Good afternoon everyone

Multiple Linear Regression Model / Linear Regression with Multiple Variables/Features/Attributes

A linear regression model that contains more than one predictor variable is called a multiple linear regression model. The following model is a multiple linear regression model with two predictor variables, X_1 and X_2. then

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon$$

Y = M*x + C

Y = m1*x1* + *m2*x2 + m3 * x3 ..... mn * xn + c + error

m1, m2,m3,m4 ..... mn

Datasets Link (https://raw.githubusercontent.com/AP-State-Skill-Development-Corporation/Datasets/master/Regression/FuelConsumptionCo2.csv

Datasets Description (https://www.kaggle.com/gangliu/oc2emission)

In [1]:

```
import pandas as pd

fuel = pd.read_csv('https://raw.githubusercontent.com/AP-State-Skill-Development-Corporatio
fuel.head()
```

Out[1]:

	MODELYEAR	MAKE	MODEL	VEHICLECLASS	ENGINESIZE	CYLINDERS	TRANSMISSION
0	2014	ACURA	ILX	COMPACT	2.0	4	AS5
1	2014	ACURA	ILX	COMPACT	2.4	4	M6
2	2014	ACURA	ILX HYBRID	COMPACT	1.5	4	AV7
3	2014	ACURA	MDX 4WD	SUV - SMALL	3.5	6	AS6
4	2014	ACURA	RDX AWD	SUV - SMALL	3.5	6	AS6

Understanding the Data

`FuelConsumption.csv` :

- **MODELYEAR** e.g. 2014
- **MAKE** e.g. Acura
- **MODEL** e.g. ILX
- **VEHICLE CLASS** e.g. SUV
- **ENGINE SIZE** e.g. 4.7
- **CYLINDERS** e.g 6
- **TRANSMISSION** e.g. A6
- **FUEL CONSUMPTION in CITY(L/100 km)** e.g. 9.9
- **FUEL CONSUMPTION in HWY (L/100 km)** e.g. 8.9
- **FUEL CONSUMPTION COMB (L/100 km)** e.g. 9.2
- **CO2 EMISSIONS (g/km)** e.g. 182 --> low --> 0

In [3]:

```
fuel['MAKE']
```

Out[3]:

```
0       ACURA
1       ACURA
2       ACURA
3       ACURA
4       ACURA
        ...
1062    VOLVO
1063    VOLVO
1064    VOLVO
1065    VOLVO
1066    VOLVO
Name: MAKE, Length: 1067, dtype: object
```

In [5]:

```
fuel.shape
```

Out[5]:

```
(1067, 13)
```

In [4]:

```
fuel['MAKE'].value_counts()
```

Out[4]:

```
FORD              90
CHEVROLET         86
BMW               64
MERCEDES-BENZ     59
AUDI              49
GMC               49
TOYOTA            49
PORSCHE           44
VOLKSWAGEN        42
DODGE             39
MINI              36
KIA               33
NISSAN            33
CADILLAC          32
JEEP              31
MAZDA             27
HYUNDAI           24
SUBARU            23
LEXUS             22
JAGUAR            22
HONDA             21
INFINITI          21
LAND ROVER        19
CHRYSLER          19
BUICK             16
MITSUBISHI        16
RAM               13
ACURA             12
LINCOLN           11
VOLVO             11
FIAT              10
SCION              9
BENTLEY            8
ROLLS-ROYCE        7
ASTON MARTIN       7
MASERATI           6
LAMBORGHINI        3
SRT                2
SMART              2
Name: MAKE, dtype: int64
```

In [7]:

```
fuel['MAKE'].value_counts().count()
```

Out[7]:

```
39
```

In [6]:

```
fuel['MODEL'].value_counts()
```

Out[6]:

```
F150 FFV                 8
F150 FFV 4X4             8
FOCUS FFV                6
BEETLE                   6
ACCORD                   6
                       ..
911 TURBO CABRIOLET      1
RAV4 LIMITED AWD         1
F-TYPE CONVERTIBLE       1
1500 DIESEL              1
RANGE ROVER V8 5.0 SC    1
Name: MODEL, Length: 663, dtype: int64
```

In [8]:

```
fuel['VEHICLECLASS'].value_counts()
```

Out[8]:

```
MID-SIZE                   178
COMPACT                    172
SUV - SMALL                154
SUV - STANDARD             110
FULL-SIZE                   86
TWO-SEATER                  71
SUBCOMPACT                  65
PICKUP TRUCK - STANDARD     62
MINICOMPACT                 47
STATION WAGON - SMALL       36
VAN - PASSENGER             25
VAN - CARGO                 22
MINIVAN                     14
PICKUP TRUCK - SMALL        12
SPECIAL PURPOSE VEHICLE      7
STATION WAGON - MID-SIZE     6
Name: VEHICLECLASS, dtype: int64
```

In [12]:

```
fuel['VEHICLECLASS'].value_counts().count()
```

Out[12]:

```
16
```

In [13]:

```
fuel['CYLINDERS'].value_counts()
```

Out[13]:

```
4     420
6     356
8     252
12     17
10      9
5       9
3       4
Name: CYLINDERS, dtype: int64
```

In [14]:

```
fuel['ENGINESIZE'].min()
```

Out[14]:

```
1.0
```

In [15]:

```
fuel['ENGINESIZE'].max()
```

Out[15]:

```
8.4
```

In [16]:

```
fuel['ENGINESIZE'].argmax()
```

Out[16]:

```
940
```

In [17]:

```
fuel.iloc[940]
```

Out[17]:

```
MODELYEAR                    2014
MAKE                          SRT
MODEL                 VIPER COUPE
VEHICLECLASS           TWO-SEATER
ENGINESIZE                    8.4
CYLINDERS                      10
TRANSMISSION                   M6
FUELTYPE                        Z
FUELCONSUMPTION_CITY           20
FUELCONSUMPTION_HWY            13
FUELCONSUMPTION_COMB         16.9
FUELCONSUMPTION_COMB_MPG       17
CO2EMISSIONS                  389
Name: 940, dtype: object
```

In [18]:

```
fuel['FUELTYPE'].value_counts()
```

Out[18]:

```
X    514
Z    434
E     92
D     27
Name: FUELTYPE, dtype: int64
```

In [19]:

```
fuel['FUELCONSUMPTION_CITY'].max()
```

Out[19]:

```
30.2
```

In [20]:

```
fuel['FUELCONSUMPTION_CITY'].argmax()
```

Out[20]:

```
228
```

In [21]:

```
fuel.iloc[228]
```

Out[21]:

```
MODELYEAR                              2014
MAKE                              CHEVROLET
MODEL                  EXPRESS 3500 PASSENGER
VEHICLECLASS                  VAN - PASSENGER
ENGINESIZE                                6
CYLINDERS                                 8
TRANSMISSION                             A6
FUELTYPE                                  E
FUELCONSUMPTION_CITY                   30.2
FUELCONSUMPTION_HWY                    20.5
FUELCONSUMPTION_COMB                   25.8
FUELCONSUMPTION_COMB_MPG                 11
CO2EMISSIONS                            413
Name: 228, dtype: object
```

In [22]:

```
fuel.iloc[fuel['FUELCONSUMPTION_HWY'].argmax()]
```

Out[22]:

```
MODELYEAR                              2014
MAKE                              CHEVROLET
MODEL                  EXPRESS 3500 PASSENGER
VEHICLECLASS                  VAN - PASSENGER
ENGINESIZE                                6
CYLINDERS                                 8
TRANSMISSION                             A6
FUELTYPE                                  E
FUELCONSUMPTION_CITY                   30.2
FUELCONSUMPTION_HWY                    20.5
FUELCONSUMPTION_COMB                   25.8
FUELCONSUMPTION_COMB_MPG                 11
CO2EMISSIONS                            413
Name: 228, dtype: object
```

In [23]:

```
fuel.iloc[fuel['FUELCONSUMPTION_HWY'].argmin()]
```

Out[23]:

```
MODELYEAR                      2014
MAKE                          HONDA
MODEL                 ACCORD HYBRID
VEHICLECLASS               MID-SIZE
ENGINESIZE                        2
CYLINDERS                         4
TRANSMISSION                     AV
FUELTYPE                          X
FUELCONSUMPTION_CITY            4.7
FUELCONSUMPTION_HWY             4.9
FUELCONSUMPTION_COMB            4.8
FUELCONSUMPTION_COMB_MPG         59
CO2EMISSIONS                    110
Name: 487, dtype: object
```

In [25]:

```
fuel['CO2EMISSIONS'].min(), fuel['CO2EMISSIONS'].max()
```

Out[25]:

```
(108, 488)
```

A+ -> 1 A- -> 2 B+ -> 3 B - -> 4 AB+ -> 5 AB- -> 6

108 - 120 - 1 121 - 140 - 2

In [26]:

```
import matplotlib.pyplot as plt
```

In [27]:

```
fuel.columns
```

Out[27]:

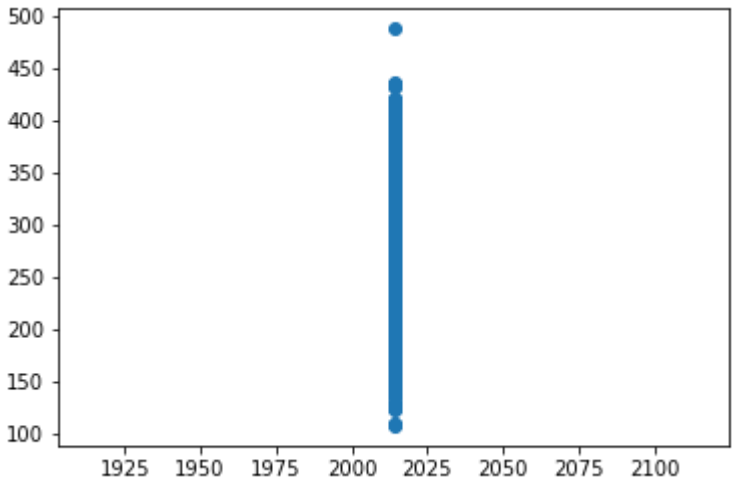
```
Index(['MODELYEAR', 'MAKE', 'MODEL', 'VEHICLECLASS', 'ENGINESIZE', 'CYLINDER
S',
       'TRANSMISSION', 'FUELTYPE', 'FUELCONSUMPTION_CITY',
       'FUELCONSUMPTION_HWY', 'FUELCONSUMPTION_COMB',
       'FUELCONSUMPTION_COMB_MPG', 'CO2EMISSIONS'],
      dtype='object')
```

In [29]:

```
plt.scatter(fuel['MODELYEAR'], fuel['CO2EMISSIONS'])
```

Out[29]:

```
<matplotlib.collections.PathCollection at 0x18a1344d130>
```



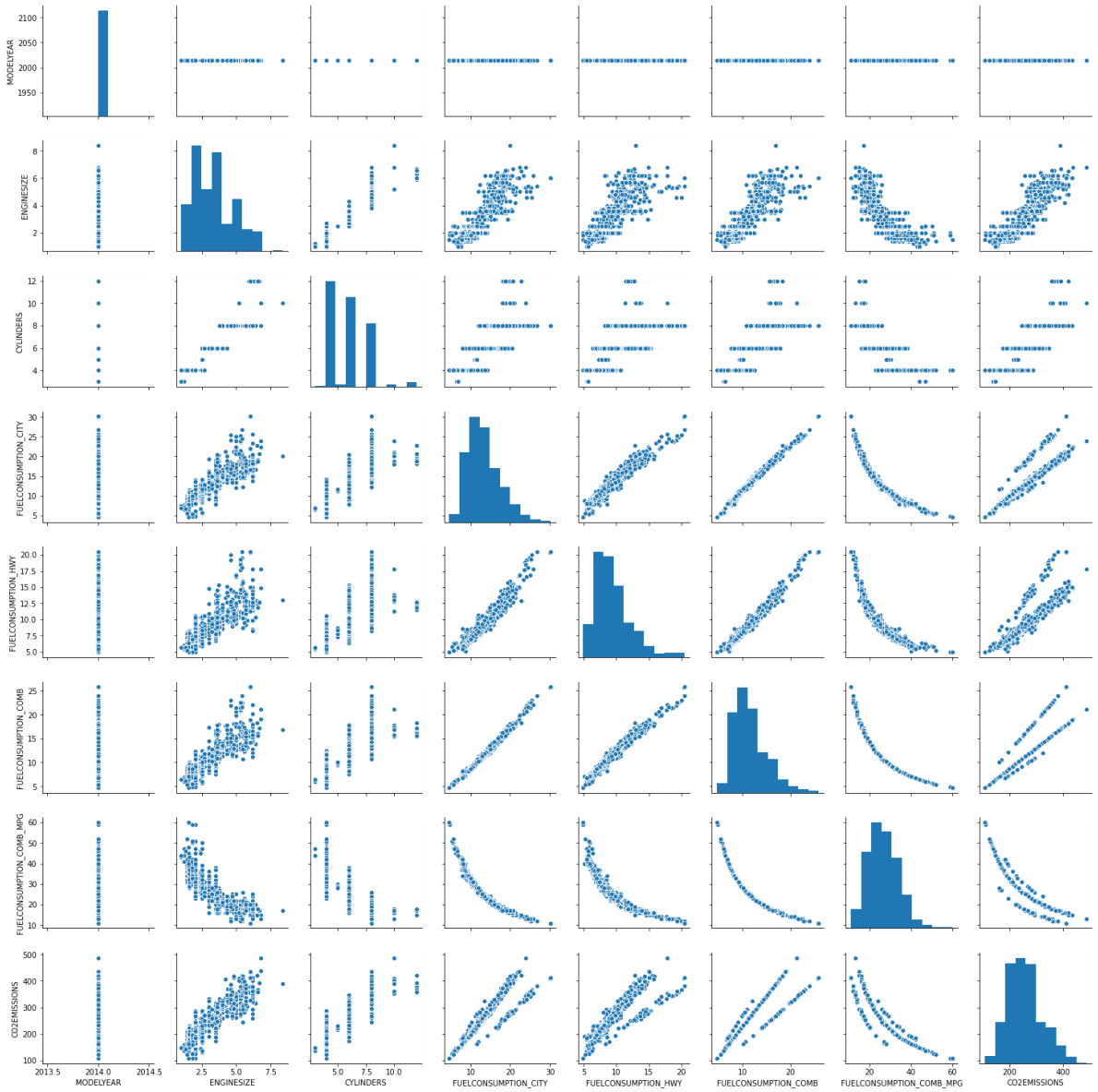
In [30]:

```
import seaborn as sns

sns.pairplot(fuel)
```

Out[30]:

```
<seaborn.axisgrid.PairGrid at 0x18a11273af0>
```



In [31]:

```
fuel.columns
```

Out[31]:

```
Index(['MODELYEAR', 'MAKE', 'MODEL', 'VEHICLECLASS', 'ENGINESIZE', 'CYLINDER
S',
       'TRANSMISSION', 'FUELTYPE', 'FUELCONSUMPTION_CITY',
       'FUELCONSUMPTION_HWY', 'FUELCONSUMPTION_COMB',
       'FUELCONSUMPTION_COMB_MPG', 'CO2EMISSIONS'],
      dtype='object')
```

In [32]:

```
reqFeatures = fuel[['ENGINESIZE', 'FUELCONSUMPTION_CITY','FUELCONSUMPTION_HWY', 'FUELCONSUM
```

In [33]:

```
sns.pairplot(reqFeatures)
```

...

In [34]:

```
corr = fuel.corr()
corr
```

Out[34]:

	MODELYEAR	ENGINESIZE	CYLINDERS	FUELCONSUMPTION_
MODELYEAR	NaN	NaN	NaN	
ENGINESIZE	NaN	1.000000	0.934011	0.83
CYLINDERS	NaN	0.934011	1.000000	0.79
FUELCONSUMPTION_CITY	NaN	0.832225	0.796473	1.00
FUELCONSUMPTION_HWY	NaN	0.778746	0.724594	0.96
FUELCONSUMPTION_COMB	NaN	0.819482	0.776788	0.99
FUELCONSUMPTION_COMB_MPG	NaN	-0.808554	-0.770430	-0.93
CO2EMISSIONS	NaN	0.874154	0.849685	0.89

In [38]:

```
x = reqFeatures.drop('CO2EMISSIONS', axis = 1)
y = reqFeatures['CO2EMISSIONS']
```

In [39]:

```
x.head()
```

Out[39]:

	ENGINESIZE	FUELCONSUMPTION_CITY	FUELCONSUMPTION_HWY	FUELCONSUMPTION_COM
0	2.0	9.9	6.7	8
1	2.4	11.2	7.7	9
2	1.5	6.0	5.8	5
3	3.5	12.7	9.1	11
4	3.5	12.1	8.7	10

In [40]:

```
y.head()
```

Out[40]:

```
0    196
1    221
2    136
3    255
4    244
Name: CO2EMISSIONS, dtype: int64
```

train = 75% test = 25%

In [41]:

```
from sklearn.model_selection import train_test_split


x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.25, random_state =
```

In [42]:

```
x_train
```

Out[42]:

	ENGINESIZE	FUELCONSUMPTION_CITY	FUELCONSUMPTION_HWY	FUELCONSUMPTION_C
309	3.6	13.2	8.7	
319	2.4	10.6	7.5	
261	5.3	16.0	11.1	
306	5.7	16.1	10.0	
737	5.5	15.1	10.7	
...	...	...	...	
330	3.6	14.2	9.4	
466	2.4	11.5	8.2	
121	4.4	16.2	10.9	
1044	1.8	10.0	6.9	
860	5.6	19.7	14.3	

800 rows × 4 columns

In [43]:

```
x_test
```

Out[43]:

	ENGINESIZE	FUELCONSUMPTION_CITY	FUELCONSUMPTION_HWY	FUELCONSUMPTION_(
732	4.7	15.4	10.4	
657	3.5	11.3	7.6	
168	3.6	15.1	9.9	
86	3.0	11.4	7.3	
411	2.0	10.5	7.1	
...	...	...	...	
110	4.4	15.0	9.8	
453	4.3	13.4	9.9	
554	5.0	15.7	10.3	
700	3.7	14.3	10.6	
1022	2.0	10.8	7.6	

267 rows × 4 columns

In [46]:

```
from sklearn.linear_model import LinearRegression
```

In [47]:

```
mlr = LinearRegression()
```

In [48]:

```
mlr.fit(x_train, y_train)
```

Out[48]:

```
LinearRegression()
```

In [52]:

```
x_test.iloc[0, :]
```

Out[52]:

```
ENGINESIZE               4.7
FUELCONSUMPTION_CITY    15.4
FUELCONSUMPTION_HWY     10.4
FUELCONSUMPTION_COMB    13.2
Name: 732, dtype: float64
```

In [54]:

```
y.iloc[0]
```

Out[54]:

196

In [55]:

```
mlr.predict([[4.7, 15.4, 10.4, 13.2]])
```

Out[55]:

array([299.12449791])

In [58]:

```
1 - mlr.score(x_train, y_train)
```

Out[58]:

0.14332101169339562

In [59]:

```
mlr.score(x_test, y_test)
```

Out[59]:

0.8659408158406949