EMISSION OF CO_2 FROM CARS

By:
Kungumaswetha A
Lekaa Darsni M
Sangeetha Sri K J
Shanmugapriya M S

INTRODUCTION

By trapping heat from the sun, greenhouse gases have kept Earth's climate habitable for humans and millions of other species. But those gases are now out of balance.Carbon dioxide being a key greenhouse gas,that drives global climate change, continues to rise every month.Between 2014 and 2016, global carbon emissions remained mostly flat, raising hopes the world’s carbon output may have peaked for good. However, emissions began to rise again in 2017 and have continued growing.

Huge cuts are needed to avoid dangerous warming and the melting of the glaciers.There is an urge to save the planet before it becomes pathetic than now.

LITERATURE SURVEY

Existing Problem

The amount of CO_2 emission from the transport sector (including cars) accounts for about 20% of total CO_2 emissions. Accordingly, from the viewpoint of preventing global warming, reducing that proportion is a key issue. In regard to CO_2 emissions from cars, fuel economy standards are getting tougher all over the world, so improving fuel economy of cars is strongly desired. From now, it is considered that fuel economy of engines will be further improved by boosting engine efficiency and by hybridization (electrification) of cars.

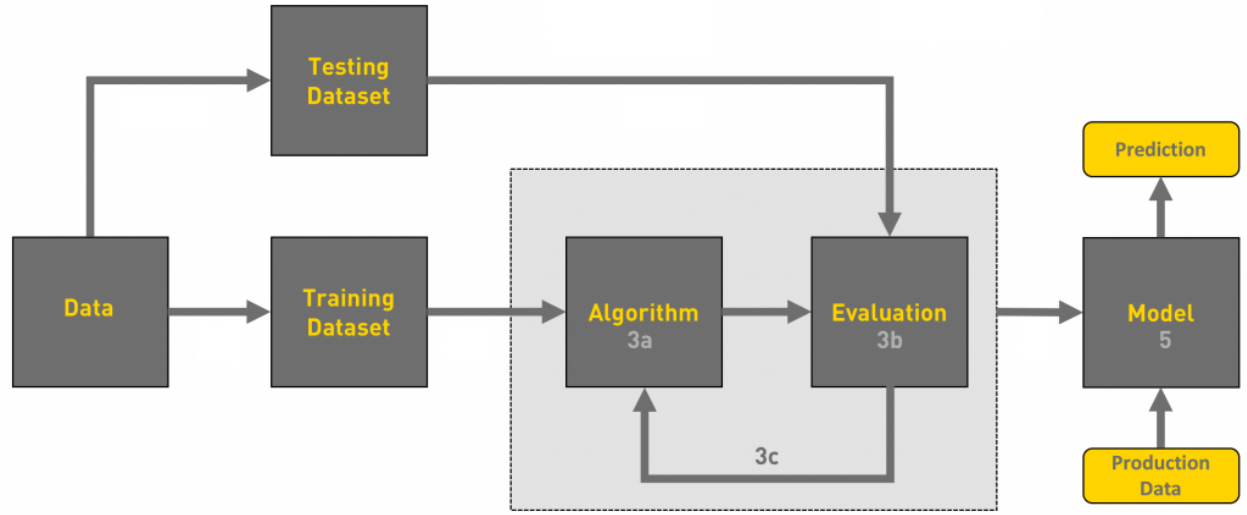
Proposed Solution

In this project,depending upon a car's features,we are going to predict the

CO_2 emission rate of a particular car. So that if, the emission rate of that particular car is more then the threshold value then, the car's owner is warned and that car's details are sent to the particular RTA region head to seize the car.

THEORITICAL ANALYSIS

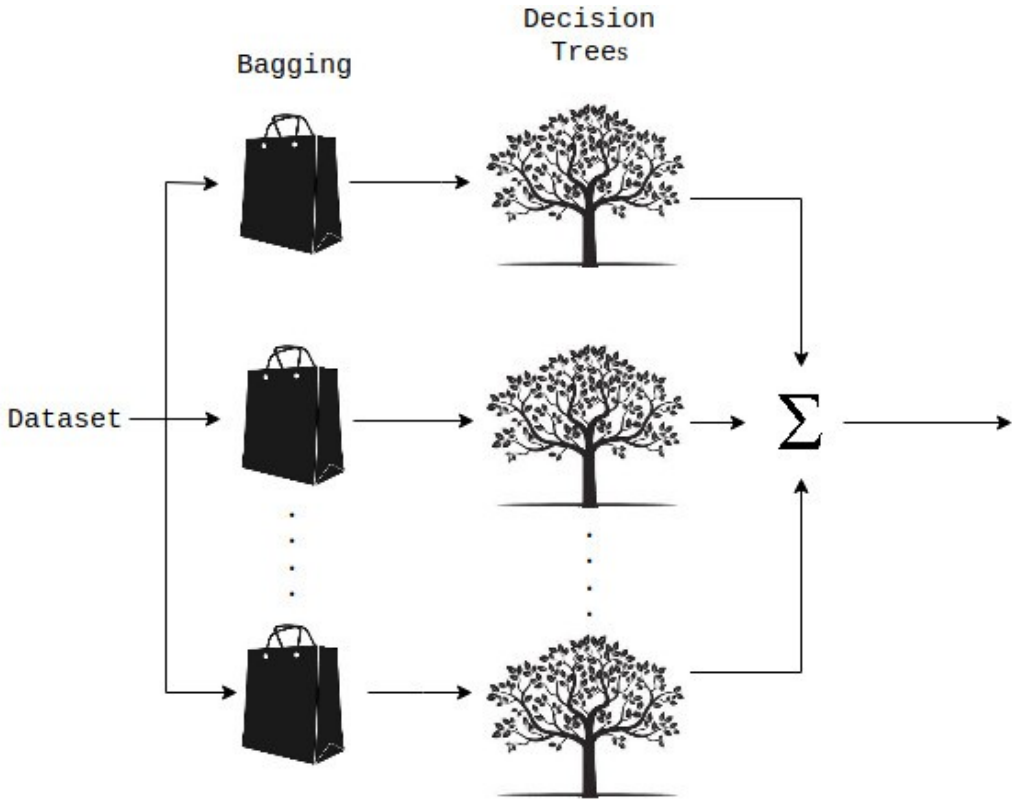
Block Diagram



Software Designing

The project provides insight of CO_2 emission prediction model using machine learning. Machine Learning provides different techniques to train the machine based on experience. In this project, Supervised Machine learning-Regression technique is used for the prediction of CO_2 emission rate from cars.An iterative and continuous improvement approach will be adapted to achieve successful results. The results are validated using R2_Score technique.

In the case of model building, random forest regression is used.A Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and bagging. The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees.

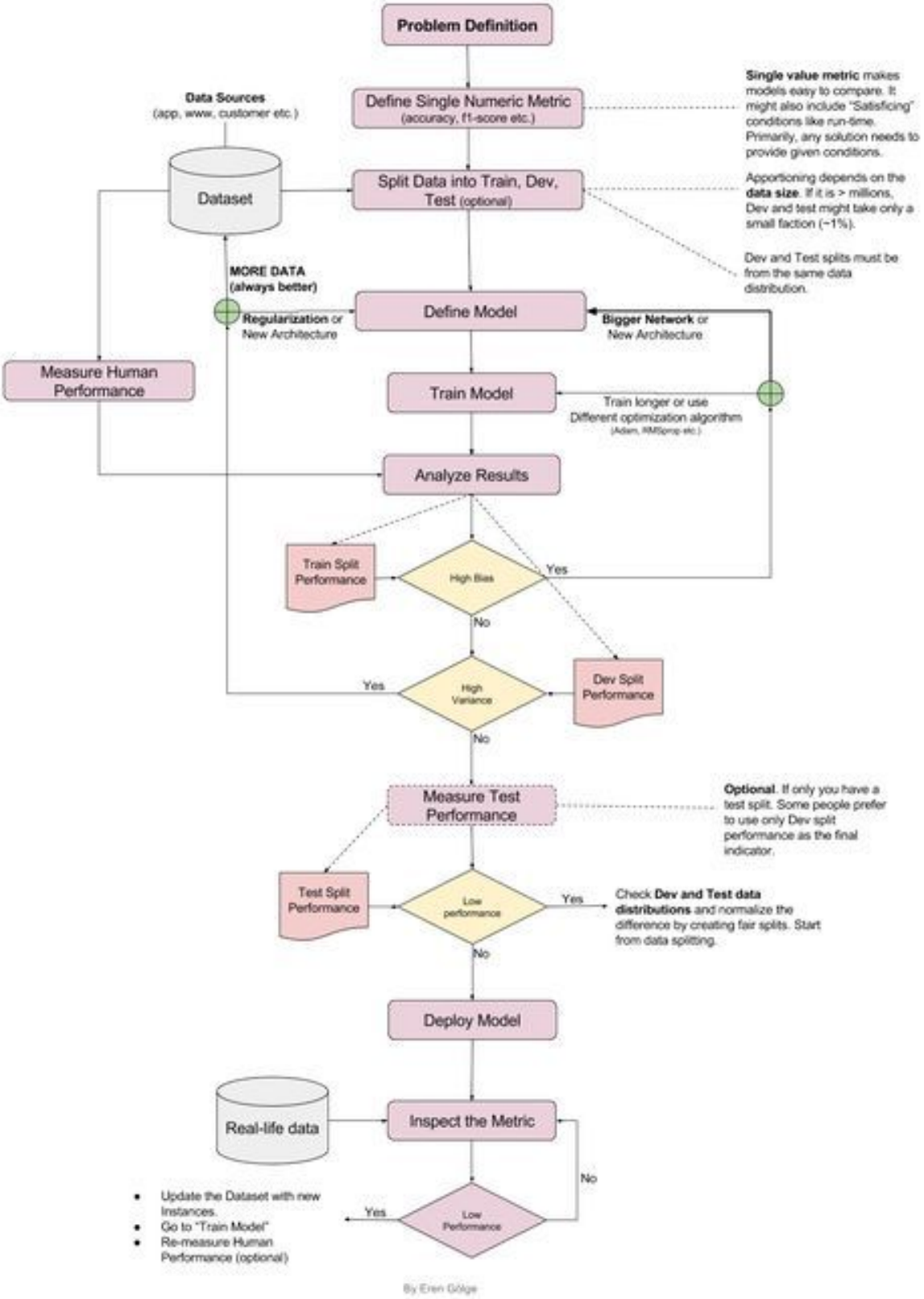


EXPERIMENTAL INVESTIGATIONS

In this project, the CO_2 emission rate of cars is predicted by the model , using the dataset that includes the parameters that are required for the prediction of CO_2 emission rate.The dataset comprises of the various parameters that are collected about each car which can be used by the model for prediction.

FLOWCHART

The flowchart, below, explains the work that the model prediction exactly does like splitting up the dataset as train and test sets, analysing the sets and predicting the values of the model and finally deploying the model based on the test performance and analysis.


Problem Definition
Define Single Numeric Metric
(accuracy, f1-score etc.)
Single value metric makes models easy to compare. It might also include "Satisficing" conditions like run-time. Primarily, any solution needs to provide given conditions.
Data Sources
(app, www, customer etc.)
Dataset
Split Data into Train, Dev, Test (optional)
Apportioning depends on the data size. If it is > millions, Dev and test might take only a small faction (~1%).
Dev and Test splits must be from the same data distribution.
MORE DATA
(always better)
Regularization or
New Architecture
Define Model
Bigger Network or
New Architecture
Measure Human Performance
Train Model
Train longer or use
Different optimization algorithm
(Adam, RMSprop etc.)
Analyze Results
Train Split Performance
High Bias
Yes
No
Dev Split Performance
High Variance
Yes
No
Measure Test Performance
Optional. If only you have a test split. Some people prefer to use only Dev split performance as the final indicator.
Test Split Performance
Low performance
Yes
Check Dev and Test data distributions and normalize the difference by creating fair splits. Start from data splitting.
No
Deploy Model
Real-life data
Inspect the Metric
No
Yes
Low Performance
Update the Dataset with new Instances.
Go to "Train Model"
Re-measure Human Performance (optional)
By Eren Gölge

RESULT

As a result of prediction, the CO_2 emission rate of the car is known. From the predicted values, it is decided whether to sent the car's details to RTA region head or not. If the predicted value is above the fixed threshold, then it should be sent to RTA region else, it is considered to be safe.

ADVANTAGES & DISADVANTAGES

- ◎ The computational cost of training a random forest is quite low,but, it requires more computational resources and is less intuitive.
- ◎ It is one of the handy algorithm that as they support default hyper parameters.
- ◎ Though,the prediction process using random forests is time-consuming than other algorithms,it has the power to handle datasets with large dimensionality.

APPLICATIONS

Using Random Forest Regressor, the value of CO_2 emitted in the cars are predicted using the entities like cylinders, fuel type and fuel consumption on various land lanes . This method builds multiple decision trees and merges them together to get a more stable and accurate prediction. It computes the score automatically for each feature after training and scales the result equals to one.

CONCLUSION

High accuracy in prediction that gives information about the CO_2 emission rate and this is because the main goal is to predict the emission of the gas in different land lanes . This prediction can help the person to know whether the greenhouse gas produced is a disturbance to the environment or not.

FUTURE SCOPE

The aim of the project is to predict the CO_2 emission rate for cars,depending on its features.Further,the project can be developed,where it can serve to be helping hand to reduce the CO_2 emission rates.An option of "Enhancement" can be

added,where the application suggests the changes that can be done to the existing features of the car,so that,it emits lesser amount of CO_2.

BIBILOGRAPHY

- https://www.nationalgeographic.com/environment/global-warming/greenhouse-gases/
- https://ieeexplore.ieee.org/document/8664935
- 2018 Fuel Consumption Guide

APPENDIX

Source Code:

CO2 Emission.ipynb:

```
In [1]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt

In [2]: ds=pd.read_csv(r'FuelConsumption.csv')
        ds.head()

In [3]: ds.corr()

In [4]: ds.drop(['MODELYEAR','MAKE','MODEL','VEHICLECLASS','TRANSMISSION','FUELCONSUMPTION_COMB_MPG'],axis=1,inplace=True)

In [5]: ds

In [6]: ds.describe()

In [7]: ds.isnull().sum()

In [8]: # correlation with one another
        ds.corr()
        # administration has little corr with others

In [9]: import seaborn as sns
        sns.boxplot(x=ds['ENGINESIZE'])
```

In [10]:
```
sns.boxplot(x=ds['CYLINDERS'])
```

In [11]:
```
sns.boxplot(x=ds['FUELCONSUMPTION_CITY'])
```

In [12]:
```
sns.boxplot(x=ds['FUELCONSUMPTION_HWY'])
```

In [13]:
```
sns.boxplot(x=ds['FUELCONSUMPTION_COMB'])
```

In [14]:
```
x=ds.iloc[:,0:6].values
x
```

In [15]:
```
y=ds.iloc[:,-1].values
y
```

In [16]:
```
ds['FUELTYPE'].value_counts()
```

In [17]:
```
import sklearn
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
```

In [18]:
```
ct=ColumnTransformer([("on",OneHotEncoder(),[2])],remainder='passthrough')
x=ct.fit_transform(x)
x
```

In [19]:
```
#remove dummy variables it reduces the accuracy
x=x[:,1:]
x
```

In [20]:
```
x.shape
```

In [21]:
```
plt.scatter(x[:,3],y)
```

In [22]:
```
plt.scatter(x[:,4],y)
```

In [23]:
```
plt.scatter(x[:,5],y)
```

In [24]:
```
plt.scatter(x[:,6],y)
```

In [25]:
```
plt.scatter(x[:,7],y)
```

In [26]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)
```

In [27]:
```
x_train.shape
```

In [28]:
```
y_train.shape
```

In [29]:
```
x_test.shape
```

In [30]:
```
y_test.shape
```

In [31]:
```
from sklearn.ensemble import RandomForestRegressor
```

In [32]:
```
rf=RandomForestRegressor(n_estimators=10,criterion='mse',random_state=0)
rf.fit(x_train,y_train)
```

In [33]:
```
from joblib import dump
dump(rf,'RFR.save')
```

In [34]:
```
y_predict=rf.predict(x_test)
y_predict
```

In [35]:
```
y_test
```

In [36]:
```
from sklearn.metrics import r2_score
r2_score(y_test,y_predict)
```

app.py:

```python
import numpy as np
from flask import Flask, request, jsonify, render_template
from joblib import load
app = Flask(__name__)
model = load('RFR.save')

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

@app.route('/y_predict',methods=['POST'])
def y_predict():
    '''
    For rendering results on HTML GUI
    '''
    x_test = [[float(x) for x in request.form.values()]]
    if(x_test[0][0] == 0):
        x_test[0][0]=0
        x_test[0].insert(1,0)
        x_test[0].insert(2,1)
    elif(x_test[0][0] == 1):
        x_test[0][0]=0
        x_test[0].insert(1,1)
        x_test[0].insert(2,0)
    elif(x_test[0][0] == 2):
        x_test[0][0]=1
        x_test[0].insert(1,0)
        x_test[0].insert(2,0)
    else:
        x_test[0][0]=0
        x_test[0].insert(1,0)
        x_test[0].insert(2,0)
    print(x_test)
    prediction = model.predict(x_test)
    print(prediction)
    output=prediction[0]

    return render_template('proj1.html', prediction_text = output)

if __name__ == "__main__":
    app.run(debug=True)
```

proj.html:

```
<!DOCTYPE html>
<html >
<head>
  <meta charset="UTF-8">
  <title>IBM project</title>
  <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='css/style1.css') }}">
<body>
    <h1 style="color:black;font-size:40px;">CO2 EMISSION RATE FROM CARS</h1>


    {% if prediction_text>200 %}
         <h1 style="color:black;font-size:25px;"> The CO2 emission rate for your vehicle is {{ prediction_text }} g/km.</h1>
         <h1 style="color:black;font-size:25px;"> Kindly stick to the standards, else your vehicle will be seized!</h1>
      {% else %}
         <h1 style="color:black;font-size:25px;">The CO2 emission rate for your vehicle is {{ prediction_text }} g/km.</h1>
         <h1 style="color:black;font-size:25px;">You are doing good!</h1>
      {% endif %}

   <br>
   <br>


</body>
</html>
```

proj1.html:

```
<!DOCTYPE html>
<html >
<head>
  <meta charset="UTF-8">
  <title>IBM project</title>
  <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='css/style1.css') }}">
<body>
 <div class="login">
    <h1 style="color:black;font-size:40px;"><center>CO2 EMISSION RATE FROM CARS</center></h1>


     <!-- Main Input For Receiving Query to our ML -->
    <form action="{{ url_for('y_predict')}}"method="post">
        <select name="State" required="required">
          <option value="">FUEL TYPE </option>
          <option value=0>Premium Gasoline</option>
          <option value=1>Regular Gasoline</option>
          <option value=2>E85</option>
          <option value=3>Diesel</option>
        </select><br>
        <p>
        <input type="text" name="ENGINESIZE" placeholder="ENGINE SIZE"  required="required" /><br>
        <input type="text" name="CYLINDERS" placeholder="NO. OF CYLINDERS" required="required" /><br>
        <input type="text" name="FUELCONSUMPTION_CITY" placeholder="FUEL CONSUMPTION-CITY" required="required" /><br>
        <input type="text" name="FUELCONSUMPTION_HWY" placeholder="FUEL CONSUMPTION-HIGHWAY" required="required" /><br>
        <input type="text" name="FUELCONSUMPTION_COMB   " placeholder="FUEL CONSUMPTION-COMBINATION " required="required" /><br>
        </p>
        <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>

    </form>

   <br>
   <br>

 </div>


</body>
</html>

```

Output Screen: