포지션 별 유망주 추천시스템

Python을 활용한 추천시스템 - CBF

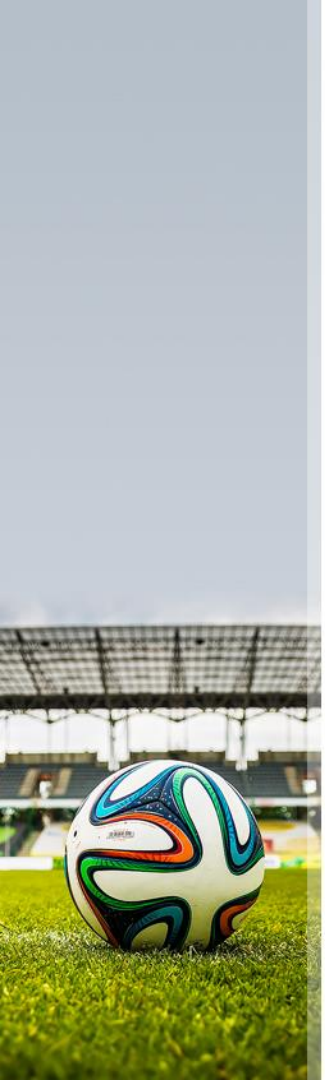
개요 및 목적

“에이전트 마인드로 포지션 별 유망주를 찾아보자!”

데이터 : Kaggle의 FIFA22 선수 데이터 18,000건

✓ 데이터 전처리
✓ 포지션 별 코사인 유사도 측정
✓ 유망주 추천 - 코사인 유사도
✓ 유망주 추천 - 가중평점

Ex) Kevin De Bruyne와 동일한 포지션 + 높은 유사도를 갖는 유망주를 추천



데이터 전처리

```
import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings('ignore')
```

```
data = pd.read_csv('fifa_22.csv', sep = '¥;')   # 구분자 ;
data = data.drop('player_id', axis = 1)
```

```
print(data.shape)
data.head(15)    # nationality에 """, '' 등 불필요한 문자 있음
```

```
(18000, 7)
```

	name	nationality	position	overall	age	potential	team
0	Lionel Messi	Argentina	ST,CF,RW	93	34	93	Paris Saint-Germain
1	Robert Lewandowski	Poland	ST	92	33	92	Bayern München
2	Kylian Mbappé	France	ST,LW	91	22	95	Paris Saint-Germain
3	Jan Oblak	Slovenia	GK	91	28	93	Atlético de Madrid
4	Kevin De Bruyne	elgium	CM,CAM	91	30	91	Manchester City
5	Neymar Jr	Brazil	CAM,LW	91	29	91	Paris Saint-Germain
6	Cristiano Ronaldo	Portugal	ST,LW	91	36	91	Manchester United
7	Harry Kane	England	ST	90	28	90	Tottenham Hotspur
8	Gianluigi Donnarumma	Italy	GK	89	22	93	Paris Saint-Germain
9	Alisson	Brazil	GK	89	28	90	Liverpool
10	Joshua Kimmich	Germany	RB,CDM	89	26	90	Bayern München
11	Ederson	Brazil	GK	89	28	91	Manchester City
12	Sadio Mané	"""Senegal"	LW	89	29	89	Liverpool
13	Virgil van Dijk	Netherlands	CB	89	30	89	Liverpool
14	Casemiro	Brazil	CDM	89	29	89	Real Madrid

```
data.nationality
```

```
0                  Argentina
1                     Poland
2                     France
3                   Slovenia
4                     elgium
                 ...
17995                 Norway
17996          """Australia"
17997       """Saudi Arabia"
17998    Republic of Ireland
17999                 Sweden
Name: nationality, Length: 18000, dtype: object
```

```
# 정규표현식으로 날림
data['nationality'] = data['nationality'].str.replace(pat=r'[^A-Za-z0-9]', repl= r' ', regex=True)
data['nationality'] = data['nationality'].str.replace(pat=r'[₩s₩s+]', repl= r' ', regex=True)
```

```
data.head(15)
```

	name	nationality	position	overall	age	potential	team
0	Lionel Messi	Argentina	ST CF RW	93	34	93	Paris Saint-Germain
1	Robert Lewandowski	Poland	ST	92	33	92	Bayern München
2	Kylian Mbappé	France	ST LW	91	22	95	Paris Saint-Germain
3	Jan Oblak	Slovenia	GK	91	28	93	Atlético de Madrid
4	Kevin De Bruyne	elgium	CM CAM	91	30	91	Manchester City
5	Neymar Jr	Brazil	CAM LW	91	29	91	Paris Saint-Germain
6	Cristiano Ronaldo	Portugal	ST LW	91	36	91	Manchester United
7	Harry Kane	England	ST	90	28	90	Tottenham Hotspur
8	Gianluigi Donnarumma	Italy	GK	89	22	93	Paris Saint-Germain
9	Alisson	Brazil	GK	89	28	90	Liverpool
10	Joshua Kimmich	Germany	RB CDM	89	26	90	Bayern München
11	Ederson	Brazil	GK	89	28	91	Manchester City
12	Sadio Mané	Senegal	LW	89	29	89	Liverpool
13	Virgil van Dijk	Netherlands	CB	89	30	89	Liverpool
14	Casemiro	Brazil	CDM	89	29	89	Real Madrid

정규표현식으로 nationality에 있는 불필요한 문자 제거

Ex) """Senegal" => Senegal

포지션 별 유사도 측정

```
from sklearn.feature_extraction.text import CountVectorizer
```

```
data['position']
```

```
0         ST,CF,RW
1               ST
2            ST,LW
3               GK
4            CM,CAM
            ...
17995           CM
17996           ST
17997           ST
17998     RB,RM,LM
17999        RB,LB
Name: position, Length: 18000, dtype: object
```

```
# CountVectorizer를 적용하기 위해 공백문자로 word 단위가 구분되는 문자열로 변환.
data['position'] = data['position'].apply(lambda x : x.replace(',',' '))
data
```

	name	nationality	position	overall	age	potential	team
0	Lionel Messi	Argentina	ST CF RW	93	34	93	Paris Saint-Germain
1	Robert Lewandowski	Poland	ST	92	33	92	Bayern München
2	Kylian Mbappé	France	ST LW	91	22	95	Paris Saint-Germain
3	Jan Oblak	Slovenia	GK	91	28	93	Atlético de Madrid
4	Kevin De Bruyne	elgium	CM CAM	91	30	91	Manchester City
...	...	...	...	...	...	...	...
17995	Ulrik Mathisen	Norway	CM	52	22	62	Lillestrøm SK
17996	Damian Tsekenis	Australia	ST	52	20	67	Central Coast Mariners
17997	Abdullah Al Radeef	Saudi Arabia	ST	52	18	70	Al Hilal
17998	Aaron McNally	Republic of Ireland	RB RM LM	52	21	66	Longford Town
17999	Lucas Forsberg	Sweden	RB LB	52	18	66	AIK

18000 rows × 7 columns

```
# CountVectorizer로 학습시켰더니 18000명 선수에 대한 279개 포지션의 '포지션 매트릭스'가 생성되었다.

count_vect = CountVectorizer(min_df=0, ngram_range=(1, 3))
position_mat = count_vect.fit_transform(data['position'])
print(position_mat.shape)
print(position_mat)
```

```
(18000, 279)
  (0, 231)      1
  (0, 43)       1
  (0, 191)      1
  (0, 240)      1
  (0, 45)       1
  (0, 242)      1
  (1, 231)      1
  (2, 231)      1
  (2, 106)      1
  (2, 259)      1
  (3, 56)       1
  (4, 47)       1
  (4, 0)        1
  (4, 48)       1
  (5, 106)      1
  (5, 0)        1
  (5, 4)        1
  (6, 231)      1
```

```
count_vect.vocabulary_
```

```
{'st': 231,
 'cf': 43,
 'rw': 191,
 'st cf': 240,
 'cf rw': 45,
 'st cf rw': 242,
 'lw': 106,
 'st lw': 259,
 'gk': 56,
 'cm': 47,
 'cam': 0,
 'cm cam': 48,
 'cam lw': 4,
 'rb': 130,
 'cdm': 26,
 'rb cdm': 139,
 'cb': 7,
 'rw lw': 192,
 'rm': 170,
```

코사인 유사도를 활용한 포지션 별 유사도 계산 및 선수 추천

```
from sklearn.metrics.pairwise import cosine_similarity

position_cos_sim = cosine_similarity(position_mat, position_mat)

print(position_cos_sim.shape)
print(position_cos_sim[:10])
```

```
(18000, 18000)
[[1.         0.40824829 0.23570226 ... 0.40824829 0.         0.        ]
 [0.40824829 1.         0.57735027 ... 1.         0.         0.        ]
 [0.23570226 0.57735027 1.         ... 0.57735027 0.         0.        ]
 ...
 [0.40824829 1.         0.57735027 ... 1.         0.         0.        ]
 [0.         0.         0.         ... 0.         0.         0.        ]
 [0.         0.         0.         ... 0.         0.         0.        ]]
```

```
# 자료를 정렬하는 것이 아니라 순서만 알고 싶다면 argsort
position_cos_sim_sorted_ind = position_cos_sim.argsort()[:, ::-1]
print(position_cos_sim_sorted_ind[:5])
```

```
[[    0  7080 12329 ... 11168 11167  8999]
 [10901 10590  1150 ... 11425 11424  8999]
 [ 6495  7446   706 ... 11243 11241  8999]
 [10431  2245 15312 ... 11480 11479     0]
 [ 1175  5849  5744 ... 11118 11117     0]]
```

함수화

```
def find_sim_player_ver1(df, sorted_ind, player_name, top_n=10):

    player_name = df[df['name'] == player_name]

    player_index = player_name.index.values
    similar_indexes = sorted_ind[player_index, :(top_n)]

    print(similar_indexes)
    similar_indexes = similar_indexes.reshape(-1)

    return df.iloc[similar_indexes]
```

```
similar_player = find_sim_player_ver1(data, position_cos_sim_sorted_ind, 'Jan Oblak', 10)
similar_player
```

```
[[10431  2245 15312 15311 15309 15308 15303 15299 15296 12590]]
```

	name	nationality	position	overall	age	potential	team
10431	Thomas Mikkelsen	Denmark	GK	65	38	65	Brøndby IF
2245	Frederik Rønnow	Denmark	GK	74	29	74	1. FC Union Berlin
15312	Adam Wilk	Poland	GK	60	23	67	Cracovia
15311	Tim Wiesner	Germany	GK	60	24	67	VfL Osnabrück
15309	Lewis Thomas	Wales	GK	60	24	66	Forest Green Rovers
15308	Miguel Vargas	Chile	GK	60	25	66	Unión La Calera
15303	Ameen Bukhari	Saudi Arabia	GK	60	24	68	Al Nassr
15299	Antoine Lejoly	elgium	GK	60	23	67	Beerschot
15296	Dean Lyness	England	GK	60	30	60	St. Mirren
12590	Koki Otani	Japan	GK	63	32	63	Hokkaido Consadole Sapporo

=> 결과는 잘 나왔지만 만족스러운 결과가 아니다. (아래 참고)
potential이 낮은 선수 위주로 나온 것 같다.
유망주라 하기엔 조금 부족한 것 같다.

```
data[data['position'] == 'GK'].sort_values(by = 'potential' , ascending=False).head(20)
```

	name	nationality	position	overall	age	potential	team
3	Jan Oblak	Slovenia	GK	91	28	93	Atlético de Madrid
8	Gianluigi Donnarumma	Italy	GK	89	22	93	Paris Saint-Germain
11	Ederson	Brazil	GK	89	28	91	Manchester City
15	Thibaut Courtois	elgium	GK	89	29	91	Real Madrid
9	Alisson	Brazil	GK	89	28	90	Liverpool
97	Mike Maignan	France	GK	84	26	89	Milan
19	Keylor Navas	Costa Rica	GK	88	34	88	Paris Saint-Germain
379	Dean Henderson	England	GK	80	24	87	Manchester United
33	Wojciech Szczesny	Poland	GK	87	31	87	Juventus
36	Hugo Lloris	France	GK	87	34	87	Tottenham Hotspur
51	Koen Casteels	elgium	GK	86	29	87	VfL Wolfsburg

가중평점(overall & potential) 반영한 선수 추천

```
C = data['potential'].mean()
m = data['overall'].quantile(0.6)

def new_rating(record):
    v = record['overall']
    R = record['potential']

    return ( (v/(v+m)) * R ) + ( (m/(m+v)) * C )
```

```
data['new_rating'] = data.apply(new_rating, axis=1)
```

```
data
```

	name	nationality	position	overall	age	potential	team	new_rating
0	Lionel Messi	Argentina	ST CF RW	93	34	93	Paris Saint-Germain	83.824787
1	Robert Lewandowski	Poland	ST	92	33	92	Bayern München	83.192442
2	Kylian Mbappé	France	ST LW	91	22	95	Paris Saint-Germain	84.854029
3	Jan Oblak	Slovenia	GK	91	28	93	Atlético de Madrid	83.709375
4	Kevin De Bruyne	elgium	CM CAM	91	30	91	Manchester City	82.564721
...	...	...	...	...	...	...	...	...
17995	Ulrik Mathisen	Norway	CM	52	22	62	Lillestrøm SK	67.256589
17996	Damian Tsekenis	Australia	ST	52	20	67	Central Coast Mariners	69.423256
17997	Abdullah Al Radeef	Saudi Arabia	ST	52	18	70	Al Hilal	70.723256
17998	Aaron McNally	Republic of Ireland	RB RM LM	52	21	66	Longford Town	68.989922
17999	Lucas Forsberg	Sweden	RB LB	52	18	66	AIK	68.989922

18000 rows × 8 columns

```
def find_sim_player_ver2(df, sorted_ind, player_name, top_n=10):
    player_name = df[df['name'] == player_name]
    player_index = player_name.index.values

    similar_indexes = sorted_ind[player_index, :(top_n*2)]
    similar_indexes = similar_indexes.reshape(-1)

    # 기준 선수 index는 제외
    similar_indexes = similar_indexes[similar_indexes != player_index]

    return df.iloc[similar_indexes].sort_values('new_rating', ascending=False)[:top_n]
```

```
similar_player = find_sim_player_ver2(data, position_cos_sim_sorted_ind, 'Kevin De Bruyne', 15)
similar_player
```

	name	nationality	position	overall	age	potential	team	new_rating
633	Florian Wirtz	Germany	CM CAM	78	18	89	Bayer 04 Leverkusen	80.745142
288	Giovani Lo Celso	Argentina	CM CAM	81	25	85	Tottenham Hotspur	78.736850
276	Lucas Paquetá	Brazil	CM CAM	81	24	85	Olympique Lyonnais	78.736850
3837	Hamed Junior Traorè	Ivory Coast	CM CAM	71	21	84	Sassuolo	77.775472
1485	Joe Willock	England	CM CAM	75	22	83	Newcastle United	77.425110
3894	Morgan Gibbs-White	England	CM CAM	71	21	81	Sheffield United	76.243098
632	Iniesta	Spain	CM CAM	79	37	79	Vissel Kobe	75.427147
621	Gylfi Sigur?sson	Iceland	CM CAM	79	32	79	Everton	75.427147
3859	Dani de Wit	Netherlands	CM CAM	71	23	77	AZ	74.199933
1175	Nicolás Filhei	Brazil	CM CAM	76	25	76	São Paulo	73.769380
3899	Joel Soñora	United States	CM CAM	71	25	76	Banfield	73.689141
5849	Josh Onomah	England	CM CAM	69	24	76	Fulham	73.655406
10082	Francesco Antonucci	elgium	CM CAM	65	22	76	Feyenoord	73.584892
9966	Julien Ponceau	France	CM CAM	65	20	75	Nîmes Olympique	73.096170
9968	Tobias Christensen	Norway	CM CAM	65	21	75	Vålerenga Fotball	73.096170

※ 가중 평점

- 완벽한 가중평점이 아님
- FIFA에서 선수의 potential을 어떤 방식으로 선정하는지 모름.
- 아마 potential에 나이가 포함되어 있는 것 같음.
- 결국 overall과 potential을 바탕으로 가중평점을 구함
- 다른 방식을 찾아볼 것.

설계 및 진행과정, 고민 과정

추천시스템 만들기 - 축구 포지션 별 유망주 추천

22.02.11

1. 데이터 수집 : Kaggle의 FIFA22 선수 데이터

2. 코사인 유사도 + CBF

3. 포지션 별 유사한 선수 추천

4. potential 기반 유망주 추천

22.02.12

1. ppt 만들기

※ 중간중간 막혔던 부분

1. 데이터 전처리
 - 읽는 과정에서 ,가 아닌 구분자 => ;로 해결
 - 나라명에 불필요한 기호 포함 => 정규표현식으로 해결

2. 가중치
 - 나이가 어리고 potential이 높은 선수를 가중치를 주고싶음.
 - age, potential, overall의 적절한 가중평균을 구할 수 있는 방안 생각해봐야함.

Thank you

코드 분석 : https://github.com/SubinKim22/project