DA-Final-1) Machine learning 개요, Evaluation measures

1. Supervised learning

1-1. Supervised learning VS Unsupervised learning

↗ HW#4

Supervised learning (e.g. classification, regression)

- The training data (observations, tuples, etc.) are accompanied by labels indicating the ground-truth of the observations.
  (동반하다)
- New data (unlabeled data) is classified or predicted based on the training set.

Unsupervised learning (e.g. clustering) → W14 ~

- The labels of training data is unknown
- To find the underlying structure of data, e.g. clusters of data

1-2. Supervised learning의 예시

① **Classification**

- The target (class) is categorical or finite-discrete.
- E.g., credit loan approval, medical diagnosis, fraud detection
  (신용 대출)

② **Regression** (회귀)

- The target (value) is continuous.
- E.g., weather forecast, stock price prediction
  ex 기온

③ +Ranking → W16

ex User의 선호 예측

1-3. Supervised learning process

~mid ①

- **Data preprocessing**
  - Data cleaning, integration, reduction, transformation (normalization, discretization)
- ② **Training (learning model)**
  - Divide the given data into (1) training, (2) validation (development), and (3) test sets
  - Learn or construct a model from training set
  - The model is represented as mathematical function, decision trees, rules, or etc.
- ③ **Validation (tuning model)**
  - Evaluate the accuracy of the model on validation set
  - Tune the hyper-parameters of the model
- ④ **Test and deploy**
  - Estimate (on test set) the accuracy of the model with the best hyper-parameter values
  - Deploy or predict future or unknown objects using the model

2. Evaluation measures

2-1. Evaluating the learning methods

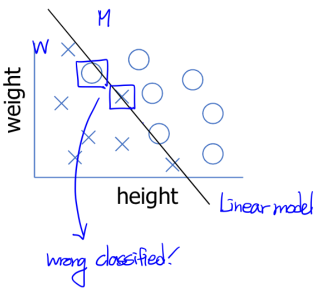
- **Accuracy**
  - How accurate is the learned model?
  - How well does the learned model generalize?
- **Speed**
  - Training time: Time to learn a model
  - Test time: Time to predict a new data
- **Interpretability**
  - The model is understandable or interpretable?

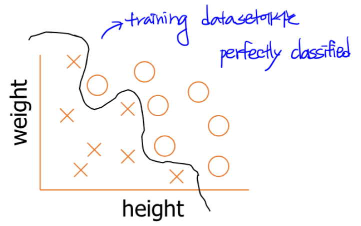
2-2. Evaluation issue - Overfitting

training set → model 학습 → training set에 overfitting?

- Fitting the model exactly to the data is usually not a good idea.
- The resulting model may not generalize well to unseen data.

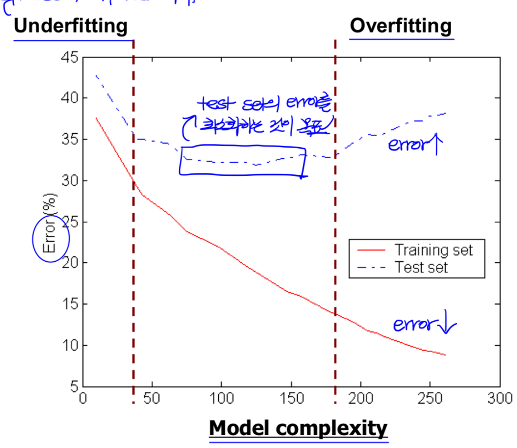
	Training	Test
acc.	↑	↓





→ 하지만, test, unseen data에서는 정확도가 떨어질 수 있다.
: Overfitting

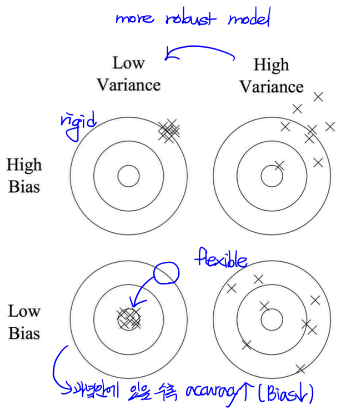
model이 너무 weak하다.



≈ parameter의 수 (cf) Deep learning은 차이가 있다)

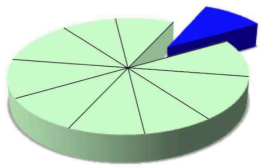
2-3. ML model의 generalization error

- $E(y - f(x))^2 = \mathrm{Var}(f) + \mathrm{Bias}(f)^2 + \mathrm{Var}(\epsilon)$
- Generalization error = Variance + Bias^2
- (분산) Variance: the amount by which f would change if we estimated it using a different training set
- (편향) Bias: the error that is introduced by approximating a real-life problem, which may be extremely complicated, by a much simpler model. test set보다 단순한 model로 인한 error
- There is a trade-off between bias and variance:
  - Flexible models: low bias but high variance
  - Rigid models: high bias but low variance



2-4. Overfitting을 방지하는 Validation methods, Test set

- **Holdout test** ⟨Hyperparameter tuning⟩
  (지양)
  - Given data is randomly partitioned into two independent sets
    - Training set (e.g. 2/3) for model construction
    - Validation set (e.g. 1/3) for accuracy estimation
- ***k*-fold cross-validation** (e.g. $k = 10$) 결국 모든 data를 test에 사용 more reliable
  (접는 부분)
  - Randomly partition the data into k *mutually exclusive* subsets, each approximately equal size
  - At i-th iteration, use D_i as validation set and others as training set
  - Repeat k times, each with different D_i for validation set
  - (계층화된) Stratified cross-validation: folds are stratified so that class distribution in each fold is approximately the same as that in the entire data (class distribution 보존)



- **Leave-one-out test**:
  - Special case of k-fold cross-validation where k = # of tuples (D_i size = 1)
  - Most stable but most inefficient

- Hyperparameters, tuned on validation set, could overfit to validation set.
- Need another set (i.e. test set) to estimate the "true" generalization error

learn parameter

tune hyperparameters

set size↓ (%보다 개수로 판단
1M 中 1000개 정도)

2-5. Evaluation measures

Measures for classification and ranking

- Confusion matrix, Accuracy, F1-score, AUC
- MAP, NDCG

Measures for regression

- MSE, RMSE, MAE, MAPE

2-5-(1). Classification - Confusion matrix, Accuracy

ex. Binary classification (P/N)

		PREDICTED CLASS	
		Positive	Negative
ACTUAL CLASS	Positive	TP	FN loss↑ 줄여주기!
	Negative	FP	TN

TP: True Positive
FN: False Negative
FP: False Positive
TN: True Negative

$$\textbf{Accuracy} = \frac{\text{True predictions}(\text{TP} + \text{TN})}{\text{Total}(\text{TP} + \text{TN} + \text{FP} + \text{FN})}$$

- Accuracy might not be the best measure.
  - E.g., two-class problem where |P|=10, |N|=9990
  - Model predicting everything to be N: Accuracy $= \frac{9990}{10000} = 99.9\%$
- Cost sensitive learning: Put different costs for TP, FN, FP, and TN, and learn to minimize the overall cost.
- Confusion matrix for multi-class classification? A, B, C

	A	B	C
A	T	F	F
B	F	T	F
C	F	F	T

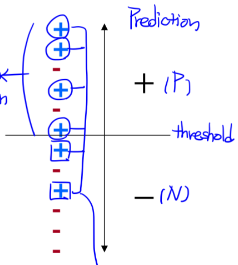
2-5-(2). Classification - Precision, Recall, F1-score, MAP

ex. binary classification

- **Precision** $p = \frac{\text{TP}}{\text{TP+FP}}$ (Precision is biased towards TP & FP)
- **Recall** $r = \frac{\text{TP}}{\text{TP+FN}}$ (Recall is biased towards TP & FN) $p \propto r$
- **F1-score** $= \frac{2rp}{r+p} = \frac{2\text{TP}}{2\text{TP+FP+FN}}$ (F-measure is biased towards all except TN)

p.r의 balanced avg.

- E.g. What is the precision and recall of the right example?
- *For multi-class classification*: Compute F-score for each class as positive and average them.
  the others → negative
- **MAP (Mean Average Precision)** is used for retrieval system (e.g. search engine). Why?



Precision (+ prediction의 정확도) : $\frac{4}{6}$

recall (+ data의 classification 정확도) : $\frac{4}{6}$

여러 threshold값에 따른 precision의 평균값

→ recall은 알 수 없으므로 F1보다 p를 선호!

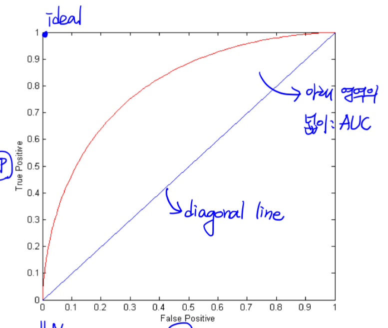
(top data 多)

놓치는 data가 없어야 하는 search engine에서는 중요!
ex. 특허 검색

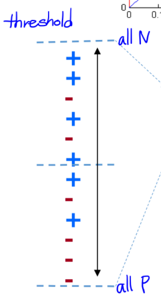
P	r	F1
0.9	0.7	$\frac{63}{16}$
		∧
0.8	0.8	$\frac{64}{16}$ (more balanced)

2-5-(3). Classification - Sensitivity, Specificity, ROC, AUC

- Alternative measures (for medical domain, e.g. cancer diagnosis)
  - **Sensitivity** = TP / (TP+FN) (= recall) >> precision 보다 더 중요! in medical domain
  - **Specificity** = TN / (TN+FP) Negative data의 classification 정확도



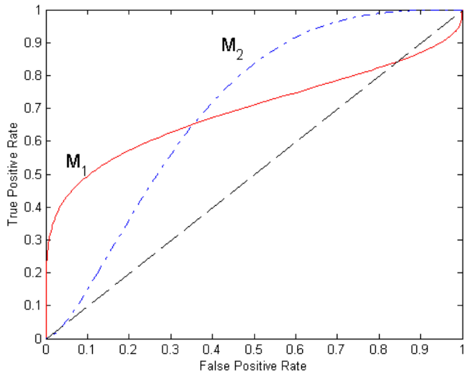
- **ROC curve** plots TP rate (on the y-axis) against FP rate (on the x-axis)
  - TP rate = TP/P (= sensitivity)
  - FP rate = FP/N (= 1 – specificity) $\leftarrow 1-\frac{TN}{TN+FP} = \frac{FP}{N}$
- Most classification methods provide a threshold that can control the tradeoff between TP and FP
  - Performance of a classifier represented as a point on the ROC curve
  - Changing the threshold of algorithm changes the location of the point



(TP,FP): FP

- (0,0): declare everything to be negative class TP X, FP X
- (1,1): declare everything to be positive class TN X, FN X
- (1,0): ideal
- Diagonal line: random guessing

- **Area Under the ROC Curve (AUC)**
  - Another metric for evaluating classification performance
  - Ideal: Area = 1
  - Random guess: Area = 0.5
- M_1 is better for small FP
- M_2 is better for large FP



⇒ Binary classification : 다양한 평가 지표 사용 가능!
ㄴ 그 중 AUC가 reliable! but, proper threshold 제안 X

2-5-(4). Ranking - NDCG

CG (Cumulative Gain)

- Sum of Relevance for top-p items: $CG_p = \sum_{i=1}^{p} rel_i$

DCG (Discounted Cumulative Gain)

- Sum of *discounted* Relevance for top-p items: $DCG_p = \sum_{i=1}^{p} \frac{rel_i}{\log_2(i+1)}$ or $\sum_{i=1}^{p} \frac{2^{rel_i}-1}{\log_2(i+1)}$

(i↓, ranking↑, weight↑)

NDCG (Normalized Discounted Cumulative Gain) ← recommendation

- Normalized DCG, which normalizes DCG regardless of p: $NDCG_p = \frac{DCG_p}{IDCG_p}$, where $IDCG_p = \cdots$

암기X

2-5-(5). Regression - MSE, RMSE, MAE

MSE (Mean Square Error):

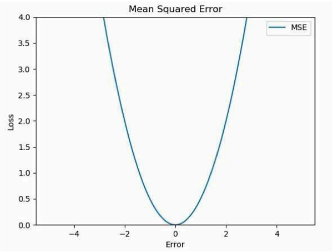
$$\frac{1}{N}\sum_{i}^{N}(\underbrace{Pred_i - Target_i}_{\text{residual}})^2$$

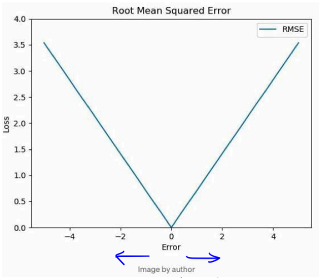
RMSE (Root Mean Square Error):

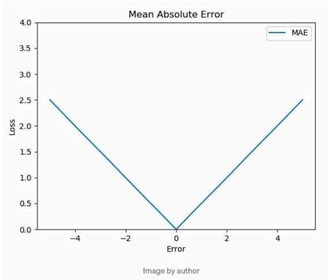
$$\sqrt{\frac{1}{N}\sum_{i}^{N}(Pred_i - Target_i)^2}$$

MAE (Mean Absolute Error):

$$\frac{1}{N}\sum_{i}^{N}|Pred_i - Target_i|$$







MSE는 quadratically 증가.

linearly 증가

→좀 더 직관적으로 분석 가능!

- MAPE (Mean Absolute Percentage Error): $\frac{100}{N}\sum_{i}^{N}\frac{|pre\ d_i - targe\ t_i|}{targe\ t_i}$: weighting depending on target? Not practical!

→큰 값의 error에 집중 / bias↑