TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN ĐHQG-HCM

KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN 3 – Linear regression

Môn Học: Toán ứng dụng & thống kê

Giảng viên:

Nguyễn Trọng Hiến

Nguyễn Văn Quang Huy

Nguyễn Đình Thúc

Võ Nam Thục Đoan

Sinh viên thực hiện:

Nguyễn Tấn Phát 20127588

1. Nội dung đồ án:

- File "wine.csv" là cơ sở dữ liệu đánh giá chất lượng của 1200 chai rượu vang theo thang điểm 1_10 dựa trên 11 tính chất khác nhau.

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcoh
0	7.4	0.700	0.00	1.9	0.076	11.0	34	0.99780	3.51	0.56	9
1	7.8	0.880	0.00	2.6	0.098	25.0	67	0.99680	3.20	0.68	9
2	7.8	0.760	0.04	2.3	0.092	15.0	54	0.99700	3.26	0.65	9
3	11.2	0.280	0.56	1.9	0.075	17.0	60	0.99800	3.16	0.58	9
4	7.4	0.700	0.00	1.9	0.076	11.0	34	0.99780	3.51	0.56	9
...	...	...	...	...	...	...	...	...	...	...	
1194	7.0	0.745	0.12	1.8	0.114	15.0	64	0.99588	3.22	0.59	9
1195	6.2	0.430	0.22	1.8	0.078	21.0	56	0.99633	3.52	0.60	9
1196	7.9	0.580	0.23	2.3	0.076	23.0	94	0.99686	3.21	0.58	9
1197	7.7	0.570	0.21	1.5	0.069	4.0	9	0.99458	3.16	0.54	9
1198	7.7	0.260	0.26	2.0	0.052	19.0	77	0.99510	3.15	0.79	10

- Xây dựng mô hình đánh giá chất lượng rượu sử dụng phương pháp hồi quy tuyến tính.

a) Sử dụng toàn bộ 11 đặc trưng đề bài cung cấp,

- $y=theta\_1x\_1+theta\_2x\_2+...+theta\_11x\_11$

b) Sử dụng duy nhất 1 đặc trưng cho kết quả tốt nhất. (Gợi ý: Phương pháp Cross Validation)

- $y=theta\_ix\_i$ (dùng mô hình lần lượt cho từng đặc trưng).

c) Xây dựng một mô hình của riêng bạn cho kết quả tốt nhất.

2. Môi trường làm việc:

- Ngôn ngữ lập trình: Python
- Text Editer: Visual Studio Code
- Thư viện hỗ trợ: numpy, pandas, matplotlib

3. Ý tưởng và các hàm:

a) Ý tưởng giải quyết bài toán

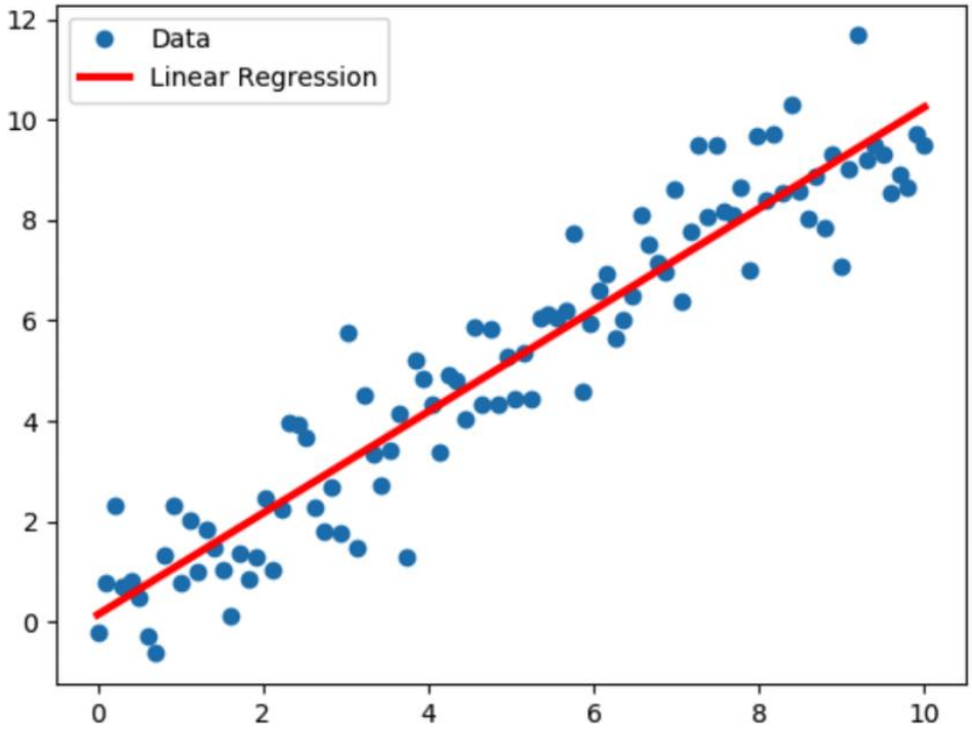
- Xác định mô hình cần phải đưa về:
  - Vd: y = ax + b
  - Hay y = [a, b].T * [x, 1]
- Phân tích dữ liệu đầu vào thành 2 phần: hồi quy và phần dư:

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

- Khi đó
  - [a, b] chính là nghiệm của Linear Regresstion cần phải tìm.
  - [x, 1] chính là ma trận hồi quy
  - [y] chính là phần dư
- Đặt A = [x, 1] khi đó:
  - $[a, b] = (A^T * A)^{-1} * A^T * [y]$

b) Các hàm:

- createMatrix(): tạo ma trận A phía trên
- resultMatrix(): tạo ma trận [y]



4. Các hàm hỗ trợ:

a. *Numpy.linalg.inv()*

- *Chức năng: Tính ma trận khả nghịch*
- VD:
  - Input: [A]
  - Output: [A] ^ -1

b. *Numpy.transpose()*

- *Chức năng: Trả về ma trận nghịch đảo*
- VD:
  - Input: [A]
  - Output: [A].T

c. *Numpy.dot*

- Chức năng: Nhân 2 ma trận với nhau
- VD:
  - Input: [A] [B]
  - Output: [A.B]

5. Hồi quy tuyến tính:

- LinearRegressstion(matrix, result): Truyền vào matrix và resultMatrix
  - Áp dụng công thức: [a, b] = $(A^T * A)^{-1} * A^T * [y]$
  - Trả về [a, b] trong y = ax + b
  - Hoặc [a, b, c] trong y = ax1 + bx2 + c
- leastSquares(data, title, titleResult): Truyền vào dữ liệu, tên tiêu chí cần tính bình phương tối tiểu và tên tiêu chí kết quả
  - Được tính qua công thức L =$\sum(y_{xi} - f(xi)^2$
  - Với y là kết quả thực tế.
  - Và f(xi) là hàm hồi quy tuyến tính
- drawLinearRegression: Dùng để vẽ hàm hồi quy tuyến tính để có cái nhìn trực quan hơn.

6. Kết quả chạy thử:

a. Sử dụng toàn bộ 11 đặc trưng đề bài cung cấp

```
titles = ["fixed acidity", "volatile acidity", "citric acid", "residual suga
Matrix = createMatrix(df, titles)
result = matrixResult(df, "quality")

x = LinearRegression(Matrix, result)
print("y = ", end='')
for i in range(len(x)):
    theta = 'x' + str(i + 1)
    print(round(x[i][0], 5), ' * ', theta, end = '')

    if i != len(x) - 1:
        print(' + ', end='')
```

✓ 0.5s Python

```
y = 0.00593 * x1 + -1.10804 * x2 + -0.26305 * x3 + 0.01532 * x4 +
-1.7305 * x5 + 0.0038 * x6 + -0.0039 * x7 + 4.33859 * x8 + -0.45854 *
x9 + 0.72972 * x10 + 0.30886 * x11
```

b. Sử dụng duy nhất 1 đặc trưng cho kết quả tốt nhất

- Tính tổng tất các bình phương tối tiểu của dữ liệu đầu vào.
- Khi đó, đặc trưng cho bình phương tối tiểu nhất sẽ tốt nhất.

```
for i in range(len(titles)):
    print(i, end=' ')
    leastSquares(df, titles[i], "quality")
```

✓ 0.8s Python

```
0 __ Least Squares of fixed acidity = 767
1 __ Least Squares of volatile acidity = 671
2 __ Least Squares of citric acid = 744
3 __ Least Squares of residual sugar = 783
4 __ Least Squares of chlorides = 773
5 __ Least Squares of free sulfur dioxide = 780
6 __ Least Squares of total sulfur dioxide = 744
7 __ Least Squares of density = 756
8 __ Least Squares of pH = 780
9 __ Least Squares of sulphates = 750
10 __ Least Squares of alcohol = 584
```

Vậy đặc trưng tốt nhất là alcohol. Sẽ có mô hình là

```
alcolMatrix = createMatrix(df, ["alcohol", None])
alcolResult = matrixResult(df, "quality")
alcol = LinearRegression(alcolMatrix, alcolResult)
a = round(alcol[0][0], 5)
b = round(alcol[1][0], 5)
print("y = ", end='')
print(a, ' * x + ', b)
```

✓ 0.3s Python

```
y = 0.37471  * x +  1.77408
```

Từ đó ta xây được mô hình thông qua 3 đặc trưng: citric acid, total sulfur dioxide, alcohol

```
myMatrix = createMatrix(df, ["citric acid", "total sulfur dioxide", "alcohol
myResult = matrixResult(df, "quality")
my = LinearRegression(myMatrix, myResult)
a = round(my[0][0], 5)
b = round(my[1][0], 5)
c = round(my[2][0], 5)
d = round(my[3][0], 5)

print("y = ", end='')
print(a, '*x^3 +', b, '*x^2 +', c, '*x +', d)
```

2] ✓ 0.4s Python

```
y = 0.63694 *x^3 + -0.00331 *x^2 + 0.3361 *x + 2.14316
```

7. Tài liệu tham khảo:

1. Slide bài giảng môn "Toán ứng dụng & thống kê"
2. Slide bài giảng môn "Đại số tuyến tính"

**********HẾT**********