# An Industry Oriented Major Project Report

On

# TRAFFIC PREDICTION FOR INTELLIGENT TRANSPORT SYSTEM USING MACHINE LEARNING

Submitted to the

## JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

In partial fulfillment of the requirement for the award of the degree of

## BACHELOR OF TECHNOLOGY

In

## COMPUTER SCIENCE AND ENGINEERING

By

## SAI CHARAN BALUSU (17WJ1A05T2)

**Under the Esteemed Guidance Of m Mrs. K. Akshitha**

**Assistant Professor, CSE Dept.**



**GURU NANAK INSTITUTIONS TECHNICAL CAMPUS (AUTONOMOUS)**

**School of Engineering and Technology**

**Ibrahimpatnam, R.R District- 501506**

**2020-2021**



**Department of Computer Science and Engineering**

**C ERTIFICATE**

This is to certify that this project report entitled **“TRAFFIC PREDICTION FOR INTELLIGENT TRANSPORT SYSTEM USING MACHINE LEARNING”** by **SAI CHARAN BALUSU (17WJ1A05T2)** submitted in partial fulfilment of the requirements for the degree of **Bachelor of Technology in Computer Science and Engineering** of the **Jawaharlal Nehru Technological University Hyderabad** during the academic year 2020-2021, is a bonafide record of work carried out under our guidance and supervision**.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **INTERNAL GUIDE** |  | **PROJECT CO-ORDINATOR** |  | **HOD CSE** |
| **Mrs. K. Akshitha** |  | **Mrs. V. Swathi** |  | **Mr. V. Devasekhar** |

## EXTERNAL EXAMINER

**Text, letter

Description automatically generated**

**ACKNOWLEDGEMENT**

I especially thank our internal guide **Mrs. K. Akshitha,** Assistant Professor for her constant guidance in every stage of the project. I would also like to thank all our lecturers for helping us in every possible way whenever the need arose.

I would like to say our sincere thanks to **Mrs. V. Swathi**, Assistant Professor, Department of CSE, Major Project Coordinator, for providing seamless support and the right suggestions given in the development of the project.

I have been truly blessed to have a wonderful adviser **Mr. V. Devasekhar,** Associate Professor, Head of the Department of CSE, GNITC for guiding us to explore the ramification of our work and I express our sincere gratitude towards him for leading us through the completion of the project.

I wish to express our sincere thanks to **Dr. Rishi Sayal,** Associate Director, GNITC for providing us with a conducive environment for carrying through our academic schedules and project with ease.

Finally, I would like to thank our family members for their moral support and encouragement to achieve our goals.

## SAI CHARAN BALUSU (17WJ1A05T2)

## 

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | **LIST OF FIGURES**  **LIST OF SYMBOLS**  **LIST OF ABBREAVIATIONS**  **ABSTRACT** | iii  iv  viii  ix |
| 1. | **CHAPTER 1: INTRODUCTION**  1.1 GENERAL  1.2 OBJECTIVE  1.3 LITERATURE SURVEY  1.4 EXISTING SYSTEM  1.5 PROPOSED SYSTEM | 1  1  2  3  8  8 |
| 2. | **CHAPTER 2: PROJECT DESCRIPTION**  2.1 GENERAL  2.2 METHODOLOGIES  2.2.1 MODULES NAME  2.3 TECHNIQUE USED OR ALGORITHM USED  2.3.1 EXISTING TECHNIQUE  2.3.2 PROPOSED TECHNIQUE | 9  9  9  9  12  12  13 |
| 3. | **CHAPTER3:REQUIREMENTS ENGINEERING**  3.1 General  3.2 Hardware REQUIREMENTS  3.3 Software REQUIREMENTS  3.4 Functional Specification  3.5 Non-Functional SpecificatioN | 14  14  14  15  15  15 |
| 4. | **CHAPTER 4: DESIGN ENGINEERING**  4.1 general  4.2 Uml  4.2.1 USE CASE diagram  4.2.2 CLASS DIAGRAM  4.2.3 OBJECT DIAGRAM  4.2.4 COMPONENT DIAGRAM  4.2.5 DEPLOYMENT DIAGRAM  4.2.6 SEQUENCED DIAGRAM  4.2.7 COLLABARATION DIAGRAM  4.2.8 STATE DIAGRAM  4.2.9 ACTIVITY DIAGRAM  4.3 SYSTEM ARCHITECTURE | 16  16  16  16  17  18  19  20  21  22  23  24  25 |
| 5. | **CHAPTER 5: DEVELOPMENT TOOLS**  5.1 OVER VIEW OF PYTHON  5.2 HISTORY OF PYTHON  5.3 IMPORTANCE OF PYTHON  5.4 FEATURES OF PYTHON  5.5 LIBRARIES USED IN PYTHON | 26  26  26  26  27  28 |
| 6. | **CHAPTER 6: SNAPSHOTS**  6.1 SNAPSHOTS | 29  32 |
| 7. | **CHAPTER 7: IMPLEMENTATION**  7.1 GENERAL COADING | 33  35 |
| 8. | **CHAPTER 8: SOFTWARE TESTING**  8.1 GENERAL  8.2 DEVELOPING METHODOLOGIES  8.3 TYPES OF TESTS | 36  36  37  37 |
| 9. | **CHAPTER 9: CONCLUSION AND FUTURE ENHANCEMENT**  9.1 CONCLUSION  9.2 FUTURE ENHANCEMENT | 38  43  44 |
|  | **REFERENCES** | 45 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **NAME OF THE FIGURE** | **PAGE NO.** |
| 4.2.1 | Use case Diagram | 16 |
| 4.2.2 | Class Diagram | 17 |
| 4.2.3 | Object Diagram | 18 |
| 4.2.4 | Component Diagram | 19 |
| 4.2.5 | Deployment Diagram | 20 |
| 4.2.6 | Sequence Diagram | 21 |
| 4.2.7 | Collaboration Diagram | 22 |
| 4.2.8 | State Diagram | 23 |
| 4.2.9 | Activity Diagram | 24 |
| 4.3 | System Architecture | 25 |
| 5.5 | NumPy, Pandas, Matplotlib, Scikit-learn | 28 |
| 6.1 | Importing libraries and importing training dataset | 29 |
| 6.2 | Examining training data | 29 |
| 6.3 | Importing testing dataset | 30 |
| 6.4 | Plotting train attributes | 30 |
| 6.5 | Visualizing data | 31 |
| 6.6 | Visualizing traffic | 31 |
| 6.7 | Making confusion matrix and checking f1-score | 32 |

**LIST OF SYMBOLS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.NO** | **NAME** | **NOTATION** | | **DESCRIPTION** | |
| 1. | Class | *Class Name*  *-attribute*  *-attribute*  *+operation*  *+operation*  *+operation*  *+ public*  *-private*  *# protected* | | Represents a collection of similar entities grouped together. | |
| 2. | Association | name  Class B  Class A  Class A  Class B | | Associations represents static relationships between classes. Roles represents the way the two classes see each other. | |
| 3. | Actor | Class A  Class A  Class B  Class B | | It aggregates several classes into a single classes. | |
| 5. | Aggregation | Interaction between the system and external environment | |
| 5. | Relation  (uses) | | Uses | | Used for additional process communication. | |
| 6. | Relation  (extends) | | extends | | Extends relationship is used when one use case is similar to another use case but does a bit more. | |
| 7. | Communication | |  | | Communication between various use cases. | |
| 8. | State | | Class B | | State of the process. | |
| 9. | Initial State | |  | | Initial state of the object | |
| 10. | Final state | |  | | Final state of the object | |
| 11. | Control flow | |  | | Represents various control flow between the states. | |
| 12. | Decision box | |  | | Represents decision making process from a constraint | |

|  |  |  |  |
| --- | --- | --- | --- |
| 13. | Use case |  | Interact ion between the system and external environment. |
| 14. | Component |  | Represents physical modules which is a collection of components. |
| 15. | Node |  | Represents physical modules which are a collection of components. |
| 16. | Data Process/State |  | A circle in DFD represents a state or process which has been triggered due to some event or action. |
| 17. | External entity |  | Represents external entities such as keyboard,sensors,etc. |
| 18. | Transition |  | Represents communication that occurs between processes. |
| 19. | Object Lifeline |  | Represents the vertical dimensions that the object communications. |
| 20. | Message | Message | Represents the message exchanged. |

**LIST OF ABBREVATION**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **ABBREVATION** | **EXPANSION** |
| 1**.** | DB | DataBase |
| 2. | JVM | Java Virtual Machine |
| 3. | JSP | Java Server Page |
| 4. | CB | Collective Behavior |
| **5.** | RSSS | Ramp secret sharing scheme |
| 6. | JRE | Java Runtime Environment |

**ABSTRACT**

This paper aims to develop a tool for predicting accurate and timely traffic flow Information. Traffic Environment involves everything that can affect the traffic flowing on the road, whether it’s traffic signals, accidents, rallies, even repairing of roads that can cause a jam. If we have prior information which is very near approximate about all the above and many more daily life situations which can affect traffic then, a driver or rider can make an informed decision. Also, it helps in the future of autonomous vehicles. In the current decades, traffic data have been generating exponentially, and we have moved towards the big data concepts for transportation. Available prediction methods for traffic flow use some traffic prediction models and are still unsatisfactory to handle real-world applications. This fact inspired us to work on the traffic flow forecast problem build on the traffic data and models. It is cumbersome to forecast the traffic flow accurately because the data available for the transportation system is insanely huge. In this work, we planned to use machine learning, genetic, soft computing, and deep learning algorithms to analyse the big-data for the transportation system with much-reduced complexity. Also, Image Processing algorithms are involved in traffic sign recognition, which eventually helps for the right training of autonomous vehicles.

**CHAPTER 1**

**INTRODUCTION**

* 1. **GENERAL**

Various Business sectors and government agencies and individual travellers require precise and appropriately traffic flow information. It helps the riders and drivers to make better travel judgement to alleviate traffic congestion, improve traffic operation efficiency, and reduce carbon emissions. The development and deployment of Intelligent Transportation System (ITSs) provide better accuracy for Traffic flow prediction. It is deal with as a crucial element for the success of advanced traffic management systems, advanced public transportation systems, and traveller information systems. [1]. The dependency of traffic flow is dependent on real-time traffic and historical data collected from various sensor sources, including inductive loops, radars, cameras, mobile Global Positioning System, crowd sourcing, social media. Traffic data is exploding due to the vast use of traditional sensors and new technologies, and we have entered the era of a large volume of data transportation. Transportation control and management are now becoming more data-driven. [2], [3].However, there are already lots of traffic flow prediction systems and models; most of them use shallow traffic models and are still somewhat failing due to the enormous dataset dimension.

Recently, deep learning concepts attract many persons involving academicians and industrialist due to their ability to deal with classification problems, understanding of natural language, dimensionality reduction, detection of objects, motion modelling. DL uses multi-layer concepts of neural networks to mining the inherent properties in data from the lowest level to the highest level [4]. They can identify massive volumes of structure in the data, which eventually helps us to visualize and make meaningful inferences from the data. Most of the ITS departments and researches in this area are also concerned about developing an autonomous vehicle, which can make transportation systems much economical and reduce the risk of lives. Also, saving time is the integrative benefit of this idea. In current decades the lots of attention have made towards the safe automatic driving. It is necessary that the information will be provided in time through driver assistance system (DAS), autonomous vehicles (AV)and Traffic Sign Recognition (TSR) predicting the traffic flow information. But these algorithms are not accurate since Traffic Flow involves data having a vast dimension, so it is not very easy to predict accurate traffic flow information with less complexity. We intend to use Genetic, Deep Learning , Image Processing,

Machine Learning and also Soft Computing algorithms for prediction of traffic flow since a lot of journals and research paper suggests that they work well when it comes to Big-Data

**1.2 OBJECTIVE**

The objective is Each purpose on the mythical monster curve represents a sensitivity specificity try resembling a selected call threshold. The area underneath the mythical monster curve (AUC) could be a live of however well a parameter will distinguish between 2 teams .

**1.3 LITERATURE SURVEY**

**Title:** Index point detection and semantic indexing of videos - a comparative review

**Author:** Mehul Mahrishi and Sudha Morwal.

**Year:** 2020

**Description:**

Primarily used for fun and entertainment, videos are now a motivation behind social, commercial, and business activities. It is presumed that by 2025, about 75% of all Internet traffic will be of videos. In education, videos are a source of learning. Study Webs of Active Learning for Young Aspiring Minds (SWAYAM), National Programme on Technology Enhanced Learning (NPTEL), Massive Open Online Courses (MOOCs), Coursera, and many other similar platforms provide not only courseware but also beyond the curriculum contents apart from the conventional syllabi. Even at the junior level, Byju’s and similar educational portals are witnessing an explosive growth in video contents. Despite that we are now able to extract semantic features from images, video sequences and besides being ubiquitous in nature, video lectures have a limitation of smooth navigation between topics. Through this paper, we want to throw light on existing automated video indexing approaches and their prerequisites that are recently proposed. We tried to analyze them based on some existing measures..

**Title:** Deep learning in mobile and wireless networking: A survey

**Author:** C. Zhang, P. Patras, and H. Had.

**Year:** 2019

**Description:**

The rapid uptake of mobile devices and the rising popularity of mobile applications and services pose unprecedented demands on mobile and wireless networking infrastructure. Upcoming 5G systems are evolving to support exploding mobile traffic volumes, real-time extraction of fine-grained analytics, and agile management of network resources, so as to maximize user experience. Fulfilling these tasks is challenging, as mobile environments are increasingly complex, heterogeneous, and evolving. One potential solution is to resort to advanced machine learning techniques, in order to help manage the rise in data volumes and algorithm-driven applications. The recent success of deep learning underpins new and powerful tools that tackle problems in this space. In this paper, we bridge the gap between deep learning and mobile and wireless networking research, by presenting a comprehensive survey of the crossovers between the two areas. We first briefly introduce essential background and state-of-the-art in deep learning techniques with potential applications to networking. We then discuss several techniques and platforms that facilitate the efficient deployment of deep learning onto mobile systems. Subsequently, we provide an encyclopedic review of mobile and wireless networking research based on deep learning, which we categorize by different domains. Drawing from our experience, we discuss how to tailor deep learning to mobile environments. We complete this survey by pinpointing current challenges and open future directions for research.

**Title:** Decision tree methods: applications for classification and prediction.

**Author:** Yan-Yan Song and LU Ying.

**Year:** 2015

**Description:**

Decision tree methodology is a commonly used data mining method for establishing classification systems based on multiple covariates or for developing prediction algorithms for a target variable. This method classifies a population into branch-like segments that construct an inverted tree with a root node, internal nodes, and leaf nodes. The algorithm is non-parametric and can efficiently deal with large, complicated datasets without imposing a complicated parametric structure. When the sample size is large enough, study data can be divided into training and validation datasets. Using the training dataset to build a decision tree model and a validation dataset to decide on the appropriate tree size needed to achieve the optimal final model. This paper introduces frequently used algorithms used to develop decision trees (including CART, C4.5, CHAID, and QUEST) and describes the SPSS and SAS programs that can be used to visualize tree structure.

**Title:** Parallel control and management for intelligent transportation systems: Concepts, architectures, and applications

**Author:** Fei-Yue Wang et al

**Year:** 2010

**Description:**

Parallel control and management have been proposed as a new mechanism for conducting operations of complex systems, especially those that involved complexity issues of both engineering and social dimensions, such as transportation systems. This paper presents an overview of the background, concepts, basic methods, major issues, and current applications of Parallel transportation Management Systems (PtMS). In essence, parallel control and management is a data-driven approach for modeling, analysis, and decision-making that considers both the engineering and social complexity in its processes. The developments and applications described here clearly indicate that PtMS is effective for use in networked complex traffic systems and is closely related to emerging technologies in cloud computing, social computing, and cyberphysical-social systems. A description of PtMS system architectures, processes, and components, including OTSt, Dyna CAS, aDAPTS, iTOP, and TransWorld is presented and discussed. Finally, the experiments and examples of real-world applications are illustrated and analyzed.

**Title:** A decentralized approach for anticipatory vehicle routing using delegate multiagent systems

**Author:** Rutger Claes, Tom Holvoet, and Danny Weyns

**Year:** 2011

**Description:**

Advanced vehicle guidance systems use real-time traffic information to route traffic and to avoid congestion. Unfortunately, these systems can only react upon the presence of traffic jams and not to prevent the creation of unnecessary congestion. Anticipatory vehicle routing is promising in that respect, because this approach allows directing vehicle routing by accounting for traffic forecast information. This paper presents a decentralized approach for anticipatory vehicle routing that is particularly useful in large-scale dynamic environments. The approach is based on delegate multiagent systems, i.e., an environment-centric coordination mechanism that is, in part, inspired by ant behavior. Antlike agents explore the environment on behalf of vehicles and detect a congestion forecast, allowing vehicles to reroute. The approach is explained in depth and is evaluated by comparison with three alternative routing strategies. The experiments are done in simulation of a real-world traffic environment. The experiments indicate a considerable performance gain compared with the most advanced strategy under test, i.e., a traffic-message-channel-based routing strategy

**1.4 EXISTING SYSTEM:**

Various Business sectors and government agencies and individual travelers require precise and appropriately traffic flow information. It helps the riders and drivers to make better travel judgment to alleviate traffic congestion, improve traffic operation efficiency, and reduce carbon emissions. The development and deployment of Intelligent Transportation System (ITSs) provide better accuracy for Traffic flow prediction. It is deal with as a crucial element for the success of advanced traffic management systems, advanced public transportation systems, and traveler information systems. The dependency of traffic flow is dependent on real-time traffic and historical data collected from various sensor sources, including inductive loops, radars, cameras, mobile Global Positioning System, crowd sourcing, social media.

**1.5 PROPOSED SYSTEM:**

Deep learning concepts attract many persons involving academicians and industrialist due to their ability to deal with classification problems, understanding of natural language, dimensionality reduction, detection of objects, motion modelling. DL uses multi-layer concepts of neural networks to mining the inherent properties in data from the lowest level to the highest level. They can identify massive volumes of structure in the data, which eventually helps us to visualize and make meaningful inferences from the data. Most of the ITS departments and researches in this area are also concerned about developing an autonomous vehicle, which can make transportation systems much economical and reduce the risk of lives. Also, saving time is the integrative benefit of this idea. In current decades the lots of attention have made towards the safe automatic driving. It is necessary that the information will be provided in time through driver assistance system (DAS), autonomous vehicles (AV) and Traffic Sign Recognition (TSR).

**CHAPTER 2**

**PROJECT DESCRIPTION**

**2.1 GENERAL:**

Deep learning is a part of machine learning algorithms, and it is a compelling tool to handle a large amount of data. DL provides a method to add intelligencies in the wireless network with complex radio data and large- scale topology. In DL, use concepts of a neural network, by using this feature, it is beneficial to find network dynamics (such as spectrum availability, congestion points, hotspots, traffic bottleneck. [8] The travel time is the essential aspect in ITS and the exact travel time forecasting also is very challenging to the development of ITS. Support Vector Machine (SVM) is one of the most effective classifiers among those which are sort of linear. It is advantageous to prevent overfitting of data. SVM is great for relatively small data sets with fewer outliers. Another algorithm (Random Forest, Deep Neural Network, etc.) require more data but always came up with very robust models

**2.2 METHODOLOGIES**

**2.2.1 MODULES NAME:**

1. Dataset

2. Training Set Composition

3. Importing Modules

4. Label encoder and One Hotencoder

5. Feature Scaling

6. Accuracy Metrics

**1 DATASET**

One such event, which impacts our daily lives, is the act of commuting. As typicalcity dwellers, most of us commute every day for work, and often we run against the strictschedule. As an example, you need to head to the office tomorrow to attend an importantearly morning meeting. So you must start early and also ensure that the duration ofcommute, or the travel time, to your office is within that safe limit that ensures that youreach in time, neither late, nor too early. What you need is a recommendation system thatcan suggest you the most favourable moment to leave from home so that you are assuredof reaching the office, just in time.

How to predict the traffic flow between two locations within a city?

Within a city, the traffic flow from point A to B depends on many dynamic factorsalong with the demographics. Even within each hour of the day, there can be variations intravel time based on busy and non-busy hour traffic. Also, the day of the week plays a role,as the weekends will witness faster travel times due to low traffic density. Apart from this,there are also some environmental conditions, such as weather conditions andtemperature that indirectly affect the travel time. So let’s consider these four data pointsthat influence the travel time.

Datapoints affecting travel time

1. Time of the day

The travel time is largely dependent on this as during busy hours the trafficdensity on the city roads is at its peak.

2. Day of the week

Weekdays always witness more traffic density due to the rush of officecommuters as compared to weekends.

3. Weather conditions

Harsh weather conditions can affect road and public infrastructure so this hasan adverse impact on travel time.

4. Temperature

Extreme temperatures tend to keep people indoors which means less trafficdensity and faster travel.

**2 Training Set Composition**

This training set will consist of the above four parameters along with theactual travel times, recorded from the past. We have collected the data for SanFrancisco between the two hospitals shown in the map.Here is how the sample training set looks like. This is real data as has been capturedusing the Mapquest API.

This data is collected from 1st January 2017 to 31st December 2017. The captured

datapoints are :

➢ Time of the day (Zone column) :A number code representing a 10 minute interval time-zone, splittingthe 24 hours of a day into 144 zones, ( For example, the 10 minute durationfrom 00:00 to 00:10 Hrs is coded as 1 and 00:10 to 00:20 Hrs is coded as 2,and so on)

➢ Day of the week (CodedDay column) :Week day in a coded number, 7 weekdays converted into to 7 numbersstarting from 1(Sunday) to 7(Saturday).

➢ Weather Conditions (CodedWeather column) :Weather in a coded number. Check out the codes representing weatherconditions that are used in this training set.

➢ Temperature (Temperature column) :Average temperature during the day, in Fahrenheit.

The training set also captures the actual time taken for travel in minutes (under theRealtime column) for each of the records that capture the four data points.So in brief, here is the list of all six parameters that constitute one data record ofthe training set.

1. The starting time from the source or point A (represented by Date column)

2. Ten minute interval time zone of the day (represented by Zone column)

3. Day of the week (represented by Day and CodedDay column, both meaningthe same)

4. Weather conditions on that day (represented by CodedWeather column)

5. The Temperature on that day (represented by Temperature column)

The actual traffic flow from A to reach destination, or point B, when someonestarted from the source at the time indicated in the Date column, (represented byRealtime column)

**3 IMPORTING MODULES**

To make use of the functions in a module, you’ll need to import the module with animport statement. An import statement is made up of the import keyword along with thename of the module. In a Python file, this will be declared at the top of the code, underany shebang lines or general comments.

Install and Load pandas Package

Pandas is a powerful data analysis package. It makes data exploration and

manipulation easy. It has several functions to read data from various sources.

**4 LABELENCODER AND ONEHOTENCODER**

In supervised learning, we usually deal with a variety of labels. These canbe in the form of numbers or words. If they are numbers, then the algorithm can usethem directly. However, a lot of times, labels need to be in human readable form. So,people usually label the training data with words. Label encoding refers to transformingthe word labels into numerical form so that the algorithms can understand how tooperate on them. Let's take a look at how to do this.

One hot encoding is a process by which categorical variables areconverted into a form that could be provided to ML algorithms to do a better job inprediction.

**5 FEATURE SCALING**

Feature scaling is a method used to standardize the range of independent variables orfeatures of data. In data processing, it is also known as data normalization and isgenerally performed during the data pre-processing step.

**6 ACCURACY METRICS**

**Confusion Matrix**

A confusion matrix is a table that is often used to describe the performance of aclassification model (or "classifier") on a set of test data for which the true values areknown. The confusion matrix itself is relatively simple to understand, but the relatedterminology can be confusing.

**F1 SCORE**

In statistical analysis of binary classification, the F1 score (also F-score or F-measure) is ameasure of a test's accuracy. The F1 score is the harmonic average of the precision and recall, where an F1 score reaches its best value at 1 (perfect precision and recall) andworst at 0.

**2.3 TECHNIQUE USED OR ALGORITHM USED**

**2.3.1 EXISTING TECHNIQUE:-**

**Support Vector Machine**

* A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane.
* In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples.
* In two dimensional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side..

**2.3.2 PROPOSED TECHNIQUE:**-

* **Decision TreeAlgorithm**

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.

**CHAPTER 3**

**REQUIREMENTS ENGINEERING**

**3.1 GENERAL**

The interpretation of the handwriting character by developing techniques and methods such as improvement of character classification techniques. The accurate and rapid classification for accurate information retrieval, sound classification, stock price forecasting.

**3.2 HARDWARE REQUIREMENTS**

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shouls what the system do and not how it should be implemented.

# Processor - Pentium –IV

* Speed - 1.1 GHz
* Ram - 256 MB
* Hard Disk - 20 GB
* Key Board - Standard Windows Keyboard
* Mouse - Two or Three Button Mouse
* Monitor - SVGA

**3.3 SOFTWARE REQUIREMENTS**

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team’s progress throughout the development activity.

**SOFTWARE REQUIREMENTS**

* Operating System - Windows XP
* Coding Language - Python

**3.4 FUNCTIONAL REQUIREMENTS**

A functional requirement defines a function of a software-system or its component. A function is described as a set of inputs, the behavior, Firstly, the system is the first that achieves the standard notion of semantic security for data confidentiality in attribute-based deduplication systems by resorting to the hybrid cloud architecture.

**3.5 NON-FUNCTIONAL REQUIREMENTS**

**EFFICIENCY**

Our multi-modal event tracking and evolution framework is suitable for multimedia documents from various social media platforms, which can not only effectively capture their multi-modal topics, but also obtain the evolutionary trends of social events and generate effective event summary details over time. Our proposed mmETM model can exploit the multi-modal property of social event, which can effectively model social media documents including long text with related images and learn the correlations between textual and visual modalities to separate the visual-representative topics and non-visual-representative topics.

**CHAPTER 4**

**DESIGN ENGINEERING**

**4.1 GENERAL**

Design Engineering deals with the various UML [Unified Modelling language] diagrams for the implementation of project. Design is a meaningful engineering representation of a thing that is to be built. Software design is a process through which the requirements are translated into representation of the software. Design is the place where quality is rendered in software engineering. Design is the means to accurately translate customer requirements into finished product.

**4.2 UML DIAGRAMS**

**4.2.1 USE CASE DIAGRAM**



Fig. 4.2.1 USE CASE DIAGRAM

**EXPLANATION:**

The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted. The above diagram consists of user as actor. Each will play a certain role to achieve the concept.

**4.2.2 CLASS DIAGRAM**

****

Fig. 4.2.2 CLASS DIAGRAM

**EXPLANATION**

In this class diagram represents how the classes with attributes and methods are linked together to perform the verification with security. From the above diagram shown the various classes involved in our project.

**4.2.3 OBJECT DIAGRAM**



Fig - 4.2.3 OBJECT DIAGRAM

**EXPLANATION:**

In the above digram tells about the flow of objects between the classes. It is a diagram that shows a complete or partial view of the structure of a modeled system. In this object diagram represents how the classes with attributes and methods are linked together to perform the verification with security.

**4.2.4 COMPONENT DIAGRAM**



Fig-4.2.4 COMPONENT DIAGRAM

**EXPLANATION**

In the Unified Modelling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems. User gives main query and it converted into sub queries and sends through data dissemination to data aggregators. Results are to be showed to user by data aggregators. All boxes are components and arrow indicates dependencies.

**4.2.5 DEPLOYMENT DIAGRAM**



Fig-4.2.5 DEPLOYMENT DIAGRAM

**EXPLANATION:**

Deployment Diagram is a type of diagram that specifies the physical hardware on which the software system will execute. It also determines how the software is deployed on the underlying hardware. It maps software pieces of a system to the device that are going to execute it.

**4.2.6 SEQUENCE DIAGRAM**



Fig-4.2.6 SEQUENCE DIAGRAM

**EXPLANATION:**

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

**4.2.7 COLLABORATION DIAGRAM**



Fig-4.2.7 COLLABORATION DIAGRAM

**EXPLANATION:**

A collaboration diagram, also called a communication diagram or interaction diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). The concept is more than a decade old although it has been refined as modeling paradigms have evolved.

**4.2.8 STATE DIAGRAM**



Fig-4.2.8 STATE DIAGRAM

**EXPLANATION:**

State diagram are a loosely defined diagram to show workflows of stepwise activities and actions, with support for choice, iteration and concurrency. State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction. Many forms of state diagrams exist, which differ slightly and have different semantics.

**4.2.9 ACTIVITY DIAGRAM**



Fig-4.2.9 ACTIVITY DIAGRAM

**EXPLANATION:**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

**4.3 SYSTEM ARCHITECTURE**

****

Fig-4.3 SYSTEM ARCHITECTURE

**EXPLANATION:**

The system architecture of E-learning with the help of machine learning consists of 3 levels. The input data or training dataset is sent into the training module and outputs certain data which is an optimized way of retrieving the processed data.

**CHAPTER 5**

**DEVELOPMENT TOOLS**

**5.1 Over view of Python**

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

## 5.2 History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

#### 5.3 Importance of Python

* **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
* **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
* **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

#### 5.4 Features of Python

* **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
* **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
* **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
* **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
* **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
* **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
* **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
* **Databases** − Python provides interfaces to all major commercial databases.
* **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
* **Scalable** − Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below −

* It supports functional and structured programming methods as well as OOP.
* It can be used as a scripting language or can be compiled to byte-code for building large applications.
* It provides very high-level dynamic data types and supports dynamic type checking.
* IT supports automatic garbage collection.
* It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

**5.5 Libraries used in python:**

* numpy - mainly useful for its N-dimensional array objects.
* pandas - Python data analysis library, including structures such as dataframes.
* matplotlib - 2D plotting library producing publication quality figures.
* scikit-learn - the machine learning algorithms used for data analysis and data mining tasks.

**CHAPTER 6**

**SNAPSHOTS**

**6.1 SNAPSHOTS**

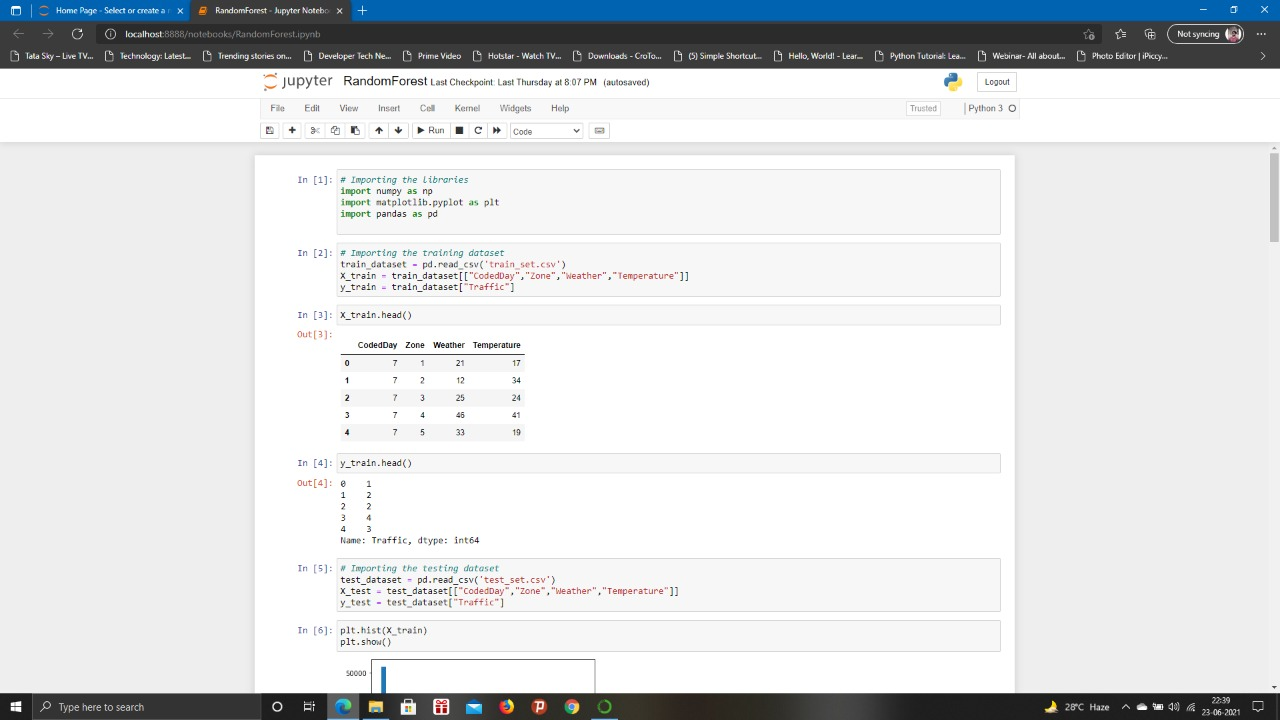


Fig 6.1 Importing libraries and importing training dataset

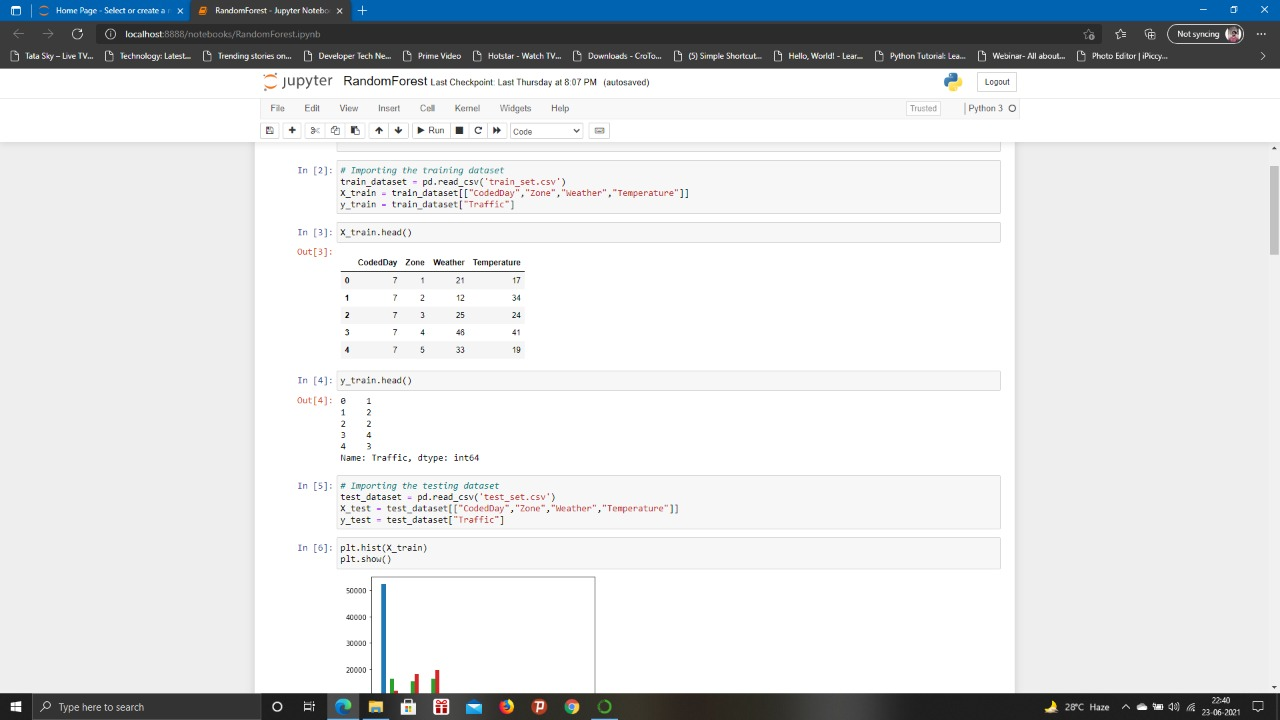


Fig 6.2 Examining training data

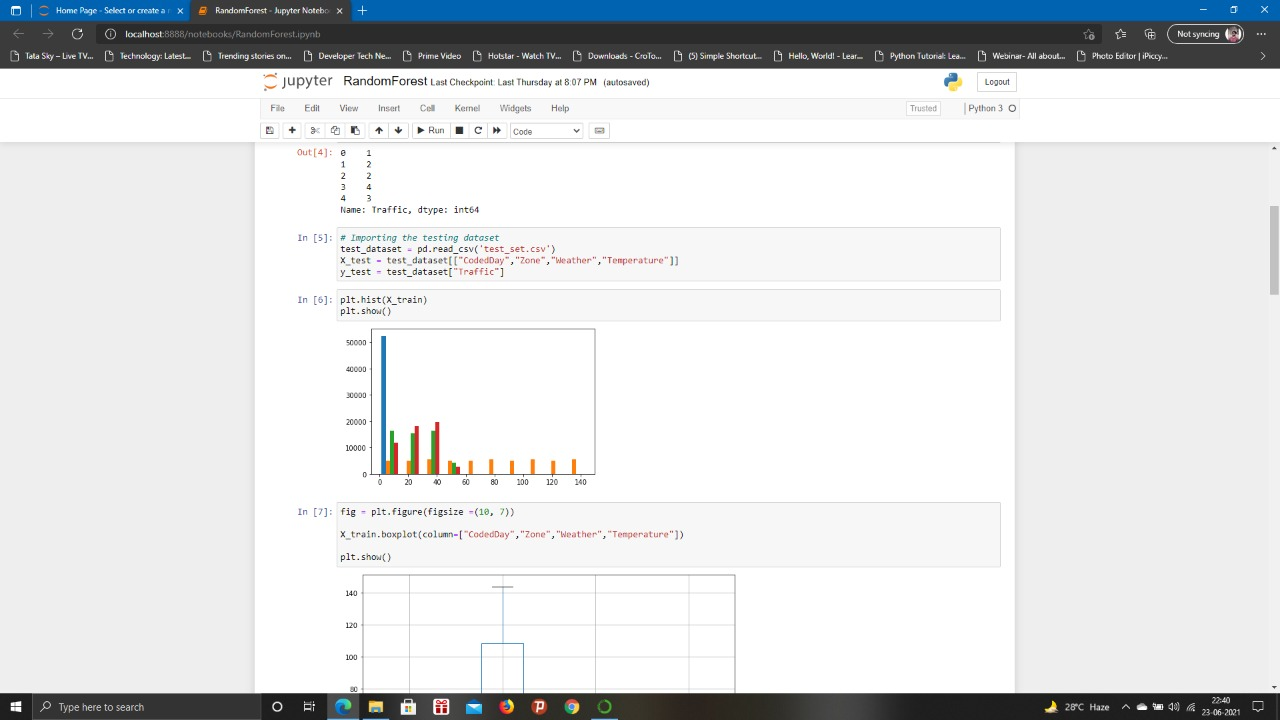


Fig 6.3 Importing testing dataset

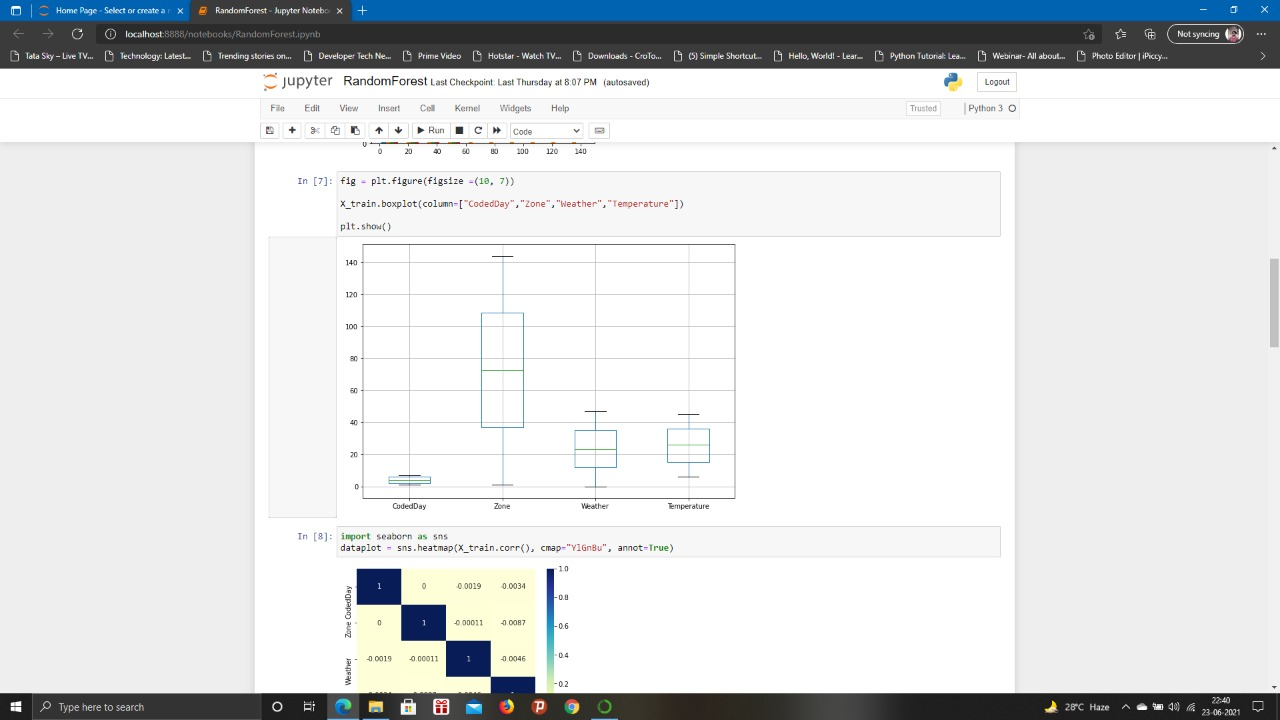


Fig 6.4 Plotting train attributes

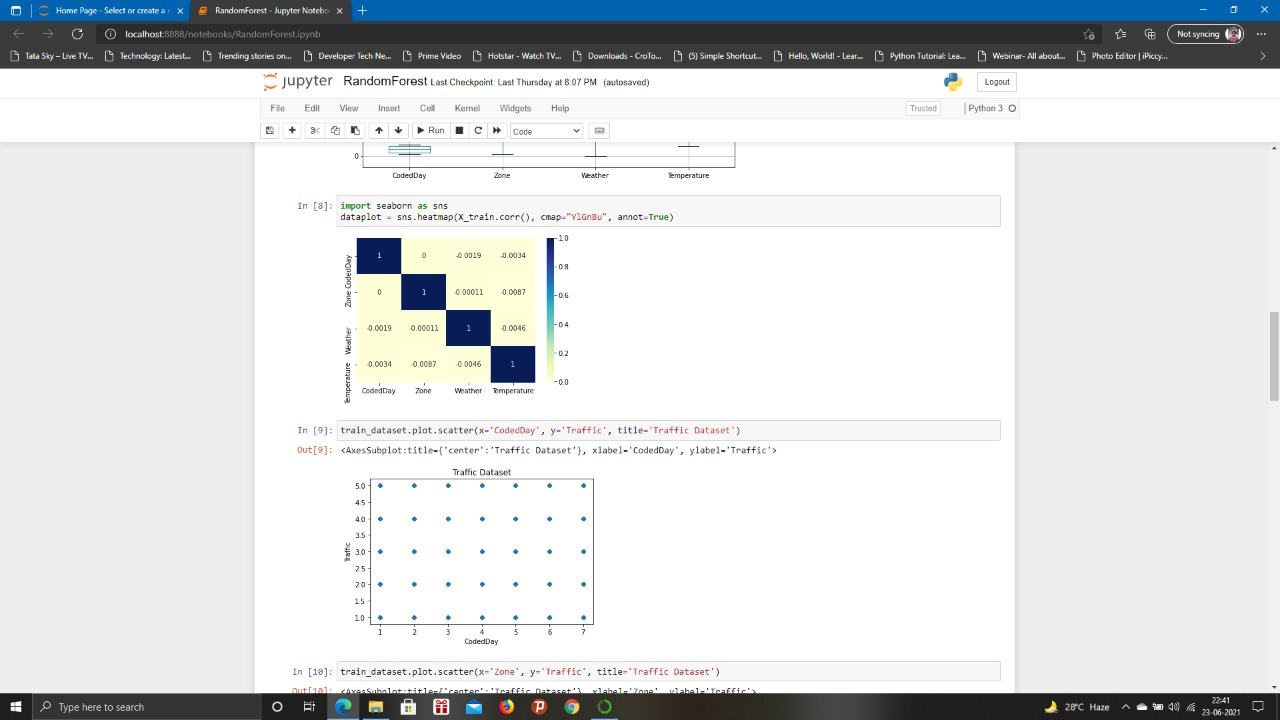


Fig 6.5 Visualizing data

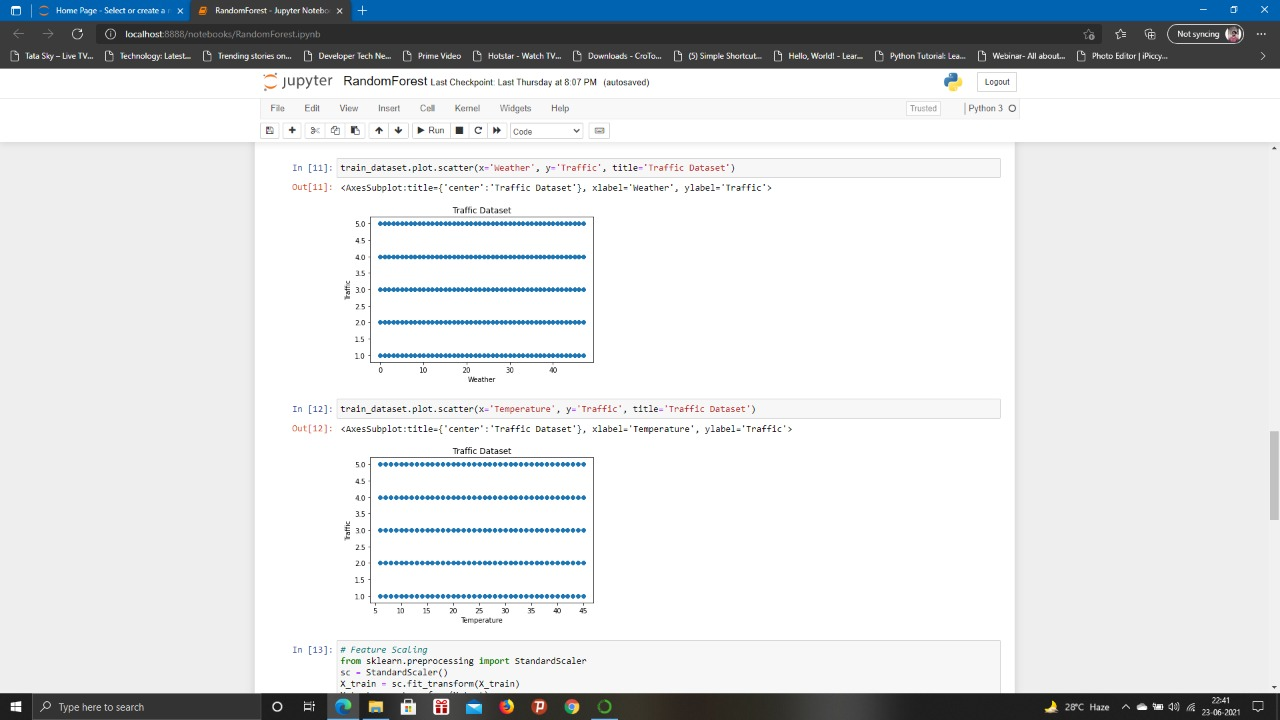


Fig 6.6 Visualizing traffic

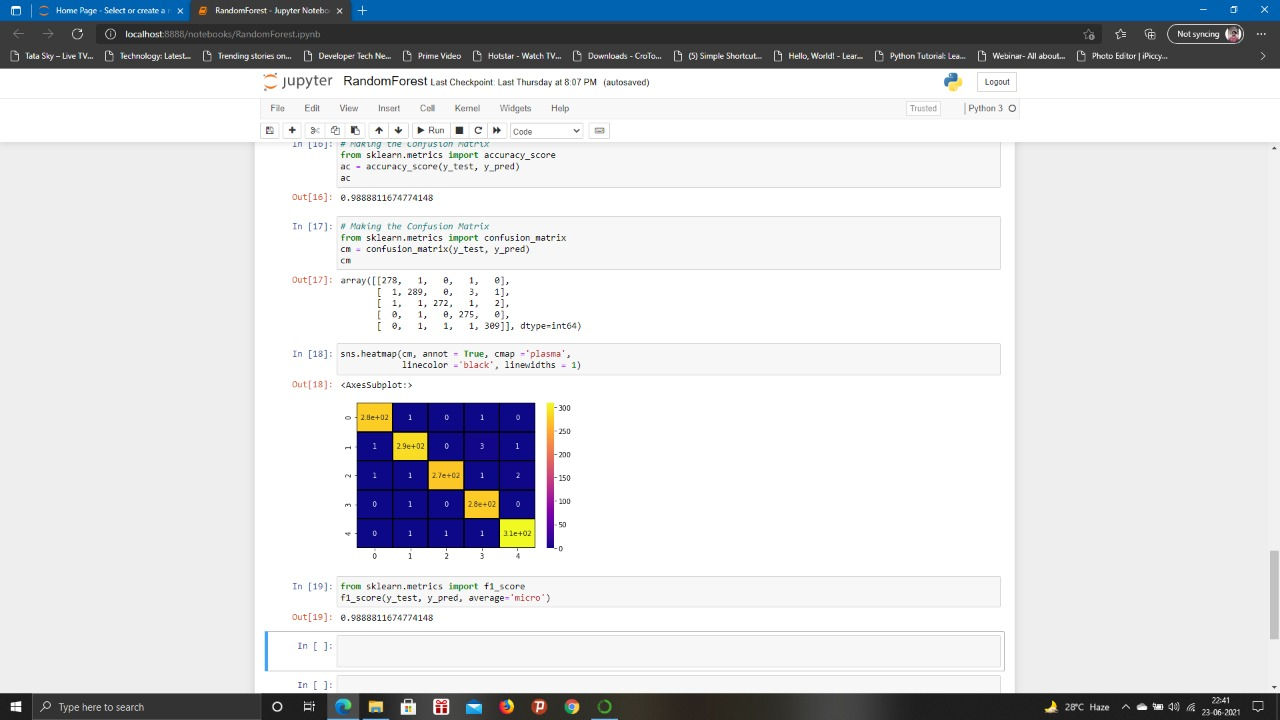


Fig 6.7 Making confusion matrix and checking f1-score

**Graphical user interface, application

Description automatically generated**

Fig 6.8 Making confusion matrix and checking f1-score for Random forest

**CHAPTER 7**

**IMPLEMENTATION**

**7.1 GENERAL**

**CODING:**

# Importing the libraries

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

# Importing the training dataset

train\_dataset = pd.read\_csv('/content/sample\_data/csvs/train\_set.csv')

X\_train = train\_dataset[["CodedDay","Zone","Weather","Temperature"]]

y\_train = train\_dataset["Traffic"]

#from google.colab import drive

#drive.mount('/content/drive')

y\_train.head()

# Importing the testing dataset

test\_dataset = pd.read\_csv('/content/sample\_data/csvs/train\_set.csv')

X\_test = test\_dataset[["CodedDay","Zone","Weather","Temperature"]]

y\_test = test\_dataset["Traffic"]

#fig = plt.figure(figsize =(10, 7))

#X\_train.boxplot(column=["CodedDay","Zone","Weather","Temperature"])

#plt.show()

train\_dataset.plot.scatter(x='CodedDay', y='Traffic', title='Traffic Dataset')

train\_dataset.plot.scatter(x='Zone', y='Traffic', title='Traffic Dataset')

train\_dataset.plot.scatter(x='Weather', y='Traffic', title='Traffic Dataset')

# Feature Scaling

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

X\_train = sc.fit\_transform(X\_train)

X\_test = sc.transform(X\_test)

# Fitting K-NN to the Training set

from sklearn.neighbors import KNeighborsClassifier

classifier = KNeighborsClassifier(n\_neighbors = 3, metric = 'minkowski', p = 2)

classifier.fit(X\_train, y\_train)

# Predicting the Test set results

y\_pred = classifier.predict(X\_test)

y\_pred

# Making the Confusion Matrix

from sklearn.metrics import accuracy\_score

ac = accuracy\_score(y\_test, y\_pred)

ac

sns.heatmap(cm, annot = True, cmap ='plasma',

            linecolor ='black', linewidths = 1)

sns.heatmap(cm, annot = True, cmap ='plasma',

            linecolor ='black', linewidths = 1)

**RANDOM FOREST IMPLEMENTATION**

# Importing the libraries

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

# Importing the training dataset

train\_dataset = pd.read\_csv('train\_set.csv')

X\_train = train\_dataset[["CodedDay","Zone","Weather","Temperature"]]

y\_train = train\_dataset["Traffic"]

X\_train.head()

# Importing the testing dataset

test\_dataset = pd.read\_csv('test\_set.csv')

X\_test = test\_dataset[["CodedDay","Zone","Weather","Temperature"]]

y\_test = test\_dataset["Traffic"]

plt.hist(X\_train)

plt.show()

fig = plt.figure(figsize =(10, 7))

X\_train.boxplot(column=["CodedDay","Zone","Weather","Temperature"])

plt.show()

import seaborn as sns

dataplot = sns.heatmap(X\_train.corr(), cmap="YlGnBu", annot=True)

train\_dataset.plot.scatter(x='CodedDay', y='Traffic', title='Traffic Dataset')

train\_dataset.plot.scatter(x='Zone', y='Traffic', title='Traffic Dataset')

train\_dataset.plot.scatter(x='Weather', y='Traffic', title='Traffic Dataset')

# Feature Scaling

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

X\_train = sc.fit\_transform(X\_train)

X\_test = sc.transform(X\_test)

# Feature Scaling

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

X\_train = sc.fit\_transform(X\_train)

X\_test = sc.transform(X\_test)

# Predicting the Test set results

y\_pred = rand\_classifier.predict(X\_test)

y\_pred

# Making the Confusion Matrix

from sklearn.metrics import accuracy\_score

ac = accuracy\_score(y\_test, y\_pred)

ac

# Making the Confusion Matrix

from sklearn.metrics import confusion\_matrix

cm = confusion\_matrix(y\_test, y\_pred)

cm

**CHAPTER 8**

**SOFTWARE TESTING**

**8.1 GENERAL**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

**8.2 DEVELOPING METHODOLOGIES**

The test process is initiated by developing a comprehensive plan to test the general functionality and special features on a variety of platform combinations. Strict quality control procedures are used.The process verifies that the application meets the requirements specified in the system requirements document and is bug free. The following are the considerations used to develop the framework from developing the testing methodologies.

**8.3TYPES OF TESTS**

**8.3.1 UNIT TESTING**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**8.3.2 FUNCTIONAL TEST**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

**8.3.3 SYSTEM TEST**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**8.3.4 PERFORMANCE TEST**

The Performance test ensures that the output be produced within the time limits,and the time taken by the system for compiling, giving response to the users and request being send to the system for to retrieve the results.

**8.3.5 INTEGRATION TESTING**

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**8.3.6 ACCEPTANCE TESTING**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**ACCEPTANCE TESTING FOR DATA SYNCHRONIZATION:**

* The Acknowledgements will be received by the Sender Node after the Packets are received by the Destination Node
* The Route add operation is done only when there is a Route request in need
* The Status of Nodes information is done automatically in the Cache Updation process

**8.2.7 BUILD THE TEST PLAN**

Any project can be divided into units that can be further performed for detailed processing. Then a testing strategy for each of this unit is carried out. Unit testing helps to identity the possible bugs in the individual component, so the component that has bugs can be identified and can be rectified from errors.

**CHAPTER 9**

**CONCLUSION AND FUTURE ENHANCEMENT**

**9.1 CONCLUSION**

Although deep learning and genetic algorithm is an important problem in data analysis, it has not been dealt with extensively by the ML community. The proposed algorithm gives higher accuracy than the existing algorithms also, It improves the complexity issues throughout the dataset. However, no research work, to the best of our knowledge does carry out to use learning data to measure content quality to improve it. Machine learning with AI has opened incredible possibilities in various fields. This fact inspired us to work on the traffic flow forecast problem build on the traffic data and models. It is cumbersome to forecast the traffic flow accurately because the data available for the transportation system is insanely huge

**9.2 FUTURE ENHANCEMENT**

Also we have planned to integrate the web server and the application. Also the things algorithms will be further improved to much more higher accuracy. , deep learning concepts attract many persons involving academicians and industrialist due to their ability to deal with classification problems, understanding of natural language, dimensionality reduction, detection of objects, motion modelling. DL uses multi-layer concepts of neural networks to mining the inherent properties in data from the lowest level to the highest level . They can identify massive volumes of structure in the data, which eventually helps us to visualize and make meaningful inferences from the data

**REFERENCES**

[1] Fei-Yue Wang et al. Parallel control and management for intelligenttransportation systems: Concepts, architectures, and applications. IEEETransactions on Intelligent Transportation Systems, 2010.

[2] Yongchang Ma, Mashrur Chowdhury, MansourehJeihani, and RyanFries. Accelerated incident detection across transportation networksusing vehicle kinetics and support vector machine in cooperation withinfrastructure agents. IET intelligent transport systems, 4(4):328–337,2010.

[3] RutgerClaes, Tom Holvoet, and Danny Weyns. A decentralizedapproach for anticipatory vehicle routing using delegate multiagentsystems. IEEE Transactions on Intelligent Transportation Systems,12(2):364–373, 2011.

[4] MehulMahrishi and SudhaMorwal. Index point detection and semanticindexing of videos - a comparative review. Advances in IntelligentSystems and Computing, Springer, 2020.

[5] Joseph D Crabtree and NikiforosStamatiadis. Dedicated short-rangecommunications technology for freeway incident detection: Performanceassessment based on traffic simulation data. Transportation ResearchRecord, 2000(1):59–69, 2007.

[6] H Qi, RL Cheu, and DH Lee. Freeway incident detection usingkinematic data from probe vehicles. In 9th World Congress on IntelligentTransport Systems ITS America, ITS Japan, ERTICO (IntelligentTransport Systems and Services-Europe), 2002.

[7] Z. Zhao, W. Chen, X. Wu, P. C. Y. Chen, and J. Liu. Lstm network:a deep learning approach for short-term traffic forecast. IET IntelligentTransport Systems, 11(2):68–75, 2017.

[8] C. Zhang, P. Patras, and H. Haddadi. Deep learning in mobile andwireless networking: A survey. IEEE Communications Surveys Tutorials,21(3):2224–2287, third quarter 2019.

[9] Chun-Hsin Wu, Jan-Ming Ho, and D. T. Lee. Travel-time predictionwith support vector regression. IEEE Transactions on IntelligentTransportation Systems, 5(4):276–281, Dec 2004.

[10] Yan-Yan Song and LU Ying. Decision tree methods: applications forclassification and prediction. Shanghai archives of psychiatry, 27(2):130,2015.

[11] Yiming He, Mashrur Chowdhury, Yongchang Ma, and PierluigiPisu.Merging mobility and energy vision with hybrid electric vehicles andvehicle infrastructure integration. Energy Policy, 41:599–609, 2012.

[12] Jason Brownlee. Bagging and random forest ensemble algorithms formachine learning.