BỘ THÔNG TIN VÀ TRUYỀN THÔNG

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Bài Báo Cáo

Môn: Lập Trình Python

Giảng Viên: Kim Ngọc Bách

Sinh Viên : Phạm Quốc Anh

MSV: B22DCCN040

Email : anhpham030704@gmail.com

Nhóm học phần : 11

Bài 1:

1.1 : Thư viện :

```
import requests
from bs4 import BeautifulSoup as BS
from bs4 import Comment
import pandas as pd
impo
    (function) def get_table(url: str) -> DataFrame
```

- ***requests***: Để gửi yêu cầu HTTP và lấy dữ liệu từ các trang web.
- ***BeautifulSoup*** : Để phân tích cú pháp HTML và lấy dữ liệu từ các thẻ HTML, bao gồm cả việc xử lý các bình luận (comments) trong HTML.
- ***pandas và numpy***: Dùng để quản lý và xử lý dữ liệu trong các bảng.

1.2: Hàm get_table:

```
def get_table(
    url: str
):
```

-
-
- Hàm này nhận một URL và trả về một bảng (DataFrame) lấy từ trang web.

- **Tải trang và phân tích cú pháp**:

```
):
    page = requests.get(url)
    soup = BS(page.text, 'html.parser')
```

  - Tải nội dung HTML của URL và tạo đối tượng BeautifulSoup để phân tích cú pháp HTML.
- **Lấy tất cả các comments từ HTML**:

```
comments = soup.find_all(string=lambda text: isinstance(text, Comment))
```

  - Tìm tất cả các phần tử trong HTML là bình luận (comments). Điều này hữu ích vì bảng HTML có thể nằm trong một comment.
- **Tạo BeautifulSoup từ comment chứa bảng**:

```
comment_soup = BS(comments[28], 'html.parser')
```

- Ở đây, mã code giả định rằng comment thứ 28 chứa bảng. comment_soup dùng để phân tích cú pháp comment đó.

- **Xử lý tiêu đề (headers)**:

```
thead = table.find('thead')

rows = thead.find_all('tr')
```

  - Lấy phần thead từ bảng để tìm các tiêu đề. Mỗi tiêu đề có thể có colspan (hợp nhất cột), vì vậy mã code xử lý điều này để tạo tên cột đúng.

- **Tạo danh sách các tên cột:**

```
prefixs = []
heads = rows[0].find_all('th')
for head in heads:
    if num := head.get('colspan'):
        content = head.text.replace(' ', '_')
        prefixs.append((int(num), content))

prefixs.append((1, ''))

columns = []
heads = rows[-1].find_all('th')
```

  - Dùng các tiêu đề để tạo danh sách các tên cột đầy đủ cho bảng.

- **Tạo DataFrame và xử lý dữ liệu trong bảng**:

```
df = pd.DataFrame(columns=columns[1:])
```

  -
  - Tạo một DataFrame trống với các cột đã được xác định từ phần tiêu đề.

- **Xử lý phần thân (body) của bảng**:

```
tbody = table.find('tbody')

rows = tbody.find_all('tr')
i = 0
```

  - Duyệt qua các hàng (row) trong phần tbody, lấy dữ liệu từ từng td (ô dữ liệu) và thêm vào DataFrame.

1.3:Tạo danh sách các URL và gọi get_table cho từng URL:

```
tables = []
urls = [
    'https://fbref.com/en/comps/9/2023-2024/stats/2023-2024-Premier-League-Stats',
    'http                                s/9/2023-2024/keepers/2023-2024-Premier-League-Stats',
    'http                                s/9/2023-2024/shooting/2023-2024-Premier-League-Stats',
    'https://fbref.com/en/comps/9/2023-2024/passing/2023-2024-Premier-League-Stats',
    'https://fbref.com/en/comps/9/2023-2024/passing_types/2023-2024-Premier-League-Stats',
    'https://fbref.com/en/comps/9/2023-2024/gca/2023-2024-Premier-League-Stats',
    'https://fbref.com/en/comps/9/2023-2024/defense/2023-2024-Premier-League-Stats',
    'https://fbref.com/en/comps/9/2023-2024/possession/2023-2024-Premier-League-Stats',
    'https://fbref.com/en/comps/9/2023-2024/playingtime/2023-2024-Premier-League-Stats',
    'https://fbref.com/en/comps/9/2023-2024/misc/2023-2024-Premier-League-Stats'
]

for i, url in enumerate(urls):
    print(i)
    table = get_table(url)
    tables.append(table)
```

Follow link (ctrl + click)

-
-
- Danh sách urls chứa các URL của các trang thống kê Premier League. get_table được gọi cho mỗi URL, và kết quả bảng (DataFrame) được lưu vào tables.

□ 1.4 : Định nghĩa các tên cột mong đợi cho từng bảng:

```
columns = [
    ['stats', ['Player', 'Nation', 'Pos', 'Squad', 'Age', 'Playing_Time_MP',
       'Playing_Time_Starts', 'Playing_Time_Min',
       'Performance_Ast',
       'Performance_G-PK', 'Performance_PK',
       'Performance_CrdY', 'Performance_CrdR', 'Expected_xG', 'Expected_npxG',
       'Expected_xAG','Progression_PrgC',
       'Progression_PrgP', 'Progression_PrgR', 'Per_90_Minutes_Gls',
       'Per_90_Minutes_Ast', 'Per_90_Minutes_G+A', 'Per_90_Minutes_G-PK',
       'Per_90_Minutes_G+A-PK', 'Per_90_Minutes_xG', 'Per_90_Minutes_xAG',
       'Per_90_Minutes_xG+xAG', 'Per_90_Minutes_npxG',
       'Per_90_Minutes_npxG+xAG']],
    ['keepers', ['Player', 'Nation', 'Pos', 'Squad', 'Age',
       'Performance_GA', 'Performance_GA90', 'Performance_SoTA',
       'Performance_Saves', 'Performance_Save%', 'Performance_W',
       'Performance_D', 'Performance_L', 'Performance_CS', 'Performance_CS%',
       'Penalty_Kicks_PKatt', 'Penalty_Kicks_PKA', 'Penalty_Kicks_PKsv',
       'Penalty_Kicks_PKm', 'Penalty_Kicks_Save%']],
    ['shooting', ['Player', 'Nation', 'Pos', 'Squad', 'Age',
       'Standard_Gls', 'Standard_Sh', 'Standard_SoT', 'Standard_SoT%',
       'Standard_Sh/90', 'Standard_SoT/90', 'Standard_G/Sh', 'Standard_G/SoT',
       'Standard_Dist', 'Standard_FK', 'Standard_PK', 'Standard_PKatt',
       'Expected_xG', 'Expected_npxG', 'Expected_npxG/Sh', 'Expected_G-xG',
       'Expected_np:G-xG']],
    ['passing', ['Player', 'Nation', 'Pos', 'Squad', 'Age', 'Total_Cmp',
       'Total_Att', 'Total_Cmp%', 'Total_TotDist', 'Total_PrgDist',
       'Short_Cmp', 'Short_Att', 'Short_Cmp%', 'Medium_Cmp', 'Medium_Att',
       'Medium_Cmp%', 'Long_Cmp', 'Long_Att', 'Long_Cmp%', 'Ast', 'xAG',
       'Expected_xA', 'Expected_A-xAG', 'KP', '1/3', 'PPA', 'CrsPA', 'PrgP']],
    ['passingtype', ['Player', 'Nation', 'Pos', 'Squad', 'Age',
```

-
-

- columns chứa tên các cột mong đợi cho từng bảng tương ứng với mỗi URL trong urls. Điều này giúp sắp xếp và chọn đúng cột từ các bảng đã lấy.

1.5 :Lọc cột và giữ lại các cột mong đợi trong mỗi bảng:

```
for i, table in enumerate(tables):
    new_table = table.filter(items=columns[i][1], axis=1)
    tables[i] = new_table
```

- Duyệt qua các bảng trong tables và lọc cột để chỉ giữ lại các cột mong muốn.

1.6: Gộp (merge) các bảng với nhau:

```
common_columns = ['Player', 'Nation', 'Pos', 'Squad', 'Age']

results = tables[0]

for i in range(1, len(tables)):
    results = results.merge(tables[i], how = 'outer', on = common_columns)
```

-
- results là bảng kết quả được gộp từ các bảng trong tables. Quá trình gộp diễn ra với cột chung (common_columns) để đảm bảo các bảng khớp đúng.

1.7 : Xử lý dữ liệu và chuyển đổi kiểu dữ liệu:

```
df = results.copy()
df = df.convert_dtypes()
df.info()
df['Playing_Time_Min'] = df['Playing_Time_Min'].str.replace(',', '').astype('Int64')
```

- Tạo một bản sao của results và chuyển đổi kiểu dữ liệu cho phù hợp.

1..8 : Xóa dấu phẩy và chuyển đổi cột thành số nguyên:

```
df['Playing_Time_Min'] = df['Playing_Time_Min'].str.replace(',', '').astype('Int64')
```

- Xóa dấu phẩy (nếu có) trong cột Playing_Time_Min và chuyển đổi thành kiểu số nguyên.

1.9: Chuyển đổi các cột còn lại thành dạng số:

```
for column in df.columns[4:]:
    df[column] = pd.to_numeric(df[column], errors='coerce')
```

)

- Duyệt qua các cột và chuyển chúng thành kiểu số, bỏ qua lỗi (nếu có).

□ 1.10: Lọc và sắp xếp dữ liệu:

```
df = df[df['Playing_Time_Min'] > 90]
df['firstname'] = df['Player'].str.split().str[0].astype('string')

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

- Lọc những cầu thủ có Playing_Time_Min lớn hơn 90 phút. Sắp xếp dữ liệu theo tên cầu thủ (firstname) và tuổi giảm dần.

□ 1.11: Xuất dữ liệu ra file CSV:

```
df_sorted.drop('firstname', axis=1, inplace=True)
df_sorted.to_csv('results.csv', index=False)
```

- Lưu bảng kết quả đã xử lý và sắp xếp vào file CSV results.csv.

Bài 2.1 :

Đọc file CSV có tên ***results.csv*** nằm trong bai1

```
df = pd.read_csv('../part1/results.csv')
```

Lấy danh sách các cột từ cột thứ 6 trở về (bắt đầu từ cột số 5 vì cột đầu tiên là số 0) và chuyển chúng thành danh sách. Cột này có thể là các cột chứa thống kê dữ liệu mà chúng tôi muốn tổng hợp

```
columns_to_sum = df.columns[5:].tolist()
```

Chuyển các giá trị trong các cột này sang kiểu số. Nếu có bất kỳ giá trị nào không thể chuyển được (ví dụ: ký tự chuỗi không phải số), chúng tôi sẽ được thay thế **NaN**(thiếu giá trị).

```
df[columns_to_sum] = df[columns_to_sum].apply(pd.to_numeric, errors='coerce')
```

Nhóm dữ liệu theo cột ***Squad***(tên nhóm bóng). Sau đó, tính tổng các giá trị trong các cột đã chọn (***columns_to_sum***) cho từng Đội và lưu kết quả vào ***team_stats.***

```
highest_team_stats = team_stats.idxmax()
```

Tìm kiếm giá trị lớn nhất cho từng cột thống kê. Kết quả trả về là một Series, trong đó tên mỗi số là chỉ mục và giá trị là tên tên có giá trị cao nhất cho chỉ số đó.

```
highest_team_stats.to_csv('best_squad.csv')
```

Lưu **DataFrame *highest_team_stats_df*** vào tệp CSV tên là Result2.csv. index=False chắc chắn rằng không ghi thêm chỉ mục cột vào tệp CSV.

Bài 2.2

1. **Tải dữ liệu**

```
table = pd.read_csv('../part1/results.csv')
```

- Dòng này tải dữ liệu từ file CSV (**results.csv**) ở thư mục ../bai1/ (tức là một thư mục bên trên thư mục hiện tại của mã nguồn).
- Kết quả được lưu vào biến **table** dưới dạng một DataFrame của pandas, để có thể xử lý dễ dàng.

2. **Đảm bảo thư mục đầu ra tồn tại**

- output_dir là đường dẫn đến thư mục nơi các hình ảnh biểu đồ sẽ được lưu trữ.
- os.makedirs(output_dir, exist_ok=True): Tạo thư mục nếu nó chưa tồn tại (nếu đã tồn tại thì không làm gì).

3. **Vòng lặp qua các cột và vẽ biểu đồ histogram**

```
for column in table.columns[4:]:
    data = table[column].dropna(axis=0)
```

- **table.columns[4:]:** Lấy các cột từ cột thứ 5 trở đi để làm việc (các cột đầu tiên có thể là metadata hoặc các thông tin không liên quan).
- **data = table[column].dropna(axis=0**): Lấy dữ liệu trong từng cột và loại bỏ các giá trị NaN (thiếu).

4.Tạo biểu đồ histogram cho từng cột

```
_, bins_edges, __ = plt.hist(data, edgecolor='black')
plt.ylabel('Number of players')
plt.xlabel(column.replace('_', ' '))
```

- **plt.hist(data, bins='auto', edgecolor='black'):** Vẽ biểu đồ histogram cho dữ liệu trong cột. bins='auto' tự động điều chỉnh số lượng bins (khoảng) để phù hợp với dữ liệu. edgecolor='black' tạo viền đen cho từng cột trong histogram.
  - **plt.ylabel('Number of players'**): Đặt tên cho trục y là "Number of players".
  - **plt.xlabel(column.replace('_', ' ')):** Đặt tên cho trục x theo tên của cột, thay thế dấu _ bằng khoảng trắng để dễ đọc.

6.Lưu hình ảnh và đặt tên tệp

```
    column = column.replace("/", "_divide_")
    plt.savefig(f"../../images/all_players/{column}.png")
    plt.clf()
```

 column.replace("/", "_divide_"): Thay thế ký tự / trong tên cột bằng _divide_ để tránh lỗi khi lưu tên file.
 plt.savefig(os.path.join(output_dir, f"{column_filename}.png")): Lưu biểu đồ histogram thành file ảnh .png trong thư mục đầu ra.
 plt.clf(): Xóa biểu đồ hiện tại để chuẩn bị cho biểu đồ của cột tiếp theo.

Bài 2.3

 Lặp qua từng đội:

- ```
  for team in teams:
      stats = table[table['Squad'] == team].drop(columns=['Player', 'Nation', 'Pos', 'Squad'])
  ```
  Đoạn này lặp qua từng đội trong danh sách các đội vừa tìm được.

 Lọc dữ liệu cho từng đội:

```
for team in teams:
    stats = table[table['Squad'] == team].drop(columns=['Player', 'Nation', 'Pos', 'Squad'])
```

- Lọc bảng table để lấy dữ liệu của từng đội cụ thể (team).
- Sau đó, bỏ đi các cột không cần thiết ('Player', 'Nation', 'Pos', 'Squad'), giữ lại các cột chỉ chứa số liệu thống kê.
```

 Vẽ biểu đồ cho từng cột dữ liệu:

```
for column in stats.columns:
    data = stats[column].dropna(axis=0)
```

- Dòng này lặp qua từng cột trong bảng stats (các số liệu thống kê cho từng đội).
- data = stats[column].dropna(axis=0) lấy dữ liệu của cột hiện tại, loại bỏ các giá trị bị thiếu (NaN).

 Tạo biểu đồ histogram:

```
    _, bins_edges, __ = plt.hist(data, edgecolor='black')
    plt.ylabel('Number of players')
    plt.xlabel(column.replace('_', ' '))
    plt.xticks(bins_edges, rotation=45)
```

- Tạo biểu đồ histogram cho dữ liệu của từng cột. Histogram thể hiện phân bố giá trị của các chỉ số (ví dụ: số bàn thắng, số lần sút trúng đích, v.v.).
- edgecolor='black' đặt màu viền cho các cột của histogram để dễ quan sát.

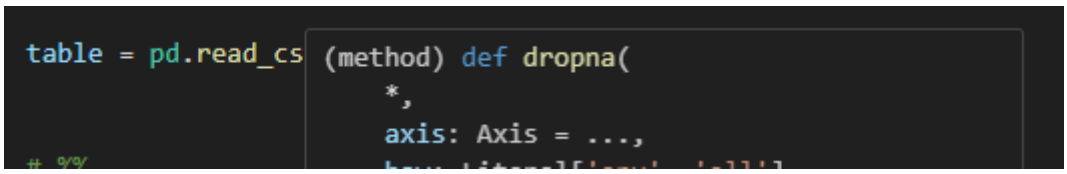
 Đặt tiêu đề và lưu biểu đồ:

```
    plt.ylabel('Number of players')
    plt.xlabel(column.replace('_', ' '))
    plt.xticks(bins_edges, rotation=45)
    column = column.replace("/", "_divide_")
    plt.savefig(f"../../images/teams/{team} {column}.png")
    plt.clf()
```

- ylabel và xlabel đặt tiêu đề cho trục tung và trục hoành.
- plt.xticks(bins_edges, rotation=45) đặt các nhãn cho trục x (giá trị của các chỉ số) và xoay chúng 45 độ.
- column.replace("/", "_divide_") thay thế ký tự / trong tên cột (nếu có) thành _divide_ để tránh lỗi khi lưu file.
- plt.savefig(f"../../images/teams/{team} {column}.png") lưu hình ảnh biểu đồ với tên file chứa tên đội và chỉ số hiện tại.
- plt.clf() xóa biểu đồ hiện tại để chuẩn bị cho biểu đồ tiếp theo.

Bài 2.4

 Đọc tệp CSV:



- Tải tệp CSV có tên results.csvvào DataFrame có tên là table. Đường dẫn cho biết tệp này nằm trong thư mục có tên là part1.

Khởi tạo trường và hàng:

```
fields = ['']
row = ['all']
```

- Khởi tạo một danh sách trống fieldsđể lưu trữ tên cột cho một DataFrame mới và một danh sách rowsẽ chứa số liệu thống kê tổng hợp.

Lặp lại qua các cột:

```
for column in table.columns[4:]:
    table.sort_values(by=column, ascending=False, inplace=True)
```

- Lặp qua các cột của DataFrame bắt đầu từ cột thứ năm (chỉ mục 4) để phân tích số liệu thống kê cho từng cột.

Sắp xếp và lọc dữ liệu:

```
datas = table.dropna(axis=0, subset=[column])
```

- Xóa các hàng có NaNgiá trị trong dòng hiện tại columnđể đảm bảo tính toán được thực hiện trên dữ liệu đầy đủ.

Hiển thị người chơi ở trên và dưới:

```
print(datas.head(3).get(['Player', 'Nation', 'Pos', 'Squad', column]))
print(datas.tail(3)[::-1].get(['Player', 'Nation', 'Pos', 'Squad', column]))
```

- In ra 3 người chơi đứng đầu (giá trị cao nhất trong cột hiện tại) và 3 người chơi đứng cuối (giá trị thấp nhất) dựa trên cột hiện tại.

Tính toán số liệu thống kê:

```
print(datas.head(3).get(['Player', 'Nation', 'Pos', 'Squad', column]))
print(datas.tail(3)[::-1].get(['Player', 'Nation', 'Pos', 'Squad', column]))

fields.append(f"Median of {column}")
fields.append(f"Mean of {column}")
fields.append(f"Std of {column}")

arr = np.array(table[column].dropna())
```

- Thêm nhãn trung vị, trung bình cộng và độ lệch chuẩn cho cột hiện tại vào fieldsdanh sách.
- Chuyển đổi các giá trị không null của cột hiện tại thành một mảng NumPy để tính toán thống kê.

Tính toán và thêm số liệu thống kê:

```
row.append(np.median(arr))
row.append(np.mean(arr))
row.append(np.std(arr))
```

- Tính toán và thêm trung vị, trung bình cộng và độ lệch chuẩn của cột hiện tại vào rowdanh sách.

Tạo một DataFrame mới:

```
df = pd.DataFrame(columns=fields)

# %%
df.loc[len(df)] = row
```

- Khởi tạo một DataFrame mới dfvới các cột được chỉ định trong fieldsdanh sách.

Thêm số liệu thống kê tổng thể:

```
df.loc[len(df)] = row

teams = table['Squad'].unique()
```

- Thêm rowsố liệu thống kê tổng thể có trong DataFrame mới df.

Nhận các đội độc đáo:

```
df.loc[len(df)] = row

teams = table['Squad'].unique()
```

- Truy xuất tên nhóm duy nhất từ cột 'Đội' của DataFrame gốc.

Chuẩn bị hàng cho mỗi đội:

```
# %%
rows = [[f"{team}"] for team in teams]
```

- Khởi tạo danh sách các hàng trong đó mỗi hàng tương ứng với một đội.

Tính toán số liệu thống kê của đội:

```
for i, team in enumerate(teams):
    stats = table[table['Squad'] == team].drop(columns=['Player', 'Nation', 'Pos', 'Squad'])
    medians = stats.median().to_list()
    means = stats.mean().to_list()
    stds = stats.std().to_list()
```

- Lặp qua từng nhóm duy nhất, lọc DataFrame để chỉ bao gồm các hàng tương ứng với nhóm hiện tại và loại bỏ các cột không liên quan.
- Tính toán trung vị, trung bình cộng và độ lệch chuẩn cho mỗi thống kê của dữ liệu đã lọc.

Điền số liệu thống kê của nhóm vào các hàng:

```
for j in range(len(medians)):
    rows[i].append(medians[j])
    rows[i].append(means[j])
    rows[i].append(stds[j])
```

- Đối với mỗi đội, thêm số liệu thống kê đã tính toán (trung vị, giá trị trung bình, độ lệch chuẩn) vào hàng tương ứng.

Thêm hàng nhóm vào DataFrame:

```
for row in rows:
    df.loc[len(df)] = row

# %%
df.to_csv('results2.csv')
```

- Thêm hàng thống kê của mỗi đội vào DataFrame mới df.

Xuất DataFrame sang CSV:


```
for row in rows:
    df.loc[len(df)] = row

# %%
df.to_csv('results2.csv')
```