BÁO CÁO ĐỒ ÁN 3

1 Các thư viện được sử dụng

Trong xuyên suốt đồ án này, ta tiến hành khai báo các thư viện được sử dụng trong việc xây dựng các mô hình được yêu cầu như sau

- **numpy** để thực hiện các phép toán về nhân ma trận, giả nghịch đảo ma trận... (Đây cũng là thư viện rất mạnh và phổ biến trên thế giới về đại số tuyến tính)
- **pandas** để đọc dữ liệu từ file về với một định dạng nào đó (Ví dụ như trong đồ án này là ngăn cách bởi dấu ';')

2 Lấy dữ liệu để xây dựng mô hình

Thư viện **pandas** hỗ trợ hàm `pd.read_csv()` cho việc lấy dữ liệu có cấu trúc

- Tham số đầu vào thứ nhất là tên file cần truyền vào để đọc dữ liệu: `wine.csv`
- Tham số đầu vào thứ hai là các dữ liệu trong file ngăn cách bởi dấu nào: `sep = ';'`
- Tham số đầu ra là biến kiểu DataFrame (xem phần output để biết thêm chi tiết)

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.700	0.00	1.9	0.076	11.0	34	0.99780	3.51	0.56	9.4	5
1	7.8	0.880	0.00	2.6	0.098	25.0	67	0.99680	3.20	0.68	9.8	5
2	7.8	0.760	0.04	2.3	0.092	15.0	54	0.99700	3.26	0.65	9.8	5
3	11.2	0.280	0.56	1.9	0.075	17.0	60	0.99800	3.16	0.58	9.8	6
4	7.4	0.700	0.00	1.9	0.076	11.0	34	0.99780	3.51	0.56	9.4	5
...	...	...	...	...	...	...	...	...	...	...	...	...
1194	7.0	0.745	0.12	1.8	0.114	15.0	64	0.99588	3.22	0.59	9.5	6
1195	6.2	0.430	0.22	1.8	0.078	21.0	56	0.99633	3.52	0.60	9.5	6
1196	7.9	0.580	0.23	2.3	0.076	23.0	94	0.99686	3.21	0.58	9.5	6
1197	7.7	0.570	0.21	1.5	0.069	4.0	9	0.99458	3.16	0.54	9.8	6
1198	7.7	0.260	0.26	2.0	0.052	19.0	77	0.99510	3.15	0.79	10.9	6

1199 rows × 12 columns

Hình 1: Dữ liệu từ file `wine.csv`

3 Xây dựng mô hình

3.1 Sử dụng toàn bộ 11 đặc trưng đề bài cung cấp

Mô hình mà ta cần xây dựng có dạng

$$y = \sum_{i=1}^{11} \theta_i * x_i = \theta x^T$$

với x là từng dòng dữ liệu đầu vào (vector dòng) và y là giá trị đầu ra ứng với x, hay nói một cách tổng quát hơn là với mỗi dòng ta có $y = f(x)$

Như vậy ta tìm hệ số θ_i trong θ của từng phần tử x_i trong x bằng công thức tổng quát

$$\theta = X^{\dagger}Y = (X^T X)^{-1} X^T y$$

với $X^{\dagger}$ là ma trận giả nghịch đảo của X (vì có thể X không khả nghịch nên ta phải dùng giả nghịch đảo hoặc biến đổi thành công thức phía sau)

Từ đó ta đưa ra ý tưởng cho việc xây dựng hàm tìm θ dựa vào dữ liệu ta vừa lấy về

- Hàm tên là `linear_regression()` với các mô tả sẽ được liệt kê bên dưới
- Tham số đầu vào duy nhất của hàm là `data_set` đại diện cho bộ dữ liệu ta cần huấn luyện cho mô hình (bao gồm dữ liệu đầu vào và dữ liệu đầu ra tương ứng)
- Tham số đầu ra duy nhất của hàm là `theta` chính là hệ số của vector x

Người đọc có thể xem mã nguồn của hàm này (và các hàm phụ trợ) trong file mã nguồn đính kèm

	x	properties	theta	value
0	x_1	fixed acidity	theta_1	0.005925161374105113
1	x_2	volatile acidity	theta_2	-1.1080375422629067
2	x_3	citric acid	theta_3	-0.2630462837012381
3	x_4	residual sugar	theta_4	0.015322283066645292
4	x_5	chlorides	theta_5	-1.730502743057372
5	x_6	free sulfur dioxide	theta_6	0.003801419076862004
6	x_7	total sulfur dioxide	theta_7	-0.003898998694536247
7	x_8	density	theta_8	4.338587684499409
8	x_9	pH	theta_9	-0.45853547521688753
9	x_10	sulphates	theta_10	0.7297186624705415
10	x_11	alcohol	theta_11	0.30885864844891386

Hình 2: Kết quả của θ sau khi sử dụng hàm `linear_regression()`

3.2 Sử dụng duy nhất 1 đặc trưng cho kết quả tốt nhất

3.2.1 Tổng quan về phương pháp K−Fold Cross Validation

Ta chia n dòng dữ liệu ban đầu thành k mẫu với số lượng phần tử trong mỗi mẫu bằng nhau hoặc chênh lệch nhau không quá 1 đơn vị.

Ta thực hiện lặp k lần. Với lần thứ i $(0 \leq i < k)$, ta chọn mẫu thứ i làm mẫu test và các mẫu còn lại làm mẫu huấn luyện cho mô hình.

- Ta sử dụng tập test để đánh giá mô hình vừa huấn luyện dựa vào giá trị sai lệch (hay dễ hiểu hơn là sai số giữa giá trị output của mô hình và giá trị output của tập test − output thực tế)
- Sau khi có giá trị sai lệch của bộ dữ liệu test, ta tính trung bình giá trị sai lệch của mô hình nếu sử dụng mẫu i làm tập test. Hay ta có công thức tổng quát để tính giá trị sai lệch trung bình của mô hình nếu sử dụng mẫu i làm tập test như sau

$$Loss(i) = \frac{\sum_{j=1}^{|test|} |y_j - f(x_j)|}{|test|}$$

với $|test|$ là số lượng dữ liệu (số lượng phần tử) trong tập test; y_j là giá trị output thực tế của dòng j trong tập test (cột cuối cùng của dòng); $f(x_j) = \theta x_j^T$ là output của mô hình f nếu nhận x_j làm tham số đầu vào

Sau khi vòng lặp hoàn thành công việc, ta có bộ giá trị $\{Loss(1), Loss(2), ..., Loss(k)\}$. Ta tính trung bình sai lệch của toàn bộ mô hình ban đầu bằng công thức

$$Loss = \frac{\sum_{i=1}^{k} Loss(i)}{k}$$

Mô hình được gọi là tốt khi giá trị $Loss$ ta vừa tính càng nhỏ

3.2.2 Tạo mẫu từ bộ dữ liệu

Trong đồ án này ta chọn $k = 6$. Như vậy ta tạo mảng 6 phần tử để lưu số lượng phần tử tại mẫu đó

- Giả sử ta có mảng a gồm $k = 6$ phần tử lần lượt là $a_1, a_2, ..., a_6$ dùng để lưu số lượng phần tử ứng với mẫu i nào đó
- Ta có công thức xác định mẫu i bắt đầu tại dòng nào và kết thúc tại dòng nào (chú ý rằng bộ dữ liệu của ta bắt đầu tại dòng 0)

$$start(i) = \sum_{j=1}^{i-1} a_j$$

$$end(i) = -1 + \sum_{j=1}^{i} a_j$$

3.2.3 Tìm giá trị sai lệch với đặc trưng xác định và với mẫu test cho trước

Tạo hàm tên `cross_validation` với các mô tả về hàm như sau

- Có 2 tham số đầu vào là `(idx, k)` đại diện cho việc ta đang xét đặc trưng idx và xét mẫu k làm mẫu test
- Bộ dữ liệu cần huấn luyện cho mô hình của ta hiện tại sẽ có số dòng bằng số dòng của bộ dữ liệu ban đầu, số cột gồm 2 cột là cột mà ta đang xét đặc trưng và cột cuối cùng (output)
- Tìm dòng bắt đầu và kết thúc của mẫu test (mẫu k) dựa vào công thức được nêu ở phần trên
- Tiếp theo ta tiến hành tách ra thành 2 bộ dữ liệu nhỏ là `training` và `test`
- Sử dụng lại hàm `linear_regression` vừa làm ở trên để tìm hệ số θ cho mô hình
- Ta tính giá trị sai lệch trung bình $Loss(i)$ theo công thức vừa được trình bày ở phần trên
- Tham số đầu ra duy nhất của hàm chính là $Loss(i)$ ta tính được

3.2.4 Tìm đặc trưng tốt nhất cho mô hình

Duyệt theo từng đặc trưng và theo từng mẫu test. Với mỗi đặc trưng ta tính trung bình sai lệch của toàn bộ đặc trưng đó rồi lưu vào mảng. Sau đó sắp xếp theo thứ tự tăng dần và in ra đặc trưng cùng với giá trị trung bình sai lệch của đặc trưng đó

	properties	loss_value
0	alcohol	0.5490115616701011
1	density	0.7078487050943097
2	pH	0.7309991988159452
3	fixed acidity	1.054806874175808
4	sulphates	1.08597890510556
5	volatile acidity	1.846487704639521
6	residual sugar	2.0148893668391956
7	chlorides	2.0936369885451094
8	citric acid	2.598740705513529
9	free sulfur dioxide	2.8451016413556
10	total sulfur dioxide	3.1411396884744636

Hình 3: Giá trị sai lệch trung bình của từng đặc trưng được sắp xếp tăng dần

Vậy đặc trưng tốt nhất cho mô hình là `alcohol`. Ta đi tìm mô hình chỉ sử dụng duy nhất 1 đặc trưng đó.

- Bộ dữ liệu cần huấn luyện có số dòng bằng số dòng của bộ dữ liệu ban đầu, số cột gồm 2 cột là cột của đặc trưng `alcohol` và cột cuối cùng (output)
- Tìm θ bằng hàm `linear_regression`

```
Mô hình nếu chỉ sử dụng đặc trưng Alcohol: y = 0.543705524123724x
```

Hình 4: Mô hình nếu chỉ sử dụng đặc trưng `alcohol`

3.3 Mô hình của bản thân cho kết quả tốt nhất

3.3.1 Như thế nào thì gọi là mô hình tốt?

Bằng việc sử dụng chuẩn vector phần dư, ta đưa ra nhận xét rằng mô hình càng tốt khi giá trị của chuẩn vector phần dư càng nhỏ

Công thức để tính chuẩn vector phần dư (giả sử ta đã tìm được θ)

$$r = ||y - \theta x^T||$$

3.3.2 Mô hình $y = ax$

Mô hình có công thức tổng quát là $y = \sum_{i=1}^{11} \theta_i x_i$ (Mô hình này đã được xây dựng ở phần 3.1)

Với bộ dữ liệu từ file `wine.csv`, ta có giá trị của chuẩn vector phần dư như hình sau (hoặc người đọc có thể xem phần mã nguồn cho việc tính toán ở trong file source code đi kèm)

```
Residual vector value of model 'y = ax' is 22.12434596534916
```

Hình 5: Chuẩn vector phần dư của mô hình $y = ax$

3.3.3 Mô hình $y = ax + b$

Mô hình có công thức tổng quát là $y = \theta_0 + \sum_{i=1}^{11} \theta_i x_i$

```
Residual vector value of model 'y = ax + b' is 22.094716807791656
```

Hình 6: Chuẩn vector phần dư của mô hình $y = ax + b$

Ta có nhận xét mô hình này đã có giá trị chuẩn vector phần dư nhỏ hơn so với mô hình $y = ax$

3.3.4 Mô hình $y = ax^2 + bx + c$

Mô hình có công thức tổng quát là $y = \theta_0 + \sum_{i=1}^{11} \theta_i x_i + \sum_{i=12}^{22} \theta_i x_{i-11}^2$

```
Residual vector value of model 'y = ax^2 + bx + c' is 21.617693192282466
```

Hình 7: Chuẩn vector phần dư của mô hình $y = ax^2 + bx + c$

3.3.5 Tại sao không sử dụng mô hình có $ln(X)$

Sau khi chạy đoạn code kiểm tra có giá trị nào trong tập dữ liệu X nhỏ hơn hoặc bằng 0 hay không, ta thấy kết quả đầu ra có 174 phần tử nhỏ hơn hoặc bằng 0. Do điều kiện của ln là biến phải lớn hơn 0 nên ta không thể sử dụng $ln(X)$

3.3.6 Mô hình $ln(y) = a + bx$

Mô hình có công thức tổng quát là $ln(y) = \theta_0 + \sum_{i=1}^{11} \theta_i x_i$

```
Residual vector value of model 'ln(y) = a + bx' is 22.12508309025727
```

Hình 8: Chuẩn vector phần dư của mô hình $ln(y) = a + bx$

3.3.7 Chọn mô hình tốt nhất cho bản thân

Dựa vào 4 mô hình và 4 giá trị chuẩn vector phần dư được tính ở trên, ta chọn mô hình dạng parabol $y = ax^2 + bx + c$ vì mô hình này cho chuẩn vector phần dư đạt giá trị nhỏ nhất