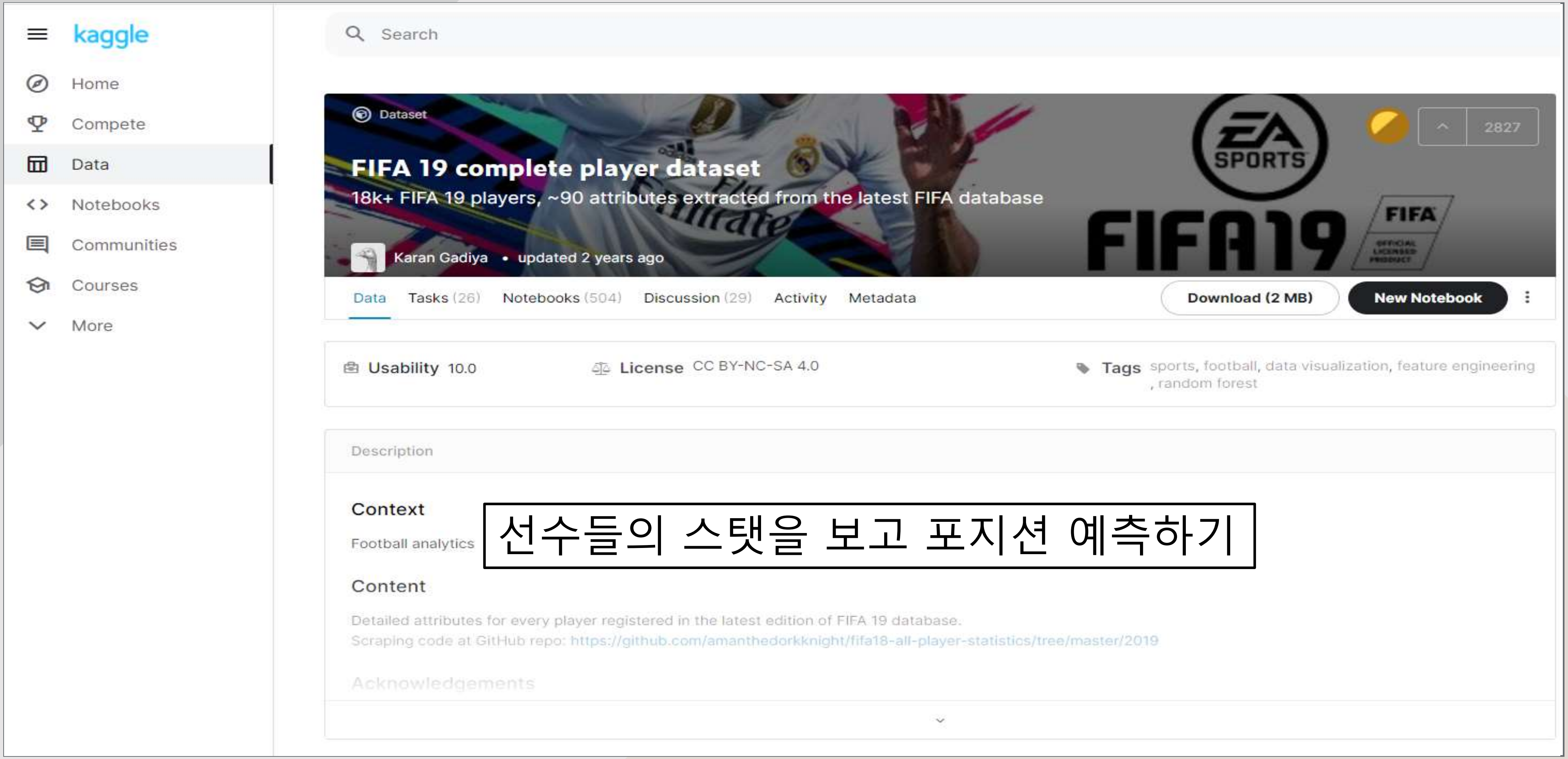
머신러닝 – 기말 프로젝트(자유)

20165517 오기용

콘텐츠 IT학과 20165517 오기용

목차

데이터 소개



kaggle
Search
Home
Compete
Data
Notebooks
Communities
Courses
More
Dataset
FIFA 19 complete player dataset
18k+ FIFA 19 players, ~90 attributes extracted from the latest FIFA database
Karan Gadiya • updated 2 years ago
2827
Data Tasks (26) Notebooks (504) Discussion (29) Activity Metadata
Download (2 MB)
New Notebook
Usability 10.0
License CC BY-NC-SA 4.0
Tags sports, football, data visualization, feature engineering, random forest
Description
Context
Football analytics
선수들의 스탯을 보고 포지션 예측하기
Content
Detailed attributes for every player registered in the latest edition of FIFA 19 database.
Scraping code at GitHub repo: https://github.com/amanthedorkknight/fifa18-all-player-statistics/tree/master/2019
Acknowledgements

데이터 소개

ID	Name	Age	Photo	Nationalit	Flag	Overall	Potential	Club	Club Logc	Value	Wage	Special	Preferred	Internatio	Weak Foo	Skill Move	Work Rate	Body Type	Real Face	Position	Jersey Nur	Joined	Loaned Fr	Contract \	Height
158023	L. Messi	31	https://cdr	Argentina	https://cdr	94	94	FC Barcelc	https://cdr	€110.5M	€565K	2202	Left	5	4	4	Medium/	Messi	Yes	RF	10	01-Jul-04		2021	5'7
20801	Cristiano F	33	https://cdr	Portugal	https://cdr	94	94	Juventus	https://cdr	€77M	€405K	2228	Right	5	4	5	High/ Low	C. Ronald	Yes	ST	7	10-Jul-18		2022	6'2
190871	Neymar Jr	26	https://cdr	Brazil	https://cdr	92	93	Paris Saint	https://cdr	€118.5M	€290K	2143	Right	5	5	5	High/ Me	Neymar	Yes	LW	10	#######		2022	5'9

Weight	LS	ST	RS	LW	LF	CF	RF	RW	LAM	CAM	RAM	LM	LCM	CM	RCM	RM	LWB	LDM	CDM	RDM	RWB	LB	LCB	CB	RCB
159lbs	88+2	88+2	88+2	92+2	93+2	93+2	93+2	92+2	93+2	93+2	93+2	91+2	84+2	84+2	84+2	91+2	64+2	61+2	61+2	61+2	64+2	59+2	47+2	47+2	47+2
183lbs	91+3	91+3	91+3	89+3	90+3	90+3	90+3	89+3	88+3	88+3	88+3	88+3	81+3	81+3	81+3	88+3	65+3	61+3	61+3	61+3	65+3	61+3	53+3	53+3	53+3
150lbs	84+3	84+3	84+3	89+3	89+3	89+3	89+3	89+3	89+3	89+3	89+3	88+3	81+3	81+3	81+3	88+3	65+3	60+3	60+3	60+3	65+3	60+3	47+3	47+3	47+3

RB	Crossing	Finishing	HeadingA	ShortPassi	Volleys	Dribbling	Curve	FKAccurac	LongPassi	BallContrc	Acceleratic	SprintSpee	Agility	Reactions	Balance	ShotPowe	Jumping	Stamina	Strength	LongShots	Aggressio	Interceptic	Positionin	Vision
59+2	84	95	70	90	86	97	93	94	87	96	91	86	91	95	95	85	68	72	59	94	48	22	94	94
61+3	84	94	89	81	87	88	81	76	77	94	89	91	87	96	70	95	95	88	79	93	63	29	95	82
60+3	79	87	62	84	84	96	88	87	78	95	94	90	96	94	84	80	61	81	49	82	56	36	89	87

Penalties	Composur	Marking	StandingT	SlidingTac	GKDiving	GKHandlir	GKKicking	GKPositioi	GKReflexe	Release Clause
75	96	33	28	26	6	11	15	14	8	€226.5M
85	95	28	31	23	7	11	15	14	11	€127.1M
81	94	27	24	33	9	9	15	15	11	€228.1M

행 : 18207개
열 : 89개

데이터 소개

데이터 소개

```python
# 사용할 모듈 불러오기
import pandas as pd
import numpy as np
import pickle
import seaborn as sns
%matplotlib inline
import matplotlib.pyplot as plt
```

```python
# 피파 데이터 읽기
fifa_data = pd.read_csv("C:/Users/multi050/fifa_data.csv")
```

```python
# 데이터 내용보기
fifa_data.head()
```

88+1개 특성

	Unnamed: 0	ID	Name	Age	Photo	Nationality	Flag	Overall	Potential	Club	...
0	0	158023	L. Messi	31	https://cdn.sofifa.org/players/4/19/158023.png	Argentina	https://cdn.sofifa.org/flags/52.png	94	94	FC Barcelona	...
1	1	20801	Cristiano Ronaldo	33	https://cdn.sofifa.org/players/4/19/20801.png	Portugal	https://cdn.sofifa.org/flags/38.png	94	94	Juventus	...
2	2	190871	Neymar Jr	26	https://cdn.sofifa.org/players/4/19/190871.png	Brazil	https://cdn.sofifa.org/flags/54.png	92	93	Paris Saint-Germain	...
3	3	193080	De Gea	27	https://cdn.sofifa.org/players/4/19/193080.png	Spain	https://cdn.sofifa.org/flags/45.png	91	93	Manchester United	...
4	4	192985	K. De Bruyne	27	https://cdn.sofifa.org/players/4/19/192985.png	Belgium	https://cdn.sofifa.org/flags/7.png	91	92	Manchester City	...

18206+1개의 샘플

5 rows × 89 columns

데이터 분석

데이터를 구성하는 정보보기

```
fifa_data.info()
'''
18207개의 샘플이 있고
54~88까지의 특성이 선수들의 스탯
28~53까지 선수들의 스탯을 종합적으로 평가해서 포지션별 점수 부여
21은 그 선수의 포지션
'''
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18207 entries, 0 to 18206
Data columns (total 89 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Unnamed: 0      18207 non-null  int64
 1   ID              18207 non-null  int64
 2   Name            18207 non-null  object
 3   Age             18207 non-null  int64
 4   Photo           18207 non-null  object
 5   Nationality     18207 non-null  object
 6   Flag            18207 non-null  object
 7   Overall         18207 non-null  int64
 8   Potential       18207 non-null  int64
 9   Club            17966 non-null  object
                 ⋮
 84  GKHandling      18159 non-null  float64
 85  GKKicking       18159 non-null  float64
 86  GKPositioning   18159 non-null  float64
 87  GKReflexes      18159 non-null  float64
 88  Release Clause  16643 non-null  object
dtypes: float64(38), int64(6), object(45)
memory usage: 12.4+ MB
```

```
 28  LS              16122 non-null  object
 29  ST              16122 non-null  object
 30  RS              16122 non-null  object
 31  LW              16122 non-null  object
 32  LF              16122 non-null  object
 33  CF              16122 non-null  object
 34  RF              16122 non-null  object
 35  RW              16122 non-null  object
 36  LAM             16122 non-null  object
 37  CAM             16122 non-null  object
 38  RAM             16122 non-null  object
 39  LM              16122 non-null  object
 40  LCM             16122 non-null  object
 41  CM              16122 non-null  object
 42  RCM             16122 non-null  object
 43  RM              16122 non-null  object
 44  LWB             16122 non-null  object
 45  LDM             16122 non-null  object
 46  CDM             16122 non-null  object
 47  RDM             16122 non-null  object
 48  RWB             16122 non-null  object
 49  LB              16122 non-null  object
 50  LCB             16122 non-null  object
 51  CB              16122 non-null  object
 52  RCB             16122 non-null  object
 53  RB              16122 non-null  object
 54  Crossing        18159 non-null  float64
 55  Finishing       18159 non-null  float64
 56  HeadingAccuracy 18159 non-null  float64
 57  ShortPassing    18159 non-null  float64
 58  Volleys         18159 non-null  float64
 59  Dribbling       18159 non-null  float64
 60  Curve           18159 non-null  float64
 61  FKAccuracy      18159 non-null  float64
 62  LongPassing     18159 non-null  float64
 63  BallControl     18159 non-null  float64
 64  Acceleration    18159 non-null  float64
 65  SprintSpeed     18159 non-null  float64
 66  Agility         18159 non-null  float64
 67  Reactions       18159 non-null  float64
 68  Balance         18159 non-null  float64
 69  ShotPower       18159 non-null  float64
 70  Jumping         18159 non-null  float64
 71  Stamina         18159 non-null  float64
 72  Strength        18159 non-null  float64
 73  LongShots       18159 non-null  float64
 74  Aggression      18159 non-null  float64
 75  Interceptions   18159 non-null  float64
 76  Positioning     18159 non-null  float64
 77  Vision          18159 non-null  float64
 78  Penalties       18159 non-null  float64
 79  Composure       18159 non-null  float64
 80  Marking         18159 non-null  float64
 81  StandingTackle  18159 non-null  float64
 82  SlidingTackle   18159 non-null  float64
 83  GKDiving        18159 non-null  float64
 84  GKHandling      18159 non-null  float64
 85  GKKicking       18159 non-null  float64
 86  GKPositioning   18159 non-null  float64
 87  GKReflexes      18159 non-null  float64
 88  Release Clause  16643 non-null  object
```

```
 21  Position        18147 non-null  object
```

데이터 분석

```python
# 위에서 언급한 선수의 포지션, 포지션별 점수, 스탯을 보여줌
player_position = pd.DataFrame({'Name': fifa_data.Name,'Position':fifa_data.Position,}) # 선수 이름, 포지션
player_Stat = fifa_data.iloc[:,54:88] # 스탯

player_information = pd.concat([player_position,player_Stat],axis=1)
player_information
```

필요한 특성 (포지션, 스탯)만 사용

	Name	Position	Crossing	Finishing	HeadingAccuracy	ShortPassing	Volleys	Dribbling	Curve	FKAccuracy	...	Penalties	Composure	Marking	StandingTackle	SlidingTackle	GKDiving	GKHandling	GKKicking	GKPositioning	GKReflexes
0	L. Messi	RF	84.0	95.0	70.0	90.0	86.0	97.0	93.0	94.0	...	75.0	96.0	33.0	28.0	26.0	6.0	11.0	15.0	14.0	8.0
1	Cristiano Ronaldo	ST	84.0	94.0	89.0	81.0	87.0	88.0	81.0	76.0	...	85.0	95.0	28.0	31.0	23.0	7.0	11.0	15.0	14.0	11.0
2	Neymar Jr	LW	79.0	87.0	62.0	84.0	84.0	96.0	88.0	87.0	...	81.0	94.0	27.0	24.0	33.0	9.0	9.0	15.0	15.0	11.0
3	De Gea	GK	17.0	13.0	21.0	50.0	13.0	18.0	21.0	19.0	...	40.0	68.0	15.0	21.0	13.0	90.0	85.0	87.0	88.0	94.0
4	K. De Bruyne	RCM	93.0	82.0	55.0	92.0	82.0	86.0	85.0	83.0	...	79.0	88.0	68.0	58.0	51.0	15.0	13.0	5.0	10.0	13.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
18202	J. Lundstram	CM	34.0	38.0	40.0	49.0	25.0	42.0	30.0	34.0	...	43.0	45.0	40.0	48.0	47.0	10.0	13.0	7.0	8.0	9.0
18203	N. Christoffersson	ST	23.0	52.0	52.0	43.0	36.0	39.0	32.0	20.0	...	43.0	42.0	22.0	15.0	19.0	10.0	9.0	9.0	5.0	12.0
18204	B. Worman	ST	25.0	40.0	46.0	38.0	38.0	45.0	38.0	27.0	...	55.0	41.0	32.0	13.0	11.0	6.0	5.0	10.0	6.0	13.0
18205	D. Walker-Rice	RW	44.0	50.0	39.0	42.0	40.0	51.0	34.0	32.0	...	50.0	46.0	20.0	25.0	27.0	14.0	6.0	14.0	8.0	9.0
18206	G. Nugent	CM	41.0	34.0	46.0	48.0	30.0	43.0	40.0	34.0	...	33.0	43.0	40.0	43.0	50.0	10.0	15.0	9.0	12.0	9.0

18207 rows × 36 columns

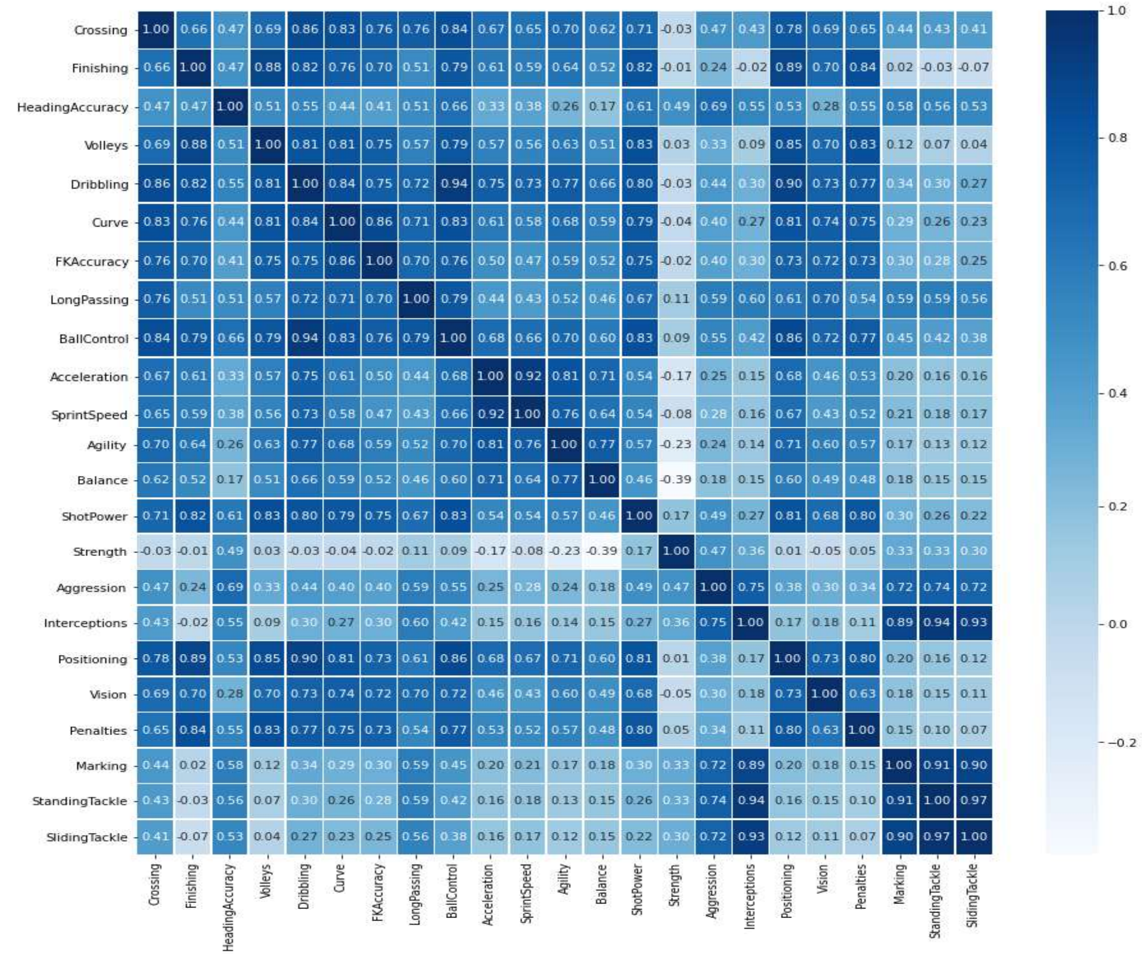
```python
fifa_data.iloc[:,28:53] # 포지션별 점수
```

	LS	ST	RS	LW	LF	CF	RF	RW	LAM	CAM	...	RM	LWB	LDM	CDM	RDM	RWB	LB	LCB	CB	RCB
0	88+2	88+2	88+2	92+2	93+2	93+2	93+2	92+2	93+2	93+2	...	91+2	64+2	61+2	61+2	61+2	64+2	59+2	47+2	47+2	47+2
1	91+3	91+3	91+3	89+3	90+3	90+3	90+3	89+3	88+3	88+3	...	88+3	65+3	61+3	61+3	61+3	65+3	61+3	53+3	53+3	53+3
2	84+3	84+3	84+3	89+3	89+3	89+3	89+3	89+3	89+3	89+3	...	88+3	65+3	60+3	60+3	60+3	65+3	60+3	47+3	47+3	47+3
3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	82+3	82+3	82+3	87+3	87+3	87+3	87+3	87+3	88+3	88+3	...	88+3	77+3	77+3	77+3	77+3	77+3	73+3	66+3	66+3	66+3
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
18202	42+2	42+2	42+2	44+2	44+2	44+2	44+2	44+2	45+2	45+2	...	44+2	44+2	45+2	45+2	45+2	44+2	45+2	45+2	45+2	45+2
18203	45+2	45+2	45+2	39+2	42+2	42+2	42+2	39+2	40+2	40+2	...	38+2	30+2	31+2	31+2	31+2	30+2	29+2	32+2	32+2	32+2
18204	45+2	45+2	45+2	45+2	46+2	46+2	46+2	45+2	44+2	44+2	...	44+2	34+2	30+2	30+2	30+2	34+2	33+2	28+2	28+2	28+2
18205	47+2	47+2	47+2	47+2	46+2	46+2	46+2	47+2	45+2	45+2	...	46+2	36+2	32+2	32+2	32+2	36+2	35+2	31+2	31+2	31+2
18206	43+2	43+2	43+2	45+2	44+2	44+2	44+2	45+2	45+2	45+2	...	46+2	46+2	46+2	46+2	46+2	46+2	46+2	47+2	47+2	47+2

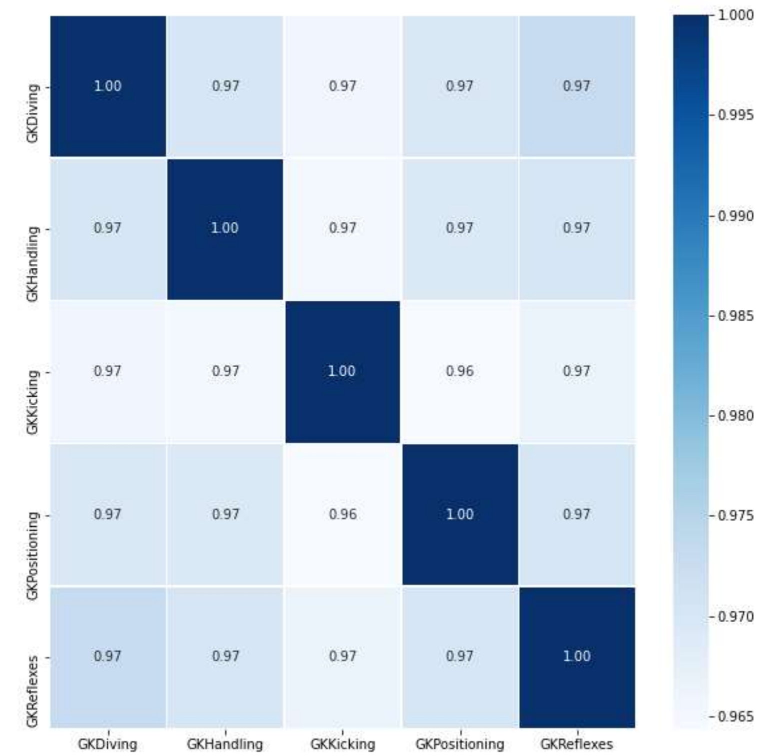
18207 rows × 25 columns

데이터 분석

```
# 히트맵 통해서 특성간 상관관계보기
%matplotlib qt5
plt.figure(figsize=(15,15))
sns.heatmap(data = player_information.iloc[:,2:25].corr(), annot=True, fmt='.2f', linewidth=.5, cmap='Blues')
```




```
plt.figure(figsize=(10,10))
sns.heatmap(data = player_information.iloc[:,25:34].corr(), annot=True, fmt='.2f', linewidth=.5, cmap='Blues')
```



데이터 수정

1. 특성(스탯) 중에서 분류에 영향을 주지않는 특성 제거
   → 포지션 별로 스탯을 비교했을 때 차이가 거의 없는 경우 삭제

2. 비슷한 영역의 포지션을 하나의 포지션으로 합쳐서 데이터 수 늘리기

   Ex) LB(왼쪽 수비수), RB(오른쪽 수비수), LCB(왼쪽 중앙 수비수), RCB(오른쪽 중앙 수비수)
   ↓
   CB(중앙 수비수)

3. NULL값 들어간 데이터 삭제

```python
#null값 존재해서 null값 제거
print("제거 전 : ", len(player_information))
player_information = player_information.dropna()
print("제거 후 : ", len(player_information))
```

```
제거 전 :  18207
제거 후 :  18147
```

4. 훈련set, 시험,set 나누기

특성 제거

대표 포지션 3개(공격, 수비, 미드필더)의 스탯 비교

```
ST=player_information.loc[player_information['Position']=='ST',:]
ST=ST[0:1000]
ST
```

	Name	Position	Crossing	Finishing	HeadingAccuracy	ShortPassing	Volleys	Drib
1	Cristiano Ronaldo	ST	84.0	94.0	89.0	81.0	87.0	
10	R. Lewandowski	ST	62.0	91.0	85.0	83.0	89.0	
16	H. Kane	ST	75.0	94.0	85.0	80.0	84.0	
23	S. Agüero	ST	70.0	93.0	77.0	81.0	85.0	
36	G. Bale	ST	87.0	86.0	84.0	85.0	85.0	
...	...	...	...	...	...	...	...	
9064	S. Davies	ST	65.0	64.0	70.0	60.0	65.0	
9073	S. Brandstetter	ST	49.0	65.0	65.0	58.0	58.0	
9075	D. Samuel	ST	49.0	65.0	67.0	60.0	57.0	
9084	Stéfano Pinho	ST	64.0	67.0	52.0	56.0	57.0	
9099	A. Baclet	ST	57.0	61.0	71.0	61.0	61.0	

1000 rows × 36 columns

```
CB=player_information.loc[player_information['Position']=='CB',:]
CB=CB[0:1000]
CB
```

	Name	Position	Crossing	Finishing	HeadingAccuracy	ShortPassing	Volleys	Dri
12	D. Godín	CB	55.0	42.0	92.0	79.0	47.0	
42	S. Umtiti	CB	69.0	51.0	79.0	81.0	70.0	
73	M. Benatia	CB	45.0	47.0	83.0	65.0	44.0	
89	N. Otamendi	CB	52.0	54.0	85.0	75.0	57.0	
102	Naldo	CB	45.0	57.0	94.0	76.0	60.0	
...	...	...	...	...	...	...	...	
11623	Kim Jungya	CB	29.0	47.0	64.0	60.0	31.0	
11640	V. Selimović	CB	32.0	21.0	65.0	54.0	26.0	
11643	C. Shaughnessy	CB	41.0	31.0	64.0	64.0	40.0	
11665	E. Wahlström	CB	43.0	31.0	61.0	50.0	22.0	
11674	M. Fontaine	CB	62.0	26.0	71.0	64.0	11.0	

1000 rows × 36 columns

```
CM=player_information.loc[player_information['Position']=='CM',:]
CM=CM[0:1000]
CM
```

	Name	Position	Crossing	Finishing	HeadingAccuracy	ShortPassing	Volleys	Dribb
67	Thiago	CM	72.0	69.0	54.0	90.0	90.0	
78	S. Milinković-Savić	CM	64.0	80.0	86.0	85.0	74.0	
121	Jorginho	CM	75.0	57.0	56.0	89.0	71.0	
136	I. Gündoğan	CM	74.0	73.0	50.0	88.0	75.0	
161	N. Keïta	CM	62.0	74.0	42.0	88.0	71.0	
...	...	...	...	...	...	...	...	
15244	P. Mbodji	CM	38.0	54.0	48.0	64.0	27.0	
15253	H. Heath	CM	50.0	39.0	54.0	66.0	45.0	
15256	T. Domgjoni	CM	36.0	39.0	58.0	61.0	38.0	
15267	K. Monlouis	CM	45.0	36.0	61.0	74.0	45.0	
15273	N. Husin	CM	59.0	55.0	44.0	63.0	44.0	

1000 rows × 36 columns

특성 제거

```
st_score=ST.iloc[:,2:].sum()
st_score
```

```
Crossing             53056.0
Finishing            72022.0
HeadingAccuracy      69315.0
ShortPassing         64010.0
Volleys              65661.0
Dribbling            68409.0
Curve                57955.0
FKAccuracy           50886.0
LongPassing          50200.0
BallControl          69878.0
Acceleration         69755.0
SprintSpeed          71157.0
Agility              68471.0
Reactions            68271.0
Balance              63652.0
ShotPower            72787.0
Jumping              71094.0
Stamina              67632.0
Strength             74172.0
LongShots            65026.0
Aggression           59417.0
Interceptions        29741.0
Positioning          72281.0
Vision               60203.0
Penalties            67313.0
Composure            67358.0
Marking              31441.0
StandingTackle       28220.0
SlidingTackle        24638.0
GKDiving             10591.0
GKHandling           10781.0
GKKicking            10793.0
GKPositioning        10560.0
GKReflexes           10578.0
dtype: float64
```

```
cb_score=CB.iloc[:,2:].sum()
cb_score
```

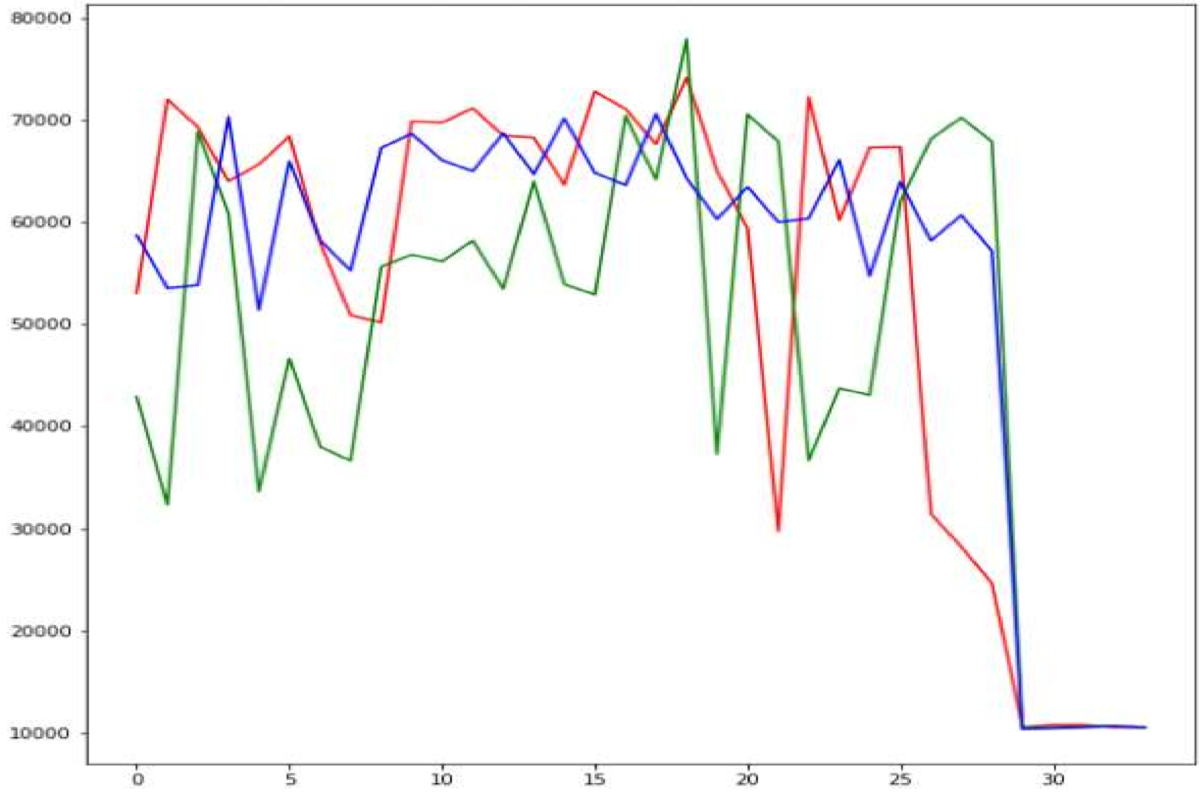
```
Crossing             42896.0
Finishing            32337.0
HeadingAccuracy      68895.0
ShortPassing         60850.0
Volleys              33652.0
Dribbling            46671.0
Curve                38030.0
FKAccuracy           36643.0
LongPassing          55647.0
BallControl          56827.0
Acceleration         56169.0
SprintSpeed          58199.0
Agility              53482.0
Reactions            63999.0
Balance              53931.0
ShotPower            52913.0
Jumping              70437.0
Stamina              64159.0
Strength             77946.0
LongShots            37305.0
Aggression           70544.0
Interceptions        67903.0
Positioning          36658.0
Vision               43737.0
Penalties            43071.0
Composure            62079.0
Marking              68152.0
StandingTackle       70247.0
SlidingTackle        67874.0
GKDiving             10591.0
GKHandling           10507.0
GKKicking            10629.0
GKPositioning        10669.0
GKReflexes           10555.0
dtype: float64
```

```
cm_score=CM.iloc[:,2:].sum()
cm_score
```

```
Crossing             58731.0
Finishing            53550.0
HeadingAccuracy      53846.0
ShortPassing         70328.0
Volleys              51412.0
Dribbling            65959.0
Curve                58236.0
FKAccuracy           55260.0
LongPassing          67274.0
BallControl          68671.0
Acceleration         66054.0
SprintSpeed          64996.0
Agility              68729.0
Reactions            64707.0
Balance              70175.0
ShotPower            64838.0
Jumping              63627.0
Stamina              70600.0
Strength             64317.0
LongShots            60303.0
Aggression           63452.0
Interceptions        60028.0
Positioning          60371.0
Vision               66104.0
Penalties            54729.0
Composure            63944.0
Marking              58184.0
StandingTackle       60701.0
SlidingTackle        57163.0
GKDiving             10388.0
GKHandling           10469.0
GKKicking            10561.0
GKPositioning        10665.0
GKReflexes           10567.0
dtype: float64
```

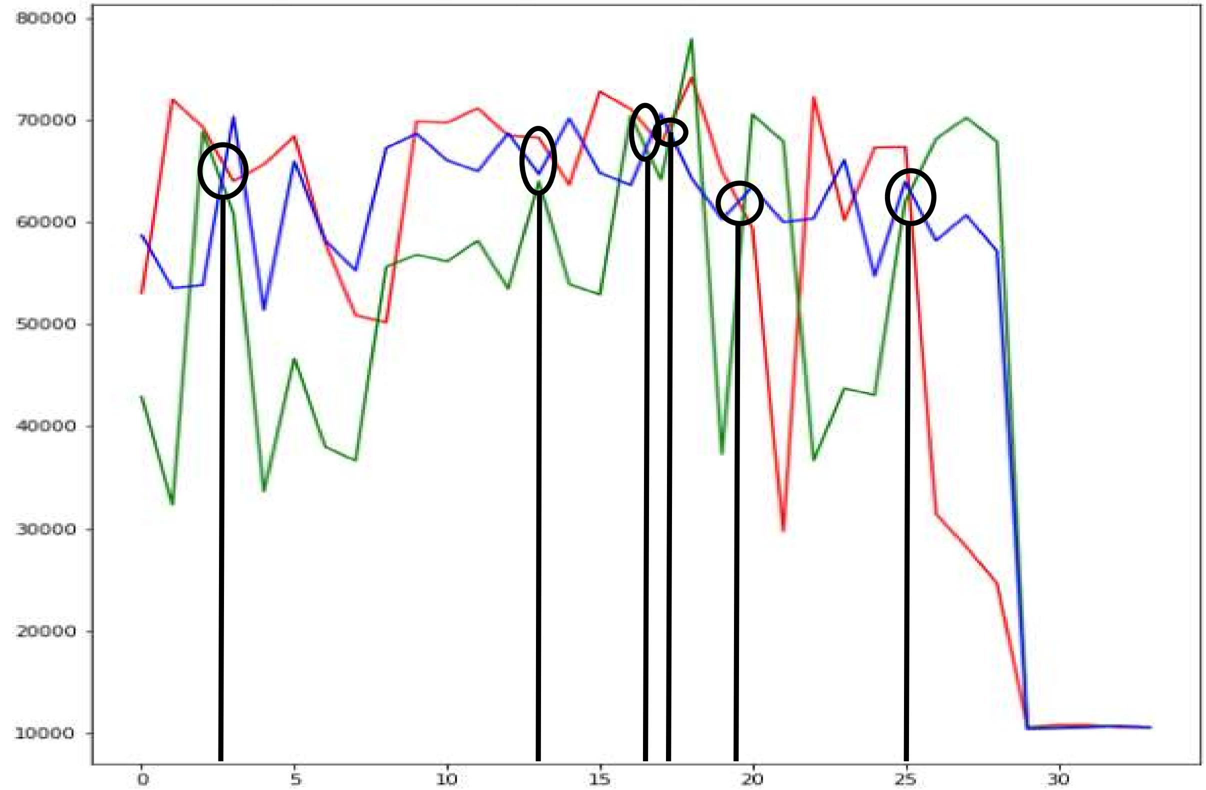
특성 제거

```
a=np.arange(0,34)
plt.figure(figsize=(10,10))
plt.plot(a,st_score,'r')
plt.plot(a,cb_score,'g')
plt.plot(a,cm_score,'b')
plt.show()
```



특성 제거

```
a=np.arange(0,34)
plt.figure(figsize=(10,10))
plt.plot(a,st_score,'r')
plt.plot(a,cb_score,'g')
plt.plot(a,cm_score,'b')
plt.show()
```

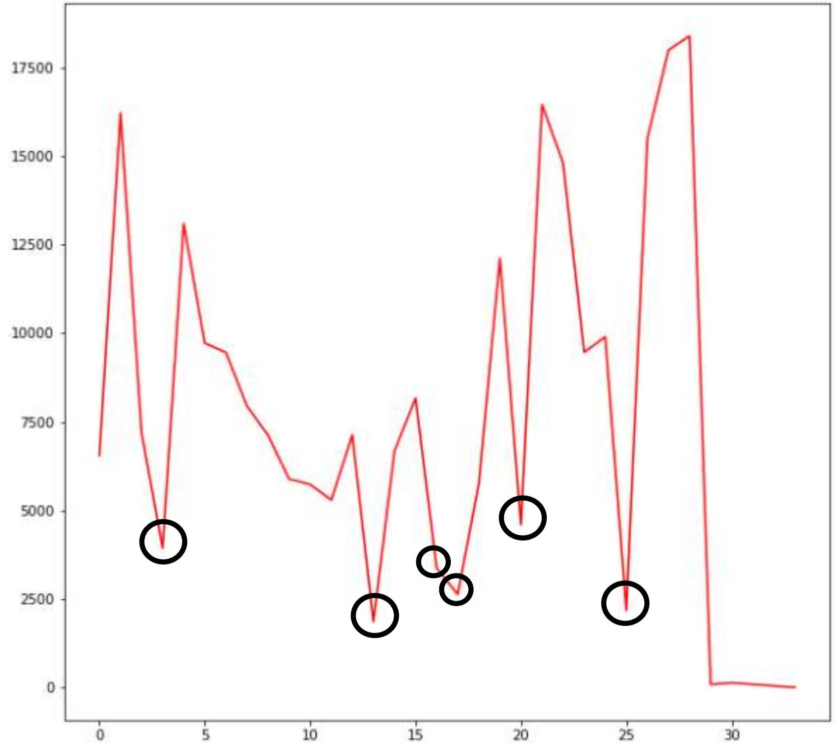


특성 제거

```
mean=(st_score+cb_score+cm_score)/3

var=((st_score-mean)**2+(cb_score-mean)**2+(cm_score-mean)**2)/3
std=var**(1/2)
std
```

```
plt.figure(figsize=(10,10))
plt.plot(a,std,'r')
plt.show()
```




```
# 분류에 필요없는 특성(스탯) 제거
data_columns = player_information.columns

del player_information[data_columns[5]]  #숏패스
del player_information[data_columns[15]] #반응
del player_information[data_columns[18]] #점핑
del player_information[data_columns[19]] #스태미나
del player_information[data_columns[21]] #멀리차기
del player_information[data_columns[27]] #평정
player_information.columns
```

```
Index(['Name', 'Position', 'Crossing', 'Finishing', 'HeadingAccuracy',
       'Volleys', 'Dribbling', 'Curve', 'FKAccuracy', 'LongPassing',
       'BallControl', 'Acceleration', 'SprintSpeed', 'Agility', 'Balance',
       'ShotPower', 'Strength', 'Aggression', 'Interceptions', 'Positioning',
       'Vision', 'Penalties', 'Marking', 'StandingTackle', 'SlidingTackle',
       'GKDiving', 'GKHandling', 'GKKicking', 'GKPositioning', 'GKReflexes'],
      dtype='object')
```

타겟 줄이고 데이터 늘리기

```python
# 현재 데이터의 포지션들과 그 개수
player_information.Position.value_counts()
```

```
ST      2152
GK      2025
CB      1778
CM      1394
LB      1322
RB      1291
RM      1124
LM      1095
CAM      958
CDM      948
RCB      662
LCB      648
LCM      395
RCM      391
LW       381
RW       370
RDM      248
LDM      243
LS       207
RS       203
RWB       87
LWB       78
CF        74
LAM       21
RAM       21
RF        16
LF        15
Name: Position, dtype: int64
```

27개의 포지션

데이터 부족

포지션 합치기

```python
'''
LF(왼쪽 공격수), RF(오른쪽 공격수), CF(중앙 공격수), LS(왼쪽 스트라이커), RS(오른쪽 스트라이커) → ST(스트라이커)

LAM(왼쪽 공격형 미드필더), RAM(오른쪽 공격형 미드필더), → CM(중앙 미드필더)
CAM(중앙 공격형 미드필더), CDM(중앙 수비형 미드필더) → CM(중앙 미드필더)
LDM(왼쪽 수비형 미드필더), RDM(오른쪽 수비형 미드필더) → CM(중앙 미드필더)
LCM(왼쪽 중앙 미드필더), RCM(오른쪽 중앙 미드필더) → CM(중앙 미드필더)

LW(왼쪽 윙어), RW(오른쪽 윙어) → WM(윙 미드필더)
LWB(좌측 윙백), RWB(우측 윙백) → WM(윙 미드필더)
LM(왼쪽 미드필더), RM(오른쪽 미드필더) → WM(윙 미드필더)

LB(왼쪽 수비수), RB(오른쪽 수비수), LCB(왼쪽 중앙 수비수), RCB(오른쪽 중앙 수비수) → CB(중앙 수비수)
'''

player_information.loc[player_information['Position']=='LF', ['Position']] = 'ST'
player_information.loc[player_information['Position']=='RF', ['Position']] = 'ST'
player_information.loc[player_information['Position']=='CF', ['Position']] = 'ST'
player_information.loc[player_information['Position']=='LS', ['Position']] = 'ST'
player_information.loc[player_information['Position']=='RS', ['Position']] = 'ST'

player_information.loc[player_information['Position']=='LAM', ['Position']] = 'CM'
player_information.loc[player_information['Position']=='RAM', ['Position']] = 'CM'
player_information.loc[player_information['Position']=='CAM', ['Position']] = 'CM'
player_information.loc[player_information['Position']=='CDM', ['Position']] = 'CM'
player_information.loc[player_information['Position']=='LDM', ['Position']] = 'CM'
player_information.loc[player_information['Position']=='RDM', ['Position']] = 'CM'
player_information.loc[player_information['Position']=='LCM', ['Position']] = 'CM'
player_information.loc[player_information['Position']=='RCM', ['Position']] = 'CM'

player_information.loc[player_information['Position']=='LW', ['Position']] = 'WM'
player_information.loc[player_information['Position']=='RW', ['Position']] = 'WM'
player_information.loc[player_information['Position']=='LWB', ['Position']] = 'WM'
player_information.loc[player_information['Position']=='RWB', ['Position']] = 'WM'
player_information.loc[player_information['Position']=='LM', ['Position']] = 'WM'
player_information.loc[player_information['Position']=='RM', ['Position']] = 'WM'

player_information.loc[player_information['Position']=='LB', ['Position']] = 'CB'
player_information.loc[player_information['Position']=='RB', ['Position']] = 'CB'
player_information.loc[player_information['Position']=='LCB', ['Position']] = 'CB'
player_information.loc[player_information['Position']=='RCB', ['Position']] = 'CB'
```

스트라이커

중앙 미드필더

윙 미드필더

중앙 수비수

타겟 줄이고 데이터 늘리기

```
# 현재 데이터의 포지션들과 그 개수
player_information.Position.value_counts()
```

```
ST      2152
GK      2025
CB      1778
CM      1394
LB      1322
RB      1291
RM      1124
LM      1095
CAM      958
CDM      948
RCB      662
LCB      648
LCM      395
RCM      391
LW       381
RW       370
RDM      248
LDM      243
LS       207
RS       203
RWB       87
LWB       78
CF        74
LAM       21
RAM       21
RF        16
LF        15
Name: Position, dtype: int64
```

27개의 포지션

포지션 합치기

```
# 변화시킨 데이터의 포지션과 그 개수
player_information.Position.value_counts()
```

```
CB      5701
CM      4619
WM      3135
ST      2667
GK      2025
Name: Position, dtype: int64
```

훈련SET, 시험SET 나누기

```python
# 데이터 나눌 때 유용한 모듈
from sklearn.model_selection import train_test_split

# 훈련데이터 80%, 시험데이터 20%로 나눔
train, test = train_test_split(player_information, test_size=0.2)
print("총 개수 : ", len(player_information))
print("훈련set 개수 : ",len(train))
print("시험set 개수 : ",len(test))
```

```
총 개수 :  18147
훈련set 개수 :  14517
시험set 개수 :  3630
```

```python
X_train = train.iloc[:,2:34]; y_train = train[['Position']].values.ravel() # 훈련데이터에 사용될 특징(스탯)과 예측 값(포지션)

X_test = test.iloc[:,2:34]; y_test = test[['Position']] # 시험데이터에 사용될 특징(스탯)과 예측 값(포지션)
```

사용할 특징(스탯) 범위

↓

이름, 포지션 값 제외

훈련(모델 선택)

RandomForest 사용

```
from sklearn.ensemble import RandomForestClassifier

forest_clf = RandomForestClassifier(n_estimators=100, random_state=42)
forest_clf.fit(X_train, y_train) #values.ravel()로 1차원 배열로 한줄로 만듦
```

```
RandomForestClassifier(random_state=42)
```

```
#훈련데이터에 대한 정확도
forest_tr=format(forest_clf.score(X_train,y_train));
forest_tr
```

```
'0.9999311152441964'
```

```
#시험데이터에 대한 정확도
forest_te=format(forest_clf.score(X_test,y_test));
forest_te
```

```
'0.8592286501377411'
```

SVM 사용

```
from sklearn.svm import SVC

svm_clf = SVC(C=10, gamma=0.0001, kernel='rbf')
svm_clf.fit(X_train, y_train)
```

```
SVC(C=10, gamma=0.0001)
```

```
#훈련데이터에 대한 정확도
svm_tr=format(svm_clf.score(X_train,y_train));
svm_tr
```

```
'0.8857890748777295'
```

```
#시험데이터에 대한 정확도
svm_te=format(svm_clf.score(X_test,y_test));
svm_te
```

```
'0.8655647382920111'
```

RandomForest < SVM

모델 세부 튜닝 - 배깅

한 모델로 데이터 중복을 허용하여 샘플링

모델 세부 튜닝 - 배깅사용

```
from sklearn.ensemble import BaggingClassifier

bag_clf = BaggingClassifier(svm_clf, n_estimators=500, max_samples=1000, bootstrap=True, n_jobs=-1)
bag_clf.fit(X_train,y_train)
```

```
BaggingClassifier(base_estimator=SVC(C=10, gamma=0.0001), max_samples=1000,
                  n_estimators=500, n_jobs=-1)
```

```
#훈련데이터에 대한 정확도
bag_tr=format(bag_clf.score(X_train,y_train));
bag_tr
```

```
'0.8640903767996142'
```

```
#시험데이터에 대한 정확도
bag_te=format(bag_clf.score(X_test,y_test));
bag_te
```

```
'0.8672176308539945'
```

큰 차이는 안남

```
# 실제값과 예측값이 어떻게 다른지 비교
comparison = pd.DataFrame({'실제 값':y_test.values.ravel(),'예측 값':bag_clf.predict(X_test)})
comparison
```

결과 보기

```
# 실제값과 예측값이 어떻게 다른지 비교
comparison = pd.DataFrame({'실제 값':y_test.values.ravel(),'예측 값':bag_clf.predict(X_test)})
comparison[:50]
```

	실제 값	예측 값
0	WM	CM
1	CB	CB
2	WM	WM
3	CB	CB
4	CB	CB
5	CM	WM
6	CM	CM
7	ST	ST
8	WM	WM
9	ST	ST
10	ST	WM
11	GK	GK
12	CM	CM
13	WM	CM
14	CM	CM
15	ST	ST
16	GK	GK
17	GK	GK
18	ST	ST
19	CB	CB
20	CB	CB
21	CM	CM
22	WM	WM
23	CB	CM
24	ST	ST
25	CM	CM
26	CB	CB
27	CB	CB
28	CM	CB
29	CB	CB
30	GK	GK
31	GK	GK
32	CB	CB
33	CM	CM
34	GK	GK
35	ST	ST
36	WM	WM
37	CM	CM
38	GK	GK
39	GK	GK
40	CM	CB
41	ST	ST
42	WM	WM
43	CM	ST
44	CM	CM
45	WM	CB
46	ST	ST
47	CB	CB
48	CB	CB
49	ST	ST