Báo cáo môn học

HỌC MÁY

Đề tài: Sử dụng Multiple Linear Regression cho đề tài Real Estate

Nhóm 8 : Nguyễn Đam Trường

Nguyễn Hoàng Vũ

Mai Tất Thắng

Nguyễn Hữu Thông

Nguyễn Đức Thành

Mục lục:

I. Linear Regression

- **Liner Regression** (hồi quy tuyến tính) đây là một thuật toán **Supervised learning**. Hay bài toán này còn có môt tên gọi khác là **Linear Fitting**(trong thống kê) hoặc **Linear Least Square**.
- **Regression** (Hồi quy) là một phương pháp thống kê để thiết lập mối quan hệ giữa một biến phụ thuộc và một nhóm tập hợp các biến độc lập. Ví dụ:

```
Tuổi = 5 + Chiều cao * 10 + Trọng lượng * 13
```

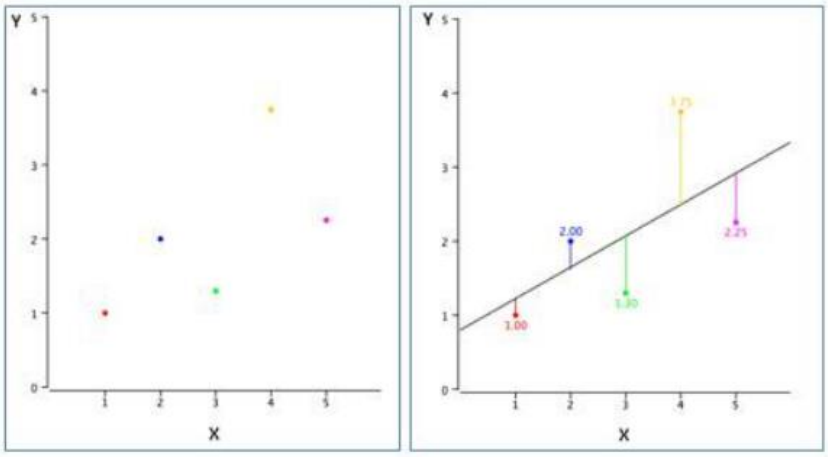
Ở đây chúng ta đang thiết lập mối quan hệ giữa Chiều cao và Trọng lượng của một người với Độ tuổi của người đó.

- **Linear Regression** (Hồi quy tuyến tính) là một phương pháp thống kê để hồi quy dữ liệu với biến phụ thuộc có giá trị liên tục trong các biến độc lập có thể có một trong hai giá trị liên tục hoặc là giá trị phân loại .

Được sử dụng cho các trường hợp chúng ta muốn dự đoán một số lượng liên tục.

- **Đường hồi quy tuyến tính** là đường thẳng có thể tạo ra sự phân bố gần nhất với hầu hết các điểm. Do đó làm giảm khoảng cách (sai số) của các điểm dữ liệu cho đến đường đó.

```
Y = B0 + B1*X
```



Đường hồi quy tuyến tính

II. Multiple Linear Regression

Chúng ta đã biết về kịch bản khi mà chúng ta chỉ có một biến độc lập. Trên thực tế để giải một bài toán cần rất nhiều biến để giải một mối liên hệ nào đó vì vậy khi chúng ta có nhiều hơn một biến độc lập, thì phương pháp phù hợp nhất là **Multiple Linear Regression** – Hồi quy tuyến tính đa biến

- **Sự khác biệt**: Về cơ bản không có sự khác biệt giữa hồi quy tuyến tính giản đơn và đa biến. Cả hai đều tuân thủ Nguyên tắc **OLS** và thuật toán để có được đường hồi quy tối ưu nhất cũng tương tự.
- **Đường hồi quy tuyến tính** trong trường hợp này có dạng :

```
Y=B0+B1*X1+B2*X2+B3*X3.....
```

III. Áp dụng Multiple Linear Regression cho bài toán Real Estate

Bài toán này chúng ta sẽ sử dụng sckit-learn để thực hiện hồi quy tuyến tính.

- **Scikit-learn** là một module Python mạnh mẽ cho việc học máy. Nó chứa hàm cho hồi quy, phân loại, phân cụm, lựa chọn mô hình và giảm kích chiều.
- Bộ Dữ liệu Nhà ở Boston bao gồm giá nhà ở những nơi khác nhau ở Boston.

Tập dữ liệu cung cấp các thông tin như Tội phạm (CRIM), Các khu vực kinh doanh không bán lẻ ở thị trấn (INDUS), tuổi chủ sở hữu ngôi nhà (AGE) và có nhiều thuộc tính khác nữa.

Các bước thiết lập:

1. **Đầu tiên : Khai báo mảng**

   Đọc dữ liệu nhờ sử dụng thư viện **Pandas**

[1]:
```
import pandas as pd
data_path = 'data.csv'
df = pd.read_csv(data_path)
df.fillna(df.mean(), inplace=True)
df.head(10)
```

[1]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90	5.33	36.2
5	0.02985	0.0	2.18	0	0.458	6.430	58.7	6.0622	3	222	18.7	394.12	5.21	28.7
6	0.08829	12.5	7.87	0	0.524	6.012	66.6	5.5605	5	311	15.2	395.60	12.43	22.9
7	0.14455	12.5	7.87	0	0.524	6.172	96.1	5.9505	5	311	15.2	396.90	19.15	27.1
8	0.21124	12.5	7.87	0	0.524	5.631	100.0	6.0821	5	311	15.2	386.63	29.93	16.5
9	0.17004	12.5	7.87	0	0.524	6.004	85.9	6.5921	5	311	15.2	386.71	17.10	18.9

2. **Bước 2 : Gán giá trị**

   Ta thấy rằng “MEDV: Median value of owner-occupied homes in $1000 “ nên ta sẽ tách bảng tính

   X là giá trị dự đoán của các căn nhà

   Y là giá thành của các căn nhà ở Boston

[2]:
```
import numpy as np
Y = np.array(df['MEDV'], dtype=np.float32)
print(Y.shape, Y.dtype)
X = df.drop(['MEDV'], axis=1)
X = np.array(X, dtype=np.float32)
print(X.shape, X.dtype)
```

```
(511,) float32
(511, 13) float32
```

3. **Bước 3: Tách dữ liệu để train-test**

   Chúng ta cần chia dữ liệu thành tệp dữ liệu để train-test, bây giờ chúng ta có thể split dữ liệu để train và test với snippet như sau.

```
[3]: from sklearn.model_selection import train_test_split
     x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size=0.3, random_state=42)
     print(x_train.shape)
     print(x_test.shape)
     print(y_train.shape)
     print(y_test.shape)
```

   Nếu chúng ta kiểm tra shape của mỗi biến, chúng ta đã có được bộ dữ liệu với tập dữ liệu thử nghiệm có tỷ lệ 70% đối với dữ liệu train và 30% đối với dữ liệu test.

```
(357, 13)
(154, 13)
(357,)
(154,)
```

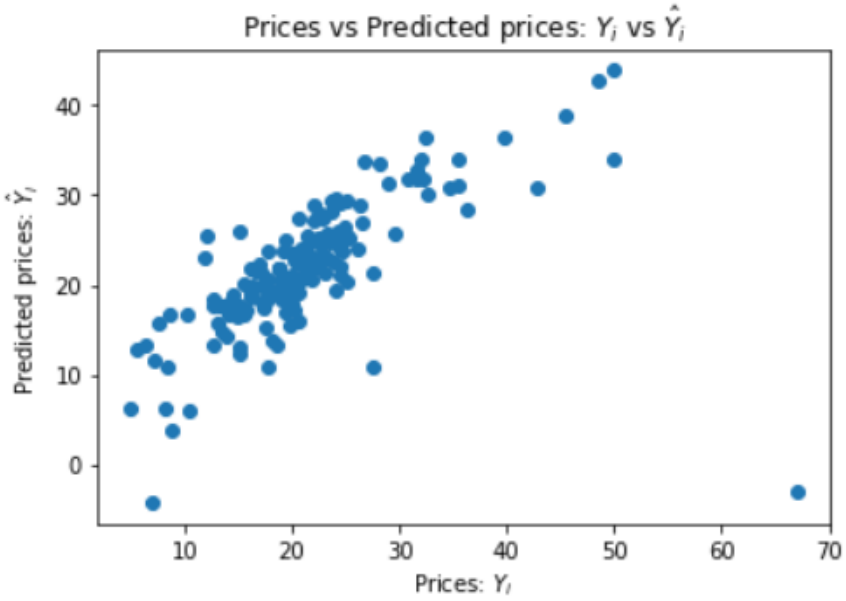
4. **Bước 4: Chạy Linear Regression**

```
[4]: import matplotlib.pyplot as plt
     from sklearn.linear_model import LinearRegression
     model = LinearRegression().fit(x_train, y_train)
     y_pred = model.predict(x_test)

     plt.scatter(y_test, y_pred)
     plt.xlabel("Prices: $Y_i$")
     plt.ylabel("Predicted prices: $\hat{Y}_i$")
     plt.title("Prices vs Predicted prices: $Y_i$ vs $\hat{Y}_i$")
```

   Bây giờ chúng ta đã có mô hình Hồi quy tuyến tính, tiếp theo chúng ta sẽ dự đoán cho x_test và các giá trị dự đoán sẽ được lưu trong y_pred
   Để hình dung em đã tạo một kiểu bảng:

```
[4]: Text(0.5, 1.0, 'Prices vs Predicted prices: $Y_i$ vs $\\hat{Y}_i$')
```



Thực tế thì đáng lẽ đồ thị ở trên phải tạo một đường tuyến tính như chúng ta đã thảo luận lý thuyết ở trên. Tuy nhiên, model không thích hợp 100%, cho nên nó đã ko thể tạo được đường tuyến tính.

Trung bình diện tích sai số

Để kiểm tra mức độ lỗi của một mô hình, chúng ta có thể sử dụng Mean Squared Error. Đây là một trong các phương pháp để đo trung bình của ô vuông của sai số. Về cơ bản, nó sẽ kiểm tra sự khác biệt giữa giá trị thực tế và giá trị dự đoán.

Ta có câu lệnh sau:

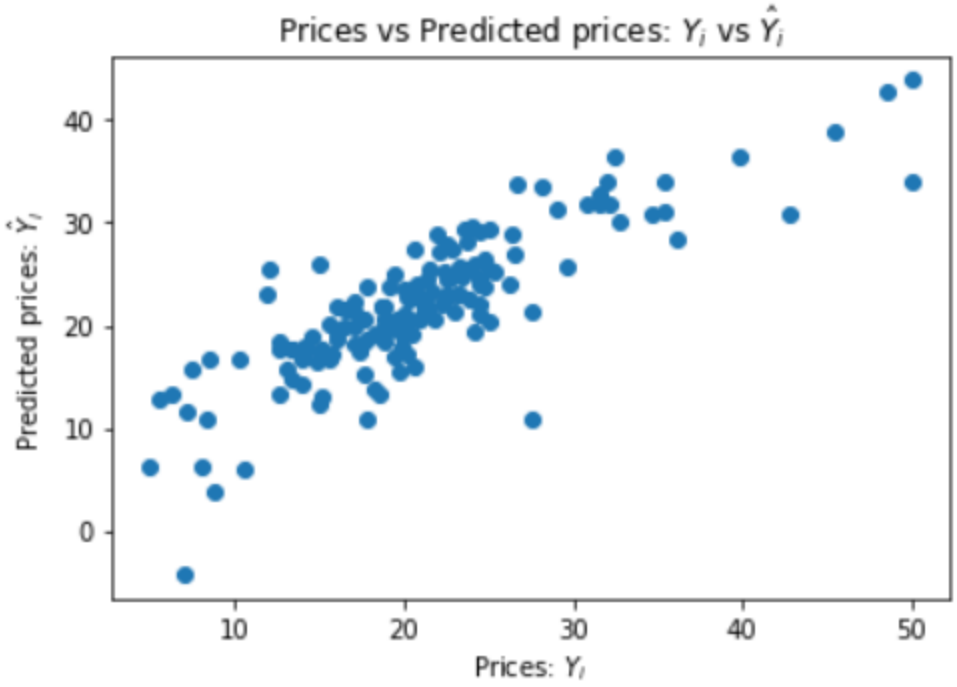
```
mse = sklearn.metrics.mean_squared_error(y_test, y_pred)
print(mse)
```

kết quả nhận được

```
52.301792
```

5. **Xử lý nhiễu**

Bằng cách loại bỏ điểm nhiễu ta có một bảng biểu diễn mới



Sai số dự đoán mới

```
mse = sklearn.metrics.mean_squared_error(y_test, y_pred)
print(mse)
```

```
20.443426
```

IV. Tổng kết

Ứng dụng của Linear Regression trong ngành bán lẻ hay kinh doanh là rất nhiều tùy vào từng trường hợp khác nhau, những mục tiêu nghiên cứu khác nhau của các nhà phân tích, các thuộc tính dữ liệu khác nhau, tổng quan Linear Regression thường có ứng dụng như :

+ Khai phá thông tin hữu ích, giá trị của đối tượng nghiên cứu
+ Đưa ra dự báo trong tương lai
+ Tối ưu hóa quá trình vận hành, hoạt động
+ Hỗ trợ ra quyết định , chiến lược.

Các hạn chế của Linear Regression

+ **Rất nhạy cảm với nhiễu** (sensitive to noise). Như các bạn đã thấy khi có nhiễu thì **Sai số dự đoán** lệch rất cao. Vì vậy trước khi thực hiện Linear Regression, các nhiễu (outlier) cần phải loại bỏ.
+ Thứ hai, đấy chính là Linear Regression **không biểu diễn được các mô hình phức tạp**