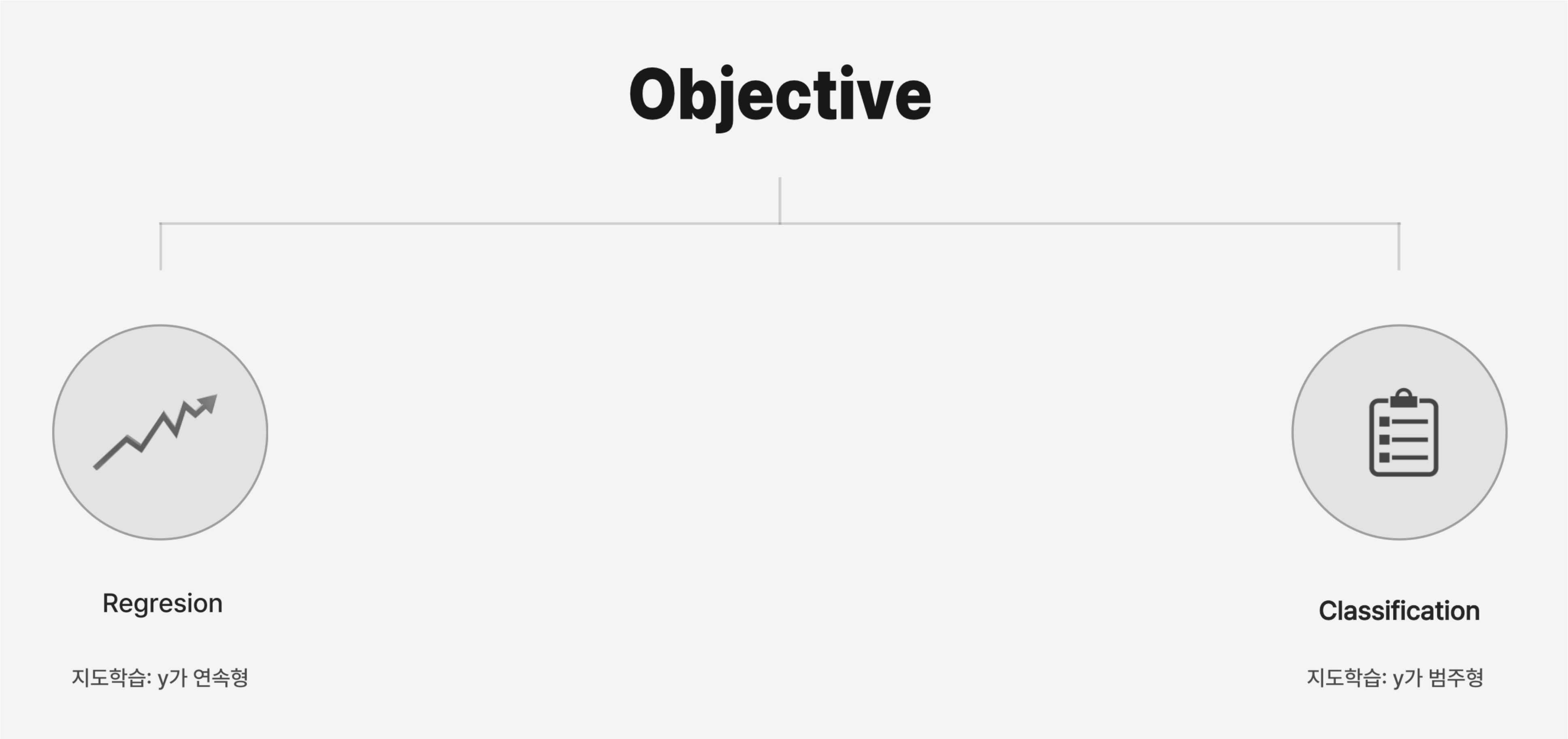
Statistical Machine Learning

3주차

담당: 17기 이서연

Classification

Classification



1. Bayesian Decision Theory

2. Parametric Method

3. Non-parametric Method

4. Model Evaluation

1. Bayesian Decision Theory

Bayes' Rule

$$\overbrace{P(C|\mathbf{x})}^{\text{posterior}} = \frac{\overbrace{P(C)}^{\text{prior}}\,\overbrace{p(\mathbf{x}|C)}^{\text{likelihood}}}{\underbrace{p(\mathbf{x})}_{\text{evidence}}}$$

$P(C=0)+P(C=1)=1$

$p(X)= p(X\,|C=1)P(C=1)+ p(X\,|C=0)P(C=0)$

$p(C=0|\,X)+P(C=1|\,X)=1$

$X = \{x_{1,} x_2\}$

$$\boxed{\begin{array}{l} \text{choose} \begin{cases} C=1 \text{ if } P(C=1|x_1,x_2) > 0.5 \\ C=0 \text{ otherwise} \end{cases} \\ \text{or} \\ \text{choose} \begin{cases} C=1 \text{ if } P(C=1|x_1,x_2) > P(C=0|x_1,x_2) \\ C=0 \text{ otherwise} \end{cases} \end{array}}$$

Bayes' Rule (K > 2 classes)

$$P(C_i \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid C_i)P(C_i)}{p(\mathbf{x})}$$

$$= \frac{p(\mathbf{x} \mid C_i)P(C_i)}{\sum_{k=1}^{K} p(\mathbf{x} \mid C_k)P(C_k)}$$

$$P(C_i) \geq 0 \; and \; \sum_{i=1}^{K} P(C_i) = 1$$

Choose C_i if $P(C_i \mid X) = max_k P(C_k|X)$

2. Parametric Method

2-1. Naïve Bayes Classifier

Parametric Estimation

$$P(C_{i.}|X) = \frac{P(X|C_{i.})P(C_i)}{P(X)} = \frac{P(X|C_{i.})P(C_i)}{\sum_{k=1}^{K} P(X|C_{k.})P(C_k)}$$

Discriminant : $g_i(x) = P(X|C_{i.})P(C_i)$ → $g_i(x) = log_2 P(X|C_{i.}) + log_2 P(C_i)$

Do know about the exact distribution? → **need Estimation!**

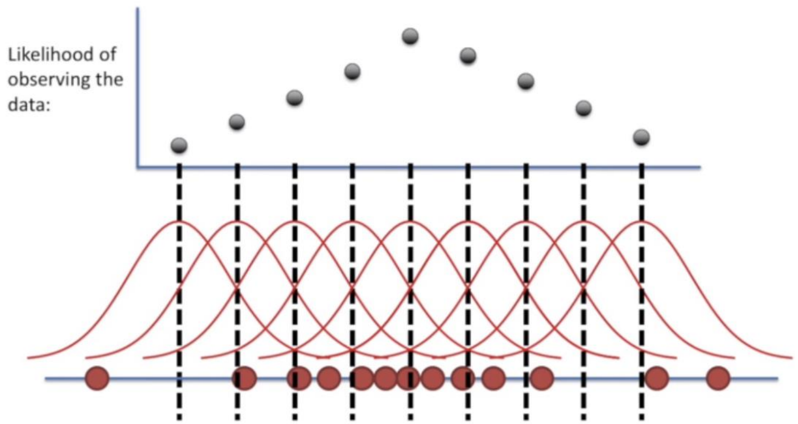
Back to MLE

$$P(C_{i.}|X) = \frac{P(X|C_{i.})P(C_i)}{P(X)} = \frac{P(X|C_{i.})P(C_i)}{\sum_{k=1}^{K} P(X|C_{k.})P(C_k)}$$

Discriminant : $g_i(x) = P(X|C_{i.})P(C_i) \rightarrow\ g_i(x) = log_2 P(X|C_{i.}) + log_2 P(C_i)$

Do know about the exact distribution? → **need Estimation!**

$$\begin{aligned}\theta_{MLE} &= \arg\max_\theta \log P(X|\theta) \\ &= \arg\max_\theta \log \prod_i P(x_i|\theta) \\ &= \arg\max_\theta \sum_i \log P(x_i|\theta)\end{aligned}$$



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**

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

Parametric Estimation

$$P(C_{i.}|X) = \frac{P(X|C_{i.})P(C_i)}{P(X)} = \frac{P(X|C_{i.})P(C_i)}{\sum_{k=1}^{K} P(X|C_{k.})P(C_k)}$$

Discriminant : $g_i(x) = P(X|C_{i.})P(C_i) \rightarrow g_i(x) = log_2 P(X|C_{i.}) + log_2 P(C_i)$

Example > $P(X|C_{i.})$ ~ Gaussian Distribution

$$P(X|C_{i.}) = \frac{1}{\sqrt{2\pi\sigma}} exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]$$

→ MLE for μ & σ

- $m = \frac{\sum_t x^t}{N}$
- $s^2 = \frac{\sum_t (x^t - m)^2}{N}$

$$g_i(x) = -\frac{1}{2}\log 2\pi - \log \sigma_i - \frac{(x-\mu_i)^2}{2\sigma_i^2} + \log P(C_i)$$

↓

$$g_i(x) = -\frac{1}{2}\log 2\pi - \log s_i - \frac{(x-m_i)^2}{2s_i^2} + \log \hat{P}(C_i)$$

Choose C_i if $P(C_i \mid X) = max_k P(C_k|X) = max_k g_k(x)$

Parametric Estimation

$$P(C_{i.}|X) = \frac{P(X|C_{i.})P(C_i)}{P(X)} = \frac{P(X|C_{i.})P(C_i)}{\sum_{k=1}^{K} P(X|C_{k.})P(C_k)}$$

Discriminant : $g_i(x) = P(X|C_{i.})P(C_i)$ → $g_i(x) = log_2 P(X|C_{i.}) + log_2 P(C_i)$

Example > $P(X|C_{i.})$ ~ Bernoulli, X = {0,1}

$P(X|C_{i.})$ =$p^X(1-p)^{(1-X)}$

→ MLE for p

- p = $\frac{\sum_t x^t}{N}$

$$g_i(x) = log \prod_t p^{X^t}(1-p)^{(1-X^t)} + log_2 P(C_i)$$

↓

Choose C_i if $P(C_i \mid X) = max_k P(C_k|X)$ = $max_k g_k(x)$

Parametric Estimation

$$P(C_{i.}|X) = \frac{P(X|C_{i.})P(C_i)}{P(X)} = \frac{P(X|C_{i.})P(C_i)}{\sum_{k=1}^{K} P(X|C_{k.})P(C_k)}$$

Discriminant : $g_i(x) = P(X|C_{i.})P(C_i)$ → $g_i(x) = log_2 P(X|C_{i.}) + log_2 P(C_i)$

Example > $P(X|C_{i.})$ ~ Multinomial, X_j = {0,1}
(X = $\{X_1, X_2, X_3, \cdots X_K\}$ | K >2)

$P(X_1, X_2, X_3, \cdots X_K|C_{i.})$ =$\prod_j p_j{}^{X_j}$

→ MLE for p_i

- $p_j = \frac{\sum_t X_j{}^t}{N}$

$$g_i(x) = \log \prod_t \prod_j p_j{}^{X_j^t} + log_2 P(C_i)$$

↓

Choose C_i if $P(C_i \mid X) = max_k P(C_k|X)$ = $max_k g_k(x)$

Naïve Bayes Classifier

Assume Independent among attributes X_j when class $C_{i.}$ is given

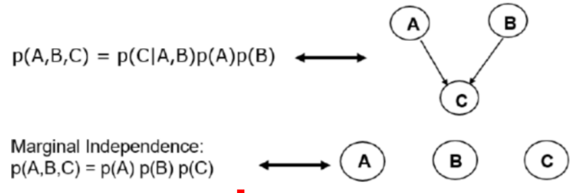
Discriminant : $g_i(x) = P(X|C_{i.})P(C_i) = P(C_i)\prod_j P(X_j|C_{i.})$

$\rightarrow log_2 P(C_i) + \sum_j P(X_j|C_{i.})$

Discrete X_j
→ Bernoulli or Multinomial

Continuous X_j
→ Gaussian (Normal) distribution

- Robust to isolated noise points
- Handle missing values by ignoring the instance during estimation
- Robust to irrelevant attributes
- Independence assumption may not hold for some attributes → BBN(Bayesian Belief Networks)



2-2. Linear Discriminant

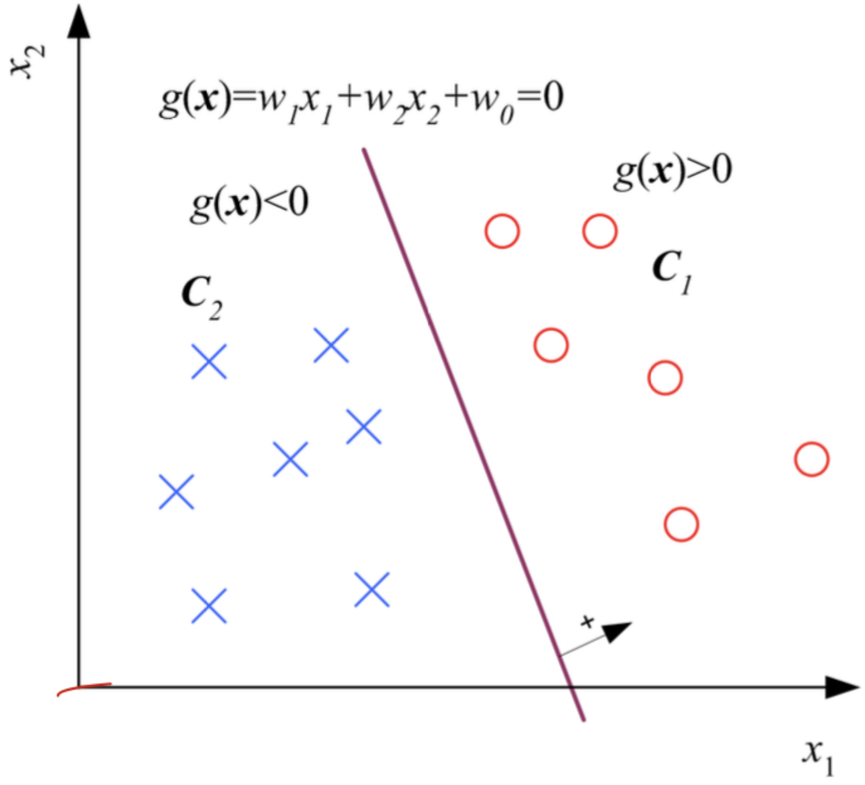
Likelihood - vs Discriminant -based Classification

Likelihood-based

- Use Bayes' Rule to calculate $P(C_{i.}|X)$
- Need Parametric estimation for $P(X|C_{i.})$
- Purpose : $g_i(x) = log_2 P(X|C_{i.}) + log_2 P(C_i)$

Discriminant Method

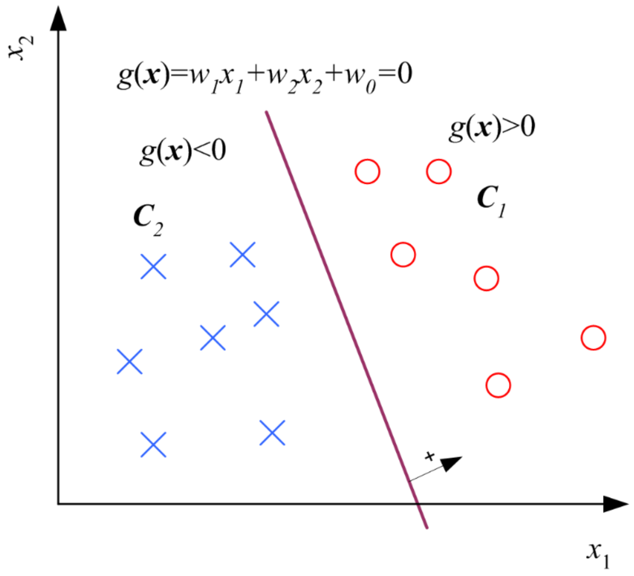
- Assume model $g_i(x)$ directly, **no density estimation**
- **Estimate boundary $g_i(x)$ from data x**



Linear Discriminant

Discriminant : $g_i(x) = \sum_j^d w_{ij}x_j + w_{i0} = w_i^T \mathrm{x} + w_{i0}$

If Two classes



$$
\begin{aligned}
g(\mathbf{x}) &= g_1(\mathbf{x}) - g_2(\mathbf{x}) \\
&= \left(\mathbf{w}_1^T\mathbf{x} + w_{10}\right) - \left(\mathbf{w}_2^T\mathbf{x} + w_{20}\right) \\
&= \left(\mathbf{w}_1 - \mathbf{w}_2\right)^T\mathbf{x} + \left(w_{10} - w_{20}\right) \\
&= \mathbf{w}^T\mathbf{x} + w_0
\end{aligned}
$$

$$
\text{choose } \begin{cases} C_1 & \text{if } g(\mathbf{x}) > 0 \\ C_2 & \text{otherwise} \end{cases}
$$

Multi-classes (k >2)

Choose C_i if $P(C_i \mid X) = max_k P(C_k|X) = max_k g_k(x)$

From Discriminant to Posterior

This is optimal solution… why?

Let assume $P(X|C_{i.})$ ~ Gaussian Distribution

$$g_i(x) = w_i^T \mathrm{x} + w_{i0}$$

$$g_i(x) = -\frac{1}{2}\log 2\pi - \log \sigma_i - \frac{(x-\mu_i)^2}{2\sigma_i^2} + \log P(C_i)$$

$$\mathbf{w}_i = \Sigma^{-1}\mu_i \quad w_{i0} = -\frac{1}{2}\mu_i^T\Sigma^{-1}\mu_i + \log P(C_i)$$

$$y \equiv P(C_1 \mid \mathbf{x}) \text{ and } P(C_2 \mid \mathbf{x}) = 1 - y$$

$$\text{choose } C_1 \text{ if} \begin{cases} y > 0.5 \\ y/(1-y) > 1 \\ \log[y/(1-y)] > 0 \end{cases} \text{ and } C_2 \text{ otherwise}$$

From Discriminant to Posterior

$$\text{logit}(P(C_1|\mathbf{x})) = \log\frac{P(C_1|\mathbf{x})}{1-P(C_1|\mathbf{x})} = \log\frac{P(C_1|\mathbf{x})}{P(C_2|\mathbf{x})}$$

$$= \log\frac{p(\mathbf{x}|C_1)}{p(\mathbf{x}|C_2)} + \log\frac{P(C_1)}{P(C_2)}$$

$$= \log\frac{(2\pi)^{-d/2}|\Sigma|^{-1/2}\exp[-(1/2)(\mathbf{x}-\mu_1)^T\Sigma^{-1}(\mathbf{x}-\mu_1)]}{(2\pi)^{-d/2}|\Sigma|^{-1/2}\exp[-(1/2)(\mathbf{x}-\mu_2)^T\Sigma^{-1}(\mathbf{x}-\mu_2)]} + \log\frac{P(C_1)}{P(C_2)}$$

$$= \mathbf{w}^T\mathbf{x} + w_0$$

where $\mathbf{w} = \Sigma^{-1}(\mu_1 - \mu_2)$ $\quad w_0 = -\frac{1}{2}(\mu_1+\mu_2)^T\Sigma^{-1}(\mu_1-\mu_2)$

The inverse of logit

$$\log\frac{P(C_1|\mathbf{x})}{1-P(C_1|\mathbf{x})} = \mathbf{w}^T\mathbf{x} + w_0$$

$$P(C_1|\mathbf{x}) = \text{sigmoid}(\mathbf{w}^T\mathbf{x}+w_0) = \frac{1}{1+\exp[-(\mathbf{w}^T\mathbf{x}+w_0)]}$$

=

$$\log(Odds(p)) = Wx + b$$

$$p(x) = \frac{1}{1+e^{-(Wx+b)}}$$

Logistic Regression (K = 2)

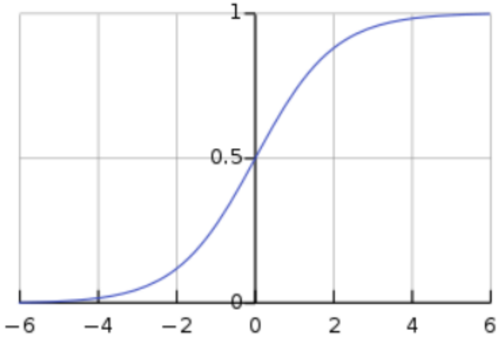
Discriminant : $g_i(x) = w_i^T \mathrm{x} + w_{i0}$ = score = z

Odds = $\frac{P(C_1|X)}{P(C_2|X)} = \frac{y}{1-y}$ → 한계가 있다(?) →**log(odds) = logit = z (실수 전체 범위)**

$$\log\frac{P(C_1|X)}{P(C_2|X)} = \log\frac{y}{1-y} = z = Wx + b$$

$y \equiv P(C_1|\mathbf{x})$ and $P(C_2|\mathbf{x}) = 1 - y$

choose C_1 if $\begin{cases} y > 0.5 \\ y/(1-y) > 1 \\ \log[y/(1-y)] > 0 \end{cases}$ and C_2 otherwise



sigmoid function

Discriminant : $g_i(x) = w_i^T \mathrm{x} + w_{i0}$ = score = z

$$p(x) = \frac{1}{1 + e^{-(Wx+b)}}$$

Choose C_1 when Wx+b> 0, y > 0.5

Q. But why sigmoid function?

Logistic Regression (K > 2)

Discriminant : $g_i(x) = w_i^T \mathrm{x} + w_{i0}$ = score = z_i

Odds = $\frac{P(C_i \mid X)}{P(C_k \mid X)}$ = e^{z_i}

$$\boxed{\frac{y}{1-y} = \exp(t)}$$

$\sum_1^{K-1} \frac{P(C_i \mid X)}{P(C_k \mid X)} = \sum_1^{K-1} e^{z_i} = \frac{1 - P(C_k \mid X)}{P(C_k \mid X)}$

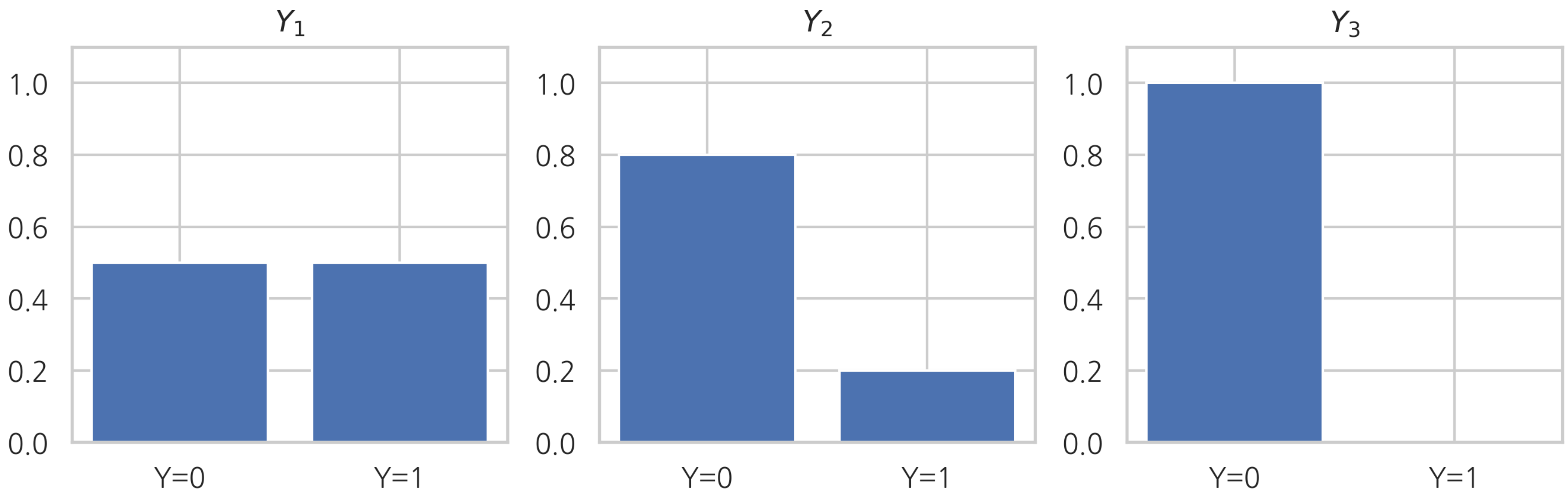
$$P(C_k \mid X) = \frac{1}{1 + \sum_1^{K-1} e^{z_i}}$$

$$P(C_i \mid X) = P(C_k \mid X) \times e^{z_i} = \frac{1}{1 + \sum_1^{K-1} e^{z_i}} \times e^{z_i} = \frac{e^{z_i}}{\sum_1^{K} e^{z_i}}$$

$$P(C_i \mid X) = \frac{e^{z_i}}{\sum_1^{K} e^{z_i}} = \mathrm{softmax}(z_i)$$

2-3. Learning Classifier

Entropy

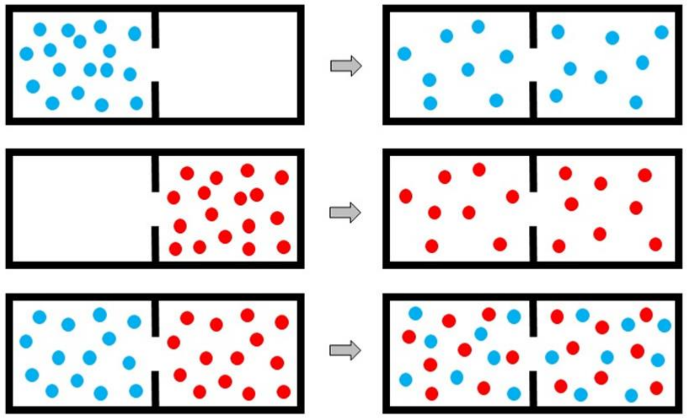


- Y1은 y값에 대해 아무것도 모르는 상태
- Y2는 y값이 0이라고 믿지만 아닐 가능성도 있다는 것을 아는 상태
-Y3는 y값이 0이라고 100% 확신하는 상태

Entropy

Entropy (불균형도)

- 특정 node t 에서 불순도
- 데이터 분포의 purity를 측정하는 척도, 여기서는 클래스의 분포의 purity를 측정
- Entropy가 낮을 수록 purity가 높은 것
- Max : $\log_2 n_c$ (n_c: 클래스 총 개수)
- Min : 0 (클래스가 1개 밖에 없을 경우)



$$Entropy(t) = -\sum_j p(j|t) \cdot log_2\, p(j|t)$$

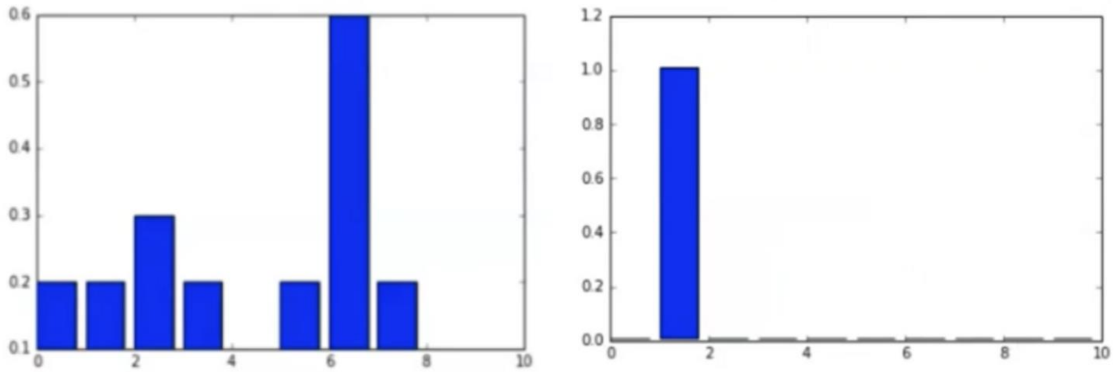
j =class

Cross-Entropy

두 분포의 차이의 척도

$$\text{Cross-entropy} = -\sum_{i=1}^{N} p_i \log q_i$$

p: 실제 정답의 분포
q: 모델을 통해 구한 답의 분포



Minimize Cross-Entropy!

Minimize Loss Function!

How to find parameters

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

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

MLE? → Loss function

If K=2 (Binary Classification)

- **Bernoulli distribution**

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

Maximize Log Likelihood

- Binary Cross Entropy

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

Minimize Loss Function

We know about p (output of model)

p $= \frac{1}{1+e^{-z}}$ = $\sigma(z) = sigmoid\ function$

$P(C_i\,|X) = \frac{e^{z_i}}{\sum_1^K e^{z_i}} = \mathrm{softmax}(z_i)$ if K>2

Gradient Descent

Minimize Loss Function

We know about p (output of model)

$p = \frac{1}{1+e^{-z}} = \sigma(z) = sigmoid\ function$

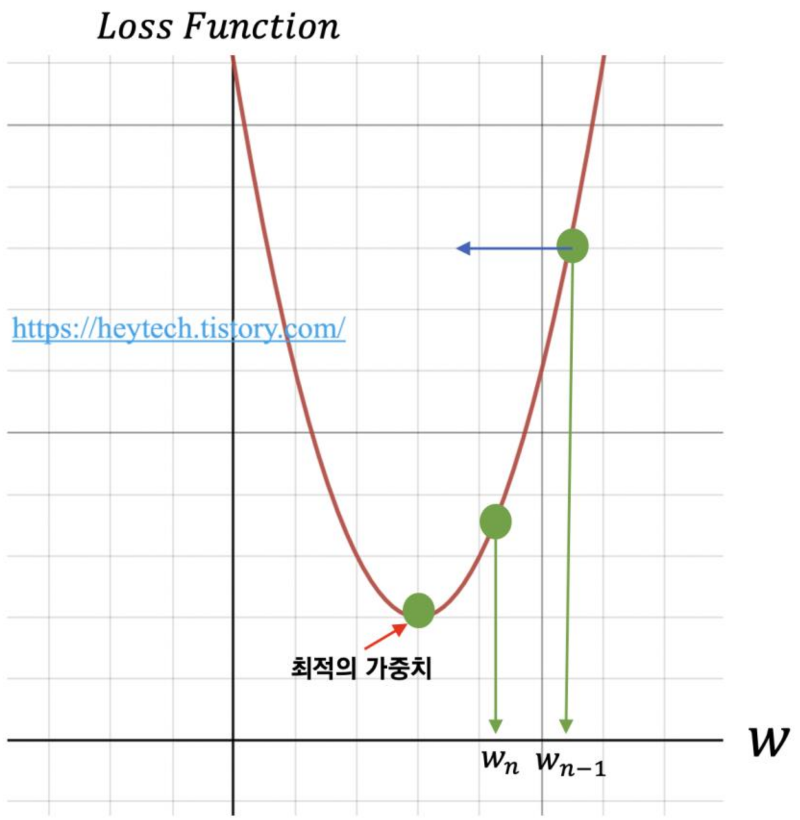
$P(C_i\ |X) = \frac{e^{z_i}}{\sum_1^K e^{z_i}} = \mathrm{softmax}(z_i)$ if K>2

1. model : $g_i(x) = w_i^T \mathrm{x} + w_{i0}$ = score = z_i
2. Loss function : E(w | X) = Cross-Entropy
3. Optimization : $w^* = \ argmin_w E(w|X)$

Gradient : $\nabla_w E$

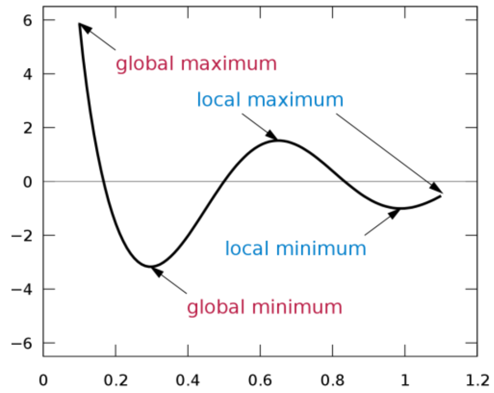
$$w_n = w_{n-1} - \alpha \nabla f(w_{n-1})$$

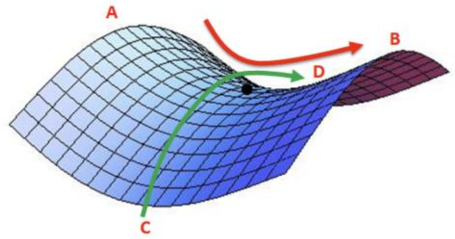
Gradient Descent



Gradient : $\nabla f(w_{n-1})$

Gradient-Descent method

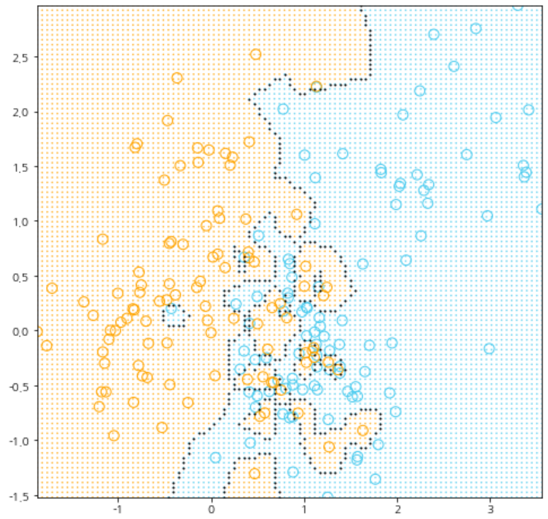




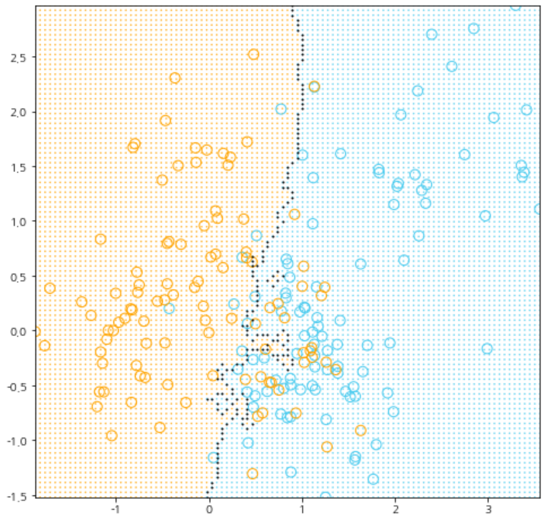
4. Non-parametric Method

KNN (K- Nearest Neighborhood)

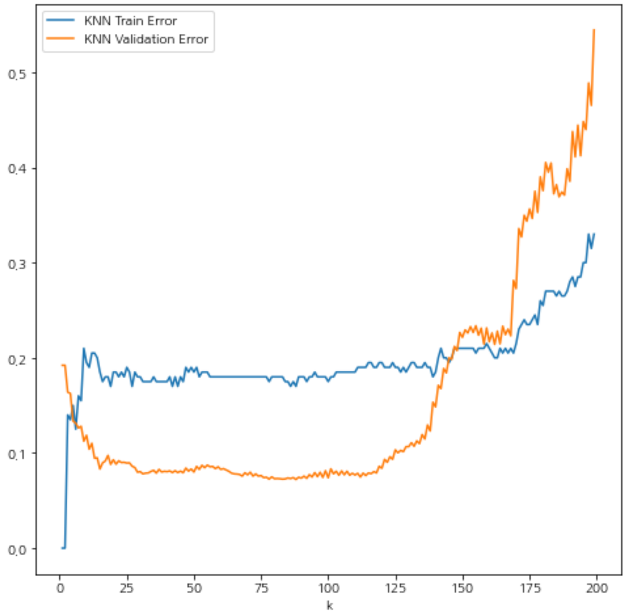
전형적인 non-parametric method



K = 1



K = 15



Best? K=88

KNN (K- Nearest Neighborhood)

- Distance measure

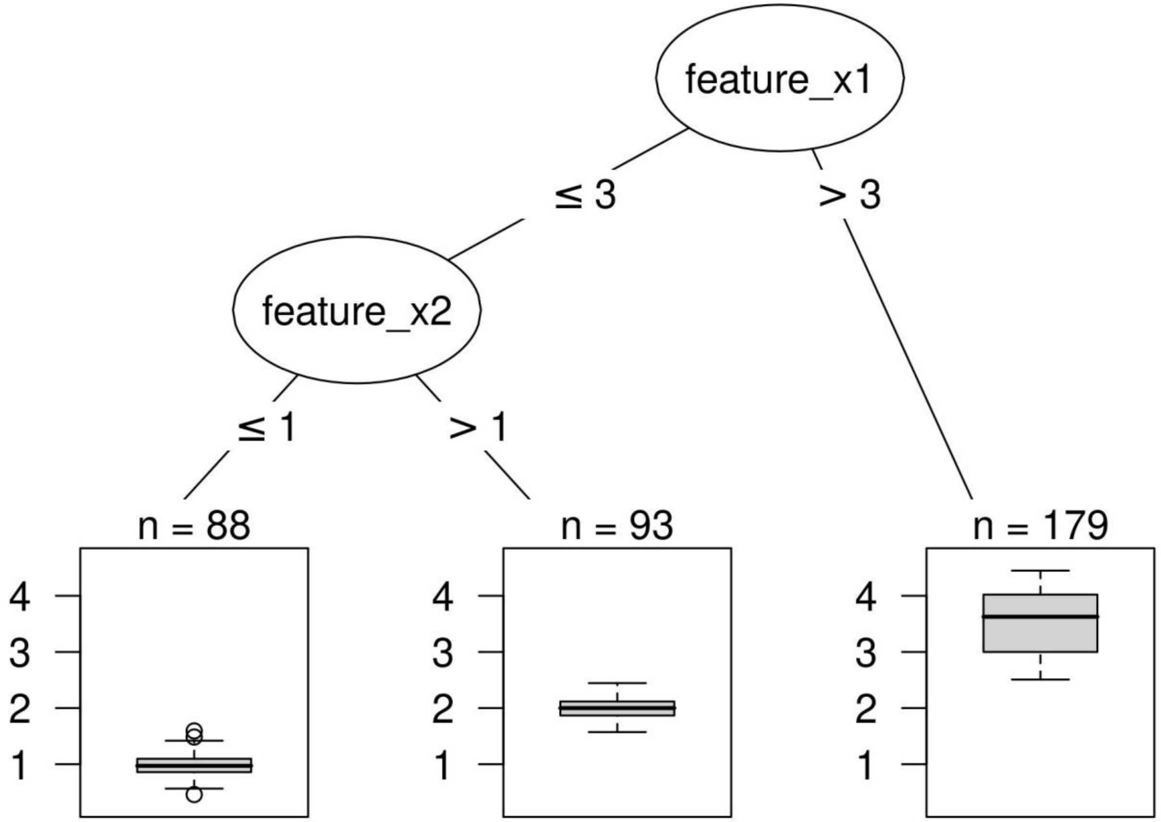
$$d(\mathbf{u},\mathbf{v}) = (\sum|u_i - v_i|^2)^{\frac{1}{2}} = ||\mathbf{u}-\mathbf{v}||_2 \qquad \textit{Euclidean (L2 norm)}$$

$$d(\mathbf{u},\mathbf{v}) = \sum|u_i - v_i| = ||\mathbf{u}-\mathbf{v}||_1 \qquad \textit{Manhattan (L1 norm)}$$

$$d(\mathbf{u},\mathbf{v}) = (\sum|u_i - v_i|^p)^{\frac{1}{p}} = ||\mathbf{u}-\mathbf{v}||_p \qquad \textit{Minkowski (Lp norm)}$$

$$d(\mathbf{u},\mathbf{v}) = \sqrt{(\mathbf{x}-\boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x}-\boldsymbol{\mu})} \qquad \textit{Mahalanobis Distance}$$

Decision Tree



5. Model Evaluation

Confusion Matrix

	Predicted Class		
Actual class		Positive	Negative
	Positive	**True Positive(TP)**	**False Negative(FN)**
	Negative	**False Positive(FP)**	**True Negative(TN)**

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

Q. What is Limitation of Accuracy?

Confusion Matrix

	Predicted Class		
Actual class		Positive	Negative
	Positive	**True Positive(TP)**	**False Negative(FN)**
	Negative	**False Positive(FP)**	**True Negative(TN)**

Precision(정밀도) : $\frac{TP}{TP+FP}$ → 양성 예측 중, 실제로 맞은 비율 / 열방향

Recall(sensitivity, 재현율, 민감도) : $\frac{TP}{TP+FN}$ → 실제 양성 중, 맞은 비율 / 행방향

Specificity(특이도) : $\frac{TN}{TN+FP}$ → 실제 음성 중, 맞은 비율

F1-score

What was limitation of Accuracy?

Precision & recall Trade-off

Precision(정밀도) : $\frac{TP}{TP+FP}$

Recall(sensitivity, 재현율, 민감도) : $\frac{TP}{TP+FN}$

→ 둘 다 높이는 것이 가능한가…?
→ 좋은 모델은 positive한 것을 모두 제대로 분류하고, positive한 것만 제대로 분류하면 된다.

About precision

About recall

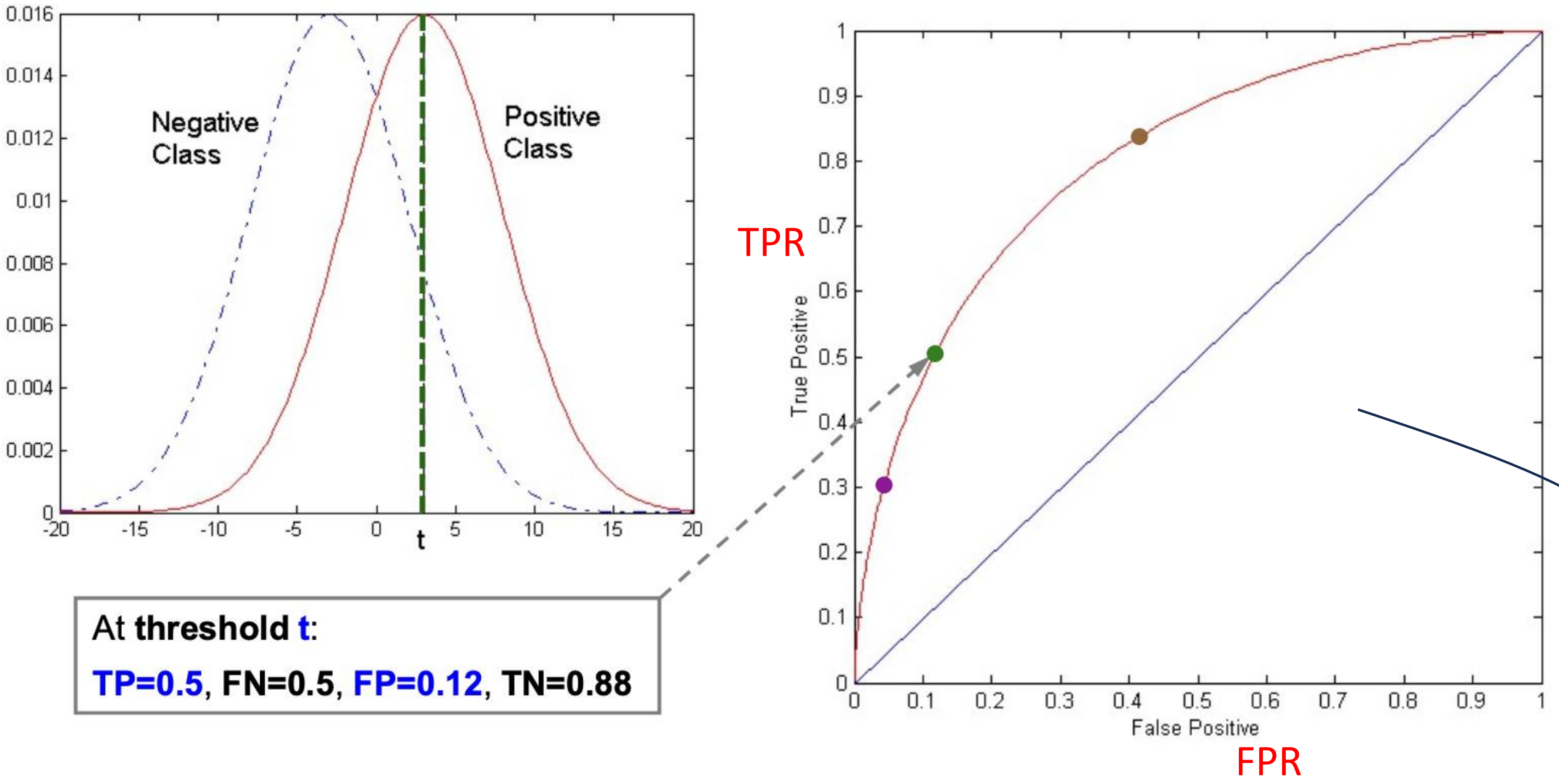
[Harmonized mean(조화 평균)]

$$\frac{1}{F1\ score} = 0.5(\frac{1}{precision} + \frac{1}{recall})$$

$$F1\ score = \frac{2 * precision\ * recall}{(precisoin + recall)}$$

ROC Curve

ROC(Reviewer Operating Characteristics) -Curve



TPR(True positive Rate) = recall

FPR(False positive Rate) = 1-specificity

AUC(Area Under Curve)

Bad= 0.5
Ideal = 1

수고하셨습니다!

해당 세션자료는 KUBIG Github에서 보실 수 있습니다!

다음은 이번 주차 과제 설명이 있습니다!