Assignment 3 Gradient descent for linear regression

By Neeraj Sharma

1. Code of the gradient descent.

```
import numpy as np

import matplotlib.pyplot as plt


x = np.array([20, 43, 63, 26, 53, 31, 58, 46, 58, 70, 46, 53, 60, 20, 63, 43, 26, 19, 31, 23])

y = np.array([120, 128, 141, 126, 134, 128, 136, 132, 140, 144, 128, 136, 146, 124, 143, 130, 124, 121,
126, 123])

learning_rate = 0.00046

iterations = 1

cost = 1


def gradient_descent(x, y, learning_rate, iterations, cost):

 m_curr = b_curr = 0

 n = len(x)


 while cost >= 0.100:

  y_predicted = m_curr * x + b_curr

  cost = 1/n * sum([val**2 for val in (y - y_predicted)])

  cost = round(cost, 3)

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

  bd = -(2/n)*sum(y-y_predicted)

  m_curr = m_curr - learning_rate * md

  b_curr = b_curr - learning_rate * bd

  iterations +=1
```

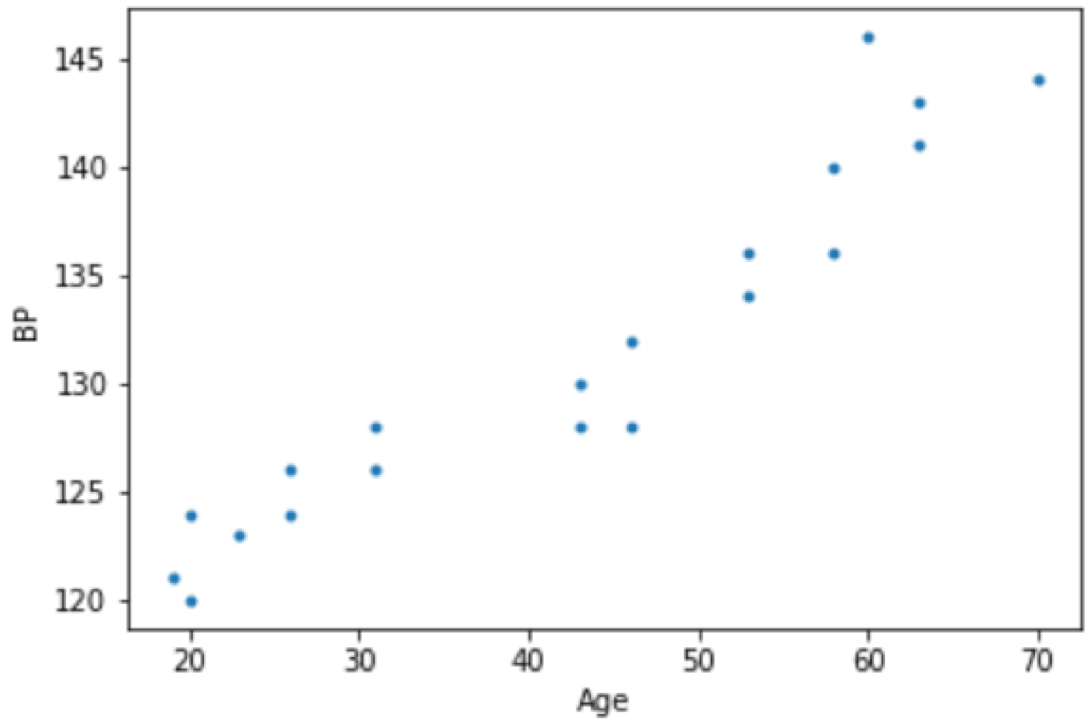
```
    print("m {}, b {}, cost {}, iteration {}".format(m_curr, b_curr, cost, iterations))
```

```
import matplotlib.pyplot as plt
m = 48
c = 113
plt.ylabel('Cost')
plt.show
plt.Xlablel('Age')
plt.Ylable('BP')
plt.plot(x, y, '.')
plt.show
```

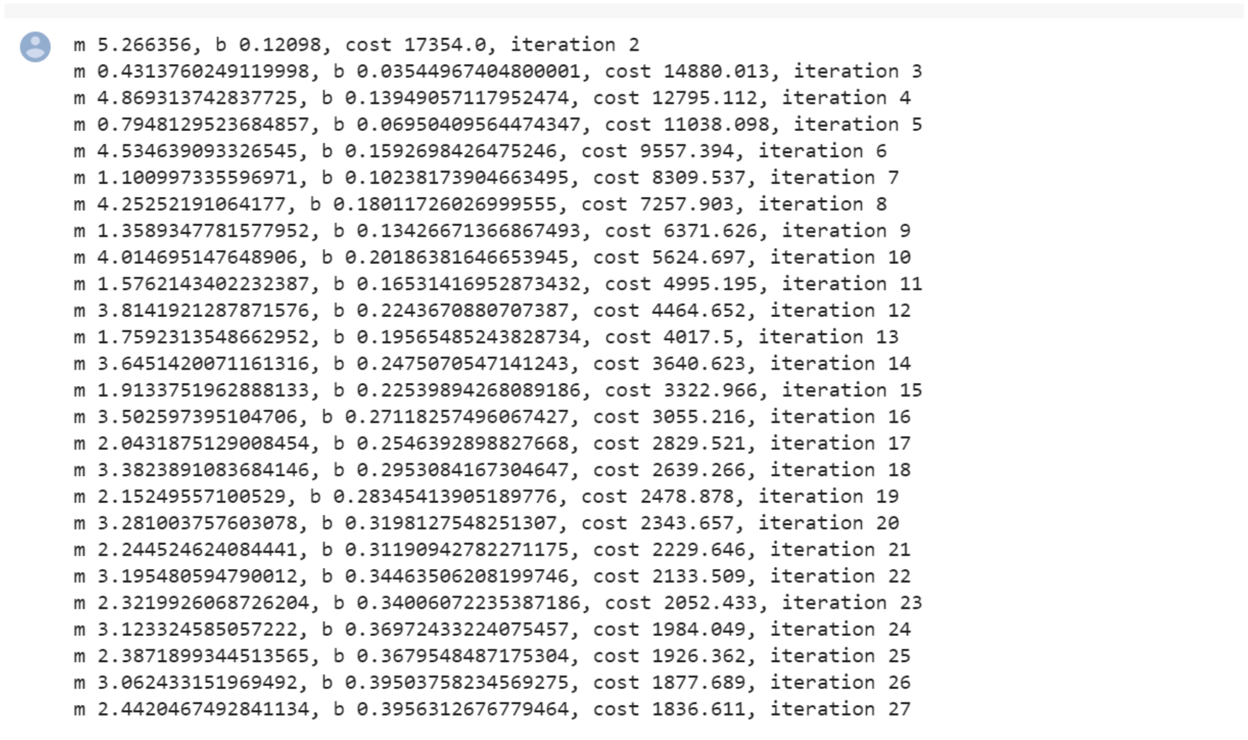
```
gradient_descent(x, y, learning_rate, iterations, cost)
```

Plotting the values given in the assignments

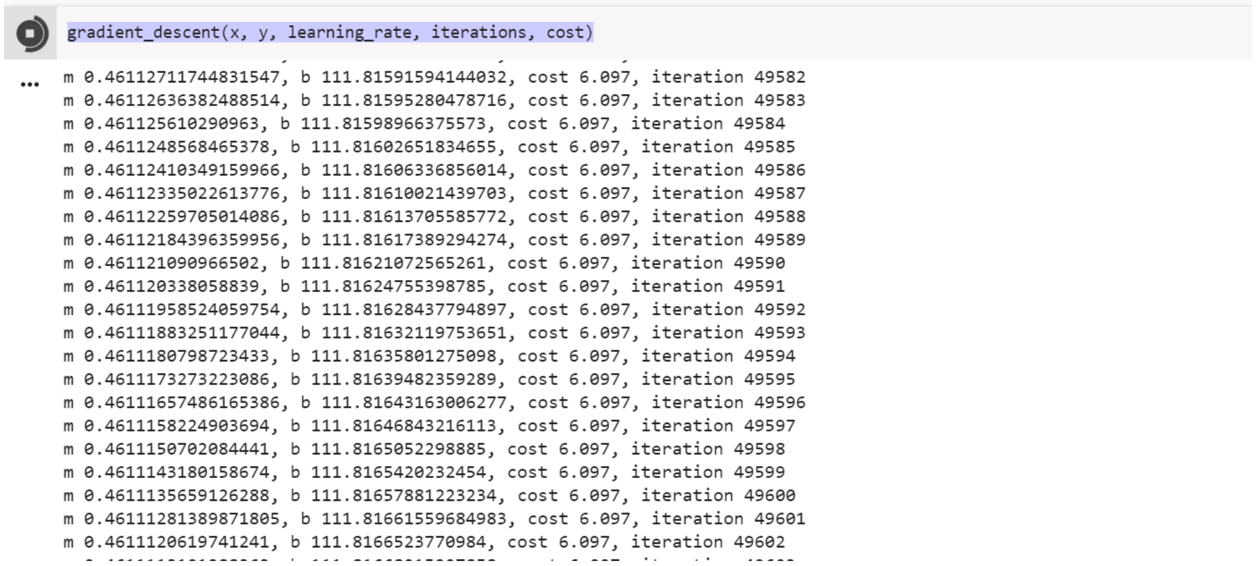
```
<function matplotlib.pyplot.show>
```



2. Screen shots of Gradient decent running

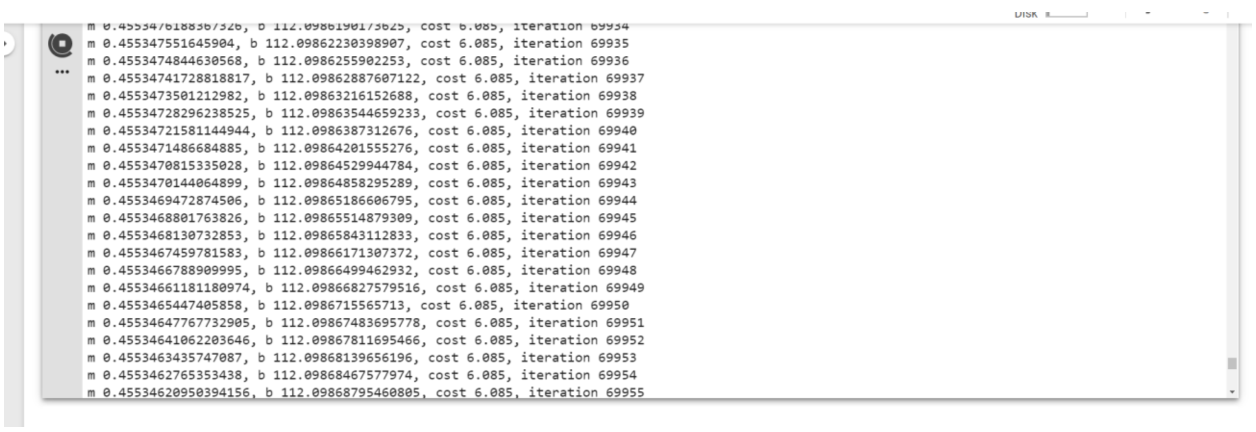


```
m 5.266356, b 0.12098, cost 17354.0, iteration 2
m 0.4313760249119998, b 0.03544967404800001, cost 14880.013, iteration 3
m 4.869313742837725, b 0.13949057117952474, cost 12795.112, iteration 4
m 0.7948129523684857, b 0.06950409564474347, cost 11038.098, iteration 5
m 4.534639093326545, b 0.1592698426475246, cost 9557.394, iteration 6
m 1.100997335596971, b 0.10238173904663495, cost 8309.537, iteration 7
m 4.25252191064177, b 0.1801172602699555, cost 7257.903, iteration 8
m 1.3589347781577952, b 0.13426671366867493, cost 6371.626, iteration 9
m 4.014695147648906, b 0.20186381646653945, cost 5624.697, iteration 10
m 1.5762143402232387, b 0.16531416952873432, cost 4995.195, iteration 11
m 3.8141921287871576, b 0.2243670880707387, cost 4464.652, iteration 12
m 1.7592313548662952, b 0.19565485243828734, cost 4017.5, iteration 13
m 3.6451420071161316, b 0.2475070547141243, cost 3640.623, iteration 14
m 1.9133751962888133, b 0.22539894268089186, cost 3322.966, iteration 15
m 3.502597395104706, b 0.27118257496067427, cost 3055.216, iteration 16
m 2.0431875129008454, b 0.2546392898827668, cost 2829.521, iteration 17
m 3.3823891083684146, b 0.2953084167304647, cost 2639.266, iteration 18
m 2.15249557100529, b 0.28345413905189776, cost 2478.878, iteration 19
m 3.281003757603078, b 0.3198127548251307, cost 2343.657, iteration 20
m 2.244524624084441, b 0.31190942782271175, cost 2229.646, iteration 21
m 3.195480594790012, b 0.34463506208199746, cost 2133.509, iteration 22
m 2.3219926068726204, b 0.34006072235387186, cost 2052.433, iteration 23
m 3.123324585057222, b 0.36972433224075457, cost 1984.049, iteration 24
m 2.3871899344513565, b 0.3679548487175304, cost 1926.362, iteration 25
m 3.062433151969492, b 0.39503758234569275, cost 1877.689, iteration 26
m 2.4420467492841134, b 0.3956312676779464, cost 1836.611, iteration 27
```

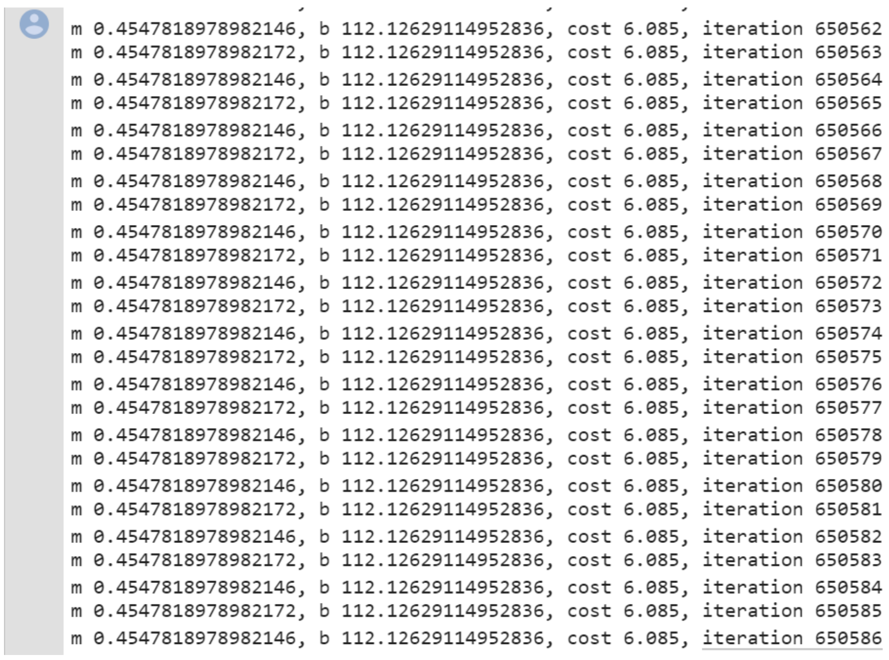


```
gradient_descent(x, y, learning_rate, iterations, cost)
```

```
m 0.46112711744831547, b 111.81591594144032, cost 6.097, iteration 49582
m 0.46112636382488514, b 111.81595280478716, cost 6.097, iteration 49583
m 0.461125610290963, b 111.81598966375573, cost 6.097, iteration 49584
m 0.4611248568465378, b 111.81602651834655, cost 6.097, iteration 49585
m 0.46112410349159966, b 111.81606336856014, cost 6.097, iteration 49586
m 0.46112335022613776, b 111.81610021439703, cost 6.097, iteration 49587
m 0.46112259705014086, b 111.81613705585772, cost 6.097, iteration 49588
m 0.46112184396359956, b 111.81617389294274, cost 6.097, iteration 49589
m 0.461121090966502, b 111.81621072565261, cost 6.097, iteration 49590
m 0.461120338058839, b 111.81624755398785, cost 6.097, iteration 49591
m 0.46111958524059754, b 111.81628437794897, cost 6.097, iteration 49592
m 0.46111883251177044, b 111.81632119753651, cost 6.097, iteration 49593
m 0.4611180798723433, b 111.81635801275098, cost 6.097, iteration 49594
m 0.4611173273223086, b 111.81639482359289, cost 6.097, iteration 49595
m 0.46111657486165386, b 111.81643163006277, cost 6.097, iteration 49596
m 0.4611158224903694, b 111.81646843216113, cost 6.097, iteration 49597
m 0.4611150702084441, b 111.8165052298885, cost 6.097, iteration 49598
m 0.4611143180158674, b 111.8165420232454, cost 6.097, iteration 49599
m 0.4611135659126288, b 111.81657881223234, cost 6.097, iteration 49600
m 0.46111281389871805, b 111.81661559684983, cost 6.097, iteration 49601
m 0.4611120619741241, b 111.8166523770984, cost 6.097, iteration 49602
```



```
m 0.4553476188367326, b 112.0986190173625, cost 6.085, iteration 69934
m 0.455347551645904, b 112.09862230398907, cost 6.085, iteration 69935
m 0.4553474844630568, b 112.0986255902253, cost 6.085, iteration 69936
m 0.45534741728818817, b 112.09862887607122, cost 6.085, iteration 69937
m 0.4553473501212982, b 112.09863216152688, cost 6.085, iteration 69938
m 0.45534728296238525, b 112.09863544659233, cost 6.085, iteration 69939
m 0.45534721581144944, b 112.0986387312676, cost 6.085, iteration 69940
m 0.4553471486684885, b 112.09864201555276, cost 6.085, iteration 69941
m 0.4553470815335028, b 112.09864529944784, cost 6.085, iteration 69942
m 0.4553470144064899, b 112.09864858295289, cost 6.085, iteration 69943
m 0.4553469472874506, b 112.09865186606795, cost 6.085, iteration 69944
m 0.4553468801763826, b 112.09865514879309, cost 6.085, iteration 69945
m 0.4553468130732853, b 112.09865843112833, cost 6.085, iteration 69946
m 0.4553467459781583, b 112.09866171307372, cost 6.085, iteration 69947
m 0.4553466788909995, b 112.09866499462932, cost 6.085, iteration 69948
m 0.45534661181180974, b 112.09866827579516, cost 6.085, iteration 69949
m 0.4553465447405858, b 112.0986715565713, cost 6.085, iteration 69950
m 0.45534647767732905, b 112.09867483695778, cost 6.085, iteration 69951
m 0.45534641062203646, b 112.09867811695466, cost 6.085, iteration 69952
m 0.4553463435747087, b 112.09868139656196, cost 6.085, iteration 69953
m 0.4553462765353438, b 112.09868467577974, cost 6.085, iteration 69954
m 0.45534620950394156, b 112.09868795460805, cost 6.085, iteration 69955
```



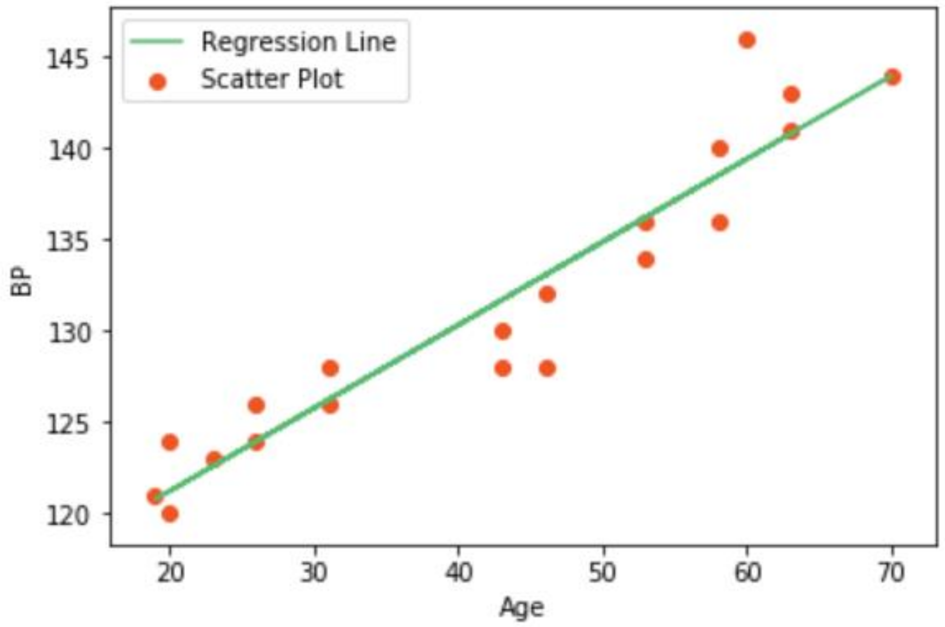
```
m 0.4547818978982146, b 112.12629114952836, cost 6.085, iteration 650562
m 0.4547818978982172, b 112.12629114952836, cost 6.085, iteration 650563
m 0.4547818978982146, b 112.12629114952836, cost 6.085, iteration 650564
m 0.4547818978982172, b 112.12629114952836, cost 6.085, iteration 650565
m 0.4547818978982146, b 112.12629114952836, cost 6.085, iteration 650566
m 0.4547818978982172, b 112.12629114952836, cost 6.085, iteration 650567
m 0.4547818978982146, b 112.12629114952836, cost 6.085, iteration 650568
m 0.4547818978982172, b 112.12629114952836, cost 6.085, iteration 650569
m 0.4547818978982146, b 112.12629114952836, cost 6.085, iteration 650570
m 0.4547818978982172, b 112.12629114952836, cost 6.085, iteration 650571
m 0.4547818978982146, b 112.12629114952836, cost 6.085, iteration 650572
m 0.4547818978982172, b 112.12629114952836, cost 6.085, iteration 650573
m 0.4547818978982146, b 112.12629114952836, cost 6.085, iteration 650574
m 0.4547818978982172, b 112.12629114952836, cost 6.085, iteration 650575
m 0.4547818978982146, b 112.12629114952836, cost 6.085, iteration 650576
m 0.4547818978982172, b 112.12629114952836, cost 6.085, iteration 650577
m 0.4547818978982146, b 112.12629114952836, cost 6.085, iteration 650578
m 0.4547818978982172, b 112.12629114952836, cost 6.085, iteration 650579
m 0.4547818978982146, b 112.12629114952836, cost 6.085, iteration 650580
m 0.4547818978982172, b 112.12629114952836, cost 6.085, iteration 650581
m 0.4547818978982146, b 112.12629114952836, cost 6.085, iteration 650582
m 0.4547818978982172, b 112.12629114952836, cost 6.085, iteration 650583
m 0.4547818978982146, b 112.12629114952836, cost 6.085, iteration 650584
m 0.4547818978982172, b 112.12629114952836, cost 6.085, iteration 650585
m 0.4547818978982146, b 112.12629114952836, cost 6.085, iteration 650586
```

Cost looks like its repeating at 6.085 because I rounded it to 3 digit, other wise it would have shown changed value

It stopped in C = 6.085 at the end because after that I believe in the next iteration it will go beyond the lowest point in curve.

3. Plotting Regression Line using the new m and b value

```
# Plotting Values and Regression Line
import numpy as np
import matplotlib.pyplot as plt

m = 0.4547818978982146
b = 112.12629114952836
X = np.array([20, 43, 63, 26, 53, 31, 58, 46, 58, 70, 46, 53, 60, 20, 63, 43, 26, 19, 31, 23])
y = np.array([120, 128, 141, 126, 134, 128, 136, 132, 140, 144, 128, 136, 146, 124, 143, 130, 124, 121, 
126, 123])
iteration = 650586

# Plotting Values and Regression Line
max_x = np.max(X) + 100
min_x = np.min(X) - 100

# Calculating line values x and y
x = np.linspace(min_x, max_x)
y_pre = b + m * X

# Ploting Line
plt.plot(X, y_pre, color='#58b970', label='Regression Line')

# Ploting Scatter Points
plt.scatter(X, y, c='#ef5423', label='Scatter Plot')

plt.xlabel('Age')
plt.ylabel('BP')
plt.legend()
plt.show()
```



Regression Line
Scatter Plot
145
140
135
130
125
120
BP
20
30
40
50
60
70
Age