Foundation of Supervised Learning

Prof. Je-Won Kang
Electronic & Electrical Engineering
Ewha Womans University

Learning from data

- What is this animal?
  - You would answer the question in few seconds
- Will you need a zoological (mathematical) definition to distinguish it?
  - Yes or **No**

We have learned a lot from data (pattern) despite being ignorant of a rigorous definition of a lion

data로 부터 내재된 정보 학습

Machine learning problems

1 **Is it a spam mail or not?**

Binary classification

2 **Image recognition**

Multi-class classification

ImageNet Large Scale Visual Recognition Challenge (ILSVRC)

3 **House prices**

Regression

Supervised learning

label

- Given a set of labeled examples $(\boldsymbol{x}^1, y^1)$, … , $(\boldsymbol{x}^N, y^N)$, learn a mapping function g: $\boxed{\boldsymbol{X} \rightarrow \boldsymbol{Y}}$, such that given an unseen sample $\boldsymbol{x}'$, associated output y' is predicted.

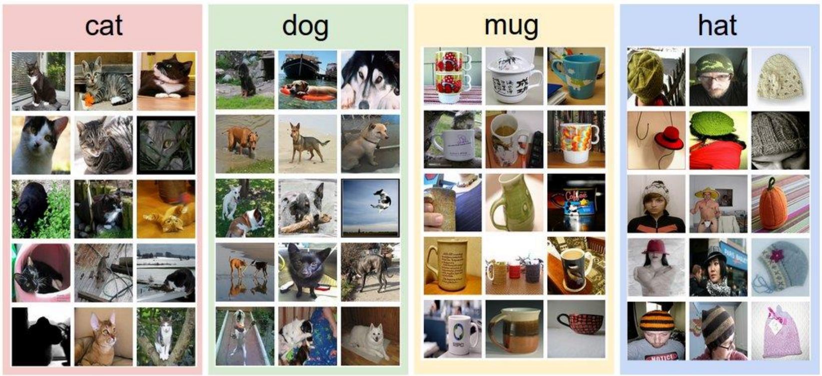
Regression (Examples)

- Exam Score Prediction (based on time)
- Stock Price Prediction (based on time)
- Weather (temperature forecasting)

Continuous variables

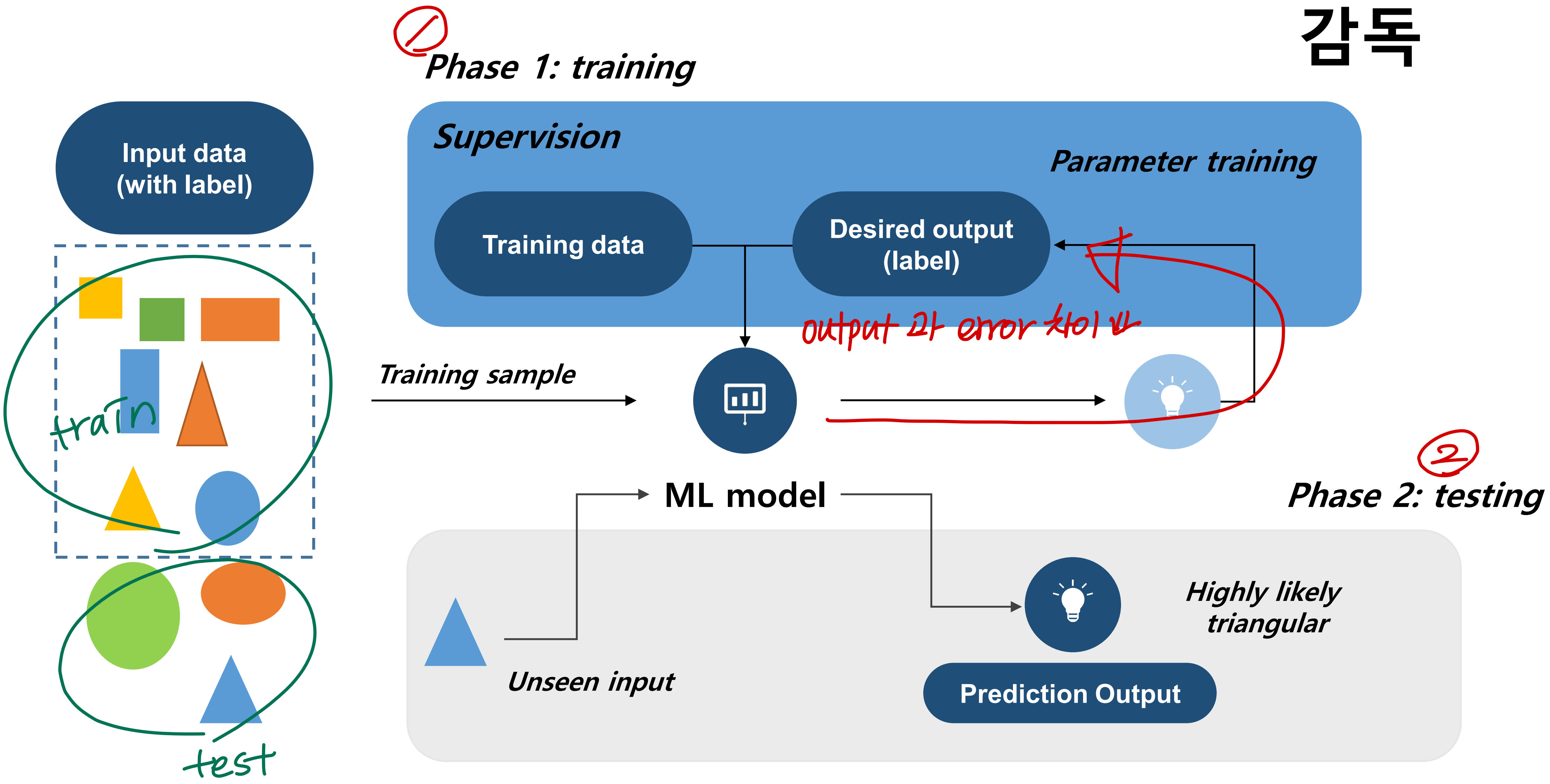
Classification (Examples)

- Pass/Fail (Binary Classification)
- Letter Grades (Multi-Level Classification)



Discrete variables

Learning pipeline

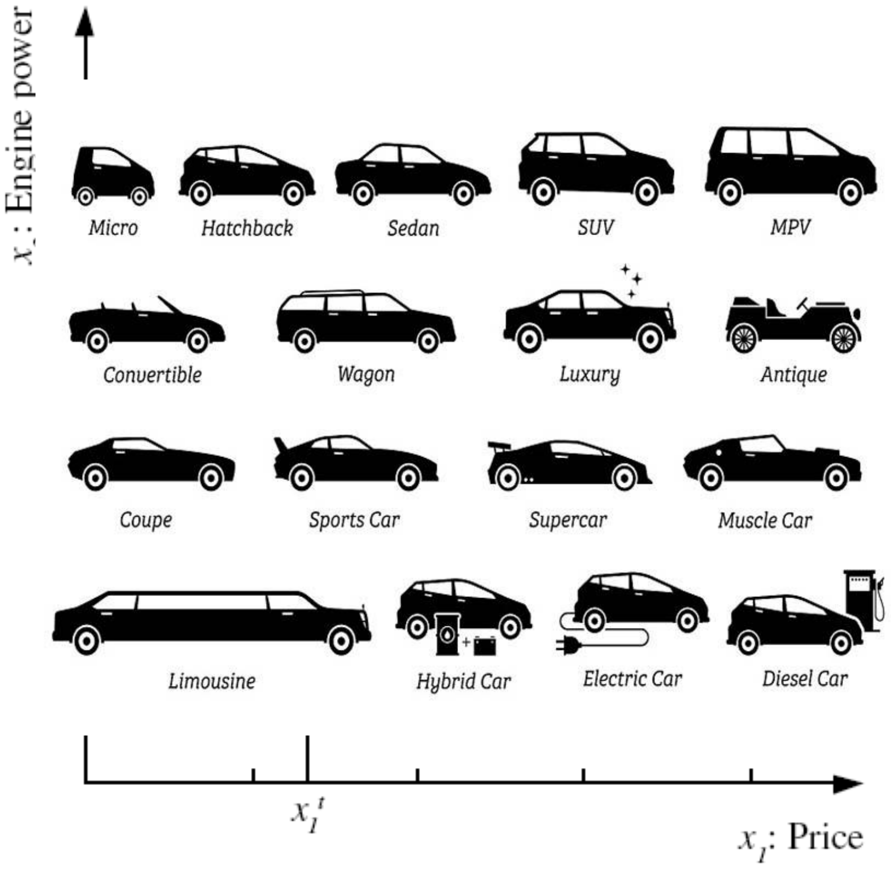


Example: family car

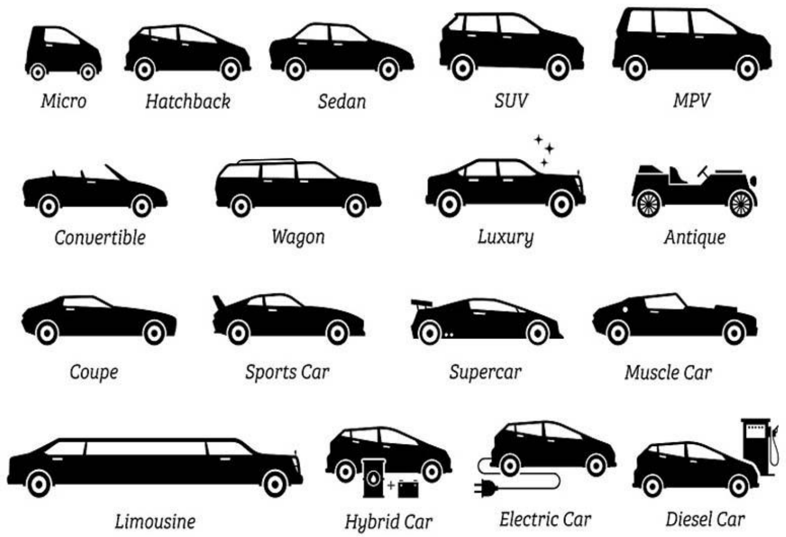
- Class C of a "family car"
  - Prediction: Is this car a family car?
- Output:

  Positive (+) and negative (–) examples, or multi-class examples
- Input representation:

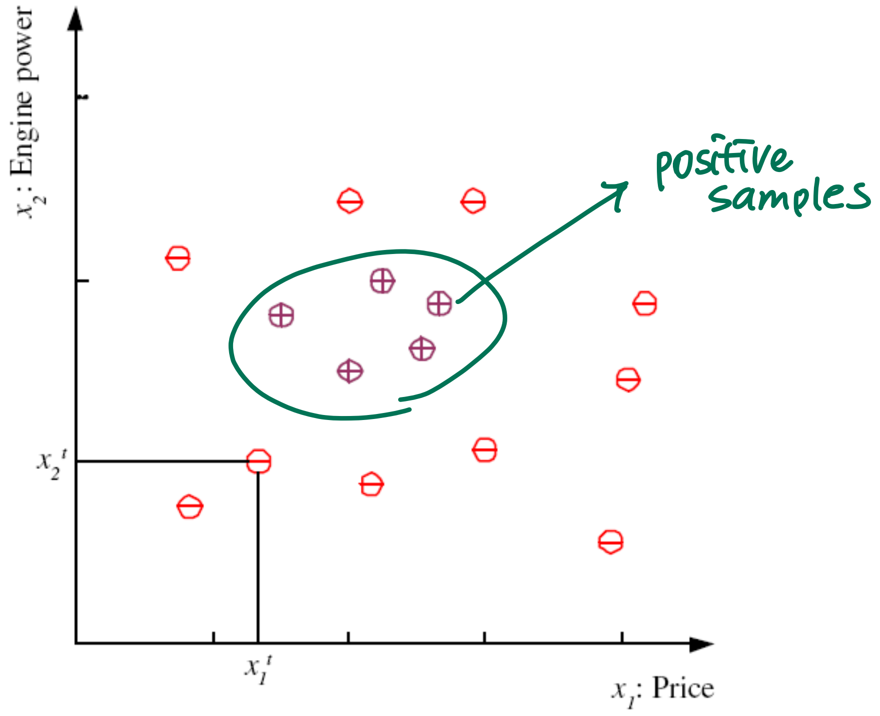
  x_1: price, x_2 : engine power



Problem formulation



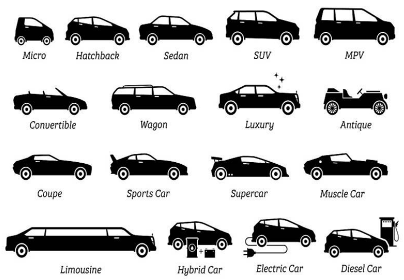
Input representation



Problem formulation

d 차원

- $\boldsymbol{X} = \mathbf{R}^d$ is an input space
  - $\mathbf{R}^d$: a d-dimensional Euclidean space
  - input vector $\boldsymbol{x} \in \boldsymbol{X}: \boldsymbol{x} = (x_1, x_2, \dots, x_d)$
- $\boldsymbol{Y}$ is an output space
  - Binary (yes/no) decision



- $\boldsymbol{X} = \mathbf{R}^d$ is an input space
  - $\mathbf{R}^d$: a d-dimensional Euclidean space
  - input vector $\boldsymbol{x} \in \boldsymbol{X}: \boldsymbol{x} = (x_1, x_2, \dots, x_d)$
- $\boldsymbol{Y}$ is an output space
  - Binary (yes/no) decision
- Now, we want to approximate a target function f
  - f: $\boldsymbol{X} \to \boldsymbol{Y}$ (unknown ideal function)
  - Data $(\boldsymbol{x}^1, y^1)$, …, $(\boldsymbol{x}^N, y^N)$; dataset where $y^N = f(\boldsymbol{x}^N)$
  - Correct label is ready for a training set
  - **Hypothesis** g: $\boldsymbol{X} \to \boldsymbol{Y}$ (ML model to approximate f) : $g \in \boldsymbol{H}$

- Now, we want to approximate a target function f
  - f: $\boldsymbol{X} \to \boldsymbol{Y}$ (unknown ideal function)
  - Data $(\boldsymbol{x}^1, y^1)$, …, $(\boldsymbol{x}^N, y^N)$; dataset where $y^N = f(\boldsymbol{x}^N)$
  - Correct label is ready for a training set
  - **Hypothesis** g: $\boldsymbol{X} \to \boldsymbol{Y}$ (ML model to approximate f) : $g \in \boldsymbol{H}$

Learning model

Learnable parameters to learn a boundary

Goal: target function
$f: X \rightarrow Y$

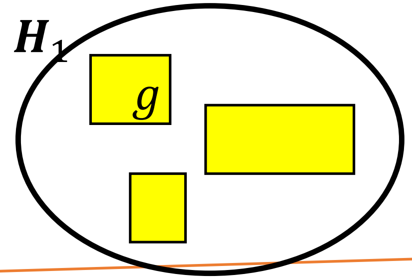
Training set
$D = \{\boldsymbol{x}^t, y^t\}_{t=1}^N$

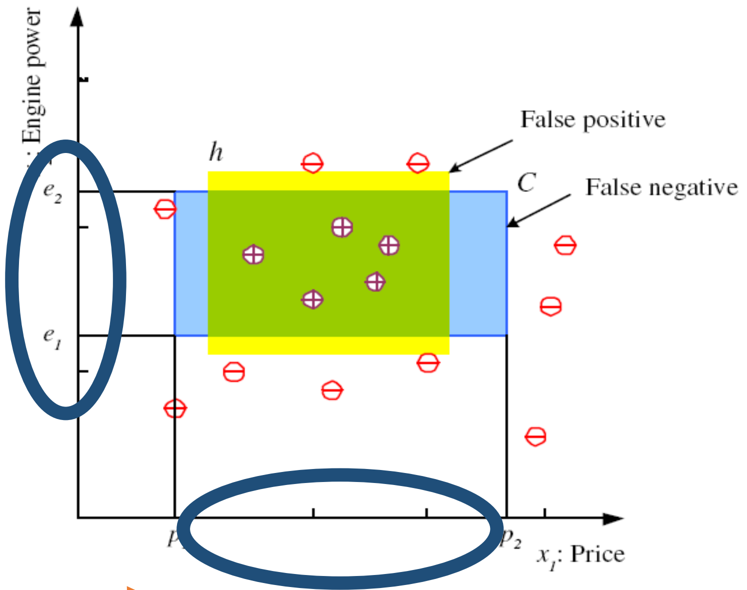
Learning model
- Feature selection
- Model selection
- Optimization

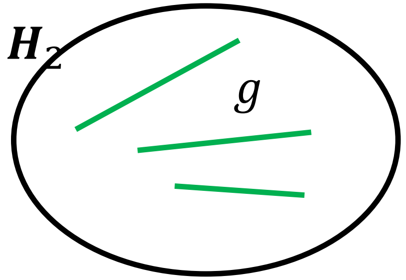
Hypothesis/Evaluation
$g \approx f$

Yes if $(p_1 \le \text{price} \le p_2)$ AND $(e_1 \le \text{engine power} \le e_2) > 0.5$

No if $(p_1 \le \text{price} \le p_2)$ AND $(e_1 \le \text{engine power} \le e_2) < 0.5$







- Linear model
- Support vector machine
- Neural network
- Decision tree

…

Attempts to find a model to match f on *new* samples

Model generalization

- Learning is an ill-posed problem; data is limited to find a unique solution
- Generalization (Goal): a model needs to perform well on unseen data

error function 최소화 하도록 노력 필요.

  - Generalization error E_{gen} ; the goal is to minimize this error, but it is impractical to compute in the real world

Learning from data	$\longrightarrow$	Learning from error (supervision)

  - Use training/validation/test set errors for the proxy

Errors

error 함수들

- Pointwise error is measured on an each input sample: $e(h(\boldsymbol{x}), y)$
- Examples:
  - ✓squared error $e(h(\boldsymbol{x}^{i}), y^{i}) = (h(\boldsymbol{x}^{i}) - y^{i})^{2}$
  - ✓binary error $e(h(\boldsymbol{x}^{i}), y^{i}) = \mathbf{1}[h(\boldsymbol{x}^{i}) \neq y^{i}]$
- From a pointwise error to overall errors
  - ✓$\boxed{E[(h(\boldsymbol{x}^{i}) - y^{i})^{2}]}$

손실함수 loss function (=cost function)

If an input sample is chosen from training, validation, and testing datasets, the errors are called a training error (E_{train}), a validation error (E_{val}), and a testing error (E_{test})

- Training error E_{train} measured on a training set, which may or may not represent E_{gen} ; used for fitting a model

- Testing error E_{test} (not used in training), which can be used for a proxy of E_{gen}

- Split into two objectives:
  1. $E_{test} \approx E_{train}$
  2. $E_{train} \approx 0$
- Objective 1: make $E_{test} \approx E_{train}$
  - Failure : overfitting → high variance
  - Cure: regularization, more data
- Objective 2: make $E_{train} \approx 0$
  - Failure : underfitting → high bias
  - Cure: optimization, more complex model

variance
학습한 모델이
일반적인 성능
갖도록 했다면
but,
overfitting의 가능성

모델의 정확도
오를 있다
bias 낮아짐
최적화 잘 수행해야 함.

Goal: $\mathrm{E_{test}} \approx \mathrm{E_{gen}} \approx 0$

How to achieve the goal in practice?

Bias and Variance

- Bias – error because the model can not represent the concept
- Variance – error because a model overreacts to small changes (noise) in the training data

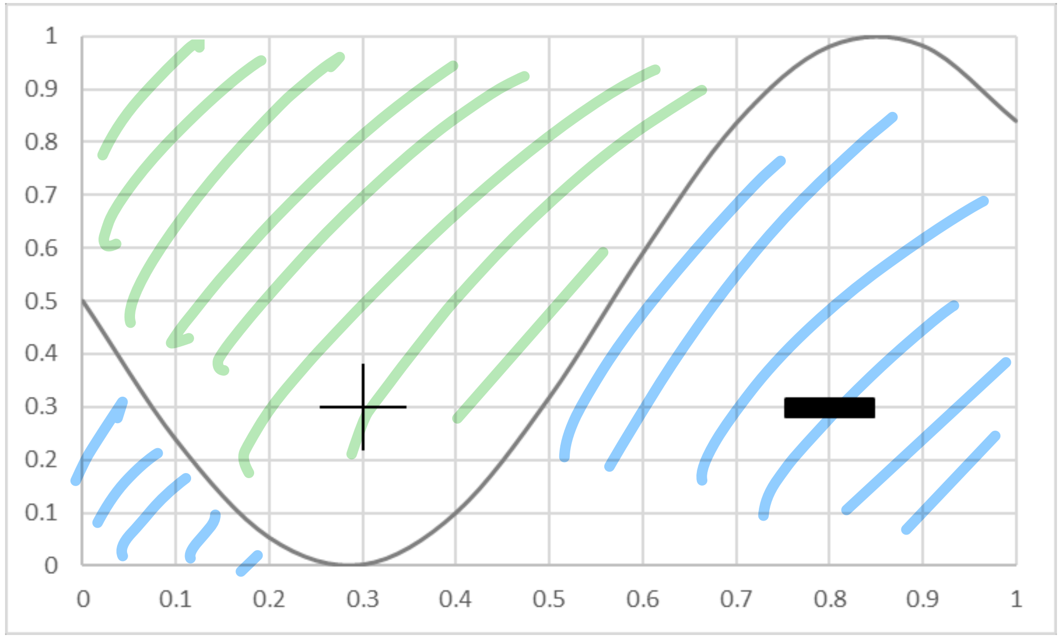
Total Loss = Bias + Variance (+ noise)

high bias

Underfitting

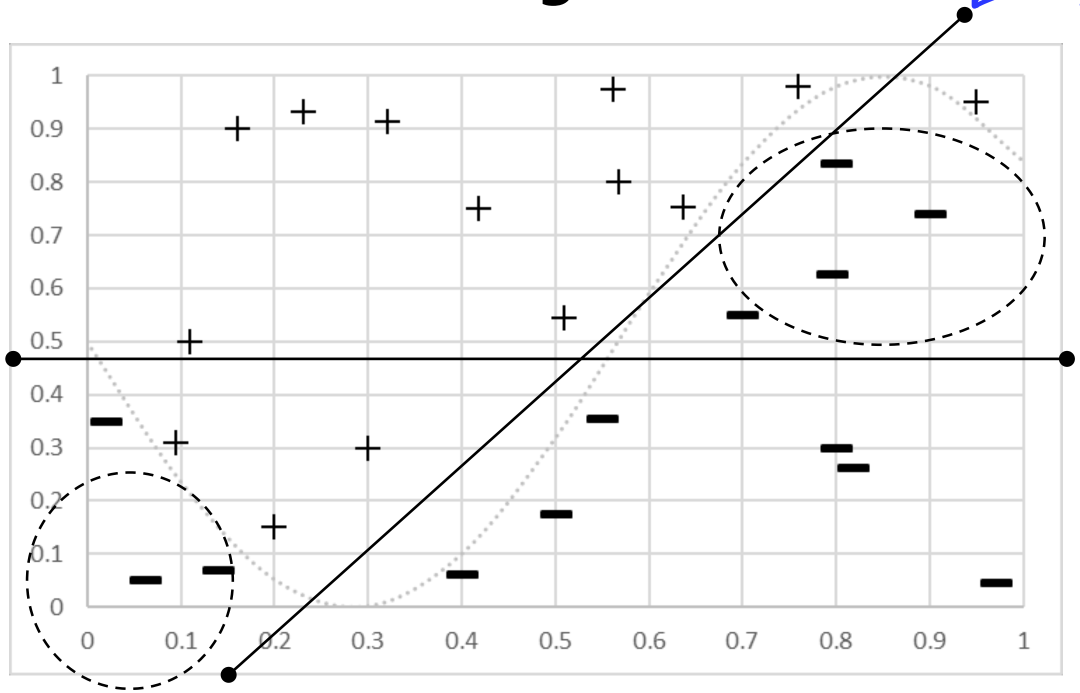
Try a simple model

True concept



Training data

① 선형
효과적 X

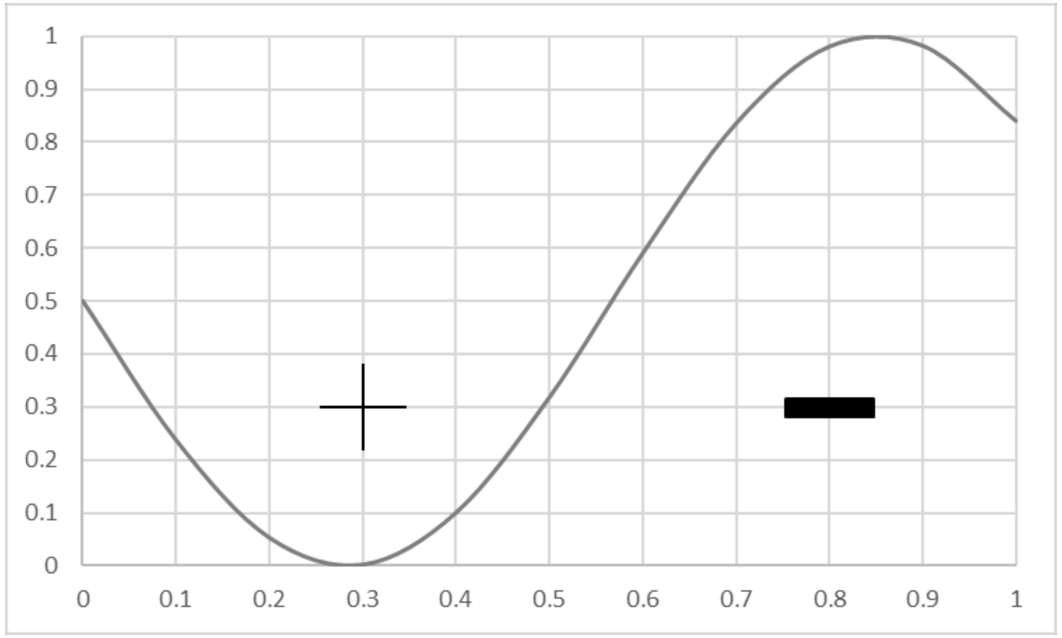


Underfitting problem because of using too simpler model than actual data distribution (high bias)

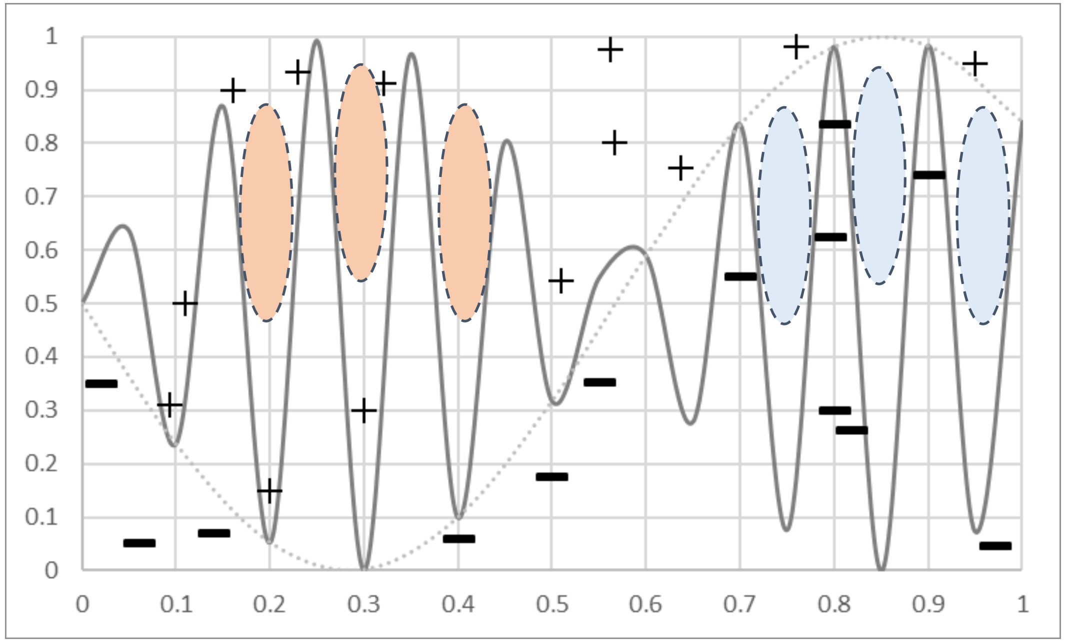
Overfitting

Try a complex model

True concept



Training data



Overfitting problem because of using more complex model than actual data distribution (high variance)

Bias-variance trade-off

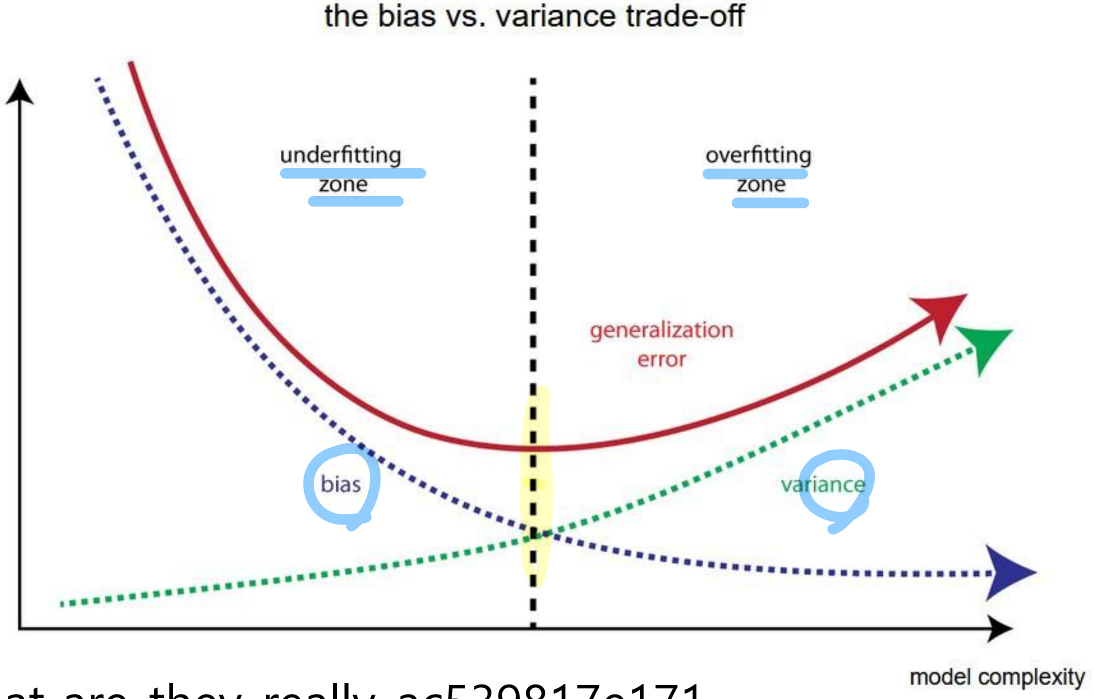
Simple model is better

$$\overbrace{E_{test} \approx \underbrace{E_{train}}_{}}^{} \approx 0$$

Complex model is better

- Split into two objectives:
  1. $E_{test} \approx E_{train}$
  2. $E_{train} \approx 0$
- Objective 1: make $E_{test} \approx E_{train}$
  - Failure : overfitting → high variance and low bias
  - If a model is too complex
- Objective 2: make $E_{train} \approx 0$
  - Failure : underfitting → high bias and low variance
  - If a model is too simple

- The two objectives have trade-off between approximation and generalization w.r.t model complexity

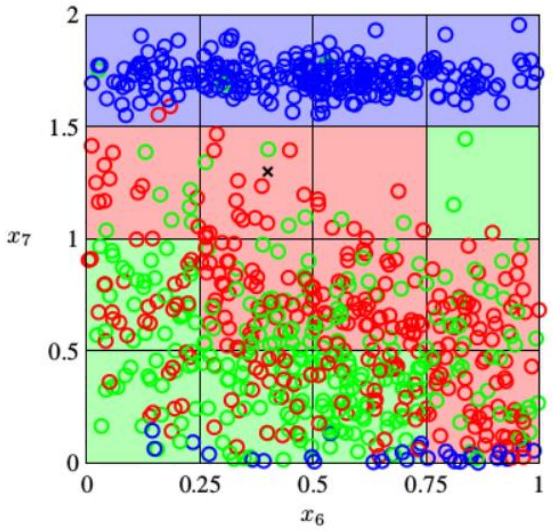


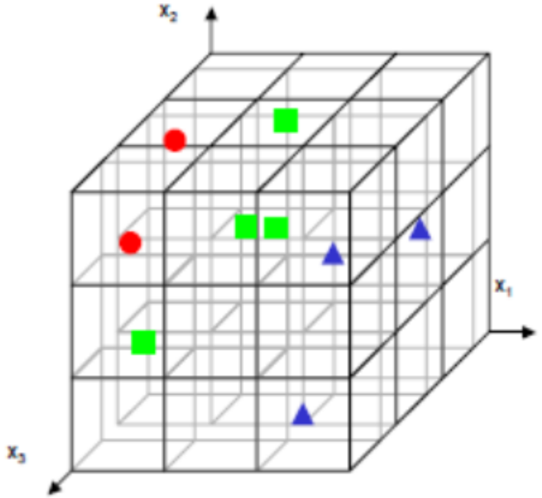
https://towardsdatascience.com/bias-and-variance-but-what-are-they-really-ac539817e171

복잡도 > 데이터수 : overfitting ⇑

Avoid overfitting

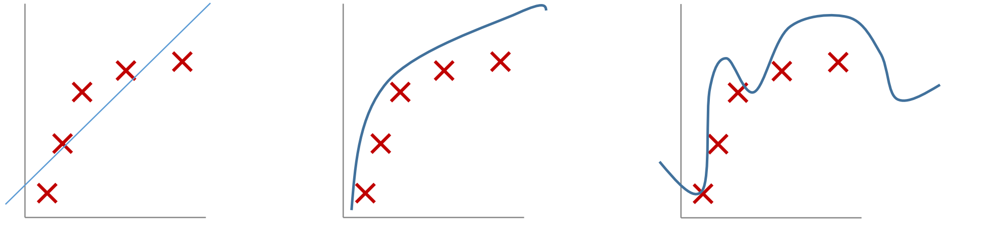
- **(Problem)** In today's ML problems, a complex model tends to be used to handle high-dimensional data (and relatively insufficient number of data) ; prone to an overfitting problem
- **(Curse of dimension)** Will you increase the dimension of the data to improve the performance as well as maintain the density of the examples per bin? If so, you need to increase the data exponentially.





Avoid overfitting

- **(Remedy)**
  - ✓**Data augmentation**
  - 1 ✓Regularization to penalize complex models (variance reduction) ; make a model not too sensitive to noise or outliers (e.g. drop-out, LASSO)
  - 2 ✓Ensemble : average over a number of models



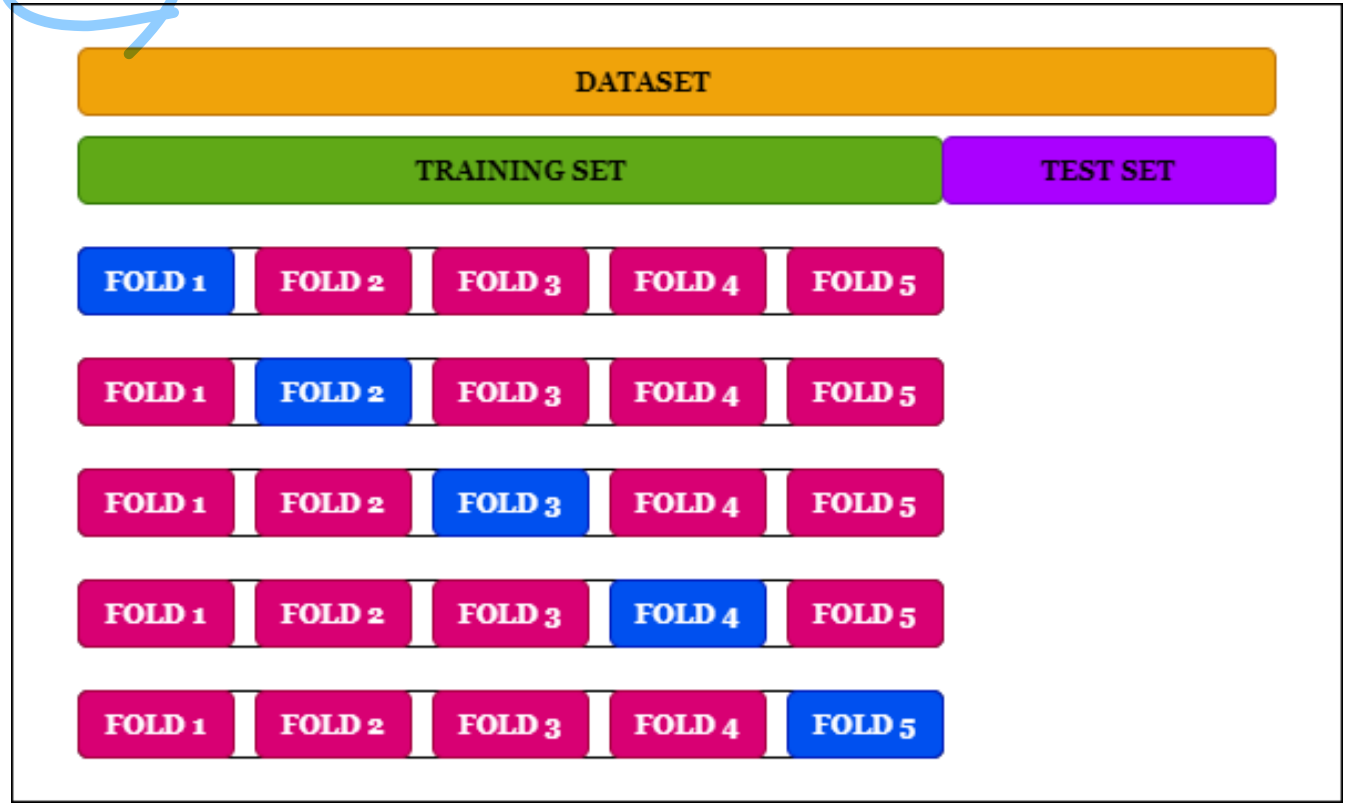
$\theta_0 + \theta_1 x$ $\quad$ $\theta_0 + \theta_1 x + \theta_2 x^2$ $\quad$ $\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \boxed{\lambda \sum_{j=1}^{n} \theta_j^2}\right]$$

$$\min_\theta J(\theta)$$

Cross-validation (CV)

k-fold cross-validation



- Training data set - used to train a model to fit data
- Validation data set – used to provide unbiased evaluation of the model's fitness
- Test data set – never been used in the training

cross-validation allows a better model to avoid overfitting (but more complexity)

Quiz

1. What are two examples of classification? Select two answers.

A. Determine when a heater is on or off based on weather data

B. Translate the numbers or letters from a handwritten message to ASCII text

C. Develop a mathematical relationship between heater level (0-100%) and temperature (20-70°C)

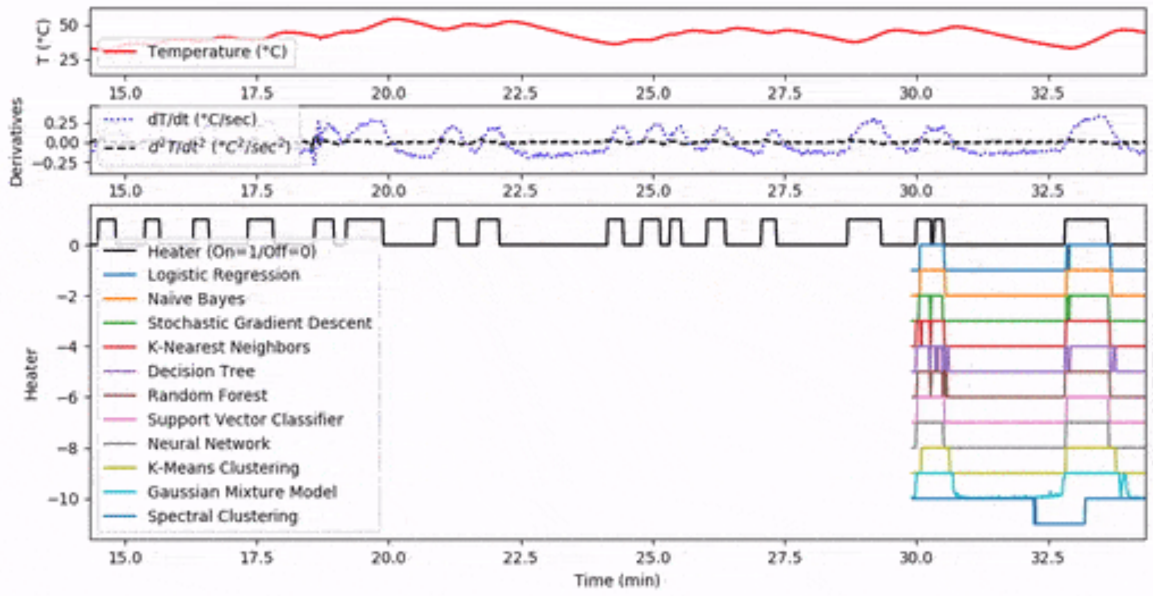
https://apmonitor.com/pdc/index.php/Main/QuizMachineLearning

Quiz

1. What are two examples of classification? Select two answers.

A. Determine when a heater is on or off based on weather data

Correct. The classifier distinguishes between on or off with temperature and temperature derivatives as the features.

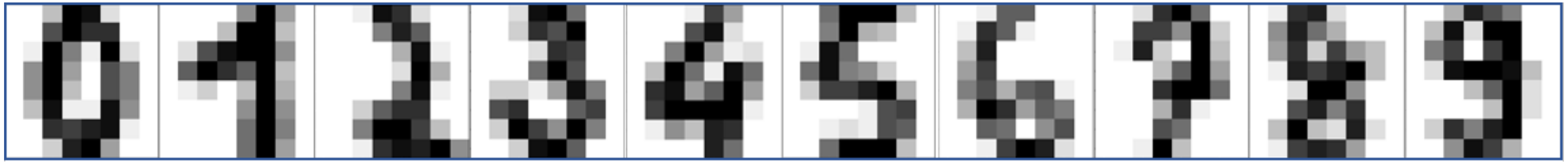


Quiz

1. What are two examples of classification? Select two answers.

B. Translate the numbers or letters from a handwritten message to ASCII text

Correct. The classifier analyzes the pixels of each letter to determine the alpha-numeric value.



Quiz

2. What answers are correct for supervised learning? Select all that apply.

A. Requires labeled data that reveals the measured or true outcome

B. Training and test samples can be overlapped

Quiz

2. What answers are correct for supervised learning? Select all that apply.

A. Requires labeled data that reveals the measured or true outcome

Correct. Supervised learning uses labeled data to compute an error with a model output.

B. Training and test samples can be overlapped

False. Training and test samples must not be overlapped

Summary

- **Introduction to supervised learning**
  - Regression and classification
  - Learning pipeline of a supervised learning
    - Learning from data (error)
  - Overfitting VS underfitting (Bias-variance trade-off)
  - Model generalization
    - Avoid overfitting and cross validation

Reference

- Book: Pattern Recognition and Machine Learning (by Christopher M. Bishop)
- Book: Machine Learning: a Probabilistic Perspective (by Kevin P. Murphy)
- https://www.andrewng.org/courses/