PREDICTING EMISSION OF CO2 FROM CARS USING MACHINE LEARNING

Mini Project submitted in partial fulfillment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

Under the esteemed guidance of

Mrs. S. Radha
Assistant Professor

By

K Navya - 18R11A0573
D Sri Nikita - 18R11A0563
G Raghav - 18R11A0565

Department of Computer Science and Engineering
Accredited by NBA

Geethanjali College of Engineering and Technology
(UGC Autonomous)
(Affiliated to J.N.T.U.H, Approved by AICTE, New Delhi)
Cheeryal (V), Keesara (M), Medchal.Dist.-501 301.

November-2021

Geethanjali College of Engineering & Technology

(UGC Autonomous)
(Affiliated to JNTUH, Approved by AICTE, New Delhi)
Cheeryal (V), Keesara (M), Medchal Dist.-501 301.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
Accredited by NBA

CERTIFICATE

This is to certify that the B.Tech Mini Project report entitled **"Predicting Emission of CO2 from cars using Machine Learning"** is a bonafide work done by **K Navya (18R11A0573), D Sri Nikita(18R11A0563), G Raghav(18R11A0565),** in partial fulfillment of the requirement of the award for the degree of Bachelor of Technology in "**Computer Science and Engineering**" from Jawaharlal Nehru Technological University, Hyderabad during the year 2021-22.

Signature of Internal Guide
Mrs S. Radha
Assistant Professor,
Dept of CSE.

Signature of the HOD
Dr. A. Sri Lakshmi
Professor and HOD,
Dept of CSE.

External Examiner

Geethanjali College of Engineering & Technology

(UGC Autonomous)
(Affiliated to JNTUH Approved by AICTE,
New Delhi) Cheeryal (V), Keesara(M),
Medchal Dist.-501 301.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Accredited by NBA

DECLARATION BY THE CANDIDATE

We, **K Navya, D Sri Nikita** and **G Raghav** bearing Roll Nos. **18R11A0573, 18R11A0563** and **18R11A0565** hereby declare that the project report entitled **"Predicting Emission of Co2 from Cars using Machine Learning "** is done under the guidance of **Mrs S. Radha**, **Assistant Professor**, Department of Computer Science and Engineering, Geethanjali College of Engineering and Technology, is submitted in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering**.

This is a record of bonafide work carried out by us in **Geethanjali College of Engineering and Technology** and the results embodied in this project have not been reproduced or copied from any source. The results embodied in this project report have not been submitted to any other University or Institute for the award of any other degree or diploma.

K NAVYA(18R11A0573)

D SRI NIKITA (18R11A0563)

G RAGHAV (18R11A0565)

ACKNOWLEDGEMENT

Before getting started we would like to thank the personalities who were part of our project in numerous ways and have provided immense support and motivation right from the start.

We are highly obliged to **Dr. S. Udaya Kumar,** Principal of Geethanjali College of Engineering and Technology, for his support during the tenure of the project.

We are extremely thankful to our HOD of Computer Science Department, **Dr. A. Sri Lakshmi,** Professor, for providing the opportunity to undertake the project and encouragement in its completion.

We hereby express our deep sense of gratitude to **Mrs S. Radha**, Assistant Professor, Department of Computer Science and Engineering, Geethanjali College of Engineering and Technology for the esteemed guidance, moral support and invaluable advice she has given us during the course of the project that has amounted to the success of the project.

We are also thankful to all the staff and faculty members of Computer Science Department who have cooperated in making our project a success.

Finally, we would like to thank all our parents and friends who extended their help and encouragement in completion of our project.

TABLE OF CONTENTS

ABSTRACT

Our personal vehicles are a major cause of global warming. Transportation is one of the main reasons for pollution and also global warming. Vehicles emit large amounts of Carbon dioxide when they have faulty engines.

A typical passenger vehicle emits about 4.6 metric tons of carbon dioxide per year. This number can vary based on a vehicle's fuel, fuel economy, and the number of miles driven per year. India is the fifth-largest passenger vehicle market globally by sales volume, with annual sales in FY 2017–2018 reaching 3.3 million. Deliveries in India expanded at an annual rate of 9% from fiscal 2015–2016 to 2017–2018 and posted a 10-year annual average growth rate of 8%. Compared with other global markets, India has had a large share of diesel engines in its passenger vehicle fleet. In FY 2010–2011, about 35% of India's passenger vehicle fleet used diesel, and the remainder predominantly gasoline.

So, this machine learning project will help us predict the values of CO2 emitted based on the input that we give. It will help us to categorize whether the vehicle is faulty or in good condition. It acts like an automated pollution check, thereby helping us to find out the faulty vehicles so that appropriate action can be taken to reduce the CO2 emissions. The minimum amount of CO2 emitted for an environmental friendly car is considered to be 250g/km.

LIST OF FIGURES

LIST OF ABBREVIATIONS

S.NO	Abbreviation	Full Form
1	UML	Unified Modeling Language
2	API	Application Programming Interface
3	HTML	Hyper Text Markup Language

LIST OF SCREENSHOTS

1. INTRODUCTION

1.1 ABOUT THE PROJECT

- Machine learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values.

- Machine learning is important because it gives enterprises a view of trends in customer behavior and business operational patterns, as well as supports the development of new products. Many of today's leading companies, such as Facebook, Google and Uber, make machine learning a central part of their operations. Machine learning has become a significant competitive differentiator for many companies.

- Classical machine learning is often categorized by how an algorithm learns to become more accurate in its predictions. There are four basic approaches: supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning. The type of algorithm data scientists choose to use depends on what type of data they want to predict.

- So, we are using machine learning for predicting the emission of co2 values from cars with help of linear regression and random forest regression algorithms. Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

- Machine Learning has following four steps**:**
  - Data Collection
  - Data Pre-processing
  - Model Building
  - Application Building

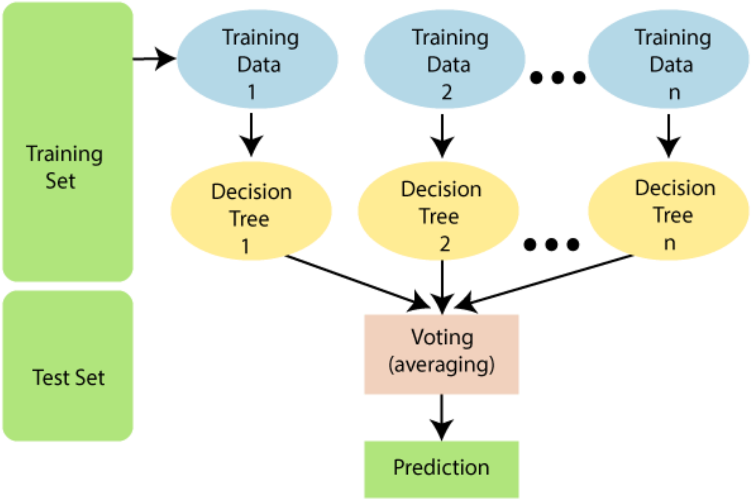


Fig: 1.1 Pictorial representation of Random Forest Regression Algorithm

1.2 OBJECTIVE

Our main objective is to develop a web application using machine learning and flask to predict amount of co2 emitted from cars. This project will help us to differentiate between the cars which are environmental friendly and the cars which are hazardous to the environment.

As technology becomes more and more advanced, developments are being made in the world faster than ever. Machinery is being manufactured in large quantities. So, this project will help the customers to understand the cars which are good for environment and also reduce the level of pollution and to help the manufacturing companies to produce a car which emits less carbon dioxide into the atmosphere.

2. SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

Climate change has mainly been affected by CO2 emissions. Analysing historic data using data mining can help to make future forecasts and formulate policies that can be used to manage carbon emissions. Across the globe, carbon emission policies have been targeted either towards the manufactures side or consumer’s side.The existing system does not provide us with the accurate values of co2 emitted from cars.

2.2 PROPOSED SYSTEM

For this machine learning project we are importing dataset from kaggle.com which has all the information for the co2 emissions of different cars, in addition to it we are training the dataset to get the accurate values and we are using random forest regression algorithm. Emission of CO2 from vehicles, machine learning project can be further improved and developed by adding features where it will record the number plate details of the vehicle finds out the problem because of which CO2 is emitted in an alarming way and also provides a quick fix to solve this problem. It also provides the detail report of condition of the vehicle and the last date and time it underwent the process of automated pollution check. In the existing system we have to enter all the details related to particular car like fuel competency, number of cylinders, type, model etc but for the proposed system it will automatically detects the number plate and using the number plate it will provide all the details related to the car. This enhanced feature will help us to provide accurate solutions to fix the faulty conditions and also suggests other environmental friendly cars and this can be achieved by using both artificial intelligence and deep learning.

2.2.1 Details

1. Firstly, we should import accurate dataset and remove all the duplicate values and null values from the data set and refine it. Then we have to train the data set accordingly.

2. We choose random forest regression algorithm to predict accurate co2 values emitted from cars.

3. After the prediction is done we are then classifying cars as environmental friendly and faulty cars based on the threshold value.

4. The threshold value that we considered for this project is 250g/km. If the emitted amount of carbon dioxide is less than 250 then it is considered to be environmental friendly or it is said to be in faulty condition.

2.2.2 Impact on Environment

1. The main purpose of this project to reduce the pollution levels by making vehicles environmental friendly. If the amount of carbon dioxide released into the atmosphere is decreased then the overall pollution and global warming will decrease.
2. India is highly populated country and the people who use cars as their main transport are increasing thereby increasing the emission of co2 into atmosphere on an alarming level.
3. This project can easily be implemented with the help of Jupiter notebook in the form of web page where the end user just requires giving input values and the accurate results will be displayed on the screen within seconds.
4. The time taken to produce the output is very minute. This web application is fast and can work efficiently with proper internet connectivity.

2.2.3 Safety

This application has a dataset which does not contain any null values or duplicate values. The dataset is free from redundant data.

- Cloud data protection provides and ensures various services and processes, such as: Integrity: Data is in the same form as its stored version. Data encryption measures are implemented to secure data from illegal editing and corruption.

2.2.4 Ethics

Our application does not use any e-content that harm (physically or virtually), we used content, which is an open-source for developers to develop any application, by using Jupiter Notebook. We did not expose any private data. We developed our own code without any plagiarism.

2.2.5 Cost

- Economic evaluation is the presentation of information regarding different profitability of the business or industry of a proposed project. It is prepared by our team and submitted to the institution.
- Cost of development
- Size of application nature of product expected quantity of output
- Amount of fixed capital required
- Raw materials required and their cost (raw material for example dataset from kaggle.com is required which is an open source website)
- Cost of machines used, cost of human work, the capacity of product, etc.
- Cost reduction due to the implementation of the project in production.

2.2.6 Type

- The interactive machine learning project is a web application, which uses technologies like HTML5, css3, JavaScript and Python.
- Jupiter Notebook and flask is also used.

2.2.7 Standards

We have followed the Agile Methodology to create our project. Initially, we gathered all the requirements that are necessary for the project. We have specifically defined the purpose of the software and have divided the entire project into small builds. These builds were worked on for days together to produce an efficient solution for each stage.

Both the design and the testing processes were simultaneous so that it would be easier for use to identify if there was any problem with the code. As soon as there were any changes, we would implement the idea and then repeat the process. This proved out to be an efficient solution that not only made the project execution faster but also easier.

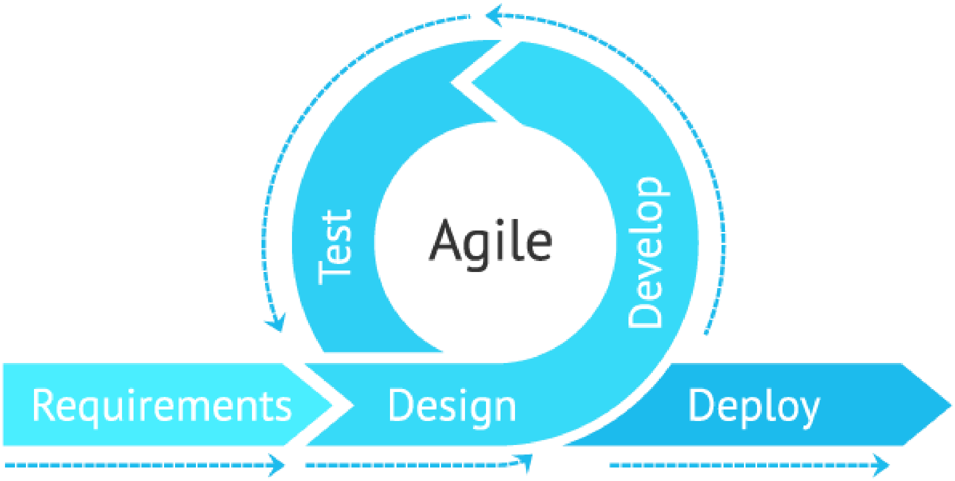


Figure 2.2.7: Agile Model

2.3 SCOPE OF THE PROJECT

This project work is completely based on interactivity of an application to the end-user where users can find out the condition of the car by simply entering some details related to the car and based on these details the machine learning algorithm will make précised predictions and then the accurate results will be displayed. Random forest regression algorithm gave more accurate results than linear regression algorithm. The accuracy of predicted values was near to 0.98 for random forest regression and its 0.92 for linear regression algorithm.

2.4 MODULES DESCRIPTION

1. Data Collection:

The given dataset is related to Emission of Co2 from cars.

Dataset was downloaded from kaggle.com website.

2. Data Pre-Processing:

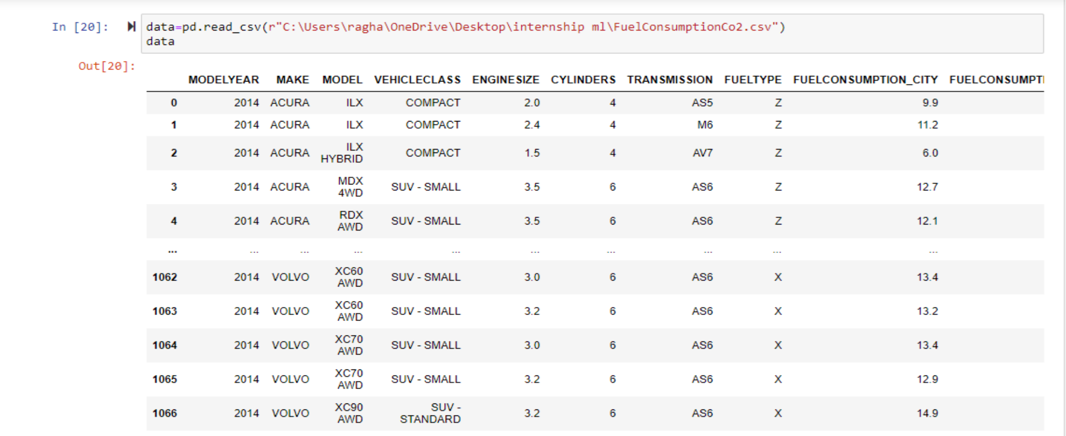
Importing the required packages:

Pandas: It is a python library mainly used for data manipulation.

Numpy: This python library is used for numerical analysis.

Train_test_split: used for splitting data arrays into training and testing data.

- Importing the dataset:



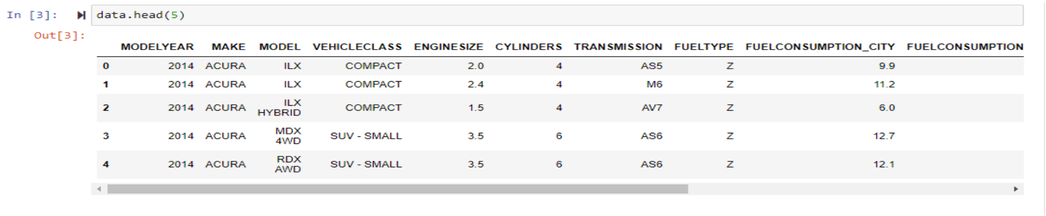
```
In [20]:  data=pd.read_csv(r"C:\Users\ragha\OneDrive\Desktop\internship ml\FuelConsumptionCo2.csv")
          data
Out[20]:
```

	MODELYEAR	MAKE	MODEL	VEHICLECLASS	ENGINESIZE	CYLINDERS	TRANSMISSION	FUELTYPE	FUELCONSUMPTION_CITY	FUELCONSUMPTI
0	2014	ACURA	ILX	COMPACT	2.0	4	AS5	Z	9.9	
1	2014	ACURA	ILX	COMPACT	2.4	4	M6	Z	11.2	
2	2014	ACURA	ILX HYBRID	COMPACT	1.5	4	AV7	Z	6.0	
3	2014	ACURA	MDX 4WD	SUV - SMALL	3.5	6	AS6	Z	12.7	
4	2014	ACURA	RDX AWD	SUV - SMALL	3.5	6	AS6	Z	12.1	
...	...	...	...	...	...	...	...	...	...	
1062	2014	VOLVO	XC60 AWD	SUV - SMALL	3.0	6	AS6	X	13.4	
1063	2014	VOLVO	XC60 AWD	SUV - SMALL	3.2	6	AS6	X	13.2	
1064	2014	VOLVO	XC70 AWD	SUV - SMALL	3.0	6	AS6	X	13.4	
1065	2014	VOLVO	XC70 AWD	SUV - SMALL	3.2	6	AS6	X	12.9	
1066	2014	VOLVO	XC90 AWD	SUV - STANDARD	3.2	6	AS6	X	14.9	

3) Data Visualization:

Exploratory data analysis is an approach to analyzing data sets to summarize their main characteristics, often with visual methods and used for determine how best to manipulate data sources to get the answers you need, making it easier for data scientists to discover patterns, spot anomalies, test a hypothesis, or check assumptions.

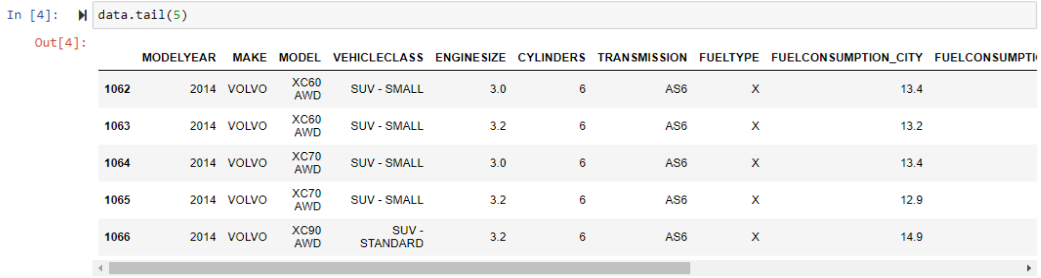
- To check first five rows of dataset, we have a function call head( ) This returns the first 5 values of the dataset.



```
In [3]:  data.head(5)
Out[3]:
```

	MODELYEAR	MAKE	MODEL	VEHICLECLASS	ENGINESIZE	CYLINDERS	TRANSMISSION	FUELTYPE	FUELCONSUMPTION_CITY	FUELCONSUMPTION
0	2014	ACURA	ILX	COMPACT	2.0	4	AS5	Z	9.9	
1	2014	ACURA	ILX	COMPACT	2.4	4	M6	Z	11.2	
2	2014	ACURA	ILX HYBRID	COMPACT	1.5	4	AV7	Z	6.0	
3	2014	ACURA	MDX 4WD	SUV - SMALL	3.5	6	AS6	Z	12.7	
4	2014	ACURA	RDX AWD	SUV - SMALL	3.5	6	AS6	Z	12.1	

- To check last 5 values of the dataset, we use the tail().



```
In [4]:  data.tail(5)
Out[4]:
```

	MODELYEAR	MAKE	MODEL	VEHICLECLASS	ENGINESIZE	CYLINDERS	TRANSMISSION	FUELTYPE	FUELCONSUMPTION_CITY	FUELCONSUMPTI
1062	2014	VOLVO	XC60 AWD	SUV - SMALL	3.0	6	AS6	X	13.4	
1063	2014	VOLVO	XC60 AWD	SUV - SMALL	3.2	6	AS6	X	13.2	
1064	2014	VOLVO	XC70 AWD	SUV - SMALL	3.0	6	AS6	X	13.4	
1065	2014	VOLVO	XC70 AWD	SUV - SMALL	3.2	6	AS6	X	12.9	
1066	2014	VOLVO	XC90 AWD	SUV - STANDARD	3.2	6	AS6	X	14.9	

Taking care of Missing Data:

We will be using isnull().any() method to see which column has missing values.

Dropping the duplicate values:

we use this syntax to drop the duplicate values in the dataset drop_duplicates()

4. Feature Scaling:

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

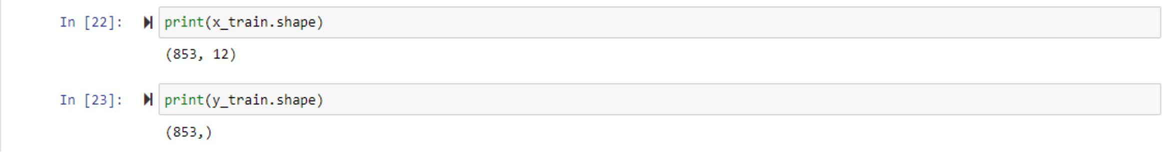
- **Train Dataset**: Used to fit the machine learning model.
- **Test Dataset**: Used to evaluate the fit machine learning model.

There are a few other parameters that we need to understand before we use the class:

a) **test_size** — this parameter decides the size of the data that has to be split as the test dataset. This is given as a fraction. For example, if you pass 0.5 as the value, the dataset will be split 50% as the test dataset

b) **train_size** — you have to specify this parameter only if you're not specifying the test_size. This is the same as test_size, but instead you tell the class what percent of the dataset you want to split as the training set.

c) **random_state** — here you pass an integer, which will act as the seed for the random number generator during the split. Or, you can also pass an instance of the Random_state class, which will become the number generator. If you don't pass anything, the Random_state instance used by np.random will be used instead.Now split our dataset into train set and test using train_test_split class from scikit learn library.



```
In [22]:  print(x_train.shape)
          (853, 12)

In [23]:  print(y_train.shape)
          (853,)
```

5. Model Building

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.

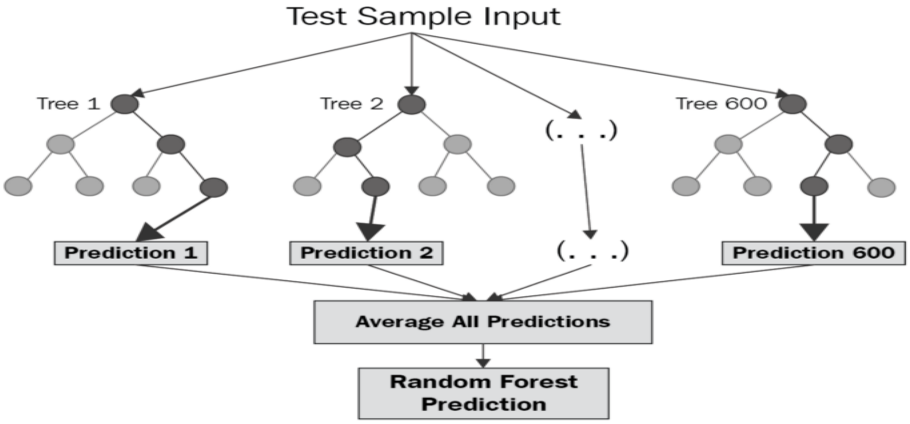
Example: 1. Linear Regression.

2. Logistic Regression.

3. Random Forest Regression / Classification.

4. Decision Tree Regression / Classification.

Now we apply Random Forest Regression algorithm on our dataset.



6. Application Building

Creating a HTML File and flask application.

Task 1: Importing Libraries

Importing the flask module in the project is mandatory. An object of Flask class is our WSGI application. Flask constructor takes the name of the current module (__name__) as argument pickle library to load the model file.

Task 2: Routing to the html Page

Here, declared constructor is used to route to the HTML page created earlier. In the above example, '/' URL is bound with home.html function. Hence, when the home page of the web server is opened in browser, the html page is rendered. Whenever you enter the values from the html page the values can be retrieved using POST Method.

Task 3: Run the application

- Open the anaconda prompt from the start menu.
- Navigate to the folder where your app.py resides.
- Now type "python co2.py" command.
- It will show the local host where your app is running on **http://127.0.0.1:5000/**
- Copy that local host URL and open that URL in the

browser. It does navigate me to where you can view your web page.

- Enter the values, click on the predict button and see the result/prediction on the web page.

2.5 SYSTEM CONFIGURATION

System analysis is done to know about the requirements in both functional and non-functional perspective. It gives software requirement specification that gives an overview of the system.

3. LITERATURE OVERVIEW

Software Requirements

i. Applications: Jupyter Notebook, Anaconda Python

ii. Language: Python

iii. Tools: Microsoft Excel

iv. Kaggle.com for dataset

Hardware Requirements

RAM: 4GB or more

Processor: 2 GHz or more

4. SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

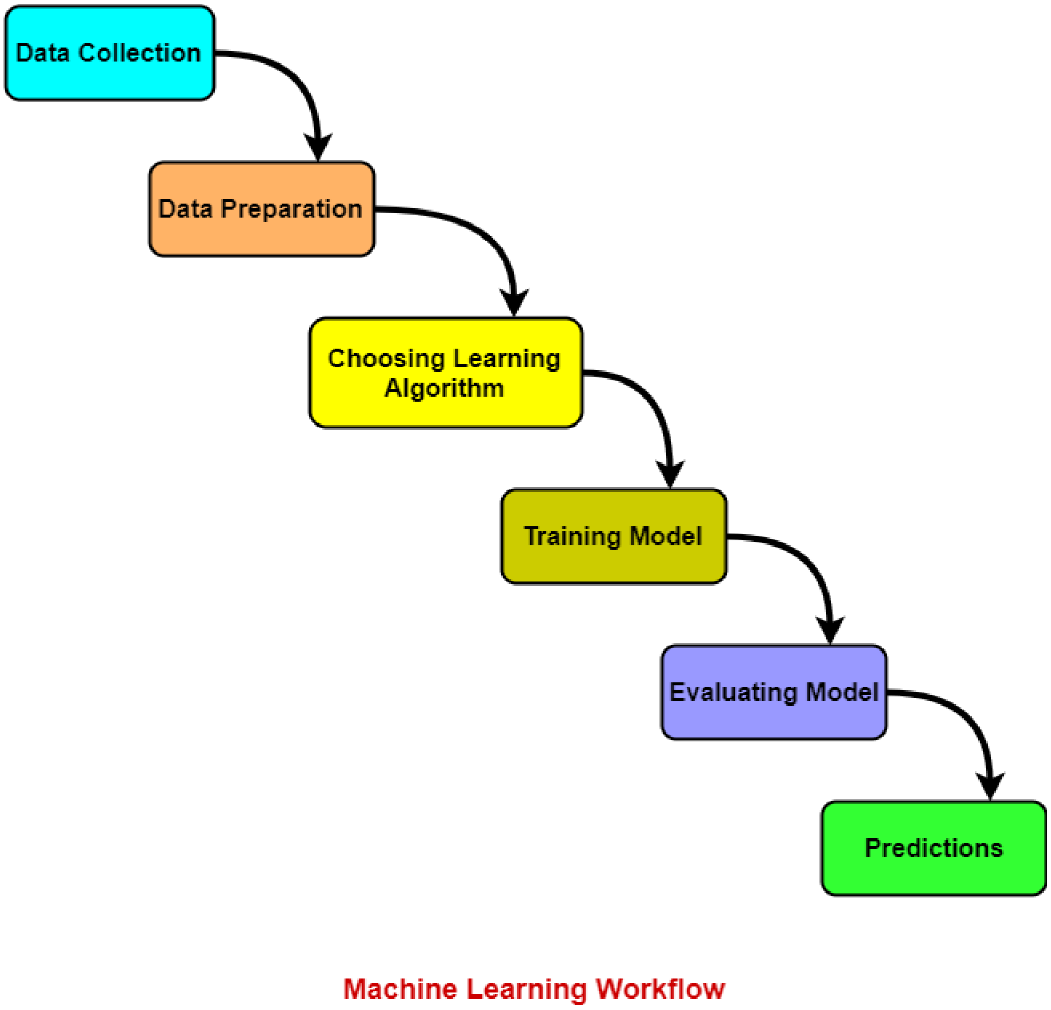


Figure 4.1:

Data Collection:

During data collection, you are defining the potential usefulness and accuracy of your project with the quality of the data you collect. To collect data, you need to identify your sources and aggregate data from those sources into a single dataset. This could mean streaming data from Internet of Things sensors, downloading open source data sets, or constructing a data lake from assorted files, logs, or media.

Data pre-processing:

Pre-processing involves cleaning, verifying, and formatting data into a usable dataset. If you are collecting data from a single source, this may be a relatively straightforward process. However, if you are aggregating several sources you need to make sure that data formats match, that data is equally reliable, and remove any potential duplicates.

Choosing Learning Algorithm:

Knowing your data is the first and foremost step of deciding on an algorithm. Before you start thinking about the different algorithms, you need to familiarize yourself with your data. A simple way to do that is to visualize the data and try to find patterns within it, try to observe it's behavior, and, most importantly of all, its size.

Training:

Once you have datasets, you are ready to train your model. This involves feeding your training set to your algorithm so that it can learn appropriate parameters and features used in classification.

Once training is complete, you can then refine the model using your validation dataset.

Evaluation:

Testing uses your test dataset and is meant to verify that your models are using accurate features. Based on the feedback you receive you may return to training the model to improve accuracy, adjust output settings, or deploy the model as needed.

Prediction:

Prediction refers to the output of an algorithm after it has been trained on a historical dataset and applied to new data when forecasting the likelihood of a particular outcome.

4.2 UML DIAGRAMS

UML, which stands for Unified Modeling Language, is a way to visually represent the architecture, design, and implementation of complex software systems. When you’re writing code, there are thousands of lines in an application, and it’s difficult to keep track of the relationships and hierarchies within a software system. UML diagrams divide that software system into components and subcomponents.

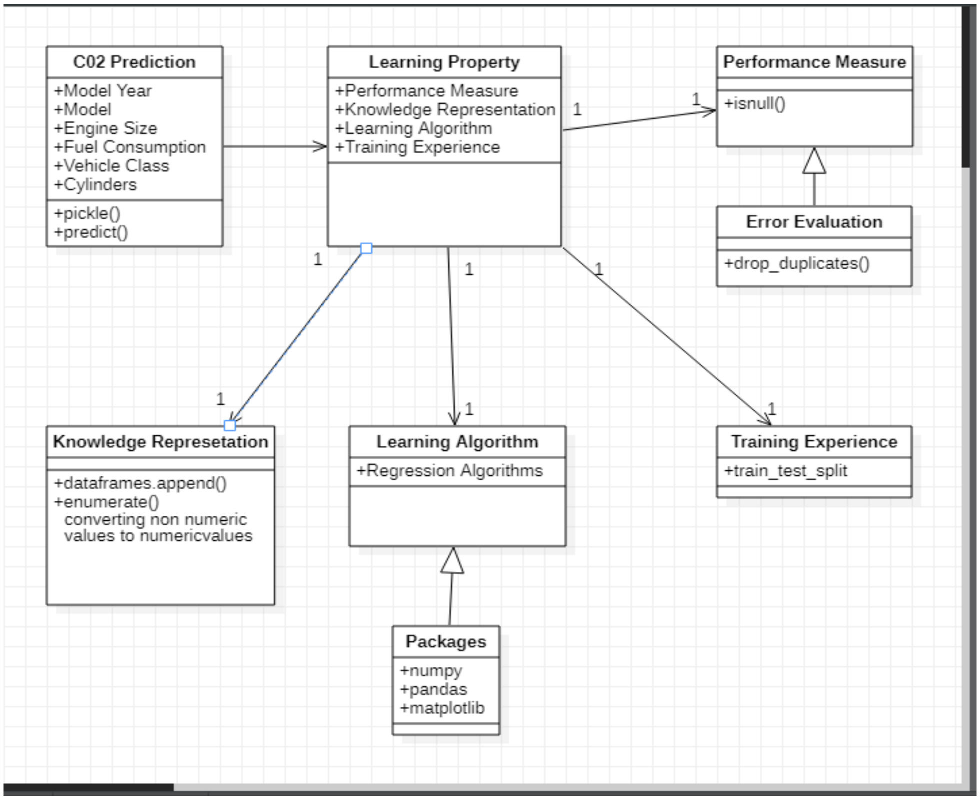
UML diagrams included in this project are:

- Class Diagram
- Use Case Diagram
- Sequence Diagram
- State Flowchart Diagram

4.2.1 Class Diagram

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the structure of the application, and for detailed modeling, translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

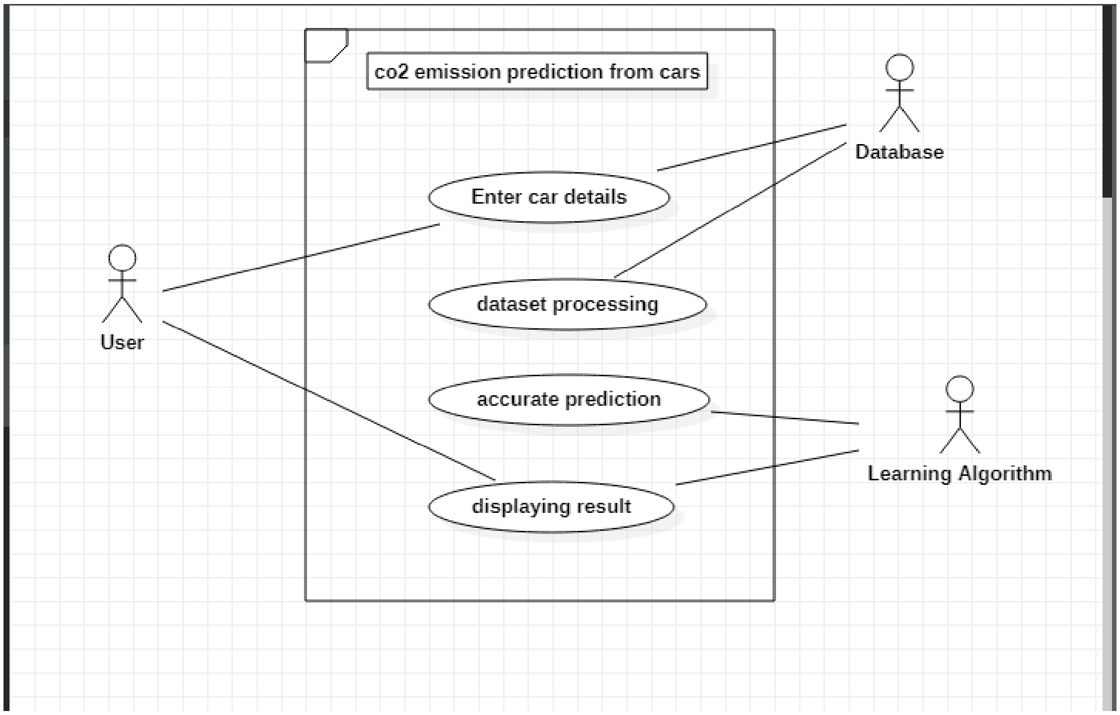


4.2.2 Use-Case Diagram

In the Unified Modeling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll use a set of specialized symbols and connectors. An effective use case diagram can help your team discuss and represent:

- Scenarios in which your system or application interacts with people, organizations, or external systems
- Goals that your system or application helps those entities (known as actors) achieve
- The scope of your system

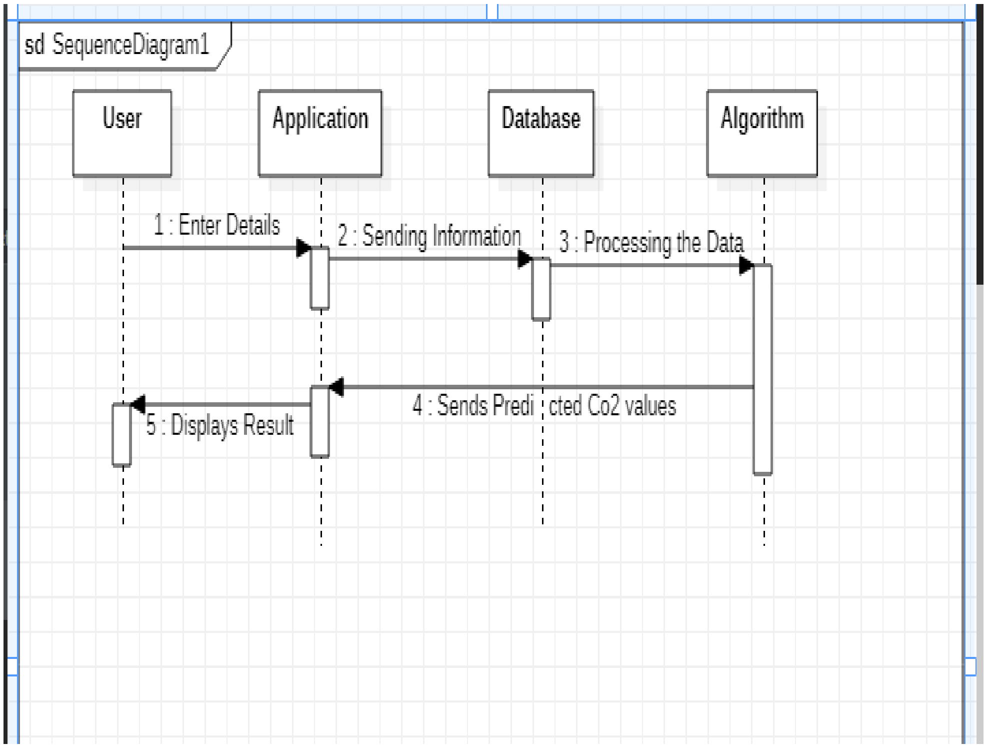
A use case diagram doesn't go into a lot of detail—for example, don't expect it to model the order in which steps are performed. Instead, a proper use case diagram depicts a high-level overview of the relationship between use cases, actors, and systems. Experts recommend that use case diagrams be used to supplement a more descriptive textual use case.



4.2.3 Sequence Diagram

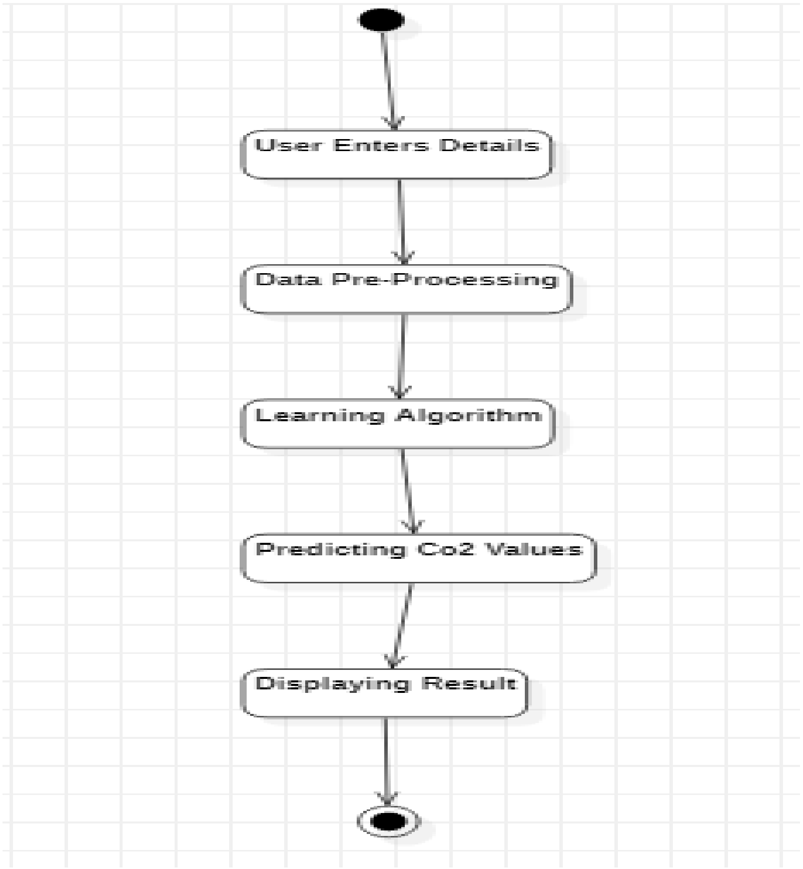
A sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process. Sequence diagrams are sometimes known as event diagrams or event scenarios.

A sequence diagram is structured in such a way that it represents a timeline which begins at the top and descends gradually to mark the sequence of interactions. Each object has a column and the messages exchanged between them are represented by arrows.



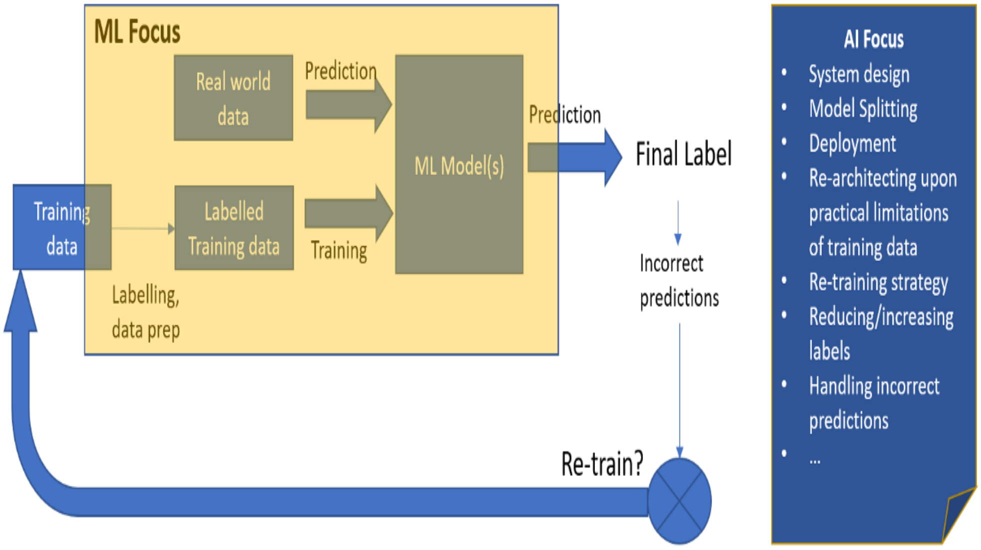
4.2.4 State-Flowchart Diagram

A state diagram is a type of diagram used in computer science and related fields to describe the behavior of systems. State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction. Many forms of state diagrams exist, which differ slightly and have different semantics.



4.3 SYSTEM DESIGN

The overall working and the design of machine learning project is explained in the below figure. Data is collected then it is processed and trained by using an appropriate learning algorithm which will predict the output based on the given input. Machine learning systems design is the process of defining the software architecture, infrastructure, algorithms, and data for a machine learning system to satisfy specified requirements. The tutorial approach has been tremendously successful in getting models off the ground.



AI Focus : Overall system

5. SAMPLE CODE

5.1 Notebook.ipynb

```
import pandas as pd #used for data manipulation
import numpy as np #used for numerical analysis
from collections import Counter as c # return counts
import seaborn as sns #used for data Visualization
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split #splits data i
n random train and test array
from sklearn.metrics import accuracy_score,mean_squared_error,
mean_absolute_error#model performance
import pickle #Python object hierarchy is converted into a byte st
ream,
from sklearn.linear_model import LinearRegression #Regresssio
n ML algorithm

data=pd.read_csv(r"C:\Users\ragha\OneDrive\Desktop\internship
 ml\FuelConsumptionCo2.csv")
data

data.head() #return you the first 5 rows values

data.tail() #return you the last 5 rows values

data.drop('MODEL',axis=1,inplace=True)  # drop is used for dro
pping the column

data.columns #return all the column names
data.columns=['Make','MODEL.1', 'Vehicle_Class', 'Engine_Size'
, 'Cylinders',
       'Transmission', 'Fuel_Type', 'Fuel_Consumption_City',
       'Fuel_Consumption_Hwy', 'Fuel_Consumption_Comb(L/100
 km)',
       'Fuel_Consumption_Comb(mpg)','CO2_Emissions']
data.columns

data.info()

data.describe()  # returns important values for continous column
data

np.unique(data.dtypes,return_counts=True)

cat=data.dtypes[data.dtypes=='O'].index.values
cat

for i in cat:
    print("Column :",i)
    print('count of classes : ',data[i].nunique())
    print(c(data[i]))
    print('*'*120)

#here we are combininng the similar types of class into one class
using where is for find
#and isin is for used for checking purpose
data["Transmission"] = np.where(data["Transmission"].isin(["A4
", "A5", "A3"]), "Automatic", data["Transmission"])
data["Transmission"] = np.where(data["Transmission"].isin(["M5
", "M6"]), "Manual", data["Transmission"])
```

```
data["Transmission"] = np.where(data["Transmission"].isin(["AS4", "AS5"]), "Automatic with Select Shift", data["Transmission"])
data["Transmission"] = np.where(data["Transmission"].isin(["AV"]), "Continuously Variable", data["Transmission"])
c(data['Transmission'])

data["Fuel_Type"] = np.where(data["Fuel_Type"]=="Z", "Premium Gasoline", data["Fuel_Type"])
data["Fuel_Type"] = np.where(data["Fuel_Type"]=="X", "Regular Gasoline", data["Fuel_Type"])
data["Fuel_Type"] = np.where(data["Fuel_Type"]=="D", "Diesel", data["Fuel_Type"])
data["Fuel_Type"] = np.where(data["Fuel_Type"]=="E", "Ethanol(E85)", data["Fuel_Type"])
data["Fuel_Type"] = np.where(data["Fuel_Type"]=="N", "Natural Gas", data["Fuel_Type"])
c(data["Fuel_Type"])

data.isnull().any()#it will return true if any columns is having null values

data.isnull().sum() #used for finding the null values

data1=data.copy()
from sklearn.preprocessing import LabelEncoder #imorting the LabelEncoding from sklearn
x='*'
for i in cat:#looping through all the categorical columns
    print("LABEL ENCODING OF:",i)
    LE = LabelEncoder()#creating an object of LabelEncoder
    print(c(data[i])) #getting the classes values before transformati
on
    data[i] = LE.fit_transform(data[i]) # trannsforming our text cla
sses to numerical values
    print(c(data[i])) #getting the classes values after transformation
    print(x*100)
a=data1.groupby('Vehicle_Class')['CO2_Emissions'].mean().sort
_values(ascending=False)[:10].reset_index()
plt.figure(figsize=(20,6))
sns.barplot(x='Vehicle_Class',y='CO2_Emissions',data=a)
corr = data.corr() #perform correlation between all continous feat
ures
plt.subplots(figsize=(16,16));
sns.heatmap(corr, annot=True, square=True) #plotting heatmap o
f correlations
plt.title("Correlation matrix of numerical features")
plt.tight_layout()
plt.show()
x = data.drop(['CO2_Emissions','Fuel_Consumption_Comb(L/10
0 km)','MODEL.1'],axis=1) #independet features
x=pd.DataFrame(x)
y = data['CO2_Emissions'] #depenent feature
y=pd.DataFrame(y)
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,ran
dom_state=2)
print(x_train.shape)
print(x_test.shape)
lr=LinearRegression() #creating object of LinearRegression mod
el
lr=lr.fit(x_train,y_train) # fitting our model
y_pred=lr.predict(x_test)
```

```
y_pred
score = lr.score(x_test,y_test)
score
from sklearn.ensemble import RandomForestRegressor

rfr=RandomForestRegressor()
rfr.fit(x_train,y_train)
rfr.score(x_test,y_test)
```

Flask

```
from flask import Flask,render_template,request
import pickle
import numpy as np

app=Flask(_name_)
lr=pickle.load(open('co2.pkl','rb'))

@app.route('/')
def home():
    return render_template("index.html")

@app.route('/rag')
def rag():
    return render_template("co2.html")

@app.route('/predict',methods=['post'])
#can use try catch or if else also
def predict():
    Modelyear=float(request.form['MODELYEAR'])
    Make=float(request.form['MAKE'])
    Model=float(request.form['MODEL'])
    VehicleClass=float(request.form['VEHICLECLASS'])
    EngineSize=float(request.form['ENGINESIZE'])
    Cylinders=float(request.form['CYLINDERS'])
    Transmission=float(request.form['TRANSMISSION'])
    FuelType=float(request.form['FUELTYPE'])

FuelConsumptionCity=float(request.form['FUELCONSUMPTIO
NCITY'])

FuelConsumptionHwy=float(request.form['FUELCONSUMPTI
ONHWY'])

FuelConsumptionComb=float(request.form['FUELCONSUMPTI
ONCOMB'])

FuelConsumptionCombMpg=float(request.form['FUELCONSU
MPTIONCOMBMPG'])


a=np.array([[Modelyear,Make,Model,VehicleClass,EngineSize,C
ylinders,Transmission,FuelType,

FuelConsumptionCity,FuelConsumptionHwy,FuelConsumptionC
omb,FuelConsumptionCombMpg]])
    print(a)

    result=lr.predict(a)
    x=result

    if(x<=250):
        return render_template('co2.html',x='Value is : {}
'.format(*x) +'    CAR IS ENVIRONMENT FRIENDLY')

    else:
        return render_template('co2.html',x='Value is : {}
'.format(*x) +'    CAR IS NOT ENVIRONMENT FRIENDLY')

if _name_ == '_main_':
    app.run()
```

HTML

```
<!DOCTYPE html>
<html>

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <link                                        rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/boots
trap.min.css"
        integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlS
AwiGgFAW/dAiS6JXm" crossorigin="anonymous">
    <style>
        select {
            width: 10%;
            padding: 8px 10px;
            border: none;
            border-radius: 4px;
            background-color: #f1f1f1;
        }

        input[type=submit] {
            background-color: #FFF8DC;

        }

        nav {
            background: black;
        }

        nav ul li {
            display: inline-block;
            padding: 0px 20px 5px 40px;
        }

        nav ul li a {
            font-size: 60%;
            text-decoration: none;
            color: white;
            transition: 0.5s;
        }

        nav ul li a:hover {
            color: white;
        }

        .bg {
            background-image:
url("https://wallpaperaccess.com/full/1372726.jpg");
```

```
            height: 80%;
            background-color: #34ebe1;
            background-position: center;
            background-repeat: no-repeat;
            background-size: cover;
        }
    </style>
</head>

<body class="bg">
    <!--<div style="background-image: url('cars.jpg');">-->
    <h1 style="text-transform: uppercase;" align="center"
>Emission Of Co2 From Cars</h1>
    <nav class="navbar navbar-inverse">
        <div class="container-fluid">
            <div class="topnav-right">
                <ul class="nav navbar-nav">
                    <li<b><a href="/">Home</a></b></li>
                </ul>
            </div>
        </div>
    </nav>
    <form action="/predict" method="POST" class="main-form
needs-validation" novalidate>
        <div class="row">
            <div class="col">
                <div class="form-group">
                    <b><label
for="MODELYEAR">MODELYEAR</label></b>
                    <input type="text" name="MODELYEAR"
id="MODELYEAR" class="form-control is-invalid">
```

```
                </div>
            </div>

            <div class="col">
                <div class="form-group">
                    <b><label for="MAKE">MAKE</label></b>
                    <select name="MAKE" id="MAKE" class="form-
control">
                        <option value="select"
default>SELECT</option>
                        <option value="3">AUDI</option>
                        <option value="5">BMW</option>
                        <option value="8">CHEVROLET</option>
                        <option value="12">FORD</option>
                        <option value="13">GMC</option>
                        <option value="26">MERCEDES-
BENZ</option>
                        <option value="30">PORSCHE</option>
                        <option value="37">TOYOTA</option>
                    </select>
                </div>
            </div>
        </div>

        <div class="row">
            <div class="col">
                <div class="form-group">
                    <b><label for="MODEL">MODEL</label></b>
                    <select name="MODEL" id="MODEL"
class="form-control">
                        <option value="select"
```

```
default>SELECT</option>
            <option value="88">ACCORD</option>
            <option value="105">BEETLE</option>
            <option value="269">F150 FFV 4X4</option>
            <option value="270">F150 FFV</option>
            <option value="283">FOCUS FFV</option>
          </select>
        </div>
      </div>

      <div class="col">
        <div class="form-group">
          <b><label
for="VEHICLECLASS">VEHICLECLASS</label></b>
          <select name="VEHICLECLASS"
id="VEHICLECLASS" class="form-control">
            <option value="select"
default>SELECT</option>
            <option value="1">COMPACT</option>
            <option value="2">FULL-SIZE</option>
            <option value="3">MID-SIZE</option>
            <option value="12">SUV-SMALL</option>
            <option value="13">SUV-STANDARD</option>
          </select>
        </div>
      </div>
    </div>

    <div class="row">
      <div class="col">
        <div class="form-group">
```

```
                    <b><label
for="ENGINESIZE">ENGINESIZE</label></b>
                    <input type="text" name="ENGINESIZE"
id="ENGINESIZE" class="form-control is-invalid">
                </div>
            </div>

            <div class="col">
                <div class="form-group">
                    <b><label
for="CYLINDERS">CYLINDERS</label></b>
                    <input type="text" name="CYLINDERS"
id="CYLINDERS" class="form-control">
                </div>
            </div>
        </div>

        <div class="row">
            <div class="col">
                <div class="form-group">
                    <b><label
for="TRANSMISSION">TRANSMISSION</label></b>
                    <select name="TRANSMISSION"
id="TRANSMISSION" class="form-control">
                        <option value="select"
default>SELECT</option>
                        <option value="3">A6</option>
                        <option value="5">A8</option>
                        <option value="12">AS6</option>
                        <option value="14">AS8</option>
                        <option value="16">AV</option>
                        <option value="21">M6</option>
                    </select>
                </div>
            </div>

            <div class="col">
                <div class="form-group">
                    <b><label
for="FUELTYPE">FUELTYPE</label></b>
                    <select name="FUELTYPE" id="FUELTYPE"
class="form-control">
                        <option value="select"
default>SELECT</option>
                        <option value="1">D</option>
                        <option value="2">E</option>
                        <option value="3">X</option>
                        <option value="4">Z</option>
                    </select>
                </div>
            </div>
        </div>

        <div class="row">
            <div class="col">
                <div class="form-group">
                    <b><label
for="FUELCONSUMPTIONCITY">FUELCONSUMPTIONCI
TY</label></b>
                    <input type="text"
name="FUELCONSUMPTIONCITY"
id="FUELCONSUMPTIONCITY"
```

```
                        class="form-control is-invalid">
                    </div>
                </div>

                <div class="col">
                    <div class="form-group">
                        <b><label
for="FUELCONSUMPTIONHWY">FUELCONSUMPTIONH
WY</label></b>
                        <input type="text"
name="FUELCONSUMPTIONHWY"
id="FUELCONSUMPTIONHWY" class="form-control">
                    </div>
                </div>
            </div>

            <div class="row">
                <div class="col">
                    <div class="form-group">
                        <b><label
for="FUELCONSUMPTIONCOMB">FUELCONSUMPTIONC
OMB</label></b>
                        <input type="text"
name="FUELCONSUMPTIONCOMB"
id="FUELCONSUMPTIONCOMB"
                            class="form-control is-invalid">
                    </div>
                </div>

                <div class="col">
                    <div class="form-group">
                    <b><label
for="FUELCONSUMPTIONCOMBMPG">FUELCONSUMPTI
ONCOMBMPG</label></b>
                    <input type="text"
name="FUELCONSUMPTIONCOMBMPG"
id="FUELCONSUMPTIONCOMBMPG" class="form-control">
                </div>
            </div>
        </div>
        <center><button type="submit" class="btn btn-
primary">Submit</button></center>
    </form>
    <center>
        <h1>{{x}}</h1>
    </center>
    </div>
</body>
</html>
```

6. TESTING

6.1 TESTING

Testing is the process where the test data is prepared and is used for testing the modules individually and later the validation given for the fields. Then the system testing takes place which makes sure that all components of the system property functions as a unit. The test data should be chosen such that it passed through all possible condition. Actually, testing is the state of implementation which aimed at ensuring that the system works accurately and efficiently before the actual operation commence. The following is the description of the testing strategies, which were carried out during the testing period.

6.2 TYPES OF TESTING

6.2.1 Unit Testing

It focuses on the smallest unit of software design. In this, we test an individual unit or group of interrelated units. It is often done by the programmer by using sample input and observing its corresponding outputs.

6.2.2 Integration Testing

The objective is to take unit tested components and build a program structure that has been dictated by design. Integration testing is testing in which a group of components is combined to produce output.

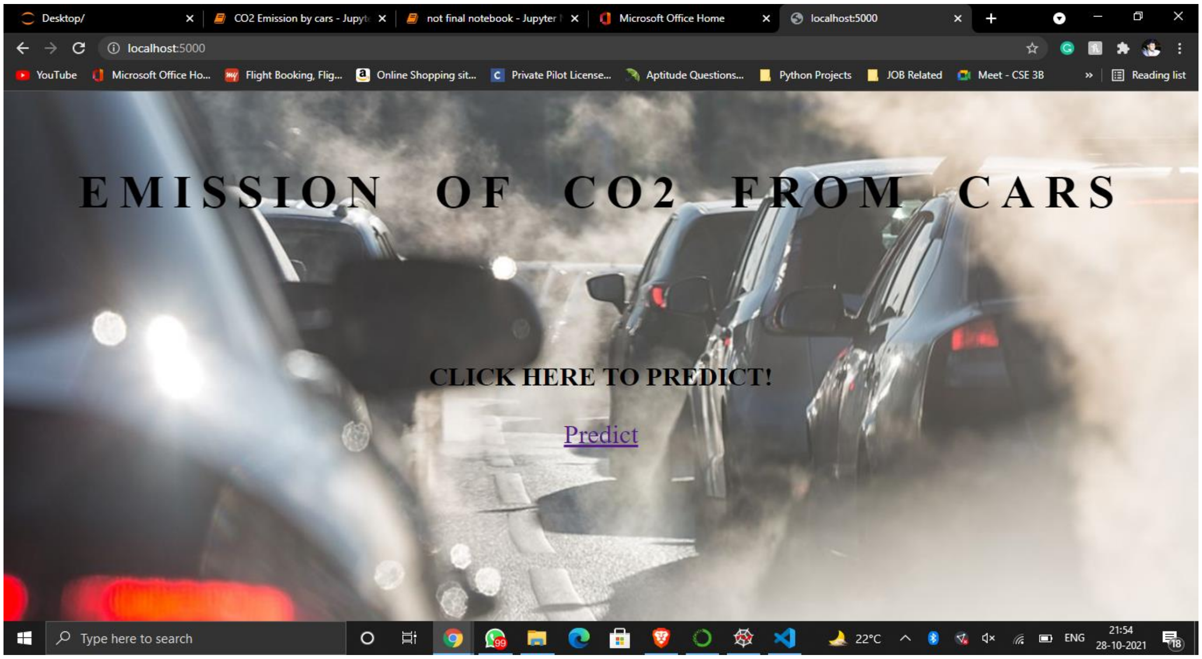
6.2.3 System Testing

This software is tested such that it works fine for the different operating systems. It is covered under the black box testing technique.

6.3 TEST CASES

Test Case 1
Input all the details related to the car by filling out form type web application
Test Case 2
Working of submit button.
Test Case 3
Predicting the emission of carbon dioxide and comparison of the predicted value with threshold value.
Test Case 4
If predicted value is less than or equal to 250g/km then output should be “Car is environmental friendly.”
Test Case 5
If predicted value is greater than 250g/km then output should be “Car is not environmental friendly.”
Test Case 6
Navigation to home page and prediction page.

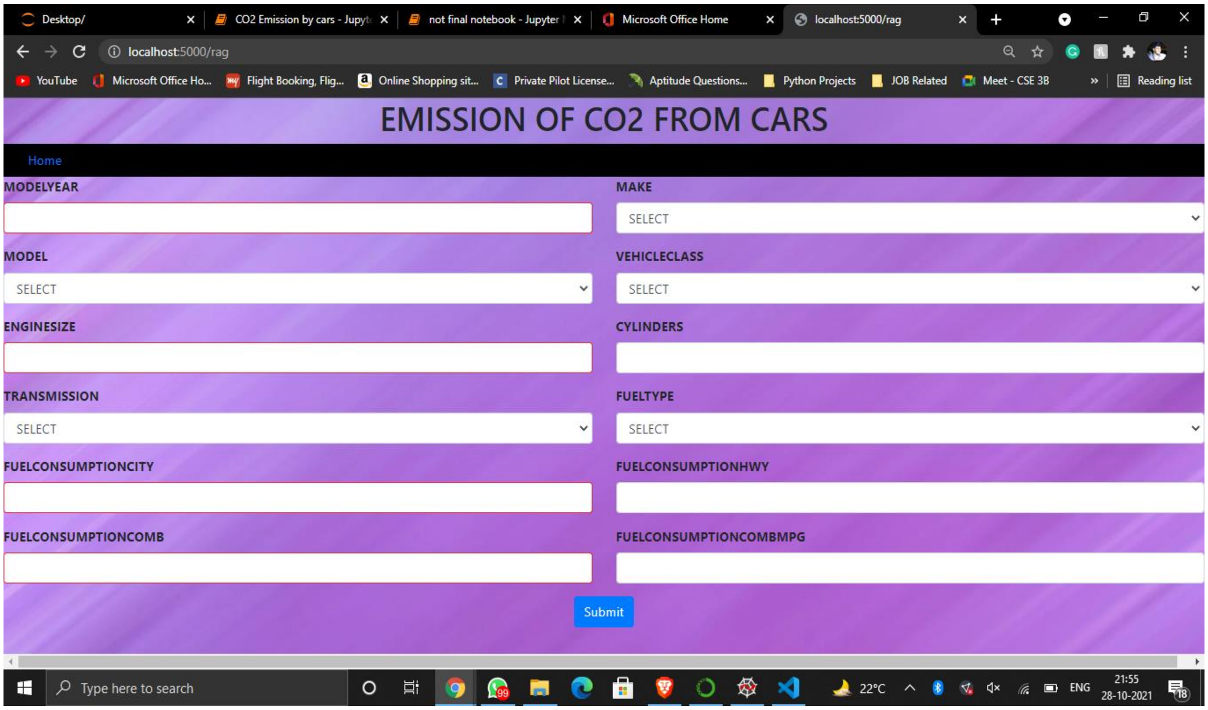
7. OUTPUT SCREENS



EMISSION OF CO2 FROM CARS

CLICK HERE TO PREDICT!

Predict



EMISSION OF CO2 FROM CARS

Home

MODELYEAR

MAKE: SELECT

MODEL: SELECT

VEHICLECLASS: SELECT

ENGINESIZE

CYLINDERS

TRANSMISSION: SELECT

FUELTYPE: SELECT

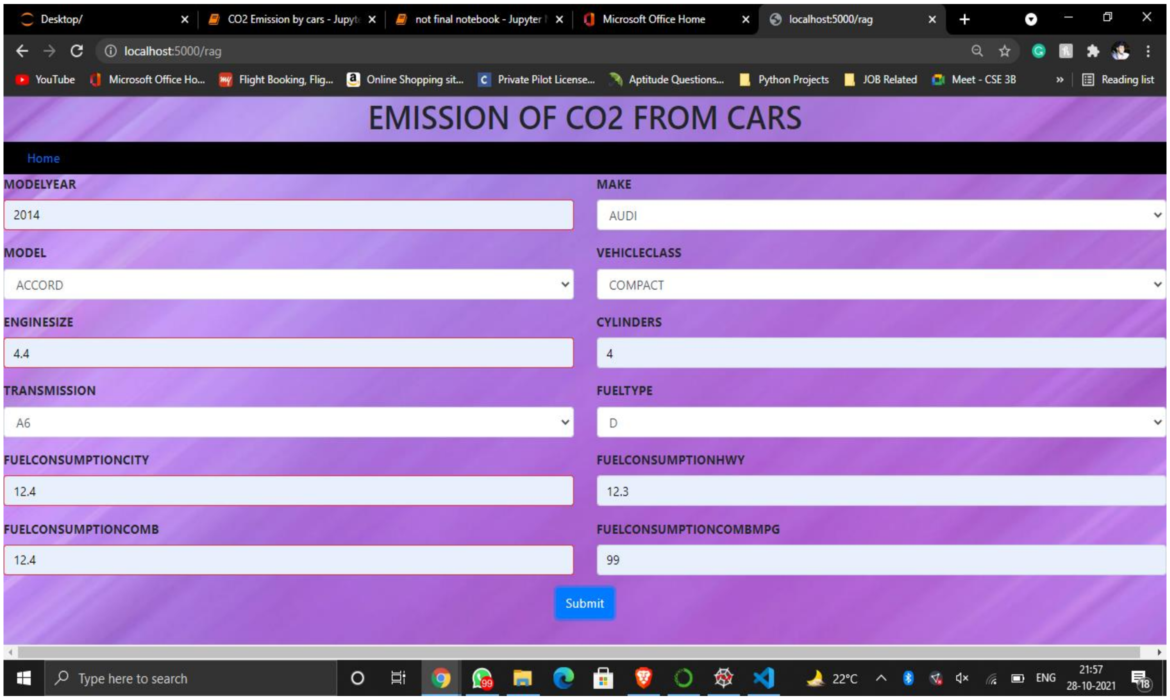
FUELCONSUMPTIONCITY

FUELCONSUMPTIONHWY

FUELCONSUMPTIONCOMB

FUELCONSUMPTIONCOMBMPG

Submit



EMISSION OF CO2 FROM CARS

Home

MODELYEAR: 2014

MAKE: AUDI

MODEL: ACCORD

VEHICLECLASS: COMPACT

ENGINESIZE: 4.4

CYLINDERS: 4

TRANSMISSION: A6

FUELTYPE: D

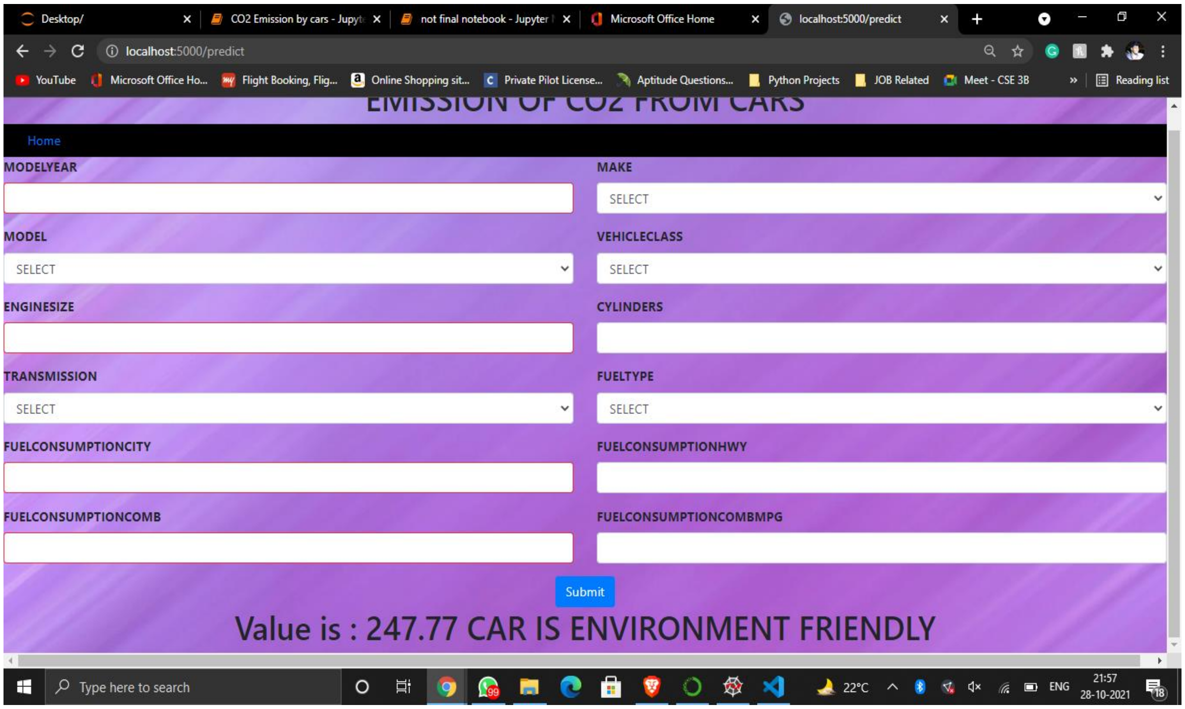
FUELCONSUMPTIONCITY: 12.4

FUELCONSUMPTIONHWY: 12.3

FUELCONSUMPTIONCOMB: 12.4

FUELCONSUMPTIONCOMBMPG: 99

Submit



EMISSION OF CO2 FROM CARS

Home

MODELYEAR:

MAKE: SELECT

MODEL: SELECT

VEHICLECLASS: SELECT

ENGINESIZE:

CYLINDERS:

TRANSMISSION: SELECT

FUELTYPE: SELECT

FUELCONSUMPTIONCITY:

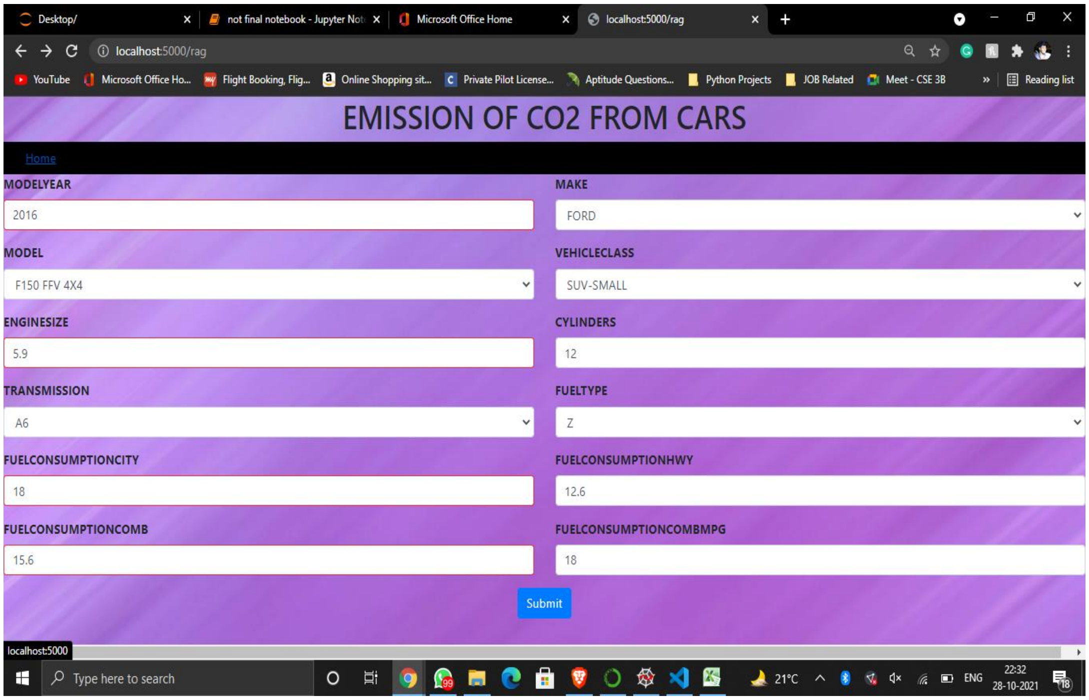
FUELCONSUMPTIONHWY:

FUELCONSUMPTIONCOMB:

FUELCONSUMPTIONCOMBMPG:

Submit

Value is : 247.77 CAR IS ENVIRONMENT FRIENDLY

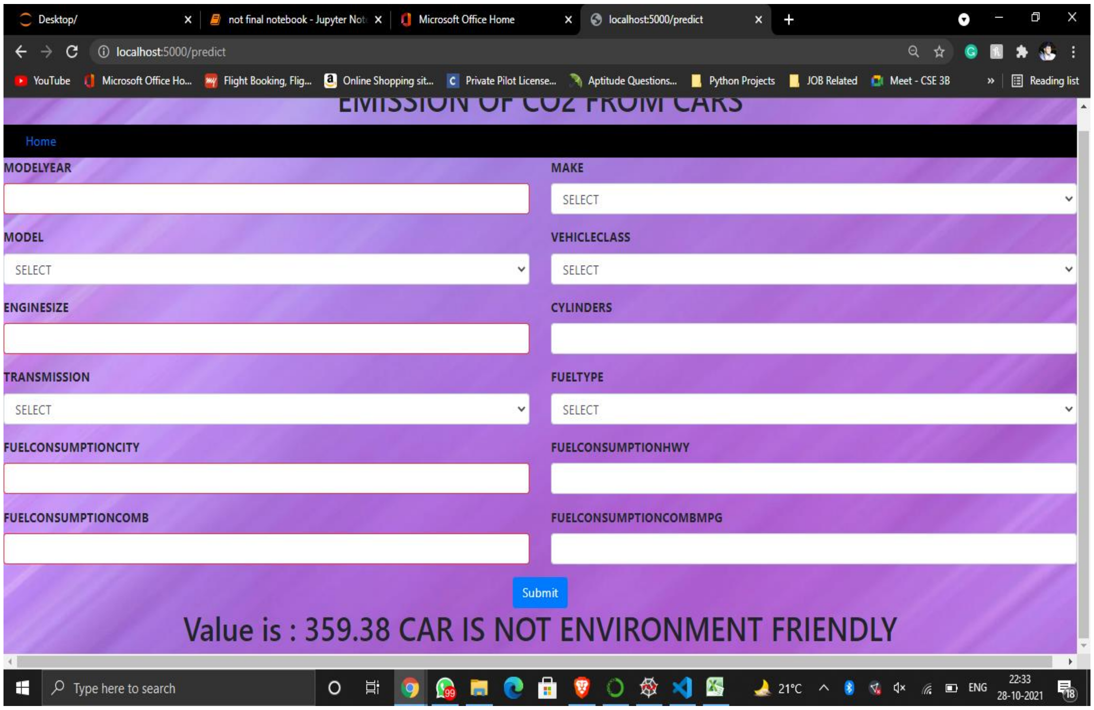


EMISSION OF CO2 FROM CARS

Home

MODELYEAR	MAKE
2016	FORD
MODEL	VEHICLECLASS
F150 FFV 4X4	SUV-SMALL
ENGINESIZE	CYLINDERS
5.9	12
TRANSMISSION	FUELTYPE
A6	Z
FUELCONSUMPTIONCITY	FUELCONSUMPTIONHWY
18	12.6
FUELCONSUMPTIONCOMB	FUELCONSUMPTIONCOMBMPG
15.6	18

Submit



EMISSION OF CO2 FROM CARS

Home

MODELYEAR	MAKE
	SELECT
MODEL	VEHICLECLASS
SELECT	SELECT
ENGINESIZE	CYLINDERS
TRANSMISSION	FUELTYPE
SELECT	SELECT
FUELCONSUMPTIONCITY	FUELCONSUMPTIONHWY
FUELCONSUMPTIONCOMB	FUELCONSUMPTIONCOMBMPG

Submit

Value is : 359.38 CAR IS NOT ENVIRONMENT FRIENDLY

8. CONCLUSION

8.1 CONCLUSION

The purpose and objective of predicting emission of carbon dioxide using machine learning project has been achieved. This project plays crucial role in the aspects of reducing pollution and global warming levels in the environment. This will give better chances and direction in future of venture.

8.2 FURTHER ENHANCEMENTS

- In version 2.0 and for future enhancement we will be adding number plate detection and getting the details of the cars.
- After getting the details we will be predicting the values.
- So instead of giving all the car details we can just give the number plate details which will return the predicted co2 values and last time it went under pollution check.
- Therefore, the purpose of this project is to predict the co2 values and finding out whether the cars are environmental friendly or not.

9. BIBLIOGRAPHY

- “Hands-On Machine Learning with Scikit-Learn, Keras, and Tensor Flow”, 2nd Edition,by Aurélien Géron, Released September 2019
- https://www.oreilly.com/library/view/hands-on-machine-learning/
- “Introduction to Machine Learning with Python: A Guide for Data Scientists”, 1st Edition by Andreas C. Müller (Author), Sarah Guido **.**
- “Machine Learning: The New AI”(The MIT Press Essential Knowledge series) by Ethem Alpaydin
- “Pattern Recognition and Machine Learning”, Christopher M. Bishop
- https://theicct.org/sites/default/files/publications/India_fuel_consumption_standards_20180925.pdf
- https://www.kaggle.com/debajyotipodder/co2-emission-by-vehicles
- https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/
- https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/
- https://towardsdatascience.com/7-of-the-most-commonly-used-regression-algorithms-and-how-to-choose-the-right-one-fc3c8890f9e3
- https://github.com/Raghav121000/Smartbridge
- https://iopscience.iop.org/article/10.1088/1742-6596/1142/1/012012/meta
- https://link.springer.com/article/10.1007/s10515-020-00277-4
- https://technicalpublications.org/te-sem-i-it-2015-course
- https://books.google.co.in/books?hl=en&lr=&id=XAqhDwAAQBAJ&oi=fnd&pg=PA19&dq=ml+supervised+learning+algorithms&ots=r2Io7RAhLr&sig=w-U3FWO_APgUjOTx9GxPiGzrp4E#v=onepage&q=ml%20supervised%20learning%20algorithms&f=false
- https://proceedings.mlr.press/v139/ding21b.html
- https://ieeexplore.ieee.org/abstract/document/8663155

10. APPENDICES

10.1 SOFTWARE USED

10.1.1 Visual Studio Code

- Visual Studio Code combines the simplicity of a source code editor with powerful developer tooling, like IntelliSense code completion and debugging.
- Visual Studio Code supports macOS, Linux, and Windows - so you can hit the ground running, no matter the platform.
- Visual Studio Code features a lightning fast source code editor, perfect for day-to-day use. With support for hundreds of languages, VS Code helps you be instantly productive with syntax highlighting, bracket-matching, auto-indentation, box-selection, snippets, and more. Intuitive keyboard shortcuts, easy customization and community-contributed keyboard shortcut mappings let you navigate your code with ease.
- VS Code includes enriched built-in support for Node.js development with JavaScript and TypeScript, powered by the same underlying technologies that drive Visual Studio. VS Code also includes great tooling for web technologies such as JSX/React, HTML, CSS, SCSS, Less, and JSON.

10.1.2 Jupyter Notebook:

- The jupyter notebook is a server client application that allows editing and running notebook documents via a web browser. The jupyternotebook can be executed on a local desktop requiring no internet access or can be installed on a remote server and accessed through the internet.
- In addition to displaying/editing/running notebook documents, the jupyter notebook has a dashboard, control panel showing local files and allowing to open notebook documents or shutting down their kernels.

10.2 METHODOLOGIES USED

Predicting emission of CO2 from cars using machine learning is a System which uses Software Prototyping Methodology for its development. In this method we first generate a model of the product which may not be a complete product and shown to the user. The changes again are made in the further development and again a prototype is made. This cycle is done till the customer satisfies. The main Advantage of prototyping is that the developer gets an idea whether the product could be developed or not.

Software Prototyping has the following stages:

Basic Requirement Identification:

This step involves understanding the very basics product requirements especially in terms of user interface. The more intricate details of the internal design and external aspects like performance and security can be ignored at this stage.

Developing the initial Prototype:

The initial Prototype is developed in this stage, where the very basic requirements are showcased and user interfaces are provided. These features may not exactly work in the same manner internally in the actual software developed. While, the work around are used to give the same look and feel to the customer in the prototype developed.

Review of the Prototype:

The prototype developed is then presented to the customer and the other important stakeholders in the project. The feedback is collected in an organized manner and used for further enhancements in the product under development.

Revise and Enhance the Prototype:

The feedback and the review comments are discussed during this stage and some negotiations happen with the customer based on factors like – time and budget constraints and technical feasibility of the actual implementation. The changes accepted are again incorporated in the new Prototype developed and the cycle repeats until the customer expectations are met.

Prototypes can have horizontal or vertical dimensions. A Horizontal prototype displays the user interface for the product and gives a broader view of the entire system, without concentrating on internal functions. A Vertical prototype on the other side is a detailed elaboration of a specific function or a sub system in the product.

10.3 TESTING METHODS USED

10.3.1 Functionality Testing

This method is used to make sure that web app functions are working without any bugs. Functionality testing will include whether all links are working, testing forms in all pages, html validation or CSS, testing the security of the database. Test cases should cover the invalid inputs should give an error message.

10.3.2 Usability Testing

In this method is suitable for the applications that are intended to manual process. This testing will verify aspects like navigation, mapping of site pages, avoid over crowded information and also application need to follow all global conventions.

10.3.3 Web UI Testing

The important interfaces in application are application, web and database server this will make sure that all components in application have proper connection and interaction between all servers need to work properly.

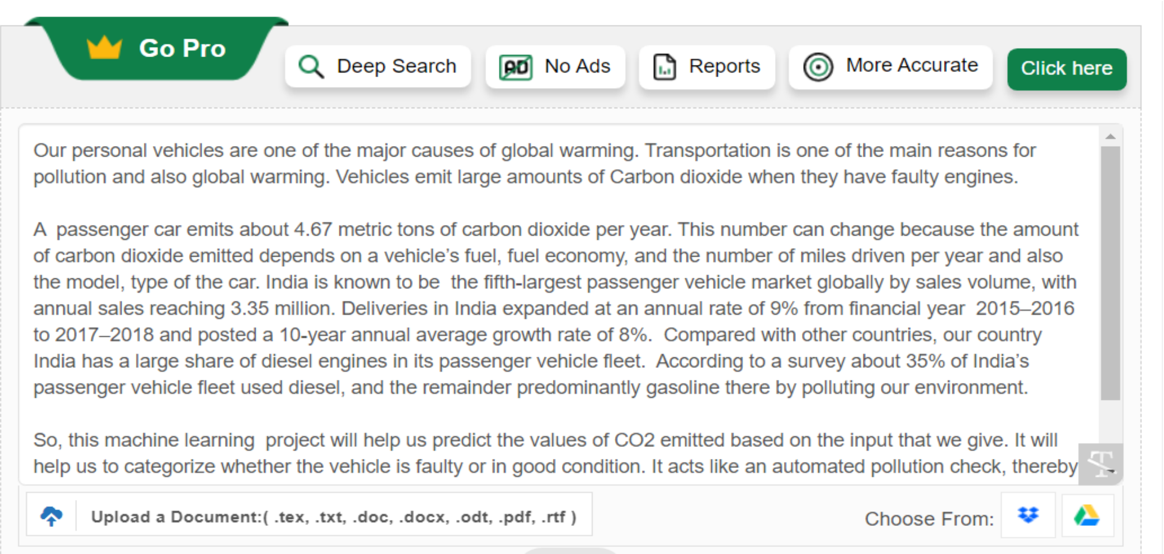
10.3.4 Compatibility Testing

Compatibility of application is the most important things that one should consider it during performing testing the application. Compatibility testing will verify the website for browser, O.S, mobile browsing compatibility and the printing options.

10.3.5 Performance Testing

Performance testing will determine the performance of a web application under the different scenarios. This testing includes stress, scalability and load testing.

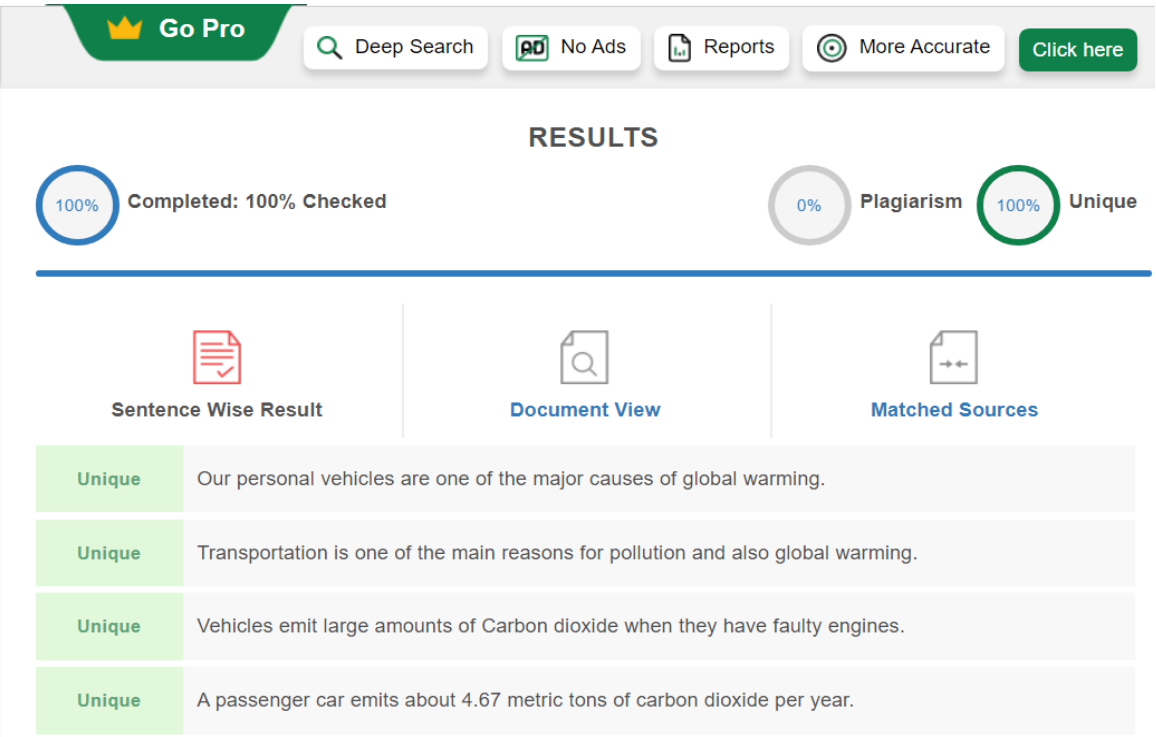
11. PLAGIARISM TESTING



Our personal vehicles are one of the major causes of global warming. Transportation is one of the main reasons for pollution and also global warming. Vehicles emit large amounts of Carbon dioxide when they have faulty engines.

A passenger car emits about 4.67 metric tons of carbon dioxide per year. This number can change because the amount of carbon dioxide emitted depends on a vehicle's fuel, fuel economy, and the number of miles driven per year and also the model, type of the car. India is known to be the fifth-largest passenger vehicle market globally by sales volume, with annual sales reaching 3.35 million. Deliveries in India expanded at an annual rate of 9% from financial year 2015–2016 to 2017–2018 and posted a 10-year annual average growth rate of 8%. Compared with other countries, our country India has a large share of diesel engines in its passenger vehicle fleet. According to a survey about 35% of India's passenger vehicle fleet used diesel, and the remainder predominantly gasoline there by polluting our environment.

So, this machine learning project will help us predict the values of CO2 emitted based on the input that we give. It will help us to categorize whether the vehicle is faulty or in good condition. It acts like an automated pollution check, thereby



Unique	Our personal vehicles are one of the major causes of global warming.
Unique	Transportation is one of the main reasons for pollution and also global warming.
Unique	Vehicles emit large amounts of Carbon dioxide when they have faulty engines.
Unique	A passenger car emits about 4.67 metric tons of carbon dioxide per year.