HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA CÔNG NGHỆ THÔNG TIN 1

BÁO CÁO BÀI TẬP LỚN

Giảng viên hướng dẫn : Kim Ngọc Bách

Lớp : D23CQCE06-B

Họ Tên : Vũ Gia Khánh

MSV : B23DCCE054

Hà Nội – 2025

LỜI PHÊ CỦA GIẢNG VIÊN HƯỚNG DẪN

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

I.Problem 1

1. Giới thiệu tổng quan

Thực hiện thu thập dữ liệu thống kê cầu thủ từ trang web FBref.com với mục tiêu chính:

- Tích hợp dữ liệu từ 8 loại thống kê khác nhau
- Xử lý và làm sạch dữ liệu
- Tạo dataset hoàn chỉnh cho phân tích sau này

2. Công nghệ sử dụng

Thư viện chính:

- **Selenium:** Tự động hóa trình duyệt để load trang web động
- **BeautifulSoup:** Phân tích cấu trúc HTML
- **Pandas:** Xử lý và hợp nhất dữ liệu dạng bảng

Driver:

- **ChromeDriverManager:** Tự động quản lý phiên bản ChromeDriver

3. Cấu trúc chương trình

3.1. Khởi tạo WebDriver

```
driver = webdriver.Chrome(service=Service(ChromeDriverManager().install()))
```

- Tự động tải và cấu hình ChromeDriver phù hợp
- Tạo instance trình duyệt ảo

3.2. Danh sách nguồn dữ liệu

```
        table_links = {
    'Standard Stats': ('https://fbref.com/en/comps/9/stats/Premier-League-
Stats', 'stats_standard'),
    'Shooting': ('https://fbref.com/en/comps/9/shooting/Premier-League-Stats',
'stats_shooting'),
    'Passing': ('https://fbref.com/en/comps/9/passing/Premier-League-Stats',
'stats_passing'),
    'Goal and Shot Creation': ('https://fbref.com/en/comps/9/gca/Premier-
League-Stats', 'stats_gca'),
    'Defense': ('https://fbref.com/en/comps/9/defense/Premier-League-Stats',
'stats_defense'),
    'Possession': ('https://fbref.com/en/comps/9/possession/Premier-League-
Stats', 'stats_possession'),
    'Miscellaneous': ('https://fbref.com/en/comps/9/misc/Premier-League-
Stats', 'stats_misc'),
    'Goalkeeping': ('https://fbref.com/en/comps/9/keepers/Premier-League-
Stats', 'stats_keeper')
}
```

- 8 loại thống kê với URL và ID bảng tương ứng
- Bao phủ đầy đủ các khía cạnh: phòng ngự, tấn công, thủ môn,...

3.3. Quy trình chính

Khởi tạo driver -> Lặp qua từng bảng -> Load trang web -> Chờ bảng tải -> Parse HTML -> Xử lý cột -> Lọc dữ liệu -> Hợp nhất dữ liệu -> Lọc cầu thủ >90p -> Xử lý thủ môn -> Xuất CSV

```
        graph TD
    A[Khởi tạo driver] --> B{Lặp qua từng bảng}
    B --> C[Load trang web]
    C --> D[Chờ bảng tải]
    D --> E[Parse HTML]
    E --> F[Xử lý cột]
    F --> G[Lọc dữ liệu]
    G --> H[Hợp nhất dữ liệu]
    H --> I[Lọc cầu thủ >90p]
    I --> J[Xử lý thủ môn]
    J --> K[Xuất CSV]
```

4. Xử lý dữ liệu chính

4.1. Xử lý MultiIndex

```
        if isinstance(df.columns, pd.MultiIndex):
    new_cols = []
    for col in df.columns:
        group = col[0].strip() if...
        subgroup = col[1].strip() if...
        col_name = f"{group} {subgroup}"...
    df.columns = new_cols
```

- Gộp các header lồng nhau thành tên cột duy nhất

  Ví dụ: ('Performance', 'Gls') → 'Performance Gls'

4.2. Lựa chọn thuộc tính

Mỗi bảng chọn các chỉ số đặc trưng:

```
# Ví dụ với Passing
required_cols = ['Player', 'Total Cmp', 'Total Cmp%',...]
```

4.3. Hợp nhất dữ liệu

- Sử dụng merge trên cột 'Player'
- Xử lý riêng cho thủ môn:

```
goalkeeping_df['Pos'] = goalkeeping_df['Player'].map(...)
```

4.4. Xử lý ngoại lệ

- Kiểm tra sự tồn tại của bảng bằng try/except
- Phục hồi dữ liệu khi lỗi lưu file:

```
except Exception as e:
    output_file = 'results_backup.csv'
```

4.5. Lọc dữ liệu

- Chỉ giữ cầu thủ có >90 phút thi đấu:

```
merged_df[min_col] = merged_df[min_col].str.replace(',', '').astype(float)
merged_df = merged_df[merged_df[min_col] > 90]
```

II. Problem 2

1. Giới thiệu tổng quan

Thực hiện phân tích thống kê chi tiết về hiệu suất cầu thủ và đội bóng từ dữ liệu đã được thu thập. Mục tiêu chính:

- Xác định cầu thủ xuất sắc nhất/theo từng chỉ số
- Phân tích hiệu suất đội bóng
- Trực quan hóa dữ liệu qua biểu đồ
- Đưa ra kết luận tổng quan về mùa giải

2. Công nghệ sử dụng

Thư viện chính:

- **Pandas**: Xử lý dữ liệu dạng bảng
- **Matplotlib**: Vẽ biểu đồ histogram
- **NumPy**: Hỗ trợ tính toán số học
- **Re**: Xử lý tên file

Dữ liệu đầu vào:

- File CSV chứa thống kê cầu thủ (results.csv)

3. Cấu trúc chương trình

3.1. Luồng xử lý chính

Đọc file CSV -> Chuyển đổi dữ liệu số -> Lọc cột số -> Ghi top/bottom cầu thủ -> Tính toán thống kê đội -> Tạo histogram -> Xuất kết quả

```
graph TD
    A[Đọc file CSV] --> B[Chuyển đổi dữ liệu số]
    B --> C[Lọc cột số]
    C --> D[Ghi top/bottom cầu thủ]
    D --> E[Tính toán thống kê đội]
    E --> F[Tạo histogram]
    F --> G[Xuất kết quả]
```

3.2. Cấu trúc hàm chính

1. **read_data**: Đọc dữ liệu từ CSV
2. **convert_to_numeric**: Xử lý cột số (%)
3. **filter_numeric_columns**: Lọc cột tấn công/phòng ngự
4. **write_top_bottom_players**: Phân hạng cầu thủ
5. **calculate_team_statistics**: Thống kê đội
6. **create_histograms**: Trực quan hóa phân bố dữ liệu

4. Xử lý dữ liệu chính

4.1. Chuyển đổi dữ liệu

```
def convert_to_numeric(df):
    # Xử lý cột chứa %
    if df[col].str.contains('%').any():
        df[col] = df[col].str.rstrip('%').astype(float)
    # Chuyển các cột số khác
    else:
        df[col] = pd.to_numeric(df[col], errors='coerce')
```

•Xử lý đặc biệt cho các cột phần tram

•Chuyển đổi an toàn sang số

4.2. Lọc chỉ số quan trọng

```
ATTACKING_COLS = ['Standard SoT/90', ...]
DEFENSIVE_COLS = ['Tackles Tkl', ...]
```

- Tập trung vào 3 chỉ số tấn công và 3 phòng ngự
- Lọc các cột hợp lệ có trong dataset

4.3. Phân hạng cầu thủ

•Ghi ra file text top 3 và bottom 3 cho từng chỉ số

•Định dạng rõ ràng:

```
Statistic: Standard SoT/90
==========================
Top 3 players:
Player          Standard SoT/90
M. Salah                   2.8
...
```

4.4. Thống kê đội bóng

Tính toán cho từng đội:

- Trung vị (Median)
- Giá trị trung bình (Mean)
- Độ lệch chuẩn (StdDev)

```
group_df[col].median()
group_df[col].mean()
group_df[col].std()
```

4.5. Trực quan hóa dữ liệu

•Tạo thư mục riêng cho từng chỉ số

•2 loại histogram:

- Toàn bộ cầu thủ
- Theo từng đội

•Xử lý tên file an toàn:

```
safe_col_name = re.sub(r'[^\w\s-]', '_', col)
```

III. Problem 3

1. Tổng quan

Thực hiện phân nhóm cầu thủ Premier League dựa trên hiệu suất thi đấu bằng kỹ thuật Machine Learning. Mục tiêu chính:

- Phát hiện các nhóm cầu thủ có đặc điểm tương đồng
- Trực quan hóa mối quan hệ giữa các chỉ số
- Cung cấp insights cho chiến lược đội bóng

2. Công nghệ chính

- **Scikit-learn**: K-Means, PCA, StandardScaler
- **Matplotlib/Seaborn**: Visualize dữ liệu
- **Pandas**: Xử lý dữ liệu

3. Luồng xử lý và xử lý dữ liệu

Đọc CSV -> Chuẩn hóa dữ liệu -> Xác định số cụm -> Phân cụm K-Means -> Thống kê cụm -> Visualize PCA -> Giải thích kết quả

```
graph TD
    A[Đọc CSV] --> B[Chuẩn hóa dữ liệu]
    B --> C[Xác định số cụm]
    C --> D[Phân cụm K-Means]
    D --> E[Thống kê cụm]
    E --> F[Visualize PCA]
    F --> G[Giải thích kết quả]
```

3.1. Tiền xử lý dữ liệu

- Chuẩn hóa về cùng tỉ lệ (Z-score)
- Tự động phát hiện cột số

3.2. Lựa chọn số cụm

- Elbow Method với range k=1-10
- Xác định điểm gập khuỷu tại k=3

3.3. Phân tích PCA

- Giảm 20+ chỉ số về 2 thành phần
- Giải thích 65-80% phương sai dữ liệu

IV. Problem 4

1. Tổng quan

Xây dựng hệ thống dự đoán giá trị chuyển nhượng cầu thủ bóng đá sử dụng kết hợp web scraping và machine learning. Mục tiêu chính:

- Thu thập dữ liệu giá trị từ các trang web uy tín

- Xây dựng mô hình dự đoán dựa trên hiệu suất thi đấu

- Đánh giá độ chính xác của mô hình

2. Công nghệ chính

- **Selenium**: Tự động hóa thu thập dữ liệu từ Transfermarkt và FootballTransfers
- **Random Forest**: Mô hình hồi quy dự đoán giá trị
- **Pandas**: Xử lý dữ liệu phức tạp
- **Scikit-learn**: Chuẩn hóa dữ liệu và đánh giá mô hình

3. Luồng xử lý chính

Đọc dữ liệu cầu thủ -> Lọc cầu thủ >900 phút -> Scrape giá trị chuyển nhượng -> Làm sạch dữ liệu -> Xây dựng mô hình -> Dự đoán và đánh giá

```
graph TD
    A[Khởi tạo driver] --> B{Lặp qua từng bảng}
    B --> C[Load trang web]
    C --> D[Chờ bảng tải]
    D --> E[Parse HTML]
    E --> F[Xử lý cột]
    F --> G[Lọc dữ liệu]
    G --> H[Hợp nhất dữ liệu]
    H --> I[Lọc cầu thủ >90p]
    I --> J[Xử lý thủ môn]
    J --> K[Xuất CSV]
```

4. Đặc điểm nổi bật

4.1. Thu thập dữ liệu động

- Tích hợp 2 nguồn dữ liệu:

- Transfermarkt (Ưu tiên)
- FootballTransfers (Dự phòng)

- Xử lý chống block với headless browser
- Cơ chế retry (3 lần thử)

4.2. Tiền xử lý dữ liệu

- Chuẩn hóa định dạng tiền tệ:

```
'€10.5m' → 10,500,000
'£75k' → 75,000
```

- Lọc các cột số quan trọng:

```
FEATURES = ['Age', 'Min', 'Gls', ..., 'Won%']
```

4.3. Mô hình machine learning

- Random Forest Regressor với 100 cây
- Chuẩn hóa dữ liệu bằng StandardScaler
- Đánh giá qua MSE và R^2 score

5. File đầu ra

1. **transfer_values.csv**: Giá trị thực tế
2. **transfer_predictions.csv**: Dự đoán + giá trị thực
3. Biểu đồ so sánh dự đoán vs thực tế