



+ Code + Text



▼ 21022008_NguyenHuuTho

▼ Hiển thị dữ liệu trên đồ thị

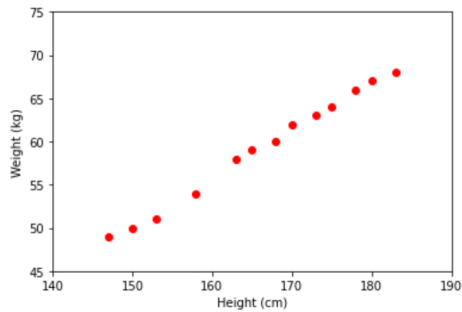
Trước tiên, chúng ta cần có hai thư viện numpy cho đại số tuyến tính và matplotlib cho việc vẽ hình.

```
[31] # To support both python 2 and python 3
      from __future__ import division, print_function, unicode_literals
      import numpy as np
      import matplotlib.pyplot as plt
```

▼ Tiếp theo, chúng ta khai báo và biểu diễn dữ liệu trên một đồ thị.

▼ Tiếp theo, chúng ta khai báo và biểu diễn dữ liệu trên một đồ thị.

```
[32] # height (cm)
      X = np.array([[147, 150, 153, 158, 163, 165, 168, 170, 173, 175, 178, 180, 183]]).T
      # weight (kg)
      y = np.array([[ 49, 50, 51, 54, 58, 59, 60, 62, 63, 64, 66, 67, 68]]).T
      # Visualize data
      plt.plot(X, y, 'ro')
      plt.axis ([140, 190, 45, 75])
      plt.xlabel('Height (cm)')
      plt.ylabel('Weight (kg)')
      plt.show()
```



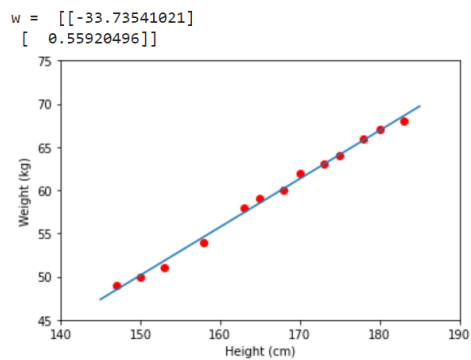
▼ Từ đồ thị này ta thấy rằng dữ liệu được sắp xếp gần như theo 1 đường thẳng, vậy mô hình Linear Regression nhiều khả năng sẽ cho kết quả tốt:

$$(\text{cân nặng}) = w_1 * (\text{chiều cao}) + w_0$$

Nghiệm theo công thức

Tiếp theo, chúng ta sẽ tính toán các hệ số w_1 và w_0 dựa vào công thức (5)(5). Chú ý: giả nghịch đảo của một ma trận A trong Python sẽ được tính bằng `numpy.linalg.pinv(A)`, `pinv` là từ viết tắt của pseudo inverse.

```
# Building Xbar
one = np.ones((X.shape[0], 1))
Xbar = np.concatenate ((one, X), axis = 1)
# Calculating weights of the fitting line
A = np.dot (Xbar.T, Xbar)
b = np.dot (Xbar.T, y)
w = np.dot (np.linalg.pinv(A), b)
print('w = ', w)
# Preparing the fitting line
w_0 = w[0][0]
w_1 = w[1][0]
x0 = np.linspace (145, 185, 2)
y0 = w_0 + w_1 * x0
# Drawing the fitting line
plt.plot(X.T, y.T, 'ro')
# data
plt.plot(x0, y0)
# the fitting line
plt.axis ([140, 190, 45, 75])
plt.xlabel('Height (cm)')
plt.ylabel('Weight (kg)')
plt.show()
```



Từ đồ thị bên trên ta thấy rằng các điểm dữ liệu màu đỏ nằm khá gần đường thẳng dự đoán màu xanh. Vậy mô hình Linear Regression hoạt động tốt với tập dữ liệu training. Bây giờ, chúng ta sử dụng mô hình này để dự đoán cân nặng của hai người có chiều cao 155 và 160 cm mà chúng ta đã không dùng khi tính toán nghiệm.

Kết quả đoạn code vừa chạy

```
[34] y1 = w_1*185 + w_0
      y2 = w_1*160 + w_0
      print( u'Predict weight of person with height 155 cm: %.2f (kg), real number: 52 (kg)' % y1)
      print( u'Predict weight of person with height 160 cm: %.2f (kg), real number: 56 (kg)' % y2)
```

Predict weight of person with height 155 cm: 69.72 (kg), real number: 52 (kg)
Predict weight of person with height 160 cm: 55.74 (kg), real number: 56 (kg)

Nghiệm theo thư viện scikit-learn

▼ Tiếp theo, chúng ta sẽ sử dụng thư viện scikit-learn của Python để tìm nghiệm.

```
[35] from sklearn import datasets, linear_model
      # fit the model by Linear Regression
      regr = linear_model.LinearRegression (fit_intercept=False) # fit_intercept = False for
      regr.fit(Xbar, y)
      # Compare two results
      print('Solution found by scikit-learn: ', regr.coef_)
      print('Solution found by (5): ', w.T)
```

Solution found by scikit-learn: [[-33.73541021 0.55920496]]
Solution found by (5): [[-33.73541021 0.55920496]]