실무 데이터 분석
(상관관계 분석)

인공지능 기반 스마트 설계
컴퓨터 AI공학부
천세진

DATA SCIENCE LABS

데이터

	A	B	C	D	E	F	G	H
1	순번	유형	컨테이너번	크기	적재상태	화주	가이드 위	실제 위치
2	CONTR_KE	WRK_TYPE	CONTR_N	CONTR_SI	CONTR_LC	BP_NM	SP_BLK	REL_BLK
3	25629	IN	FTAU10	20	F	한도물류	A	A
4	25306	OT	HDMU68	40	E	로드스타	R	R
5	24852	OT	GCXU57	40	F	DTC	B	B
6	25605	IN	OOCU50	40	F	태성로지스	D	D
7	23610	OT	EITU94	40	F	태성로지스		D
8	27967	IN	CAAU52	40	F	태성로지스	D	D
9	27972	IN	NYKU36	20	E	태성로지스	A	A
10	27980	IN	FCIU55	20	F	로드스타	A	A
11	27979	IN	TEMU15	20	F	로드스타	A	A
12	27996	IN	FCIU58	20	F	에스에이치	A	A
13	24913	OT	MSKU54	20	F	태성로지스	A	A
14	27995	IN	TGBU97	40	F	한타특수운	E	E
15	27982	IN	NYKU47	40	F	한타특수운	E	E
16	27981	IN	TCKU78	40	F	한타특수운	E	E
17	27983	IN	HMMU64	40	F	한타특수운	E	E
18	27991	IN	MSCU51	40	F	한타특수운	E	E
19	27990	IN	KOCU47	40	F	한타특수운	E	E
20	27989	IN	TCNU23	40	F	한타특수운	E	E
21	27958	IN	TEMU56	20	F	삼일익스프	A	A
22	28001	IN	SEKU56	40	F	디앤디로직	E	E
23	28000	IN	TLLU77	40	F	디앤디로직	E	E
24	27971	IN	HDMU68	40	F	태성로지스	D	D
25	25461	OT	HDMU26	20	F	에스에이치	A	A
26	27937	OT	TEMU71	40	F	한타특수운	D	D

데이터 처리 단계

1. 데이터 이해
(목적, 구성, 특징)

2. 데이터 전처리
(결측값, 이상치, 중복값)

3. 데이터 탐색
(데이터의 분포, 상관관계, 이상치 탐색)

4. 통계적인 분석
(Aggregation/Summarization)

5. 시각화

6. 결론 도출

1. 데이터 이해

컨테이너 야드 장치장 (Container yard)

항구나 항만 근처에 위치하며, 컨테이너화된 화물을 일시적으로 저장하거나 운송 수단에 싣기 위해 사용

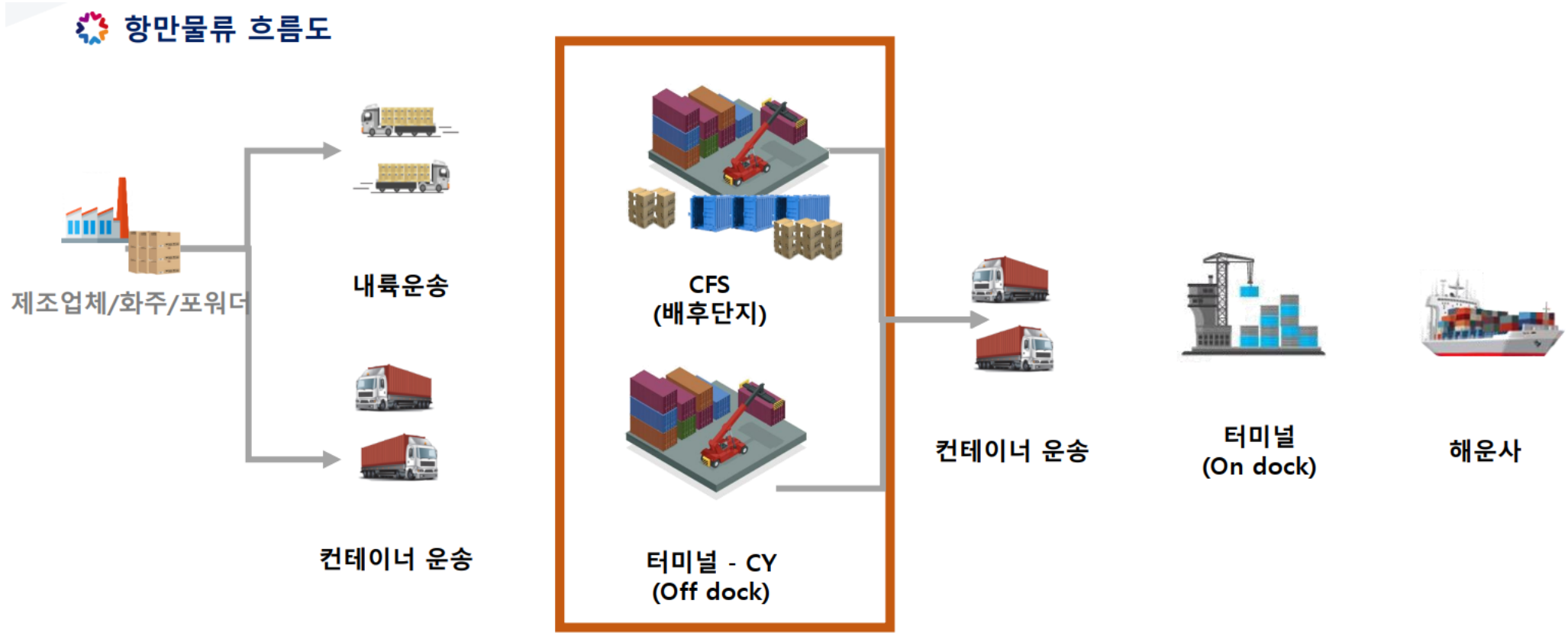
양·적하 작업
화물 보관 및 이동

배후부지 CFS (Container Freight Station)

항만에서 도착한 컨테이너 화물을 잠시 보관하거나 분류, 정리, 출고 등의 작업을 수행

화물 보관
분류 및 정리

1. 데이터 이해



1. 데이터 이해

문제점

- 대부분 업장에서 반입일 혹은 화주 단위로 적재
- 무분별한 적재는 이적량을 증가시키고 작업 능률을 낮춤
- 낮은작업 능률은 운송기사 대기시간 및 교통체증 또한 유발
- 일부 업장에선 수기로 사무실에서 컨테이너의 정보로 적절한 적재 위치를 기사에게 제공
  –〉 시행 이전에 비해 이적량이 감소한 효과

KULS 고객사 컨테이너 적재 모습

No.	컨테이너	유형	반출일	오더	추가정보
62 대기	CAXU 3292569 Full	20GP	06-12	6621 (A7)	ZIM ANTWERP 69E (로드스타)
61 대기	CMAU 8627455 Full	40GP	06-12	7383 (B40)	CMA CGM IGUACU (삼성전자로지텍)

컨테이너 적재위치 안내부분 (오더 내부 A7, B40)

1. 데이터 이해

- 순번: 순번 ID
- 유형: 유형 IN/OUT
- 컨테이너번호:
- 크기: 컨테이너 크기
- 적재상태: 적재 상태 (Full,Empty)
- 화주: 화주
- 가이드 위치: 가이드 위치
- 실제 위치: 실제 위치

	A	B	C	D	E	F	G	H
1	순번	유형	컨테이너번	크기	적재상태	화주	가이드 위	실제 위치
2	CONTR_KE	WRK_TYPE	CONTR_N	CONTR_SI	CONTR_LC	BP_NM	SP_BLK	REL_BLK
3	25629	IN	FTAU10	20	F	한도물류	A	A
4	25306	OT	HDMU68	40	E	로드스타	R	R
5	24852	OT	GCXU57	40	F	DTC	B	B
6	25605	IN	OOCU50	40	F	태성로지스	D	D
7	23610	OT	EITU94	40	F	태성로지스		D
8	27967	IN	CAAU52	40	F	태성로지스	D	D
9	27972	IN	NYKU36	20	E	태성로지스	A	A
10	27980	IN	FCIU55	20	F	로드스타	A	A
11	27979	IN	TEMU15	20	F	로드스타	A	A
12	27996	IN	FCIU58	20	F	에스에이치	A	A
13	24913	OT	MSKU54	20	F	태성로지스	A	A
14	27995	IN	TGBU97	40	F	한타특수운	E	E
15	27982	IN	NYKU47	40	F	한타특수운	E	E
16	27981	IN	TCKU78	40	F	한타특수운	E	E
17	27983	IN	HMMU64	40	F	한타특수운	E	E
18	27991	IN	MSCU51	40	F	한타특수운	E	E
19	27990	IN	KOCU47	40	F	한타특수운	E	E
20	27989	IN	TCNU23	40	F	한타특수운	E	E
21	27958	IN	TEMU56	20	F	삼일익스프	A	A
22	28001	IN	SEKU56	40	F	디앤디로직	E	E
23	28000	IN	TLLU77	40	F	디앤디로직	E	E
24	27971	IN	HDMU68	40	F	태성로지스	D	D
25	25461	OT	HDMU26	20	F	에스에이치	A	A
26	27937	OT	TEMU71	40	F	한타특수운	D	D

1. 데이터 이해

- **데이터의 수**

```
1 # 데이터의 수 (레코드수, 특징수)
2 df.shape
```

```
(198, 8)
```

1. 데이터 이해

- 데이터의 유형

```
df.dtypes # 데이터 유형
```

```
CONTR_KEY          int64
WRK_TYPE          object
CONTR_NO          object
CONTR_SIZE         int64
CONTR_LOAD_STS    object
BP_NM             object
SP_BLK            object
REL_BLK           object
dtype: object
```

DATA SCIENCE LABS

1. 데이터 이해

- 특징
  - 기본적인 데이터 통계, 결측값 확인

```
1 # 기본적인 통계
2 df.describe()
```

	CONTR_KEY	CONTR_SIZE
count	198.000000	198.000000
mean	27010.398990	33.737374
std	1512.653452	9.298857
min	22760.000000	20.000000
25%	25463.250000	20.000000
50%	27991.500000	40.000000
75%	28057.500000	40.000000
max	28155.000000	40.000000

1. 데이터 이해

- **문자열 -> 카테고리 데이터 변환**

```
1 # 문자열 카테고리값으로 변환
2 categorical_columns = df.select_dtypes(include=['object']).columns
3 df[categorical_columns] = df[categorical_columns].astype('category')
4
5 df.dtypes
```

```
CONTR_KEY           int64
WRK_TYPE         category
CONTR_NO         category
CONTR_SIZE          int64
CONTR_LOAD_STS   category
BP_NM            category
SP_BLK           category
REL_BLK          category
dtype: object
```

1. 데이터 이해

- 카테고리 값 (고유값들 확인)

```
# 모든 카테고리값을 출력
category_values = {col: df[col].cat.categories for col in categorical_columns}
# category_values

#
df['BP_NM'].cat.categories
```

```
Index(['DTC', 'HK종합운수', 'KCTC부산', 'SITC', 'SKON', '더블피씨', '덕창로지스틱스', '동우로지스틱',
       '디앤디로직스', '로드스타', '삼일익스프레스', '서중로직스', '에스에이치피물류', '영풍물류', '우진종합물류',
       '유니온로지스', '지투비', '태성로지스', '트레이스로지스틱스', '한도물류', '한타특수운송'],
      dtype='object')
```

2. 데이터 전처리

- 카테고리 값에 대한 표현통계

```
1 # 카테고리 데이터에 대한 표현 통계
2 df[categorical_columns].describe()
```

	WRK_TYPE	CONTR_NO	CONTR_LOAD_STS	BP_NM	SP_BLK	REL_BLK
count	198	198	198	198	193	198
unique	2	186	2	21	6	6
top	IN	TXGU70	F	태성로지스	A	A
freq	136	2	184	64	60	62

2. 데이터 전처리

- 카테고리 값을 숫자로 변환

```
# 카테고리 값을 숫자로 변환
df_copy = df.copy()
for col in categorical_columns:
    df_copy[col] = df_copy[col].cat.codes
```

3. 데이터 탐색

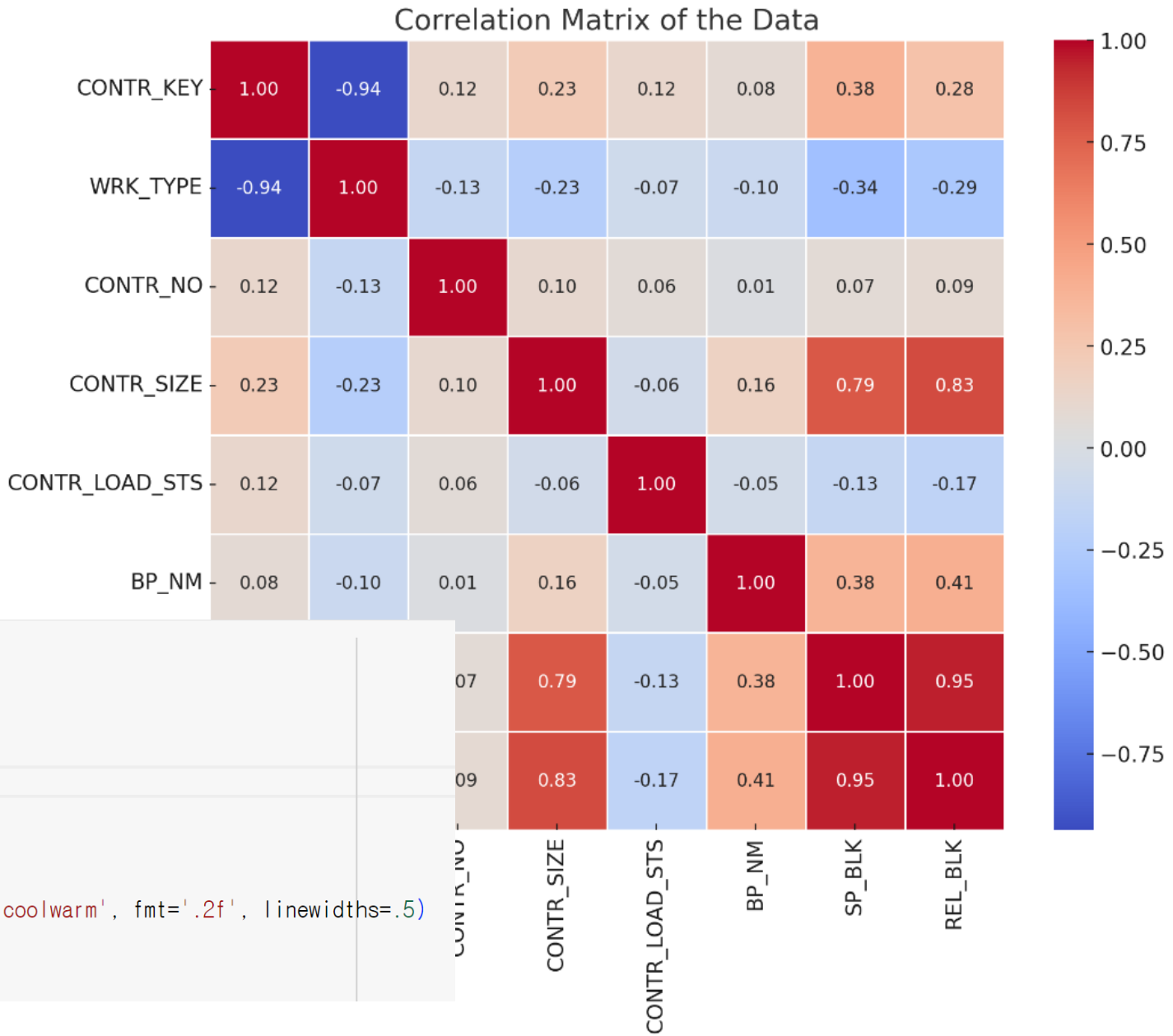
- 카테고리 값을 숫자로 변환

```
1 df_copy.corr()
```

	CONTR_KEY	WRK_TYPE	CONTR_NO	CONTR_SIZE	CONTR_LOAD_STS	BP_NM	SP_BLK	REL_BLK
CONTR_KEY	1.000000	-0.937569	0.118860	0.227375	0.119922	0.080604	0.384894	0.281532
WRK_TYPE	-0.937569	1.000000	-0.134634	-0.225095	-0.068661	-0.103247	-0.342967	-0.286132
CONTR_NO	0.118860	-0.134634	1.000000	0.100280	0.064466	0.011712	0.070398	0.087505
CONTR_SIZE	0.227375	-0.225095	0.100280	1.000000	-0.058791	0.163479	0.785694	0.826965
CONTR_LOAD_STS	0.119922	-0.068661	0.064466	-0.058791	1.000000	-0.045351	-0.128209	-0.170462
BP_NM	0.080604	-0.103247	0.011712	0.163479	-0.045351	1.000000	0.377021	0.409501
SP_BLK	0.384894	-0.342967	0.070398	0.785694	-0.128209	0.377021	1.000000	0.950548
REL_BLK	0.281532	-0.286132	0.087505	0.826965	-0.170462	0.409501	0.950548	1.000000

3. 데이터 탐색

- 히트맵을 사용

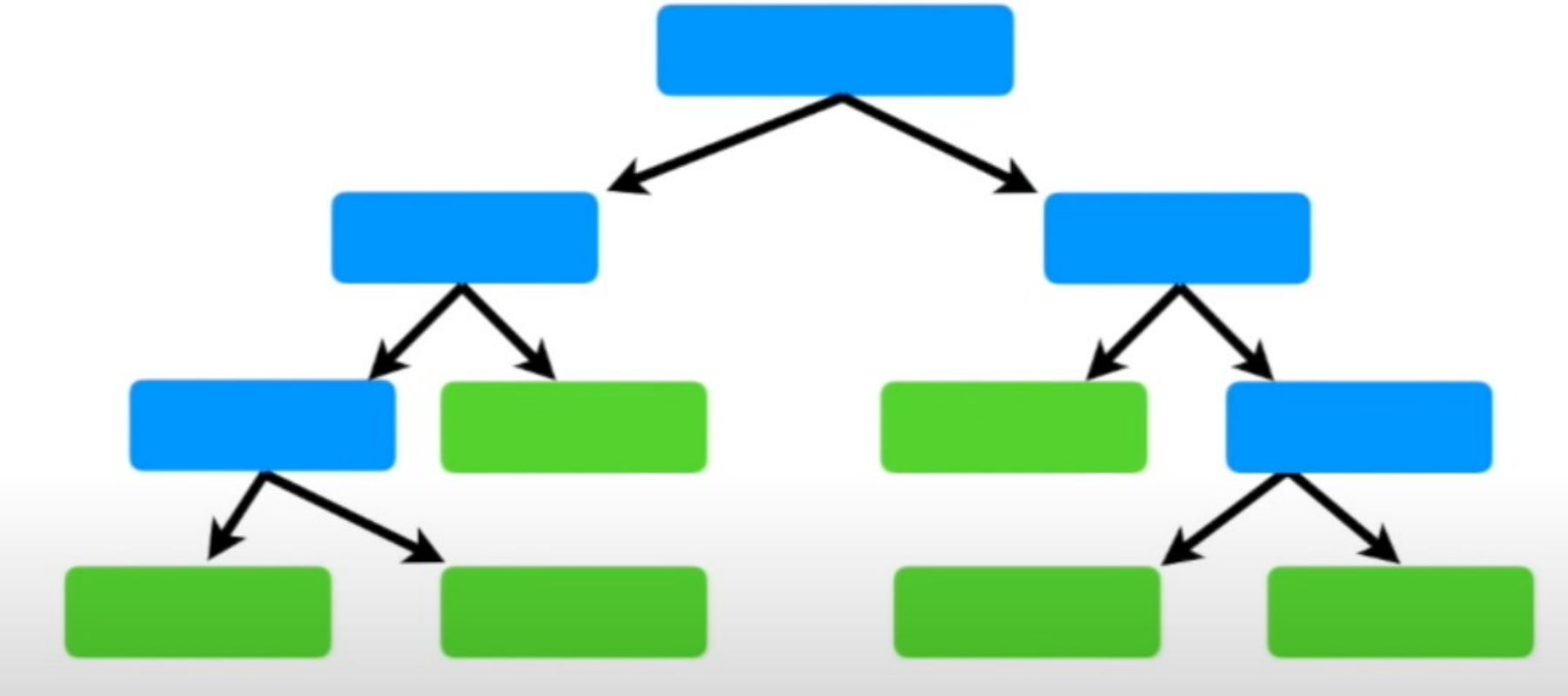



```python
import seaborn as sns
import matplotlib.pyplot as plt

# Calculate the correlation matrix
correlation_matrix = df_copy.corr()

# Plot the heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f', linewidths=.5)
plt.title('Correlation Matrix of the Data')
plt.show()
```

DATA SCIENCE LABS

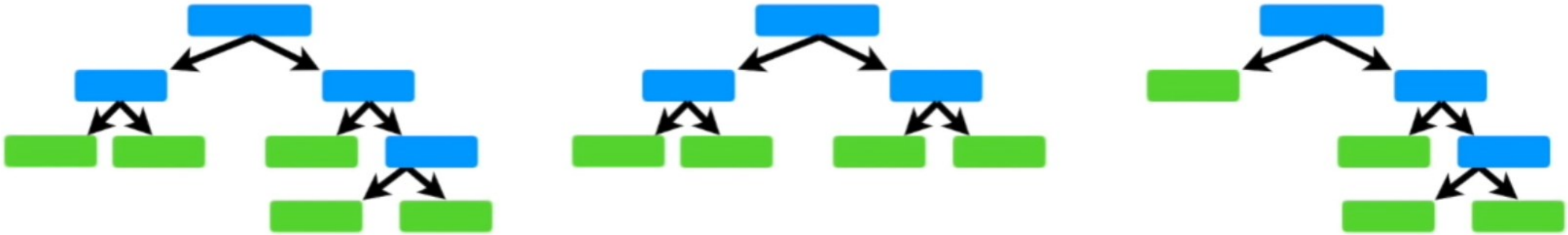
4. 랜덤포레스트(분류기)

- 결정 트리(Decision tree)



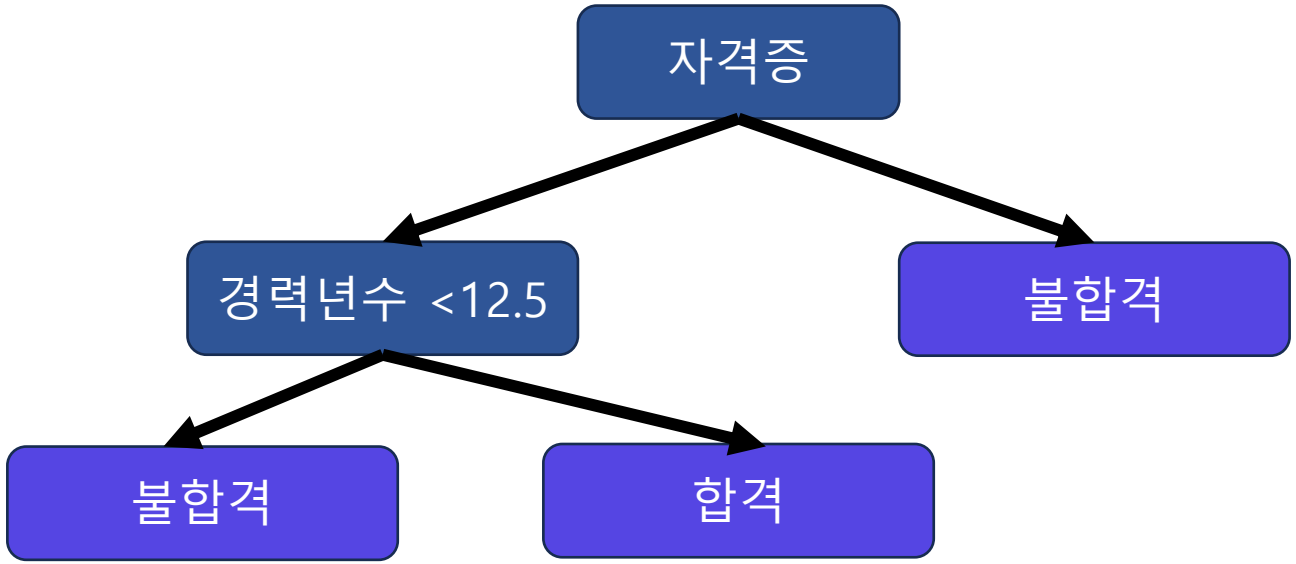
4. 랜덤포레스트(분류기)

- 다양한 결정 트리를 이용



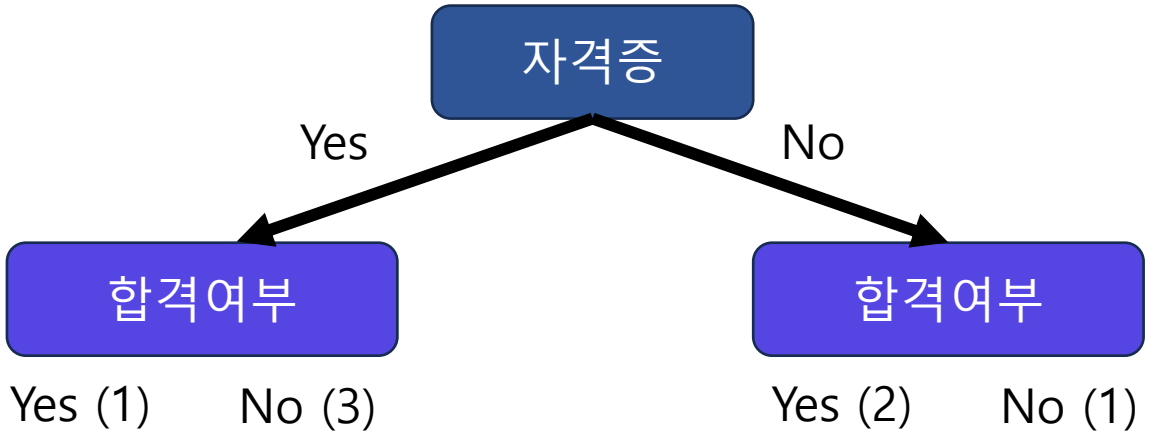
4. 결정트리

자격증	결혼여부	경력년수	합격여부
Yes	Yes	7	**No**
Yes	No	12	**No**
No	Yes	18	**Yes**
No	Yes	35	**Yes**
Yes	Yes	38	**Yes**
Yes	No	50	**No**
No	No	83	**No**



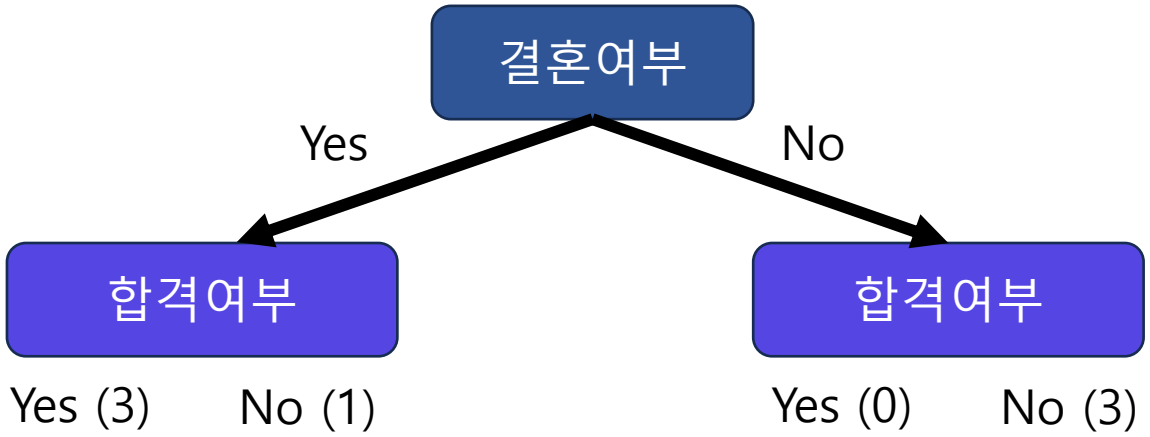
4. 결정트리

자격증	결혼여부	경력년수	합격여부
Yes	Yes	7	**No**
Yes	No	12	**No**
No	Yes	18	**Yes**
No	Yes	35	**Yes**
Yes	Yes	38	**Yes**
Yes	No	50	**No**
No	No	83	**No**



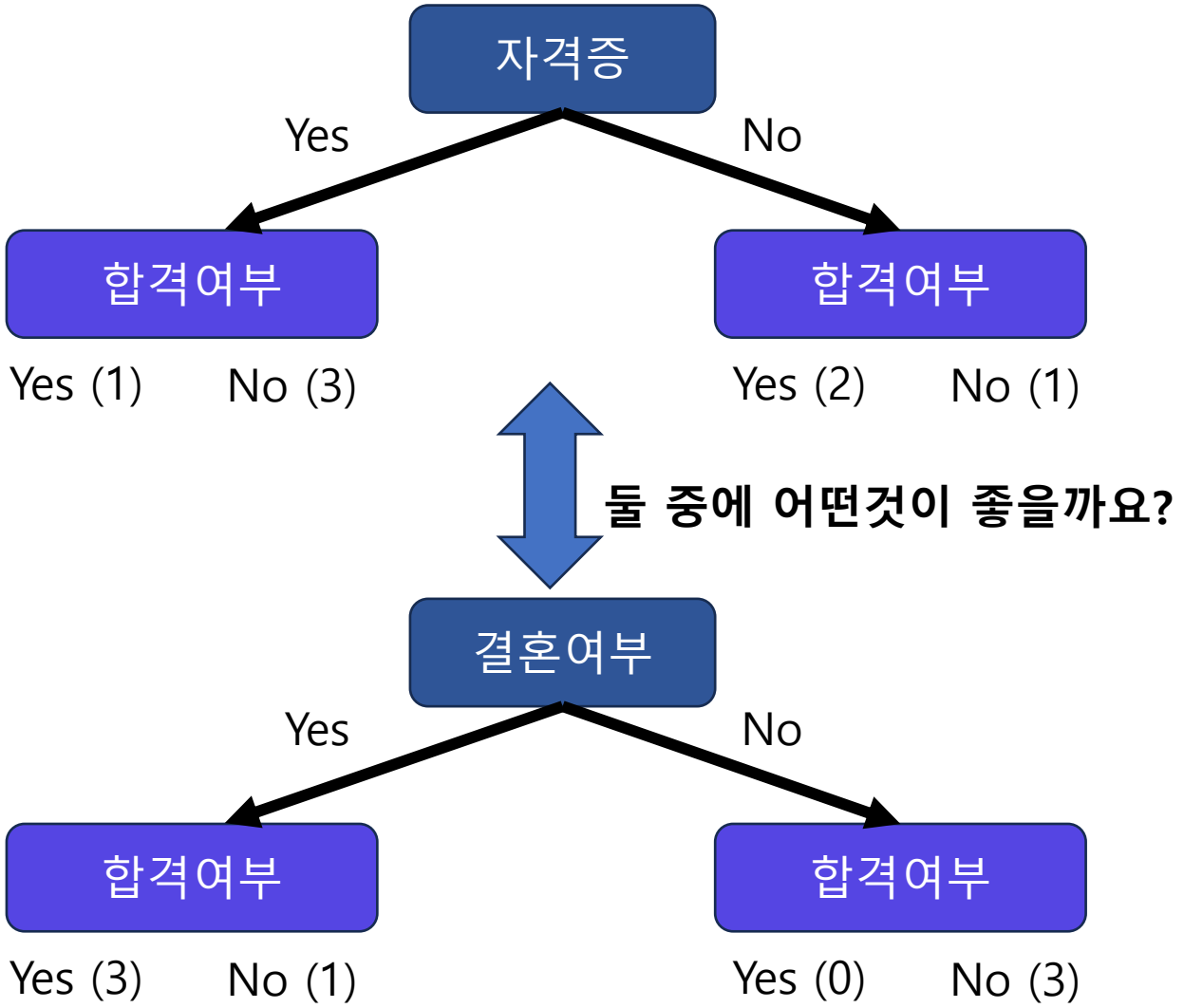
4. 결정트리

자격증	결혼여부	경력년수	합격여부
Yes	Yes	7	**No**
Yes	No	12	**No**
No	Yes	18	**Yes**
No	Yes	35	**Yes**
Yes	Yes	38	**Yes**
Yes	No	50	**No**
No	No	83	**No**



4. 결정트리

자격증	결혼여부	경력년수	합격여부
Yes	Yes	7	**No**
Yes	No	12	**No**
No	Yes	18	**Yes**
No	Yes	35	**Yes**
Yes	Yes	38	**Yes**
Yes	No	50	**No**
No	No	83	**No**

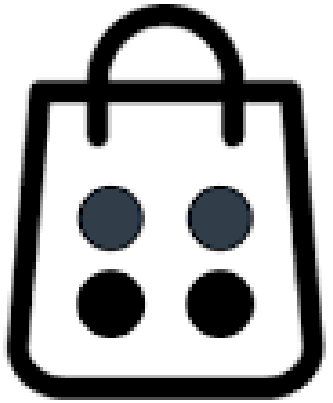


4. 결정트리

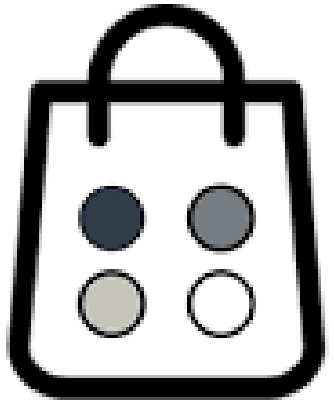
지니 불순도: 분류된 케이스에 데이터가 불순한 정도.

$$Gini(D) = 1 - \sum_{i=1}^{k} p_i^2$$

그외
Information Gain, 엔트로피



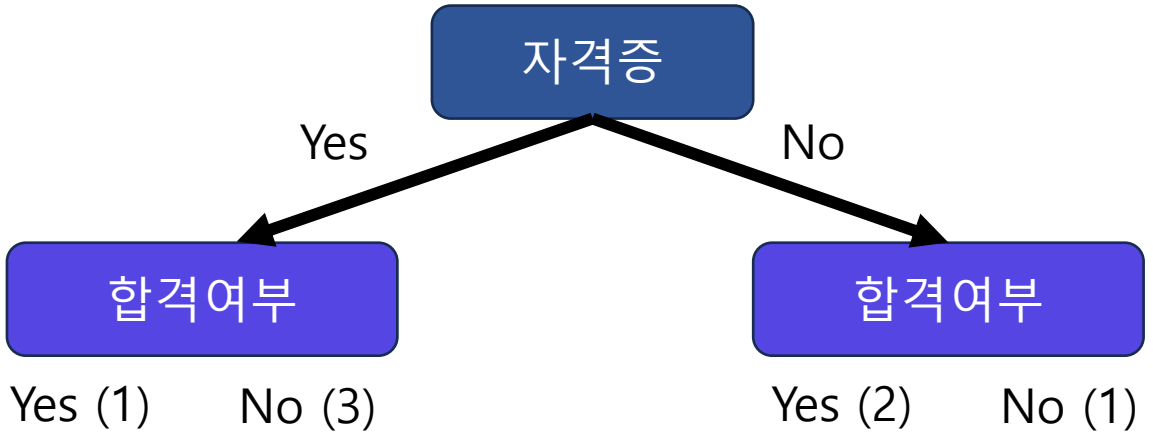
Low Gini
impurity index



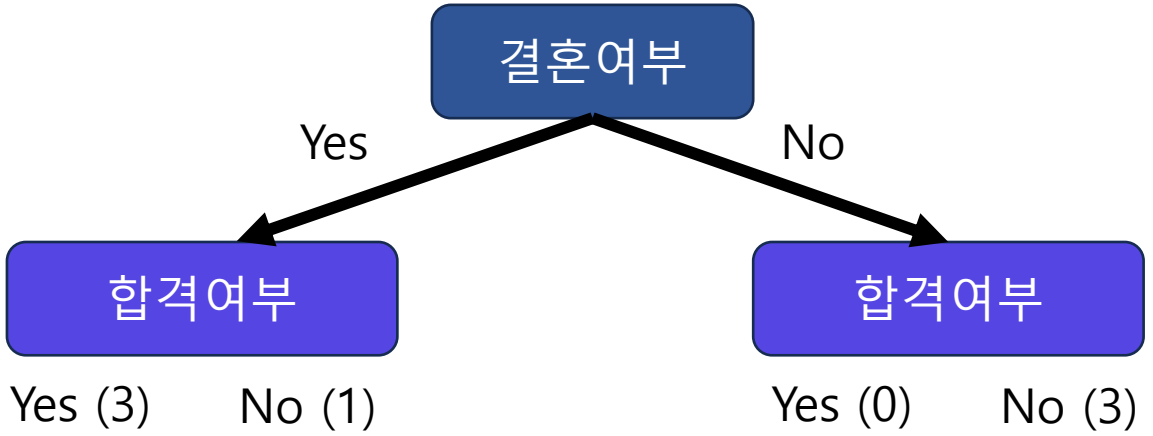
High Gini
impurity index

4. 결정트리

자격증	결혼여부	경력년수	합격여부
Yes	Yes	7	**No**
Yes	No	12	**No**
No	Yes	18	**Yes**
No	Yes	35	**Yes**
Yes	Yes	38	**Yes**
Yes	No	50	**No**
No	No	83	**No**

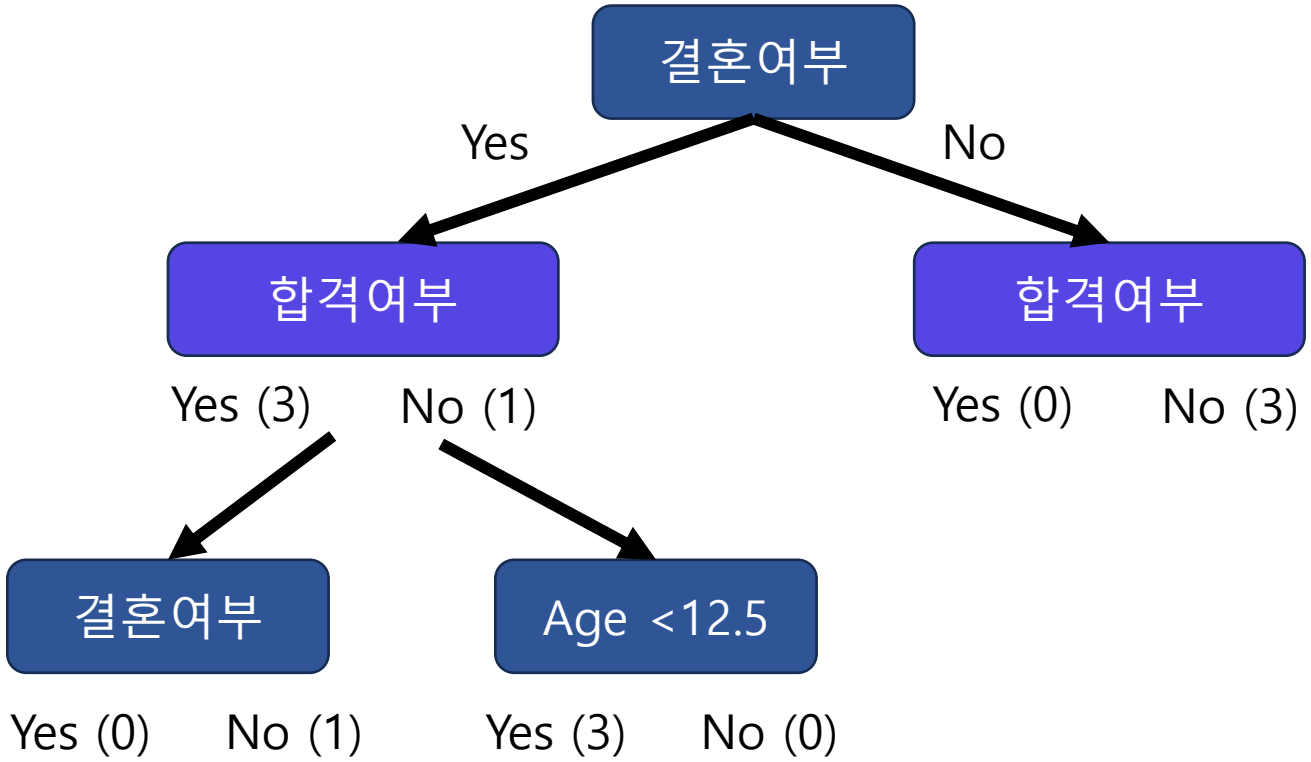


지니 불순도

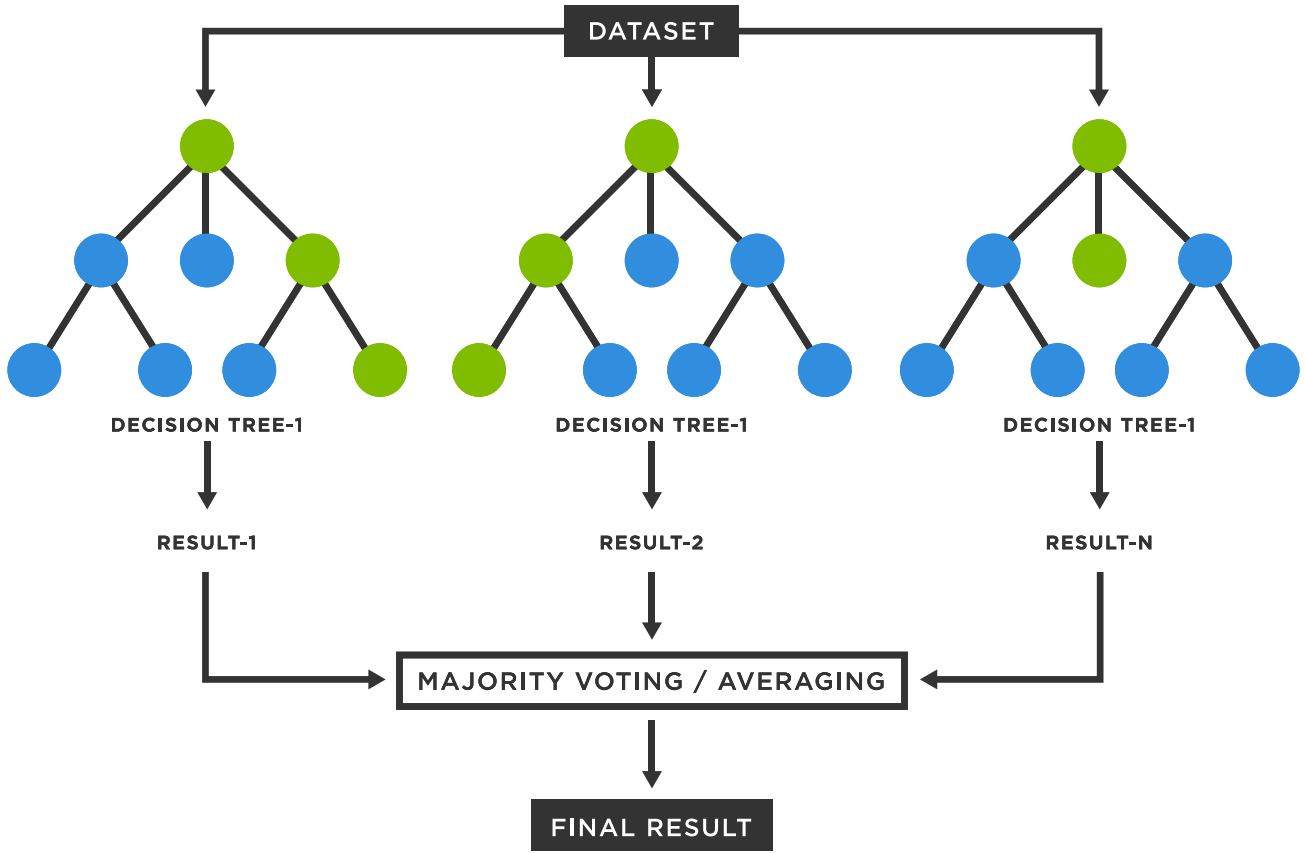


4. 결정트리

자격증	결혼여부	경력년수	합격여부
Yes	Yes	7	**No**
Yes	No	12	**No**
No	Yes	18	**Yes**
No	Yes	35	**Yes**
Yes	Yes	38	**Yes**
Yes	No	50	**No**
No	No	83	**No**



4. 랜덤포레스트



4. 랜덤포레스트(분류기)

```
[26]  1 from sklearn.model_selection import train_test_split
      2 from sklearn.ensemble import RandomForestClassifier
      3 from sklearn.metrics import accuracy_score, classification_report
```

```
[27]  1 # Prepare the data
      2 X = df_copy.drop('REL_BLK', axis=1)  # Features
      3 y = df_copy['REL_BLK']  # Target
      4
      5 # Split the data into training and test sets
      6 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

4. 랜덤포레스트(분류기)

```
# Initialize the RandomForestClassifier
rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)

# Train the model
rf_classifier.fit(X_train, y_train)

# Predict on the test set
y_pred = rf_classifier.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
class_report = classification_report(y_test, y_pred)

accuracy, class_report
```

4. 결과 확인

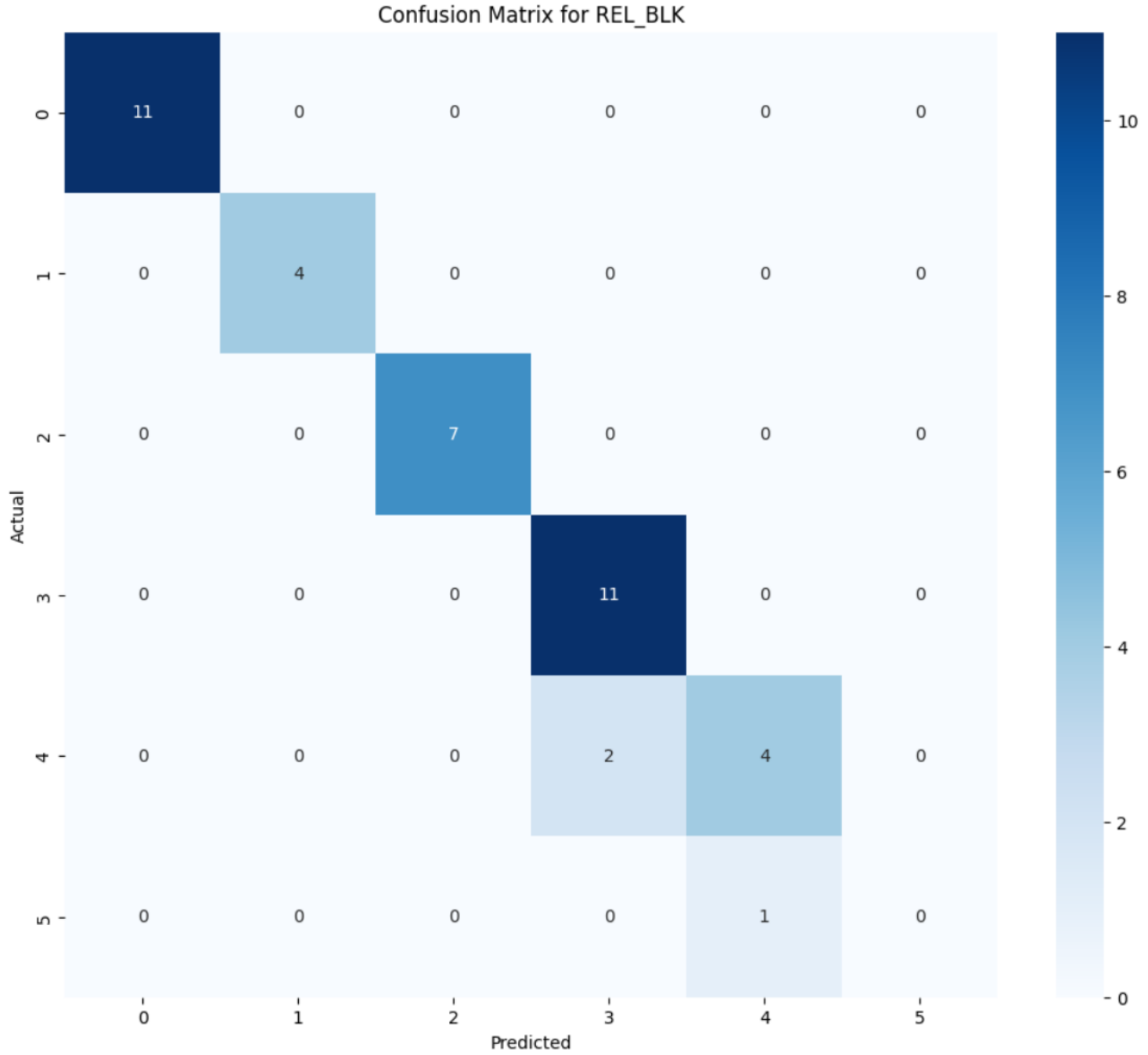
- Confusion Matrix
  - 실제와 예측의 결과를 비교

```
1 from sklearn.metrics import confusion_matrix
2 import numpy as np
```

```
1 cm_rel_blk = confusion_matrix(y_test, y_pred)
2
3 # Plot the confusion matrix as a heatmap for REL_BLK
4 plt.figure(figsize=(12, 10))
5 sns.heatmap(cm_rel_blk, annot=True, fmt='d', cmap='Blues', xticklabels=np.unique(y), yticklabels=np.unique(y))
6 plt.title('Confusion Matrix for REL_BLK')
7 plt.ylabel('Actual')
8 plt.xlabel('Predicted')
9 plt.show()
```

4. 결과 확인

- Confusion Matrix
  - 실제와 예측의 결과를 비교
  - X축은 예측된결과
  - Y축은 실제



Further work

- 서포트 벡터 머신(SVM)
- KNN 최근접이웃
- 그라디언트 부스팅(Gradient Boosting)