기초빅데이터프로그래밍 : 기말 프로젝트

전공: 컴퓨터공학 학년: 3학년 학번: 20201635 이름: 전찬

**0. 목차**

1. 개요 및 구현 목표

2. 구현한 프로젝트 설명

- 데이터 crawling 과정

- 데이터 전처리 과정

- 데이터 분석 과정

- 데이터 시각화 과정

- 추가적인 기능 구현

3. 개선 사항 및 느낀 점

**1. 개요**

4차 산업 혁명에서 중요한 것은 방대한 양의 데이터를 활용해서 새로운 가치를 창출해내는 것이다. 또한 이러한 방대한 양의 빅데이터를 가장 손쉽게 얻을 수 있는 방법은, 매일매일 사용하는 인터넷을 활용하는 것이다. 현재 인터넷은 총 20억개의 웹사이트, 53억명의 사용자가 있을 정도로 세상에 있는 거의 모든 정보가 존재하며, 이러한 인터넷에서 필요한 정보를 잘 추출해내는 능력 또한 4차 산업 혁명 시대를 살아가는 데에 필요한 능력이라고 할 수 있다.

최근에 위와 같은 빅데이터 기술을 활발하게 사용하고 있는 분야가 바로 스포츠 분야이다. 야구에서는 각 선수의 신체를 모델링해서 투수, 혹은 타자에게 가장 적절한 포즈를 제시해주기도 한다. 또한 빅데이터와 인공지능을 이용한 야구 심판 또한 지속적으로 개발되고 있다. 또한 부상 위험이 많은 미식 축구에서 부상을 낮추기 위해 각 선수의 움직임 데이터를 분석해서, 부상 확률을 낮출 수 있도록 데이터를 활용하기도 한다. 축구 분야에서는 축구 한 경기를 스스로 분석하는 인공지능이 존재하며, 데이터를 통해 분석한 각 선수의 활동량, 골 결정력 등은 선수의 가치를 측정하는 데에 큰 비중을 차지하기도 한다.

따라서 이번 기말 프로젝트에서는, 위에서 이야기한 것처럼 세계에서 가장 많은 사람들이 즐기는 스포츠 중 하나인 축구 분야에서, 웹 crawling을 통해 인터넷에서 필요한 정보를 직접 추출하며, pandas, OOP 토대로 수업 시간에 배운 내용을 활용해 추출해낸 정보를 분석하는 프로젝트를 진행했다.

**2. 구현한 프로젝트 설명**

구현한 프로젝트는 총 5가지로 나눌 수 있는데, 아래와 같다.

- 데이터 crawling 과정

- 데이터 전처리 과정

- 데이터 분석 과정

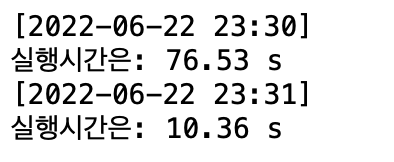
- 데이터 시각화 과정

- 추가적인 기능 구현

따라서 각각의 과정에서의 구현은 아래와 같다.

* **데이터 crawling 과정**

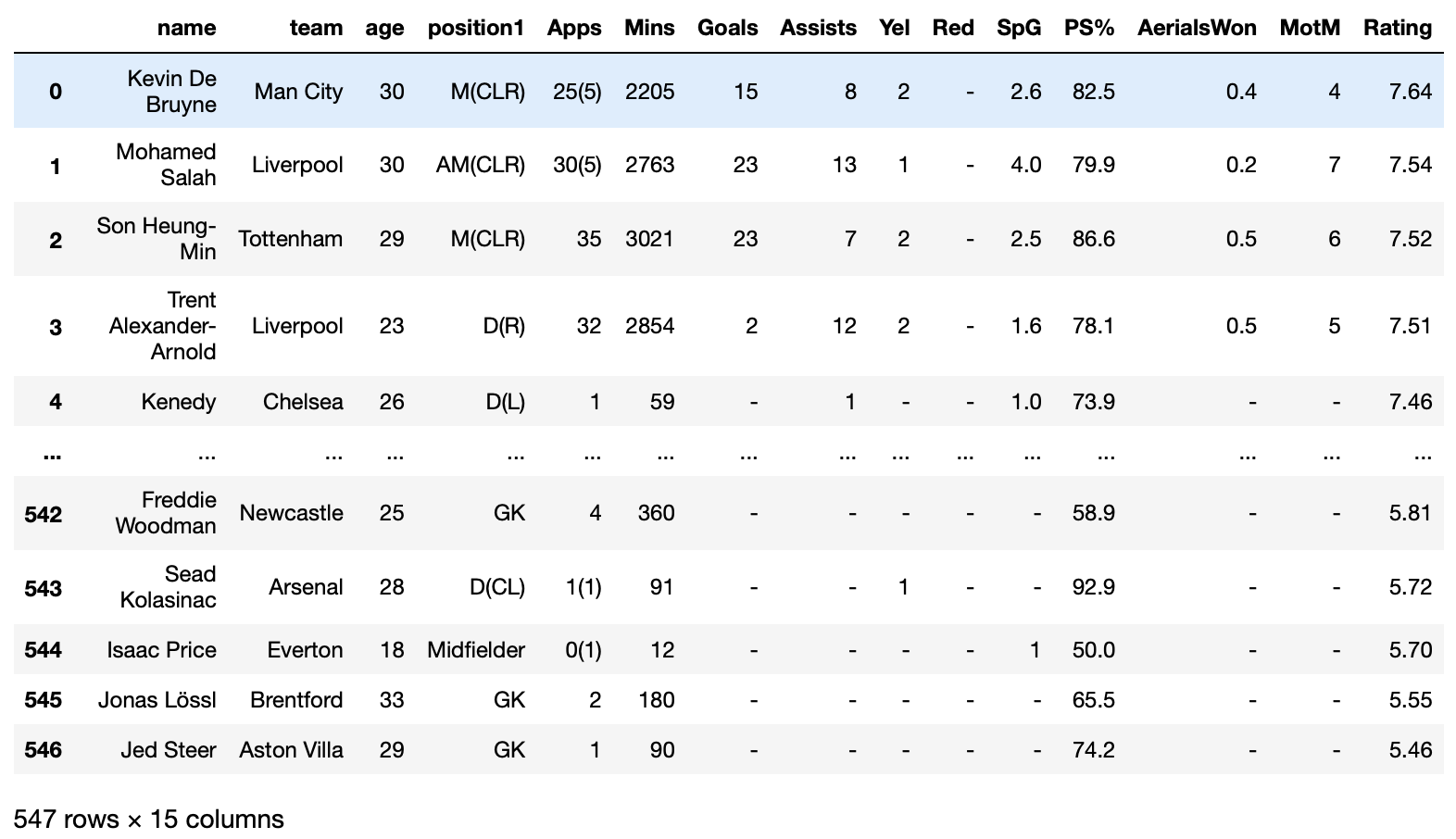
데이터 crawling 과정에서는, 필요한 데이터를 다양한 경로로 얻어내는 것을 의미한다. 그 중에서 개인적으로는 whoscored.com 이라는 사이트를 이용했다. 위 사이트는 축구 선수의 다양한 정보를 얻어낼 수 있는 축구 분석 사이트이다. 따라서 위 사이트에서 필요한 정보를 얻어내기 위해 selenium, pandas.io.html 등의 module을 사용했다. 또한 crawling 과정에서 걸리는 시간을 측정하기 위해 수업 시간에 배운 decorator을 활용하여, @checkTime 형태로 각 crawling 과정에서 걸리는 시간을 측정할 수 있도록 했다. 마지막으로 처음 crawling을 수행하고, 이후에는 crawling 과정을 거치지 않고, 데이터를 바로 사용하기 위해 import pickle 을 바탕으로, 데이터를 저장했다. 왜냐하면 selenium을 바탕으로 crawling을 수행할 때, 시간 딜레이를 일부로 만들어야 하며(서버에서 사람으로 인식할 수 있도록), 이렇게 하는 경우, 꽤나 많은 시간이 걸리기 때문이다. 따라서 pickle.dump() 로 저장하고, 이후에 pickle.load() 로 crawling한 데이터를 다시 불러오는 형식으로 코드를 작성했다. 이 과정에서의 출력 값은 아래와 같다.



<player, team 데이터 crawling에 소요되는 시간 출력>

* **데이터 전처리 과정**

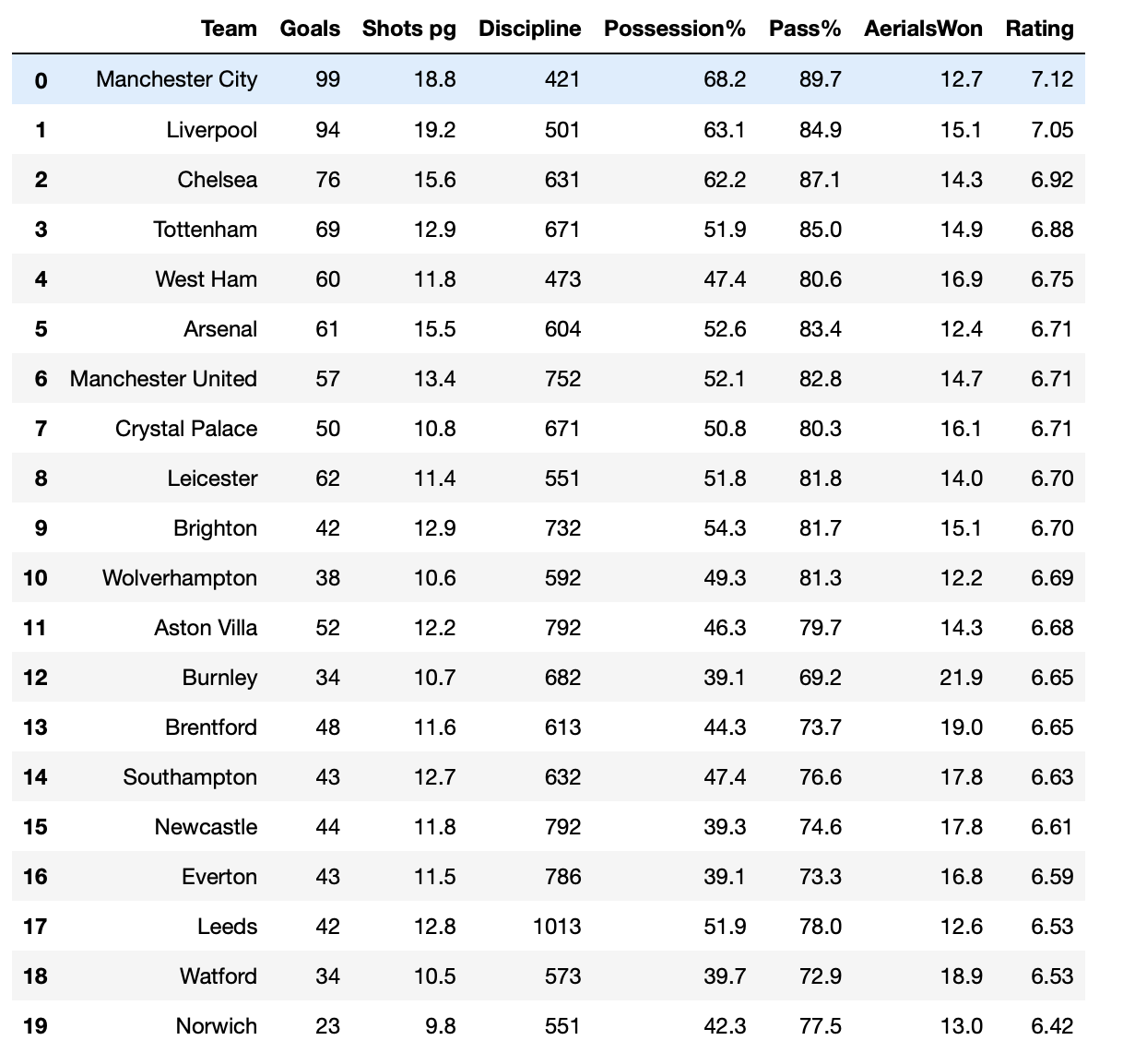
위와 같이 웹에서 crawling으로 얻어낸 데이터는 사용자가 원하는 형식이 아닐 확률이 높다. 따라서 이러한 데이터를 전처리 해서 사용자가 원하는 형식의 데이터로 가공해주어야 한다. 개인적으로 whoscored.com에서 얻어낸 데이터에 문제가 존재했는데, 중복되는 정보가 존재하며, DataFrame의 ‘name’ 부분에 선수 이름 뿐만 아니라, 팀 이름이 포함되어 있는 형태였다. 따라서 중복되는 정보를 제거하며, 선수 이름에서 팀을 분리해 주었으며, 각 column의 순서를 보기 좋은 형태로 iloc method를 통해 수정해주었다. 이의 결과는 아래와 같다.



<crawling, 전처리를 수행한 선수 DataFrame>

이를 통해 프리미어리그의 547명 선수 데이터를 포함한 선수 DataFrame을 얻어낼 수 있었다.

또한 팀 데이터에서는 팀 이름과 팀 번호가 합쳐져 있어, 이를 분리해주는 작업을 수행했다. 이렇게 얻어낸 팀 DataFrame은 아래와 같다.



< crawling, 전처리를 수행한 선수 DataFrame >

또한 위에서 얻어낸 선수 DataFrame과 팀 DataFrame을 바탕으로, 각 팀에 해당하는 팀 객체를 구현했다. 이러한 팀 객체는 다음과 같은 변수를 포함하고 있다.

- player\_index : 해당 팀의 선수 목록을 int 값으로 저장하는 list 변수이다. int값은 각 선수의 df\_player(선수 DataFrame) 의 해당하는 번호 index 값을 의미한다.

- name : 해당 팀의 이름을 저장하는 변수이다.

- goal\_percent : 해당 팀의 골 결정력을 의미하는 변수이다.

- possession\_percent : 해당 팀의 점유율을 의미하는 변수이다.

- pass\_percent : 해당 팀의 패스 성공률을 의미하는 변수이다.

- rating : 해당 팀의 평점을 의미하는 변수이다.

이러한 변수들은 외부에서 변경할 수 없도록, ‘\_\_’와 함께, @property 처리를 수행했다. 이를 통해 모든 instance 변수들에 private 처리를 해 주었다.

또한 이러한 팀 객체는 다음과 같은 method을 포함하고 있다. 아래 method 들은 데이터 분석 과정에서 사용된다.

- calculate\_avg\_rating : 팀 선수들을 바탕으로 평균 평점을 계산한다.

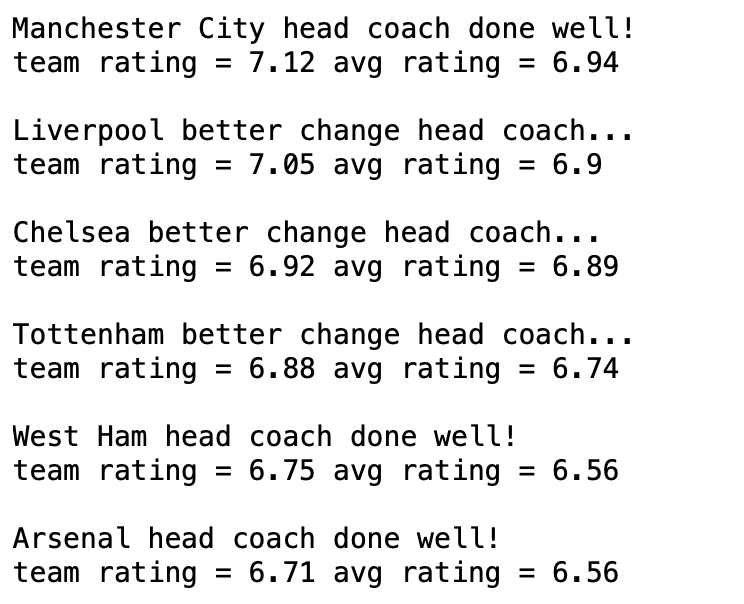
- calculate\_11\_rating : 팀 선수 상위 11명의 평균 평점을 계산한다. 축구가 11명이 하는 게임이기 때문에, 더 정확한 측정이 될 수 있다.

- evatulate\_head\_coach : 팀 평점과 팀 선수들의 평균 평점을 비교하여, 해당 팀의 감독이 감독직을 잘 수행했는지 파악하는 method 이다.

- print\_player\_list : 각 팀의 선수 명단을 출력하는 method 이다.

* **데이터 분석 과정**

따라서 위의 method 들을 바탕으로 여러 데이터 분석을 수행할 수 있다. 첫 번째 과정은 위에서 구현한 evaluate\_head\_coach() method을 바탕으로 1~20위 팀의 감독이 감독직을 잘 수행했는지 판단할 수 있다. 이의 출력 결과는 아래와 같다.



<evaluate\_head\_coach() method를 통한 각 팀의 감독 평가>

이를 통해 ‘done well!’ 문구가 존재하는 감독은 감독직을 잘 수행했으며, ‘better change head coach…’ 문구가 출력된 팀은 감독이 감독직을 잘 수행하지 못했음을 파악할 수 있다.

2~4 번째 과정은 실제 순위와 여러 측도로 분석한 순위 사이의 차이를 비교해보는 분석이다. 우선 두 번째 과정은 팀 선수들의 평균 rating을 기준으로 내림차순으로 정렬해서 순위를 예측해볼 때, 이를 실제 순위와 비교해서 정확도가 어느 정도인지 파악해보는 과정이다. 이는 위에서 정의한 calculate\_avg\_rating() method와 sorted function을 통해 구현할 수 있다. 따라서 이렇게 구현하는 경우, 아래와 같은 결과를 얻어낼 수 있다.



<팀 선수 평균을 바탕으로 예측한 결과값>

또한 세 번째 과정은 이와 비슷하지만, 위에서 정의한 calculate\_11\_rating() method을 통해 팀 선수 상위 11명의 평균 rating을 바탕으로 순위를 예측하는 형태이다. 이렇게 측정하는 이유는, 위에서 말한 것처럼 각 프로 축구 팀에는 주로 Best 11 와 같은 형태로 가장 잘하는 11명이 정해져 있기 때문이다. 따라서 이렇게 측정한 예측 결과는 아래와 같다.



<11명 상위 선수의 평균을 바탕으로 예측한 결과값>

이를 토대로, 총 선수단을 바탕으로 한 예측보다, 11명 상위 선수의 평균을 바탕으로 예측한 결과값이 더 정확한 측도가 될 수 있음을 파악할 수 있다.

마지막으로 각 객체에 저장되어 있는 rating 변수를 바탕으로 동일한 측정을 수행할 수 있는데, 이 결과는 아래와 같다.



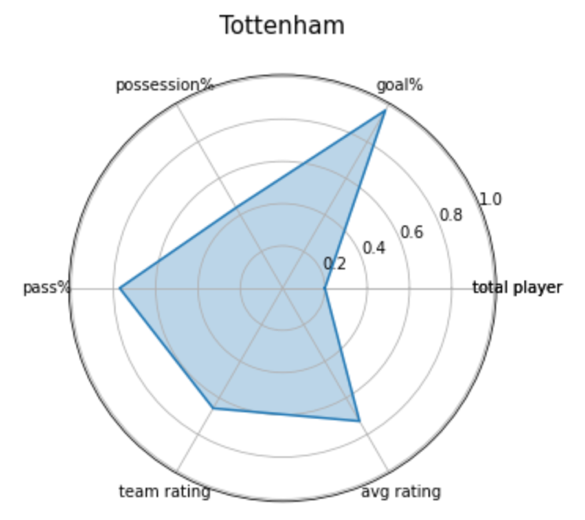
<team rating을 바탕으로 예측한 결과값>

따라서 위의 25%, 40% 와 비교했을 때 훨씬 더 정확한 값을 얻을 수 있음을 파악할 수 있다. 결과적으로는 선수들만이 중요한 것이 아닌, 감독이 각 팀의 성적(순위)에 중요한 영향을 미친다고도 분석해볼 수 있다.

* **데이터 시각화 과정**

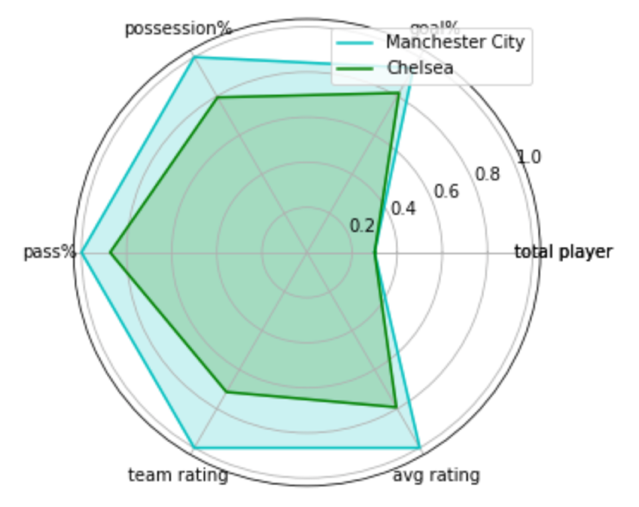
여러 게임에서, 능력치 등을 다각형 형태로 표현하곤 하며, 이는 '레이더 차트' 라는 이름으로 실생활에 다양한 영역에서 적용된다. 따라서 데이터 시각화 과정에서는, 레이더 차트를 바탕으로 해당 팀의 레이더 차트를 표현해보며, 두 팀을 입력으로 받았을 때, 레이더 차트를 출력해서 비교해볼 수 있는 시각화 과정을 구현했다.

구현을 수행할 때, 각 단위가 다른 경우가 존재하는데, 예를 들어, 평점은 0~10점 사이에 형성되어 있지만 패스 성공률은 0~100 사이에서 형성되어 있다. 따라서 이렇게 데이터를 그 자체로 레이더 차트에 표현한다면 한쪽으로 많이 쏠린 도형이 만들어질 것이다. 따라서 이를 방지하기 위해 하나의 정규화 방법이라 할 수 있는 rescaling을 각 데이터에 적용하여, 육각형에 맞게 표현해주었다. 더 나아가 rescaling 중 가장 간단한 방법인 (X - min) / (max - min) 식을 바탕으로 각 데이터가 0~1 사이의 값을 가질 수 있도록 rescaling을 적용했다. 따라서 이를 토대로 표현한 레이더 차트는 아래와 같다. 또한 이 기능을 수행하는 함수를 print\_radar\_chart() 로 정의했다.



<토트넘 핫스퍼 팀의 능력치 레이더 차트 표현>

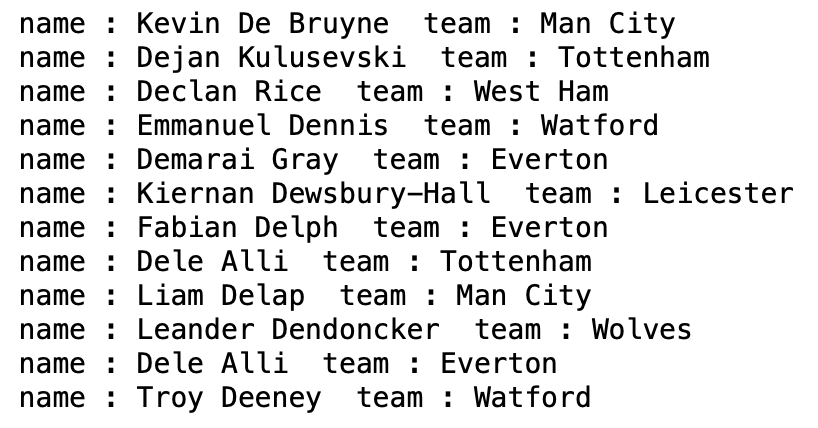
더 나아가 위 차트를 통해 두 팀을 비교할 수 있는 함수인 radar\_compare() 함수 또한 정의했다. 위 함수의 경우에는 4개의 parameter이 존재하는데, 각각 첫 번째 팀, 두 번째 팀, 첫 번째 팀의 색, 두 번째 팀의 색을 parameter로 가진다. 한 예시인 radar\_compare(1, 3, 'c', 'g') 호출의 출력 차트는 아래와 같다.



<1번째 팀인 machester city와 3번째 팀인 chelsea의 비교>

* **추가적인 기능 구현**

위와 같은 데이터 분석에서 더 나아가, 정규식을 이용한 선수 이름 검색을 수행할 수 있는 함수인 search\_by\_regex() 을 정의했다. parameter로 정규식을 받으면, 그 정규식에 대해 re.search를 수행해서 적합한 선수가 있으면 선수의 이름과 팀을 출력해주는 함수이다. 이 함수에서 이름에 ‘De’가 포함된 선수를 검색했을 때, 출력은 아래와 같다.



<’De’ 가 포함된 선수의 이름과 팀 출력>

또한 위에서 정의한 print\_player\_list() method 또한 실행해볼 수 있다. 아래는 4위인 토트넘의 선수 명단 출력이다.



<토트넘 선수 명단 출력>

**3. 개선 사항 및 느낀 점**

위와 같이 pandas DataFrame, OOP를 기본으로, regex, decorator, @property, plt, pickle 등의 수업 시간에 배운 내용을 활용해가며 축구 선수와 팀을 바탕으로 여러 분석과 시각화 과정을 수행할 수 있었다. 하지만 위 프로젝트에서 개선점이 존재하는데, 프로젝트를 시작하면서 하나의 구현 목표로 정했던, crawling한 데이터를 바탕으로 딥 러닝 과정을 거쳐 각 선수의 이적 시장 상 가격을 예측해보는 구현을 수행해보고 싶었는데, 이를 시간, 그리고 선수 몸값 데이터를 구하는 데에 한계가 존재해서 수행해보지는 못했다. 따라서 나중에 이를 구현해보고도 싶다. 추가로 선수 데이터 자체의 분석 과정도 약간은 부족하다고 생각하여, 선수 데이터로 분석하는 것도 이번 프로젝트의 한 개선 사항이라고 생각한다.

마지막으로 이번 프로젝트를 진행하며 이번 학기에 기초빅데이터프로그래밍 강의를 들으며 배웠던 파이썬 기본 내용들, 여러 module의 응용 등을 다시 한 번 공부하면서 프로젝트를 구현할 수 있어서 좋은 시간이었다. 또한 컴퓨터공학과에서는 할 수 없는, 분석 하고 싶은 주제를 바탕으로 직접 crawling을 수행하고, 분석하는 과정을 수행할 수 있어서 유익하기도 했다.