18.10.2023

Football Team Evolution

First week - Progress Report

Damian Urbański,
Jakub Mikuła,
Michał Wiśniewski

Project Overview: The goal of our project is to trace the evolution of three football teams: Real Madrid, FC Barcelona, and Atletico de Madrid, by analyzing every match they have played since 1970. We aim to create links between players who participated in winning matches to understand how the teams have evolved over the years.

Key Achievements:

1. **Data Source Discovery:** Our project began with the significant challenge of sourcing detailed match data, including player information, scores, and match dates. After thorough research, we identified the website "https://www.bdfutbol.com/" as a valuable data source for our selected teams. This website provides a comprehensive repository of match records for our teams dating back to 1970.

2. **Web Scraping Script Development:** In the first week, we successfully developed a Python web scraping script that can access the "https://www.bdfutbol.com/" website, extract relevant data, and save it in CSV files for further analysis. This script is a crucial component of our data collection process.

    1. search_for_players: This function scrapes player names from match pages on the website. It takes the URL ending, team names, and the target team for the project as parameters.

```python
def search_for_players(ending_of_url, team1, team2, team_for_project):

    url = f'https://www.bdfutbol.com/en/p/p.php?id={ending_of_url}'
    response = requests.get(url)
    players = []

    if response.status_code == 200:
        soup = BeautifulSoup(response.text, 'html.parser')
        sections = soup.find_all('table', class_='taula_estil')

        for section_number in range(0,4):
            section=sections[section_number]
            rows = section.find_all('tr')[1:]

            for row in rows:
                columns = row.find_all('td')
                player_name = columns[3].text

                if player_name!='' and team1==team_for_project:
                    if section_number==0 or section_number==2:
                        players.append(player_name)

                elif player_name!='' and team2==team_for_project:
                    if section_number==1 or section_number==3:
                        players.append(player_name)

    return players
```

2. save_to_excel: This function saves player data to an Excel file, including match details and player names. It is called for each team, and data is saved in separate CSV files.

```
def save_to_excel(name_of_team, name_of_excel, matches_list):

    worbook = Workbook()
    sheet=worbook.active
    workbook_row=1

    for list_row in matches_list:
        try:
            if (list_row[5]== name_of_team and (list_row[6]>list_row[7])) or (list_row[9]==name_of_team and (list_row[7]>list_row[6])):
                players=search_for_players(list_row[8], list_row[5], list_row[9], name_of_team)

                sheet.cell(row=workbook_row, column=1, value=list_row[0])
                sheet.cell(row=workbook_row, column=2, value=list_row[5])
                sheet.cell(row=workbook_row, column=3, value=list_row[9])
                sheet.cell(row=workbook_row, column=4, value=f'{list_row[6]}-{list_row[7]}')
                print(list_row[0])
                for col, name in enumerate(players, start=5):
                    sheet.cell(row=workbook_row, column=col, value=name)

                workbook_row+=1
        except:
            worbook.save(f'{name_of_excel}.csv')

            worbook.close()
            quit()

    worbook.save(f'{name_of_excel}.csv')
    worbook.close()
```

3. **GitHub Repository Creation:** We have created GitHub repository for our project to facilitate collaboration and version control. Our code, including the web scraping script, has been uploaded to this repository.

4. **Data Collection Initiated:** With our web scraping script in place, we initiated data collection from the chosen website. The script captures match details, player information, and match scores. The data is then structured and saved into CSV files for ease of access and analysis.