**Code:**

import random

import numpy as np

import matplotlib.pyplot as plt

iterations = 10000

learningRate = 0.001

mse = []

def gradient\_descent(x,y):

m\_curr = b\_curr = 0

n= len(x)

prev\_mse = 0

for i in range(iterations):

y\_predicted = m\_curr \* x + b\_curr

#printing line progress

if not i%2000:

plt.plot(x,y\_predicted)

md = -(2/n)\*sum(x\*(y-y\_predicted))

bd = -(2/n)\*sum(y-y\_predicted)

curr\_mse = (1/n)\*sum((y-y\_predicted)\*\*2)

if(round(curr\_mse)!= round(prev\_mse)):

print("itr: {} => appending to mse: {}".format(i,curr\_mse))

mse.append(curr\_mse)

m\_curr = m\_curr - learningRate \* md

b\_curr = b\_curr - learningRate \* bd

prev\_mse = curr\_mse

if i==0 or i==iterations-1:

print("itr:{} \n m: {}, b: {}".format(i,m\_curr,b\_curr))

newY = []

for i in range(n):

newY.append(m\_curr\*i + b\_curr)

newY = np.array(newY)

return newY

random.seed(1)

dataSet = 50

dataLine\_m = 3

dataLine\_b = 4

weight = 2

x = []

y = []

for c in range(dataSet):

x.append(c)

y.append(dataLine\_m\*c+dataLine\_b + weight\*random.random()\*(- 1 + 2\*random.random()))

x = np.array(x)

y = np.array(y)

fitLine = gradient\_descent(x,y)

plt.scatter(x,y)

plt.plot(x,fitLine,'-r')

plt.xlabel('X\_data')

plt.ylabel('Y\_data')

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

plt.show()

err\_x = np.linspace(0,len(mse),len(mse))

plt.figure(2)

plt.plot(err\_x,mse);

plt.title('Error vs iterations plot')

plt.xlabel('Iterations')

plt.ylabel('Mean Sqaured Error')

plt.show()

**Output:**



