SQUARESPACE.COM/LOGO,
ICONS BY THE NOUN PROJECT

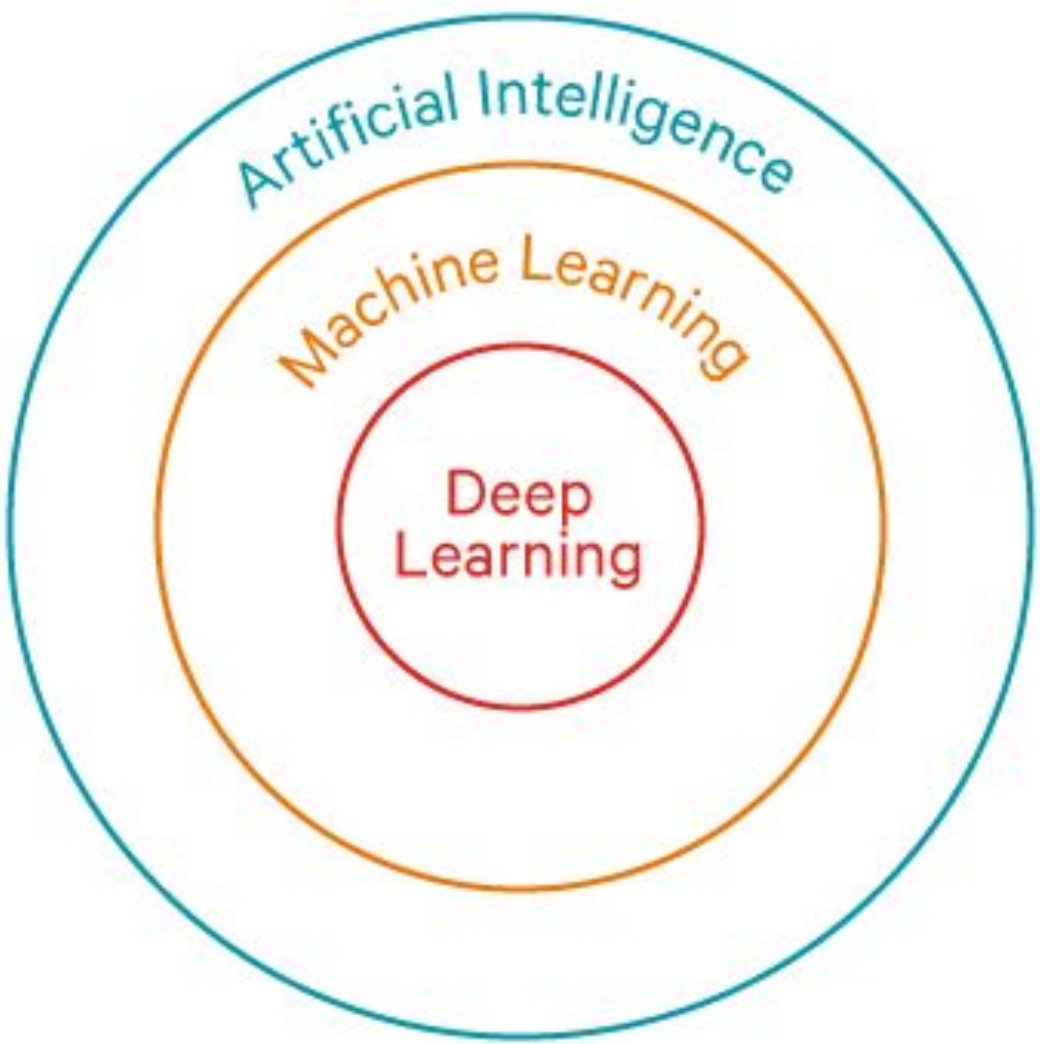
머신러닝 스터디

이 혁

CONTENTS

Chapter 1

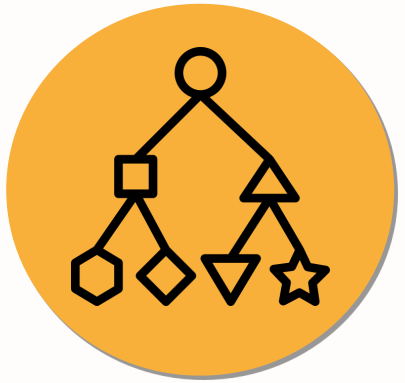
- Basic -


Artificial Intelligence
Machine Learning
Deep Learning

Chap. 1

Supervised Learning

Unsupervised Learning



Classification

Regression


scikit
learn

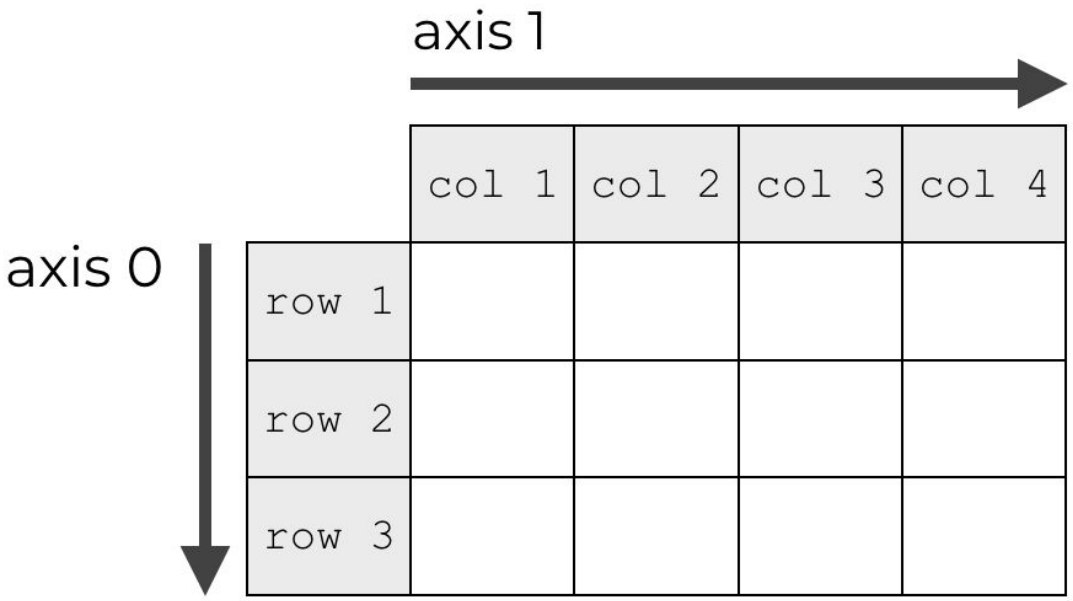

NumPy


pandas


matplotlib

Chap. 1

NumPy = Numerical Python

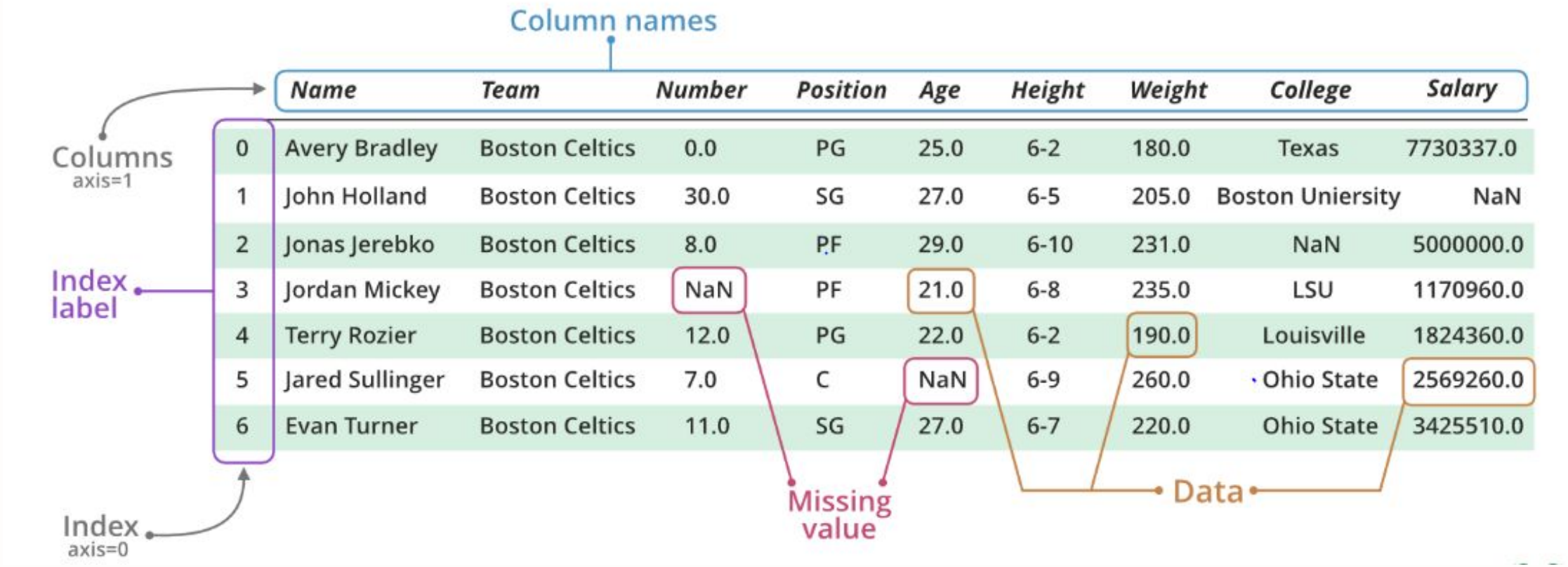


NumPy

```
import numpy as np

array1 = np.array([1,2,3])
print('array1 type:',type(array1))
print('array1 array 형태:',array1.shape)

array2 = np.array([[1,2,3],[2,3,4]])
print('array2 type:',type(array2))
print('array2 array 형태:',array2.shape)

array3 = np.array([[1,2,3]])
print('array3 type:',type(array3))
print('array3 array 형태:',array3.shape)
```

```
array1 type: <class 'numpy.ndarray'>
array1 array 형태: (3,)
array2 type: <class 'numpy.ndarray'>
array2 array 형태: (2, 3)
array3 type: <class 'numpy.ndarray'>
array3 array 형태: (1, 3)
```

pandas



	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0.0	PG	25.0	6-2	180.0	Texas	7730337.0
1	John Holland	Boston Celtics	30.0	SG	27.0	6-5	205.0	Boston Uniersity	NaN
2	Jonas Jerebko	Boston Celtics	8.0	PF	29.0	6-10	231.0	NaN	5000000.0
3	Jordan Mickey	Boston Celtics	NaN	PF	21.0	6-8	235.0	LSU	1170960.0
4	Terry Rozier	Boston Celtics	12.0	PG	22.0	6-2	190.0	Louisville	1824360.0
5	Jared Sullinger	Boston Celtics	7.0	C	NaN	6-9	260.0	Ohio State	2569260.0
6	Evan Turner	Boston Celtics	11.0	SG	27.0	6-7	220.0	Ohio State	3425510.0

pandas

```
import pandas as pd

titanic_df = pd.read_csv('titanic_train.csv')
print('titanic 변수 type:',type(titanic_df))
titanic_df
```

```
titanic 변수 type: <class 'pandas.core.frame.DataFrame'>
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN	Q
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750	NaN	S
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.1333	NaN	S
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0708	NaN	C
10	11	1	3	Sandstrom, Miss. Marguerite Rut	female	4.0	1	1	PP 9549	16.7000	G6	S
11	12	1	1	Bonnell, Miss. Elizabeth	female	58.0	0	0	113783	26.5500	C103	S
12	13	0	3	Saundercock, Mr. William Henry	male	20.0	0	0	A/5. 2151	8.0500	NaN	S
13	14	0	3	Andersson, Mr. Anders Johan	male	39.0	1	5	347082	31.2750	NaN	S

Chapter 2

- Scikit Learn -

```python
from sklearn.datasets import load_iris
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
import pandas as pd

iris = load_iris()

iris_data = iris.data
iris_label = iris.target
print('iris target값:', iris_label)
print('iris target명:', iris.target_names)
```

```
iris target값: [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2]
iris target명: ['setosa' 'versicolor' 'virginica']
```

```
iris_df = pd.DataFrame(data=iris_data, columns=iris.feature_names)
iris_df['label'] = iris.target
iris_df
```

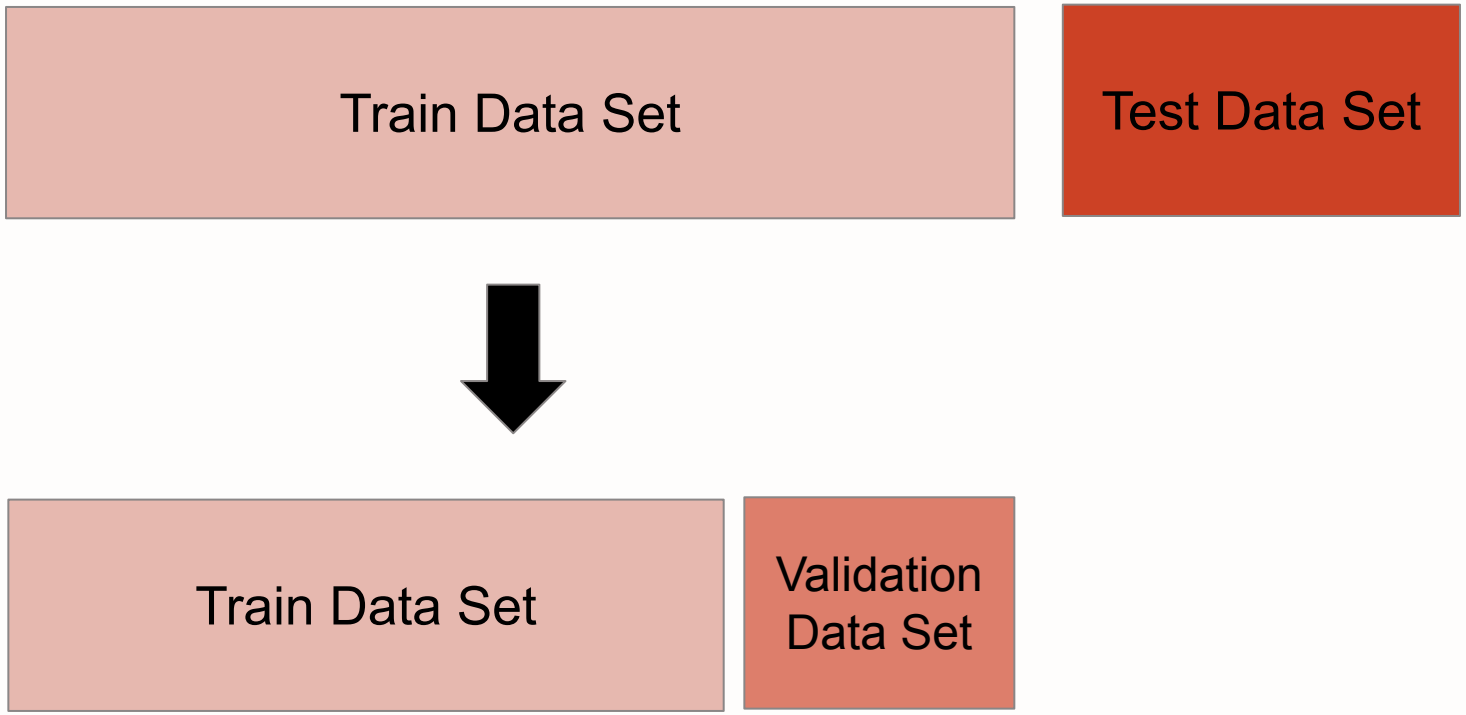
	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	label
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	2
146	6.3	2.5	5.0	1.9	2
147	6.5	3.0	5.2	2.0	2
148	6.2	3.4	5.4	2.3	2
149	5.9	3.0	5.1	1.8	2

150 rows × 5 columns

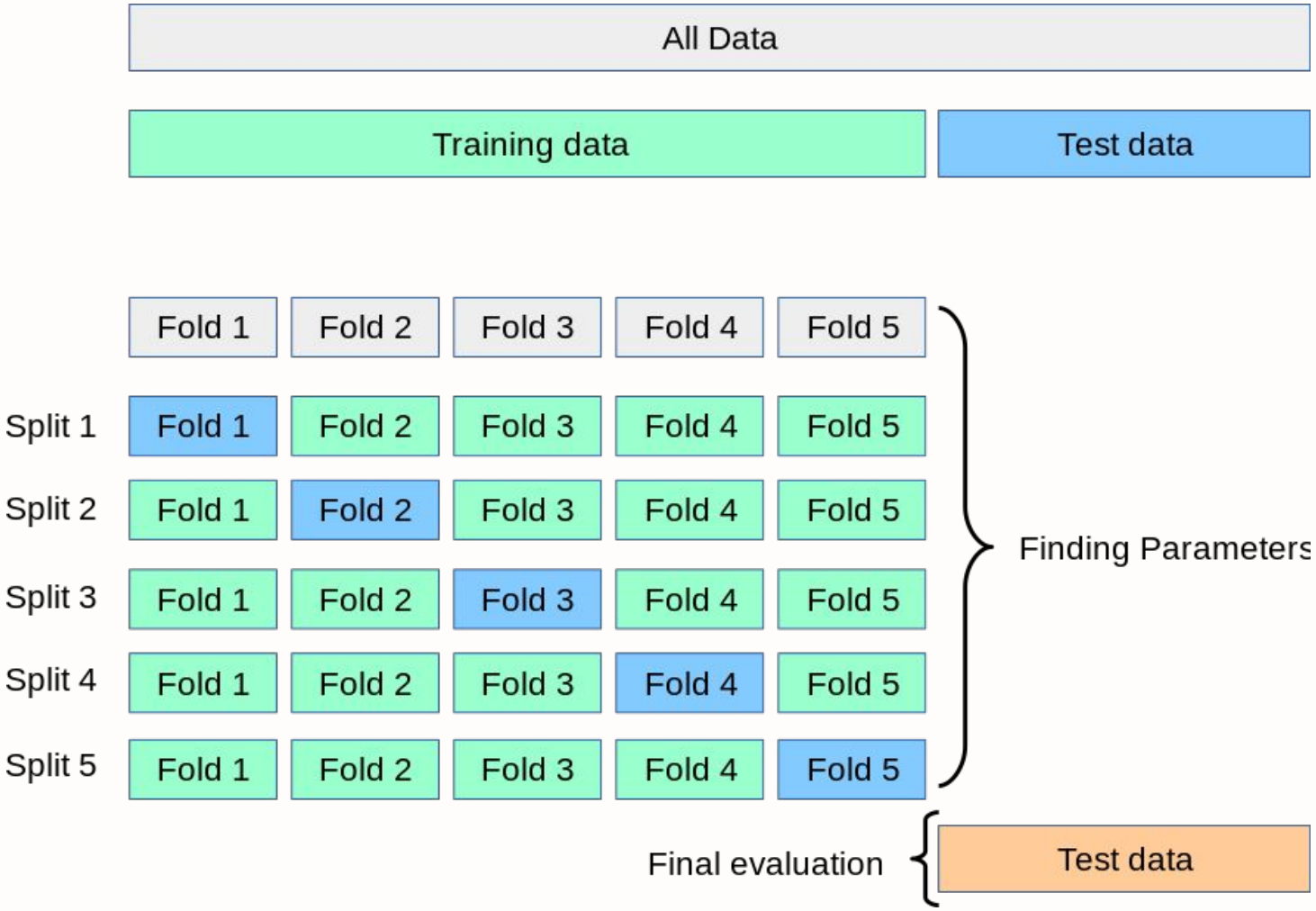
```
X_train, X_test, y_train, y_test = train_test_split(iris_data,iris_label,test_size=0.2,random_state=11)

# DecisionTreeClassifier 객체 생성
dt_clf = DecisionTreeClassifier(random_state=11)

#학습 수행
dt_clf.fit(X_train, y_train)

#예측 수행
pred = dt_clf.predict(X_test)

print('예측 정확도: {0:.4f}'.format(accuracy_score(y_test,pred)))
```

예측 정확도: 0.9333

Chap. 2 Cross Validation

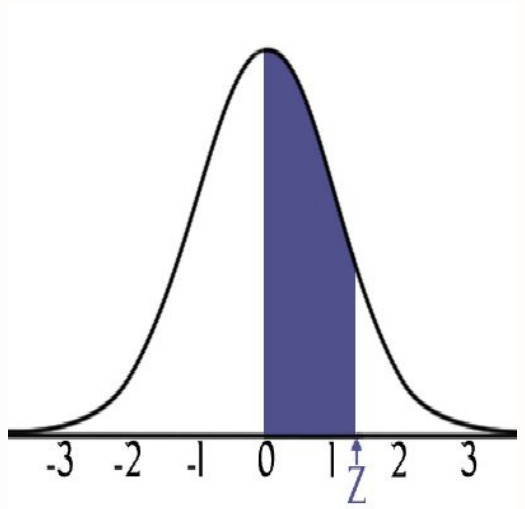


Chap. 2 K Fold Cross Validation



Chap. 2 Stratified K Fold

Chap. 2 데이터 전처리(1)



Label Encoding

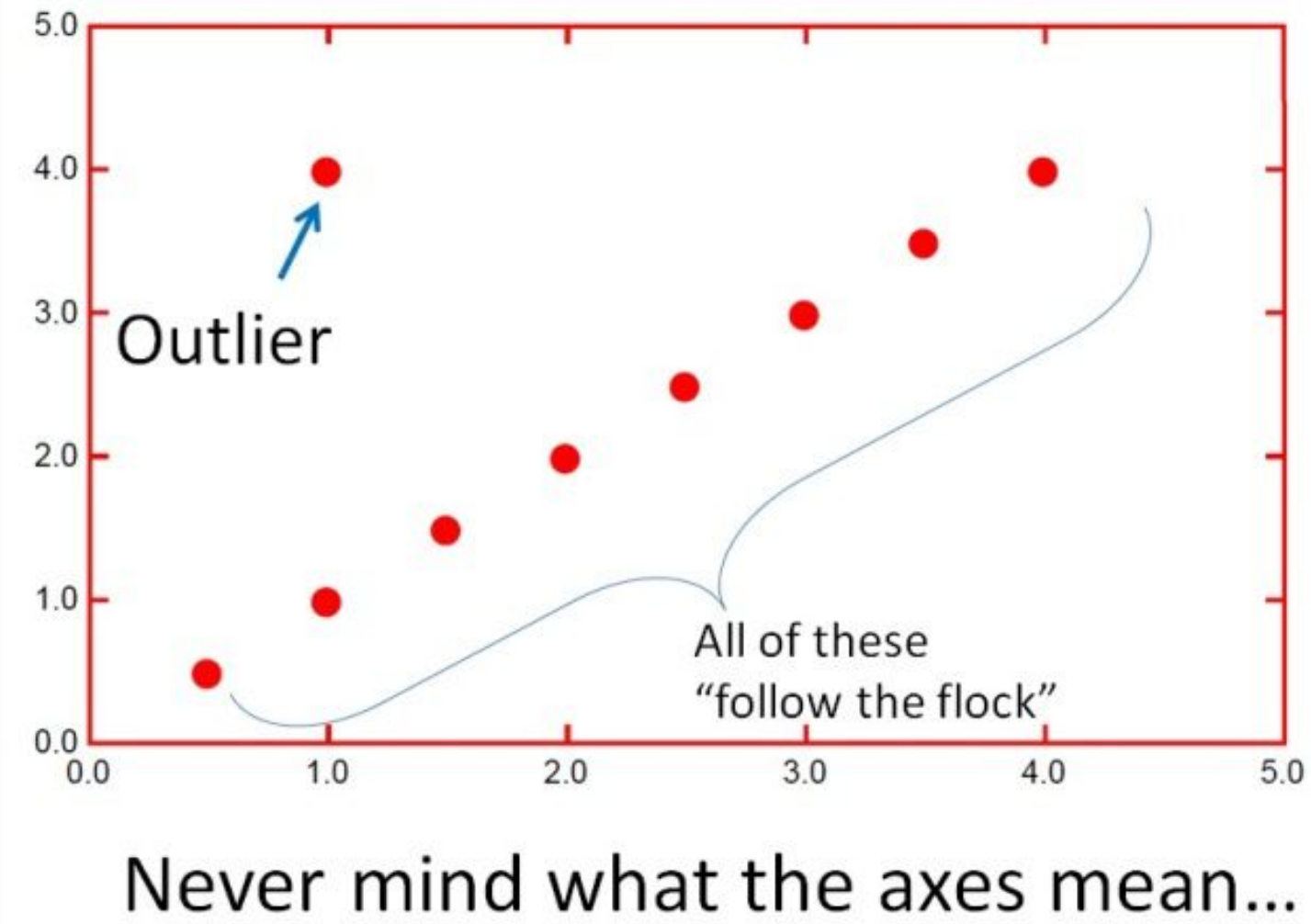
Food Name	Categorical #	Calories
Apple	1	95
Chicken	2	231
Broccoli	3	50

→

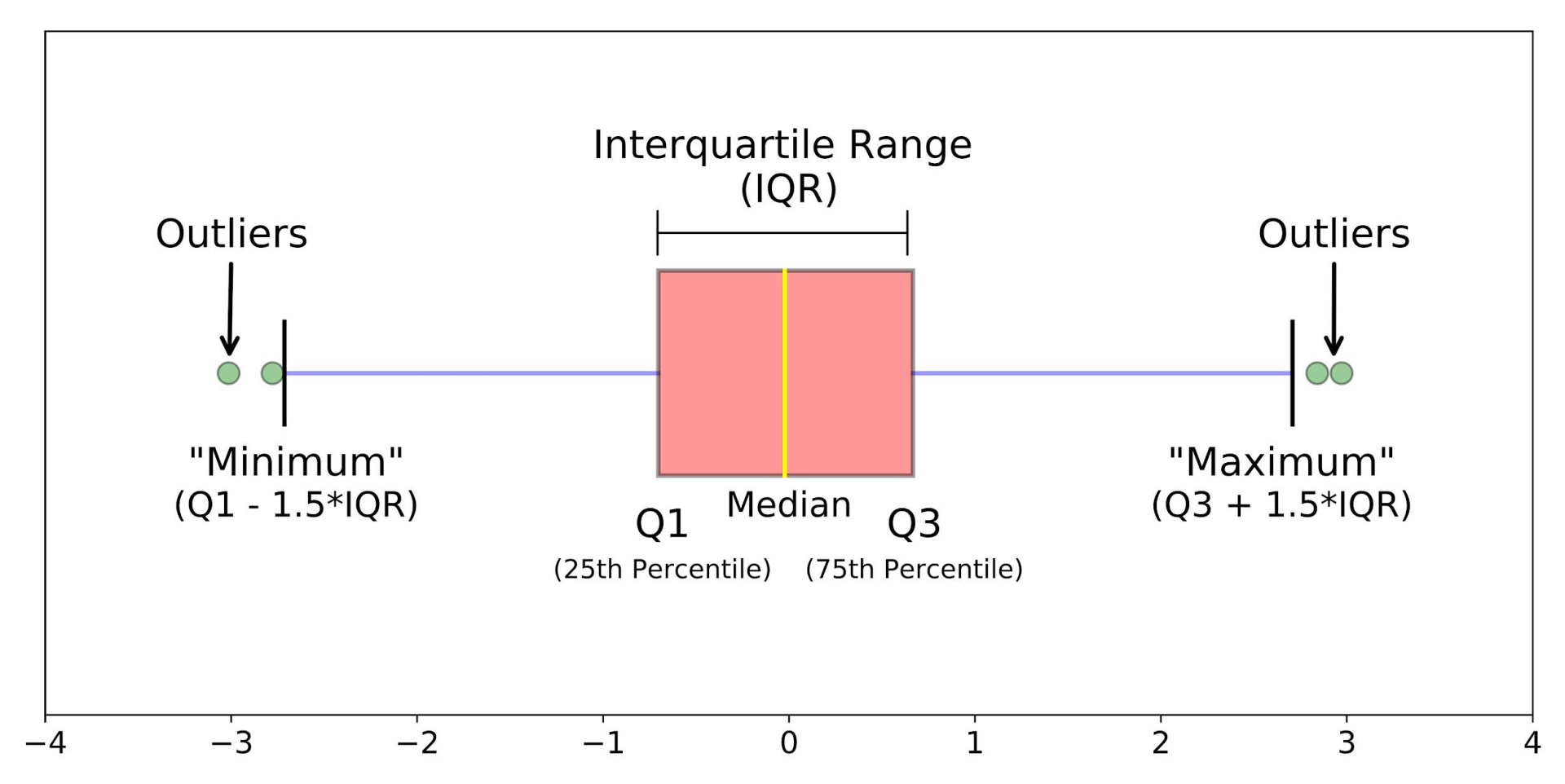
One Hot Encoding

Apple	Chicken	Broccoli	Calories
1	0	0	95
0	1	0	231
0	0	1	50

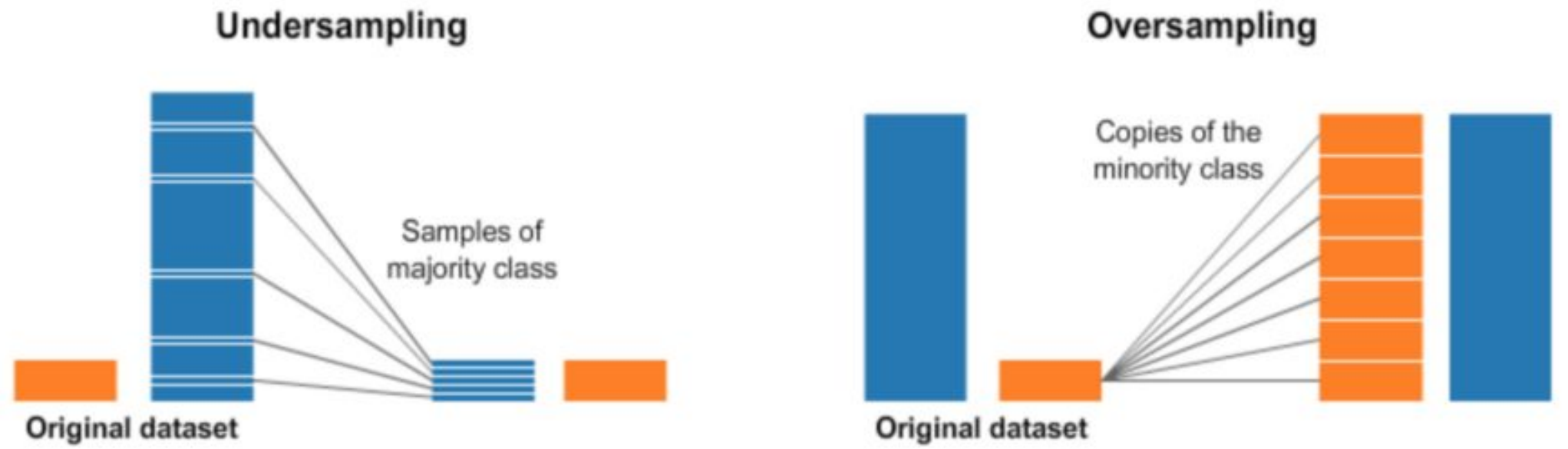
Chap. 2 데이터 전처리(2)



Chap. 4 데이터 전처리(3)



Chap. 4 데이터 전처리 (4)

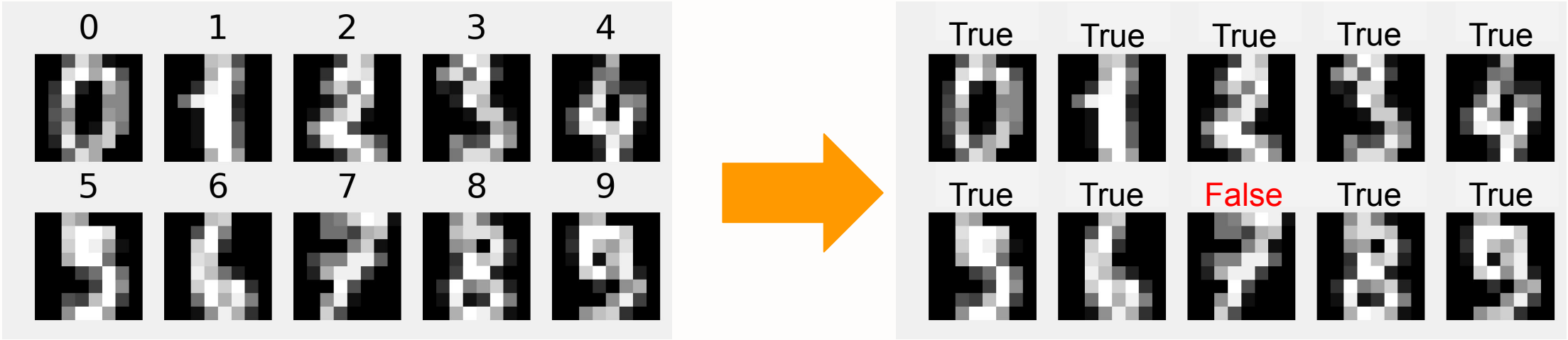


Chapter 3

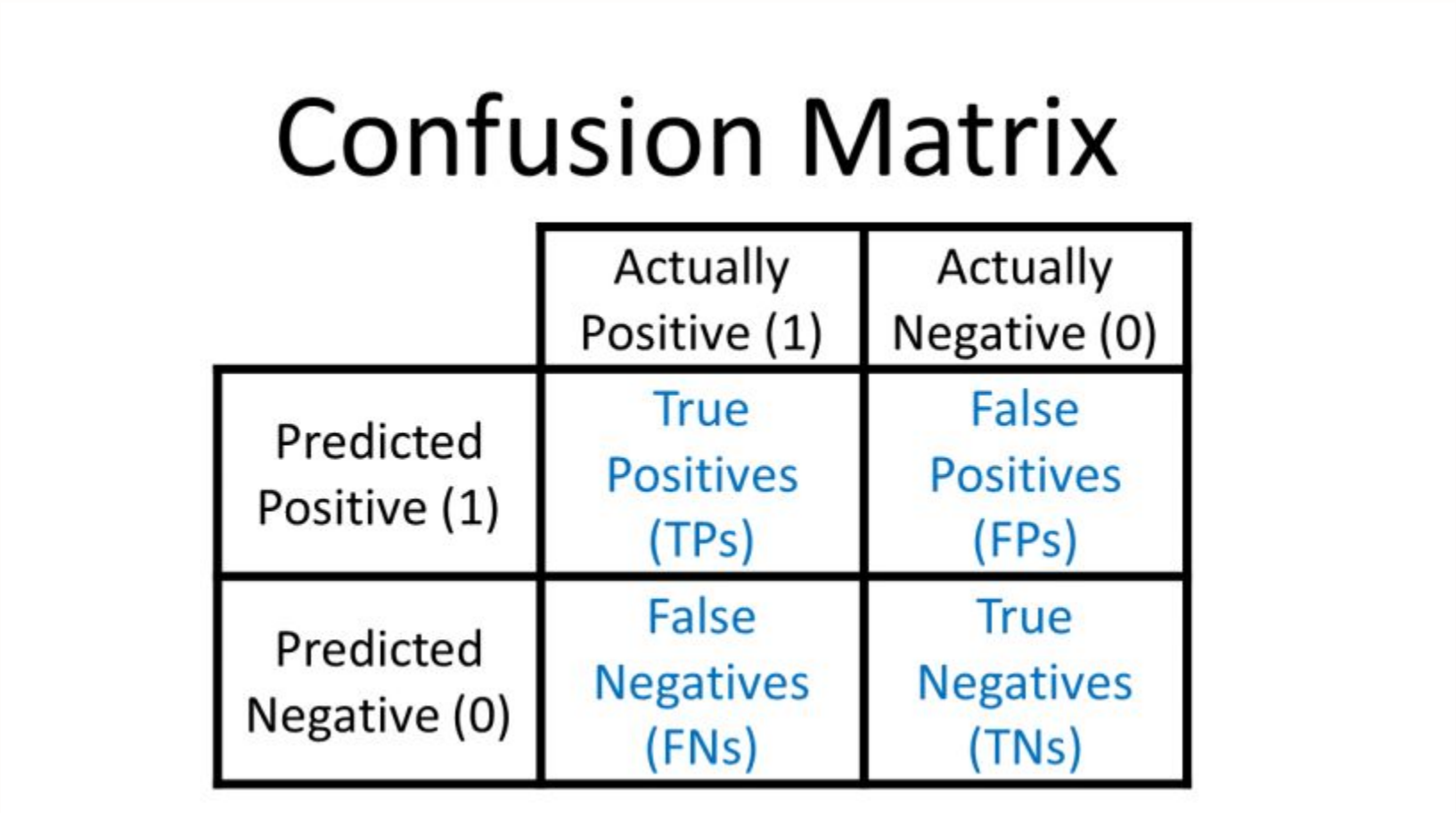
- Evaluation -

Chap. 3 Accuracy

$$\text{Accuracy} = \frac{\text{True } \cancel{\text{Positives}} + \text{True } \cancel{\text{Negatives}}}{\text{All Samples}}$$



Confusion Matrix



Precision & Recall (1)

Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

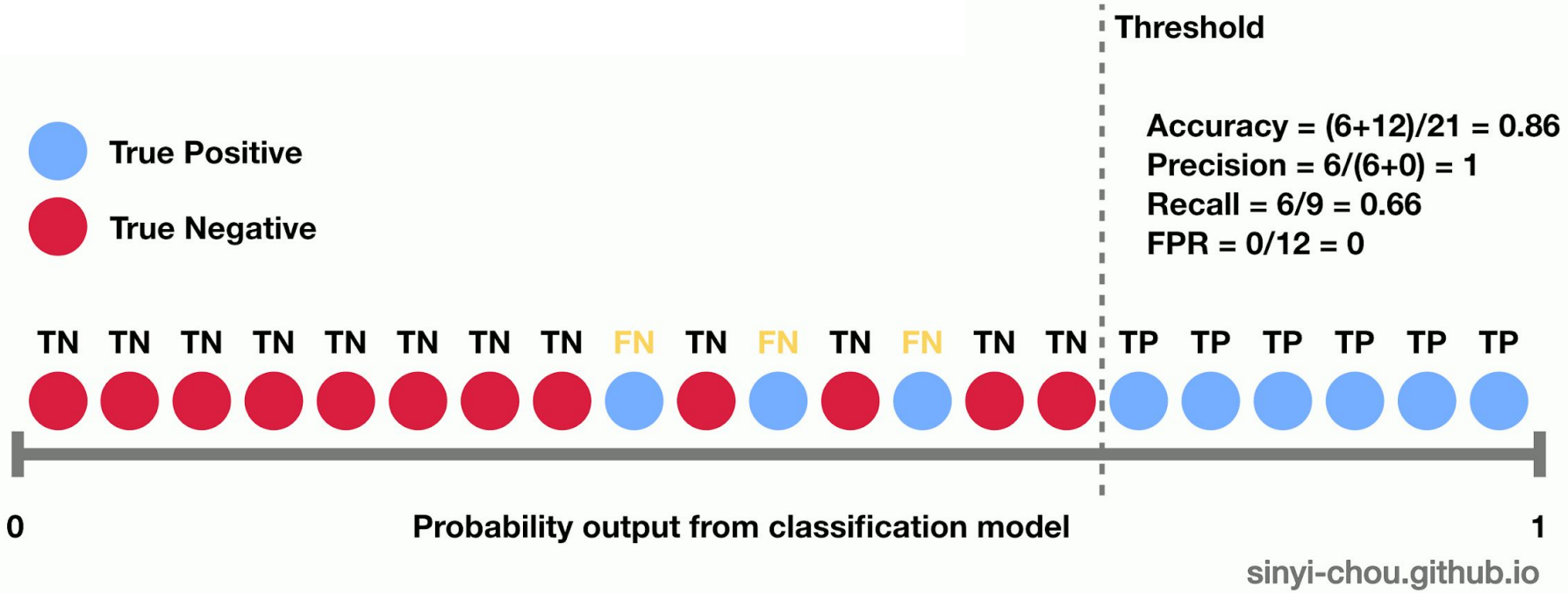
Precision

Recall

Precision & Recall (2)

$$\text{Precision} = \frac{\textit{True Positives}}{\textit{True Positives} + \textit{False Positives}}$$

$$\text{Recall} = \frac{\textit{True Positives}}{\textit{True Positives} + \textit{False Negatives}}$$



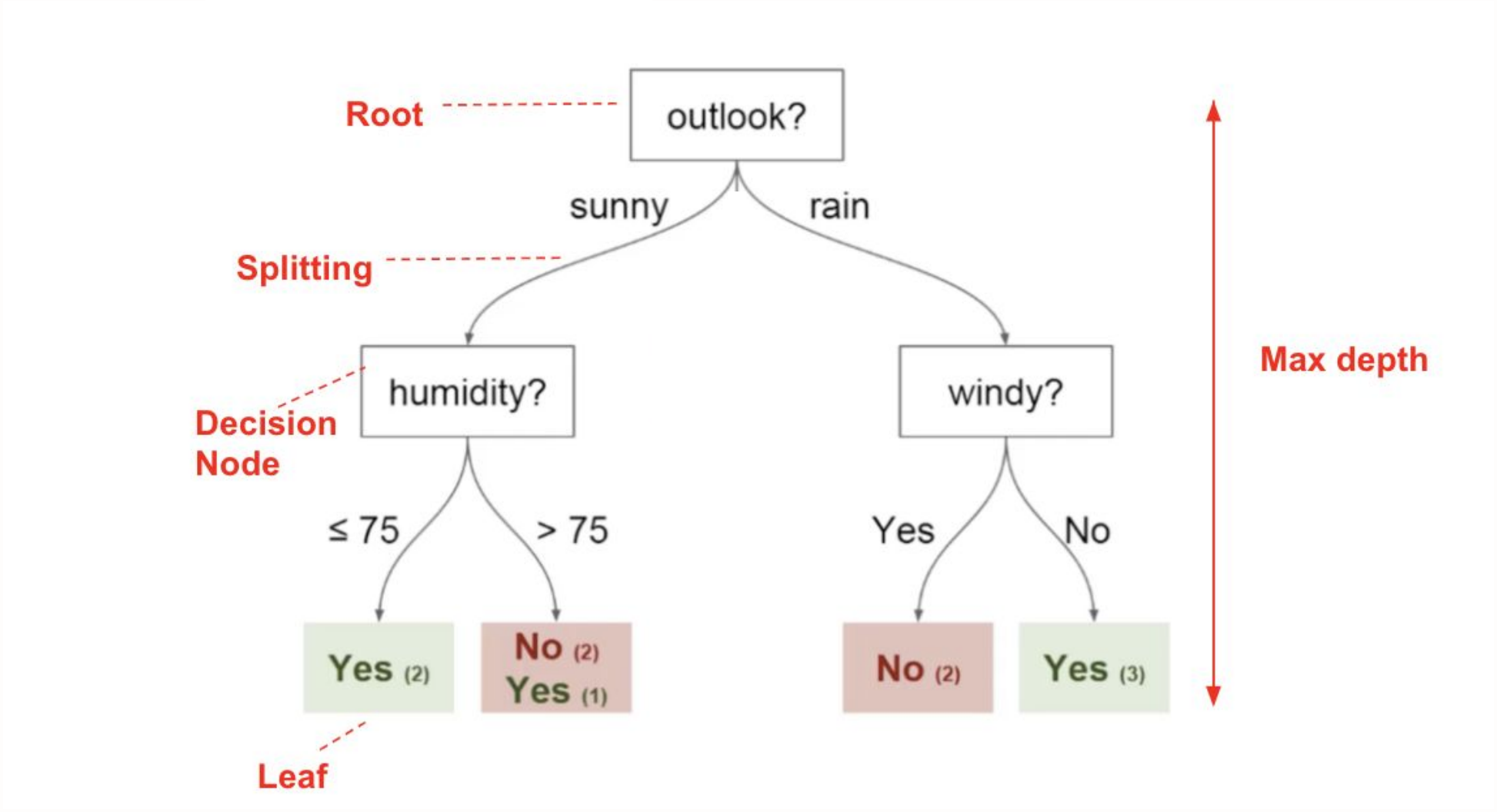
Chap. 3 F1 Score

$$\text{F1 Score} = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}}$$

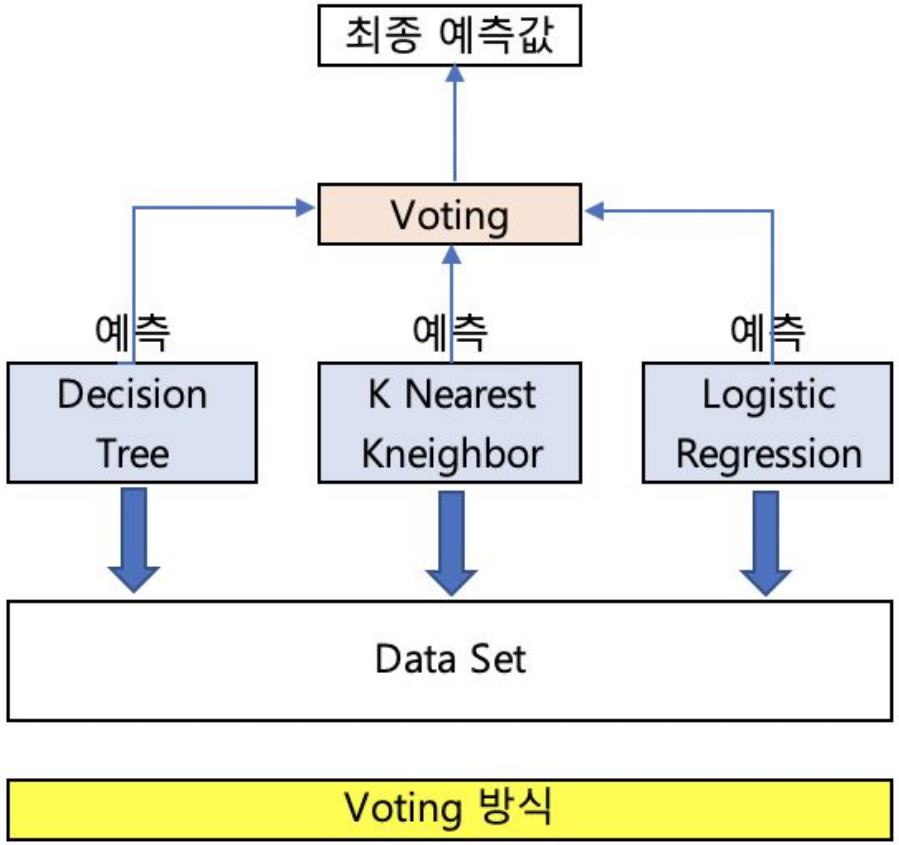
Chapter 4

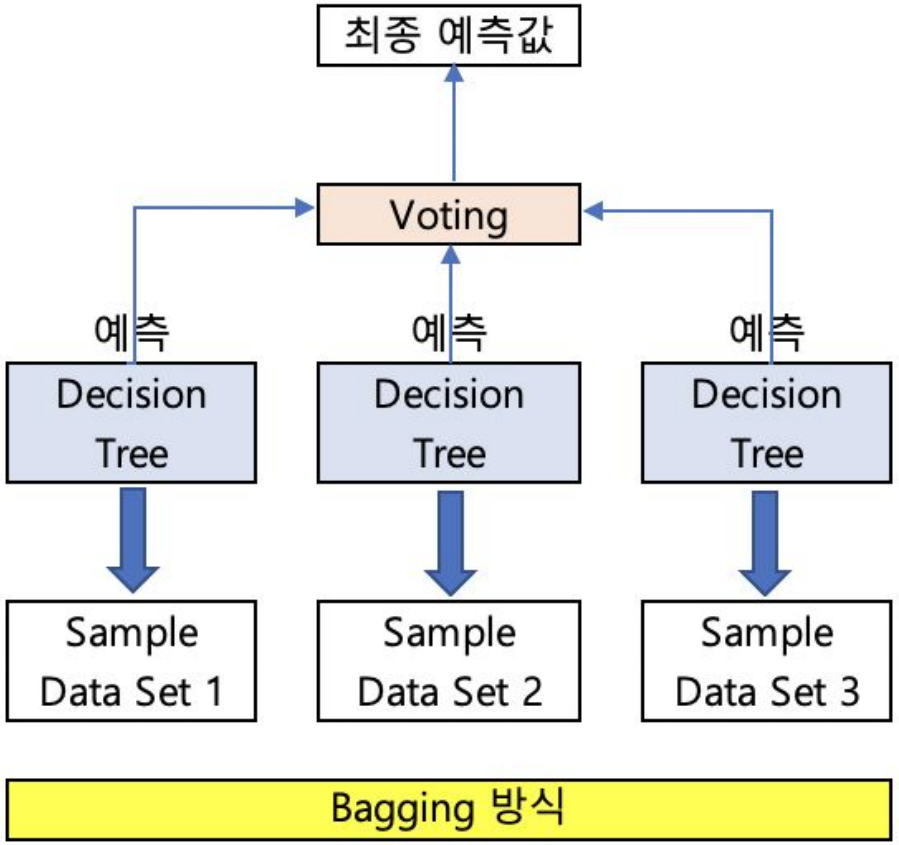
- Classification -

Chap. 4 Decision Tree

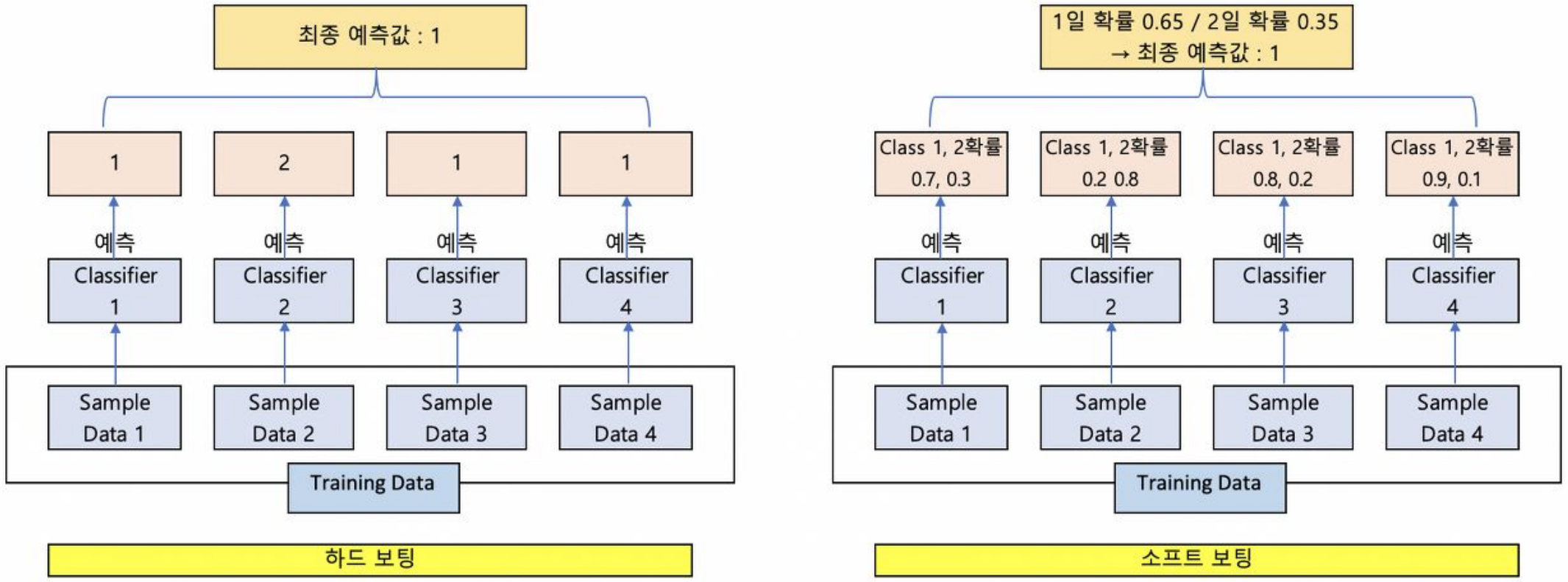


Chap. 4 Ensemble Learning (1)

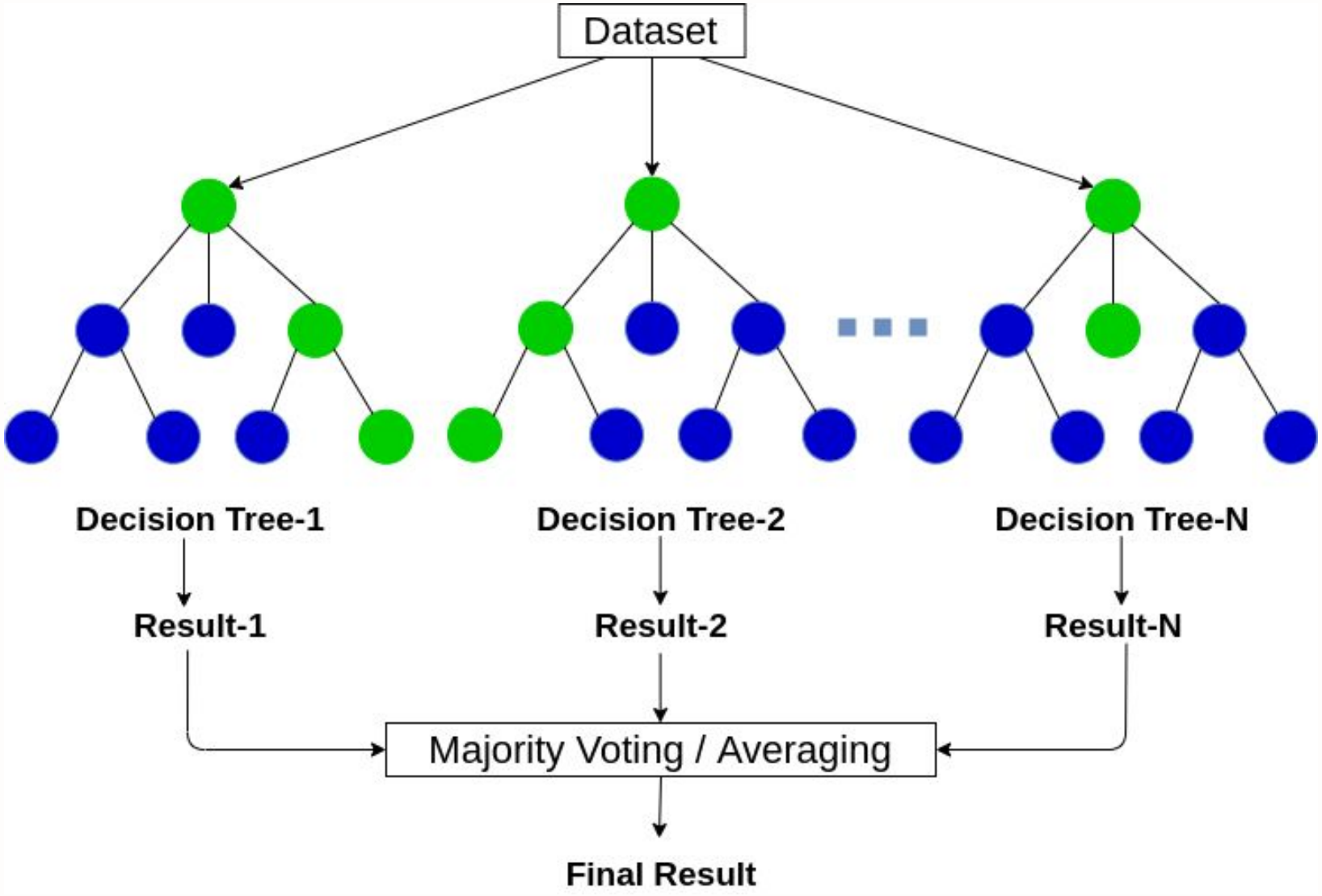




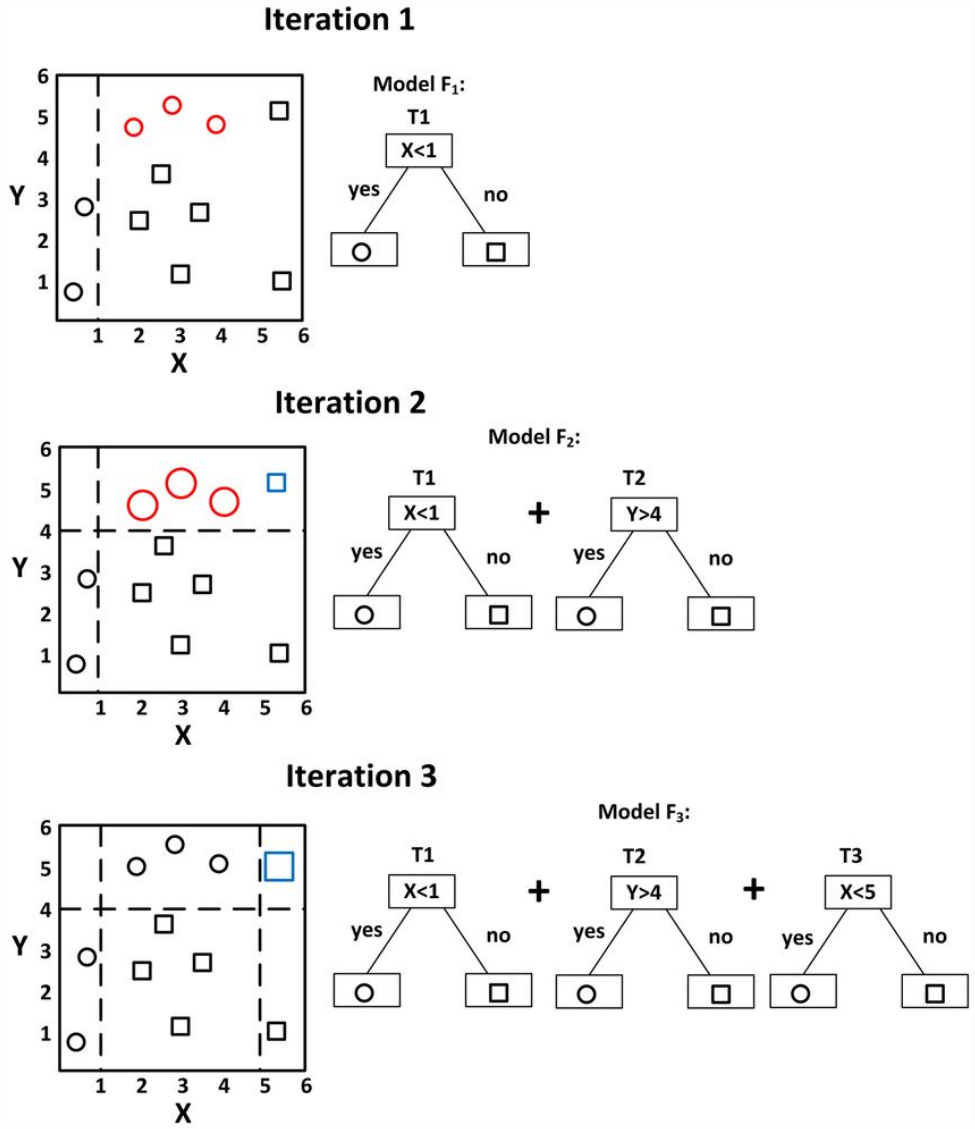
Chap. 4 Ensemble Learning (2)



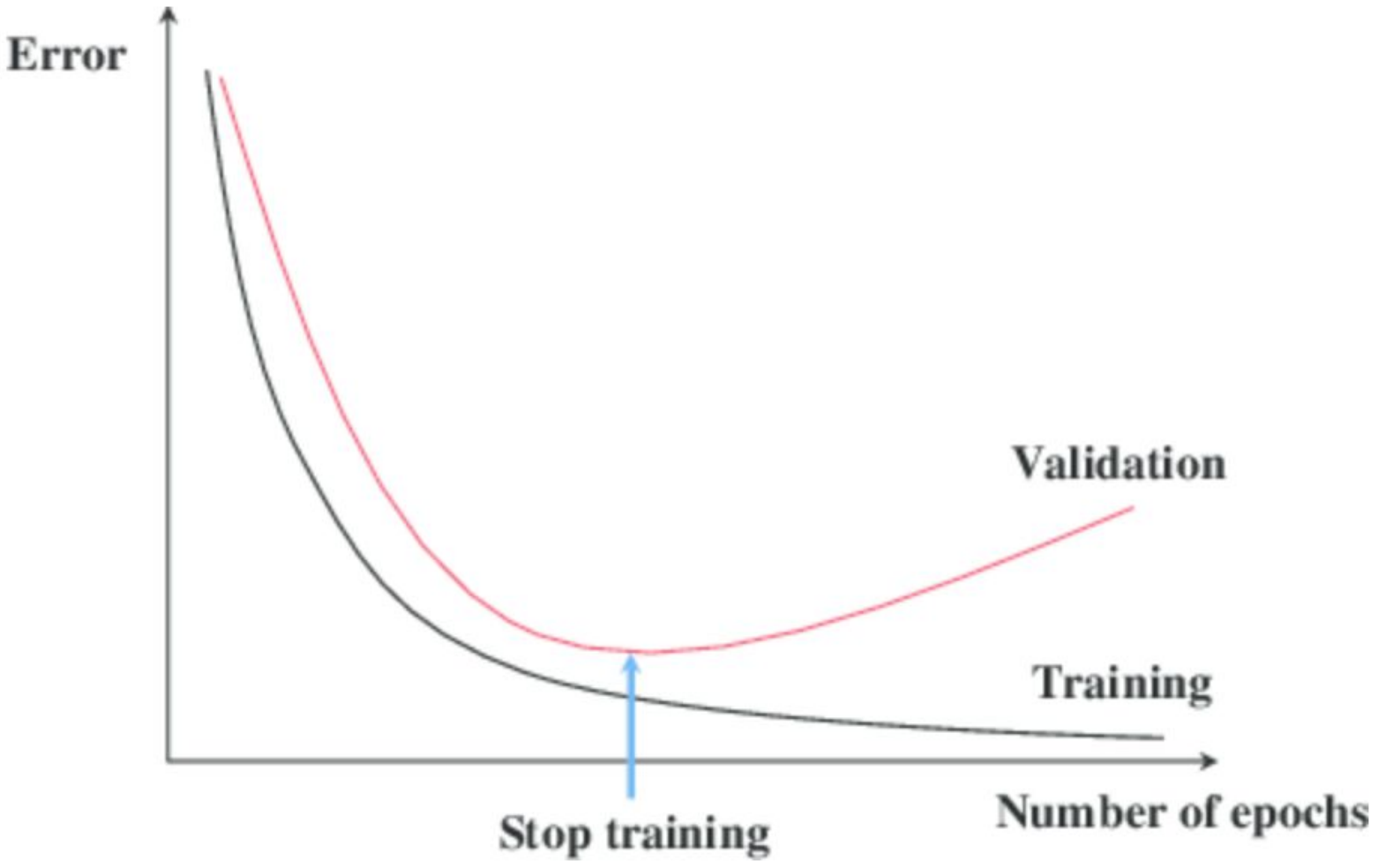
Chap. 4 Random Forest



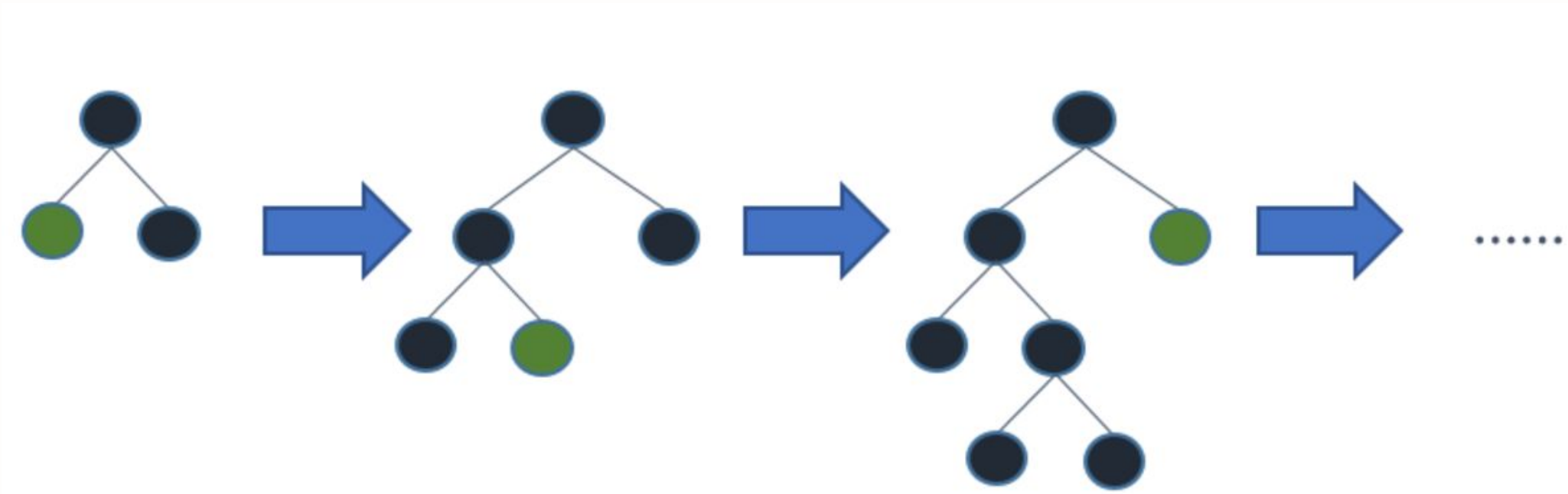
Chap. 4 GBM (Gradient Boosting Machine)



Chap. 4 XGBOOST (eXtra Gradient Boost)



Chap. 4 LightGBM (eXtra Gradient Boost)



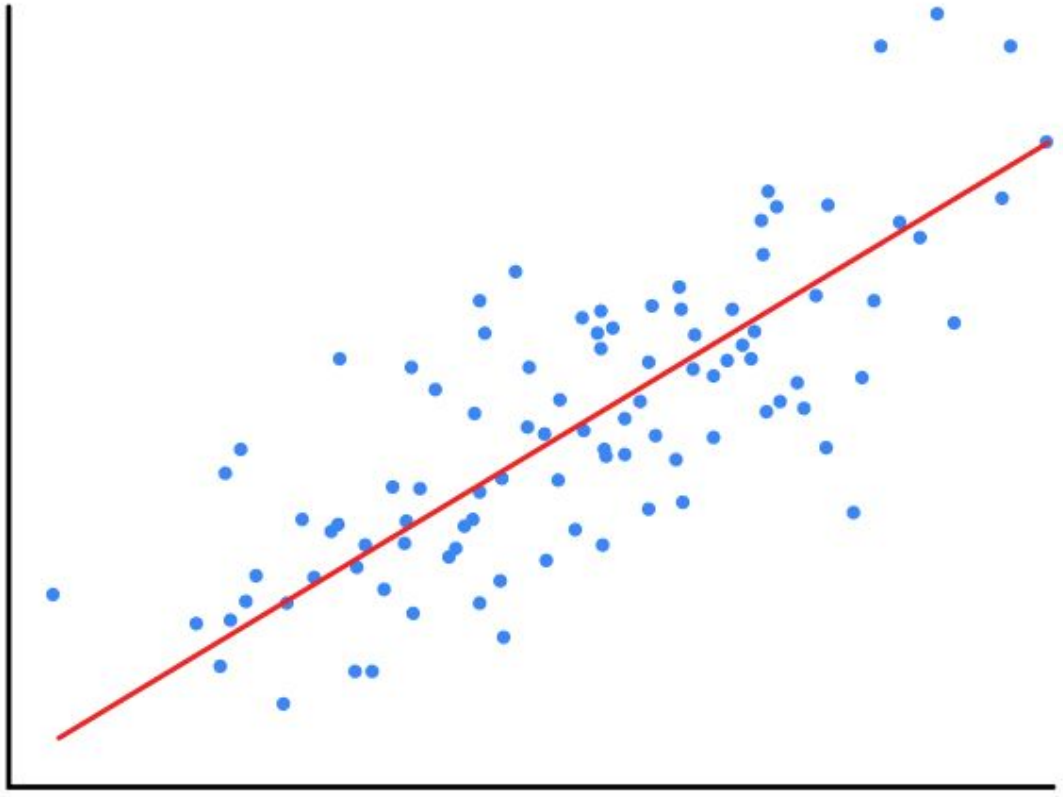
Leaf-wise tree growth

Chapter 5

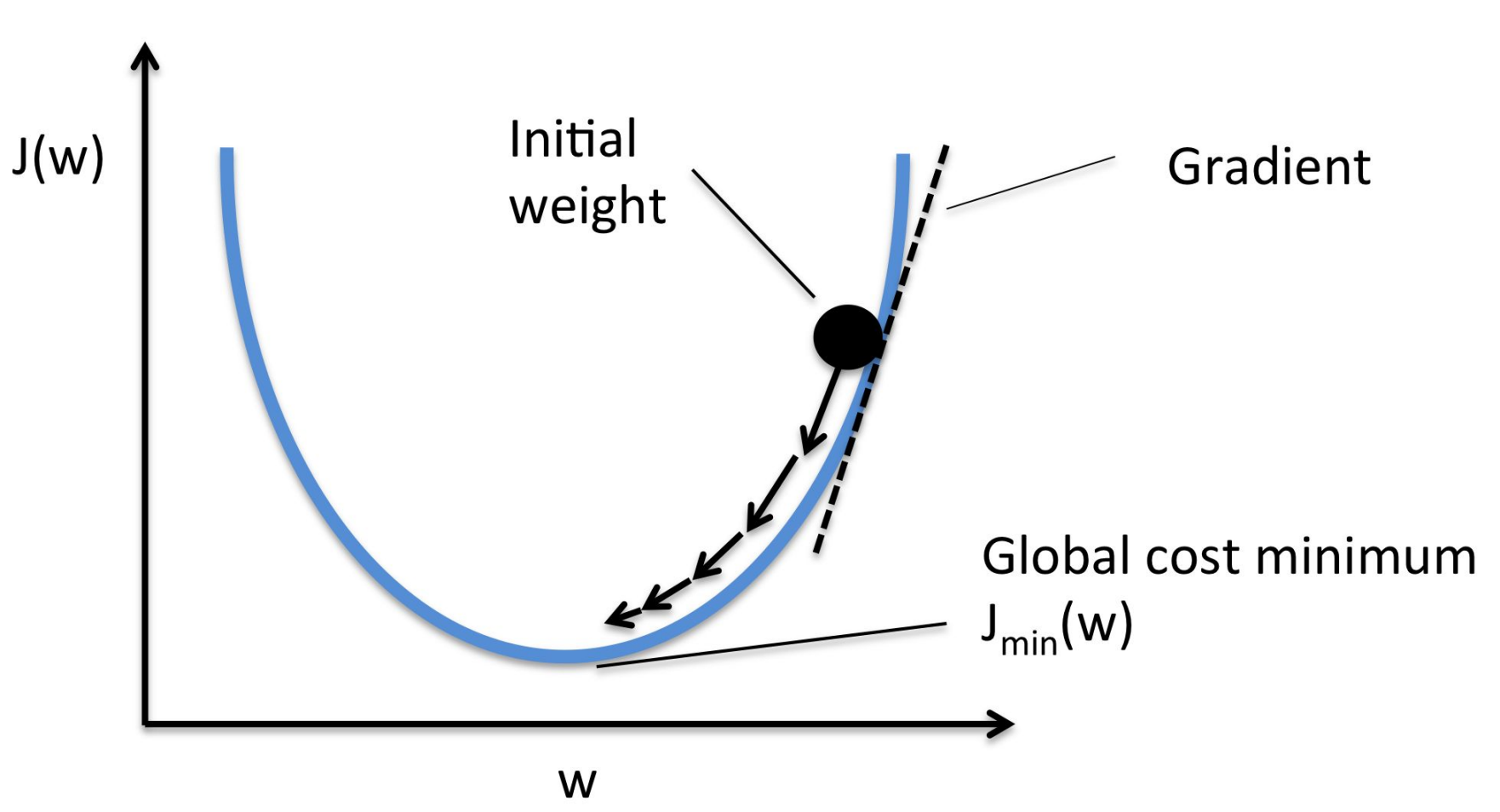
- Regression -

Chap. 5 Regression?

"데이터 값이 평균과 같은 일정한 값으로 돌아가려는 경향을 이용한 통계학 기법"



Gradient Descent



Regression Assessment

MAE
- Mean Absolute Error
- 실제 값과 예측값의 차이를 절댓값으로 변환해 평균

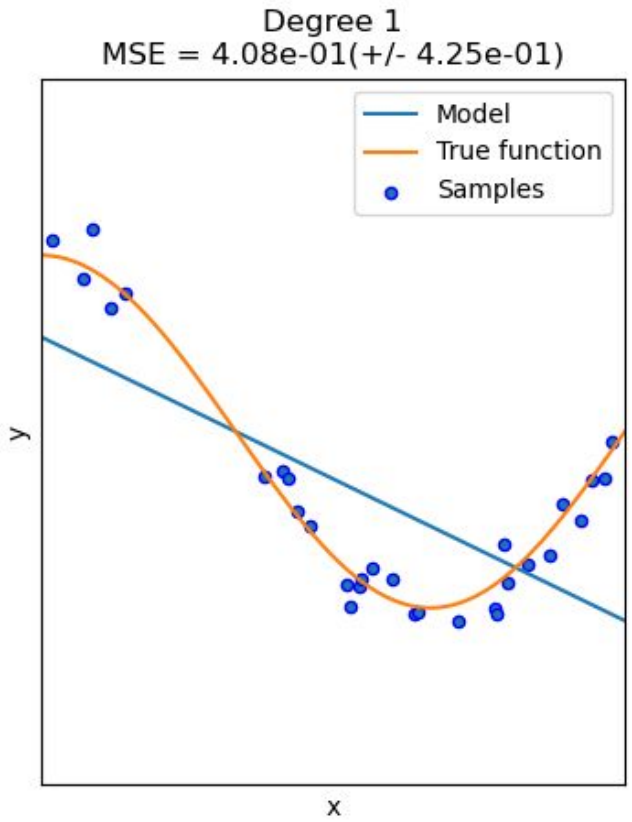
MSE
- Mean Squared Error
- 실제 값과 예측값의 차이를 제곱해 평균

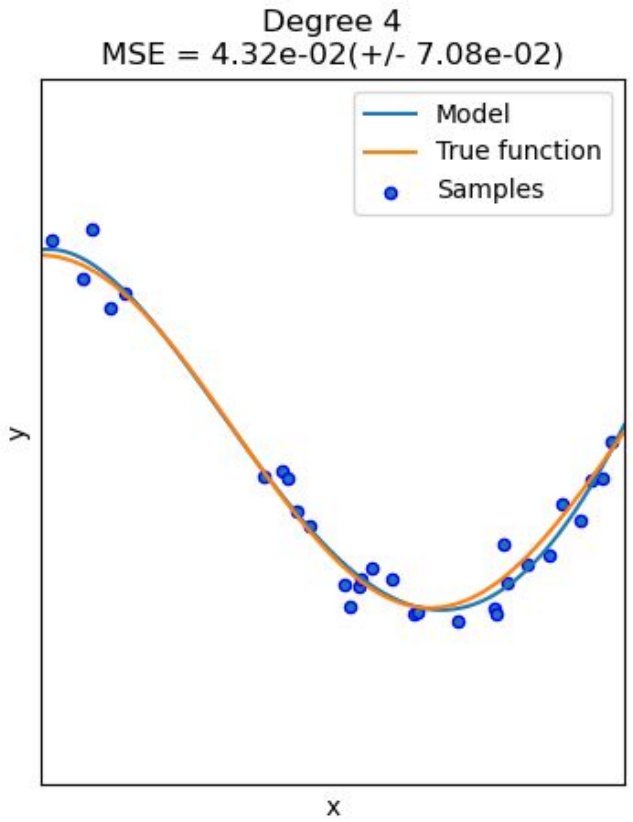
RMSE
- Root Mean Squared Error
- MSE에 루트를 씌움

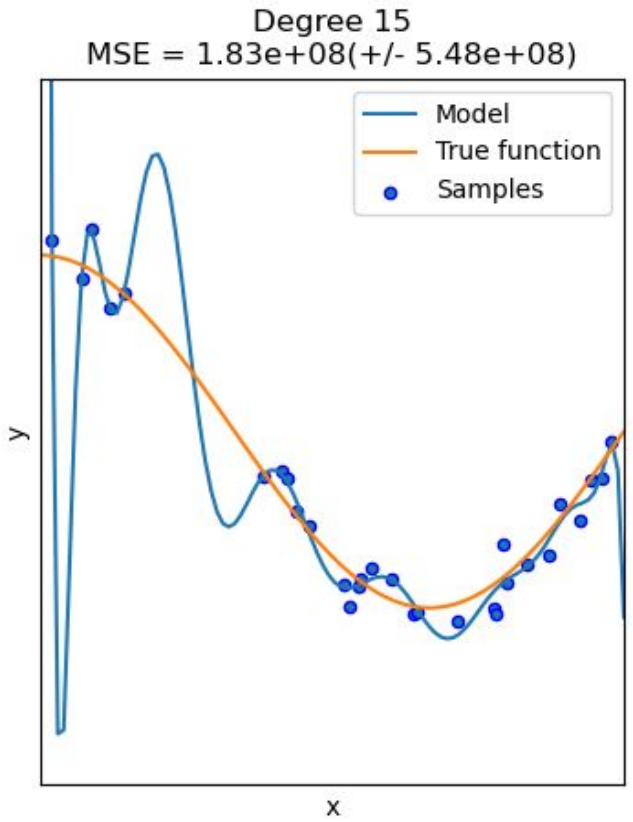
RMSLE
- Root Mean Squared Log Error
- RMSE의 각 인자에 로그화

R^2
- 실제 값의 분산 대비 예측값의 분산 비율
- 1에 가까울수록 예측 정확도 높음

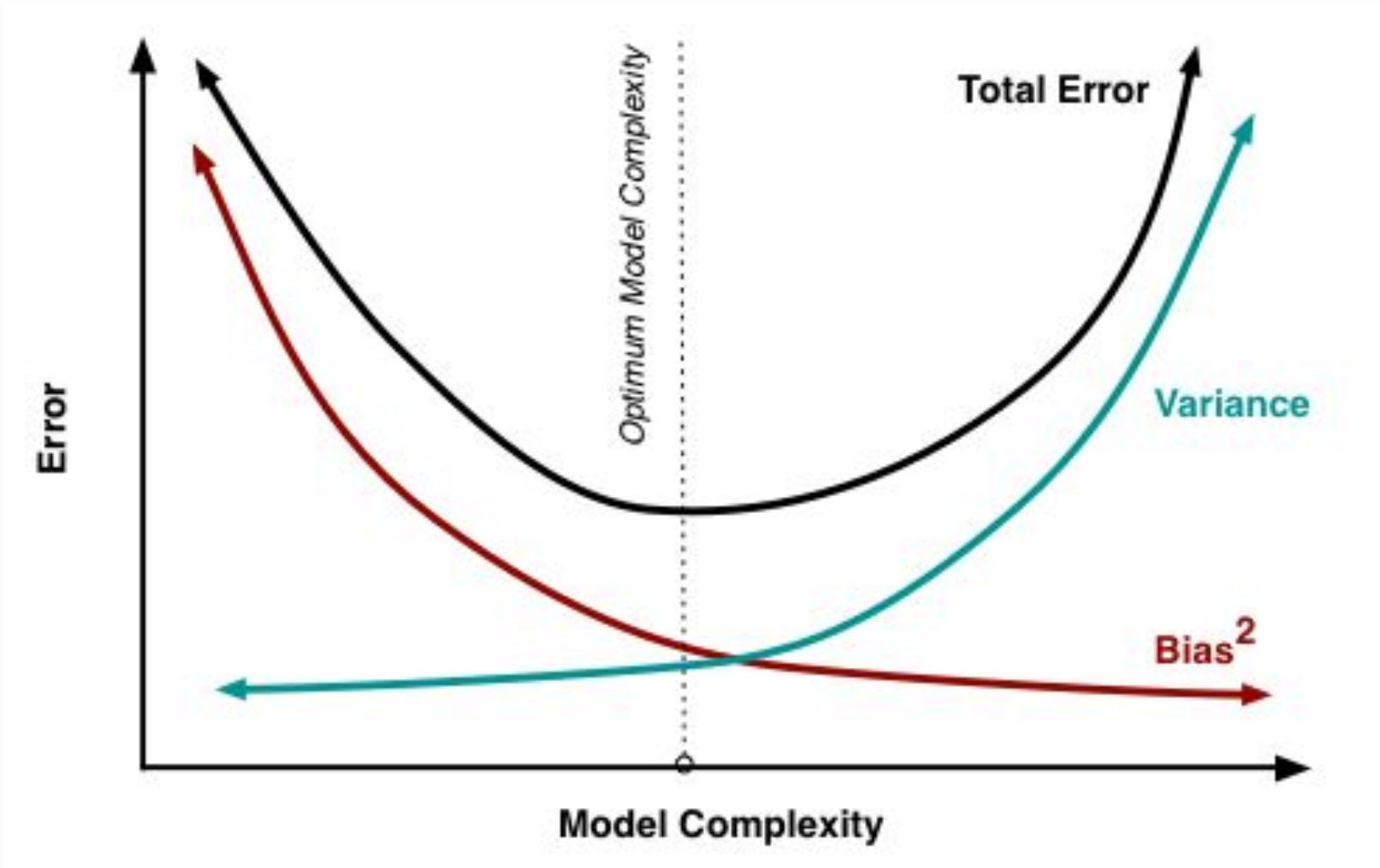
Chap. 5 Underfitting & Overfitting



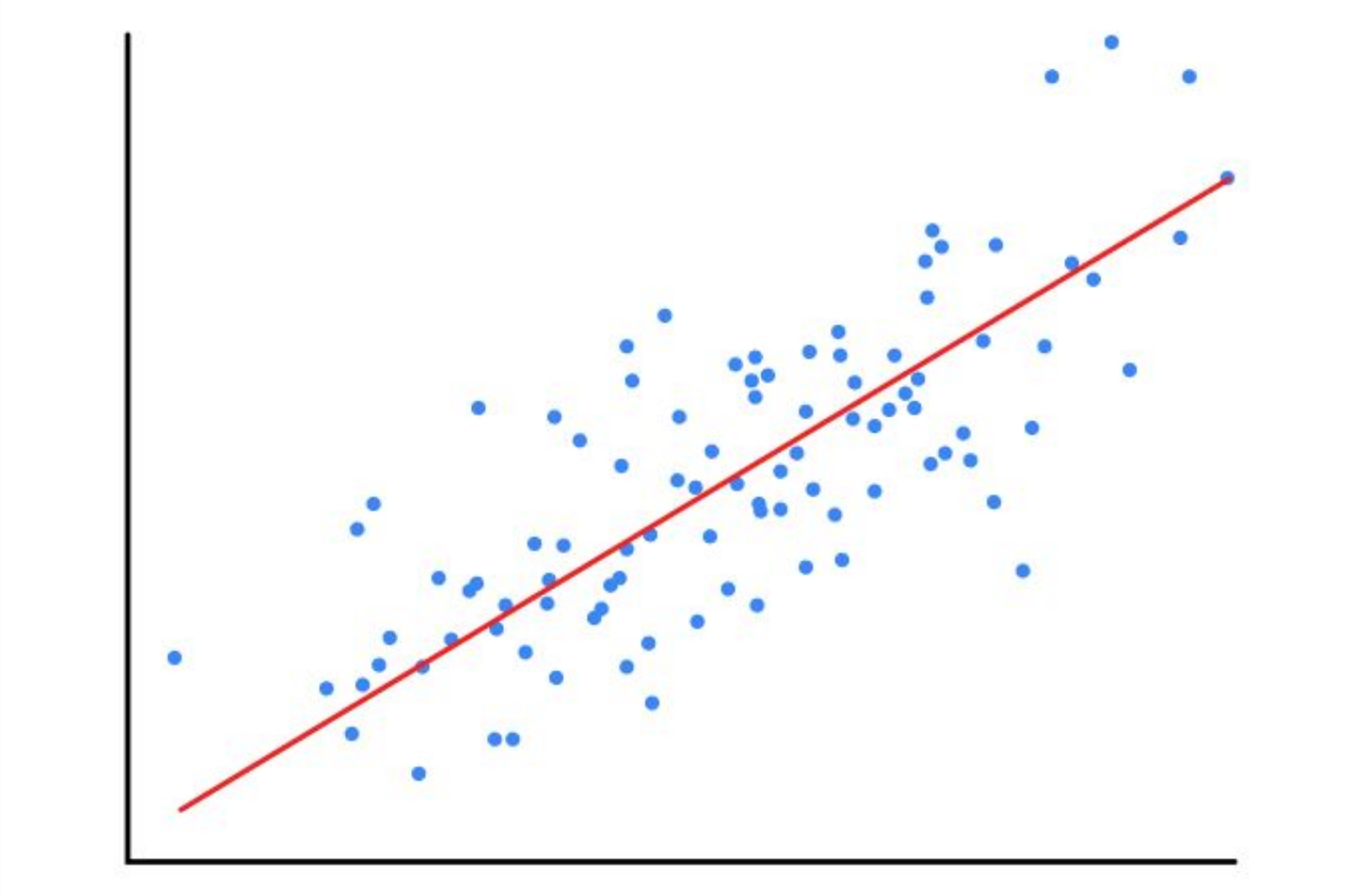




Chap. 5 Bias-Variance Trade Off

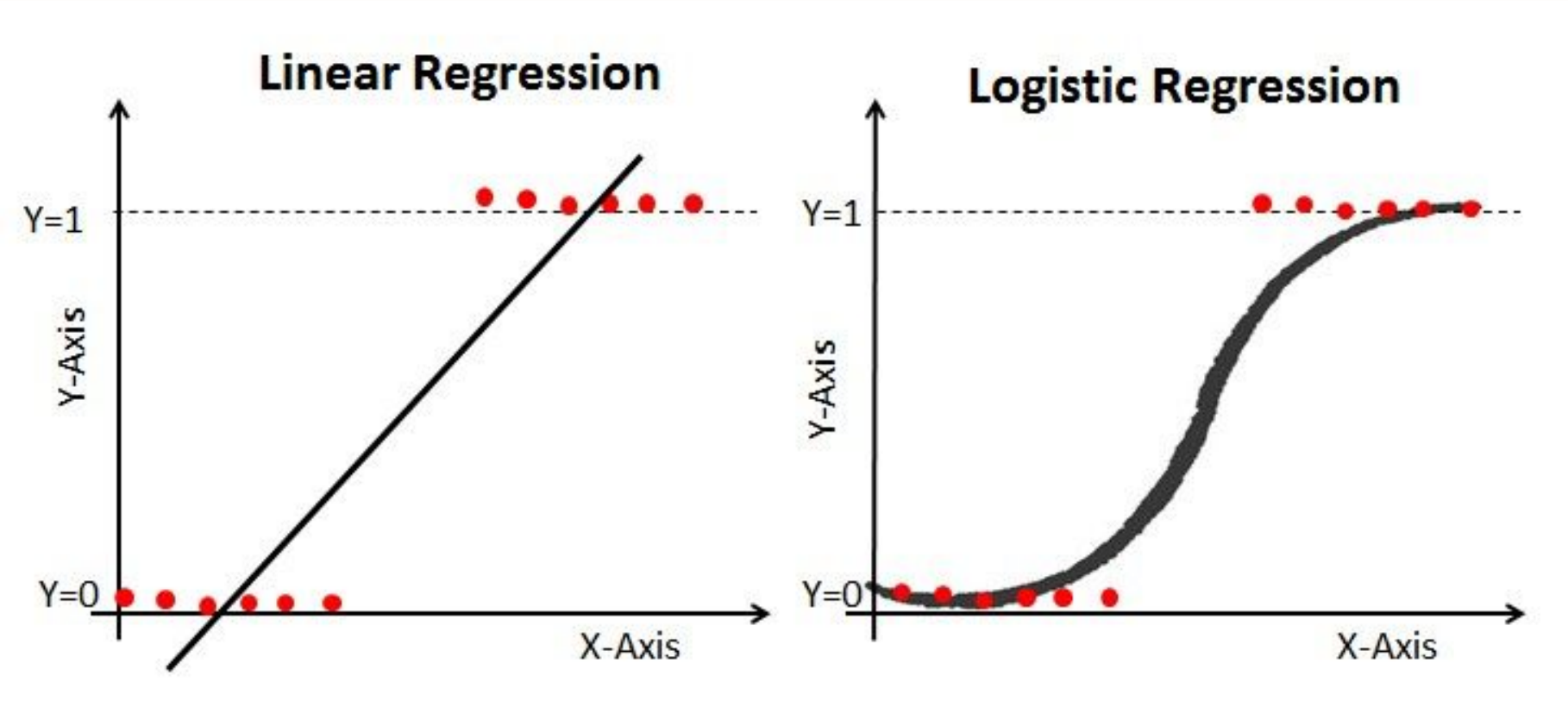


Chap. 5 Linear Regression

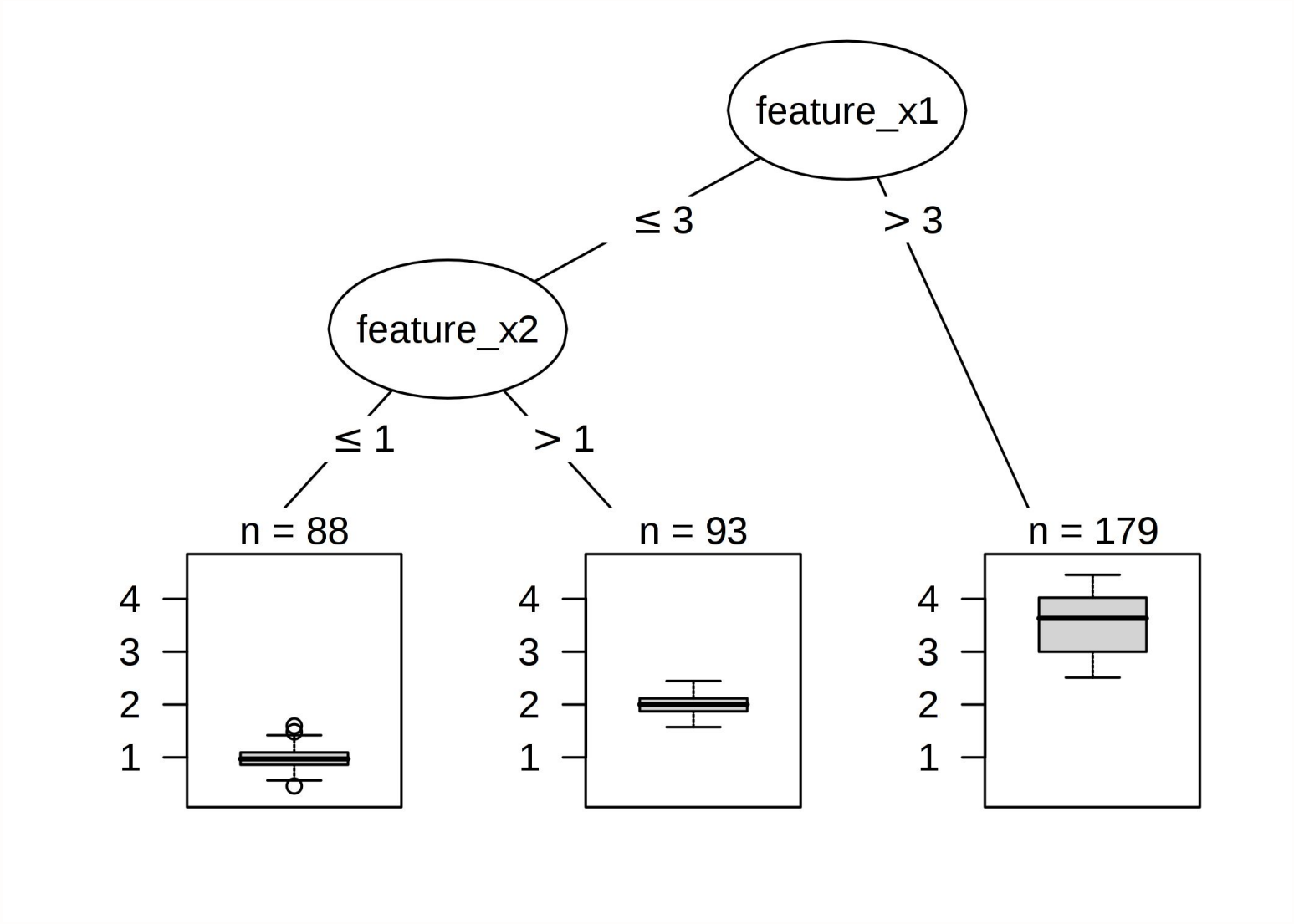


“ Cost Function = Min(RSS(W) + alph*W) ”

Chap. 5 Logistic Regression



Chap. 5 Regression Tree



Chapter 6

- Project -

Chap. 6 체지방량 예측

회귀

Feature
- 나이, 성별, 전압, 키, 몸무게

데이터
- 총: 438개 (1개 outlier)
- train: 421개
- test: 16개

Chap. 6 체지방량 예측

생체 데이터 전송

ML Model

회귀 값 전송

Chap. 6 대사 증후군 예측

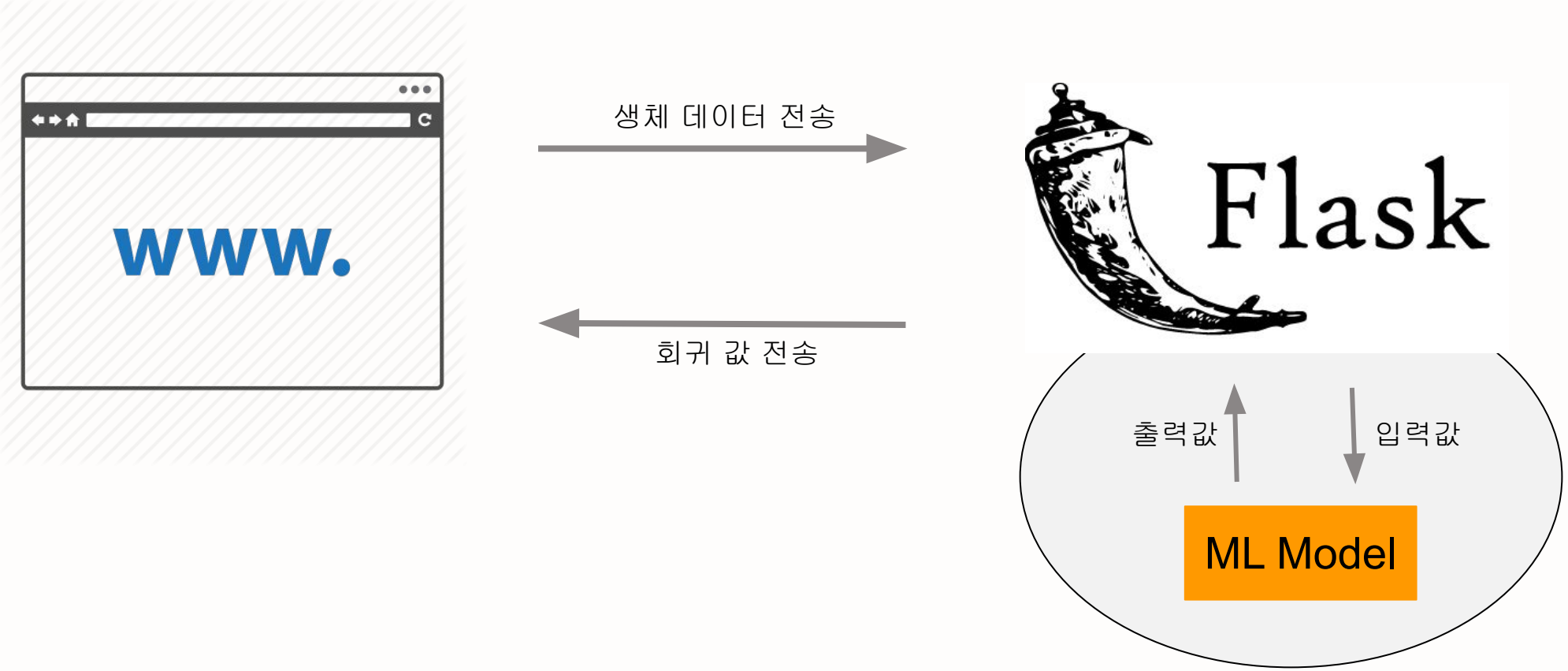
회귀

Feature
- 성별, 나이, SBP, DBP, HR, FBS, hbA1c, TC, TG, HDL, BMI, 표준 체중, 체지방량, 근육량, 체수분량, 기초 대사량, 허리둘레
- 허리둘레 위험군, BP 위험군, TG 위험군, HDL 위험군, Glucose 위험군

대사 증후군 분류
- 0: 정상
- 1: 저위험군
- 2: 고위험군
- 3: 환자

데이터
- 총: 99개 (아웃라이어 4개)
- Train: 85개
- Test: 10개

Chap. 6 체지방량 예측



Q & A

[감사합니다]