import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error, r2\_score

%matplotlib notebook

#generating random data with slope=2 and intercept=-2

delta = np.random.uniform(-10,10, size=(20,)) #for generating random data

x = np.arange(20)

y = 2\*x-2+delta #y=mx+b; m=slope; b-y-intercept; delta=random scatter

plt.scatter(x,y, c = 'b', label='random data') #scatter plot

#print(x)

#print(y)

#generating the desired line which is y=2x-2

y = 2\*x-2 #y=mx+b; m=slope; b-y-intercept

plt.plot(x,y, c='r', linewidth=1, label = 'desired') #plot line

#generating the linear regression from the random data

x = x.reshape(-1, 1)

model = LinearRegression()

model.fit(x, y)

y\_predict = model.predict(x)

#print(y\_predict)

plt.plot(x, y\_predict, 'g', linewidth=3, linestyle=':', dash\_capstyle='round', label='output') #plot linear regression

#styling the plot

plt.legend(loc=2)

plt.xticks(np.arange(0, 21, 1))

plt.yticks(np.arange(-10, 50, 10))

plt.show()

plt.savefig('MP4.png')

print("Linear regression coefficient = ", model.coef\_[0])

print("Linear regression intercept = ", model.intercept\_)

print("coefficient of determination (R\u00b2) = ", r2\_score(y, y\_predict))

print("root mean squared error (RMSE) = ", np.sqrt(mean\_squared\_error(y, y\_predict)))

#Rane Gillian Villanueva

#2018101140

