# Machine Learning

# Contents

# 8.1. Why?

- There are various types of learners which use different
  - Algorithms
  - Hyper-parameters
  - Representations
  - Training sets
  - Subproblems

⇒ Different learners attain different performance!

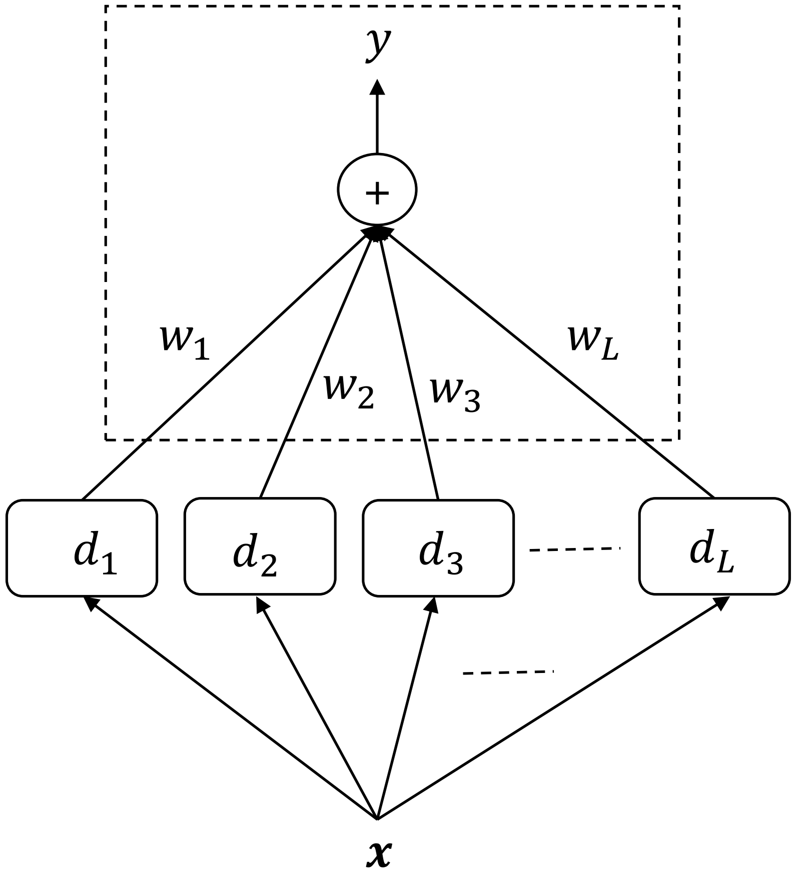
- No Free Lunch Theorems for Machine Learning
  - If an algorithm performs well on a certain class of problems then it necessarily pays for that with degraded performance on the set of all remaining problems
  - There is no single algorithm that always shows the most accurate performance.
  - Fine tuning of algorithms → there are instances the algorithm is inaccurate and there may be another algorithm that is accurate on those
- Generating a group of base learners and combining them for higher accuracy

# 8.2. Voting

- Linear combination of outputs (learners)

$$y = \sum_{j=1}^{L} w_j d_j$$

$$w_j \geq 0 \text{ and } \sum_{j=1}^{L} w_j = 1$$




- Classification

$$y_i = \sum_{j=1}^{L} w_j d_{ji}$$

  - Simple voting: equal weight
  - Plurality voting: class with max vote is the winner
  - Majority voting (2 class problem)

- Bayesian model

$$P(C_i|x) = \sum_{\text{all models } \mathcal{M}_j} P(C_i|x, \mathcal{M}_j)P(\mathcal{M}_j)$$

- If $d_j$ are i.i.d.(independent, identically distributed)

$$E[y] = E\left[\sum_j \frac{1}{L} d_j\right] = \frac{1}{L} L E[d_j] = E[d_j]$$

$$Var[y] = Var\left[\sum_j \frac{1}{L} d_j\right] = \frac{1}{L^2} Var[\sum_j d_j] = \frac{1}{L^2} L Var[d_j] = \frac{1}{L} Var[d_j]$$

  - Bias → no change, variance → decreasing by factor *L*
- Supposing that base learners are discriminant/regression function with random noises
  - Then, we are averaging over noise
  - If noises are uncorrelated with 0 mean, the noise effect average out to zero

Given an input value $\boldsymbol{x}$ and $L$ base learners, seek the weights such that

$$\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} E[(r^* - \sum_{j=1}^{L} w_j d_j(\boldsymbol{x}))^2]$$

- Expectation done with regard to the true input distribution
- Note $E[(r^* - \sum_{j=1}^{L} \hat{w}_j d_j(\boldsymbol{x}))^2] \le E[(r^* - \sum_{j=1}^{L} \frac{1}{L} d_j(\boldsymbol{x}))^2]$

Input distribution is not available in practice

- Use empirical distribution from training dataset

참조: 독립 확률변수의 분산

$N$개의 독립 확률변수 $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N$가 있으며 이들의 평균과 분산을 각각 $m_i = E[\boldsymbol{x}_i]$와 $\sigma_i^2 = E[(\boldsymbol{x}_i - m_i)^2]$이라고 하자. 이를 합한 것을 $z$라고 하면

$$z = \sum_{i=1}^{N} \boldsymbol{x}_i \tag{8.2.9}$$

이다. 그러면, 평균은

$$E[z] = \sum_{i=1}^{N} E[\boldsymbol{x}_i] = \sum_{i=1}^{N} m_i \tag{8.2.10}$$

이다. 분산은

$$\sigma_z^2 = E[(z - E[z])^2] = E[(\sum_{i=1}^{N} (\boldsymbol{x}_i - m_i))^2] \tag{8.2.11}$$

이다. 여기서, $\boldsymbol{x}_i$와 $\boldsymbol{x}_j (i \neq j)$는 독립이므로

$$E[(\boldsymbol{x}_i - m_i)(\boldsymbol{x}_j - m_j)] = E[(\boldsymbol{x}_i - m_i)]E[(\boldsymbol{x}_j - m_j)] = 0 \tag{8.2.12}$$

인 것을 이용하면 식 (8.2.11)는

$$\sigma_z^2 = \sum_{i=1}^{N} E[(\boldsymbol{x}_i - m_i)^2] = \sum_{i=1}^{N} \sigma_i^2 \tag{8.2.13}$$

이 된다.

이제 $\boldsymbol{x}_i$의 가중치 합에 의해

$$y = \sum_{i=1}^{N} w_i \boldsymbol{x}_i \tag{8.2.14}$$

로 주어진 경우의 평균을 구하면

$$E[y] = E[\sum_{i=1}^{N} w_i \boldsymbol{x}_i] = \sum_{i=1}^{N} w_i E[\boldsymbol{x}_i] = \sum_{i=1}^{N} w_i m_i \tag{8.2.15}$$

이고, 분산은 식 (8.2.13)을 이용하면

$$\sigma_y^2 = E[(\sum_{i=1}^{N} (w_i \boldsymbol{x}_i - w_i m_i))^2] = \sum_{i=1}^{N} E[(w_i \boldsymbol{x}_i - w_i m_i)^2] = w_i^2 \sum_{i=1}^{N} \sigma_i^2 \tag{8.2.16}$$

으로 구해진다.

참조: 확률변수의 상관관계 계수

2 개의 확률변수 $\boldsymbol{x}$와 $\boldsymbol{y}$ 사이의 상관관계 계수(correlation coefficient)는

$$r_{xy} = \frac{E[\boldsymbol{xy}] - E[\boldsymbol{x}]E[\boldsymbol{y}]}{\sigma_x \sigma_y} \tag{8.2.17}$$

로 정의된다. 여기서,

$$C_{xy} = E[\boldsymbol{xy}] - E[\boldsymbol{x}]E[\boldsymbol{y}] \tag{8.2.18}$$

를 공분산이라고 한다. $r_{xy} = 0$이면 $\boldsymbol{x}$와 $\boldsymbol{y}$ 는 상관관계가 없다고 한다.

# 8.3. Bagging

- Bootstrapping method
  - Generate training sets, train one base-learner with each, and combine them
  - From $X = \{\boldsymbol{x}^t, r^t\}_{t=1}^{N}$ generate $X_l, l = 1, 2, \dots, L$ with replacement
  - Train L base-learners for $X_l \rightarrow g_l(\boldsymbol{x})$
  - Output: average $g_{bag}(\boldsymbol{x}) = \frac{1}{L}\sum_{l=1}^{L} g_l(\boldsymbol{x})$

- Why?
  - $g^*(x)$ the best base-learner (expectation with regard to the input distribution)

$$
\begin{aligned}
E[(r^* - g^*(\boldsymbol{x}))^2] &= E[(r^* - g_{bag}(\boldsymbol{x}) + g_{bag}(\boldsymbol{x}) - g^*(\boldsymbol{x}))^2] \\
&= E[(r^* - g_{bag}(\boldsymbol{x}))^2] + E[(g_{bag}(\boldsymbol{x}) - g^*(\boldsymbol{x}))^2] \\
&\geq E[(r^* - g_{bag}(\boldsymbol{x}))^2]
\end{aligned}
$$

# 8.4. Evaluation of Classifiers

- "Accuracy" or "Recognition Ratio"

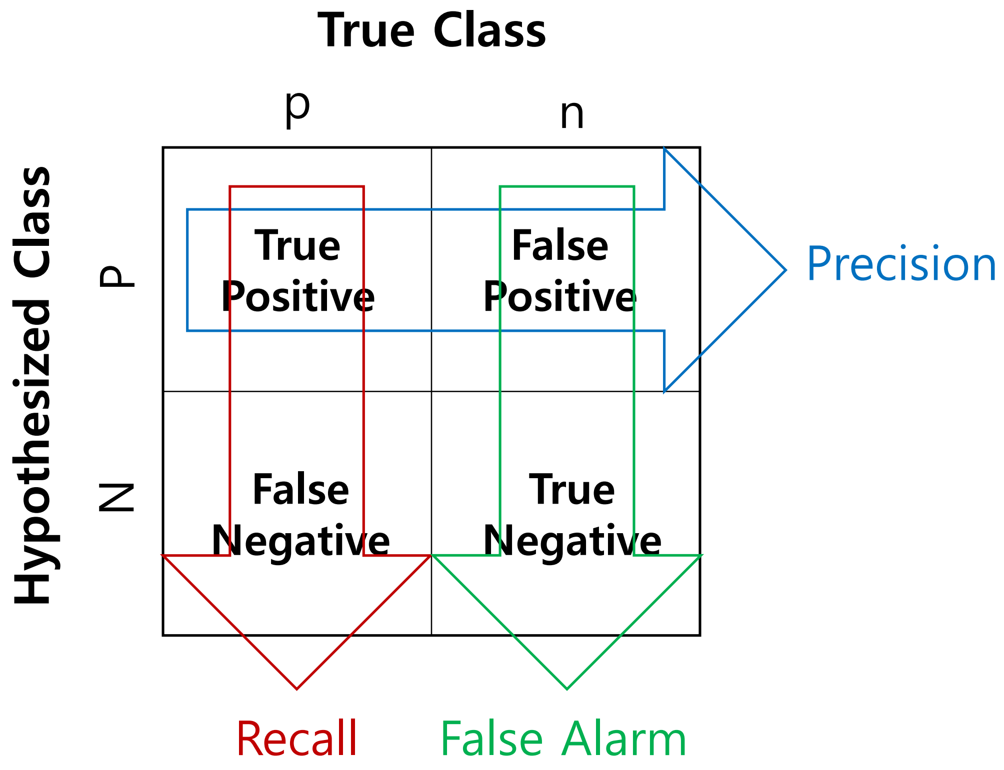
$$Accuracy = \frac{No.(Correctly\ Classified\ Samples)}{No.(Presented\ Samples)} \quad (8.4.1)$$

- "Precision and Recall"
  - TP : True Positive
  - FP : False Positive
  - TN : True Negative
  - FN : False Negative Positive

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$



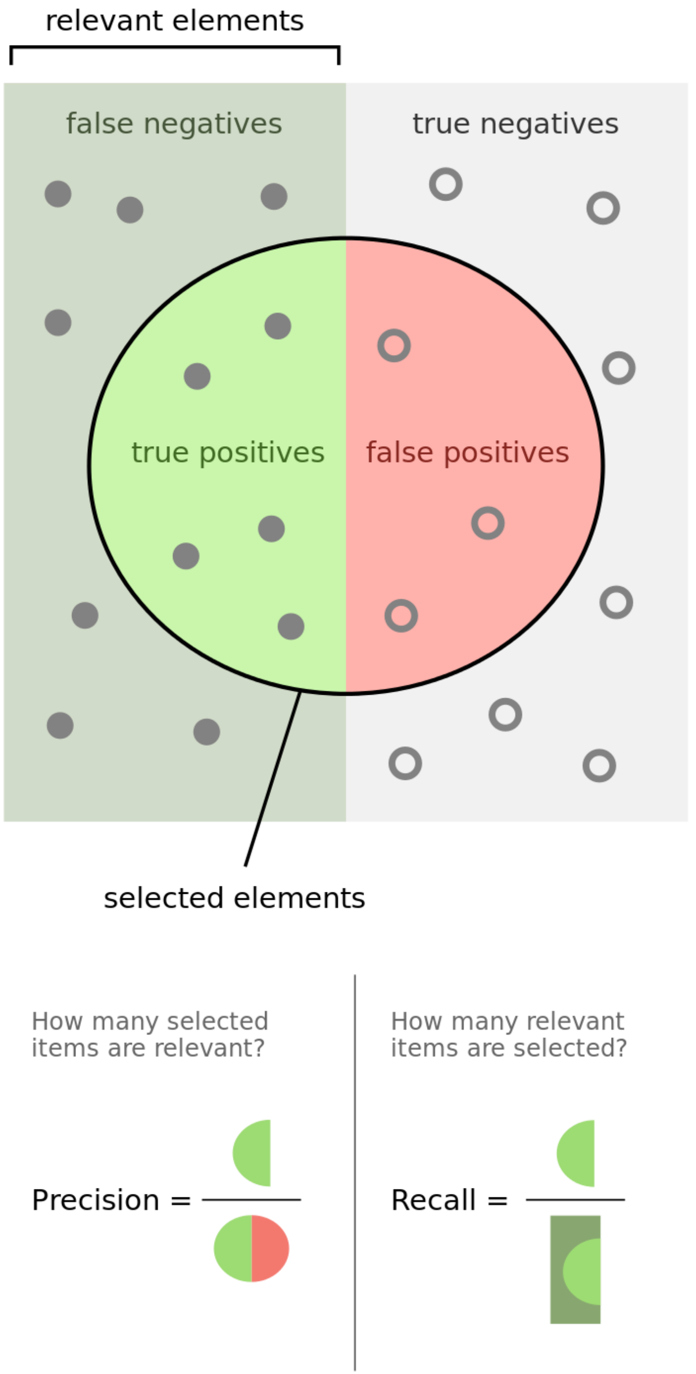

[Confusion Matrix]

- A precision score of 1.0 for a class C means that every item labeled as belonging to class C does indeed belong to class C (but says nothing about the number of items from class C that were not labeled correctly).

- A recall of 1.0 means that every item from class C was labeled as belonging to class C (but says nothing about how many other items were incorrectly also labeled as belonging to class C)

- An inverse relationship between precision and recall, where it is possible to increase one at the cost of reducing the other.

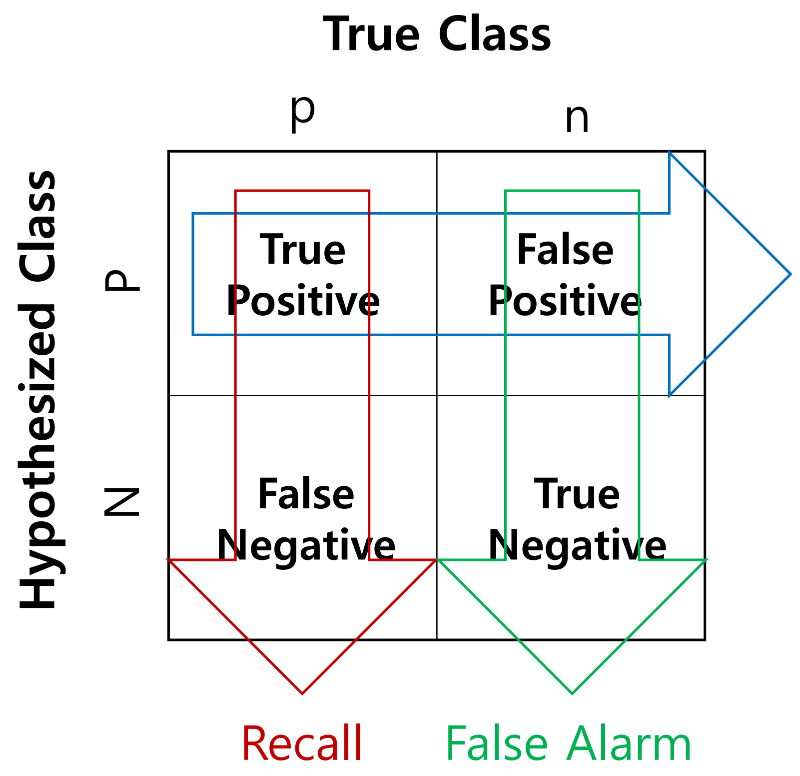
- Brain surgery provides an illustrative example of the tradeoff. Consider a brain surgeon tasked with removing a cancerous tumor from a patient’s brain. The surgeon needs to remove all of the tumor cells since any remaining cancer cells will regenerate the tumor. Conversely, the surgeon must not remove healthy brain cells since that would leave the patient with impaired brain function.
- The surgeon may be more liberal in the area of the brain he removes to ensure he has extracted all the cancer cells. This decision increases recall but reduces precision. On the other hand, the surgeon may be more conservative in the brain he removes to ensure he extracts only cancer cells. This decision increases precision but reduces recall.
- That is to say, greater recall increases the chances of removing healthy cells (negative outcome) and increases the chances of removing all cancer cells (positive outcome). Greater precision decreases the chances of removing healthy cells (positive outcome) but also decreases the chances of removing all cancer cells (negative outcome).

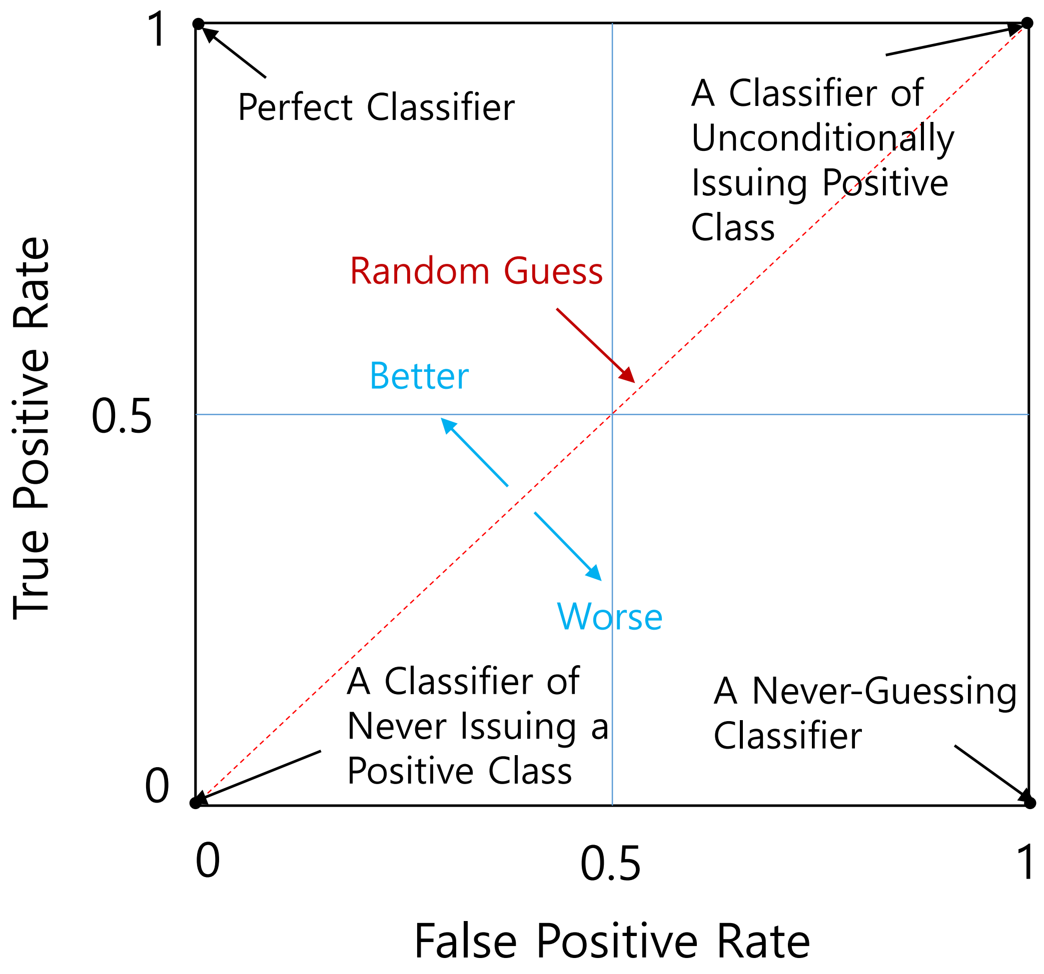



[from Wikipedia]

- **ROC(Receiver Operating Characteristic)**
  - a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied.
  - plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings.

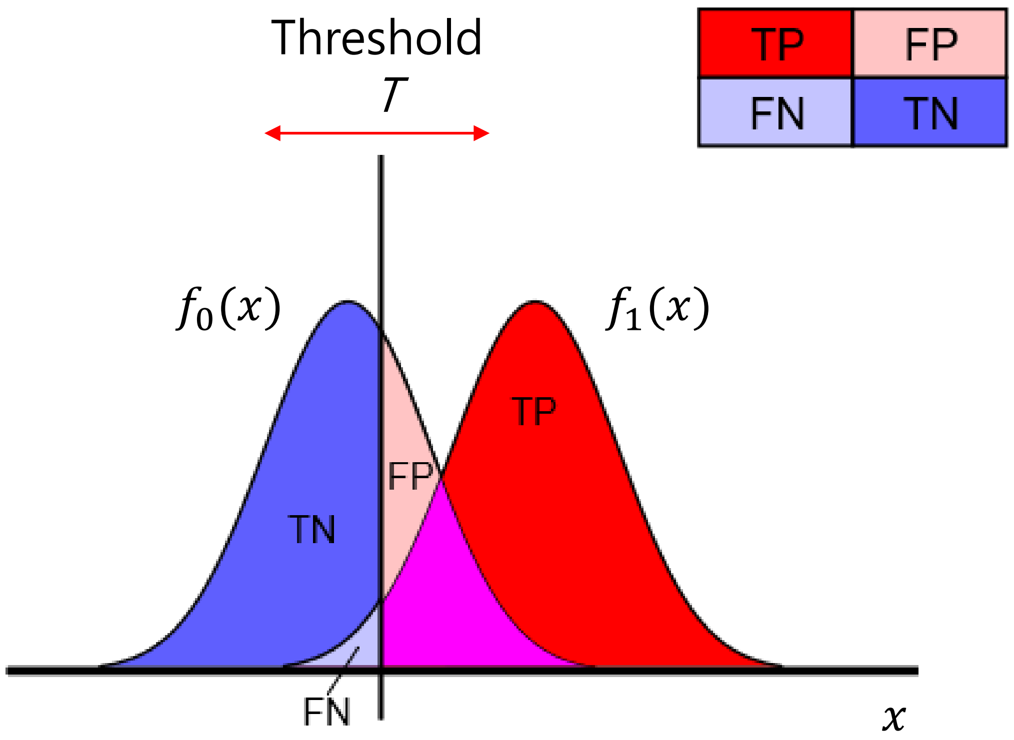
$$TPR = \frac{TP}{TP + FN} \qquad FPR = \frac{FP}{FP + TN}$$

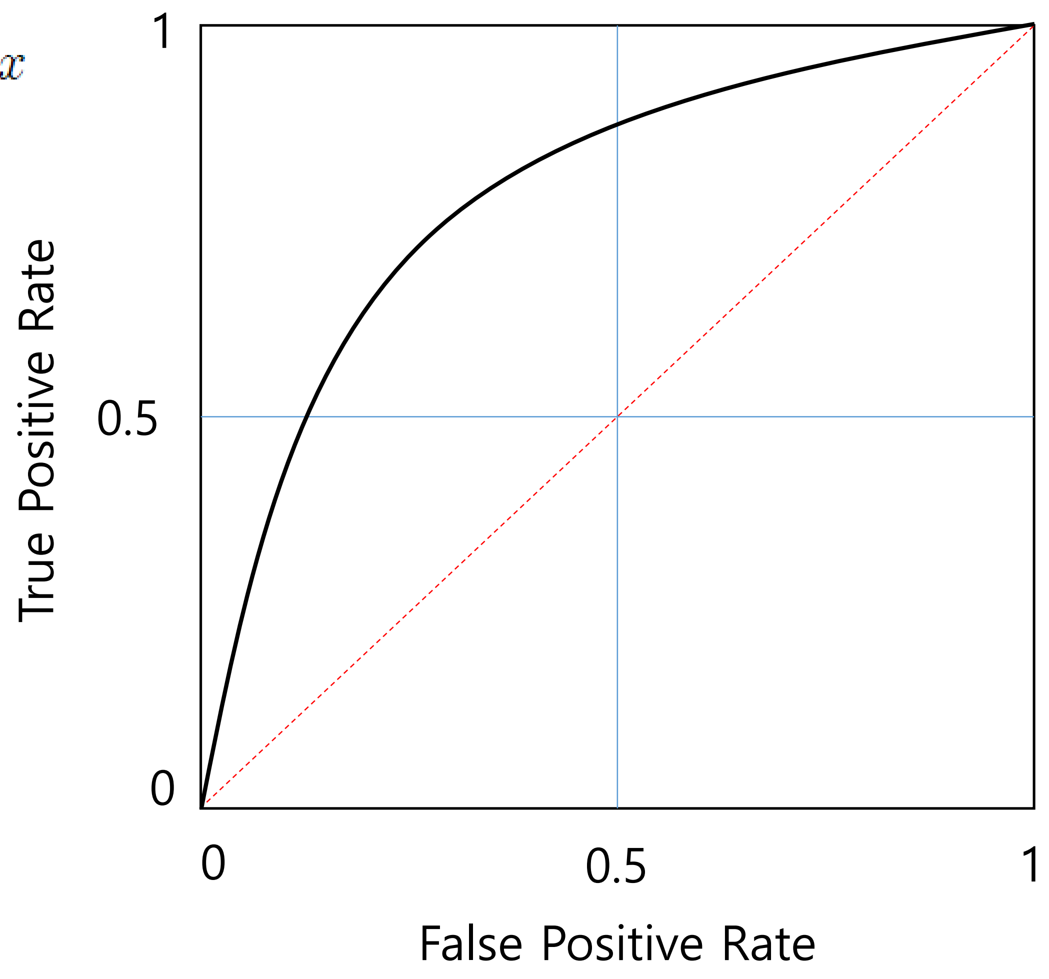


- **Curves in ROC Space : Useful for Imbalanced Data!**
  - Given a threshold *T*, the instance is classified as "positive" if $X > T$, and "negative" otherwise.
  - *X* follows a probability density $f_1(x)$ if the instance actually belongs to class "positive", and $f_0(x)$ if otherwise.

$$TPR(T) = \int_T^\infty f_1(x)dx \qquad FPR(T) = \int_T^\infty f_0(x)dx$$



**예제 8.4-1**

혼동행렬이 아래와 같이 주어진 경우에 정확도, 정밀도와 상기율, TPR, FPR을 계산하여라. 그리고, (FPR,TPR)을 ROC 공간상에 점으로 표시하라.

| | |
|---|---|
| TP=80 | FP=10 |
| FN=20 | TN=5 |