ĐẠI HỌC QUỐC GIA TPHCM

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

Báo cáo Đồ án Thực hành Cuối kỳ
Đề 2 - Bài toán data fitting

Môn học: Toán ứng dụng và thống kê - MTH00051

Trương Thành Nhân (21120105)

Lớp: 21_2
Giáo viên hướng dẫn:
Thầy Nguyễn Hữu Toàn

Ngày 20 tháng 6 năm 2023

Mục lục

1 Thông tin chung

1.1 Thông tin sinh viên

- Họ tên: Trương Thành Nhân
- MSSV: 21120105
- Lớp: 21_2

1.2 Đề bài lựa chọn

Đề 2. Bài toán data fitting

- Sử dụng bài toán data fitting trong xây dựng mô hình đánh giá lương nhân viên từ các yếu tố tác động theo dữ liệu được cung cấp.
- Sử dụng bài toán data fitting trong việc xây dựng mô hình đánh giá giá nhà từ các yếu tố tác động từ dữ liệu được cung cấp

2 Tóm tắt kiến thức và các bước xác định một số dạng của mô hình hồi quy:

2.1 Tóm tắt

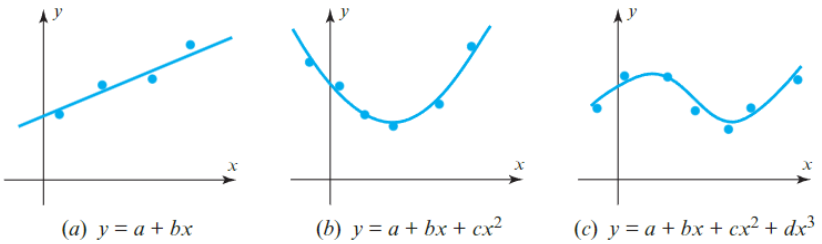
Mô hình hồi quy là một phương pháp thống kê dùng để dự đoán một biến số liên tục dựa trên các biến số độc lập khác. Trong mô hình hồi quy, ta tìm ra một mối quan hệ giữa biến phụ thuộc (có giá trị cần dự đoán) và các biến độc lập (biến đầu vào được sử dụng để dự đoán) ví dụ như mối quan hệ $y = f(x)$ giữa x và y bằng cách "khớp" một đường cong với các điểm trong mặt phẳng tương ứng với các giá trị khác nhau được xác định bằng thực nghiệm của x and y, chẳng hạn $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$.

Trên cơ sở xem xét lý thuyết hoặc quan sát mô hình của các điểm, ta có thể quyết định dạng tổng quát của đường $f(x)$ được khớp. Đường cong này được gọi là mô hình toán học của dữ liệu.

Ngoài ra, để tìm ra các hệ số hồi quy, chúng ta sử dụng *phương pháp bình phương tối thiểu* sai số giữa các giá trị dự đoán và giá trị thực tế.

Có nhiều loại hồi quy, bao gồm hồi quy tuyến tính đơn giản, hồi quy tuyến tính đa biến, hồi quy phi tuyến, vv. Mỗi loại hồi quy có đặc điểm riêng và được áp dụng trong các tình huống khác nhau.

Một số đồ thị ví dụ về mô hình hồi quy:



(a) $y = a + bx$ (b) $y = a + bx + cx^2$ (c) $y = a + bx + cx^2 + dx^3$

2.2 Xây dựng mô hình hồi quy sử dụng bình phương nhỏ nhất (least squares)

2.2.1 Mô hình hồi quy tuyến tính $y = \theta_1 + \theta_2 x$

Ví dụ ta có bảng dữ liệu

x	x_1	x_2	...	x_n
y	y_1	y_2	...	y_n

Đặt ma trận cột chứa các tham số của mô hình viết các dữ liệu được cho thành ma trận như sau:

$$\theta = \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix}, A = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix}, Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

Gọi $\hat{y}$ là giá trị dự đoán được từ mô hình. Khi đó, **tổng bình phương sai số**:

$$RSS = \sum_{i=1}^{m} (\hat{y}_i - y_i)^2 = \sum_{i=1}^{m} (\theta_1 + \theta_2 x - y_i)^2 = ||A\theta - Y||^2$$

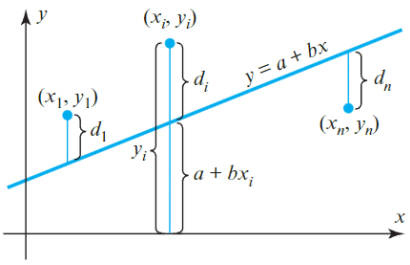
$$RSS_{min} = ||A\theta - Y||^2_{\min} \Leftrightarrow \theta = (A^\top A)^{-1} A^\top Y$$

Do đó ta có công thức tổng quát để tìm ma trận cột chứa các hệ số của mô hình hồi quy như sau:

$$\boxed{\theta = (A^\top A)^{-1} A^\top Y}$$

Sau khi tính ra được các tham số của mô hình $\theta = \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix}$ thì $y = \theta_1 + \theta_2 x$ chính là đường hồi quy của mô hình (**least squares line of best fit** or the **regression line**).

Ngoài ra, các đại lượng $d_1 = |\theta_1 + \theta_2 x_1 - y_1|, d_2 = |\theta_1 + \theta_2 x_2 - y_2|, ..., d_n = |\theta_1 + \theta_2 x_n - y_n|$ được gọi là các phần dư (**residuals**). Vì phần dư là khoảng cách giữa điểm dữ liệu (x_i, y_i) và đường hồi quy nên chúng ta có thể hiểu giá trị của nó là "lỗi" (**error**) ở điểm x_i.



Nếu chúng ta giả sử giá trị của mỗi x_i là chính xác, thì tất cả các "**error**" đều nằm trong y_i. Nên *đường hồi quy* có thể được mô tả là đường cực tiểu hóa tổng bình phương của các lỗi dữ liệu (*the line that minimizes the sum of the squaresof the data errors.*

Do đó nó có tên là "đường bình phương nhỏ nhất phù hợp nhất" (*least squares line of best fit*).
Ta có thể tính chuẩn của vector phần dư dựa vào công thức:

$$r = \hat{Y} - Y = \begin{pmatrix} \hat{y_1} - y_1 \\ \hat{y_2} - y_2 \\ \dots \\ \hat{y_n} - y_n \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ \dots \\ r_n \end{pmatrix} \Longrightarrow \boxed{||\mathrm{r}|| = \sqrt{\sum_{i=1}^{n} r_i^2}}$$

Nhận xét: Từ đó ta có thể thấy, những mô hình hồi quy nào có **||r||** càng **nhỏ** thì mô hình đó sẽ cho kết quả **tốt** hơn và **tối ưu** hơn

2.2.2 Mô hình đa thức $y = \theta_0 + \theta_1 x + \ + \theta_m x^m$

Tương tự như ở mô hình hồi quy tuyến tính ở **2.1.1** Đặt ma trận cột chứa các tham số của mô hình là:

$$\theta = \begin{pmatrix} \theta_1 \\ \vdots \\ \theta_n \end{pmatrix}, Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

Tuy nhiên, ở ma trận **A**, ta lập như sau: **Ứng với mỗi cột trong cùng hàng thứ** i **là** x_i **với hệ số mũ tương ứng giống như mô hình đã cho ban đầu**

$$A = \begin{pmatrix} 1 & x_1 & x_1^2 & \dots & x_1^m \\ 1 & x_2 & x_2 2 & \dots & x_2^m \\ \vdots & \vdots & & & \\ 1 & x_n & x_n^2 & \dots & x_n^m \end{pmatrix}$$

Sau đó làm tương tự như trên, ta tìm $\theta = (A^\top A)^{-1} A^\top Y$ và tính chuẩn vector phần dư **||r||**

2.2.3 Mô hình log - tuyến tính $\ln y = \theta_0 + \theta_1 x$

Tương tự ở mô hình hồi quy tuyến tính ở **2.1.1**.
Đối với ma trận **Y**, ta sẽ lấy **ln** của các phần tử trong **Y**. Ta có các ma trận:

$$\theta = \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix}, A = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix}, Y = \begin{pmatrix} \ln y_1 \\ \ln y_2 \\ \vdots \\ \ln y_n \end{pmatrix}$$

Sau đó làm tương tự như trên, ta tìm $\theta = (A^\top A)^{-1} A^\top Y$ và tính chuẩn vector phần dư **||r||**

2.2.4 Mô hình chứa nhiều biến $y = \theta_0 + \theta_1 x_1 + ... + \theta_n x_n$

Trong đó:

- y là biến phụ thuộc (biến mục tiêu).
- $x_1, x_2, ..., x_n$ là các biến độc lập (biến đầu vào)

- $\theta_0, \theta_1, ..., \theta_n$ là các tham số mô hình (hệ số hồi quy).

Giống với mô hình hồi quy tuyến tính ở **2.1.1** Đặt ma trận cột chứa các tham số của mô hình là:

$$\theta = \begin{pmatrix} \theta_1 \\ \vdots \\ \theta_n \end{pmatrix}, Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

Với ma trận **A**, ta lập như sau: **Ứng với mỗi cột trong cùng hàng thứ i là x_i tương ứng trong hàm số đã cho**

$$A = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{pmatrix}$$

Sau đó làm tương tự như trên, ta tìm $\theta = (A^\top A)^{-1}(A^\top Y)$ và tính chuẩn vector phần dư $||\mathbf{r}||$

2.2.5 Một số dạng mô hình hồi quy khác

- Mô hình cubic: $Y = \theta_1 + \theta_2 X^2$
- Mô hình tuyến tính - log: $Y = \theta_1 + \theta_2 \ln X$
- Mô hình log - log: $\ln Y = \theta_1 + \theta_2 \ln X$

3 Ứng dụng data fitting vào bài toán thực tế

3.1 Định nghĩa một số hàm

3.1.1 Một số thư viện hỗ trợ

- **pandas** hỗ trợ đọc các dữ liệu từ tập tin data cho trước.
- **numpy** hỗ trợ chuyển các dữ liệu đọc được thành array.
- **matplotlib.pyplot** hỗ trợ việc vẽ các mô hình hồi quy.
- **math** hỗ trợ việc tính log ở một số mô hình hồi quy.

3.1.2 Các hàm hỗ trợ

1. Hàm **inverse(Matrix)**: Hàm tìm ma trận nghịch đảo của ma trận *Matrix* ban đầu.
   - Input: Ma trận vuông *Matrix*.
   - Output: Ma trận nghịch đảo của *Matrix*.

2. Hàm **transpose(Matrix)**: Hàm tìm ma trận chuyển vị của *Matrix*.
   - Input: Ma trận *Matrix*.
   - Output: Ma trận chuyển vị của *Matrix*.

3. Hàm **multiplyMatrix(A, B)**: Hàm tính tích ma trận của hai ma trận A và B.
   - Input: Ma trận A và B.
   - Output: Ma trận *result* là tích của A và B.

4. Hàm **addColumn(Matrix)**: Hàm bổ sung thêm cột số 1 vào ma trận *Matrix*.
   - Input: Ma trận *Matrix*.
   - Output: Ma trận *Matrix* đã được bổ sung thêm cột số 1.

5. Hàm **linearRegressionModel(A, Y)**: Hàm tính vector v* chứa các hệ số của mô hình hồi quy dựa vào ma trận các yếu tố tác động A và ma trận cột chứa giá trị thực tế Y.
   - Input: Ma trận dữ liệu A và ma trận cột Y chứa các giá trị thực tế.
   - Output: $v*$ chứa các giá trị θ.
   Dựa theo công thức sau:

$$v* = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \dots \\ \theta_n \end{bmatrix} = (A^\top A)^{-1} A^\top Y$$

```
def linearRegressionModel(A, Y):
    _A = transpose(A)                          // A^T
    T1 = inverse(multiplyMatrix(_A, A))        // (A^T.A)^-1
    T2 = multiplyMatrix(_A, Y)                 // A^T.Y
    y = multiplyMatrix(T1, T2)                 // (A^T.A)^-1.A^T.Y
    for i in range(len(y)):                    // Lam tron cac he so den chu so thap phan thu 7
        y[i][0] = round(y[i][0], 7)
    return y
```

6. Hàm **findResidual(A, Y, y)**: Hàm tính chuẩn vector phần dư (residual) **||r||**.
   -Input: Ma trận dữ liệu A, ma trận cột chứa các giá trị thực Y và ma trận cột/vector y chứa các giá trị θ vừa tìm được.
   - Output: Gía trị của chuẩn vector phần dư $||r||$.

   Theo công thức sau:

$$r = \hat{Y} - Y = \begin{pmatrix} \hat{y_1} - y_1 \\ \hat{y_2} - y_2 \\ \dots \\ \hat{y_n} - y_n \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ \dots \\ r_n \end{pmatrix} \Longrightarrow ||r|| = \sqrt{\sum_{i=1}^{n} r_i^2}$$

```
def findResidual(A, Y, y):
    R = [[None]*(len(Y[0])) for _ in range(len(Y))]
    for i in range(len(R)):
        tmp = 0
        for j in range(len(y)):
            tmp += A[i][j] * y[j][0]
        R[i][0] = tmp - Y[i][0]     // Tim cac gia tri yi^ - yi
    result = 0
    for i in range(len(R)):
        result += R[i][0] * R[i][0]
    return round(result**(1/2),7)   // Lam tron ket qua den chu so thap phan thu 7
```

7. Hàm **printLinearRegressionModel(y, check)**: Hàm in ra mô hình hồi quy tìm được.
   - Input: Ma trận cột y chứa các giá trị θ vừa tìm được, và biến số nguyên *check* ($1 \leq check \leq 4$)

   - **check = 1:** mô hình $y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$

- **check = 2:** mô hình $\ln y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$
- **check = 3:** mô hình $\ln y = \theta_0 + \theta_1 lnx_1 + \theta_2 lnx_2 + \theta_3 lnx_3 + \theta_4 lnx_4$
- **check = 4:** mô hình $y = \theta_0 + \theta_1 x_1 + \theta_2 x_2^2 + \theta_3 x_3^3$

- Output: Mô hình hồi quy tuyến tính ứng với các giá trị θ và check đó.

8. Hàm **printResult(result, A, y, check)**: Hàm in ra kết quả cuối cùng.
- Input: Ma trận cột *result* chứa các hệ số của mô hình, ma trận dữ liệu A, ma trận cột y và số nguyên *check*.
- Output: In ra các **hệ số** của mô hình hồi quy tuyến tính, **mô hình hồi quy tuyến tính** và **chuẩn của vector phần dư** của mô hình tìm được ứng với giá trị **check** ở trên.

3.2 Xây dựng mô hình đánh giá lương nhân viên từ các yếu tố tác động theo dữ liệu được cung cấp

Ta sẽ xét một vài mô hình, sau đó dựa trên chuẩn vector phần dư ||r|| để tìm ra mô hình tối ưu hơn so với những mô hình còn lại.

Đầu tiên, sử dụng thư viện **panda** đọc các dữ liệu từ file **cps4_small.xlsx**

```
df = pd.read_excel('cps4_small.xlsx')
```

	wage	educ	exper	hrswk
0	18.70	16	39	37
1	11.50	12	16	62
2	15.04	16	13	40
3	25.95	14	11	40
4	24.03	12	51	40
...	...	...	...	...
995	16.83	16	27	40
996	28.85	13	7	40
997	11.25	14	5	40
998	7.50	8	36	40
999	8.50	13	22	40

1000 rows × 4 columns

Dùng thư viện **numpy** chuyển các giá trị đọc được sang array.
Lấy các giá trị dữ liệu về yếu tố tác động (*educ, exper, hrswk*) tương ứng với 3 cột cuối thành một ma trận a và các giá trị thực (*wage*) tương ứng với cột đầu tiên thành một ma trận cột y khác

```
data = np.array(df)
a = data[:,-3:]
y = data[:,:-3]
```

3.2.1 Sử dụng mô hình tuyến tính $y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$

- Bổ sung thêm cột số 1 vào ma trận a bằng hàm **A = addColumn(a)**
- Tìm các hệ số của mô hình hồi quy bằng hàm **result = linearRegressionModel(A, y)**
- Sử dụng hàm **printResult(result, A, y)** để in các kết quả ra màn hình, bao gồm:
  - Các hệ số của mô hình hồi quy:

$$v* = (A^\top A)^{-1} A^\top Y = \begin{pmatrix} -16.4432252 \\ 2.0119945 \\ 0.1437088 \\ 0.1373133 \end{pmatrix}$$

  - Mô hình hồi quy tuyến tính tương ứng

$$\boxed{\mathbf{y} = \text{-}16.4432252 + 2.0119945\text{x}_1 + 0.1437088x_2 + 0.1373133x_3}$$

  - Chuẩn vector phần dư $||r_1|| = 361.0899821$

3.2.2 Sử dụng mô hình log - tuyến tính $\ln y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$

- Tìm ma trận cột $newY$ mới bằng cách lấy **ln** các phần tử trong y

```
newY = [[None]*(len(y[0])) for _ in range(len(y))]
for i in range(len(y)):
    newY[i][0] = math.log(y[i])
```

- Bổ sung thêm cột số 1 vào ma trận a bằng hàm **A = addColumn(a)**
- Tìm các hệ số của mô hình hồi quy bằng hàm **result = linearRegressionModel(A, newY)**
- Dùng hàm **printResult(result, A, newY)** để in các kết quả ra màn hình, bao gồm:
  - Các hệ số của mô hình hồi quy:

$$v* = (A^\top A)^{-1} A^\top Y = \begin{pmatrix} 1.1005398 \\ 0.0903056 \\ 0.0057759 \\ 0.0089411 \end{pmatrix}$$

  - Mô hình hồi quy tương ứng

$$\boxed{\ln y = 1.1005398 + 0.0903056x_1 + 0.0057759x_2 + 0089411x_3}$$

    hay

$$\boxed{\text{y} = \text{e}^{1.1005398+0.0903056x_1+0.0057759x_2+0.0089411x_3}}$$

  - Chuẩn vector phần dư $||r_2|| = 16.2111707$

3.2.3 Sử dụng mô hình đa thức $y = \theta_0 + \theta_1 x_1 + \theta_2 x_2^2 + \theta_3 x_3^3$

- Tìm ma trận cột *newA* mới bằng cách lấy lấy mũ các phần tử trong y theo các vị trí cột tương ứng x_1, x_2^2 và x_3^3

```
newA = [[None] * (len(a[0])) for _ in range(len(a))]
for i in range(len(newA)):
    for j in range(len(newA[0])):
        newA[i][j] = a[i][j]**(j+1)
```

- Bổ sung thêm cột số 1 vào ma trận *newA* vừa tìm được bằng hàm **A = addColumn(newA)**
- Tìm các hệ số của mô hình hồi quy bằng hàm **result = linearRegressionModel(A, Y)**
- Dùng hàm **printResult(result, A, y)** để in các kết quả ra màn hình, bao gồm:
  - Các hệ số của mô hình hồi quy:

$$v* = (A^\top A)^{-1} A^\top Y = \begin{pmatrix} -9.45003 \\ 2.0132602 \\ 0.001659 \\ 1.11.10^{-5} \end{pmatrix}$$

  - Mô hình hồi quy tương ứng

$$\boxed{\mathbf{y} = 9.45003 + 2.0132602\mathrm{x}_1 + 0.001659x_2^2 + 1.11.10^{-5}x_3^3}$$

  - Chuẩn vector phần dư $||r_3|| = 365.9717226$

3.2.4 Nhận xét

Dựa vào **chuẩn vector phần dư ||r||**, mô hình nào có **||r|| càng nhỏ** thì sẽ **tối ưu hơn** và cho **kết quả tốt hơn**.

Ta thấy $||r_3|| > ||r_1|| >> ||r_2||$ (365.9717226 > 361.0899821 >> 16.2111707)
$\Longrightarrow$ Mô hình ở **3.2.2 (log - tuyến tính** $\ln y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$**)** sẽ cho kết quả tốt hơn.

Vậy ta lựa chọn **mô hình đánh giá lương nhân viên** là:

$$\boxed{\ln y = 1.1005398 + 0.0903056x_1 + 0.0057759x_2 + 0.0089411x_3}$$

hay

$$\boxed{\mathrm{y} = \mathrm{e}^{1.1005398+0.0903056x_1+0.0057759x_2+0.0089411x_3}}$$

với x_1: educ, x_2: exper, x_3: hrswk

Khi đó, chuẩn vector phần dư $\boxed{||\mathrm{r}|| = 16.2111707}$

3.3 Xây dựng mô hình đánh giá giá nhà từ các yếu tố tác động từ dữ liệu được cung cấp.

Ta cũng sẽ xét một vài mô hình, sau đó dựa trên chuẩn vector phần dư ||r|| để tìm ra mô hình tối ưu hơn so với những mô hình còn lại.

Đầu tiên, sử dụng thư viện **panda** đọc các dữ liệu từ file **cps4_small.xlsx**

```
house = pd.read_excel('br2.xlsx')
```

	price	sqft	Bedrooms	Baths	Age
0	66500	741	1	1	18
1	66000	741	1	1	18
2	68500	790	1	1	18
3	102000	2783	2	2	18
4	54000	1165	2	1	35
...	...	...	...	...	...
1075	122570	2853	5	3	25
1076	185000	4599	5	3	13
1077	1280000	7086	5	3	13
1078	123808	3148	5	2	25
1079	374000	6203	7	4	25

1080 rows × 5 columns

Dùng thư viện **numpy** chuyển các giá trị đọc được sang array.
Lấy các giá trị dữ liệu về yếu tố tác động (*sqft, Bedrooms, Baths, Age*) tương ứng với 3 cột cuối thành một ma trận m và các giá trị thực (*price*) tương ứng với cột đầu tiên thành một ma trận cột b khác

```
data = np.array(house)
m = data[:,-4:]
b = data[:,:-4]
```

3.3.1 Sử dụng mô hình tuyến tính $y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4$

- Bổ sung thêm cột số 1 vào ma trận m bằng hàm **M = addColumn(m)**
- Tìm các hệ số của mô hình hồi quy bằng hàm **result = linearRegressionModel(M, b)**
- Sử dụng hàm **printResult(result, M, b)** để in các kết quả ra màn hình, bao gồm:
  - Các hệ số của mô hình hồi quy:

$$v* = (A^\top A)^{-1} A^\top Y = \begin{pmatrix} -26733.1238772 \\ 87.5006884 \\ -29192.6536254 \\ 41420.7677794 \\ -553.4032903 \end{pmatrix}$$

- Mô hình hồi quy tuyến tính tương ứng

$$\boxed{\mathbf{y} = \text{-26733.1238772} + 87.5006884\text{x}_1 - 29192.6536254x_2 + 41420.7677794x_3 - 553.4032903x_4}$$

- Chuẩn vector phần dư $||r_1|| = 2493863.7509459$

3.3.2 Sử dụng mô hình log - log $\ln y = \theta_0 + \theta_1 lnx_1 + \theta_2 lnx_2 + \theta_3 lnx_3 + \theta_4 lnx_4$

- Tìm ma trận cột *newB* mới bằng cách lấy **ln** các phần tử trong b

```
    newB = [[None]*(len(b[0])) for _ in range(len(b))]
    for i in range(len(b)):
        newB[i][0] = math.log(b[i])
```

- Tìm ma trận dữ liệu *newm* bằng cách lấy giá trị **ln** của các phần tử trong m

```
newm = [[None]*(len(m[0])) for _ in range(len(m))]
for i in range(len(m)):
    for j in range(len(m[0])):
        newm[i][j] = math.log(m[i][j])
```

- Bổ sung thêm cột số 1 vào ma trận *newm* bằng hàm **M = addColumn(newm)**
- Tìm các hệ số của mô hình hồi quy bằng hàm **result = linearRegressionModel(M, newB)**
- Sử dụng hàm **printResult(result, M, newB)** để in các kết quả ra màn hình, bao gồm:
  - Các hệ số của mô hình hồi quy:

$$v* = (A^\top A)^{-1} A^\top Y = \begin{pmatrix} 5.4879381 \\ 0.8514173 \\ -0.2802541 \\ 0.3992267 \\ -0.0669347 \end{pmatrix}$$

  - Mô hình hồi quy tương ứng

$$\boxed{\ln y = 5.4879381 + 0.8514173 \ln x_1 - 0.2802541 \ln x_2 + 0.3992267 \ln x_3 - 0.0669347 \ln x_4}$$

  hay

$$\boxed{\text{y} = \text{e}^{5.4879381 + 0.8514173 \ln x_1 - 0.2802541 \ln x_2 + 0.3992267 \ln x_3 - 0.0669347 \ln x_4}}$$

  - Chuẩn vector phần dư $||r_1|| = 9.683687$

3.3.3 Sử dụng mô hình log - tuyến tính $\ln y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4$

- Tìm ma trận cột *newB* mới bằng cách lấy **ln** các phần tử trong b

```
    newB = [[None]*(len(b[0])) for _ in range(len(b))]
    for i in range(len(b)):
        newB[i][0] = math.log(b[i])
```

- Bổ sung thêm cột số 1 vào ma trận *newm* bằng hàm **M = addColumn(newm)**
- Tìm các hệ số của mô hình hồi quy bằng hàm **result = linearRegressionModel(M, newB)**
- Sử dụng hàm **printResult(result, M, newB)** để in các kết quả ra màn hình, bao gồm:
  - Các hệ số của mô hình hồi quy:

$$v* = (A^\top A)^{-1} A^\top Y = \begin{pmatrix} 10.9189639 \\ 0.0003308 \\ -0.0589539 \\ 0.2145707 \\ -0.0066016 \end{pmatrix}$$

  - Mô hình hồi quy tương ứng

$$\boxed{\ln y = 10.9189639 + 0.0003308x_1 - 0.0589539x_2 + 0.2145707x_3 - 0.0066016x_4}$$

    hay

$$\boxed{\text{y} = \text{e}^{10.9189639+0.0003308x_1-0.0589539x_2+0.2145707x_3-0.0066016x_4}}$$

  - Chuẩn vector phần dư $||r_1|| = 9.1239914$

3.3.4 Nhận xét

Dựa vào **chuẩn vector phần dư ||r||**, mô hình nào có **||r|| càng nhỏ** thì sẽ **tối ưu hơn** và cho **kết quả tốt hơn**.

Ta thấy $||r_1|| >> ||r_2|| > ||r_3||$ (2493863.7509459 >> 9.683687 > 9.1239914)
$\Longrightarrow$ Mô hình ở **3.3.3 (log - tuyến tính** $\ln y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4$**)** sẽ cho kết quả tốt hơn.

Vậy ta lựa chọn **mô hình đánh giá giá nhà** là:

$$\boxed{\ln y = 10.9189639 + 0.0003308x_1 - 0.0589539x_2 + 0.2145707x_3 - 0.0066016x_4}$$

hay

$$\boxed{\text{y} = \text{e}^{10.9189639+0.0003308x_1-0.0589539x_2+0.2145707x_3-0.0066016x_4}}$$

với x_1: sqft, x_2: Bedrooms, x_3: Baths, x_4: Age

Khi đó, chuẩn vector phần dư $\boxed{||\text{r}|| = 9.1239914}$

4 Kết luận

Bên cạnh hai bài toán thực tế được đề cập trên đây, mô hình hồi quy còn có thể được áp dụng trong nhiều lĩnh vực như kinh tế học, y học, khoa học xã hội, khoa học dữ liệu:

- **Dự báo:** Mô hình hồi quy được sử dụng rộng rãi trong việc dự báo giá cổ phiếu, giá nhà, doanh số bán hàng, lợi nhuận, dự đoán kết quả của cuộc bầu cử đến dự báo sản xuất, tài chính hoặc y tế...

- **Phân tích xu hướng:** Mô hình hồi quy có thể được sử dụng để phân tích xu hướng trong dữ liệu, nhằm giúp người dùng hiểu rõ hơn về sự tăng trưởng hoặc suy giảm của các biến số.
- **Tối ưu hóa:** Với mô hình hồi quy, người sử dụng có thể tối ưu hóa các yếu tố quan trọng, như chi phí sản xuất, tỷ lệ chuyển đổi khách hàng, để đưa ra quyết định kinh doanh thông minh.
- **Phân tích liên quan giữa các biến:** Mô hình hồi quy cũng có thể được sử dụng để phân tích mối quan hệ giữa các biến, như tác động của giới tính, độ tuổi và thu nhập đến sức khỏe, hạnh phúc của mỗi người.
- **Kiểm tra giả thuyết:** Mô hình hồi quy cũng được sử dụng trong kiểm tra giả thuyết, xác định liệu có mối liên hệ giữa biến phụ thuộc và độc lập hay không, hoặc để kiểm tra sự ảnh hưởng của một biến đến biến phụ thuộc.

Tuy nhiên, mô hình hồi quy có thể chỉ đúng với một số giả định nhất định, và chúng ta cần phải kiểm tra các giả định này để đảm bảo tính chính xác và độ tin cậy của dự đoán.