Hồi quy tuyến tính

Mục tiêu

1. Mục tiêu của mô hình hồi quy tuyến tính
2. Ý tưởng hồi quy tuyến tính
3. Hồi quy tuyến tính đơn biến
4. Hồi quy tuyến tính đa biến
5. Ưu điểm và hạn chế của hồi quy tuyến tính
6. Thực hành với Python

Hồi quy tuyến tính (Linear Regression)

Hồi quy tuyến tính là phương pháp học máy có giám sát đơn giản, được sử dụng để dự đoán giá trị biến mục tiêu liên tục Y dựa trên phương trình tuyến tính của biến dự báo X.

Hồi quy tuyến tính (Linear Regression)

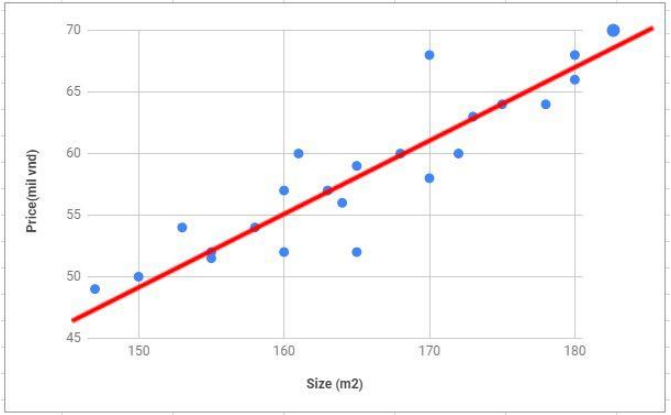
Ví dụ:

- Dự đoán giá nhà dựa vào các thông số về diện tích, số phòng ngủ và khoảng cách tới trung tâm,..
- Dự đoán mức lương sau khi ra trường dựa vào điểm trung bình khóa học, giới tính, các hoạt động ngoại khoá đã tham gia, ...
- Dự đoán giá chứng khoán ngày mai dựa vào lịch sử giá trước đó, các sự kiện xã hội, số lượng vốn đầu kỳ, ...

Hồi quy tuyến tính một biến

- Biến mục tiêu y và biến dự báo x_1 có mối quan hệ tuyến như sau:

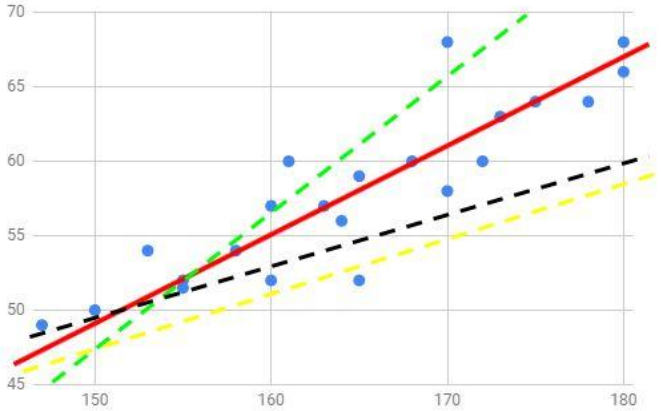
$$\hat{y}_i = f(x_i) = w_0 + w_1 * x_1$$



- w_0, w_1 là các hằng số chưa biết => Ta ước tính giá trị của chúng từ dữ liệu đầu vào
- y là giá trị thực của outcome (dựa trên số liệu thống kê chúng ta có trong tập training data), ŷ là giá trị mà mô hình dự đoán được.

Hồi quy tuyến tính một biến

- Làm thế nào để chúng ta ước lượng các hệ số (“fit the model”)?
- Đánh giá độ phù hợp của mô hình từ dữ liệu quan sát được?



Hồi quy tuyến tính một biến

- Sai số dự đoán: mong muốn rằng sự sai khác giữa giá trị thực *y* và giá trị dự đoán ŷ là nhỏ nhất
- => Tổng bình phương lỗi được tính bằng công thức:

$$\mathcal{L}(\mathbf{w}|\mathbf{X}) = \frac{1}{2n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \frac{1}{2n}\sum_{i=1}^{n}(y_i - w_0 - w_1 * x_i)^2$$

$L(w|X)$: hàm mất mát (loss function)

- Các giá trị tham số w_0, w_1 ước lượng bằng cách cực tiểu hóa tổng bình phương lỗi

Hồi quy tuyến tính một biến

- Đạo hàm Loss function:

$$\frac{\delta \mathcal{L}(\mathbf{w}|\mathbf{X})}{\delta w_0} = \frac{-1}{n}\sum_{i=1}^{n}(y_i - w_0 - w_1 * x_i) = 0$$

$$\frac{\delta \mathcal{L}(\mathbf{w}|\mathbf{X})}{\delta w_1} = \frac{-1}{n}\sum_{i=1}^{n} x_i(y_i - w_0 - w_1 * x_i) = 0$$

- Nghiệm w:

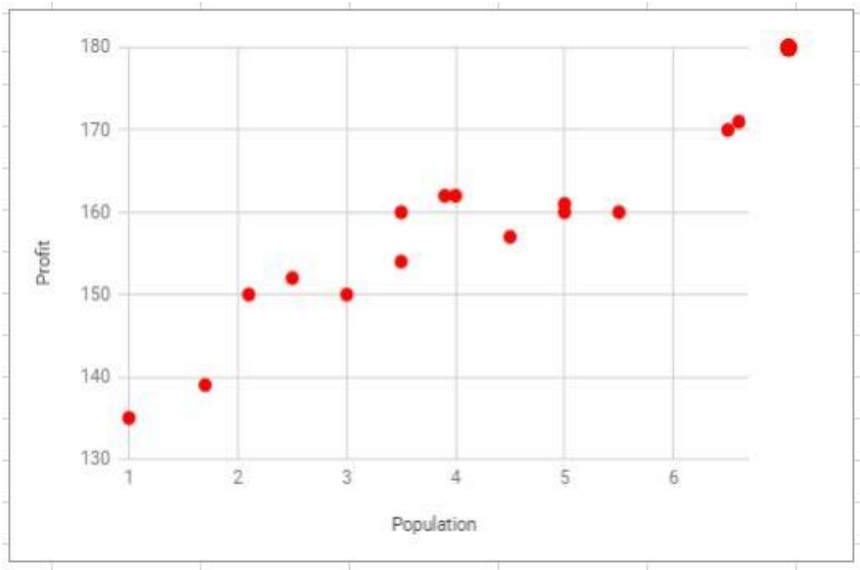
$$w_0 = \bar{\mathbf{y}} - w_1\bar{\mathbf{x}}$$

$$w_1 = \frac{\bar{\mathbf{x}}\bar{\mathbf{y}} - \overline{\mathbf{x}\mathbf{y}}}{\bar{\mathbf{x}}^2 - \overline{\mathbf{x}^2}}$$

Hồi quy tuyến tính một biến

- Ví dụ: Cho trước tập dữ liệu gồm thông tin số dân và lợi nhuận thu được khi mở hàng ăn ở 15 thành phố lớn, phân bố như hình sau:

Hãy dự đoán lợi nhuận của cửa hàng ăn nào đó, nếu biết số dân của thành phố tại cửa hàng ăn đó?



Hồi quy tuyến tính một biến

Population	Profit	Population	Profit
1.7	139	4	162
2.1	150	5	160
3.5	160	1	135
3.9	162	6.6	171
5	161	4.5	157
6.5	170	5.5	160
2.5	152	3	150
3.5	154		

Hồi quy tuyến tính một biến

```
# To support both python 2 and python 3
from __future__ import division, print_function, unicode_literals
import numpy as np
from sklearn import linear_model
import matplotlib.pyplot as plt

# # Population
X = np.array([[1.7, 2.1, 3.5, 3.9, 5, 6.5, 4, 5, 1, 6.6, 4.5, 5.5, 3, 2.5, 3.5]]).T
print(X)
# # Profit
y = np.array([[ 139, 150, 160, 162, 161, 170, 162, 160, 135, 171, 157, 160, 150, 152, 154]]).T

# Visualize data
plt.plot(X, y, 'o')
plt.show()

# fit the model by Linear Regression
model = linear_model.LinearRegression()
model.fit(X, y)

# Plot the data and the model prediction
X_fit = np.linspace(1, 7, 2)[:, np.newaxis]
y_fit = model.predict(X_fit)

plt.plot(X, y, 'o')
plt.plot(X_fit, y_fit)
plt.show()
```

Linear Regression with multiple variables

- Hồi quy tuyến tính đa biến: mô hình có số biến nhiều hơn 1 dùng để dự đoán biến đích (output)

$$y = w_0 + w_1x_1 + w_2x_2 + ...+ w_nx_n$$

- Loss function:

$$\mathcal{L}(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{N}(y_i - \bar{\mathbf{x}}_i\mathbf{w})^2$$

$$= \frac{1}{2}\|\mathbf{y} - \bar{\mathbf{X}}\mathbf{w}\|_2^2$$

Linear Regression with multiple variables

- Đạo hàm loss function:

$$\frac{\partial \mathcal{L}(\mathbf{w})}{\partial \mathbf{w}} = \bar{\mathbf{X}}^T(\bar{\mathbf{X}}\mathbf{w} - \mathbf{y})$$

- Nghiệm w:

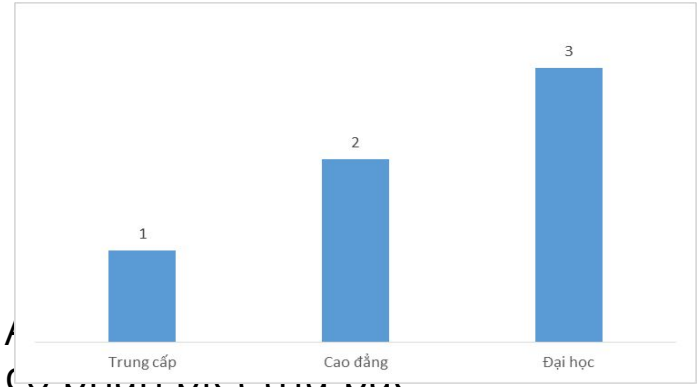
$$\mathbf{w} = \mathbf{A}^\dagger \mathbf{b} = (\bar{\mathbf{X}}^T\bar{\mathbf{X}})^\dagger \bar{\mathbf{X}}^T\mathbf{y}$$

Linear Regression with multiple variables

Kích thước (m2)	Số phòng ngủ	Số tầng	Giá ($1000)
2100	5	1	460
1416	3	2	232
1534	3	2	315
852	2	1	178
1600	3	2	329
1985	5	1	420
1535	4	2	330
1050	2	1	195
2300	4	2	450
1200	3	2	250

Chuẩn hóa dữ liệu

- Tạo biến thức bậc:



Áp dụng đối với các biến có phân biệt thứ bậc

Onehot coding:

Male	Female
1	0
0	1
1	0

Áp dụng đối với các biến không có tính thứ bậc.

Chuẩn hóa dữ liệu

- Một căn nhà có giá 2100$, số phòng ngủ 5, số tầng 1. Giá nhà được dự báo:

$$w_0 + w_1 \times 2100 + w_2 \times 5 + w_3 \times 1$$

=> Khác biệt lớn trong range giữa các biến có thể làm giảm độ chính xác của mô hình trong một số trường hợp.

- Giải pháp: chuẩn hóa dữ liệu về cùng range.

Min max scale

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Standard scale

$$x' = \frac{x - \bar{x}}{\sigma}$$

Unit length scale

$$\mathbf{x}' = \frac{\mathbf{x}}{\|\mathbf{x}\|_2}$$

- Apply normalization trong linear regression.

```
LinearRegression(fit_intercept = True, normalize = True)
```

Ưu / nhược điểm

- Ưu điểm
  - Mô hình đơn giản, dễ hiểu
  - Dự báo được các biến liên tục
  - Lời giải cho nghiệm tối ưu đơn giản
  - Dễ diễn giải mô hình thông qua hệ số hồi qui
- Nhược điểm
  - Mô hình đơn giản nên không linh hoạt nếu biểu diễn được các quan hệ dữ liệu phức tạp.
  - Rất nhạy cảm với dữ liệu ngoại lai (nhiễu)

Tóm tắt

Qua bài học này, chúng ta đã tìm hiểu những kiến thức sau:

1. Mục tiêu của mô hình hồi quy tuyến tính
2. Ý tưởng hồi quy tuyến tính
3. Hồi quy tuyến tính đơn biến
4. Hồi quy tuyến tính đa biến
5. Ưu điểm và hạn chế của hồi quy tuyến tính
6. Thực hành với Python

Thank you!