**Practical – 1: Application of Regression Analysis in Data Analytics.**

**1) Linear Regression –**

**Description:** We download a dataset that is related to fuel consumption and Carbon dioxide emission of cars. Then, we split our data into training and test sets, create a model using training set, evaluate model using test set, and finally use model to predict unknown value.

**Code:**

import matplotlib.pyplot as plt

import pandas as pd

import pylab as pl

import numpy as np

**# data import**

df = pd.read\_csv("data-genrator/FuelConsumptionCo2.csv")

cdf = df[['ENGINESIZE','CYLINDERS','FUELCONSUMPTION\_COMB','CO2EMISSIONS']]

**# relation visuallization**

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

plt.xlabel("Engine size")

plt.ylabel("Emission")

plt.show()

**# split data**

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

train = cdf[msk]

test = cdf[~msk]

**# model fitting**

from sklearn import linear\_model

regr = linear\_model.LinearRegression()

train\_x = np.asanyarray(train[['ENGINESIZE']])

train\_y = np.asanyarray(train[['CO2EMISSIONS']])

regr.fit (train\_x, train\_y)

print ('Coefficients: ', regr.coef\_)

print ('Intercept: ',regr.intercept\_)

**# Plot regression line**

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

plt.plot(train\_x, regr.coef\_[0][0]\*train\_x + regr.intercept\_[0], '-r')

plt.xlabel("Engine size")

plt.ylabel("Emission")

**# r2 calculation**

from sklearn.metrics import r2\_score

test\_x = np.asanyarray(test[['ENGINESIZE']])

test\_y = np.asanyarray(test[['CO2EMISSIONS']])

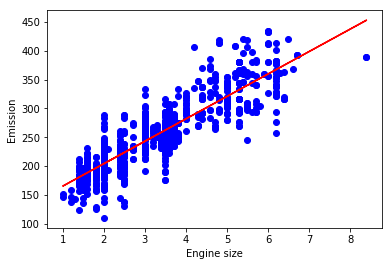
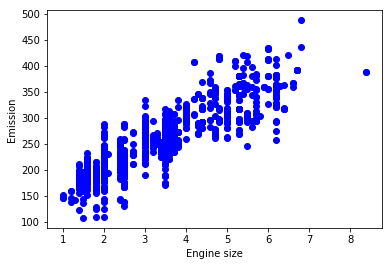
test\_y\_ = regr.predict(test\_x)

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) )

**Output:**



Coefficients: [[38.89929855]]

Intercept: [126.30857997]

Mean absolute error: 25.23

Residual sum of squares (MSE): 1090.08

R2-score: 0.64