Vehicle CO2 Emissions Analysis

Research on Transportation and Environmental Factor

Hoang Nha Nguyen

Contents

1. Why CO2 Emissions Reduction is important?

- CO2 Vehicle Emissions is a big contributor to climate change and air pollution.
- A typical passenger vehicle emits about 4.6 metric tons of carbon dioxide per year. This number can vary ??

Tasks

- Integrate Machine Learning Algorithms that deploy regression techniques
- Predict the amount of carbon footprint of the vehicles using Python Programming Language
- Build and evaluate supervised machine learning models for multivariate regression
- Worked with NumPy, Pandas, Seaborn and Matplotlib libraries for data visualization and analytics

Purpose

Prevent the environmental disasters and global warming effects and assist the consumers decision-making as we want to live in a world that strives for zero carbon emissions and that we are favorable of vehicles with minimized greenhouse gas emissions.

What I try to do...

The general idea of machine learning is to get a model to learn trends from historical data on any topic and be able to reproduce those trends on comparable data in the future.

In this project, I'm going to use the “ 2022 Fuel Consumption Ratings” dataset from the Government of Canada Official Website on vehicles' attributes to predict the amount of Carbon Dioxide / CO2 Emissions or Greenhouse Gas Emissions that the vehicles produce.

2022Fuel_Consumption_Ratings.csv

2. Data Exploration - A quick overview of Car Attributes

We have 1067 instances (total number of car models) and 13 features that describes the car attributes and car behaviors, also knowns as rows and columns including:

1. **Year** – refers to model year when the vehicle was produced
2. **Make** – brand of the vehicle
3. **Model** – name of the car product or a range of products including:
    - 4WD/4X4 = Four-wheel drive
    - AWD = All-wheel drive
    - FFV = Flexible-fuel vehicle
    - SWB = Short wheelbase
    - LWB = Long wheelbase
    - EWB = Extended wheelbase

4. **Vehicle Class** – designation of automobile vehicle types that are not limited to compact, SUV, mid-size, station wagon, pickup truck, minivan, etc.
5. **Engine Size** – refer to as 'engine capacity' or 'engine displacement' and is the measurement of the total volume of the cylinders in the engine

2. Data Exploration (cont.)

6. **Cylinder** – the power unit of an engine; it's the chamber where the gasoline is burned and turned into power.
    - Transmission type - Gear transmission of the car including:
    - A = Automatic
    - AM = automated manual
    - AS = automatic with select shift
    - AV = continuously variable transmission
    - M = Manual
    - 3 – 10: number of gears
7. **Fuel Type** – energy sources by which the motor vehicles are powered including:
    - X = regular gasoline
    - Z = premium gasoline
    - D = Diesel
    - E = Ethanol
    - N = Natural Gas
8. **Fuel Consumption City** – represent fuel consumption while driving in inner-city conditions with heavy, stop-start, local traffic
9. **Fuel Consumption** – City and highway fuel consumption ratings are described in liters per 100 kilometers (L/100 km) unit and the combined rating (55% city, 45% highway) is in L/100 km unit and in miles per imperial gallon (mpg)
10. **CO2 Emissions** – the emissions of carbon dioxide (in grams per kilometer) for combined city and highway driving
11. **CO2 Ratings** – the emissions of carbon dioxide rated on a scale from 1 (worst) to 10 (best)
12. **Smog Ratings** – the emissions of smog-forming pollutants rated on a scale from 1 (worst) to 10 (best)

2022CarEmissions

Model_Year	Make	Model	Vehicle_Class	Engine_Size	Cylinders	Transmission_type	Fuel_Type	CT_FC	HWG_FC	CB_FC	CB_FC_MPG	CO2Emissions	CO2_rating	Smog_rating
2022	Acura	ILX	Compact	2.4	4	AM8	Z	9.9	7.0	8.6	33	200	6	3
2022	Acura	MDX SH-AWD	SUV: Small	3.5	6	AS10	Z	12.6	9.4	11.2	25	263	4	5
2022	Acura	RDX SH-AWD	SUV: Small	2.0	4	AS10	Z	11.0	8.6	9.9	29	232	5	6
2022	Acura	RDX SH-AWD A-SPEC	SUV: Small	2.0	4	AS10	Z	11.3	9.1	10.3	27	242	5	6
2022	Acura	TLX SH-AWD	Compact	2.0	4	AS10	Z	11.2	8.0	9.8	29	230	5	7
2022	Acura	TLX SH-AWD A-SPEC	Compact	2.0	4	AS10	Z	11.3	8.1	9.8	29	231	5	7
2022	Acura	TLX Type S	Compact	3.0	6	AS10	Z	12.3	9.4	11.0	26	256	5	5
2022	Acura	TLX Type S (Performance Tire)	Compact	3.0	6	AS10	Z	12.3	9.8	11.2	25	261	4	5
2022	Alfa Romeo	Giulia	Mid-size	2.0	4	A8	Z	10.0	7.2	8.7	32	205	6	3
2022	Alfa Romeo	Giulia AWD	Mid-size	2.0	4	A8	Z	10.5	7.7	9.2	31	217	5	3
2022	Alfa Romeo	Giulia Quadrifoglio	Mid-size	2.9	6	A8	Z	13.5	9.3	11.6	24	271	4	3
2022	Alfa Romeo	Stelvio	SUV: Small	2.0	4	A8	Z	10.3	8.1	9.3	30	218	5	3
2022	Alfa Romeo	Stelvio AWD	SUV: Small	2.0	4	A8	Z	10.8	8.3	9.6	29	226	5	3
2022	Alfa Romeo	Stelvio AWD Quadrifoglio	SUV: Small	2.9	6	A8	Z	13.9	10.3	12.3	23	288	4	3
2022	Aston Martin	DB11 V8	Minicompact	4.0	8	A8	Z	13.0	9.8	11.5	25	271	4	5
2022	Aston Martin	DB11 V12	Minicompact	5.2	12	A8	Z	16.4	10.7	13.8	20	324	3	3
2022	Aston Martin	DBS V12	Minicompact	5.2	12	A8	Z	16.4	10.7	13.8	20	324	3	3
2022	Aston Martin	DBX V8	SUV: Standard	4.0	8	A9	Z	16.8	11.9	14.6	19	343	3	5
2022	Aston Martin	Vantage V8	Two-seater	4.0	8	A8	Z	13.1	9.6	11.5	25	270	4	5
2022	Audi	A3 Sedan 40 TFSI quattro	Subcompact	2.0	4	AM7	X	8.5	6.6	7.6	37	178	7	7
2022	Audi	A4 Sedan 40 TFSI quattro	Compact	2.0	4	AM7	Z	9.1	7.0	8.2	34	190	6	5
2022	Audi	A4 Sedan 45 TFSI quattro	Compact	2.0	4	AM7	Z	9.8	7.6	8.8	32	205	6	5
2022	Audi	A4 allroad 45 TFSI quattro	Station wagon: Small	2.0	4	AM7	Z	9.8	7.9	8.9	32	208	6	5
2022	Audi	A5 Cabriolet 45 TFSI quattro	Subcompact	2.0	4	AM7	Z	10.4	7.5	9.1	31	214	5	5
2022	Audi	A5 Coupe 45 TFSI quattro	Subcompact	2.0	4	AM7	Z	9.8	7.6	8.8	32	205	6	5
2022	Audi	A5 Sportback 45 TFSI quattro	Mid-size	2.0	4	AM7	Z	9.8	7.6	8.8	32	205	6	5
2022	Audi	A6 Sedan 45 TFSI quattro	Mid-size	2.0	4	AM7	Z	10.2	7.4	8.9	32	208	6	5
2022	Audi	A6 Sedan 55 TFSI quattro	Mid-size	3.0	6	AM7	Z	11.1	7.8	9.6	29	224	5	5
2022	Audi	A6 allroad 55 TFSI quattro	Station wagon: Mid-size	3.0	6	AM7	Z	11.5	8.3	10.0	28	234	5	5
2022	Audi	A7 Sportback 55 TFSI quattro	Mid-size	3.0	6	AM7	Z	11.1	7.8	9.6	29	224	5	5
2022	Audi	A8 L Sedan 55 TFSI quattro	Full-size	3.0	6	AS8	Z	12.6	8.3	10.6	27	248	5	5

Data Types_Categorical

```
df["Fuel_Type"].unique()
```

```
array(['Z', 'X', 'D', 'E'], dtype=object)
```

```
df["Vehicle_Class"].unique()
```

```
array(['Compact', 'SUV: Small', 'Mid-size', 'Minicompact',
       'SUV: Standard', 'Two-seater', 'Subcompact',
       'Station wagon: Small', 'Station wagon: Mid-size', 'Full-size',
       'Pickup truck: Small', 'Pickup truck: Standard', 'Minivan',
       'Special purpose vehicle'], dtype=object)
```

```
df["Transmission_type"].unique()
```

```
array(['AM8', 'AS10', 'A8', 'A9', 'AM7', 'AS8', 'M6', 'AS6', 'AV', 'AS9',
       'A10', 'A6', 'M5', 'M7', 'AV7', 'AV1', 'AM6', 'AS7', 'AV8', 'AV6',
       'AV10', 'AS5', 'A7'], dtype=object)
```

Standardize Data Types - Get Dummy Variable

```
#Encoding Categorical Variable
df2 = pd.get_dummies(df,columns = ['Fuel_Type', 'Vehicle_Class','Transmission_type','Make','Model'])
```

```
df2
```

	Engine_Size	Cylinders	CT_FC	HWG_FC	CB_FC	CB_FC_MPG	CO2Emissions	CO2_rating	Smog_rating	Fuel_Type_D	...	Model_Yukon	Model_Yukon (No Stop-Start)	Mo
0	2.4	4	9.9	7.0	8.6	33	200	6	3	0	...	0	0	
1	3.5	6	12.6	9.4	11.2	25	263	4	5	0	...	0	0	
2	2.0	4	11.0	8.6	9.9	29	232	5	6	0	...	0	0	
3	2.0	4	11.3	9.1	10.3	27	242	5	6	0	...	0	0	
4	2.0	4	11.2	8.0	9.8	29	230	5	7	0	...	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
939	2.0	4	10.7	7.7	9.4	30	219	5	5	0	...	0	0	
940	2.0	4	10.5	8.1	9.4	30	219	5	5	0	...	0	0	
941	2.0	4	11.0	8.7	9.9	29	232	5	7	0	...	0	0	
942	2.0	4	11.5	8.4	10.1	28	236	5	5	0	...	0	0	
943	2.0	4	12.4	8.9	10.8	26	252	5	7	0	...	0	0	

944 rows × 802 columns

Pearson's Correlation

```
drop_ft = []


for col in df2:
    if(pearsonr(df2[col],df2['CO2Emissions'])[0] > -0.5 and pearsonr(df2[col],df2['CO2Emissions'])[0] < 0.5):
        drop_ft.append(col)

drop_ft
```

```
# calculate the Pearson's correlation between two variables
from numpy.random import randn
from numpy.random import seed
from scipy.stats import pearsonr
df2['Engine_Size']
df2['CO2Emissions']

pearsonr(df2['Engine_Size'],df2['CO2Emissions'])[0]
```

```
0.822577426427435
```

Cleaned Dataset

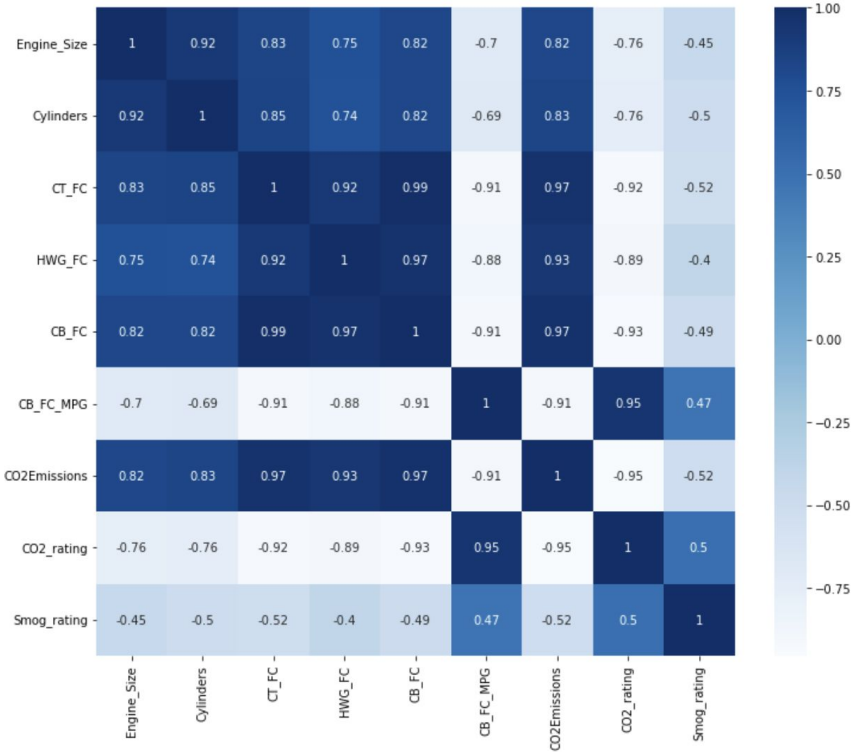
```
df3 = df2.drop(drop_ft, axis = 1)
```

```
df3
```

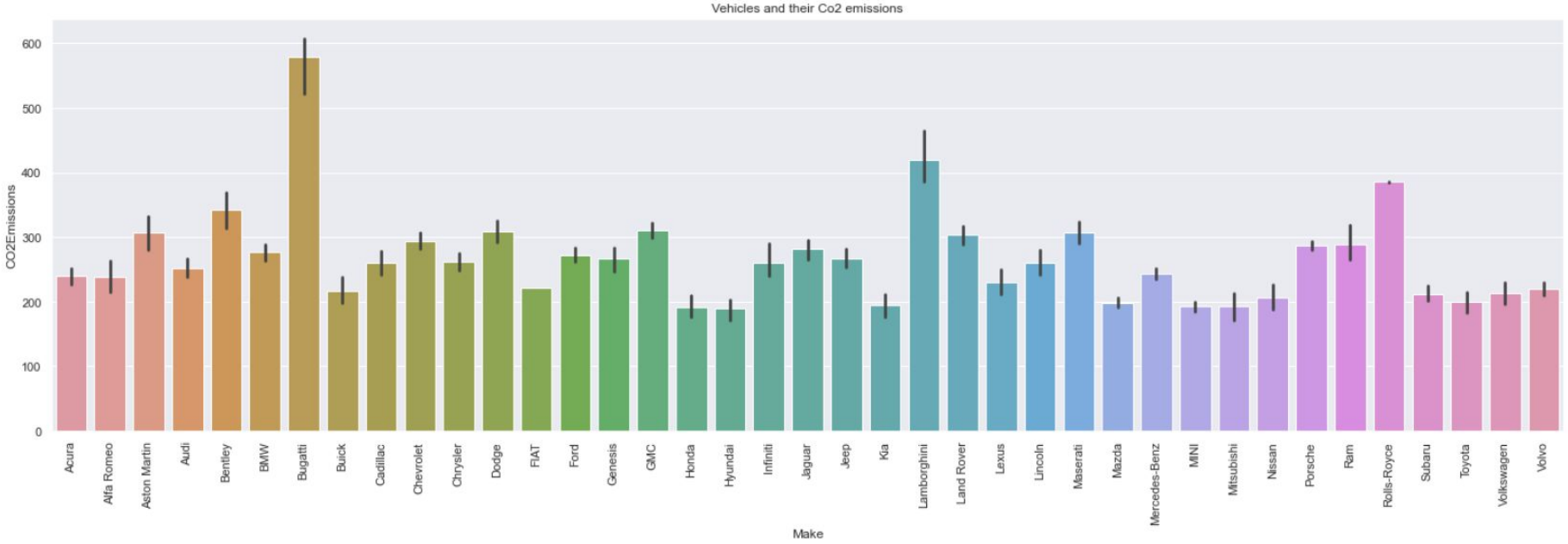
	Engine_Size	Cylinders	CT_FC	HWG_FC	CB_FC	CB_FC_MPG	CO2Emissions	CO2_rating	Smog_rating
0	2.4	4	9.9	7.0	8.6	33	200	6	3
1	3.5	6	12.6	9.4	11.2	25	263	4	5
2	2.0	4	11.0	8.6	9.9	29	232	5	6
3	2.0	4	11.3	9.1	10.3	27	242	5	6
4	2.0	4	11.2	8.0	9.8	29	230	5	7
...	...	...	...	...	...	...	...	...	...
939	2.0	4	10.7	7.7	9.4	30	219	5	5
940	2.0	4	10.5	8.1	9.4	30	219	5	5
941	2.0	4	11.0	8.7	9.9	29	232	5	7
942	2.0	4	11.5	8.4	10.1	28	236	5	5
943	2.0	4	12.4	8.9	10.8	26	252	5	7

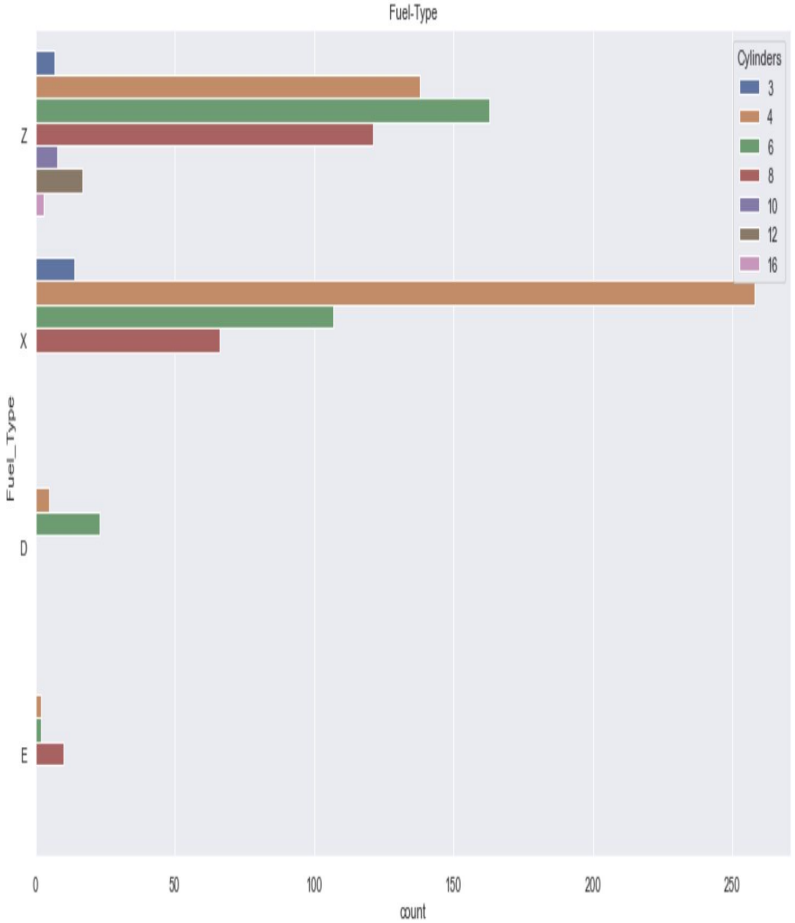
944 rows × 9 columns

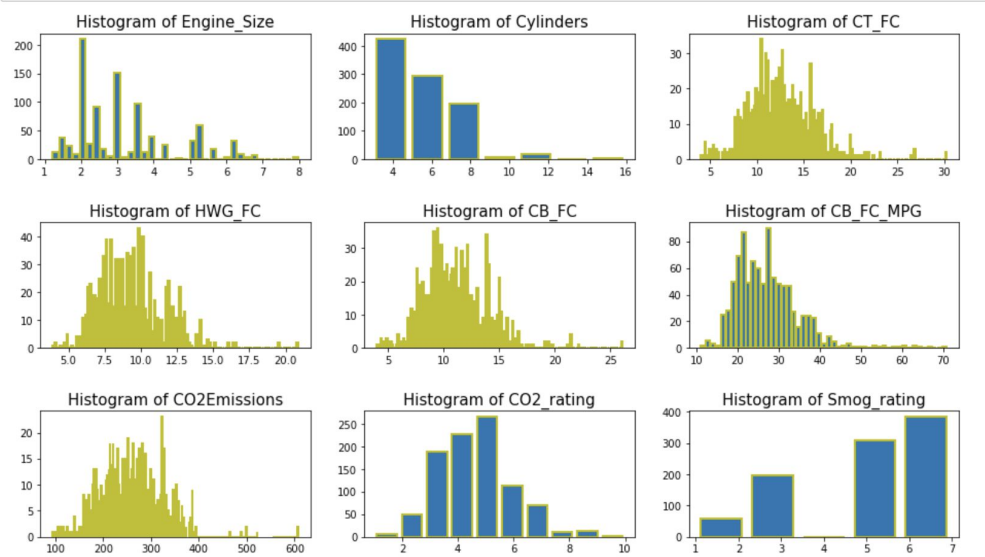
Correlation Heatmap



3. Data Visualization




Fuel-Type
Cylinders
3
4
6
8
10
12
16
Z
X
D
E
Fuel_Type
0
50
100
150
200
250
count


Histogram of Engine_Size
Histogram of Cylinders
Histogram of CT_FC
Histogram of HWG_FC
Histogram of CB_FC
Histogram of CB_FC_MPG
Histogram of CO2Emissions
Histogram of CO2_rating
Histogram of Smog_rating

4. Data Preprocessing

1.No missing values are found within this variables.

2.Converted the ‘**object’** data type to ‘**numerical’** data type

3.To complete the analysis we use python programming and it’s libraries like:

- NumPy
- Pandas
- Matplotlib
- Seaborn

5. Data Analysis

From the sklearn module we will use the LinearRegression() method to create a linear regression object. This object has a method called fit() that takes the independent and dependent values as parameters and fills the regression object with data that describes the relationship:

$$\hat{Y} = b_0 + b_1 X_1 + b_2 X_2 + \ldots + b_p X_p$$

```
X = df3.drop('CO2Emissions', axis=1)
Y= df3['CO2Emissions']

# spliting into train and test dataset with train_test_split :

x_train, x_test, y_train, y_test = train_test_split(X, Y , test_size=0.2, random_state=42)
```

Linear Regression Model

```
# Creating the model with the train dataset
model = linear_model.LinearRegression()
# Now we have a regression object that are ready to predict CO2 values based on a car's attributes
```

```
# # I use training dateset to train the model y=ax+b
model.fit(x_train,y_train)
```

```
LinearRegression()
```

```
# # Finding coeficients A and B of the model
#Coefficient - The coefficient is a factor that describes the relationship with an unknown variable.

print(f'intercept: {model.intercept_} | slope: {model.coef_}')
```

```
intercept: 156.12783329147743 | slope: [-3.12780050e-02  2.45308052e+00 -1.79007347e-01  1.02943229e-02
  1.39912063e+01  1.08508067e+00 -1.95564641e+01 -8.72005458e-01]
```

```
predict
```

```
array([326.34092535, 365.95731801, 274.26554396, 222.90826082,
       244.04404995, 315.03253828, 337.79296171, 329.30791989,
       357.2729118 , 279.21883841, 326.5569377 , 231.81831545,
       310.90472046, 304.44016493, 167.86921731, 238.82277868,
       194.23712591, 239.6506682 , 247.52323993, 313.19419197,
       190.57870838, 194.79267852, 193.29910992, 332.85146446,
       223.51521439, 281.38276296, 271.42414313, 190.947755  ,
       195.52334095, 191.69644203, 286.32001668, 229.34195608,
       287.28345681, 229.07274696, 231.76936153, 225.95202741,
       289.24637176, 162.31008494, 164.56491009, 340.65664005,
       286.32001668, 292.0062231 , 237.766958  , 344.39753469,
       188.2053424 , 241.42535133, 191.74994136, 268.36041694,
       190.89619066, 286.2290452 , 281.45087187, 319.08617188,
       326.5569377 , 227.35217747, 161.06569681, 229.18432334,
       235.70406374, 162.28786361, 332.0170039 , 198.096659  ,
```

Statistical Measures

```
from math import sqrt
# # Showing metrics to check the acuracy of our model
print(f'Sum of squared error (SSE): {np.sum((predict - y_test)**2)}')
print(f'Mean squared error (MSE): {mean_squared_error(y_test, predict)}')
print (f'Sqrt of mean squared error (RMSE):  {sqrt(mean_absolute_error(y_test, predict))}')
print(f'Accuracy: {100*r2_score(predict, y_test)}',"%")
```

```
Sum of squared error (SSE): 28929.262769058176
Mean squared error (MSE): 153.06488237596918
Sqrt of mean squared error (RMSE):  2.6823618312611655
Accuracy: 95.82742676307564 %
```

```
acc_lr = 100*r2_score(predict, y_test)
acc_lr
```

```
95.82742676307564
```

Decision Tree

```
print("Mean Absolute Error of the Decision Tree on test set is", mean_absolute_error(y_test, y_pred))
print("Mean Squared Error of the Decision Tree on test set is", mean_squared_error(y_test, y_pred))
print("Root Mean Squared Error of the Decision Tree on test set is", sqrt(mean_absolute_error(y_test, y_pred)))
```

```
Mean Absolute Error of the Decision Tree on test set is 1.5996472663139332
Mean Squared Error of the Decision Tree on test set is 11.458553791887127
Root Mean Squared Error of the Decision Tree on test set is 1.264771626149928
```

```
acc_dt = 100*tree.score(x_test, y_test)
print(acc_dt,'%')
```

```
99.66474236302727 %
```

Random Forest

```
print("Mean Absolute Error of the RF on test set is", mean_absolute_error(y_test, y_pred))
print("Mean Squared Error of the RF on test set is", mean_squared_error(y_test, y_pred))
print("Root Mean Squared Error of the RF on test set is", sqrt(mean_absolute_error(y_test, y_pred)))
```

```
Mean Absolute Error of the RF on test set is 1.7968124632569085
Mean Squared Error of the RF on test set is 16.752814102821336
Root Mean Squared Error of the RF on test set is 1.3404523353170408
```

```
acc_rf = 100*rf.score (x_test, y_test)
print(acc_rf,'%')
```

```
99.50984138393346 %
```

```
print('R squared of the Random Forest Regressor on training set: {:.2%}'. format (rf.score(x_train, y_train)))
print( 'R squared of the Random Forest Regressor on test set: {:.2%}'. format (rf. score (x_test, y_test)))
```

```
R squared of the Random Forest Regressor on training set: 99.89%
R squared of the Random Forest Regressor on test set: 99.51%
```

Conclusion

ML Models	Linear Regression	Decision Tree	Random Forest
Accuracy	95.8%	99.7%	99.5%
RMSE	2.68	1.26	1.34

Future Research

-Collect more relevant dataset

-Create a Graphical User Interface

-Research on more vehicle attributes that contribute as the environmental factors.