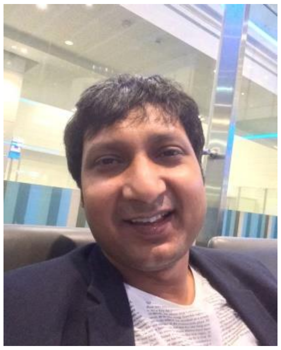
Polynomial Regression



Aparajito Sengupta (IISWBM)

PGD Analytics

coding: utf-8

Data Set : ***FUEL CONSUMPTION CO2 EMISSION***

Tool : Python

Jupyter Notebook

In[3]:

import pandas as pd

import pylab as pl

import numpy as np

import matplotlib.pyplot as plt

get_ipython().run_line_magic('matplotlib', 'inline')

#Lets Load the Dataset

In[9]:

df = pd.read_csv("C:/Users/Lenovo/Desktop/IISWBM BA/IBM DATA SCIENCE COURSE/Fuel Consumption CO2 Emission/FuelConsumptionCo2.csv")

In[12]:

#Checking the dataset

df.head(5)

df.tail(2)

In[13]:

#Let's select some features that we want to select as regression

cdf = df[["ENGINESIZE","CYLINDERS","FUELCONSUMPTION_COMB","CO2EMISSIONS"]]

cdf.head(2)

In[17]:

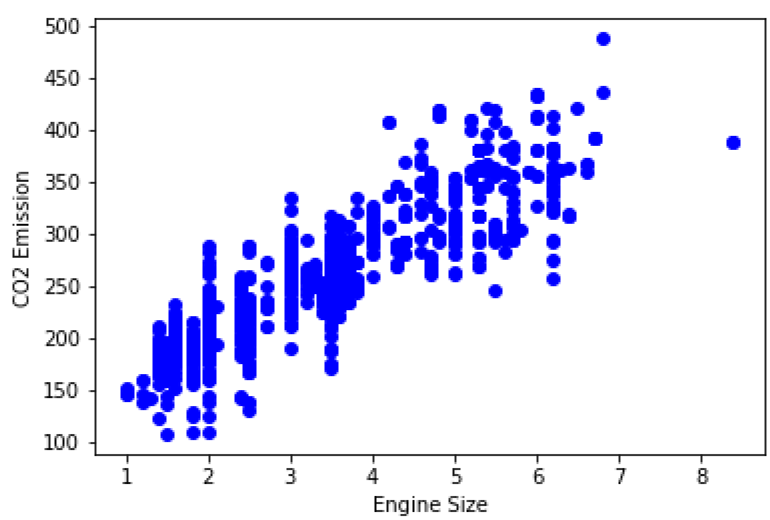
#Let's plot Emission wrt Engine Size

plt.scatter(cdf.ENGINESIZE, cdf.CO2EMISSIONS, color = 'blue')

plt.xlabel("Engine Size")

plt.ylabel("CO2 Emission")

plt.show()



In[18]:

#Creating Train & Test Dataset

msk = np.random.rand(len(df)) <0.8

train = cdf[msk]

test = cdf[~msk]

In[21]:

#Let us create ploynomial features using sklearn

from sklearn.preprocessing import PolynomialFeatures

from sklearn import linear_model

train_x =np.asanyarray(train[['ENGINESIZE']])

train_y = np.asanyarray(train[['CO2EMISSIONS']])

test_x = np.asanyarray(test[['ENGINESIZE']])

test_y = np.asanyarray(test[['CO2EMISSIONS']])

poly = PolynomialFeatures(degree = 2)

train_x_poly = poly.fit_transform(train_x)

train_x_poly #(fittransform takes x values and output list of data raised from the power 0 to power 2 as we have chosen the degree of our polynomial to 2)

Result :

```
array([[ 1.  , 1.5  , 2.25],
```

```
[ 1.  , 3.5 , 12.25],
[ 1.  , 3.5 , 12.25],
...,
[ 1.  , 3.  , 9.  ],
 [ 1.  , 3.  , 9.  ],
 [ 1.  , 3.2 , 10.24]])
```

In[22]:

#Polynomial Regression is a special case of multiple linear regression

#If we replece the higher powers with variables like x1,x2 the regression equation could be re written as

y = b + Ɵ1x1 + Ɵ2x2 and so on

#Thus, we can use LinearRegression function to solve it

clf = linear_model.LinearRegression()

train_y_ = clf.fit(train_x_poly, train_y)

#The Coeeficients

print('Coefficients:', clf.coef_)

print('Intercept:', clf.intercept_)

Result :

```
Coefficients: [[ 0. 48.22032646 -1.26481472]]
Intercept: [110.18314369]
```

In[24]:

#Let's plot it

plt.scatter(train.ENGINESIZE, train.CO2EMISSIONS, color ='blue')

XX = np.arange(0.0,10.0,0.1)

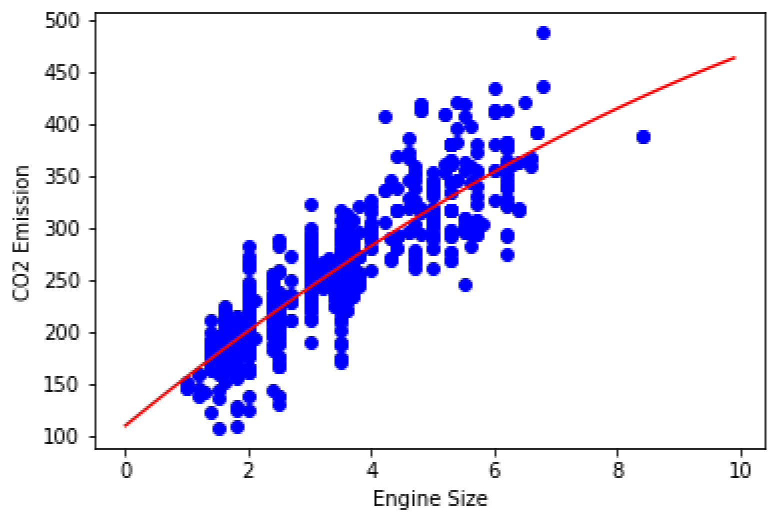
yy = clf.intercept_[0]+clf.coef_[0][1]*XX + clf.coef_[0][2]*np.power(XX,2)

plt.plot(XX,yy, '-r')

plt.xlabel("Engine Size")

plt.ylabel("CO2 Emission")

```
Text(0,0.5,'CO2 Emission')
```



In[30]:

#Evaluation

from sklearn.metrics import r2_score

test_x_poly = poly.fit_transform(test_x)

test_y_ = clf.predict(test_x_poly)

print ("Mean Absolute Error : %.2f" % np.mean(np.absolute(test_y_ - test_y)))

print("Residual Sum Of Squares (MSE) : %.2f" % np.mean((test_y_ - test_y) ** 2))

```
print("R2-Score : %.2f" % r2_score(test_y_ , test_y))
```

Result :

Mean Absolute Error : 23.87

Residual Sum Of Squares (MSE) : 963.56

R2-Score : 0.68