Librerías

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

Divisón del data set en entrenamiento y prueba

```
# Función que divide los datasets en entrenamiento y prueba
import random

def split_dataset(porcentaje, dataset):

  columna0 = dataset.columns[0]
  columna1 = dataset.columns[1]
  X = dataset[columna0].values
  Y = dataset[columna1].values

  cantidad = int((len(X) * porcentaje) / 100)
  lst = list(range(0,len(X)))
  arr_train_X = np.array([])
  arr_train_Y = np.array([])

  for i in range(0, cantidad):
    indice = random.choice(lst)

    # Creación de DataFrame de entrenamiento de X
    arr_train_X = np.append(arr_train_X, X[indice])
    train_X = pd.DataFrame(arr_train_X, columns = [columna0])

    # Creación de DataFrame de entrenamiento de Y
    arr_train_Y = np.append(arr_train_Y, Y[indice])
```

```
    train_Y = pd.DataFrame(arr_train_Y, columns = [columna1])

    lst.remove(indice)

  test_X = pd.DataFrame(X[lst], columns = [columna0])
  test_Y = pd.DataFrame(Y[lst], columns = [columna1])

  return train_X, train_Y, test_X, test_Y
```

Construcción de modelo de regresión

$\mathbf{y}$ = β1*x + β0

Fit del modelo

```
def fit(dataset_X, dataset_Y):

  columna0 = dataset_X.columns[0]
  columna1 = dataset_Y.columns[0]
  X = dataset_X[columna0].values
  Y = dataset_Y[columna1].values

  # Cálculo de los coeficientes
  suma_x = X.sum()
  suma_y = Y.sum()

  suma_xy = (X*Y).sum()

  suma_x2 = (X**2).sum()
  suma_y2 = (Y**2).sum()

  promedio_x = np.mean(X)
  promedio_y = np.mean(Y)
```

```
19
20   n = len(X)
21
22   b0 = (suma_y*suma_x2 - suma_x*suma_xy) / (n*suma_x2 - (suma_x**2))
23   b1 = (n*suma_xy - suma_x*suma_y) / (n*suma_x2 - (suma_x)**2)
24   coeficientes = [b0, b1]
25   train_y = b1*X + b0
26
27   return X, Y, coeficientes, train_y
```

Entrenamiento

```
1 def regresion(dataset_X, dataset_Y, fit):
2   X, Y, coeficientes, train_y = fit
3
4   X_test = dataset_X[dataset_X.columns[0]].values
5   Y_test = dataset_Y[dataset_Y.columns[0]].values
6
7   test_y = coeficientes[1]*X_test + coeficientes[0]
8   resultado = pd.DataFrame({test_X.columns[0] : test_X.iloc[:, 0], test_Y.columns[0]: test_Y.iloc[:, 0], 'Predicción '+ t
9
10  return test_y, resultado
```

Evaluación

```
1 # Mean Squared Error, Mean Absolute Error, Root Mean Squared Error, Mean Percent Absolute Error, Sum of Squared Errors, C
2 def evaluacion(resultado):
3   n = len(resultado)
4   resultado['Error'] = resultado.iloc[:,1] - resultado.iloc[:,2]
5   resultado['Error Absoluto'] = abs(resultado.iloc[:,1] - resultado.iloc[:,2])
6   resultado['Squared Error'] = (resultado.iloc[:,1] - resultado.iloc[:,2])**2
7   media_y = resultado.iloc[:,1].mean()
8
```

```
  SSE = resultado['Squared Error'].sum()
  MSE = SSE / n
  MAE = resultado['Error Absoluto'].sum() / n
  RMSE = MSE**(1/2)
  MAPE = (abs(resultado['Error']/resultado.iloc[:,1]).sum()) / n
  R2 = (((resultado.iloc[:,2] - media_y)**2).sum()) / (((resultado.iloc[:,1] - media_y)**2).sum())

  print('Mean Squared Error: ', MSE)
  print('Mean Absolute Error: ', MAE)
  print('Root Mean Squared Error', RMSE)
  print('Mean Percent Absolute Error', MAPE)
  print('Sum of Squared Errors', SSE)
  print('Coeficiente de determinación', R2)

  return resultado, MSE, MAE, RMSE, MAPE, SSE, R2
```

Cargar datos

```
data = pd.DataFrame()
data['income'] = np.array([3.86264742, 4.97938138, 4.92395694, 3.21437244, 7.19640925,
       3.72964348, 4.67451739, 4.49810382, 3.12163053, 4.63991444,
       4.63283951, 2.7731789 , 7.11947859, 7.4666532 , 2.11774233,
       2.55916582, 2.35479322, 2.38815725, 4.75568027, 1.99427505,
       7.31091603, 3.52831896, 2.42875167, 3.54274787, 5.22720124,
       6.69199314, 3.90040994, 2.29105548, 2.38051271, 2.54960878,
       6.93329583, 1.85564517, 3.58902314, 6.82647791, 2.07060188,
       5.22420527, 2.24311363, 7.07616637, 4.1906725 , 1.95648612,
       5.06175818, 3.98218993, 3.06505862, 3.68287749, 3.78942929,
       5.35871566, 5.19611978, 5.24118957, 7.10161959, 3.42402109,
       2.25339902, 5.3703369 , 6.225606  , 5.48286223, 4.03417211,
       6.51021872, 6.02921386, 6.94911289, 7.19503729, 2.7573385 ,
       6.95607948, 4.67019258, 6.36829267, 6.16668117, 6.07415829,
       5.48471897, 1.58957474, 1.68047393, 5.49994796, 4.0438909 ,
       5.00509324, 4.86358173, 1.50627504, 2.86466385, 5.87790629,
       6.48398379, 4.93803659, 5.62543442, 7.22826521, 5.33746029,
       2.82582659, 5.93136667, 3.5202552 , 3.23994059, 3.49838637,
```

```
19        7.18611226, 4.71916607, 3.59480196, 3.23394243, 1.51415322,
20        4.00253706, 6.19810376, 2.28065086, 2.18986551, 3.43415113,
21        5.93226968, 5.30783944, 5.66434491, 7.43924761, 2.13470215,
22        6.50127472, 3.65118317, 2.28649539, 4.74885914, 5.45916076,
23        3.43306477, 7.17639965, 5.50639512, 3.09761573, 4.64755599,
24        1.82830634, 3.53456623, 4.60617584, 5.36150313, 6.87933304,
25        4.31703231, 3.38316358, 4.93220723, 4.93559652, 2.60155331,
26        5.71126354, 6.11753055, 3.77141515, 7.11722011, 2.19488185,
27        5.95200192, 3.92230276, 7.08158906, 6.950745  , 3.66087655,
28        1.78909178, 3.5403411 , 4.53339519, 4.86733944, 4.05600544,
29        5.63464348, 5.46163588, 3.18617583, 4.41766556, 5.76028922,
30        3.71670025, 2.18256157, 4.29198369, 3.41003028, 3.58109704,
31        3.50966309, 6.66021636, 6.27178643, 3.73501763, 4.39320834,
32        3.5122169 , 6.23973963, 3.68148559, 7.241313  , 6.34536973,
33        5.93974152, 2.45932138, 2.53908889, 6.70860373, 6.83132233,
34        5.08265793, 6.03060737, 6.57459503, 3.57429693, 5.52990843,
35        2.40938175, 4.26478955, 3.53034483, 6.14314982, 5.15769738,
36        4.71084746, 6.84751527, 5.46464009, 4.17653234, 3.74809321,
37        2.27452311, 1.57636572, 1.9241337 , 5.90424604, 5.18903108,
38        1.87986818, 2.54434765, 3.22139361, 7.26037414, 6.48161696,
39        5.68848772, 6.63361882, 5.97274135, 3.89773792, 6.46124332,
40        6.6280363 , 3.1189592 , 4.69596361, 1.57369372, 3.67037693,
41        7.19440696, 1.78047866, 2.14235955, 3.65648599, 2.09035414,
42        3.36309717, 2.42314385, 7.11158364, 3.03994186, 2.37323204,
43        1.98456369, 2.62848285, 7.13675997, 3.10491835, 1.55865675,
44        7.47844662, 2.81390005, 5.74453964, 6.54098836, 6.56279413,
45        5.47012499, 2.08513053, 4.5895725 , 5.07450177, 7.46350981,
46        5.85390576, 3.76453996, 7.06279211, 6.37737643, 1.92086276,
47        7.36421338, 6.53579886, 7.30090272, 3.03723172, 6.70326683,
48        1.92799668, 4.22355358, 2.92270632, 4.4271086 , 2.07056241,
49        5.22407023, 7.16187287, 2.21069648, 7.2070597 , 2.18408499,
50        4.41499847, 5.01480967, 2.60203712, 2.91704867, 6.24434163,
51        6.85965449, 2.37123042, 5.96405759, 7.15367409, 2.28989742,
52        6.37622803, 3.5405043 , 3.13982572, 6.4607423 , 2.64134846,
53        2.00221355, 6.39142813, 2.76371977, 6.83125755, 3.82725452,
54        3.7706709 , 3.15985489, 5.09941716, 5.61039082, 1.85637175,
55        5.36372958, 2.33613356, 4.97585137, 2.62954677, 2.6464581 ,
56        3.85989239, 4.12153062, 6.38594064, 3.84270968, 4.99054854,
57        3.40059694, 3.82011537, 1.90949915, 2.85846412, 7.45150065,
```

```
58        3.35425211, 6.7078252 , 6.32590607, 1.93118068, 5.12854523,
59        4.27820987, 1.57947029, 2.90768218, 5.64471409, 3.57117544,
60        2.34510757, 5.84519703, 5.29848041, 3.43469993, 1.86599523,
61        5.09561549, 1.53080773, 1.61831146, 6.68767722, 7.34724552,
62        5.9838457 , 5.19608237, 4.19300743, 2.34768755, 4.70948938,
63        2.30719771, 2.73076882, 3.88251351, 3.57439365, 4.15957638,
64        1.54463433, 3.38389667, 3.61474518, 6.50388232, 1.84841256,
65        4.4203119 , 6.47715324, 6.56141998, 7.18090719, 2.80909078,
66        5.68620541, 4.80034375, 2.41291216, 2.92570352, 3.17417576,
67        2.68552963, 2.12442924, 2.69402213, 4.23088879, 5.35051566,
68        5.09157951, 6.25030221, 5.32463256, 2.04849771, 2.38138525,
69        6.91994297, 2.35950779, 4.54570746, 7.31050264, 1.69746369,
70        4.48358582, 2.07995534, 4.83439739, 4.76134448, 4.27003655,
71        5.37979542, 4.25555723, 4.21646848, 5.69372506, 2.9965207 ,
72        2.84945821, 3.74497364, 3.44151329, 4.28007538, 1.82445572,
73        2.889242  , 2.1699775 , 6.9768987 , 2.68373123, 1.65115609,
74        4.41632058, 3.88109869, 3.78456055, 6.2672605 , 1.89032676,
75        1.76205818, 6.08916251, 6.37425386, 6.39248312, 3.20431634,
76        2.90814691, 2.34995207, 7.00439884, 3.48359912, 2.5174638 ,
77        5.69845134, 4.38192328, 4.77205909, 5.39501645, 5.45644131,
78        1.78835992, 3.11259278, 5.08096845, 4.85131392, 6.60747211,
79        3.12685618, 6.56276283, 2.34307547, 1.60003302, 5.24817554,
80        3.03546132, 4.94737215, 1.57495362, 5.74172488, 5.91047527,
81        6.19500399, 5.49258972, 4.51724457, 4.311786  , 4.63374137,
82        7.4238325 , 2.50776855, 5.71258666, 6.82237691, 3.36759182,
83        6.32019131, 2.29864337, 4.69989588, 7.24675483, 5.66695115,
84        4.08746198, 3.31455607, 6.76833623, 5.40574906, 3.63228339,
85        4.92494969, 3.69920552, 6.89692929, 7.02551769, 6.47562522,
86        1.63938057, 6.63595412, 4.60440921, 7.0320571 , 3.10234604,
87        3.0441142 , 6.18289923, 5.41505293, 3.86069598, 5.69533881,
88        6.36491894, 2.09909347, 3.49896752, 5.99460228, 6.84093095,
89        5.75494099, 6.92672097, 1.62381374, 5.90635848, 7.48152138,
90        2.60785316, 1.6606505 , 6.23558536, 2.96599172, 6.27926507,
91        2.22040386, 2.07095025, 2.86841863, 3.9281409 , 6.83574773,
92        3.43716472, 5.13956522, 6.0045213 , 3.2831669 , 5.91912926,
93        3.91019855, 2.64209859, 5.72087741, 3.39991642, 1.61160631,
94        4.78368549, 4.59148911, 5.67183974, 6.0527314 , 4.31579418,
95        6.49585476, 6.54613258, 4.68821963, 3.12652403, 6.78156412,
96        7.11950538, 7.29772228, 5.87349354, 1.58731535, 6.00180686,
```

```
 97        5.7232261 , 5.23789305, 6.44994534, 4.17233667, 6.52277168,
 98        7.44811656, 7.22519186, 3.27533481, 2.03181776, 2.84014075,
 99        6.1015647 , 6.0743278 , 4.24387255, 5.31857559, 6.05238132,
100        3.15404259, 2.4823816 , 3.43510175, 5.24920879, 3.47179858,
101        6.08760962, 3.44084688, 4.53054541])
102 data['happiness'] = np.array([2.31448898, 3.43348976, 4.5993734 , 2.7911138 , 5.59639827,
103        2.45855587, 3.19299181, 1.90713683, 2.94244987, 3.7379416 ,
104        3.17540615, 2.00904646, 5.9518141 , 5.96054731, 1.44579886,
105        2.89858314, 1.23116752, 2.31298805, 2.66611603, 2.58472902,
106        5.7474441 , 2.54652459, 1.20078553, 3.07829338, 4.31776092,
107        5.38147874, 3.56522432, 0.953413  , 2.16916126, 2.06079431,
108        6.29910126, 1.59035586, 2.25092941, 5.9142477 , 2.1918337 ,
109        5.76781437, 0.97288292, 5.01057743, 2.23966499, 1.92757883,
110        3.35807157, 2.40008729, 3.40798004, 2.57617631, 2.47307938,
111        3.75265949, 4.08763118, 3.54320365, 5.34835294, 3.05637665,
112        1.55842256, 3.22513275, 5.03423101, 3.85742426, 3.61905549,
113        4.00453774, 4.80209181, 4.65889039, 5.23170297, 2.48060647,
114        5.49814719, 4.55063696, 3.57001357, 4.71966533, 4.50310817,
115        5.04608177, 0.66971595, 1.60607236, 4.8266027 , 2.20824052,
116        4.0564931 , 3.56790523, 1.30848726, 4.1596093 , 4.63391506,
117        5.06874788, 3.04079729, 3.80429886, 5.03400379, 3.70343793,
118        2.18893812, 5.53804753, 3.58387516, 3.09688561, 2.20098217,
119        5.15159829, 5.95098633, 2.9681871 , 2.39956133, 0.85949905,
120        1.77593255, 4.66126121, 0.72722119, 0.77128657, 3.34878821,
121        3.96621541, 2.89044742, 3.77326065, 6.3596    , 0.2687221 ,
122        4.37483232, 2.15584332, 1.89355686, 4.90299158, 4.83350643,
123        3.17229946, 5.02995166, 4.261013  , 1.67239058, 1.49702413,
124        1.26548892, 2.66746543, 1.99932553, 5.23186331, 5.21140132,
125        3.66165646, 1.41503466, 4.93304409, 4.13077827, 2.28226693,
126        3.90117033, 4.69199885, 3.57780067, 5.56254547, 2.39322812,
127        3.56472374, 2.25372152, 4.12164775, 4.16910076, 3.82389869,
128        0.45837757, 2.57693996, 2.94753147, 3.7399958 , 3.57144651,
129        4.80815042, 4.01761121, 1.83980201, 3.46857379, 4.75878549,
130        2.39167752, 0.99291736, 3.16938024, 2.08904235, 1.84367577,
131        1.61660744, 5.9493475 , 4.94022778, 2.84123868, 2.94439134,
132        3.02691818, 5.09780251, 3.03681994, 4.68281699, 4.00081918,
133        4.57080113, 2.04273931, 1.74865112, 6.06401305, 5.13240147,
134        3.11334209, 4.82181026, 4.17952102, 1.63126111, 3.82217964,
135        1.85754189, 3.75108928, 3.15861859, 4.92713262, 4.60011476,
```

```
136       2.4083799 , 4.48670366, 2.72771571, 3.02072445, 3.74918073,
137       2.31155424, 0.98760317, 1.46114442, 4.57685653, 4.76759564,
138       0.43817157, 2.35120247, 3.45310274, 5.38139847, 4.82009125,
139       4.65874268, 5.38007021, 3.31598948, 2.79974749, 4.20675491,
140       4.40266319, 2.76911813, 2.78424455, 0.68809058, 3.47649914,
141       5.83619666, 2.00392609, 0.97133245, 2.85761436, 1.85211771,
142       3.51517918, 2.10055249, 6.08647828, 4.08382116, 1.52478229,
143       2.57951652, 1.61922613, 5.50662154, 1.09599927, 0.96852733,
144       4.87772554, 1.75969867, 5.03221103, 5.71380229, 4.79584287,
145       4.66028996, 2.74039541, 4.25056806, 3.94101932, 4.50344538,
146       4.63322873, 4.03086494, 6.86338795, 5.0794759 , 1.62526839,
147       6.61827984, 4.65699857, 6.00498747, 2.41063557, 4.26121997,
148       1.06562381, 2.29570014, 1.9195111 , 3.58138104, 0.62894206,
149       3.34410837, 6.1604341 , 2.86127444, 4.20947057, 1.22629813,
150       4.79332516, 3.31379652, 2.15269843, 2.92368387, 3.45206723,
151       4.25831153, 3.96103277, 3.42772316, 4.72098734, 1.57194981,
152       6.18694561, 3.55273696, 1.24578594, 3.42691165, 1.87714704,
153       1.81788911, 5.2399095 , 1.38404411, 3.83797777, 2.32169769,
154       3.31157823, 3.05738784, 4.73704595, 4.59097046, 0.65424215,
155       3.98112774, 2.33627603, 4.54063615, 2.19274486, 1.78817634,
156       2.71943218, 4.21269734, 4.47727139, 2.46885999, 3.45177254,
157       0.68692085, 2.24674571, 1.62769509, 2.45320906, 4.10994465,
158       2.5725982 , 4.81042406, 4.8274163 , 2.46576113, 3.75737899,
159       3.69134819, 1.98180991, 1.41022957, 3.75430142, 4.2452156 ,
160       1.43787672, 4.15988714, 3.51489178, 2.71449804, 1.35737314,
161       5.63846039, 2.42146467, 1.51120511, 5.52289167, 4.39662232,
162       3.98845283, 3.31760678, 2.77968263, 0.84650553, 2.45792171,
163       3.50960469, 2.64690454, 2.97791672, 1.42445055, 1.79034232,
164       1.6446366 , 0.73163629, 2.90178859, 4.31713316, 2.09736906,
165       4.39159298, 5.3139569 , 6.28137019, 5.60796663, 2.94687043,
166       3.87278885, 2.84286835, 1.46061633, 3.75282477, 3.12660324,
167       2.82119264, 2.73496006, 2.19759212, 4.15554086, 4.07820884,
168       4.45696361, 4.93925896, 3.73056996, 2.57505486, 2.17632215,
169       6.23695531, 0.8987328 , 3.62942439, 5.92334607, 0.54619018,
170       3.03149707, 2.84781684, 4.51696059, 4.59169701, 3.15776509,
171       4.28799968, 3.61882979, 3.20079786, 3.99970301, 1.6622532 ,
172       2.39155   , 3.06760945, 2.01751359, 3.64088397, 0.26604366,
173       1.29542729, 1.85334761, 5.3672361 , 2.46629234, 1.72359356,
174       4.30728842, 3.15636744, 1.69045809, 5.08552301, 2.48162387,
```

```
175       1.35331734, 5.27529415, 3.67029153, 3.69639041, 2.67988732,
176       2.36356467, 1.4385961 , 5.36244826, 2.32530923, 1.52956094,
177       4.03572702, 3.55033718, 3.55460032, 4.19972853, 3.13547515,
178       1.14529581, 3.53985413, 3.82220396, 3.83557765, 5.68375056,
179       2.51740088, 4.32466038, 2.9586968 , 1.68099713, 4.2535013 ,
180       2.26039569, 3.61547105, 1.50221163, 3.77648463, 3.20104243,
181       4.05831162, 4.10522452, 3.72209764, 3.82227031, 3.70666098,
182       4.90968847, 2.15573363, 5.74373608, 4.33757145, 2.58241664,
183       4.28430556, 2.96216821, 3.99041589, 5.8901487 , 3.68853952,
184       3.21784736, 2.22047033, 5.91028488, 4.85303252, 3.80297363,
185       3.09216465, 3.23331375, 5.26547896, 4.81153338, 5.36804117,
186       1.7987183 , 4.76001433, 3.41580648, 5.04017458, 1.96563593,
187       1.87195785, 5.32350739, 4.50115834, 2.43509486, 4.55321655,
188       4.36170927, 1.4787872 , 2.76852313, 3.96063637, 5.9157814 ,
189       3.98019535, 4.70820299, 1.41666106, 4.3284171 , 6.19629555,
190       1.94596105, 0.69845996, 5.13545532, 1.56035538, 4.23499122,
191       0.68890923, 0.68584887, 1.79175357, 3.22747217, 5.15933124,
192       3.80297067, 4.21605759, 4.93240998, 2.23344058, 4.37825945,
193       3.48392568, 2.03029704, 3.59639832, 3.2611821 , 3.48113777,
194       3.39343915, 4.22966335, 4.3213643 , 4.40045515, 3.35759738,
195       5.335718  , 5.58838934, 2.99420428, 2.41338175, 4.98536659,
196       4.98198112, 6.3842745 , 3.26747544, 1.31275898, 4.03954799,
197       4.42599508, 3.3427494 , 3.49231019, 3.75122546, 5.38655933,
198       5.96342209, 4.98525493, 2.79816073, 2.96732327, 1.22877318,
199       3.96172964, 4.64914365, 4.7541684 , 2.96195666, 4.63344105,
200       1.29413658, 0.54836519, 2.11513648, 4.56870456, 2.53500211,
201       4.39745128, 2.07066377, 3.71019298])
```

Pruebas

```
1 # Prueba 1
2 train_X, train_Y, test_X, test_Y = split_dataset(70, datos) # División de dataset de prueba y entrenamiento
3 test_y, resultado = regresion(test_X, test_Y, fit(train_X, train_Y))   # Predicción de y del dataset de prueba
4 resultado, MSE, MAE, RMSE, MAPE, SSE, R2 = evaluacion(resultado)
5 resultado
```

```
Mean Squared Error:  0.4784252664607663
Mean Absolute Error:  0.5471487170441045
Root Mean Squared Error 0.6916829233548898
Mean Percent Absolute Error 0.27988871657177916
Sum of Squared Errors 71.76378996911494
Coeficiente de determinación 0.7151342963811769
```

	income	happiness	Predicción happiness	Error	Error Absoluto	Squared Error
0	4.639914	3.737942	3.531611	0.206330	0.206330	0.042572
1	7.466653	5.960547	5.521604	0.438943	0.438943	0.192671
2	3.528319	2.546525	2.749060	-0.202536	0.202536	0.041021
3	2.428752	1.200786	1.974977	-0.774191	0.774191	0.599372
4	3.542748	3.078293	2.759218	0.319075	0.319075	0.101809
...	...	...	...	...	...	...
145	7.448117	5.963422	5.508555	0.454867	0.454867	0.206904
146	7.225192	4.985255	5.351618	-0.366363	0.366363	0.134222
147	4.243873	4.754168	3.252802	1.501366	1.501366	2.254101
148	5.249209	4.568705	3.960548	0.608157	0.608157	0.369855
149	3.440847	2.070664	2.687481	-0.616817	0.616817	0.380463

150 rows × 6 columns

En la segunda prueba se modificó el tamaño de los dataset de entrenamiento y prueba y el resultado fue que los valores de los errores disminuyó

```
# Prueba 2
train_X, train_Y, test_X, test_Y = split_dataset(90, datos) # División de dataset de prueba y entrenamiento
test_y, resultado = regresion(test_X, test_Y, fit(train_X, train_Y))   # Predicción de y del dataset de prueba
resultado, MSE, MAE, RMSE, MAPE, SSE, R2 = evaluacion(resultado)
resultado.head()
```

```
Mean Squared Error:  0.5413035373328922
Mean Absolute Error:  0.6061574065890772
Root Mean Squared Error 0.7357333330309918
Mean Percent Absolute Error 0.22451272185720014
Sum of Squared Errors 27.06517686664461
Coeficiente de determinación 0.7544831690314232
```

	income	happiness	Predicción happiness	Error	Error Absoluto	Squared Error
0	4.755680	2.666116	3.591761	-0.925645	0.925645	0.856819
1	1.994275	2.584729	1.620063	0.964666	0.964666	0.930581
2	3.542748	3.078293	2.725703	0.352590	0.352590	0.124320
3	3.065059	3.407980	2.384624	1.023357	1.023357	1.047259
4	5.358716	3.752659	4.022341	-0.269681	0.269681	0.072728

1

1

Colab paid products - Cancel contracts here

✓ 0s completed at 7:14 PM