# Fundamentals of AI and DS

## Machine Learning

Tien-Lam Pham
Anh-Tuan Nguyen
Phenikaa School of Computing

# Intelligent Machine

**Học từ dữ liêu**

**(Learning from data)**

**Học máy**

**(Machine learning)**

understand the world

**Tri thức hoặc trí tuệ**

**(Knowledge or Intelligence)**

**Quyết định hành động, hoặc dự đoán**

**action, $y$**

**sensor data, $x$**

**Dữ liệu vào**

**Quan trắc thế giới**

**(World observation)**

# Intelligent Machine

**INPUT**
**Information**
**(X)**

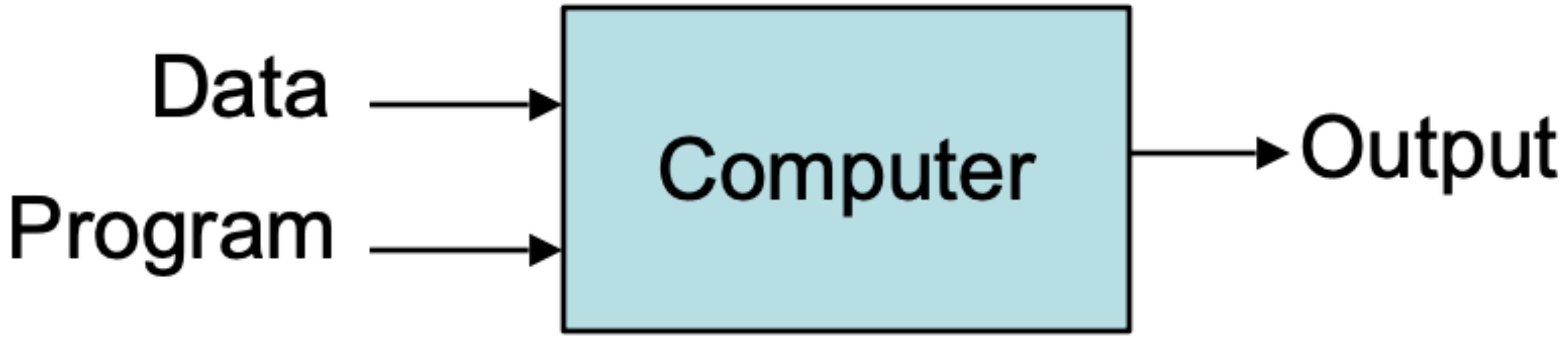
**Give machine  set of rules**

**Learning from data**
**(Machine learning)**

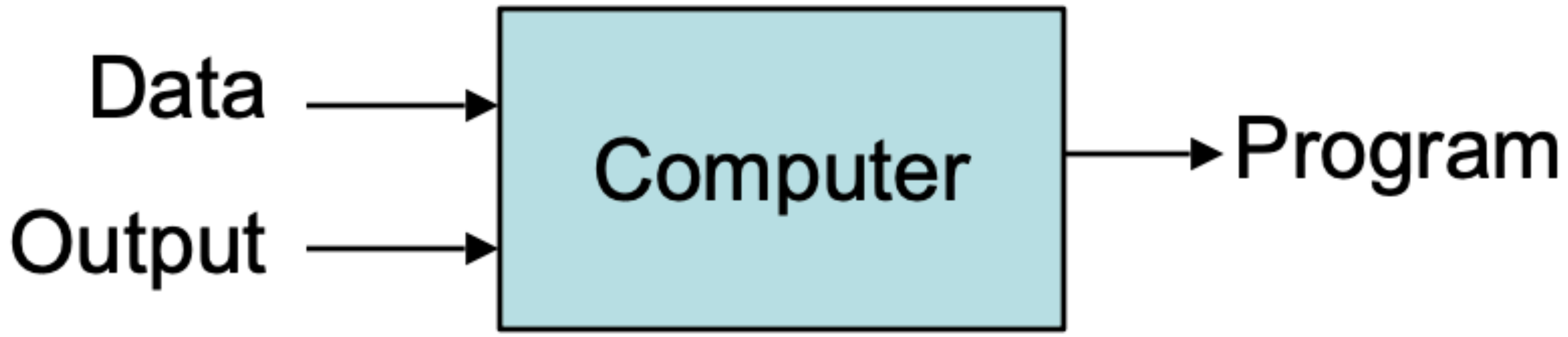

**OUTPUT**
**Prediction (y)**
**Action (0, 1)**

# Data processing

## Traditional Programming




## Machine Learning



# Machine Learning

**"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."**

***Tom Mitchell***

**“Một chương trình máy tính được cho là có khả năng học từ kinh nghiêm (dữ liệu), E, để thực hiện một chức năng T với hiệu năng P, nếu nó thực hiện chức năng T tốt hơn (được đo bằng hiệu năng P) nếu có mặt dữ liệu E”**

# Data-Driven Approach

**(1) Problem setting: inputs -> outputs**

**(2) Data collection**

$$D = \{(x_i, y_i), i = 1, 2, ..., m\}$$

**(3) Modeling and Training Models**

```
def train(images, labels):
  # Machine learning!
  return model
```

**(4) Model selection**

**(5) Deploy suitable model (Using the best model to make prediction)**

```
def predict(model, test_images):
  # Use model to predict labels
  return test_labels
```

# Example: Linear Regression

**T: predict profit of startups**
**E: data**
**P: Accuracy**

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | R&D Spend | Administration | Marketing Spend | State | Profit |
| 2 | 165349.2 | 136897.8 | 471784.1 | New York | 192261.83 |
| 3 | 162597.7 | 151377.59 | 443898.53 | California | 191792.06 |
| 4 | 153441.51 | 101145.55 | 407934.54 | Florida | 191050.39 |
| 5 | 144372.41 | 118671.85 | 383199.62 | New York | 182901.99 |
| 6 | 142107.34 | 91391.77 | 366168.42 | Florida | 166187.94 |
| 7 | 131876.9 | 99814.71 | 362861.36 | New York | 156991.12 |

$$D = \{(x_i, y_i), i = 1, 2, ..., m\}$$

$$x_i = (x_{i1}, x_{i2}, x_{i3})$$

$y_i$ = profit of startup i

$x_{i1}$: R&D spend of startup i

$x_{i2}$: Administration fee of startup i

$x_{i3}$: Marketing fee of startup i

# Example: Linear Regression

**Step 1: Data collection**

$$D = \{(x_i, y_i), i = 1, 2, ..., m\}$$

$$x_i = (x_{i1}, x_{i2}, x_{i3})$$

$$y_i = \text{profit of startup i}$$

**Step 2: Modeling and Training model**

$$\begin{aligned} \hat{y}_i &= f(x_{i1}, x_{i2}, x_{i3}) \\ &= w_1 \times x_{i1} + w_2 \times x_{i2} + w_3 \times x_{i3} \\ &= \vec{x}_i \cdot \vec{w} \end{aligned}$$

# Example: Linear Regression

**Step 2: Modeling and Training model**

$$\begin{aligned}\hat{y}_i &= f(x_{i1}, x_{i2}, x_{i3}) \\ &= w_1 \times x_{i1} + w_2 \times x_{i2} + w_3 \times x_{i3} \\ &= \vec{x}_i \cdot \vec{w}\end{aligned}$$

**Loss Function (Cost function)**

$$\begin{aligned}L(w_1, w_2, w_3) &= \frac{1}{2m}\sum_i^m (\hat{y}_i - y_i)^2 \\ &= \frac{1}{2m}\sum_i^m (w_1 x_{i1} + w_2 x_{i2} + w_3 x_{i3} - y_i)^2\end{aligned}$$

**Training = finding w which minimizes L**

**Cần tăng w1 hay giảm w1 để loss function giảm?**
**(L = w**2, w=1, muốn giảm L thì tăng w hay giảm ư?)**

# Example: Vector Representation

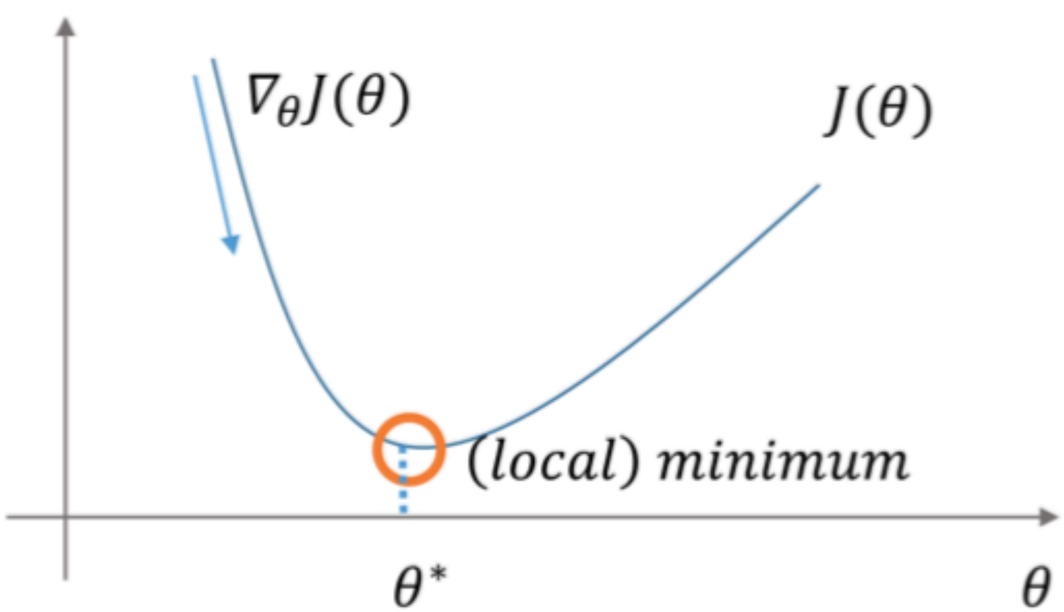
$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ \vdots & \vdots & \vdots \\ x_{m1} & x_{m2} & x_{m3} \end{bmatrix}$$

$$\begin{aligned} \hat{Y} &= X \cdot w^T \\ &= \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ \vdots & \vdots & \vdots \\ x_{m1} & x_{m2} & x_{m3} \end{bmatrix} \cdot \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_m \end{bmatrix} \end{aligned}$$

$$L = \frac{1}{2m}(\hat{Y} - Y)^T \cdot (\hat{Y} - Y)$$

# Gradient descent

- Gradient descent is a way to minimize an objective function $J(\theta)$
  - $\theta \in \mathbb{R}^d$: model parameters
  - $\eta$: learning rate
  - $\nabla_\theta J(\theta)$: gradient of the objective function with regard to the parameters
- Updates parameters **in opposite direction** of gradient.
- Update equation: $\theta = \theta - \eta \cdot \nabla_\theta J(\theta)$



# Gradient Descent

$$w_1 := w_1 - \eta \frac{\partial L}{\partial w_1}$$

$$w_2 := w_2 - \eta \frac{\partial L}{\partial w_2}$$

$$w_3 := w_3 - \eta \frac{\partial L}{\partial w_3}$$

$$\begin{aligned}\frac{\partial L}{\partial w_1} &= \frac{1}{m}\sum_i^m x_{i1}(\hat{y} - y_i) \\ &= \frac{1}{m} x_1^{(c)T} \cdot (\hat{(Y)} - Y)\end{aligned}$$

$$\begin{aligned}\frac{\partial L}{\partial w_2} &= \frac{1}{m}\sum_i^m x_{i2}(\hat{y} - y_i) \\ &= \frac{1}{m} x_2^{(c)T} \cdot (\hat{(Y)} - Y)\end{aligned}$$

$$\begin{aligned}\frac{\partial L}{\partial w_3} &= \frac{1}{m}\sum_i^m x_{i3}(\hat{y} - y_i) \\ &= \frac{1}{m} x_3^{(c)T} \cdot (\hat{(Y)} - Y)\end{aligned}$$

# Gradient Descent: Vectorization

$$w_1 := w_1 - \eta \frac{\partial L}{\partial w_1}$$

$$w_2 := w_2 - \eta \frac{\partial L}{\partial w_2}$$

$$w_3 := w_3 - \eta \frac{\partial L}{\partial w_3}$$

$$W^T := W^T - \eta \nabla_W L$$

$$\nabla_W L = X^T \cdot (\hat{Y} - Y)$$

# Algebra Solution
# (Nghiệm bằng phương pháp đại số)

$$X \cdot w^T = Y$$

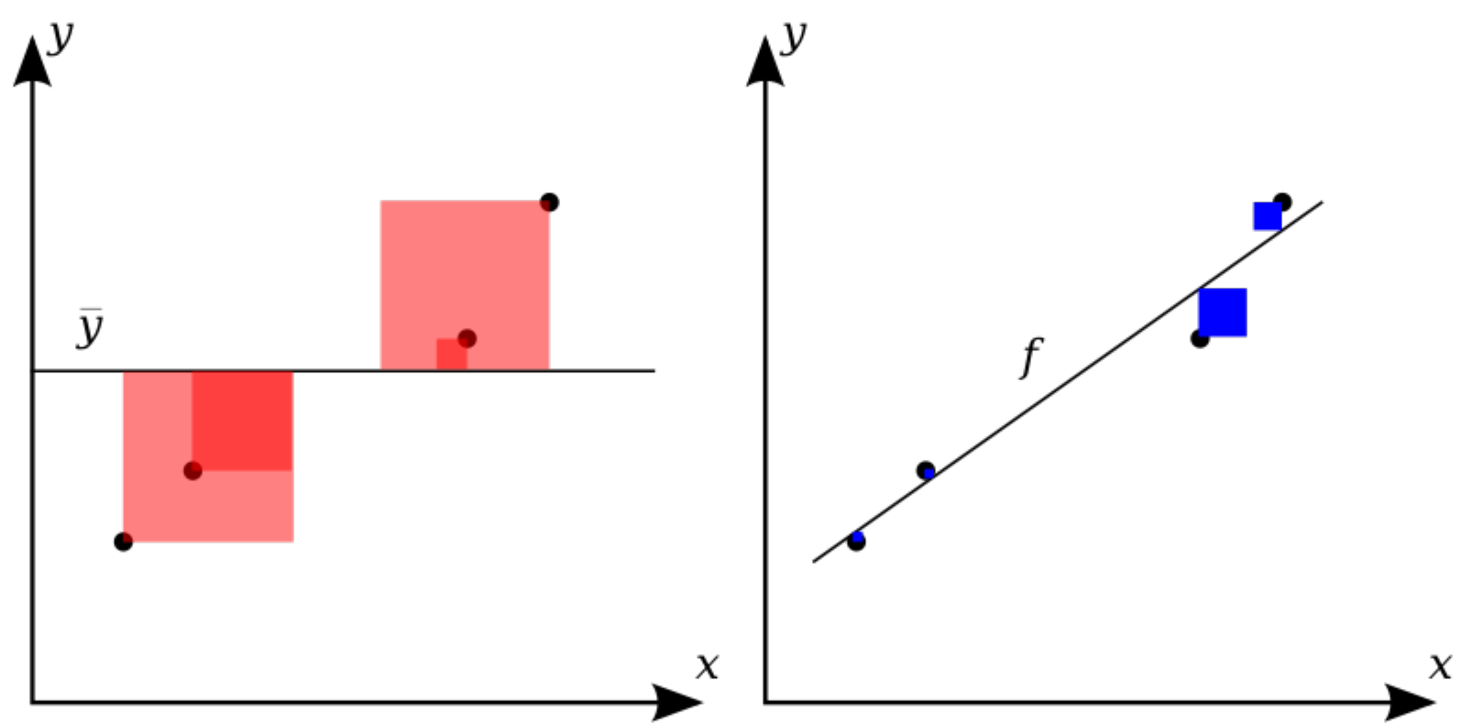
$$X^T \cdot X \cdot w^T = X^T \cdot Y$$

$$w^T = (X^T \cdot X)^{-1} \cdot X^T \cdot Y$$

# Đánh giá hiệu quả của mô hình (Evaluation of Model)

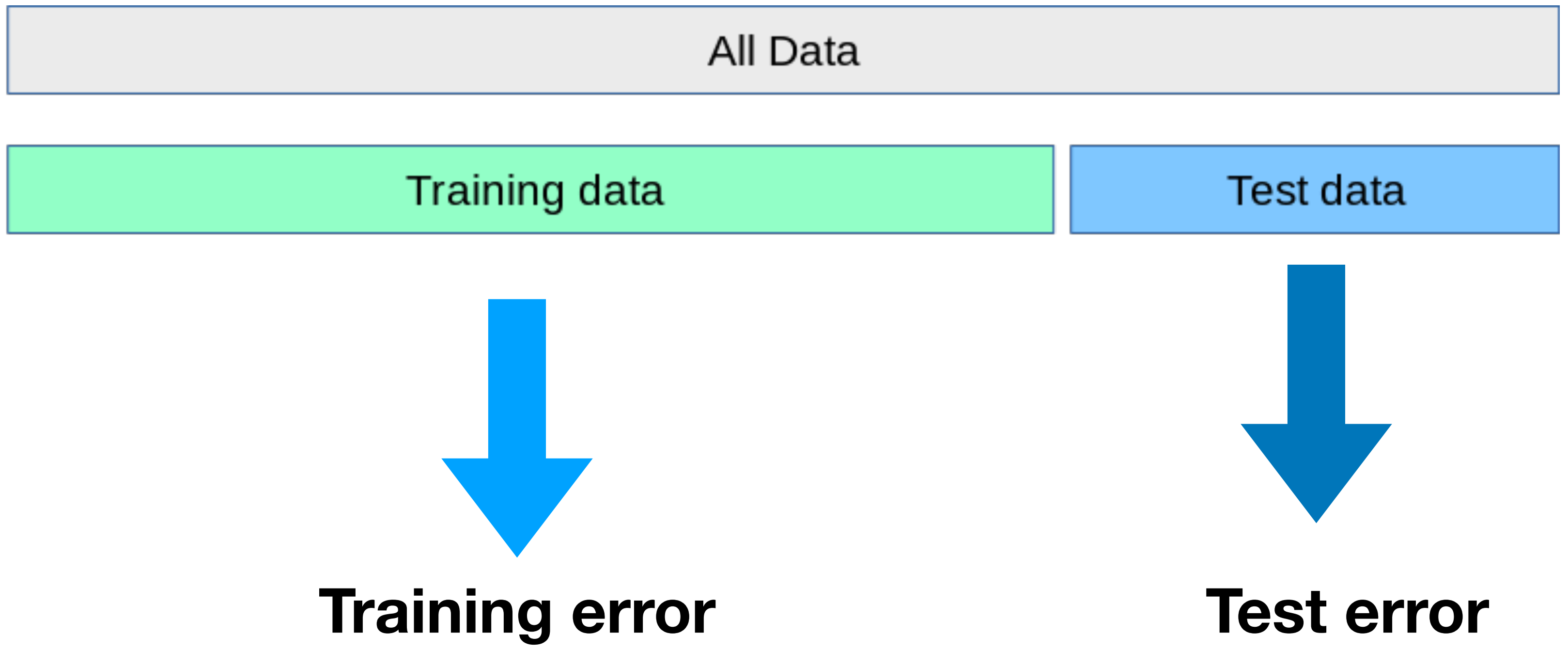
- Root Mean Square Error (RMSE)
- Mean Absolute Error (MAE)
- R2 score (coefficients of determination)

$$R^2 = 1 - \frac{SSR}{SST}$$

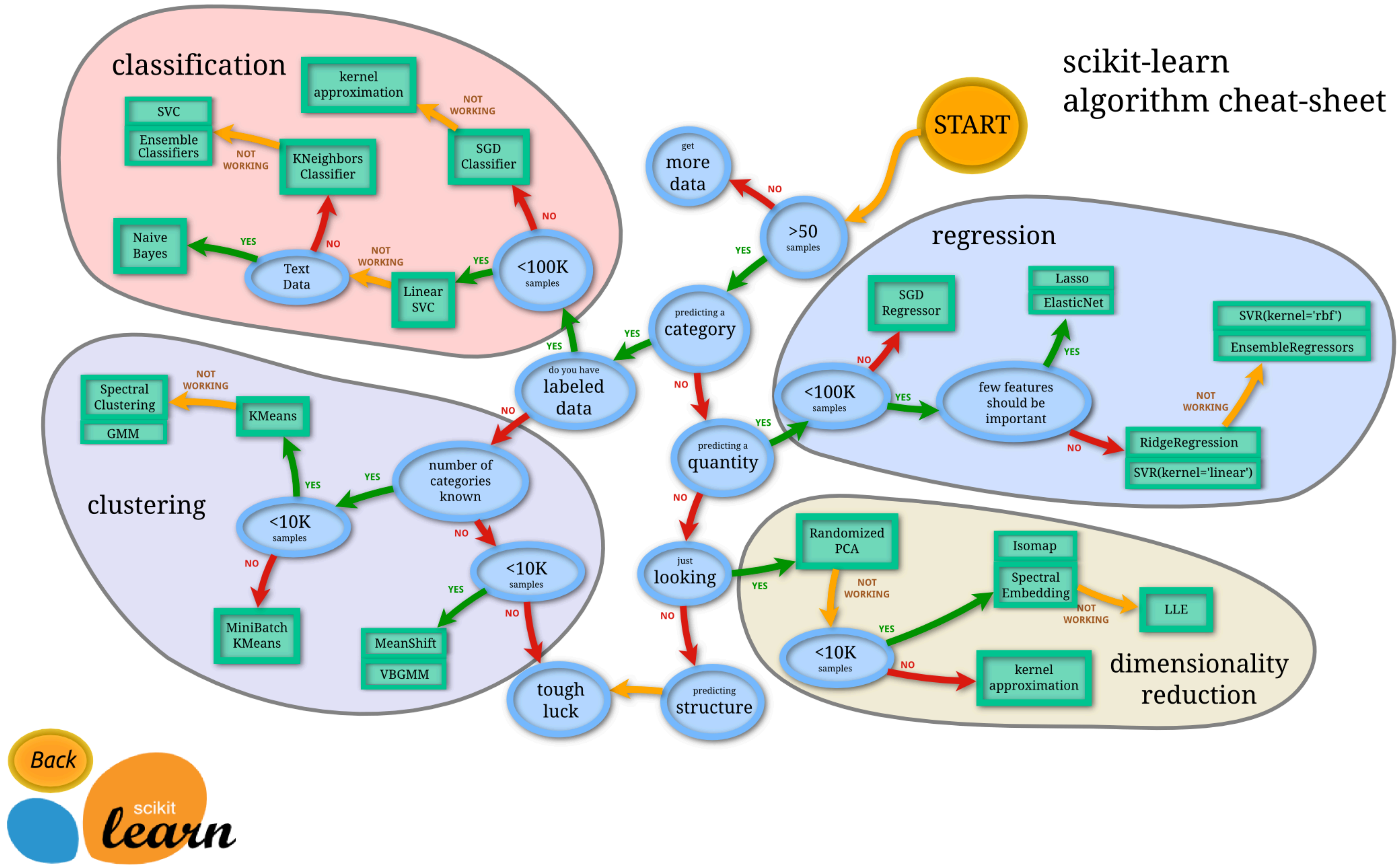



https://en.wikipedia.org/wiki/Coefficient_of_determination

# Phân chia Dữ liệu Huấn luyện và Dữ liệu Test



# Scikit-Learn Libs



**Nhiệm vụ dự đoán mức lương (salary)**
**dựa vào số năm kinh nghiệm (YearsExperience)**

**(1) Thu thập dữ liệu**

| YearsExperie | Salary |
|---|---|
| 1.1 | 39343 |
| 1.3 | 46205 |
| 1.5 | 37731 |
| 2 | 43525 |

**(2) Xây dựng (modeling) và Huấn luyện mô hình (training model)**

**Modeling**

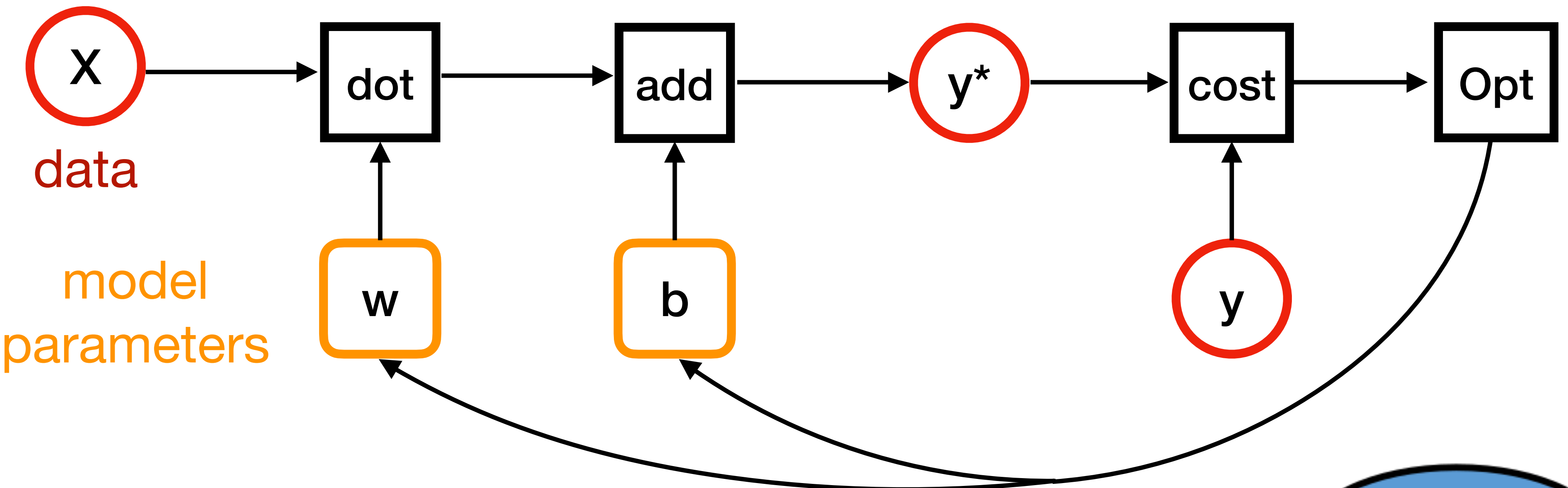
$$Salary = f(YearsExperience) = a \times YearsExperience + b$$

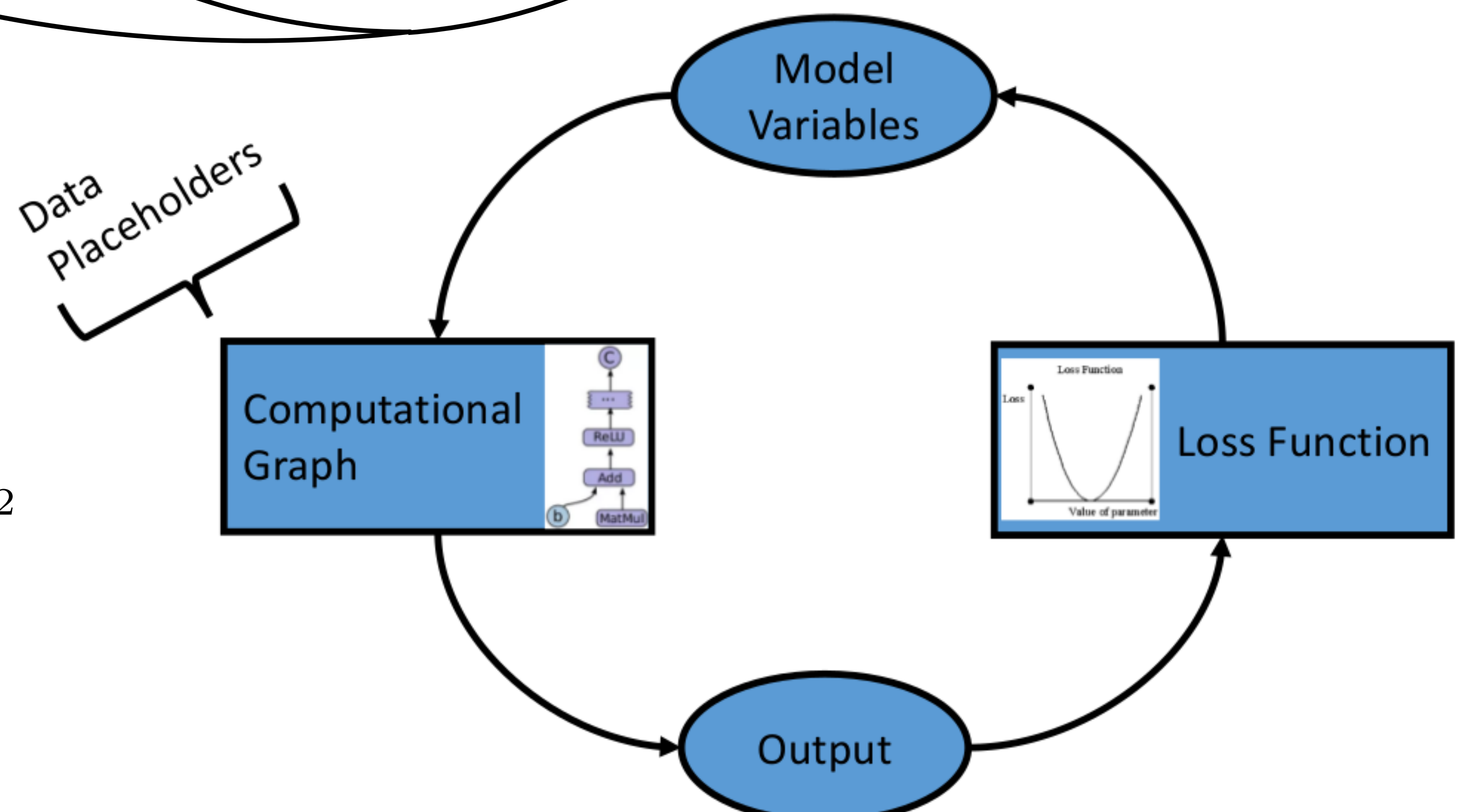
# Training??

**(3) Dự đoán mức lương của đ/c X có YearsExperience = 1.8**

# Học Máy (Machine Learning)

$$y = X \cdot w^T + b$$




$$cost(X, y) = \frac{1}{2}\sum_m (y_m - \hat{y}_m)^2$$



https://www.slideshare.net/NicholasMcClure1/introduction-to-neural-networks-in-tensorflow

# Học Máy (Machine Learning)

COMMUNICATIONS OF THE ACM | OCTOBER 2012 | VOL. 55 | NO. 10

**Tapping into the "folk knowledge" needed to advance machine learning applications.**

BY PEDRO DOMINGOS

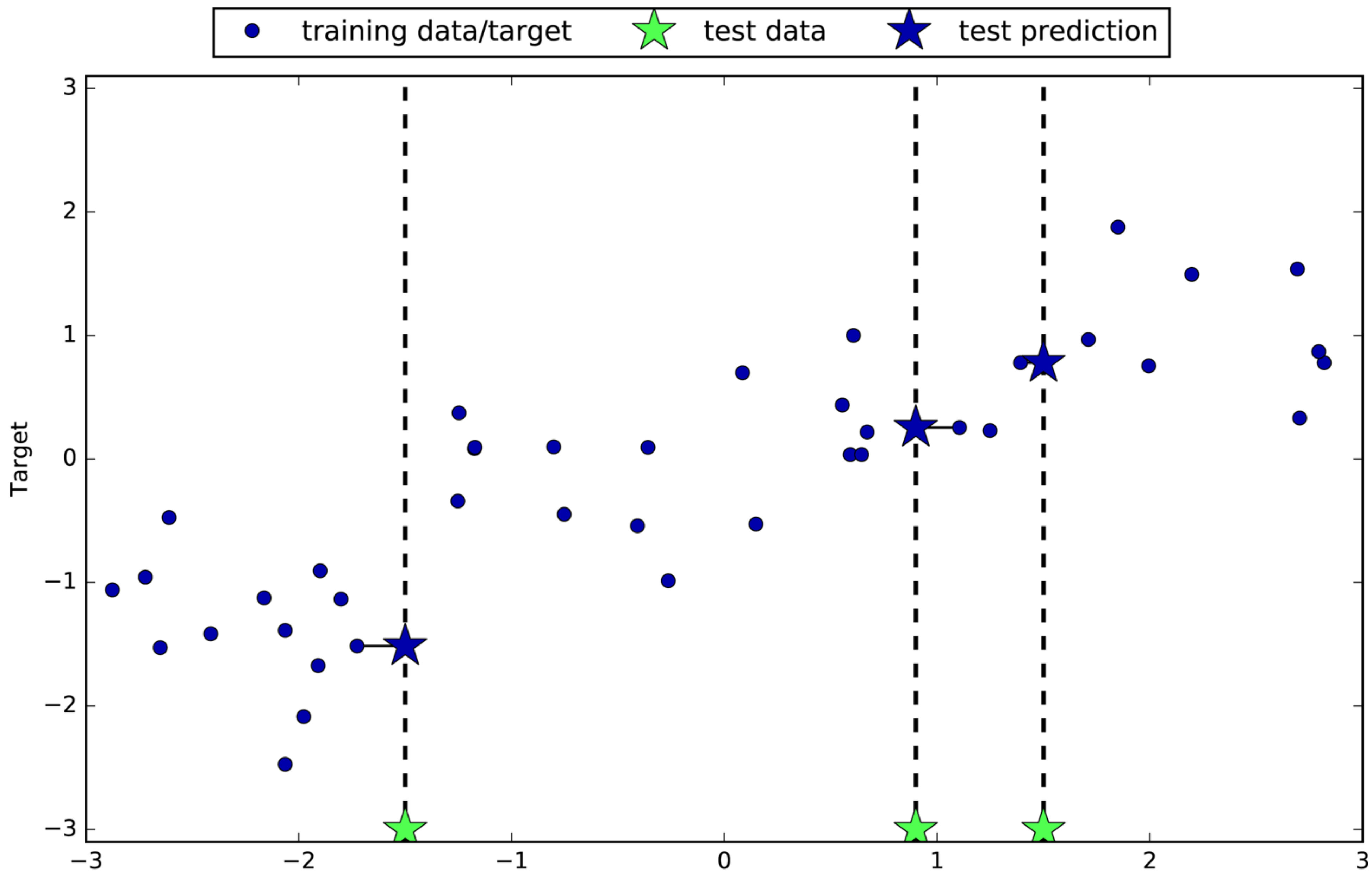
## A Few Useful Things to Know About Machine Learning

## Learning = Representation + Evaluation + Optimization

**Table 1. The three components of learning algorithms.**

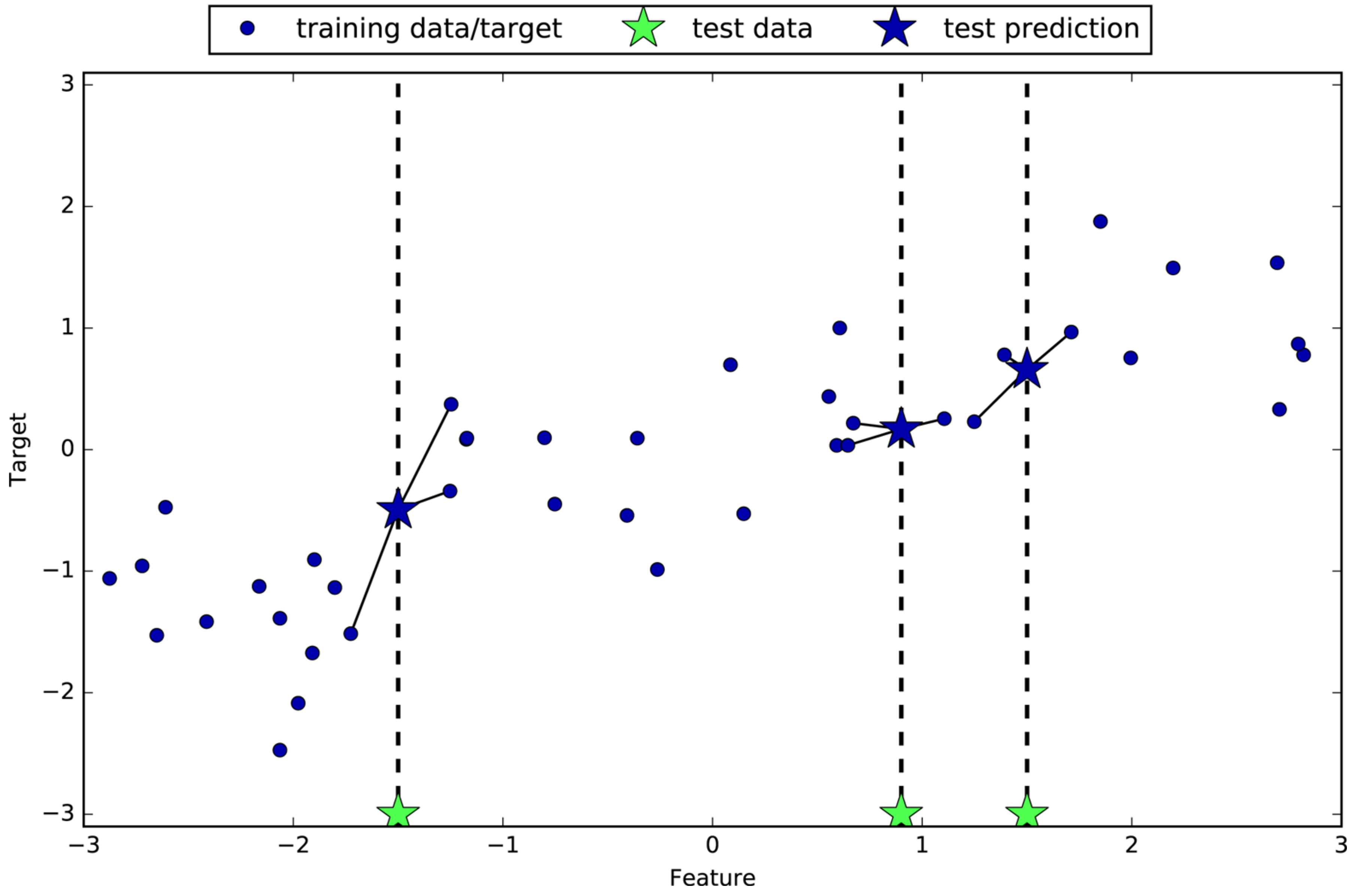
| Representation | Evaluation | Optimization |
|---|---|---|
| Instances | Accuracy/Error rate | Combinatorial optimization |
| K-nearest neighbor | Precision and recall | Greedy search |
| Support vector machines | Squared error | Beam search |
| Hyperplanes | Likelihood | Branch-and-bound |
| Naive Bayes | Posterior probability | Continuous optimization |
| Logistic regression | Information gain | Unconstrained |
| Decision trees | K-L divergence | Gradient descent |
| Sets of rules | Cost/Utility | Conjugate gradient |
| Propositional rules | Margin | Quasi-Newton methods |
| Logic programs | | Constrained |
| Neural networks | | Linear programming |
| Graphical models | | Quadratic programming |
| Bayesian networks | | |
| Conditional random fields | | |

# K-nearest Regression




Andreas C. Müller and Sarah Guido, "Introduction to Machine Learning with Python", O’Reilly Media (2017). p41

# K-nearest Regression




Andreas C. Müller and Sarah Guido, "Introduction to Machine Learning with Python", O'Reilly Media (2017). P42

# K-nearest Regression

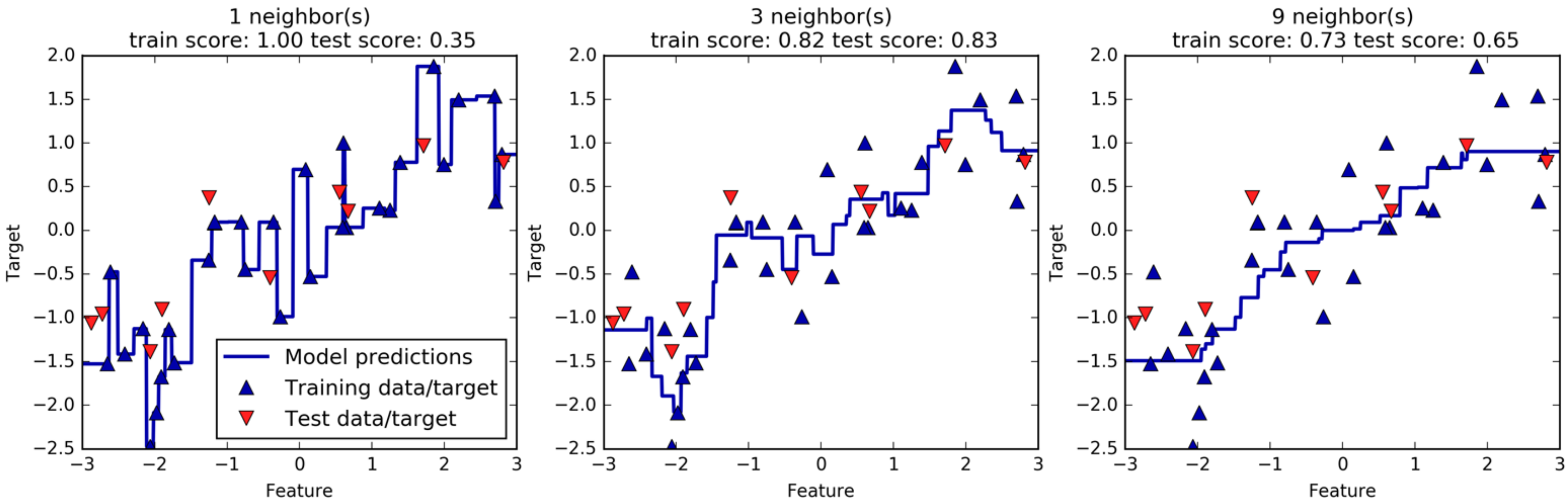
**(1) Chọn số neighbors (K)**

**(2) Xác định K điểm gần nhất của điểm dữ liệu mới cần dự đoán**

**(3) Trong K điểm dữ liệu gần nhất, đếm số điểm dữ liệu thuộc mỗi lớp**

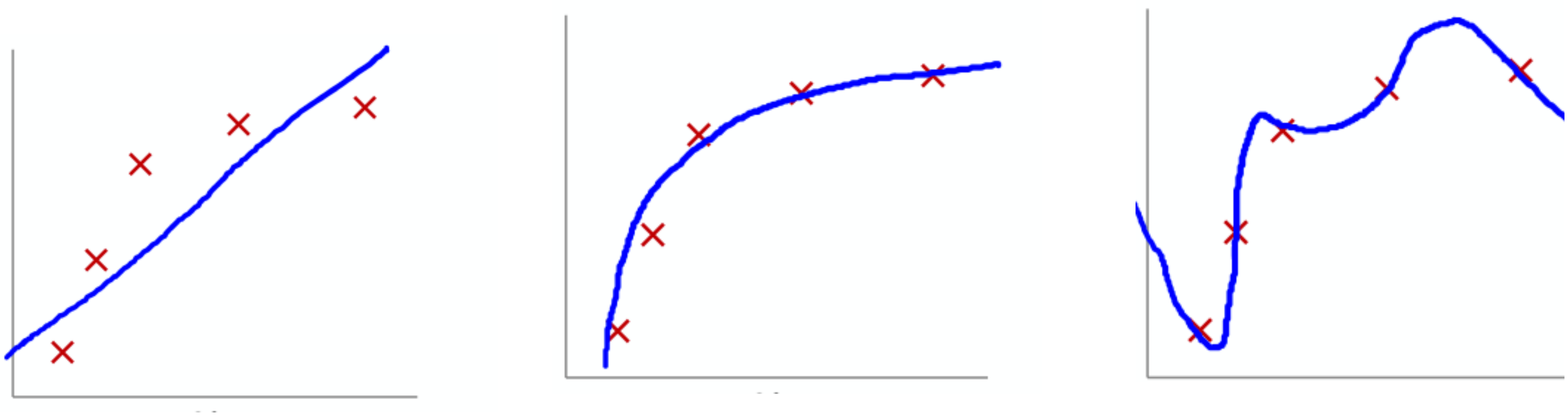
**(4) Điểm dữ liệu mới sẽ là giá trị trung bình của các điểm lân cận**

# K-nearest Regression




Andreas C. Müller and Sarah Guido, "Introduction to Machine Learning with Python", O'Reilly Media (2017). P44

# Overfit vs. Underfit



**Underfit** **Just right** **Overfit**

1. **Reduce number of features**
2. **Regularization**

# Regularization

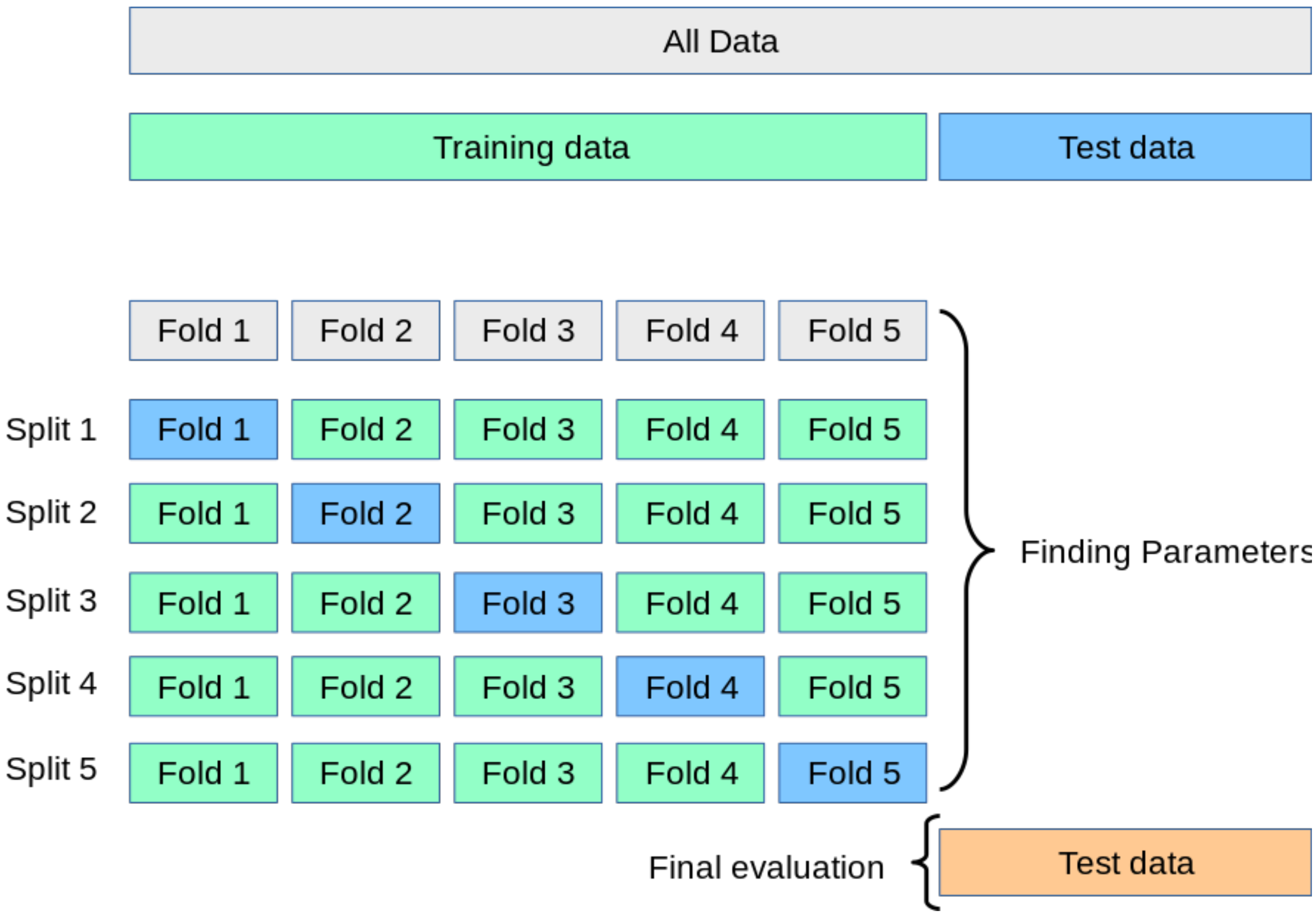
**L2-Regularization**

$$cost(X, y) = \frac{1}{2}\sum_{m}(y_m - \hat{y}_m)^2 + \lambda||w||_2^2$$

**L1-Regularization**

$$cost(X, y) = \frac{1}{2}\sum_{m}(y_m - \hat{y}_m)^2 + \lambda||w||_1^2$$

# Model Selection by Cross Validation




**https://scikit-learn.org/stable/modules/cross_validation.html**