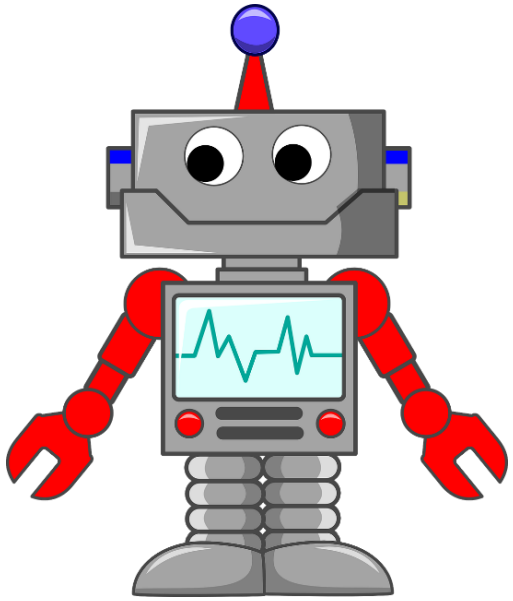
ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

CS116 – LẬP TRÌNH PYTHON CHO MÁY HỌC

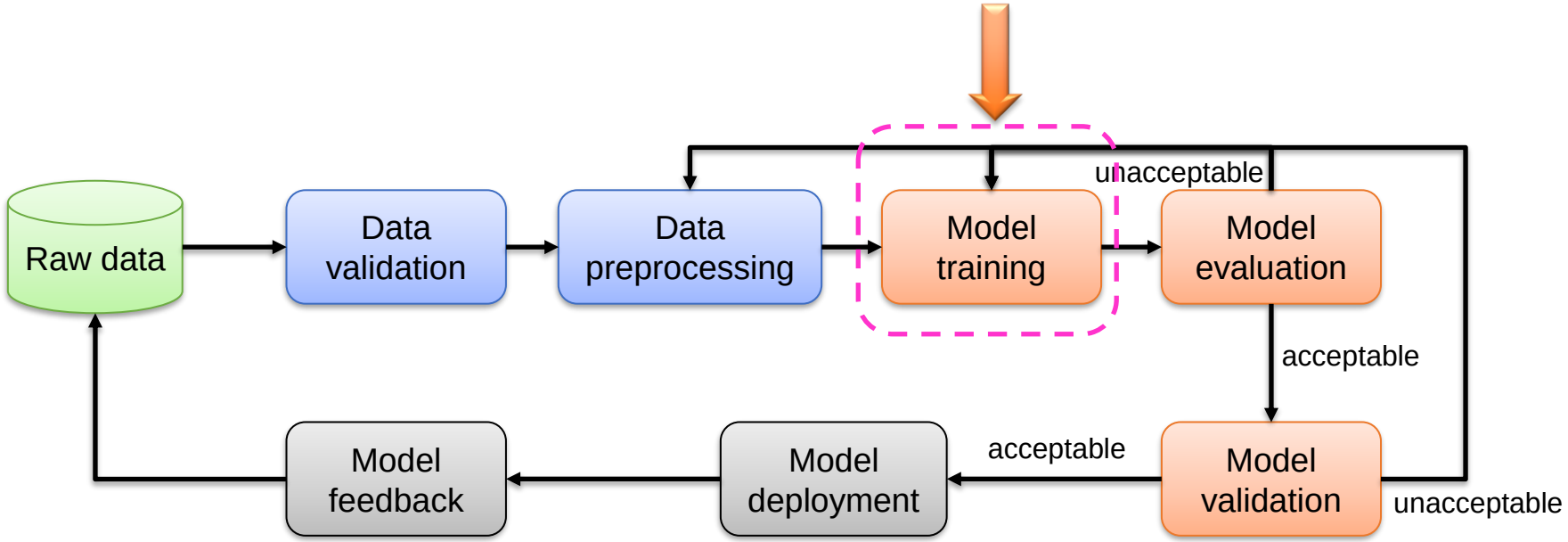
BÀI 07

HỌC CÓ GIÁM SÁT – MÔ HÌNH HỒI QUY (REGRESSION)



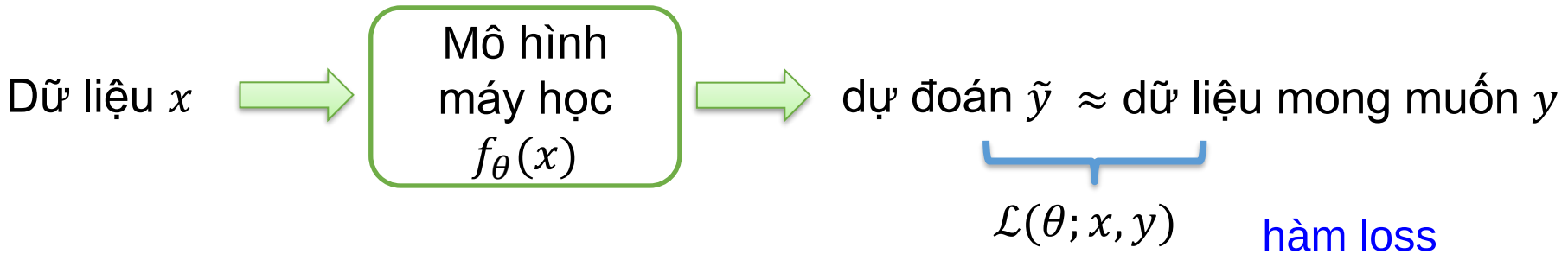
TS. Nguyễn Vinh Tiệp

Vị trí của bài học



Học có giám sát

- **Học có giám sát (supervised learning)**: là một nhánh của máy học, nhằm dự đoán giá trị đầu ra từ một đặc trưng đầu vào dựa trên các dữ liệu huấn luyện trước đó
- Dữ liệu huấn luyện bao gồm cặp đặc trưng đầu vào và giá trị đầu ra mong muốn (x, y)



- Có hai loại bài toán chính: hồi quy và phân lớp

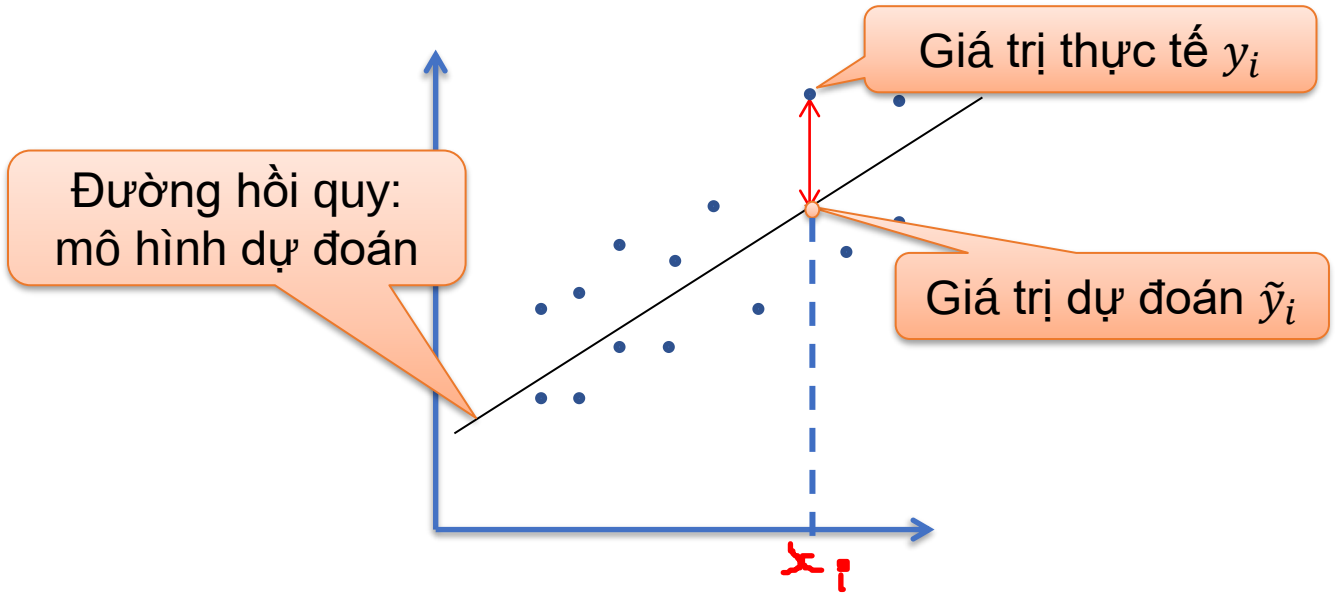
tìm theta với x,y đầu vào sao cho L min

NỘI DUNG

Mô hình hồi quy tuyến tính – Linear Regression

- Mô hình thực tế: $y = \varepsilon + \beta x$
- Hàm mô hình dự đoán: $\hat{y} = \hat{\beta} x$
- Hàm độ lỗi: $\mathrm{MSE} = \mathrm{L}(\hat{\beta}) = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} (y_i - \hat{\beta} x_i)^2 = \|Y - \hat{\beta} X\|^2$

tổng các sai số bình phương



Mô hình hồi quy tuyến tính – Linear Regression

- Huấn luyện mô hình:
  - Bằng normal equation: tham số ước lượng: $\hat{\beta} = (X^T X)^{-1}(X^T Y)$
  - Bằng thuật toán gradient descent với công thức cập nhật:

$\hat{\beta}$ = random() khi init ban đầu

$$\hat{\beta} = \hat{\beta} - \alpha \nabla_{\hat{\beta}} L$$

thì dừng

lớp la(đạo hàm) của Loss theo $\hat{\beta}$

- Trong sklearn đã cài đặt module: **LinearRegression**

hàm fit để huấn luyện
predict để test với input vào

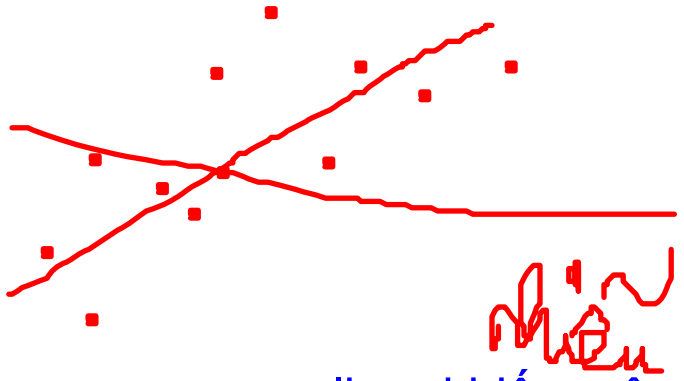
Mô hình hồi quy tuyến tính – Linear Regression

- **Ưu điểm:**
  - Đơn giản, dễ hiểu
  - Phù hợp với dữ liệu có mối quan hệ tuyến tính (đồng biến, nghịch biến)

x tăng thì y tăng theo, hoặc x tăng y giảm => tuyến tính

- **Khuyết điểm:**
  - Không hiệu quả khi dữ liệu có mối quan hệ phức tạp
  - Dễ bị ảnh hưởng khi có dữ liệu nhiễu

phi tuyến



outliner khiến mô hình lệnh nhiều

NỘI DUNG

1. MÔ HÌNH HỒI QUY TUYẾN TÍNH

2. BIAS VÀ VARIANCE liên quan đến overfitting và under fitting

3. MÔ HÌNH LASSO, RIDGE VÀ ELASTIC NET

4. MỘT SỐ MÔ HÌNH PHI TUYẾN

Khái niệm Bias

- **Bias**: là sai số giữa trung bình (kỳ vọng) mô hình dự đoán với mô hình thực tế:

E là trung bình của mô hình dự đoán

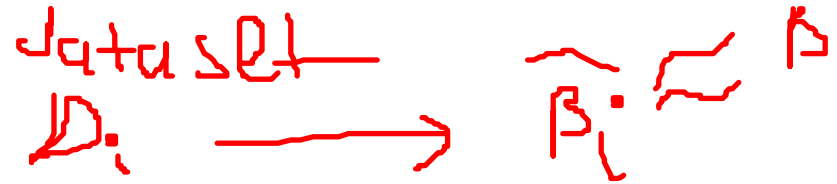
$$\text{bias}(\hat{\beta}) = E(\hat{\beta}) - \beta$$

- **Bias thấp**: thể hiện mô hình học được mối quan hệ của dữ liệu huấn luyện
- **Bias cao:** thể hiện mô hình không học được quan hệ của dữ liệu huấn luyện

Bias ≠ noise
Noise: lúc cao lúc thấp
Bias: lệch một chiều

Ví dụ: cái cân luôn dư +2kg → bias dương

Khái niệm Variance



sự thay đổi của dataset cũng làm thay đổi Bi dự đoán khiến nó tăng hoặc giảm không ổn định

- **Variance**: là sai số trung bình của mô hình dự đoán so với kỳ vọng (trung bình) mô hình được huấn luyện trên toàn bộ dữ liệu thực

$$\text{variance}(\hat{\beta}) = E\left[\left(E[\hat{\beta}] - \hat{\beta}\right)^2\right]$$

nghĩa là nếu vào Dj ra Bj dự đoán thì Bi dự đoán cũng không quá khác bj dự đoán

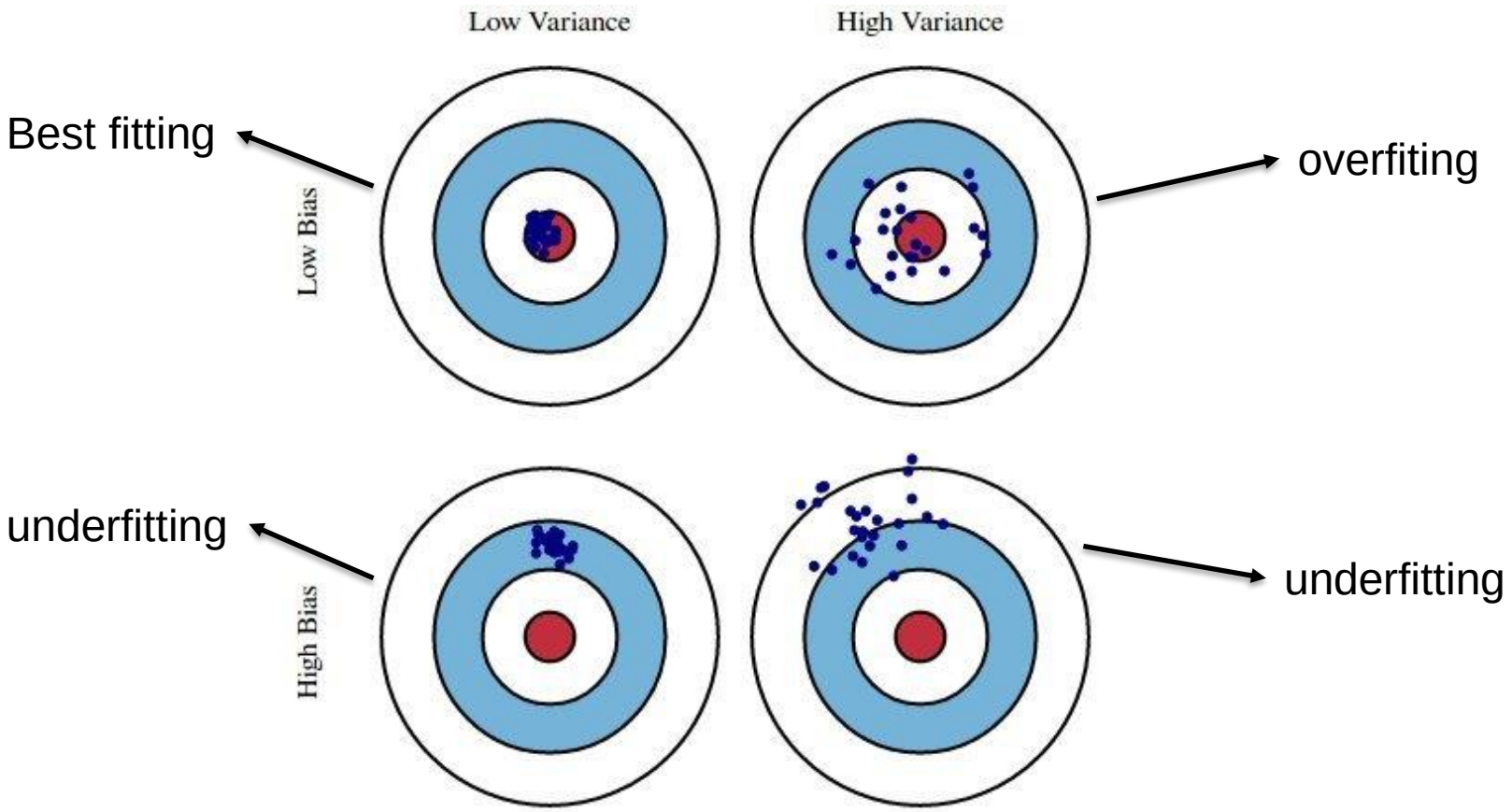
- **Variance thấp**: thể hiện tính tổng quát cao của mô hình, dù huấn luyện trên một tập con vẫn đoán đúng trên dữ liệu chưa thấy (tập test)
- **Variance cao**: thể hiện mô hình không đoán tốt trên dữ liệu chưa gặp

đây hiểu là độ lệch nhưng bình phương lên để 1. triệt tiêu âm, 2. phạt mạnh điểm ở xa

Variance lớn → các điểm dữ liệu văng xa khỏi mean
Variance nhỏ → dữ liệu tập trung sát mean
Nếu mean là “trung tâm khối”, variance là “độ rung” quanh tâm đó.

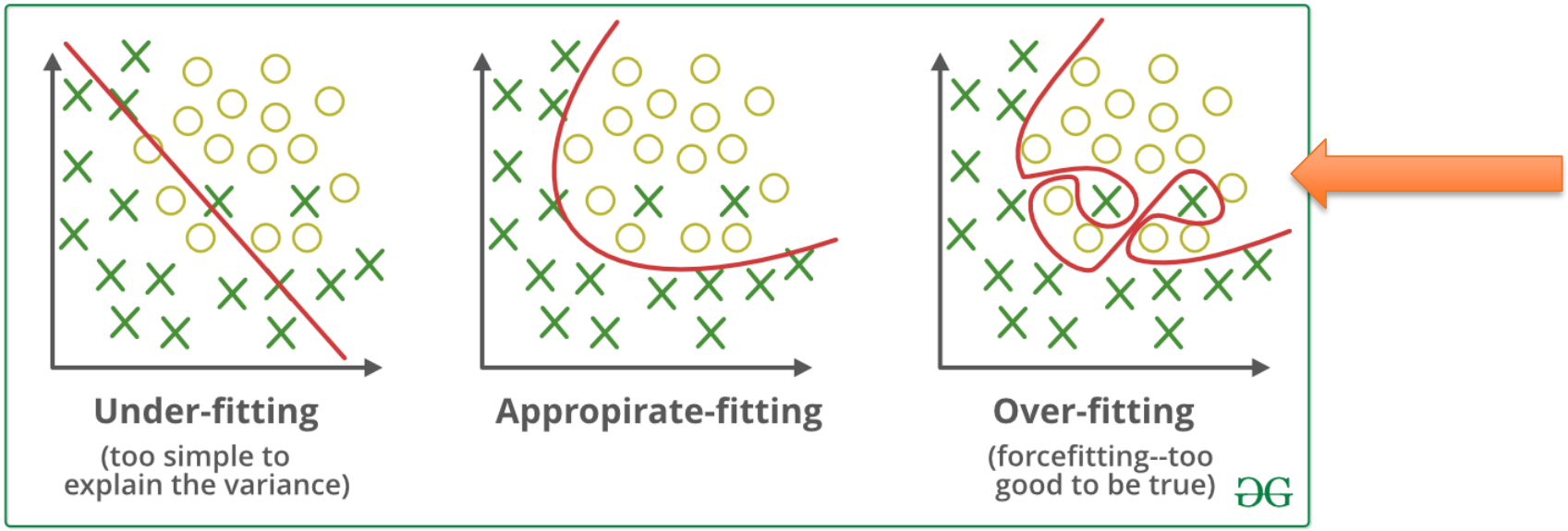
Cân bằng bias-variance

- Khi xây dựng mô hình và huấn luyện trên tập dữ liệu cần chú ý đảm bảo cả bias và variance đều thấp



Hiện tượng overfiting

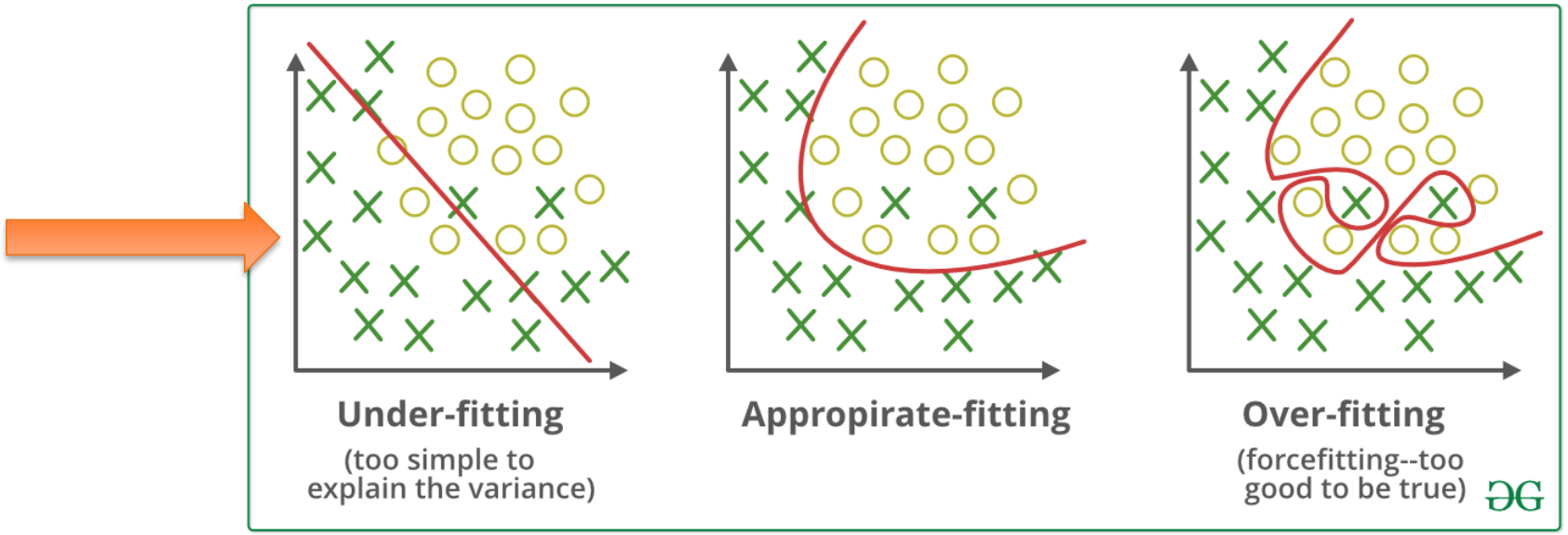
- **Nguyên nhân**: Xảy ra khi mô hình quá phức tạp hoặc dữ liệu không đủ khái quát



- **Để tránh overfiting**:
  - ➔ Giảm bớt sự phức tạp của mô hình (**giảm tham số**)
  - ➔ Lấy mẫu thêm dữ liệu để bao gồm các tình huống thực tế (**tăng dữ liệu**)

Hiện tượng underfitting

- **Nguyên nhân**: Xảy ra khi mô hình quá đơn giản so với tính chất phức tạp của dữ liệu, hoặc dữ liệu không đủ khái quát



- **Để tránh underfiting**:
  - ➔ Chuyển sang mô hình có khả năng biểu diễn phức tạp hơn (**thay hàm mô hình**)
  - ➔ Lấy mẫu thêm dữ liệu để bao gồm các tình huống thực tế (**tăng dữ liệu**)

NỘI DUNG

1. MÔ HÌNH HỒI QUY TUYẾN TÍNH

2. BIAS VÀ VARIANCE

3. MÔ HÌNH LASSO, RIDGE VÀ ELASTIC NET

4. MỘT SỐ MÔ HÌNH PHI TUYẾN

LASSO Regression

- Hồi quy tuyến tính + chính quy hóa L_1

L1 =

- Có thể dùng để lựa chọn đặc trưng (Feature selection): mô hình cố gắng đưa các hệ số về 0 đối với những đặc trưng không quan trọng

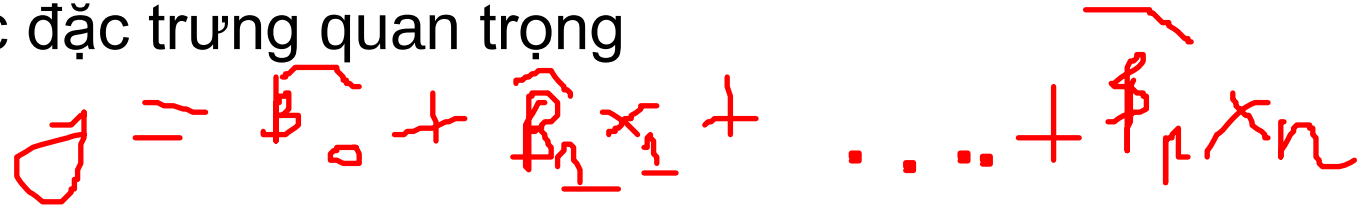
$$\underset{\hat{\beta}}{\operatorname{argmin}} \|Y - \hat{\beta}X\|^2 + \lambda \|\hat{\beta}\|_1$$

thêm 1 hệ số bên cạnh hàm loss

norm bậc 1

➔hướng đến chọn lựa các đặc trưng quan trọng

có thể dùng trong bước feature selection



giả sử x1 không liên quan lắm với y

thì x1, hay đại lượng nào khi thay đổi mà không ảnh hưởng y thì đại lượng Bi dự đoán đó giảm xuống dần 0 để loại bớt đặc trưng đó

Ridge Regression

- Hồi quy tuyến tính + chính quy hóa L_2

$$\underset{\hat{\beta}}{\operatorname{argmin}} \left\| Y - \hat{\beta} X \right\|^2 + \lambda \left\| \hat{\beta} \right\|_2^2$$

➔hướng đến khai thác hết các đặc trưng

β = [1,0,0,0] => 1^2 +3* o^2 =1

β = [0.25,0.25,0.25,0.25] => 4* (0.25)^2 <1

cả 2 cách chọn đều có tổng bằng 1 tuy nhiên trường hợp 2 thì khai thác hết các đặc trưng được trải đều ra khi ta muốn tìm giá trị nhỏ nhất, với trường hợp 1 thì nó bỏ các đăc trưng còn lại mô hình này ưu tiên khai thác hết đặc trưng

Elastic Net

ensemble model

- Hồi quy tuyến tính + chính quy hóa $L_1 + L_2$

$$\underset{\hat{\beta}}{\operatorname{argmin}} \|Y - \hat{\beta}X\|^2 + \lambda \left(\frac{1-\alpha}{2} \|\hat{\beta}\|_2^2 + \alpha \|\hat{\beta}\|_1 \right)$$

L2 L1

MSE

đại diện sai số giá trị dự đoán và mong muốn

➔ Cân bằng các tiêu chí của LASSO và Ridge Regression

lambda ở 3 mô hình trên là

mong muốn đưa vào tham s, Lambda càng lớn thì muốn tập trung đưa các B dự đoán càng nhỏ , nghĩa là phạt càng mạnh đối với tham số không ảnh hưởng đến y

Lambda nhỏ thì ta ưu tiên tối ưu sai số MSE hơn

Nghĩa là loss để phạt dự đoán, còn L1,L2 thì phạt trọng số

NỘI DUNG

1. MÔ HÌNH HỒI QUY TUYẾN TÍNH

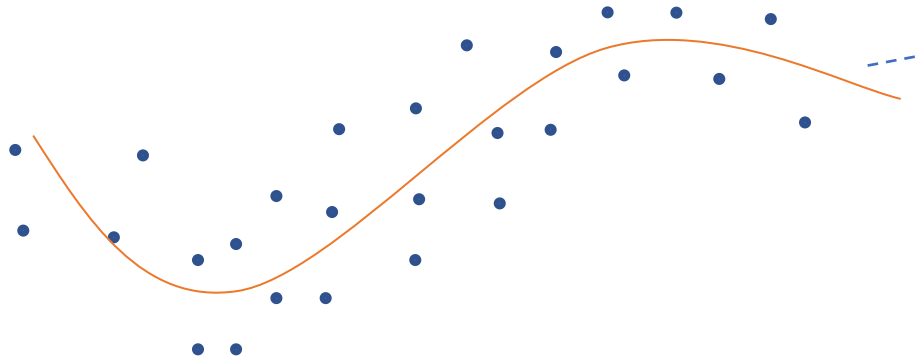
2. BIAS VÀ VARIANCE

3. MÔ HÌNH LASSO, RIDGE VÀ ELASTIC NET

4. MÔ HÌNH VỚI DỮ LIỆU CÓ QUAN HỆ PHI TUYẾN

Một số mô hình Regression khác

- Vẫn dùng LinearRegression, nhưng với feature engineer:



Dự đoán dạng của hàm, ta tạo các đặc trưng tương ứng.
Ví dụ: mô hình hàm số bậc 3

khi biết được hoặc dự đoán được hình dạng của hàm thì ta có thể đưa vào thêm x tương ứng để dễ hình thành nên mô hình hơn

- Trước đây $X = \begin{bmatrix} 1 \\ x \end{bmatrix}$, ta tạo thêm các đặc trưng $X_{\text{new}} = \begin{bmatrix} 1 \\ x \\ x^2 \\ x^3 \end{bmatrix}$

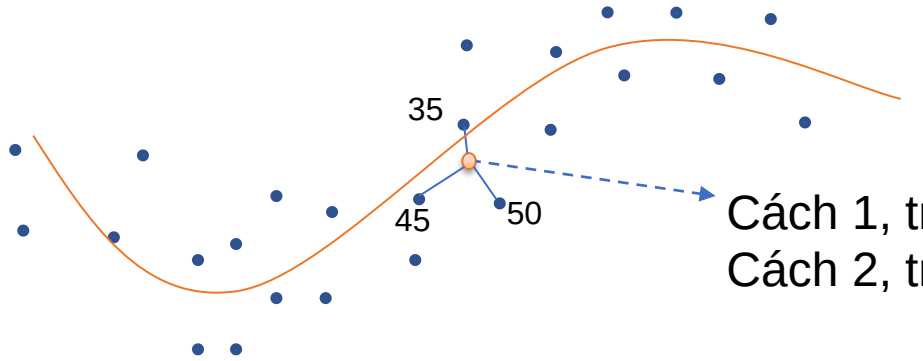
bổ sung thêm biến bậc 2 và bậc 3 để tạo nên dạng hàm đã dự đoán

Một số mô hình Regression

k nearest neigbor

- **KNNRegressor**

k đặc trưng nào gần điểm đầu vào nhất thì lấy ra, dự đoán dựa trên các điểm gần, nhưng lại bỏ qua khoảng cách xa gần giữa điểm đặc trưng với các điểm gần nhất này rồi (cách 1)



Cách 1, trung bình: (35+45+50)/3 = 43.3
Cách 2, trung bình trọng số theo khoảng cách

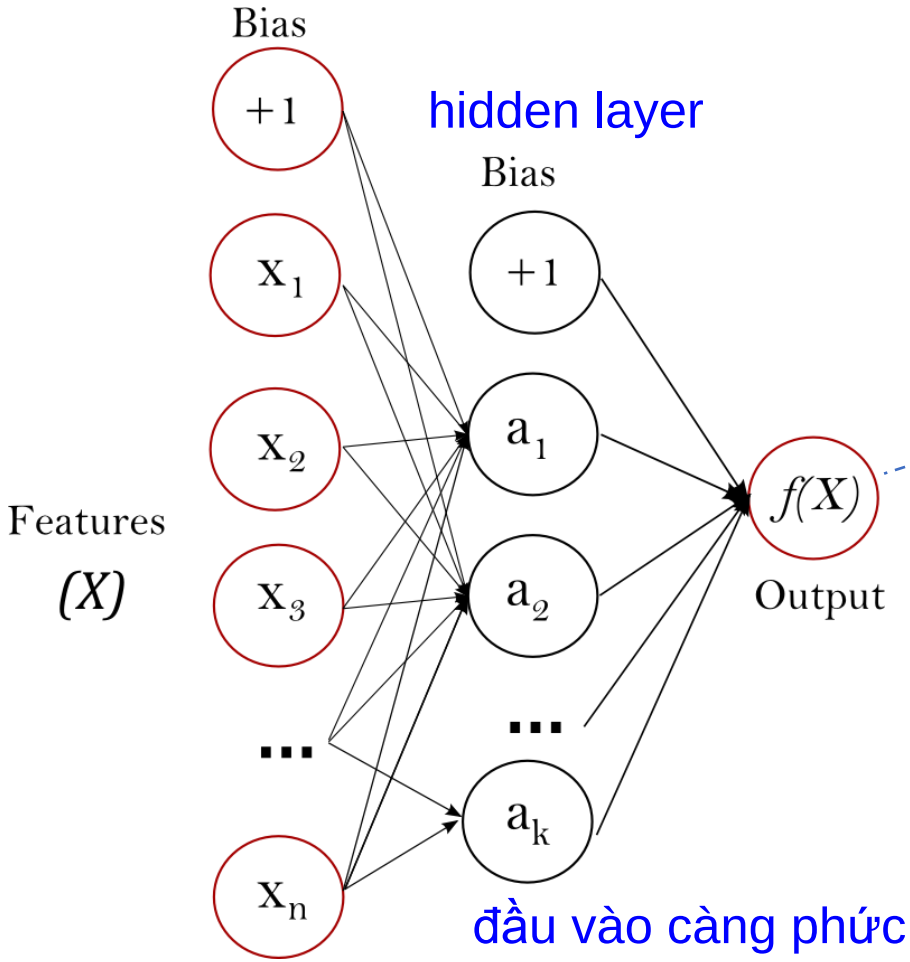
càng xa trọng số càng nhỏ, càng gần trọng số càng lớn

rất tốn tài nguyên vì phải lưu trữ all dataset để tính toán

Một số mô hình Regression

trong scikit learn thì có MLPRegresor để dùng cái này

- **Mạng Neural Network** (hay Multi-Layer Perceptron)



Không có hàm kích hoạt cuối

không có hàm này vì sigmoid sẽ ép hàm thành 0 ->1

đầu vào càng phức tạp thì hidden layer nên càng nhiều để giảm bớt feature

Một số mô hình Regression

- Các mô hình có nguồn gốc từ mô hình phân lớp:
  - Support Vector Regressor
  - Decision Tree Regressor
  - Random Forest Regressor
  - Gradient Boost, XGBoost, LightGBM, CatBoost Regressor

BÀI QUIZ VÀ HỎI ĐÁP