```
import numpy as np
import pandas as pd
from sklearn.linear_model import LinearRegression
import math

def predict_using_sklearn():
  df = pd.read_csv('/content/drive/MyDrive/Salary_dataset.csv')
  r = LinearRegression()
  r.fit(df[['YearsExperience']],df.Salary)
  return r.coef_, r.intercept_

def gradient_descent(x, y):
  m_curr = 0
  b_curr = 0
  iterations = 100
  n = len(x)
  learning_rate = 0.0002

  cost_previous = 0
  for i in range(iterations):
    y_predicted = m_curr * x + b_curr
    cost = (1/n) * sum([val**2 for val in (y-y_predicted)])
    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
    if math.isclose(cost, cost_previous, rel_tol=1e-20):
      break
    cost_previous = cost
    print ("m {}, b {}, cost {}, iteration {}".format(m_curr,b_curr,cost, i))

  return m_curr, b_curr

if __name__ == "__main__":
  df = pd.read_csv('/content/drive/MyDrive/Salary_dataset.csv')
  x = np.array(df.YearsExperience)
  y = np.array(df.Salary)

  m, b = gradient_descent(x, y)
  print("Using gradient descent function: Coef {} Intercept {}".format(m, b))

  m_sklearn, b_sklearn = predict_using_sklearn()
  print("Using sklearn: Coef {} Intercept {}".format(m_sklearn,b_sklearn))
```

```
m 194.00176533333337, b 30.401600000000002, cost 6503107277.733334, iteration 0
m 385.05954261211883, b 60.37096087079823, cost 6311763265.6603155, iteration 1
m 573.2179439998646, b 89.9146302235138, cost 6126173846.451653, iteration 2
m 758.520905628157, b 119.03905645001669, cost 5946165936.795322, iteration 3
m 941.0116978409328, b 147.75059022644984, cost 5771571659.28162, iteration 4
m 1120.7329352835138, b 176.0554859939677, cost 5602228185.822952, iteration 5
m 1297.726586838758, b 203.9599034170362, cost 5437977585.78319, iteration 6
m 1472.0339854126423, b 231.46990881963453, cost 5278666678.674858, iteration 7
m 1643.695837571559, b 258.5914765996932, cost 5124146891.286864, iteration 8
m 1812.7522330335735, b 285.33049062209835, cost 4974274119.109419, iteration 9
m 1979.2426540158558, b 311.6927455905875, cost 4828908591.926963, iteration 10
m 2143.2059844404657, b 337.6839483988556, cost 4687914743.4537115, iteration 11
m 2304.6805190006367, b 363.3097194611876, cost 4551161084.890215, iteration 12
m 2463.703972089675, b 388.57559402292713, cost 4418520082.283026, iteration 13
m 2620.3134865945544, b 413.48702345108643, cost 4289868037.573075, iteration 14
m 2774.5456425562547, b 438.0493765053999, cost 4165084973.221792, iteration 15
m 2926.4364656988682, b 462.2679405901159, cost 4044054520.3073945, iteration 16
m 3076.0214358294597, b 486.14792298681994, cost 3926663809.986942, iteration 17
m 3223.335495110636, b 509.6944520685758, cost 3812803368.2229323, iteration 18
m 3368.413056207761, b 532.9125784956689, cost 3702367013.6762695, iteration 19
m 3511.2880103127086, b 555.8072763932287, cost 3595251758.670319, iteration 20
m 3651.9937350460286, b 578.3834445110076, cost 3491357713.1337466, iteration 21
m 3790.5631022393673, b 600.6459073655835, cost 3390587991.4324775, iteration 22
m 3927.0284855999557, b 622.599416365255, cost 3292848622.0039287, iteration 23
m 4061.4217682589547, b 644.2486509178898, cost 3198048459.709212, iteration 24
m 4193.774350205415, b 665.5982195219859, cost 3106099100.821554, iteration 25
m 4324.11715560759, b 686.652660841199, cost 3016914800.5716453, iteration 26
m 4452.480640023296, b 707.4164447625869, cost 2930412393.173013, iteration 27
m 4578.8947975010215, b 727.8939734388181, cost 2846511214.2528405, iteration 28
```

```
m 4703.389167573423, b 748.089582314587, cost 2765133025.615841, iteration 29
m 4825.992842144843, b 768.0075411374755, cost 2686201942.2710705, iteration 30
m 4946.734472274464, b 787.6520549534962, cost 2609644361.6535635, iteration 31
m 5065.6422748566665, b 807.0272650875498, cost 2535388894.9748154, iteration 32
m 5182.744039200164, b 826.1372501090252, cost 2463366300.6380644, iteration 33
m 5298.067133507431, b 844.9860267827669, cost 2393509419.65626, iteration 34
m 5411.6385112559565, b 863.5775510056324, cost 2325753113.0125093, iteration 35
m 5523.484717482788, b 881.9157187288572, cost 2260034200.904549, iteration 36
m 5633.631894973851, b 900.004366866443, cost 2196291403.816593, iteration 37
m 5742.105790359469, b 917.8472741897797, cost 2134465285.3635855, iteration 38
m 5848.931760117528, b 935.4481622087121, cost 2074498196.8545542, iteration 39
m 5954.134776485658, b 952.8106960392541, cost 2016334223.5233557, iteration 40
m 6057.739433283834, b 969.9384852581549, cost 1959919132.3766603, iteration 41
m 6159.769951648738, b 986.8350847445143, cost 1905200321.6105368, iteration 42
m 6260.250185681231, b 1003.5039955086464, cost 1852126771.54846, iteration 43
m 6359.203628008239, b 1019.9486655083812, cost 1800648997.0549667, iteration 44
m 6456.653415260363, b 1036.1724904529974, cost 1750719001.3806064, iteration 45
m 6552.622333466476, b 1052.1788145949724, cost 1702290231.395091, iteration 46
m 6647.132823366587, b 1067.970931509735, cost 1655317534.1669433, iteration 47
m 6740.206985644178, b 1083.5520848636013, cost 1609757114.8491013, iteration 48
m 6831.866586079265, b 1098.9254691700744, cost 1565566495.8312173, iteration 49
m 6922.133060623373, b 1114.0942305346828, cost 1522704477.1205604, iteration 50
m 7011.027520397596, b 1129.0614673885325, cost 1481131097.9145498, iteration 51
m 7098.570756614931, b 1143.8302312107428, cost 1440807599.3290925, iteration 52
m 7184.783245428012, b 1158.4035272399349, cost 1401696388.247956, iteration 53
m 7269.685152703391, b 1172.7843151749387, cost 1363761002.2594616, iteration 54
m 7353.296338723466, b 1186.9755098648816, cost 1326966075.6477869, iteration 55
m 7435.636362817161, b 1200.9799819888196, cost 1291277306.4071689, iteration 56
```