Statistical Machine Learning

2주차

담당: 15기 염윤석

1.What is Supervised Learning?

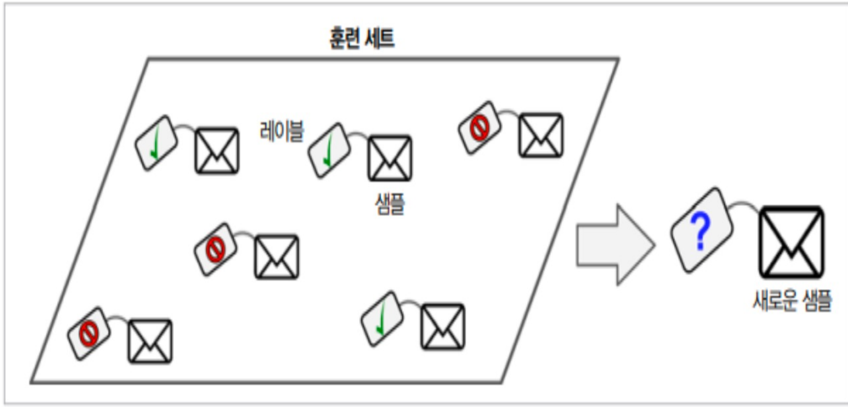
2. Train model

3. Model Selection

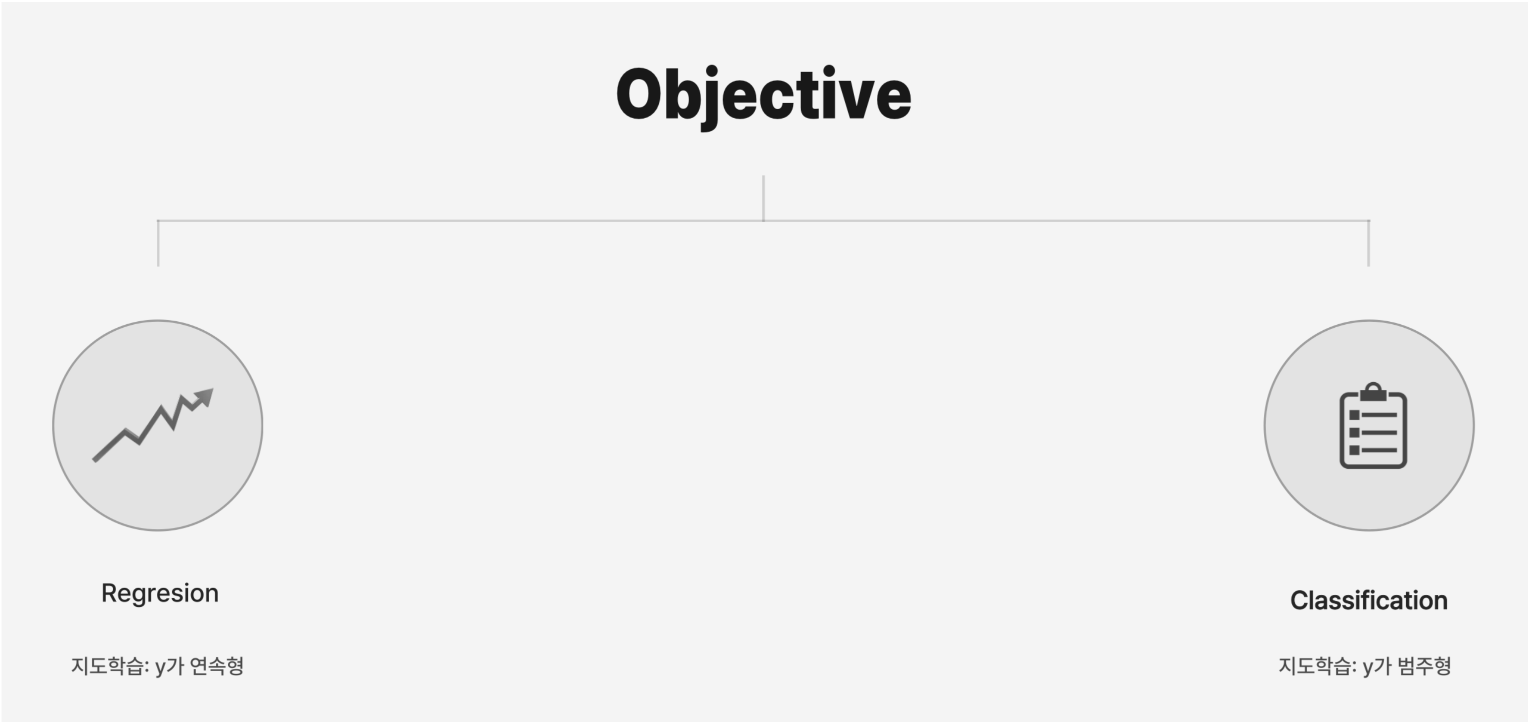
1. What is Supervised Learning?

Supervised Learning

Supervised Learning vs Unsupervised Learning



Supervised Learning



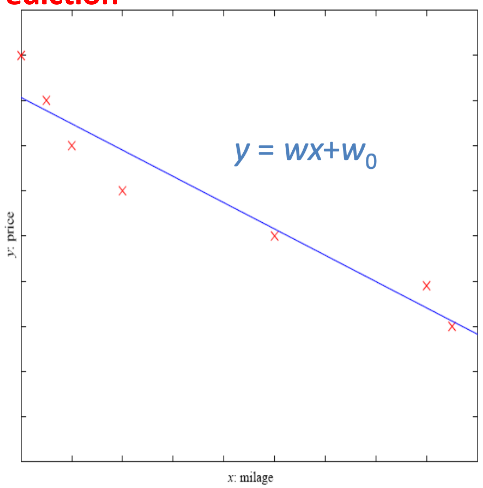
Supervised Learning

Objective

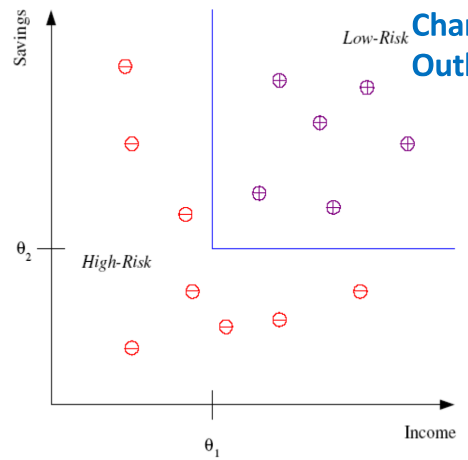
Regression

Classification

Price of car used
Housing price prediction



Face Recognition
Character Recognition
Outlier / Novelty Detection



Supervised Learning

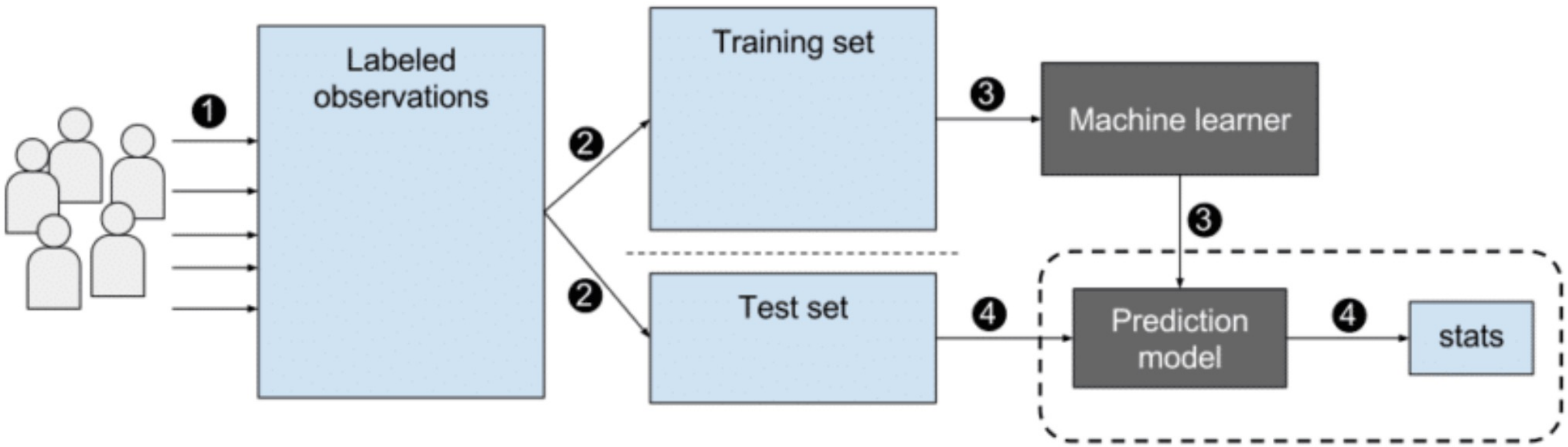
동일한 항목의 많은 재고가 있습니다. 다음 3달 동안 이러한 항목 중 몇 개가 판매될지 예측하려고 합니다.

→ Regression or Classification

당신은 개별 고객 계정을 검사하는 소프트웨어를 만들려고 합니다.
각 계정에 대해 해킹/손상 여부를 결정하는 기능을 만들려고 합니다.

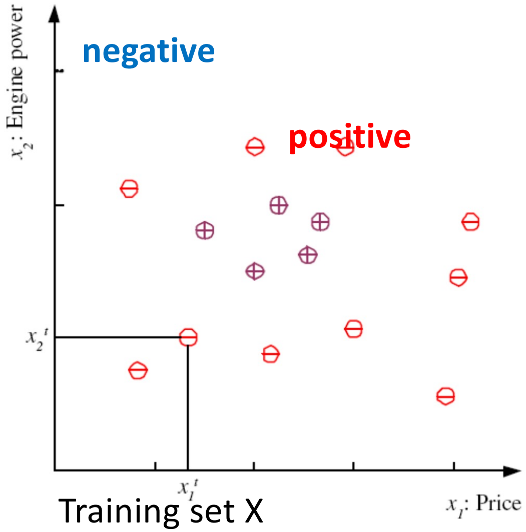
→ Regression or Classification

Supervised Learning



2. Train Model

Learning a Class



Training set X

Class C : Family car

→ Is this car x a family car? = classification task

Input representation:

X_1 :price, X_2 : engine power

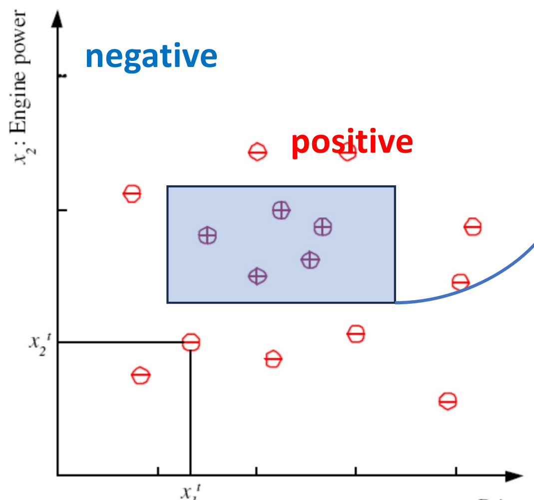
"Learning a Class"

= input feature을 통해서 class를 서술하는 것

Output:

positive(+) or negative(-)

Learning a Class



Training set X

$(p_1 \le \text{price} \le p_2)$ & $(e_1 \le \text{engine power} \le e_2)$

학습의 목표 : Class description → example classification

Inductive Bias:
- 학습이 가능도록 하기 위한 장치 → Aligned Rectangle
- 학습 시에는 만나보지 않았던 상황에 대하여 정확한 예측을 하기 위해 사용하는 추가적인 가정
- Parameter : {p1, p2, e1, e2}

!! 결국 classification을 위해서 {p1, p2, e1, e2} 만 찾으면 된다 !!

수 많은 {p1, p2, e1, e2} 조합 = Hypothesis H = Assumption = Model

Q. 이 중에서 "최적"의 선택은 어떤 것…?

Dimensions of a Supervised Learner

모델을 훈련한다. = Task를 이행하기 위해서, 훈련 데이터 셋에 **가장 잘 맞도록** 모델 파라미터를 설정

1. Model: $g(\mathbf{x} \mid \theta)$

모델을 설정.

2. Loss function: $E(\theta \mid \mathcal{X}) = \sum_t L\left(r^t, g\left(\mathbf{x}^t \mid \theta\right)\right)$

정답을 통한 모델의 성능을 측정

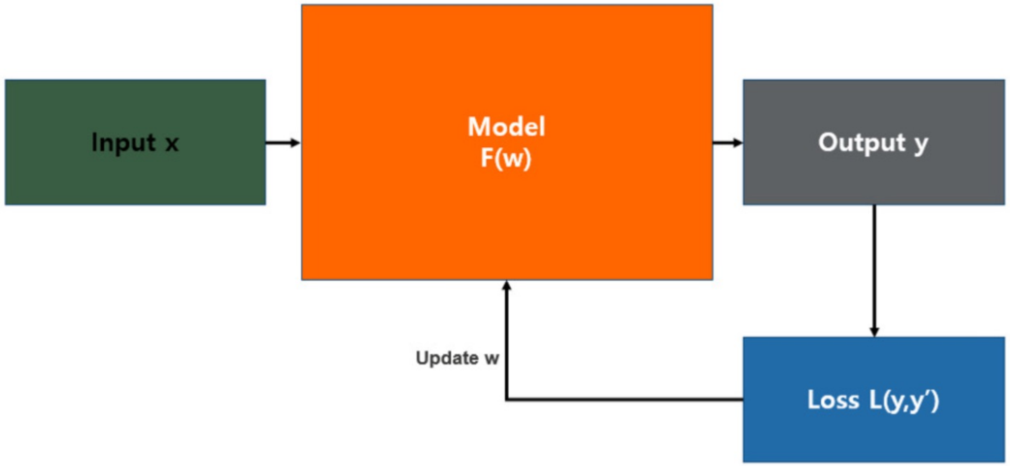
3. Optimization procedure: $\theta^* = \arg\min_\theta E(\theta \mid \mathcal{X})$

정답을 가장 잘 맞추는 모델 파라미터를 찾는다.

Loss Function

- What is Loss Function?　예측값과 실제값(레이블)의 차이를 구하는 기준

  Quantifies the error between output of the algorithm and given target value.



Loss Function

Loss function penalizes bad predictions.

Regression

- Mean Squared Error

$$MSE = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}$$

Others:
Mean absolute error and mean bias error

Classification

- Binary Cross Entropy

$$BCE = -\frac{1}{N}\sum_{i=0}^{N} y_i \cdot \log(\hat{y_i}) + (1 - y_i) \cdot \log(1 - \hat{y_i})$$

- Categorical Cross Entropy

$$CCE = -\frac{1}{N}\sum_{i=0}^{N}\sum_{j=0}^{J} y_j \cdot \log(\hat{y_j}) + (1 - y_j) \cdot \log(1 - \hat{y_j})$$

Others:
Hinge loss / SVM loss.

Dimensions of a Supervised Learner

모델을 훈련한다. = Task를 이행하기 위해서, 훈련 데이터 셋에 **가장 잘 맞도록** 모델 파라미터를 설정

1. Model: $g(\mathbf{x} \mid \theta)$ 모델을 설정.

2. Loss function: $E(\theta \mid \mathcal{X}) = \sum_t L\left(r^t, g\left(\mathbf{x}^t \mid \theta\right)\right)$ 정답을 통한 모델의 성능을 측정

3. Optimization procedure: $\theta^* = \arg\min_\theta E(\theta \mid \mathcal{X})$ 정답을 가장 잘 맞추는 모델 파라미터를 찾는다.

고려해야 할 point!

1. 어떤 모델을 써야할까
2. Task에 맞는 어떤 loss function 을 써야할 까
3. Parameter estimation하는 어떤 estimator 을 써야 할까

Maximum Likelihood Estimator

From Bayes Theorem

모델을 훈련한다. = Task를 이행하기 위해서, 훈련 데이터 셋에 **가장 잘 맞도록** 모델 파라미터를 설정

훈련 데이터 셋에 잘 맞는 파라미터

관측 훈련 데이터 셋이 주어졌을 때, 특정 파라미터의 그럴듯함.

Posterior Probability of parameters

$$P(\theta|X) = \frac{\overbrace{\overbrace{P(X|\theta)}^{\text{MLE}}\, p(\theta)}^{\text{MAP}}}{P(X)}$$

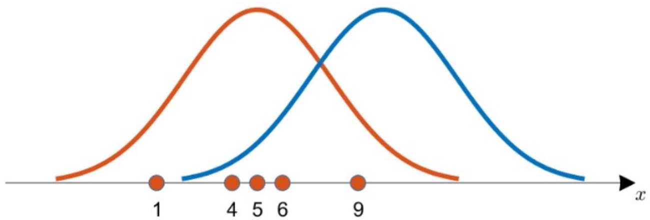
Likelihood Function

$$\underbrace{P(\theta|X)}_{\text{Unknown}} = \frac{P(X|\theta)p(\theta)}{P(X)} \propto \underbrace{P(X|\theta)}_{\text{Likelihood function}}$$

다음과 같이 5개의 데이터를 얻었다고 가정하자.

$$x = \{1, 4, 5, 6, 9\}$$

이 때, 아래의 그림을 봤을 때 데이터 x는 주황색 곡선과 파란색 곡선 중 어떤 곡선으로부터 추출되었을 확률이 더 높을까?



Log Likelihood Function

Definition (Likelihood)

For $X_1, \cdots, X_n \overset{iid}{\sim} f_X(x;\theta)$, where θ denotes a parameter of interest. The likelihood function is

$$L(\theta;\mathbf{X}) = L(\theta; X_1, \cdots, X_n) = \prod_{i=1}^{n} f_X(X_i;\theta)$$

$$\theta_{MLE} = \arg\max_\theta P(X|\theta)$$

$$= \arg\max_\theta \prod_i \underline{P(x_i|\theta)}$$

0~1 사이 값으로 이루어진 확률값들의 곱

→ 0으로 가까워져 버린다.

$$\theta_{MLE} = \arg\max_\theta \log P(X|\theta)$$

$$= \arg\max_\theta \log \prod_i P(x_i|\theta)$$

Log Likelihood function

$$= \arg\max_\theta \sum_i \log P(x_i|\theta)$$

Maximum Likelihood Estimator

- What is MLE?

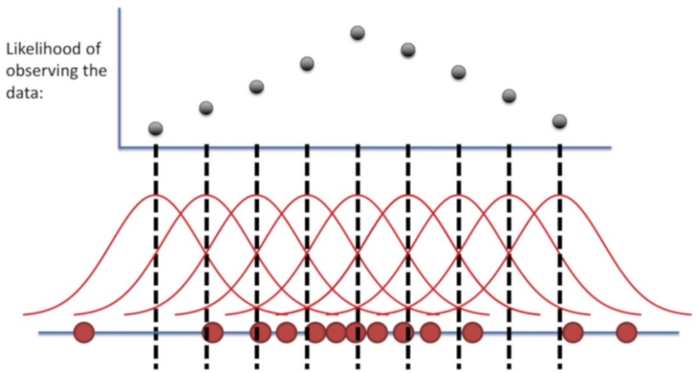
Definition (Maximum likelihood estimator, MLE)

For $X_1, \cdots, X_n \overset{iid}{\sim} f_X(x;\theta)$, the MLE of θ is

$$\hat{\theta}_{MLE} = \underset{\theta}{\operatorname{argmax}}\, L(\theta; \mathbf{x}).$$

which is equivalent to maximize the logarithm of $L(\theta; \mathbf{x})$ which we call the log-likelihood

$$\ell(\theta; \mathbf{x}) = \log L(\theta; \mathbf{x}).$$



Log Likelihood Function

- **Bernoulli distribution**

$$\log L(p) = \sum_{i=1}^{n} (y_i \log p + (1 - y_i) \log (1 - p))$$

- **Binomial distribution**

$$\log L(p) = \log \binom{n}{c} + \sum_{i=1}^{n} (y_i \log p + (1 - y_i) \log (1 - p))$$

- **Multinomial distribution**

$$\log L(p) = \sum_{i=1}^{n} \sum_{j=1}^{c} y_{ij} \log p_j$$

- **Normal distribution**

$$\log L(\mu) \approx - \frac{\sum_{i=1}^{n} (y_i - \mu)}{\sigma^2}$$

MLE → Loss Function

"적합한 파라미터를 찾는" 같은 솔루션을 얻는 방법
Argmax → Argmin으로 문제를 바꿔, penalty의 성격을 띄게 된다.
결국, 손실함수를 최소화하는 parameter를 찾는 문제임

Loss function penalizes bad predictions.

Regression

- Mean Squared Error

$$MSE = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}$$

Others:
Mean absolute error and mean bias error

Classification

- Binary Cross Entropy

$$BCE = -\frac{1}{N}\sum_{i=0}^{N} y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)$$

- Categorical Cross Entropy

$$CCE = -\frac{1}{N}\sum_{i=0}^{N}\sum_{j=0}^{J} y_j \cdot \log(\hat{y}_j) + (1 - y_j) \cdot \log(1 - \hat{y}_j)$$

Others:
Hinge loss / SVM loss.

3. Model Selection

Error

- Error : deviation from an actual value by a prediction or expectation of that value
- Loss function: 모델의 학습 과정에서 최소화되어야 하는 함수로서 모델의 **오류(Error)**를 정량화하는 역할

Error = Variance + Bias

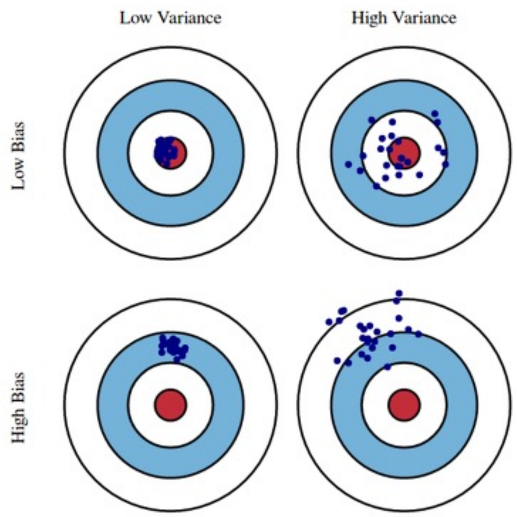
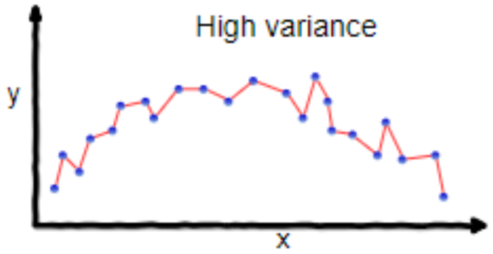
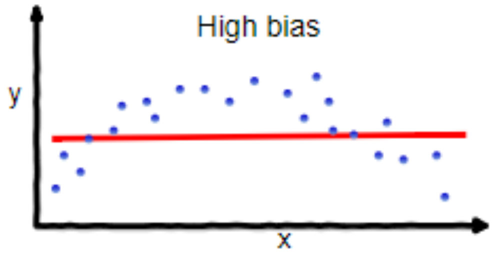


Fig. 1 Graphical illustration of bias and variance.

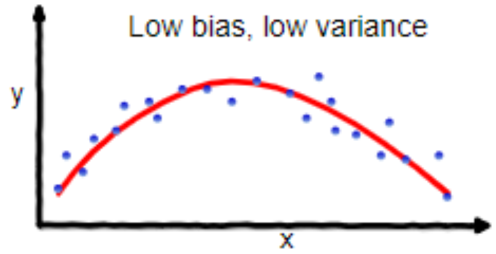
Underfitting vs Overfitting



overfitting

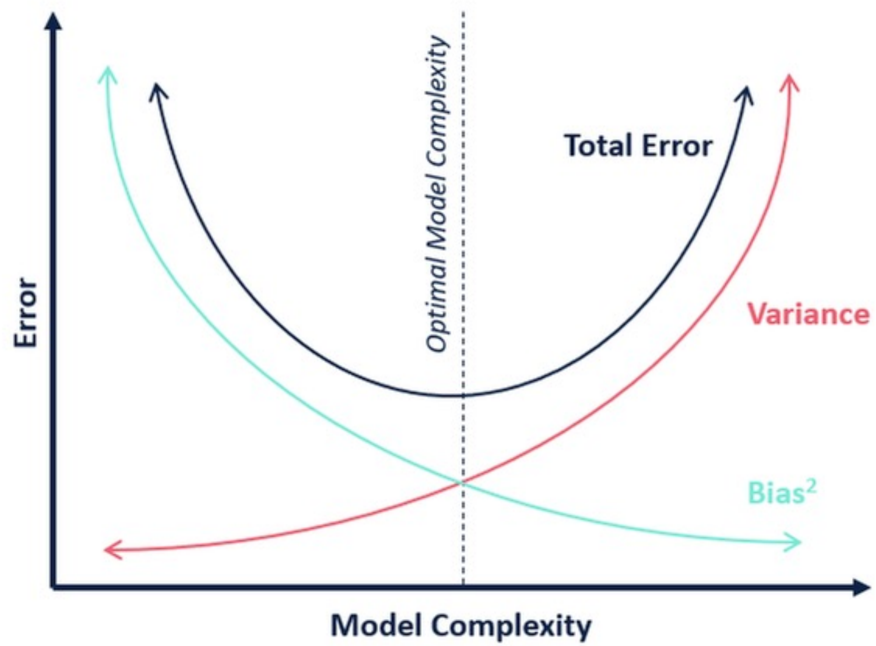


underfitting



Good balance

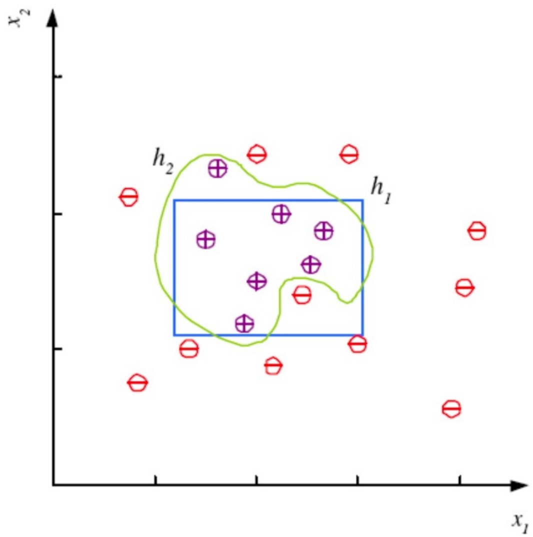
Trade-off



Bias / Variance dilemma : Geman et al. 1992

Model Selection

Inductive Bias : Occam's Razor



<u>If performances are similar,</u>

Use the simpler one because

- Simpler to use (lower computational complexity)
- Easier to train (lower space complexity)
- Easier to explain (more interpretable)
- Generalizes better (lower variance)

Cross Validation

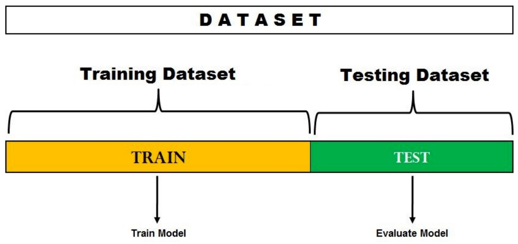
To estimate generalization error, we need data unseen during training.

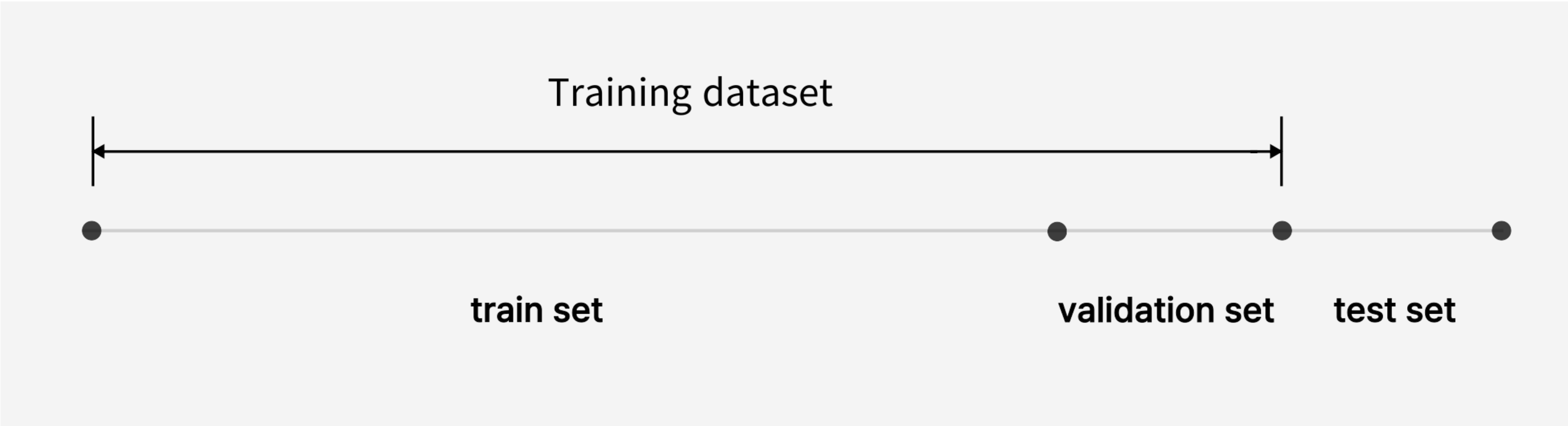
We split the data as

- Training set (50%)
- Validation set (25%)
- Test (publication) set (25%)

Measure generalization accuracy by testing on data unused during training

Hold out





Regularization

Penalize complex models

- E'=error on data + λ*model complexity

* If λ increases, variance decreases, but bias increases

In regression...

Regularization (L2): $$E(\mathbf{w}|\mathcal{X})=\frac{1}{2}\sum_{t=1}^{N}\left[r^t-g\left(x^t|\mathbf{w}\right)\right]^2+\lambda\sum_i w_i^2$$

수고하셨습니다!

해당 세션자료는 KUBIG Github에서 보실 수 있습니다!

다음은 이번 주차 과제 설명이 있습니다!