클래스 분류 (2)

과목명: 데이터사이언스

AI융합학부 박건우

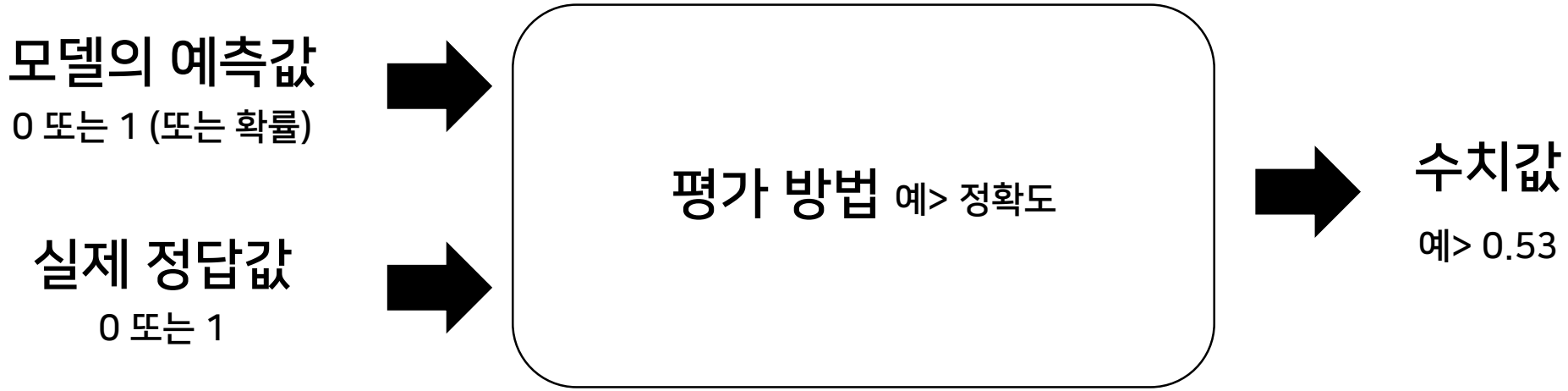
Contents

Model Evaluation and Selection

- Q. Which classification model should be selected for data mining?

- Things to consider
  - Which metrics can be used to evaluate model performance?
  - Validation set is required for model comparison. How can we construct a set?
  - How can we determine a model is better than another?

Evaluation Metrics



Confusion Matrix (혼동 행렬)

- Confusion Matrix:

Actual class\Predicted class	C_1	¬ C_1
C_1	**True Positives (TP)**	**False Negatives (FN)**
¬ C_1	**False Positives (FP)**	**True Negatives (TN)**

- An example:

Actual class\Predicted class	buy_computer = yes	buy_computer = no	Total
buy_computer = yes	**6954**	**46**	7000
buy_computer = no	**412**	**2588**	3000
Total	7366	2634	10000

- Given *m* classes, an entry, $\boldsymbol{CM_{i,j}}$ in a **confusion matrix** indicates # of tuples in class *i* that were labeled by the classifier as class *j*
- May have extra rows/columns to provide totals

Classifier Evaluation Metrics:
Accuracy, Error Rate

A\P	C	¬C	
C	**TP**	**FN**	**P**
¬C	**FP**	**TN**	**N**
	P'	**N'**	**All**

- **Classifier Accuracy** (정확도), or recognition rate: percentage of test set tuples that are correctly classified

  Accuracy = (TP + TN)/All

- **Error rate** (오류율): *1 – accuracy*, or

  Error rate = (FP + FN)/All

Classifier Evaluation Metrics:
Sensitivity and Specificity

Suppose a classifier
always predicting a negative class

A\P	C	¬C	
C	**0**	**10**	**P**
¬C	**0**	**90**	**N**
	P'	**N'**	**All**

Accuracy = 90/100 = **0.9** (?!)

- **Class Imbalance Problem**:
  - One class may be rare,
    e.g. fraud, or HIV-positive
  - Significant majority of the negative class and minority of the positive class
  - Accuracy is a poor measure for an **imbalanced** dataset
- We need to have a diverse view
  - **Sensitivity (민감도)**: True Positive recognition rate
    - **Sensitivity = TP/P**
  - **Specificity (특이도)**: True Negative recognition rate
    - **Specificity = TN/N**

Classification Evaluation Metrics:
Precision and Recall, and F-measures

- **Precision (정밀도)**: exactness – what % of tuples that the classifier labeled as positive are actually positive
- **Recall (재현율)**: completeness – what % of positive tuples did the classifier label as positive?
  - Perfect score is 1.0
  - Inverse relationship between precision & recall

$$precision = \frac{TP}{TP + FP} \quad recall = \frac{TP}{TP + FN}$$

- **F measure (F_1 or F-score)**: harmonic mean of precision and recall,

$$F = \frac{2 \times precision \times recall}{precision + recall}$$

- $F_ß$: weighted measure of precision and recall
  - assigns ß times as much weight to recall as to precision

$$F_\beta = \frac{(1 + \beta^2) \times precision \times recall}{\beta^2 \times precision + recall}$$

Classifier Evaluation Metrics: Example

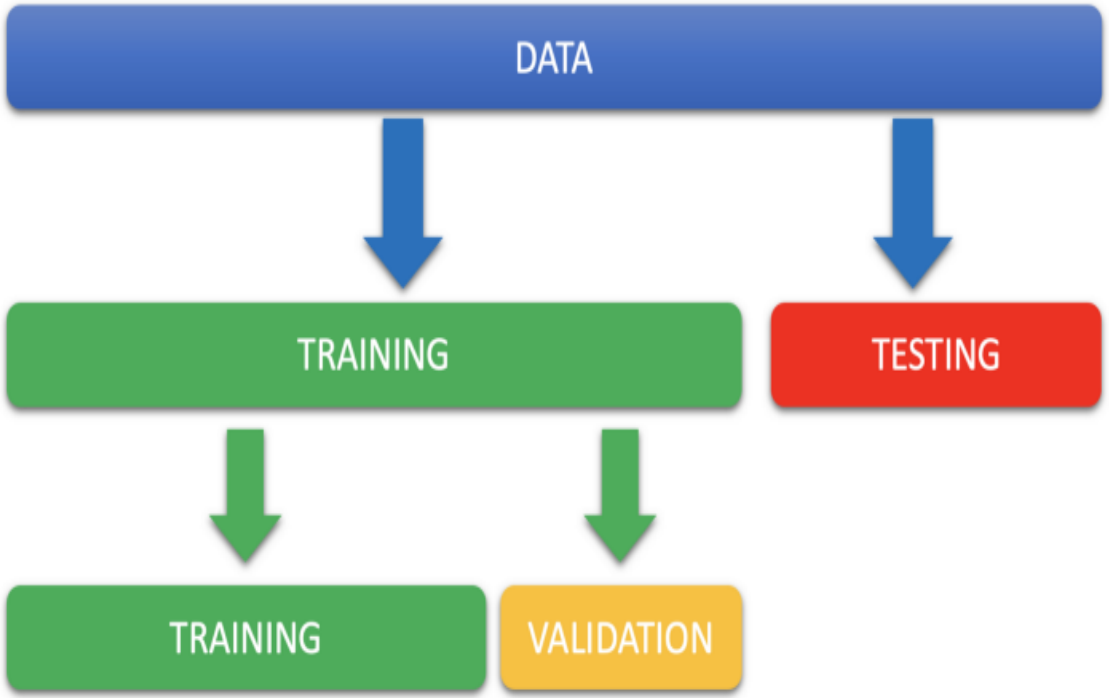
Actual Class\Predicted class	cancer = yes	cancer = no	Total	Recognition(%)
cancer = yes	**90**	**210**	300	30.00 (*sensitivity*
cancer = no	**140**	**9560**	9700	98.56 (*specificity)*
Total	230	9770	10000	96.40 (*accuracy*)

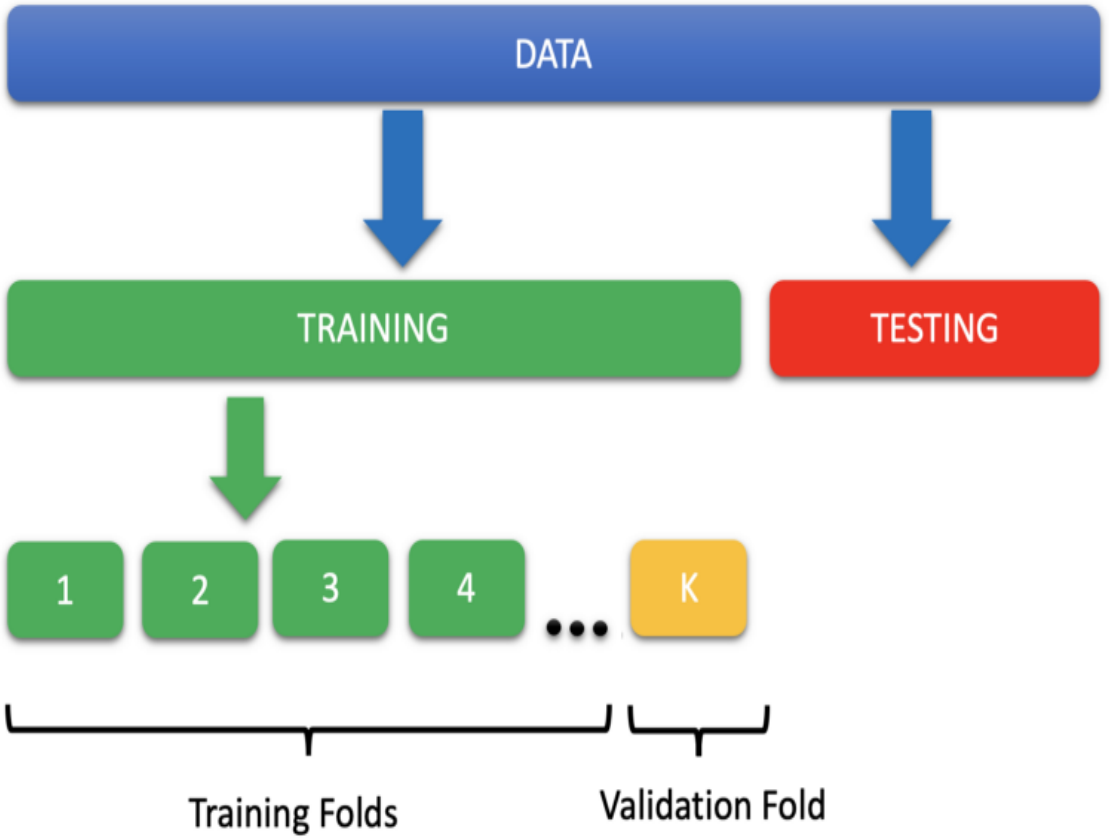
- *Precision* = 90/230 = 0.3913 *Recall* = 90/300 = 0.3
- F_1?

Evaluating Classifier Accuracy:
Holdout & Cross-Validation Methods

- **Holdout method**
  - Given data is randomly partitioned into two independent sets
    - Training set (e.g., 2/3) for model construction
    - Test set (e.g., 1/3) for accuracy estimation
  - Repeated holdout
    - Repeat holdout k times, accuracy = avg. of the accuracies obtained
- **Cross-validation** (k-fold, where k = 10 is most popular)
  - Randomly partition the data into k mutually exclusive subsets, each approximately equal size
  - At i-th iteration, use D_i as test set and others as training set
  - **Leave-one-out**: k folds where k = # of tuples, for small sized data
  - ***Stratified cross-validation***: folds are stratified so that class dist. in each fold is approx. the same as that in the initial data

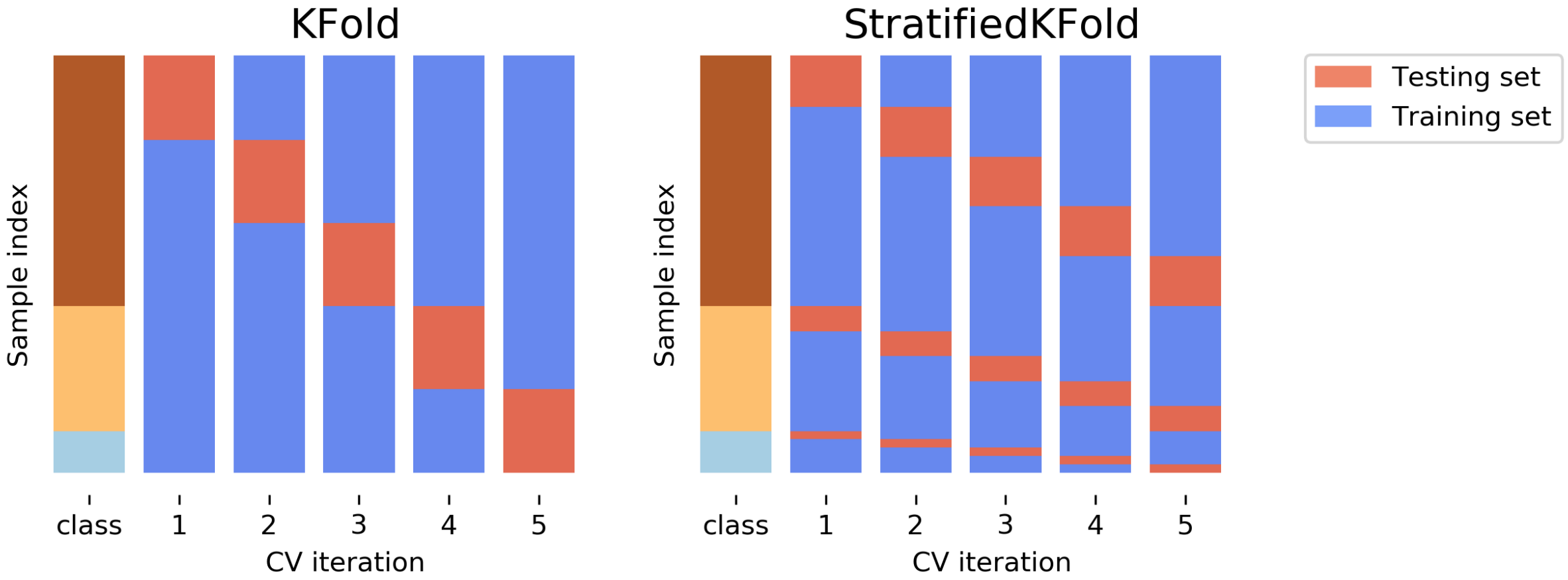
Holdout vs. K-Fold





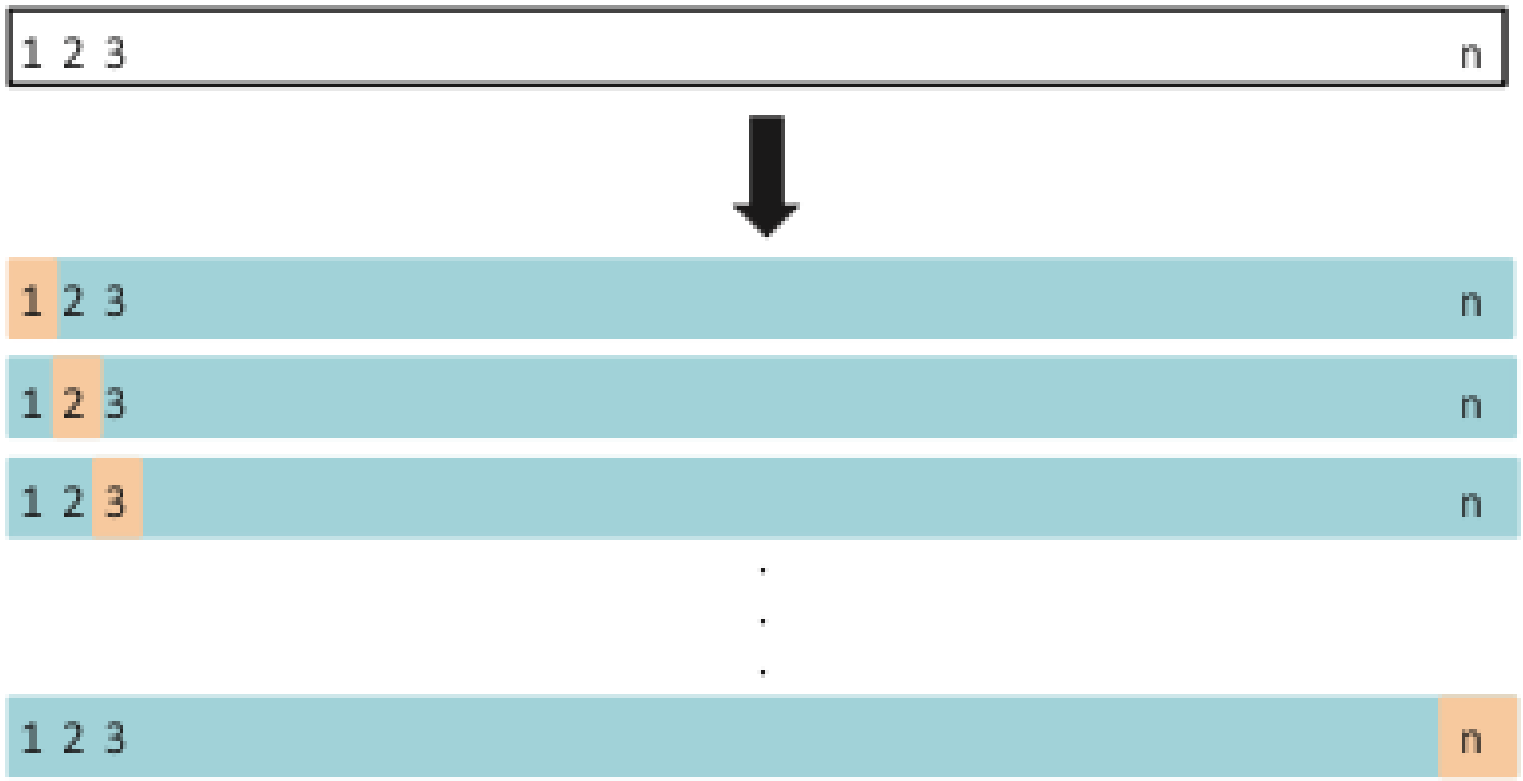
Source: AI Graduate

K-Fold vs. Stratified K-Fold



Source: Andreas Mueller

Leave-One-Out Cross Validation



Source: Jacob's Coding Playground

Model Evaluation and Selection

- Q. Which classification model should be selected for data mining?

- Things to consider
  - Which metrics can be used to evaluate model performance?
  - Validation set is required for model comparison. How can we construct a set?
  - How can we determine a model is better than another?

Estimating Confidence Intervals:
Classifier Models M_1 vs. M_2

- Suppose we have 2 classifiers, M_1 and M_2, which one is better?
- Use 10-fold cross-validation to obtain $\overline{err}(M_1)$ and $\overline{err}(M_2)$
- These mean error rates are just *estimates* of error on the true population of *future* data cases
- What if the difference between the 2 error rates is just attributed to *chance*?
  → Use a **test of statistical significance**

Estimating Confidence Intervals:
Null Hypothesis

- Perform 10-fold cross-validation
- Assume samples follow a **t distribution** with k–1 **degrees of freedom** (here, k=10)
- Use **t-test** (or **Student's t-test**)
  - **Null Hypothesis**: M_1 & M_2 are the same
  - If we can **reject** null hypothesis, then
    - we conclude that the difference between M_1 & M_2 is **statistically significant**
    - we choose model with lower error rate

Estimating Confidence Intervals:
t-test

- Using the same (validation) test set: **pairwise comparison**
  - For i^{th} round of 10-fold cross-validation, the same test fold is used to obtain $err(M_1)_i$ and $err(M_2)_i$
  - Average over 10 rounds to get $\overline{err}(M_1)$ and $\overline{err}(M_2)$
  - **t-test** computes **t-statistic** with k-1 **degrees of freedom**:

$$t = \frac{\overline{err}(M_1) - \overline{err}(M_2)}{\sqrt{var(M_1 - M_2)/k}} \quad \text{where} \quad var(M_1 - M_2) = \frac{1}{k}\sum_{i=1}^{k}\left[err(M_1)_i - err(M_2)_i - (\overline{err}(M_1) - \overline{err}(M_2))\right]^2$$

- (참조)
  If two different test sets are used: use **non-paired t-test** where

$$var(M_1 - M_2) = \sqrt{\frac{var(M_1)}{k_1} + \frac{var(M_2)}{k_2}},$$

  where k_1 & k_2 are # of cross-validation samples used for M_1 & M_2, resp.

Estimating Confidence Intervals:
Statistical Significance

- Are M_1 & M_2 **significantly different**?
  - Select significance level (유의 수준): e.g., $sig = 5\%$
  - Consult table for t-distribution:
    Find *t value* corresponding to $k - 1$ *degrees of freedom* (자유도): here, 9
  - Look up value for **confidence limit** $z = sig/2$ (here, 0.025)

    → t-distribution is symmetric: typically upper % points of distribution shown
  - **If t > z or t < -z**, then t value lies in rejection region:
    - Null hypothesis: $\mathrm{err}(M_1)$ and $\mathrm{err}(M_2)$ are the same
    - **Reject null hypothesis** that mean error rates of M_1 & M_2 are same
    - Conclude: statistically significant difference between M_1 & M_2
  - **Otherwise**, conclude that the difference is **by chance**

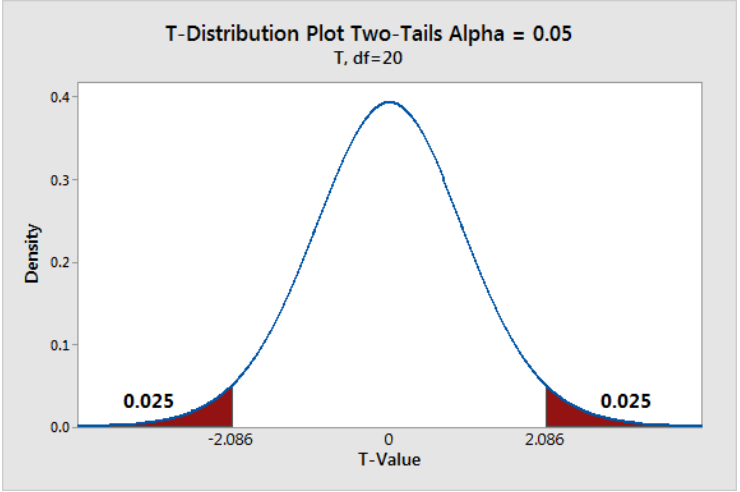


TABLE B: *t*-DISTRIBUTION CRITICAL VALUES

	Tail probability p											
df	.25	.20	.15	.10	.05	.025	.02	.01	.005	.0025	.001	.0005
1	1.000	1.376	1.963	3.078	6.314	12.71	15.89	31.82	63.66	127.3	318.3	636.6
2	.816	1.061	1.386	1.886	2.920	4.303	4.849	6.965	9.925	14.09	22.33	31.60
3	.765	.978	1.250	1.638	2.353	3.182	3.482	4.541	5.841	7.453	10.21	12.92
4	.741	.941	1.190	1.533	2.132	2.776	2.999	3.747	4.604	5.598	7.173	8.610
5	.727	.920	1.156	1.476	2.015	2.571	2.757	3.365	4.032	4.773	5.893	6.869
6	.718	.906	1.134	1.440	1.943	2.447	2.612	3.143	3.707	4.317	5.208	5.959
7	.711	.896	1.119	1.415	1.895	2.365	2.517	2.998	3.499	4.029	4.785	5.408
8	.706	.889	1.108	1.397	1.860	2.306	2.449	2.896	3.355	3.833	4.501	5.041
9	.703	.883	1.100	1.383	1.833	2.262	2.398	2.821	3.250	3.690	4.297	4.781
10	.700	.879	1.093	1.372	1.812	2.228	2.359	2.764	3.169	3.581	4.144	4.587
11	.697	.876	1.088	1.363	1.796	2.201	2.328	2.718	3.106	3.497	4.025	4.437
12	.695	.873	1.083	1.356	1.782	2.179	2.303	2.681	3.055	3.428	3.930	4.318
13	.694	.870	1.079	1.350	1.771	2.160	2.282	2.650	3.012	3.372	3.852	4.221
14	.692	.868	1.076	1.345	1.761	2.145	2.264	2.624	2.977	3.326	3.787	4.140
15	.691	.866	1.074	1.341	1.753	2.131	2.249	2.602	2.947	3.286	3.733	4.073
16	.690	.865	1.071	1.337	1.746	2.120	2.235	2.583	2.921	3.252	3.686	4.015
17	.689	.863	1.069	1.333	1.740	2.110	2.224	2.567	2.898	3.222	3.646	3.965
18	.688	.862	1.067	1.330	1.734	2.101	2.214	2.552	2.878	3.197	3.611	3.922

Recap: 임곗값(threshold)

- 많은 이진 분류 모델의 출력값은, 주어진 샘플이 positive 클래스에 속할 확률로 표현된다.
- 가능한 정답의 개수는 2가지 이므로 (Negative-0 또는 Positive-1), 일반적으로 0.5를 기준값으로 하여 0.5보다 크면 Positive, 0.5보다 작으면 Negative 클래스로 예측한다.
- 이 0.5 값을 임곗값(threshold)이라고 하며 이를 조정하여, 같은 모델 예측값에 대해 실제 예측 클래스를 다르게 할 수 있다.

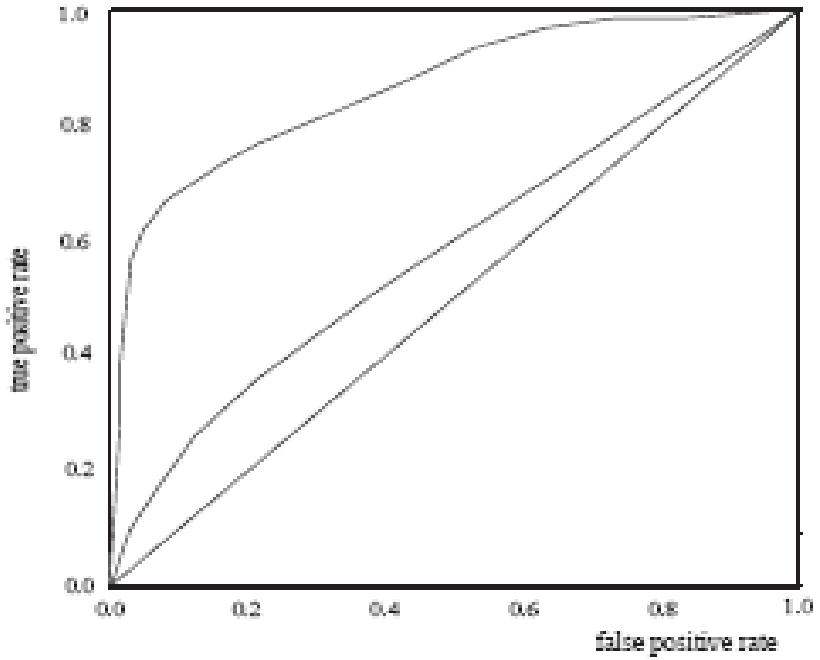
Model Selection: ROC Curves

True Positive Rate

= TP / (TP + FN)

= 재현율(Recall)

= 실제 값이 Positive인 케이스가 정확히 예측된 정도



False Positive Rate

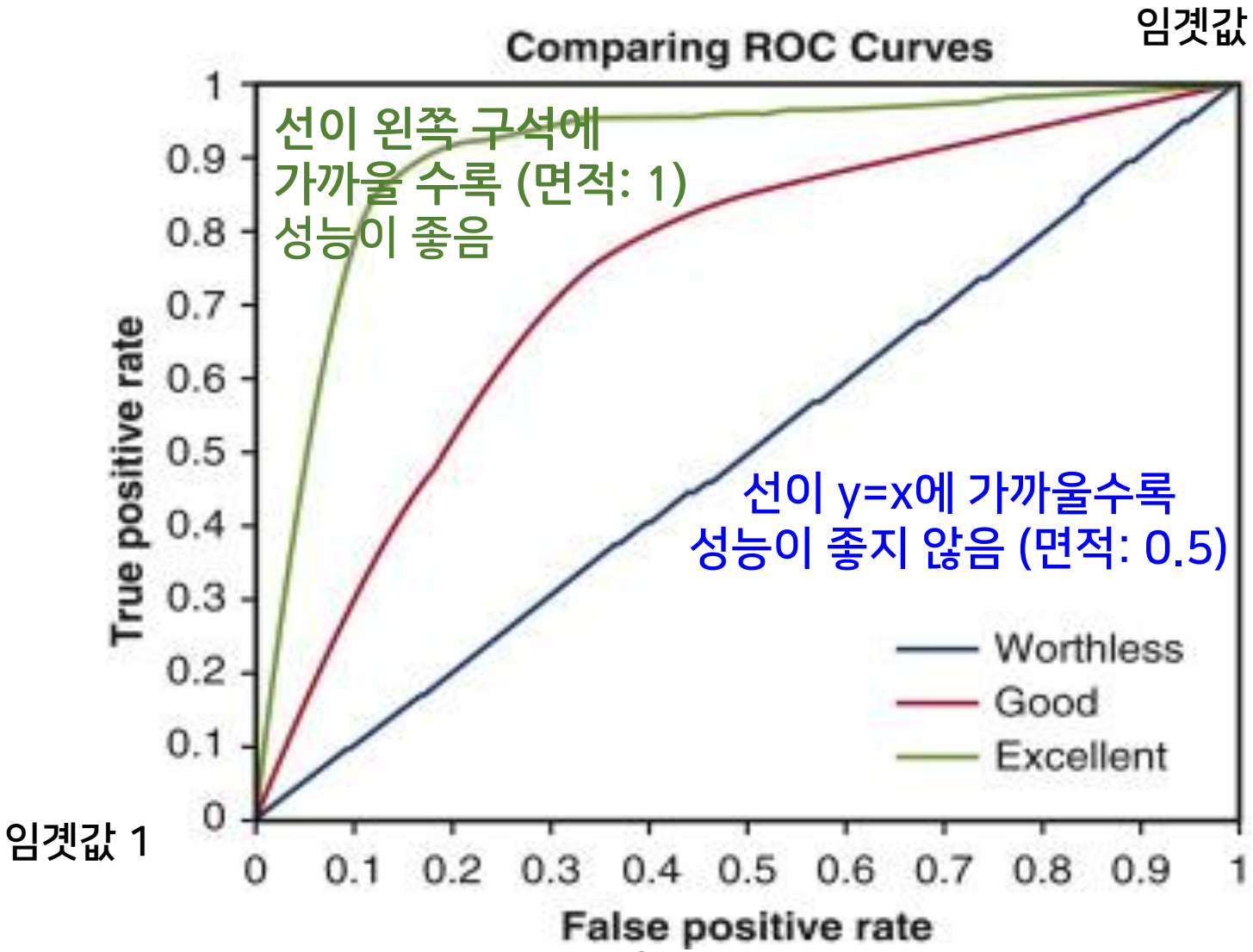
= FP / (FP + TN)

=1 - 특이성(Specificity)

=1 – TNR(True Negative Rate)

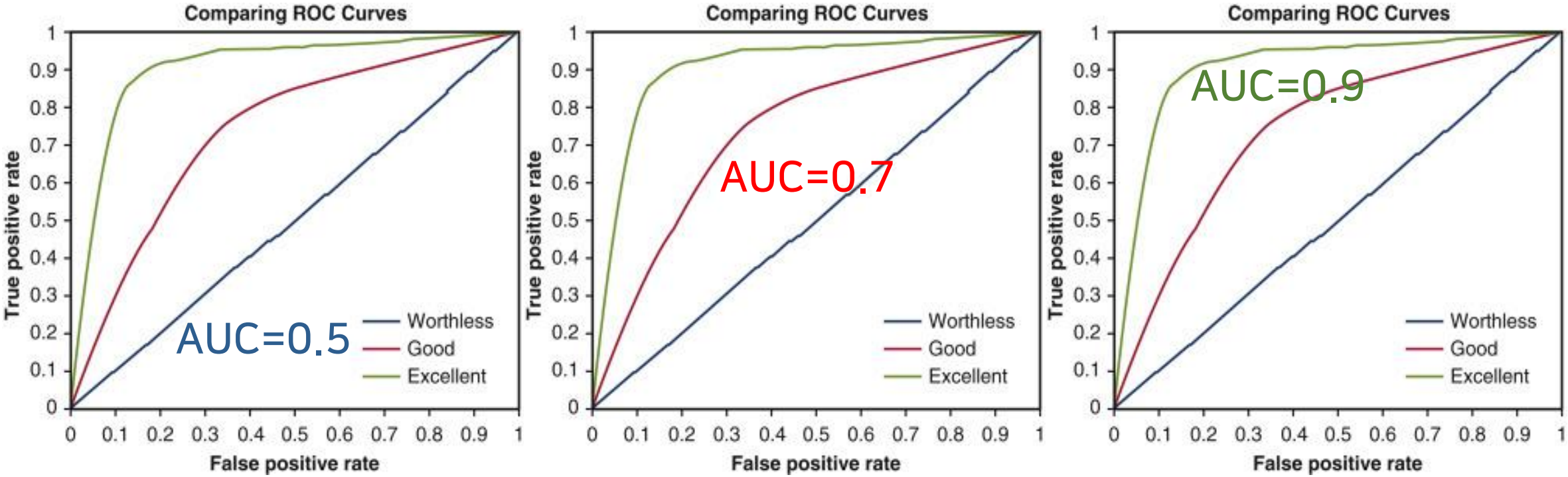
- ROC (Receiver Operating Characteristics) curves: for **visual** comparison of classification models
- Measures TPR and FPR while varying prediction threshold (0 to 1)

Model Selection: ROC Curves



ROC Curves and AUC

- AUC: ROC Curve로 만들어지는 면적(적분값)
  → FPR과 TPR을 동시에 고려하기 때문에 클래스 분포에 robust 하다 (영향을 덜 받는다)

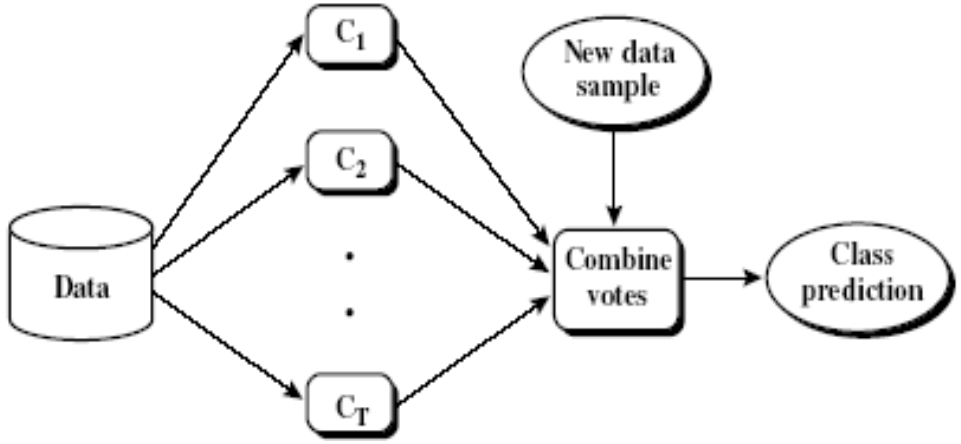


Issues Affecting Model Selection

- **Accuracy (예측 정확도)**
- **Speed**
  - time to construct the model (training time)
  - time to use the model (classification/prediction time)
- **Robustness (강건성)**: 노이즈에 굳건한 정도
- **Scalability**: 데이터 사이즈가 커 졌을때 적용 가능한지
- **Interpretability**: 모델이 학습한 예측 패턴을 사람이 이해할 수 있는지

Contents

Ensemble Methods: Increasing the Accuracy

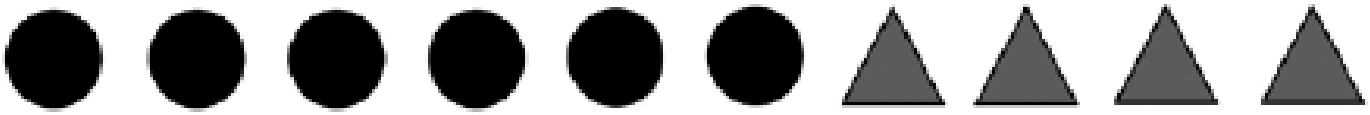


- Ensemble methods
  - Use a combination of models to increase accuracy
  - Combine a series of k learned models, $M_1, M_2, \cdots, M_k$, with the aim of creating an improved model M^*
- Popular ensemble methods
  - **Voting Ensemble**: combining a set of heterogeneous classifiers
  - Bagging: averaging the prediction over a collection of classifiers (not covered)
  - Boosting: weighted vote with a collection of classifiers (not covered)

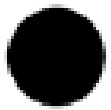
Voting Ensemble

- 같은 데이터에 대해 학습된 서로 다른 개별 분류기를 이용해 주어진 샘플의 예측 레이블 값을 얻고, 많은 선택을 받은 클래스 레이블을 선택하는 방법
- 쉽게 이야기 해, 다수결 방식의 머신러닝 구현으로 생각할 수 있다.
- 예) 각 도형이 하나의 분류기 예측을 나타낸다면, 동그라미 클래스로 여섯 개 분류기가 예측하고, 네 개 분류기가 세모 클래스로 예측을 하여 과반수 선택을 받은 동그라미 클래스로 최종 예측을 한다.

개별 분류기 예측

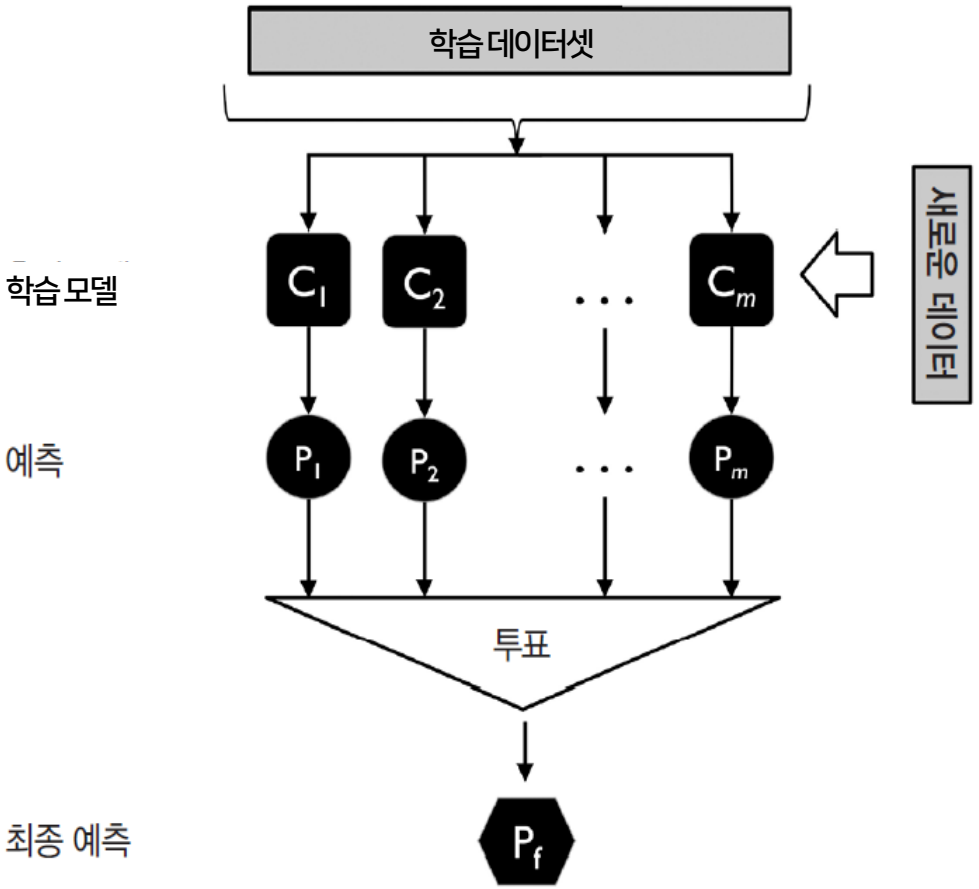


과반수 투표 예측



Voting Ensemble: Learning

- 같은 학습 데이터셋을 사용하여 m 개의 다른 개별 분류기 $(C_1, C_2, \cdots, C_m)$ 를 학습
- C_i 는 결정 트리, 로지스틱 회귀 등등 임의의 분류기가 될 수 있음



Voting Ensemble: Inference

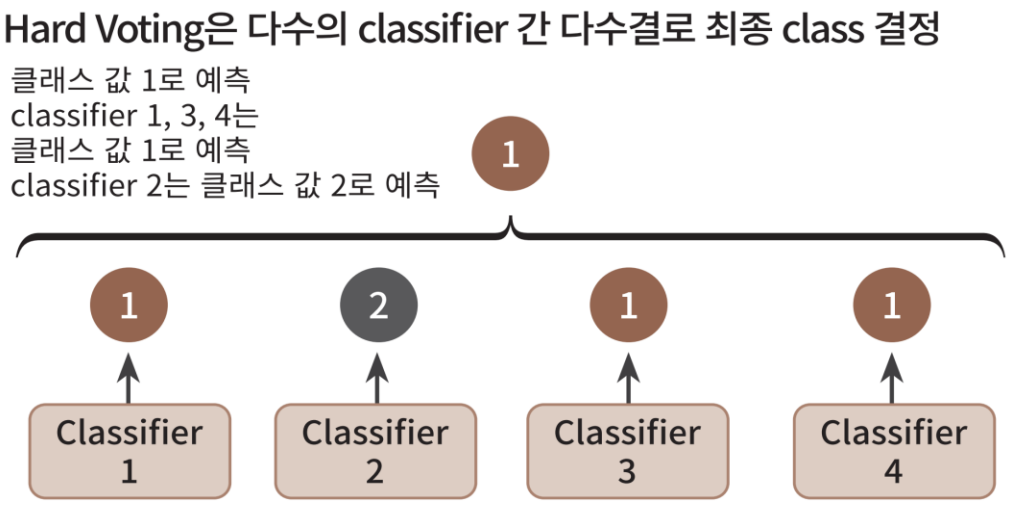
- 주어진 샘플 x 에 대해, 개별 분류기들의 예측 레이블을 모아, 가장 많은 표를 받은 레이블 $\hat{y}$ 를 선택

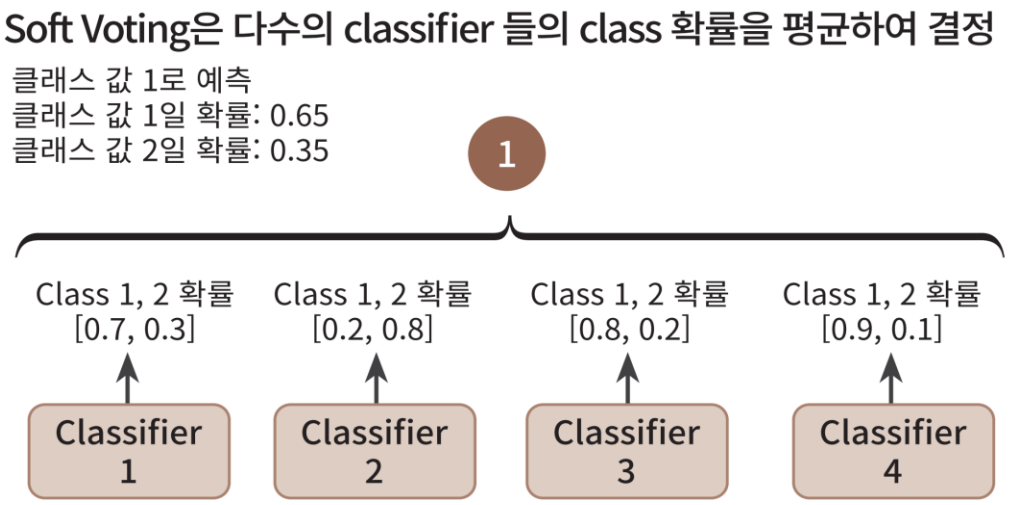
$$\hat{y} = mode\{C_1(\boldsymbol{x}), C_2(\boldsymbol{x}), \cdots, C_m(\boldsymbol{x})\}$$

- 최빈값(mode): 주어진 값들 중 가장 많이 등장한 수. 예를 들어, $mode\{0, 1, 1, 1, 0, 0, 1, 1\} = 1$
- 즉, 다수결 방법을 머신러닝 예측에 그대로 구현한 것이다.

Variations of Voting Ensemble

- 투표 방법은 예측 레이블을 이용하는지, 예측 확률을 이용하는지에 따라 하드 보팅, 소프트 보팅으로 나뉜다.
- 하드 보팅: 개별 분류기 예측 레이블 값들의 최빈값
- 소프트 보팅: 개별 분류기 예측 확률 값들의 평균값





Examples of Voting Ensemble

- 같은 데이터셋에 대해 학습된 세 개의 이진 분류기 C_1, C_2, C_3 가 있을 때,
  주어진 샘플 x 에 대한 예측 확률이 다음과 같다고 하자:

  $C_1(\boldsymbol{x}) \rightarrow [0.6, 0.4],\ C_2(\boldsymbol{x}) \rightarrow [0.55, 0.45],\ C_3(\boldsymbol{x}) \rightarrow [0.1, 0.9]$

- 하드 보팅을 이용한 예측 결과 (예측 임곗값 0.5 기준)

  $\hat{y} = mode\{0, 0, 1\} = 0$

- 소프트 보팅을 이용한 예측 결과

  예측 확률 평균값 $= [0.417, 0.583]$. 따라서, 예측 임곗값 0.5 기준 $\hat{y} = 1$.

Classification of Class-Imbalanced Data Sets

- **Class-imbalance problem**: Rare positive example but numerous negative ones, e.g., medical diagnosis, fraud, oil-spill, fault, etc.
- Traditional methods assume a balanced distribution of classes and equal error costs: not suitable for class-imbalanced data
- Typical methods for imbalance data in 2-class classification:
  - **Oversampling**: re-sampling of data from positive class
  - **Under-sampling**: randomly eliminate tuples from negative class
  - **Threshold-moving**: moves the decision threshold, t, so that the rare class tuples are easier to classify, and hence, less chance of costly false negative errors
- (참고) 클래스 불균형 해결 방법 관련해 별도 강의 자료가 제공될 예정입니다.

Summary (1/2)

- **Classification** is a form of data analysis that extracts **models** describing important data classes.
- Effective and scalable methods have been developed for **decision tree induction** and many other classification methods.
- **Evaluation metrics** include: accuracy, sensitivity, specificity, precision, recall, F measure, and F_β measure.
- **Stratified k-fold cross-validation** is recommended for accuracy estimation. Ensemble methods can be used to increase overall accuracy by learning and combining a series of individual models.

Summary (2/2)

- **Significance tests** and **ROC curves** are useful for model selection.
- No single method has been found to be superior over all others for all data sets
- Issues such as accuracy, training time, robustness, scalability, and interpretability must be considered and can involve trade-offs, further complicating the quest for an overall superior method