MULTIPLE LINEAR REGRESSION
HỒI QUI TUYẾN TÍNH ĐA BIẾN

1. TS. Nguyễn Tấn Trần Minh Khang
2. ThS. Võ Duy Nguyên
3. Cao học. Nguyễn Hoàn Mỹ
4. Tình nguyện viên. Lê Ngọc Huy
5. Tình nguyện viên. Cao Bá Kiệt

DATASET

Dataset

- Tên tập dữ liệu: 50 Startups.
- Nguồn: https://www.superdatascience.com/pages/machine-learning.

Dataset

- Tập dữ liệu gồm 50 điểm dữ liệu, mỗi điểm dữ liệu gồm 5 thuộc tính:
  + R&D Spend: Số tiền chi trả cho nghiên cứu và phát triển.
  + Administrator: Số tiền chi trả cho quản trị và điều hành.
  + Marketing Spend: Số tiền chi trả cho quảng cáo.
  + State: Là một chuỗi ký tự, đại diện cho bang mà công ty khởi nghiệp.
  + Profit: Là một số thực dương, đại diện cho lợi nhuận thu được của startup.

Dataset

- Bài toán đặt ra là cho những dữ kiện về:
  - + Số tiền chi trả cho nghiên cứu và phát triển.
  - + Số tiền chi trả cho quản trị và điều hành.
  - + Số tiền dành cho quảng cáo.
  - + Vị trí (bang - state) của startup.

➢ Ta cần dự đoán lợi nhuận (profit) mà startup đó thu được.

Dataset

STT	R&D Spend	Administator	Marketing Spend	State	Profit
1	165,349.2	136,897.8	471,784.1	New York	192,261.83
2	162,597.7	151,377.59	443,898.53	California	191,792.06
3	153,441.51	101,145.55	407,934.54	Florida	191,050.39
4	144,372.41	118,671.85	383,199.62	New York	182,901.99
5	142,107.34	91,391.77	366,168.42	Florida	166,187.94
6	131,876.9	99,814.71	362,861.36	New York	156,991.12

MULTIPLE LINEAR REGRESSION

Multiple Linear Regression

– Mô hình hồi quy tuyến tính đa biến Multiple Linear Regression:

$$y = w_0 + w_1 \times x_1 + w_2 \times x_2 + \cdots + w_n \times x_n$$

– Trong đó:

+ y là biến phụ thuộc (dependent variable), trong bài toán của chúng ta, đó là giá trị lợi nhuận (profit).
+ $x_1, x_2, \ldots x_n$ là các biến độc lập (indenpendent variable), trong bài toán của chúng ta, đó là chi phí nghiên cứu phát triển, chi phí quản trị, chi phí quảng cáo và thông tin tiểu bang startup.
+ $w_0, w_1, \ldots, w_n$ là những tham số mô hình.

TIỀN XỬ LÝ DỮ LIỆU

Tiền xử lý dữ liệu

– Ban đầu, đọc toàn bộ dữ liệu và phân chia các giá trị đầu vào – ký hiệu là X, và đầu ra – ký hiệu là Y.

```
1. import pandas as pd
2. dataset = pd.read_csv("50_Startups.csv")
3. X = dataset.iloc[:, 0:4].values
4. Y = dataset.iloc[:, -1].values
```

Tiền xử lý dữ liệu

- Vì thuộc tính "State" ở dạng chuỗi ký tự, ta chưa thể đưa vào công thức Multiple Linear Regression, ta cần phải chuyển nó về dạng số.
- Lớp `LabelEncoder` ở module `sklearn.preprocessing` đã được xây dựng sẵn cho việc đánh số cho các nhãn có dạng "không ở dạng số".

```
5. from sklearn.preprocessing import LabelEncoder
6. le = LabelEncoder()
7. le.fit(X[:, 3])
8. X[:, 3] = le.transform(X[:, 3])
```

Tiền xử lý dữ liệu

State
New York
California
Florida
California
New York

Bảng Ánh Xạ	
New York	0
Califonia	1
Florida	2

State
New York
California
Florida
California
New York

State
0
1
2
1
0

Tiền xử lý dữ liệu

- Vì "State" là loại thuộc tính không có tính thứ tự, tức giá trị của các loại "State" khác nhau có tầm quan trọng như nhau.
- Do đó, dùng các chữ số 0,1,2,... để gán nhãn cho thuộc tính "State" là không hợp lý.

Tiền xử lý dữ liệu

- Khái niệm one-hot vector:
  - + One-hot vector là vector chỉ có đúng một phần tử có giá trị là 1, các phần tử còn lại có giá trị là 0.
  - + Ví dụ: $[0,0,1], [1,0,0]$.
- One-hot vector thường được sử dụng để gán nhãn các dữ liệu dạng danh mục không thứ tự (non-order categorial data).

Tiền xử lý dữ liệu

- Chuyển các giá trị của thuộc tính "State" (đang ở dạng số nguyên) thành dạng one-hot vector.
- Lớp `OneHotEncoder` của module `sklearn.preprocessing` đã được xây dựng sẵn cho việc chuyển dữ liệu dạng số nguyên sang dạng one-hot vector.

```
9. from sklearn.preprocessing import OneHotEncoder
10.ohe = OneHotEncoder(categorical_features= [3])
11.X = ohe.fit_transform(X).toarray()
```

Tiền xử lý dữ liệu

State
New York
California
Florida
California
New York

→

State
0
1
2
1
0

→

State		
1	0	0
0	1	0
0	0	1
0	1	0
1	0	0

Tiền xử lý dữ liệu

- Phân chia tập dữ liệu hiện tại thành hai tập con, một tập là dữ liệu training, tập còn lại là dữ liệu test.
- Tỉ lệ phân chia là 80% dữ liệu cho tập training.
- Hàm `train_test_split` được xây dựng sẵn trong module `sklearn.model_selection` để phân chia tập dữ liệu.

```
12.from sklearn.model_selection import train_test_split
13.X_train, X_test, Y_train, Y_test =
    train_test_split(X, Y, train_size = 0.8, random_state = 0)
```

HUẤN LUYỆN DỮ LIỆU

Huấn luyện dữ liệu

– Để huấn luyện mô hình, ta sử dụng lớp `LinearRegression` trong module `sklearn.linear_model`.

```
14.from sklearn.linear_model import LinearRegression
15.lin_reg = LinearRegression()
16.lin_reg.fit(X_train, Y_train)
```

KIỂM TRA KẾT QUẢ MÔ HÌNH

Kiểm tra kết quả mô hình

- Ở tập dữ liệu này, ta rất khó để trực quan hóa kết quả của mô hình.
  - + Tập dữ liệu có 5 chiều.
  - + Có bốn chiều là dữ liệu dạng số.
  - + Một chiều là dữ liệu dạng liệt kê.
- Nên ta cần một cách đánh giá mới.

Kiểm tra kết quả mô hình

– Khái niệm hệ số đánh giá R^2:

+ R^2 là hệ số để đánh giá chất lượng của một mô hình hồi quy.

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - y_{mean})^2}$$

+ Trong đó:

- y_i là giá trị outcome thực sự của một điểm dữ liệu.
- $\hat{y}_i$ là giá trị outcome mô hình dự đoán của một điểm dữ liệu.
- y_{mean} là giá trị outcome trung bình trên tập dữ liệu huấn luyện.

Kiểm tra kết quả mô hình

- Giá trị của hệ số R^2 luôn nằm trong đoạn $(-\infty, 1]$:
  - + Nếu $R^2 < 0$: Mô hình tệ hơn mô hình cơ sở.
  - + Nếu $R^2 = 0$: Mô hình giống như mô hình cơ sở (vẫn rất tệ).
  - + Nếu $R^2 = 1$: Mô hình chính xác tuyệt đối.
- R^2 càng lớn thì độ chính xác của mô hình càng cao.
- Một mô hình được xem là tốt nếu $R^2 > 0.8$.

Kiểm tra kết quả mô hình

– Ta sẽ sử dụng phương thức `score` trong lớp `LinearRegression` để đánh giá mô hình hiện tại bằng hệ số đánh giá R^2.

```
17.lin_reg.score(X_train, Y_train)
>>> 0.950184762749360 7
18.lin_reg.score(X_test, Y_test)
>>> 0.934706847328230 3
```

Kiểm tra kết quả mô hình

– Xây dựng hàm:

+ Dự đoán kết quả của một điểm dữ liệu.

+ In ra màn hình và so sánh với đầu ra thực.

```
19.def compare(i_example):
20.    x = X_test[i_example : i_example + 1]
21.    y = Y_test[i_example]
22.    y_pred = lin_reg.predict(x)[0]
23.    label = ohe.inverse_transform(x[:, 0:3]).astype(int)
24.    state = le.inverse_transform(label)
25.    print(x[:, 3:6], state, y, y_pred)
```

Kiểm tra kết quả mô hình

– Gọi thực hiện hàm compare cho tất cả các điểm dữ liệu trong tập test.

```
26.for i in range(len(X_test)):
27.    compare(i)
```

Kiểm tra kết quả mô hình

STT	R&D Spending	Administrator	Marketing Spending	State	Profit	Predicted Profit
1	28,754	118,546	172,795	New York	78,239	79,786
2	27,892	84,710	164,470	California	77,798	78,474
3	23,640	96,189	148,001	Florida	71,498	76,185
4	15,505	127,382	35,534	New York	69,758	61,898
5	22,177	154,806	28,334	Florida	65,200	68,900

Kiểm tra kết quả mô hình

STT	R&D Spending	Administrator	Marketing Spending	State	Profit	Predicted Profit
6	1,000	124,153	1,903	New York	64,926	49,996
7	1,315	115,816	297,114	California	49,490	60,628
8	0	135,426	0	Florida	42,559	52,036
9	542	51,743	0	New York	35,673	52,906
10	0	116,983	45,173	California	14,681	54,195

Chúc các bạn học tốt
Thân ái chào tạm biệt các bạn

ĐẠI HỌC QUỐC GIA TP.HCM
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN TP.HCM
TOÀN DIỆN – SÁNG TẠO – PHỤNG SỰ