Báo cáo triển khai bài tập thu thập và xử lý dữ liệu

1. Mục tiêu

Mục đích của bài tập là tự động thu thập thông tin thống kê về các cầu thủ trong giải Premier League từ trang FBRef, xử lý dữ liệu để làm sạch và loại bỏ các cột không cần thiết, và xuất kết quả cuối cùng dưới dạng tệp CSV.

2. Các thư viện sử dụng

- **requests**: Gửi yêu cầu HTTP để tải nội dung trang web.
- **BeautifulSoup (bs4)**: Phân tích cú pháp HTML và tìm kiếm dữ liệu trong trang.
- **pandas**: Tạo và xử lý dữ liệu dưới dạng bảng, dễ dàng để lưu trữ và làm sạch dữ liệu.

3. Các bước triển khai

Bước 1: Tải nội dung trang web

- Đầu tiên, URL của trang thống kê Premier League được tải xuống bằng cách sử dụng requests.get() với tiêu đề mô phỏng một trình duyệt để đảm bảo khả năng truy cập.
- Nội dung HTML trả về từ yêu cầu này được phân tích cú pháp thành đối tượng BeautifulSoup để dễ dàng thao tác.

Code:

```
# Send request and get the page content
r = requests.get(url, headers={'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.0.0 Safari/537.36'})
soup = bs(r.content, 'html.parser')
```

Bước 2: Xác định và lưu các liên kết đến bảng dữ liệu

- Từ nội dung HTML, script xác định các liên kết chứa các bảng thống kê cần thiết. Các liên kết này được tìm thấy trong danh sách ul, nằm ngay sau phần tử p có lớp listhead.
- Mỗi liên kết được lưu trữ dưới dạng một từ điển gồm tên bảng và URL, để dễ dàng xử lý trong các bước sau.

Code:

```
# Gather links
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})
```

Bước 3: Truy xuất nội dung bảng từ các bình luận HTML

- Các bảng dữ liệu được đặt trong các phần tử bình luận (HTML comments), do đó, chúng không thể truy cập trực tiếp qua các thẻ HTML thông thường. Script này xác định các phần tử bình luận chứa bảng bằng: **soup.find_all(string=lambda text: isinstance(text, Comment)).**
- Hàm **get_url** được thiết kế để trả về tất cả các bình luận từ trang URL tương ứng, giúp trích xuất bảng từ trong các bình luận.

Code:

```
# Function to fetch comments from URL
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
```

Bước 4: Trích xuất và xử lý dữ liệu trong bảng

- Hàm **get_info** đọc từng bình luận trong danh sách, xác định các bảng (nếu có) và tiến hành trích xuất dữ liệu từ tiêu đề và các hàng dữ liệu.
- Trong quá trình trích xuất, các giá trị số có dấu phẩy sẽ được chuyển đổi thành dạng số để thuận tiện cho việc xử lý, còn các giá trị trống được điền mặc định là 0.

Code:

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

Bước 5: Làm sạch dữ liệu bằng cách xóa các cột không cần thiết

- Để giữ lại những thông tin cần thiết, danh sách **labels_to_delete** được tạo để chứa các cột cần xóa ở từng bảng. Dựa vào danh sách này, hàm **get_info** sẽ loại bỏ các cột không quan trọng sau khi đọc dữ liệu từ mỗi bảng.

Code:

```
# Remove unnecessary columns
df = df.drop(columns=[col for col in columns_to_delete if col in df.columns])

return df
```

Bước 6: Gộp dữ liệu từ nhiều bảng và lọc dữ liệu có giá trị

- Các bảng đã làm sạch được gộp lại thành một bảng lớn với **pd.merge(),** dựa trên các cột chung như **Player, Nation, Pos, Squad, Age, 90s, và Born.**
- Chỉ giữ lại các cầu thủ có ít nhất 1 trận đấu bằng cách lọc trên cột **90s.**

Code:

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

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

Bước 7: Xuất dữ liệu cuối cùng ra file CSV

- Dữ liệu cuối cùng được lưu thành tệp CSV, giúp dễ dàng sử dụng trong các công cụ phân tích hoặc cho các bước xử lý tiếp theo.

Code:

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

4. Kết quả và Đánh giá

Kết quả là file results.csv chứa dữ liệu đã được làm sạch và tổ chức tốt, với thông tin tổng hợp từ nhiều bảng. Dữ liệu đã qua xử lý này có thể dễ dàng dùng trong các bài toán phân tích chuyên sâu hoặc trình bày thống kê.

5. Tổng kết

Bài tập đã áp dụng các kỹ thuật web scraping và làm sạch dữ liệu để thu thập thông tin một cách tự động từ một trang web có cấu trúc phức tạp. Thao tác này đặc biệt hữu ích khi cần thu thập và làm sạch dữ liệu từ nhiều bảng khác nhau. Việc sử dụng BeautifulSoup và Pandas đã giúp giảm thiểu thời gian xử lý và tăng tính hiệu quả trong việc xử lý dữ liệu.