TRƯỜNG ĐẠI HỌC
VĂN LANG
Đạo đức - Ý chí - Sáng tạo
KHOA CÔNG NGHỆ THÔNG TIN

THỰC HÀNH NHẬP MÔN HỌC MÁY

THỰC HÀNH – HỒI QUY TUYẾN TÍNH

BỘ MÔN KHOA HỌC DỮ LIỆU

THỰC HÀNH – HỒI QUY TUYẾN TÍNH

1. Phương trình hồi qui tuyến tính

-Phân tích hồi qui tuyến tích đơn giản (Simple Linear Regression Analysis) là tìm sự liên hệ giữa 2 biến số liên tục: biến độc lập (biến dự đoán) trên trục hoành x với biến phụ thuộc (biến kết cục) trên trục tung y.

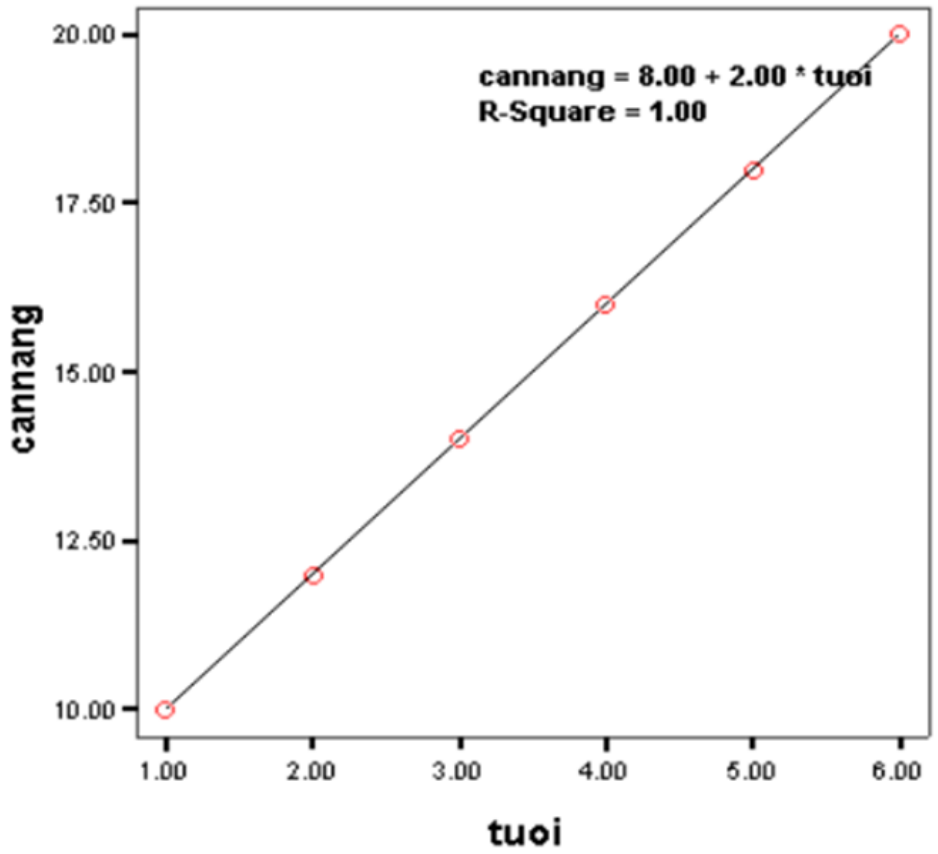
-Vẽ một đường thẳng hồi qui và từ phương trình đường thẳng này ta có thể dự đoán được biến y.

Ví dụ 1.

Ta có 1 mẫu gồm 6 trẻ từ 1-6 tuổi, có cân nặng như bảng sau:

Tuổi	Cân nặng (kg)
1	10
2	12
3	14
4	16
5	18
6	20

THỰC HÀNH – HỒI QUY TUYẾN TÍNH



Nối các cặp (x,y) này ta thấy có dạng 1 phương trình bậc nhất: **y=2x+8**

(trong đó 2 là độ dốc và 8 là điểm cắt trên trục tung y khi x=0). Trong thống kê phương trình đường thẳng (bậc nhất) này được viết dưới dạng:

$y = \beta x + \alpha$ [1]

THỰC HÀNH – HỒI QUY TUYẾN TÍNH

Đây là phương trình hồi qui tuyến tính, trong đó β gọi là độ dốc (slope) và α là chặn (intercept), điểm cắt trên trục tung khi x=0.

Thực ra phương trình hồi qui tuyến tính này chỉ có trên lý thuyết, nghĩa là các trị số của x_i (i=1,2,3,4,5,6) và y_i tương ứng, liên hệ với nhau 100% (hoặc hệ số tương quan R=1)

Trong thực tế hiếm khi có sự liên hệ 100% này mà thường có sự sai lệch giữa trị số quan sát y_i và trị số y_i' ước đoán nằm trên đường hồi qui.

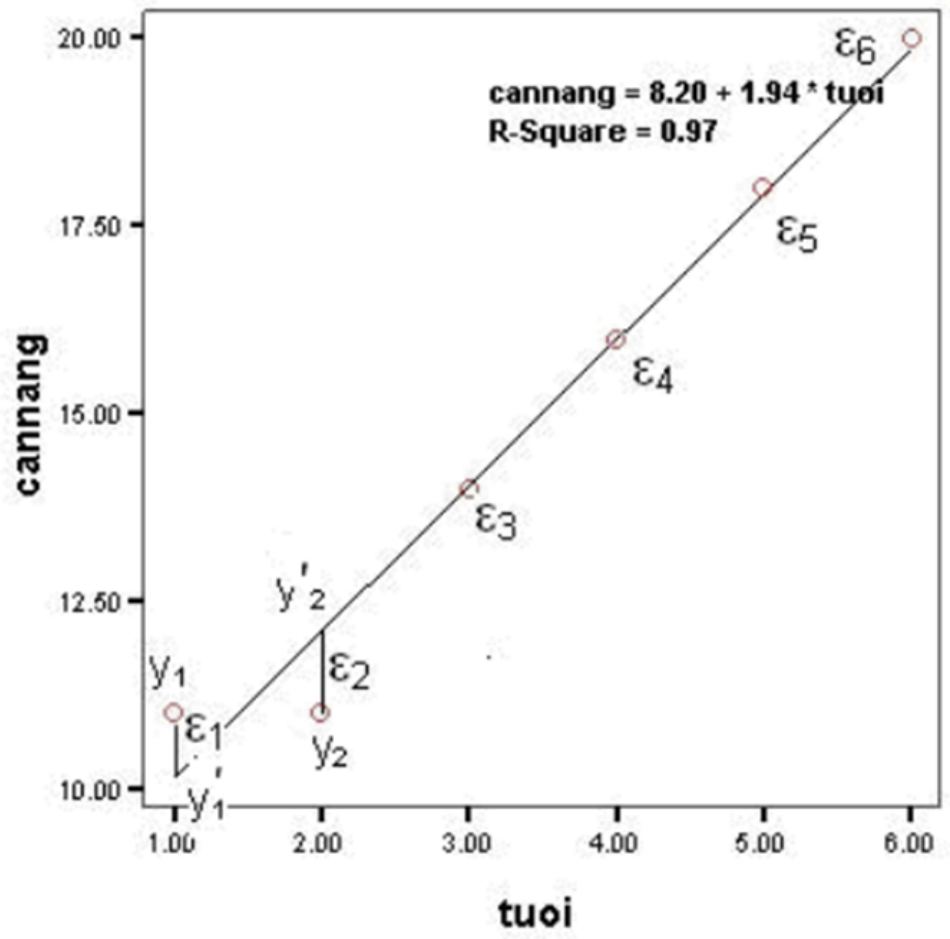
THỰC HÀNH – HỒI QUY TUYẾN TÍNH

Ví dụ 2. Mô hình hồi quy tuyến tính

Ta có 1 mẫu gồm 6 trẻ em khác có cân nặng theo bảng sau:

Tuổi	Cân nặng (kg)
1	11
2	11
3	14
4	16
5	18
6	20

THỰC HÀNH – HỒI QUY TUYẾN TÍNH



Khi vẽ đường thẳng hồi qui, ta thấy các trị số quan sát y_3, y_4, y_5, y_6 nằm trên đường thẳng, còn y_1 và y_2 không nằm trên đường thẳng này và sự liên hệ giữa x_i và y_i

THỰC HÀNH – HỒI QUY TUYẾN TÍNH

không còn là 100% mà chỉ còn 97% vì có sự sai lệch tại y_1 và y_2. Sự sai lệch này trong thống kê gọi là phần dư (residual) hoặc errors.

Gọi $y_1, y_2, y_3, y_4, y_5, y_6$ là trị số quan sát và $y'_1, y'_2, y'_3, y'_4, y'_5, y'_6$ là trị số ước đoán nằm trên đường hồi qui, $\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4, \varepsilon_5, \varepsilon_6$ là phần dư.

Như vậy $\varepsilon_1 = y_1 - y'_1$

$\varepsilon_2 = y_2 - y'_2$

$\varepsilon_3 = y_3 - y'_3$

$\varepsilon_4 = y_4 - y'_4$

$\varepsilon_5 = y_5 - y'_5$

$\varepsilon_6 = y_6 - y'_6$

Khi đó phương trình hồi qui tuyến tính được viết dưới dạng tổng quát như sau:

$y' = \beta x_i + \alpha_i + \varepsilon_i$ [2]

Như vậy nếu phần dư ε_i càng nhỏ sự liên hệ giữa x,y càng lớn và ngược lại. Phần liên hệ còn đượi gọi là phần hồi qui. Mô hình hồi qui tuyến tích được mô tả như sau:

Dữ liệu = Hồi quy (Regression) + Phần dư (Residual)

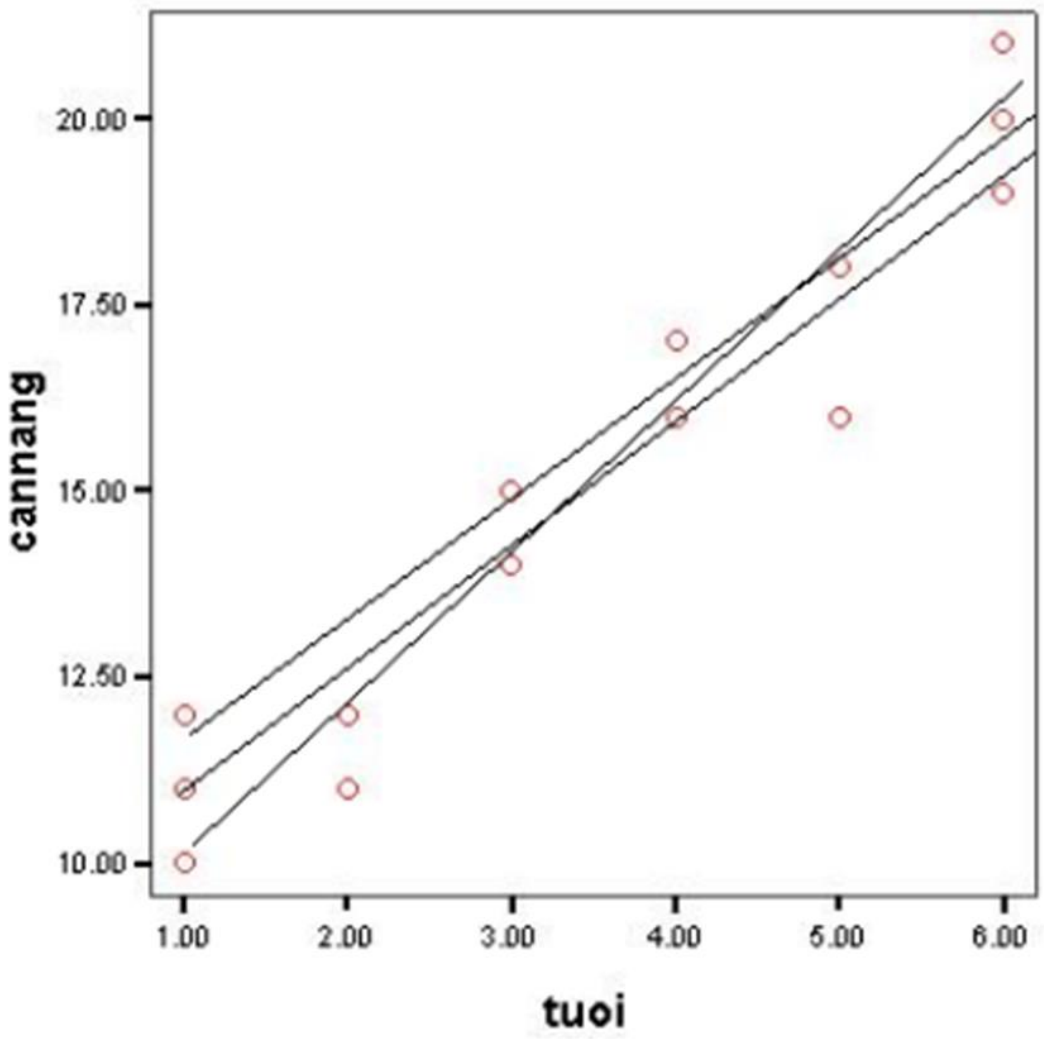
THỰC HÀNH – HỒI QUY TUYẾN TÍNH

Ước tính hệ số tương quan β và chặn α

Muốn vẽ được phương trình hồi qui tuyến tính cần phải ước tính được độ dốc β và chặn α trên trục tung.

Ví dụ 3.

Nếu chúng ta chọn một mẫu thực tế gồm 30 em từ 1-6 tuổi và kết quả cân nặng tương ứng của 30 em được vẽ trong biểu đồ sau:



THỰC HÀNH – HỒI QUY TUYẾN TÍNH

Lúc này ta không thể nối 30 điểm trên biểu đồ mà phải vẽ 1 đường thẳng đi càng gần với tất cả các điểm càng tốt. Như vậy 3 đường thẳng ở biểu đồ ta chọn đường thẳng nào?. Nguyên tắc chọn đường thẳng nào đi gần cả 30 điểm, có nghĩa làm sao để tổng các phần dư $\Sigma\varepsilon_i$ nhỏ nhất:

$$\Sigma\ \varepsilon_i = \Sigma\ (y_i - \beta x - \alpha)$$

và tổng bình phương của phần dư:

$$\Sigma\ (\varepsilon_i)^2 = \Sigma\ (y_i - \beta x - \alpha)^2$$

Đây là phương trình bậc 2 theo x. Trong toán học, muốn tìm trị cực tiểu của 1 phương trình bậc 2, người ta lấy đạo hàm và cho đạo hàm triệt tiêu (bằng 0) sẽ tìm được trị cực tiểu của x. Giải phương trình này, ta sẽ tính được 2 thông số β và α và từ 2 thông số này ta sẽ vẽ được đường thẳng hồi qui. Phương pháp này trong toán học gọi là **phương pháp bình phương nhỏ nhất** (least square method).

THỰC HÀNH – HỒI QUY TUYẾN TÍNH

Giải phương trình trên ta có:

$$\beta = r\frac{Sy}{Sx}$$

(r là hệ số tương quan; Sy là độ lệch chuẩn của y và Sx là độ lệch chuẩn của x)

$$r = \frac{1}{n-1}\Sigma\left(\frac{xi-\overline{x}}{Sx}\right)\left(\frac{yi-\overline{y}}{Sy}\right)$$

$$\alpha = y - \beta x$$

và phương trình hồi qui tuyến tính của y theo x (bình phương nhỏ nhất) là:

$$y' = \beta x_i + \alpha$$

THỰC HÀNH – HỒI QUY TUYẾN TÍNH

2. Phân tích hồi qui tuyến tính

Nhập số liệu tuổi và cân nặng cân của 30 trẻ 1-6 tuổi:

cân được của 30 trẻ 1-6 tuổi gồm cột 1-tuổi; cột 2-cân nặng

Bảng 1 – Tóm tắt mô hình

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.918[a]	.843	.837	1.49794

a. Predictors: (Constant), tuoi

Hệ số tương quan R=0,918 và R^2=0,843

	tuoi	cannang
1	1	9
2	2	8
3	3	12
4	4	16
5	5	18
6	6	18
7	1	12
8	2	13
9	3	14
10	4	16
11	5	18
12	6	20
13	1	10
14	2	11
15	3	14
16	4	17
17	5	18
18	6	19
19	1	10
20	2	11
21	3	15
22	4	16
23	5	13
24	6	21
25	1	11
26	2	10
27	3	14
28	4	16
29	5	18
30	6	21

THỰC HÀNH – HỒI QUY TUYẾN TÍNH

Bảng 2 – Phân tích với biến phụ thuộc là cân nặng

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	336.140	1	336.140	149.808	.000[a]
	Residual	62.827	28	2.244		
	Total	398.967	29			

a. Predictors: (Constant), tuoi

b. Dependent Variable: cannang

Tổng bình phương phần hồi qui (Regression)=336,14

Tổng bình phương phần dư (Residual)=62,8

Trung bình bình phương hồi qui: 336,14/ 1 (bậc tự do)=336,14

Trung bình bình phương phần dư: 62,8/ 28(bậc tự do=n-2)=2,24

$$F= \frac{336,14}{2,24} = 149,8 \text{ và } p<0,000$$

THỰC HÀNH – HỒI QUY TUYẾN TÍNH

Bảng 3 – Hệ số tương quan β và chặn α

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	7.773	.624		12.464	.000
	tuoi	1.960	.160	.918	12.240	.000

a. Dependent Variable: cannang

Kết quả: Ta thấy hệ số tương quan β (độ dốc) = 1,96 và điểm cắt tại trung tung là α=7.773

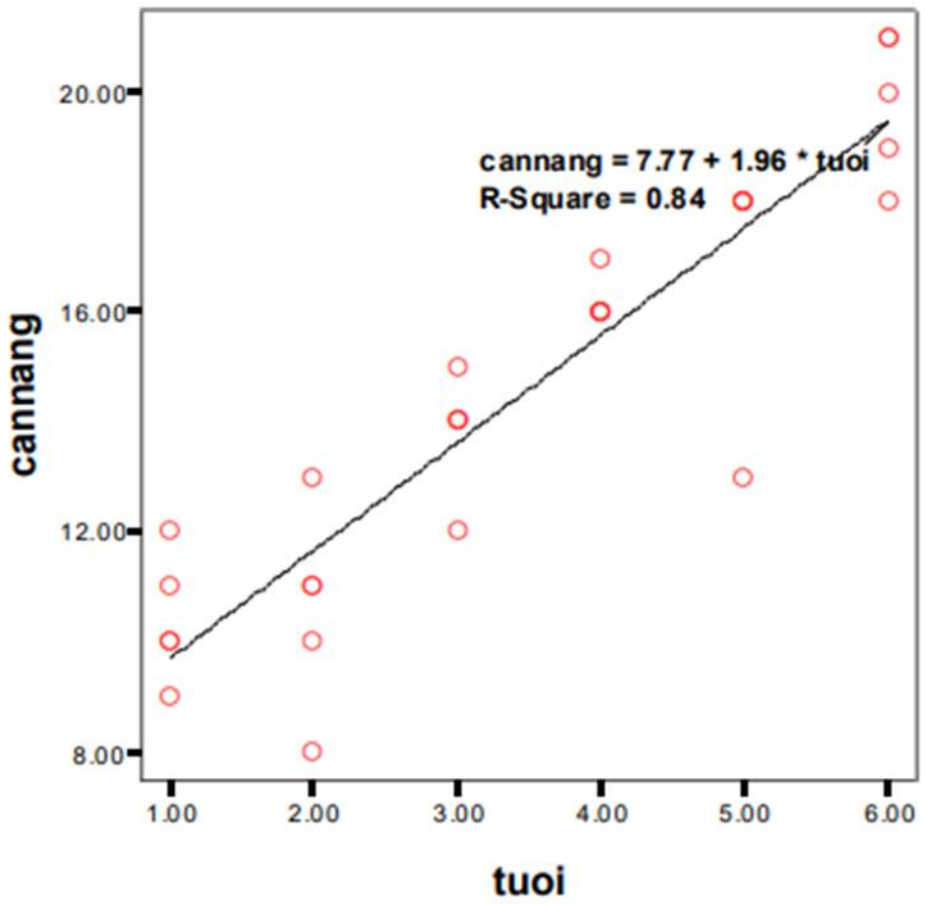
Phương trình đường thẳng hồi qui được viết:

Cân nặng= 7,77 + 1,96 *x* tuổi

Như vậy, khi em bé tăng lên 1 tuổi thì cân nặng tăng lên 1,96 kg.

THỰC HÀNH – HỒI QUY TUYẾN TÍNH

Vẽ đường thẳng hồi qui sau:



Linear Regression

THỰC HÀNH – HỒI QUY TUYẾN TÍNH

Từ phương trình trên, ta có thể ước đoán được cân nặng theo tuổi của trẻ.

Tuy nhiên nằm trong một giới hạn nào đó chẳng hạn như từ 1-12 tuổi, vì sau tuổi này cân nặng trẻ sẽ tăng vọt trong thời kỳ dậy thì và không còn quan hệ tuyến tính với tuổi nữa.

Chẳng hạn, ta muốn ước đoán cân nặng của trẻ từ quần thể thì:

7 tuổi → Cân nặng= 7,77 + 1,96 x 7 = 21,49 kg

8 tuổi → Cân nặng= 7,77 + 1,96 x 8 = 23,45 kg

THỰC HÀNH – HỒI QUY TUYẾN TÍNH

3. Các giả định trong phân tích hồi qui tuyến tính

Phân tích hồi qui tuyến tính không chỉ là việc mô tả các dữ liệu quan sát được trong mẫu (sample) nghiên cứu mà cần phải suy rộng cho mối liên hệ trong dân số (population).

Vì vậy, trước khi trình bày và diễn dịch mô hình hồi qui tuyến tính cần phải dò tìm vi phạm các giả định.

Nếu các giả định bị vi phạm thì các kết quả ước lượng không đáng tin cậy được.

Các giả định cần thiết trong hồi qui tuyến tính:

(1) x_i là biến số cố định, không có sai sót ngẫu nhiên trong đo lường.

(2) Phần dư (trị số quan sát trừ cho trị số ước đoán) phân phối theo luật phân phối chuẩn

(3) Phần dư có trị trung bình bằng 0 và phương sai không thay đổi cho mọi trị x_i

(4) Không có tương quan giữa các phần dư

THỰC HÀNH – HỒI QUY TUYẾN TÍNH

#THỰC HÀNH 1

```
#Step 1: Nhập gói và lớp
>>> import numpy as np
>>> from sklearn.linear_model import LinearRegression
#Step 2: Cung cấp dữ liệu
>>> x = np.array([5, 15, 25, 35, 45, 55]).reshape((-1, 1))
>>> y = np.array([5, 20, 14, 32, 22, 38])
>>> print(x)
[[ 5]
 [15]
 [25]
 [35]
 [45]
 [55]]
>>> print(y)
[ 5 20 14 32 22 38]
```

THỰC HÀNH – HỒI QUY TUYẾN TÍNH

#THỰC HÀNH 1

```
#Step 3: Tạo một mô hình phù hợp
>>> model = LinearRegression()
>>> model.fit(x, y)
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
normalize=False)
      # copy_X là một Boolean (Đúng là mặc định) quyết định sao chép
(Đúng) hay ghi đè các biến đầu vào (Sai);
      # fit_intercept là một Boolean (Đúng là mặc định) quyết định
tính toán khoảng chặn b0 (Đúng) hay nó bằng 0 (Sai);
      # n_jobs là một số nguyên hoặc Không (mặc định) và đại diện cho
số lượng công việc được sử dụng trong tính toán song song. Không có
nghĩa là một công việc và -1 để sử dụng các bộ xử lý;
      # normalize là một Boolean (False là mặc định) chuẩn hóa các
biến đầu vào (True) hay (False);
>>> model = LinearRegression().fit(x, y)
       # Câu lệnh này thực hiện tương tự như hai câu trước, thực hiện
ngắn hơn
```

THỰC HÀNH – HỒI QUY TUYẾN TÍNH

#THỰC HÀNH 1

```
#Step 4: Nhận kết quả
>>> r_sq = model.score(x, y)
>>> print('coefficient of determination:', r_sq)
coefficient of determination: 0.715875613747954
      #lấy hệ số xác định (R²) với .score () được gọi từ mô hình;
>>> print('intercept:', model.intercept_)
intercept: 5.633333333333329
>>> print('slope:', model.coef_)
slope: [0.54]
      #thuộc tính của mô hình .intercept_, đại diện cho hệ số b₀ và
.coef_, đại diện cho hệ số b₁;
>>> new_model = LinearRegression().fit(x, y.reshape((-1, 1)))
>>> print('intercept:', new_model.intercept_)
intercept: [5.63333333]
>>> print('slope:', new_model.coef_)
slope: [[0.54]]
```

```
#Step 5: Dự đoán DL hiện có hoặc DL mới
>>> y_pred = model.predict(x)
>>> print('predicted response:', y_pred, sep='\n')
predicted response:
[ 8.33333333 13.73333333 19.13333333 24.53333333 29.93333333
35.33333333]
>>> y_pred = model.intercept_ + model.coef_ * x
>>> print('predicted response:', y_pred, sep='\n')
predicted response:
[[ 8.33333333]
 [13.73333333]
 [19.13333333]
 [24.53333333]
 [29.93333333]
 [35.33333333]]
>>> x_new = np.arange(5).reshape((-1, 1))
>>> print(x_new)
[[0]
 [1]
 [2]
 [3]
 [4]]
>>> y_new = model.predict(x_new)
>>> print(y_new)
[5.63333333 6.17333333 6.71333333 7.25333333 7.79333333]
```

#THỰC HÀNH 2

```
import numpy as np
import matplotlib.pyplot as plt
def estimate_coef(x, y):
    # số điểm
    n = np.size(x)
    # trung bình của vecto x và y
    m_x = np.mean(x)
    m_y = np.mean(y)
    # tính toán độ lệch và độ lệch về x;
    SS_xy = np.sum(y*x) - n*m_y*m_x
    SS_xx = np.sum(x*x) - n*m_x*m_x
    # hệ số hồi quy;
    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):
    # biểu đồ các điểm thực tế - phân tán;
    plt.scatter(x, y, color = "m",
                marker = "o", s = 30)
    # dự đoán
    y_pred = b[0] + b[1]*x
    # vẽ đường hồi quy
    plt.plot(x, y_pred, color = "g")
    # dán nhãn
    plt.xlabel('x')
    plt.ylabel('y')
    plt.show()
```

```
def main():
    # data
    x = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
    y = np.array([1, 3, 2, 5, 7, 8, 8, 9, 10, 12])
    # hệ số ước lượng
    b = estimate_coef(x, y)
    print("Estimated coefficients:\nb_0 = {}  \
          \nb_1 = {}".format(b[0], b[1]))
    # vẽ đường hồi quy
    plot_regression_line(x, y, b)

if __name__ == "__main__":
    main()
```

Estimated coefficients:

b_0 = 1.2363636363636363

b_1 = 1.1696969696969697

TRƯỜNG ĐẠI HỌC
VĂN LANG
Đạo đức - Ý chí - Sáng tạo

KHOA CÔNG NGHỆ THÔNG TIN