COM-480 Data Visualization
May 31, 2024

FIFA Player Visualization

Process Book

Authors:
Han Miao
Ilias Merigh
Zhuofu Zhou

Introduction

Inspiration for the Idea

The idea of our project arises from the fact that all of us are football fans and FIFA online players. We decided to work on the player data in the FIFA-Online game. When we decided what story we want to tell, we find that though there are many visualizations on the players themselves, there is not much about visualization on the transfer market. Football enthusiasts around the global can only know the ongoing activities about the transfer market through plain texts and uninformative numbers.

Overview

In this milestone, we constructed a visualization website aimed at displaying the dynamics of the football transfer markets. We split the visualization into categories: player, team, league, and region, providing a comprehensive picture of the thrilling and lively transfer market from 2015 to 2023.

Player: Career Trajectories

The career trajectories timeline is used to show the career of a specific player. Our aim is to identify the path to success of the most promising players in the world. This timeline provides the audience with a detailed explanation of the player's choices in the transfer market.

Team: Team Categories

The team categories are used to highlight three types of football clubs: Clubs that develop young, talented players and sell them later. Clubs that directly purchase developed players, typically having more financial resources to afford such acquisitions. Clubs that buy older players at the end of their careers, not primarily for their performance but to enhance the club's image and increase their viewing audience.

League: Transfer Sankey Diagram

The Sankey diagram is used to show the changes in leagues during different stages of players’ careers. We identify the league a player is in during the early, middle, and late stages of his career. This provides the audience with an overview of transfer dynamics through league-based visualization.

Region: Transfer Flight Route Map

The flight route map is used to show the most popular transfer destinations of players exiting each country, represented with arrows. When the user hovers their mouse over the interactive map, it provides an overview of transfer dynamics through region-based visualization.

Visualization Toolkit

Python

Python is used for exploratory data analysis, data preprocessing, data structure conversion, and collaboration with other visualization tools.

HTML/CSS/Javascript

HTML/CSS/Javascript is used for website construction.

D3.js

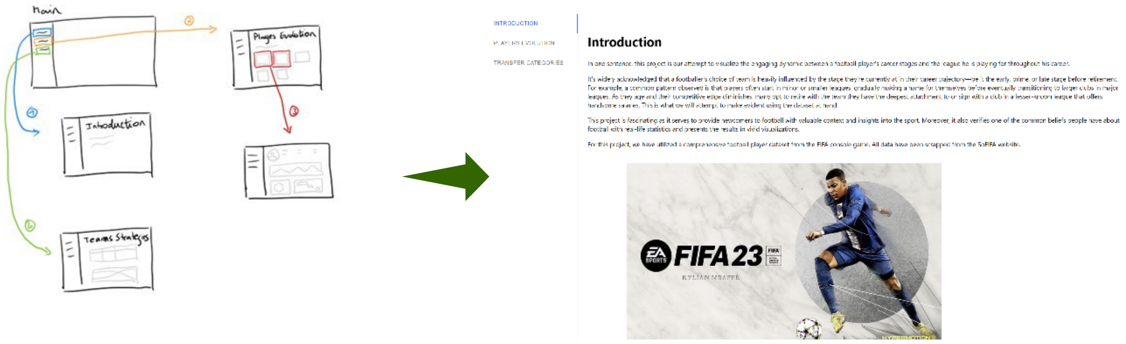
D3.js is one of the main visualization tools we used for our project. It provides a clear and powerful way to create and manipulate complex data visualizations, allowing us to dynamically render interactive charts, graphs, and maps with precision and ease.

Plotly.js

Plotly.js is the other visualization tool we used for our project. It offers a user-friendly interface for creating highly interactive and visually appealing plots and charts. Additionally, Plotly.js integrates seamlessly with Python, allowing for robust data analysis and visualization through collaboration with Python libraries such as Pandas and NumPy.

React

React provides us with a powerful tool for building dynamic and responsive user interfaces. It enables us to create reusable components, manage application state efficiently, and ensure a seamless user experience through its virtual DOM and component-based architecture.



Introduction

In one sentence, this project is our attempt to visualize the engaging dynamic between a football player's career stages and the league he is playing for throughout his career.

It's widely acknowledged that a footballer's choice of team is heavily influenced by the stage they're currently at in their career trajectory—be it the early, prime, or late stage before retirement. For example, a common pattern observed is that players often start in minor or smaller leagues, gradually making a name for themselves before eventually transitioning to larger clubs in major leagues. As they age and their competitive edge diminishes, many opt to retire with the team they have the deepest attachment to or sign with a club in a lesser-known league that offers handsome salaries. This is what we will attempt to make evident using the dataset at hand.

This project is fascinating as it serves to provide newcomers to football with valuable context and insights into the sport. Moreover, it also verifies one of the common beliefs people have about football with real-life statistics and presents the results in vivid visualizations.

For this project, we have utilized a comprehensive football player dataset from the FIFA console game. All data have been scrapped from the SoFIFA website.

Website structure(left) and website overview (right)

Career Trajectories

Data Prepossessing

We first develop an algorithm to define the most improved players in the current professional football players recorded by the FIFA.

Algorithm: Decile Classification

We use decile-classification algorithms to rank players based on their overall scores, which reflect a player's real-life level. We assume that this level can be used to rank players into different categories: bad players, below-average players, above-average players, good players, and world-class players.

We then identified the players who improved the most in terms of levels throughout their careers and recognized them as the most improved players. Using these algorithms, we have obtained a list of players that aligns well with the current football transfer market. For example, we have selected Frenkie de Jong, one of the best midfielders in the world, as well as Erling Haaland and Kylian Mbappe, who are considered potential GOATs (Greatest of All Time) following Cristiano Ronaldo and Lionel Messi. In total, there are 31 players on the list, all of whom are considered the world's top-tier young players in various positions across the game globally.

Implement Visualization with Plotly.js

Visualize the career trajectories of each player

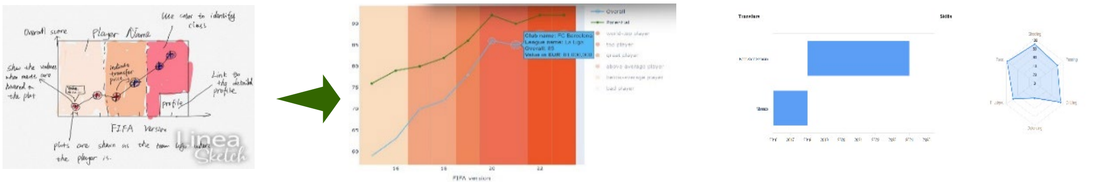
Based on the trajectory visualization completed during Milestone 1, in Milestone 3, we enhanced the visualization to be interactive and more informative. For this milestone, we used Plotly.js for implementation.

First, we display the animation of a player's career trajectory as a time series. With this animation, we can provide the audience with an immersive experience of the player's career. Second, we added another plot to show the potential score, which displays the player’s sustainability in his future career. We also included basic information in each score point so that when the audience hovers their mouse over the plot, it displays the team and the value of the player.

Visualize the player card

When a career trajectory is clicked, a player card will be displayed on the website, providing more details on the player's trajectory. We implemented this using React, offering the audience detailed information and the evolution of the player's skills. With the player's overall score as the main visualization, we display the player's team over time, aligning with the main topic of our project. We also provide a hexagon visualization of the player's skills in different areas, such as dribbling, pace, and strength, presented as a time series.

Overall, we have created a comprehensive information database on the most improved players in today's football transfer market. The purpose of this is to give the audience a better understanding of the career trajectory of a successful professional football player through real examples.



Original design(left), Plotly.js-implemented career trajectories(middle), Skill Hexagon and Transfer timeline (right)

Team Categories

The objective is to highlight three categories of football clubs: those that tend to develop young, talented players and sell them later; those that directly purchase developed players because these clubs usually have more financial resources and can afford such acquisitions; and finally, those that buy older players at the end of their careers, not for their performance but to enhance the club's image and increase their viewing audience. Therefore, the goal is to establish a ranking of clubs for each category, resulting in three separate rankings.

Algorithms

Naive Approach

Initially, we focused only on the age and skill level of players, neglecting transfers and changes in skill over time. This approach failed to consider club transfers and the variation in player skills across different years.
Initially, our approach focused solely on player age and skill level, overlooking transfers between clubs and changes in skill over time. Our main concern was that this method failed to consider how player abilities evolve and the impact of club transfers.
This meant that an excellent player could fall into the category of clubs that attract peak players, even though the player was developed by the club and not acquired after his development. These limitations led us to develop a second, more advanced approach.

Advanced Approach

In this second approach, we revisited the algorithm from the "Most Improved Player" section to incorporate a ranking system for players. We excluded all players whose performance levels remained exceptionally high throughout their careers from 2015 to 2023, as we considered them outliers and not typical of the usual club recruitment process (for example, Lionel Messi has been at the top level from 2015 to 2023, which did not interest us). We also filtered out players who stayed with the same club throughout the period since our project focuses on transfers and we wanted to avoid the pitfalls of the first approach.

Discussion of Results

Attracts Rising Stars: The results from the advanced approach appear much more interesting than those from the naive method: Ajax, PSV, and Olympic Lyonnais, Sporting CP, RB Leipzig, FC Barcelona are excellent candidates. Moreover, it's important to note that our data has limitations as it only includes players of a certain age from the FIFA game, whereas training centers primarily focus on teenage players who are not represented in our dataset.
Attracts Peak Players: The advanced algorithm demonstrated much more relevant results than the naive method, with all outcomes being pertinent: Liverpool, Napoli, Chelsea, Manchester United, Tottenham Hotspur etc… As we can observe, all these clubs compete in the Champions League and strive to have the best talent performing at their peak.
Attracts Falling Stars: Neither approach showed interesting results. Our hypothesis is that the trend among some wealthy, lower-tier clubs to buy very expensive players at the end of their careers solely to enhance their image is a new phenomenon. For example, we know that Miami FC recruited Lionel Messi, Luis Suarez, Jordi Alba, and Sergio Busquets in 2023-2024. Other clubs in Saudi Arabia also signed players like Karim Benzema and Neymar Jr. this year. Therefore, it is normal that these new transfer mechanisms do not appear in our dataset, which stops in 2023.

Relevant Players for Each Team

When a user hovers over a club, the top three players who have most significantly contributed to the club's ranking in our algorithm are displayed. This includes the specific year in question, their age at that time, and their overall level. In other words, the players shown are the three most relevant individuals who justify the club's inclusion in one of our categories. The data on the players, including the year, level, and age, is very consistent. Additionally, as football fans, we confirm that the inclusion of certain players such as De Jong, Neymar, and E. Hazard aligns with popular opinion.

Transfer Sankey Diagram

Data Processing

Identify a cohort of players

We aim to study players who were in their early career stage in FIFA 15, excluding those with exceptionally long careers, to observe their league transitions through early, middle, and late stages over the limited 8-year dataset.
In this case, we only consider the players who have been present for all of the years from 15~23 and who were younger than 25 years old in 2015.

Define the early, middle and late stages, and the serving league

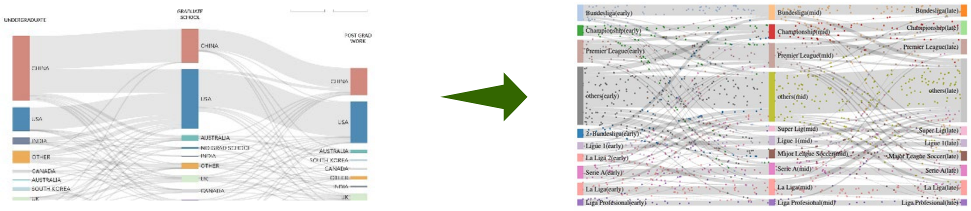
We set the league a player is in for 2015 as the early-stage league, and for 2023 as the late-stage league. The middle stage league is determined by finding the mode of the leagues the player played in from 2015 to 2023. After defining these stages, we store the transfer routes of all players with the stages sorted by leagues. The top 10 most frequently occurring leagues in each column are identified, and all other entries are replaced with the string “others”. This method simplifies the analysis by focusing on predominant leagues while accommodating the inherent variability in players' career paths and league affiliations.

Implementation in D3.js

The implementation of the Sankey diagram itself is fairly standard, involving steps like creating an SVG container, setting up the Sankey diagram using the Sankey generator from D3.js, appending the node rectangles, etc.
One particular challenge is worth noting:

Load and Bind the Data from Python to JSON Suitable for Sankey diagram

To do this, we implement a function using python.
The function converts a NumPy array (the sorted array containing the serving leagues of the players, discussed in the last section) into a JSON format suitable for creating a Sankey diagram. The function starts by converting the dataset into a list of lists using the tolist method. It then identifies unique transfer routes (e.g. Premier League to Bundesliga to English Championship, etc.) along with their counts. These unique transfer routes are transformed into a list of node objects, each containing a "name" property. The function iterates through each row of the NumPy array, creating links between consecutive node objects by identifying their indices in the array. If a link between two nodes already exists, its "value" is incremented; otherwise, a new link is created. The resulting structure, containing "nodes" and "links" arrays, is then converted to a JSON string with formatted line breaks for better readability. The contents of this JSON string were then copied to the D3JS script to create the Sankey Diagram.



Template Sankey diagram (left) and D3.js-implemented Sankey diagram(right)

Transfer Flight Route Map

Data Processing

The flight route map is used to show the most popular transfer destinations of players exiting each country with arrows, when the user hovers his/her mouse over it on the interactive map. There was one challenge that was time consuming to resolve:

Contain the country of the team a player is playing at a certain time

We create a look-up table, by getting all the leagues that are present in the dataset, and then manually labeling what countries they are from. Then, we proceed to identify the top 5 transfer destinations for each country. For each unique player, it sorts their records by FIFA version to maintain chronological order. As it iterates through each player's records, it tracks transfers between different countries, recording each route when a player moves from one country to another. These routes are then stored in the country_transfer_routes dictionary, where each key is a country and the value is a list of transfer routes originating from that country. Next, the code initializes another dictionary, top_destination_countries, to store the top 5 destination countries for each country. It counts the occurrences of each destination country from the transfer routes, sorts them in descending order based on the count, and selects the top 5. Finally, it prints the top 5 destination countries for each country, providing a clear overview of the most common transfer destinations in the dataset.

Implementation in D3.js

The implementation of the flight route map, involves the standard steps for creating an interactive map like defining the SVG and the projection used to draw the map, loading the GeoJSON data, drawing the countries as paths on the canvas, etc.
There are two main difficulties worth noting:

Map Country Names to GeoJSON Features

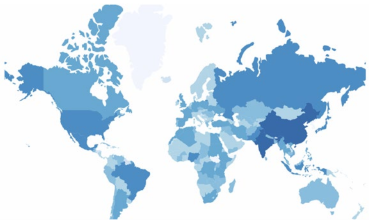
The top 5 destination countries are provided as names, which need to be matched with the corresponding GeoJSON features to accurately draw transfer routes.
A lookup mechanism is implemented where destination country names are mapped to their respective GeoJSON features. This involves iterating through the GeoJSON features and comparing the names to find the correct feature objects. By filtering out undefined matches, we ensure that only valid destination countries are processed.

Highlight Countries and Managing State

Upon interaction, the visualization requires dynamically changing the styles of countries (source and destination) and ensuring these changes are reverted when the interaction ends.
Two functions, mouseOver and mouseLeave, manage the state changes:

mouseOver: Changes the opacity and stroke of the source country to highlight it, logs the country name, maps destination names to GeoJSON features, draws arrows to these destinations, and changes their fill color to red.

mouseLeave: Restores the original style for all countries and removes the arrows. This ensures that the map returns to its initial state, allowing for repeated and smooth interactions.



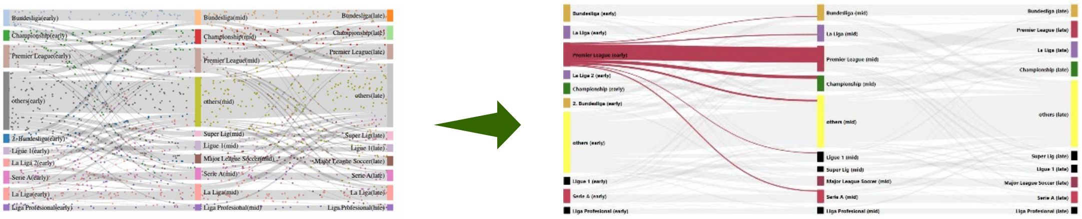
The D3.js-implemented transfer map

React

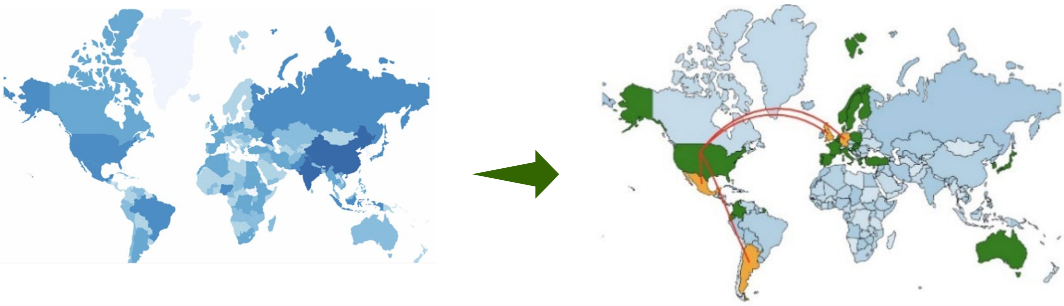
The website was developed using React and JavaScript. We incorporated interactive charts from various libraries such as ApexCharts and D3.js.

Challenges Encountered

Initially, we used the Map and Sankey Diagram from the D3.js library, as mentioned earlier, to generate an individual HTML file for each, which we then embedded into the website. However, since code quality is a critical evaluation criterion, we determined that embedding HTML files did not constitute clean coding practices. Consequently, we converted both the map and the Sankey diagram into widgets that could be directly integrated into React, for example using the library [React Google Charts]. We retained the same logic as initially explained but switched libraries to achieve cleaner code. The original HTML files for the map and Sankey diagram can still be found in the GitHub repository under the "d3_diagrams" folder.



D3.js-impleneted Sankey diagram (left) and React-implemented Sankey diagram (right)



D3.js-impleneted transfer map (left) and React-implemented transfer map (right)

Peer Review

Han Miao

Created the D3.js scripts for the visualization of flight-route map and the Sankey diagram, and its data processing notebook. Authored the sections on Sankey Diagram and Flight Route Map in this process book. Produced the screencast.

Ilias Merigh

Developed the React website, set up the server configuration, and integrated everyone's contributions into the website. Created a data processing notebook for team categories and implemented its visualization on the website. Transformed the Map and Sankey diagrams into React widgets, based on Han's work with D3.js. Authored the sections on Team Categories and React in this process book. Compiled the README and organized the repository for clarity and accessibility.

Zhuofu Zhou

Created the Plotly.js script for the visualization of the career trajectories and the processing notebook. Created the React project based on the career trajectories, skill hexagon and transfer timeline in the Most Improved Player of the website. Authored the sections on Career Trajectories in this process book. Designed and produced the process book.