Predicting Traffic Volume Using Machine Learning

1. INTRODUCTION

1.1 OVERVIEW

Machine learning can be used to predict traffic volume by analyzing data from various sources such as traffic sensors, weather data, and historical traffic patterns. By using supervised learning algorithms such as regression and decision trees, the model can be trained to accurately predict traffic volume. The model can then be used to make predictions about future traffic volume based on current and past data. Additionally, unsupervised learning algorithms such as clustering can be used to identify patterns in the data that can be used to make more accurate predictions.

Growth in the number of vehicles and degree of urbanization means that the annual cost of traffic jams is increasing in cities. This leads to a decrease in the quality of life among citizens through a considerable waste of time and excessive fuel consumption and air pollution in congested areasTraffic congestion has been one of the major issues that most metropolises are facing despite measures being taken to mitigate and reduce it. The safe and time-efficient movement of the people and goods is dependent on Traffic flow, which is directly connected to the traffic characteristics. Early analysis of congestion events and prediction of traffic volumes is a crucial step to identify traffic bottlenecks, which can be utilized to assist traffic management centres.

1.2 PURPOSE

The purpose of Predicting Traffic Volume is to help transportation planners anticipate traffic patterns and plan accordingly. It can also help drivers plan their routes and adjust their driving habits to avoid congested areas. Additionally, it can help businesses plan their staffing needs and adjust their operations to accommodate changes in traffic volume. Finally, it can help cities and towns better manage their resources and plan for future growth.

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

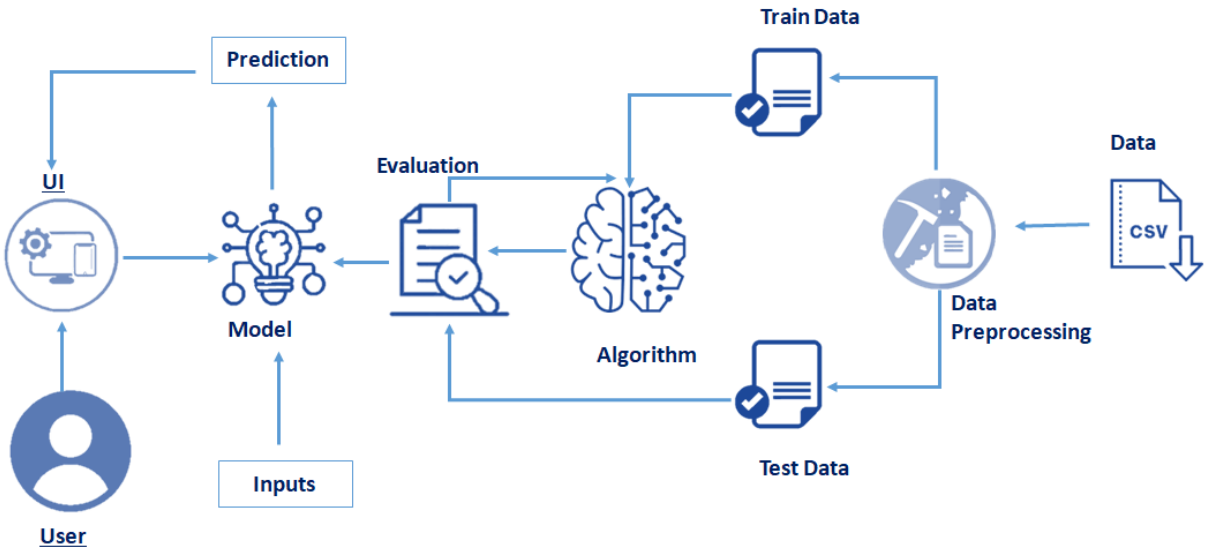
Traffic volume can be measured in a variety of ways, including manual counts, automated traffic counters, and aerial surveys. Manual counts involve counting the number of vehicles passing a certain point over a given period of time. Automated traffic counters use sensors to detect and count vehicles as they pass. Aerial surveys involve taking aerial photographs of roads and then counting the number of vehicles in the photographs.

2.2 PROPOSED SYSTEM

Proposed systems of traffic volume prediction typically use a combination of historical data, machine learning algorithms, and real-time data to predict future traffic volumes. Historical data is used to create a baseline for traffic patterns, while machine learning algorithms are used to identify patterns in the data and make predictions. Real-time data is used to adjust the predictions based on current conditions. Additionally, some systems use sensors to measure traffic flow and provide more accurate predictions.

3. THEORETICAL ANALYSIS

3.1 BLOCK DIAGRAM



3.2 HARDWARE AND SOFTWARE DESIGNING

Python

Python is an interpreted, object-oriented, high-level programming language withdynamic semantics. It was created by Guido van Rossum , and first releasedonFebruary 20, 1991. Its high-level built in data structures, combined with dynamictyping and dynamic binding , make it very attractive for Rapid ApplicationDevelopment, as well as for use as a scripting or glue language to connect existingcomponents together. Python's simple, easy to learn syntax emphasizes readabilityand therefore reduces the cost of program maintenance. Python supports modules andpackages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary formwithout charge for all major platforms, and can be freely distributed.

Anaconda Navigator

Anaconda Navigator is a free and open-source distribution of the Python andRprogramming languages for data science and machine learning related applications. Itcan be installed on Windows, Linux, and macOS.Conda is an open-source, crossplatform, package management system. Anaconda comes with so very nice toolslike JupyterLab, Jupyter Notebook, QtConsole, Spyder, Glueviz, Orange, Rstudio, Visual Studio Code. For this project, we will be using Jupyter notebook and Spyder.

Jupyter Notebook

The Jupyter Notebook is an open source web application that you can use to createand share documents that contain live code, equations, visualizations, and text. Jupyter Notebook is maintained by the people at Project Jupyter. Jupyter Notebooksare a spin-off project from the IPython project, which used to have an IPythonNotebook project itself. The name, Jupyter, comes from the core supportedprogramming languages that it supports: Julia, Python, and R. Jupyter ships withtheIPython kernel, which allows you to write your programs in Python, but there arecurrently over 100 other kernels that you can also use.

Spyder

Spyder, the Scientific Python Development Environment, is a free integrateddevelopment environment (IDE) that is included with Anaconda. It includes editing, interactive testing, debugging, and introspection features. Initially created anddeveloped by Pierre Raybaut in 2009, since 2012 Spyder has been maintainedandcontinuously improved by a team of scientific Python developers andthecommunity.Spyder is extensible with first-party and third party plugins includes support for interactive tools for data inspection and embeds Pythonspecific code. Spyder is also pre-installed in Anaconda Navigator, which is included in Anaconda.

Flask

Webframework used for building. It is a web application framework written in pythonwhich will be running in local browser with a user interface. In this application, whenever the user interacts with UI and selects emoji, it will suggest the best andtopmovies of that genre to the user. Hardware Requirements: Operating system: window 7 and above with 64bit

Processor Type -Intel

Core i3-3220

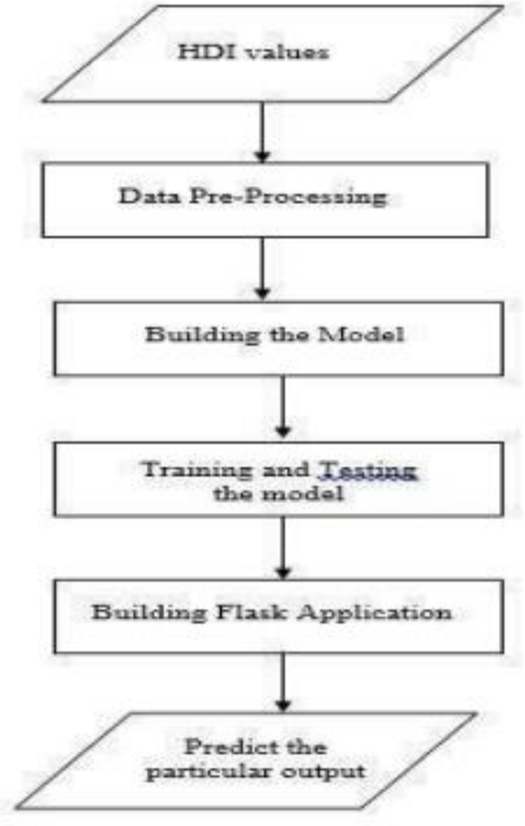
RAM: 4Gb and above

Hard disk: min 100GB

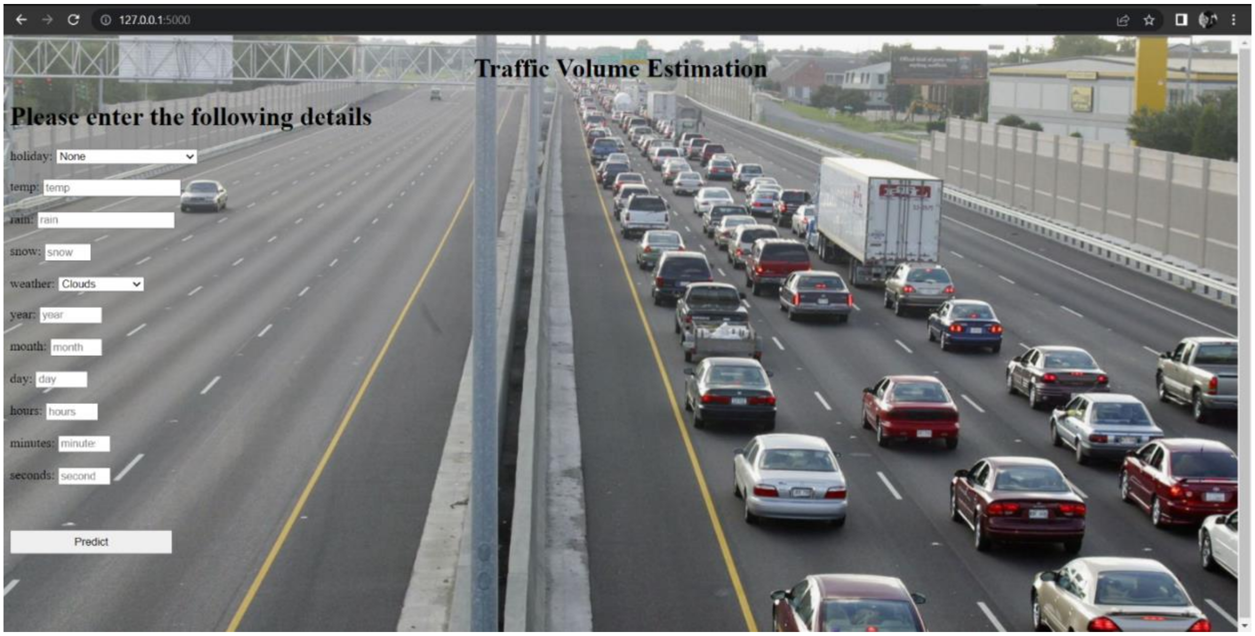
4. EXPERIMENTAL INVESTIGATION

The text data need to be organized before proceeding with the project. The original dataset has a single folder. We will be using the traffic volume.csv file to fetch the text data of training data. The datas need to be unique and all fields need to be filled. The data set images are to be pre-processed before giving to the model. We will create a function that uses the pre-trained model for predicting custom outputs. Then we have to test and train the model. After the model is build, we will be integrating it to a web application

5. FLOW CHART


HDI values
Data Pre-Processing
Building the Model
Training and Testing the model
Building Flask Application
Predict the particular output

6. RESULT



Traffic Volume Estimation

Please enter the following details

holiday: None

temp: temp

rain: rain

snow: snow

weather: Clouds

year: year

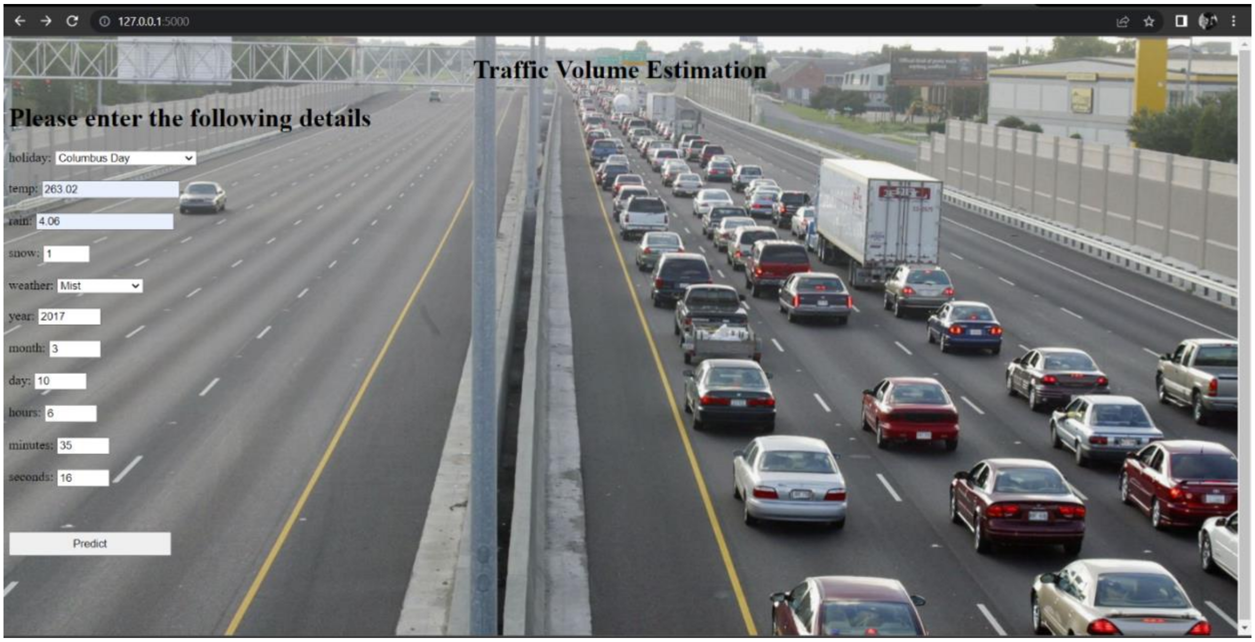
month: month

day: day

hours: hours

minutes: minute

seconds: second



Traffic Volume Estimation

Please enter the following details

holiday: Columbus Day

temp: 263.02

rain: 4.06

snow: 1

weather: Mist

year: 2017

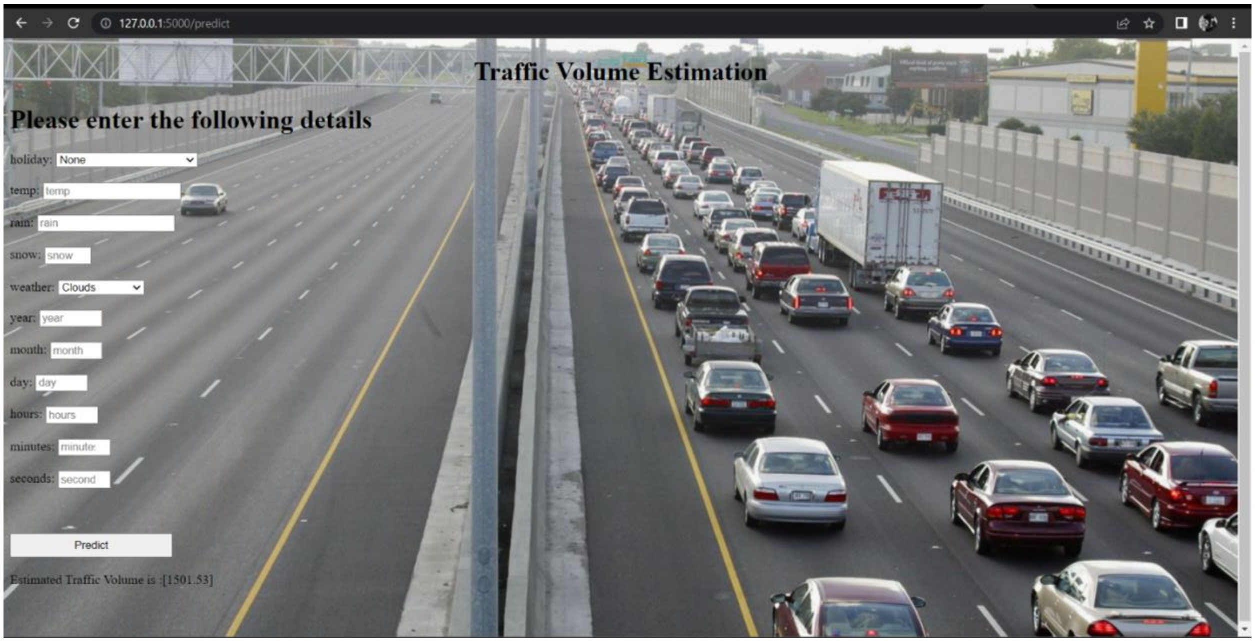
month: 3

day: 10

hours: 6

minutes: 35

seconds: 16



Traffic Volume Estimation

Please enter the following details

holiday: None
temp: temp
rain: rain
snow: snow
weather: Clouds
year: year
month: month
day: day
hours: hours
minutes: minute
seconds: second

Estimated Traffic Volume is :[1501.53]

7. ADVANTAGES

- Easy to use
- Cost efficient
- Time efficie

8. CONCLUSION

In conclusion, traffic volume prediction is an important tool for transportation planning and management. It can help to identify areas of congestion, plan for future infrastructure needs, and optimize traffic flow. With the help of advanced machine learning algorithms, traffic volume prediction can be made more accurate and reliable. However, it is important to note that traffic volume prediction is not a perfect science and there are many factors that can affect the accuracy of the predictions. Therefore, it is important to consider all the factors that can affect traffic volume when making predictions.

9. FUTURE SCOPE

The future scope of predicting traffic volume is very promising. With the advancement of technology, more sophisticated methods of predicting traffic volume can be developed. This includes the use of machine learning algorithms, artificial intelligence, and big data analytics. These methods can be used to accurately predict traffic volume in real-time, allowing for better traffic management and improved safety. Additionally, predictive analytics can be used to identify potential traffic problems before they occur, allowing for proactive solutions to be implemented.

10. BIBILOGRAPLY

https://www.altexsoft.com/blog/trafficprediction/#:~:text=traffic%20flow%20prediction.,Deep%20learning%20approach,are%20based%20on%20neural%20networs.

APPENDIX

SOURCE CODE

APP.PY

```
import numpy as np

import pickle

import joblib

import matplotlib

import matplotlib.pyplot as plt

import time

import pandas

import os

from flask import Flask, request, jsonify, render_template



app = Flask(__name__)

model = pickle.load(open('model.pkl','rb'))

#scale = pickle.load(open('encoder.pkl','rb'))


@app.route('/')# route to display the home page

def home():

    return render_template('index.html') #rendering the home page


@app.route('/predict',methods=["POST","GET"])# route to show the predictions in a web UI

def predict():
    #  reading the inputs given by the user
    input_feature=[float(x) for x in request.form.values() ]
    features_values=[np.array(input_feature)]
    names = [['holiday','temp', 'rain', 'snow', 'weather', 'year', 'month', 'day','hours', 'minutes', 'seconds']]
    data = pandas.DataFrame(features_values,columns=names)
    #data = scale.fit_transform(data)
     # predictions using the loaded model file
    prediction=model.predict(data)
    print(prediction)
    text = "Estimated Traffic Volume is :"
    return render_template("index.html",prediction_text = text + str(prediction))
     # showing the prediction results in a UI
if __name__=="__main__":

    # app.run(host='0.0.0.0', port=8000,debug=True)    # running the app
    port=int(os.environ.get('PORT',5000))
    app.run(port=port,debug=True,use_reloader=False)
```

INDEX.HTML

```
<!DOCTYPE html>
<html >
<head>
  <meta charset="UTF-8">
  <title>Traffic Volume Estimation</title>
</head>
```

```
<body background="https://cdn.vox-
cdn.com/thumbor/voARJfEKvTp6iMSzW3ExPn06TDM=/0x78:3000x1766/1600x900/cdn.vox-
cdn.com/uploads/chorus_image/image/44219366/72499026.0.0.jpg" text="black">


 <div class="login">
                  <center><h1>Traffic Volume Estimation</h1></center>


     <!-- Main Input For Receiving Query to our ML -->
    <form action="{{ url_for('predict')}}"method="post">
<h1>Please enter the following details</h1>




</style></head>


<br>   <label>temp:</label>
      <input type="number"   name="temp" placeholder="temp         " required="required" /><br>


 <br>
       <label>rain:</label>
      <input type="number"  min="0" max="1"  name="rain  " placeholder="rain"
required="required" /><br>


<br>
       <label>snow:</label>
      <input type="number" min="0" max="1"   name="snow            " placeholder="snow   "
required="required" /><br>




<br>
```

```
      <label for="weather">weather:</label>
        <select id="weather" name="weather">
            <option value=1>Clouds</option>
            <option value=0>Clear</option>
            <option value=6>Rain</option>
            <option value=2>Drizzle</option>
            <option value=5>Mist</option>
            <option value=4>Haze</option>
            <option value=3>Fog</option>
            <option value=10>Thunderstorm</option>
            <option value=8>Snow</option>
            <option value=9>Squall</option>
            <option value=7>Smoke</option><



        </select> &nbsp;&nbsp;<br>
<br>
       <label>year:</label>
      <input type="number" min="2012" max="2022"   name="year  " placeholder="year    "
required="required" /><br>


<br>
            <label>month:</label>
      <input type="number" min="1" max="12"   name="month        " placeholder="month "
required="required" /><br>


<br>
           <label>day:</label>
       <input type="number" min="1" max="31"   name="day" placeholder="day     "
required="required" /><br>


<br>
        <label>hours:</label>
       <input type="number" min="0" max="24"   name="hours         " placeholder="hours  "
required="required" /><br>


<br>
             <label>minutes:</label>
       <input type="number" min="0" max="60"   name="minutes      " placeholder="minutes
                    " required="required" /><br>


<br>
        <label>seconds:</label>
       <input type="number" min="0" max="60"   name="seconds      " placeholder="seconds
                    " required="required" /><br>


<br>


<br><br>




<button type="submit" class="btn btn-primary btn-block btn-large"
style="height:30px;width:200px">Predict</button>


    </form>
<br>
```

```
   {{ prediction_text }}



  <br>
  <br>
  <img src="data:image/png;base64,{{url_3}}" alt="Submit Form" height="180" width="233" onerror="this.style.display='none'"/>
  <img src="data:image/png;base64,{{url_1}}" alt="Submit Form" height="180" width="233" onerror="this.style.display='none'"/>
  <img src="data:image/png;base64,{{url_4}}" alt="Submit Form" height="180" width="233" onerror="this.style.display='none'"/>
  <br>
  <br>
  <img src="data:image/png;base64,{{url_2}}" alt="Submit Form" height="150" width="711" onerror="this.style.display='none'"/>

 </div>



</body>
</html>
```