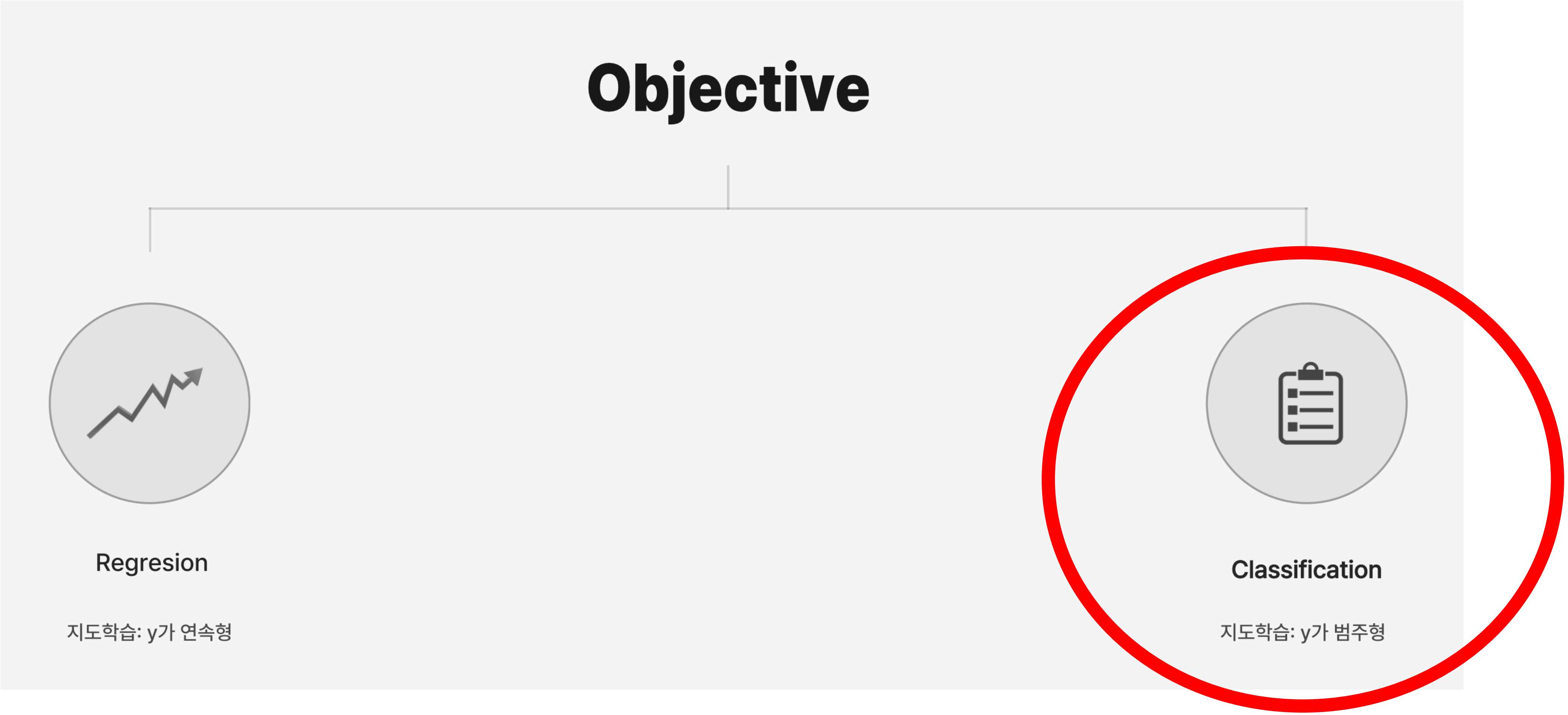
Statistical Machine Learning

3주차

담당: 18기 방서연

Classification

Classification



1. Bayesian Decision Theory

2. Parametric Method

3. Non-parametric Method

4. Model Evaluation

5. 이론과 응용

1. Bayesian Decision Theory

Bayes' Rule

posterior *prior* *likelihood*

$$P(C|\mathbf{x}) = \frac{P(C)p(\mathbf{x}|C)}{p(\mathbf{x})}$$

evidence

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

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

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

$X = \{x_1, x_2\}$

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

or

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

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!**

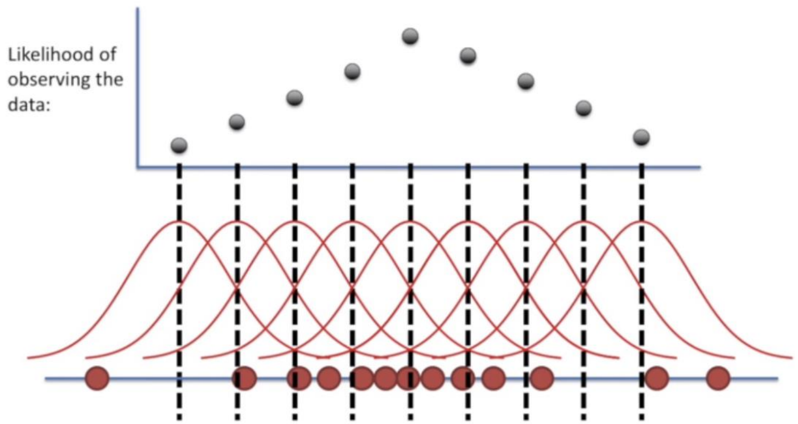
Back to MLE

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

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!**

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



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

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

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

$$\downarrow$$

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

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

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

Naïve Bayes Classifier

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

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

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

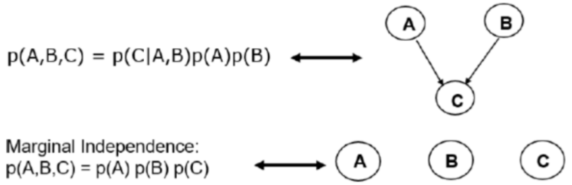
Discrete X_j

$\rightarrow$ Bernoulli or Multinomial

Continuous X_j

$\rightarrow$ 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 $\rightarrow$ BBN(Bayesian Belief Networks)



2-2. Linear Discriminant

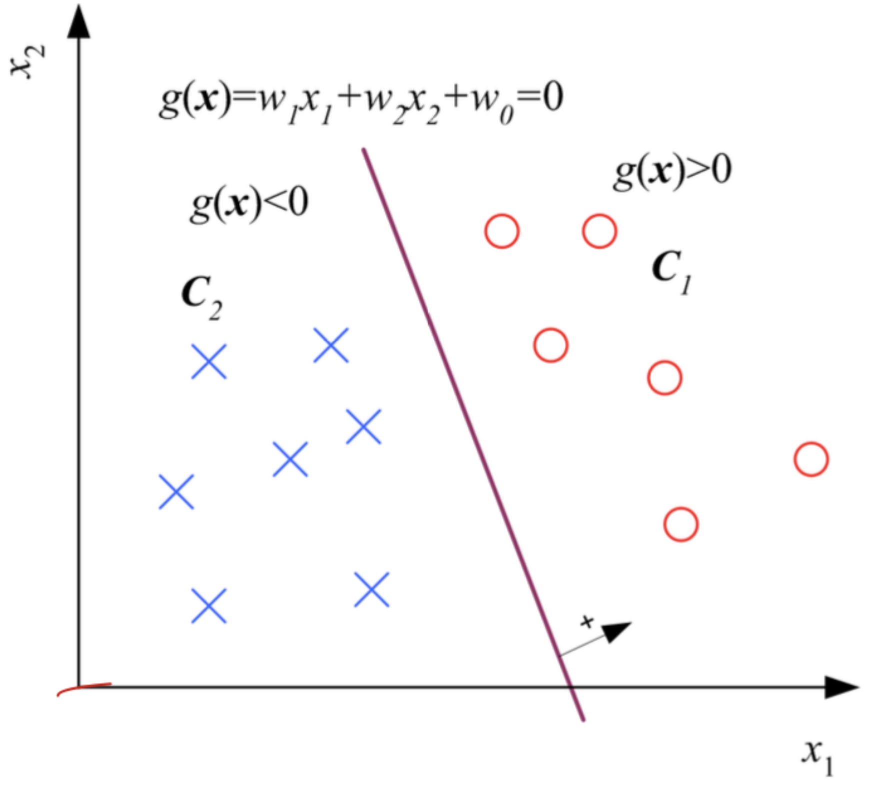
Likelihood - vs Discriminant -based Classification

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

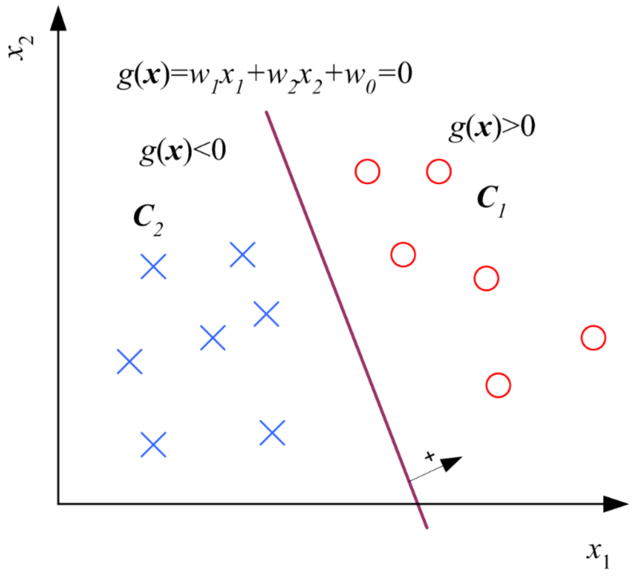
- Assume model $g_i(x)$ directly, **no density estimation**
- **Estimate boundary $g_i(x)$ from data x**



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)

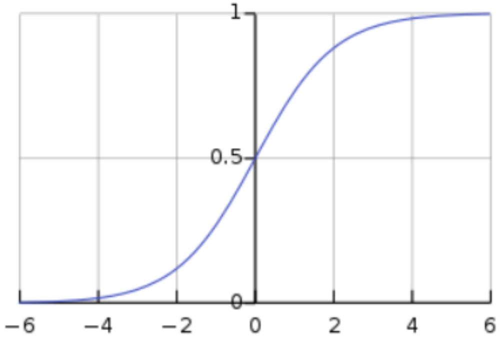
Choose C_i if $P(C_i \mid X) = max_k P(C_k|X)$ = $max_k g_k(x)$

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)**

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



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

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

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

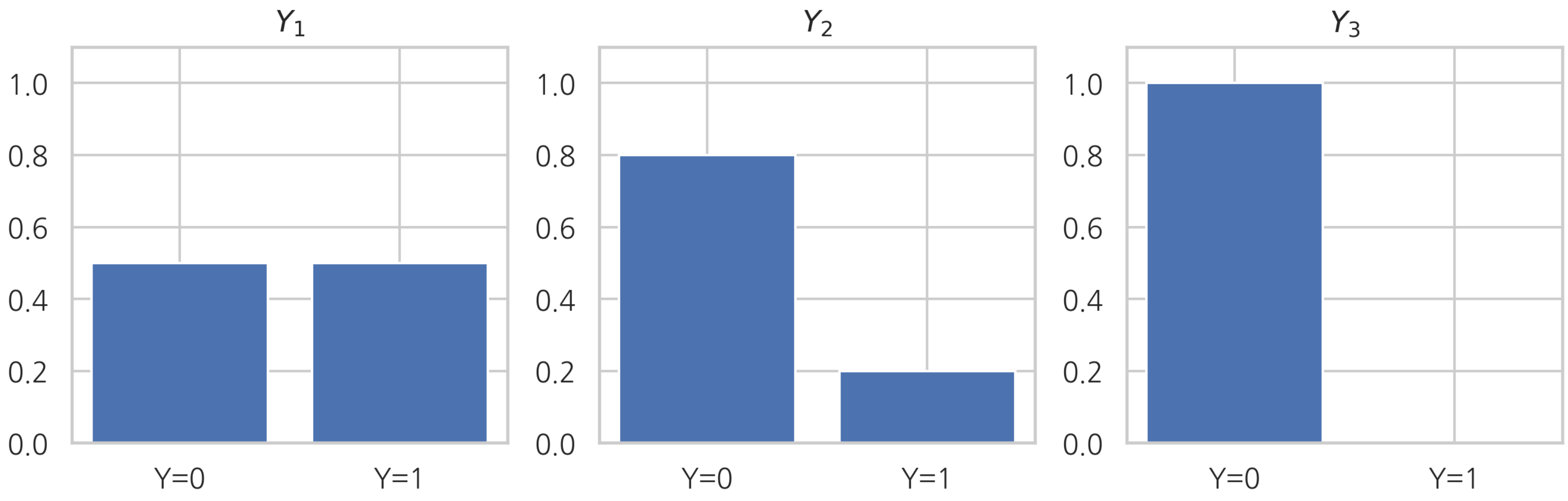
sigmoid function

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

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

2-3. Learning Classifier

Entropy

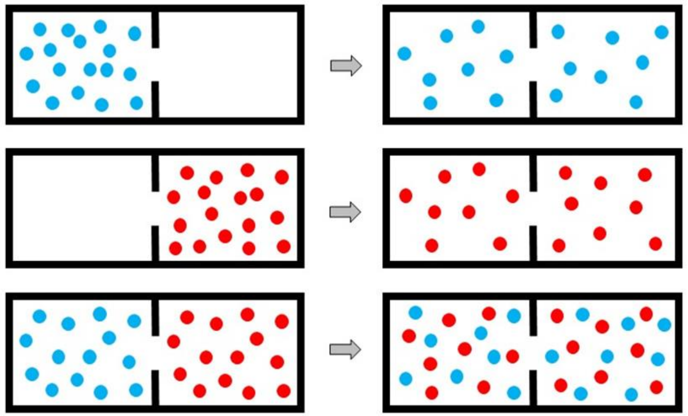


- Y1은 y값에 대해 아무것도 모르는 상태
- Y2는 y값이 0이라고 믿지만 아닐 가능성도 있다는 것을 아는 상태
-Y3는 y값이 0이라고 100% 확신하는 상태

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

Entropy

Classifier A

N = 100

	Number
A	40
B	10

	Number
A	20
B	30

Classifier B

N = 100

	Number
A	50
B	0

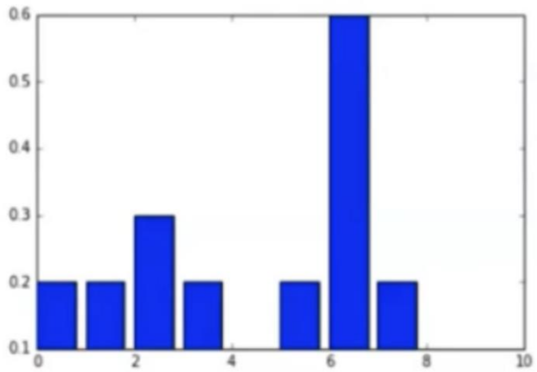
	Number
A	5
B	45

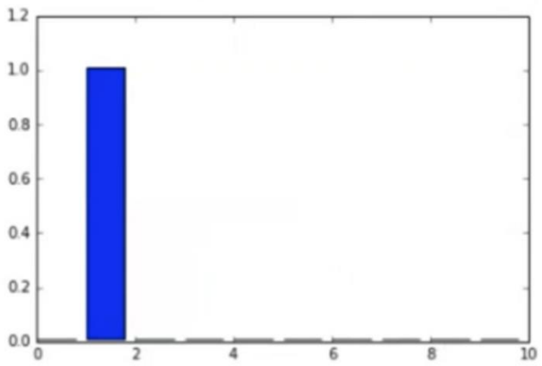
Cross-Entropy

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

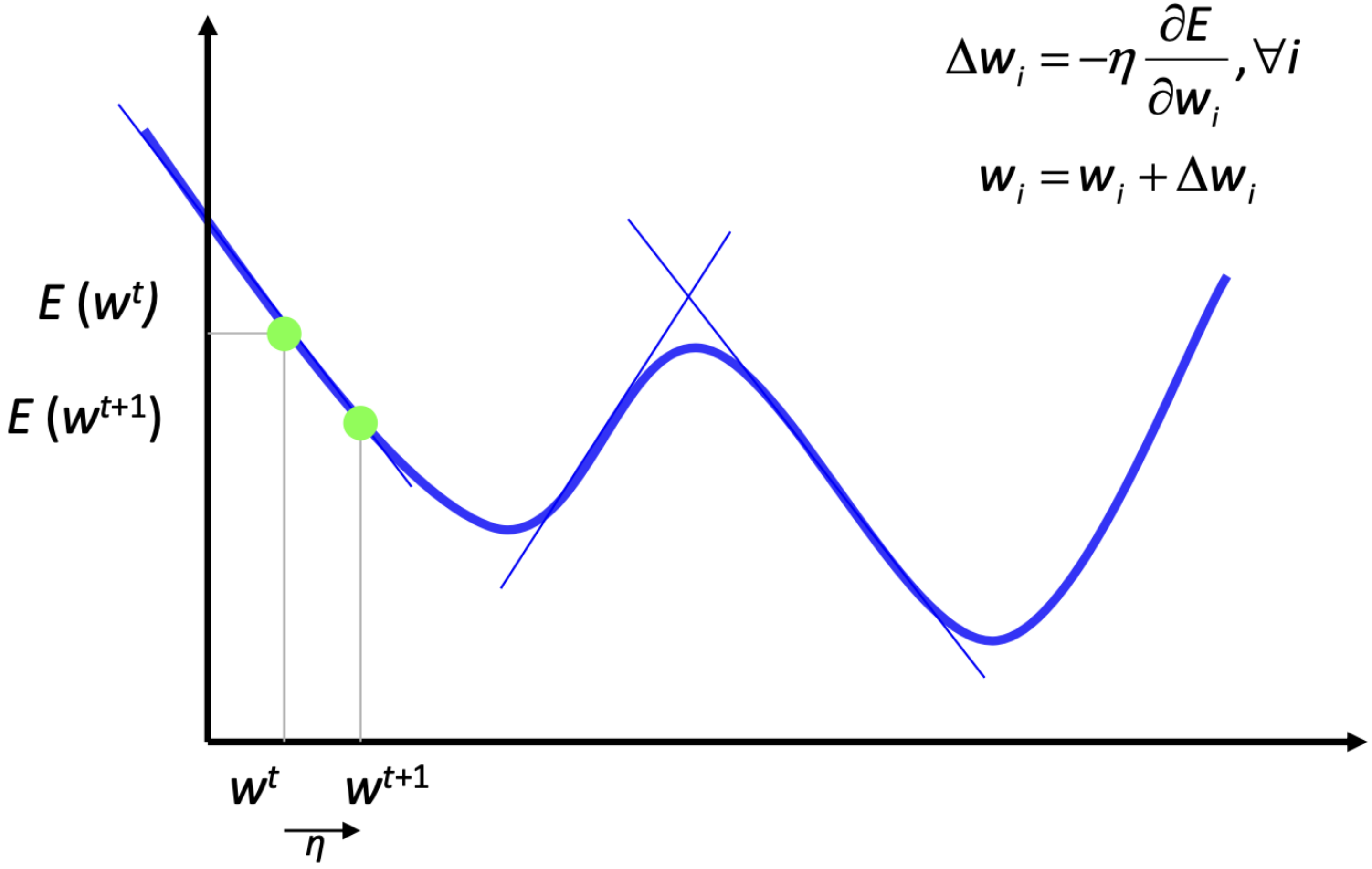
$P(C_i\,|X) = \frac{e^{z_i}}{\sum_1^K e^{z_i}} = \mathrm{softmax}(z_i)$ if K>2

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



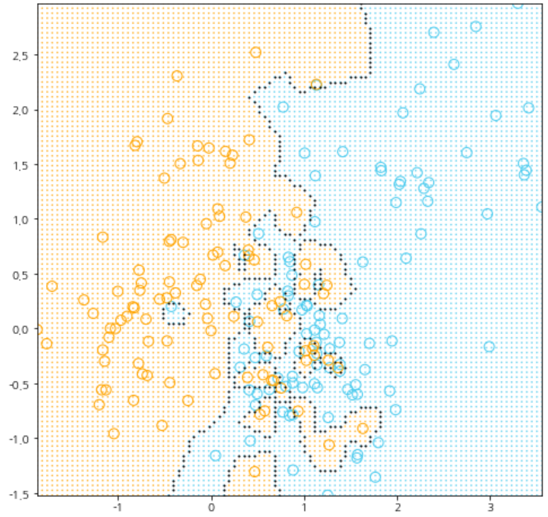
$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i}, \forall i$$

$$w_i = w_i + \Delta w_i$$

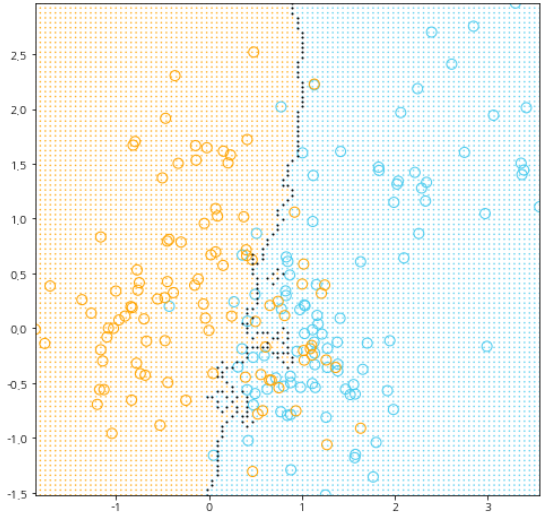
3. Non-parametric Method

KNN (K- Nearest Neighborhood)

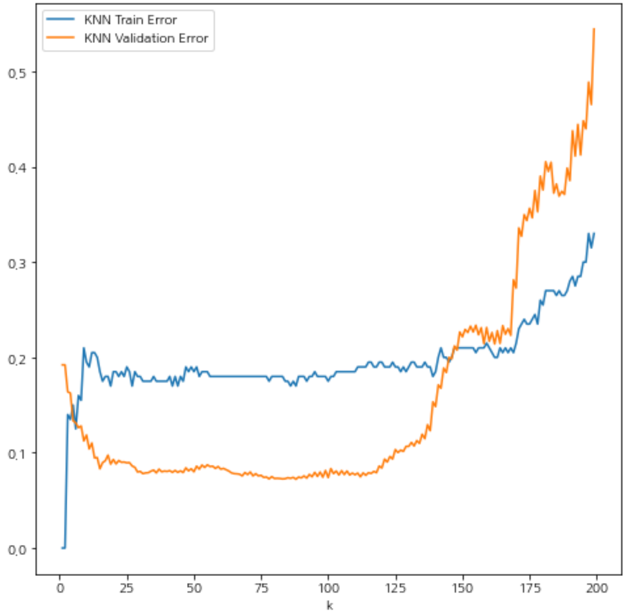
전형적인 non-parametric method



K = 1



K = 15



Best?
K=88

KNN (K- Nearest Neighborhood)

- Distance measure

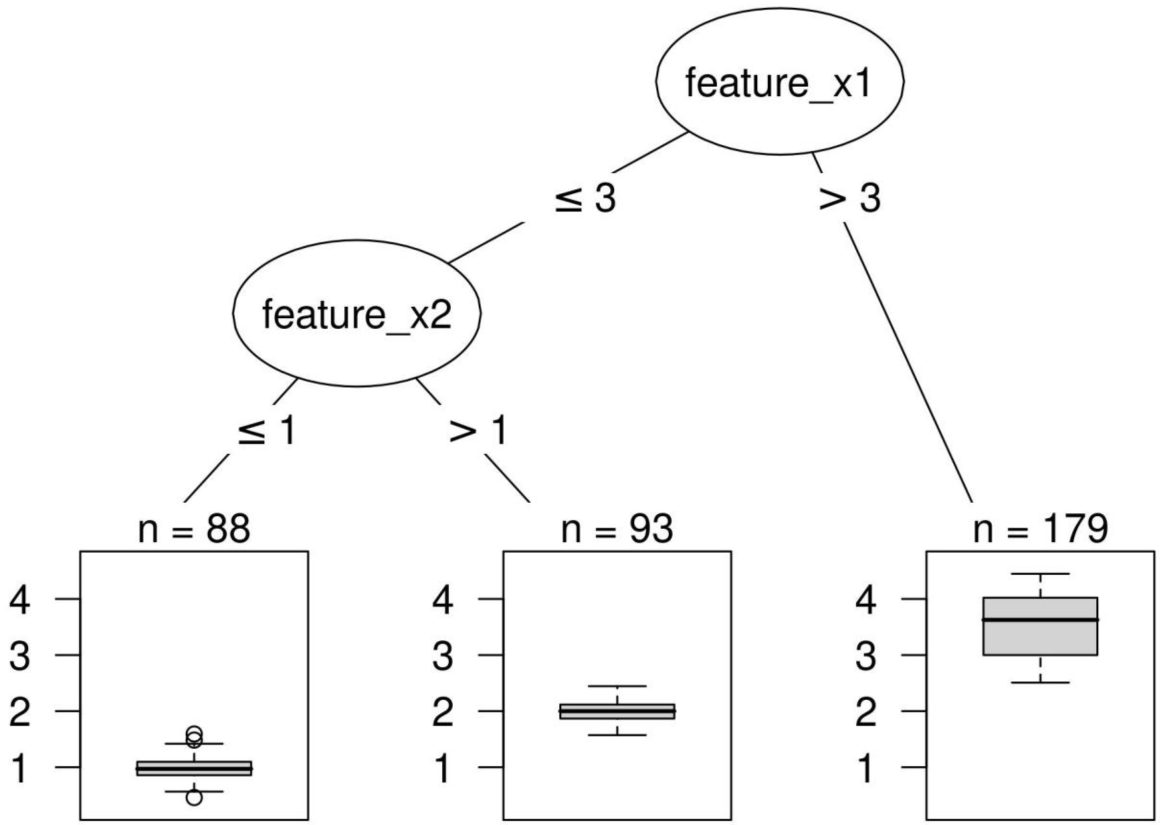
$$d(\mathbf{u},\mathbf{v}) = (\sum|u_i - v_i|^2)^{\frac{1}{2}} = ||\mathbf{u}-\mathbf{v}||_2$$ *Euclidean (L2 norm)*

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

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

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

Decision Tree



4. Model Evaluation

Confusion Matrix

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

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

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

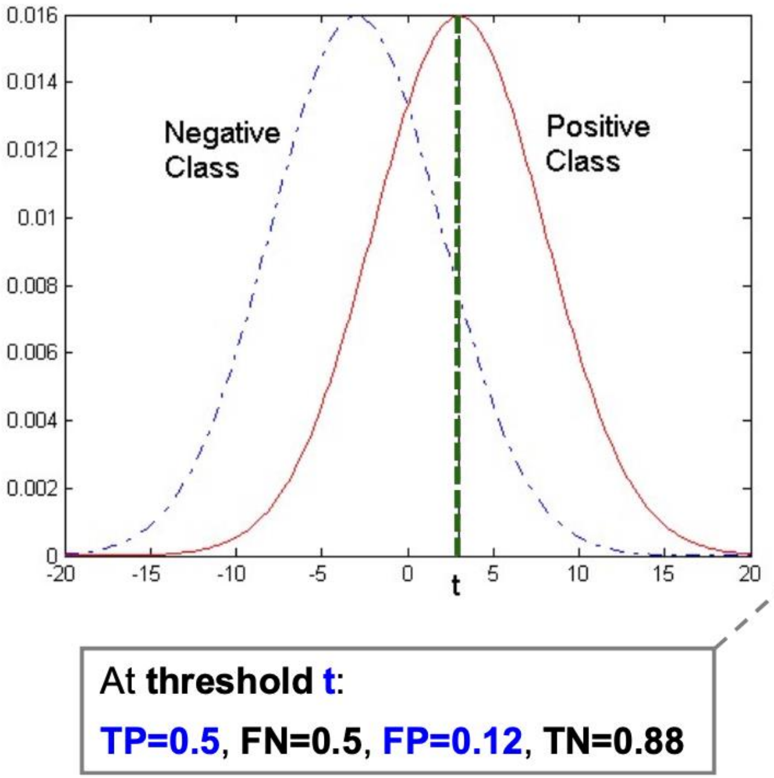
[Harmonized mean(조화 평균)]

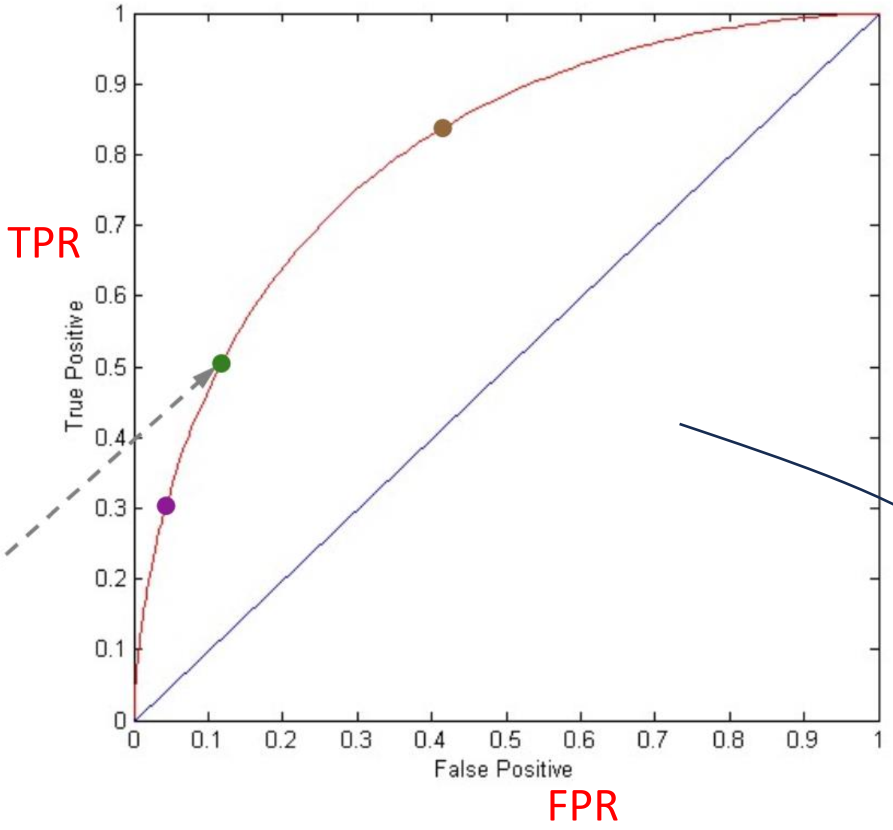
$$\frac{1}{F1\ score} = 0.5\left(\frac{1}{precision} + \frac{1}{recall}\right)$$

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

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

5. 이론과 응용

이론과 응용

Classification 문제 가정

A 회사에서, 직원이 이직할지(1) 안할지(0)를 분류 예측하는 모델을 개발하고자 함.
> df.head(6)

이름	성별	나이	입사연도	부서	근무평가	초과 근무 시간	결혼 여부	만나이	몸무게	입사 여부
김하나	여	28	2021	가	B	5	Y	27	55	1
이둘	남	25	2019	나	A	10	N	23	78	0
박삼식	남	35	2014	나	C	3	N	33	88	1
강너울	남	44	2011	나	B	5	Y	43	81	1
한다섯	여	31	2017	가	A	12	N	30	63	0
육개장	남	29	2022	가	A	19	Y	27	79	0

생각해보기 - 모델링까지, 어떤 과정을 거쳐야 할까?

이론과 응용

Classification 문제

LG 스마트공장 제품 품질 분류 AI 모델 개발 (https://dacon.io/competitions/official/236055/overview/description)

train.csv | test.csv | sample_submission.csv

Views | Grid view | Hide fields | Filter | Group | Sort

	PRODUCT_ID	Y_Class	Y_Quality	TIMESTAMP	LINE	PRODUCT_CODE	X_1
1	TRAIN_000	1	0.53343333	2022-06-13 05:14	T050304	A_31	
2	TRAIN_001	2	0.54181905	2022-06-13 05:22	T050307	A_31	
3	TRAIN_002	1	0.53126667	2022-06-13 05:30	T050304	A_31	
4	TRAIN_003	2	0.53732540	2022-06-13 05:39	T050307	A_31	
5	TRAIN_004	1	0.53159048	2022-06-13 05:47	T050304	A_31	
6	TRAIN_005	2	0.53783333	2022-06-13 05:55	T050307	A_31	
7	TRAIN_006	1	0.53366508	2022-06-13 06:03	T050304	A_31	
8	TRAIN_007	2	0.54000317	2022-06-13 06:11	T050307	A_31	
9	TRAIN_008	1	0.53182064	2022-06-13 06:19	T050304	A_31	
10	TRAIN_009	2	0.53804921	2022-06-13 06:28	T050307	A_31	
10 records		Sum 15	Sum 5.35680635				

Airtable | Copy base | View larger version

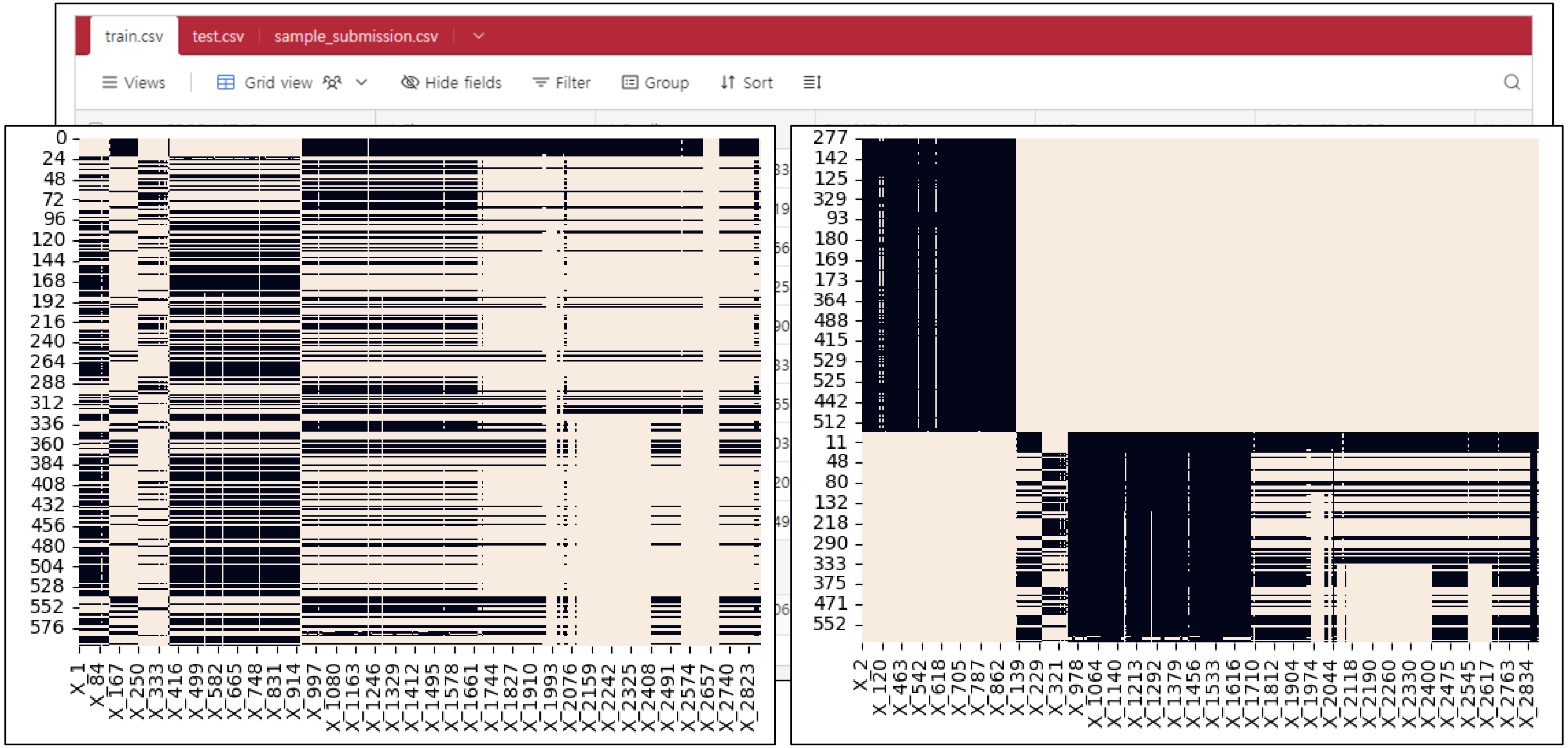
고민해봐야 할 포인트

- 비식별화된 설명변수
- 행 개수 < 열 개수
- 차원 축소 방법
- 적절한 모델

이론과 응용

Classification 문제

LG 스마트공장 제품 품질 분류 AI 모델 개발 (https://dacon.io/competitions/official/236055/overview/description)



고민해봐야 할 포인트

- 비식별화된 설명변수
- 행 개수 < 열 개수
- 차원 축소 방법
- 적절한 모델

이론과 응용

Classification 코드로 구현하기

변수선택? Feature importance, SHAP value

모델선택? AutoML (pycaret)

```
from sklearn.naive_bayes import GaussianNB # model 생성
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
from sklearn.linear_model import SGDClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from xgboost as xgb # xgb.XGBClassifier()

from sklearn.model_selection import train_test_split # train/test set
from sklearn.metrics import accuracy_score, confusion_matrix # model 평가

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2, random_state = 1)
model = GaussianNB() # 하이퍼파라미터 튜닝

model.fit(X=x_train, y=y_train) # 모델 학습
y_pred = model.predict(x_test) # 예측치
con_mat = confusion_matrix(y_true, y_pred)
```

GridsearchCV, BayessearchCV
Optuna 패키지

수고하셨습니다!

해당 세션자료는 KUBIG Github에서 보실 수 있습니다!

다음은 이번 주차 과제 설명이 있습니다!