**Code:**

from random import random, seed

import math

import numpy as np

import matplotlib.pyplot as plt

data\_m = 3

data\_c = 4

def err(r): return r\*(- 1 + 2\*random())

learningRate = 0.0001

Iterations = 5000

Points = 100

data = np.array([[x, data\_m\*x + data\_c + err(2)] for x in range(1, Points+1)])

w = random()

b = random()

MSEgraph = []

MSE = 0

prevMSE = 0

for j in range(Iterations):

prevMSE = MSE

MSE = 0

for i in range(Points):

predicted = w \* data[i, 0] + b

expected = data[i, 1]

error = expected-predicted

w += learningRate\*error\*data[i, 0]

b += learningRate\*error

MSE += error\*\*2

MSE = round(MSE/Points, 2)

if MSE != prevMSE:

MSEgraph.append([len(MSEgraph), MSE])

print("iteration({0}): MSE = {1}".format(j, MSE))

MSEgraph = np.array(MSEgraph)

finalPoints = []

for i in range(Points):

finalPoints.append(w\*data[i, 0] + b)

print("FINAL VALUES:\nW:{0}\nB:{1}\nMSE:{2}".format(w, b, MSE))

plt.figure(1)

plt.plot(data[:, 0], data[:, 1])

plt.plot(data[:, 0], finalPoints)

plt.xlabel('X\_data')

plt.ylabel('Y\_data')

plt.title('Gradient Descent: Linear Regression')

plt.figure(2)

plt.plot(MSEgraph[:, 0], MSEgraph[:, 1])

plt.title('Error vs iterations plot')

plt.xlabel('Iterations')

plt.ylabel('Mean Sqaured Error')

plt.show()

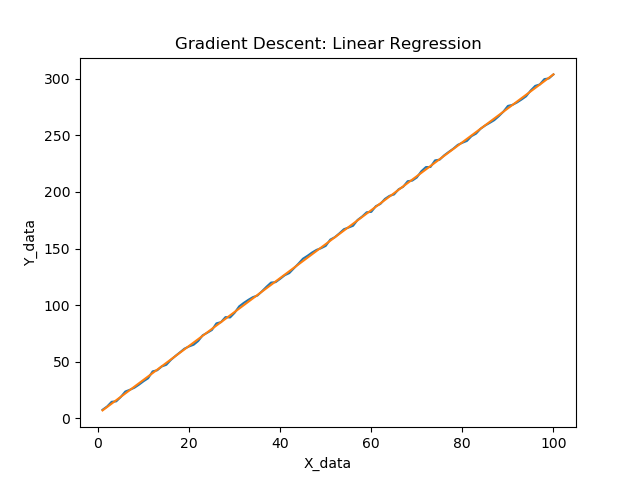
**Outputs:**

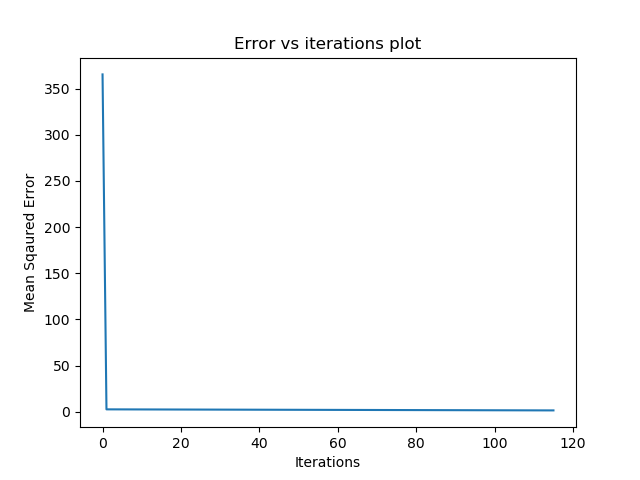
FINAL VALUES:

W:2.99818031340653

# B:3.985904571671755

MSE:1.52

****

****