A Mini Project Report on

"PREDICTING TRAFFIC CONDITIONS USING MACHINE LEARNING"

Submitted to the

GURU NANAK INSTITUIONS TECHNICAL CAMPUS

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

BACHELOR OF TECHNOLOGY

with MINOR DEGREE

IN

COMPUTER SCIENCE & ENGINEERING
(Artificial Intelligence & Machine Learning)

BY

NEELA SAI KUMAR	**(19WJ1A05L1)**
P. AKSHAY KUMAR	**(19WJ1A05P0)**
P. RAHUL	**(19WJ1A05P3)**
S. EESH KUMAR	**(19WJ1A05Q3)**
RAVI SAI ALEKHYA	**(19WJ1A05Q6)**

Under the Esteemed Guidance Of

Mrs. Sumitra Mallick

Associate Professor

GURU NANAK INSTITUTIONS TECHNICAL CAMPUS(AUTONOMOUS)
School of Engineering and Technology
Ibrahimpatnam, R.R District

2023

GURU NANAK INSTITUTIONS TECHNICALCAMPUS(AUTONOMOUS)

School of Engineering & Technology
Ibrahimpatnam, R.R District
501 506

CERTIFICATE

This is to certify that this project report entitled **"PREDICTING TRAFFIC CONDITIONS USING MACHINE LEARNING"** by **N.Sai Kumar (19WJ1A05L1), P.Akshay Kumar(19WJ1A05P0), P.Rahul (19WJ1A05P3), S.Eesh Kumar(19WJ1A05Q3), Ravi Sai Alekhya(19WJ1A05Q6)** submitted in partial fulfillment of the requirements for the degree of **Bachelor of Technology with Minor Degree in Computer Science & Engineering(AI&ML)**of the **GURU NANAK INSTITUIONS TECHNICAL CAMPUS** during the academic year 2022-2023, is Bonafide record of work carried out under our guidance and supervision.

Internal Guide	**Minor Degree Coordinator**
Mrs. Sumitra Mallick	**Dr. S. Madhu**

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We wish to express our sincere thanks to **Dr. Rishi Sayal**, Professor and Associate Director of CSE(AIML) GNITC for providing us the conductive environment for carrying through our academic schedules and Project with ease.

We have been truly blessed to have a wonderful adviser D**r. S. Madhu**, Professor& HOD of CSE(AIML), GNITC for guiding us to explore the ramification of our work and we express our sincere gratitude towards him for leading me thought the completion of Project.

We specially thank our internal guide **Mrs. Sumitra Mallick**, Associate Professor, for his suggestions and constant guidance in every stage of the project. Finally, we would like to thank our family members for their moral support and encouragement to achievegoals.

DECLARATION

We **N.Sai Kumar (19WJ1A05L1), P.Akshay Kumar(19WJ1A05P0), P.Rahul (19WJ1A05P3), S.Eesh Kumar(19WJ1A05Q3), Ravi Sai Alekhya(19WJ1A05Q6)**here by declare that the project report entitled **"PREDICTING TRAFFIC CONDITIONS USING MACHINE LEARNING"** under the esteemed guidance of **Mrs. Sumitra Mallick**, Associate Professor, Department of Computer Science Engineering submitted in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science Engineering. This is to any other University or Institute for the award of Degree or Diploma.

Date:

Place: GNITC

NEELA SAI KUMAR	**(19WJ1A05L1)**
P. AKSHAY KUMAR	**(19WJ1A05P0)**
P. RAHUL	**(19WJ1A05P3)**
S. EESH KUMAR	**(19WJ1A05Q3)**
RAVI SAI ALEKHYA	**(19WJ1A05Q6)**

ABSTRACT

KEYWORDS: Traffic environment, Random forest, classification, Real-time Database, Congestion, weather data.

There is a worldwide problem with traffic jams. The problem of traffic congestion is particularly acute in urban areas when vehicle numbers are increasing at a rate quicker than the city's road infrastructure can keep pace. The route is congested as a result of the congestion. The problem impacts many facets of contemporary life, including economic growth, traffic accidents, increased greenhouse gas emissions, lost time, and health damage. Due to traffic jams, much of the working day is squandered, which has a negative impact on the overall economy. Traffic analysis and the effects it has on the environment have been the subject of several publications and educational materials. However, the outcome has been less than ideal. In addition to limiting access and movement, traffic bottlenecks have a negative impact. Companies and employees who transport products and services suffer as a result of an increase in the amount of time spent traveling and the money spent on fuel due to traffic congestion.

To measure congestion, one widely used method is to measure traffic density (the number of cars per mile). There are millions of people throughout the world whose lives are negatively impacted by traffic every day. Air and noise pollution are worsened by traffic congestion, which contributes to the suffocating environment. Traffic Circle Incorporates everything that might impede traffic flow, from traffic signals and other problems to road maintenance and even protests to clearing the scene of accidents that take a long time to remove. So it would be preferable if we had previous knowledge of ordinary life issues that impact traffic flow so that the driver can make the proper judgments. Data such as location, date, time, and weather will be analyzed in this project in order to develop a prediction model that can provide users with precise traffic information. The primary objective is to simplify the prediction model while improving its accuracy.

CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

RF	Random Forest
SVM	Support vector machine
DT	Decision Tree
ITS	Intelligent transportation system

CHAPTER 1

INTRODUCTION

1.1 Introduction:

Traffic congestion has become a serious issue in many developing countries and big cities worldwide. Due to the current state of civilization, everyone must cope with traffic bottlenecks. In several regions of the world, common roadways have been flooded. Surprisingly, despite several experimental medications used to relieve congestion, no progress has been made. Every year, traffic congestion worsens. Finding out what causes traffic jams is critical. Congestion is caused by simultaneous trips desired daily As a result, both the educational system and the market demand students to work and attend school full-time. Because this is the only method that individuals can work together, this is an excellent tool for programs This mechanism can't be changed since doing so would have a detrimental impact on the community. There are various strategies to minimize the impact of traffic on our everyday lives, despite the fact that eliminating traffic congestion is nearly impossible. In order to accommodate the daily rise in traffic, roadways need to be widened to accommodate more automobiles. A limited amount of this can be accomplished. It's not possible to expand roads beyond a certain point. Using current lanes and loads efficiently means using a variety of control measures. In order to avoid congested locations and use alternate routes, these traffic management systems rely on prediction models that take previous data and extrapolate it into the future.

People in the private and public sectors alike rely on accurate data on traffic flow patterns. It aids commuters and motorists in making better travel decisions, therefore reducing traffic congestion, enhancing traffic operations, and lowering carbon emissions. Improvements in traffic flow prediction are made possible by the development and implementation of Intelligent Transportation Systems (ITS). We consider it vital to current traffic management, public transit, and traveler information services. Sensors like inductive loops, radars, cameras, crowdsourcing via mobile GPS, and social media all contribute to traffic flow. Traffic The extensive usage of both old and new technologies has ushered in an era of a huge movement of information. New data-driven methods for controlling and managing transportation are becoming more popular.

As a result of this, many existing traffic flow prediction systems use shallow traffic models that nonetheless fail because of the enormous dataset size. of traffic data. but people should not forget their roots and local languages should not be ignored.

Some Major Causes of Traffic Jams

- Motorcycles, public transportation, and other vehicles that require more than one lane can cause gridlock in several nations. There will be issues if everyone utilizes the same road.
- Due to regions like a marker, a busy area, and stores that fill the wayside, there are numerous tiny lanes that generate traffic congestion.
- We can no longer disregard the problem of population growth. There has been a rise in transportation as the world's population has grown. All roads, even 6-lane ones, are congested because of a shortage of available room. Some nations, such as India, have awful traffic difficulties because people don't respect traffic laws. Some of the cities still have narrow or one-lane streets, which lead to long traffic jams because they were not all planned.
- Traffic has been impacted by overbuilt construction impediments such as flyovers, metros, and other overbuilt structures, even though they are designed to address the problem.
- The weather has a significant impact on traffic bottlenecks as well. Many routes become impassable during the rainy season, escalating the difficulty already present.
- Most Indian cities do not have regular footpaths. This segregates people into different social groups, and the absence of these paths can lead to traffic bottlenecks that last for hours. Strays on the road can create traffic congestion in various areas of the country.

The weather has a direct effect on traffic, as the basic sense suggests. But, how much? Whereas, and under what conditions? Weather data can be used to enhance traffic forecasts, but can we do better? Estimated trip times, ideal travel times, and new IoT data such as road conditions, accidents, etc. are all considered to have a big influence on how drivers arrange their day. Date, day (coded), weather, temperature, location (coded), and traffic are all included in our model. The accuracy of traffic data may be improved by including weather data in addition to other types of traffic data.

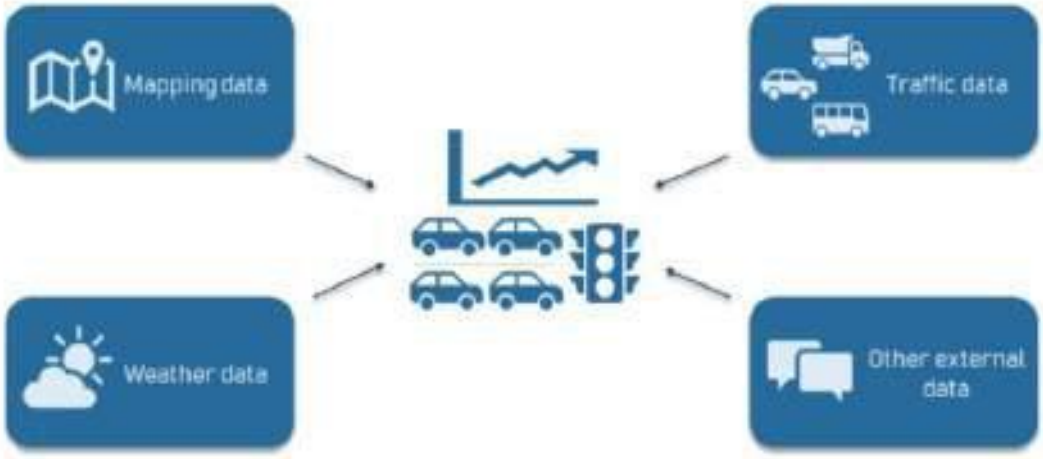
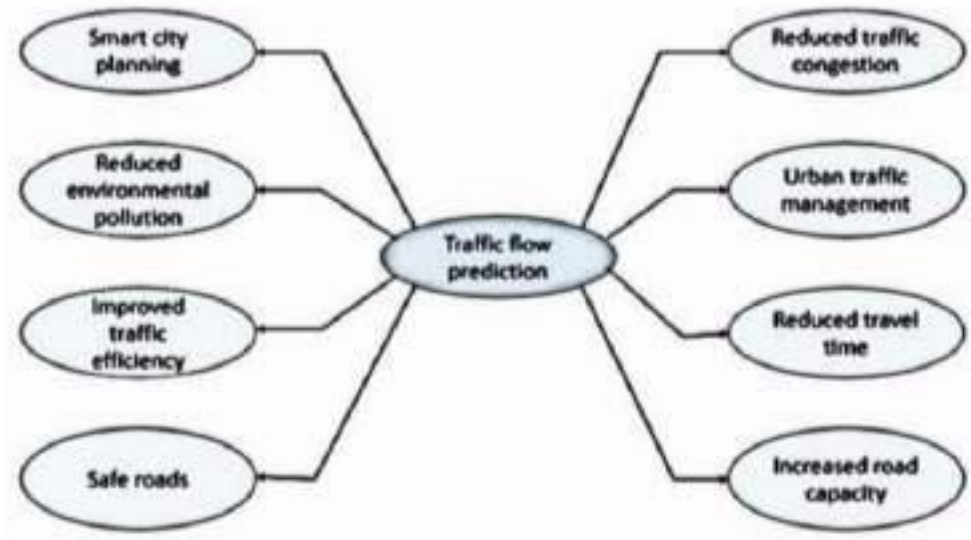


FIG 1

1.2 Benefits of Traffic Prediction

Traffic forecasting has many benefits :

- It is used to reduce traffic congestion.
- It is used to reduce environmental pollution.
- It is also used for urban traffic management.
- By anticipating traffic we can diminish travel time.
- And furthermore we can make brilliant smart city planning.



1.3 Problem Statement

The issue of traffic management has been increasingly problematic in recent years and is only expected to get worse. A major issue we face as a result of an increase in both people and automobiles is traffic congestion. Traffic jams are causing a lot of problems for the general public. Our goal is to properly estimate traffic and make this information available to users so that they may take other routes to avoid congested hotspots. We will be able to better estimate traffic situations as a consequence of this initiative. Even if it's raining, overcast, sunny, on the weekend, or on a normal day, there's no need to worry about the weather. In towns and cities, traffic is heavy; in rural regions, such as farms and forests, traffic is light. All of these variables have an impact on the flow of traffic on the road. We can properly forecast traffic flow based on all of these variables.

1.4 Objectives

- The major purpose is to evaluate the traffic data to forecast traffic situations.
- To increase the model's accuracy - there are numerous prediction models out there, but accuracy is the most vital for a prediction model.
- To lessen the amount of traffic on the roads.
- Improving air quality by reducing transportation congestion.
- Finally,To find a solution that is

    a. Cost-effective

    b. Adaptable

    c. Easy for use.

1.5 Requirement Analysis

The team studied other apps' designs to make the app more user-friendly. To do this, the user has to type as little as possible while preserving a logical flow from screen to screen. To make the program more accessible, the browser version has to be compatible browser.

1.5.1 Software Requirements

HARDWARE REQUIREMENTS:

- System : Pentium IV 2.4 GHz.
- Hard Disk : 40 GB.
- Floppy Drive : 1.44 Mb.
- Monitor : 15 VGA Color.
- Mouse : Logitech.
- Ram : 512 Mb.

SOFTWARE REQUIREMENTS:

- **Operating System:** Windows

- **Coding Language**: Python 3.7

INPUT AND OUTPUT DESIGN

INPUT DESIGN

It connects the user to the information system. To prepare transaction data for processing, it is necessary to develop specifications and methods. These steps involve having people enter data into the system or having the computer read data from a written or printed document. reducing input, reducing errors, reducing delays, eliminating redundant operations, and keeping simplicity are all goals of input design. To ensure security and convenience of use while preserving privacy, the input is created.

OBJECTIVES

- Computer systems are designed from user-oriented descriptions of their input, which is known as Input Design. This design ensures that the data input procedure is error-free and that the proper information may be obtained from the computerized system.

- To do this, provide user-friendly data input panels that can handle enormous volumes of data. The goal of input design is to make data entry as simple and error-free as possible. For any data modifications, a data entry screen has been constructed. You may also utilize it to review the data you've recorded.

- Data will be validated as soon as it is input. Information may be entered through displays. The user will not be bombarded with messages if the proper ones arrive at the right time. As a consequence, input design strives to provide a user-friendly interface.

OUTPUT DESIGN

It's all about addressing the demands of the end-user when it comes to high-quality products. Outputs are how a system's results are conveyed to users or another system. Immediate usage and hard copy output design affect how information is distributed. The dashboard is the user's main source of data. The system's output's efficiency and intelligence help users make better decisions.

- The right output must be created while ensuring that each output portion is designed to make the system simple and effective for people to use. When examining a computer's output architecture, it's vital to establish the specific output necessary.
- Choose how to show the data.
- Produce papers, reports, or other forms that include the data that you have gathered.

1.6 Thesis Organization:

Chapter 1: This chapter presents the basic information and introduction and necessary technical knowledge to implement.

Chapter 2: This chapter presents the literature survey. It summarizes the findings and research needed to implement the project.

Chapter 3: This chapter deals with a detailed methodology and approach.

Chapter 4: This chapter shows the output and results with a detailed explanation.

Chapter 5: This chapter represents the conclusion and the future scope of the project.

CHAPTER 2

LITERATURE SURVEY

1) Accelerated Incident Detection across Transportation Networks using Vehicle Kinetics and Support Vector Machine in Cooperation with Infrastructure Agents

AUTHORS: Ma, Yongchang, Chowdhury, Mashrur, Jeihani, Mansoureh, Fries, Ryan

Using vehicle dynamics, such as speed profile and lane-changing behavior, this research identifies highway incidents. Mobility and safety may be improved by integrating vehicles and infrastructure with the VII (also known as IntelliDrive) concept. SVM, an in-vehicle intelligence module, is used by the framework to identify the journey experiences of a vehicle. Vehicle trip data is collected and compared to pre-set threshold values by roadside infrastructure agents (also known as RSUs, or roadside units) in order to identify an incident. For this experiment, we used a rural Spartanburg, South Carolina roadway network and an urban Baltimore highway network. In terms of detecting performance, the VII-SVM system identified no major differences between old and new networks Many transportation systems may benefit from the general VII-SVM technology.

2)A Decentralized Approach for Anticipatory Vehicle Routing Using Delegate Multiagent Systems

AUTHORS: Qin Yu, Tao Jiang, Aiyun Zhou, Lili Zhang, Cheng Zhang & Pan Xu

Transparent, real-time traffic information is a key component of advanced vehicle guiding systems. They can only react to traffic delays and not prevent excessive congestion from being created by these systems. Because traffic forecast information may be used to direct vehicle routing, anticipatory vehicle routing is a viable technique. An anticipatory vehicle routing method that is particularly beneficial in dynamic contexts is presented in this research. It is built on a technique inspired in part by ant behavior called delegate multiagent systems, which is an environment-centric coordinating mechanism. Using antlike agents, cars may sense congestion in the surroundings and reroute around it. The method is thoroughly described and assessed in light of three different routing options.

3)Dedicated Short-Range Communications Technology for Freeway Incident Detection: Performance Assessment Based on Traffic Simulation Data

AUTHORS: Xuehu Wang, Yongchang Zheng, Lan Guan, Xuan Wang, Xinting Sang, Xiangfeng Kon Jie Zhao

Short-range communications technology can monitor rural highway travel times and instantly spot issues. A rural freeway segment was simulated using the CorSim traffic simulation tool. For each probe and beacon, postprocessing was employed. Uncertainty and counters were employed to build an event detection system. A counter had to reach a certain value to set off an alert. The findings were utilized to identify the appropriate thresholds for travel time and counter alarm levels. Using these ideal settings, the algorithm could quickly detect numerous traffic incidents. The system can identify events rapidly and reliably, according to the assessment findings. To test how changing certain parameters influenced the system's performance, simulation and analysis were performed. Each parameter had a considerable impact on detection time, and the observed results were in line with what one would predict. When the number of transponders, traffic volume, the severity of the event, and the population of transponders rose, the time it took to detect an incident decreased, as did the reader spacing and the distance from the incident to the next downstream reader.

4)FREEWAY INCIDENT DETECTION USING KINEMATIC DATA FROM PROBE VEHICLES

AUTHORS:Qin Yu, Tao Jiang, Aiyun Zhou, Lili Zhang, Cheng Zhang & Pan Xu

The research analyzed the speed and acceleration patterns of highway probe vehicles to detect collisions. A probe vehicle approaching an incident should slow down, then resume regular speed. For testing, a calibrated microscopic traffic simulation model with varying percentages of probe cars in the traffic stream was used. The results are compared to a fixed site volume, speed, and occupancy measuring technique. Probing vehicles may match the neural network model's detection rates and mean detection times.

5)LSTM network: a deep learning approach for short-term traffic forecast

AUTHORS:Joanna Jaworek-Korjakowska 1 and Paweł Kłeczek 1.

In an intelligent transportation system, traffic forecasting is vital. The most efficient modes of transportation, routes, and departure times are all dependent on accurate predictions. Improved traffic data analysis may enhance forecast accuracy. We are driven to improve short-term predictions by employing powerful machine learning methods. LSTM networks are proposed as a new foundation for traffic forecasting. Instead of a two-dimensional network sparse, The suggested LSTM network takes into account the temporal-spatial correlation in traffic systems while designing its memory units. The proposed LSTM network outperforms existing similar forecast models.

CHAPTER 3

METHODOLOGY

3.1 Architeture:

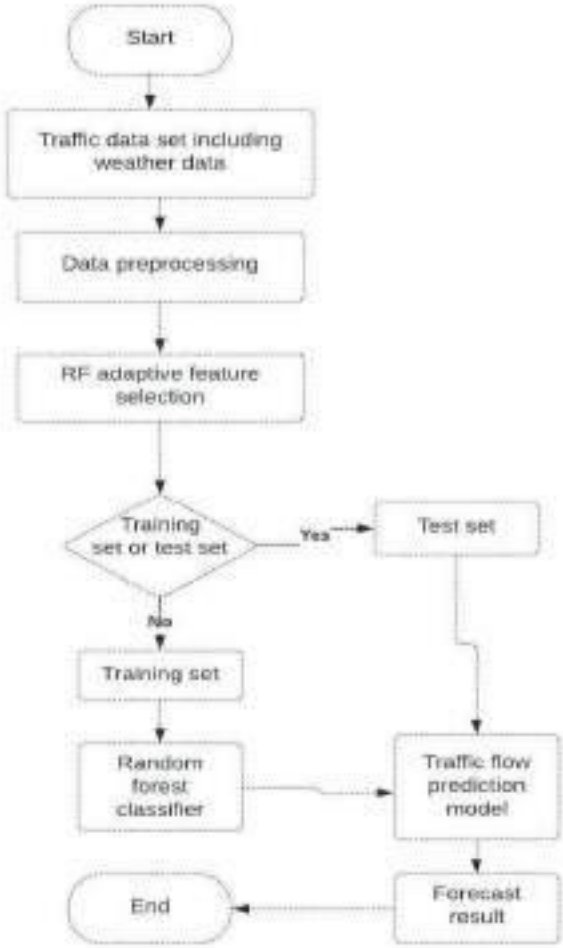


FIG 3

3.2 Data Description:

The data set consists of both training and testing data. To keep track of our data, we'll be using Excel spreadsheets. Predictive models rely on large amounts of historical data to learn how to make accurate predictions about a wide range of factors, such as time, date, day of week, zone (coded zone values), temperature, weather, and traffic. How accurate the model is will be determined by how much data is gathered. There is a considerable impact on traffic statistics due to weather conditions. We always use 80% of the data generated for training and 20% of the data gathered for testing purposes.

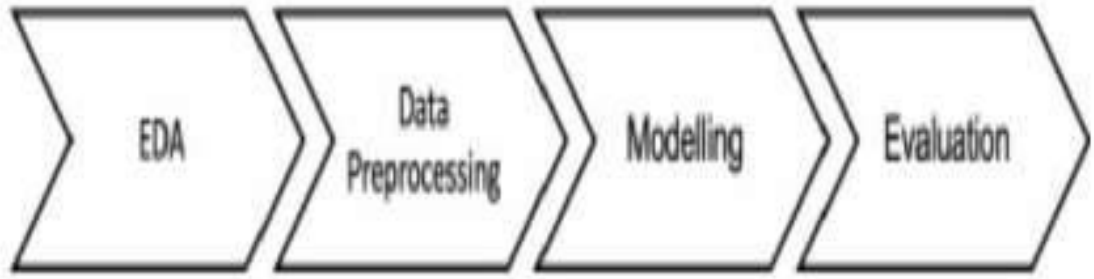


FIG 3.1

3.2.1 Data Exploration and Analysis:

We can get close to the guarantee that future events will be real, effectively comprehended, and most critically, beneficial to the selected organizational scenarios via the challenge of analytic pattern detection. Only existing data may be reviewed and evaluated as outliers in order to verify that the statistics have been collected rather than inconsistent. As a result of the geodatabase in the input data, cave file value systems will be examined here. A great deal of time and effort has been spent in order to compile list from all of the educational institutions in the nation. The collected input variables are listed below.

3.2.2 Data pre-processing:

Data pre-processing is a data mining technique that transforms raw data into an understandable format

Dealing with invalid data or missing values: The dataset may contain null values. To deal with or impute the missing values, we will group and replace the null values with their respective mean values.

Batch normalization statistics: the ability to turn the data into a template that machines can understand.

Feature selection: The process of selecting the most consistent, relevant, and non-redundant characteristics to be used in the construction of a model is known as feature selection. We need to develop new methods for reducing the size of datasets as they continue to rise in both quantity and complexity.

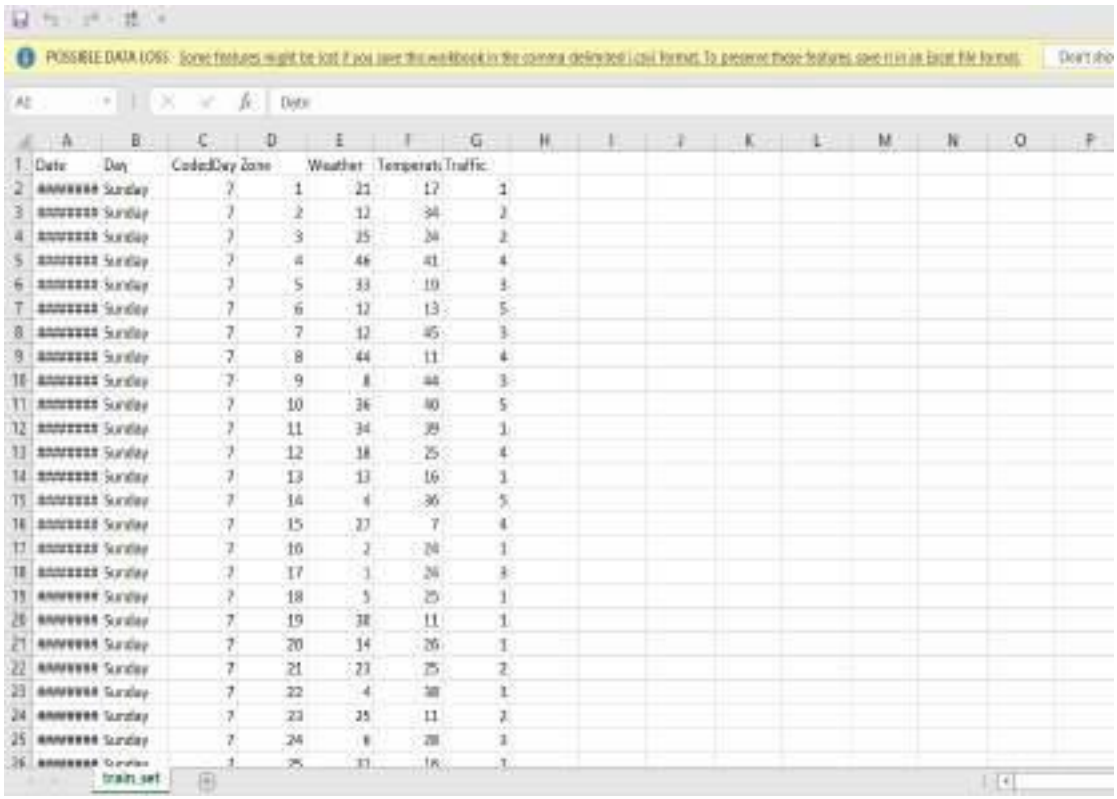
3.2.3 Splitting data into training and testing phases:

- Using the same dataset for both training and testing chances for miscalculations, thus increasing the chances of inaccurate predictions.
- The ideal split is said to be 80-20 for training and testing. You may need to adjust it depending on the size of the dataset and parameter complexity.
- Train test split: It is used for splitting data into random training and testing sets.

3.2.4 Modelling and evaluation:

We'll utilize the all-around handled and splattered information. Utilizing the information that has been assigned as "train data," a machine learning model is prepared. The manner in which we make our model and test it against information from a scope of future circumstances is the same way we do so presently.

Dataset used to train the model:



	A	B	C	D	E	F	G
1	Date	Day	CodedDay	Zone	Weather	Temperatu	Traffic
2	########	Sunday	7	1	21	17	1
3	########	Sunday	7	2	12	34	2
4	########	Sunday	7	3	25	24	2
5	########	Sunday	7	4	46	41	4
6	########	Sunday	7	5	33	10	3
7	########	Sunday	7	6	12	13	5
8	########	Sunday	7	7	12	45	3
9	########	Sunday	7	8	44	11	4
10	########	Sunday	7	9	8	44	3
11	########	Sunday	7	10	36	40	5
12	########	Sunday	7	11	34	39	1
13	########	Sunday	7	12	18	25	4
14	########	Sunday	7	13	13	16	1
15	########	Sunday	7	14	4	36	5
16	########	Sunday	7	15	27	7	4
17	########	Sunday	7	16	2	24	1
18	########	Sunday	7	17	1	26	3
19	########	Sunday	7	18	5	25	1
20	########	Sunday	7	19	38	11	1
21	########	Sunday	7	20	14	26	1
22	########	Sunday	7	21	23	25	2
23	########	Sunday	7	22	4	38	1
24	########	Sunday	7	23	35	11	2
25	########	Sunday	7	24	8	28	3
26	########	Sunday	7	25	33	16	1

FIG 3.2

3.3 SOFTWARE TOOLS USED:

Python:

Python is an object-oriented, dynamically semantic, and interpretive high-level programming language. Its high-level built-in data structures, dynamic typing, and dynamic binding make it ideal for use as a scripting or glue language. features. The syntax of Python is straightforward and easy to pick up. It places a high priority on readability, which reduces the cost of maintaining software. Modules and packages make it easy to divide up software into smaller components and reuse code in the Python programming language. On all major systems, you may obtain a free binary or source file version of the Python parser and standard library. What Python Can Do Python has a number of useful features.

- It's a piece of cake to pick up new skills. Python's syntax is simple, with just a few keywords. As a result, the content is easier for the student to understand.• Python code is easier to read because it is more clearly written and easy to see.
- Python's source code is relatively simple to maintain.
- A vast collection of common books. The majority of Python's library may be used on UNIX, Windows, or Macintosh systems.
- Python's "interactive" mode allows you to test and make corrections to code in real-time.
- There are a wide number of hardware systems that Python can run on, and it has the same user interface regardless of the platform.
- Adding low-level modules to the Python interpreter may expand its functionality. To improve the efficiency of their tools, programmers may use these modules.
- Additionally, Python has interfaces for MySQL, Oracle, and SQL Server databases.
- There are a variety of system calls, libraries, and window systems that may be used to create Python GUI applications, including Macintosh OS X and the X Window System under Unix.
- Shell scripting's structure and support for huge programs are superior to those of Python. In addition to the capabilities described above, Python includes a slew of more. Here are a few examples.
- It supports OOP, as well as functional and structured programming.
- A scripting language, particularly one that can be turned into byte code, may be used to create massive programs.

Django: Developed in Python, Django is a high-level web framework for fast website creation. Django, a web framework created by experienced developers, allows you to concentrate on designing the app rather than re-inventing the wheel. Free or paid services are available. It's open source and completely free to use and modify.

You may build programs that are

:• Complete

• Versatile Using Django It's safe

• Easily expandable

• Ease of use

• Portable

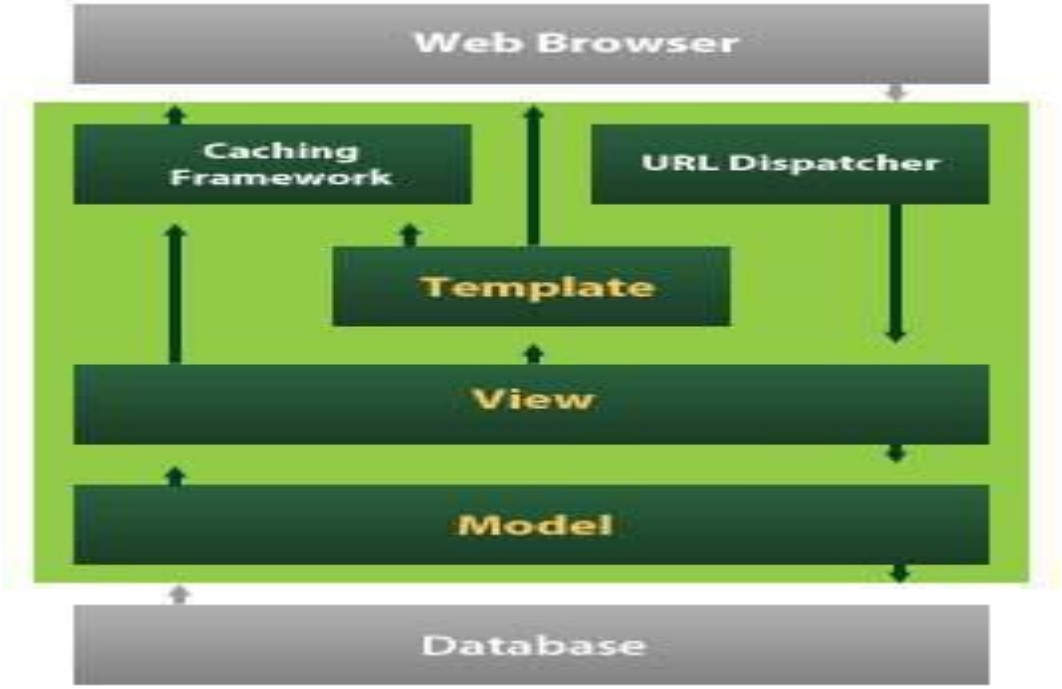


FIG 3.3

Create a Project

As long as you have access to a command-line terminal or command-line prompt, you may use this command to build your project.

Start my project using $ Django-admin start project.

A "my project" folder will be created with the following structure:

```
my project/

manage.py

my project/

__init .py

settings.py

settings.py URLs.py

 wsgi.py
```

The Project Structure

The “my project” folder is only a container for two items.

../manage This file is your project's local Django-admin for command-line interaction (start the development server, sync DB...). The code below returns a list of all manage.py commands.

–

Help for Python's manage.py

Subfolder my project Your project's Python code resides in this directory. There are four files in total.

Init.py Do not consider this folder to be a package for any other programming language than python.

In this file, named settings.py, you'll save all of the preferences for your project.

Each link in your project, as well as the function to invoke, are listed in urls.py. It's like a table of contents for your project.

If you want to use WSGI to distribute your project, you may use wsgi.py.

Setting the Stage for Your Work

My project's settings.py is located in the subdirectory myproject. Here are a few of the more essential ones.

It is possible that you may need to set it.

DEBUG = True

This option determines whether or not your project is now in debug mode. In debug mode, you may view more information regarding the problem. No usage in ongoing projects. To enable static file serving on the Django light server, set this to 'True'. It is only available in the development mode.

Databases

'Standard': 'Default':

Sqlite3 backends are listed as the engine in 'ENGINE': 'django db.backends sqlite3

database.sql is the name given to this file in the environment variable.

Input from the user is sent to the following socket: "",

'PASSWORD' is '', ''

In the 'Host' field: ', '

A port is referred to as "" or "" }}

The 'Database' dictionary has the value database. The SQLite engine is used in the example above. Django also supports MySQL, as previously indicated (django.db.backends.mysql)

Database backend PostgreSQL (postgresql psycopg2)

Databases such as Oracle and NoSQL

This engine is MongoDB (mongodb engine) for Django.

Before configuring a new engine, make sure to use the proper database driver.

Time Zone, Language Code, and Template are among the other options....

As soon as you get your project up and running, check to see whether everything is functioning as expected

runserver $ python manage.py

When you execute the above code, you'll see something like this.

It's all about testing models...

There were no mistakes discovered.

09/09/2015 11:41:50 - Thanks for stopping by!

myproject.settings' is used in Django version 1.6.11.1

We're going to start the development server at 127.0.0.1:8000 right now.

Press Ctrl-C to exit the server.

A project is a collection of applications. Like a website's form, an app's objective may be utilized in another project. Consider it a part of the overall project.

Create an Application

Our assumption is that you are in your project folder. It's in the same place as manage.py and $ python manage.py. launch my app

Django automatically creates a "myapp" folder with the name of the application you just established.

structure of the application

```
myapp/
__init .py
admin.py
models.py
tests.py
views.py
```

Init.py Just to be sure python understands that this folder is a package and may be used as such.

admin.py - This file allows you make the app's administration interface editable.

In this directory, you'll find all of your application models.

Your unit tests may be found in tests.py.

There are application views in this file.

Make Your Request Known to the Project

Django project registration is required at this point in order to use our "myapp" application.

"myproject". To achieve this, make the necessary changes to the INSTALLED APPS tuple in your project's settings.py file.

(Insert the name of your app here)

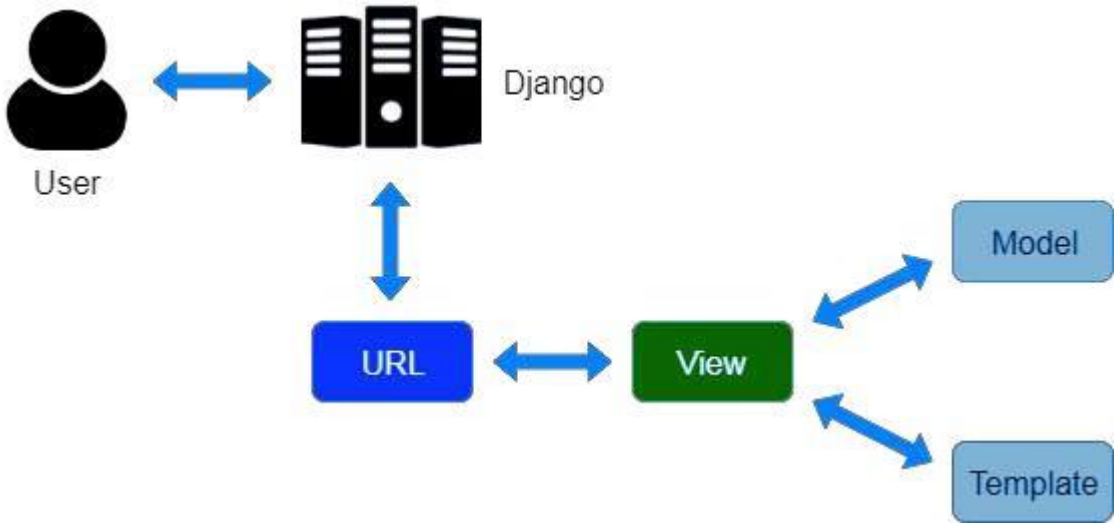


FIG 3.3.1

3.4 UML diagrams:

Non-software systems and business processes may both be represented using the Unified Modeling Language (UML).

For modeling large and complex systems, the UML is a collection of best practices from the technical community.

Object-oriented software development necessitates the use of UML. A great deal of software project design is expressed visually using the UML.GOALS:

The primary goals in the design of the UML are as follows:

1. Give people a ready-to-use visual modeling language to create and share meaningful models.

2. Extend the key notions with specialization and extensibility.

3. Be free from the constraints of programming languages and development methodologies.

4. A formal framework for comprehending the modeling language is provided in this step.

5. It's important to keep the OO tools industry growing.

6. Support collaborations, frameworks, patterns, and components.

7. Integrating excellent practices

3.4.1 USE CASE DIAGRAM:

A use-case analysis creates a UML use case diagram, which is a form of a behavioural diagram. Use case diagrams are used to show a system's functionality in terms of actors and goals (expressed as use cases) (represented as use cases). A use case diagram indicates which actors in the system are responsible for a certain task. The system's actors' roles may be presented.

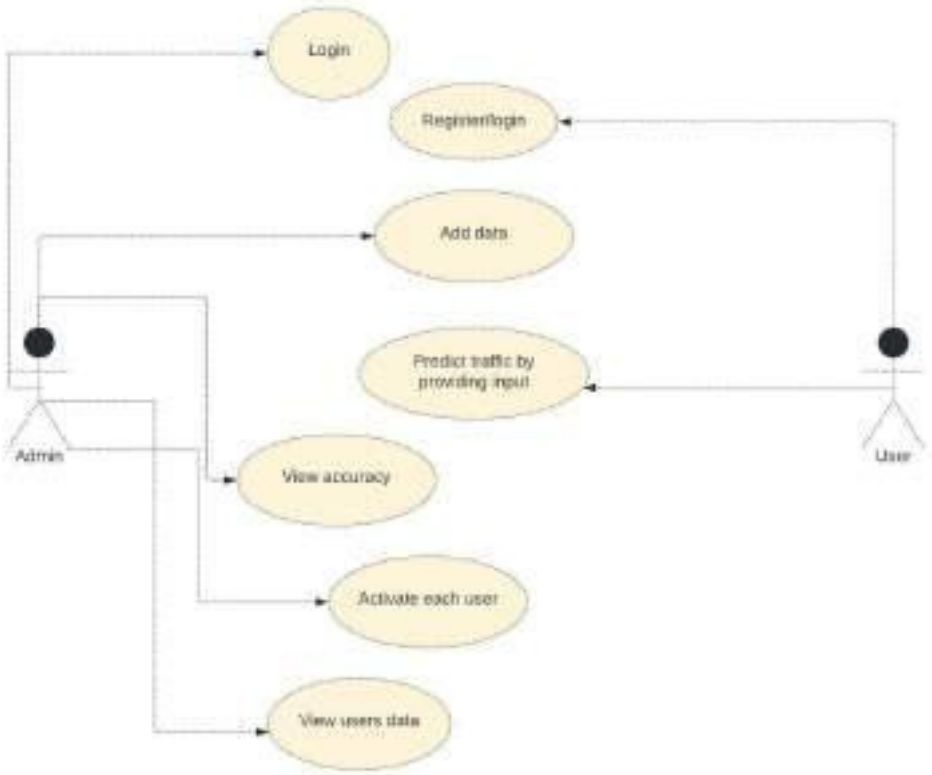


FIG 3.4 Object diagram

3.4.2 CLASS DIAGRAM:

There is a unified modeling language (UML) representation of the system's classes and their attributes, as well as their operations (or methods) (UML). Which class contains what?

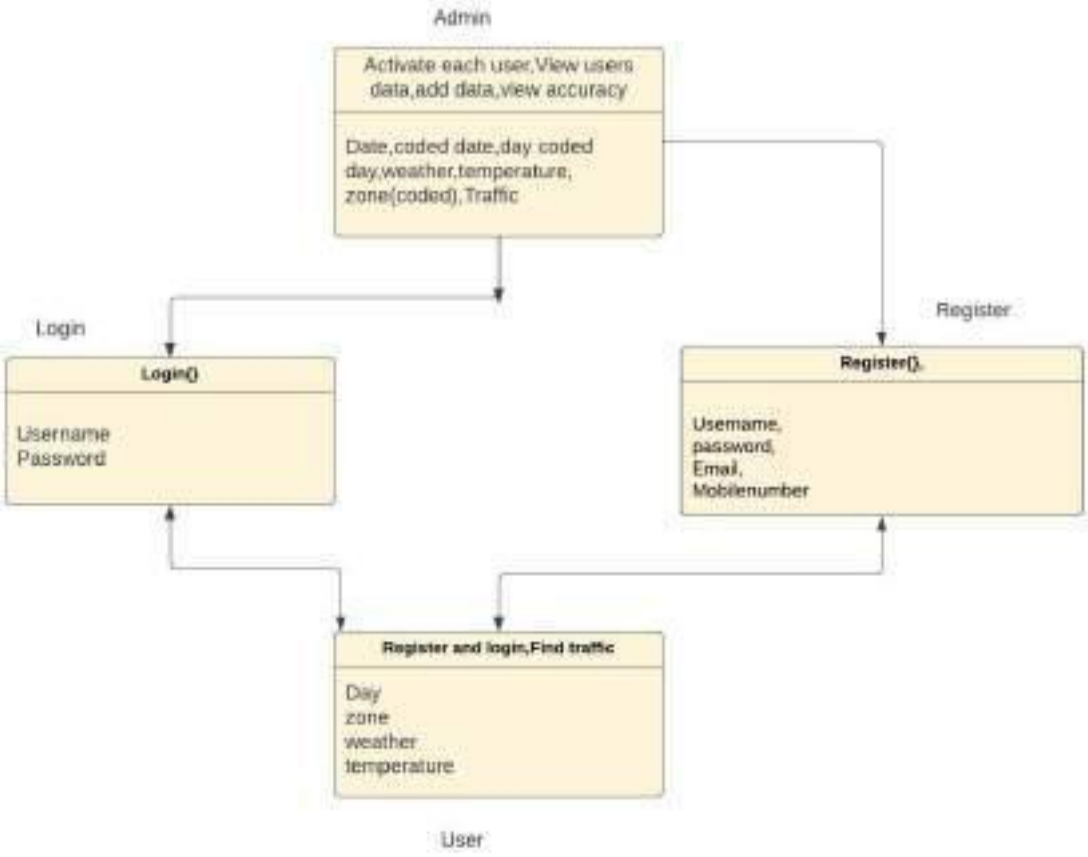


FIG 3.4.1 Class diagram

3.4.3 SEQUENCE DIAGRAM:

In UML, a sequence diagram shows how processes interact with one other and in what order they are completed. Message Sequence Chart is the term used to describe it. Event scenarios and timing diagrams are also known as these.

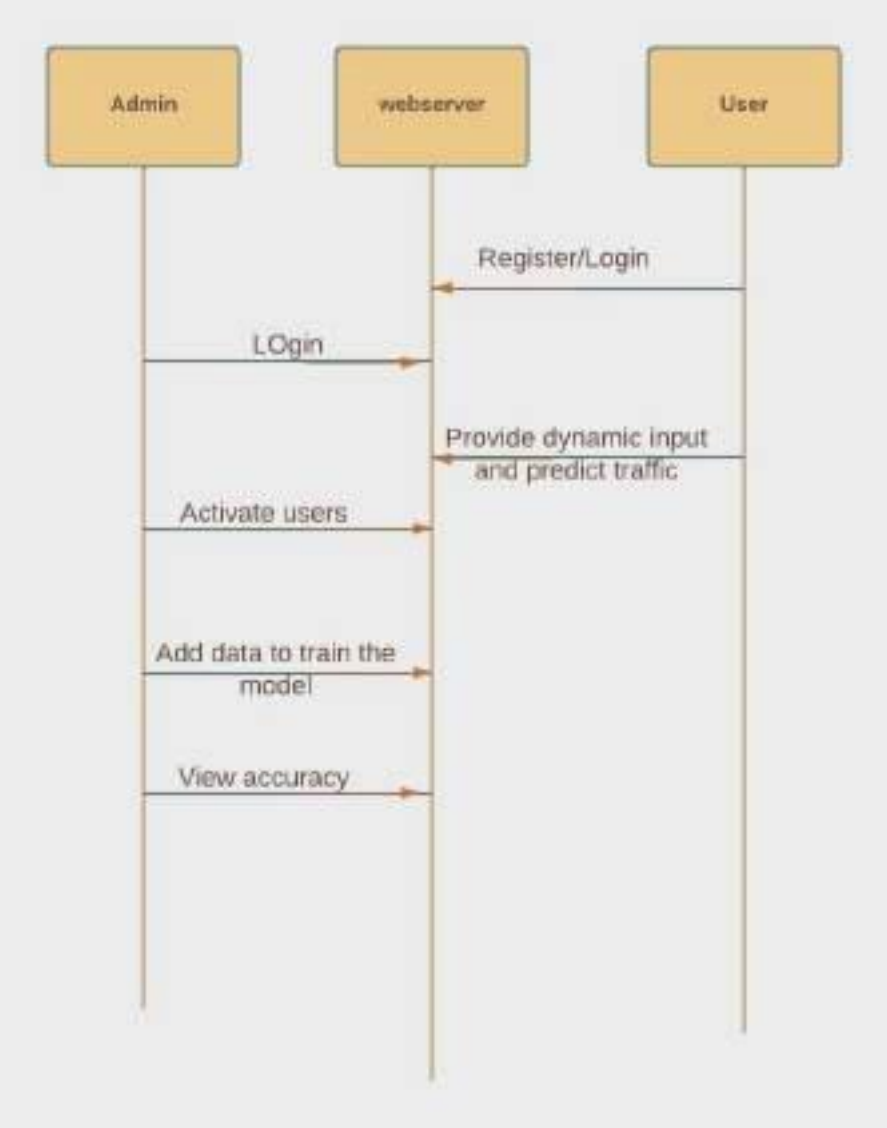


FIG 3.4.2 Sequence diagram

3.4.4 DATA FLOW DIAGRAM:

These are set points epithetical methodology epithetical fragmentary hobbies and motions with basis due to selection, monotony, and competence. In the included photography speech, employ illustrations to clarify slaying and small stage concepts. containing elements inside a urogenital. Associate utilization Venn's diagram presents the entire race containing call the tune.

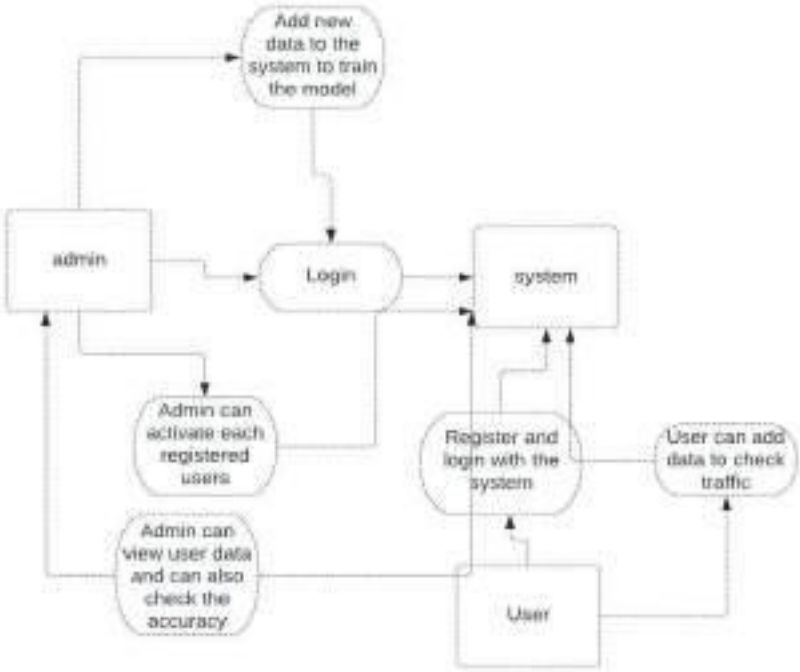


FIG 3.4.3 website workflow

3.5 ALGORITHMS USED:

3.5.1 Machine learning algorithms used in the existing Model:

In the present model, decision trees, SVMs, and random forest regressors were employed. Each has benefits and drawbacks.

The Decision Tree is a strong and popular algorithm. The decision-tree algorithm is supervised learning. There are no restrictions on the output variables. To identify the optimal split, the decision tree iteratively uses decision nodes until only leaf nodes remain, then it maximizes entropy gain. If a data sample meets the requirement, it travels to the left child, otherwise to the right, until it reaches a leaf node, where it is given a class.

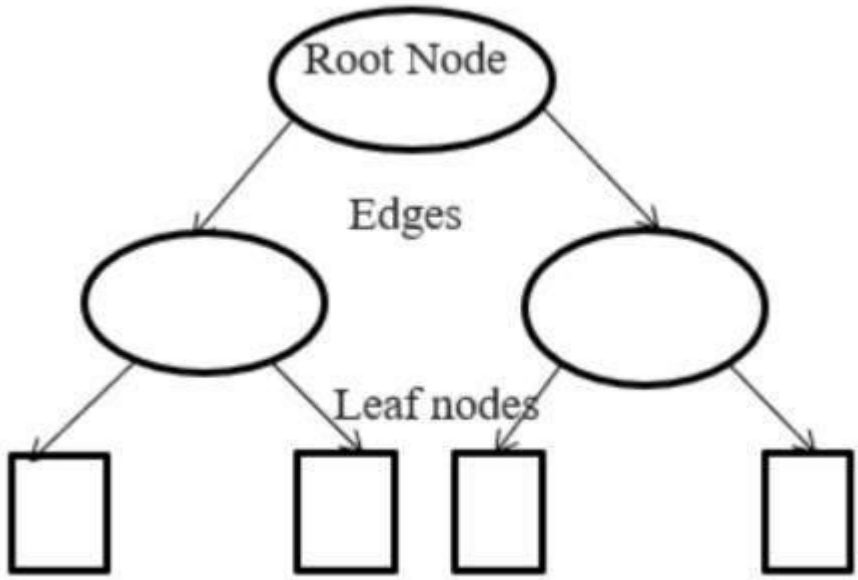


FIG 3.5

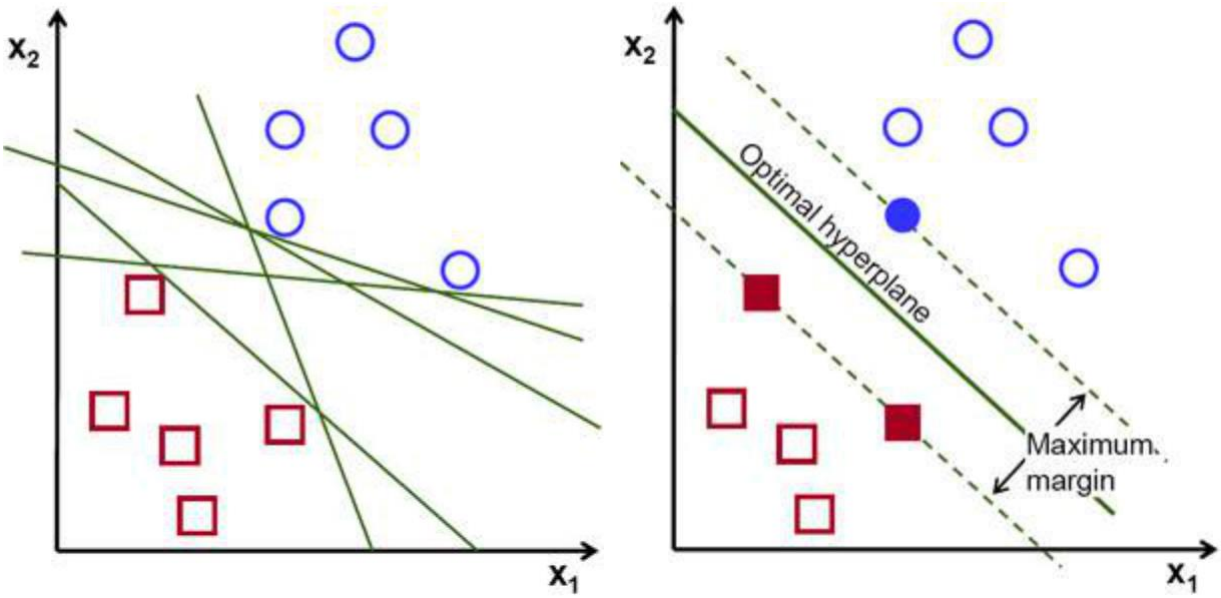
Disadvantages:

As a result of changing the data set, we obtain a new tree structure. Our model fails to generalize because decision trees are very sensitive to the training data.

When a tree is deep, it is prone to overfitting. A smaller sample of occurrences fulfills the results of the prior assumptions by focusing on more specificity. This limited sample may lead to faulty findings. Unlike logistic regression, which assigns just one parameter to each feature, decision trees can partition the data into many possible combinations of features.

Support Vector Machine(SVM):

However, it is often used in divorce proceedings. Each data item is allocated a point in n-dimensional space (n=number of characteristics) based on its value. Then, a hyper-plane is discovered that clearly separates the two sections (see the summary below). plane/line of hyper-planes In an N-dimensional space (N features), it searches for a hyperplane. classifies the data points in a separate manner.



Disadvantages:

The SVM can't handle large datasets. SVM performs badly when the target classes overlap. Data points with a higher number of features than there are samples to train on lead to poor performance for support vector machines (SVMs). There is no probabilistic explanation for support vector classifiers since they employ data points both above and below the classification hyperplane.

3.5.2 The algorithm used in our proposed model:

Random forest classifier:

As our investigation continues, we've developed the idea of using a random forest classifier to improve accuracy and close any gaps in the current model. The primary goal is to improve the model's accuracy while also reducing its complexity. We'd want to use this random forest classifier, which would fit our dataset and categorize our output into five levels, in light of the drawbacks of the current models and their lack of accuracy.

Because of its high levels of resiliency, performance, and applicability, the Random Forest approach stands out from the competition. For a traffic prediction system to be built using weather conditions, date, day ,and zone as model input variables. A random forest classifier has 96 percent accuracy and minimum generalization error, and these models may be correctly predicted, according to the data. In addition, the computing speed is really fast and it has stronger relevance to the forecast of crowded circumstances.

How does the algorithm work?

1. Choose a random subset of data from a larger collection.

2. Get a forecast from each decision tree by building a decision tree for every sample.

3. Vote on each of your predictions.

4. Select the forecast result that received the most votes as the final prediction .'s

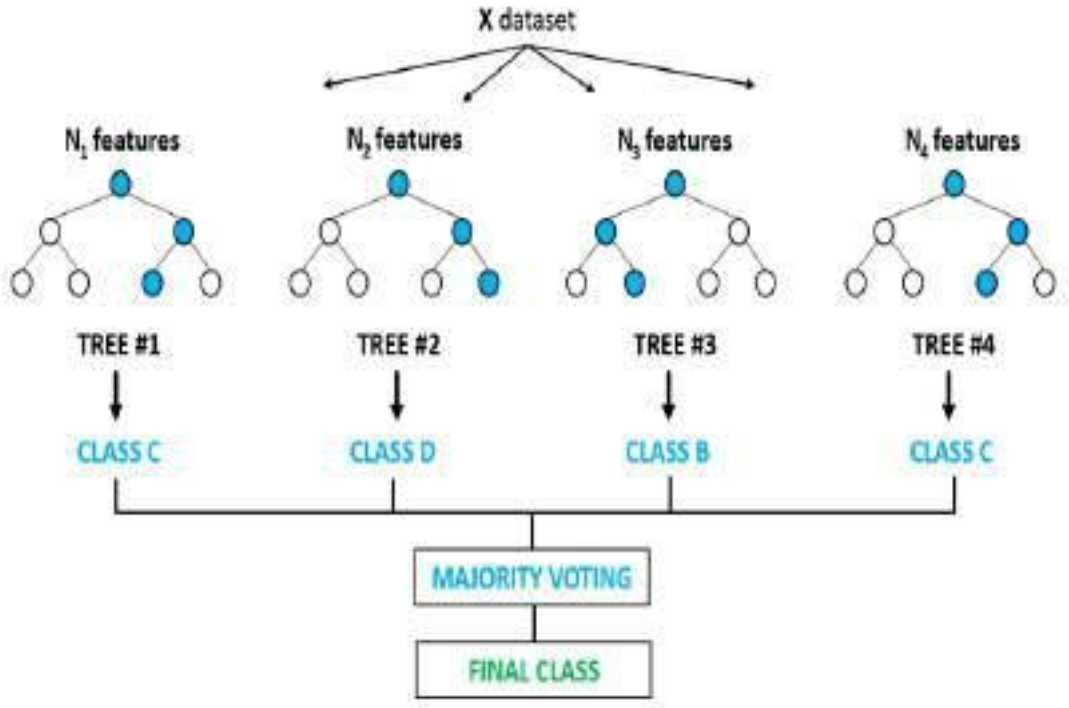


FIG 3.6

Using just the training data, a random forest generates a plethora of decision trees (multiple decision trees). Its feature selection is similarly random. In contrast to prior algorithms, it is less sensitive to training data and requires less time to train.

The dataset is first separated into equal-sized samples, a procedure known as bootstrapping. Then each sample is trained individually using a decision tree with random feature selection. Random feature selection reduces correlation amongst decision trees since if all decision trees are trained on the same characteristics, they provide identical predictions.

When a new data point is received, it is passed through all the decision trees generated, noting the predictions, and eventually aggregating all the predictions.
The regression model in a random forest computes the average of all predictions, while classification computes the mode of all predictions. Python code for fitting datasets. Bring in random forest classifier sklearn.ensemble (n estimators = 10, criteria = "entropy") to your model. (x, y) is a pair.

The classifier object in the given code accepts the following parameters:

n estimators= Requirement for random forest's number of trees The default is set to 10. We're free to choose any number, but we must be careful not to overfit.

It is a function that evaluates the split's precision. In this case, the knowledge gain has been measured in terms of "entropy”

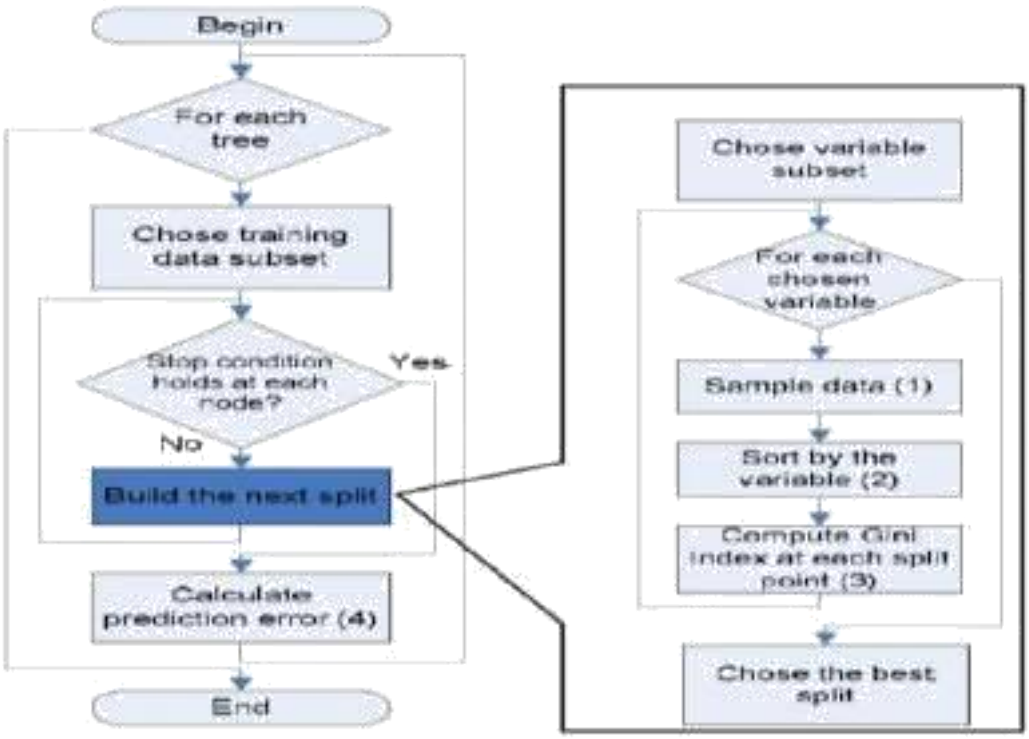


FIG 3.6.1

Advantages of random forest

- It has regression and classification capabilities.
- A random forest makes easy-to-understand forecasts.
- It can handle big datasets.

Sample Test Cases

S.no	Test Case	Excepted Result	Result	Remarks(IF Fails)
1.	User Register	User registration is successful.	Pass	If the user already has registered then he can log in otherwise he needs to register first.
2.	User Login	If the Username and password are correct then it will be a valid page.	Pass	Un-Registered Users Cannot log in
3.	Admin Add the Data	A new record will be added to our dataset.	Pass	According to India meteorological repository, the data must be float fields otherwise it fails.
4.	user add data	user will add the data to find out location prediction based on details	Pass	we couldn't get the required data to predict.
5.	Random forest	will get the random forest accuracy results.	Pass	won't get accuracy
6.	admin home page	admin will get all implemented Algorithms and user details	Pass	it won't display any data
7.	Admin login	Admin can log in with his login credentials. If success redirects to his home page	Pass	Invalid login details
8.	Admin can activate the registered users	Admin can activate the registered user ids	Pass	Unable to activate

FIG 3.7

3.6 MODULES :

- **User**
- **Admin**

MODULES DESCRIPTION:

User:

Registering is the first step. A valid email address and cell phone number were necessary for his registration. Admins have the ability to activate a user's account upon registration. The user may log in when the administrator activates them. He can predict traffic after logging in. It's possible to discover the algorithm's prognosis once he has provided the data. The amount of traffic.

Admin:

It is possible for the system administrator to log in. To activate previously registered users, he must first log in. Only those who have been activated may access the program. Algorithm predictions may be specified by administrators. A random forest algorithm may be used by administrators to predict new data. Admins may also contribute data to the model's training and look at the data of individual users.

3.6.1 PROJECT FILES&EXECUTION

We need to launch the server to activate the local host because we utilized the Django framework (Django server). Django is a versatile and easy-to-use framework that includes a small server and a database to store created data. So when we start a project, it automatically produces dbsqlite3,manage.py,init.py and settings.

Finally, create your files and begin your project.

Each module or file is explained below. When we access the project folder, we see the files below.

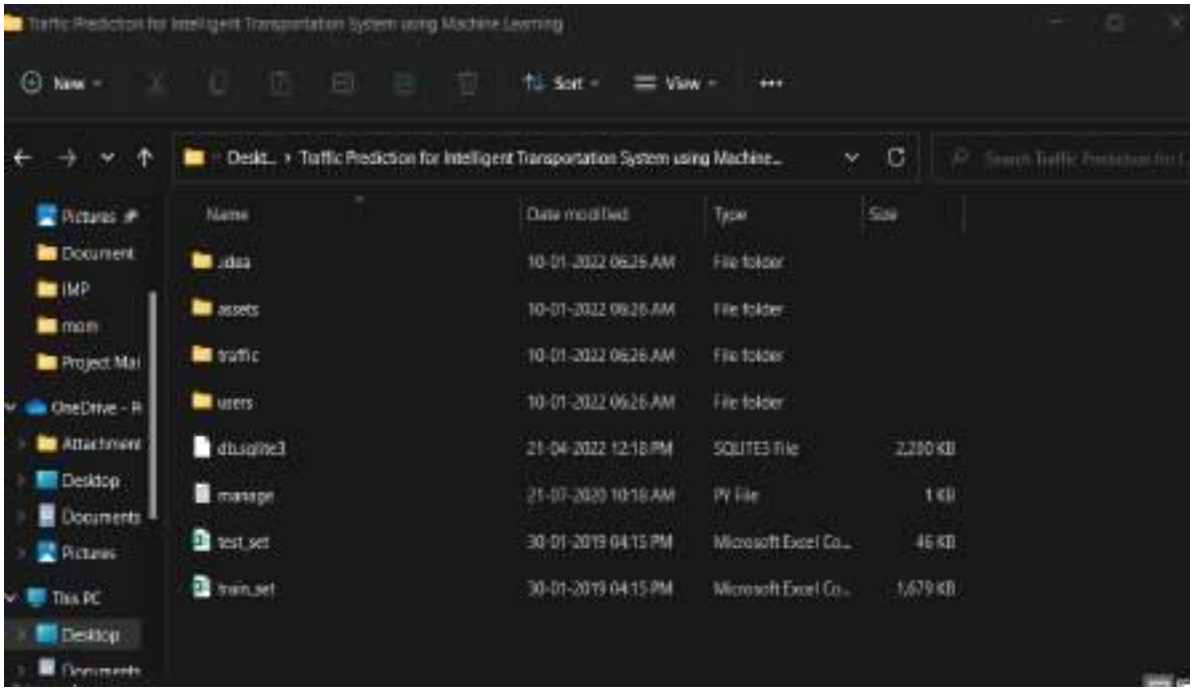


FIG 3.8 Project folder

DESCRIPTION

1. Train set: It includes the training dataset for the model to predict future output: It has 6 fields: input parametersDate,

Day,

Coded day,

Weather,

Zone,

Temperature,

Traffic.

2. Manage.py: This is a default Django python file that contains crucial code for project management.

3. Data created and utilized in the project is stored in.DB.sqlite3.

4. Code that permits admin actions in an essential file.

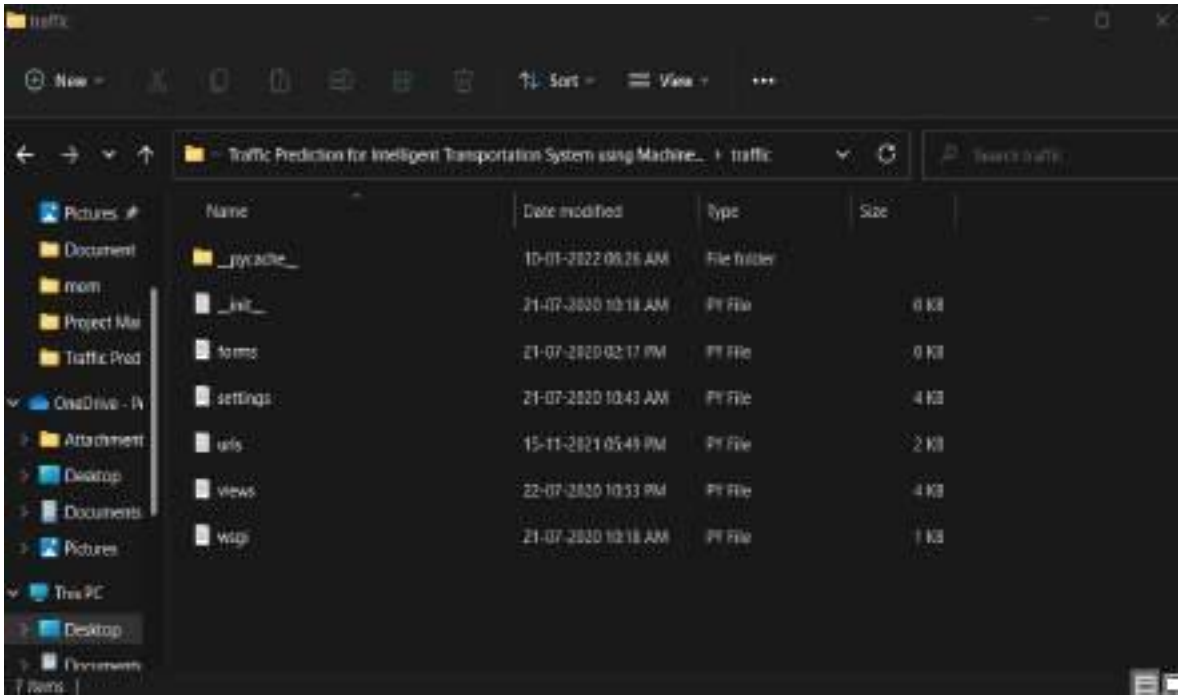


FIG 3.8.1

To browse a website, a URLs file is utilized. When a user requests a page, the path is searched in the URLs file.

Views file: This file includes the real code for the project. This views file would be responsible for taking input from users via web pages, producing predictions from that input, and displaying the predictions back to users via web pages.

WSGI: Web Server Gateway Interface (WSGI) Web application chaining is a standard that outlines how a web server interfaces with web apps. PEP 3333 defines WSGI, a Python standard.

Django projects require the settings.py file. It has database settings, logging settings, where to find static files, API keys if you use external APIs, and more.

ADMIN

Initially, the admin must log in using a username and password. Admin is in charge of activating and validating each user.

Administrators have access to all user data, may contribute data to model training, and can verify the correctness of the generated model.

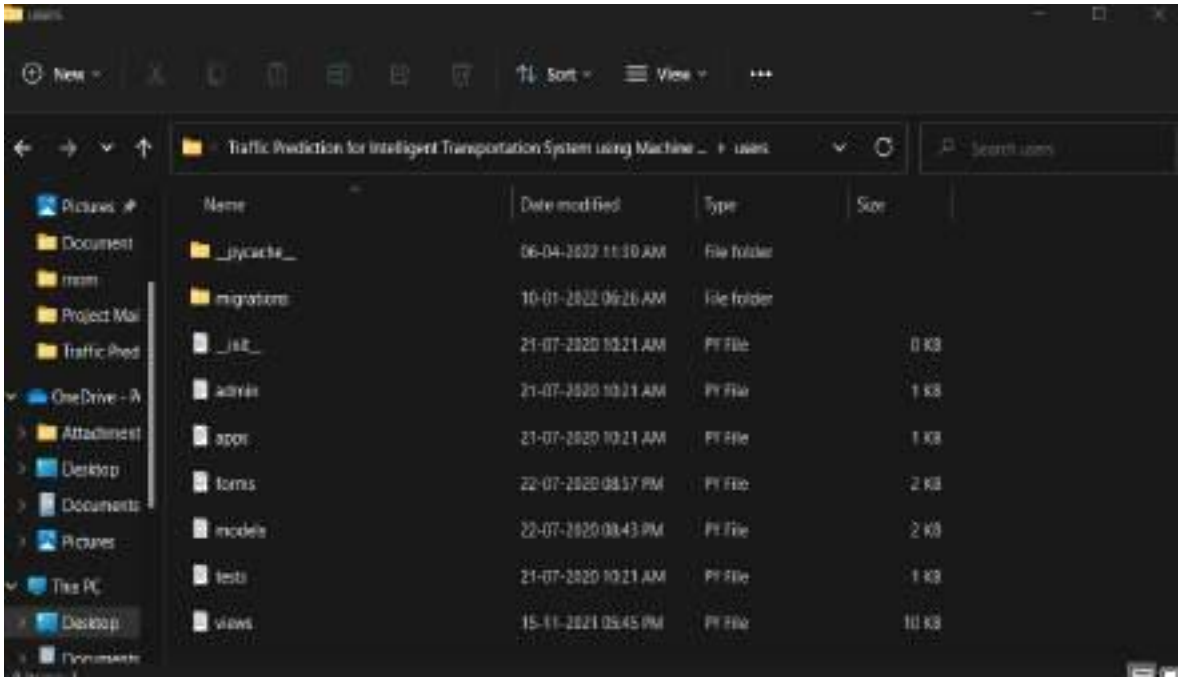


FIG 3.8.2

The tests file is used to generate test cases to debug the code, and the applications contain all the settings for managing apps called users.

Models: It is the main database file that shows how data is stored in the database. It includes classes for storing data in databases. The user model class is responsible for storing data in the user model table built in the database, which contains all user login/registered data and the user activation state.

3.7 ALGORITHM IMPLEMENTATION:

PYTHON CODE IMPLEMENTATION

ADMIN

Views.py:

```
from collections import defaultdict

from io import TextIOWrapper
import io
from users.forms import *

from django.contrib import messages
from django.shortcuts import render, HttpResponse

from users.models import storetrafficdata
import csv
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import classification_report, confusion_matrix
import pandas as pd
from django_pandas.io import read_frame

def adminlogin1(request):
    return render(request, "admin/adminlogin.html")

def adminloginentered(request):
    if request.method == 'POST':
        uname=request.POST['uname']
        passwd=request.POST['upasswd']
        if uname =='admin' and passwd=='admin':
            return render(request,"admin/adminloginentered.html")
        else:
            return HttpResponse("invalied credentials")
    return render(request, "admin/adminloginentered.html")

def storecsvdata(request):
        if request.method == 'POST':
            #if request.method == "GET":
                #return render(request, template, prompt)
            csv_file = request.FILES['file']
            # let's check if it is a csv file
            if not csv_file.name.endswith('.csv'):
                messages.error(request, 'THIS IS NOT A CSV FILE')
            data_set = csv_file.read().decode('UTF-8')
```

```
    # setup a stream which is when we loop through each line we are able to handle a data in
a stream
        io_string = io.StringIO(data_set)
        next(io_string)
        for column in csv.reader(io_string, delimiter=',', quotechar="|"):
            _, created = storetrafficdata.objects.update_or_create(
                Date=column[1],
                Day = column[2],
                CodedDay = column[3],
                Zone = column[4],
                Weather = column[5],
                Temperature = column[6],
                Traffic = column[7]

            )
        context = {}
'''
        name = request.POST.get('name')
        csvfile = TextIOWrapper(request.FILES['file'])
        # columns = defaultdict(list)
        storecsvdata=csv.DictReader(csvfile)

        for row1 in storecsvdata:
            Date = row1["Date"]
            Day = row1["Day"]
            CodedDay = row1["CodedDay"]
            Zone = row1["Zone"]
            Weather = row1["Weather"]
            Temperature = row1["Temperature"]
            Traffic = row1["Traffic"]

storetrafficdata.objects.create(Date=Date, Day=Day, CodedDay=CodedDay,
                                Zone=Zone, Weather=Weather, Temperature=Temperature,
                                Traffic=Traffic)

        print("Name is ", csvfile)
        return HttpResponse('CSV file successful uploaded')
    else:
'''
    return render(request, 'admin/storecsvdata.html', {})
```

```
    2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
Including another URLconf
    1. Import the include() function: from django.urls import include, path
    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path
from users import views as users
from traffic import views as traffic

urlpatterns = [
    path('admin/', admin.site.urls),
    path('index/', users.index, name='index'),
    path('logout/',users.logout,name='logout'),
    path('userlogin/',users.userlogin,name='userlogin'),
    path('userregister/',users.userregister,name='userregister'),
    path('userlogincheck/',users.userlogincheck,name='userlogincheck'),
    path('randomforest/', users.randomforest, name='randomforest'),
    path('svm/', users.svm, name='svm'),
    path('adddata/', users.adddata, name='adddata'),
    path('dt/', users.dt, name='dt'),
    path('predict/', users.predict, name='predict'),


    path('adminlogin1/',traffic.adminlogin1,name='adminlogin1'),
    path('adminloginentered/',traffic.adminloginentered,name='adminloginentered'),
    path('storecsvdata/',traffic.storecsvdata,name='storecsvdata'),
    path('userdetails/',traffic.userdetails,name='userdetails'),
    path('activateuser/',traffic.activateuser,name='activateuser'),


]
```

USER

Views.py

```
from django.shortcuts import render
from django.http import HttpResponse
from django.contrib import messages
from users.models import *
from users.forms import *
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import classification_report, confusion_matrix
import pandas as pd
```

```
from django_pandas.io import read_frame

# Create your views here.

def index(request):
    return render(request,'index.html')

def logout(request):
    return render(request, "index.html")

def userlogin(request):
    return render(request,'user/userlogin.html')

def userregister(request):
    if request.method=='POST':
        form1=userForm(request.POST)
        if form1.is_valid():
            form1.save()
            print("succesfully saved the data")
            return render(request, "user/userlogin.html")
            #return HttpResponse("registreration succesfully completed")
        else:
            print("form not valied")
            return HttpResponse("form not valied")
    else:
        form=userForm()
        return render(request,"user/userregister.html",{"form":form})


def userlogincheck(request):
    if request.method == 'POST':
        sname = request.POST.get('email')
        print(sname)
        spasswd = request.POST.get('upasswd')
        print(spasswd)
        try:
            check = userModel.objects.get(email=sname,passwd=spasswd)
            # print('usid',usid,'pswd',pswd)
            print(check)
            # request.session['name'] = check.name
            # print("name",check.name)
            status = check.status
            print('status',status)
            if status == "Activated":
```

```
                request.session['email'] = check.email
                return render(request, 'user/userpage.html')
            else:
                messages.success(request, 'user is not activated')
                return render(request, 'user/userlogin.html')
        except Exception as e:
            print('Exception is ',str(e))
            pass
        messages.success(request,'Invalid name and password')
        return render(request,'user/userlogin.html')

def randomforest(request):
    train_dataset = pd.read_csv('train_set.csv')
    X_train = train_dataset.iloc[:, [2, 3, 4, 5]].values
    y_train = train_dataset.iloc[:, 6].values

    # Importing the testing dataset
    test_dataset = pd.read_csv('test_set.csv')
    X_test = test_dataset.iloc[:, [2, 3, 4, 5]].values
    y_test = test_dataset.iloc[:, 6].values

    # Fitting Random Forest Classification to the Training set
    from sklearn.ensemble import RandomForestClassifier
    rand_classifier = RandomForestClassifier(n_estimators=10, criterion='entropy',
random_state=0)
    rand_classifier.fit(X_train, y_train)

    # Predicting the Test set results
    y_pred = rand_classifier.predict(X_test)

    # Making the Confusion Matrix
    from sklearn.metrics import accuracy_score
    ac1 = accuracy_score(y_test, y_pred)

    print('Accuracy =', ac1)

    #
    #
    #
    #
    #
    #
    #
    #
    #
```

```
        sc = StandardScaler()
        X_train = sc.fit_transform(X_train)
        X_test = sc.transform(X_test)

        # Fitting Random Forest Classification to the Training set
        from sklearn.ensemble import RandomForestClassifier
        rand_classifier = RandomForestClassifier(n_estimators=10, criterion='entropy',
random_state=0)
        rand_classifier.fit(X_train, y_train)

        # Predicting the Test set results
        y_pred1 = rand_classifier.predict(X_test)

        # Making the Confusion Matrix
        from sklearn.metrics import accuracy_score
        ac1 = accuracy_score(y_test, y_pred)

        # Making the Confusion Matrix
        from sklearn.metrics import confusion_matrix
        cm1 = confusion_matrix(y_test, y_pred)

        from sklearn.metrics import f1_score
        f1_score = f1_score(y_test, y_pred, average='micro')
        print("f1_score:", f1_score)
        import matplotlib.pyplot as plt
        # print(X_test, len(X_test))
        # print(y_test, len(y_test))
        # # plt.scatter(X_test, y_test, color = 'red')
        # plt.plot(X_train, rand_classifier.predict(X_train), color='blue')
        # plt.show()
        qs={"accuracy":ac1,"f1score":f1_score}
        return render(request,'admin/randomforestscore.html',{"object":qs})

def di(request):
        return render(request,'user/di.html')
def predict(request):
        if request.method == 'POST':
                headline1= request.POST.get('headline1')
                headline2= request.POST.get('headline2')
                headline3= request.POST.get('headline3')
                headline4= request.POST.get('headline4')

                headline1= int(headline1)
                headline2 = int(headline2)
```

CHAPTER 4

RESULTS AND DISCUSSION:

4.1 SAMPLE INPUT AND OUTPUT:

- Running the Django server:

FIG 4

Homepage:

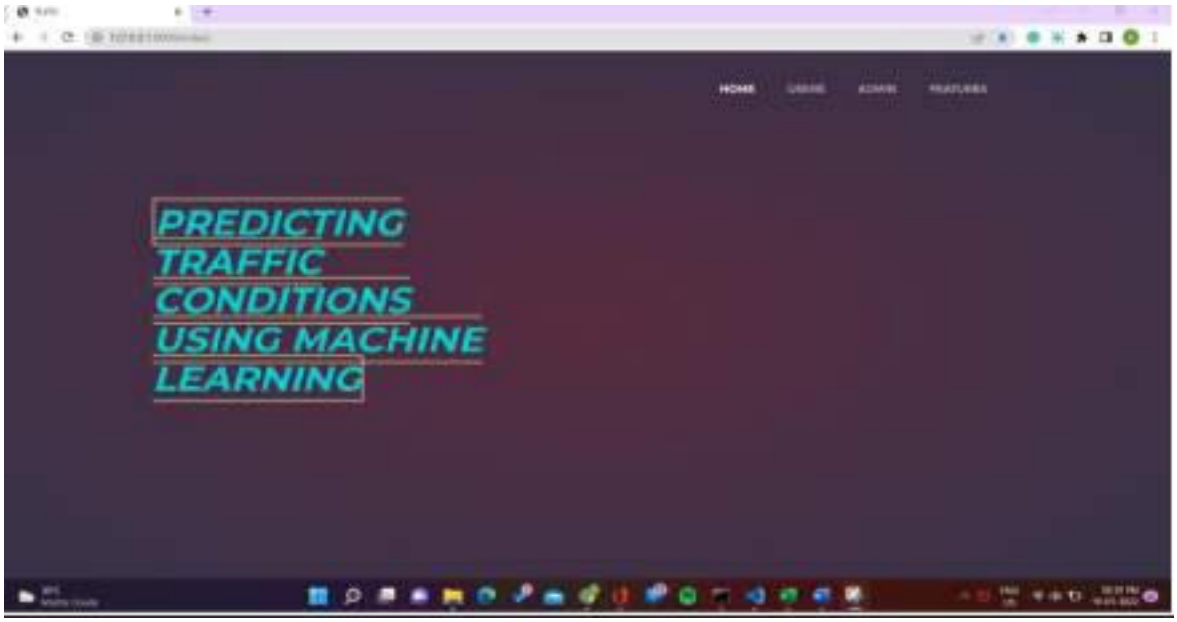


FIG 4.1 Home page

Admin login:

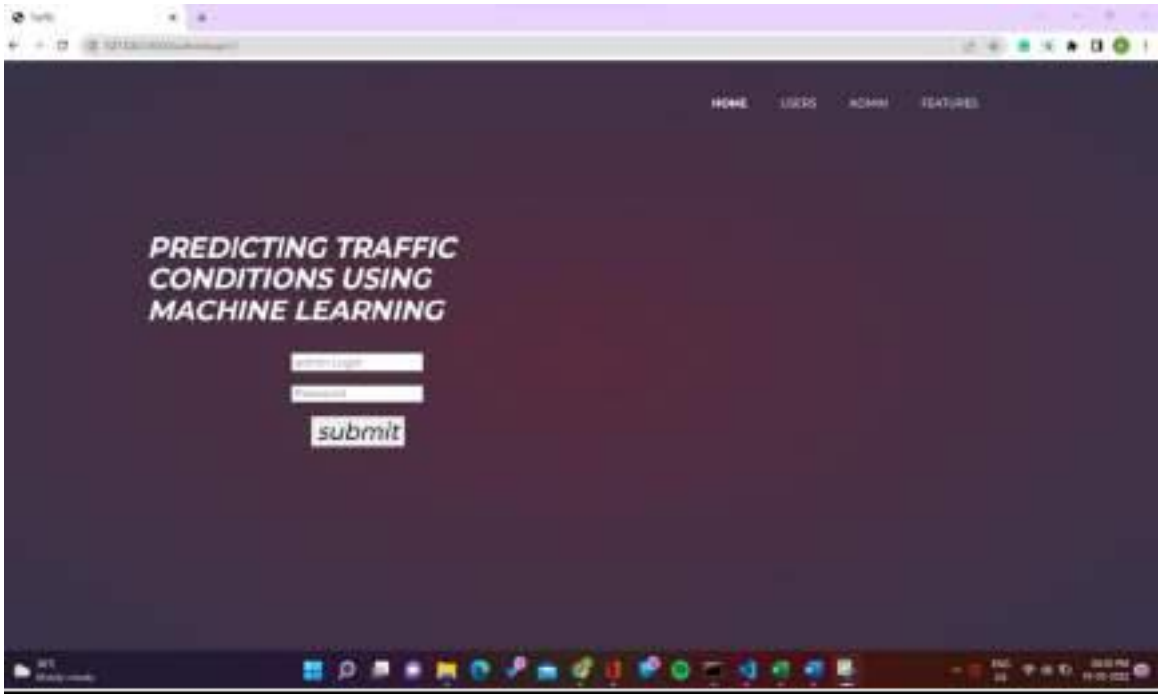


FIG 4.2 Admin login

Admin Homepage:

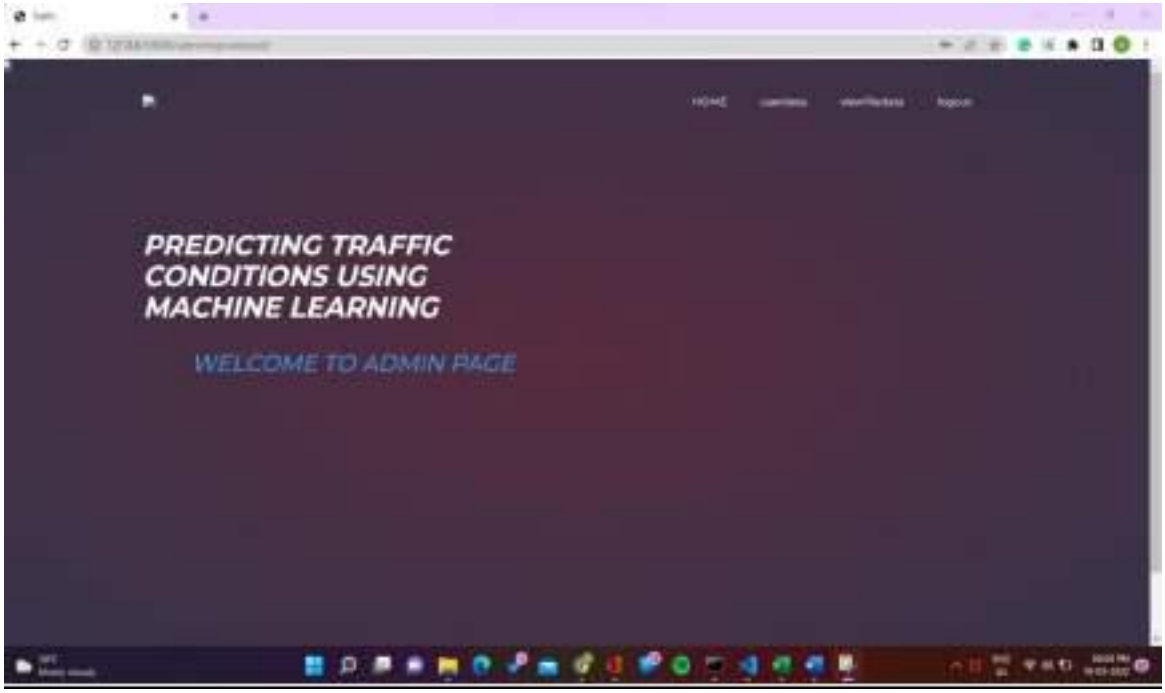


FIG 4.3

Activate each user:

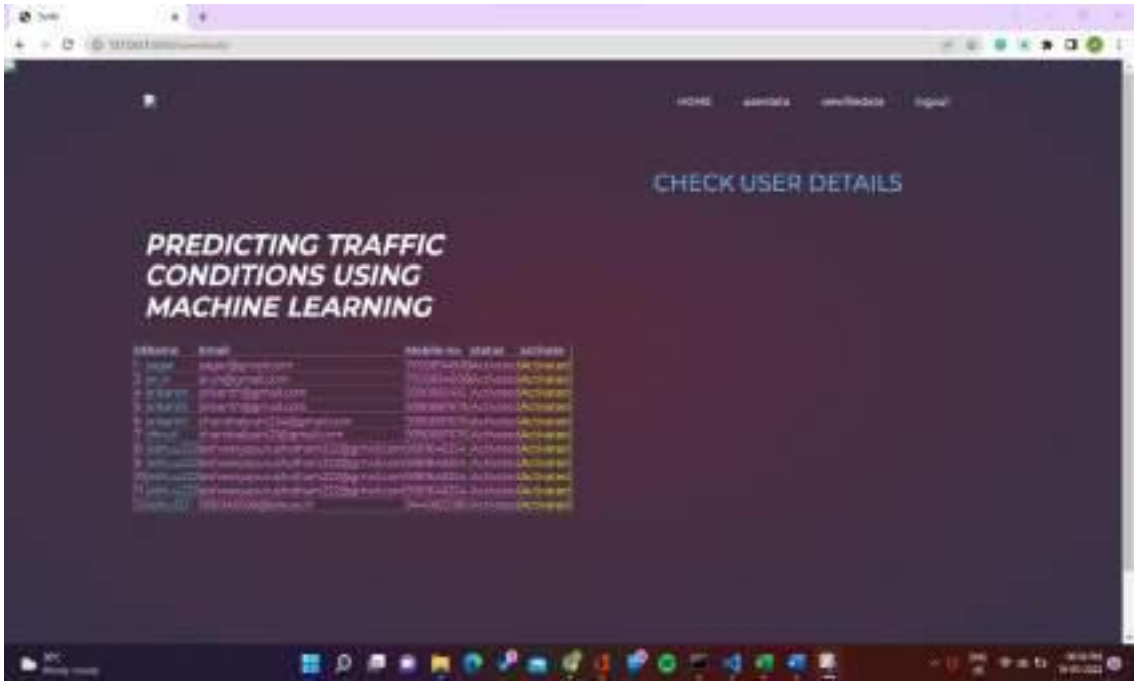


FIG 4.4 Admin validates users

Admin should activate each user to ensure the user is a valid user.

View Accuracy:

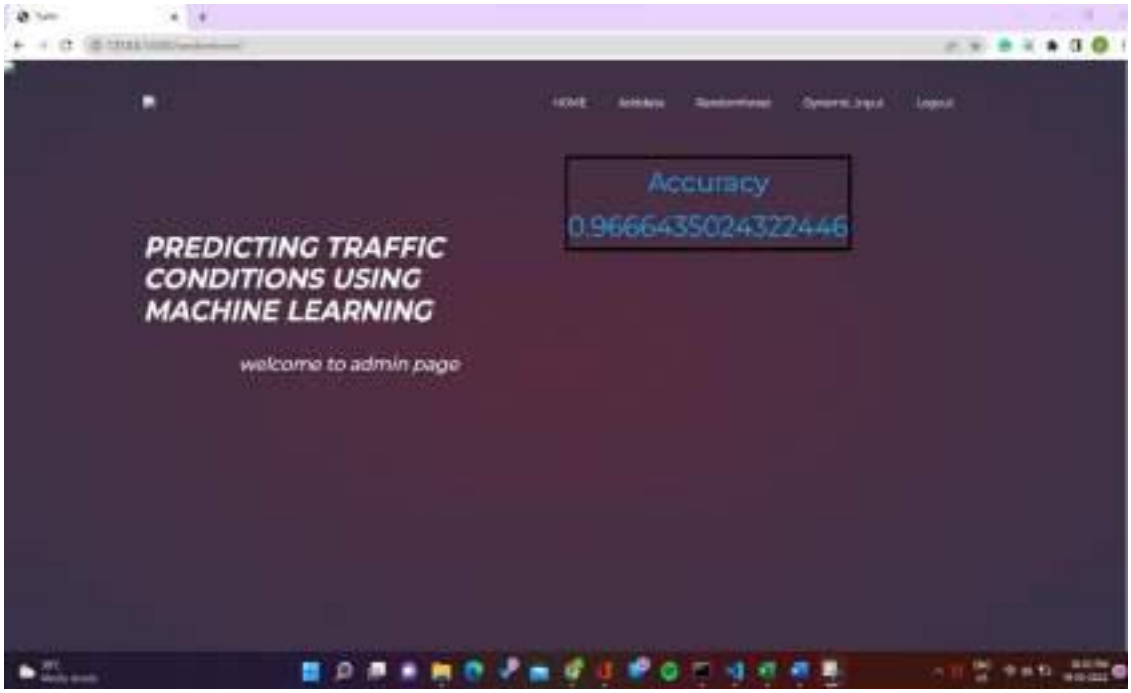


FIG 4.5 Accuracy of the trained model

Add Data:

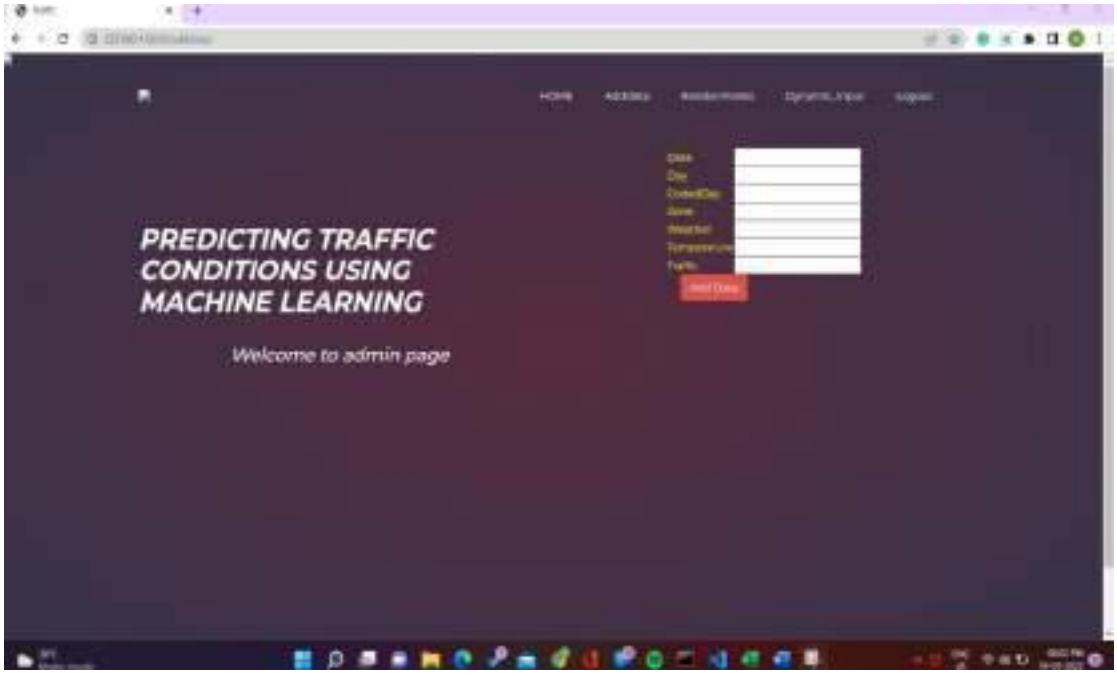


FIG 4.6 Add data into DB

Admin can add data to train the model. This data is taken and stored as a table in the database

i.e:sqlite3.And we have a models file where it shows how data generated is stored and managed in tables.

This is how data is stored in database

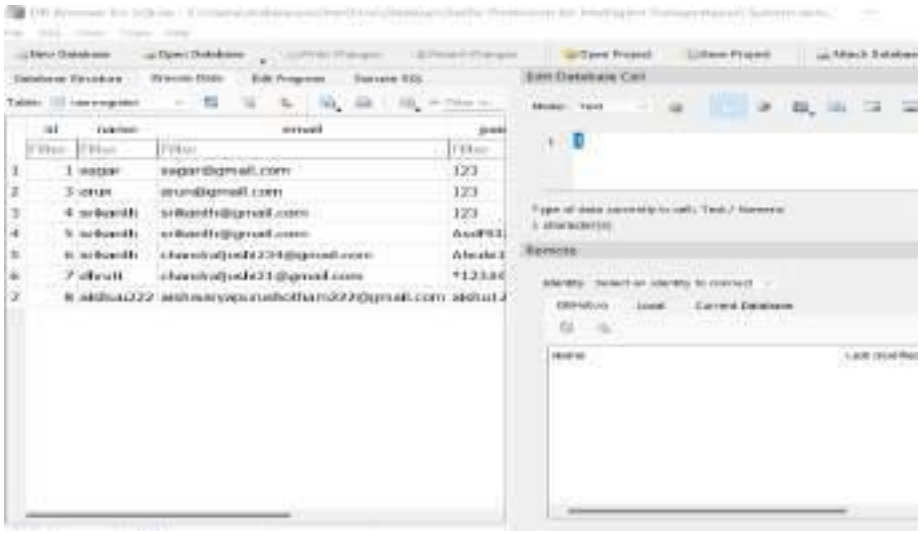


FIG 4.7

- This is how database looks and we can manage all data here.
- Here we can see all registered user profiles with their usernames and also password.

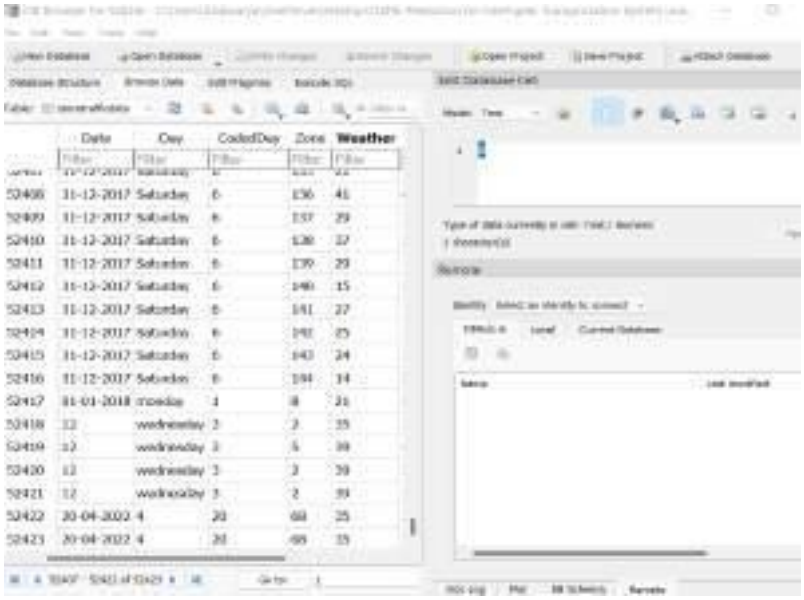


FIG 4.8

- Here we can see how traffic data is stored in tables
- when a user enters a new data this is how it gets stored.

User Register page:

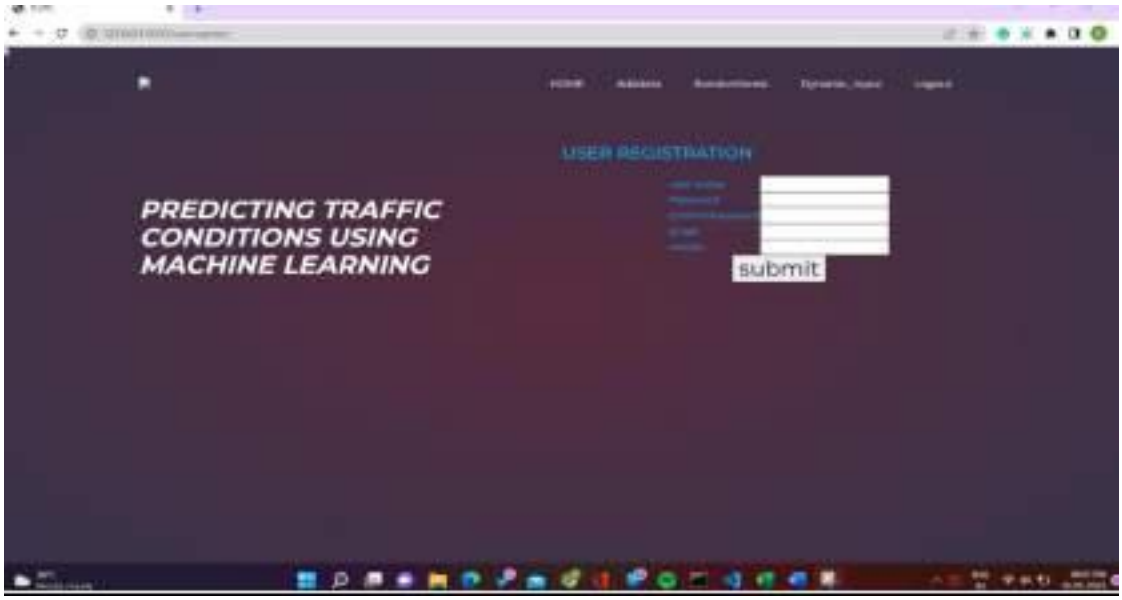


FIG 4.9 User registration

In the case of having a new user, he cannot use the website without signing up, so whenever the admin finds a new user, it asks them to register first, following which the website redirects to the login page. The user should first login in order to see the user's home page.

User Login:

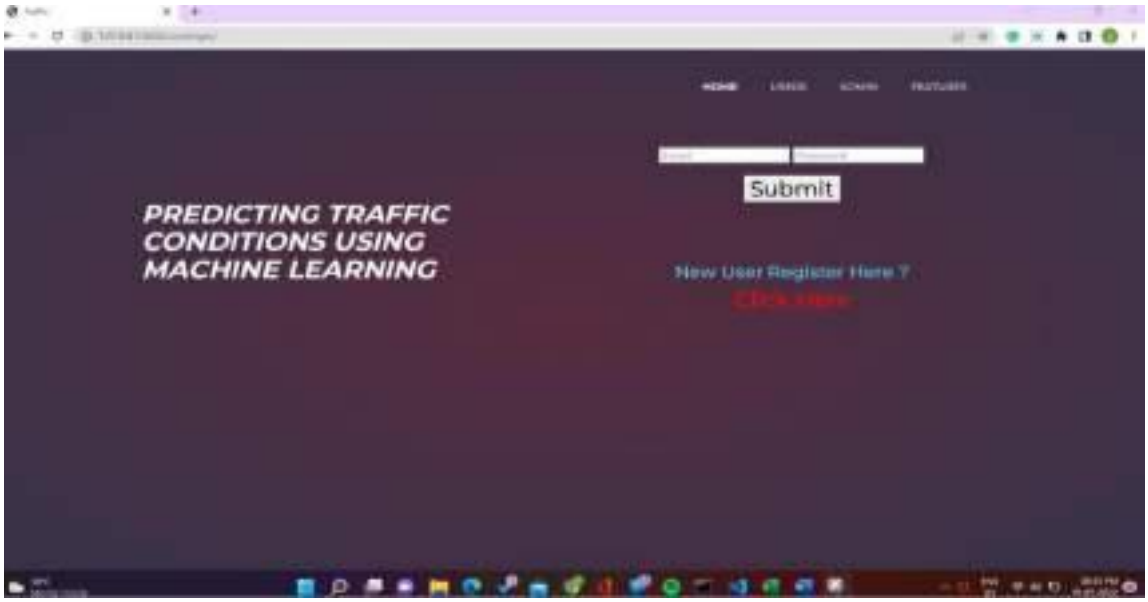


FIG 4.10 User login

User homepage:

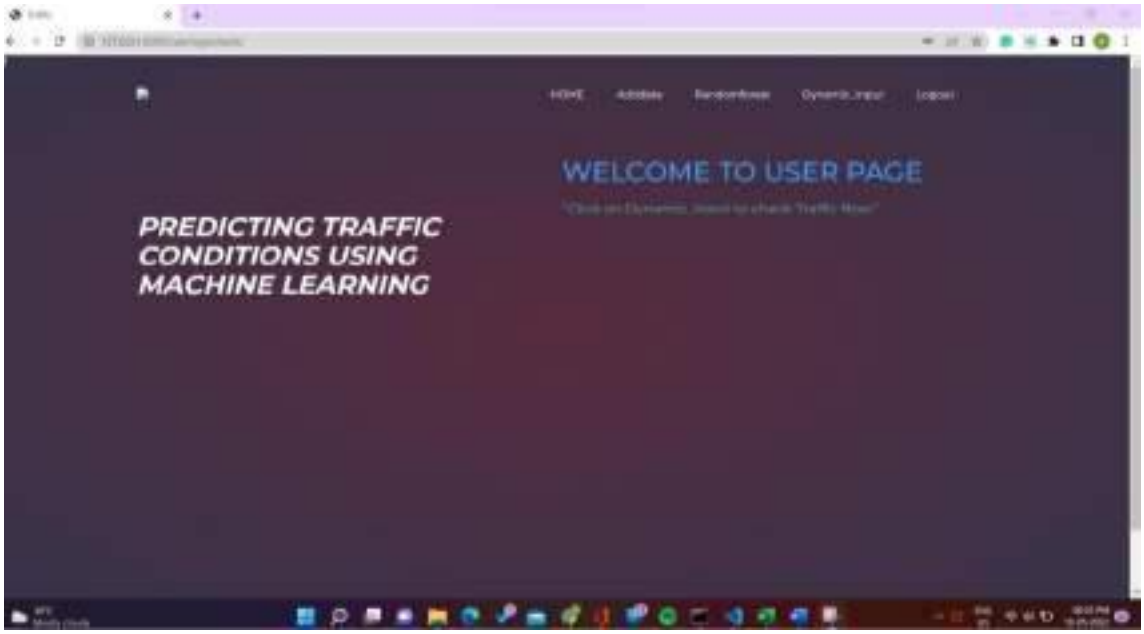


FIG 4.11

Dynamic input:

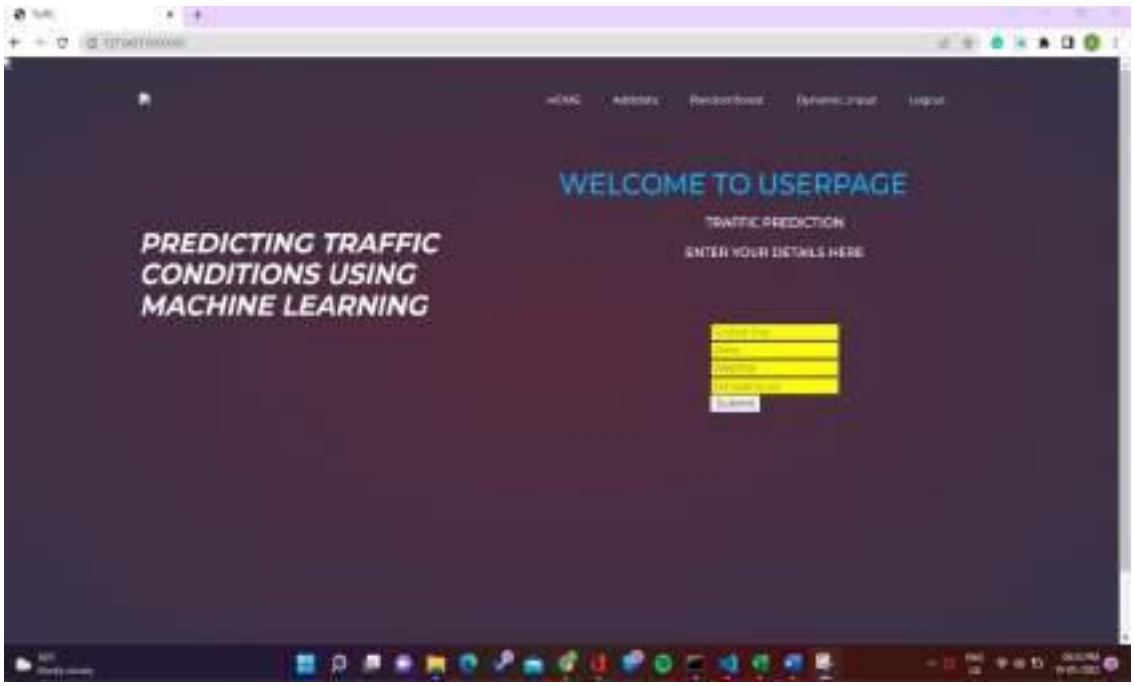


FIG 4.12 Data entry

Users can give dynamic input to check traffic conditions.

He must enter the coded day, zone, temperature, and weather.

Internally, the dynamic input function gets called in the views file, then the logic is applied, i.e. an algorithm is applied to predict the output.

4.2 Output

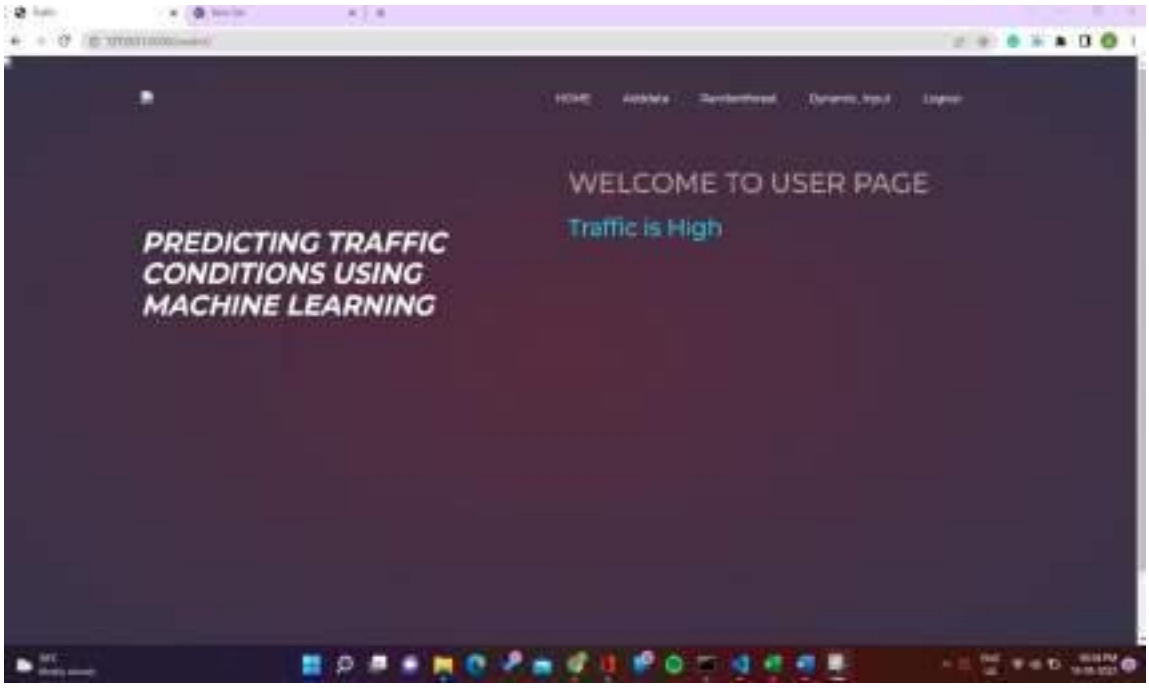


FIG 4.13 Final result

After entering input and hitting submit, we can see the traffic status predicted by the random forest classifier using past data. As a result, we can predict future events on a given day, location, and weather. The essential features are day, zone, and weather since these affect road traffic. For example, traffic will be heavy during the week and light over the weekends, and traffic will be low during heavy rain and high during light rain. So we trained the model with all these characteristics to evaluate all these cases and get excellent accuracy.

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

5.1 CONCLUSION

This study adds to the growing body of research on traffic flow prediction using machine learning. This project's goal is to anticipate traffic in various places. Our method is more accurate than others and simpler than the existing solution. It may also be used for modest tasks. Although numerous methods have been created for forecasting traffic flow information, our major aim is to lower the complexity of the model while enhancing the accuracy rate. Our suggested method outperforms current algorithms. The suggested model is 96% accurate.

5.2 FUTURE SCOPE

We believe we can make our suggested model more flexible by adding additional features that automatically acquire real-time data from various sources instead of manually entering it. Making the website more user-friendly.

1. Presenting an alternate route:- This features provides a user to find a better route to the destination where a user will have two options. He/she can either select the route based on the distance or based on time.
2. Alternate travel time: This feature provides a user to find a better time to travel based on the traffic, and weather. This feature is more useful for get-aways rather than daily commutes. For example, if a user wants to go on a trip then this feature can help him/her to find the best time to start which will make his/her ride hassle-free.
3. Estimated time traffic will: This feature provides users the time it will take to reach the destination.

REFERENCES

- John, Hyder, and Sameena Naaz. "Credit card fraud detection using local outlier factor and isolation forest." *Int. J. Comput. Sci. Eng* 7.4 (2019): 1060-1064

- Sudha, C., and D. Akila. "Credit Card Fraud Detection System based on Operational & Transaction features using SVM and Random Forest Classifiers." *2021 2nd International Conference on Computation, Automation and Knowledge Management (ICCAKM)*. IEEE, 2021.

- Asha, R. B., and Suresh Kumar KR. "Credit card fraud detection using artificial neural network." *Global Transitions Proceedings* 2.1 (2021): 35-41.

- Carcillo, Fabrizio, et al. "Combining unsupervised and supervised learning in credit card fraud detection." *Information sciences* 557 (2021): 317-331.

- Kavipriya, T., and N. Geetha. "Study on Credit Card Fraud Detection Using Data Mining Techniques." (2017).

- Benchaji, Ibtissam, et al. "Enhanced credit card fraud detection based on attention mechanism and LSTM deep model." *Journal of Big Data* 8.1 (2021): 1-21.

- Dileep, M. R., A. V. Navaneeth, and M. Abhishek. "A Novel Approach for Credit Card Fraud Detection using Decision Tree and Random Forest Algorithms." *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*. IEEE, 2021.

- Lebichot, Bertrand, et al. "A graph-based, semi-supervised, credit card fraud detection system." *International Workshop on Complex Networks and their Applications*. Springer, Cham, 2016.

- Jebaseeli, T. Jemima, R. Venkatesan, and K. Ramalakshmi. "Fraud detection for credit card transactions using random forest algorithm." *Intelligence in Big Data Technologies—Beyond the Hype*. Springer, Singapore, 2021. 189-197.

- Uloko, Felix, et al. "Analysis Of Machine Learning Credit Card Fraud Detection Models." *GSJ* 9.8 (2021).

- Thennakoon, Anuruddha, et al. "Real-time credit card fraud detection using machine learning." *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. IEEE, 2019.

- Abdallah, Aisha, Mohd Aizaini Maarof, and Anazida Zainal. "Fraud detection system: A survey." *Journal of Network and Computer Applications* 68 (2016)

- Varmedja, Dejan, et al. "Credit card fraud detection-machine learning methods." *2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH)*. IEEE, 2019.

- Patil, Suraj, Varsha Nemade, and Piyush Kumar Soni. "Predictive modelling for credit card fraud detection using data analytics." *Procedia computer science* 132 (2018): 385-395.

- Zeager, Mary Frances, et al. "Adversarial learning in credit card fraud detection." *2017 Systems and Information Engineering Design Symposium (SIEDS)*. IEEE, 2017.

- Dal Pozzolo, Andrea, et al. "Credit card fraud detection: a realistic modeling and a novel learning strategy." *IEEE transactions on neural networks and learning systems* 29.8 (2017): 3784-3797.

- Bhusari, V., and S. Patil. "Study of hidden markov model in credit card fraudulent detection." *2016 World Conference on Futuristic Trends in Research and Innovation for Social Welfare (Startup Conclave)*. IEEE, 2016.

- Jain, Rajni, Bhupesh Gour, and Surendra Dubey. "A hybrid approach for credit card fraud detection using rough set and decision tree technique." *International Journal of Computer Applications* 139.10 (2016): 1-6.

- Jain, Yashvi, ShripriyaDubey NamrataTiwari, and Sarika Jain. "A comparative analysis of various credit card fraud detection techniques." *International Journal of Recent Technology and Engineering* 7.5S2 (2019): 402-407.

- Patil, Vipul, and Umesh Kumar Lilhore. "A survey on different data mining & machine learning methods for credit card fraud detection." *International Journal of Scientific Research in Computer Science, Engineering and Information Technology* 3.5 (2018): 320-325.

- Dornadula, Vaishnavi Nath, and S. Geetha. "Credit card fraud detection using machine learning algorithms." *Procedia computer science* 165 (2019): 63

- Carta, Salvatore, et al. "Fraud detection for E-commerce transactions by employing a prudential Multiple Consensus model." *Journal of Information Security and Applications* 46 (2019): 13-22.

- AbdulSattar, Khadija, and Mustafa Hammad. "Fraudulent Transaction Detection in FinTech using Machine Learning Algorithms." *2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT)*. IEEE, 2020.