data JSON parsing is performed on that and required fields are stored in the MySQL database.
3. mysql.connector –
    - The module handles all the database related functionalities such as, creation of database if not exists, creation of table to store various fields, to establish connection with the database etc. All of the above operations are performed by writing SQL queries.
    - After successfully parsing each of the necessary data fields, parsed data is stored in the MySQL table by performing write operation on the database.
    - To avoid the duplicate entry of the parsed data various conditions are implemented.
    - If the requests module fails to get the response from the URL for particular second then manual entry of the data will be entered with ‘None’ value for categorical fields and ‘0’ for numerical fields.

  - The main reason to carry out manual entry is to track the status of the camera – one can get to know that during this time the camera was off or there was some issue in the camera.

4. calendar –
    - This module is used to get the weekday from the date which is also stored in the database as a separate field in the table.
5. time –
    - time module is imported to perform necessary sleep operations as per the need, it will halt the particular process for some amount of time.
6. multiprocessing –
    - multiprocessing module is imported to carry out the concept of multiprocessing to handle the data coming from all of the URL's at the same time.
    - It will allocate separate memory for each of the processes. So, all the process will work as a separate program.

3. Data Cleaning

- In 'Predictive Analytics for Traffic' data cleaning is also carried out during the data storage phase.
- During data cleaning missing data handling, checking of proper time format and some other tasks are carried out.
- In missing data handling manual data entry is performed to track the status of the camera.
- The time at which data occurred from camera is in the UTC format so, it is necessary to convert UTC time format in to human readable form to get the idea regarding the occurred time at which data came from the camera.

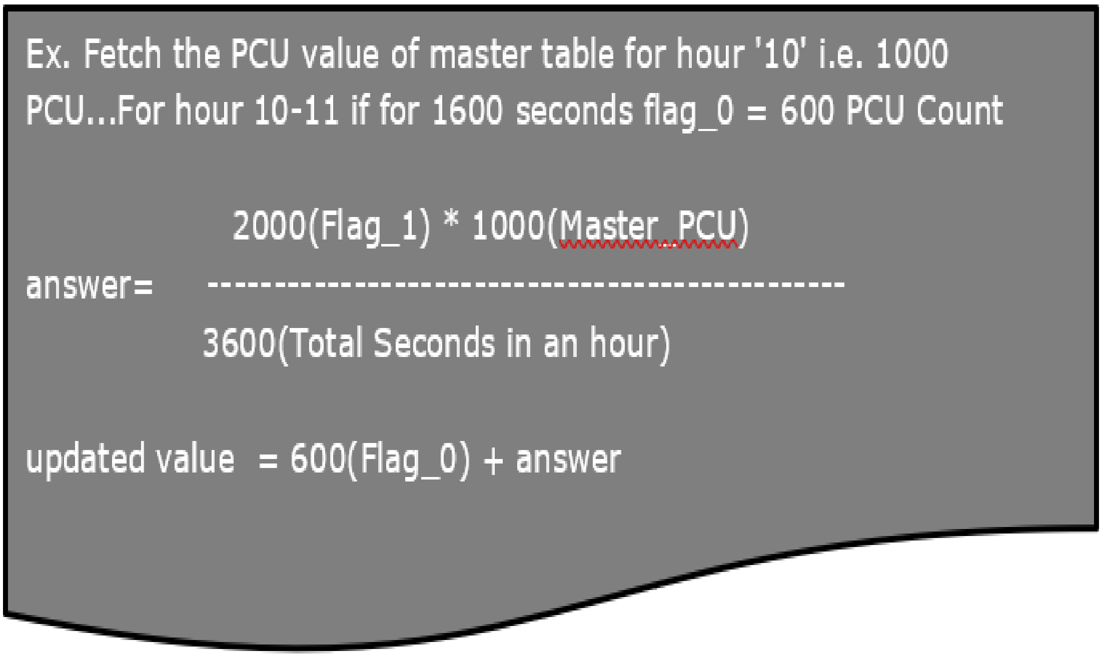
4. Data Separation

- In data separation whole data stored in the database is separated on the basis of various factors.
- Data separation is performed to get the data arm wise, junction wise and on the other factors to generate various reports and graphs for data visualization part.
- In this research project the data coming from the camera was in the intervals of seconds.
- To carry out prediction hourly and minutely, the data must be converted in minute wise and hour wise format which is also carried out as a module of this phase.

- In this phase the data separation is carried out in such a way that data is divided in train and test data sets as per the internal logic developed by the team.
- As per the logic the train data set will contain 80 % of the data to train the model and test data set will store rest 20 % of the data.
- As per the logic the prediction models will run at the end of the day only for once and the logic is designed in the FIFO format i.e. First In First Out.
- When the data of the latest date will come then that will be appended in the test dataset and the first date’s data of test dataset will be transferred to the train dataset and the first date’ data of train data will be removed. The whole logic is developed to create the train dataset and test dataset dynamically.

5. Data Processing

- Data processing is the phase which will be carried out every time. This phase contains one of the internal logics developed to handle the real-life scenarios if the camera will not work during some duration either due to technical issue or due to maintenance task.
- This phase includes the creation of the master table (Table based on the internal logic). Master table will store the PCU value for each arm for each day and time parameter (i.e. hour for hourly_master_table and minute for minutewise_master_table).
- The table will be updated every day before performing the data modelling module.
- The value of the master table will be updated as per below formulas,



Ex. Fetch the PCU value of master table for hour '10' i.e. 1000 PCU...For hour 10-11 if for 1600 seconds flag_0 = 600 PCU Count

$$\text{answer} = \frac{2000(\text{Flag\_1}) * 1000(\text{Master\_PCU})}{3600(\text{Total Seconds in an hour})}$$

updated value = 600(Flag_0) + answer

Figure 3.1 Master Table Internal Logic

6. Data Modelling

- Data modelling is the main module of the ‘Predictive Analytics for Traffic’ as it performs the prediction part on the data. Various algorithms used in this phase are described in the next chapter in brief.

7. Data Visualization

- Data visualization is the module which will generate reports in various formats i.e. in .CSV file and in graphical representation to get the idea of the prediction results.

CHAPTER 4 Proposed Hypothesis / Model / Algorithm

4.1 Algorithm

- All the algorithms implemented during the research-based project ‘Predictive Analytics for Traffic’ are described here in detail.
- Through data modelling in the research-based project formation of train and test dataset is done by using algorithms based on internal logics.
- Train and test dataset are built on the basis of 20 days and 10 days intervals. The module will create datasets dynamically at the end of the day.
- The dataset will be updated each and every day, which I would like to explain through one example. If currently train dataset consists data of 1-20 April, and currently test dataset contains the data of 21-30 April then at the end of the 1st may when the program will run dynamically then it will append the data of 1st may at the end of the test dataset and data of 21st April will be appended at the end of the train dataset and 1st April’s data from train dataset will be removed.
- The reason behind the creation of FIFO (First in First Out) algorithm is that if there is bridge construction work is going on then there will be more traffic but when the bridge construction will complete then it may possible that most of the traffic will be diverted on the bridge. This will result in the sudden decline in the number of traffic flows.

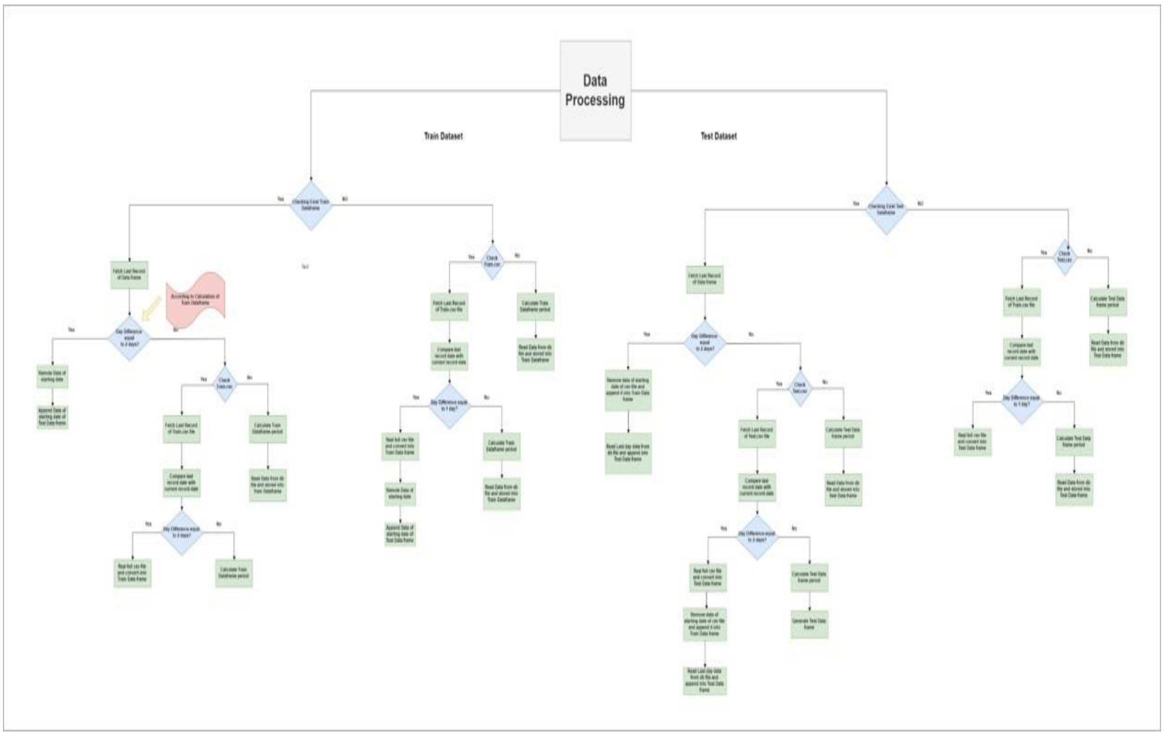


Figure 4.1 Train Dataset and Test Dataset Generation Algorithm

- Various statistical approaches, time-series methods and machine learning models are used to perform the prediction task on the traffic data. The algorithms are implemented in Python. Algorithms which are used to perform prediction of time series data are as listed below,
    1. Decision Tree Regression
    2. XGBoost Algorithm
    3. ARIMA (Auto Regressive Integrated Moving Average)
    4. SARIMA (Seasonal Auto Regressive Integrated Moving Average)
    5. FbProphet (Facebook Prophet)
- From the above listed algorithms Decision Tree Regression and XGBoost are machine learning algorithms while ARIMA, SARIMA and FbProphet are time-series based statistical algorithms.
- To predict the red light and green light signal time one of the important or key parameter is PCU count so, there is no need to consider Vehicle Type, Speed, Queue Length of traffic and other parameters.
- So, to predict the PCU count necessary parameters are the Occurred Time, either Arm number to predict the PCU for particular Arm or Event to get the junction name and the PCU value.

4.2 Implementation Details (Tools and Technologies used)

- Time is the major parameter on which the PCU count varies i.e. the traffic will be higher during the morning and evening while it may possible that the frequency of the vehicles passing through the signal will be less during noon and in the midnight.
- So, to perform forecasting of the traffic various machine learning and statistical based time-series methods are used in the research-based project ‘Predictive Analytics for Traffic Flow’.
- All the algorithms are implemented in the Python using appropriate libraries such as, Pandas to work with datasets (.CSV files), NumPy to work with numerical operations, Sklearn to implement various machine learning models, statsmodels to implement statistical models.
- During this whole project we have utilized GridSearch to hyper tune the parameters to choose the best parameters for each of the models.

Machine Learning Approaches:

- Various machine learning models such as, Linear Regression, Polynomial Regression (Non-Linear Regression), Decision Tree Regression, Gradient Boosting techniques and other approaches are used for predictive analytics to forecast the data.
- At which time which algorithm to use depends on the data.
- In case of traffic data, after plotting the data there doesn’t exists any linear relationship so, it is not possible to apply Linear Regression.
- After applying Non-Linear Regression, I get to know that model overfits the data most of the time i.e. model fits fully on the train data set but doesn’t able to provide proper results on test dataset. So, we get to know that non-linear regression will also won’t work.
- Decision Tree Regressor and one of the powerful gradient boosting approaches i.e. XGBoost are implemented on the data to forecast the values of traffic flow which are described below.

  1. **Decision Tree Regression:**

  - Decision Tree is one of the supervised machine learning techniques.
  - Decision Tree is a type of CART (Classification and Regression Technique).
  - Decision Tree is a decision-making tool which makes decisions on the basis of flowchart-like tree structure.
  - Decision Tree Regression observes features of an object and trains a model in the structure of a tree to predict or forecast the data values to produce meaningful and effective continuous outputs.
  - Decision Tree Regression in python is implemented using sklearn library, dataset import and conversion of that into data frame is performed using pandas, various numerical operations are performed using NumPy.
  - The output after implementing Decision Tree Regression is as shown below,

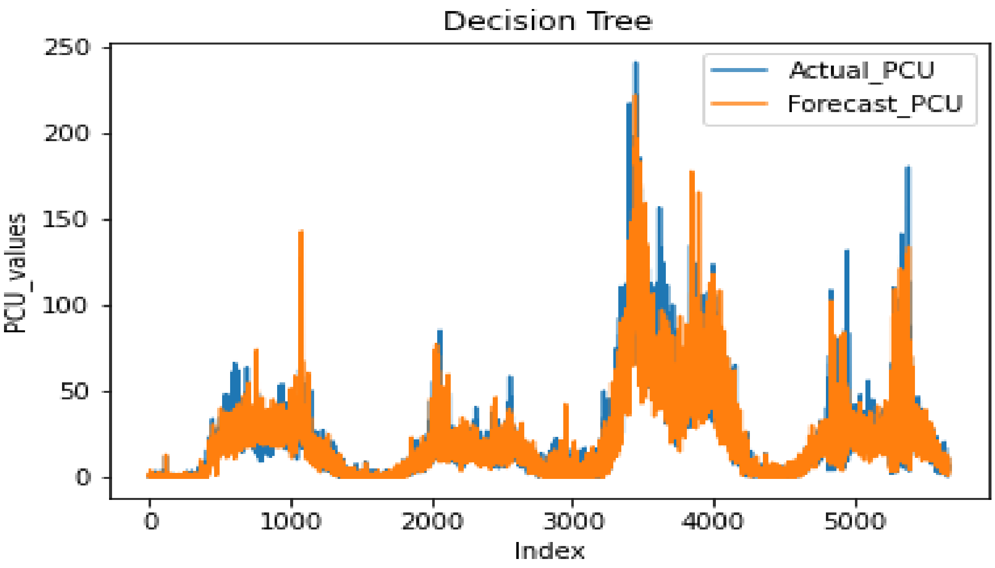


Figure 4.2 Decision Tree Algorithm Implementation

- Drawback of decision tree regression is discussed in the next chapter.

2. **XGBoost Algorithm:**

- Boosting is one of the machine learning technique which is used to solve complex data driven real-world problems.
- Three types of boosting algorithms are: Adaptive Boosting, Gradient Boosting and XGBoost.
- Adaptive Boosting is implemented by combining some weak learners in one single strong learner. Adaptive boosting starts by assigning equal weight edge to all the data points and the decision stump for a unique input feature is drawn out.
- If any observations are misclassified then they are assigned higher weights. After that new decision stump is drawn by considering the representation of higher pressures as more significant.
- Adaptive boosting will keep looping until all of the observations will fall into the right class. The end goal of the technique is to classify the data correctly.
- Gradient Boosting is based on sequential and symbol learning model. In this technique base learners are generated in such a way that the present learner is always more effective than the previous one. The overall model improves sequentially with each new iteration.
- The difference in this technique is that the weights for misclassified outcomes are not incremented, instead in this technique the loss function of the previous learner is optimized by adding a new adaptive model that combines various weak learners.

- The main reason behind reducing the loss function is to overcome the errors in the previous learner's prediction. Gradient Boosting is also used for both classification and regression.
- XGBoost is the advance version of gradient boosting.
- XGBoost has the tendency to fill the missing values of the data.
- The main aim of the XGBoost is to increase the speed and increase the efficiency of computations.
- XGBoost was introduced to overcome the limitation of Gradient Boosting i.e. the long time consuming due to sequential analysis of the data set to generate appropriate outputs.
- To increase the speed and efficiency XGBoost has couple of features. It supports parallelization by creating decision trees.
- Decision Tree Regression observes features of an object and trains a model in the structure of a tree to predict or forecast the data values to produce meaningful and effective continuous outputs.
- After choosing appropriate parameters XGBoost is implemented and the result is shown as per the below image,

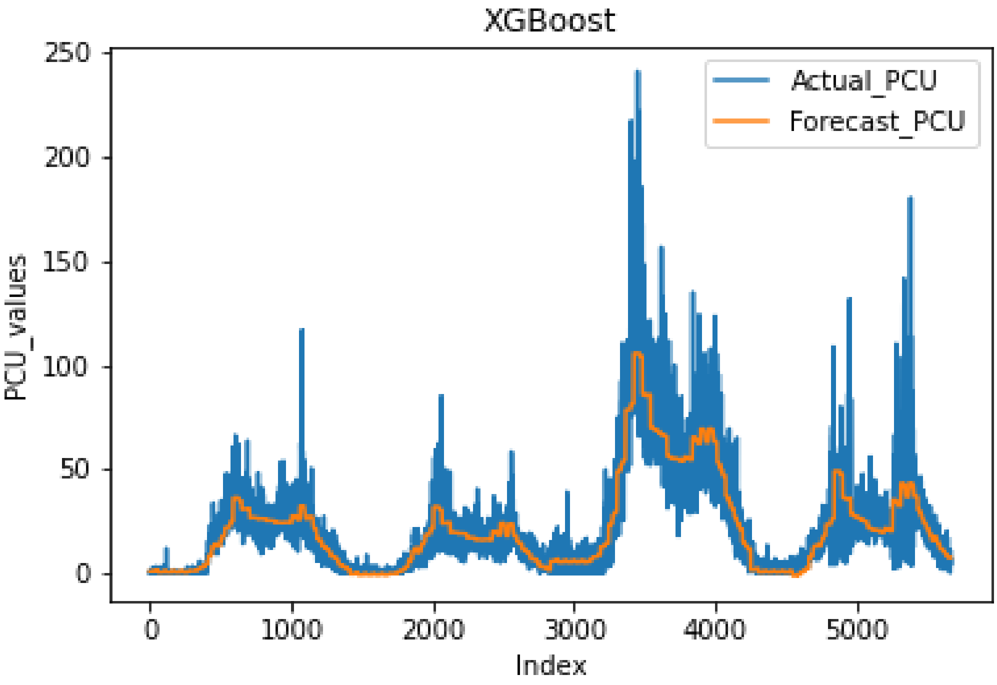


Figure 4.3 XGBoost Algorithm Implementation

Time Series based statistical approaches:

- To get the idea of which time series model to apply time series decomposition and other statistical approaches are used during unified modelling.
- Time series always inherits different patterns i.e. Trend, Seasonality / Cycle, Residual / Error.
- Every timeseries can be expressed as the combination of Trend, Seasonality, Residual which can be shown by the decomposition.
- Timeseries observation is expressed either additive or multiplicative or the combination of both additive and multiplicative of Trend, Seasonality and Residual as shown below,

  Additive: Observed = Trend + Seasonality + Cycle + Residual

  Multiplicative: Observed = Trend * Seasonality * Cycle * Residual

- Time series which follows long term pattern is known as trend.
- The time series which follows short term pattern is known as seasonal / cyclical. The difference between seasonal and cyclical is that, seasonal follows repetitive patterns at particular fixed interval while in cyclical time series follows repetitive patterns at irregular intervals.
- Time series decomposition decomposes the time series into Trend, Seasonality / Cycle and Residual / Error.
- Time series decomposition of the traffic data is as shown below,

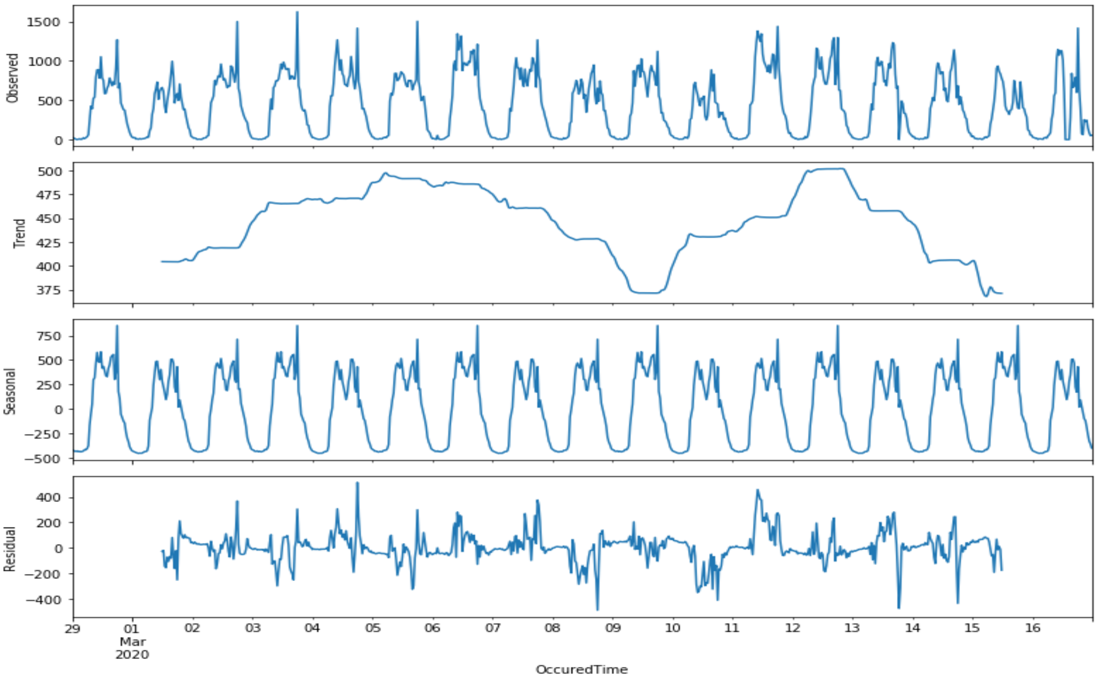


Figure 4.4 Time Series Decomposition

- Before applying any statistical forecasting model, the assumption is taken as the timeseries data must be stationary i.e. there should not exist trend and seasonality / cyclicity.
- As per the above decomposition it is shown that there exists seasonality in the time series data. To get the idea about stationarity various statistical tests are performed i.e. Augmented Dicky Fuller Test (ADF Test).
- So, to make the time series stationary by removing the seasonality various statistical methods are used i.e. moving average, exponential smoothing, differencing and many more.
- As per the main objective of the research we have to predict the traffic for the next day for different intervals of time i.e. for T + 5 minutes, T + 10 minutes, T + 15 minutes, T + 30 minutes, T + 1 hour.
- Here, we have to forecast the data for small amount of time intervals it is known as short term time series forecasting. Various time series forecasting models used during this research-based project are as below,

  1. **ARIMA (Auto Regressive Integrated Moving Average):**

  - ARIMA models (which consists of AR and MA models) are a general class of models to forecast stationary time series. ARIMA models are made of three parts:
  - AR part consists weighted sum of lagged values of the series.
  - MA part consists weighted sum of lagged forecasted errors of the series.
  - I part stands for the difference of the time series.
  - An ARIMA model is often noted ARIMA (p, d, q) where p represents the order of the AR term, d the order of differencing (I part), and q the order of the MA term.
  - The Autoregressive Integrated Moving Average (ARIMA) method models the next step in the sequence as a linear function of the differenced observations and residual errors at prior time steps i.e. combination of AR and MA parts.
  - It combines Auto regression (AR) and Moving Average (MA) models as well as a differencing pre-processing step of the sequence to make the sequence stationary, called integration (I).
  - The notation for the model involves specifying the order for the AR(p), I(d), and MA(q) models as parameters to an ARIMA function, e.g. ARIMA (p, d, q). An ARIMA model can also be used to develop AR, MA, and ARMA models.

- The method is appropriate for univariate time series with trend and without seasonal components.
- The Arima model in python is implemented using pandas, matplotlib, NumPy, statstools and other libraries.
- After installing the above libraries, we have used pandas to read the csv file which is grouped on minute of interval and store the csv file in dataframe(df).
- The columns on the dataframe are Occurred time, Arm, Total Count, PCU.
- Now we have to convert occuredtime column to datetime using datetime method and the datetime format will be of dd-mm-yyyy HH:MM:SS i.e. date-month-year Hour : Minute : Second..
- After converting occurred time column to datetime we will perform statistics test to check whether the time series is stationary or not. For checking stationarity, we will use Augmented Dicker fuller's test in which we will check p-value of the data.
- There is another test Kwiatkowski–Phillips–Schmidt–Shin (KPSS) tests to check the stationarity of the data.
- Most popular statistical test to check whether a given time series is stationary or not is Augmented Dickey Fuller test (ADF Test). It is one of the most commonly used statistical tests when it comes to analyzing the stationarity of a time series.
- So, we will check the p-value if the p-value is lesser than 0.05 then the data is stationary else the data is not stationary and you have to do differencing to make the data stationary.
- Now we will start implementing ARIMA model on one-minute interval data.
- The first step of fitting an ARIMA model is to determine the differencing order to stationaries the series. To do that, we look at the ACF plot, and keep in mind these two rules:
  1. Rule 1: If the series has positive autocorrelations out to a high number of lags, then it probably needs a higher order of differencing.
  2. Rule 2: If the lag-1 autocorrelation is zero or negative, or the autocorrelations are all small and pattern less, then the series does not need a higher order of differencing. If the lag-1 autocorrelation is -0.5 or more negative, then it indicates that the series may be over differenced.

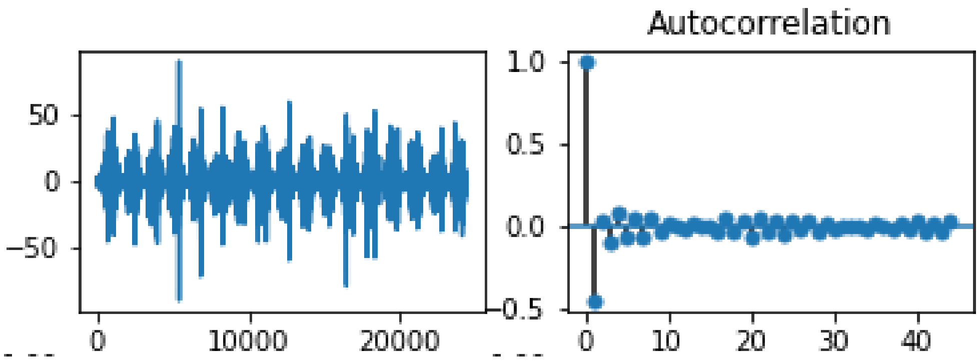


Figure 4.5 Time Series Differencing and ACF

- As It is easily shown in the image differencing and the corresponding Auto Correlation Function (ACF) is plotted. As per the above rules the autocorrelation should be less than -0.5 if it exceeds then time series is over differenced.
- So, when first order of differencing is applied on the one-minute interval data then ACF fulfills the condition i.e. the time series becomes stationary in first order differencing.
- The second step of fitting an ARIMA model is to determine the MA order.
- Now we know we have to include a 1st order difference in our model, we need to choose the Moving-Average order.
- This is done by looking at the differenced series (because we just saw that the first-order difference series was stationary).
- Again, we look at our ACF & PACF plots, with this rule in mind:

1. RULE: If the lag-1 autocorrelation of the differenced series ACF is negative, and/or there is a sharp cutoff, then choose a **MA** order of 1.

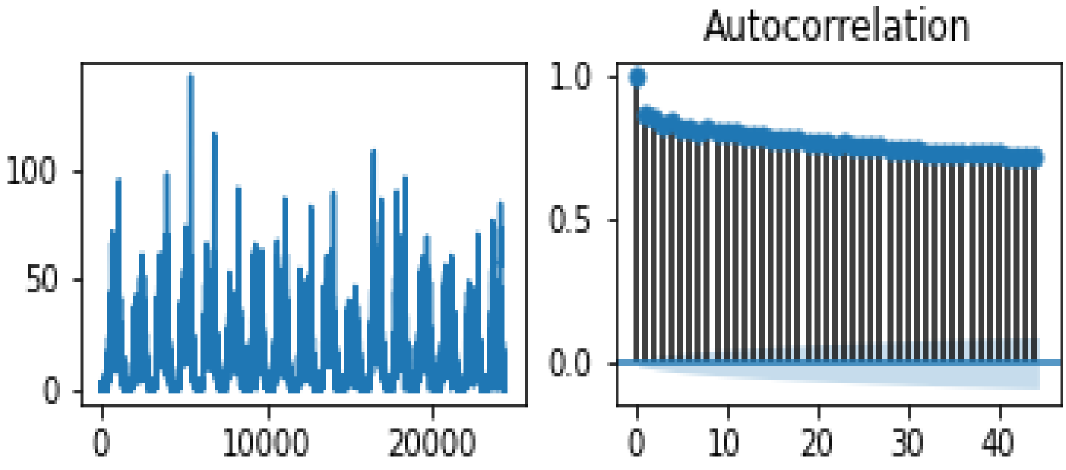


Figure 4.6 Original Time Series and ACF

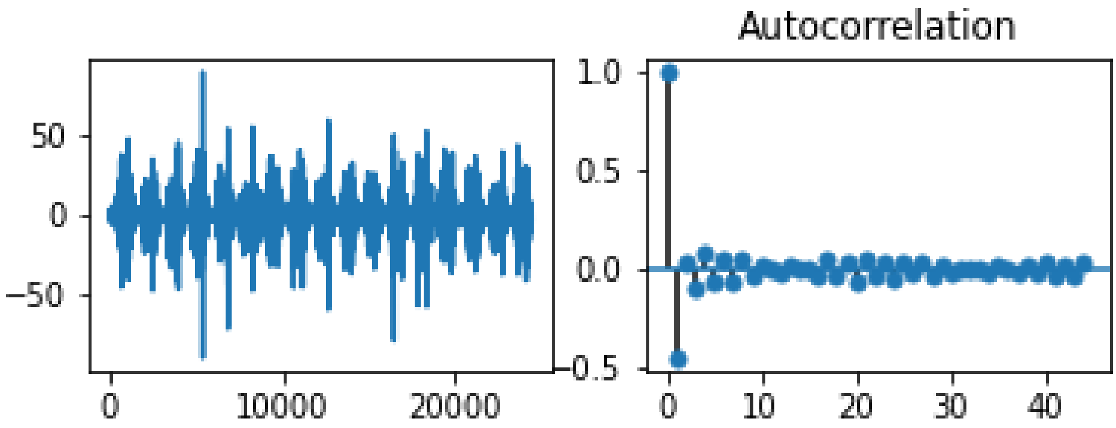


Figure 4.7 1st Differencing of Time Series and ACF

- As it is seen in the image, I have plotted MA without any differencing so as per the above rule the autocorrelation shouldn't be negative and the autocorrelation shouldn't be linear.
- So, by looking at the original series the rule is already satisfied so no need to change MA order to 1.
- But for understanding purpose I have used MA order still my time series is not going negative or linear so we can go with 1st MA order.
- The third step of fitting an ARIMA model is to determine the AR order.
- Now the question arises that when to consider AR term? The answer is given as per the following rule,
  1. RULE: If the lag-1 autocorrelation of the differenced series PACF is negative, and/or there is a sharp cutoff, then choose a AR order of 1.

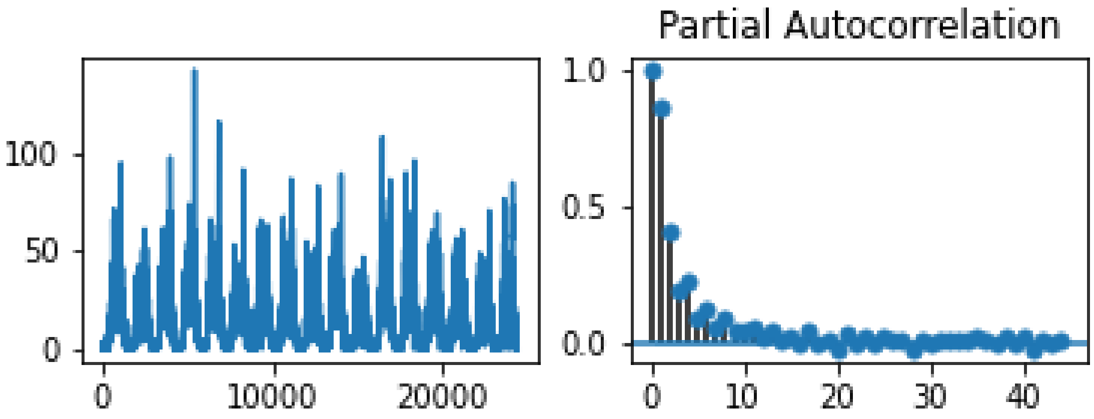


Figure 4.8 Original Time Series and PACF

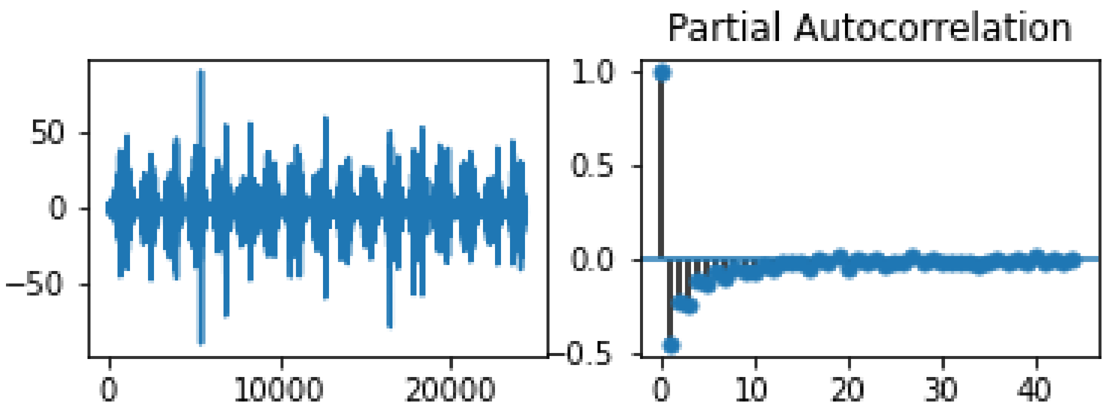


Figure 4.9 1st Differencing of Time Series and PACF

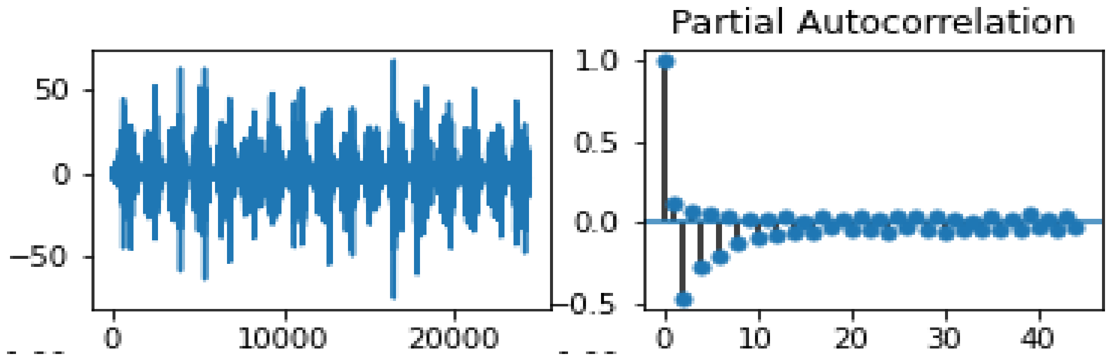
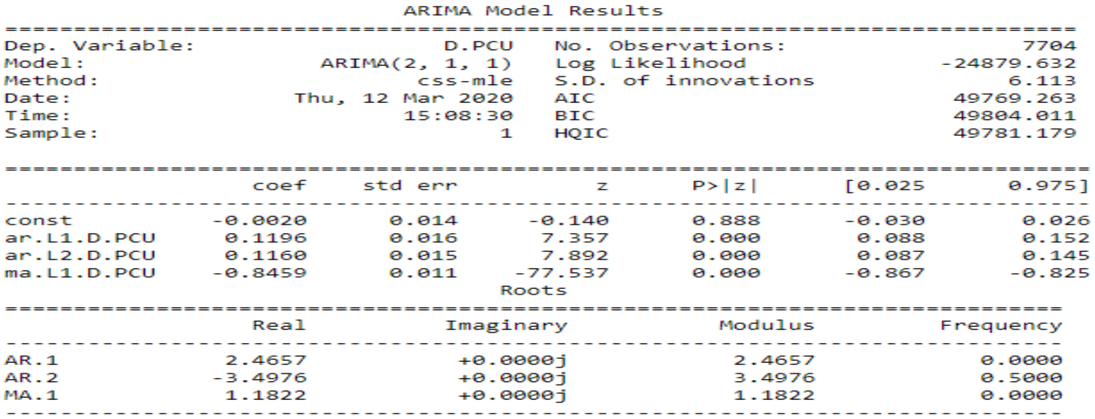


Figure 4.10 2nd Differencing of Time Series and PACF

- As it is shown in the image, I have plotted AR so as per the above rule the partial autocorrelation shouldn't be negative and linear in nature.
- So, by looking at the original series the rule is already satisfied so no need to change AR order to 1.
- But for understanding purpose I have used AR order 1 still my time series is not going negative or linear so no need to go with 1st order.
- So now we will fit the Arima model with (2, 1, 1) order and check the results.



```
                             ARIMA Model Results
==============================================================================
Dep. Variable:                  D.PCU   No. Observations:                 7704
Model:                 ARIMA(2, 1, 1)   Log Likelihood              -24879.632
Method:                       css-mle   S.D. of innovations              6.113
Date:                Thu, 12 Mar 2020   AIC                          49769.263
Time:                        15:08:30   BIC                          49804.011
Sample:                             1   HQIC                         49781.179

================================================================================
                  coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------------
const          -0.0020      0.014     -0.140      0.888      -0.030       0.026
ar.L1.D.PCU     0.1196      0.016      7.357      0.000       0.088       0.152
ar.L2.D.PCU     0.1160      0.015      7.892      0.000       0.087       0.145
ma.L1.D.PCU    -0.8459      0.011    -77.537      0.000      -0.867      -0.825
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1            2.4657           +0.0000j            2.4657            0.0000
AR.2           -3.4976           +0.0000j            3.4976            0.5000
MA.1            1.1822           +0.0000j            1.1822            0.0000
-----------------------------------------------------------------------------
```

Figure 4.11 ARIMA result summary

- After fitting the Arima model on (2, 1, 1) order we have to check Akaike's Information Criterion.
- The lower the AIC then we will have better results. To get lower AIC we will perform grid search to get the lowest AIC.
- So, to do grid search to get lowest AIC we will take p, d & q and provide the range from 0 to 3 and loop through it.
- So, we get 10 to 15 different combinations to apply on Arima for lowest AIC.
- After doing grid search for lowest AIC from the above different combinations we got.
- Then the prediction results of Arima model looks like these:

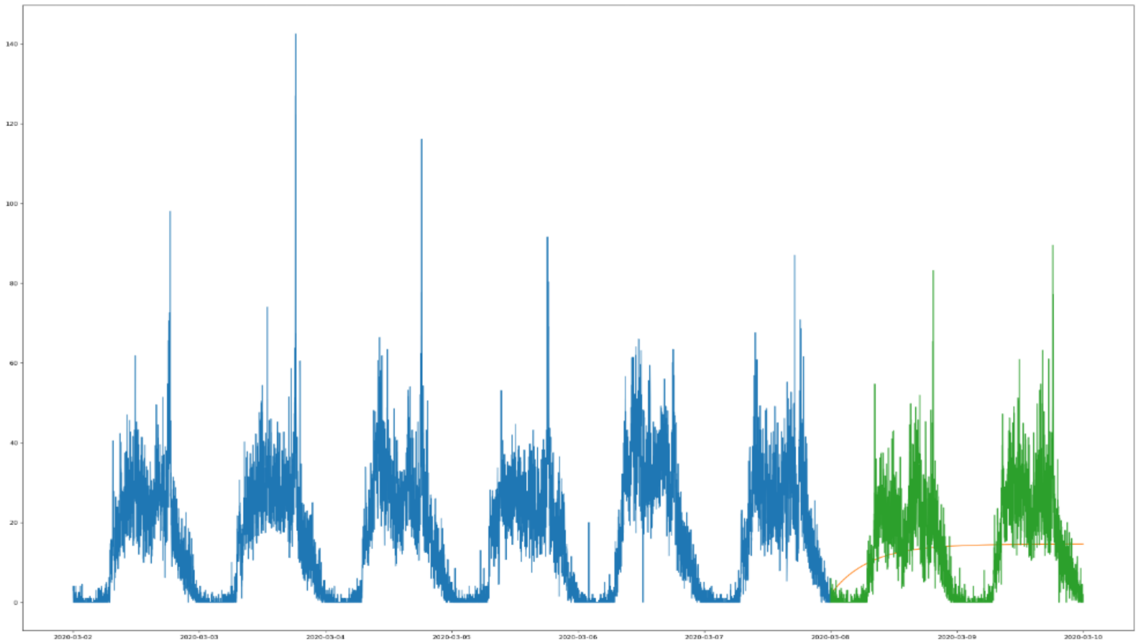


Figure 4.12 Forecasting by ARIMA

- The orange line is the prediction performed by ARIMA model.
- It is clearly visible that the ARIMA model is not giving better results because of the seasonality factor in the data.
- So, we have to take SARIMA model because of the seasonality in the data.

2. SARIMA (Seasonal Auto Regressive Integrated Moving Average):

- SARIMA is also known as Seasonal Auto Regressive Integrated Moving Average i.e. Seasonal ARIMA.
- As per the time series decomposition it is easily shown that there exists seasonality the data. Due to which ARIMA doesn't provide proper prediction. So, SARIMA is used to overcome that problem.

- Before applying SARIMA various other simple forecasting techniques are used to get the idea of the predicted output of SARIMA.
- First of all, basic, NAÏVE approach is used to forecast the traffic flow.
- In NAÏVE approach last period's actual values are used as the forecast, without adjusting them.

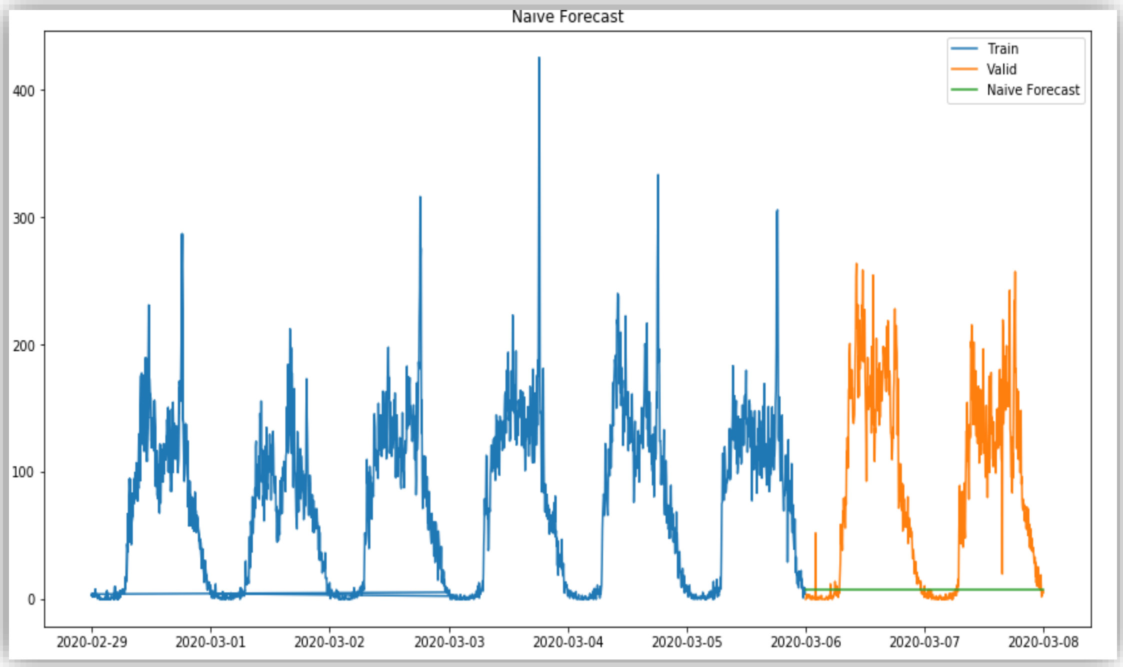


Figure 4.13 Forecasting of Time Series using Naïve Approach

- The next technique used is Simple Average Forecast.
- Here, the forecasts of all future values are expressed as the average of the historical data.

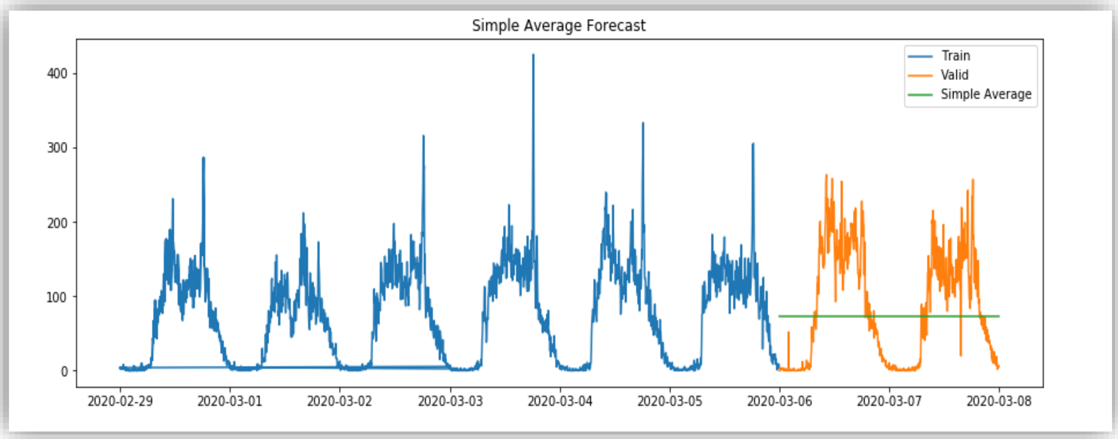


Figure 4.14 Forecasting of Time Series using Simple Average method

- The next approach is the same average approach by taking mean value as 5, 10, 60.

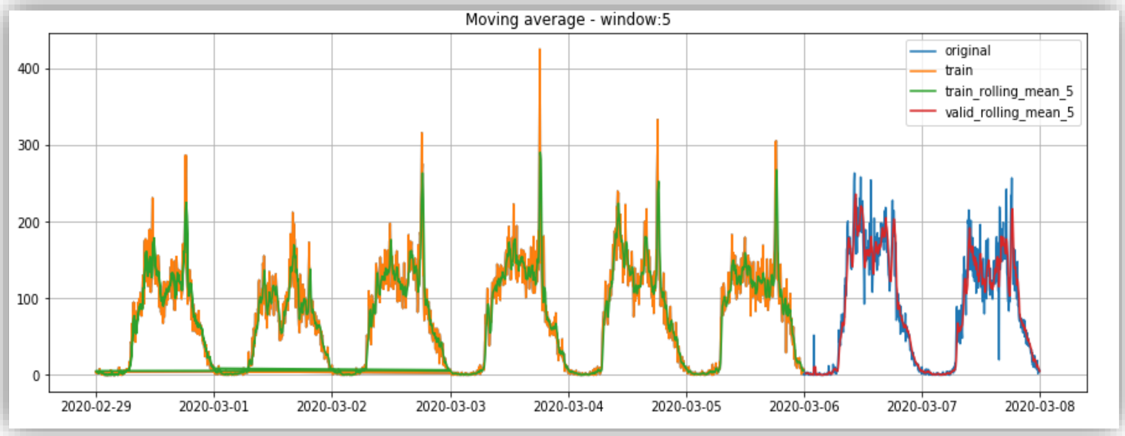


Figure 4.15 Forecasting using Moving Average of 5 window size

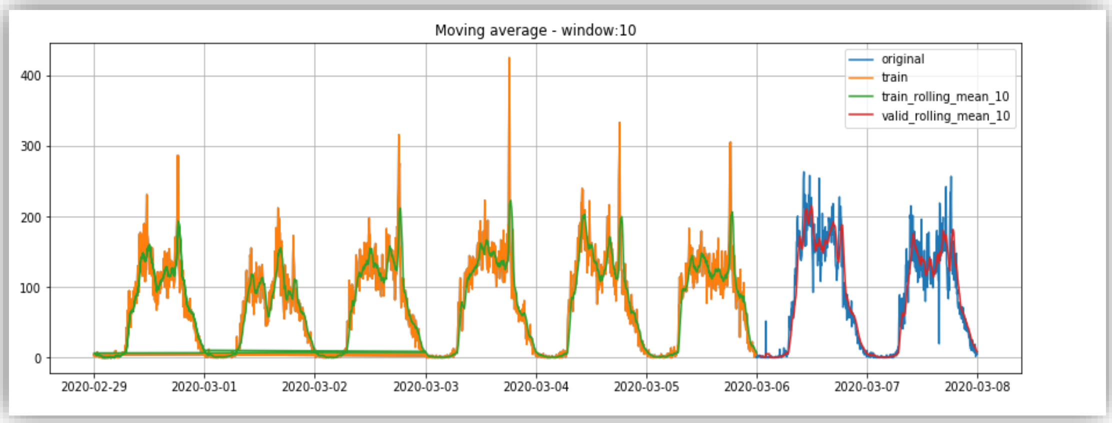


Figure 4.16 Forecasting of Time Series using Moving Average of 10 window size

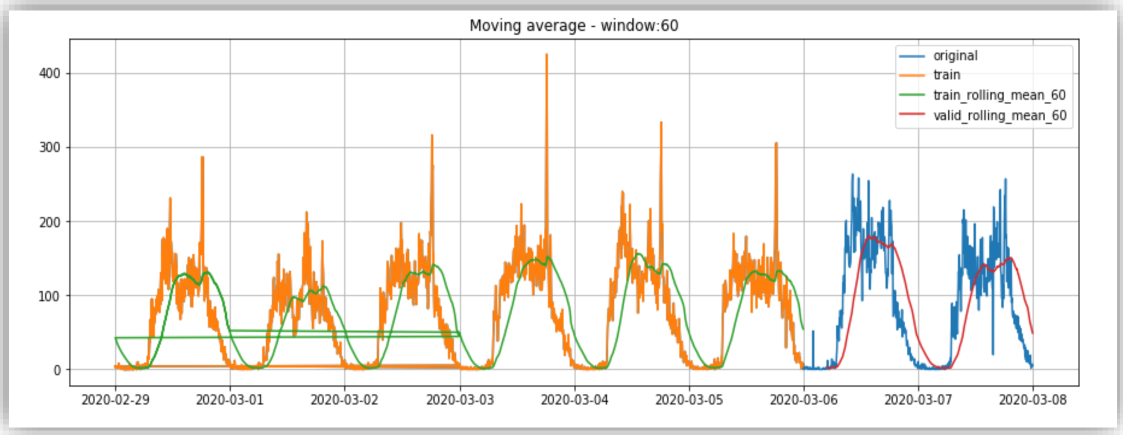


Figure 4.17 Forecasting of Time Series using Moving Average of 60 window size

- Next approach is Simple Exponential Smoothing. Simple Exponential Smoothing techniques works well when the time series shows a clear trend and / or seasonal behavior, which is not presented in the traffic data.

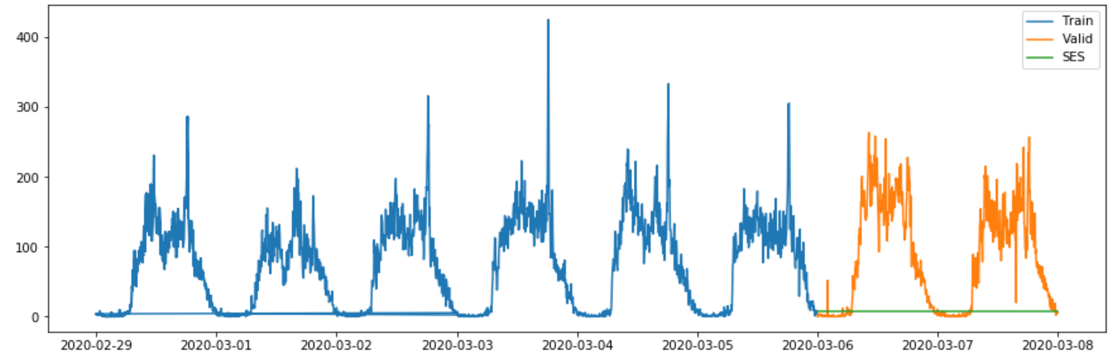


Figure 4.18 Forecasting of Time Series using Simple Exponential Smoothing

- Next technique applied is Holt Linear's Method.

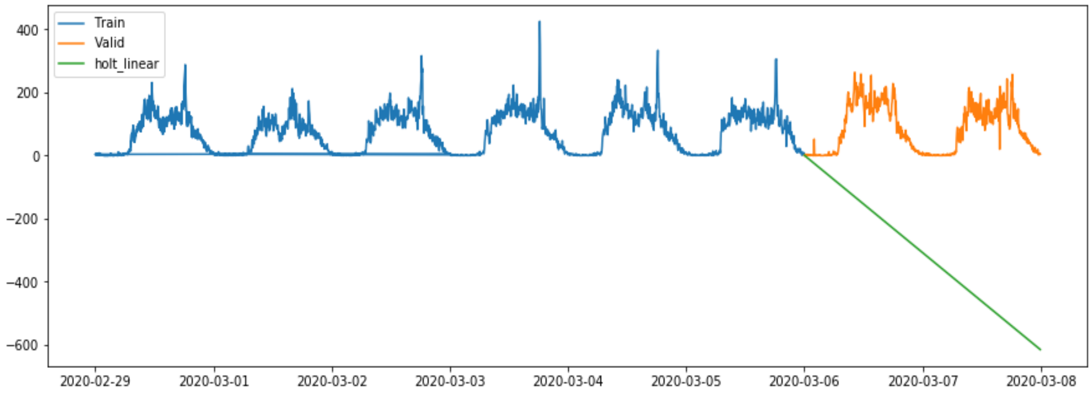


Figure 4.19 Forecasting of Time Series using Holt Linear's Approach

- Next technique is the Holt Winter Method.

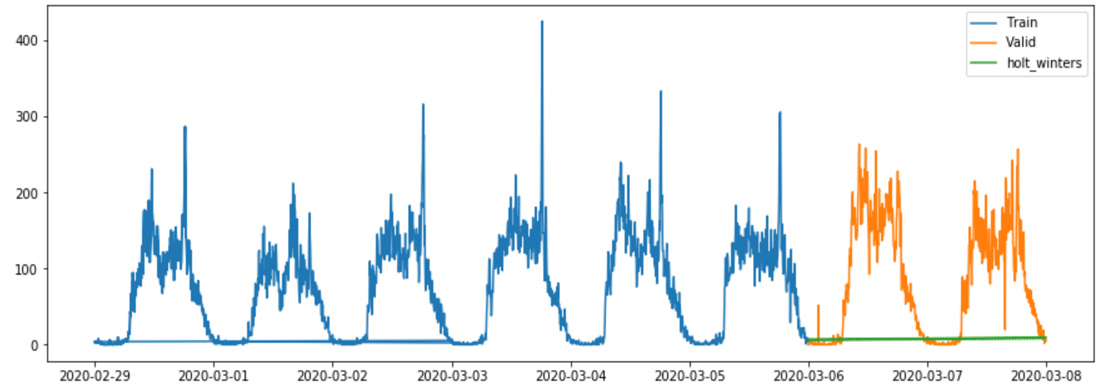


Figure 4.20 Forecasting of Time Series using Holt Winters Approach

- The above method doesn't seem to capture the overall multiplicative seasonality and trend patterns in the data, on looking at the plot of the time series data one can see there is seasonal patterns and few irregular patterns, therefore we have applied SARIMA model for our forecasting.
- Parameters of SARIMA model are (P, D, Q, S) along with (p, d, q). The P, D, Q values are similar to the parameters described above but it's applied to the seasonality component of the SARIMA model. S is the periodicity of the time series i.e. 12 for yearly, 4 for quarterly.
- Here, as our data is based on 30 minutes so there will be 2 data points at each hour.
- So, for 24 hours 24 will be multiplied by 2 that will give 48 data points in one day. So, S term in the SARIMA will be 48.
- Likewise, if the data is based on one-minute interval then S will be 24*60 i.e. 1440. Same way, S parameter for 15 minutes interval will have 4 data points at each hour i.e. S will be 24*4 = 96.
- As, SARIMA works well on 30 minutes data which is plotted as below,

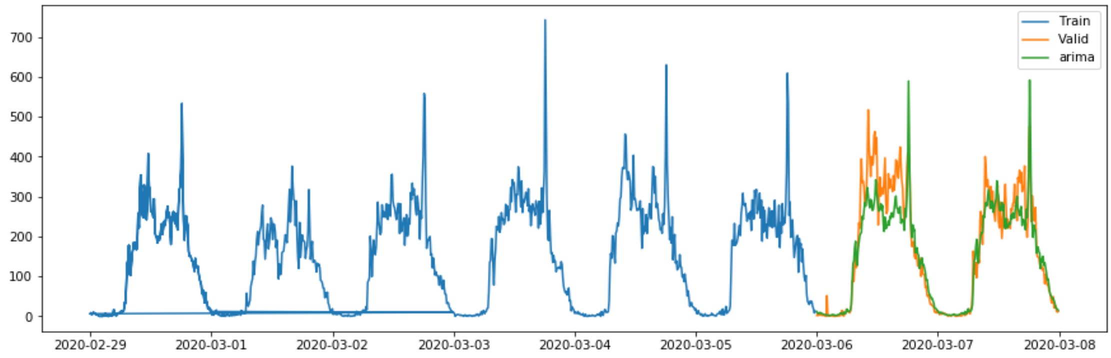


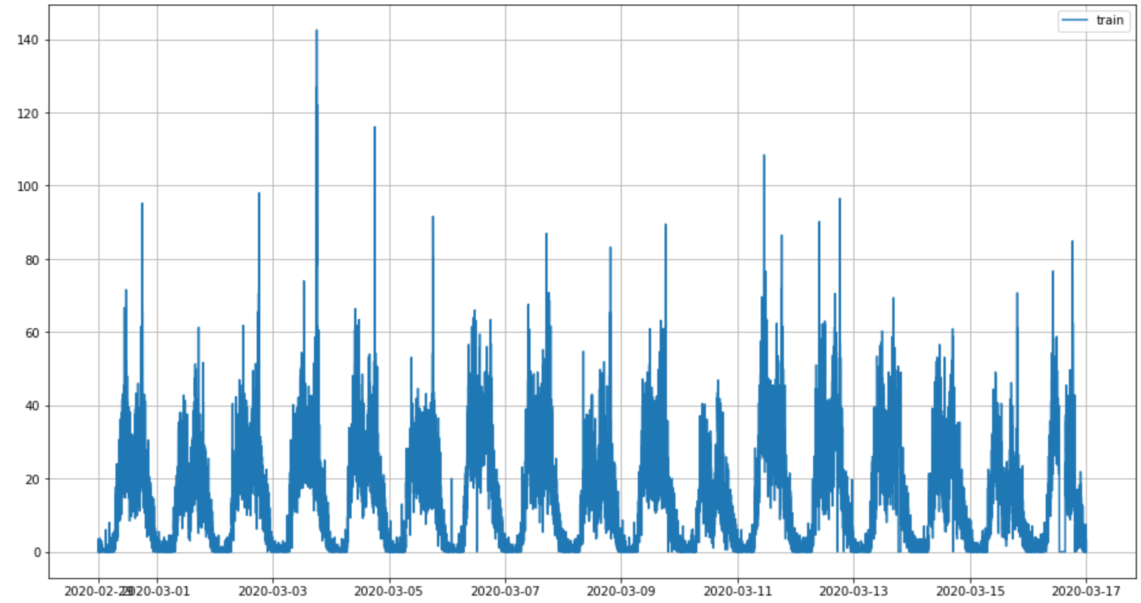
Figure 4.21 Forecasting of Time Series using SARIMA

- From the results one can say that, SARIMA is capable enough to identify the seasonality so next after performing parameter tuning, we have found the lowest AIC to get the better prediction.
- But SARIMA doesn't work well for short time interval spans i.e. for 1 minute and for 5 minutes due to large computational time, we need to search for other alternatives so, we used Facebook Prophet.

3. FbProphet (Facebook Prophet):

- As shown above, the ARIMA and SARIMAX fails to provide more appropriate output for the one minute time span although after taking large amount of time, there was a need for an alternative so, we – the team of research project searched out for alternatives and found Prophet and used it for short amount of time span i.e. for one minute time intervals to forecast the traffic flow.

- FbProphet stands for Facebook Prophet. This is the open source time series algorithm developed by research team of Facebook. Prophet decomposes time series into three components i.e. trend, seasonality and holiday. It has set of intuitive hyper parameters which are easy to tune.
- Prophet is interesting, sophisticated and quite easy to implement.
- Even by using the default argument or parameter, this model allows one to generate appropriate forecasting output with little effort or domain knowledge of time-series analysis.
- Prophet time series is expressed as, Trend + Seasonality + Holiday + error
- Trend models non periodic changes in the value of the time series.
- Seasonality in FBProphet represents the periodic changes like daily, weekly, or yearly seasonality.
- Holiday effect occurs on irregular schedules over a day or a period of days.
- The one of the limitations as of now is that Prophet works only for univariate time series i.e. it will take two parameters as inputs one of them will be the time dependent parameter and the other will be the predicted parameter. The research-based project is currently only on the basis of univariate time series forecasting Prophet is the perfect choice to use as it provides appropriate forecasting values as compared to other time series forecasting models for short amount of time span.
- The time dependent variable of Prophet is ds i.e. date stamp column which should be as per pandas datetime format, YYYY – MM - DD or YYYY – MM - DD HH: MM: SS for a timestamp.
- The predicted parameter is y which is of numeric type.
- Prophet follows sklearn model API to create an instance of the Prophet, to fit the data on Prophet Object and then predict the future values.
- The data look like these:



.

Figure 4.22 Time Series Plotting

- After the proper data formatting into ds and y it is divided in the train dataset to train the Prophet model and in test dataset to test the model.
- The train dataset consists of 15 days data while the test dataset contains data of 1 day as per the available data.
- Now Prophet is fit to the train data to train the model to perform forecasting.
- By default, FBProphet fits the data into a linear model. When forecasting grows, then there will be some points on their maximum achievable positions. So, parameter tuning is performed to get the better forecasting results.
- Besides the linear model, FBProphet can opt to use logistics growth trend model instead by changing its argument, logistics model is used when your data is non-linear in behavior.
- Define whether you want to use the linear model or logistics model (it is optional parameter). For this case, we won't use the logistics growth number and let the model use its default setting for the linear model.
- Other parameters are holidays, seasonality_mode, daily_seasonality, weekly_seasonality, yearly_seasonality, period, fourier_order etc. are set.
- For the research purpose we set the values of each and every parameter to True and get the idea of FB Prophet's working.
- When the daily_seasonality was set to True the period was set to 24 * 60, as each day consists of 24 hours and each hour consists of 60 minutes.

- Fourier_order is set to 1440 by multiplying 24 with 60 according to the seasonality period.
- Holiday is set to none you can set it if you want to include holiday or event.
- On FBProphet, you can make yearly, monthly, weekly, daily or even hourly predictions in one go in an easy way.
- As of now, after the research and findings we get the better results using Fb Prophet algorithm in forecasting of short time span i.e. of 1 minute.
- The actual values are shown by the black dots while the predicted value on the train dataset is shown by the blue line up to 15th march.
- The forecasting value by the Prophet algorithm is shown by the blue line i.e. the mean forecasted value and the upper bound and lower bound of forecasted value by sky blue shadow. As per the model the forecasted value will be lie between upper and lower bound of values.

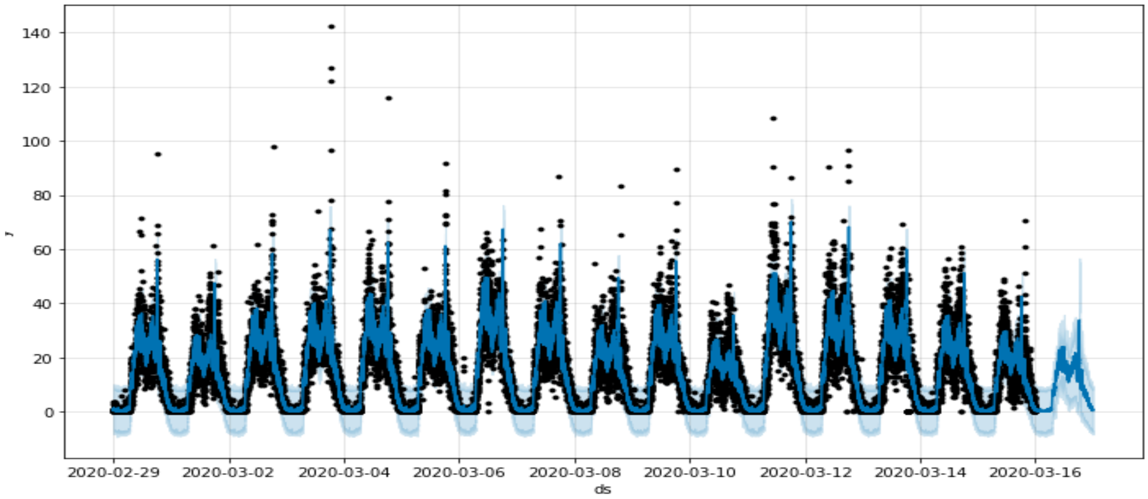


Figure 4.23 Forecasting of Time Series using FBProphet

- The reason behind using the FBProphet and other appropriate algorithms that is SARIMA is discussed in the next chapter.

4.3 Complexity Analysis

- In research-based project ‘Predictive Analytics For Traffic’, as per the aim of the research we wanted to forecast the traffic flow for the short term time intervals i.e. for 5 min, 10 min, 15 min and for some other time intervals which was difficult task as the data was in the time interval of seconds and at the time of conversion complex thing was the format after the conversion.

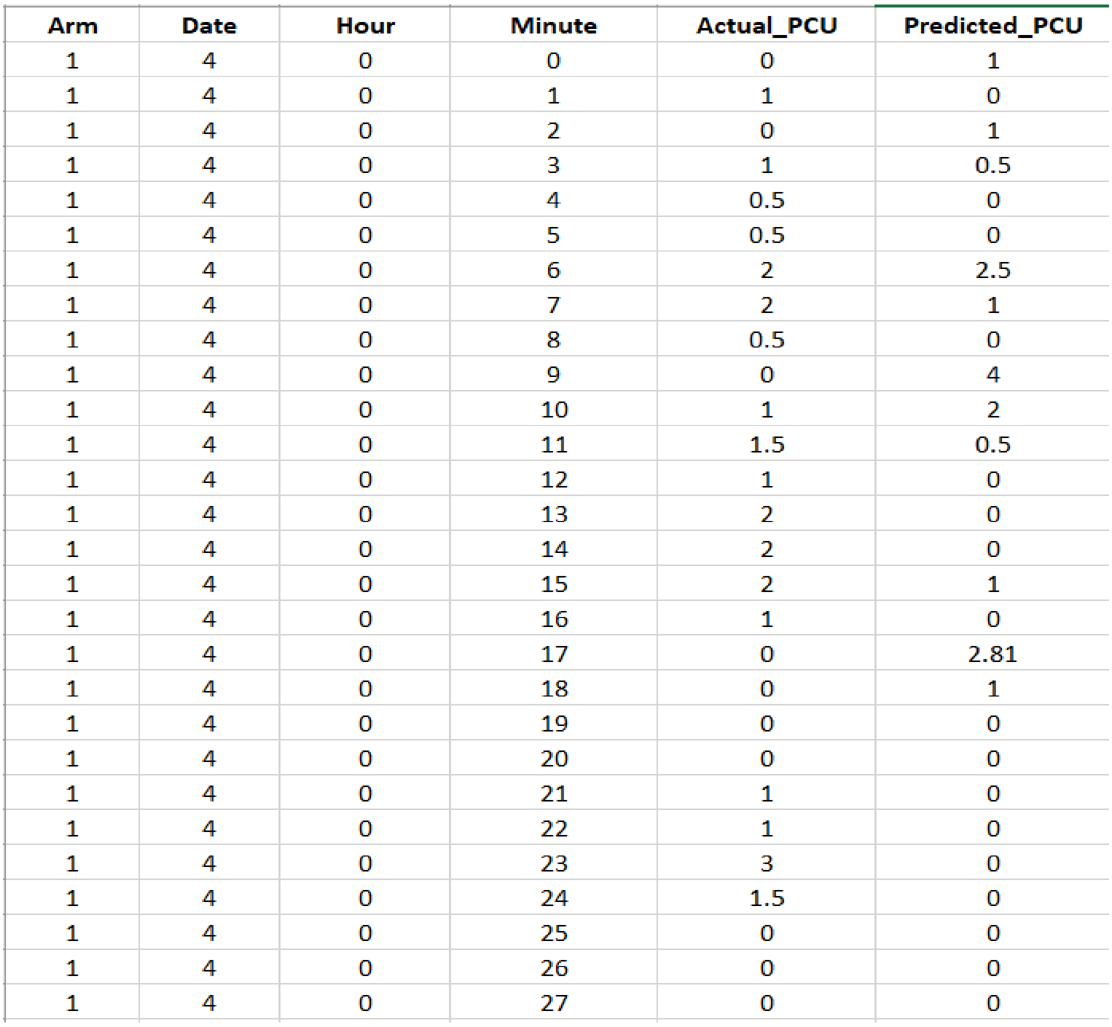
- Another complexity occurred while applying time series forecasting because before applying any time series models there should be one criterion which must be followed that is the removal of trend and seasonality. But after applying various techniques we were not able to achieve this fully.
- At the end of the day I have 24*60*60*4 = 3,45,600 total of entries so, to analyze these amounts of data can’t be handled using another data analysis tools such as Microsoft Excel which also increases the complexity.
- Due to the computational issues machines are not able to perform as we want which may result in the missing data and the handling of them is also a complex issue because most of the time series models require none of the missing values.
- While predicting for short time intervals i.e. 1 minutes, 5 minutes most of the time series models require complex computations which also require higher configurations that increases the complexity of the project.

CHAPTER 5 Results and Discussion on Results

- In this section results are shown which are obtained after the applying the selected algorithms in the various scenarios analyzed.
- Since the traffic data is collected at the interval of every seconds, prediction of traffic congestion has been made for 1 minute, 5 minutes, 10 minutes, 15 minutes and 30 minutes of intervals and the algorithms implemented during this algorithm are Decision Tree Regressor, XGBoost, ARIMA, SARIMA and FB Prophet.
- To check which machine learning model and statistical methods performs well or provides more accurate results various parameters are taken into account such as, RMSE i.e. Root Mean Square Error, R^2 (R-Squared) and some other values. R^2 is also known as the coefficient of determination.
- R^2 shows percentage variation in y i.e. dependent variables which is explained by all the x variables together. R^2 is one of the accuracy measures to get the idea of which model is more accurate. So, all the results of models will be compared on this basis.
- On the basis of various graphical visualizations, it comes to know that Decision Tree and XGBoost provides proper results up to some extent.
- To overcome their limitations various time series models are used to forecast the traffic flow which is described in details as below,

1. **Discussion on the results of Decision Tree Regression:**
    - Decision Tree Regression creates tree on the basis of Information Gain and Entropy for each of the parameters.
    - The decision tree is used to forecast the results on one-minute data.
    - The result generated by decision tree depends on various parameters such as,
        - ➢ max_depth which defines the maximum depth up to which the tree will expand.
        - ➢ Random_state defines the seed used by the random number generator, i.e. if it is set to None then it will take random value to get the data from the dataset to train the model.
        - ➢ And many more such as, min_samples_split, max_leaf_nodes, min_samples_leaf, criterion, etc.
    - From the above listed parameters max_depth is one of the important parameters, when max_depth is set to lowest value at that time it underfits the data and when max_depth is set to largest value at that time it overfits the data most of the time.

- Result forecasted by using Decision Tree Regressor is shown as below,



Arm	Date	Hour	Minute	Actual_PCU	Predicted_PCU
1	4	0	0	0	1
1	4	0	1	1	0
1	4	0	2	0	1
1	4	0	3	1	0.5
1	4	0	4	0.5	0
1	4	0	5	0.5	0
1	4	0	6	2	2.5
1	4	0	7	2	1
1	4	0	8	0.5	0
1	4	0	9	0	4
1	4	0	10	1	2
1	4	0	11	1.5	0.5
1	4	0	12	1	0
1	4	0	13	2	0
1	4	0	14	2	0
1	4	0	15	2	1
1	4	0	16	1	0
1	4	0	17	0	2.81
1	4	0	18	0	1
1	4	0	19	0	0
1	4	0	20	0	0
1	4	0	21	1	0
1	4	0	22	1	0
1	4	0	23	3	0
1	4	0	24	1.5	0
1	4	0	25	0	0
1	4	0	26	0	0
1	4	0	27	0	0

Figure 5.1 Decision Tree Result

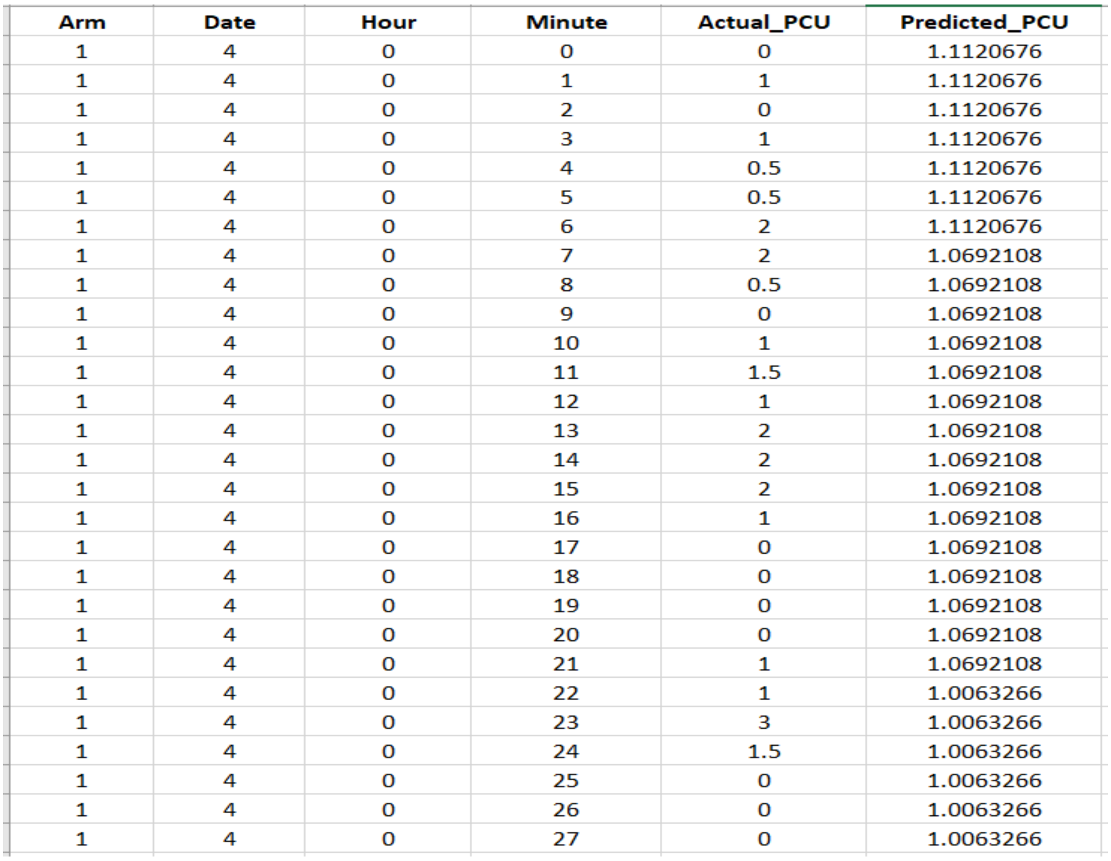
- Decision tree regression most of the time forecasts the data on the basis of the conditions such as, if there is any node available for hour 23 minute 12 then it will always return the fixed value of that particular node for every observations of hour 23 and minute 12.
- When the max_depth is increased at that time the model becomes subjective and fails to provide the appropriate results by overfitting the data.

2. Discussion on the results of XGBoost

- For better forecasting results, we jumped to XGBoost which is a robust boosting algorithm.
- The XGBoost is used to forecast the traffic flow for 1-minute time interval.
- The result generated by XGBoost depends on various parameters such as,

- ➢ max_depth which defines the maximum depth up to which the tree will expand.
- ➢ Learning_rate defines the step size shrinkage used to prevent overfitting. It ranges between [0,1].
- ➢ Colsample_bytree defines the percentage of features used per tree. High value can lead to overfitting.
- ➢ Subsample defines the percentage of samples used per tree. Low value can lead to underfitting.

- Result forecasted by using XGBoost algorithm is as shown below,



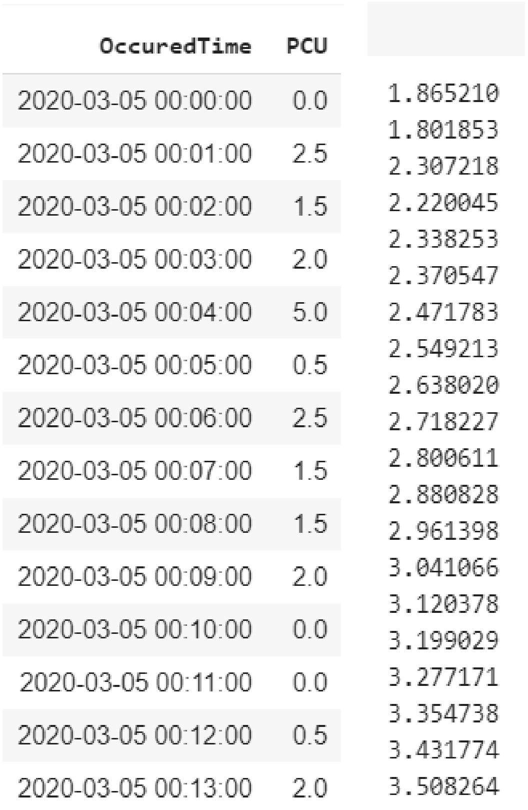
Arm	Date	Hour	Minute	Actual_PCU	Predicted_PCU
1	4	0	0	0	1.1120676
1	4	0	1	1	1.1120676
1	4	0	2	0	1.1120676
1	4	0	3	1	1.1120676
1	4	0	4	0.5	1.1120676
1	4	0	5	0.5	1.1120676
1	4	0	6	2	1.1120676
1	4	0	7	2	1.0692108
1	4	0	8	0.5	1.0692108
1	4	0	9	0	1.0692108
1	4	0	10	1	1.0692108
1	4	0	11	1.5	1.0692108
1	4	0	12	1	1.0692108
1	4	0	13	2	1.0692108
1	4	0	14	2	1.0692108
1	4	0	15	2	1.0692108
1	4	0	16	1	1.0692108
1	4	0	17	0	1.0692108
1	4	0	18	0	1.0692108
1	4	0	19	0	1.0692108
1	4	0	20	0	1.0692108
1	4	0	21	1	1.0692108
1	4	0	22	1	1.0063266
1	4	0	23	3	1.0063266
1	4	0	24	1.5	1.0063266
1	4	0	25	0	1.0063266
1	4	0	26	0	1.0063266
1	4	0	27	0	1.0063266

Figure 5.2 XGBoost Result

- From the above listed parameters max_depth, learning_rate and subsample are important parameters. To get the best fit model for the data GridSearch has been implemented.
- GridSearch takes lots of amount of time to give the best parameters.
- Although after choosing best parameters if there happens a small change in the training data then the previously trained model won’t work proper way. So, XGBoost also became subjective this way.

3. Discussion on the results of ARIMA

- ARIMA is a simple timeseries forecasting model which is the combination of AR and MA models with additional Integrated part (I).
- ARIMA is also said Non-Seasonal ARIMA as it doesn't contain any seasonal part.
- The decomposition of time series shows that, there exists a seasonal component in the data which can be removed by the integrated part of ARIMA using differencing.
- When the ARIMA applies to the appropriate differenced data it still doesn't work appropriately.
- As differenced time series data still contains the seasonal component, but to apply ARIMA on any timeseries data that must be stationary i.e. the time series should not have trend and seasonal component.



OccuredTime	PCU
2020-03-05 00:00:00	0.0
2020-03-05 00:01:00	2.5
2020-03-05 00:02:00	1.5
2020-03-05 00:03:00	2.0
2020-03-05 00:04:00	5.0
2020-03-05 00:05:00	0.5
2020-03-05 00:06:00	2.5
2020-03-05 00:07:00	1.5
2020-03-05 00:08:00	1.5
2020-03-05 00:09:00	2.0
2020-03-05 00:10:00	0.0
2020-03-05 00:11:00	0.0
2020-03-05 00:12:00	0.5
2020-03-05 00:13:00	2.0

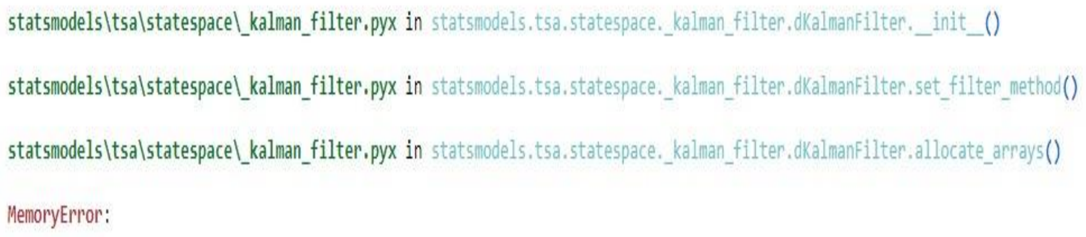
```
1.865210
1.801853
2.307218
2.220045
2.338253
2.370547
2.471783
2.549213
2.638020
2.718227
2.800611
2.880828
2.961398
3.041066
3.120378
3.199029
3.277171
3.354738
3.431774
3.508264
```

Figure 5.3 ARIMA 1-minute Results

- The result of ARIMA for 1-minutes time interval data is as shown above, which is not that much appropriate as non-seasonal ARIMA can't handle the seasonality of the timeseries data.

4. Discussion on SARIMA results

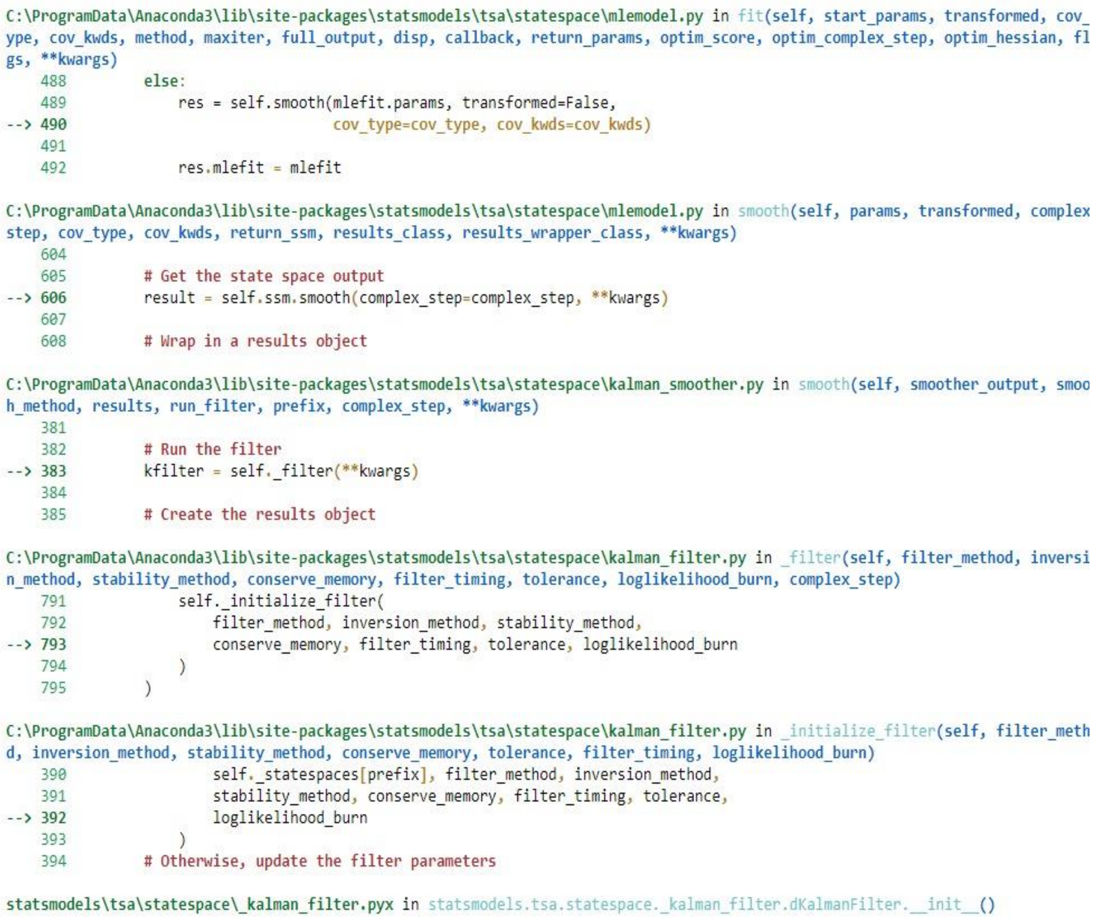
- From the decomposition it is shown that, there exists a seasonal component in the data due to which ARIMA fails to provide more accurate forecasting even after differencing the time series. So, it is necessary to consider SARIMA which includes one additional seasonal component.
- SARIMA is applied to 1 minute, 5 minutes, 15 minutes of time intervals data.
- The parameters on which results of the SARIMA depends are enlisted as below,
  - ➢ p = non seasonal AR order
  - ➢ d = non seasonal differencing
  - ➢ q = non seasonal MA order
  - ➢ P = seasonal AR order
  - ➢ D = seasonal differencing
  - ➢ Q = seasonal MA order
  - ➢ S = time span of repeating seasonal pattern
- In SARIMA value of S parameter will affect the performance i.e. if seasonality is taken according to one day then for 24 hours there will be 60 minutes so, S will be 24*60 i.e. 1440.
- With S value equals to 1440 i.e. for one day seasonality SARIMA takes more time approximately 48 hours or more to train the model. Although after taking this much amount of time it didn't provide the needed results so for seasonality of one day we need to search for other models.
- SARIMA doesn't work well on 1-minute time intervals data which is shown below, it provides Memory Error when it runs on the system having normal configurations.



```
statsmodels\tsa\statespace\_kalman_filter.pyx in statsmodels.tsa.statespace._kalman_filter.dKalmanFilter.__init__()

statsmodels\tsa\statespace\_kalman_filter.pyx in statsmodels.tsa.statespace._kalman_filter.dKalmanFilter.set_filter_method()

statsmodels\tsa\statespace\_kalman_filter.pyx in statsmodels.tsa.statespace._kalman_filter.dKalmanFilter.allocate_arrays()

MemoryError:
```

Figure 5.4 SARIMA 1-minute Error

- For 5-minute intervals of time there will be 12 intervals in each hour so, for 24 hours S will be 24*12 = 288.
- The result generated by 5 minutes time intervals of data is shown below,



```
C:\ProgramData\Anaconda3\lib\site-packages\statsmodels\tsa\statespace\mlemodel.py in fit(self, start_params, transformed, cov_
ype, cov_kwds, method, maxiter, full_output, disp, callback, return_params, optim_score, optim_complex_step, optim_hessian, fl
gs, **kwargs)
    488         else:
    489             res = self.smooth(mlefit.params, transformed=False,
--> 490                               cov_type=cov_type, cov_kwds=cov_kwds)
    491 
    492             res.mlefit = mlefit

C:\ProgramData\Anaconda3\lib\site-packages\statsmodels\tsa\statespace\mlemodel.py in smooth(self, params, transformed, complex
step, cov_type, cov_kwds, return_ssm, results_class, results_wrapper_class, **kwargs)
    604 
    605         # Get the state space output
--> 606         result = self.ssm.smooth(complex_step=complex_step, **kwargs)
    607 
    608         # Wrap in a results object

C:\ProgramData\Anaconda3\lib\site-packages\statsmodels\tsa\statespace\kalman_smoother.py in smooth(self, smoother_output, smoo
h_method, results, run_filter, prefix, complex_step, **kwargs)
    381 
    382         # Run the filter
--> 383         kfilter = self._filter(**kwargs)
    384 
    385         # Create the results object

C:\ProgramData\Anaconda3\lib\site-packages\statsmodels\tsa\statespace\kalman_filter.py in _filter(self, filter_method, inversi
n_method, stability_method, filter_timing, tolerance, loglikelihood_burn, complex_step)
    791             self._initialize_filter(
    792                 filter_method, inversion_method, stability_method,
--> 793                 conserve_memory, filter_timing, tolerance, loglikelihood_burn
    794             )
    795         )

C:\ProgramData\Anaconda3\lib\site-packages\statsmodels\tsa\statespace\kalman_filter.py in _initialize_filter(self, filter_meth
d, inversion_method, stability_method, conserve_memory, tolerance, filter_timing, loglikelihood_burn)
    390                 self._statespaces[prefix], filter_method, inversion_method,
    391                 stability_method, conserve_memory, filter_timing, tolerance,
--> 392                 loglikelihood_burn
    393             )
    394         # Otherwise, update the filter parameters

statsmodels\tsa\statespace\_kalman_filter.pyx in statsmodels.tsa.statespace._kalman_filter.dKalmanFilter.__init__()
```

Figure 5.5 SARIMA 5-minute Error

- When SARIMA on 5 minutes intervals data runs on the higher configuration machine then it provides following results,

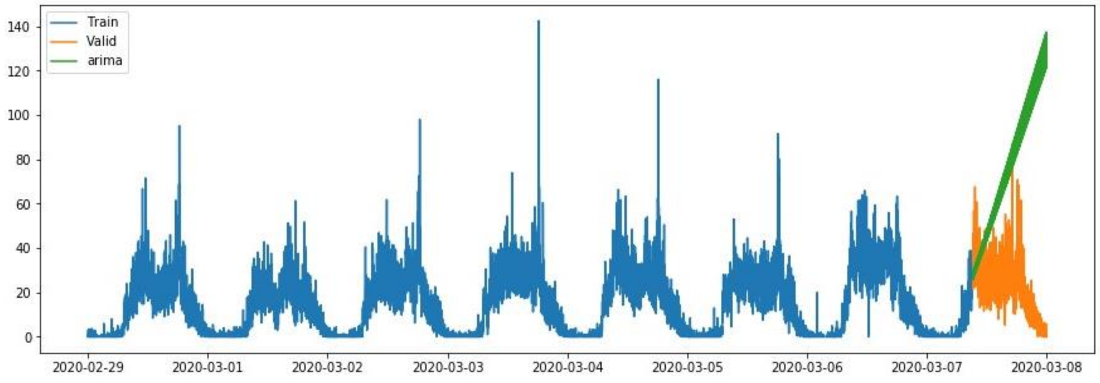


Figure 5.6 SARIMA 5-minutes Result

- For 15-minute intervals of time there will be 4 intervals in each hour so, for 24 hours S will be 24*4 = 96.

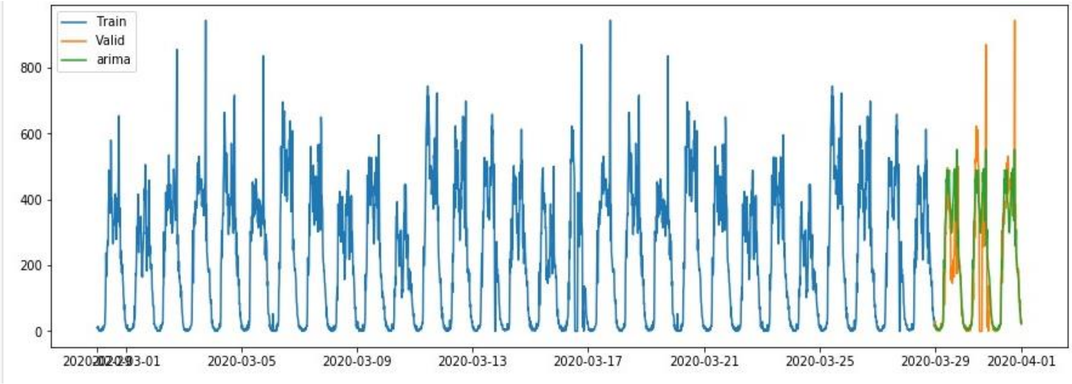


Figure 5.7 SARIMA 15-minute Result

- For 30-minute intervals of time there will be 2 intervals in each hour so, for 24 hours S will be 24*2 = 48.
- The result generated for 30 minutes time intervals of data is shown below,

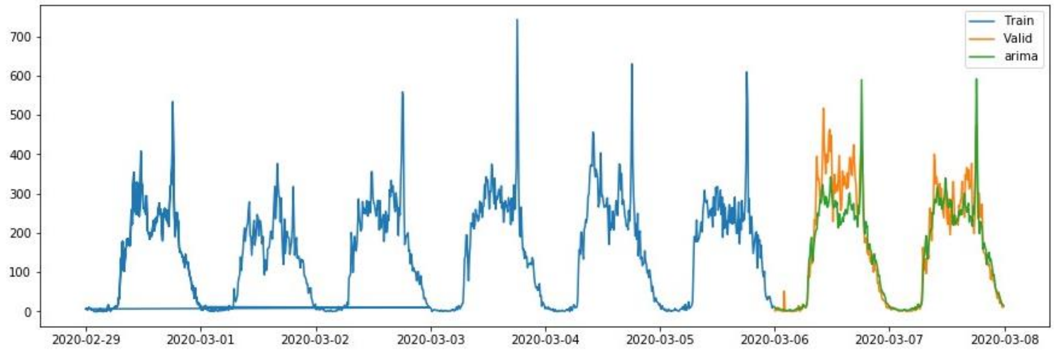


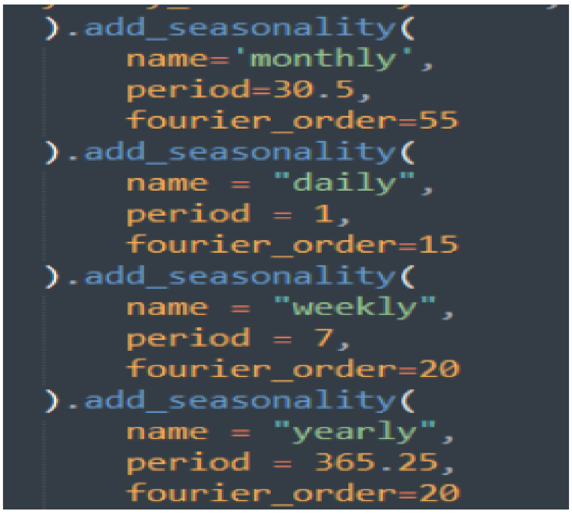
Figure 5.8 SARIMA 30-minute Result

- On the basis of results and some other values it is shown that SARIMA doesn't work for short term time intervals i.e. for 1 minutes.
- Due to the complexity and more time consumption and resource consumption we researched for other time series forecasting models and we found FB Prophet which works well compare to above models.

5. Discussion on the results of FbProphet

- Resource consumption and time consumption for short time intervals i.e. 1 minute of data lead us to search for other robust algorithms and by research it is found that FbProphet developed by the researchers of Facebook is the perfect choice.
- FbProphet accommodates seasonality with multiple periods.
- FbProphet is resilient to missing values i.e. the performance won't affected by the missing values.

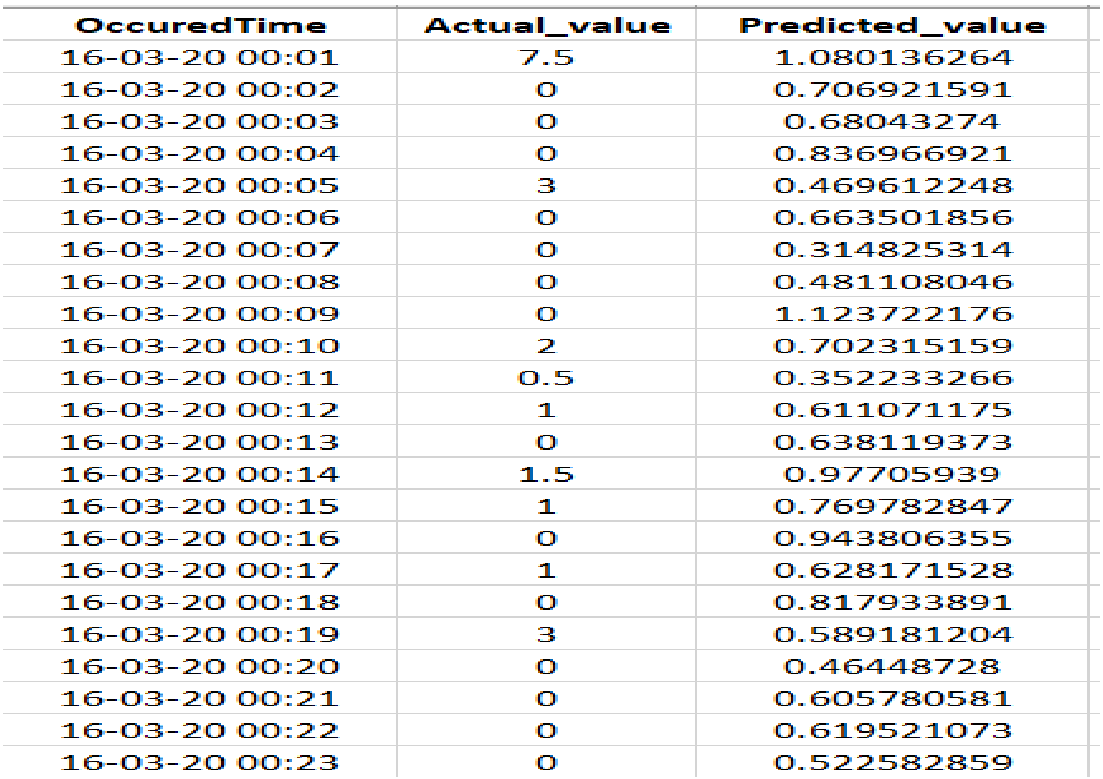
- FbProphet is the best way to handle outliers as it won’t consider them at the training of model.
- FbProphet is faster than above all the algorithms in model fitting time.
- FbProphet has intuitive hyper parameters which are easy to tune.
- The parameters on which results of the FbProphet depends are enlisted as below,
  - Growth – this parameter takes two values linear or logistic.
  - Holidays – this parameter defines the periods of time where the days have the same sort of effect each year i.e. Sunday if the road is taken by office guys then it will carry less traffic mostly on each of the Sunday.
  - Holidays_prior_scale – this parameter defines how much an effect of holiday should have on the prediction.
  - Changepoints – this parameter defines the points in the data where there are sudden and abrupt changes occurs in the trend.
  - Seasonalities – this parameter indicates how the seasonality component should be integrated with the predictions. It contains two values i.e. additive or multiplicative.
  - fourier_order – higher value of this parameter means that the data has higher frequency terms and so it will be able to fit more quickly-changing and complex seasonality patterns.
- In FbProphet one can define the seasonality as per the data which is explained as below,



```
).add_seasonality(
    name='monthly',
    period=30.5,
    fourier_order=55
).add_seasonality(
    name = "daily",
    period = 1,
    fourier_order=15
).add_seasonality(
    name = "weekly",
    period = 7,
    fourier_order=20
).add_seasonality(
    name = "yearly",
    period = 365.25,
    fourier_order=20
```

Figure 5.9 FBProphet Seasonality

- In FbProphet using add_seasonality one can easily define the seasonality period and fourier_order for each of the parameters i.e. for daily, monthly, weekly, yearly separately.
- The forecasting performed using FBProphet is as shown below,



OccuredTime	Actual_value	Predicted_value
16-03-20 00:01	7.5	1.080136264
16-03-20 00:02	0	0.706921591
16-03-20 00:03	0	0.68043274
16-03-20 00:04	0	0.836966921
16-03-20 00:05	3	0.469612248
16-03-20 00:06	0	0.663501856
16-03-20 00:07	0	0.314825314
16-03-20 00:08	0	0.481108046
16-03-20 00:09	0	1.123722176
16-03-20 00:10	2	0.702315159
16-03-20 00:11	0.5	0.352233266
16-03-20 00:12	1	0.611071175
16-03-20 00:13	0	0.638119373
16-03-20 00:14	1.5	0.97705939
16-03-20 00:15	1	0.769782847
16-03-20 00:16	0	0.943806355
16-03-20 00:17	1	0.628171528
16-03-20 00:18	0	0.817933891
16-03-20 00:19	3	0.589181204
16-03-20 00:20	0	0.46448728
16-03-20 00:21	0	0.605780581
16-03-20 00:22	0	0.619521073
16-03-20 00:23	0	0.522582859

Figure 5.10 FBProphet Result

- Result comparison on the basis of R^2 i.e. co. efficient of determination is as shown in the below table,

ALGORITHM	Type of Traffic Data (minute wise)	R^2 (Accuracy Parameter)
Decision Tree Regressor	1 - minute	0.72
XGBoost	1 - minute	0.78
ARIMA	1 - minute	NaN
SARIMA	1 - minute	-
SARIMA	5 - minutes	-0.25
SARIMA	15 - minutes	0.70
SARIMA	30 - minutes	0.86
Facebook Prophet	1 - minutes	0.65

Table 5-1 Comparison of the results of Algorithms on various data

- On the basis of above table, it is shown that Decision Tree Regressor, XGBoost algorithms doesn’t provide appropriate and accurate results.
- While, time series models such as ARIMA is also failed to provide the result due to the seasonality of the data and complex computations.
- SARIMA provides appropriate forecasting results for the time series data of 15 minutes and 30 minutes time interval.
- Due to the memory error thrown by SARIMA for short term time series data having time intervals of 1 minute and 5-minute, Facebook Prophet is used for the 1-minute time intervals data for forecasting purpose.
- As per the above results, SARIMA provides 70 % and 86 % accuracy in the forecasting of 15-minutes and 30 minutes time intervals data respectively.
- FBProphet provides 65 % accuracy on 1-minute time intervals data right now.
- We are trying to increase the performance of time series forecasting by performing various parameters tuning.

CHAPTER 6 Conclusion

The research project has been under the guidance of company professionals considering the research criteria. The project is an honest attempt in providing a proper solution to the traffic operators to solve the major issue of traffic jam in the smart cities of India. The project is an innovative approach to solve the real-life issue using machine learning and statistical approaches by predicting the traffic flow. The project will help to automate the traffic system in future. In a way it will help government to solve one of the major issues of metro cities.

Problems Encountered and Their Solutions:

- During the implementation of research-based project ‘Predictive Analytics For Traffic’ various problems were encountered which are enlisted as below with the solutions,

    1. *Problem:* While implementing data storage in MySQL the main problem was encountered during the concurrent storage of data of all the 4 cameras using the approach of multiprocessing / multithreading in the same table of the database.

       Solution: This challenge was handled by providing separate connections to the database for each camera for each of the threads or processes.

    2. *Problem:* While using the concept of multiple threading most of the data was missing as multiple threads have shared memory to store the data among all the threads due to which most of the data was losing during the storage of the data in the database.

       Solution: This challenge was handled partially by using the multiprocessing concept, because multiprocessing allocates separate memory to store the data for every process. The problem is solved partially because, after implementing multiprocessing the rate of missing data became less then multithreading.

    3. *Problem:* The data coming from the camera comes at each second but there is an issue – if the data comes at 12:00:12.635489 the given time then the next data will hit at 12:00:13 but it may come on 635489 millisecond or before or may be after that. That is the data coming from the camera considers millisecond of time too. As per the current scenario some of the data is missed due to this issue.

Solution: The issue will be solved by implementing by performing storage pool concept. Storage pool concept will implement the file that will store the data in the file which will be stored in the database later on by another separate process.

Summary of Research Project:

- During this research project 'Predictive Analytics For Traffic', I got a chance to learn various new tools and technologies including,
  - ➢ MySQL
  - ➢ Python (Pandas, Matplotlib, NumPy, Datetime, requests, mysql.connector)
  - ➢ Various statistical approaches
  - ➢ Time Series Forecasting methods
  - ➢ Core Machine Learning models
- I had a great experience working on research project.
- I also got to know coding according to industry standards and knowledge regarding various other technologies the organization is working on.
- While working on the project I got to learn about how to optimize the code and write it effectively.
- I also got to know ins and outs of how a project is actually being developed at industry level.
- I have got a great exposure working on this project and working on the project has enhanced my skills and experience about working on industrial project.

CHAPTER 7 Future Extensions

- New enhanced research and development in the field of transportation is a boon to solve various problems.
- Various innovations in this field will solve many of the traffic related issues. We would like to extend the area of traffic analytics from one particular junction to a number of junctions.
- In future ‘Predictive Analytics for Traffic’ will implemented on various advanced Time Series Forecasting models and Neural Networks to improve the results of forecasting.
- Future enhancement in this field can provide the most congested junction if there are more than one junction are connected in the network.
- It will also provide most congested Arm i.e. the particular way which consists more traffic than others.
- As of now google provides the shortest path from the existing paths on the basis of GPS. In future this technology will help to divert the traffic and it will solve one of the major issues.
- In future on the basis of traffic flow, the traffic will be diverted across the city to solve the issue of traffic jam.

Bibliography

1. https://ashwin-ks.github.io/2018-05-02-Time-Series-Modelling-using-Python/
2. https://towardsdatascience.com/predicting-the-future-with-facebook-s-prophet-bdfe11af10ff
3. https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/
4. https://towardsdatascience.com/https-medium-com-lorrli-classification-and-regression-analysis-with-decision-trees-c43cdbc58054
5. http://afterinc.com/brief-history-predictive-analytics-part-1/
6. https://stackoverflow.com/questions/42822902/can-someone-help-me-in-installing-python-package-prophet-on-windows-10
7. https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0218626
8. https://link.springer.com/article/10.1007/s12544-015-0170-8
9. https://towardsdatascience.com/time-series-in-python-exponential-smoothing-and-arima-processes-2c67f2a52788
10. https://ashwin-ks.github.io/2018-05-02-Time-Series-Modelling-using-Python/

16CE109

ORIGINALITY REPORT

25%	19%	11%	19%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Charotar University of Science And Technology Student Paper	6%
2	towardsdatascience.com Internet Source	4%
3	www.edureka.co Internet Source	2%
4	afterinc.com Internet Source	1%
5	medium.com Internet Source	1%
6	ivyproschool.com Internet Source	1%
7	ashwin-ks.github.io Internet Source	1%
8	link.springer.com Internet Source	1%
9	Submitted to Indian School of Business	

	Student Paper	<1%
10	Submitted to Symbiosis International University Student Paper	<1%
11	www.geeksforgeeks.org Internet Source	<1%
12	Submitted to University of Computer Studies Student Paper	<1%
13	Submitted to Christ University Student Paper	<1%
14	Submitted to Xavier Institute of Management and Entrepreneurship, Bangalore Student Paper	<1%
15	www.hcltech.com Internet Source	<1%
16	Submitted to National Institute of Construction Management and Research Student Paper	<1%
17	Submitted to Nanyang Polytechnic Student Paper	<1%
18	Submitted to October University for Modern Sciences and Arts (MSA) Student Paper	<1%
19	"Intelligent Information and Database Systems",	<1%

	Springer Science and Business Media LLC, 2020 Publication	
20	Tharindu D. Gamage, Jayathu G. Samarawickrama, A. A. Pasqual. "GPU based non-overlapping multi-camera vehicle tracking", 7th International Conference on Information and Automation for Sustainability, 2014 Publication	<1%
21	Submitted to Liverpool John Moores University Student Paper	<1%
22	Prifiyia Nunes, Dippal Israni, Karthick D., Arpita Shah. "A novel approach for mitigating atmospheric turbulence using weighted average Sobolev gradient and Laplacian", International Journal of Computational Vision and Robotics, 2019 Publication	<1%
23	Submitted to University of Sydney Student Paper	<1%
24	www.charusat.ac.in Internet Source	<1%
25	Submitted to National College of Ireland Student Paper	<1%
26	"Advances in Network Security and Applications", Springer Science and Business	<1%

	Media LLC, 2011 Publication	
27	G.C. Montanari. "A parametric approach to the prediction of the time-behavior of harmonic-quantities in electrical networks", IAS 95 Conference Record of the 1995 IEEE Industry Applications Conference Thirtieth IAS Annual Meeting IAS-95, 1995 Publication	<1%
28	"Foreseeing Employee Attritions using Diverse Data Mining Strategies", International Journal of Recent Technology and Engineering, 2019 Publication	<1%
29	Submitted to Nottingham Trent University Student Paper	<1%
30	Submitted to University of Strathclyde Student Paper	<1%
31	Submitted to University of Johannsburg Student Paper	<1%
32	Submitted to Intercom Programming & Manufacturing Company Limited (IPMC) Student Paper	<1%
33	Submitted to Middlesex University Student Paper	<1%
34	Submitted to iGroup Student Paper	<1%

35	Submitted to Higher Education Commission Pakistan Student Paper	<1%
36	Chieh-Chang Li, Shuo-Yan Chou, Shih-Wei Lin. "An agent-based platform for drivers and car parks negotiation", IEEE International Conference on Networking, Sensing and Control, 2004, 2004 Publication	<1%
37	Submitted to University of Portsmouth Student Paper	<1%
38	Submitted to University College London Student Paper	<1%
39	www.sciencepub.net Internet Source	<1%
40	manualzz.com Internet Source	<1%
41	Submitted to University of Brighton Student Paper	<1%
42	Submitted to Birkbeck College Student Paper	<1%
43	www.variousetc.com Internet Source	<1%
44	Submitted to Deakin University Student Paper	<1%

45	www.mdpi.com Internet Source	<1%
46	Submitted to University of Warwick Student Paper	<1%
47	Gao, C.. "Price forecast in the competitive electricity market by support vector machine", Physica A: Statistical Mechanics and its Applications, 20070801 Publication	<1%
48	eagletechsolutions.co.uk Internet Source	<1%
49	Submitted to National Research University Higher School of Economics Student Paper	<1%
50	Submitted to University of Newcastle upon Tyne Student Paper	<1%
51	www.jiwaji.edu Internet Source	<1%
52	Submitted to Higher Ed Holdings Student Paper	<1%
53	www.vibgyorjournal.com Internet Source	<1%
54	Submitted to Associatie K.U.Leuven Student Paper	<1%

55	www.dtic.mil Internet Source	<1 %
56	pt.scribd.com Internet Source	<1 %
57	cte.rockhurst.edu Internet Source	<1 %
58	www.researchoptimus.com Internet Source	<1 %
59	Submitted to University of Wales Institute, Cardiff Student Paper	<1 %
60	www.tantiagroup.com Internet Source	<1 %
61	Submitted to Leeds Metropolitan University Student Paper	<1 %
62	"Proceedings of the International Conference on ISMAC in Computational Vision and Bio-Engineering 2018 (ISMAC-CVB)", Springer Science and Business Media LLC, 2019 Publication	<1 %
63	"Information and Communication Technology for Intelligent Systems (ICTIS 2017) - Volume 1", Springer Science and Business Media LLC, 2018 Publication	<1 %

64	"Advanced Informatics for Computing Research", Springer Science and Business Media LLC, 2019 Publication	<1%
65	bans.tf2.gaming-servers.net Internet Source	<1%
66	Submitted to Aston University Student Paper	<1%
67	www.listendata.com Internet Source	<1%
68	ncap.res.in Internet Source	<1%
69	M. Inada, T. Terano. "Qc Chart Mining: Extracting Systematic Error Patterns from Quality Control Charts", 2005 IEEE International Conference on Systems, Man and Cybernetics, 2005 Publication	<1%
70	Submitted to Campus Hochschule fur angewandte Wissenschaften Augsburg Student Paper	<1%
71	Submitted to University of Hong Kong Student Paper	<1%
72	tobias-lib.uni-tuebingen.de Internet Source	<1%

73	Submitted to University of Bristol Student Paper	<1%
74	hdl.handle.net Internet Source	<1%
75	Submitted to Universiti Sains Malaysia Student Paper	<1%
76	Submitted to Maulana Azad National Institute of Technology Bhopal Student Paper	<1%
77	Younes Oulad Sayad, Hajar Mousannif, Hassan Al Moatassime. "Predictive modeling of wildfires: A new dataset and machine learning approach", Fire Safety Journal, 2019 Publication	<1%
78	Mohammad Amin Kuhail, Manohar Boorlu, Neeraj Padarthi, Collin Rottinghaus. "Parking Availability Forecasting Model", 2019 IEEE International Smart Cities Conference (ISC2), 2019 Publication	<1%
79	Submitted to University Tun Hussein Onn Malaysia Student Paper	<1%
80	Submitted to University of Lancaster Student Paper	<1%

81	Submitted to Rutgers University, New Brunswick Student Paper	<1%
82	Jun Sun, Hanping Mao, Yiqing Yang. "Chapter 30 THE RESEARCH ON THE JUDGMENT OF PADDY RICE’S NITROGEN DEFICIENCY BASED ON IMAGE", Springer Science and Business Media LLC, 2009 Publication	<1%
83	quizsolution.in Internet Source	<1%
84	Submitted to Queen Mary and Westfield College Student Paper	<1%
85	Submitted to Napier University Student Paper	<1%
86	Submitted to University of Sunderland Student Paper	<1%
87	Submitted to University of Wales Swansea Student Paper	<1%
88	Submitted to University of Moratuwa Student Paper	<1%
89	www.ijstr.org Internet Source	<1%
90	Submitted to Kwame Nkrumah University of Science and Technology Student Paper	<1%

91	Chase, . "ARIMA Models", Demand-Driven Forecasting A Structured Approach to Forecasting, 2013. Publication	<1%
92	Submitted to Federal University of Technology Student Paper	<1%
93	Submitted to Taylor’s Education Group Student Paper	<1%
94	Sebastian Raschka, Anne M. Scott, Mar Huertas, Weiming Li, Leslie A. Kuhn. "Chapter 16 Automated Inference of Chemical Discriminants of Biological Activity", Springer Science and Business Media LLC, 2018 Publication	<1%
95	Submitted to Sheffield Hallam University Student Paper	<1%

Exclude quotes	On	Exclude matches	< 5 words
Exclude bibliography	On		