

DECLARATION

I, CHINTHA GOWTHAMI (Reg No: 23MK1A0519) hereby declare that the Project Report entitled "Advanced Traffic Volume Estimation With Machine Learning" done by me under the guidance of Mr. Ganesh at Smart Bridge (Smart Internz) is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in Computer Science and Engineering.

DATE:

PLACE: SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of SATHYABAMA for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to Dr. T. Sasikala M.E., Ph.D., Dean, School of Computing , Dr.S.Vigneshwari M.E., Ph.D., and Dr.L.Lakshmanan M.E., Ph.D., Heads of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide MR. Ganesh, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Computer Science and Engineering who were helpful in many ways for the completion of the project

ABSTRACT

The purpose of this project is to design and develop a traffic assessment system. Traffic estimate is determined by the amount of traffic congestion. Traffic jams cause people to lose valuable time, energy and frustration every day. Congestion is a global problem that affects all levels of society. The most common causes of traffic congestion are any driver getting stuck in a traffic jam on their journey. Accidents such as road accidents and road accidents often lead to unexpected unforeseen delays. There are also bad weather conditions due to low traffic flow speeds. It is difficult to accurately estimate traffic flow due to the very large data of the

transportation system. This fact prompted us to work on a traffic prediction system to accurately and timely assess traffic flow information. We plan to use machine learning for prediction and regression based algorithm for image detection to analyze the bulk data of the transport system, we will use various graphical user fronts for interactive application. Machine learning provides better accuracy for Traffic volume flow prediction. It's addressed as a major element for the success of advanced traffic volume management systems, advanced public transportation systems, and traveller information systems. The rationale of this extension is to develop a prescient demonstration utilizing different machine learning calculations and to record the end-to-end steps. The Metro Interstate Activity Volume dataset could also be a relapse circumstance where we are trying to anticipate the esteem of a ceaseless variable. We'll be analysing how the drift of month-to-month interstate activity volume changes over an extended time between 2012 and 2018.

Key Words: Traffic Volume, Random Forest, Machine Learning, RSME, Flask

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
4.1	Heat Map	10
4.2	Seaborn Axis Grid	11
4.3	Box Plot	12

LIST OF TABLES		
TABLE NO.	TABLERNAME	PAGENO.
3.1	Data Header	4
3.2	Describe the Data	5
3.3	Data Head After Splitting in Date and Time	5
4.1	Correlation of the Data	9
4.1	Feature Scaling of X-GRID	13

CHAPTER 1. INTRODUCTION

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 areas. Traffic 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. Traffic jams on Urban Network are increasing day by day, because the traffic demand increases, and the speed of the vehicles is drastically reduced thus causing longer vehicular queuing and more such cases substantially hamper the traffic flow by giving rise to holdup.

Motivation

With the progress of urbanization and therefore the recognition of automobiles, transportation problems are becoming more and more challenging: the traffic volume flow is congested, wear and tear of vehicles, delays end in the late time of arrival at the meeting, accidents are frequent, and wastage of fuel while waiting in traffic, the traffic environment is becoming worse, to unravel this problem and to assist society, we've chosen our topic as traffic volume prediction.

1.2 Problem Definition

Now? The question arises of how to improve the capacity of the road network. To solve this problem the first solution that occurs to most of us is to build more highways, expanding the number of lanes on the road. However, according to the study done by scholars, expanding the road capacity will cause more serious traffic conditions. Therefore, traffic volume prediction is one of the most famous.

CHAPTER 2. AIM AND SCOPE OF THE PRESENT INVESTIGATION

2.1 AIM:

We will be using Regression algorithms such as Linear Regression, Decision tree, Random forest, and xgboost to predict the count of traffic volume. We will train and test the data with these algorithms. From this best model is selected and saved in .pkl (Pickle) format. Once the model is saved, we integrate it with flask application and also deploy the model in IBM.

2.2 SCOPE:

The objective of this study is to seek out a traffic volume predictor suitable for real implications. This predictor must be accurate in terms of computation cost and power consumption. Within the go after such a predictor, we've included the subsequent contributions: We compare existing schemes to seek out their effectiveness for real-time applications.

CHAPTER 3. EXPERIMENTAL OR MATERIALS AND METHODS; ALGORITHMS USED

3.1 Pre Requisites

To complete this project, you must require the following software's, concepts, and packages

Anaconda navigator

Packages

Open anaconda prompt as administrator.

Type "pip install numpy" and click enter

Type "pip install pandas" and click enter.

Type "pip install matplotlib" and click enter.

Type "pip install scikit-learn" and click enter.

Type "pip install Flask" and click enter.

Type "pip install xgboost" and click enter.

3.2 Project Objectives

By the end of this project:

You'll be able to understand the problem to classify if it is a regression or a classification kind of problem.

You will be able to know how to pre-process/clean the data using different data pre-processing techniques.

You will be able to analyze or get insights into data through visualization.

Applying different algorithms according to a dataset and based on visualization.

You will be able to know how to find the accuracy of the model.

You will be able to know how to build a web application using the Flask framework.

3.3 Project Flow

User interacts with the UI (User Interface) to enter the input values.

Entered input values are analyzed by the model which is integrated.
Once the model analyses the input the prediction is showcased on the UI.
To accomplish this, we have to complete all the activities and tasks listed below

Data Collection.

Collect the dataset or Create the dataset

Data Pre-processing.

Import the Libraries.

Importing the dataset.

Checking for Null Values.

3.4 Data Collection

ML depends heavily on data, without data, it is impossible for an “AI” model to learn. It is the most crucial aspect that makes algorithm training possible. In Machine Learning projects, we need a training data set. It is the actual data set used to train the model for performing various actions.

3.1 DATA HEADER

3.2 DESCRIBE THE DATA

3.4 DATA HEAD AFTER SPLITTING IN DATE AND TIME

3.5 Import Necessary Libraries

It is important to import all the necessary libraries such as pandas, NumPy, matplotlib.

NumPy- It is an open-source numerical Python library. It contains a multi-dimensional array and matrix data structures. It can be used to perform mathematical operations on arrays such as trigonometric, statistical, and algebraic routines.

Pandas- It is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language.

Seaborn- Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

Matplotlib- Visualisation with python. It is a comprehensive library for creating static, animated, and interactive visualizations in Python

Sklearn – which contains all the modules required for model building.

CHAPTER 4. RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS

4.1 Project Structure:

Flask files consist of template folder which has HTML pages, app.py file and .pkl files which are used for application building

IBM folder has flask files and scoring endpoint.ipynb- model training code file.

We need the model which is saved and the saved model in this content is Traffic volume. Pkl

Templates folder which contains index.HTML file, chance.HTML file, noChance.HTML file.
Scale.pkl for scaling, encoder.pkl file for encoding the categorical data, imputer.pkl file for filling out the missing values

4.2 Importing The Dataset

You might have your data in .csv files, .excel files

Let's load a .csv data file into pandas using read_csv() function. We will need to locate the directory of the CSV file at first (it's more efficient to keep the dataset in the same directory as your program).

If your dataset is in some other location, Then

```
Data=pd.read_csv(r"File_location/datasetname.csv")
```

If the dataset is in the same directory of your program, you can directly read it, without giving raw as r.

Our Dataset weatherAus.csv contains the following Columns

Holiday - working day or holiday

Temp- temperature of the day

Rain and snow – whether it is raining or snowing on that day or not

Weather = describes the weather conditions of the day

Date and time = represents the exact date and time of the day

Traffic volume – output column

The output column to be predicted is Traffic volume. Based on the input variables we predict the volume of the traffic. The predicted output gives them a fair idea of the count of traffic

Analyse The Data

head() method is used to return top n (5 by default) rows of a DataFrame or series.

4.3 Handling the Missing values

1. The Most important step in data pre-processing is dealing with missing data, the presence of missing data in the dataset can lead to low accuracy.

2. Check whether any null values are there or not. if it is present then the following can be done. There are missing values in the dataset, we will fill the missing values in the columns.

3. We are using mean and mode methods for filling the missing values

Columns such as temp, rain, and snow are the numeric columns, when there is a numeric column you should fill the missing values with the mean/median method. so here we are using the mean method to fill the missing values.

Weather column has a categorical data type, in such case missing data needs to be filled with the most repeated/ frequent value. Clouds are the most repeated value in the column, so imputing with clouds value.

4.4 Data Visualization:

Data visualization is where a given data set is presented in a graphical format. It helps the detection of patterns, trends and correlations that might go undetected in text-based data.

Understanding your data and the relationship present within it is just as important as any algorithm used to train your machine learning model. In fact, even the most sophisticated

machine learning models will perform poorly on data that wasn't visualized and understood properly.

To visualize the dataset we need libraries called Matplotlib and Seaborn.

The Matplotlib library is a Python 2D plotting library that allows you to generate plots, scatter plots, histograms, bar charts etc.

Let's visualize our data using Matplotlib and seaborn library.

Before diving into the code, let's look at some of the basic properties we will be using when plotting.

xlabel: Set the label for the x-axis.

ylabel: Set the label for the y-axis.

title: Set a title for the axes.

Legend: Place a legend on the axes.

`data.corr()` gives the correlation between the columns

4.1 Correlation of the Data

Correlation is a statistical term describing the degree to which two variables move in coordination with one another. If the two variables move in the same direction, then those variables are said to have a positive correlation. If they move in opposite directions, then they have a negative correlation.

Correlation strength varies based on colour, lighter the colour between two variables, more the strength between the variables, darker the colour displays the weaker correlation

We can see the correlation scale values on the left side of the above image

4.1 HEAT MAP

Pair Plot: Plot pairwise relationships in a dataset.

A pair plot is used to understand the best set of features to explain a relationship between two variables or to form the most separated clusters. It also helps to form some simple classification models by drawing some simple lines or making a linear separation in our data-set.

By default, this function will create a grid of Axes such that each numeric variable in data will be shared across the y-axes across a single row and the x-axes across a single column. The diagonal plots are treated differently: a univariate distribution plot is drawn to show the marginal distribution of the data in each column.

We implement this using the below code.

Pair plot usually gives pairwise relationships of the columns in the dataset

From the above pair plot, we infer that

1. From the above plot we can draw inferences such as linearity and strength between the variables. how features are correlated (positive, neutral and negative)

4.2 SEABORN AXIS GRID

BOXPLOT:

4.3 BOX PLOT

Box-plot is a type of chart often used in explanatory data analysis. Box plots visually show the distribution of numerical data and skewness through displaying the data quartiles (or percentiles) and averages.

Box plots are useful as they show the average score of a data set. The median is the average value from a set of data and is shown by the line that divides the box into two parts. Half the scores are greater than or equal to this value and half are less.

jupyter has a built-in function to create a boxplot called `boxplot()`. A boxplot plot is a type of plot that shows the spread of data in all the quartiles.

4. Data and time columns need to be split into columns so that analysis and training of the model can be done in an easy way, so we use the split function to convert date into the year, month and day. time column into hours, minutes and seconds.

4.5 Splitting The Dataset Into Dependent And Independent Variable

- In machine learning, the concept of the dependent variable (y) and independent variables(x) is important to understand. Here, the Dependent variable is nothing but output in dataset and the independent variable is all inputs in the dataset.

- With this in mind, we need to split our dataset into the matrix of independent variables and the vector or dependent variable. Mathematically, Vector is defined as a matrix that has just one column.

To read the columns, we will use `iloc` of pandas (used to fix the indexes for selection) which takes two parameters — [row selection, column selection].

Let's split our dataset into independent and dependent variables.

```
y = data[traffic_volume] – independent x = data.drop(traffic_volume,axis=1)
```

Feature Scaling

4.1 FEATURE SCALING OF X-GRID

After scaling the data will be converted into an array form

Loading the feature names before scaling and converting them back to data frame after standard scaling is applied

4.7 Splitting The Data Into Train And Test

When you are working on a model and you want to train it, you obviously have a dataset. But after training, we have to test the model on some test datasets. For this, you will a dataset which is different from the training set you used earlier. But it might not always be possible to have so much data during the development phase. In such cases, the solution is to split the dataset into two sets, one for training and the other for testing.

The train-test split is a technique for evaluating the performance of a machine learning algorithm.

Train Dataset: Used to fit the machine learning model.

Test Dataset: Used to evaluate the fit machine learning model.

In general you can allocate 80% of the dataset to the training set and the remaining 20% to test.

Now split our dataset into train set and test using `train_test_split` class from `sci-kit learn` library.

4.8 Training And Testing The Model

Once after splitting the data into train and test, the data should be fed to an algorithm to build a model.

There are several Machine learning algorithms to be used depending on the data you are going to process such as images, sound, text, and numerical values. The algorithms that you can choose according to the objective that you might have it may be Classification algorithms are Regression algorithms.

- 1.Linear Regression
- 2.Decision Tree Regressor
- 3.Random Forest Regressor
- 4.KNN
- 5.svm
- 5.xgboost

We're going touse the x-train and y-train obtained above in the `train_test_split` section to train our Random forest regression model. We're using the `fit` method and passing the parameters as shown below.

We are using the algorithm from Scikit learn library to build the model as shown below, Once the model is trained, it's ready to make predictions. We can use the `predict` method on the model and pass `x_test` as a parameter to get the output as `y_pred`.

Notice that the prediction output is