TRƯỜNG ĐẠI HỌC NGOẠI NGỮ - TIN HỌC THÀNH PHỐ HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO KẾT THÚC HỌC PHẦN

MÁY HỌC

XÂY DỰNG MÔ HÌNH HỒI QUY DỰ ĐOÁN CHỈ SỐ KỸ NĂNG CỦA CẦU THỦ

Giảng viên hướng dẫn: **ThS. Huỳnh Thành Lộc**

Sinh viên thực hiện:

1. Nguyễn Hoàng Bảo 22DH110271
2. Lương Tiến Đạt 22DH114497
3. Nguyễn Trần Bảo Long 22DH112007

Thành phố Hồ Chí Minh, tháng 11 năm 2024

TRƯỜNG ĐẠI HỌC NGOẠI NGỮ - TIN HỌC THÀNH PHỐ HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO KẾT THÚC HỌC PHẦN

MÁY HỌC

XÂY DỰNG MÔ HÌNH HỒI QUY DỰ ĐOÁN CHỈ SỐ KỸ NĂNG CỦA CẦU THỦ

Mã lớp học phần: **241123018401**

Năm học: **2024 – 2025**

Học kỳ: **1**

Sinh viên thực hiện:

1. Nguyễn Hoàng Bảo 22DH110271
2. Lương Tiến Đạt 22DH114497
3. Nguyễn Trần Bảo Long 22DH112007

Thành phố Hồ Chí Minh, tháng 11 năm 2024

MỤC LỤC

DANH MỤC HÌNH

DANH MỤC BẢNG

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1. Giới thiệu bài toán

Bối cảnh**:** FIFA 19 là một trong những tựa game bóng đá phổ biến nhất trên toàn cầu, thu hút hàng triệu người chơi mỗi năm. Một trong những yếu tố quan trọng tạo nên trải nghiệm của người chơi chính là các chỉ số kỹ năng của các cầu thủ. Những chỉ số này không chỉ phản ánh khả năng thực tế của cầu thủ trong game mà còn ảnh hưởng trực tiếp đến cách người chơi tương tác với trò chơi.

Nhiệm vụ chính của nghiên cứu này là xây dựng một mô hình hồi quy tuyến tính để dự đoán chỉ số tổng thể (Overall) của cầu thủ trong game FIFA 19, dựa trên các chỉ số kỹ năng quan trọng như tốc độ, sức mạnh, khả năng dứt điểm,… . Cụ thể, các nhiệm vụ sẽ bao gồm:

- **Thu thập và chuẩn bị dữ liệu**: Tập hợp dữ liệu từ các chỉ số kỹ năng của cầu thủ trong FIFA 19, bao gồm các thông số kỹ thuật và chỉ số tổng thể của các cầu thủ. Sau đó, thực hiện các bước tiền xử lý dữ liệu để đảm bảo chất lượng và tính nhất quán của dữ liệu.
- **Phân tích mối quan hệ giữa các chỉ số kỹ năng và chỉ số tổng thể**: Phân tích các chỉ số kỹ năng và đánh giá mối quan hệ của chúng với chỉ số tổng thể của cầu thủ. Từ đó, xác định các yếu tố quan trọng ảnh hưởng đến chỉ số tổng thể.
- **Xây dựng mô hình hồi quy tuyến tính**: Áp dụng thuật toán hồi quy tuyến tính để xây dựng mô hình dự đoán chỉ số tổng thể dựa trên các chỉ số kỹ năng. Kiểm tra độ chính xác và hiệu quả của mô hình thông qua các chỉ số đánh giá như RMSE (Root Mean Squared Error), R-squared và các kỹ thuật kiểm định mô hình khác.
- **Đánh giá và cải tiến mô hình**: Đánh giá kết quả dự đoán của mô hình, so sánh với dữ liệu thực tế và phân tích những hạn chế của mô hình. Dựa vào đó, đề xuất các biện pháp cải tiến để tăng độ chính xác của dự đoán.

Mục tiêu nghiên cứu**:** Mục tiêu của nghiên cứu này là phân tích các kỹ năng của cầu thủ trong FIFA 19 và đánh giá ảnh hưởng của các kỹ năng này đến chỉ số tổng thể (Overall) của cầu thủ. Dựa trên các kỹ năng đó chúng tôi sẽ xây dựng một mô hình hồi quy tuyến tính để dự đoán chỉ số tổng thể của cầu thủ, giúp hiểu rõ hơn về mối quan hệ giữa các chỉ số kỹ năng và đánh giá tổng thể trong trò chơi.

1.2. Các công trình liên quan

Trong việc dự đoán chỉ số tổng thể (Overall) của cầu thủ trong game FIFA 19 dựa trên các kỹ năng, nhiều mô hình và thuật toán học máy có thể được áp dụng để xây dựng một hệ thống dự đoán hiệu quả. Dưới đây là một số mô hình phổ biến được sử dụng trong các nghiên cứu và bài toán tương tự:

Bảng 1.1. Các thuật toán Hồi Quy

STT	Tên Mô Hình	Mô Tả	Ưu Điểm	Nhược Điểm
1	Hồi Quy Tuyến Tính	Mô hình đơn giản dự đoán giá trị mục tiêu dựa trên mối quan hệ tuyến tính giữa biến độc lập và biến phụ thuộc.	Dễ hiểu, dễ triển khai, và nhanh chóng.	Nhạy cảm với các điểm ngoại lệ và giả định tuyến tính.
2	Hồi Quy Ridge	Phiên bản điều chỉnh của hồi quy tuyến tính với regularization L2, giúp giảm thiểu overfitting.	Giảm thiểu overfitting, làm cho mô hình ổn định hơn.	Không thể thực hiện chọn lọc đặc trưng tự động.
3	Hồi Quy Lasso	Tương tự như hồi quy Ridge nhưng sử dụng regularization L1, giúp loại bỏ một số đặc trưng không cần thiết.	Tự động chọn lọc đặc trưng, giảm thiểu độ phức tạp mô hình.	Có thể loại bỏ quá nhiều đặc trưng quan trọng.
4	Hồi Quy Logistic	Sử dụng cho các bài toán phân loại, dự đoán xác	Đơn giản và hiệu quả cho các bài	Không thể sử dụng cho các biến biên tục.

		suất của các lớp khác nhau.	toán phân loại nhị phân.	

Những thuật toán trên đều có ứng dụng và ưu điểm riêng khi áp dụng vào bài toán dự đoán chỉ số tổng thể trong FIFA 19. Tuy nhiên, việc lựa chọn mô hình hồi quy tuyến tính sẽ phù hợp với với bài toán vì nó có nhiều lợi thế:

- Đơn Giản và Dễ Hiểu: Mô hình hồi quy tuyến tính có cấu trúc đơn giản và dễ hiểu.
- Đặc điểm của dữ liệu: Dữ liệu có mối quan hệ tuyến tính giữa các kỹ năng và chỉ số "Overall", mô hình hồi quy tuyến tính sẽ hoạt động rất hiệu quả.
- Dễ Dàng Kiểm Định và Đánh Giá: Các chỉ số đánh giá như R^2 và RMSE, giúp ta dễ dàng đánh giá độ chính xác và hiệu suất của mô hình..

Kết Luận: Với những lý do trên, hồi quy tuyến tính là một lựa chọn phù hợp và hiệu quả cho bài toán.

CHƯƠNG 2. CHUẨN BỊ DỮ LIỆU

2.1. Giới thiệu tập dữ liệu

Dataset FIFA 19 là một tập dữ liệu nổi tiếng được cung cấp bởi Kaggle sử dụng để nghiên cứu và phân tích trong lĩnh vực bóng đá. Nó cung cấp thông tin chi tiết về các cầu thủ trong trò chơi FIFA, bao gồm nhiều chỉ số kỹ năng và thông tin cá nhân, giúp người dùng hiểu rõ hơn về khả năng của cầu thủ và đưa ra các dự đoán chính xác về hiệu suất của họ trong trò chơi cũng như trong thực tế.

Quy mô dữ liệu:

- Số lượng bản ghi: Tập dữ liệu bao gồm hơn **18.000** cầu thủ từ khắp nơi trên thế giới.
- Số lượng thuộc tính: Có tổng cộng **88 cột** đại diện cho các thông tin khác nhau của cầu thủ.

Nội dung của Dataset được tổ chức thành nhiều nhóm thuộc tính khác nhau gồm một số cột cơ bản sau:

- Thông tin cầu thủ:
  - Name: Tên cầu thủ.
  - Age: Tuổi của cầu thủ.
  - Height: Chiều cao cầu thủ.
  - Weight: Cân nặng cầu thủ.
  - Nationality: Quốc tịch của cầu thủ.
- Chỉ số kỹ năng:
  - Overall: Chỉ số tổng quan, phản ánh khả năng tổng thể của cầu thủ.
  - Potential: Chỉ số tiềm năng, cho thấy khả năng phát triển trong tương lai.
  - Skill Moves: Số lượng kỹ năng đi bóng cầu thủ có thể thực hiện, thể hiện khả năng sáng tạo trên sân.
  - International Reputation: Đánh giá mức độ nổi tiếng và uy tín của cầu thủ trên đấu trường quốc tế.
  - Sprint Speed: Tốc độ của cầu thủ

- Finishing: kỹ năng dứt điểm
- Các chỉ số kỹ năng khác: Bao gồm Dribbling, Shooting, Passing, Defending, Physical, giúp phân tích kỹ lưỡng hơn về khả năng thi đấu của cầu thủ.

- Thông tin về đội bóng:
  - Club: Đội bóng hiện tại của cầu thủ, ảnh hưởng đến lối chơi và chiến thuật.
  - Position: Vị trí thi đấu của cầu thủ, ảnh hưởng đến vai trò trong đội hình.
- Thông tin tài chính:
  - Value: Giá trị thị trường của cầu thủ, phản ánh độ hot và tiềm năng trong thị trường chuyển nhượng.
  - Wage: Mức lương cầu thủ nhận được, thể hiện sự đền bù cho khả năng thi đấu của họ.

2.2. Tiền xử lý và trực quan hóa dữ liệu

2.2.1. Tiền xử lý dữ liệu

Trước tiên, chúng ta cần phải **xóa các cột không cần thiết** vì nó chỉ mang tính chất nhận diện hoặc thông tin không liên quan trực tiếp đến các bài toán phân tích, không có ý nghĩa định lượng hoặc định tính. Việc giảm số lượng của các này cột này sẽ giúp tăng hiệu quả tính toán, dễ dàng xử lý và trực quan hóa trong việc phân tích dữ liệu hoặc xây dựng mô hình học máy.

Các cột cần xóa như:

- ‘Photo’, ‘Flag’, ‘Club Logo’: Các cột này chứa hình ảnh đại diện của cầu thủ, quốc gia và logo câu lạc bộ không ảnh hưởng đến các bài toán liên quan đến phân tích hiệu suất hoặc giá trị cầu thủ.
- ‘Jersey Number’: Cột này chứa số áo của cầu thủ. Đây là thông tin mang tính nhận dạng hoặc thẩm mỹ, không liên quan đến hiệu suất hoặc khả năng thi đấu.

- ‘Loaned From’: Cột này cho biết cầu thủ được cho mượn từ câu lạc bộ nào. Thông tin này còn thiếu khá nhiều và cũng không tác động trực tiếp đến chỉ số chuyên môn hoặc giá trị của cầu thủ.
- ‘Real Face’: Cột này cho biết khuôn mặt của cầu thủ trong game có giống ngoài đời thật không.
- ‘Release Clause’: Cột này chứa giá trị điều khoản giải phóng hợp đồng của cầu thủ. Mặc dù có thể hữu ích trong một số bài toán liên quan đến tài chính tuy nhiên cột này trùng lặp với cột Value, Wage và không phù hợp với bài toán hiện tại.
- 'LS', 'ST', 'RS', 'LW', 'LF', 'CF', 'RF', 'RW', 'LAM', 'CAM', 'RAM', 'LM', 'LCM', 'CM', 'RCM', 'RM', 'LWB', 'LDM', 'CDM', 'RDM', 'RWB', 'LB', 'LCB', 'CB', 'RCB', 'RB': Các cột này chứa chỉ số kỹ năng của cầu thủ khi thi đấu ở từng vị trí cụ thể. Thông tin từ các cột này đã được phản ánh đầy đủ qua các chỉ số tổng quát khác như Overall, Potential. Việc giữ các cột lẻ có thể làm tăng độ phức tạp của dữ liệu, ngoài ra các cột này còn bị thiếu khá nhiều dữ liệu, khiến cho mô hình có thể tăng thêm chi phí khi xử lý các cột bị thiếu.

```
print(f'Trước khi xóa các cột không cần thiết: {fifa.shape[1]}')

fifa.drop(['Photo','Flag','Club Logo', 'Jersey Number', 'Loaned From', 'Real Face',
           'Release Clause', 'LS', 'ST', 'RS', 'LW', 'LF', 'CF', 'RF', 'RW', 'LAM', 'CAM', 'RAM', 'LM', 'LCM', 'CM',
           'RCM', 'RM', 'LWB', 'LDM', 'CDM', 'RDM', 'RWB', 'LB', 'LCB', 'CB', 'RCB', 'RB'], axis=1,inplace=True)

print(f'Sau khi xóa các cột không cần thiết: {fifa.shape[1]}')
```

- Trước khi xóa các cột không cần thiết: 88 cột
- Sau khi xóa các cột không cần thiết: 55 cột

Sau khi loại bỏ các cột không cần thiết, tập dữ liệu được rút gọn và tập trung vào các thông tin quan trọng. Tiếp theo chúng ta tiến hành **phân tích và làm sạch tập dữ liệu** để mô hình có thể hoạt động tốt hơn và đưa ra dự đoán chính xác:

1) Kiểm tra dữ liệu tổng quan:

- Sử dụng lệnh info(): trên tập dữ liệu fifa để ta thấy tổng quan tập dữ liệu có 18207 dòng và 55 cột. Trong đó có 37 cột thuộc kiểu dữ liệu float64, 5 cột thuộc kiểu int64, 13 cột kiểu object. Số lượng dòng trên mỗi cột không đủ

18207 dòng sẽ ảnh hưởng đến quá trình huấn luyện mô nên dữ liệu thu thập cần phải xử lý lại.

- Sử dụng lệnh describe() trên tập dữ liệu fifa để ta thấy các giá trị thống kê từ những cột có kiểu dữ liệu là số nguyên hay số thực. Trong đó ta có thể thấy giá trị trung bình, độ lệch chuẩn trong một số cột có sự chênh lệch nhau khá lớn có thể sẽ ảnh hưởng đến quá trình huấn luyện mô nên dữ liệu thu thập cần phải xử lý lại.

2) Kiểm tra và xử lý các giá trị trùng lặp

Khi kiểm tra các giá trị trùng lắp trong các cột ta có thể thấy tất cả các cột đều có khả năng bị trùng lắp dữ liệu ngoại trừ cột 'ID'. Điều này là hợp lý vì các giá trị như chỉ số kỹ năng, tuổi, quốc gia,… trên thực tế có thể trùng nhau.

Tuy nhiên, sự trùng hợp trên cột 'Name' là một trường hợp đặc biệt do quá trình thu thập dữ liệu có khả năng thu thập dữ liệu cầu thù đó nhiều lần gây ra dữ liệu nhiễu khi đưa vào mô hình. Tuy nhiên, thực tế khả năng trùng tên của mọi người cũng có khả năng cao xảy ra. Vì thế ta phải xét thêm các điều kiện khác như tuổi, quốc gia để xác định chính xác có phải cùng một người hay không để tránh làm mất đi các dữ liệu quan trọng đáng quý.

```
print(f'Trước khi lọc tên bị trùng: {fifa.shape[0]}')

# Lọc ra các bản ghi có trùng tên
list_name = fifa[fifa['Name'].duplicated(keep=False)]
print(f'Sau khi lọc tên bị trùng: {list_name["Name"].count()}')

# Lọc các bản ghi có trùng cả Name và Nationality
duplicate_players = list_name[list_name.duplicated(subset=['Name', 'Nationality'], keep=False)]
print(f'Sau khi lọc tên và quốc gia cùng bị trùng: {duplicate_players["Name"].count()}')

# Lọc các bản ghi có trùng cả Name và Nationality và Age
duplicate_players = duplicate_players[duplicate_players.duplicated(subset=['Name', 'Nationality', 'Age'], keep=False)]
print(f'Sau khi lọc tên và quốc gia và tuổi cùng bị trùng: {duplicate_players["Name"].count()}')

# Sắp xếp kết quả theo Name và Nationality
sorted_duplicates = duplicate_players.sort_values(by=['Name', 'Nationality'])
print(sorted_duplicates[['Name', 'Nationality', 'Age']])
```

Ta có được kết quả:

- Trước khi lọc tên bị trùng: 18207
- Sau khi lọc tên bị trùng: 1775
- Sau khi lọc tên và quốc gia cùng bị trùng: 956
- Sau khi lọc tên và quốc gia và tuổi cùng bị trùng: 78

Sau khi thực hiện lọc cột ‘Name’, ‘National’, ‘Age’ có cùng các giá trị trùng lặp ta thu được kết quả còn 78 dòng. Từ đó, ta có thể xử lý các giá trị trùng lặp này bằng cách xóa nó đi.

```
print(f'Trước khi xóa tên bị trùng: {fifa.shape[0]}')
fifa.drop_duplicates(subset ="Name",  keep = False, inplace = True)
print(f'Sau khi xóa tên bị trùng: {fifa.shape[0]}')
```

- Trước khi xóa cột bị trùng: 18207 dòng
- Sau khi xóa cột bị trùng: 18129 dòng

3) Kiểm tra và xử lý các giá trị còn thiếu

Xác định các cột hoặc hàng có giá trị bị thiếu (NaN) là một bước rất quan trọng trong tiền xử lý dữ liệu. Khi kiểm tra các cột bị thiếu trong các cột ta có thể thấy dữ liệu còn bị thiếu khá nhiều 46/55 cột bị thiếu trong đó cột ‘Joined’ là cột thiếu nhiều nhất.

Xử lý các trị còn thiếu đối với hai kiểu dữ liệu:

- Đối với các cột thuộc tính dạng chuỗi:
  - Nếu cầu thủ không có câu lạc bộ (Club), điền giá trị 'No Club'.
  - Nếu không có thông tin về chân thuận (Preferred Foot), điền 'Right' (chân phải).
  - Nếu thiếu thông tin về Work Rate, điền 'Medium/ Medium'.
  - Nếu thiếu Body Type, điền 'Normal', và thay thế những giá trị đặc biệt như 'C. Ronaldo', 'Messi', v.v. bằng 'Normal'.
  - Nếu không có vị trí (Position), mặc định điền 'ST' (Striker - Tiền đạo).
- Đối với các cột dạng số: Sẽ được điền bằng giá trị trung bình (mean) của cột đó. Vì mean đại diện cho mức trung bình, giúp duy trì phân phối dữ liệu nhằm tránh bỏ sót thông tin quan trọng.

```
fifa['Club'].fillna('No Club', inplace = True)
fifa['Preferred Foot'].fillna('Right', inplace = True)
fifa['International Reputation'].fillna(fifa['International Reputation'].mean(), inplace= True)
fifa['Weak Foot'].fillna(fifa['Weak Foot'].mean(), inplace = True)
fifa['Skill Moves'].fillna(fifa['Skill Moves'].mean(), inplace = True)
fifa['Work Rate'].fillna('Medium/ Medium', inplace = True)
fifa['Body Type'].fillna('Normal', inplace = True)
fifa.loc[fifa['Body Type'] == 'C. Ronaldo', 'Body Type'] = 'Normal'
fifa.loc[fifa['Body Type'] == 'Courtois', 'Body Type'] = 'Normal'
fifa.loc[fifa['Body Type'] == 'Messi', 'Body Type'] = 'Normal'
fifa.loc[fifa['Body Type'] == 'Neymar', 'Body Type'] = 'Normal'
fifa.loc[fifa['Body Type'] == 'PLAYER_BODY_TYPE_25', 'Body Type'] = 'Normal'
fifa.loc[fifa['Body Type'] == 'Shaqiri', 'Body Type'] = 'Normal'
fifa.loc[fifa['Body Type'] == 'Akinfenwa', 'Body Type'] = 'Normal'
fifa.loc[fifa['Body Type'] == 'normal', 'Body Type'] = 'Normal'
fifa['Position'].fillna('ST', inplace = True)
fifa['Crossing'].fillna(fifa['Crossing'].mean(), inplace = True)
fifa['Finishing'].fillna(fifa['Finishing'].mean(), inplace = True)
fifa['HeadingAccuracy'].fillna(fifa['HeadingAccuracy'].mean(), inplace = True)
fifa['ShortPassing'].fillna(fifa['ShortPassing'].mean(), inplace = True)
fifa['Volleys'].fillna(fifa['Volleys'].mean(), inplace = True)
fifa['Dribbling'].fillna(fifa['Dribbling'].mean(), inplace = True)
fifa['Curve'].fillna(fifa['Curve'].mean(), inplace = True)
fifa['FKAccuracy'].fillna(fifa['FKAccuracy'].mean(), inplace = True)
fifa['LongPassing'].fillna(fifa['LongPassing'].mean(), inplace = True)
fifa['BallControl'].fillna(fifa['BallControl'].mean(), inplace = True)
fifa['Acceleration'].fillna(fifa['Acceleration'].mean(), inplace = True)
fifa['SprintSpeed'].fillna(fifa['SprintSpeed'].mean(), inplace = True)
fifa['Agility'].fillna(fifa['Agility'].mean(), inplace = True)
fifa['Reactions'].fillna(fifa['Reactions'].mean(), inplace = True)
fifa['Balance'].fillna(fifa['Balance'].mean(), inplace = True)
fifa['ShotPower'].fillna(fifa['ShotPower'].mean(), inplace = True)
fifa['Jumping'].fillna(fifa['Jumping'].mean(), inplace = True)
fifa['Stamina'].fillna(fifa['Stamina'].mean(), inplace = True)
fifa['Strength'].fillna(fifa['Strength'].mean(), inplace = True)
fifa['LongShots'].fillna(fifa['LongShots'].mean(), inplace = True)
fifa['Aggression'].fillna(fifa['Aggression'].mean(), inplace = True)
fifa['Interceptions'].fillna(fifa['Interceptions'].mean(), inplace = True)
fifa['Positioning'].fillna(fifa['Positioning'].mean(), inplace = True)
fifa['Vision'].fillna(fifa['Vision'].mean(), inplace = True)
fifa['Penalties'].fillna(fifa['Penalties'].mean(), inplace = True)
fifa['Composure'].fillna(fifa['Composure'].mean(), inplace = True)
```

```
fifa['Marking'].fillna(fifa['Marking'].mean(), inplace = True)
fifa['StandingTackle'].fillna(fifa['StandingTackle'].mean(), inplace = True)
fifa['SlidingTackle'].fillna(fifa['SlidingTackle'].mean(), inplace = True)
fifa['GKDiving'].fillna(fifa['GKDiving'].mean(), inplace = True)
fifa['GKHandling'].fillna(fifa['GKHandling'].mean(), inplace = True)
fifa['GKKicking'].fillna(fifa['GKKicking'].mean(), inplace = True)
fifa['GKPositioning'].fillna(fifa['GKPositioning'].mean(), inplace = True)
fifa['GKReflexes'].fillna(fifa['GKReflexes'].mean(), inplace = True)
```

4) Xử lý và chuyển đổi đơn vị đo, tiền tệ

Trong tập dữ liệu FIFA, một vài cột chứa dữ liệu ở các định dạng khác nhau (object), không trực tiếp phù hợp để phân tích hoặc xử lý. Vì vậy, việc chuẩn hóa và chuyển đổi các đơn vị đo lường, tiền tệ là cần thiết để dễ dàng phân tích và trực quan hóa dữ liệu.

```
print(fifa[['Value', 'Wage', 'Height','Weight', 'Contract Valid Until']].head(2))
print("Kiểu dữ liệu của các cột trên:")
print(fifa[['Value', 'Wage', 'Height','Weight', 'Contract Valid Until']].dtypes)
```

```
     Value   Wage Height  Weight Contract Valid Until
0  €110.5M  €565K    5'7  159lbs                 2021
1    €77M  €405K    6'2  183lbs                 2022
Kiểu dữ liệu của các cột trên:
Value                  object
Wage                   object
Height                 object
Weight                 object
Contract Valid Until   object
dtype: object
```

- Chuyển đổi đơn vị tiền tệ: Trong hai cột 'Value', 'Wage' chứa các ký hiệu như €, K, M ta sẽ biến đổi thành (đơn vị nghìn, triệu $).
- Chuyển đổi đơn vị đo cân nặng và chiều cao: Trong cột ‘Height’, ‘Weight’ sử dụng đơn vị (feet-inch, pound), cần chuyển đổi sang đơn vị chuẩn như cm và kg vì hai đơn vị này phổ biến hơn ở Việt Nam.
- Chuyển đổi đơn vị năm: Trong cột ‘Contract Valid Until’ tuy một số cột chỉ chứa năm nhưng còn một số cột chưa dữ liệu là ngày tháng năm vd: 'May 31, 2020’ nên ta phải chuyển đổi về cùng một dạng là năm.

Sau khi xử xong việc chuyển đổi dữ liệu, việc phân tích và trực quan hóa dữ liệu sẽ trở nên đơn giản hơn.

```
print('Sau khi thực hiện chuyển đổi dữ liệu:')
print(fifa[['Value', 'Wage', 'Height','Weight', 'Joined', 'Contract Valid Until']].head())
```

```
Sau khi thực hiện chuyển đổi dữ liệu:
       Value    Wage  Height  Weight Joined Contract Valid Until
0  110000000  565000     174      72   2004                 2021
1   77000000  405000     189      83   2018                 2022
2  118000000  290000     180      68   2017                 2022
3   72000000  260000     195      76   2011                 2020
4  102000000  355000     156      70   2015                 2023
```

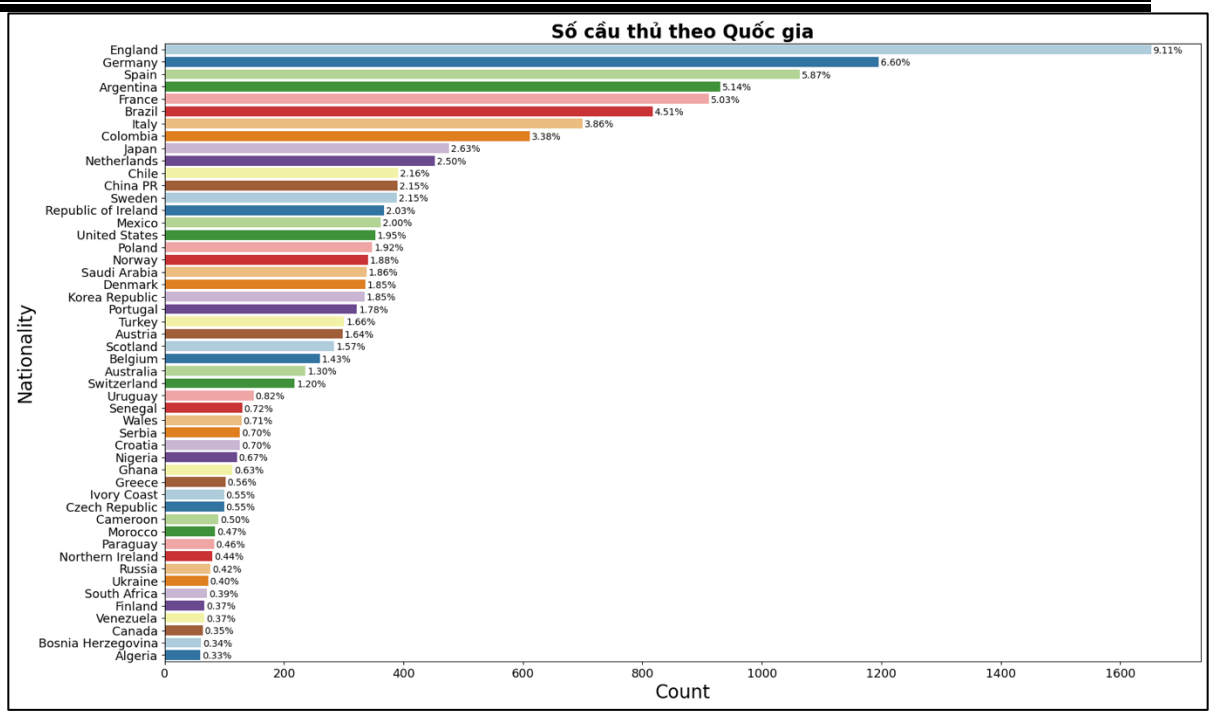
5) Chuyển đổi các trị thành kiểu số nguyên

Để đảm bảo dữ liệu phù hợp với mô hình học máy, các cột số liệu về chỉ số đang là dữ liệu số float64 cần được chuyển đổi về dạng kiểu int64. Việc chuyển đổi giúp tối ưu hóa hiệu suất xử lý và đảm bảo rằng tất cả các cột liên quan đến thông tin định lượng được đưa vào mô hình đều có định dạng thống nhất.

```
#Đổi kiểu dữ liệu thành int phục vụ cho việc chạy model
fifa['International Reputation']= fifa['International Reputation'].astype('int64')
fifa['Weak Foot']=fifa['Weak Foot'].astype('int64')
fifa['Skill Moves']=fifa['Skill Moves'].astype('int64')
fifa['Crossing']=fifa['Crossing'].astype('int64')
fifa['Finishing']=fifa['Finishing'].astype('int64')
fifa['HeadingAccuracy']=fifa['HeadingAccuracy'].astype('int64')
fifa['ShortPassing']=fifa['ShortPassing'].astype('int64')
fifa['Volleys']=fifa['Volleys'].astype('int64')
fifa['Dribbling']=fifa['Dribbling'].astype('int64')
fifa['Curve']=fifa['Curve'].astype('int64')
fifa['FKAccuracy']=fifa['FKAccuracy'].astype('int64')
fifa['LongPassing']=fifa['LongPassing'].astype('int64')
fifa['BallControl']=fifa['BallControl'].astype('int64')
fifa['Acceleration']=fifa['Acceleration'].astype('int64')
fifa['SprintSpeed']=fifa['SprintSpeed'].astype('int64')
fifa['Agility']=fifa['Agility'].astype('int64')
fifa['Reactions']=fifa['Reactions'].astype('int64')
fifa['Balance']=fifa['Balance'].astype('int64')
fifa['ShotPower']=fifa['ShotPower'].astype('int64')
fifa['Jumping']=fifa['Jumping'].astype('int64')
fifa['Stamina']=fifa['Stamina'].astype('int64')
fifa['Strength']=fifa['Strength'].astype('int64')
fifa['LongShots']=fifa['LongShots'].astype('int64')
fifa['Aggression']=fifa['Aggression'].astype('int64')
fifa['Interceptions']=fifa['Interceptions'].astype('int64')
fifa['Positioning']=fifa['Positioning'].astype('int64')
fifa['Vision']=fifa['Vision'].astype('int64')
fifa['Penalties']=fifa['Penalties'].astype('int64')
fifa['Composure']=fifa['Composure'].astype('int64')
fifa['Marking']=fifa['Marking'].astype('int64')
fifa['StandingTackle']=fifa['StandingTackle'].astype('int64')
fifa['SlidingTackle']=fifa['SlidingTackle'].astype('int64')
fifa['GKDiving']=fifa['GKDiving'].astype('int64')
fifa['GKHandling']=fifa['GKHandling'].astype('int64')
fifa['GKKicking']=fifa['GKKicking'].astype('int64')
fifa['GKPositioning']=fifa['GKPositioning'].astype('int64')
fifa['GKReflexes']=fifa['GKReflexes'].astype('int64')
```

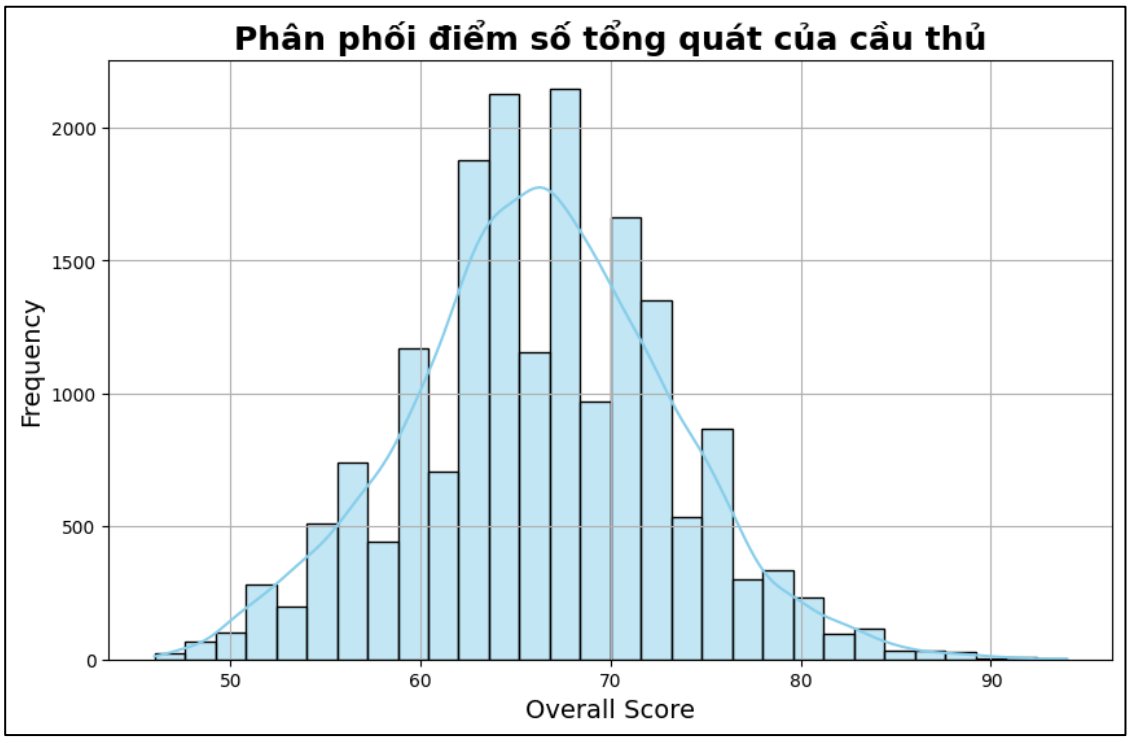
2.2.2. Trực quan hóa dữ liệu

Thống kê xếp hạng các quốc gia có số lượng cầu thủ từ cao đến thấp và tính toán hiển thị số phần trăm của quốc gia đó với tổng các quốc gia:



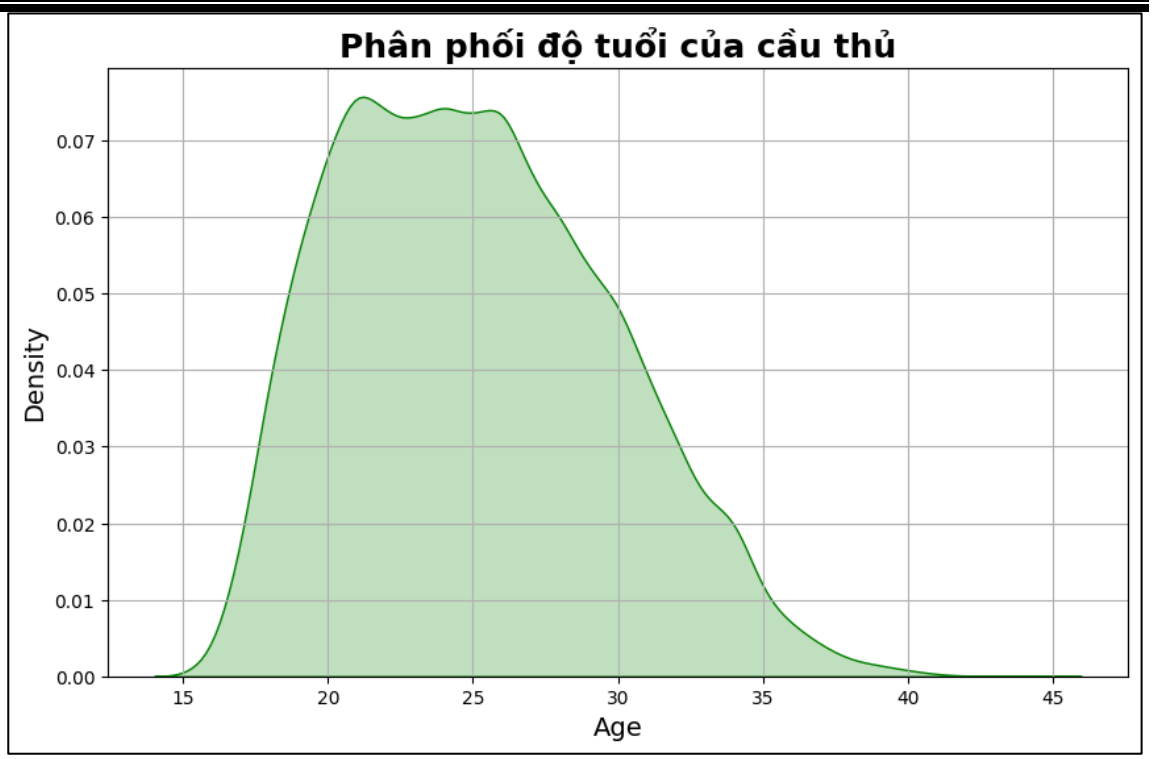
Hình 2.1. Trực quan hóa số lượng cầu thủ theo quốc gia

Thống kê số điểm đánh giá chỉ số cầu thủ bằng biểu đồ Histogram:



Hình 2.2. Trực quan hóa số điểm chỉ số cầu thủ

Thống kế độ tuổi của cầu thủ bằng biểu đồ phân phối mật độ xác suất (density plot):

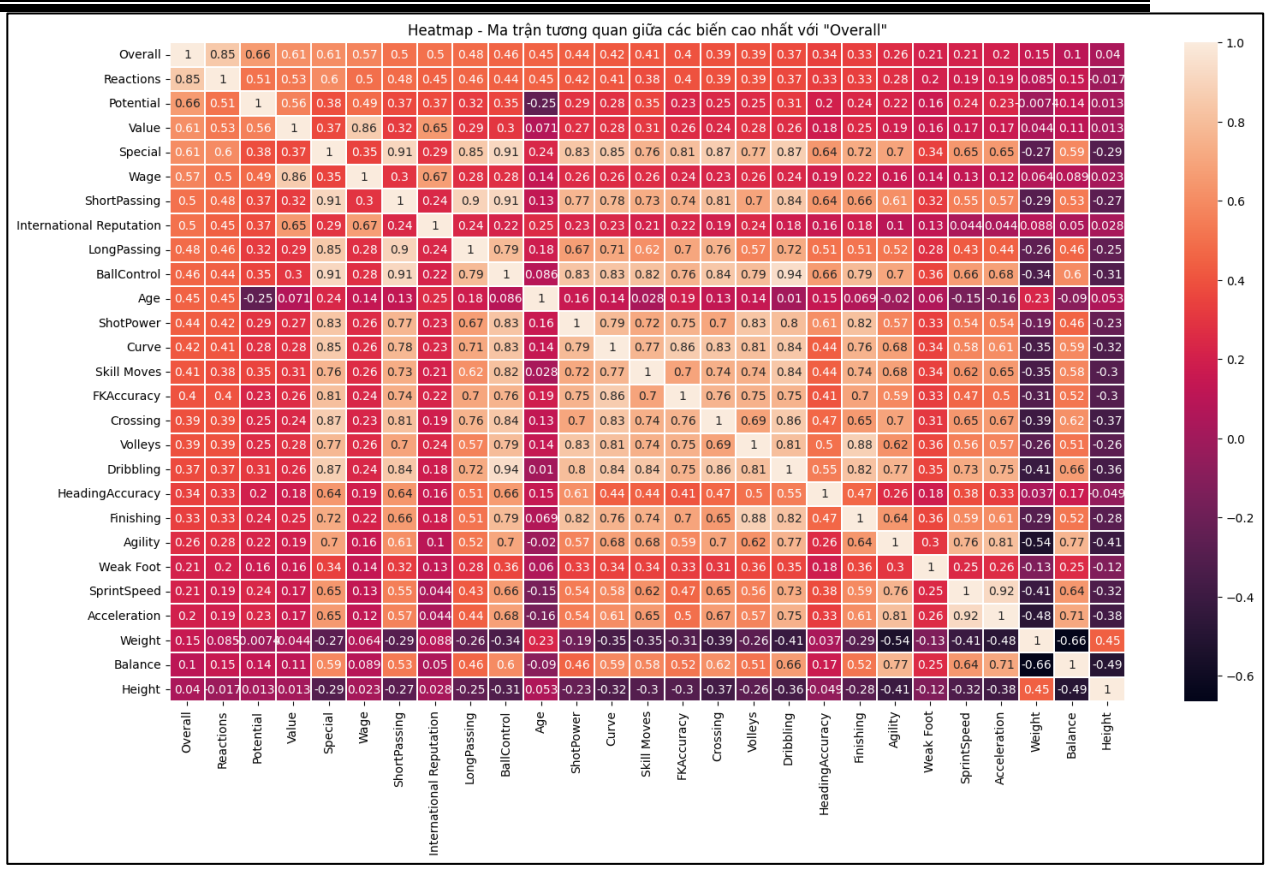


Hình 2.3. Trực quan hóa độ tuổi của cầu thủ

2.3. Trích chọn đặc trưng

Trích chọn đặc trưng là một bước quan trọng trong việc chuẩn bị dữ liệu để đưa vào mô hình học máy. Mục tiêu chính là biến đổi dữ liệu thô thành các đặc trưng có ý nghĩa, giúp mô hình học được quy luật và đưa ra dự đoán chính xác. Việc trích chọn đặc trưng sẽ giúp giảm độ phức tạp của dữ liệu để tập trung vào các thông tin hữu ích nhằm loại bỏ nhiễu và tăng hiệu quả tính toán, cải thiện hiệu suất của mô hình.

Chúng ta sẽ sử dụng biểu đồ heatmap để phân tích mối tương quan giữa Overall (đánh giá tổng quát của cầu thủ) và các biến khác, từ đó chọn các đặc trưng liên quan mạnh nhất.



Hình 2.4. Biểu đồ heatmap thể hiện độ tương quan với 'Overall'

Heatmap cho thấy mức độ tương quan giữa các biến trong dữ liệu với biến ‘Overall’:

- Các biến như Potential, Value, và Special có tương quan cao với Overall, thể hiện rằng tiềm năng, giá trị thị trường và kỹ năng đặc biệt là những yếu tố quan trọng quyết định đánh giá tổng quát của cầu thủ.
- Các đặc trưng như 'Finishing', 'Agility',... có thể được cho là có tính chất rất đặc thù hoặc tương quan cao với các đặc trưng khác, do đó không cần giữ chúng trong mô hình.
- Các biến liên quan đến thủ môn hoặc đặc điểm thể chất (như chiều cao, cân nặng) có tương quan thấp hoặc không đáng kể.

CHƯƠNG 3. XÂY DỰNG MÔ HÌNH

3.1. Tổng quan mô hình hồi quy tuyến tính

3.1.1. Khái niệm

Hồi quy tuyến tính là một phương pháp thống kê được sử dụng để mô hình hóa mối quan hệ giữa một biến phụ thuộc và một hoặc nhiều biến độc lập. Mục tiêu của hồi quy tuyến tính là tìm ra một đường thẳng (hoặc mặt phẳng trong trường hợp nhiều biến độc lập) tốt nhất mô tả mối quan hệ này.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n + \epsilon$$

Trong đó:

- y: Biến phụ thuộc (mục tiêu).
- x: Biến độc lập.
- β: Hệ số chặn (intercept).
- β n: Hệ số hồi quy (slope).
- ϵ : Sai số ngẫu nhiên.

3.1.2. Ước Lượng Hệ Số Hồi Quy

Để ước lượng các hệ số hồi quy, phương pháp bình phương tối thiểu (Ordinary Least Squares - OLS) được sử dụng. Mục tiêu là giảm thiểu tổng bình phương sai số giữa giá trị thực tế và giá trị dự đoán:

$$\text{SSE} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Trong đó:

- yi : là giá trị thực tế.
- $\hat{y}_i$ là giá trị dự đoán từ mô hình

3.1.3. Dự Đoán

Dựa trên các hệ số hồi quy ước lượng, giá trị của biến phụ thuộc có thể được dự đoán bằng công thức:

$$\hat{y} = Xw$$

Trong đó:

- $\hat{y}_i$ là giá trị dự đoán từ mô hình

3.1.4. Thước đo đánh giá mô hình

Các chỉ số đánh giá hiệu suất của mô hình hồi quy bao gồm:

1) **R^2 (Hệ số xác định)**: Cho biết tỷ lệ biến thiên của biến phụ thuộc được giải thích bởi các biến độc lập. Giá trị R^2 nằm trong khoảng [0, 1], với giá trị càng gần 1 thì mô hình càng tốt.

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

Trong đó:

- $SSres = \sum(\text{yi} - \hat{\text{y}}_\text{i})^2$ là tổng bình phương sai số giữa giá trị thực tế và giá trị dự đoán.
- $SStot = \sum(\text{yi} - \bar{\text{y}})^2$ là tổng bình phương sai số giữa giá trị thực tế và giá trị trung bình của nó.

2) **RMSE (Root Mean Square Error)**: Là độ lệch chuẩn của các sai số dự đoán. RMSE cho biết mức độ sai lệch giữa giá trị thực tế và giá trị dự đoán. Giá trị càng nhỏ thì mô hình càng chính xác.

$$RMSE = \sqrt{\frac{1}{n}\sum(y_i - \hat{y}_i)^2}$$

3.1.5. Độ Quan Trọng Của Các Đặc Trưng

Trong hồi quy tuyến tính, độ quan trọng của các biến độc lập có thể được đánh giá thông qua độ lớn của các hệ số hồi quy:

- Các hệ số hồi quy cho biết mức độ ảnh hưởng của từng biến độc lập đến biến phụ thuộc.
- Độ lớn của hệ số cho biết rằng khi biến độc lập thay đổi một đơn vị, biến phụ thuộc sẽ thay đổi bao nhiêu đơn vị.

3.2. Cài đặt mô hình hồi quy tuyến tính trên Python

3.2.1. Yêu cầu về hệ thống và thư viện

1) Python

Python là một ngôn ngữ thông dịch, điều này nghĩa là ngôn ngữ này trực tiếp chạy từng dòng mã. Sử dụng từ ngữ giống trong tiếng Anh gần gũi với ngôn ngữ con người hơn các ngôn ngữ lập trình khác. Không giống như các ngôn ngữ lập trình khác. Python là một ngôn ngữ cấp cao. Do đó, các lập trình viên không cần phải lo lắng về những chức năng cơ bản của nó như kiến trúc và quản lý bộ nhớ.

Trong hệ thống của chúng tôi sử dụng phiên bản Python 3.10.12.

Hình 3.1. Hình minh họa python

2) Google Colaboratory

Colaboratory hay còn gọi là Google Colab, là một sản phẩm từ Google Research, nó cho phép thực thi Python trên nền tảng đám mây. Google Colab cho phép tạo và chạy các Jupyter Notebooks trực tuyến mà không cần cài đặt môi trường phát triển phức tạp trên máy tính cá nhân. Tương tự như Jupyter Notebook truyền thống với các cell cho phép thực thi mã Python hoặc viết markdown để tạo nội dung hướng dẫn. Ngoài ra cùng với thuật toán đề xuất code cực kì nhanh chóng và chính xác.

Hình 3.2. Hình minh họa google Colab

Google Colab cung cấp truy cập miễn phí đến GPU và TPU, đặc biệt hữu ích cho các tác vụ tính toán nặng về mặt số học, đặc biệt là trong lĩnh vực học máy và deep learning.

3) Các thư viện cần thiết

Matplotlib: Các nhà phát triển sử dụng Matplotlib để hiển thị dữ liệu dưới dạng đồ họa hai và ba chiều (2D và 3D) chất lượng cao. Thư viện này thường được sử dụng trong các ứng dụng khoa học

Seaborn: Thư viện vẽ biểu đồ và trực quan hóa dữ liệu, mở rộng matplotlib với các biểu đồ trực quan và dễ sử dụng. Trong thị giác máy tính, nó hữu ích cho việc phân tích và biểu diễn dữ liệu.

Pandas: cung cấp cấu trúc dữ liệu được tối ưu hóa và linh hoạt mà có thể sử dụng để thao tác với dữ liệu chuỗi thời gian và dữ liệu có cấu trúc, chẳng hạn như bảng và nhóm.

NumPy: là một thư viện phổ biến mà các nhà phát triển sử dụng để dễ dàng tạo và quản lý nhóm, thao tác với các hình dạng logic và thực hiện các phép toán đại số tuyến tính.

3.2.2. Triển khai mô hình hồi quy tuyến tính

Trong phần này, chúng ta sẽ cài đặt mô hình hồi quy tuyến tính bội để dự đoán chỉ số tổng thể (Overall) của cầu thủ trong dữ liệu FIFA. Các bước làm từ việc xử lý dữ liệu, thêm cột bias, đến tính toán trọng số w bằng cách sử dụng phương pháp ma trận:

- Bước 1: Chuẩn bị dữ liệu

Tách biến mục tiêu, loại bỏ các biến không cần thiết và xử lý các biến phân loại. nhằm giữ lại các đặc trưng cần thiết cho mô hình, loại bỏ biến không liên quan để tăng tính chính xác.

```
target = df['Overall']
df2=df.drop(['Overall','Dribbling','HeadingAccuracy','Finishing','Agi
lity','Weak Foot', 'SprintSpeed', 'Acceleration', 'Weight', 'Balance',
'Height'], axis=1)
```

Chia dữ liệu thành tập 80% huấn luyện và 20% kiểm tra, đảm bảo mô hình được đánh giá trên dữ liệu chưa từng thấy:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(df2, target,
test_size=0.2)
```

- Bước 2: Xử lý các biến phân loại và các trường hợp khác

Chuyển đổi các biến phân loại thành dạng nhị phân One-hot encoding để mô hình hồi quy tuyến tính có thể xử lý.

```
X_train = pd.get_dummies(X_train)
X_test = pd.get_dummies(X_test)
```

Sử dụng hàm align() để đảm bảo cốt cột giữa tập huấn luyện và kiểm tra để tránh lỗi

```
X_train, X_test = X_train.align(X_test, join='left', axis=1,
fill_value=0)
```

- Bước 3: Thêm cột bias

Cột chứa các giá trị 1 được thêm vào cả tập huấn luyện và tập kiểm tra để mô hình có thêm hệ số chặn (bias) giúp mô hình dự đoán chính xác hơn trong phương trình hồi quy.

```
X_train = np.column_stack((np.ones(X_train.shape[0]), X_train))
X_test = np.column_stack((np.ones(X_test.shape[0]), X_test))
```

- Bước 4: Tính toán trọng số (w)

Sử dụng phương pháp ma trận để tính trọng số của các đặc trưng. Phương pháp ma trận này giúp tìm trọng số tối ưu để giảm thiểu sai số bình phương giữa giá trị dự đoán và giá trị thực tế.

```
w = np.linalg.inv(X_train.T.dot(X_train)).dot(X_train.T).dot(y_train)
```

- Bước 5: Dự đoán và Đánh giá mô hình

```
predictions = X_test.dot(w)
```

Tính toán giá trị dự đoán từ ma trận X_test và trọng số w.

Sử dụng R^2 và RMSE đánh giá: R^2 đánh giá mức độ mô hình giải thích được biến mục tiêu, còn RMSE đo độ lệch trung bình giữa dự đoán và thực tế.

```
# Tính r2 score
ss_res = np.sum((y_test - predictions) ** 2)  # residual sum of squares
ss_tot = np.sum((y_test - np.mean(y_test)) ** 2)  # total sum of squares
r2 = 1 - (ss_res / ss_tot)
print('r2 score:', r2)

# Tính RMSE
rmse = np.sqrt(np.mean((y_test - predictions) ** 2))
print('RMSE:', rmse)
```

- Bước 6: Phân tích độ quan trọng của các đặc trưng

Tính giá trị tuyệt đối của các trọng số w và sắp xếp theo thứ tự giảm dần nhằm đo lường độ quan trọng của các đặc trưng dựa trên trọng số tuyệt đối. Các đặc trưng có trọng số lớn hơn có ảnh hưởng nhiều hơn đến biến mục tiêu.

```
importances = np.abs(w[1:])

feature_names = df2.columns
sorted_indices = np.argsort(importances)[::-1]

print("Độ quan trọng của các đặc trưng:")
for idx in sorted_indices:
    print(f"{feature_names[idx]}: {importances[idx]}")
```

3.3. Hồi quy tuyến tính từ thư viện scikit-learn

1) Chuẩn bị dữ liệu

Tách biến mục tiêu và biến đặc trưng: Tương tự như cách hồi quy tự cài đặt, biến mục tiêu Overall được tách ra khỏi dữ liệu và các biến không cần thiết được loại bỏ.

Phân chia tập huấn luyện và kiểm tra: Sử dụng train_test_split() để chia dữ liệu.

Biến đổi One-hot Encoding: Các biến phân loại được chuyển thành biến giả (dummy variables) bằng hàm pd.get_dummies(). Điều này chuyển đổi các giá trị phân loại thành dạng ma trận nhị phân (one-hot encoding), giúp mô hình hồi quy tuyến tính xử lý chúng một cách hiệu quả.

2) Sử dụng mô hình LinearRegression từ thư viện scikit-learn

- Bước 1: Tạo Mô Hình

Đầu tiên, cần tạo một đối tượng của lớp LinearRegression từ thư viện scikit-learn: LinearRegression() là một đối tượng đại diện cho mô hình hồi quy tuyến tính. Đây là mô hình mà chúng ta sẽ huấn luyện để dự đoán giá trị của biến mục tiêu (ở đây là Overall).

```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
```

- Bước 2: Huấn Luyện Mô Hình

Tiếp theo, sử dụng hàm fit() để huấn luyện mô hình trên tập huấn luyện, trong đó X_train là các đặc trưng và y_train là biến mục tiêu.

```
model.fit(X_train, y_train)
```

Quá trình fit() sẽ tính toán các trọng số (coefficients) của các đặc trưng, giúp mô hình tìm ra mối quan hệ giữa các đặc trưng và giá trị mục tiêu. Các trọng số này là những giá trị mà mô hình sẽ sử dụng để thực hiện dự đoán.

- Bước 3: Dự Đoán và đánh giá mô hình

Sau khi mô hình được huấn luyện, có thể sử dụng hàm predict() để dự đoán kết quả cho tập kiểm tra (X_test):

```
predictions = model.predict(X_test)
```

Tính R^2 và RMSE: Các giá trị R^2 và RMSE được tính toán tự động bằng cách sử dụng r2_score và mean_squared_error từ thư viện scikit-learn.

```
from sklearn.metrics import mean_squared_error, r2_score
r2 = r2_score(y_test, predictions)
print('R2 score:', r2)
rmse = np.sqrt(mean_squared_error(y_test, predictions))
print('RMSE:', rmse)
```

- Bước 4: Phân tích độ quan trọng của các đặc trưng

Các hệ số trọng số của mô hình được truy xuất từ coef_ của đối tượng hồi quy. Tương tự, các đặc trưng quan trọng cũng được sắp xếp theo giá trị tuyệt đối của trọng số.

```
importances = np.abs(model.coef_)


feature_names = X_train.columns
sorted_indices = np.argsort(importances)[::-1]

print("Độ quan trọng của các đặc trưng:")
for idx in sorted_indices:
    print(f"{feature_names[idx]}: {importances[idx]}")
```

3.4. So sánh hai mô hình

1) So sánh số điểm đánh giá mô hình:

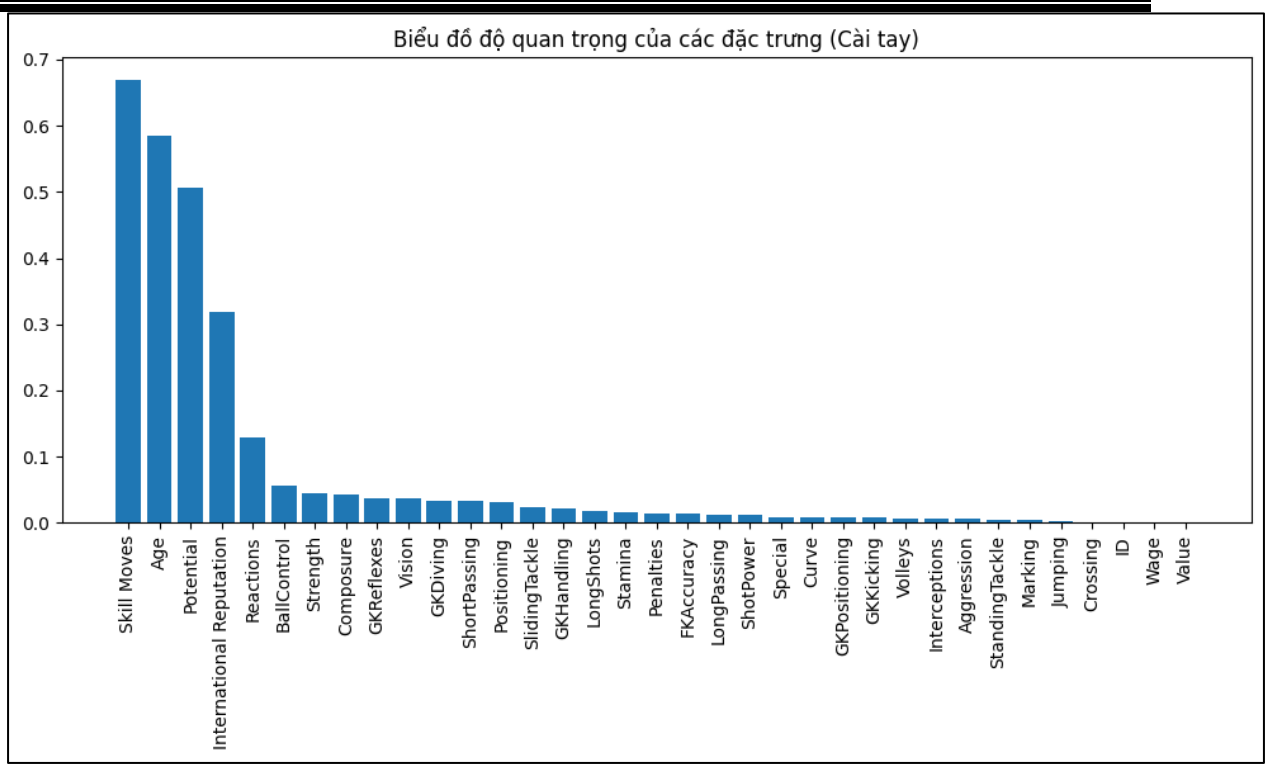
Mô hình hồi quy tự cài đặt đã thể hiện hiệu suất tốt với các chỉ số đánh giá như sau:

- R^2 đạt **0.9287346206364916**: Điều này cho thấy khoảng 92.87% biến thiên của biến mục tiêu có thể được giải thích bởi mô hình, cho thấy mối quan hệ mạnh mẽ giữa các đặc trưng và biến mục tiêu.
- RMSE là **1.8197159201491957**: Giá trị này cho thấy độ lệch trung bình giữa giá trị dự đoán và giá trị thực tế. Một giá trị RMSE thấp cho thấy mô hình đã dự đoán chính xác các giá trị mục tiêu.

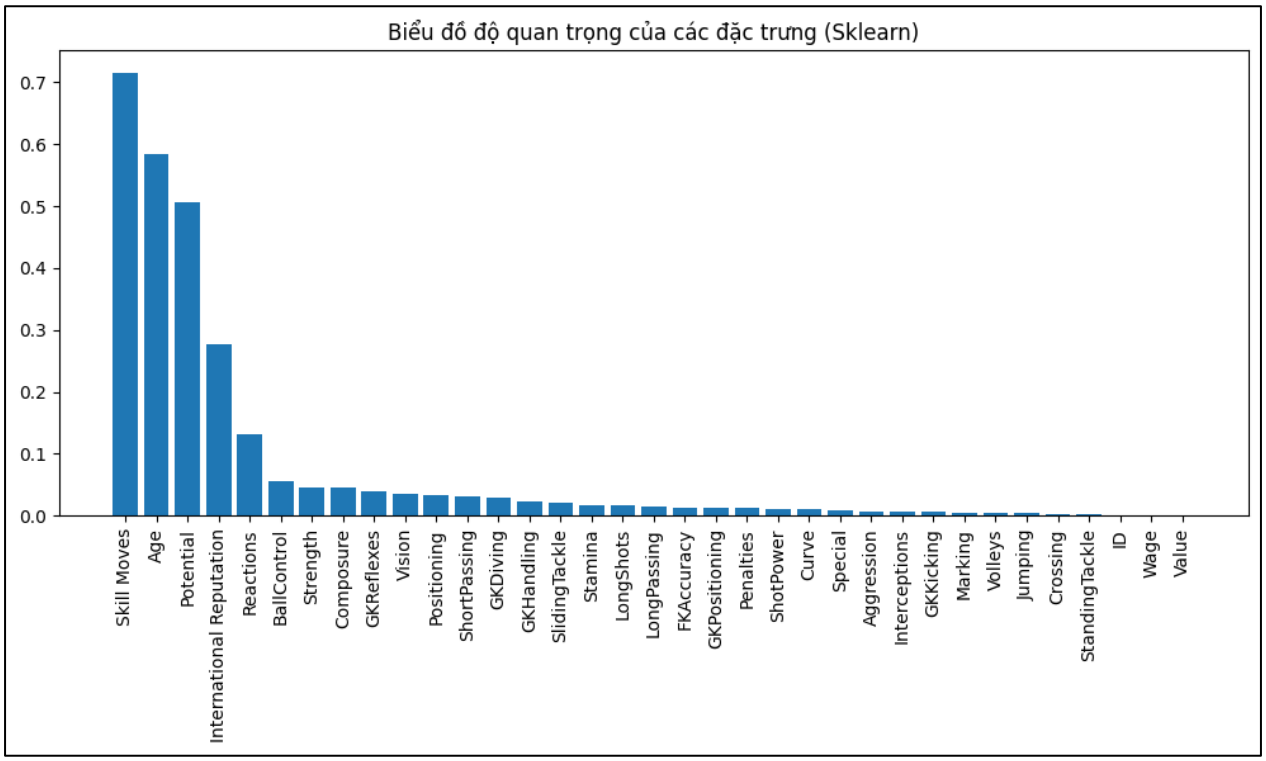
Mô hình hồi quy sử dụng thư viện có chỉ số đánh giá là:

- R^2 là **0.9283811658406068**, cho thấy nó giải thích khoảng 92.83% biến thiên của biến mục tiêu.
- RMSE là **1.8310829102557435**, cho thấy độ lệch trung bình lớn hơn so với mô hình tự cài đặt.

2) So sánh độ quan trọng của các đặc trưng:



Hình 3.3. Biểu đồ thể hiện độ quan trọng của các đặc trưng (Cài tay)



Hình 3.4. Biểu đồ thể hiện độ quan trọng của các đặc trưng (Sklearn)

Kết quả trực quan trên biểu đồ thể hiện độ quan trọng của các đặc trưng cho thấy:

- **Các đặc trưng quan trọng nhất:**
    1. **Skill Moves** (xếp đầu, ~65%-70%): Đây là yếu tố có tác động lớn nhất đến dự đoán.
    2. **Age** (~58%): Tuổi của cầu thủ có tác động quan trọng thứ hai.
    3. **Potential** (~51%): Khả năng tiềm năng của cầu thủ cũng rất quan trọng.
    4. **International Reputation** (~29%): Danh tiếng quốc tế là một yếu tố đáng kể.
- **Đặc trưng có ảnh hưởng thấp hơn nhưng đáng chú ý:**
    1. **Reactions**, **BallControl**, **Strength**, và **Composure**.
    2. Các chỉ số liên quan đến kỹ năng thủ môn như **GKReflexes**, **GKDiving**, và **GKHandling**.
- **Đặc trưng có ảnh hưởng không đáng kể:**
    1. **ID**, **Wage**, và **Value** có giá trị quan trọng gần như bằng 0, chứng tỏ không đóng góp gì nhiều vào mô hình.

3.5. Áp dụng mô hình hồi quy cho dự đoán chỉ số 'Overall'

Sau khi đã xác định rõ 5 chỉ số quan trọng gồm "Age", "Potential", "Skill Moves", "International Reputation" và "Reactions" có ảnh hưởng đến chỉ số "Overall", chúng ta tiến hành áp dụng mô hình hồi quy tuyến tính để dự đoán chỉ số này thông qua hai hàm chính:

1) Hàm manual_linear_regression():

Hàm này xây dựng mô hình hồi quy tuyến tính thủ công bằng cách thêm một cột bias (hệ số tự do) vào ma trận X để đại diện cho hệ số không phụ thuộc.

Sau đó, áp dụng công thức hồi quy tuyến tính $\boldsymbol{w} = (\boldsymbol{X^T X})^{-1} \boldsymbol{X^T y}$. Công thức này cho phép tính toán các hệ số hồi quy (w), biểu thị mức độ ảnh hưởng của từng đặc trưng lên biến đầu ra.

```
def manual_linear_regression(X, y):
    X = np.column_stack((np.ones(X.shape[0]), X))
    w = np.linalg.inv(X.T.dot(X)).dot(X.T).dot(y)
    return w
```

2) Hàm manual_predict():

Hàm này sử dụng ma trận đặc trưng đầu vào (sau khi đã thêm cột bias) và các hệ số hồi quy đã tính được (w) để dự đoán chỉ số "Overall".

Kết quả dự đoán cho thấy mối quan hệ giữa các chỉ số đầu vào và chỉ số "Overall", được tính toán dựa trên mô hình hồi quy tuyến tính.

```
def manual_predict(X, w):
    X = np.column_stack((np.ones(X.shape[0]), X))
    return X.dot(w)
```

CHƯƠNG 4. KẾT QUẢ THỰC NGHIỆM

Kết quả thực nghiệm của mô hình hồi quy tuyến tính tự cài tay so với mô hình từ thư viện sklearn. Dự đoán chỉ số Overall của cầu thủ dựa trên các đặc trưng đầu vào bao gồm "Age" (tuổi), "Potential" (tiềm năng), "Skill Moves" (kỹ năng đi bóng), "International Reputation" (danh tiếng quốc tế), và "Reactions" (phản xạ).

Bảng 4.1. Bảng thực nghiệm kết quả

Name	**Age**	**Potential**	**Skill Moves**	**International Reputation**	**Reactions**	**Overall**	**Overall cài tay**	**Overall sklearn**
L. Messi	31	94	4	5	95	94	94.99	94.99
C.Ronaldo	33	94	5	5	96	94	97.27	97.27
Neymar Jr	26	93	5	5	94	92	91.4	91.4
De Gea	27	93	1	4	90	91	88.17	88.17
De Bruyne	27	92	4	4	91	91	89.81	89.81
E. Hazard	27	91	4	4	90	91	88.91	88.91
L. Modric	32	91	4	4	90	91	92.29	92.29
L. Suarez	31	91	3	5	92	91	91.61	91.61
S.Ramos	32	91	3	4	85	91	90.42	90.42
Lewandoski	29	90	4	4	90	90	89.6	89.6

Nhận xét chung: Dữ liệu cho thấy các mô hình dự đoán hoạt động hiệu quả với độ chính xác cao, thể hiện qua việc giá trị dự đoán Overall, Overall cài tay, và Overall sklearn có các giá trị dự đoán giống nhau chứng tỏ mô hình hồi quy tuyến tính cài tay rất đúng và các giá trị dự đoán rất gần với giá trị thực tế do giá trị R^2 khá cao (~ 92) chứng tỏ mô hình cũng được huấn luyện rất tốt.

4.1. Điểm mạnh của mô hình

Đối với những cầu thủ có chỉ số Overall cao và rõ ràng, như **L. Messi**, **Neymar Jr**, và **Lewandowski**, mô hình có khả năng dự đoán khá chính xác. Ví dụ, chỉ số dự đoán cho Messi là 94.99, rất gần với chỉ số thực tế 94. Điều này cho thấy mô hình có thể nắm bắt tốt các mối quan hệ giữa các đặc trưng và chỉ số Overall đối với các cầu thủ có hiệu suất ổn định.

Các dự đoán cho những cầu thủ như **De Bruyne**, **E. Hazard**, **L. Suarez**, và **L. Modric** cũng rất gần với giá trị thực tế. Điều này cho thấy mô hình đang học tốt từ dữ liệu huấn luyện và có thể dự đoán khá chính xác trong nhiều trường hợp.

4.2. Điểm yếu của mô hình

Mặc dù dự đoán của mô hình cho **C. Ronaldo** là 97.27, cao hơn đáng kể so với giá trị thực tế là 94, đây là một ví dụ rõ ràng về việc mô hình có thể bị sai lệch đáng kể. Sự khác biệt lớn này có thể do mô hình không đủ tinh tế để xử lý các yếu tố đặc biệt liên quan đến Ronaldo như danh tiếng quốc tế và kỹ năng đi bóng.

Mô hình dự đoán chỉ số của **De Gea** là 88.17, thấp hơn so với chỉ số thực tế là 91. Điều này có thể do "Skill Moves" của De Gea rất thấp, khiến mô hình không đánh giá cao khả năng của anh.

Điều này cho thấy mô hình có thể cần thêm các đặc trưng hoặc tinh chỉnh hơn nữa để giảm thiểu sự sai lệch trong các trường hợp đặc biệt và chưa hoàn toàn hiểu được tầm quan trọng của các đặc trưng trong những trường hợp đặc biệt như thủ môn.

4.3. Các yếu tố ảnh hưởng đến hiệu quả mô hình

Mô hình có vẻ đánh giá cao các đặc trưng như "Potential"**,** "Skill Moves"**,** và "Reactions" trong việc dự đoán chỉ số Overall. Những cầu thủ như C. Ronaldo và De Gea có thể được ảnh hưởng mạnh mẽ bởi các đặc trưng này, và điều này cho thấy các yếu tố này đóng vai trò quan trọng trong việc xác định chỉ số Overall.

Mô hình hồi quy tuyến tính có thể chưa cân nhắc hết các yếu tố có ảnh hưởng mạnh đến chỉ số Overall của cầu thủ, chẳng hạn như vị trí thi đấu, phong độ thi đấu trong mùa giải, hoặc các yếu tố ngoài sân cỏ (tinh thần, sự chăm chỉ, v.v.). Điều này có thể là một lý do cho sự sai lệch trong dự đoán đối với một số cầu thủ.

4.4. Kết luận chung

Mô hình hồi quy tuyến tính đã thể hiện hiệu quả trong việc dự đoán chỉ số Overall của các cầu thủ dựa trên các đặc trưng như Age**,** Potential**,** Skill Moves**,** International Reputation và Reactions**.** Tuy nhiên, có một số trường hợp mà mô hình chưa hoàn hảo, đặc biệt là với những cầu thủ có các đặc điểm đặc biệt. Những kết quả sai lệch này chỉ ra rằng

mô hình cần được tinh chỉnh thêm hoặc cần bổ sung thêm đặc trưng để cải thiện độ chính xác.

Ngoài ra, có thể thử nghiệm các mô hình phức tạp hơn như **hồi quy tuyến tính đa thức** hoặc các mô hình học máy khác như **random forest** hoặc **gradient boosting** để cải thiện độ chính xác trong dự đoán chỉ số Overall, đặc biệt là đối với các cầu thủ có các đặc điểm hoặc phong độ thi đấu rất đặc biệt.

CHƯƠNG 5. KẾT LUẬN

5.1. Kết quả đạt được

Qua quá trình nghiên cứu và thực nghiệm, đề tài đã đạt được những kết quả đáng ghi nhận:

- Xây dựng thành công mô hình hồi quy tuyến tính thủ công để dự đoán chỉ số Overall của các cầu thủ dựa trên 5 đặc trưng chính: **Age** (Tuổi), **Potential** (Tiềm năng), **Skill Moves** (Kỹ năng đi bóng), **International Reputation** (Danh tiếng quốc tế), và **Reactions** (Phản xạ).
- Mô hình đã cho ra những dự đoán khá chính xác, với độ chênh lệch nhỏ giữa chỉ số thực tế và chỉ số dự đoán, đặc biệt đối với các cầu thủ có đặc trưng đồng đều. Một số trường hợp điển hình như **L. Messi** và **Neymar Jr** có kết quả dự đoán sát với thực tế, thể hiện tính hiệu quả của mô hình.
- Việc áp dụng phương pháp hồi quy tuyến tính giúp hiểu rõ hơn về mức độ ảnh hưởng của từng đặc trưng đến chỉ số **Overall**. Các hệ số hồi quy cũng cung cấp cái nhìn trực quan về cách các yếu tố này tương tác trong việc xác định giá trị tổng thể của cầu thủ.

5.2. Những khó khăn, hạn chế

Dù đạt được nhiều kết quả tích cực, đề tài vẫn gặp phải một số khó khăn và hạn chế:

- **Độ chính xác dự đoán chưa hoàn toàn tối ưu**: Trong một số trường hợp, như đối với **C. Ronaldo**, sự chênh lệch giữa dự đoán và thực tế còn khá lớn. Điều này có thể do mô hình hồi quy tuyến tính chưa mô phỏng đầy đủ các tương tác phức tạp giữa các đặc trưng, đặc biệt đối với những cầu thủ có phong cách thi đấu đa dạng.
- **Giới hạn về số lượng đặc trưng**: Đề tài chỉ tập trung vào 5 đặc trưng chính, trong khi còn nhiều yếu tố khác cũng ảnh hưởng đến chỉ số **Overall** của cầu thủ như **Work Rate** (Tần suất hoạt động), **Positioning** (Vị trí thi đấu), hay

Strength (Sức mạnh). Điều này dẫn đến mô hình chưa phản ánh được toàn diện khả năng của cầu thủ.

- **Dữ liệu không đồng nhất**: Một số cầu thủ có dữ liệu đặc trưng không đồng đều, đặc biệt là về tuổi tác và kinh nghiệm thi đấu quốc tế, gây ảnh hưởng đến độ chính xác của mô hình. Các cầu thủ lớn tuổi có thể bị đánh giá thấp hơn do ảnh hưởng của đặc trưng **Age**, mặc dù họ vẫn giữ được phong độ cao.
- **Mô hình đơn giản**: Hồi quy tuyến tính là phương pháp đơn giản và dễ hiểu, nhưng lại thiếu linh hoạt trong việc mô hình hóa các quan hệ phi tuyến giữa các đặc trưng. Điều này có thể được cải thiện bằng cách sử dụng các mô hình phức tạp hơn như hồi quy phi tuyến hay các mô hình học sâu (deep learning).

Nhìn chung, mặc dù còn tồn tại một số hạn chế, đề tài đã đạt được những kết quả quan trọng, làm nền tảng cho việc phát triển các mô hình dự đoán cầu thủ chính xác hơn trong tương lai.

TÀI LIỆU THAM KHẢO

[1] A. Géron, Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly Media, Inc, 2019.

[2] G. Bonaccorso, Machine Learning Algorithms, Packt Publishing Ltd., 2017.

[3] D. Sarkar, R. Bali and T. Sharma, Practical Machine Learning with Python, Apress Media, LLC, 2018.

[4] V. H. Tiệp, Machine Learning cơ bản, 2018.

PHỤ LỤC

Bảng phân công công việc của các thành viên trong nhóm

STT	Nội dung công việc	22DH114497 Lương Tiến Đạt	22DH110271 Nguyễn Hoàng Bảo	22DH112007 Nguyễn Trần Bảo Long
1	Chuẩn bị và làm sạch dữ liệu	35%	30%	35%
2	Khám phá dữ liệu (EDA)	35%	30%	35%
3	Xây dựng mô hình hồi quy tuyến tính	30%	30%	40%
4	Thiết kế Web	35%	30%	35%
5	Viết báo cáo	35%	35%	30%