BỘ THÔNG TIN VÀ TRUYỀN THÔNG
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

BÀI TẬP LỚN

MÔN:	**Lập trình Python**
Giảng viên:	**Kim Ngọc Bách**
Sinh viên thực hiện:	**Nguyễn Hải Nam**
Lớp:	**D23CQCE06-B**

Hà Nội, Tháng 5/2025

Mục lục

1 Phân tích luồng chạy mã nguồn thu thập dữ liệu thống kê bóng đá trong bài 1

1.1 Giới thiệu

Các phần sau mô tả luồng chạy của mã nguồn dùng để thu thập và xử lý dữ liệu thống kê chi tiếtcủa cầu thủ bóng đá từ trang FBref.com, bao gồm việc tự động truy cập, trích xuất, làm sạch và tổng hợp các bảng thống kê thành một tập dữ liệu hoàn chỉnh phục vụ cho các phân tích tiếp theo

1.2 Cấu trúc mã và các thư viện phụ thuộc

Mã nguồn bắt đầu bằng việc nhập các thư viện Python cần thiết, mỗi thư viện đảm nhiệm một vai trò cụ thể trong quy trình thu thập và xử lý dữ liệu:

- `time`: Quản lý thời gian chờ để đảm bảo trang web tải hoàn toàn.
- `pandas`: Xử lý và lưu trữ dữ liệu dưới dạng bảng (DataFrame).
- `BeautifulSoup`, `Comment`: Phân tích nội dung HTML để trích xuất dữ liệu.
- `selenium.webdriver`: Tự động hóa tương tác trình duyệt để thu thập nội dung web động.
- `selenium.webdriver.chrome.options`, `Service`, `webdriver_manager.chrome`: Cấu hình và quản lý ChromeDriver cho Selenium.
- `io.StringIO`: Xử lý chuỗi HTML như một tệp để Pandas đọc.
- `os`: Thao tác với hệ thống tệp, như tạo thư mục và đường dẫn.

1.3 Phân tích luồng thực thi

Quá trình thực thi mã nguồn có thể được chia thành một số giai đoạn chính, mỗi giai đoạn đóng góp vào mục tiêu tổng thể là thu thập và xử lý thống kê bóng đá.

1.3.1 Khởi tạo và cấu hình

Mã bắt đầu bằng việc xác định thư mục gốc để lưu trữ tệp bằng cách sử dụng `os.path`, đảm bảo tương thích đa nền tảng. Hai hàm tiện ích được định nghĩa:

- `convert_age_to_decimal(age_str)`: Chuyển đổi tuổi từ định dạng chuỗi (ví dụ: “22-123”) thành số thập phân (ví dụ: 22.34), giả sử một năm có 365 ngày.
- `extract_country_code(nation_str)`: Trích xuất mã quốc gia từ chuỗi quốc tịch (ví dụ: “eng-Eng” thành “Eng”).

Selenium được cấu hình để chạy Chrome ở chế độ ẩn (không hiển thị cửa sổ trình duyệt) với các tùy chọn như `-disable-gpu` và `-no-sandbox` để tối ưu hóa hiệu suất và tương thích. ChromeDriver được tự động cài đặt và khởi tạo bằng `webdriver_manager`.

1.3.2 Xác định nguồn dữ liệu

Mã xác định danh sách các URL và ID bảng tương ứng để thu thập dữ liệu từ các danh mục cụ thể (ví dụ: thống kê tiêu chuẩn, thủ môn, sút bóng) trên trang FBref. Các URL nhắm đến các trang như:

- `https://fbref.com/en/comps/9/2024-2025/stats/...` cho thống kê tiêu chuẩn.
- `https://fbref.com/en/comps/9/2024-2025/keepers/...` cho thống kê thủ môn.

Các ID bảng (ví dụ: `stats_standard`, `stats_keeper`) xác định các bảng HTML chứa dữ liệu mong muốn. Danh sách `required_columns` chỉ định các chỉ số cần trích xuất, như tên cầu thủ, số bàn thắng, kiến tạo và kỳ vọng bàn thắng (xG). Một `column_rename_dict` ánh xạ tên cột thô từ trang web sang tên chuẩn hóa để đảm bảo nhất quán.

1.3.3 Quy trình thu thập dữ liệu web

Mã lặp qua các URL và ID bảng song song bằng vòng lặp `for`. Với mỗi cặp URL-bảng:

- Selenium (`driver.get(url)`) tải trang web, với thời gian chờ 3 giây (`time.sleep(3)`) để đảm bảo nội dung JavaScript được hiển thị.
- `BeautifulSoup` phân tích HTML của trang (`driver.page_source`) để tìm bảng với ID được chỉ định.
- Nếu tìm thấy bảng, `read_html` của Pandas chuyển bảng HTML thành DataFrame, sử dụng hàng đầu tiên làm tiêu đề.
- Các cột được đổi tên bằng `column_rename_dict`, và các cột trùng lặp được loại bỏ để tránh dư thừa.
- Một cột `Player_Team` được tạo bằng cách nối `Player` và `Team` để làm định danh duy nhất cho việc gộp dữ liệu.
- Cột `Age`, nếu có, được chuyển đổi sang định dạng thập phân bằng `convert_age_to_decimal`.
- DataFrame được lọc để chỉ bao gồm các `required_columns` và lưu vào từ điển `all_tables`.

1.3.4 Gộp dữ liệu

Mã gộp tất cả các DataFrame trong `all_tables` thành một `merged_df` duy nhất:

- DataFrame đầu tiên khởi tạo `merged_df`.
- Các DataFrame tiếp theo được gộp bằng `pd.merge` với phép nối ngoài (`outer join`) dựa trên cột `Player_Team`.
- Các hàng trùng lặp trong mỗi DataFrame được loại bỏ trước khi gộp để tránh dư thừa.
- Các cột xung đột (ví dụ: `Player` từ các bảng khác nhau) được xử lý bằng cách gộp giá trị và xóa cột thừa.
- Cột `Player_Team` được xóa khỏi DataFrame cuối cùng.

1.3.5 Làm sạch và định dạng dữ liệu

DataFrame được gộp trả qua các bước làm sạch:

- Các cột được sắp xếp lại theo `required_columns`, loại bỏ các cột không có mặt.
- Các kiểu dữ liệu được áp dụng:
  - Các cột số nguyên (ví dụ: `Minutes`, `Gls`) được chuyển thành `Int64` (kiểu số nguyên có thể chứa giá trị null).
  - Các cột số thực (ví dụ: `xG`, `Save%`) được chuyển thành số thực và làm tròn đến hai chữ số thập phân.
  - Các cột chuỗi (ví dụ: `Player`, `Nation`) được giữ nguyên dưới dạng chuỗi.
- Các cầu thủ có thời gian thi đấu dưới 90 phút bị loại bỏ.
- Cột `Nation` được làm sạch bằng `extract_country_code` để chuẩn hóa mã quốc gia.

1.3.6 Tạo tệp đầu ra

Mã tạo một thư mục `csv` trong thư mục gốc nếu chưa tồn tại. DataFrame `merged_df` đã làm sạch được lưu dưới dạng `result.csv` với:

- Mã hóa UTF-8 với BOM (`utf-8-sig`) để tương thích với Excel.
- Không bao gồm cột chỉ số (`index=False`).
- Các giá trị thiếu được biểu diễn bằng "N/A" (`na_rep="N/A"`).

Mã in thông báo xác nhận với đường dẫn tệp và kích thước DataFrame, sau đó đóng Selenium WebDriver.

1.4 Kết luận

Mã Python thu thập và xử lý thống kê bóng đá từ FBref một cách hiệu quả, chuyển đổi các bảng HTML thô thành một tệp CSV có cấu trúc. Luồng thực thi được tổ chức tốt, tiến triển từ cấu hình và trích xuất dữ liệu đến gộp, làm sạch và tạo đầu ra. Thiết kế mã ưu tiên tính mạnh mẽ và khả năng thích nghi, phù hợp cho các tác vụ thu thập dữ liệu tự động.

2 Phân tích luồng chạy mã nguồn phân tích dữ liệu thống kê bóng đá trong bài 2

2.1 Giới thiệu

Các phần sau sẽ phân tích luồng thực thi của mã Python được thiết kế để xử lý và phân tích dữ liệu thống kê bóng đá từ tệp `result.csv`. Mã sử dụng các thư viện `pandas`, `os`, và `matplotlib` để thực hiện các tác vụ như xếp hạng cầu thủ, tính toán thống kê đội bóng, tạo biểu đồ histogram, và xác định đội bóng có hiệu suất tốt nhất. Phân tích này trình bày chi tiết cấu trúc mã, các thành phần chính và các bước xử lý dữ liệu.

2.2 Cấu trúc mã và các thư viện phụ thuộc

Mã nguồn bắt đầu bằng việc nhập các thư viện Python cần thiết, mỗi thư viện đảm nhiệm một vai trò cụ thể:

- `pandas`: Xử lý và phân tích dữ liệu dưới dạng bảng (DataFrame).
- `os`: Quản lý đường dẫn tệp và tạo thư mục.
- `matplotlib.pyplot`: Tạo biểu đồ histogram để trực quan hóa dữ liệu.

2.3 Phân tích luồng thực thi

Mã được tổ chức thành một hàm chính `main()` gọi các hàm riêng biệt cho từng chức năng. Luồng thực thi có thể được chia thành các giai đoạn chính sau.

2.3.1 Khởi tạo và cấu hình

Mã bắt đầu bằng việc xác định thư mục gốc (`base_dir`) bằng `os.path`, đảm bảo tương thích đa nền tảng. Các thư mục `csv`, `txt`, và `histograms` (với các thư mục con `all_players` và `teams`) được tạo nếu chưa tồn tại. Tệp `result.csv` được đọc vào DataFrame `df` với các giá trị "N/A" được xử lý thành `NaN`. Một bản sao `df_calc` được tạo để tính toán mà không ảnh hưởng đến dữ liệu gốc. Các cột được phân loại:

- `exclude_columns`: Các cột không tính toán (`Player`, `Nation`, `Team`, `Position`).
- `numeric_columns`: Các cột số còn lại, được ép kiểu thành số và điền `NaN` bằng 0.

2.3.2 Xếp hạng cầu thủ

Hàm `generate_top_3_rankings()` tạo xếp hạng cho mỗi cột số:

- Với mỗi cột, lấy 3 cầu thủ có giá trị cao nhất và thấp nhất, kèm theo `Player` và `Team`.
- Đổi tên cột số thành `Value` và thêm cột `Rank` ("1st", "2nd", "3rd").
- Lưu kết quả vào từ điển `rankings` với hai khóa: `Highest` và `Lowest`.
- Ghi kết quả vào `top_3.txt` với định dạng dễ đọc, phân tách bằng dấu gạch ngang.

2.3.3 Tính toán thống kê đội bóng

Hàm `compute_team_statistics()` tính toán median, mean và độ lệch chuẩn:

- Cho toàn giải: Tính các chỉ số cho tất cả cầu thủ, lưu vào từ điển `all_stats`.
- Cho từng đội: Lọc dữ liệu theo `Team`, tính các chỉ số và lưu vào `team_stats`.
- Tạo DataFrame từ danh sách các từ điển, làm tròn giá trị đến 2 chữ số thập phân.
- Lưu vào `results2.csv` với mã hóa UTF-8 BOM cho tương thích Excel.

2.3.4 Tạo biểu đồ histogram

Hàm `generate_histograms()` tạo biểu đồ cho các thống kê được chọn (`Glsper90`, `xGper90`, `SCA90`, `GA90`, `TklW`, `Blocks`):

- Lọc cầu thủ chơi ít nhất 300 phút (nếu có cột `Min`).
- Cho toàn giải: Vẽ histogram với 20 bin, lưu vào `histograms/all_players`.
- Cho từng đội: Vẽ histogram với 10 bin, sử dụng màu xanh lá cho thống kê phòng ngự, lưu vào `histograms/teams`.
- Các biểu đồ có tiêu đề, nhãn trục và lưới, được lưu dưới dạng PNG.

2.3.5 Xác định đội dẫn đầu từng thống kê

Hàm `compute_highest_team_stats()` tìm đội có giá trị trung bình cao nhất cho mỗi thống kê:

- Nhóm dữ liệu theo `Team`, tính trung bình các cột số.

- Với mỗi cột, tìm đội có giá trị trung bình lớn nhất, lưu thông tin vào danh sách.
- Tạo DataFrame và lưu vào `highest_team_stats.csv`.

2.3.6 Xác định đội xuất sắc nhất

Hàm `identify_best_team()` xác định đội dẫn đầu nhiều thống kê tích cực nhất:

- Đọc `highest_team_stats.csv`.
- Loại bỏ các thống kê tiêu cực (ví dụ: `GA90`, `CrdY`).
- Đếm số lần mỗi đội xuất hiện trong các thống kê tích cực.
- Xác định đội có số lần dẫn đầu cao nhất và ghi kết quả vào `Thebest-performingteam.txt`.

2.3.7 Thực thi chính

Hàm `main()` gọi các hàm trên theo thứ tự, đảm bảo `compute_highest_team_stats()` chạy trước `identify_best_team()` do phụ thuộc dữ liệu. Mã in thông báo xác nhận sau mỗi tác vụ.

2.4 Kết luận

Mã Python phân tích dữ liệu bóng đá từ `result.csv` một cách hiệu quả, tạo ra các xếp hạng, thống kê, biểu đồ và đánh giá đội bóng xuất sắc. Luồng thực thi được tổ chức thành các hàm độc lập, đảm bảo tính mô-đun và dễ bảo trì. Kết quả được lưu dưới dạng CSV, TXT và PNG, phù hợp cho phân tích tiếp theo và trực quan hóa dữ liệu.

3 Phân tích số lượng cụm tối ưu và vẽ biểu đồ bằng PCA trong bài 3

Thuật toán K-means được sử dụng để gom các cầu thủ trong tệp `result.csv` thành các cụm. Phương pháp Elbow Method được áp dụng để xác định số cụm tối ưu, với kết quả cho thấy số cụm tối ưu là $k = 4$.

3.1 Phương pháp Elbow Method

Phương pháp Elbow Method phân tích giá trị inertia (tổng bình phương khoảng cách từ các điểm dữ liệu đến tâm cụm), được định nghĩa bởi công thức:

$$\text{Inertia} = \sum_{i=1}^{m} \min \|x_i - \mu_j\|^2$$

trong đó x_i là điểm dữ liệu, μ_j là tâm cụm, và m là số cầu thủ. Điểm khuỷu tay (elbow point) là giá trị k mà tại đó inertia giảm chậm lại, được xác định bằng `KneeLocator`.

3.2 Kết quả

Biểu đồ Elbow Method (lưu tại `histograms/K-means/Theoptimalnumberofclusters.png`) cho thấy:

- Khi $k = 1, 2$, inertia rất lớn, cho thấy gom cụm kém.
- Từ $k = 1$ đến $k = 4$, inertia giảm mạnh, nghĩa là chất lượng gom cụm cải thiện đáng kể.
- Sau $k = 4$, đường cong inertia trở nên phẳng hơn, cho thấy việc thêm cụm chỉ giảm inertia không đáng kể.

Do đó, `KneeLocator` xác định $k = 4$ là điểm khuỷu tay, là số cụm tối ưu.

3.3 Lý do chọn $k = 4$

- **Hiệu quả gom cụm**: Từ $k = 1$ đến $k = 4$, mỗi cụm thêm vào cải thiện đáng kể chất lượng gom cụm, thể hiện qua sự giảm mạnh của inertia.
- **Cân bằng độ phức tạp**: Chọn $k = 4$ tránh tạo quá nhiều cụm (dẫn đến quá khớp) hoặc quá ít cụm (không đủ chi tiết).
- **Ý nghĩa bóng đá**: Số cụm $k = 4$ có thể đại diện cho các nhóm cầu thủ với hiệu suất khác nhau, ví dụ:
  - Cụm 1: Cầu thủ ngôi sao (cao về `Gls`, `KeyPasses`, `Save%`).
  - Cụm 2: Cầu thủ trung bình khá (hiệu suất ổn định).
  - Cụm 3: Cầu thủ trung bình yếu (hiệu suất thấp).
  - Cụm 4: Cầu thủ ít thi đấu (thống kê thấp).

3.4 Quy trình thực hiện K-means

Quy trình gom cụm bao gồm:

- **Tiền xử lý dữ liệu**:
  - Loại bỏ các cột không tính toán (`Player`, `Nation`, `Team`, `Position`).
  - Chuyển đổi sang kiểu số, điền `NaN` bằng 0.
  - Chuẩn hóa bằng `StandardScaler` để trung bình = 0, độ lệch chuẩn = 1.

- **Chạy K-means**: Thử nghiệm với $k = 1$ đến $k = 15$, sử dụng `random_state=42` và `n_init=10` để đảm bảo kết quả tái lập.
- **Xác định k tối ưu**: Sử dụng `KneeLocator` để tìm điểm khuỷu tay tại $k = 4$.
- **Trực quan hóa**: Vẽ biểu đồ Elbow Method và lưu tại thư mục `histograms/K-means`.

3.5 Trực quan hóa cụm bằng PCA

Để trực quan hóa các cụm, dữ liệu được giảm chiều từ không gian đa chiều xuống 2 chiều bằng phương pháp Phân tích Thành phần Chính (PCA). Các bước thực hiện bao gồm:

- **Giảm chiều dữ liệu**: Sử dụng PCA với `n_components=2` để chuyển dữ liệu đã chuẩn hóa thành tọa độ trong không gian 2D, gồm hai thành phần chính:
  - PCA1: Đại diện cho khả năng sáng tạo và tấn công (ví dụ: ghi bàn, kiến tạo).
  - PCA2: Đại diện cho khả năng phòng ngự và kiểm soát bóng (ví dụ: chuyền bóng chính xác, hoạt động phòng ngự).
- **Phân cụm**: Áp dụng thuật toán K-means với $k = 4$ (số cụm tối ưu) để gán nhãn cụm cho từng cầu thủ.
- **Vẽ biểu đồ**: Một biểu đồ phân tán được tạo, trong đó:
  - Mỗi điểm đại diện cho một cầu thủ, với tọa độ là giá trị PCA1 và PCA2.
  - Màu sắc của các điểm tương ứng với nhãn cụm, sử dụng bảng màu `viridis`.
  - Trục x biểu thị "Khả năng sáng tạo và tấn công", trục y biểu thị "Khả năng phòng ngự và kiểm soát bóng".
  - Thanh màu hiển thị các cụm, và lưới được thêm để dễ quan sát.
- **Lưu kết quả**: Biểu đồ được lưu tại `histograms/K-means/PCA_2D_Cluster_Plot.png`.

Biểu đồ này giúp trực quan hóa sự phân bố và mức độ tách biệt giữa các cụm. Tuy nhiên, việc giảm chiều có thể làm mất một phần thông tin, và ý nghĩa của PCA1, PCA2 cần được kiểm chứng thêm dựa trên các đặc trưng gốc.

3.6 Kết luận

Số cụm tối ưu để phân loại cầu thủ là $k = 4$, được xác định bởi phương pháp Elbow Method với điểm khuỷu tay tại $k = 4$. Lựa chọn này cân bằng giữa hiệu quả gom cụm và ý nghĩa bóng đá, phản ánh căng thẳng về sự phân bố các cụm.

4 Phân tích đặc trưng dự đoán giá trị chuyển nhượng ước tính cho các vị trí bóng đá trong bài 4

4.1 Giới thiệu

Các phần sau sẽ phân tích các đặc trưng được sử dụng để dự đoán giá trị chuyển nhượng ước tính (ETV) của cầu thủ bóng đá ở bốn vị trí: Tiền vệ (MF), Thủ môn (GK), Tiền đạo (FW) và Hậu vệ (DF). Dựa trên dữ liệu từ tệp `result.csv`, các đặc trưng được chọn nhằm phản ánh vai trò đặc thù của từng vị trí và tương quan với giá trị thị trường. Mục tiêu là đánh giá tính toàn diện của bộ đặc trưng, phân tích cách chúng tương tác với mô hình hồi quy tuyến tính, đồng thời đề xuất các cải tiến để nâng cao độ chính xác dự đoán ETV. Báo cáo cung cấp cái nhìn chi tiết về cách các đặc trưng được xử lý, so sánh giữa các vị trí và định hướng tối ưu hóa mô hình trong tương lai.

4.2 Tổng quan về đặc trưng theo vị trí

Mỗi vị trí có bộ đặc trưng riêng, được chọn để bao quát vai trò chính, tương quan với giá trị thị trường, và phù hợp với dữ liệu có sẵn. Bảng tóm tắt số lượng và nhóm đặc trưng chính.

4.3 Phân tích chi tiết đặc trưng

Dưới đây là phân tích chi tiết từng đặc trưng, bao gồm định nghĩa, lý do chọn, cách xử lý trong mã, ví dụ minh họa, và so sánh giữa các vị trí.

4.3.1 Tiền vệ (MF) - 14 đặc trưng

- **KeyPasses (Số đường chuyền tạo cơ hội)**
  Định nghĩa: Số đường chuyền dẫn đến cú sút (ví dụ: 50 lần).
  Lý do chọn: Đo lường sáng tạo, tương quan với ETV (như Kevin De Bruyne), phổ biến trong `result.csv`.
  Xử lý trong mã: Chuyển thành số, điền giá trị thiếu bằng trung vị, biến đổi log (`np.log1p`), trọng số 2.0, chuẩn hóa bằng `StandardScaler`.
  Ví dụ: Tiền vệ A (KeyPasses = 60) > Tiền vệ B (KeyPasses = 30).

- **SCA (Shot-Creating Actions)**
  Định nghĩa: Số hành động dẫn đến cú sút (chuyền, rê, phạm lỗi) (ví dụ: 100 lần).
  Lý do chọn: Đo số lượng sáng tạo, bổ sung cho KeyPasses, tương quan với ETV.
  Xử lý: Như KeyPasses, trọng số 2.0.
  Ví dụ: Tiền vệ A (SCA = 120) > Tiền vệ B (SCA = 60).

- **xAG (Expected Assists)**
  Định nghĩa: Giá trị kỳ vọng kiến tạo dựa trên chất lượng đường chuyền (ví dụ: xAG = 8.0).
  Lý do chọn: Đo chất lượng sáng tạo, tương quan với ETV.
  Xử lý: Như KeyPasses, trọng số 2.0.
  Ví dụ: Tiền vệ A (xAG = 10.0) > Tiền vệ B (xAG = 5.0).

- **Ast (Assists)**
  Định nghĩa: Số pha kiến tạo dẫn đến bàn thắng (ví dụ: 10 lần).
  Lý do chọn: Kết quả sáng tạo, tương quan cao với ETV.
  Xử lý: Như KeyPasses, trọng số 2.0.
  Ví dụ: Tiền vệ A (Ast = 12) > Tiền vệ B (Ast = 5).

- **PrgP (Progressive Passes)**
  Định nghĩa: Số chuyền đưa bóng tiến gần khung thành (ví dụ: 150 chuyền).
  Lý do chọn: Khởi tạo tấn công, tương quan với ETV.
  Xử lý: Như KeyPasses, trọng số 2.0.
  Ví dụ: Tiền vệ A (PrgP = 200) > Tiền vệ B (PrgP = 100).

- **PPA (Passes into Penalty Area)**
  Định nghĩa: Số chuyền vào khu vực vòng cấm (ví dụ: 40 chuyền).
  Lý do chọn: Đo tấn công trực tiếp, tương quan với ETV.
  Xử lý: Như KeyPasses, trọng số 2.0.
  Ví dụ: Tiền vệ A (PPA = 50) > Tiền vệ B (PPA = 20).

- **Pass_into_1/3 (Passes into Final Third)**
  Định nghĩa: Số chuyền vào 1/3 sân cuối (ví dụ: 200 chuyền).
  Lý do chọn: Đo khả năng đưa bóng vào khu vực nguy hiểm.
  Xử lý: Như KeyPasses, trọng số 2.0.
  Ví dụ: Tiền vệ A (Pass_into_1/3 = 250) > Tiền vệ B (Pass_into_1/3 = 150).

- **Carries1_3 (Carries into Final Third)**
  Định nghĩa: Số lần mang bóng vào 1/3 sân cuối (ví dụ: 50 lần).
  Lý do chọn: Đo tấn công trực tiếp qua rê bóng.
  Xử lý: Như KeyPasses, trọng số 2.0.
  Ví dụ: Tiền vệ A (Carries1_3 = 60) > Tiền vệ B (Carries1_3 = 30).

- **ProDist (Progressive Distance)**
  Định nghĩa: Tổng khoảng cách mang bóng tiến gần khung thành (ví dụ: 5,000 mét).
  Lý do chọn: Đo đóng góp tấn công qua rê bóng.
  Xử lý: Như KeyPasses, trọng số 2.0.
  Ví dụ: Tiền vệ A (ProDist = 6,000) > Tiền vệ B (ProDist = 3,000).

- **Cmp% (Pass Completion Percentage)**
  Định nghĩa: Tỷ lệ chuyền chính xác ($\frac{\text{Số chuyền hoàn thành}}{\text{Tổng số chuyền}} \times 100$), ví dụ: 90%.
  Lý do chọn: Đo độ tin cậy kiểm soát bóng.
  Xử lý: Chuyển đổi, điền giá trị thiếu, biến đổi log, không trọng số, chuẩn hóa.
  Ví dụ: Tiền vệ A (Cmp% = 92%) > Tiền vệ B (Cmp% = 80%).

- **Touches**
  Định nghĩa: Số lần chạm bóng (ví dụ: 1,500 lần).
  Lý do chọn: Đo mức độ tham gia kiểm soát bóng.
  Xử lý: Như Cmp%, không trọng số.
  Ví dụ: Tiền vệ A (Touches = 1,800) > Tiền vệ B (Touches = 1,200).

- **Rec (Receptions)**
  Định nghĩa: Số lần nhận bóng thành công (ví dụ: 1,200 lần).
  Lý do chọn: Đo khả năng tham gia lối chơi.
  Xử lý: Như Cmp%, không trọng số.
  Ví dụ: Tiền vệ A (Rec = 1,500) > Tiền vệ B (Rec = 1,000).

- **Dis (Dispossessed)**
  Định nghĩa: Số lần mất bóng do bị cướp (ví dụ: 20 lần).
  Lý do chọn: Đo sai lầm kỹ thuật, Dis thấp tăng ETV.
  Xử lý: Như Cmp%, không trọng số.
  Ví dụ: Tiền vệ A (Dis = 10) > Tiền vệ B (Dis = 30).

- **Mis (Miscontrols)**
  Định nghĩa: Số lần kiểm soát bóng kém dẫn đến mất bóng (ví dụ: 15 lần).
  Lý do chọn: Đo độ tin cậy kỹ thuật.
  Xử lý: Như Cmp%, không trọng số.
  Ví dụ: Tiền vệ A (Mis = 10) > Tiền vệ B (Mis = 25).

4.3.2 Thủ môn (GK) - 4 đặc trưng

- **GA90 (Goals Against per 90)**
  Định nghĩa: Số bàn thua trung bình mỗi 90 phút (ví dụ: GA90 = 1.0).
  Lý do chọn: Đo hiệu quả ngăn bàn thua, GA90 thấp tăng ETV (như Alisson Becker).
  Xử lý: Chuyển đổi, điền giá trị thiếu, biến đổi log, trọng số 2.0 (hệ số âm), chuẩn hóa.
  Ví dụ: Thủ môn A (GA90 = 0.8) > Thủ môn B (GA90 = 1.5).

- **CS% (Clean Sheet Percentage)**
  Định nghĩa: Tỷ lệ trận giữ sạch lưới ($\frac{\text{Số trận không thủng lưới}}{\text{Tổng số trận}} \times 100$), ví dụ: 40%.
  Lý do chọn: Đo khả năng ngăn bàn thua tổng thể.

Xử lý: Chuyển đổi, điền giá trị thiếu, biến đổi log, trọng số 2.0, chuẩn hóa.
Ví dụ: Thủ môn A (CS% = 45%) > Thủ môn B (CS% = 25%).

- **Save% (Save Percentage)**
  Định nghĩa: Tỷ lệ cản phá cú sút trúng đích ($\frac{\text{Số lần cản phá}}{\text{Tổng số cú sút trúng đích}} \times 100$), ví dụ: 75%.
  Lý do chọn: Đo kỹ năng cản phá, tương quan với ETV.
  Xử lý: Như CS%, trọng số 2.0.
  Ví dụ: Thủ môn A (Save% = 80%) > Thủ môn B (Save% = 65%).

- **PKSave% (Penalty Kick Save Percentage)**
  Định nghĩa: Tỷ lệ cản phá phạt đền ($\frac{\text{Số lần cản phá phạt đền}}{\text{Tổng số phạt đền}} \times 100$), ví dụ: 30%.
  Lý do chọn: Đo kỹ năng đặc biệt, tăng ETV.
  Xử lý: Như CS%, trọng số 2.0.
  Ví dụ: Thủ môn A (PKSave% = 35%) > Thủ môn B (PKSave% = 15%).

4.3.3 Tiền đạo (FW) - 11 đặc trưng

- **Gls (Goals)**
  Định nghĩa: Số bàn thắng ghi được (ví dụ: 25 bàn).
  Lý do chọn: Cốt lõi của tiền đạo, tương quan cao với ETV (như Erling Haaland).
  Xử lý: Chuyển đổi, điền giá trị thiếu bằng trung vị, biến đổi log, trọng số 2.0, chuẩn hóa.
  Ví dụ: Tiền đạo A (Gls = 30) > Tiền đạo B (Gls = 15).

- **xGper90 (Expected Goals per 90)**
  Định nghĩa: Giá trị kỳ vọng bàn thắng mỗi 90 phút (ví dụ: xGper90 = 0.8).
  Lý do chọn: Đo chất lượng cơ hội ghi bàn.
  Xử lý: Như Gls, trọng số 2.0.
  Ví dụ: Tiền đạo A (xGper90 = 0.9) > Tiền đạo B (xGper90 = 0.4).

- **Glsper90 (Goals per 90)**
  Định nghĩa: Số bàn thắng trung bình mỗi 90 phút (ví dụ: Glsper90 = 0.7).
  Lý do chọn: Đo hiệu suất ghi bàn ổn định.
  Xử lý: Như Gls, trọng số 2.0.
  Ví dụ: Tiền đạo A (Glsper90 = 1.0) > Tiền đạo B (Glsper90 = 0.5).

- **Gpersh (Goals per Shot)**
  Định nghĩa: Số bàn thắng trên mỗi cú sút (ví dụ: Gpersh = 0.2).
  Lý do chọn: Đo hiệu quả ghi bàn.
  Xử lý: Như Gls, trọng số 2.0.
  Ví dụ: Tiền đạo A (Gpersh = 0.3) > Tiền đạo B (Gpersh = 0.1).

- **Ast (Assists)**
  Định nghĩa: Số pha kiến tạo (ví dụ: 10 lần).
  Lý do chọn: Đo sáng tạo, bổ sung ghi bàn.
  Xử lý: Như Gls, trọng số 2.0.
  Ví dụ: Tiền đạo A (Ast = 12) > Tiền đạo B (Ast = 5).

- **SCA90 (Shot-Creating Actions per 90)**
  Định nghĩa: Số hành động dẫn đến cú sút mỗi 90 phút (ví dụ: SCA90 = 3.5).
  Lý do chọn: Đo số lượng sáng tạo.
  Xử lý: Như Gls, trọng số 2.0.
  Ví dụ: Tiền đạo A (SCA90 = 4.0) > Tiền đạo B (SCA90 = 2.0).

- **GCA90 (Goal-Creating Actions per 90)**
  Định nghĩa: Số hành động dẫn đến bàn thắng mỗi 90 phút (ví dụ: GCA90 = 0.5).
  Lý do chọn: Đo chất lượng sáng tạo.
  Xử lý: Như Gls, trọng số 2.0.
  Ví dụ: Tiền đạo A (GCA90 = 0.6) > Tiền đạo B (GCA90 = 0.2).

- **PrgC (Progressive Carries)**
  Định nghĩa: Số lần mang bóng tiến gần khung thành (ví dụ: 70 lần).
  Lý do chọn: Đo tấn công trực tiếp.
  Xử lý: Như Gls, trọng số 2.0.
  Ví dụ: Tiền đạo A (PrgC = 80) > Tiền đạo B (PrgC = 40).

- **Carries1/3 (Carries into Final Third)**
  Định nghĩa: Số lần mang bóng vào 1/3 sân cuối (ví dụ: 50 lần).
  Lý do chọn: Đo khả năng tạo nguy hiểm.
  Xử lý: Như Gls, trọng số 2.0.
  Ví dụ: Tiền đạo A (Carries1/3 = 60) > Tiền đạo B (Carries1/3 = 30).

- **SoT% (Shots on Target Percentage)**
  Định nghĩa: Tỷ lệ sút trúng đích ($\frac{\text{Số cú sút trúng đích}}{\text{Tổng số cú sút}} \times 100$), ví dụ: 50%.
  Lý do chọn: Đo độ chính xác.
  Xử lý: Chuyển đổi, điền giá trị thiếu, biến đổi log, không trọng số, chuẩn hóa.
  Ví dụ: Tiền đạo A (SoT% = 55%) > Tiền đạo B (SoT% = 35%).

- **AerialWon% (Aerial Duels Won Percentage)**
  Định nghĩa: Tỷ lệ thắng tranh chấp bóng bổng ($\frac{\text{Số lần thắng}}{\text{Tổng số tranh chấp}} \times 100$), ví dụ: 70%.
  Lý do chọn: Đo khả năng không chiến, quan trọng với tiền đạo cao lớn.
  Xử lý: Như SoT%, không trọng số.
  Ví dụ: Tiền đạo A (AerialWon% = 80%) > Tiền đạo B (AerialWon% = 60%).

4.3.4 Hậu vệ (DF) - 13 đặc trưng

- **Tkl (Tackles)**
  Định nghĩa: Số lần tắc bóng (ví dụ: 60 lần).
  Lý do chọn: Cốt lõi phòng ngự, tương quan với ETV (như Aaron Wan-Bissaka).
  Xử lý: Chuyển đổi, điền giá trị thiếu bằng trung vị, biến đổi log, trọng số 2.0, chuẩn hóa.
  Ví dụ: Hậu vệ A (Tkl = 70) > Hậu vệ B (Tkl = 40).

- **TklW (Tackles Won)**
  Định nghĩa: Số lần tắc bóng giành lại bóng (ví dụ: 35/50 lần).
  Lý do chọn: Đo hiệu quả tắc bóng, bổ sung Tkl.
  Xử lý: Như Tkl, trọng số 2.0.
  Ví dụ: Hậu vệ A (TklW = 40) > Hậu vệ B (TklW = 20).

- **Int (Interceptions)**
  Định nghĩa: Số lần đánh chặn đường chuyền (ví dụ: 50 lần).
  Lý do chọn: Đo khả năng đọc trận đấu, tương quan với ETV (như Virgil van Dijk).
  Xử lý: Như Tkl, trọng số 2.0.
  Ví dụ: Hậu vệ A (Int = 60) > Hậu vệ B (Int = 30).

- **Blocks**
  Định nghĩa: Số lần chặn cú sút hoặc chuyền nguy hiểm (ví dụ: 30 lần).
  Lý do chọn: Bảo vệ khung thành, tăng ETV.
  Xử lý: Như Tkl, trọng số 2.0.
  Ví dụ: Hậu vệ A (Blocks = 35) > Hậu vệ B (Blocks = 15).

- **Recov (Recoveries)**
  Định nghĩa: Số lần thu hồi bóng lỏng (ví dụ: 200 lần).
  Lý do chọn: Đo ý thức chiến thuật, tăng ETV.
  Xử lý: Như Tkl, trọng số 2.0.
  Ví dụ: Hậu vệ A (Recov = 250) > Hậu vệ B (Recov = 150).

- **AerialWon (Aerial Duels Won)**
  Định nghĩa: Số lần thắng tranh chấp bóng bổng (ví dụ: 80 lần).
  Lý do chọn: Quan trọng với hậu vệ trung tâm, tăng ETV.
  Xử lý: Như Tkl, trọng số 2.0.
  Ví dụ: Hậu vệ A (AerialWon = 90) > Hậu vệ B (AerialWon = 50).

- **AerialWon% (Aerial Duels Won Percentage)**
  Định nghĩa: Tỷ lệ thắng tranh chấp bóng bổng (ví dụ: 80%).
  Lý do chọn: Đo hiệu quả không chiến, bổ sung AerialWon.
  Xử lý: Như Tkl, trọng số 2.0.

Ví dụ: Hậu vệ A (AerialWon% = 85%) > Hậu vệ B (AerialWon% = 60%).

- **Cmp (Completed Passes)**
  Định nghĩa: Số đường chuyền hoàn thành (ví dụ: 1,000 chuyền).
  Lý do chọn: Đo phát triển bóng, tăng ETV.
  Xử lý: Chuyển đổi, điền giá trị thiếu, biến đổi log, không trọng số, chuẩn hóa.
  Ví dụ: Hậu vệ A (Cmp = 1,200) > Hậu vệ B (Cmp = 800).

- **Cmp% (Pass Completion Percentage)**
  Định nghĩa: Tỷ lệ chuyền chính xác (ví dụ: 90%).
  Lý do chọn: Đo độ tin cậy chuyền bóng.
  Xử lý: Như Cmp, không trọng số.
  Ví dụ: Hậu vệ A (Cmp% = 92%) > Hậu vệ B (Cmp% = 80%).

- **PrgP (Progressive Passes)**
  Định nghĩa: Số chuyền tiến gần khung thành (ví dụ: 150 chuyền).
  Lý do chọn: Đo khả năng khởi tạo tấn công.
  Xử lý: Như Cmp, không trọng số.
  Ví dụ: Hậu vệ A (PrgP = 200) > Hậu vệ B (PrgP = 100).

- **LongCmp% (Long Pass Completion Percentage)**
  Định nghĩa: Tỷ lệ chuyền dài thành công (ví dụ: 65%).
  Lý do chọn: Đo chuyền bóng chiến thuật.
  Xử lý: Như Cmp, không trọng số.
  Ví dụ: Hậu vệ A (LongCmp% = 70%) > Hậu vệ B (LongCmp% = 50%).

- **Dis (Dispossessed)**
  Định nghĩa: Số lần mất bóng do bị cướp (ví dụ: 20 lần).
  Lý do chọn: Đo sai lầm kỹ thuật.
  Xử lý: Như Cmp, không trọng số.
  Ví dụ: Hậu vệ A (Dis = 10) > Hậu vệ B (Dis = 30).

- **Mis (Miscontrols)**
  Định nghĩa: Số lần kiểm soát bóng kém (ví dụ: 15 lần).
  Lý do chọn: Đo độ tin cậy kỹ thuật.
  Xử lý: Như Cmp, không trọng số.
  Ví dụ: Hậu vệ A (Mis = 10) > Hậu vệ B (Mis = 25).

4.4 So sánh đặc trưng giữa các vị trí

Bảng so sánh các nhóm đặc trưng chính giữa các vị trí.

Bảng 1: So sánh đặc trưng giữa các vị trí

Điểm tương đồng	
PrgP	Xuất hiện ở MF (trọng số 2.0) và DF (không trọng số), đo khả năng khởi tạo
Cmp%	Có ở MF và DF (không trọng số), đo độ tin cậy chuyền bóng.
Dis, Mis	Có ở MF và DF (không trọng số), đo sai lầm kỹ thuật.
Ast	Có ở MF và FW (trọng số 2.0), đo sáng tạo.
AerialWon%	Có ở FW và DF (không trọng số ở FW, trọng số 2.0 ở DF), nhưng FW tập tru
Điểm khác biệt	
Tiền vệ (MF)	Tập trung sáng tạo (4) và tấn công (5), nhưng thiếu phòng ngự.
Thủ môn (GK)	Chỉ có phòng ngự (4), thiếu phát triển bóng hoặc kiểm soát.
Tiền đạo (FW)	Ưu tiên ghi bàn (4) và tấn công (2), thiếu kiểm soát bóng.
Hậu vệ (DF)	Cân bằng phòng ngự (7) và phát triển bóng (4), nhưng thiếu sáng tạo.

4.5 Cách đặc trưng tương tác với mô hình

Mô hình hồi quy tuyến tính được sử dụng cho tất cả vị trí, với công thức tổng quát:

$$\text{ETV} = \beta_0 + \beta_1 \cdot x_1 + \beta_2 \cdot x_2 + \cdots + \beta_n \cdot x_n$$

trong đó x_i là đặc trưng, β_i là hệ số.

4.5.1 Tiền vệ (MF)

- **Hệ số**: Dương, lớn cho KeyPasses, SCA, xAG, Ast, PrgP, PPA, Pass_into_1/3, Carries1_3, ProDist (trọng số 2.0); âm cho Dis, Mis.
- **Tiền xử lý**: Biến đổi log (`np.log1p`), chuẩn hóa bằng `StandardScaler`, trọng số 2.0 cho sáng tạo/tấn công.

4.5.2 Thủ môn (GK)

- **Hệ số**: Âm cho GA90; dương, lớn cho CS%, Save%, PKSave% (trọng số 2.0).
- **Tiền xử lý**: Như MF, trọng số 2.0 cho phòng ngự.

4.5.3 Tiền đạo (FW)

- **Hệ số**: Dương, lớn cho Gls, xGper90, Glsper90, Gpersh, Ast, SCA90, GCA90, PrgC, Carries1/3 (trọng số 2.0); nhỏ hơn cho SoT%, AerialWon%.
- **Tiền xử lý**: Như MF, trọng số 2.0 cho ghi bàn/sáng tạo/tấn công.

4.5.4 Hậu vệ (DF)

- **Hệ số**: Dương, lớn cho Tkl, TklW, Int, Blocks, Recov, AerialWon, AerialWon% (trọng số 2.0); nhỏ hơn cho Cmp, Cmp%, PrgP, LongCmp%; âm cho Dis, Mis.
- **Tiền xử lý**: Như MF, trọng số 2.0 cho phòng ngự.

4.6 Quy trình huấn luyện mô hình

Quy trình huấn luyện mô hình hồi quy tuyến tính để dự đoán giá trị chuyển nhượng ước tính (ETV) bao gồm các bước chuẩn bị dữ liệu, chia tập dữ liệu, chọn siêu tham số, huấn luyện, đánh giá và tối ưu hóa. Mô hình sử dụng các đặc trưng được chọn (như KeyPasses, Gls, GA90, v.v.) từ tệp `result.csv` để dự đoán ETV, với mục tiêu giảm thiểu sai số dự đoán.

4.6.1 Chuẩn bị dữ liệu

Dữ liệu được tiền xử lý để đảm bảo phù hợp với mô hình hồi quy tuyến tính:

- **Xử lý giá trị thiếu**: Các giá trị thiếu trong các đặc trưng (như KeyPasses, Gls, GA90) được điền bằng trung vị của cột tương ứng. Nếu trung vị là dãy, giá trị 0 được sử dụng.
- **Chuyển đổi kiểu dữ liệu**: Tất cả đặc trưng được chuyển thành kiểu số (float hoặc integer) để đảm bảo tính toán chính xác.
- **Biến đổi log**: Áp dụng hàm $\ln(1+x)$ (`np.log1p`) cho các đặc trưng có phân phối lệch (như Gls, KeyPasses) để giảm độ lệch và cải thiện tính tuyến tính.
- **Tăng trọng số cho đặc trưng quan trọng**: Các đặc trưng quan trọng (ví dụ: KeyPasses cho tiền vệ, Gls cho tiền đạo, Save% cho thủ môn) được nhân với hệ số 2.0 để nhấn mạnh vai trò trong dự đoán ETV.
- **Chuẩn hóa**: Sử dụng `StandardScaler` để chuẩn hóa các đặc trưng về trung bình 0 và độ lệch chuẩn 1, đảm bảo các đặc trưng có thang đo tương đương.
- **Đảo ngược đặc trưng tiêu cực**: Đối với các đặc trưng mà giá trị thấp là tốt (như GA90), đảm bảo tương quan đúng với ETV.

4.6.2 Chia tập dữ liệu

Dữ liệu được chia thành ba tập:

- **Tập huấn luyện (70%)**: Dùng để huấn luyện mô hình, học các hệ số β_i của mô

hình hồi quy tuyến tính:

$$\text{ETV} = \beta_0 + \beta_1 \cdot x_1 + \beta_2 \cdot x_2 + \cdots + \beta_n \cdot x_n$$

trong đó x_i là các đặc trưng (như KeyPasses, Gls, Save%).

- **Tập xác nhận (15%)**: Dùng để điều chỉnh siêu tham số và đánh giá hiệu suất mô hình trong quá trình huấn luyện.
- **Tập kiểm tra (15%)**: Dùng để đánh giá cuối cùng hiệu suất của mô hình trên dữ liệu chưa thấy.

Việc chia dữ liệu được thực hiện ngẫu nhiên với `train_test_split` từ thư viện `scikit-learn`, đảm bảo phân phối đồng đều giữa các vị trí (MF, FW, GK).

4.6.3 Chọn siêu tham số

Mô hình hồi quy tuyến tính có ít siêu tham số, nhưng các tham số sau được điều chỉnh:

- **Hệ số phạt (Regularization)**: Sử dụng L2 regularization (Ridge Regression) với tham số α để tránh hiện tượng quá khớp (overfitting). Giá trị α được chọn dựa trên thử nghiệm.
- Quá trình điều chỉnh sử dụng chỉ số đánh giá Mean Squared Error (MSE) trên tập xác nhận để chọn bộ siêu tham số tốt nhất.

4.6.4 Huấn luyện mô hình

Mô hình hồi quy tuyến tính được huấn luyện trên tập huấn luyện sử dụng thư viện `scikit-learn`:

- **Hàm mất mát**: Mô hình tối ưu hóa hàm mất mát MSE:

$$\text{MSE} = \frac{1}{m} \sum_{i=1}^{m} \left(\text{ETV}_i - \hat{\text{ETV}}_i \right)^2$$

trong đó ETV_i là giá trị thực tế, $\hat{\text{ETV}}_i$ là giá trị dự đoán, và m là số mẫu.

- **Tối ưu hóa**: Sử dụng phương pháp Gradient Descent hoặc phương pháp giải tích (Normal Equation) để tìm các hệ số β_i tối ưu.
- **Xử lý đặc trưng tiêu cực**: Đối với các đặc trưng như GA90, hệ số β_i âm được kiểm tra để đảm bảo ảnh hưởng ngược chiều với ETV.

Quá trình huấn luyện được lặp lại với các giá trị siêu tham số khác nhau, sử dụng tập xác nhận để chọn mô hình tốt nhất.

4.6.5 Đánh giá mô hình

Mô hình được đánh giá trên tập kiểm tra bằng các chỉ số:

- **Mean Squared Error (MSE)**: Đo lường sai số bình phương trung bình giữa giá trị thực và dự đoán.
- **R-squared (R^2)**: Đo lường mức độ giải thích của mô hình đối với phương sai của ETV:

$$R^2 = 1 - \frac{\sum_{i=1}^{m} \left(\text{ETV}_i - \hat{\text{ETV}}_i\right)^2}{\sum_{i=1}^{m} \left(\text{ETV}_i - \overline{\text{ETV}}\right)^2}$$

  trong đó $\overline{\text{ETV}}$ là giá trị trung bình của ETV.
- **Mean Absolute Error (MAE)**: Đo lường sai số tuyệt đối trung bình, dễ diễn giải hơn MSE.

Ví dụ: Nếu mô hình đạt $R^2 = 0.85$ trên tập kiểm tra, điều này có nghĩa là 85% phương sai của ETV được giải thích bởi các đặc trưng.

4.6.6 Ví dụ minh họa

Giả sử dữ liệu của một tiền đạo có các đặc trưng sau: `Gls = 20`, `Ast = 8`, `xGper90 = 0.7`. Sau tiền xử lý (biến đổi log, chuẩn hóa, tăng trọng số), mô hình dự đoán:

$$\text{ETV} = \beta_0 + 2.0 \cdot \beta_1 \cdot \text{Gls}_{\text{scaled}} + 2.0 \cdot \beta_2 \cdot \text{Ast}_{\text{scaled}} + 2.0 \cdot \beta_3 \cdot \text{xGper90}_{\text{scaled}}$$

Nếu ETV thực tế là 50 triệu euro và ETV dự đoán là 48 triệu euro, sai số tuyệt đối là 2 triệu euro, cho thấy mô hình có độ chính xác cao.

4.7 Kết luận

Báo cáo đã phân tích chi tiết các đặc trưng dự đoán ETV cho bốn vị trí bóng đá. Tiền vệ (14 đặc trưng) bao quát sáng tạo và tấn công, thủ môn (4 đặc trưng) tập trung phòng ngự, tiền đạo (11 đặc trưng) ưu tiên ghi bàn, và hậu vệ (13 đặc trưng) cân bằng phòng ngự và phát triển bóng. Quy trình huấn luyện mô hình hồi quy tuyến tính bao gồm chuẩn bị dữ liệu kỹ lưỡng, chia tập dữ liệu hợp lý, điều chỉnh siêu tham số, và đánh giá hiệu suất bằng các chỉ số như MSE, R^2, và MAE. Các bước tối ưu hóa như kiểm tra đa cộng tuyến, thử nghiệm mô hình phi tuyến, và kiểm định chéo giúp cải thiện độ chính xác và độ tin cậy của mô hình trong dự đoán ETV. Mặc dù các đặc trưng phản ánh đúng vai trò và tương quan với thị trường, việc bổ sung đặc trưng mới, đánh giá hiệu suất mô hình, và yếu tố phi thống kê sẽ nâng cao độ chính xác dự đoán ETV.