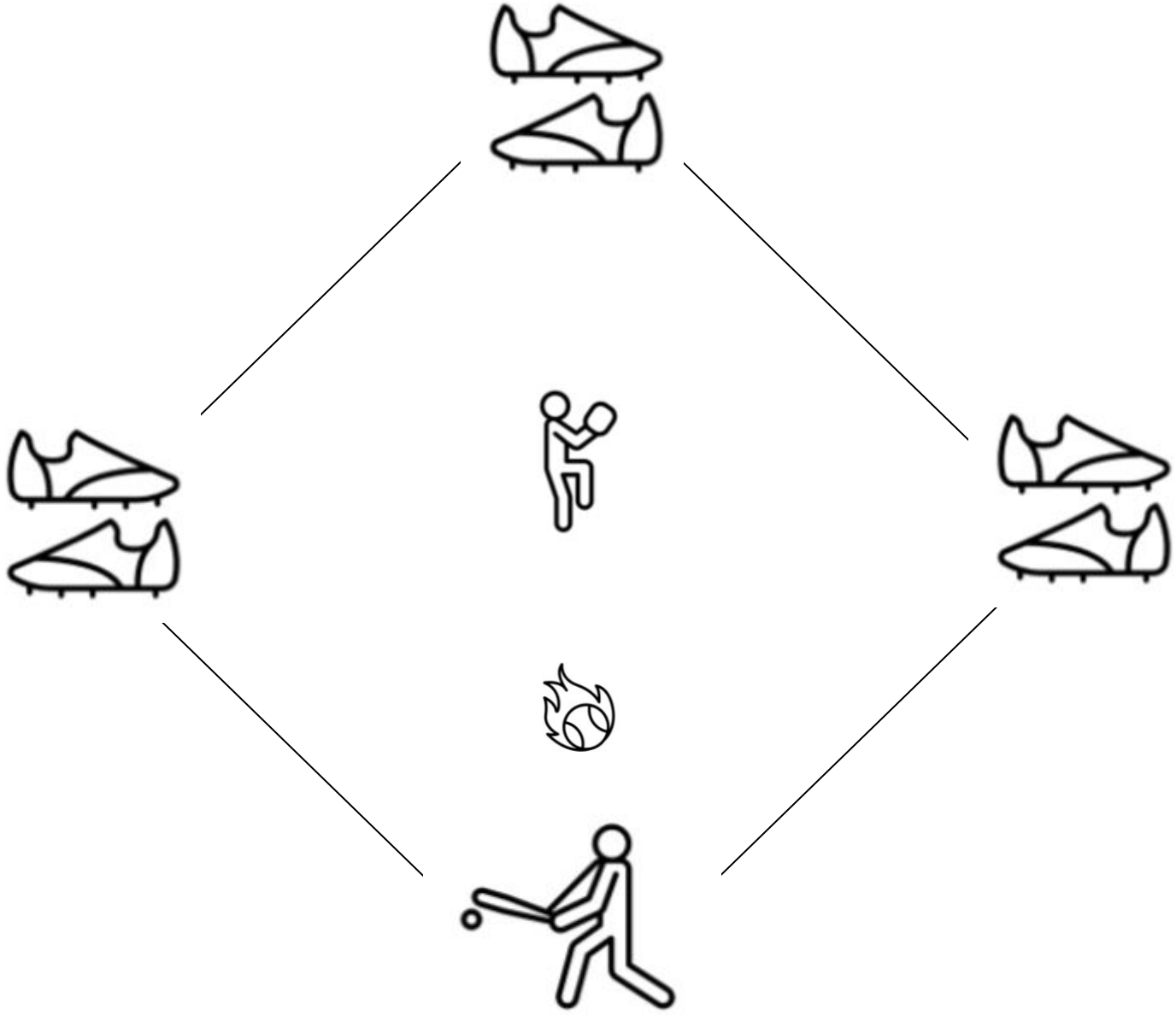
[ML]kt_wiz 황재균 타점 예측하기

홍익대학교 빅데이터 교육과정 김원중

타점이란?



INDEX

01 전처리

02 회귀

03 다중분류

문제정의

kt_wiz 황재균 선수의 타점을 구해보자

- 2007년부터 2021년도 까지의 데이터 셋을 만든다.
- Train data는 07 ~ 20년도 까지의 데이터를 사용한다.
- Test data는 21년도 데이터를 사용한다.
- 독립변수 X는 상대팀, 타석, 타율, 안타, 2루타, 3루타, 홈런 feature를 사용한다.
- 타겟 y는 타점을 사용한다.

전처리

과정

- Import library
- 결측치 제거
- 열 삭제
- One_Hot_Encoding
- Train, test 데이터 슬라이싱을 위한 컬럼 위치 변경

개발환경

```
!python --version
```

```
Python 3.7.10
```

```
import pandas as pd
import numpy as np
import seaborn as sns
import plotly.express as px
# 랜덤 시드 고정시키기
np.random.seed(5)

import tensorflow as tf
from keras.models import Sequential
from keras.layers import Dense
from keras.activations import relu
from keras.optimizers import Adam
```

Develop Environment

파이썬 버전은 3.7.10

import pandas, matplotlib, numpy, seaborn

keras, plotly

한글깨짐방지

데이터 셋

<KBO 기록실>

<복사 & 붙여넣기> .tsv파일로 생성

KT 위즈

선수명: 황재균
생년월일: 1987년 07월 28일
신장/체중: 183cm/96kg
입단 계약금: 6000만원
지명순위: 06 현대 2차 3라운드 24순위
등번호: No.10
포지션: 내야수(우투우타)
경력: 사당초-서울이수중-경기고-현대-우리-히어로즈-넥센-롯데
연봉: 80000만원
입단년도: 06현대

타자 | 투수 | 기록용어

기본기록 | 통산기록 | 일자별기록 | 경기별기록 | 상황별기록 | 등록일수

2007년 일자별 성적

2007 | KBO 정규시즌

4월	상대	AVG1	PA	AB	R	H	2B	3B	HR	RBI	SB	CS	BB	HBP	SO	GDP	AVG2
04.19	두산	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.000
04.21	롯데	0.000	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0.000
04.22	롯데	0.000	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0.000
04.26	두산	1.000	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0.333
합계		0.333	3	3	0	1	0	0	0	0	0	0	0	0	1	0	0.333

5월	상대	AVG1	PA	AB	R	H	2B	3B	HR	RBI	SB	CS	BB	HBP	SO	GDP	AVG2
05.01	LG	0.000	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0.250
05.02	LG	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.250

Hwang_07_21_data.txt

4월	상대	타율	타격	타석	득점	안타	2루타	3루타	홈런	타점	도루	도루실패	볼넷	사구	삼진	병살타	AVG2
04.19	두산	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0.000	
04.21	롯데	0.000	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0.000
04.22	롯데	0.000	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0.000
04.26	두산	1.000	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0.333
05.01	LG	0.000	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0.250
05.02	LG	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0.250	
05.04	SK	-	0	0	1	0	0	0	0	0	0	0	0	0	0	0.250	
05.05	SK	0.500	2	2	0	1	0	0	0	1	0	0	0	0	0	0	0.333
05.06	SK	-	0	0	1	0	0	0	0	0	0	0	0	0	0	0.333	
05.10	한화	0.000	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0.286
06.05	한화	0.000	2	2	0	0	0	0	0	0	0	0	0	0	1	0	0.222
06.07	한화	0.000	2	2	0	0	0	0	0	0	0	0	0	1	0	0	0.182
06.10	롯데	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0.182	
06.13	LG	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0.182	
06.14	LG	0.000	2	2	0	0	0	0	0	0	0	0	0	2	0	0	0.154
06.16	삼성	0.333	3	3	0	1	0	0	0	0	0	0	0	1	0	0	0.188
06.17	삼성	0.000	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0.176
07.22	KIA	1.000	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0.222
07.25	SK	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0.222	
07.26	SK	0.000	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0.211
07.27	LG	0.000	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0.200
07.28	LG	0.000	3	2	0	0	0	0	0	0	0	0	0	1	0	0	0.182
07.29	LG	0.000	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0.167
07.31	롯데	0.333	3	3	0	1	0	0	0	0	0	0	0	0	0	0	0.185
08.01	롯데	-	0	0	1	0	0	0	0	0	0	0	0	0	0	0.185	
08.02	롯데	1.000	2	1	1	1	1	0	0	1	0	0	1	0	0	0	0.214
08.03	한화	0.500	4	4	0	2	0	0	0	0	0	0	0	0	0	0	0.250
08.05	한화	1.000	2	2	0	2	0	0	0	0	0	0	0	0	0	0	0.294
08.10	삼성	0.500	4	4	0	2	0	0	0	1	0	0	0	1	0	0	0.316
08.11	삼성	0.333	3	3	0	1	0	0	0	0	1	0	0	0	2	0	0.317
08.15	한화	0.200	5	5	0	1	0	0	0	0	0	0	0	0	0	0	0.304
08.16	한화	0.250	4	4	1	1	0	0	1	3	0	0	0	0	0	0	0.300
08.17	롯데	0.500	4	4	1	2	1	0	0	1	0	0	0	0	0	0	0.315
08.18	롯데	0.000	2	2	0	0	0	0	0	0	0	0	0	0	1	0	0.304
08.19	롯데	0.500	4	4	1	2	0	0	0	0	0	0	0	0	1	0	0.317
08.21	LG	0.250	4	4	1	1	0	0	1	1	0	0	0	0	0	0	0.313
08.22	LG	0.000	2	2	0	0	0	0	0	0	0	0	0	1	0	0	0.303
08.23	LG	0.000	3	3	0	0	0	0	0	0	0	0	0	1	0	0	0.290
08.24	두산	0.333	4	3	1	1	0	0	0	0	0	0	1	0	1	0	0.292
08.25	두산	0.250	4	4	0	1	0	0	0	0	1	0	0	0	0	0	0.289
08.26	두산	0.200	5	5	0	1	0	0	0	1	0	0	0	0	0	0	0.284
08.28	SK	0.667	4	3	1	2	0	0	0	1	0	1	0	0	0	0	0.298
08.30	SK	0.000	3	3	0	0	0	0	0	0	0	0	0	0	2	0	0.287

데이터 셋

<데이터 불러오기>

```
df = pd.read_csv('/content/Hwang_07_21_data.txt', sep='₩t')
df.tail()
```

	4월	상대	타율	타격	타석	득점	안타	2루타	3루타	홈런	타점	도루	도루실패	볼넷	사구	삼진	병살타	AVG2
1623	6.27	한화	0.500	4	4	2	2	0	0	1	4	0	0	0	0	2	0	0.309
1624	6.30	LG	0.500	4	4	1	2	0	0	0	1	0	1	0	0	0	0	0.314
1625	7.01	LG	0.000	5	5	1	0	0	0	0	0	0	0	0	0	2	0	0.304
1626	7.02	키움	0.500	4	4	2	2	0	0	0	0	0	0	0	0	0	0	0.309
1627	7.04	키움	0.400	5	5	2	2	0	0	0	3	0	0	0	0	0	0	0.312

<데이터 확인>

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1628 entries, 0 to 1627
Data columns (total 18 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   4월      1628 non-null   float64
 1   상대      1628 non-null   object
 2   타율      1628 non-null   object
 3   타격      1628 non-null   int64
 4   타석      1628 non-null   int64
 5   득점      1628 non-null   int64
 6   안타      1628 non-null   int64
 7   2루타     1628 non-null   int64
 8   3루타     1628 non-null   int64
 9   홈런      1628 non-null   int64
 10  타점      1628 non-null   int64
 11  도루      1628 non-null   int64
 12  도루실패    1628 non-null   int64
 13  볼넷      1628 non-null   int64
 14  사구      1628 non-null   int64
 15  삼진      1628 non-null   int64
 16  병살타     1628 non-null   int64
 17  AVG2    1628 non-null   float64
dtypes: float64(2), int64(14), object(2)
memory usage: 229.1+ KB
```

타율도 타점을 구하기 위한 독립변수이므로 전처리 필요

결측치 제거

<'-' 글자 제거>

4월	상대	타율	타격	타석
04.19	두산	-	0	
04.21	롯데	0.000		
04.22	롯데	0.000		
04.26	두산	1.000		
05.01	LG	0.000		
05.02	LG	-	0	
05.04	SK	-	0	
05.05	SK	0.500		
05.06	SK	-	0	
05.10	한화	0.000		
06.05	한화	0.000		
06.07	한화	0.000		
06.10	롯데	-	0	
06.13	LG	-	0	

<replace 함수 사용(np.nan)>

	4월	상대	타율	타격
0	4.19	두산	NaN	0
1	4.21	롯데	0.000	1
2	4.22	롯데	0.000	1
3	4.26	두산	1.000	1
4	5.01	LG	0.000	1

<dropna 함수 사용>

	4월	상대	타율	타격
1	4.21	롯데	0.000	1
2	4.22	롯데	0.000	1
3	4.26	두산	1.000	1
4	5.01	LG	0.000	1
7	5.05	SK	0.500	2

열 삭제

<열 삭제 전>

<df.drop(['컬럼명', ...], axis=1) 사용>

	4월	상대	타율	타격	타석	득점	안타	2루타	3루타	홈런	타점	도루	도루실패	볼넷	사구	삼진	병살타	AVG2
1	4.21	롯데	0.000	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0.000
2	4.22	롯데	0.000	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0.000
3	4.26	두산	1.000	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0.333
4	5.01	LG	0.000	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0.250
7	5.05	SK	0.500	2	2	0	1	0	0	0	1	0	0	0	0	0	0	0.333
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1623	6.27	한화	0.500	4	4	2	2	0	0	1	4	0	0	0	0	2	0	0.309
1624	6.30	LG	0.500	4	4	1	2	0	0	0	1	0	1	0	0	0	0	0.314
1625	7.01	LG	0.000	5	5	1	0	0	0	0	0	0	0	0	0	2	0	0.304
1626	7.02	키움	0.500	4	4	2	2	0	0	0	0	0	0	0	0	0	0	0.309
1627	7.04	키움	0.400	5	5	2	2	0	0	0	3	0	0	0	0	0	0	0.312

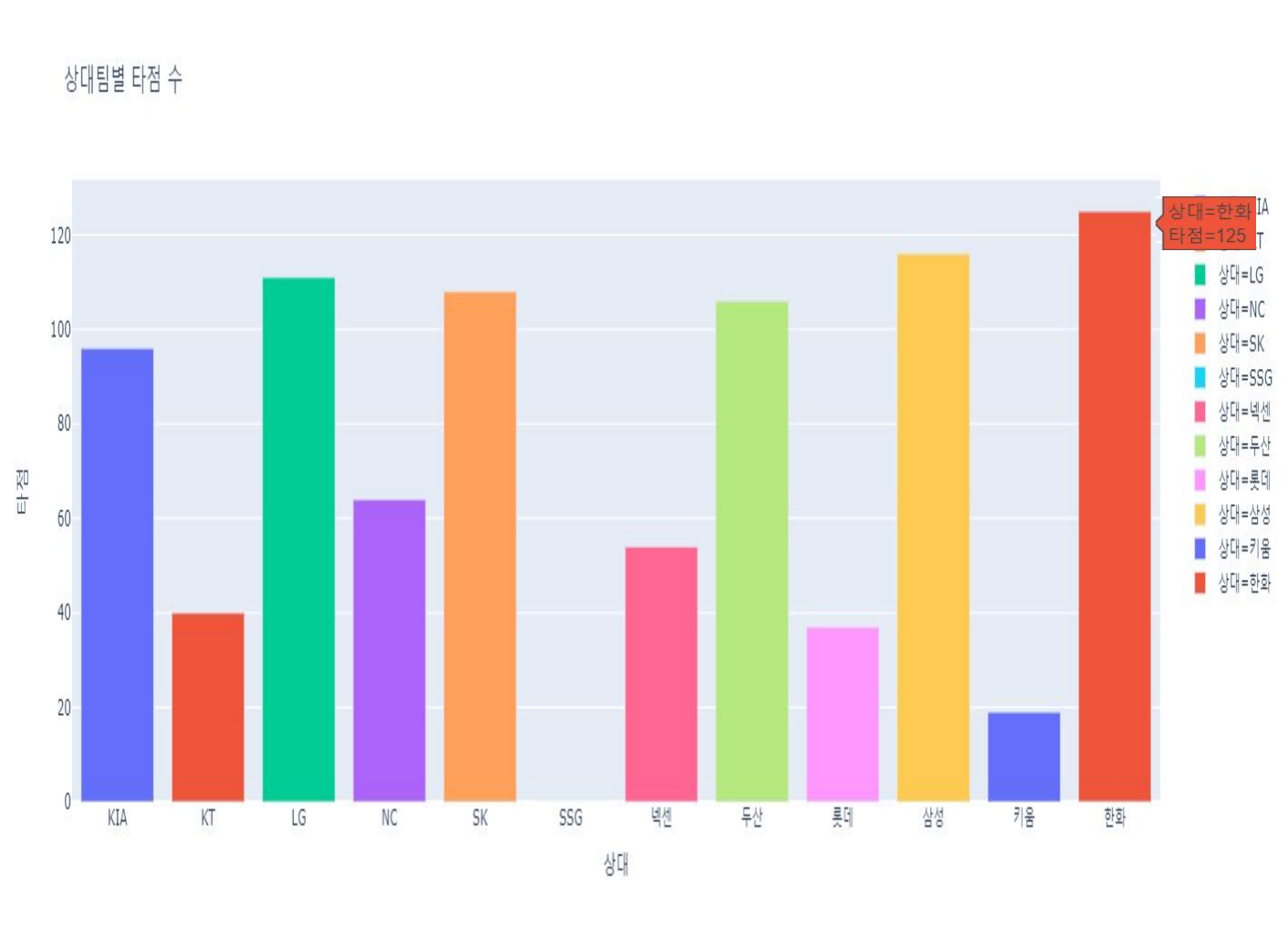
1591 rows × 18 columns

<열 삭제 후>

	상대	타율	타석	안타	2루타	3루타	홈런	타점
1	롯데	0.000	1	0	0	0	0	0
2	롯데	0.000	1	0	0	0	0	0
3	두산	1.000	1	1	0	0	0	0
4	LG	0.000	1	0	0	0	0	0
7	SK	0.500	2	1	0	0	0	1
...	...	...	...	...	...	...	...	...
1623	한화	0.500	4	2	0	0	1	4
1624	LG	0.500	4	2	0	0	0	1
1625	LG	0.000	5	0	0	0	0	0
1626	키움	0.500	4	2	0	0	0	0
1627	키움	0.400	5	2	0	0	0	3

1591 rows × 8 columns

상대팀 컬럼 처리



- ☑ 선수마다 팀 별 상성이 있을 것이라고 생각
- ☑ 실제로 강한 팀과 약한 팀이 존재해 X변수에 포함하도록 함

Onehotencoding

<pd.get_dummies(df['상대'])>

	상대	타율	타석	안타	2루타	3루타	홈런	타점
1	롯데	0.000	1	0	0	0	0	0
2	롯데	0.000	1	0	0	0	0	0
3	두산	1.000	1	1	0	0	0	0
4	LG	0.000	1	0	0	0	0	0
7	SK	0.500	2	1	0	0	0	1
...	...	...	...	...	...	...	...	...
1623	한화	0.500	4	2	0	0	1	4
1624	LG	0.500	4	2	0	0	0	1
1625	LG	0.000	5	0	0	0	0	0
1626	키움	0.500	4	2	0	0	0	0
1627	키움	0.400	5	2	0	0	0	3

1591 rows × 8 columns

<'상대' 열 더미 값 구성>

	KIA	KT	LG	NC	SK	SSG	넥센	두산	롯데	삼성	키움	한화
1	0	0	0	0	0	0	0	0	1	0	0	0
2	0	0	0	0	0	0	0	0	1	0	0	0
3	0	0	0	0	0	0	0	1	0	0	0	0
4	0	0	1	0	0	0	0	0	0	0	0	0
7	0	0	0	0	1	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...
1623	0	0	0	0	0	0	0	0	0	0	0	1
1624	0	0	1	0	0	0	0	0	0	0	0	0
1625	0	0	1	0	0	0	0	0	0	0	0	0
1626	0	0	0	0	0	0	0	0	0	0	1	0
1627	0	0	0	0	0	0	0	0	0	0	1	0

1591 rows × 12 columns

Onehotencodin

<pd.concat([x_df, y_df], axis=1)> 함수 사용하여 붙이기

- x_df : 붙임 당하는 df
- y_df : 붙이려는 df

	상대	타율	타석	안타	2루타	3루타	홈런	타점	KIA	KT	LG	NC	SK	SSG	넥센	두산	롯데	삼성	키움	한화
1	롯데	0.000	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
2	롯데	0.000	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
3	두산	1.000	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
4	LG	0.000	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
7	SK	0.500	2	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1623	한화	0.500	4	2	0	0	1	4	0	0	0	0	0	0	0	0	0	0	0	1
1624	LG	0.500	4	2	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0
1625	LG	0.000	5	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
1626	키움	0.500	4	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1627	키움	0.400	5	2	0	0	0	3	0	0	0	0	0	0	0	0	0	0	1	0

1591 rows x 20 columns

<pd.drop(['상대'], axis=1)> 함수로 기존 컬럼 삭제

	타율	타석	안타	2루타	3루타	홈런	타점	KIA	KT	LG	NC	SK	SSG	넥센	두산	롯데	삼성	키움	한화
1	0.000	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
2	0.000	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
3	1.000	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
4	0.000	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
7	0.500	2	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1623	0.500	4	2	0	0	1	4	0	0	0	0	0	0	0	0	0	0	0	1
1624	0.500	4	2	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0
1625	0.000	5	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
1626	0.500	4	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1627	0.400	5	2	0	0	0	3	0	0	0	0	0	0	0	0	0	0	1	0

1591 rows x 19 columns

열 위치 변경

<열 위치를 변경>

- X_train, y_train, X_test, y_test 슬라이싱을 위해

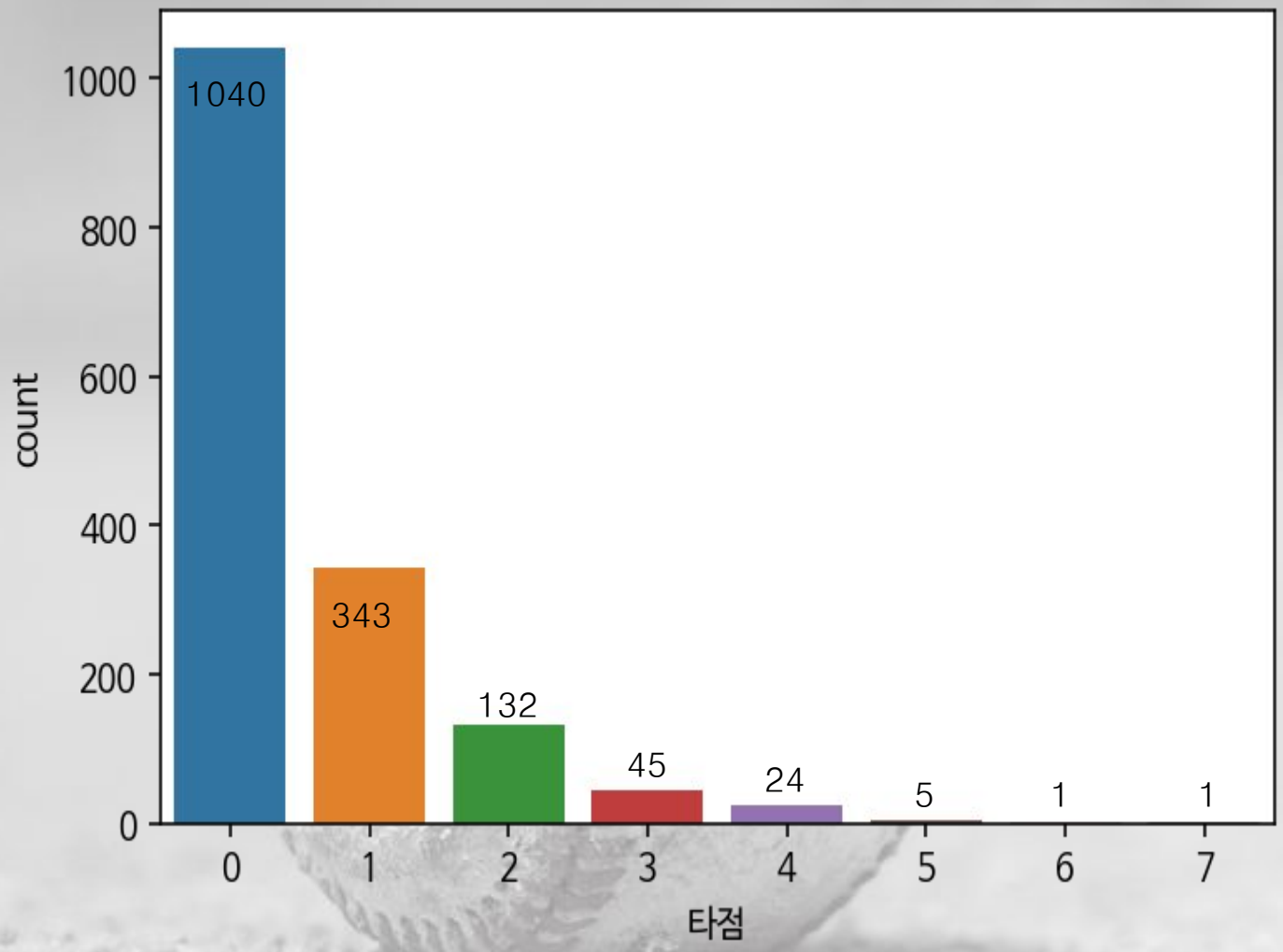
	타율	타석	안타	2루타	3루타	홈런	KIA	KT	LG	NC	SK	SSG	넥센	두산	롯데	삼성	키움	한화	타점
1	0.000	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
2	0.000	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
3	1.000	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
4	0.000	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
7	0.500	2	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1623	0.500	4	2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	4
1624	0.500	4	2	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
1625	0.000	5	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
1626	0.500	4	2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
1627	0.400	5	2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	3

1591 rows x 19 columns

```
# 데이터 셋이 전부 숫자이므로 numpy로 np.array로 바꿔준다.
Hwang_data = np.array(merged_df)
type(Hwang_data)
```

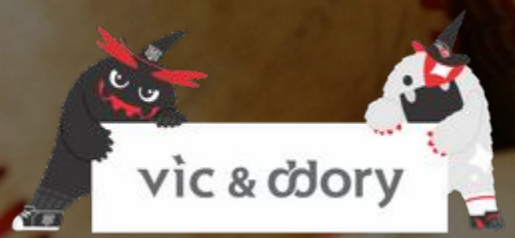
```
numpy.ndarray
```

레이블(타점)확인



- 0 타점이 전체 데이터의 약 65%를 차지

회귀



```
# 학습 데이터 만들기
# X_train에 학습 데이터에 1547개와 18개 변수
X_train = Hwang_data[:1547, 0:18].astype('float')
print(X_train.shape)

# y_train에 학습 데이터 1547개와 1개 타겟
y_train = Hwang_data[:1547, 18].astype('float')
print(y_train.shape)

# X_test 데이터에 44개와 18개 변수
X_test = Hwang_data[1547:, 0:18].astype('float')
print(X_test.shape)

# y_test에 학습 데이터 44개와 1개 타겟
y_test = Hwang_data[1547:, 18].astype('float')
print(y_test.shape)
```

```
(1547, 18)
(1547,)
(44, 18)
(44,)
```

- ☑ 학습 데이터와 테스트 데이터 슬라이싱하기
- ☑ 테스트 데이터로 사용할 데이터들은 21년도의 황재균의 성적

```
# 검증 데이터 셋 만들기

from sklearn.model_selection import train_test_split

# 훈련 데이터에서 검증데이터를 7대 3의 비율로 분리하기
X_train, X_val, y_train, y_val = train_test_split(X_train,
                                                  y_train,
                                                  test_size = 0.3,
                                                  random_state = 777)

print(X_train.shape, X_val.shape, y_train.shape, y_val.shape)

# X_train: 1082개의 데이터가 들어감
# X_val: 465개의 데이터가 들어감
```

```
(1078, 18) (462, 18) (1078,) (462,)
```

- ☑ 검증 데이터를 만들기
- ☑ 약 1500개의 학습데이터에서 약 1000개의 학습데이터와 460개의 검증 데이터로 나눔

모델 구성하기

```
[ ]  model = Sequential()

     model.add(Dense(64, activation='relu', input_shape=(18,)))
     model.add(Dense(32, activation='relu'))
     model.add(Dense(1, activation='linear'))
```

모델 설정하기

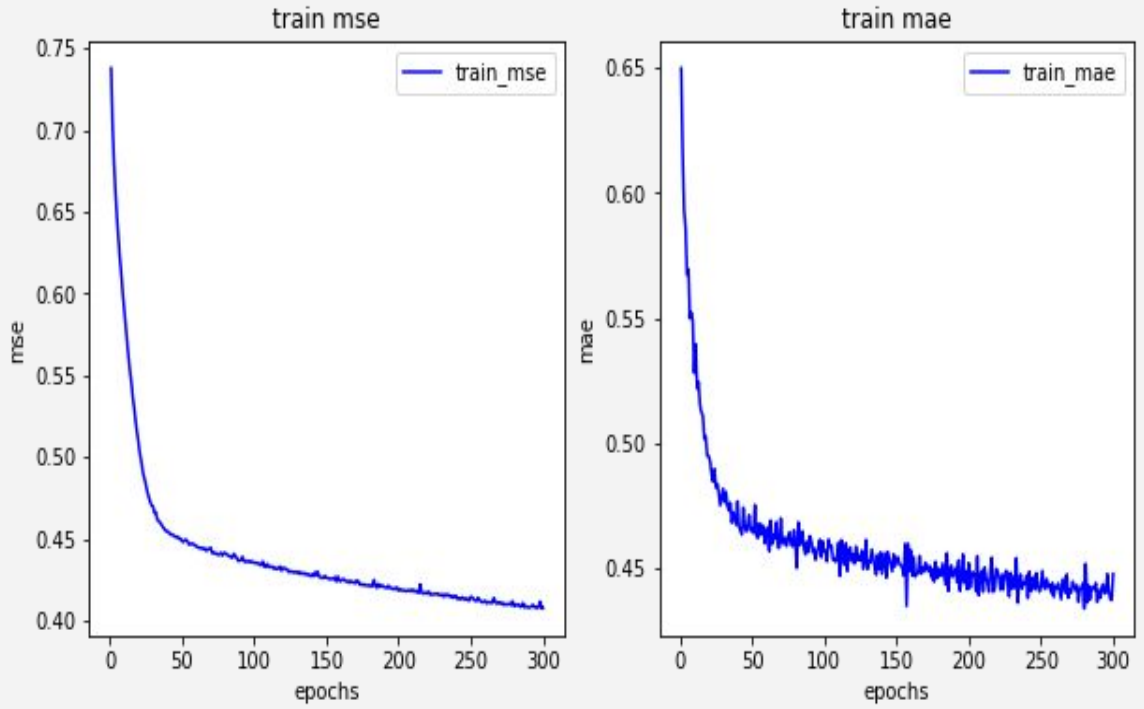
```
[ ]  model.compile(optimizer=Adam(learning_rate=0.0001), loss='mse', metrics=['mae', 'mse'])
```

모델 학습하기

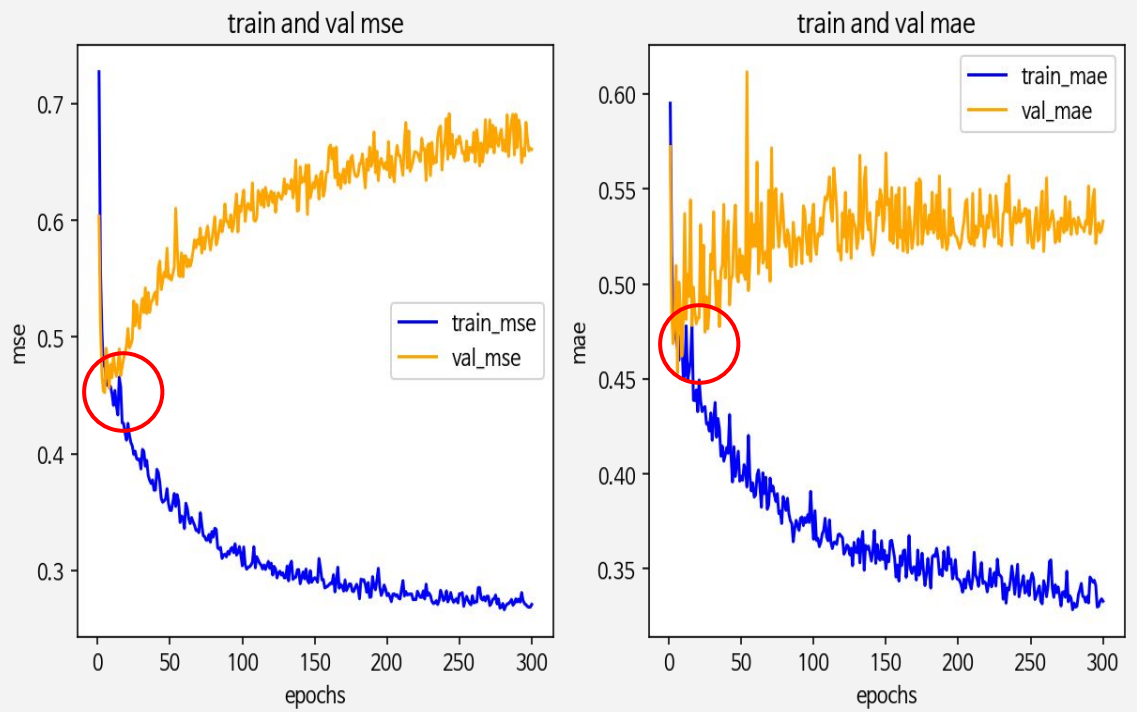
```
[ ]  history = model.fit(X_train,
                         y_train,
                         epochs = 300)
```

- ☑ 활성화 함수는 relu와 linear사용
- ☑ optimizer는 Adam,
  손실함수는 mse,
  metrics는 mae와 mse를 사용
- ☑ epochs는 300번을 줌

<모델 그려보기>



<모델 그려보기>



<과대적합>

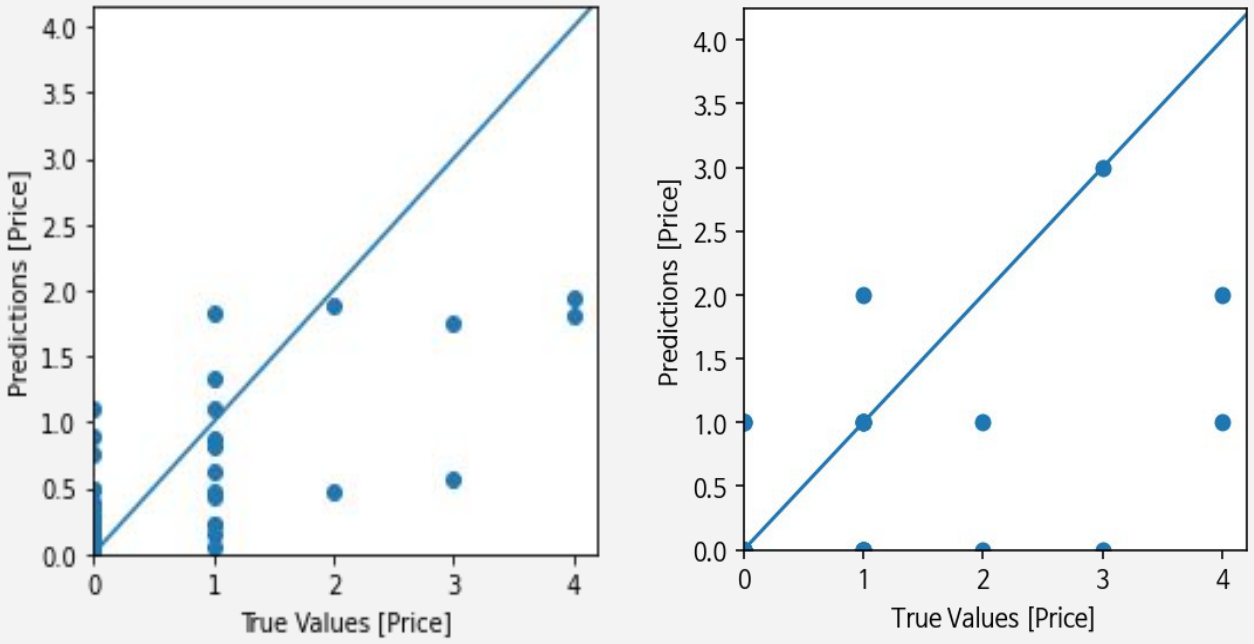
모델 평가하기

```
[ ]  print(model.evaluate(X_test, y_test))
```

```
2/2 [==============================] - 0s 7ms/step - loss: 0.6204 - mae: 0.5365 - mse: 0.6204
[0.6203550100326538, 0.536532461643219, 0.6203550100326538]
```

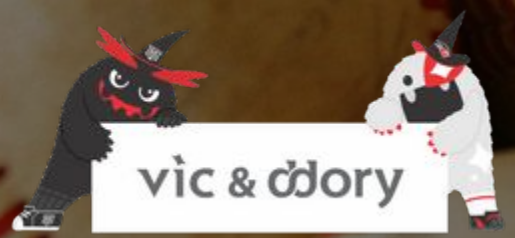
☑ 모델 평가에서 mae와 mse는 각각 0.53, 0.62로 좋지 않은 결과를 보이고 있다.

<모델 예측하기>

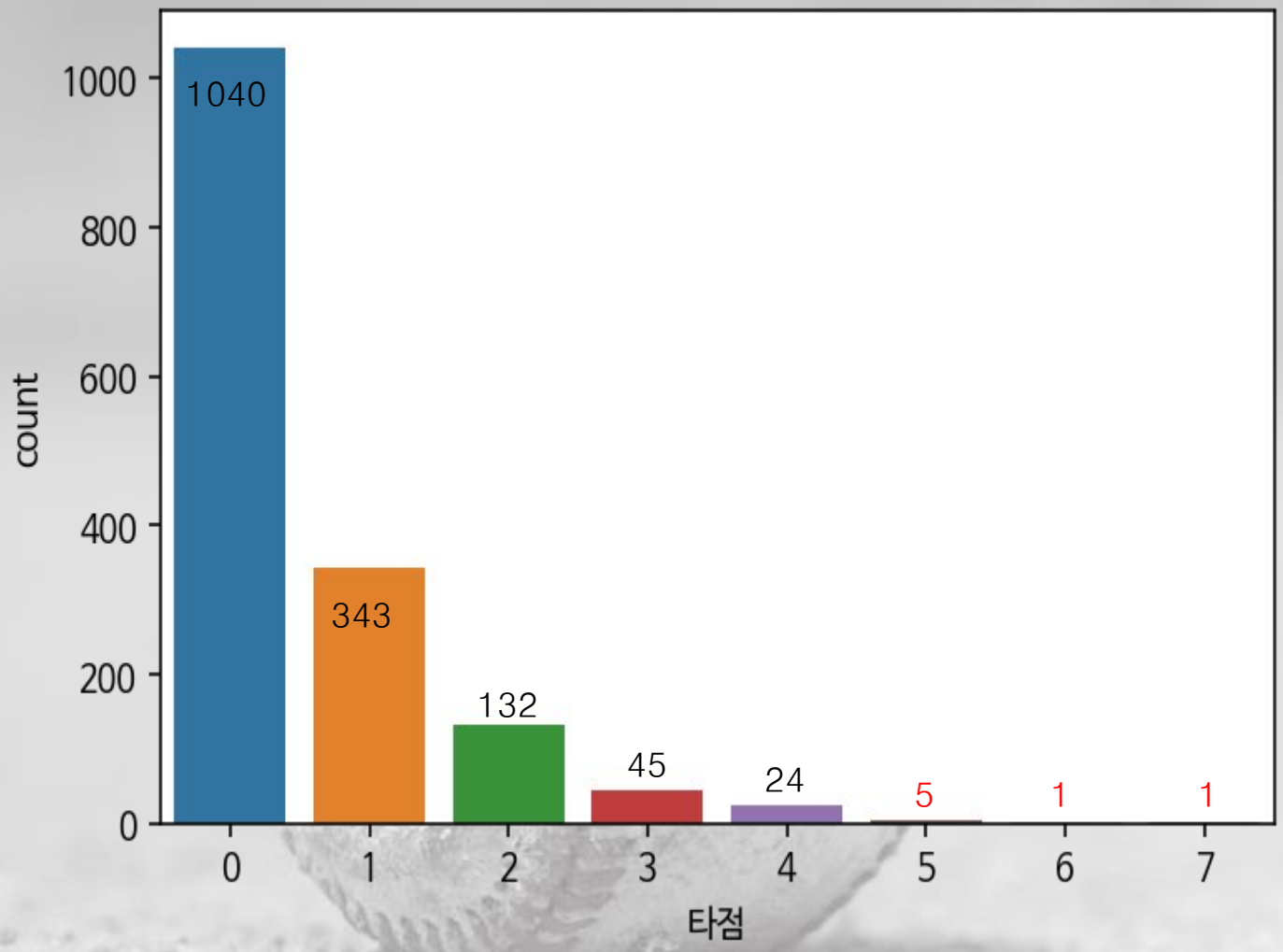


- ☑ 왼쪽의 scatter 그래프는 예측값을 실수로 계산함
- ☑ **타점은 정수로 실수가 나올 수 없음**
- ☑ 오른쪽의 scatter 그래프는 예측값을 반올림하여 정수로 만듦

다중분류



결측치 제거



☑ 5 타점부터 데이터가 부족해 이상치로 판단해 제거함

```
from tensorflow.keras.utils import to_categorical
# 타점 데이터의 레이블 0 ~ 5 숫자 값을 범주형 형태로 변경해준다.
y_train = to_categorical(y_train)
y_val = to_categorical(y_val)
y_test = to_categorical(y_test)
```

☑ softmax 함수를 사용하기 위해 레이블을 범주형 데이터로 만듦

모델 구성하기

```
[ ]  model = Sequential()

     model.add(Dense(64, activation='relu', input_shape=(18,)))
     model.add(Dense(32, activation='relu'))
     model.add(Dense(5, activation='softmax'))
```

모델 설정하기

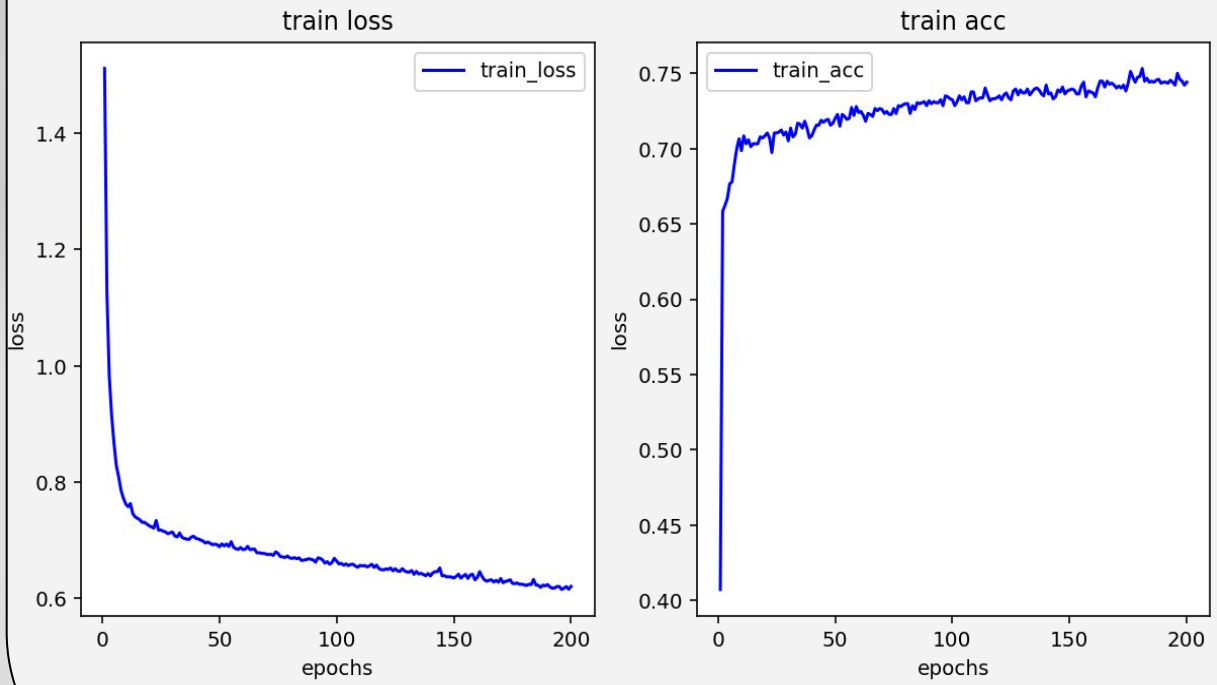
```
[ ]  model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

모델 학습하기

```
[ ]  history = model.fit(X_train,
                         y_train,
                         epochs = 200,
                         batch_size=128,
                         validation_data = (X_val, y_val))
```

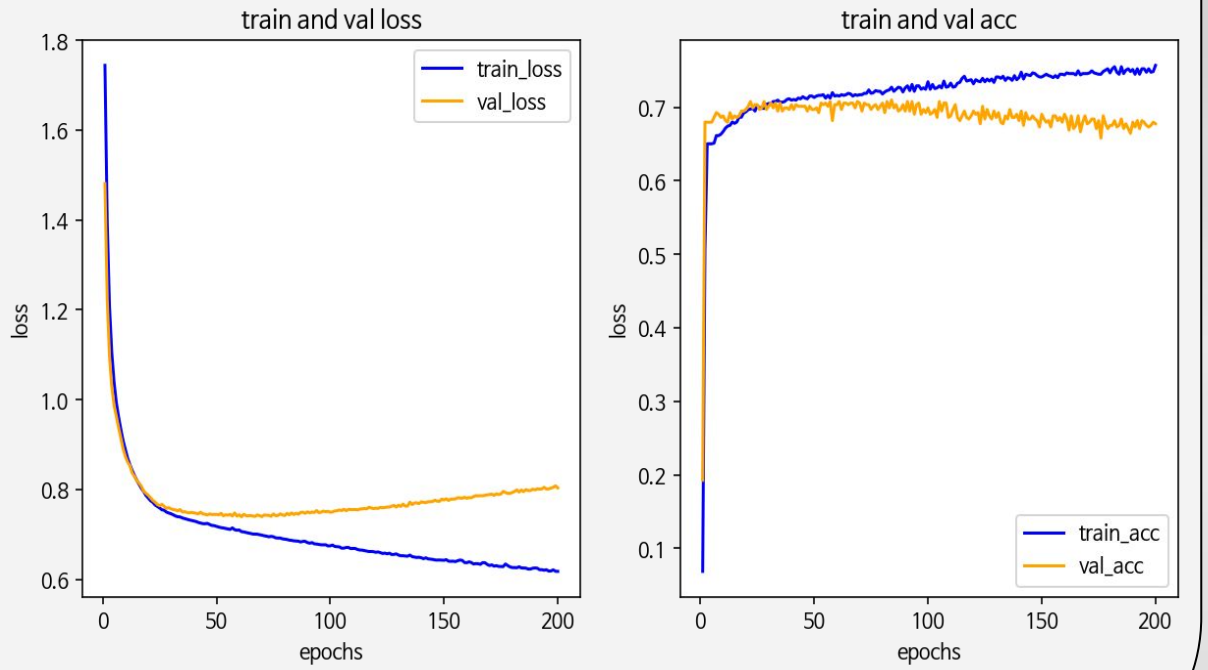
- ☑ 마지막 밀집층에는 softmax 함수를 사용
- ☑ 다중분류이므로 손실함수로 categorical_crossentropy를 사용
- ☑ 정확도 파악을 위해 accuracy를 사용

<모델 그려보기>



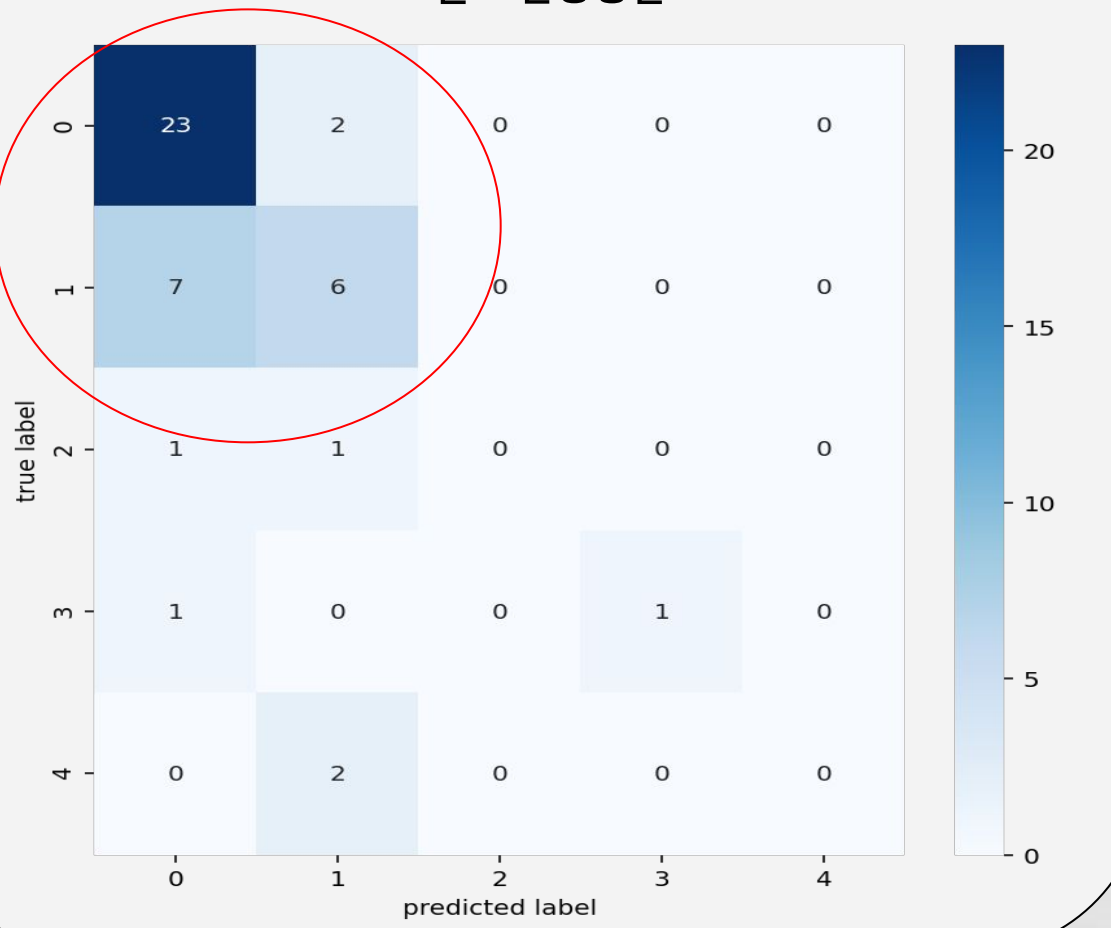
정확도 : 68.18%

<모델 그려보기>

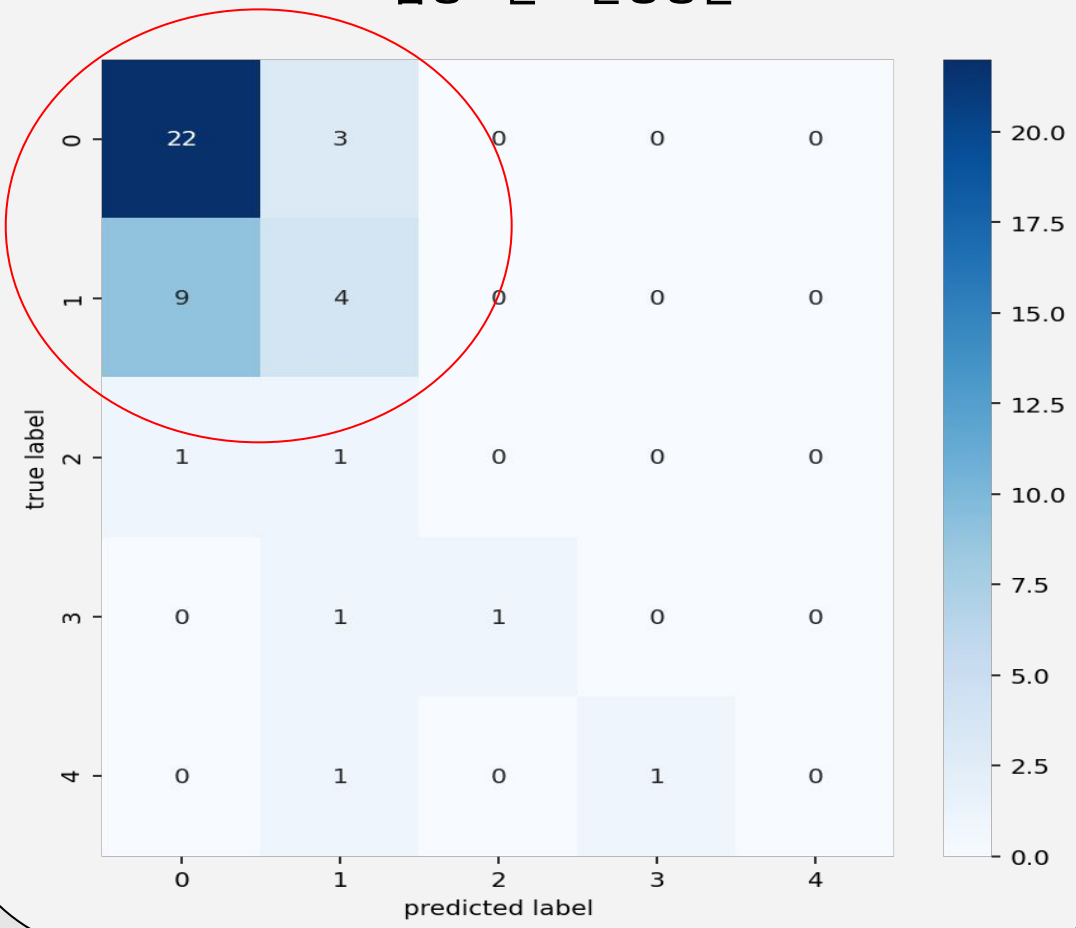


정확도 : 59.09%

<모델 – 혼동행렬>



<검증모델 – 혼동행렬>

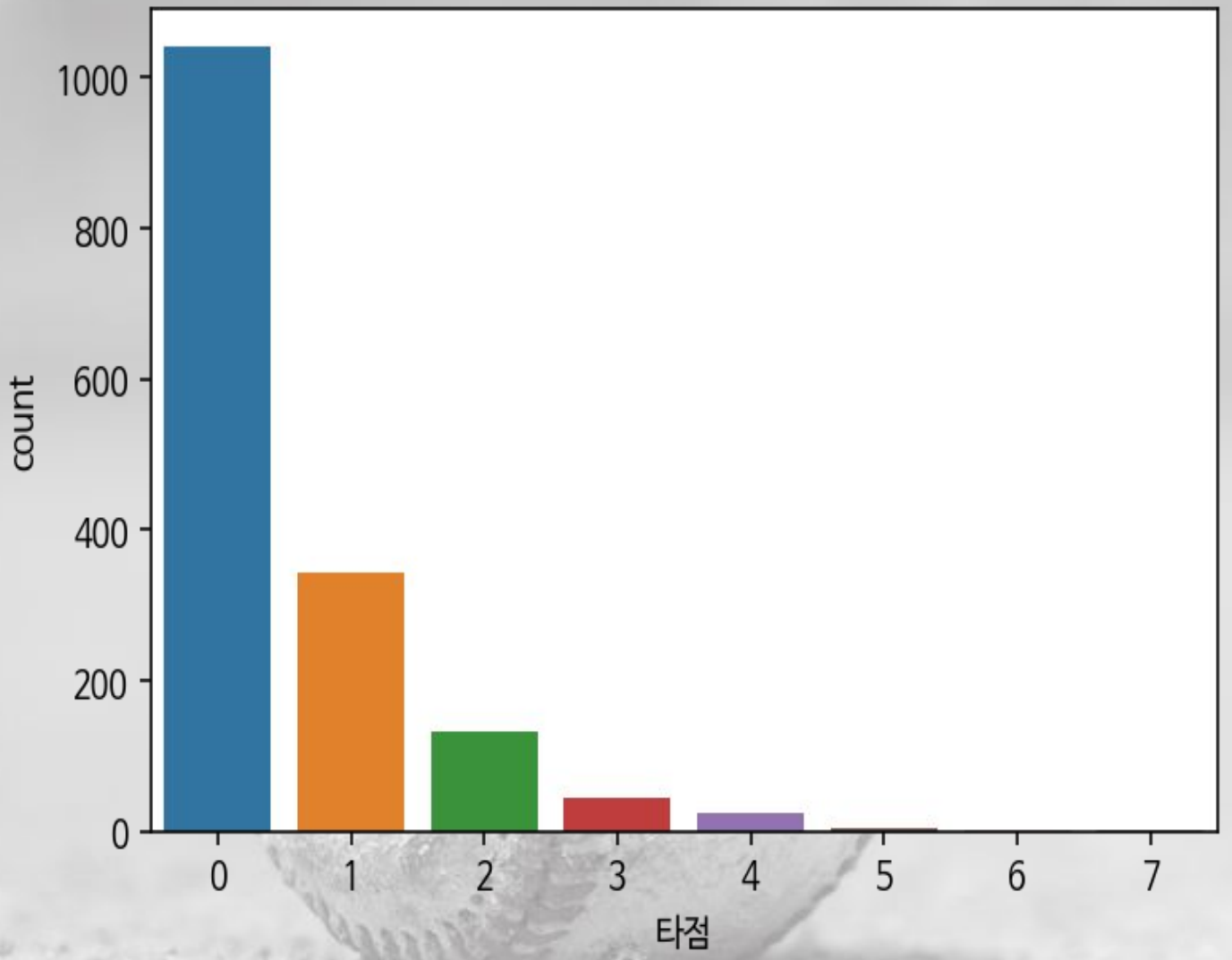


<모델 – 보고서>

	precision	recall	f1-score	support
0	0.72	0.92	0.81	25
1	0.55	0.46	0.50	13
2	0.00	0.00	0.00	2
3	1.00	0.50	0.67	2
4	0.00	0.00	0.00	2
accuracy			0.68	44
macro avg	0.45	0.38	0.39	44
weighted avg	0.61	0.68	0.64	44

<검증모델 – 보고서>

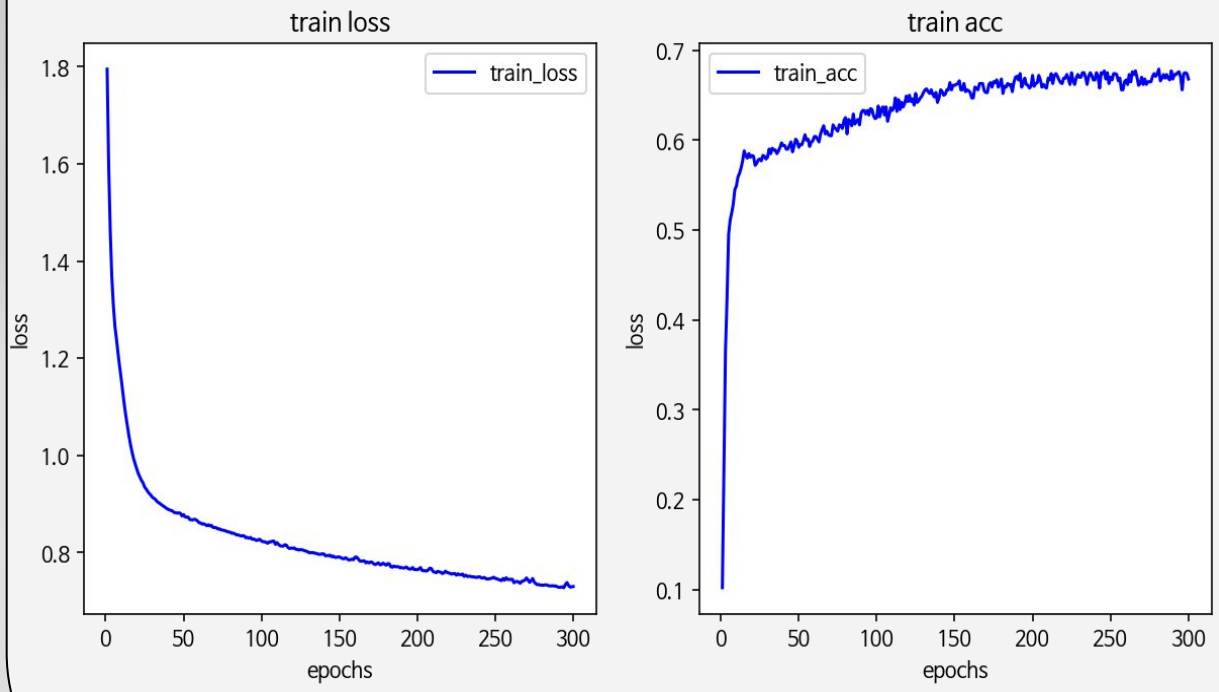
	precision	recall	f1-score	support
0	0.69	0.88	0.77	25
1	0.40	0.31	0.35	13
2	0.00	0.00	0.00	2
3	0.00	0.00	0.00	2
4	0.00	0.00	0.00	2
accuracy			0.59	44
macro avg	0.22	0.24	0.22	44
weighted avg	0.51	0.59	0.54	44



KEY POINT

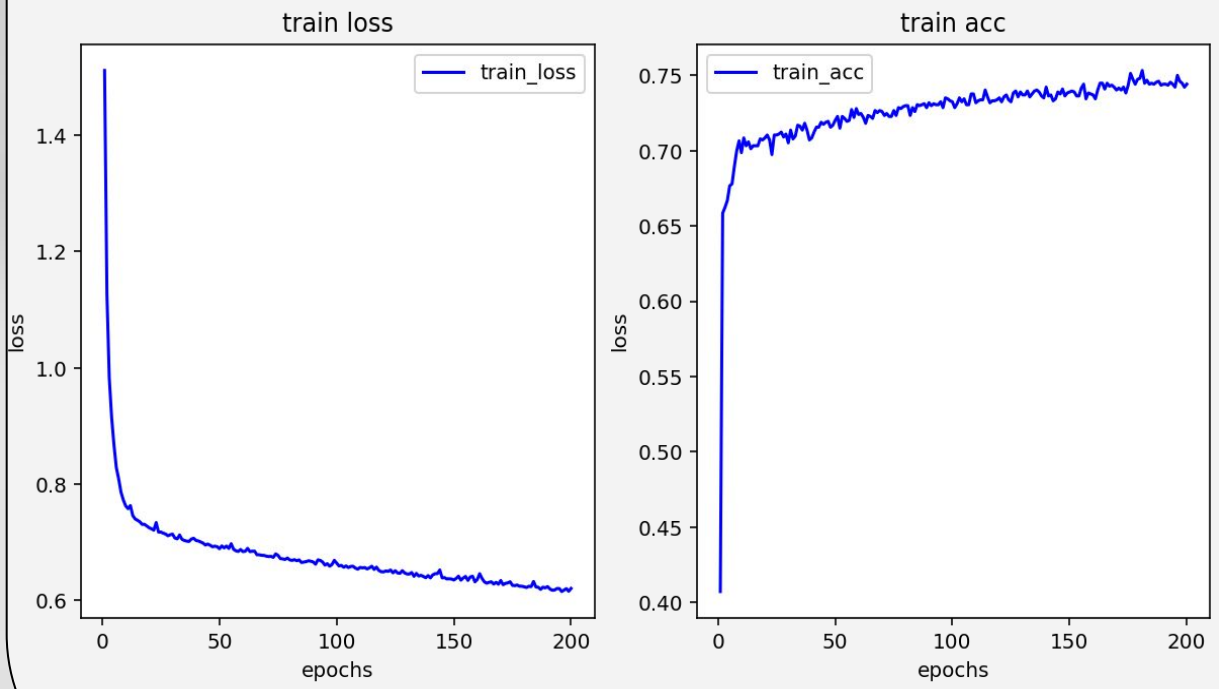
- 혼동행렬과 보고서를 보면 0 타점 예측은 높은 정확도를 보이며 맞추고 있음
- 하지만 2 타점부터의 정확도는 굉장히 좋지 않음
- 0 타점의 데이터가 너무 많기 때문인 것 같아 0 타점의 데이터를 절반으로 줄이고 다시 학습시켜보기

<'0 타점' 줄인 모델 그려보기>



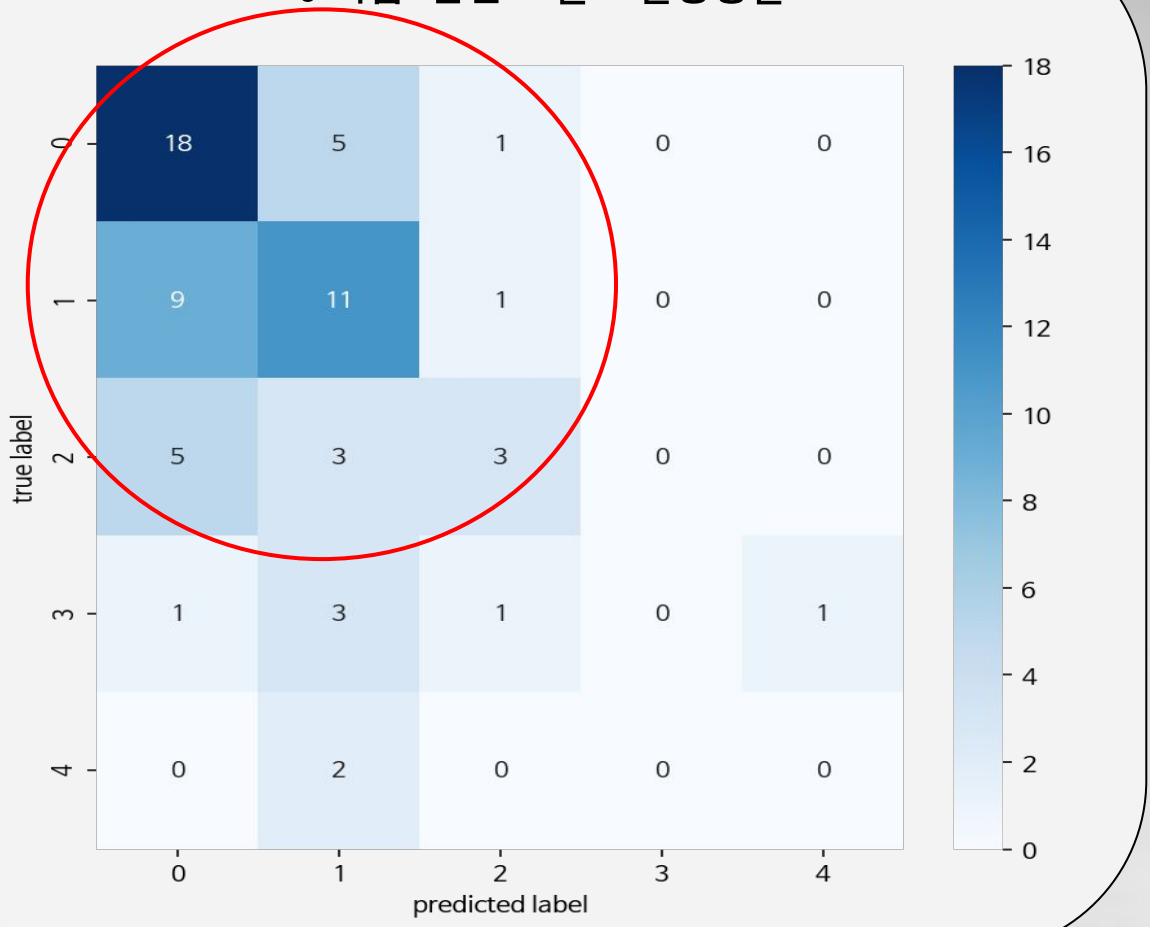
정확도 : 50.00%

<모델 그려보기>

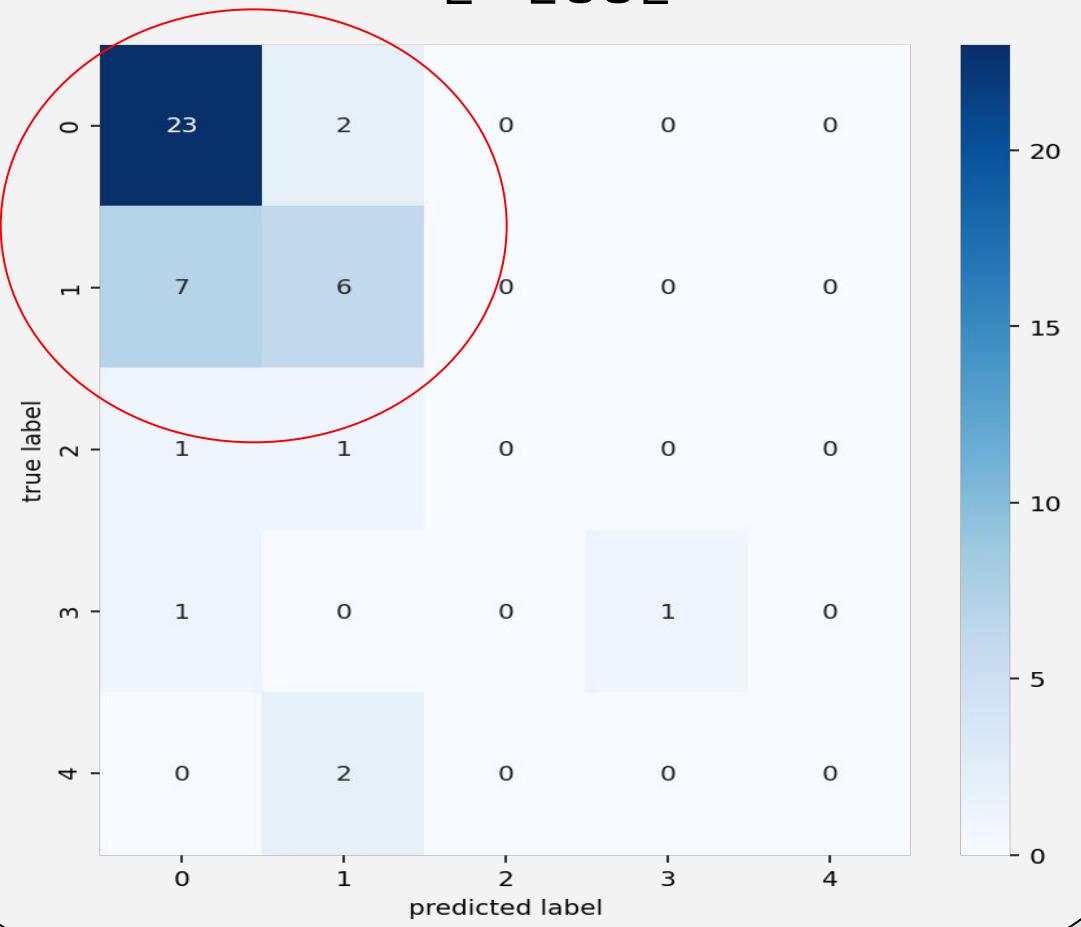


정확도 : 68.18%

<'0 타점' 줄인 모델 – 혼동행렬>



<모델 – 혼동행렬>



<'0 타점' 줄인 모델 – 보고서>

	precision	recall	f1-score	support
0	0.55	0.75	0.63	24
1	0.46	0.52	0.49	21
2	0.50	0.27	0.35	11
3	0.00	0.00	0.00	6
4	0.00	0.00	0.00	2
accuracy			0.50	64
macro avg	0.30	0.31	0.29	64
weighted avg	0.44	0.50	0.46	64

<모델 – 보고서>

	precision	recall	f1-score	support
0	0.72	0.92	0.81	25
1	0.55	0.46	0.50	13
2	0.00	0.00	0.00	2
3	1.00	0.50	0.67	2
4	0.00	0.00	0.00	2
accuracy			0.68	44
macro avg	0.45	0.38	0.39	44
weighted avg	0.61	0.68	0.64	44

KEY POINT

- ☑ 0 타점을 절반 줄인 데이터로 학습을 시켰을 때 정확도는 낮아졌지만 1타점과 2타점을 맞추는 정확도가 높아졌다는 점은 유의미한 결과라고 생각함
- ☑ 2, 3, 4 타점의 데이터의 양이 더 많았다면 정확도가 더 높아질 것이라고 예측
- ☑ 타점에 영향을 주는 변수들은 상대팀, 안타, 타율, 타석 뿐만 아니라 날씨, 컨디션, 운, 부상여부, 관중 수 등 더 다양하고 복잡한 변수들에 영향을 받지만 그 부분을 포함하지 못함 (데이터의 한계)
- ☑ 눈으로 응원만 해오던 팀을 이번 기회를 통해 분석해 봄으로써 더 큰 애정과 팬심이 생겼음


Coca-Cola
VOLVO
Bank of America
10 SHEFFIELD
2009 WITH RISP
AVG AB H HR
.250 28 7 0
COVIDIEN
STANLEY
MAKE SOMETHING GREAT

THANK YOU!!