Comparing La Masia and Hale End Academy Graduates

By Angelina Cottone, Taarunya Sekaran, and Kieran Sullivan

Contributions

Angelina: Acquired match logs from FBref, calculated per-90-minutes data, wrote report
Taarunya: Acquired transfer value data from TransferMarkt, created interactive data visualization, wrote report
Kieran: Acquired match rating data from Sofascore, calculated average match rating variable, wrote report

Prepared for:

Dr. Peter Kramlinger
STA 141B
March 16, 2025

Abstract

This project utilizes Python to gather data on youth academy graduates from Arsenal's Hale End and FC Barcelona's La Masia, two elite European football academies. The primary objectives are to scrape data from different sources on these players and construct interactive visualizations to compare players. La Masia has a long history of producing world-class talent that rises to the first team, while Arsenal's Hale End has been criticized for their limited utilization of academy graduates in recent years. By examining key performance metrics, market value, and more, we hope to provide coaches, recruiters, and more with valuable insights into each academy's effectiveness in developing talent. To effectively convey the data, interactive visualizations were created to help these users identify trends, compare performance and see changes in market value over time.

Introduction

In football, youth academies act as the foundation for developing the next generation of players. Arsenal's 2024-25 Premier League title hopes have been brought down by an injury crisis in the attack that has prompted manager Mikel Arteta to rely on their young academy graduates Ethan Nwaneri and Myles Lewis-Skelly to help them through the season. Arsenal is often criticized for not involving their youth players in their main team, with only one real superstar emerging in recent years–Bukayo Saka.

On the other hand, La Masia, FC Barcelona's youth academy, has produced worldwide legends such as Lionel Messi, Xavi, and Carles Puyol and is often regarded as the most prestigious in the world. When Barcelona pulls players from the academy, they cut down on costs that come with buying superstars, aligning with their philosophy: "Barcelona doesn't buy stars, they make them."

This project seeks to gather data for the future analysis of these academies' output that could answer questions such as:

1. How do Hale End and La Masia graduates compare in terms of key performance metrics, including goals, assists, passing, and more?
2. How does a player's performance and match scores influence their market value? Does La Masia produce higher value players in comparison to Hale End?
3. Are there identifiable patterns that help predict the success of academy graduates?

To gather data to answer these questions, Python was used to perform web scraping and API calls to collect the data from multiple football analytics platforms, including:

- FBref: Provides detailed match logs for each player's professional career.
- Transfermarkt: For market value fluctuations.

- Sofascore: Provides match ratings.

Interactive visualizations were created using Plotly to allow for dynamic comparisons between players from each academy.

Methods

We sought to collect match statistics, match ratings, and market valuations to assess the development of youth academy graduates, and found FBref, Transfermarkt, and Sofascore to be strong fits. Interactive visualizations were created using Plotly, for users to be able to look at the data gathered for each player in a dynamic and informative format. All code used for collecting, processing, and visualizing the data for this project is available in the following GitHub repository: https://github.com/acottone/comparing_academy_graduates

1. FBref

FBref (Football Reference) provides comprehensive football statistics for hundreds of competitions and thousands of players worldwide. From this source, we aimed to gather detailed match statistics for each player's professional career, while excluding friendlies and national competitions. By excluding these competitions, we can focus on club game performance and provide a more accurate representation of each player's development in these environments.

Initial Data Extraction

In early stages of the project, we focused on gathering data for four players, two from Arsenal and two from FC Barcelona, with additional focus on the first 1000 minutes of their careers to analyze their initial performances as first-team players. Our initial approach involved attempting to use one of FBref's undocumented APIs to retrieve these match logs. However, implementing these proved difficult when trying to gather data with our specific criteria. Therefore, we opted to manually scrape from FBref directly to allow for greater control and flexibility over gathering the data we needed.

Web Scraping Implementation

To scrape and process the necessary data from FBref, a Python script was developed to perform and automate the necessary steps. Since FBref's structure is dynamic and subject to periodic change, a flexible scraping approach was utilized. The code uses 'requests' for HTTP requests and fetching the web pages, 'BeautifulSoup' library to parse and extract the HTML content, as well as 'pandas' for cleaning and organizing the dataset and 'datetime' for per-90-statistics generation.

Extracting player-specific match logs was done by constructing the base URL for each player based on their player ID. The script followed internal FBref links to gather the match logs for multiple seasons, only including matches for competitive club competitions. To ensure consistency in merging the datasets, the script parsed table headers in the page and mapped them to predefined categories. Only matches where a player's playing time was greater than 0 minutes were included.

Challenges

We encountered several challenges during the scraping process for this source, however. The initial code was developed to extract the data from FBref for the four initial players based on their player IDs. Later, as we decided to expand to include more players, and gather match logs for their entire career, our previous code structure no longer worked, likely due to some changes in FBref's HTML structure. This resulted in having to rewrite the code to accommodate the new format and include the additional players. FBref also limits excessive requests by blocking users for a period of time. To address this, a 6-second delay between requests and the use of the 'fake_useragent' library was used to prevent detection and getting blocked by the site. Also, since we did not use an API for this source, we needed to clean and structure the data appropriately during the scraping process.

Per-90 Statistics

For the purposes of visualization and future analysis, per-90-minute statistics were calculated. This allowed for normalized performance metrics for each player. The generation of these per-90 statistics was aligned with dates of market value change retrieved from Transfermarkt. For each player, all matches played between two valuation dates were used to generate the average performance leading up to the market value change. For the first date of valuation, all matches starting from their first match up to the date were used. The formula used for this is as follows $per90 = (\frac{total\ value\ (in\ time\ period)}{total\ minutes\ played\ (in\ time\ period)}) \times 90.$

2. Transfermarkt

Transfermarkt is a comprehensive online football database that provides detailed information about players, teams, and their market values. For our analysis, Transfermarkt was a crucial source for obtaining the market value history of players who graduated from the Arsenal and Barcelona academies and are currently playing for the senior squads. The market value data provides insights into how a player's perceived value has changed over time, which can be influenced by their performance, age, contract status, injuries and other factors.

Data Extraction

The process began by scraping player IDs and positions from the academy squad pages of Arsenal and Barcelona. This was done by sending HTTP requests to the respective URLs of the academy squad pages using the requests library in Python. The HTML content of the pages was parsed using BeautifulSoup, and the relevant player information (name, ID, and position) was extracted by identifying specific HTML elements and attributes. For example, the player ID was extracted from the href attribute of the player profile link, while the position was extracted from a specific table cell. Once the player IDs were obtained, the next step was to fetch the market value history for each player by sending requests to Transfermarkt's API endpoint, which contained the market value development data. This endpoint returned JSON data that included the player's market value over time. The JSON response was parsed to extract the market value data, including the date, value in euros, and the club the player was associated with at that time. The extracted market value data for all players was then stored in a list of dictionaries, where each dictionary represented a single market value entry for a player. This data was converted into a Pandas DataFrame for easier manipulation and analysis. Finally, the DataFrame was saved to a CSV file: Transfermarkt_values.csv

Challenges

To avoid being blocked by Transfermarkt's servers, a delay (time.sleep(2)) was introduced between requests. This slowed down the scraping process but was necessary to ensure the script could run without interruptions. Another significant challenge was encoding issues. Many of the Spanish players playing for Barcelona had accents and umlauts, which caused problems with displaying the names correctly in the CSV file, resulting in unusual formatting causing errors while retrieving information. To address this, UTF-8 encoding and decoding was used to maintain their original names while saving the data to the CSV file.

3. Sofascore

Sofascore is an online football database that provides a variety of collections of data about players, teams, and competitions. Key for our analysis are their match performance ratings, which are given to each player in a match, assuming they played at least ten minutes. The Sofascore ratings range between 3 and 10, with each player starting at 6.5, and are calculated with an algorithm that takes over 100 data categories into consideration, such as successful dribbles, unsuccessful passes, or shots on target. Thus, this provides us with a well-rounded variable to consider players performance in individual matches. We aimed to extract these match ratings for our 15 Hale End

and La Masia graduates currently playing in the first team, and filtered it to just their competitive performances for their club senior team (so any friendly, youth, or international appearances were excluded).

Data Extraction

To gather the match ratings for the 15 players, the first step was to collect their unique Player IDs from Sofascore, which was possible by searching them by name and acquiring their six or seven digit ID from the url. We then needed to find the team IDs for Arsenal and Barcelona, to be able to select only the matches played for those teams. This was possible with the same method of searching the teams and pulling out the IDs from the url (which were 42 and 2817, respectively).

We used a Python script that incorporated a while loop to send GET requests for each page of each player's match ratings, and converted the response into JSON. The script iterated through the pages of ratings by increasing the page parameter in the url being used in the request, and once the response did not contain a valid section containing match ratings data, the loop stopped. To process the JSON response from the Sofascore's semi-structured API, we needed to look for where the important features were. We found the match ratings in the statisticsMap section, and the dates, team IDs, and competition names in the event details portion of the response. After filtering to the types of matches that we wanted, we were able to compile all match ratings for all players in a dataframe.

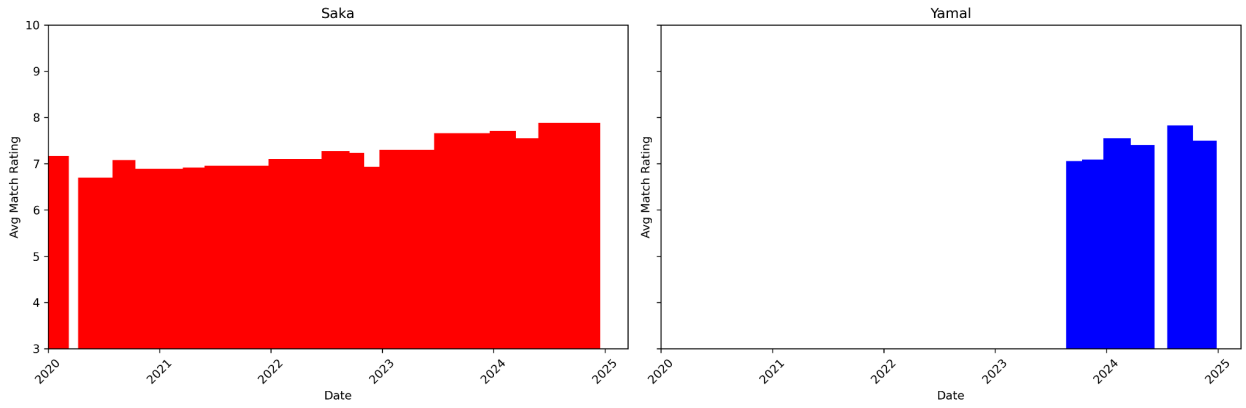
Challenges

The main challenges encountered during this process was acquiring match ratings from the players' entire time at their respective clubs. At first, the script would only return the last twenty or so matches they played. Realizing this was due to only the first page of matches being used, we had to implement the pagination process described above. Excluding matches that we were not interested in was somewhat challenging as well, as it involved finding and using several criteria to select by.

Average Match Ratings for Transfer Value Periods

To add to the visualization, average match ratings between changes in Transfermarkt transfer value were calculated. This involved using both the transfer values data that was collected and the match rating data that was collected to create intervals of dates that match ratings would be averaged from. One challenge with this is that sometimes a player's transfer value would change and during that period the player had played no matches for Arsenal or Barcelona, due to injury, loan, international break, or other reason, leading to NA values. Below is another way to view match

ratings by transfer value period, comparing Bukayo Saka and Lamine Yamal on the metric of average match rating during those intervals.



4. Interactive visualization

The interactive visualization was designed to provide a comprehensive and dynamic view of a player's career progression by integrating data from three key sources: Transfermarkt (market values), per90 statistics (performance metrics), and Sofascore (match ratings). The goal was to create a tool that allows users to explore how a player's market value, on-field performance, and match ratings have evolved over time. The visualization was built using Plotly, a powerful Python library for creating interactive and customizable graphs.

Data Preparation

The first step in creating the visualization was preparing the data. Three datasets were loaded into the script: Transfermarkt_values.csv, which contains the market value history of players; all_players_per90_stats.csv, which includes performance metrics such as goals, assists, and shots per 90 minutes; and player_match_ratings.csv, which provides Sofascore match ratings for each player. Initially, there were issues matching the data from the three CSVs due to differences in how player names were stored. For example, one dataset listed "Cubarsi" as the player's name for match ratings, while "Pau-Cubarsi" appeared in the per 90 stats, and "Pau Cubarsí" in the transfer values. To ensure consistency across the datasets, the script standardized player names by converting them to lowercase, removing accents and special characters, and using FuzzyWuzzy to match names that might have slight variations. This step was crucial because discrepancies in player names across datasets could lead to mismatched or missing data. Additionally, there were issues matching dates as well. The date formats varied, with the match ratings using "11/30/2024 11:30,"

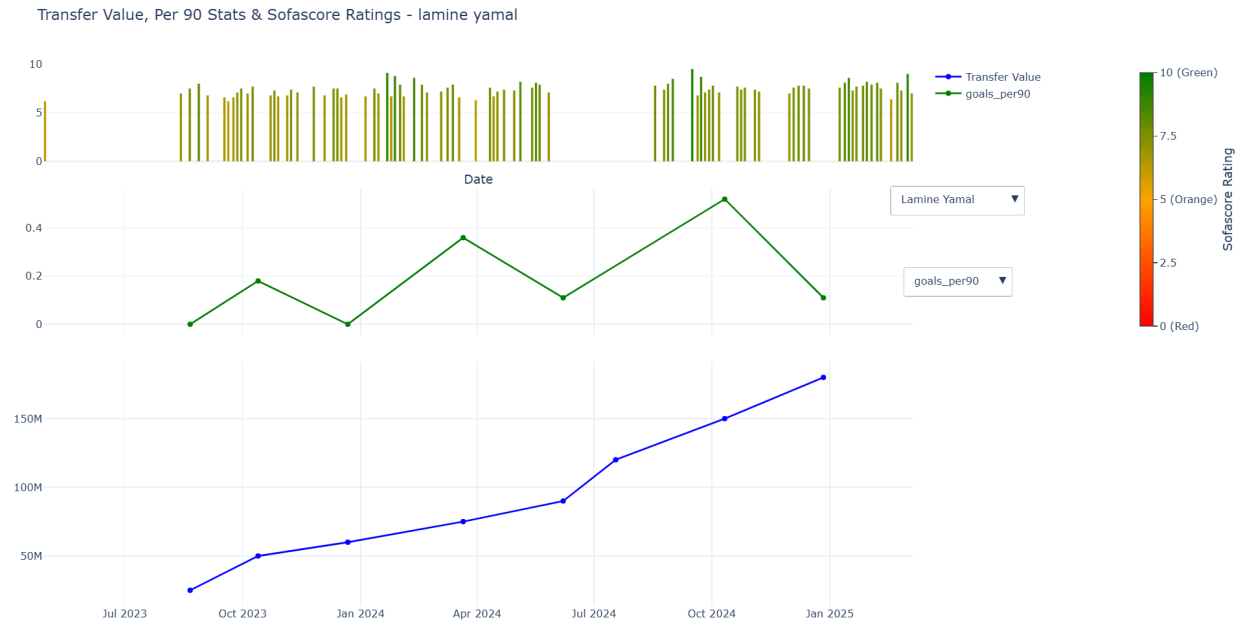
the per 90 stats using "3/21/2024," and the transfer values using "13-Oct-23." All date columns were converted to a standardized datetime format to enable accurate time-based plotting and analysis.

Visualization Layout

The visualization is structured as a multi-subplot figure with three rows, each representing a different aspect of the player's career. The top graph (Row 1) displays Sofascore match ratings as a heatmap-style bar plot. Each bar represents a match, and the color of the bar corresponds to the player's rating in that match. A custom color gradient was applied to the bars, ranging from red (for poor ratings of 0) to orange (for average ratings of 5) to green (for excellent ratings of 10). This gradient, combined with a color bar on the right, makes it easy to interpret the player's performance in individual matches at a glance. The middle graph (Row 2) shows per90 performance metrics, such as goals, assists, and shots per 90 minutes, as a line graph. This graph is interactive, allowing users to toggle between different metrics using a dropdown menu. The bottom graph (Row 3) displays the player's market value over time as a line graph, providing insights into how their perceived value has changed throughout their career. The subplots are arranged vertically.

Interactive Features

One of the key strengths of this visualization is its interactivity, which allows users to explore the data in depth. Two dropdown menus were added to the layout to enhance usability. The first dropdown allows users to switch between players. When a player is selected, all three graphs update dynamically to display data for that player. The dropdown is populated with the 15 player names from the Transfermarkt dataset, ensuring that users can explore data for any player included in the analysis. The second dropdown allows users to toggle between 5 different per90 metrics in the middle graph. This feature provides flexibility and enables users to focus on the metrics that are most relevant to their analysis. The interactivity is powered by Plotly's update and restyle methods. When a player is selected, the update method is used to refresh all three graphs with the corresponding data. When a per90 metric is toggled, the restyle method is used to update only the middle graph, ensuring a smooth and efficient user experience. Additionally, the visualization includes a dynamic title that updates based on the selected player and per90 metric, providing context and making it easy to keep track of what is being displayed.



Challenges

One of the earliest and most significant challenges was that the dropdown menus, designed to toggle between players and per90 metrics, were affecting all three graphs instead of just the intended one. For ex: when switching between per90 metrics, the Sofascore match ratings graph would unexpectedly transform into a bar graph, and the y-axis would shift dramatically (e.g., from 0–10 to 0–100). Similarly, the transfer value graph would sometimes display incorrect data or axes. This issue stemmed from the way the `args` in the dropdown buttons were structured. The `args` were being applied to all traces in the figure, causing unintended changes to the Sofascore and transfer value graphs. To address this, the `restyle` method was used for the per90 dropdown, ensuring that only the middle graph (per90 metrics) was updated. This involved isolating the `y` data for the per90 metrics and applying changes exclusively to the second trace (middle graph). This fix successfully resolved the issue of the dropdowns affecting the other graphs.

Another issue that emerged was that the labels on the per90 stats graph did not update correctly. For example, when toggling to "shots_per90," the graph would display the correct data but the label would still say "goals_per90." This was fixed by explicitly updating the trace name and axis labels when toggling between per90 metrics. The `restyle` method was used to update the `y` data, while the `update` method was used to modify the graph title and labels dynamically.

One of the most persistent issues was that the graphs would revert to Lamine Yamal's data when switching between players or toggling metrics. This happened because the dropdown buttons were created before the player selection was made, causing them to always pull data from the initial player (Lamine Yamal). This issue was resolved by restructuring the code to ensure that the per90

dropdown buttons dynamically updated based on the selected player. The `get_player_data` function was called within the dropdown update logic to refresh the data for the selected player. This ensured that the correct stats were displayed for each player.

However, this fix only resolved the problem with the match ratings and the transfer values, not with the per90 stats. While the "goals_per90" metric worked and updated perfectly for all players, the other four metrics would still show Yamal's data. The number of data points corresponded to the selected player, but the data itself was Yamal's. Several attempts were made to correct this, including manually creating separate traces in a dictionary and compromising on sophistication, but none of these solutions worked as intended. The root of the problem likely lies in how the `update` method is being used in the dropdown buttons. The `update` method is not correctly updating the `per90` data for the selected player. Instead, it seems to be using the initial `per90_data` for all players. Despite multiple attempts to fix this issue, a solution could not be found within the integrated visualization.

Alternative

To address the persistent issues with the per90 stats in the main interactive visualization, a temporary alternative graph was created. This standalone visualization focuses exclusively on displaying per90 metrics (e.g., goals, assists, shots, sca, passes) for each player in a multi-subplot layout, allowing users to view all five metrics simultaneously. While it lacks the integration of the main visualization, it provides a reliable and accurate way to analyze player performance independently, ensuring that the per90 data is presented clearly and without interference from other graphs. This solution serves as a practical workaround until the per90 functionality can be fully integrated into the main interface.

Results

While we did not complete any high-level analysis, as the focus of our project was on data acquisition that could later be used for analysis, we were able to generate some preliminary takeaways. First, in recent years La Masia have generated far more high-value players for the first team than Hale End. Barcelona currently has six academy graduates playing in the first team with market values of €50 million or higher, while Arsenal only have one in Bukayo Saka. Additionally, Barcelona have the highest valued player of both academies combined, with Lamine Yamal valued at €180 million. It was also observed that players who took breaks during their careers, whether due to injuries or loans, experienced a drop in value. This was seen with Ansu Fati, who was considered a top prospect worth €80 million before his injuries, and after Eric Garcia's temporary move to Manchester City. Additionally, the graphs show that the performance stats of players in forward and attacking midfielder roles had a more positive impact on their transfer values.

The data collected and the interactive visualizations created provide a foundation for further high-level analysis of player performance and market valuations in the future.

Conclusion

This project aimed to collect data on youth academy graduates from Arsenal's Hale End and FC Barcelona's La Masia to assess their development and performance as professional players. Through the use of web scraping and APIs, we were able to successfully retrieve detailed match logs from FBref and generate per-90 statistics, aligned with market value fluctuations from Transfermarkt. We also collected individual match ratings through Sofascore for a well-rounded analysis of an individual match performance. This approach provides a way to quantify and visualize multiple aspects of a player's development throughout their career, offering insights that could help recruiters, coaches, and analysts in evaluating the effectiveness of each academy in producing world-class players.

Limitations

Despite the insights this project provides, there are several limitations to consider. The dataset currently focuses on only 3 Hale End graduates compared to 12 La Masia graduates, so there are some imbalances when comparing the two academies, and some players have significantly less playing time in comparison to others. A more even sample size could provide stronger conclusions. The data gathered also doesn't take extraneous circumstances such as injuries into account, and also doesn't include data on B teams.

Future analysis could also examine players who have graduated from Hale End or La Masia and gone on to have success at other clubs, or take into account the profit made from selling those players as part of its analysis of the success of the academy. There could also be work to be done to put these results in the context of the English and Spanish leagues to understand how well Arsenal and Barcelona are doing at producing talent in comparison to their domestic competition.

This project would also benefit from gathering data from more sources, and including sentiment analysis to see how media perception may affect variables like market valuation. Furthermore, future work could focus on integrating the alternative graph functionality into the main visualization while addressing the technical challenges encountered. It would be valuable to use all per90 stats instead of just the top 5, sorting them by relevance to the players' positions. For instance, metrics like goals per 90 are less relevant for a left-back like Lewis Skelly than stats like tackles per 90. Similarly, for a goalkeeper like Inaki Pena, most outfield stats would not be applicable.

Appendix

The code used in the project for web scraping and processing the sources, and visualization, can be found on GitHub at the following link:
https://github.com/acottone/comparing_academy_graduates