Program No:6

Date:08-12-2021

Aim: Program to implement linear and multiple regression techniques using any standard dataset available in the public domain (without using built in function)

Program code

import numpy as np

import matplotlib.pyplot as plt

def estimate\_coef(x, y):

# number of observations/points

n = np.size(x)

# mean of x and y vector

m\_x = np.mean(x)

m\_y = np.mean(y)

# calculating cross-deviation and deviation about x

SS\_xy = np.sum(y\*x) - n\*m\_y\*m\_x

SS\_xx = np.sum(x\*x) - n\*m\_x\*m\_x

# calculating regression coefficients

b\_1 = SS\_xy / SS\_xx

b\_0 = m\_y - b\_1\*m\_x

return (b\_0, b\_1)

def plot\_regression\_line(x, y, b):

# plotting the actual points as scatter plot

plt.scatter(x, y, color = &quot;m&quot;, marker = &quot;o&quot;, s = 30)

# predicted response vector

y\_pred = b[0] + b[1]\*x

# plotting the regression line

plt.plot(x, y\_pred, color = &quot;g&quot;)

# putting labels

plt.xlabel(&#39;x&#39;)

plt.ylabel(&#39;y&#39;)

# function to show plot

plt.show()

def main():

# observations / data

x = np.array([12,45,45,77,3,19,12,45,33,76])

y = np.array([98,45,66,12,34,88,61,89,45,61])

# estimating coefficients

b = estimate\_coef(x, y)

print(“Estimated coefficients:\nb\_0 = {} \

\nb\_1 = {}”.format(b[0], b[1]))

# plotting regression line

plot\_regression\_line(x, y, b)

if \_\_name\_\_ =="\_\_main\_\_":

main()

OUTPUT

