|  |  |
| --- | --- |
| **bỘ GIÁO DỤC VÀ ĐÀO TẠO** | **BỘ NÔNG NGHIỆP VÀ PTNT** |

TRƯỜNG ĐẠI HỌC THỦY LỢI



NGUYỄN THÀNH TRUNG

Ứng dụng các mô hình học máy dự đoán giá trị chuyển nhượng của cầu thủ bóng đá chuyên nghiệp

ĐỒ ÁN TỐT NGHIỆP

HÀ NỘI, NĂM 2023

|  |  |
| --- | --- |
| **bỘ GIÁO DỤC VÀ ĐÀO TẠO** | **BỘ NÔNG NGHIỆP VÀ PTNT** |

TRƯỜNG ĐẠI HỌC THỦY LỢI

NGUYỄN THÀNH TRUNG

Ứng dụng các mô hình học máy dự đoán giá trị chuyển nhượng của cầu thủ bóng đá chuyên nghiệp

|  |  |
| --- | --- |
| Ngành: | Công nghệ thông tin |
| Mã số: | 7480201 |

|  |  |
| --- | --- |
| NGƯỜI HƯỚNG DẪN: | ThS. Nguyễn Đắc Hiếu |

HÀ NỘI, NĂM 2023

LỜI CAM ĐOAN

Tác giả xin cam đoan đây là Đồ án tốt nghiệp của bản thân tác giả. Các kết quả trong Đồ án tốt nghiệp này là trung thực, và không sao chép từ bất kỳ một nguồn nào và dưới bất kỳ hình thức nào. Việc tham khảo các nguồn tài liệu (nếu có) đã được thực hiện trích dẫn và ghi nguồn tài liệu tham khảo đúng quy định.

|  |  |
| --- | --- |
|  | **Tác giả ĐATN**  *Trung*  **Nguyễn Thành Trung** |
|  |  |

LỜI CẢM ƠN

Trong suốt quá trình học tập tại khoa Công nghệ thông tin trường Đại học Thủy Lợi, em cảm thấy vô cùng may mắn khi đã nhận được sự quan tâm, giúp đỡ và sự chỉ bảo nhiệt tình của các thầy, cô giáo trong cả học tập và đời sống.

Em muốn gửi lời cảm ơn chân thành và sâu sắc nhất đến với các thầy, cô giáo trường Đại học Thủy Lợi, những người đã tận tâm truyền đạt các kiến thức và niềm cảm hứng học tập đến cho cá nhân em và các bạn sinh viên khác. Và đặc biệt hơn, trong quãng thời gian làm đồ án tốt nghiệp, em đã nhận được sự hướng dẫn tận tình của Thạc sĩ, thầy giáo Nguyễn Đắc Hiếu. Em xin gửi long biết ơn sâu sắc đến thầy, người đã giúp đỡ, hỗ trợ và cho em những lời khuyên hữu ích để em có thể hoàn thành đồ án tốt nghiệp của mình hiệu quả và kịp tiến độ của học phần.

Trong thời gian thực hiện đồ án tốt nghiệp của mình với đề tài “ **Ứng dụng các mô hình học máy dự đoán giá trị chuyển nhượng của các cầu thủ bóng đá chuyên nghiệp**”, em đã cố gắng hết sức để xây dựng và hoàn thiện đồ án một cách tốt nhất, nhưng do kiến thức còn nhiều thiếu sót, thời gian hoàn thiện có hạn và thiếu kinh nghiệm thực tế nên không thể tránh khỏi các sai sót. Em rất mong nhận được sự góp ý của các thầy, cô để đồ án trở nên hoàn thiện hơn.

Em xin chân thành cảm ơn !

# CHƯƠNG 1: **GIỚI THIỆU BÀI TOÁN, GIỚI THIỆU TỔNG QUAN**

## Lý do chọn đề tài:

Với hơn 5 tỷ người hâm mộ trên toàn thế giới, bóng đá không chỉ là một môn thể thao phổ biến nhất trên thế giới mà còn là một hiện tượng văn hóa mạnh mẽ. Sự hấp dẫn của nó đã tạo ra một cộng đồng đa dạng và đồng lòng, làm cho bóng đá trở thành một phần không thể tách rời từ cuộc sống hàng ngày của hàng tỷ người trên toàn thế giới.

Với sự yêu thích và đón nhận mạnh mẽ như vậy, bóng đá không chỉ là trò chơi mà còn là cầu nối văn hóa kết nối mọi ngóc ngách của thế giới. Các giải đấu hàng đầu như World Cup, Premier League,… và các sự kiện thể thao tầm cỡ thế giới đã trở thành những điểm nhấn quan trọng trong lịch sử của từng quốc gia và đội bóng. Và chính vì những giải đấu này là nơi diễn ra những cuộc cạnh tranh khốc liệt giữa các đội bóng, việc lựa chọn cầu thủ phù hợp đã trở thành yếu tố không thể thiếu, đóng góp quan trọng vào thành công và danh tiếng của mỗi đội. Sự thành công của bất kỳ trận đấu bóng đá nào đều nằm ở việc lựa chọn cầu thủ, đây là một quyết định khó khăn đối với các tuyển trạch viên của mỗi đội bóng. Quá trình này không chỉ đòi hỏi sự nhạy bén với kỹ thuật và tình hình trận đấu mà còn đòi hỏi sự hiểu biết sâu sắc về các yếu tố như kỹ năng cá nhân, thống kê hiệu suất, sự kết hợp giữa các cầu thủ, thể lực, yếu tố tâm lý và chấn thương. Thị trường chuyển nhượng ngày càng trở nên phức tạp với sự tăng cường của nguồn lực tài chính, sự cạnh tranh giữa các đội bóng lớn, và sự tăng giá của các cầu thủ tiêu biểu. Quản lý chuyển nhượng không chỉ đòi hỏi sự hiểu biết vững về bóng đá mà còn đòi hỏi khả năng phân tích và đánh giá dữ liệu một cách chính xác. Giá trị chuyển nhượng của một cầu thủ ảnh hưởng trực tiếp đến cơ cấu đội hình và chiến lược của đội bóng. Việc đưa ra quyết định chuyển nhượng chính xác có thể tối ưu hóa hiệu suất trận đấu và đồng thời duy trì sự ổn định tài chính của đội.

Thị trường chuyển nhượng cầu thủ bóng đá chắc chắn đã trở thành một ngành kinh doanh lớn, và với số tiền lớn như vậy được chi cho một số lượng tài sản tương đối nhỏ sẽ có rủi ro cao. Sai lầm là chuyện thường tình và các câu lạc bộ thường rơi vào tình thế khó khăn khi phải trả hàng chục triệu euro cho một cầu thủ không đáp ứng được kỳ vọng. Những sai lầm như vậy có thể gây ra hậu quả thảm khốc cho câu lạc bộ và người hâm mộ, bao gồm cả việc xuống hạng và phá sản. Chính vì lý do này, đồ án này sẽ sử dụng kỹ thuật học máy để xây dựng mô hình hỗ trợ ra quyết định trong quá trình lựa chọn cầu thủ cho một đội bóng đá chuyên nghiệp dựa vào các kỹ năng và đặc điểm của các cầu thủ. Với sự hỗ trợ của mô hình này không chỉ giúp nhà quản lý đưa ra quyết định thông minh và chiến lược mà còn giúp họ tối ưu hóa nguồn lực tài chính, tạo ra đội hình mạnh mẽ và cạnh tranh trên mọi mặt trận. Việc này không chỉ là quan trọng đối với sự thành công của đội bóng mà còn góp phần vào hấp dẫn và sức sống của môn thể thao vua này.

## Mục tiêu:

Mục tiêu chính của đồ án là phát triển một mô hình dự đoán giá trị chuyển nhượng của cầu thủ bóng đá, tập trung vào việc khám phá mức độ có thể dự đoán của giá trị chuyển nhượng dựa trên kỹ năng và đặc điểm cá nhân của từng cầu thủ. Dữ liệu sẽ được thu thập bao gồm thông tin chi tiết về các kỹ năng và các yếu tố quan trọng khác từ trang web Sofifa, cùng với dữ liệu về giá trị chuyển nhượng từ trang web Transfermarkt. Sau đó chúng tôi sẽ sử dụng dữ liệu này để xây dựng một mô hình học máy, có khả năng dự đoán giá trị chuyển nhượng của cầu thủ.

Quá trình nghiên cứu sẽ tập trung vào việc xác định mức độ ảnh hưởng của từng yếu tố, như kỹ năng chi tiết, hiệu suất trong trận đấu, và các chỉ số tâm lý, đối với giá trị chuyển nhượng. Mục tiêu cuối cùng là cung cấp một công cụ dự đoán linh hoạt, giúp nhà quản lý đội bóng và đội ngũ tuyển trạch viên có cái nhìn chi tiết về giá trị của cầu thủ dựa trên những yếu tố cụ thể, từ đó hỗ trợ quyết định chuyển nhượng thông minh và tối ưu hóa nguồn lực tài chính của đội. Dựa vào đó cũng sẽ khám phá và đánh giá tiềm năng của các phương pháp học máy trong việc ước tính giá trị chuyển nhượng, tạo ra sự đóng góp mới và nâng cao hiệu suất so với các phương pháp truyền thống.

## Đối tượng, phạm vi nghiên cứu:

### Đối tượng nghiên cứu:

Nghiên cứu sẽ tập trung vào cầu thủ bóng đá chuyên nghiệp, đặc biệt là những cá nhân có tầm ảnh hưởng lớn đối với thị trường chuyển nhượng. Điều này bao gồm cả các cầu thủ nổi tiếng và những tài năng mới nổi có tiềm năng phát triển. Trọng tâm của nghiên cứu sẽ được đặt vào việc đánh giá các đặc điểm và kỹ năng cá nhân của cầu thủ, nhằm mục đích hiểu rõ hơn về những yếu tố nào góp phần vào giá trị chuyển nhượng của họ. Phân tích sẽ được tiến hành trên cầu thủ từ các vị trí khác nhau trên sân, bao gồm tiền đạo, tiền vệ, và hậu vệ, để khám phá ảnh hưởng của từng vị trí đối với giá trị chuyển nhượng.

### Phạm vi nghiên cứu:

Phạm vi nghiên cứu sẽ được giới hạn trong một khoảng thời gian cụ thể, tập trung vào các mùa chuyển nhượng hoặc chu kỳ thương vụ quan trọng trong lịch sử giải đấu bóng đá lớn và đầy uy tín. Nghiên cứu sẽ chú trọng vào việc thu thập và phân tích dữ liệu liên quan đến kỹ năng cá nhân của cầu thủ, thông tin cá nhân và giá trị chuyển nhượng. Cụ thể, dữ liệu sẽ bao gồm thông tin chi tiết về kỹ năng của cầu thủ, các yếu tố tâm lý và dữ liệu về giá trị chuyển nhượng của 20 giải đấu hàng đầu, được xác định bởi tổng số tiền mà các đội bóng của giải đấu đó đầu tư vào thị trường chuyển nhượng trong khoảng thời gian xác định. Điều này sẽ giúp nghiên cứu theo dõi và đánh giá sự biến động của giá trị chuyển nhượng theo thời gian, cung cấp cái nhìn sâu sắc về các yếu tố ảnh hưởng đến thị trường chuyển nhượng trong ngữ cảnh của những giải đấu danh tiếng. Tiếp theo, sẽ triển khai các mô hình học máy như mô hình hồi quy để dự đoán giá trị chuyển nhượng dựa trên các biến số nghiên cứu. Điều này sẽ giúp xác định mối quan hệ giữa các yếu tố như kỹ năng, thông tin cá nhân và giá trị chuyển nhượng.

# **:** CÁC CÔNG CỤ VÀ KỸ THUẬT SỬ DỤNG TRONG BÀI

## Tổng quan lý thuyết về học máy:

Học máy là một lĩnh vực trong trí tuệ nhân tạo (AI) mà nó tập trung vào việc phát triển các phương pháp và thuật toán để máy tính có thể học từ dữ liệu mà không cần phải được lập trình một cách tường minh. Mục tiêu của học máy là xây dựng mô hình dự đoán và phân loại có khả năng tự cải thiện thông qua trải nghiệm. Học máy có thể ứng dụng hầu hết các loại dữ liệu từ cấu trúc (bảng số liệu kinh tế, số liệu nông nghiệp,…) và cả dữ liệu phi cấu trúc (hình ảnh, văn bản, tín hiệu âm thanh,…). Học máy được chia thành 3 loại chính: học có giám sát, học không giám sát và học tăng cường.

Học có giám sát (Supervised Learning) là một dạng học máy trong đó mô hình được huấn luyện trên dữ liệu có nhãn. Mục tiêu là dự đoán hoặc phân loại dữ liệu mới dựa trên mối liên kết giữa đầu vào và đầu ra đã biết. Sử dụng khi bạn muốn nhận dự đoán một kết quả đầu ra từ dữ liệu đầu vào và bạn có các cặp dữ liệu (đầu vào/đầu ra) tương ứng. Để xây dựng mô hình học có giám sát thường phải có sự nổ lực từ con người để gán nhãn cho tập dữ liệu.

Học không giám sát (Unsupervised Learning) không sử dụng dữ liệu có nhãn trong quá trình huấn luyện. Mô hình phải tự tìm ra cấu trúc hoặc mẫu trong dữ liệu mà không có sự hướng dẫn nào từ bên ngoài. Các nhiệm vụ thường gặp trong học không giám sát bao gồm phân loại tự nhiên (clustering) để nhóm các điểm dữ liệu có tính chất tương đồng và giảm chiều dữ liệu.

Học tăng cường (Reinforcement Learning) liên quan đến việc mô hình tương tác với một môi trường và tự cập nhật dự đoán và hành động dựa trên phản hồi từ môi trường. Mô hình học từ các tình huống trải nghiệm và tìm cách tối ưu hóa một chính sách hành động để đạt được mục tiêu cụ thể. Điều này thường được áp dụng trong các bài toán như điều khiển robot, chơi trò chơi, và quản lý tài chính.

Học máy có thể được áp dụng trong nhiều lĩnh vực khác nhau như y tế, tài chính, giáo dục, sản xuất, v.v. Học máy không chỉ là một công cụ mạnh mẽ để giải quyết các vấn đề thực tế mà còn mang lại những hiểu biết sâu rộng về mối quan hệ giữa dữ liệu và kiến thức, đóng góp quan trọng vào sự tiến bộ của trí tuệ nhân tạo và công nghệ thông tin.

## Các thuật toán hồi quy:

### Hồi quy rừng ngẫu nhiên (Random Forest Regression):

Random Forest Regression (RFR) là một phương pháp mạnh mẽ trong học máy, kết hợp sức mạnh của nhiều cây quyết định để thực hiện nhiệm vụ hồi quy. Phương pháp này sử dụng một kỹ thuật gọi là Bootstrap và Aggregation, hay còn được biết đến là bagging, để tạo ra nhiều cây quyết định độc lập nhau. Bagging cho phép lựa chọn ngẫu nhiên một nhóm nhỏ các thuộc tính tại mỗi nút của cây để phân chia dữ liệu, giúp mô hình có khả năng phân loại một cách linh hoạt và nhanh chóng.

Một đặc điểm quan trọng của RFR là khả năng chọn ngẫu nhiên một số lượng đặc trưng để sử dụng trong quá trình xây dựng mỗi cây. Điều này tăng tính đa dạng và linh hoạt của mô hình, đồng thời giảm nguy cơ quá khớp. Kết quả của RFR được xác định thông qua quá trình biểu quyết, nơi kết quả của từng cây được tổng hợp theo nguyên tắc đa số hoặc lấy trung bình, tùy thuộc vào loại bài toán là phân loại hay hồi quy. Đối với bài toán hồi quy, kết quả cuối cùng của mô hình RFR sẽ là trung bình của tất cả các kết quả dự báo của các cây (Hình 2-1).

Cách thuật toán RFR hoạt động có thể được tóm tắt như sau:

* Dùng phương pháp bootstrap để tạo ra nhiều tập dữ liệu con, mỗi tập con được tạo ngẫu nhiên từ tập dữ liệu huấn luyện với việc thay thế mẫu để tạo tính đa dạng.
* Mỗi tập dữ liệu con được sử dụng để xây dựng một cây hồi quy độc lập. Quá trình này tối ưu hóa tiêu chuẩn chia nút để tạo cây hiệu quả.
* Kết quả dự đoán cuối cùng được tính toán bằng cách biểu quyết kết quả từ tất cả các cây hồi quy. Đối với bài toán hồi quy, kết quả cuối cùng là giá trị trung bình của các dự đoán từ các cây như công thức sau:

(2‑1)

Trong đó:

* là giá trị dự đoán cuối cùng.
* N là số lượng cây hồi quy.
* là dự đoán của cây thứ

A diagram of a tree

Description automatically generated

Hình 2‑1: Sơ đồ biểu diễn cách hoạt động của mô hình Random Forest Regression

### Hồi quy XGBoost Regression (Extreme Gradient Boosting Regression):

XGBoost (Extreme Gradient Boosting), là một thuật toán học máy hàng đầu trong lĩnh vực cây quyết định, xây dựng trên khung tăng cường độ dốc. Phát triển ban đầu bởi Tianqi Chen và Carlos Guestrin tại Đại học Washington, XGBoost đã đem lại sự đột phá trong lĩnh vực Machine Learning khi được giới thiệu tại Hội nghị SIGKDD (Special Interest Group on Knowledge Discovery and Data Mining) vào năm 2016. Ban đầu nhắm đến giải quyết các bài toán phân loại và hồi quy, XGBoost đã nhanh chóng trở thành một trong những thuật toán phổ biến nhất.

XGBoost thuộc họ thuật toán "boosting," trong đó các mô hình được xây dựng theo kiểu tuần tự để cải thiện từng bước và khắc phục sai lệch của mô hình trước đó. Thuật toán chủ yếu tập trung vào việc tối ưu hóa hàm mất mát thông qua phương pháp Gradient Descent.

Các tối ưu và cải tiến của XGBoost dựa trên khung Gradient Boosting Model (GBM) bao gồm (Hình 2-2):

* Song Song Hóa: XGBoost sử dụng song song hóa để tiếp cận quá trình xây dựng cây một cách hiệu quả. Điều này thực hiện được bằng cách lồng các vòng lặp và sắp xếp chúng một cách song song để giảm thời gian chạy.
* Tỉa Cây: Sử dụng tham số 'max\_depth' để cắt tỉa cây từ phía sau, cải thiện độ hiệu quả tính toán.
* Tối Ưu Hóa Phần Cứng: XGBoost được tối ưu hóa cho việc sử dụng tài nguyên phần cứng thông qua việc nhận thức bộ nhớ cache và phân bổ bộ đệm nội bộ.
* Tránh Overfitting: Thêm vào hàm mất mát một thành phần chặn để kiểm soát độ lớn của trọng số và giảm thiểu nguy cơ quá khớp, sử dụng alpha và lambda làm tham số kiểm soát.
* Xử Lý Dữ Liệu Thiếu: XGBoost tự động xử lý tính năng thiếu thốn bằng cách 'học' giá trị tốt nhất từ sự mất mát trong quá trình đào tạo.
* Xác Thực Chéo: Phương pháp xác thực chéo được tích hợp sẵn giúp quá trình đào tạo được kiểm định một cách hiệu quả.

A diagram of a diagram

Description automatically generated

Hình 2‑2: Các tối ưu hóa thuật toán của XGBoost so với GBM tiêu chuẩn

Công thức của XGBoost Regression được biểu diễn bằng công thức sau:

(2‑2)

Trong đó:

* ​ là giá trị dự đoán cho mẫu thứ .
* K là số lượng cây hồi quy.
* là dự đoán của cây thứ *k* cho mẫu thứ *i*.

Cách thuật toán hoạt động có thể tóm tắt như sau:

* Tạo cây quyết định (Decision Trees): Mỗi cây quyết định được xây dựng bằng cách chia dữ liệu thành các nhóm dựa trên các đặc trưng của dữ liệu. XGBoost sử dụng cây quyết định có cấp độ sâu thấp, thường chỉ vài chục nút lá, để tránh quá mức phức tạp và giữ cho mô hình tốt hơn trong việc tổng hợp thông tin.
* Tối ưu hóa hàm mất mát (Loss Function Optimization): Mục tiêu là tối thiểu hóa hàm mất mát, là sự chênh lệch giữa giá trị dự đoán và giá trị thực tế. Hàm mất mát thường là một hàm phi tuyến, và XGBoost sử dụng kỹ thuật Gradient Boosting để cập nhật dự đoán mô hình sao cho mất mát giảm thiểu.
* Regularization để tránh overfitting, XGBoost thêm các thành phần chặn vào hàm mất mát, bao gồm cả l1 (Lasso) và l2 (Ridge) regularization. Các thành phần này giúp kiểm soát trọng số của cây, ngăn chặn chúng trở nên quá phức tạp và giảm khả năng tổng hợp dữ liệu nhiễu.
* Quyết định cây tiếp theo dựa trên Gradient: XGBoost không xây dựng tất cả các cây cùng một lúc, mà là từng cây một. Sau khi xây dựng một cây, nó sẽ tính gradient của hàm mất mát đối với dự đoán hiện tại và sử dụng gradient này để xây dựng cây tiếp theo.
* Tích hợp các cây để tạo mô hình cuối cùng: Khi tất cả các cây đã được xây dựng, dự đoán cuối cùng của mô hình là tổng của các dự đoán từ tất cả các cây.
* Learning Rate (Tốc độ học): XGBoost có một tham số được gọi là "learning rate" để kiểm soát mức độ cập nhật của dự đoán sau mỗi cây được thêm vào. Learning rate giúp kiểm soát tốc độ học và tránh quá mức điều chỉnh mô hình.
* Cross-validation và Early Stopping: XGBoost thường sử dụng kỹ thuật cross-validation để đánh giá hiệu suất của mô hình. Early stopping được sử dụng để dừng quá trình huấn luyện khi không có sự cải thiện đáng kể nữa, giúp giảm thời gian và nguồn lực.

### Hồi quy LightGBM ( Light Gradient Boosting Machine Regression):

LightGBM là một thuật toán học máy thuộc dòng Gradient Boosting Frameworks, đã đem lại đột phá đáng kể trong việc xử lý dữ liệu lớn và tăng cường hiệu suất của các mô hình. Được Microsoft phát triển và cho ra mắt bản thử nghiệm vào tháng 1 năm 2016, và nhanh chóng trở thành thuật toán ensemble được ưa chuộng nhất.

LightGBM phát triển cây theo lá dựa trên "leaf-wise tree growth", trong khi hầu hết các thuật toán ensemble khác dựa trên phương pháp tang trưởng cây theo cấp "level (depth)-wise tree growth". "Leaf-wise tree growth" lựa chọn nút để phát triển cây dựa trên tối ưu toàn bộ tree, trong khi "level (depth)-wise tree growth" tối ưu trên nhánh đang xét, do đó với số node nhỏ, các cây xây dựng từ "leaf-wise tree growth" thường thực hiện tốt hơn "level (depth)-wise tree growth" (Hình 2-3).

A close-up of a diagram

Description automatically generated

Hình 2‑3: Sơ đồ hoạt động của LightGBM

LightGBM sử dụng "histogram-based algorithms" thay vì "pre-sort-based algorithms" thường được sử dụng trong các công cụ boosting khác để tìm "split point" trong quá trình xây dựng cây. Cải tiến này giúp LightGBM tăng tốc quá trình đào tạo và giảm lượng bộ nhớ cần thiết. LightGBM cũng sử dụng hai thuật toán đặc biệt là GOSS (Gradient Based One Side Sampling) và EFB (Exclusive Feature Bundling), đồng thời tối ưu hóa quá trình tính toán

Công thức của LightGBM Regression được biểu diễn bằng công thức sau:

(2‑3)

Trong đó:

* ​ là giá trị dự đoán cho mẫu thứ .
* K là số lượng cây hồi quy.
* là dự đoán của cây thứ *k* cho mẫu thứ *i*.

Cách thuật toán hoạt động có thể tóm tắt như sau:

* LightGBM sử dụng phương pháp "leaf-wise tree growth," làm tối ưu toàn bộ cây dựa trên các nút lá thay vì theo cấp "level (depth)-wise tree growth" như nhiều thuật toán khác. Cách tiếp cận này giúp tối ưu hóa hiệu suất của cây với số lượng node ít hơn, đặc biệt là hiệu quả với dữ liệu có kích thước lớn.
* LightGBM sử dụng "histogram-based algorithms" thay vì "pre-sort-based algorithms" để tìm kiếm "split points" khi xây dựng cây. Cải tiến này giúp tăng tốc quá trình đào tạo và giảm lượng bộ nhớ cần sử dụng bằng cách sử dụng các biểu đồ histogram để nhanh chóng tính toán các giá trị tối ưu.
* Sử dụng thuật toán GOSS (Gradient Based One Side Sampling) để tối ưu hóa quá trình lấy mẫu (sampling) bằng cách tập trung vào các mẫu có gradient lớn, giảm kích thước dữ liệu được sử dụng trong quá trình đào tạo.
* LightGBM sử dụng EFB (Exclusive Feature Bundling) để kết hợp các đặc trưng một cách độc quyền, giảm độ phức tạp của cây và tăng tốc độ tính toán. Việc này giúp giảm số lượng cây cần xây dựng và tăng hiệu suất.

## Các phương pháp đánh giá mô hình:

Để đánh giá mức độ dự đoán chính xác của mô hình, bốn độ đo sau được sử dụng: MAE (**Mean Absolute Error)**, MSE (Mean Squared Error), RMSE (**Root Mean Squared Error)**, R2 (**Coefficient of Determination).**

MSE (Mean Squared Error): Đây là trung bình giá trị bình phương của sai số giữa giá trị dự đoán và giá trị thực tế. MSE được sử dụng để đo lường độ lỗi trung bình của mô hình. MSE càng thấp thì mô hình càng chính xác. MSE có thể bị ảnh hưởng bởi các giá trị ngoại lai.

(2‑1)

Trong đó:

* N là số lượng điểm dữ liệu.
* là giá trị thực tế.
* là giá trị dự đoán.

MAE (Mean Absolute Error): Đo độ lớn trung bình các lỗi trong một tập hợp các dự đoán mà không cần xem xét hướng của chúng. Đó là giá trị trung bình trên tập mẫu kiểm tra về sự khác biệt tuyệt đối giữa dự đoán và quan sát thực tế, trong đó tất cả các khác biệt riêng lẻ có trọng số bằng nhau.

(2‑2)

Trong đó:

* N là số lượng điểm dữ liệu.
* là giá trị thực tế.
* là giá trị dự đoán.

MAE luôn không âm và giá trị 0 (hầu như không bao giờ đạt được trong thực tế) sẽ chỉ ra sự phù hợp hoàn hảo với dữ liệu. MAE càng thấp thì kết quả dự báo càng gần với thực tế.

R2 (Coefficient of Determination): Đây là độ đo đánh giá mức độ giải thích của mô hình hồi quy. R2 có giá trị từ 0 đến 1, trong đó 1 là một mô hình hoàn hảo và 0 là một mô hình không có khả năng giải thích. R2 càng gần 1 thì mô hình càng tốt.

(2‑3)

Trong đó:

* là tổng bình phương sai số (Sum of Squared Errors), tức là tổng của bình phương hiệu giữa giá trị dự đoán và giá trị thực tế
* là tổng bình phương độ lệch (Sum of Squared Total), tức là tổng của bình phương hiệu giữa giá trị thực tế và giá trị trung bình của biến phụ thuộc.

RMSE (Root Mean Squared Error): Đây là căn bậc hai của trung bình giá trị bình phương của sai số giữa giá trị dự đoán và giá trị thực tế. RMSE được sử dụng để đo lường độ lỗi trung bình của mô hình. RMSE càng thấp thì mô hình càng chính xác. RMSE có thể được sử dụng để so sánh các mô hình khác nhau.

(2‑4)

## Các công cụ sử dụng trong bài:

### CloudScraper:

CloudScraper là một thư viện Python dựa trên Python Requests, được thiết kế để vượt qua hệ thống bảo vệ chống bot của Cloudflare. Điều này cho phép thu thập dữ liệu từ các trang web triển khai trên nền tảng Cloudflare, nơi thường áp dụng các biện pháp bảo vệ bot. Thư viện này sử dụng các kỹ thuật mô phỏng hành vi trình duyệt, bao gồm chạy trình duyệt headless, thực hiện JavaScript, và giả mạo User-Agent để mô phỏng trình duyệt thực sự và tránh được các biện pháp bảo vệ bot.

### BeautifulSoup:

Beautiful Soup là một thư viện Python mạnh mẽ giúp phân tích cú pháp HTML và XML một cách linh hoạt. Nó tạo cây cú pháp từ nguồn HTML hoặc XML, cung cấp công cụ để điều hướng và trích xuất thông tin. Thư viện này cho phép tìm kiếm thẻ, thuộc tính, truy cập nội dung thẻ, và tìm kiếm đoạn văn bản cụ thể. Beautiful Soup thích hợp để trích xuất dữ liệu từ trang web và lưu trữ nó trong nhiều định dạng khác nhau như CSV, Excel, JSON, hoặc SQL để phân tích hoặc xây dựng ứng dụng. Điều đặc biệt là nó linh hoạt và có thể sử dụng cho nhiều ngôn ngữ đánh dấu, không chỉ riêng HTML hoặc XML.

### Lập trình bất đồng bộ (Asynchronous):

Lập trình bất đồng bộ (asynchronous programming) là một phong cách lập trình mà không yêu cầu đợi đến khi một tác vụ kết thúc trước khi bắt đầu thực hiện tác vụ khác. Thay vào đó, nó tận dụng khả năng chạy nhiều tác vụ đồng thời mà không chặn quá trình thực thi chương trình chính (Hình 2-4). Các ngôn ngữ lập trình hiện đại như Python, JavaScript, và Kotlin hỗ trợ lập trình bất đồng bộ thông qua các khái niệm như coroutine, promise, asyncio và async/await.

A screenshot of a computer

Description automatically generated

Hình 2‑4: Cách hoạt động của lập trình bất đồng bộ và đồng bộ

Dưới đây là các khái niệm chính của bất đồng bộ trong Python:

* Asyncio: asyncio là một thư viện trong Python cung cấp cơ chế để viết mã bất đồng bộ sử dụng các coroutine, event loops, và các công cụ khác để quản lý việc thực thi các tác vụ không đồng bộ.
* Async: async là một từ khóa được sử dụng để định nghĩa một hàm coroutine. Khi một hàm được đánh dấu bằng từ khóa async, nó trở thành một coroutine, có thể được gọi mà không làm chương trình bị chặn và có thể chờ đợi (await) các tác vụ không đồng bộ.
* Await: await là một từ khóa được sử dụng bên trong một coroutine để đợi một tác vụ không đồng bộ hoàn thành. Khi gặp từ khóa await, coroutine hiện tại sẽ tạm thời chuyển quyền thực thi cho một tác vụ khác mà không làm chương trình bị chặn.
* Coroutine: Coroutine là một loại hàm đặc biệt trong Python được đánh dấu bằng từ khóa async. Hàm coroutine có khả năng tạm thời dừng thực thi và chuyển quyền thực thi cho các coroutine khác thông qua await. Các coroutine thường được sử dụng để xử lý các tác vụ không đồng bộ, chẳng hạn như gọi API mạng, xử lý sự kiện, hoặc thực hiện các tác vụ đòi hỏi đồng thời.

Cách chúng hoạt động với nhau:

* Coroutine và từ khóa async: Khi bạn định nghĩa một hàm với từ khóa async, đó trở thành một coroutine. Coroutine có khả năng chờ đợi các tác vụ không đồng bộ mà không làm chương trình bị chặn.
* Từ khóa await: Trong một coroutine, từ khóa await được sử dụng để chờ đợi kết quả của một tác vụ không đồng bộ. Khi gặp từ khóa await, quyền điều khiển tạm thời chuyển cho event loop để thực thi các tác vụ khác.
* Event Loop và asyncio.run(): asyncio sử dụng event loop để quản lý việc thực thi các coroutine. Event loop là một vòng lặp vô hạn chạy qua danh sách các tác vụ, kiểm tra xem chúng đã hoàn thành hay chưa, và chuyển đến các tác vụ khác khi cần thiết. asyncio.run() là một hàm chạy một coroutine trong event loop. Nó chạy coroutine chính (main coroutine) và quản lý tất cả các coroutine con.

A diagram of a program

Description automatically generated

Hình 2‑5: Cách hoạt động của bất đồng bộ với python

### Biểu thức chính quy (****regular expression):****

Regular Expression (Regex hoặc RegExp) là chuỗi ký tự đặc biệt sử dụng để mô tả mẫu xâu ký tự, là công cụ mạnh mẽ trong xử lý chuỗi cho việc tìm kiếm, so sánh, và thay thế dựa trên quy tắc cụ thể. Regex cung cấp ngôn ngữ mô tả mẫu linh hoạt và có thể sử dụng trong nhiều ngôn ngữ lập trình như Python, JavaScript, Java, C#, và nhiều ngôn ngữ khác.

### Các thư viện của Python:

Pandas: là một thư viện Python cung cấp các cấu trúc dữ liệu và công cụ phân tích dữ liệu mạnh mẽ, đặc biệt là DataFrame. DataFrame là một bảng dữ liệu hai chiều với các hàng và cột được đặt tên, cho phép thực hiện nhanh chóng các thao tác như lọc, chọn, và xử lý dữ liệu. Pandas cũng hỗ trợ đọc và ghi dữ liệu từ nhiều định dạng như CSV, Excel, SQL databases.

NumPy: là thư viện Python cho tính toán số học và thống kê, cung cấp đối tượng mảng nhiều chiều (ndarray) linh hoạt. NumPy giúp thực hiện các phép toán số học nhanh chóng trên dữ liệu mảng, làm cho việc xử lý và tính toán trên dữ liệu lớn trở nên hiệu quả. NumPy còn chứa các chức năng linh hoạt cho đại số tuyến tính, biến đổi Fourier, và nhiều tính năng khác.

Matplotlib: là một thư viện vẽ đồ thị và biểu đồ 2D cho Python, cho phép hiển thị dữ liệu một cách trực quan. Nó cung cấp giao diện để tạo các biểu đồ, đồ thị đường, đồ thị phức tạp, và nhiều loại biểu đồ khác. Matplotlib giúp tạo ra đồ thị chất lượng cao, có thể tùy chỉnh đồ họa và chú thích để phản ánh thông tin một cách rõ ràng.

Seaborn: là một thư viện trực quan hóa dữ liệu dựa trên Matplotlib, tối ưu hóa cho việc tạo ra các biểu đồ thống kê hấp dẫn và dễ hiểu. Seaborn cung cấp các giao diện đơn giản để vẽ các biểu đồ phức tạp như biểu đồ violin, biểu đồ hộp, và biểu đồ phân phối. Nó cũng có khả năng tương tác mạnh mẽ với Pandas DataFrame, giúp dễ dàng thực hiện phân tích thống kê và thăm dò dữ liệu.

### Chuẩn hóa và mã hóa dữ liệu:

Chuẩn hóa là quá trình biến đổi dữ liệu về một phạm vi chuẩn để đảm bảo rằng các đặc trưng của dữ liệu có cùng một đơn vị đo. Mục tiêu của chuẩn hóa là giảm biên độ của dữ liệu và đưa chúng về một khoảng giá trị chung. Các phương pháp chuẩn hóa thường được áp dụng khi các đặc trưng của dữ liệu có phạm vi giá trị quá lớn hoặc quá nhỏ, và khi sự chênh lệch này có thể ảnh hưởng đến hiệu suất của các mô hình máy học.

StandardScaler là một phương pháp chuẩn hóa phổ biến. Nó thực hiện việc chuyển đổi dữ liệu sao cho giá trị trung bình của mỗi đặc trưng là 0 và độ lệch chuẩn là 1. Công thức chuẩn hóa z-score

(2‑4)

Trong đó:

* 𝑧 là giá trị tiêu chuẩn hóa
* 𝑋 là giá trị ban đầu
* 𝜇 là giá trị trung bình của một biến
* 𝜎 là độ lệch chuẩn của một biến

Mã hóa là quá trình chuyển đổi các biến phân loại hoặc chuỗi thành các giá trị số, giúp mô hình máy học có thể hiểu và xử lý chúng. Các biến phân loại thường không thể sử dụng trực tiếp trong các mô hình, nên việc mã hóa chúng trở thành bước quan trọng trong xử lý dữ liệu.

LabelEncoder là một công cụ mã hóa sự thứ bậc của các biến phân loại thành các giá trị số nguyên. Đối với mỗi giá trị duy nhất trong biến, LabelEncoder gán một số nguyên duy nhất tương ứng. Mã hóa này thích hợp cho biến mục tiêu (target variable) trong các mô hình phân loại. Tuy nhiên, nó cũng có thể được sử dụng cho biến độc lập nếu biến đó có thứ bậc tự nhiên.

### Tối ưu siêu tham số:

Bayesian Search là một phương pháp tối ưu hóa siêu tham số (hyperparameter) trong quá trình đào tạo mô hình máy học. Nó dựa trên ý tưởng của Bayes Optimization, một phương pháp tối ưu hóa toàn diện, linh hoạt và hiệu quả. Trong ngữ cảnh của việc tinh chỉnh hyperparameter, phương pháp này thường được gọi là Bayesian Hyperparameter Optimization hoặc Bayesian Search for Hyperparameter Tuning.

Cách Hoạt Động:

* Mô hình ước lượng số đối tượng (Surrogate Model): Bayesian Search sử dụng một mô hình ước lượng số đối tượng để xấp xỉ hàm mất mát (loss function) tại các điểm chưa biết của không gian siêu tham số. Gaussian Process là một trong những mô hình phổ biến được sử dụng cho mục đích này.
* Hàm đánh giá (Acquisition Function): Một hàm đánh giá được sử dụng để chọn ra điểm kế tiếp để đánh giá. Hàm này kết hợp giữa sự cân nhắc giữa việc khám phá các khu vực chưa biết và khai thác các khu vực mà chúng ta có đủ thông tin.
* Quá trình lựa chọn và cập nhật mô hình: Dựa vào hàm đánh giá, một điểm mới được chọn để đánh giá. Sau đó, mô hình ước lượng số đối tượng được cập nhật với thông tin mới. Quá trình này lặp đi lặp lại cho đến khi đạt được sự hội tụ hoặc đã đạt đến số lượng lần lặp quy định.

### Giao diện:

Tkinter là một thư viện GUI (Graphical User Interface) tích hợp sẵn trong Python, cung cấp các công cụ và widgets để xây dựng các giao diện người dùng đồ họa đơn giản. Tkinter chủ yếu dựa trên thư viện GUI Tk (Tcl/Tk) và là một phần của thư viện tiêu chuẩn của Python. Tkinter cung cấp cơ chế xử lý sự kiện thông qua việc sử dụng các hàm callback. Khi một sự kiện xảy ra, hàm được gọi để xử lý sự kiện đó.

# : ỨNG DỤNG PHƯƠNG PHÁP VÀ XÂY DỰNG MÔ HÌNH

## Phân tích bài toán và lựa chọn dữ liệu:

### Mô tả bài toán:

Để đội bóng hoạt động tốt nhất, các đội bóng thường tìm kiếm những cầu thủ mới có nhiều tiềm năng, hứa hẹn có khả năng phát triển để đáp ứng mục tiêu và định hình tương lai của đội. Có ba lựa chọn khác nhau để ký hợp đồng với một cầu thủ mới, đó là đi mượn từ các đội bóng khác, ký hợp đồng với cầu thủ tự do hoặc mua một cầu thủ. Nghiên cứu này tập trung vào lựa chọn mua một cầu thủ, có nghĩa là một câu lạc bộ bóng đá trả tiền bồi thường cho một câu lạc bộ khác để có thể có cầu thủ đó chơi cho họ, khoản bồi thường này được gọi là phí chuyển nhượng.

Việc đạt được sự đồng thuận về mức phí chuyển nhượng giữa câu lạc bộ mua và bán là quan trọng để duy trì cân bằng hợp lý. Sự phù hợp giữa số tiền thực tế đã trả và giá trị được đánh giá của cầu thủ là quyết định quan trọng, giúp tránh rủi ro thoái vốn lớn khi một cầu thủ không đáp ứng kỳ vọng. Nghiên cứu này nhằm mục đích khám phá khả năng dự đoán giá trị chuyển nhượng của cầu thủ, một thông tin quan trọng mà câu lạc bộ có thể sử dụng trong quá trình đàm phán, cũng như mang lại những đánh giá về giá trị của một cầu thủ trong cộng đồng bóng đá.

Có rất nhiều yếu tố để có thể định giá một cầu thủ bóng đá có thể kể đến như hiệu suất trên sân, dựa vào sự cạnh tranh của thị trường chuyển nhượng, các quy tắc của hợp đồng, ảnh hưởng của truyền thông,… nhưng nghiên cứu này đặt ra mục tiêu là xác định mức độ có thể dự đoán giá trị chuyển nhượng của cầu thủ bóng đá dựa trên kỹ năng và đặc điểm cá nhân của họ. Câu hỏi nghiên cứu chính là: "Giá trị chuyển nhượng của các cầu thủ có thể được dự đoán ở mức độ nào dựa trên kỹ năng và đặc điểm cá nhân của họ?"

### Dữ liệu:

#### Lựa chọn dữ liệu cầu thủ:

Hệ thống trò chơi điện tử EA Sports FIFA được phát triển bởi Electronic Arts (EA). Nó cung cấp sẵn các thông tin cá nhân, về các kỹ năng chơi bóng và đặc điểm của hơn 17,000 cầu thủ bóng đá trên thế giới. Thông tin này có sẵn trên trang web chính thức của họ. EA Sports tích hợp một mạng lưới tuyển trạch viên ngoài đời thực rộng lớn, tham gia trực tiếp các sự kiện và xem trực tiếp các trận đấu để đánh giá và cập nhật điểm kỹ năng cho từng người chơi trong FIFA. Thực tế, điểm số của cầu thủ có thể biến động theo mùa giải thực tế. Ví dụ: nếu một cầu thủ trẻ thể hiện sự xuất sắc trong việc sút và rê bóng, điểm của anh ấy sẽ tăng trong phiên bản tiếp theo của FIFA. Ngược lại, nếu một cầu thủ lớn tuổi mất đi tốc độ hoặc thể lực, điều này cũng sẽ được phản ánh qua điểm số [1]. Quá trình này giúp đảm bảo tính mạch lạc và tính đại diện cho bộ dữ liệu.

Các nghiên cứu trước đây về giá trị của các cầu thủ bóng đá chuyên nghiệp, dữ liệu từ các trò chơi điện tử đã được sử dụng.

* Nghiên cứu của Kirschstein và Liebscher (2019) [2] tập trung vào việc sử dụng kỹ thuật học máy để dự đoán giá trị thị trường của cầu thủ bóng đá. Họ đã phát triển một mô hình dựa trên các biến số kỹ năng từ FIFA 16 bằng cách so sánh với giá trị thị trường thực tế từ transfermarkt.com, nghiên cứu này cung cấp thông tin quan trọng về khả năng dự đoán giá trị cầu thủ thông qua mô hình học máy và dữ liệu từ trò chơi FIFA.
* Trong nghiên cứu của Yiğit et al. (2020) [3], họ đã sử dụng mô hình học máy, bao gồm Random Forests, Gradient Boosting, và Ridge And Lasso Regressions, dựa trên kỹ năng của cầu thủ trong trò chơi điện tử Football Manager để dự đoán giá trị thị trường. Kết quả chỉ ra rằng việc kết hợp dữ liệu từ Football Manager với giá trị thị trường từ transfermarkt.com cung cấp dự đoán chính xác hơn về giá trị cầu thủ.
* Behravan và Razavi (2021) [4] đã sử dụng dữ liệu FIFA 20 để dự đoán giá trị thị trường, với nhận định rằng vị trí của cầu thủ trên sân cần được xem xét. Họ cũng nhấn mạnh tầm quan trọng của xếp hạng tổng thể, cho biết nó phản ánh trình độ kỹ năng hiện tại của người chơi.

Dựa trên các dẫn chứng và các nghiên cứu đã nêu trên, có thể kết luận rằng việc sử dụng dữ liệu cầu thủ từ trò chơi điện tử EA Sports FIFA là một quyết định có lý. Các nghiên cứu này đã chứng minh rằng thông tin từ trò chơi này, đặc biệt là về kỹ năng và đánh giá của cầu thủ, có thể đóng vai trò quan trọng trong việc dự đoán giá trị thị trường của họ. Sự kết hợp giữa dữ liệu từ FIFA và các thông tin thị trường thực tế mang lại kết quả dự đoán chính xác hơn, cũng như sự cân nhắc đối với các yếu tố khác. Do đó, sử dụng dữ liệu cầu thủ từ trò chơi EA Sports FIFA có thể cung cấp thông tin độc đáo và hữu ích để đánh giá giá trị và hiệu suất của cầu thủ bóng đá.

#### Mô tả dữ liệu cầu thủ:

Bộ dữ liệu này được thu thập từ trang web sofifa.com từ mùa giải 2014/2015 – 2021/2022 (tương ứng với FIFA15 – FIFA22). Bộ dữ liệu SOFIFA bao gồm các thuộc tính về khả năng, hồ sơ và vị trí như một loại thuộc tính trong trò chơi được đo lường định lượng dựa trên hiệu suất thực tế của cầu thủ. Thuộc tính kỹ năng được mô tả bằng dữ liệu hiển thị các số liệu thống kê liên quan đến bóng đá của cầu thủ từ 1 đến 99 (Bảng 3-2), với tổng cộng 34 thuộc tính về khả năng. Ngoài ra, nghiên cứu này phân loại 34 thuộc tính về kỹ năng thành bảy loại thuộc tính khả năng như "attacking", "skill", "movement", "power", "mentality", "defending", và "goalkeeping" (Hình 3-1). Dữ liệu này cũng cung cấp các thuộc tính hồ sơ là dữ liệu thực tế của cầu thủ bóng đá bao gồm như: tên, tuổi, quốc tịch, chiều cao, cân nặng, câu lạc bộ đang chơi,… (Bảng 3-3). Bộ dữ liệu SOFIFA thường được cung cấp từ ít nhất một đến ba vị trí, xem xét lịch sử vị trí thi đấu thực tế của cầu thủ, chủ yếu là một vị trí của thủ môn, và lên đến ba vị trí phù hợp cho tiền đạo, hậu vệ và tiền vệ (Bảng 3-1).

|  |  |  |  |
| --- | --- | --- | --- |
| **Loại vị trí** | **Vị trị viết tắt** | **Tên vị trí** | **Tên tiếng việt** |
| Attacker | ST | Striker | Tiền đạo |
| CF | Centre Forward | Tiền đạo trung tâm |
| LW | Left Winger | Tiền đạo cánh trái |
| RW | Right Winger | Tiền đạo cánh phải |
| Midfielder | CAM | Centre Attacking Midfielder | Tiền vệ trung tâm tấn công |
| CM | Central Midfielder | Tiền vệ trung tâm |
| LM | Left Midfielder | Tiền vệ cánh trái |
| RM | Right Midfielder | Tiền vệ cánh phải |
| CDM | Central Defensive Midfielder | Tiền vệ trung tâm phòng ngự |
| Defender | CB | Central Back | Trung vệ trung tâm |
| LB | Left Back | Hậu vệ cánh trái |
| RB | Right Back | Hậu vệ cánh phải |
| Goalkeeper | GK | Goalkeeper | Thủ môn |

Bảng 3‑1: Mô tả các vị trí của cầu thủ



Hình 3‑1: Dữ liệu cầu thủ trên web SoFIFA

|  |  |  |  |
| --- | --- | --- | --- |
| **Loại kỹ năng** | **Thuộc tính chi tiết** | **Mô tả** | **Phạm vi giá trị** |
| Attack | Crossing | Chuyền bóng | 1-99 |
| Finishing | Kỹ thuật dứt điểm | 1-99 |
| Heading Accuracy | Độ chính xác đánh đầu | 1-99 |
| Short Passing | Chuyền ngắn | 1-99 |
| Volleys | Kỹ thuật vô lê | 1-99 |
| Skill | Dribbling | Dẫn bóng | 1-99 |
| Curve | Độ cong khi chuyền | 1-99 |
| Free Kick Accuracy | Độ chính xác sút phạt | 1-99 |
| Long Passing | Chuyền xa | 1-99 |
| Ball Control | Kiểm soát bóng | 1-99 |
| Movement | Acceleartion | Tăng tốc | 1-99 |
| Agility | Linh hoạt | 1-99 |
| Sprint Speed | Tốc độ nhanh | 1-99 |
| Reactions | Phản ứng | 1-99 |
| Balance | Cân bằng | 1-99 |
| Power | Shot Power | Lực sút | 1-99 |
| Jumping | Nhảy cao | 1-99 |
| Stamina | Thể lực | 1-99 |
| Strength | Sức mạnh | 1-99 |
| Long Shots | Sút xa | 1-99 |
| Mentality | Aggression | Quyết tâm | 1-99 |
| Penalties | Phạt đền | 1-99 |
| Positioning | Vị trí | 1-99 |
| Interceptions | Chặn bóng | 1-99 |
| Vision | Tầm nhìn | 1-99 |
| Composure | Bình tĩnh | 1-99 |
| Defending | Marking | Theo kèm đối thủ | 1-99 |
| Sliding Tackle | Pha vào bóng trượt | 1-99 |
| Standing Tackle | Pha vào bóng đứng | 1-99 |
| Goalkeeping | Positioning | Chọn vị trí | 1-99 |
| Diving | Ném mình | 1-99 |
| Handling | Xử lý bóng | 1-99 |
| Kicking | Đá bóng | 1-99 |
| Reflexes | Phản xạ | 1-99 |

Bảng 3‑2: Mô tả phạm vi giá trị của các thuộc tính kỹ năng

|  |  |
| --- | --- |
| **Tên thuộc tính** | **Mô tả thuộc tính** |
| sofifa\_id | ID của sofifa.com |
| player\_url | URL tới hồ sơ người chơi trên sofifa.com |
| short\_name | Tên viết tắt của cầu thủ |
| long\_name | Tên đầy đủ của cầu thủ |
| age | Tuổi của cầu thủ |
| dob | Ngày sinh của cầu thủ |
| height\_cm | Chiều cao của cầu thủ tính bằng cm |
| weight\_kg | Trọng lượng của cầu thủ tính bằng kg |
| nationality | Quốc tịch của cầu thủ |
| club\_name | Tên câu lạc bộ của cầu thủ |
| league\_name | Giải đấu mà câu lạc bộ tham gia |
| league\_rank | Thứ hạng của giải đấu mà câu lạc bộ tham gia |
| overall | Đánh giá chung của cầu thủ |
| potential | Đánh giá tiềm năng, tương lai của cầu thủ |
| value\_eur | Giá trị cầu thủ theo FIFA |
| wage\_eur | Mức lương cầu thủ theo FIFA |
| player\_positions | Vị trí của cầu thủ |
| preferred\_foot | Chân thuận của cầu thủ |
| international\_reputation | Danh tiếng quốc tế của cầu thủ |
| weak\_foot | Mức độ chân yếu của cầu thủ (1-5) |
| skill\_moves | Cấp độ di chuyển kỹ năng của cầu thủ (1-5) |
| work\_rate | Tốc độ làm việc của cầu thủ |
| body\_type | Kiểu cơ thể của cầu thủ |
| real\_face | Khuôn mặt của cầu thủ |
| release\_clause\_eur | Điều khoản giải phóng cầu thủ bằng euro theo FIFA |
| player\_tags | Thẻ cầu thủ trong FIFA |
| team\_position | Vị trí với đội của họ trong FIFA |
| team\_jersey\_number | Số áo đấu với đội của họ trong FIFA |
| loaned\_from | Cầu thủ mượn từ đâu |
| joined | Khi một cầu thủ gia nhập câu lạc bộ của họ |
| contract\_valid\_until | Hiệu lực hợp đồng của cầu thủ |
| nation\_position | Vị trí của trên đội tuyển quốc gia |
| nation\_jersey\_number | Số áo đấu trên đội tuyển quốc gia |

Bảng 3‑3: Mô tả các thuộc tính trong bộ dữ liệu

#### Lựa chọn dữ liệu giá chuyển nhượng:

Transfermarkt là một trang web chuyên về thống kê và thông tin liên quan đến thị trường chuyển nhượng, cũng như thông tin về cầu thủ, câu lạc bộ, và các giải đấu bóng đá trên khắp thế giới. Đây là một nguồn dữ liệu quan trọng mà nhiều chuyên gia bóng đá, đội quản lý, và người hâm mộ thường sử dụng để theo dõi và đánh giá giá trị chuyển nhượng của cầu thủ. Sự độc lập và liên tục của dữ liệu là điểm mạnh của Transfermarkt. Với hàng ngàn cầu thủ và câu lạc bộ được cập nhật mỗi ngày, trang web này giúp người hâm mộ và chuyên gia theo dõi những thay đổi trên thị trường chuyển nhượng một cách chính xác. Thông tin chi tiết về kỹ năng, hiệu suất, hợp đồng, và nhiều khía cạnh khác của cầu thủ cung cấp một cái nhìn toàn diện

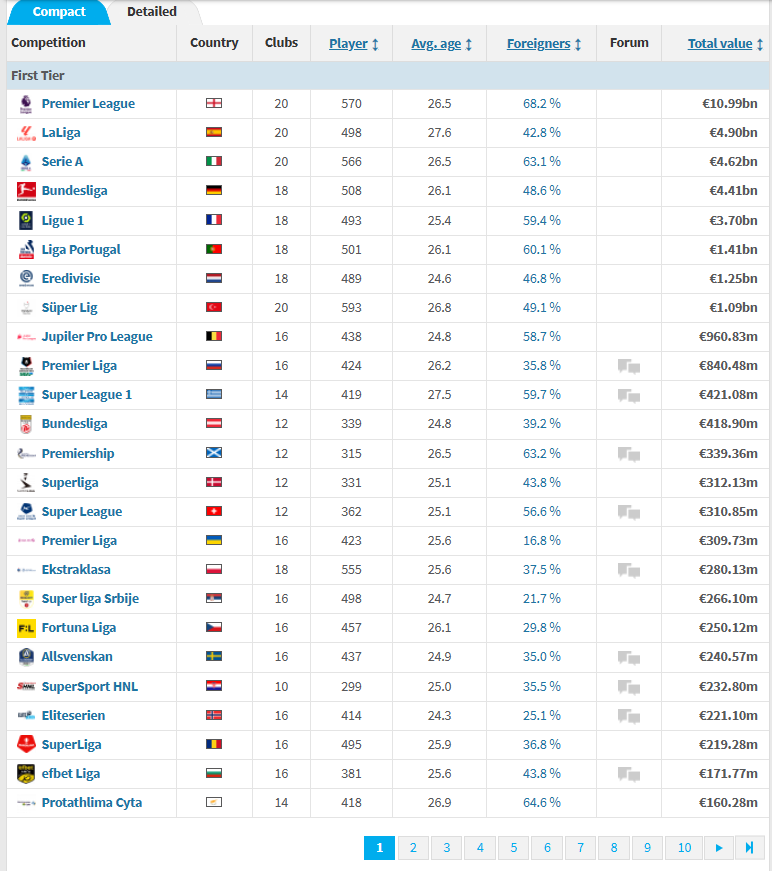
Có nhiều bằng chứng chứng minh sự đáng tin cậy của dữ liệu từ Transfermarkt. Thương vụ chuyển nhượng thực tế thường được cập nhật nhanh chóng và chính xác trên trang web này. Sự đồng thuận và tin tưởng từ cộng đồng bóng đá là một lý do khác chứng minh sự uy tín của trang web. Sự cam kết với sự đổi mới và cập nhật liên tục cũng làm cho Transfermarkt trở thành một nguồn dữ liệu linh hoạt và phản ánh chính xác xu hướng thị trường.

Một số nghiên cứu đã sử dụng dữ liệu từ Transfermarkt như:

* Barbuscak (2018) [5] đã nghiên cứu giá trị thị trường của các cầu thủ bóng đá bằng cách sử dụng dữ liệu từ transfermarkt.com. Barbuscak đã chạy một phép hồi quy tuyến tính, cho thấy tác động đáng kể của nhiều biến số lên giá trị thị trường, chẳng hạn như số năm còn lại của hợp đồng.
* Poli và cộng sự (2022) [6] cố gắng dự đoán phí chuyển nhượng cầu thủ bằng mô hình hồi quy tuyến tính bội. Họ dung transfermarkt.com và các nguồn dữ liệu uy tín trong thế giới bóng đá để xác thực phí chuyển nhượng trả cho cầu thủ.

#### Mô tả bộ dữ liệu giá chuyển nhượng:

Dữ liệu về giá chuyển nhượng của cầu thủ được thu thập từ trang web Transfermarkt trong khoảng thời gian từ mùa giải 2014/2015 – 2022/2023 để phù hợp với dữ liệu cầu thủ đã thu thập ở trang web sofifa.com.



Hình 3‑2: Các giải đấu có tổng số tiền chi cho thị trường chuyển nhượng cao nhất

Dựa vào (Hình 3-2) ta có thể thấy các giải đấu đã chi bao nhiêu tiền cho thị trường chuyển nhượng, giải đấu có cấp độ càng cao thì mức chi tiêu cho việc mua, bán các cầu thủ cũng tỷ lệ thuận theo. Do đó để có thể có được nhiều dữ liệu nhất tôi sẽ lựa chọn 20 giải đấu giảm dần theo số tiền, chi tiết các giải đấu như sau (Bảng 3-4)

|  |  |
| --- | --- |
| **Tên các giải đấu** | **Mô tả** |
| Premier League | Giải bóng đá hàng đầu ở Anh |
| La Liga | Giải bóng đá hàng đầu ở Tây Ban Nha. |
| Bundesliga | Giải bóng đá hàng đầu ở Đức. |
| Serie A | Giải bóng đá hàng đầu ở Ý |
| Ligue 1 | Giải bóng đá hàng đầu ở Pháp |
| POR Liga Portugal | Giải bóng đá hàng đầu ở Bồ Đào Nha. |
| TUR Süper Lig | Giải bóng đá hàng đầu ở Thổ Nhĩ Kỳ. |
| NLD Eredivisie | Giải bóng đá hàng đầu ở Hà Lan. |
| BEL Jupiler Pro League | Giải bóng đá hàng đầu ở Bỉ. |
| RUS Premier Liga | Giải bóng đá hàng đầu ở Nga. |
| GRE Super League 1 | Giải bóng đá hàng đầu ở Hy Lạp. |
| AUS Bundesliga | Giải bóng đá hàng đầu ở Áo. |
| Scottish Premiership | Giải bóng đá hàng đầu ở Scotland. |
| Super League | Giải bóng đá hàng đầu ở Thụy Sĩ. |
| Premier Liga | Giải bóng đá hàng đầu ở Ukraine. |
| Ekstraklasa | Giải bóng đá hàng đầu ở Ba Lan. |
| Superliga | Giải bóng đá hàng đầu ở Đan Mạch. |
| Super liga Srbije | Giải bóng đá hàng đầu ở Serbia. |
| Allsvenskan | Giải bóng đá hàng đầu ở Thụy Điển. |
| SuperSport HNL | Giải bóng đá hàng đầu ở Croatia. |

Bảng 3‑4: 20 giải đấu bóng đá với tổng số tiền chi cho việc chuyển nhượng cao nhất

Như đã đề cập ở trên, nghiên cứu này tập trung vào việc mua một cầu thủ, nghĩa là có tồn tại giao dịch giữa 2 câu lạc bộ. Do đó nên các dữ liệu chuyển nhượng tự do hay đi mượn sẽ không được thu thập trong bộ dữ liệu này (Hình 3-3)

A screenshot of a football match

Description automatically generated

Hình 3‑3: Dữ liệu chuyển nhượng của Transfermarkt

|  |  |
| --- | --- |
| **Tên các thuộc tính** | **Mô tả** |
| Club name | Câu lạc bộ của cầu thủ |
| Player name | Tên của cầu thủ |
| Age | Tuổi của cầu thủ |
| Position | Vị trí của cầu thủ |
| Nationality | Quốc tịch của cầu thủ |
| Dealing\_club | Câu lạc bộ mua |
| Fee | Phí chuyển nhượng trả cho cầu thủ |
| Movement | Chuyển khoản đến hoặc đi |
| Window | Kỳ chuyển nhượng |
| League name | Tên giải đấu mà club\_name thi đấu |
| Year | Năm chuyển giao |
| Season | Mùa chuyển nhượng |

Bảng 3‑5: Mô tả các thuộc tính trong bộ dữ liệu giá chuyển nhượng

## Thu thập dữ liệu:

### Thu thập dữ liệu cầu thủ từ trang web sofifa:

A diagram of a flowchart

Description automatically generated

Hình 3‑4: Quy trình thu thập dữ liệu Sofifa

Dựa vào mô tả dữ liệu đã nói ở trên, dữ liệu cầu thủ sẽ được thu thập từ FIFA15 – FIFA22 (tương ứng với mùa giải 2014/2015 – 2021/2022) tôi sẽ xây dựng các bước thu thập dữ liệu cầu thủ như sau:

Bước 1: Gửi yêu cầu đến trang web

Khi bắt đầu quá trình thu thập dữ liệu từ trang web Sofifa.com, ta đối mặt với thách thức là trang web này áp đặt các hạn chế và chặn yêu cầu, thường trả về mã lỗi 403 khi phát hiện các yêu cầu tự động. Để vượt qua giới hạn này, chúng ta sử dụng thư viện Cloudscraper. Thay vì sử dụng thư viện requests thông thường, Cloudscraper giúp giả mạo thông tin người dùng và thực hiện các bước phức tạp để tránh bị chặn bot.

Bước 2: Lấy danh sách liên kết các cầu thủ

Trên trang web Sofifa, thông tin chi tiết của từng cầu thủ không được hiển thị trực tiếp trên danh sách chính. Để có được dữ liệu chi tiết, quá trình crawl bao gồm việc lấy các đường liên kết của từng cầu thủ từ trang web. Mỗi đường liên kết này đại diện cho một cầu thủ cụ thể và chứa thông tin chi tiết về cầu thủ đó. Chúng ta sẽ lấy các các đường liên kết này và sau đó tiếp tục truy cập từng đường liên kết để thu thập dữ liệu chi tiết.

A screenshot of a computer screen

Description automatically generated

Hình 3‑5: Giao diện chứa đường liên kết của cầu thủ

Bước 3: Lấy dữ liệu chi tiết của cầu thủ

Sau khi đã có được đường liên kết đến trang web chứa thông tin chi tiết cầu thủ (Hình 3-1) ta sẽ tiến hành thu thập dữ liệu. Trong bước này ta sẽ sử dụng BeautifulSoup để phân tích và lấy ra giá trị của thuộc tính dựa trên cấu trúc HTML của trang web.

Với các thuộc tính như short\_name, player\_name, nationality, overall, potential, value và wage ta dùng phương thức ‘find’ để định vị thẻ và class chứa các dữ liệụ sau đó loại bỏ các khoảng trắng sử dụng ‘strip()’.

Với dữ liệu là một biểu thức văn bản, ta sử dụng Regular Expression (Regex) để có thể lấy được dữ liệu chính xác. Trong bài tôi đã sử dụng để lấy dữ liệu từ biểu thức có dạng như sau: “32y.o. (Aug 21, 1988) 185cm / 6'1" 81kg / 179lbs” để lấy ra các thông tin về tuổi, ngày tháng năm sinh, vị trí, chiều cao (cm), cân nặng (kg).

A screenshot of a computer

Description automatically generated

Hình 3‑6: Dữ liệu thông tin cá nhân cầu thủ

Đối với các thuộc tính có cùng cấu trúc HTML giống nhau tôi sử dụng hàm find\_all để tìm tất cả các thẻ chứa dữ liệu, sau đó tôi sẽ lặp qua từng phần tử của find\_all để lấy dữ liệu. Các dữ liệu thuộc trường hợp này (Hình 3-7). Chúng ta sẽ ví dụ cách để thu thập các dữ liệu này như sau:

card\_divs = soup.find\_all('div', class\_='card')

for card\_div in card\_divs:

ul\_element = card\_div.find('ul', class\_='pl')

if ul\_element:

li\_elements = ul\_element.find\_all('li')

for li in li\_elements:

span\_tag = li.find('span', class\_='bp3-tag')

attribute\_name\_span = li.find('span', role='tooltip')

if span\_tag and attribute\_name\_span:

attribute\_value = span\_tag.text.strip()

attribute\_name = attribute\_name\_span.text.strip()

player\_data.append((attribute\_name, attribute\_value))

else:

pass

else:

pass

Mã nguồn trích xuất thông tin chi tiết về cầu thủ từ các phần tử div có class 'card'. Đối với mỗi phần tử này, ta tìm thẻ ul với class 'pl' để lấy thông tin cầu thủ.

Trong mỗi ul, ta tìm tất cả các phần tử li để trích xuất dữ liệu. Trong mỗi phần tử li, ta kiểm tra thẻ span và thẻ span có vai trò 'tooltip'. Nếu cả hai thẻ này tồn tại, ta lấy tên thuộc tính và giá trị tương ứng, nếu không ta bỏ qua phần tử đó. Làm tương tự với các dữ liệu khác.

A screenshot of a computer screen

Description automatically generated

Hình 3‑7: Dữ liệu về đặc điểm và chỉ số kỹ năng của cầu thủ

Bước 4: Sau khi hoàn tất quá trình trích xuất dữ liệu từ trang web Sofifa, chúng ta chuyển sang bước xử lý theo năm. Đầu tiên, cần xác định các mã số được đánh cho từng năm trên trang web (Hình 3-8). Điều này sẽ giúp xây dựng đường dẫn đến trang danh sách cầu thủ của từng năm khi chạy mã nguồn.

Với mỗi năm, chúng ta sẽ lặp qua các trang, mỗi trang chứa thông tin về 60 cầu thủ. Bằng cách này, ta có thể lấy dữ liệu cho tất cả các cầu thủ trên trang và di chuyển đến trang tiếp theo cho đến khi không còn dữ liệu cầu thủ nào nữa. Điều này giúp đảm bảo việc thu thập toàn bộ dữ liệu từ Sofifa theo năm.

A screenshot of a computer screen

Description automatically generated

Hình 3‑8: Mã số từng năm của Sofifa

Bước 5: Để phòng tránh quá trình thu thập xảy ra lỗi, ta sẽ sử dụng hàm bắt lỗi và sẽ chạy lại sau mỗi khoảng thời gian nhất định. Mỗi dữ liệu cầu thủ khi được thu thập sẽ được lưu vào file csv để tránh bị quá tải bộ nhớ. Dưới đây là kết quả sau khi thu thập (Hình 3-9).

A screen shot of a computer screen

Description automatically generated

Hình 3‑12: Kết quả thu được

### Thu thập dữ liệu giá chuyển nhượng từ trang web transfermarkt:

A diagram of a network

Description automatically generated

Hình 3‑9: Quy trình thu thập dữ liệu giá chuyển nhượng

Dựa vào mô tả dữ liệu đã nói ở trên, dữ liệu cầu thủ sẽ được thu thập từ mùa giải 2014/2015 – 2021/2022, tôi sẽ xây dựng các bước thu thập dữ liệu giá chuyển nhượng như sau:

Bước 1: Để xây dựng đường liên kết để có thể gửi yêu cầu ta hãy phân tích một đường dẫn sau đây:

<https://www.transfermarkt.com/premier_league/transfers/wettbewerb/GB1/plus/?saison_id=2022&s_w=&leihe=1&intern=0&intern=1>

Trong đó:

* <https://www.transfermarkt.com/premier_league/transfers/wettbewerb/GB1/plus/>. Với các thông tin bao gồm giải đấu, mã giải đấu và thông tin đang là về giá trị chuyển nhượng.
* saison\_id=2022: Tham số này chỉ định mùa giải cụ thể, trong trường hợp này là mùa giải 2022/2023.
* s\_w=: Tham số này đại diện cho thời điểm chuyển nhượng (summer - mùa hè, winter - mùa đông).
* leihe=1: Tham số này chỉ định cách xử lý chuyển nhượng liên quan đến việc cho mượn và nó có các giá trị sau: Giá trị = 0 hiển thị chuyển nhượng không bao gồm các khoản vay. Giá trị = 1hiển thị tất cả các chuyển nhượng. Giá trị = 2 chỉ hiển thị các khoản vay. Giá trị = 3 hiển thị tất cả chuyển nhượng, nhưng loại trừ các cầu thủ quay về từ khoản cho mượn.

Ta sẽ lựa chọn các tham số để có thể tạo ra các đường dẫn phù hợp với dữ liệu muốn thu thập (Hình 3-10).

A screenshot of a computer program

Description automatically generated

Hình 3‑10: Lựa chọn tham số để tạo đường dẫn

Bước 2: Xử lý dữ liệu đầu vào: Đối với các giá trị khác, chúng tôi sẽ chỉ giữ lại một giá trị duy nhất. Trong khi đó, đối với giải đấu và mùa giải, chúng tôi sẽ chọn theo phạm vi nhất định. Phạm vi mùa giải sẽ bắt đầu từ năm 1992 đến hiện tại, và chỉ giữ lại các giải đấu nằm trong danh sách top 20. Sử dụng hàm \_parse\_hyphenated\_string để phân tích thông tin về mùa bóng đá nhập từ người dùng. Hàm này sẽ chuyển đổi thông tin mùa bóng đá từ định dạng "1992-2023" thành một danh sách các năm chạy từ 1992 đến 2023. Tương tự với giải đấu cũng vậy.

Bước 3: Sau khi đã xây dựng được URL, ta sẽ sử dụng Cloudscraper để gửi yêu cầu đến trang web và sau đó sẽ sử dụng BeautifulSoup để phân tích và trích xuất dữ liệu.

Bước 4: Dữ liệu chúng ta cần thu thập nằm trong bảng (Hình 3-3) của từng câu lạc bộ vì vậy ta sẽ lấy tên câu lạc bộ trước. Sử dụng soup.select(".content-box-headline--logo a"), mã này chọn tất cả các thẻ <a> nằm trong lớp CSS .content-box-headline—logo. Với mỗi thẻ được chọn sẽ trích xuất tên của các câu lạc bộ tham gia vào chuyển nhượng. Kết quả được lưu vào một list.

Ta sẽ sử dụng phương thức find-all để chọn tất cả các thẻ <div> có lớp CSS là 'responsive-table' và lưu trữ chúng trong một list. Sau đó ta dùng cơ chế cắt list để phân biệt giữa bảng IN và OUT. Tiếp đến ta sẽ lặp qua từng câu lạc bộ cùng với bảng dữ liệu chuyển nhượng tương ứng. Thêm dữ liệu của bảng IN và OUT vào các dataframe tương ứng sau đó gộp chúng lại để thành một file kết quả.

Bước 5: Xử lý các dữ liệu đặc biệt (Hình 3-11)

* Xử lý phí và cho mượn: Kiểm tra xem có một số từ khóa nhất định trong fee và thực hiện các bước xử lý tương ứng:
  + Nếu bắt đầu bằng "end of loan", cập nhật fee thành "$0" và thiết lập các trạng thái về cho mượn.
  + Nếu bắt đầu bằng "loan fee", loại bỏ phần "loan fee:" và thiết lập các trạng thái.
  + Nếu bắt đầu bằng "loan transfer", cập nhật fee thành "$0" và thiết lập các trạng thái.
  + Nếu bắt đầu bằng "free transfer", cập nhật fee thành "$0".
* Xử lý các giá trị để chuyển đổi chúng thành số: Áp dụng cho các cột market\_value và fee.
  + Nếu giá trị là NaN, hoặc là "-", hoặc là "?", trả về NaN.
  + Nếu giá trị là "0", trả về 0.
  + Nếu giá trị kết thúc bằng 'm', nhân với 1 triệu.
  + Nếu giá trị kết thúc bằng 'th.', nhân với 1 nghìn.
* Chỉnh sửa kiểu dữ liệu và xử lý NaN:
  + Dùng.assign(is\_loan=False, loan\_status='')để thêm cột is\_loan với giá trị mặc định là False và loan\_status với giá trị rỗng cho mỗi hàng.
  + Sử dụng pd.to\_numeric để chuyển đổi cột age thành kiểu dữ liệu số
  + Chuyển cột season thành kiểu dữ liệu năm từ định dạng ngày tháng và thiết lập các giá trị NaN nếu cần.
  + Đối với các cột nhất định (nationality, position, short\_pos, dealing\_club, dealing\_country), sử dụng.fillna('', inplace=True)để điền giá trị rỗng bằng chuỗi trống.
  + Trong cột league, thay thế các dấu gạch ngang bằng khoảng trắng và chuyển đổi thành chữ in hoa.

A screenshot of a screen

Description automatically generated

Hình 3‑11: Các dữ liệu đặc biệt

Bước 6: Dữ liệu sau khi thu thập sẽ được xuất vào file csv ứng với mùa giải và giải đấu bóng đá. Lặp lại quy trình thu thập qua các mùa giải và giải đấu dựa vào các tham số đầu vào.

Kết quả sau quá trình thu thập như sau (Hình 3-12):

A screenshot of a computer

Description automatically generated

Hình 3‑12: Kết quả sau quá trình thu thập giá trị chuyển nhượng

### Hợp nhất dữ liệu:

Hai bộ dữ liệu được hợp nhất dựa trên tên người chơi và năm chuyển nhượng. Nếu một vụ chuyển nhượng diễn ra vào năm 2015 tôi sẽ sử dụng dữ liệu của người chơi đó vào năm 2015. Với dữ liệu giá chuyển nhượng sau khi thu thập, mỗi folder chứa dữ liệu theo năm hiện tại đang có 20 file csv theo 20 giải đấu (Hình 3-13) vì vậy nên tôi sẽ gộp tất cả dữ liệu 20 giải đấu lại theo từng năm.

A screenshot of a computer

Description automatically generated

Hình 3‑13: Dữ liệu từng năm của giá chuyển nhượng

Trong tập dữ liệu giá chuyển nhượng vì chỉ tập trung vào những cầu thủ đã được trả phí nên các dữ liệu có fee = 0 hoặc fee = NaN sẽ được loại bỏ. Cả chuyển khoản đến và đi đều được thu thập, điều này dẫn đến một số giao dịch sẽ xuất hiện 2 lần, do đó nên chỉ chuyển khoản đến sẽ được giữ lại. Kết quả sau khi loại bỏ (Hình 3-14).

A screenshot of a computer

Description automatically generated

Hình 3‑14: Kết quả sau khi lọc của bộ dữ liệu giá chuyển nhượng

Để hợp nhất 2 bộ dữ liệu với nhau một cách chính xác và đẩy đủ thì phải thêm một số thuộc tính, cụ thể như sau (Hình 3‑23):

* Với bộ dữ liệu giá chuyển nhượng tên của cầu thủ chứa nhiều ký tự đặc thù của mỗi quốc gia, và có tên không đầy đủ so với bộ dữ liệu cầu thủ. Vì vậy tôi đã them một trường là normalized\_name để chuyển đổi tên cầu thủ về dạng bình thường.
* Với bộ dữ liệu cầu thủ tôi cũng làm tương tự với 2 cột là player\_name và short\_name để dễ dàng đối chiếu với bộ dữ liệu giá chuyển nhượng.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

Hình 3‑15: Kết quả sau khi thêm trường vào 2 bộ dữ liệu

Sau khi đã thêm các trường mới, tôi tiến hành kết hợp 2 bộ dữ liệu với nhau bằng cách đối chiếu giữa normalized\_name và nationality của bộ giá trị chuyển nhượng với player\_name, player\_short\_name và nationality của bộ dữ liệu cầu thủ. Bởi vì đây là 2 trường không thể nhầm lẫn, khi xác định được tên chính xác và quốc tịch của cầu thủ thì các thuộc tính khác sẽ khớp với nhau. Kết quả sau khi hợp nhất 2 bộ dữ liệu, tôi thu được 4158 quan sát, những dữ liệu không khớp sẽ bị bỏ qua.

## Tiền xử lý dữ liệu:

### Lựa chọn thuộc tính (Feature Selection):

Từ tập dữ liệu đã hợp nhất, có một số thuộc tính sẽ bị loại bỏ vì chúng không mang lại giá trị cho nghiên cứu này. Các thuộc tính sofifa\_id, player\_name, short\_name, long\_name, name vì những thuộc tính này chỉ có giá trị duy nhất, do đó không hữu ích trong việc đưa vào mô hình học máy.

Trong tập dữ liệu FIFA, có một thuộc tính được gọi là league\_rank. Biến league\_rank được sử dụng trong các trò chơi FIFA được phát hành từ năm 2015 đến năm 2020 để mô tả cấp độ của giải đấu và có thang điểm từ 0-4. Thuộc tính này không mang giá trị đối với các cầu thủ nên cũng sẽ bị loại bỏ.

Các thuộc tính age, nationality, league\_name, club\_name, club sẽ bị xóa bỏ vì trùng lặp với nhau. Thuộc tính ‘value\_eur’ thể hiện giá trị của một cầu thủ và ‘wage\_eur’ thể hiện mức lương của một cầu thủ theo FIFA. Vì không rõ các giá trị của các thuộc tính này được tính như thế nào và liệu chúng có thực tế hay không nên chúng bị loại trừ.

Một số biến bắt nguồn từ trò chơi FIFA không cung cấp thông tin về kỹ năng hoặc đặc điểm của cầu thủ. Các thuộc tính loại bỏ bao gồm: player\_positions, work\_rate, body\_type,real\_facerelease\_clause\_eur,player\_tags,team\_position,team\_jersey\_number,loaned\_from, joined, nation\_position, nation\_jersey\_number, pace, shooting, passing, dribbling, defending, physic, gk\_diving, gk\_handling, gk\_kicking, gk\_reflexes, gk\_speed, gk\_positioning, player\_traits, short\_pos, normalized\_name, market\_value, movement, is\_loan,loan\_status,dealing\_club,dealing\_country, international\_reputation, weak\_foot, skill\_moves.

### Xử lý dữ liệu thiếu (Handle Missing Data):

Tôi đã sử dụng isna()để tạo một DataFrame mới với các giá trị boolean (True nếu là NaN, False nếu không phải), sau đó gọi phương thức any()để xác định xem có ít nhất một giá trị True nào đó trong mỗi cột không.

missing\_data = df1.isna().any()

missing\_data[missing\_data].index

Sử dụng df1.isna().sum()tôi xác định được số dữ liệu đang thiếu của mỗi thuộc tính, với mentality\_composure là 884 và goalkeeping\_reflexes là 20.

Dựa vào biểu đồ (Hình 3-16) tôi thấy rằng các giá trị của goalkeeping\_reflexes phân phối theo các vị trị của cầu thủ. Với đặc thù là khả năng của thủ môn nên các chỉ số của các vị trí khác sẽ thấp hơn so với goalkeeper, vì vậy nên không thể điền các dữ liệu còn thiếu bằng cách tính trung bình của cả cột thuộc tính đó được như vậy sẽ khiến các giá trị các vị trí khác sẽ cao bất thường và các giá trị của nhóm goalkeeper sẽ bị giảm đáng kể.

Điều này trái so với bộ dữ liệu nên tôi đã tính trung bình của goalkeeping\_reflexes theo từng vị trí sau đó điền các giá trị đó với các dữ liệu thiếu tương ứng với nhóm vị trí của chúng.

A graph with blue dots

Description automatically generated A graph with blue dots

Description automatically generated

Hình 3‑16: Phân bố dữ liệu của goalkeeping\_reflexes và mentality\_composure với các vị trí của cầu thủ

### Kỹ thuật thuộc tính (Feature Engineering):

Vì tập dữ khá nhỏ chỉ với 4158 mẫu quan sát nên các vị trí của thuộc tính position sẽ không thể đánh giá tốt được với số mẫu ít như vậy. Do đó tôi đã nhóm các vị trí theo khu vực họ chơi lại với nhau thành một thuộc tính là position\_group với 4 giá trị như sau: attacker, midfielder, defender, goalkeeper.

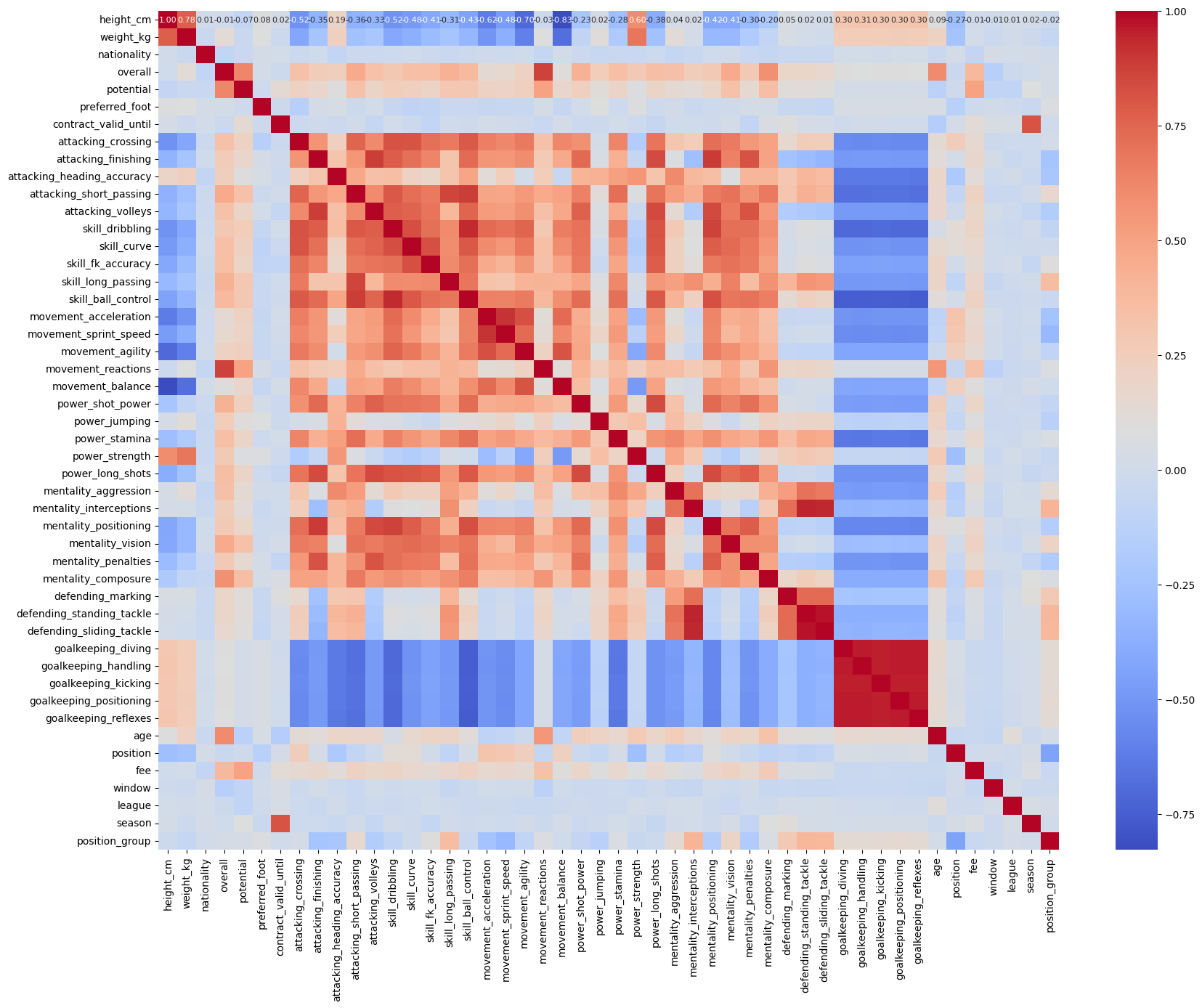
Trên thực tế thì các thủ môn sẽ có các bài đánh giá khác so với các vị trí khác trên sân. Cộng thêm số mẫu của vị trí thủ môn trong bộ dữ liệu cũng rất ít (Hình 3-17), điều này khiến cho việc so sánh thủ môn với các cầu thủ trên sân trở nên khó khăn và do đó chúng tôi sẽ loại thủ môn khỏi các mô hình chính của nghiên cứu này

A graph of a number of players

Description automatically generated

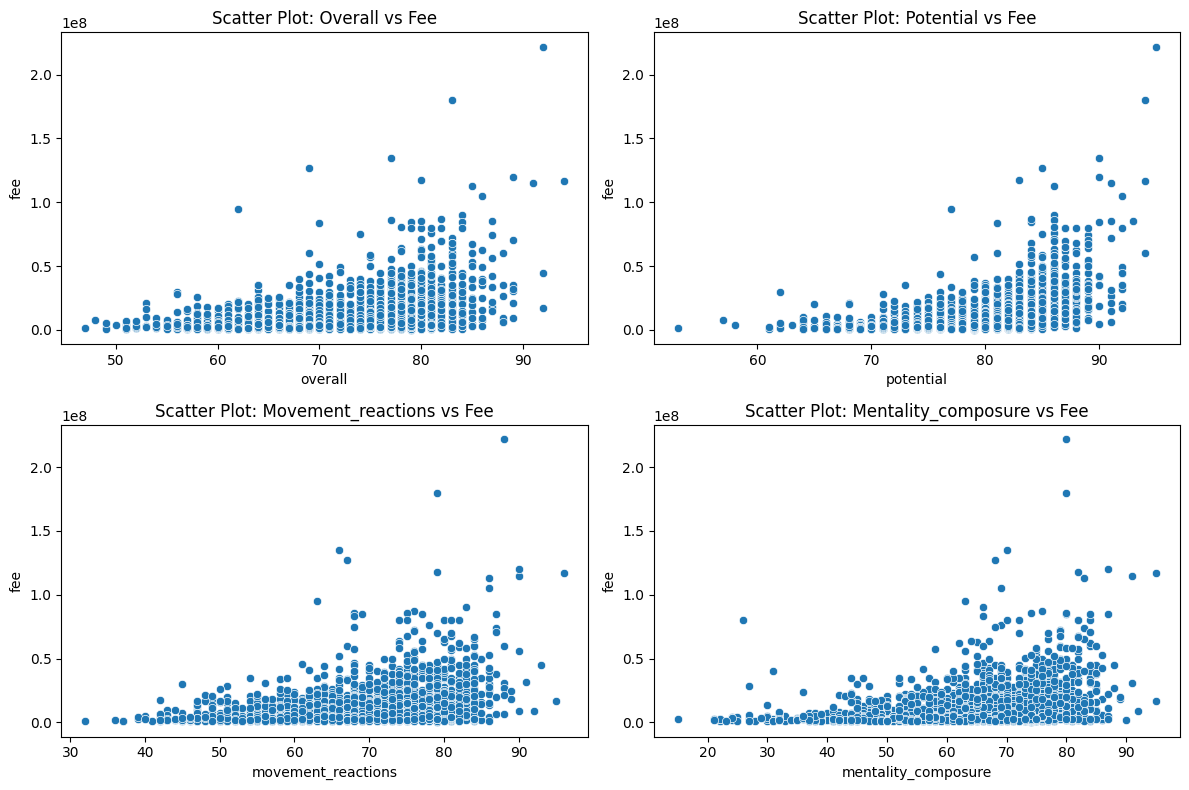
Hình 3‑17: Phân bố dữ liệu theo nhóm các vị trí

### Xử lý dữ liệu ngoại lai (Remove Outlier):



Hình 3‑18: Biểu đồ heatmap thể hiện sự tương quan giữa các thuộc tính

Dựa vào biểu đồ (Hình 3-18), thấy rằng có một vài biến có sự tương quan cao với biến mục tiêu (ở đây là thuộc tính ‘fee’) như overall, potentinal, movement\_reactions, mentality\_composure. Tôi sẽ trực quan hóa mối quan hệ giữa các biến tương quan mạnh nhất với biến mục tiêu ‘fee’ để xác định các giá trị ngoại lai (Hình 3-19).



Hình 3‑19: Biểu đồ trực quan mối quan hệ giữa các thuộc tính overall, potentinal, movement\_reactions, mentality\_composure với biến mục tiêu fee

Biểu đồ trong Hình 3-19 cho thấy có một vài điểm dữ liệu phi logic. Nhìn vào biểu đồ giữa overall và fee ta thấy có các điểm dữ liệu chỉ có overall từ 60 -70 nhưng giá trị lại cao hơn mức overall 80 – 90, tương tự potential cũng vậy. Các điểm dữ liệu này nằm không tuân theo mô hình chung của dữ liệu nên ta có thể xác định chính là dữ liệu ngoại lai và tiến hành xóa bỏ nó.

Trong bài này tôi sử dụng Z-score để loại bỏ các dữ liệu ngoại lai. Z-score là một phép đo thống kê đo khoảng cách của một giá trị từ trung bình của tập dữ liệu, tính theo đơn vị độ lệch chuẩn. Để loại bỏ dữ liệu ngoại lai sử dụng Z-score:

* Xác định Z-Score: Tính Z-score cho từng quan sát trong tập dữ liệu.
* Đặt ngưỡng Z-Score: Chọn một ngưỡng Z-score (thường là 2 hoặc 3) để xác định giá trị ngoại lai.
* Loại bỏ hoặc đánh dấu ngoại lai: Loại bỏ hoặc đánh dấu các giá trị có Z-score vượt quá ngưỡng.

Kết quả sau khi loại bỏ dữ liệu ngoại lai (Hình 3-20):

A group of images of a graph

Description automatically generated with medium confidence

Hình 3‑20: Biểu đồ trực quan mối quan hệ giữa các thuộc tính overall, potentinal, movement\_reactions, mentality\_composure với biến mục tiêu fee sau khi loại bỏ các dữ liệu ngoại lai

### Chuẩn hóa và mã hóa dữ liệu:

Để thực hiện tiêu chuẩn hóa dữ liệu, chúng ta có thể sử dụng StandardScaler trong thư viện Scikit-learn. Công thức như sau:

Trong bài này các dữ liệu đều có một phạm vi nhất định không bị lệch quá nhiều ngoại trừ biến mục tiêu fee. Vì vậy, tôi sẽ scale dữ liệu cho biến fee để ổn định hiệu suất và cải thiện cho mô hình học máy.

Về mã hóa dữ liệu tôi sử dụng LabelEncoder, một thư viện của Python được dùng để chuyển đổi các giá trị của biến phân loại thành các biến số nguyên. Trong bài này các thuộc tính sau sẽ được sử dụng : nationality, preferred\_foot, position\_group, window, league

## Xây dựng mô hình:

A diagram of a system

Description automatically generated

Hình 3‑21: Sơ đồ quy trình xây dựng mô hình học máy

### Mô hình LightGBM:

Mô hình được xây dựng ở đây là một Regressor sử dụng thuật toán LightGBM, một mô hình Gradient Boosting Framework hiệu quả. Mục tiêu của mô hình là dự đoán giá trị của biến mục tiêu 'fee' dựa trên các biến độc lập trong bộ dữ liệu.

Trước khi xây dựng mô hình, dữ liệu đã được chuẩn bị bằng cách mã hóa biến phân loại và chuẩn hóa biến mục tiêu ‘fee’. Trước khi chia tập huấn luyện và tập kiểm tra ta sẽ logarit biến ‘fee’ việc này có thể là để giảm biến động của dữ liệu và đảm bảo phân phối của nó gần với phân phối chuẩn.

Sau đó, tập dữ liệu được chia thành tập huấn luyện (X\_train, y\_train) và tập kiểm tra (X\_test, y\_test) sử dụng hàm train\_test\_split.

Mô hình LightGBM được tinh chỉnh thông qua việc tìm kiếm siêu tham số sử dụng phương pháp Bayesian Optimization. Phạm vi của các siêu tham số được định nghĩa trong param\_space, và các giá trị tốt nhất được chọn dựa trên độ đo 'neg\_mean\_squared\_error'. Các tham số tốt nhất và giá trị đạt được cao nhất được in ra màn hình. Các tham số cụ thể như sau:

* colsample\_bytree (0.7): Quyết định tỷ lệ số cột được lấy ngẫu nhiên khi xây dựng mỗi cây trong quá trình huấn luyện. Giá trị này giúp kiểm soát sự đa dạng của các cây trong ensemble.
* learning\_rate (0.02): Tốc độ học của mô hình, tức là mức độ điều chỉnh trọng số sau mỗi bước huấn luyện. Learning rate càng nhỏ thì mô hình càng ổn định, nhưng có thể yêu cầu thêm thời gian để hội tụ.
* max\_depth (8): Độ sâu tối đa của mỗi cây trong ensemble. Điều này kiểm soát độ phức tạp của mô hình và giúp tránh tình trạng quá mức fitting.
* min\_child\_samples (31): Số lượng mẫu tối thiểu được yêu cầu để tạo một nút lá. Điều này giúp kiểm soát sự phức tạp của cây và tránh tình trạng overfitting.
* min\_child\_weight (8): Trọng lượng tối thiểu của một nút lá. Giảm giá trị này có thể dẫn đến việc tăng độ phức tạp của mô hình.
* min\_split\_gain (0.08): Ngưỡng tối thiểu để thực hiện một phân chia. Điều này kiểm soát độ nhảy của mô hình khi phân loại dữ liệu.
* n\_estimators (350): Số lượng cây (estimators) được xây dựng trong ensemble. Điều này kiểm soát số lượng bước huấn luyện.
* num\_leaves (57): Số lượng lá tối đa trên mỗi cây. Số lượng lá càng lớn, mô hình càng có khả năng fit tốt dữ liệu huấn luyện, nhưng cũng có nguy cơ overfitting.
* reg\_alpha (0.2): Hệ số alpha của L1 regularization. Hỗ trợ kiểm soát việc sử dụng các biến quan trọng.
* reg\_lambda (0.9): Hệ số lambda của L2 regularization. Cũng là một hệ số kiểm soát regularization, giúp tránh overfitting.
* subsample (0.9): Tỷ lệ mẫu được sử dụng để huấn luyện mỗi cây. Điều này giúp kiểm soát độ phức tạp và đa dạng của mô hình.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

Sau khi xác định được các tham số tốt nhất, mô hình LightGBM được xây dựng lại (best\_model\_bayes) và được sử dụng để dự đoán trên tập kiểm tra. Các phương pháp đánh giá như Mean Squared Error (MSE), R-squared, Mean Absolute Error (MAE), và Root Mean Squared Error (RMSE) được sử dụng để đánh giá hiệu suất của mô hình trên tập kiểm tra.

### Mô hình XGBoost:

Mô hình XGBoost được xây dựng là một mô hình học máy sử dụng thuật toán cực mạnh và hiệu quả, chuyên dụng cho bài toán dự đoán giá trị 'fee' trong DataFrame df2. Dưới đây là mô tả về cách mô hình hoạt động.

Đầu tiên, dữ liệu được chuẩn bị bằng cách tách các đặc trưng (biến độc lập) và biến mục tiêu (fee), sau đó thực hiện biến đổi logarit cho biến mục tiêu để làm giảm độ lớn của giá trị và cải thiện tính chất phân phối.

Tiếp theo, dữ liệu được chia thành tập huấn luyện (X\_train, y\_train) và tập kiểm tra (X\_test, y\_test) bằng cách sử dụng hàm train\_test\_split.

Mô hình XGBoost được chọn làm mô hình dự đoán. Một không gian siêu tham số để tìm kiếm được xác định (param\_space\_xgb). Trong trường hợp này, tìm kiếm siêu tham số được thực hiện bằng cách sử dụng Bayesian Optimization (BayesSearchCV). Các siêu tham số chi tiết:

* colsample\_bylevel: Xác định tỷ lệ số cột được lấy ngẫu nhiên cho mỗi cấp độ cây (level). Giảm giá trị này có thể giúp kiểm soát việc overfitting.
* colsample\_bynode: Tỷ lệ số cột được lấy ngẫu nhiên cho mỗi nút trong cây. Điều này kiểm soát độ đa dạng của cây.
* colsample\_bytree: Xác định tỷ lệ số cột được lấy ngẫu nhiên cho mỗi cây trong ensemble. Giúp kiểm soát sự đa dạng của các cây.
* gamma: Ngưỡng cắt cho cây để thực hiện một phân chia. Giảm giá trị này có thể kiểm soát sự phức tạp của mô hình và giảm overfitting.
* learning\_rate: Tốc độ học của mô hình, tức là mức độ điều chỉnh trọng số sau mỗi bước huấn luyện. Giảm learning rate có thể làm tăng ổn định của mô hình.
* max\_depth: Độ sâu tối đa của mỗi cây trong ensemble. Kiểm soát độ phức tạp của mô hình và giúp tránh overfitting.
* min\_child\_weight: Trọng lượng tối thiểu của mỗi nút lá. Giảm giá trị này có thể dẫn đến mô hình phức tạp hơn.
* n\_estimators: Số lượng cây (estimators) trong ensemble. Kiểm soát số lượng bước huấn luyện.
* reg\_alpha: Hệ số alpha của L1 regularization. Giúp kiểm soát việc sử dụng các biến quan trọng.
* subsample: Tỷ lệ mẫu được sử dụng để huấn luyện mỗi cây. Giảm giá trị này có thể kiểm soát sự đa dạng của mô hình.

Sau khi tìm kiếm siêu tham số, mô hình tốt nhất được chọn và lưu vào biến best\_model\_xgb. Mô hình này sau đó được sử dụng để dự đoán trên tập kiểm tra (X\_test), và các kết quả được lưu vào biến predictions\_xgb.

Cuối cùng, một số thông số đánh giá mô hình như Mean Squared Error (MSE), R-squared, Mean Absolute Error (MAE) và Root Mean Squared Error (RMSE) được in ra để đánh giá hiệu suất của mô hình trên tập kiểm tra.

### Mô hình Random Forest Regression:

Mô hình RandomForestRegressor và thực hiện tinh chỉnh siêu tham số thông qua phương pháp Bayesian Optimization để cải thiện hiệu suất dự đoán giá trị 'fee' trong DataFrame df2. Dưới đây là mô tả về cách xây dựng mô hình:

* Chuẩn bị Dữ liệu: Dữ liệu được chia thành hai phần chính: tập dữ liệu huấn luyện (X\_train, y\_train) và tập dữ liệu kiểm tra (X\_test, y\_test). Biến mục tiêu 'fee' được chuyển đổi bằng hàm log1p để đảm bảo phân phối gần với phân phối chuẩn.
* Xây Dựng và Tinh chỉnh Mô hình RandomForest:
  + n\_estimators: Số lượng cây quyết định trong ensemble. Một số cây lớn có thể dẫn đến overfitting, trong khi một số cây nhỏ có thể làm giảm hiệu suất.
  + max\_depth: Độ sâu tối đa của mỗi cây quyết định. Giới hạn độ sâu giúp kiểm soát độ phức tạp của mô hình và ngăn chặn overfitting.
  + min\_samples\_split: Số mẫu tối thiểu cần để một nút có thể được chia. Giảm giá trị này có thể dẫn đến các cây với các nhánh nhỏ hơn và có thể giảm overfitting.
  + min\_samples\_leaf: Số mẫu tối thiểu cần để một lá của cây. Giảm giá trị này có thể làm tăng độ chính xác của lá nhưng cũng có thể dẫn đến overfitting.
  + bootstrap: Xác định liệu có nên thực hiện lấy mẫu có thay thế hay không. Lấy mẫu có thay thế giúp tăng tính ngẫu nhiên và đa dạng của các cây.
* Tinh chỉnh Tham số: Các tham số được lựa chọn thông qua Bayesian Optimization dựa trên độ đo 'neg\_mean\_squared\_error', với mục tiêu là tối ưu hóa hiệu suất mô hình trên tập dữ liệu huấn luyện.
* Kết quả và Hiệu suất: Kết quả tinh chỉnh tham số được in ra màn hình, bao gồm các giá trị tối ưu của số cây, độ sâu tối đa của cây, và các tham số khác. Mô hình tốt nhất được chọn dựa trên kết quả tinh chỉnh.
* Đánh giá Mô hình: Mô hình tinh chỉnh được đánh giá trên tập kiểm tra thông qua các số liệu đánh giá như Mean Squared Error (MSE), R-squared, Mean Absolute Error (MAE), và Root Mean Squared Error (RMSE) để đảm bảo khả năng dự đoán chính xác và hiệu quả của mô hình.

# : KẾT QUẢ VÀ ĐÁNH GIÁ MÔ HÌNH

## Kết quả các độ đo của các mô hình:

Trong phần này, các kết quả chạy của từng mô hình được tổng hợp dưới bảng sau (Bảng 4-1):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên các mô hình** | **MAE** | **MSE** | **RMSE** | **R2** |
| **LightGBM** | 0.29693 | 0.14828 | 0.38507 | 0.652034 |
| **XGBoost** | 0.29520 | 0.14864 | 0.38554 | 0.64700 |
| **Random Forest** | 0.32074 | 0.17217 | 0.41494 | 0.59595 |

Bảng 4‑1: Kết quả các độ đo của các mô hình học máy

LightGBM và XGBoost: Cả hai mô hình gradient boosting, LightGBM và XGBoost, đều cho thấy hiệu suất tốt với các giá trị R-squared lớn, đặc biệt là LightGBM với R2 đạt 0.652034. MAE và RMSE của cả hai mô hình cũng thấp, cho thấy khả năng dự đoán chính xác và độ chính xác của mô hình.

Random Forest: Mô hình Random Forest cũng có hiệu suất tốt, nhưng thấp hơn so với các mô hình gradient boosting. Mặc dù có R2 tốt (0.59595), nhưng MAE và RMSE cao hơn so với LightGBM và XGBoost, chỉ ra rằng có thể có sự chệch trong dự đoán.

## Đánh giá mô hình:

# : KẾT LUẬN

Trong nghiên cứu này, chúng tôi muốn tìm hiểu xem giá trị chuyển nhượng của các cầu thủ bóng đá chuyên nghiệp có thể dự đoán được ở mức độ nào dựa trên kỹ năng và đặc điểm. Để kiểm tra điều này, chúng tôi đã xây dựng một mô hình để dự đoán giá trị chuyển nhượng của các cầu thủ bóng đá chuyên nghiệp. Mô hình hoạt động tốt nhất của chúng tôi, mô hình hồi quy LightGBM (𝑅2 = 0,652), hoạt động kém hơn các mô hình dự đoán giá trị thị trường từ công việc liên quan. Điều này có thể là do giá trị thị trường là giá trị được tính toán, trong khi giá trị chuyển nhượng là phí thực tế phải trả để mua một cầu thủ. Có thể có những yếu tố khác góp phần tạo ra phí chuyển nhượng thực tế so với giá trị thị trường mà chúng tôi không xem xét trong nghiên cứu này. Điều này có thể gây ra hiệu suất tồi tệ hơn. Dự đoán về giá trị chuyển nhượng của các cầu thủ bóng đá chuyên nghiệp thay vì giá trị thị trường, kết hợp với việc sử dụng nhiều thuật toán học máy, là điểm mới chính của nghiên cứu này vì điều này chưa từng được thực hiện trước đây theo hiểu biết của chúng tôi.

Nghiên cứu này cung cấp cái nhìn sâu sắc về những tính năng quan trọng nhất trong việc dự đoán giá trị chuyển giao. Một số tính năng là đặc điểm và những tính năng khác là xếp hạng kỹ năng từ trò chơi điện tử FIFA, được chứng minh là một nguồn hữu ích để dự đoán giá trị chuyển nhượng, vì chín trong số mười tính năng quan trọng nhất trong mô hình hồi quy LightGBM của chúng tôi đến từ dữ liệu trò chơi điện tử. Các tính năng quan trọng nhất là xếp hạng tổng thể hiện tại và tiềm năng của người chơi. Nghiên cứu trước đây đề xuất nên cân nhắc các vị trí khác nhau trong lĩnh vực này khi xem xét các đặc điểm quan trọng nhất, vì mỗi vị trí sẽ có sự khác biệt. Trong nghiên cứu này, chúng tôi nhận thấy có sự khác biệt về các đặc điểm quan trọng nhất của mỗi vị trí.

Chúng ta có thể kết luận rằng giá trị chuyển giao có thể dự đoán được ở một mức độ nào đó, với một số tính năng quan trọng hơn những tính năng khác. Các tính năng quan trọng nhất khác nhau tùy theo vị trí. Dự đoán về giá trị chuyển nhượng không chính xác như dự đoán về giá trị thị trường của nghiên cứu trước đó. Điều này có thể là do chúng tôi đã sử dụng các biến tương tự như nghiên cứu trước đây để dự đoán giá trị thị trường. Đối với nghiên cứu trong tương lai, có thể sẽ thú vị nếu đưa các biến mới vào mô hình để xem liệu điều này có ảnh hưởng đến hiệu suất của mô hình hay không, chẳng hạn như ngân sách chuyển nhượng của các câu lạc bộ.