

Report Lab04 – Linear Regression

AUGUST 30

Toán Ứng dụng và Thống Kê

Tác giả: Phạm Ngọc Thùy Trang – 18127022

PROJECT 03 – LAB04

THÔNG TIN CÁ NHÂN

Họ và tên: Phạm Ngọc Thùy Trang

MSSV: 18127022

Lớp: 18CLC1

GIẢNG VIÊN HƯỚNG DẪN

GV. Bùi Huy Thông

Ý TƯỞNG XÂY DỰNG MÔ HÌNH ĐÁNH GIÁ CHẤT LƯỢNG SỬ DỤNG PHƯƠNG PHÁP HỒI QUY TUYẾN TÍNH

1. Giới thiệu hồi quy tuyến tính là gì?

Như ta đã biết, trong thống kê, hồi quy là một phương pháp được dùng để tìm hiểu và định lượng một mối quan hệ giữa 2 hay nhiều biến bất kỳ. Các mô hình hồi quy rất đa dạng từ đơn giản đến phức tạp, linh hoạt áp dụng cho từng bộ dữ liệu có các đặc tính khác nhau, trong đó có linear regression (hồi quy tuyến tính) là một trong những kỹ thuật cơ bản và quan trọng để dự đoán các giá trị cho một attribute (y). Ý tưởng của Linear Regression là chúng ta sẽ được cho một tập các observations và mỗi observations sẽ được liên kết tới một vài features, chúng ta sẽ muốn dự đoán giá trị thực cho mỗi observations. Và **mục tiêu của giải thuật hồi quy tuyến tính** là dự đoán giá trị của một hoặc nhiều biến mục tiêu liên tục (continuous target variable) y dựa trên một vector đầu vào x .

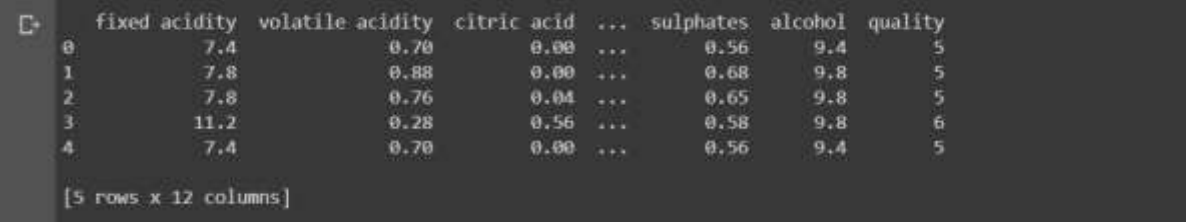
2. Ứng dụng của mô hình hồi quy nói chung và nói riêng trong bài toán hiện tại

Mô hình hồi quy tuyến tính có rất nhiều ứng dụng để giúp các tổ chức, các công ty áp dụng phân tích cho nguồn dữ liệu của mình với 4 mục đích chính (dưới góc độ dữ liệu): **mô tả dữ liệu** (*hình thành phương trình hồi quy để đánh giá tổng quan các mối liên hệ giữa các biến, đồng thời cũng là bài toán chúng ta đang làm*), **ước lượng hệ số hồi quy dựa trên khoảng tin cậy**, **dự báo giá trị của biến phụ thuộc**, **biến mục tiêu** và **kiểm soát các biến độc lập** (biến phụ thuộc bị ảnh hưởng tích cực hay tiêu cực nếu các biến độc lập được điều chỉnh)

3. Ý tưởng xây dựng và làm việc

Như được phân loại ở trên, bài toán chúng ta cần giải quyết chính là **tìm và xây dựng một hàm số $f(x)$ xấp xỉ với điểm dữ liệu dùng để huấn luyện mô hình**, từ đó, ta có thể **thay thế các giá trị đặc trưng** (trong xử lý ngôn ngữ tự nhiên, chúng ta thường gọi các yếu tố dự đoán như số tính từ mơ hồ là các **feature**) **vào và nhận ra được giá trị dự đoán của bài toán**. Cụ thể hơn đối với bài toán “Xây dựng mô hình đánh giá chất lượng rượu được sử dụng phương pháp hồi quy tuyến tính” để đánh giá chất lượng của 1200 chai rượu vang theo thang điểm từ 1 – 10 dựa trên 11 tính chất khác nhau. Chúng ta sẽ biểu diễn mỗi observation (mỗi một chai rượu vang để kiểm định) bằng một vector của những features này. Bằng trực giác, chúng ta sẽ mong muốn chọn được các weights (đi kèm với các giá trị feature sẽ có weight của nó) sao cho giá trị y ước lượng sẽ gần nhất với giá trị thực tế mà ta nhìn thấy trong tập training test

(Hình ảnh dưới đây cho thấy một vài dòng thông tin của 5 chai rượu vang đầu trong số các chai rượu vang của file wine.csv)



	fixed acidity	volatile acidity	citric acid	...	sulphates	alcohol	quality
0	7.4	0.70	0.00	...	0.56	9.4	5
1	7.8	0.88	0.00	...	0.68	9.8	5
2	7.8	0.76	0.04	...	0.65	9.8	5
3	11.2	0.28	0.56	...	0.58	9.8	6
4	7.4	0.70	0.00	...	0.56	9.4	5

[5 rows x 12 columns]

a. Xây dựng mô hình dự đoán chất lượng rượu bằng 11 thuộc tính

Như vậy, với một vector 11 chiều bất kỳ tương ứng với 11 đặc trưng cần hồi quy là x , ta có label là điểm đánh giá được cho sẵn của 1199 chai rượu là y , thì khi đó ta có: $x = [x_0, x_1, x_2, \dots, x_{10}]^T \in R^{11}$

Gọi θ là vector tham số huấn luyện tương ứng với model, bên cạnh đó, để siêu phẳng (siêu phẳng trong không gian n chiều) chúng ta cần thêm một hệ số tự do (bias) vào hàm $f(x)$, vì nếu không có siêu phẳng

thì nó chỉ giới hạn đi qua gốc tọa độ. Do đó mà mô hình của chúng ta sẽ có dạng, trong đó bias là hệ số tự do

$$y = f(x) = bias + theta_0x_0 + theta_1x_1 + \dots + theta_{10}x_{10}$$

Do đó, vector điểm dữ liệu và tham số huấn luyện của chúng ta bây giờ sẽ có dạng như sau:

$$x = [x_0, x_1, x_2, \dots, x_{10}, 1]^T \in R^{12}$$

$$theta = [theta_0, theta_1, theta_2, \dots, theta, 1]^T \in R^{12}$$

Sau đó, xếp toàn bộ 1199 điểm dữ liệu thành một ma trận A với kích thước 1199 x 12 rồi áp dụng phương pháp bình phương tối thiểu (mean least square) để tìm tham số theta

b. Tìm ra thuộc tính nào ảnh hưởng đến chất lượng rượu nhất

Với bài toán này ta sẽ sử dụng phương pháp cross validation để đánh giá thuộc tính rượu, vì dù có 11 thuộc tính nhưng không phải tất cả các thuộc tính đều ảnh hưởng hết đến chất lượng rượu, do đó chúng ta cần tìm ra thuộc tính nào có thể gây ra sự ảnh hưởng lớn nhất. Chúng ta sẽ chia tập data thành k tập con nhỏ không giao nhau và sau đó chọn 1 tập con bất kỳ để làm bộ test, và k – 1 tập còn lại sẽ làm bộ training, chính vì vậy mà tương ứng với mỗi thuộc tính sẽ có k lần huấn luyện.

Chúng ta vẫn sẽ còn sử dụng phương pháp tìm tham số model như câu a (*dùng bias và bình phương tối thiểu*) nhưng sẽ có sự khác biệt trong việc chọn dữ liệu để huấn luyện. Và ứng với mỗi một thuộc tính sẽ có k lần huấn luyện và độ lỗi của thuộc tính đó chính là trung bình độ lỗi của k lần, với tham số là tương ứng với lần có độ lỗi nhỏ nhất, tính độ

lỗi model bằng trung bình độ lỗi của các điểm dữ liệu trong bộ test, cuối cùng là chọn ra thuộc tính tốt nhất tương ứng với độ lỗi nhỏ nhất trong 11 thuộc tính đó.

c. Tự xây dựng mô hình dự đoán của riêng bạn để đạt được kết quả tốt nhất

Ý tưởng thực hiện: từ việc xét theo thuộc tính câu B, ta sẽ lựa chọn kết hợp các thuộc tính tốt nhất lại để thử nghiệm. Cụ thể hơn, ta sẽ thử các trường hợp sử dụng từng đặc trưng khác nhau, ví dụ như là hai đặc trưng tốt nhất, 3 đặc trưng tốt nhất hoặc 4 đặc trưng tốt nhất, v.v... Với cách làm của em thì em chọn 6 thuộc tính có độ lỗi thấp nhất để chạy thử, sau đó tự thay thế thí nghiệm lựa chọn cho cái nào phù hợp rồi tiếp tục lặp lại bước huấn luyện mô hình và dự đoán trên tập test

CÁC CÔNG VIỆC ĐÃ HOÀN THÀNH

Tên chức năng/ công việc	Đánh giá tỉ lệ hoàn thành
Sử dụng toàn bộ 11 đặc trưng đề bài cung cấp	1/1
Sử dụng duy nhất 1 đặc trưng cho kết quả tốt nhất. <i>Gợi ý: dùng phương pháp Cross Validation</i>	1/1
Xây dựng một mô hình của riêng bạn để cho ra kết quả tốt nhất	0.5/1
Tổng:	2.5/3

CÁC THƯ VIỆN SỬ DỤNG:

Thư viện đọc file csv dữ liệu đầu vào: **pandas**

Thư viện tính toán trên ma trận: **numpy**

Thư viện matplotlib để trực quan hóa dữ liệu: **seaborn** và **matplotlib**. (Lưu ý rằng seaborn là thư viện base trên matplotlib, nó như 1 interface để người dùng có thể dễ dàng visualization hơn và vì seaborn là 1 wrapper của matplotlib nên khi sử dụng cần import cả matplotlib, thực tế thì bài này không cần đến 2 thư viện này)

Thư viện dùng cho việc xáo trộn và phân chia dữ liệu cho quá trình huấn luyện model: **sklearn.model_selection**

Thư viện dùng cho việc tính toán các giá trị số: **math**

MÔ TẢ CÁC CÔNG VIỆC VÀ HÀM CỤ THỂ

(Một số chức năng sẽ không được đưa vào hàm, thay vào đó ta sẽ code trực tiếp trên từng cell của file notebook)

Tên chức năng/ tên hàm	Mô tả chính
<pre>df['quality'].value_counts()</pre>	Đếm số lượng các giá trị ở cột quality
<pre>X_data = df.iloc[:, :-1].values y_data = df.iloc[:, -1].values bias = np.ones(X_data.shape[0]) bias = np.resize(bias, (1, X_data.shape[0])) X_data = np.concatenate((X_data, bias.T), axis=1)</pre>	Như đã nói ở trên, ta cần lấy 11 thuộc tính để làm dữ liệu training, thì cột cuối cùng sẽ đóng vai trò là nhãn dự đoán. Dòng code này được dùng để đọc thông tin dữ liệu
<pre>def scale_data(X_data)</pre>	Chuẩn hóa dữ liệu đầu vào là tập một vector 11 chiều bất kỳ tương ứng với 11 đặc trưng cần hồi quy là x, kết quả trả về sẽ là một tập dữ liệu đã được chuẩn hóa để dễ sử dụng
<pre>def shuffle_split_data (X,y,split_rate = 80)</pre>	Chia dữ liệu thành hai tập training và tập test, tương ứng với từng tập training sẽ trả ra giá trị x và y khác

	<p>nhau (một vector 11 chiều bất kỳ tương ứng với 11 đặc trưng cần hồi quy là x, ta có label là điểm đánh giá được cho sẵn của 1199 chai rượu là y).</p> <p>Tham số đầu vào là tập X và tập Y ta tìm được khi đọc thông tin dữ liệu và một tỉ lệ chia = 80</p>
<code>def predict(theta, input)</code>	Hàm này được dùng để dự đoán kết quả bằng cách truyền vào tham số θ và từng giá trị trong bộ X test
<code>def linear_regression(X_train, y_train)</code>	Được dùng để tính giá trị θ , với tham số đầu vào là tập train và trả về kết quả là giá trị θ
<code>def calculate_score(y_test, y_pred)</code>	Hàm này được dùng để tính độ lỗi của model, với tham số truyền vào là tập Y test và tập Y pred (tập Y_pred này ban đầu là một mảng rỗng, sau đó qua vòng lặp dựa trên bộ y_test thì các giá trị mới

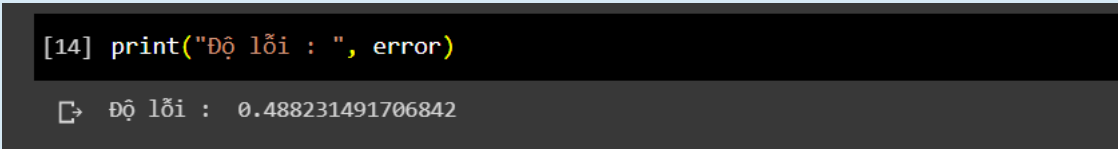
	(dựa trên hàm <code>predict(theta, input)</code> sẽ được đưa vào)
<pre> n_split = 5 kf = Kfold(n_splits=n_split) list_output = [] list_name = [] for index,name in enumerate(df.columns[1:-1]): print("\n\t\tTrình: ", name) list_name.append(name) X_data = np.array(df[name]) X_data = X_data.reshape((X_data.shape[0], 1)) bias = X_data[:, X_data.shape[1]-1].reshape((X_data.shape[0], 1)) X_data = np.concatenate((X_data, bias), axis=1) sum_mae = 0.0 for train_index, test_index in kf.split(X_data): X_train, X_test = X_data[train_index], X_data[test_index] y_train, y_test = y_data[train_index], y_data[test_index] theta = linear_regression(X_train,y_train) y_pred = [] for id in range(len(y_test)): y_pred.append(predict(theta,X_test[id])) y_test = [item for item in y_test] mae = calculate_score(y_test,y_pred) sum_mae += mae print("\nAverage MAE: ",sum_mae/n_split) list_output.append(sum_mae/n_split) print("-----") print(list_output) print("\n\t\tGiá trị tối thiểu là: ", list_name[list_output.index(min(list_output))]) print("\n\t\tĐộ lệch nhỏ là: ", min(list_output)) </pre>	<ul style="list-style-type: none"> Chuẩn bị k tập data cho cross validation, sử dụng thư viện sklearn cho việc tạo ra k tập validation Tìm độ lỗi và tham số model tương ứng với từng thuộc tính Tìm thuộc tính ảnh hưởng nhất (độ lỗi của model và chỉ số của thuộc tính ảnh hưởng nhất)
<pre> features = ["volatile acidity","chlorides","residual sugar","fixed acidity","density","ph","alcohol","quality"] df_feature = pd.DataFrame(df, columns=features) X_data2 = df_feature.iloc[:,1:].values y_data2 = df_feature.iloc[:,1].values bias2 = np.ones(X_data2.shape[0]) bias2 = np.resize(bias2, (1, X_data2.shape[0])) X_data2 = np.concatenate((X_data2, bias2.T), axis=1) X_train2, y_train2, X_test2, y_test2 = shuffle_split_data(X_data2, y_data2) print("Số lượng dữ liệu là:") print(len(X_train2),len(y_train2)) print(len(X_test2),len(y_test2)) print(X_train2[0]) </pre>	<p>Lựa chọn các đặc trưng sau đây làm mô hình. Dựa vào mô hình trên ta chọn 6 đặc trưng có độ lỗi thấp nhất chạy thử.</p> <p>Tự thay thế thí nghiệm lựa chọn cho cái nào phù hợp. Thêm bớt thì chỗ features thì tùy.</p>


```
# Huấn luyện mô hình
theta2 = linear_regression(X_train2,y_train2)
print(theta2)
# Dự đoán trên tập test
y_pred2 = []
for id in range(len(y_test2)):
    y_pred2.append(predict(theta2,X_test2[id]))
y_test2 = [item for item in y_test2]

mse2 = calculate_score(y_test2,y_pred2)
print("error: ",mse2)
```

Huấn luyện mô hình và
tiếp tục dự đoán trên tập
test

KẾT QUẢ - HÌNH ẢNH TƯƠNG ỨNG VỚI TỪNG CHỨC NĂNG

Tên chức năng/công việc	Hình ảnh minh họa kết quả
Xây dựng mô hình dự đoán chất lượng rượu bằng 11 thuộc tính	 <pre>[14] print("Độ lỗi : ", error) Độ lỗi : 0.488231491706842</pre>
Tìm ra thuộc tính nào ảnh hưởng đến chất lượng rượu nhất	 <pre>list_output.append(sum_mae/n_split) print("-----") print(list_output) print("Đặc trưng có giá trị tốt nhất là : ", list_name[list_output.index(min(list_output))]) print("Độ đo thấp nhất là : ", min(list_output)) Đặc trưng có giá trị tốt nhất là : alcohol Độ đo thấp nhất là : 0.5476279559423002</pre>

<p>Tự xây dựng mô hình dự đoán của riêng bạn cho kết quả tốt nhất</p>	<div data-bbox="357 193 1502 331">  <pre data-bbox="446 210 1315 304">[-1.07225035e+00 -5.84361319e-01 6.77647278e-03 3.60181215e-02 -8.07829007e+00 -3.06464242e-01 3.53262903e-01 1.13461380e+01] error: 0.5583622074409094</pre> </div>
---	---

REFERENCES

- [1]: <https://machinelearningcoban.com/2016/12/28/linearregression/>
- [2]: <https://dominhhai.github.io/vi/2017/12/ml-linear-regression/#3-1-gi%E1%BB%AF-nguy%C3%AAn-%C4%91%E1%BA%A7u-v%C3%A0o>
- [3]: https://www.youtube.com/playlist?list=PLLssT5z_DsK-h9vYZkQkYNWcItqhlRJLN&app=desktop