SESHADRI RAO GUDLAVALLERU ENGINEERING COLLEGE

Trip Based Modeling Of Fuel Consumption In Modern Fleet Vehicles Using IBM Watson

SUBMITTED BY: Chandra shekhar Borra
PIN:19481A0311

CONTENT

1.INTRODUCTION

1.1 ABSTRACT

The use of this project. What can be achieved using this.

1.2 OVERVIEW

A brief description about your project

2 LITERATURE SURVEY

2.1 Existing problem

Existing approaches or method to solve this problem

2.2 Proposed solution

What is the method or solution suggested by you

3 THEORITICAL ANALYSIS

3.1 Block diagram

Diagrammatic overview of the project.

3.2 Hardware / Software requirements

Hardware and software requirements of the project

4 EXPERIMENTAL INVESTIGATIONS

Analysis or the investigation made while working on the solution

5 FLOWCHART

Diagram showing the control flow of the solution

6 RESULT

Final findings (Output) of the project along with screenshots.

7 CONCLUSION

Conclusion summarizing the entire work and finding

1. INTRODUCTION

Abstract :

Heavy-duty trucks contribute approximately 20% of fuel consumption . The fuel economy of heavy-duty vehicles (HDV) is affected by several real-world parameters like road parameters, driver behavior, weather conditions, and vehicle parameters, etc. Although modern vehicles comply with emissions regulations, potential malfunction of the engine, regular wear and tear, or other factors could affect vehicle performance. Predicting fuel consumption per trip based on dynamic on-road data can help the automotive industry to reduce the cost and time for on-road testing. Data modeling can easily help to diagnose the reason behind fuel consumption with a knowledge of input parameters. In this paper, an artificial neural network (ANN) was implemented to model fuel consumption in modern heavy-duty trucks for predicting the total and instantaneous fuel consumption of a trip based on very few key parameters, such as engine load (%), engine speed (rpm), and vehicle speed (km/h). Instantaneous fuel consumption data can help to predict patterns in fuel consumption for optimized fleet operations. In this work, the data used for modeling was collected at a frequency of 1Hz during on-road testing of modern heavy-duty vehicles (HDV) at the West Virginia University Center for Alternative Fuels Engines and Emissions (WVU CAFEE) using the portable emissions monitoring system (PEMS). The performance of the artificial neural network was evaluated using mean absolute error (MAE) and root mean square error (RMSE). The model was further evaluated with data collected from a vehicle on-road trip. The study shows that artificial neural networks performed slightly better than other machine learning techniques such as linear regression (LR), and random forest (RF), with high R-squared (R2) and lower root mean square error.

OVERVIEW:

Ability to model and predict the fuel consumption is vital in enhancing fuel economy of vehicles and preventing fraudulent activities in fleet management. Fuel consumption of a vehicle depends on several internal & external factors However, not all these factors may be measured or available for the fuel consumption analysis.

The main aim of the project is to build Machine Learning algorithm to predict the fuel consumption of fleet vehicles based on the gas type . A web application is built which is integrated with ML mode

LITERATURE SURVEY:

2.1 Existing problem

Nowdayas people faces a problems like How much fuel the vehical consumps to travel a certain distance and how much fuel we want to fill in the tank etc ... They faces a problems like

1.Sudden break down of the vehical due to lack of fuel

2.lack of fuel stations

3.Waste of time

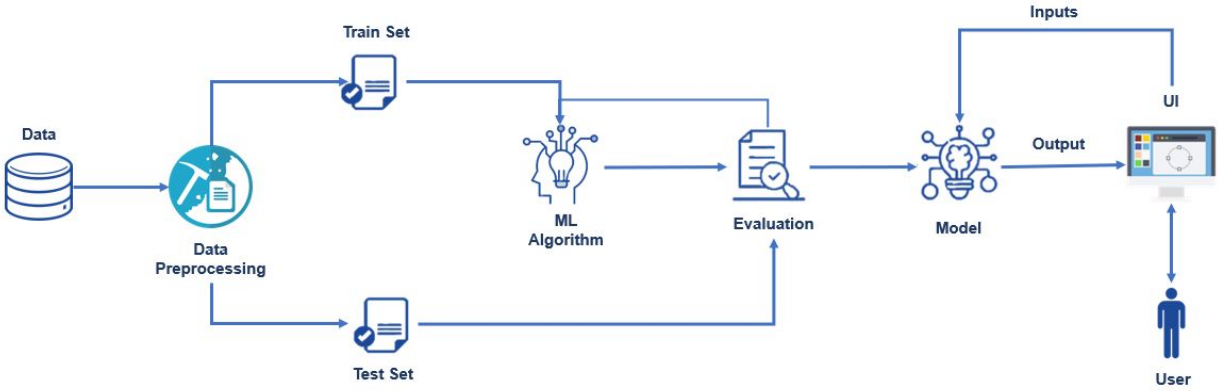
2.2 Proposed solution

Due to above problems this app provides the how much fuel will the vehical consumps to travel a certain distance , how much fuel you have to fill.. If you give the distance surroundings temperature and the type of fuelit will give you the amount of fuel you have to pour in the tank.

All above are free of cost by using a app..it can reduce the tension and can ride comfartably

3 THEROTICAL ANALYSIS

3.1 BLOCK DIAGRAM:



3.2 Hardware / Software Requirements

Recommended System Requirements

● Processors:Intel® CoreTM i5 processor 4300M at 2.60 GHz or2.59 GHz (1socket, 2 cores, 2 threads per core), 8 GB ofDRAMIntel® Xeon® processor E5- 2698 v3 at 2.30 GHz (2sockets, 16 cores each, 1 thread per core), 64 GB of DRAMIntel® Xeon PhiTM processor 7210 at 1.30 GHz (1 socket,64 cores, 4 threads per core), 32 GB of DRAM, 16 GB ofMCDRAM (flat mode enabled)
● Disk space: 2 to 3 GB

● Operating systems: Windows® 10, macOS*, and Linux*Minimum System Requirements

● Processors: Intel Atom® processor or Intel® CoreTM i3 processor

● Disk space: 1 GB

● Operating systems: Windows* 7 or later, macOS, and Linux

● Python* versions: 3.9

Software requirements:

anaconda navigator:

Anaconda is an open-source distribution for python and R. It isused for data science, machine learning, deep learning, etc. With theavailability of more than 300 libraries for data science, it becomesfairly optimal for any programmer to work on anaconda for data science.

Pycharm:

PyCharm is a dedicated Python Integrated DevelopmentEnvironment (IDE) providing a wide range of essential tools forPython developers, tightly integrated to create a convenientenvironment for productive Python, web, and data science development

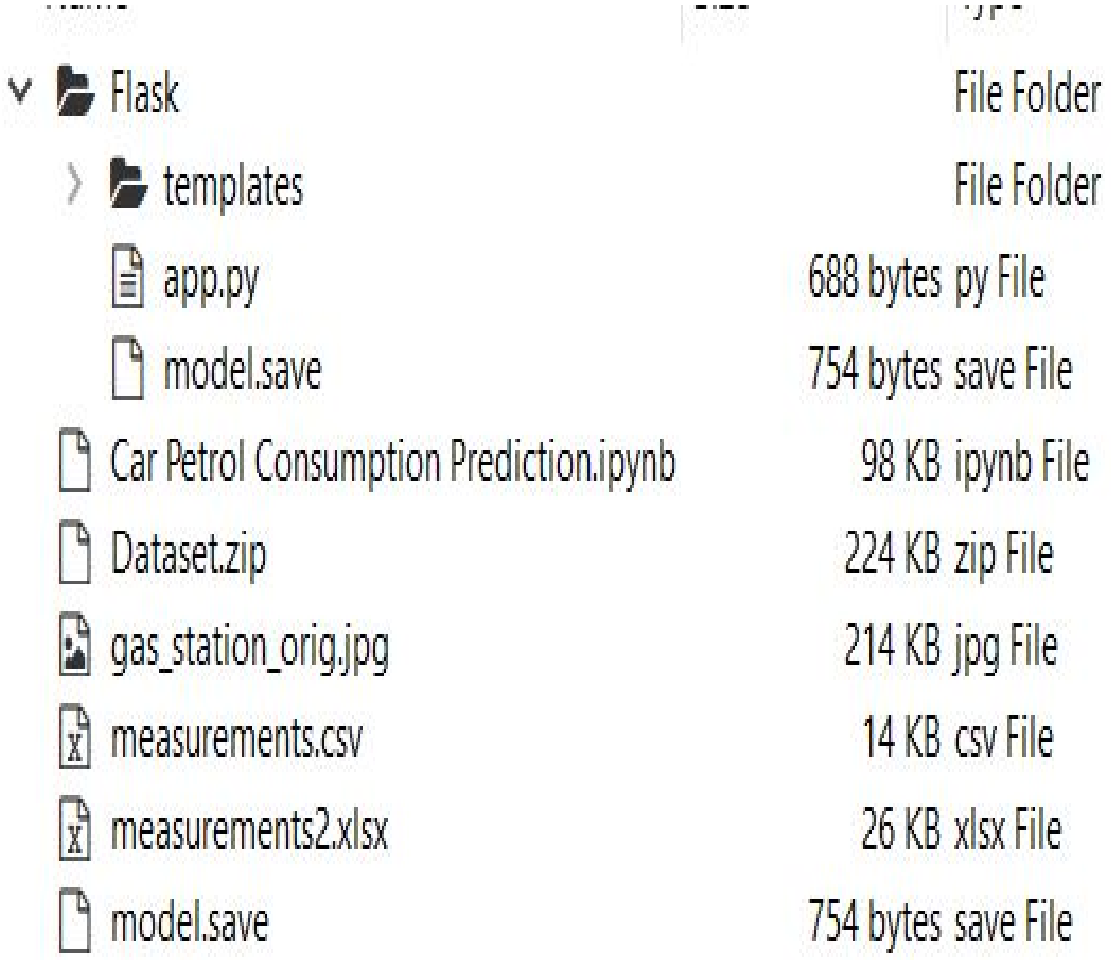
4 EXPERIMENTAL INVESTIGATIONS

Logistic Regression:

This type of statistical model (also known as logit model) is often used for classification and predictive analytics. Logistic regression estimates the probability of an event occurring, such as voted or didn't vote, based on a given dataset of independent variables. Since the outcome is a probability, the dependent variable is bounded between 0 and 1. In logistic regression, a logit transformation is applied on the odds—that is, the probability of success divided by the probability of failure. Linear Regression: Linear regression models are used to identify the relationship between a continuous dependent variable and one or more independent variables. When there is only one independent variable and one dependent variable, it is known as simple linear regression, but as the number of independent variables increases, it is referred to as multiple linear regression. For each type of linear regression, it seeks to plot a line of best fit through a set of data points, which is typically calculated using the least squares method.

5 FLOW CHART

PROJECT STRUCTURE:



•Car Petrol Consumption Prediction.ipynb is the jupyter notebook file where the model is built.

•Dataset.zip is the dataset file used in this project

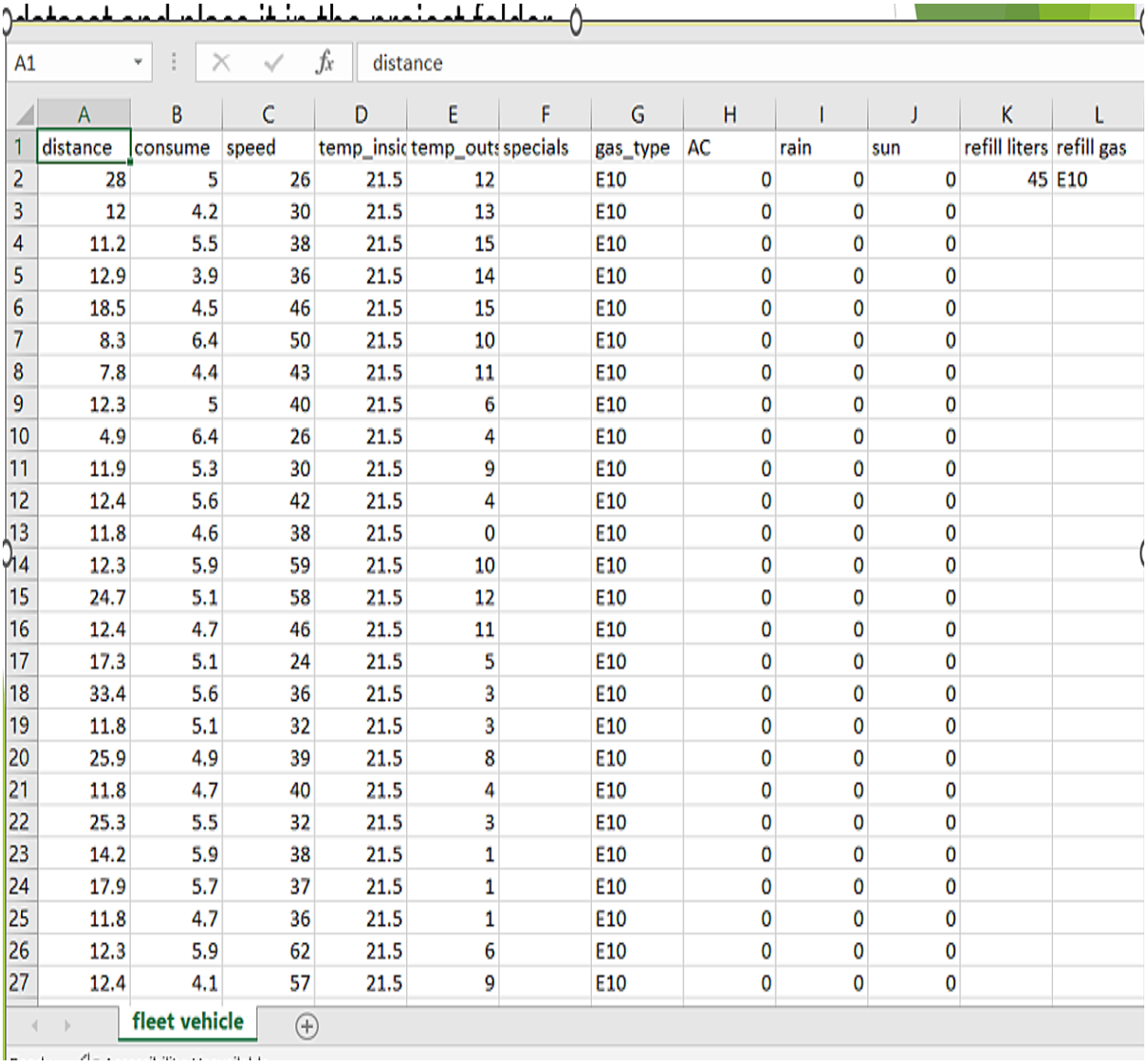
•model.save is the model file that generates when the notebook file is executed.

•Flask folder is the application folder where the web application and server-side program are present.

•Measurements.csv & measurements.xlsx are the dataset files.

DATA COLLECTION

For any Machine learning project, data is the primary source. Download the dataset and place it in the project folder.



A1 | distance

	A	B	C	D	E	F	G	H	I	J	K	L
1	distance	consume	speed	temp_insic	temp_outs	specials	gas_type	AC	rain	sun	refill liters	refill gas
2	28	5	26	21.5	12		E10	0	0	0	45	E10
3	12	4.2	30	21.5	13		E10	0	0	0		
4	11.2	5.5	38	21.5	15		E10	0	0	0		
5	12.9	3.9	36	21.5	14		E10	0	0	0		
6	18.5	4.5	46	21.5	15		E10	0	0	0		
7	8.3	6.4	50	21.5	10		E10	0	0	0		
8	7.8	4.4	43	21.5	11		E10	0	0	0		
9	12.3	5	40	21.5	6		E10	0	0	0		
10	4.9	6.4	26	21.5	4		E10	0	0	0		
11	11.9	5.3	30	21.5	9		E10	0	0	0		
12	12.4	5.6	42	21.5	4		E10	0	0	0		
13	11.8	4.6	38	21.5	0		E10	0	0	0		
14	12.3	5.9	59	21.5	10		E10	0	0	0		
15	24.7	5.1	58	21.5	12		E10	0	0	0		
16	12.4	4.7	46	21.5	11		E10	0	0	0		
17	17.3	5.1	24	21.5	5		E10	0	0	0		
18	33.4	5.6	36	21.5	3		E10	0	0	0		
19	11.8	5.1	32	21.5	3		E10	0	0	0		
20	25.9	4.9	39	21.5	8		E10	0	0	0		
21	11.8	4.7	40	21.5	4		E10	0	0	0		
22	25.3	5.5	32	21.5	3		E10	0	0	0		
23	14.2	5.9	38	21.5	1		E10	0	0	0		
24	17.9	5.7	37	21.5	1		E10	0	0	0		
25	11.8	4.7	36	21.5	1		E10	0	0	0		
26	12.3	5.9	62	21.5	6		E10	0	0	0		
27	12.4	4.1	57	21.5	9		E10	0	0	0		

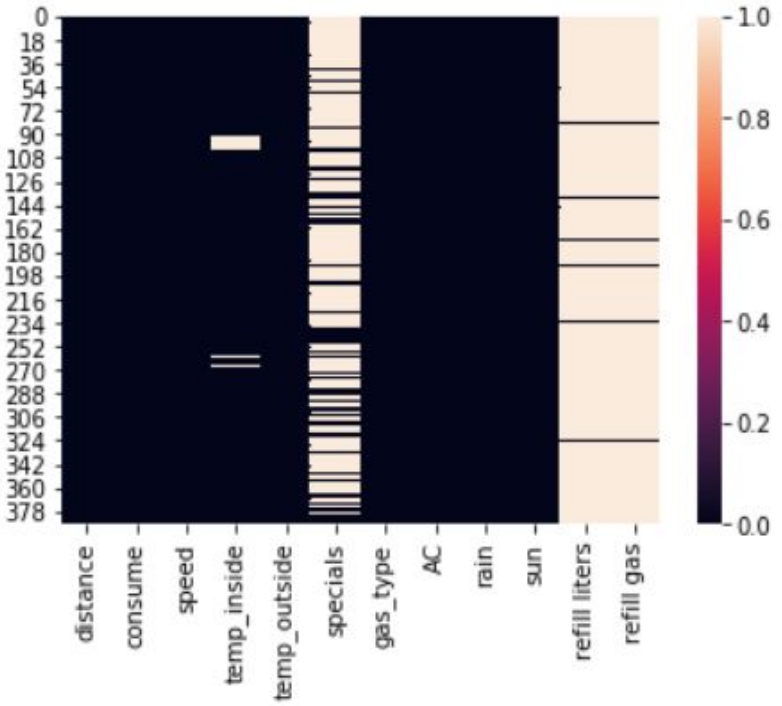
fleet vehicle

Check Null Values

For checking the null values, df.isnull() function is used. To sum those null values we use .sum() function to it. To visualize the null values heatmap() and barplot() from seaborn package is used.

```
In [3]:
1 import seaborn as sns
2 sns.heatmap(df.isnull())
```

```
Out[3]: <matplotlib.axes._subplots.AxesSubplot at 0x2c192931c88>
```



In [6]:
```
1 temp_inside_mean=np.mean(df['temp_inside'])
```

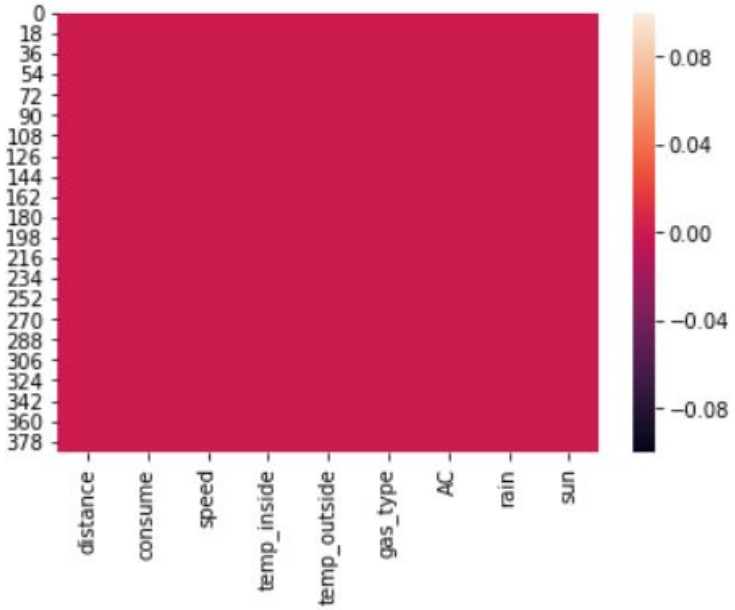
In [7]:
```
1 print(temp_inside_mean)
```

```
21.929521276595743
```

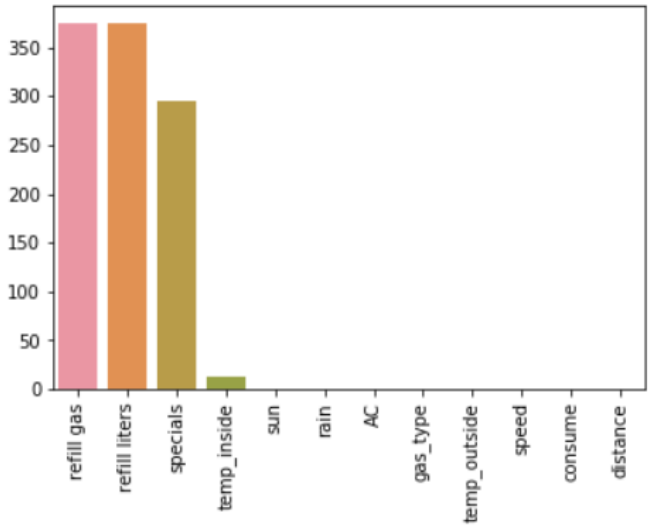
In [8]:
```
1 df['temp_inside'].fillna(temp_inside_mean,inplace=True)
```

In [9]:
```
1 sns.heatmap(df.isnull())
```

Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x2174356b7f0>



In [4]:
```
null_values=df.isnull().sum().sort_values(ascending=False)
ax=sns.barplot(null_values.index,null_values.values)
ax.set_xticklabels(ax.get_xticklabels(),rotation=90)
import matplotlib.pyplot as plt
plt.show()
```



Seperating Independent And Dependent Variables

In [10]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
l=LinearRegression()
```

In [11]:
```
x=df.drop(['consume','gas_type'],axis=1)
```

In [12]:
```
y=df['consume']
```

In [13]:
```
x.columns
```

Out[13]:
```
Index(['distance', 'speed', 'temp_inside', 'temp_outside', 'AC', 'rain',
       'sun'],
      dtype='object')
```

In [14]:
```
x=x.values
y=y.values
```

Applying Linear Regression

Now we are going to create our model with linear regression. As an initial step we have to initialize the linear model. Then train the model with fit() method. Now our model is trained and to test the model predict() method is used. To find the loss of linear regression model

mean_squared_error and mean_absolute_error are used.

Link: https://www.geeksforgeeks.org/ml-linear-regression/#:~:text=Linear%20Regression%20is%20a%20machine,relationship%20between%20variables%20and%20forecasting.

```
In [16]:
1 l.fit(x_train,y_train)
Out[16]: LinearRegression()

In [17]:
1 x_train.shape
Out[17]: (271, 7)

In [18]:
1 y_pred=l.predict(x_test)

In [19]:
1 print(l.coef_,l.intercept_)
[ 0.00523674 -0.02371772 -0.14711979 -0.03724498  0.41456804  0.61676684
 -0.06407861] 9.389308142257136

In [20]:
1 from sklearn import metrics
2 print(metrics.mean_squared_error(y_test,y_pred))
3 print(metrics.mean_absolute_error(y_test,y_pred))
4 print(np.sqrt(metrics.mean_squared_error(y_test,y_pred)))
0.7424532609047081
0.6635761182069623
0.8616572757800564
```

Build The Python Flask App

Load the saved model. 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.

```
1 from flask import Flask, request,render_template
2 import joblib
3 app = Flask(__name__)
4 model = joblib.load("model.save")
5
6
7 app = Flask(__name__)
8
```

Load the html page

```python
@app.route('/')
def predict():
    return render_template('Manual_predict.html')
```

Build An HTML Page

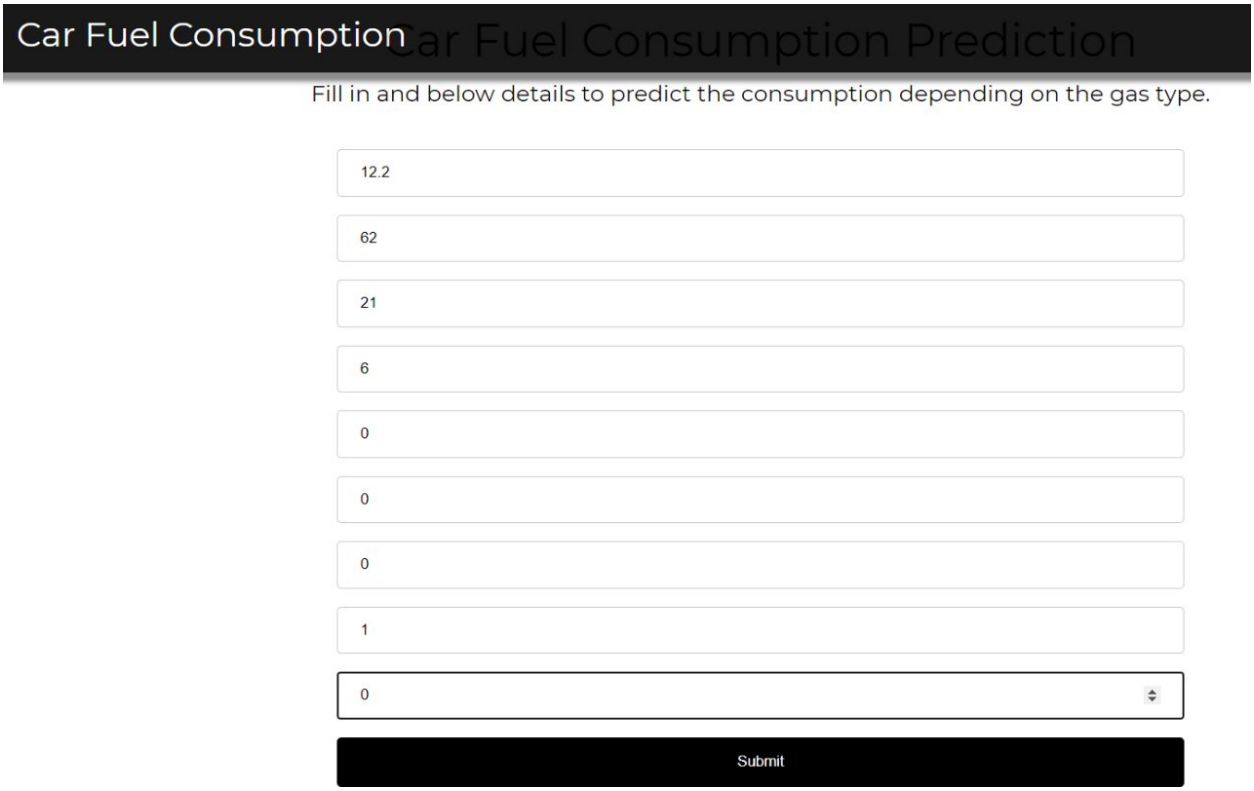
We Build an HTML page to take the values from the user in a form and upon clicking on the predict button we get the fuel consumption predicted.

```html
<html>
<head>
<title>
    Prediction
</title>
<link href='https://fonts.googleapis.com/css?family=Montserrat' rel='stylesheet'>
<style>
        * {
            box-sizing: border-box;
        }

        body {
            font-family: 'Montserrat' ;
        }

        .header {
            top:0;
            margin:0px;
            left: 0px;
            right: 0px;
            position: fixed;
            background-color: black;
            color: white;
            box-shadow: 0px 8px 4px grey;
            overflow: hidden;
            padding: 15px;
            font-size: 2vw;
            width: 100%;
            text-align: left;
            padding-left: 100px;
            opacity:0.9;
        }
        .header_text{
            font-size:40px;
            text-align:center;
        }
        .content{
        margin-top:100px;
        }
```

```
        .text{
            font-size:20px;
            margin-top:10px;
            text-align:center;
        }
        input[type=number], select {
  width: 50%;
  padding: 12px 20px;
  margin: 8px 0;
  display: inline-block;
  border: 1px solid #ccc;
  border-radius: 4px;
  box-sizing: border-box;
}

input[type=submit] {
  width: 50%;
  background-color: #000000;
  color: white;
  padding: 14px 20px;
  margin: 8px 0;
  border: none;
  border-radius: 4px;
  cursor: pointer;
}

input[type=submit]:hover {
  background-color: #5d6568;
  color:#ffffff;
  border-color:black;
}
form{
margin-top:20px;
}
.result{
color:black;
margin-top:30px;
margin-bottom:20px;
font-size:25px;
color:red;
}
```

```
</style>
</head>
<body align=center>
<div class="header">
        <div>Car Fuel Consumption </div>
</div>
<div class="content">
<div class="header_text">Car Fuel Consumption Prediction</div>
<div class="text">Fill in and below details to predict the consumption depending on the gas type.</div>
<div class="result">
  {{ prediction_text }}
  </div>
<form action="{{ url_for('y_predict') }}" method="POST">
    <input type="number" step= "any" id="distance" name="distance" placeholder="distance(km)">
    <input type="number" id="speed" name="speed" placeholder="speed(km/h)">
    <input type="number"  id="temp_inside" name="temp_insidet" placeholder="temp_inside(°C)">
    <input type="number"  id="temp_outside" name="temp_outside" placeholder="temp_outside(°C)">
    <input type="number"  id="AC" name="AC" placeholder="AC">
    <input type="number"  id="rain" name="rain" placeholder="rain">
    <input type="number"  id="sun" name="sun" placeholder="sun">
    <input type="number"  id="E10" name="E10" placeholder="E10">
    <input type="number"  id="SP98" name="SP98" placeholder="SP98">

    <input type="submit" value="Submit">
  </form>

</div>
</body>
</html>
```

6 RESULT:



Car Fuel Consumption

Car Fuel Consumption Prediction

Fill in and below details to predict the consumption depending on the gas type.

('Car fuel Consumption(L/100km) : ', 4.707891280435151)

7 CONCLUSION:

The fuel consumption of the is predicted depending of the gas type