BÁO CÁO BÀI TẬP LỚN MÔN LẬP TRÌNH PYTHON

GIẢNG VIÊN: KIM NGỌC BÁCH

HỌ VÀ TÊN SV: NGUYỄN HỒNG QUANG

MSV: B23DCVT361

LỚP: D23CQCE04 - B

**🧠 1. Mục tiêu bài tập lớn**

Mục tiêu của bài là xây dựng một quy trình phân tích dữ liệu cầu thủ Ngoại hạng Anh từ trang FBref và FootballTransfers:

* **Thu thập dữ liệu chi tiết về các cầu thủ** qua web scraping.
* **Xử lý và phân tích thống kê** để rút ra các đặc trưng nổi bật.
* **Áp dụng học máy (Machine Learning)** để phân cụm (clustering) cầu thủ.
* **Ghép dữ liệu định giá cầu thủ** từ nguồn khác và xuất ra file tổng hợp.

**📦 2. Tóm tắt từng phần code**

**🔹 B1.py – Thu thập & xử lý dữ liệu từ FBref**

**Chức năng:**

* Sử dụng Selenium và BeautifulSoup để scrape dữ liệu từ trang FBref của 20 đội Ngoại hạng Anh.
* Lấy 10 bảng thống kê (standard, shooting, passing, defense, etc.) và hợp nhất theo các cột chung như "Player", "Nation", "Squad", "Pos".
* Làm sạch dữ liệu: xóa cột thừa, chuẩn hóa tên quốc gia, chỉ giữ cầu thủ đá trên 90 phút.
* Kết quả được lưu thành file results.csv.

**Thư viện sử dụng:**

* Selenium, BeautifulSoup (web scraping)
* Pandas, StringIO, time

**🔹 B2.py – Phân tích thống kê và trực quan hóa dữ liệu**

**Chức năng:**

* Đọc results.csv, xác định các cột số để phân tích.
* Tìm **Top 3 cao nhất / thấp nhất** cho từng chỉ số → ghi vào top\_3.txt.
* Tính **trung vị, trung bình, độ lệch chuẩn** cho từng đội và toàn bộ giải → ghi vào results2.csv.
* Vẽ **biểu đồ histogram** cho từng chỉ số, cả toàn giải và từng đội → lưu vào thư mục histograms/.
* Xác định **cầu thủ có chỉ số cao nhất** mỗi cột → ghi vào best\_players.txt.

**Thư viện sử dụng:**

* Pandas, NumPy, Matplotlib, re, os

**🔹 B3.py – Phân cụm cầu thủ với KMeans & PCA**

**Chức năng:**

* Đọc lại results.csv, chuẩn hóa dữ liệu số.
* Dùng **Elbow method** để xác định số lượng cụm tối ưu → lưu hình elbow\_method.png.
* Áp dụng **KMeans** để chia cụm (ví dụ: 3 cụm).
* Dùng **PCA** để giảm chiều dữ liệu và trực quan hóa kết quả phân cụm trong không gian 2D → lưu hình player\_clusters.png.

**Thư viện sử dụng:**

* Pandas, NumPy, Matplotlib, Seaborn
* Sklearn (KMeans, StandardScaler, PCA)

**🔹 B4.py – Thu thập giá trị cầu thủ và ghép dữ liệu**

**Chức năng:**

* Dùng Selenium để thu thập danh sách cầu thủ có giá trị cao từ trang FootballTransfers (22 trang).
* Làm sạch tên cầu thủ (hàm normalize\_name), dùng **fuzzy matching** để ghép tên giữa hai bảng dữ liệu.
* Lọc ra những cầu thủ đã đá > 900 phút, ghép với giá trị thị trường → lưu file transfer\_player.csv.

**Thư viện sử dụng:**

* Pandas, Selenium, BeautifulSoup, RapidFuzz, re, time, Pathlib

**🛠️ 3. Các công nghệ & thư viện sử dụng chính**

| **Thư viện/Công nghệ** | **Mục đích sử dụng** |
| --- | --- |
| **Selenium** | Truy cập và tương tác với website để scrape dữ liệu |
| **BeautifulSoup** | Phân tích HTML từ trang web |
| **Pandas / NumPy** | Xử lý, phân tích và thao tác dữ liệu bảng |
| **Matplotlib / Seaborn** | Vẽ biểu đồ phân tích và phân cụm |
| **Scikit-learn** | Chuẩn hóa, phân cụm (KMeans), giảm chiều (PCA) |
| **RapidFuzz** | So khớp tên cầu thủ bằng fuzzy string matching |
| **Regex, OS, Pathlib** | Hỗ trợ xử lý chuỗi, quản lý file hệ thống |

**📌 4. Tổng kết**

Dự án của bạn là một ví dụ thực tiễn và chuyên nghiệp về cách kết hợp kỹ năng lập trình, xử lý dữ liệu và machine learning để phân tích dữ liệu thể thao. Quy trình được chia thành 4 bước rõ ràng:

1. **Thu thập dữ liệu** (Web scraping từ nhiều nguồn)
2. **Làm sạch & xử lý** (filter, chuẩn hóa, loại bỏ thiếu)
3. **Phân tích thống kê & trực quan hóa**
4. **Kết hợp học máy & nâng cao dữ liệu**

**Báo cáo chi tiết dự án phân tích dữ liệu cầu thủ Premier League 2024–2025**

**Phần 1: B1.py – Thu thập & tổng hợp dữ liệu từ FBref**

**1.1 Mục đích và chức năng**

* **Mục đích**: Tự động thu thập các bảng thống kê thi đấu (standard, shooting, passing, defense, ... tổng cộng 10 loại) của tất cả 20 đội Premier League mùa 2024–2025 từ FBref.
* **Chức năng**: Duyệt web lấy HTML, parse thành DataFrame, gộp và làm sạch trước khi xuất ra results.csv.

**1.2 Công nghệ & thư viện**

* **Selenium**: Điều khiển trình duyệt Chrome headless để tải trang động.
* **BeautifulSoup (bs4)**: Phân tích và trích xuất HTML các bảng theo id.
* **pandas**: Xử lý và gộp DataFrame.
* **os, time, re**: Tương tác hệ thống, chờ tải trang và xử lý chuỗi tên đội.

**1.3 Cấu trúc code**

1. **Khởi tạo WebDriver**
2. options = webdriver.ChromeOptions()
3. options.add\_argument('--headless')
4. driver = webdriver.Chrome(options=options)
5. **Danh sách URL**: Lưu sẵn liên kết 20 đội, ví dụ https://fbref.com/en/squads/xyz/Team-Stats
6. **Hàm parse\_table(url, table\_id)**: Truy cập URL, chờ load, BeautifulSoup tìm <table id=table\_id> rồi pd.read\_html chuyển thành DataFrame.
7. **Chuẩn hóa bảng**:
   * Thêm cột Squad (tên đội) từ URL.
   * Loại bỏ cột Rk (rank) thừa.
8. **Gộp các bảng phụ vào bảng stats\_standard**:
9. df = dfs['stats\_standard']
10. for key in ['stats\_shooting', ..., 'stats\_gca']:
11. df = df.merge(dfs[key], on=['Player','Nation','Squad','Pos'], how='left')
12. **Lọc và làm sạch**:
    * Chuyển cột Min sang int; giữ Min >= 90.
    * Chuẩn hóa Nation: lấy 2 ký tự cuối.
    * df.fillna('N/A').
13. **Xuất file**: df.to\_csv('results.csv', index=False).

**1.4 Hướng dẫn sử dụng**

1. Cài đặt môi trường: pip install selenium bs4 pandas.
2. Tải ChromeDriver tương ứng với Chrome.
3. Chạy: python B1.py; file results.csv sẽ sinh ra trong thư mục hiện tại.

**Phần 2: B2.py – Phân tích thống kê mô tả & trực quan hóa**

**2.1 Mục đích và chức năng**

* **Mục đích**: Từ results.csv, tính toán và xuất báo cáo mô tả, top cầu thủ, và vẽ histogram phân bố.
* **Chức năng**: Thống kê mean/median/std, tìm top 3, cầu thủ giỏi nhất, lưu text và hình ảnh.

**2.2 Công nghệ & thư viện**

* **pandas, numpy**: Xử lý dữ liệu, tính toán số liệu.
* **matplotlib**: Vẽ biểu đồ histogram.
* **os, re**: Tự động tạo thư mục và xử lý tên file.

**2.3 Cấu trúc code**

1. **Đọc dữ liệu**: df = pd.read\_csv('results.csv').
2. **Xác định cột số**: lọc df.select\_dtypes(include=[np.number]).columns.
3. **Top 3 cao/thấp**:
4. for col in numeric\_cols:
5. top3 = df.nlargest(3, col)[['Player', col]]
6. bottom3 = df.nsmallest(3, col)[['Player', col]]
7. # ghi top\_3.txt
8. **Thống kê mô tả**:
9. stats\_all = df[numeric\_cols].agg(['mean','median','std']).T
10. stats\_by\_team = df.groupby('Squad')[numeric\_cols].agg(['mean','median','std'])
11. stats\_all.to\_csv('results2.csv')
12. **Histogram**:
13. os.makedirs('histograms', exist\_ok=True)
14. for col in numeric\_cols:
15. plt.figure()
16. df[col].hist(bins=20)
17. plt.title(col)
18. plt.savefig(f'histograms/{col}.png')
19. plt.close()
20. **Cầu thủ xuất sắc**:
21. best = df.loc[df[numeric\_cols].idxmax()]
22. # ghi best\_players.txt

**2.4 Hướng dẫn sử dụng**

1. Đảm bảo đã cài: pip install pandas numpy matplotlib.
2. Có sẵn results.csv từ B1.
3. Chạy: python B2.py; sinh ra thư mục histograms/, top\_3.txt, results2.csv, best\_players.txt.

**Phần 3: B3.py – Phân cụm cầu thủ với K-Means & PCA**

**3.1 Mục đích và chức năng**

* **Mục đích**: Phân nhóm cầu thủ dựa trên chỉ số thi đấu, trực quan hóa với PCA.
* **Chức năng**: Chuẩn hóa, chọn số cụm tối ưu (Elbow), chạy K-Means, PCA 2D.

**3.2 Công nghệ & thư viện**

* **scikit-learn**: StandardScaler, KMeans, PCA
* **pandas, numpy**: Xử lý dữ liệu.
* **matplotlib, seaborn**: Vẽ đồ thị Elbow và phân cụm.

**3.3 Cấu trúc code**

1. **Đọc**: df = pd.read\_csv('results.csv') và lọc numeric\_cols.
2. **Impute**: SimpleImputer(strategy='mean') cho NaN.
3. **Chuẩn hóa**:
4. scaler = StandardScaler()
5. X\_scaled = scaler.fit\_transform(df[numeric\_cols])
6. **Elbow Method**:
7. inertias = []
8. for k in range(1,11):
9. km = KMeans(n\_clusters=k).fit(X\_scaled)
10. inertias.append(km.inertia\_)
11. plt.plot(range(1,11), inertias, '-o')
12. plt.savefig('elbow\_method.png')
13. **K-Means & gán nhãn** (giả sử k=3):
14. km = KMeans(n\_clusters=3).fit(X\_scaled)
15. df['Cluster'] = km.labels\_
16. **PCA 2D**:
17. pca = PCA(n\_components=2)
18. pcs = pca.fit\_transform(X\_scaled)
19. df['PCA1'], df['PCA2'] = pcs[:,0], pcs[:,1]
20. sns.scatterplot(data=df, x='PCA1', y='PCA2', hue='Cluster')
21. plt.savefig('player\_clusters.png')
22. **Xuất**: df.to\_csv('clustered\_players.csv', index=False) (nếu cần).

**3.4 Hướng dẫn sử dụng**

1. Cài: pip install scikit-learn seaborn matplotlib pandas.
2. Chạy: python B3.py; nhận elbow\_method.png, player\_clusters.png và DataFrame có cột Cluster, PCA1,PCA2.

**Phần 4: B4.py – Ghép giá trị thị trường từ FootballTransfers**

**4.1 Mục đích và chức năng**

* **Mục đích**: Lấy giá trị thị trường của cầu thủ và gán vào dữ liệu thống kê (Min > 900).
* **Chức năng**: Crawl trang FootballTransfers, fuzzy match tên cầu thủ, xuất transfer\_player.csv.

**4.2 Công nghệ & thư viện**

* **Selenium + BeautifulSoup**: Lấy HTML trang động.
* **rapidfuzz**: So khớp chuỗi tên cầu thủ.
* **pandas, re**: Xử lý DataFrame và chuẩn hóa tên.

**4.3 Cấu trúc code**

1. **Load & lọc**:
2. df\_stats = pd.read\_csv('results.csv')
3. df\_stats = df\_stats[df\_stats['Min'] > 900]
4. **Crawl dữ liệu MarketValue**:
5. base\_url = 'https://footballtransfers.com/en/players?filter%5Bleague\_id%5D=1&page='
6. for page in range(1,23):
7. driver.get(base\_url + str(page))
8. soup = BeautifulSoup(driver.page\_source, 'html.parser')
9. # parse từng thẻ player-card
10. **Bảng market\_df**: chứa Player, MarketValue, Club, Nation.
11. **Chuẩn hóa tên**:
12. def normalize(name):
13. name = re.sub(r"[^a-zA-Z ]", "", name)
14. parts = name.split()[:2]
15. return ' '.join(parts)
16. **Fuzzy matching**:
17. for idx, row in df\_stats.iterrows():
18. match, score = process.extractOne(normalize(row['Player']), market\_df['clean\_name'])
19. if score >= 85:
20. df\_stats.at[idx,'MarketValue'] = market\_df.loc[market\_df['clean\_name']==match,'MarketValue'].values[0]
21. **Xuất CSV**: df\_stats[['Player','Nation','Min','MarketValue']].to\_csv('transfer\_player.csv', index=False).

**4.4 Hướng dẫn sử dụng**

1. Cài: pip install selenium bs4 rapidfuzz pandas.
2. Chạy: python B4.py; sinh file transfer\_player.csv với các cột chính.

**Kết luận chung**: Mỗi script đảm nhận một bước trong pipeline: từ thu thập thô, tiền xử lý, phân tích thống kê, phân cụm đến tích hợp giá trị thị trường. Bằng cách chạy tuần tự B1 → B2 → B3 → B4, bạn có được góc nhìn đầy đủ về hiệu suất và giá trị của cầu thủ Premier League mùa 2024–2025.