BÁO CÁO BÀI TẬP LỚN PYTHON

1 Bài 1

Viết chương trình thu thập thông tin cầu thủ Ngoại hạng Anh mùa 2023-2024

1.1 Thư viện Sử Dụng

- `requests`: Gửi yêu cầu HTTP để lấy dữ liệu từ trang web.
- `BeautifulSoup` từ `bs4`: Phân tích HTML của trang web để trích xuất dữ liệu.
- `pandas`: Tạo và xử lý dữ liệu dưới dạng DataFrame để lưu vào tệp CSV.

1.2 Gửi Yêu Cầu và Phân Tích Nội Dung Trang

```
url = 'https://fbref.com/en/comps/9/2023-2024/stats/2023-2024-Premier-League-Stats'

# Send request and get the page content
r = requests.get(url, headers={'User-Agent': 'Mozilla/5.0'})
soup = bs(r.content, 'html.parser')
```

Đoạn mã gửi yêu cầu HTTP để lấy nội dung của trang web. Sau đó, `BeautifulSoup` phân tích HTML và lấy các thành phần cần thiết.

1.3 Danh Sách Các Cột Cần Xóa

```
labels_to_delete = [
    ['Rk', 'G+A', 'PK', 'PKatt', 'npxG+xAG', 'Matches'],
    ['Rk', 'MP', 'Starts', 'Min', 'Matches'],
    [],
    ['Rk', 'Matches'],
    ['Rk', 'Matches'],
    ['Rk', 'Att', 'Matches'],
    ['Rk', 'Matches'],
    ['Rk', 'Matches'],
    ['Rk', 'Matches'],
    ['Rk', 'MP', 'Min', 'Mn/MP', 'Min%', 'Matches', '+/-', '+/-90', 'On-Off'],
    ['Rk', 'CrdY', 'CrdR', '2CrdY', 'Int', 'TklW', 'PKwon', 'PKcon', 'Matches']
]
```

Đây là danh sách chứa các cột sẽ bị loại bỏ khỏi mỗi bảng dữ liệu được lấy từ các liên kết. Điều này giúp làm gọn dữ liệu, chỉ giữ lại các thông tin cần thiết.

1.4 Thu Thập Các Liên Kết Dẫn Đến Bảng Dữ Liệu

```
data = []
for item in soup.find('p', class_='listhead').find_next('ul').find_all('li'):
    title = item.text.strip()
    link = 'https://fbref.com' + item.find('a')['href']
    data.append({'title': title, 'link': link})
```

Đoạn mã này duyệt qua các mục trong một danh sách trên trang, thu thập tiêu đề và liên kết đến các trang con chứa dữ liệu thống kê.

1.5 Hàm `get_url`

Hàm này gửi yêu cầu HTTP đến từng liên kết, sau đó tìm các bình luận HTML (`Comment`) trong trang. Những bình luận HTML này có thể chứa các bảng dữ liệu mà trang web ẩn khỏi người dùng bình thường.

```
def get_url(url):
    r = requests.get(url)
    soup = bs(r.content, 'html.parser')
    comments = soup.find_all(string=lambda text: isinstance(text, Comment))
    return comments
```

1.6 Hàm `get_info`

Hàm này phân tích nội dung của các bình luận HTML để tìm bảng (`table`) có chứa dữ liệu thống kê. Các bước trong hàm bao gồm:

- Tìm bảng có lớp `stats_table`.
- Thu thập tên cột và dữ liệu từ từng hàng.
- Chuyển dữ liệu thành một `DataFrame`.
- Tùy chọn chỉnh sửa:
  - Giữ lại mã quốc gia trong cột `Nation` (chỉ giữ lại 3 ký tự cuối).
  - Loại bỏ các cột không cần thiết dựa trên danh sách `columns_to_delete`.

```
def get_info(comments, columns_to_delete):
    table = None
    for comment in comments:
        if 'table_container' in comment:
            table = bs(comment, 'html.parser')

    if table:
        if_table = table.find('table', {'class': 'stats_table'})

        headers_list = []
        headers = if_table.find_all('tr', attrs={'class': 'thead'})

        for header in headers:
            cols = header.find_all('th')
            for col in cols:
                headers_list.append(col.text.strip())
            break

        rows = []
        for row in if_table.find('tbody').find_all('tr'):
            row_data = []
            for th in row.find_all(['th', 'td']):
                if 'data-stat' in th.attrs:
                    value = th.text.strip()
                try:
                    if ',' in value:
                        value = float(value.replace(',', '.'))*1000
                    else:
                        value = int(value) if value.isdigit() else float(value)
                except ValueError:
                    pass
                    if value == '':
                        value = 0
                row_data.append(value)

            if len(row_data) == len(headers_list) and row_data != headers_list:
                rows.append(row_data)

        df = pd.DataFrame(rows, columns=headers_list)

        if 'Nation' in df.columns:
            df['Nation'] = df['Nation'].str[-3:]

        df = df.drop(columns=[col for col in columns_to_delete if col in df.columns])

        return df
```

1.7 Xử Lý và Hợp Nhất Dữ Liệu

1.7.1 Hợp nhất dữ liệu

Ở đây chúng ta cần thu thập dữ liệu ở các bảng: Stats, Goalkeeping, Shooting, Passing, Pass Types, Goal and Shot Creation, Defensive Actions, Possession, Playing Time, Miscellaneous Stats.

```
dataframes = [get_info(get_url(data[i]['link']), labels_to_delete[i])
for i in range(len(data)) if i != 2]

df = dataframes[0]

for data_frame in dataframes[1:]:
    df = pd.merge(df, data_frame, on=['Player', 'Nation', 'Pos' , 'Squad', 'Age', '90s'
, 'Born'], how='outer')
```

1.7.2 Xử lý dữ liệu

. Sau khi thu thập đầy đủ các dữ liệu cần thiết, chúng ta xử lý theo yêu cầu của đề bài là:

- Thống kê các cầu thủ có số phút thi đâu trên 90p
- Sắp xếp thứ tự theo tên, nếu trùng thì xếp theo độ tuổi

```
df = df[df['90s'] >= 1]

df = df.sort_values(by=['Player', 'Age'], ascending=[True, False])
```

1.8 Xuất Dữ Liệu

`df.to_csv('result.csv', sep=';', index=False)`

Cuối cùng, dữ liệu được xuất ra một tệp CSV với tên `result.csv`. Mỗi cột được phân cách bằng dấu `;`, và không bao gồm chỉ số hàng (`index=False`).