



Universidad Torcuato Di Tella

Master in Management+Analytics

Thesis

**Development of a Football Analytics Web Application for
Player Scouting**

Student: **Fabricio Pretto**

Advisor: **Guido de Caso**

March, 2022

Desarrollo de una aplicación web de analítica de fútbol para el reclutamiento de jugadores.

Tesis de Maestría en Gestión y Análisis de Datos

Fabrizio Pretto

Resumen Ejecutivo

La modernización y globalización del fútbol y de la economía de los clubes ha provocado un crecimiento significativo en la importancia del reclutamiento de jugadores. La competencia para buscar jóvenes talentos es extremadamente intensa, y los clubes líderes con presupuestos más abultados llevan la delantera. En la presente tesis se desarrolla una plataforma inteligente para optimizar el proceso de reclutamiento de jugadores, haciendo uso de nuevos tipos de datos como estadísticas de cada jugada, incluyendo información sobre cada tiro o pase realizado en un partido.

La tesis cubre el diseño, desarrollo e implementación de todas las etapas del proceso, desde la extracción de los datos y el almacenamiento en un datawarehouse, hasta el análisis estadístico y la representación gráfica en una aplicación web de fácil uso. A través de la exploración descriptiva y el análisis factorial se resume el rendimiento de los jugadores en un solo índice, facilitando a los reclutadores la comparación entre ellos, sin importar su equipo, liga y rol en el campo de juego. El índice de performance es validado utilizando las calificaciones de los partidos como benchmark y a través de un análisis de sensibilidad.

Se presenta un prototipo de la aplicación web y se exponen diferentes casos de uso para mostrar las fortalezas de la plataforma y cómo se puede utilizar para tomar mejores decisiones en el proceso de reclutamiento de jugadores, optimizando el tiempo y los costos asociados.

Development of a Football Analytics Web Application for Player Scouting

Master in Management & Analytics Thesis

Fabricio Pretto

Abstract

The modernization and globalization of football and the football club economy has brought about a growth in stature and importance of scouting. Competition to search for young talents is extremely keen, and top clubs with higher budgets have the lead. In this thesis we develop a smart platform for optimizing the player scouting process, by leveraging new types of data such as play-by-play stats including information on each shot or pass made in a match.

The thesis covers the design, development, and implementation of all stages of the pipeline, from the extraction of the data and storage in a datawarehouse, to the statistical analysis and graphical representation in a user-friendly web application. Through descriptive exploration and factor analysis the performance of football players is summarized in one single index, making it easier for scouts to compare between them, regardless of their team, league, and role in the football pitch. The performance index is validated using the players' match ratings as a benchmark, and through sensitivity analysis.

A prototype of the web application is presented and different use cases are exposed to show its strengths and how it can be used to make better decisions in the scouting process, optimizing associated time and costs.

Contents

1 Introduction	1
1.1 Background	2
1.2 Problem	3
1.3 Objective	5
2 Data	6
3 Methods & Procedures	8
3.1 Methodology	8
3.2 Data Modelling	9
3.2.1 Data Warehouse	9
3.2.2 ELT vs ETL vs EtLT	12
3.3 Analytical Base Table	15
3.4 Statistical Analysis	17
3.5 Web Application	28
4 Results	29
4.1 Player Performance Index	29
4.1.1 Forwards	29
4.1.2 Midfielders	33
4.1.3 Defenders	37
4.1.4 Goalkeepers	40
4.1.5 General results	44
4.2 Football Analytics Web Application	45
4.2.1 App Scouting	46
4.2.2 App Player	47
5 Conclusions	51
5.1 Achievements	51
5.2 Limitations and potential improvements	51
Appendix A	53
Appendix B	60

Appendix C

61

Bibliography

63

Chapter 1

Introduction

With 3.5 billion fans around the world and a total of 41.7 billion dollars of revenue generated by year, football is considered to be the most popular sport on the planet [2]. In recent years, new types of data have been collected for many games in various countries, such as play-by-play data including information on each shot or pass made in a match.

The collection of this data has placed Data Science on the forefront of the football industry with many possible uses and applications:

- Match strategy, tactics, and analysis
- Identifying players' playing styles
- Player acquisition, player valuation, and team spending
- Training regimens and focus
- Injury prediction and prevention using test results and workloads
- Performance management and prediction
- Match outcome and league table prediction
- Tournament design and scheduling
- Betting odds calculation

Many football clubs have been focusing their attention on creating data analysis departments and hiring experts to help them manage millions of stats about players' performance and the upcoming opposition to help the club's chances of winning [3]. Moreover, the use of data science as a scouting tool has proven to be invaluable to clubs, allowing them to tailor the search to the specific attributes needed to fit the team's playing style, narrowing down the area of interest and saving time and money [4]. In the academic landscape, several studies have been conducted in football match prediction, either using machine learning ([5], [6], [7]) or statistical techniques ([8], [9], [10]). In addition to forecasting, researchers have also analyzed optimal frameworks for sport result prediction [11], performance analysis [12], and the statistical identification of the principal factors for match analysis and match outcome ([13], [14]).

1.1 Background

Football scouts are the link between new football players and well-known clubs. Their basic task is to find talented players who can strengthen the club and contribute to its prosperity. A football scout attends football matches on behalf of clubs to collect intelligence. Primarily, there are two types of scouts: player scouts and tactical scouts.

Player scouts evaluate the talent of footballers with a view to signing them on a professional contract for their employers. Some scouts focus on discovering promising young players and future stars, others are employed to run the rule on potential signings. While smaller clubs might only scout within their own country or region, larger, richer clubs can have extensive international scouting networks.

Tactical scouts assess the matches of upcoming opponents of the club and prepare dossiers for their teams' tactical preparations. Instead of identifying talent in these matches, the scout assesses the team and each individual player to identify the relative tactical threats and weaknesses in the opposition. Tactical scouts are typically full-time employees of clubs as their knowledge and findings are considered precious to clubs.

A player scout typically attends as many football matches as possible to evaluate targets first hand. Scouts who wish to identify promising young players typically attend lower-league club games, where their talent can be compared to older peers, or under-16, -18 and -21 international tournaments. Scouts may also receive tips from agents, peers and club colleagues.

On the first evaluation, player scouts determine whether a player has the desired technical attributes to succeed at the sport. They then highlight this player to the club management. Some of the desired attributes that scouts look in players include:

- **Goalkeepers:** good reflexes, communication with defense, one-on-one ability, command of the penalty area and aerial intelligence.
- **Center-backs:** good heading and tackling ability, height, bravery in attempting challenges, concentration.
- **Full-backs:** pace, stamina, anticipation, tackling and marking abilities, work rate and team responsibility.
- **Central midfielders:** stamina, passing ability, team responsibility, positioning, marking abilities.
- **Wingers:** pace, technical ability like dribbling and close control, off-the-ball intelligence, creativity.

- **Forwards:** finishing ability, composure, technical ability, heading ability, pace, off-the-ball intelligence.

Once a player has been recommended to a club, the club may continue to monitor his progress over a period from as little as a few months to as many as a few seasons. Scouts continue to evaluate whether a player has turned in consistent performances, if he has retained his appetite for team responsibilities, and so on.

The importance of scouting offers football clubs with several distinct advantages:

- **Global reach:** Scouting allows clubs to cast the largest possible net to find players from all around the world.
- **Cheap players:** Players from lower leagues can be available at cheaper transfer prices, and command smaller wages. In particular, a talented cheap player can help a football club to progress in a league, knockout cup competition, or into continental cup competition, potentially even ahead of other clubs with superior financial clout.
- **Specialist tactical advice:** Scouting opposition matches allows clubs to build up a knowledge base about opponents that club coaches would otherwise not have the time and resources to research on their own.

1.2 Problem

With the modernization and globalization of football and the football club economy, scouting has grown in stature and importance. Competition to search for young talents is extremely keen. Although it is difficult to quantify the prevalence of scouting in modern football, circumstantial evidence of its magnitude is readily available. The former Chelsea FC Chief Scout Gwyn Williams is reported to have used a database containing up to 77,000 players while working there. It has also been reported that home games of the French second division club Tours FC are attended by an average of 15 to 20 scouts per game.

The use of data and analytics in the scouting process of European leagues has been a growing trend in the last few years, and for good reasons. First and foremost, searching for players in large and detailed databases allows clubs to save incredible amounts of time and money. After applying the desired filters, the scout can restrict the area of interest to a selected number of players and start seeing videos of only this group, and further reducing the need for trips and live interviews. Of course databases cannot replace scouts, but rather they can complement their talent identification skills.



Figure 1.1: Soccerment's hypothetical scouting funnel [4]

In addition, while a computer is not biased towards a favorite team or player, it can remember everything that has occurred in past seasons. This is invaluable for scouts, who are often misled by the over- or under-performance of a player in a specific game, or even over a more extended period.

Furthermore, a smart approach to scouting can allow teams with lower budgets to maximize returns in the transfer market. Player transfers give football clubs the opportunity to reinvest the money in either undervalued talents or players who better suit the team's playing style.

Someone who understands this concept very well is Matthew Benham, owner of Brentford FC in England and FC Midtjylland in Denmark. Table 1.1 represents Brentford's most profitable player transfers, taken from the book *The Expected Goals Philosophy*, by James Tippett [15]:

Player	Purchasing fee (£M)	Selling fee (£M)	Profit (£M)
N. Maupay	1.8	20	18.2
A. Gray	0.5	12	11.5
S. Hogan	0.75	12	11.25
C. Mephan	0	11	11
E. Konsa	2.5	12	9.5
R. Woods	1	6.5	5.5
N. Yennaris	0.2	5	4.8
Jota	1.5	6	4.5
J. Tarkowski	0.3	4.5	4.2
J. Egan	0.4	4	3.6
D. Bentley	0.45	4	3.55
R. Sawyer	0.3	2.9	2.6
M. Odubajo	1	3.5	2.5
M. Colin	0.9	3	2.1
F. Jozefzoon	0.9	2.8	1.9
Total	12.5	109.2	96.7

Table 1.1: Brentford FC most profitable player transfers.

As shown, Brentford FC paid £12.5 million for that list of players and received £109.2 million when they were transferred out, for a gross capital gain of £96.7 million.

According to the book, their success in the transfer market allowed Brentford FC to successfully compete in the English Championship, despite having a salary budget of less than £15m, 60% lower than the league average (£39m).

There are several tools available in the market for smart scouting, provided by companies such as Wyscout, InStat and Soccerment. However, all these platforms are focused mainly in European leagues.

1.3 Objective

The aim of this project is to develop a data-driven web application for football analytics in Latin America that could appeal to football clubs as a performance analysis and scouting tool. Different techniques of Data Analytics will be needed to provide relevant statistics, and a good graphical representation is sought to bring a more user-friendly way to draw conclusions from the data. Some of the questions needed to answer during the course of this thesis are:

- How can sports game data through data models and data analytics be treated to make valuable sports statistics and show them in a proper way?
- Which are the most important statistics that clubs would find valuable?
- How could the performance of players be summarized in one single index, making it easier to compare between them, regardless of their role in the pitch, current team, etc.?

Research has been done on the top football analytics platforms (StatsPerform [16], StatsBomb [17]) in order to identify the variables and stats most commonly used to describe a player's performance in each role of the pitch. These stats help describe a player's contribution to different phases of the game (e.g. *shots*, *goals* and *assists* for forwards; *passing accuracy*, *through balls* and *dribbling* for midfielders; *tackles*, *blocks* and *recoveries* for defenders; and *saves*, *duels won* and *goals conceded* for goalkeepers), shooting tendencies, spatial tendencies and possession involvement.

Furthermore, an attempt will be made to summarize the performance of the players in one single index, taking into consideration the relevant stats for each role. Platforms like Soccerment [18] and InStat [19] have already developed an index for measuring players' performance, but logically the procedure and calculations made to create them are not publicly available since their algorithms are proprietary.

Therefore, following the guidelines of the Handbook on Constructing Composite Indicators, by the Organization for Economic Co-operation and Development (OECD) [20], a custom index will be developed with the available data for players in Latin America. This index is sought to speed up the scouting process, being the primary selection filter used by many professionals, as stated by InStat about their tool in its website.

Chapter 2

Data

In order to address the above stated problem, relevant data has been extracted from the API of <https://api-sports.io/> for the first division of the following countries: Argentina, Brazil, Colombia, Mexico, Chile, Perú, Spain, England, Italy, France, Germany, Netherlands and Portugal. Even though the web application will be focused in Latin America, the stats collected from the European leagues will be used to develop the overarching model that calculates performance indexes, so that it is not biased towards the performance of players only in Latin America, but rather takes into consideration a worldwide approach and then it is applied only to the region.

The data collected for these leagues include the following information:

- **Leagues:** seasons and rounds.
- **Teams:** information and statistics.
- **Venues:** name, location, capacity, surface.
- **Matches:** lineups, events, team statistics, player statistics.
- **Transfers:** transfers of players between teams.

A sample of the player's statistics data available for a match is provided:

```
{'player': {'id': 6627,
  'name': 'Nicolás Blandi',
  'photo': 'https://media.api-sports.io/football/players/6627.png'},
 'statistics': [{'games': {'minutes': 17,
  'number': 9,
  'position': 'F',
  'rating': '7.3',
  'captain': False,
  'substitute': False},
 'offsides': None,
 'shots': {'total': 1, 'on': 1},
 'goals': {'total': 1, 'conceded': None, 'assists': None, 'saves': None},
 'passes': {'total': 4, 'key': 0, 'accuracy': '66%'},
 'tackles': {'total': None, 'blocks': 0, 'interceptions': 0},
 'duels': {'total': 1, 'won': 1},
 'dribbles': {'attempts': 0, 'success': 0, 'past': None},
 'fouls': {'drawn': None, 'committed': None},
 'cards': {'yellow': 0, 'red': 0},
 'penalty': {'won': None,
  'committed': None,
  'scored': 1,
  'missed': 0,
  'saved': None}}]}]}
```

A sample for each of the endpoints used from the API is included in Appendix A.

The information ranges from season 2016 to 2021, depending on each specific league. The raw data will be extracted from the API and stored in a PostgreSQL database, in which further transformations are necessary to create a dataset with relevant features. The final data model contains statistics of circa 15,625 players of 345 teams in 20,000 matches across 13 leagues.

Chapter 3

Methods & Procedures

3.1 Methodology

The problem will be broken down into several steps that will serve as building blocks of the overarching solution:

1. **Data Warehouse:** study of the API documentation, and design and implementation of the database for storing the raw data.
2. **ETLs:** development of scripts for extracting, transforming, and loading the data from the API to the data warehouse.
3. **Analytical Base Table (ABT):** creation of a dataset with all the relevant features needed to analyze players' performance and draw insights through the creation of data visualizations.
4. **Statistical Analysis:** Exploratory Data Analysis (EDA) and Data Viz. Analysis of the data to discover trends, patterns, and relevant features for describing players' performance. Feature engineering and possible need for extracting more data to create better variables. Creation of a single player performance index. Plots and visualizations to include in the Web App.
5. **Web Application Development:** design and development of back-end and front-end of the application, including callback functions to allow the user interact with the data.

The project will be developed entirely in Python using PyCharm as Integrated Development Environment (IDE) for the final solution, and Jupyter Notebooks for the EDA and initial data visualizations. A Github repository has been used to back the code and track its progress. All necessary code files and datasets can be found in https://github.com/fpretto/MiM_Analytics_Tesis. A summary with the contents of each folder and files in the repository can be found in Appendix B.

3.2 Data Modelling

The two first steps of the overarching solution are the foundations of all the posterior analysis and visualizations to be done. They entail extracting the data from the API, cleaning and harmonizing it, and finally storing it in a structured way in a database. This process is depicted graphically in Figure 3.1.

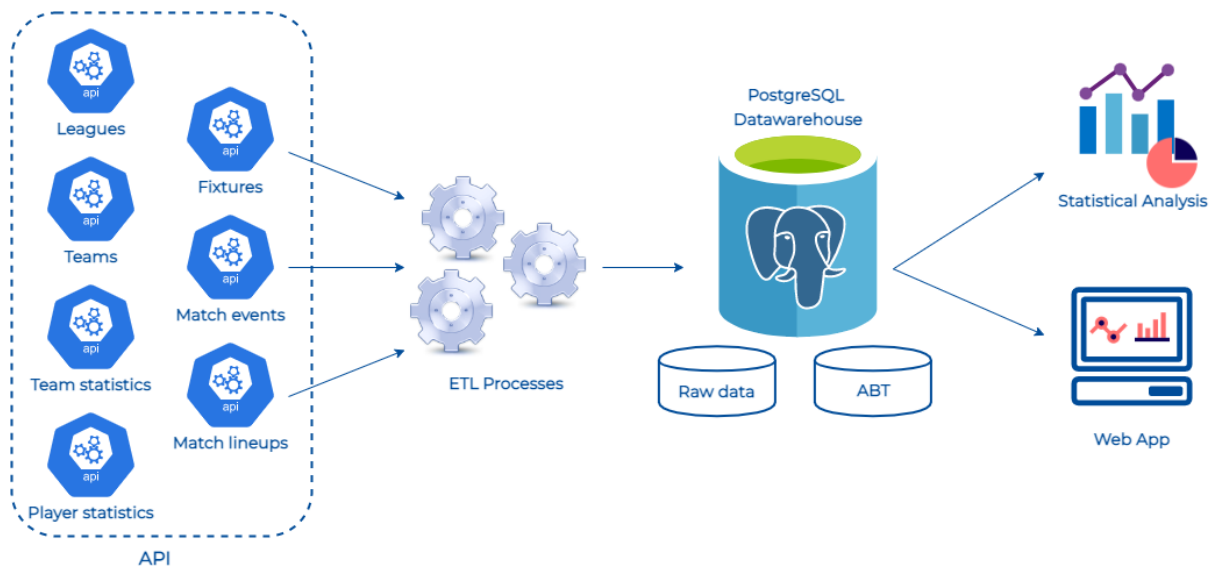


Figure 3.1: Data modelling pipeline

3.2.1 Data Warehouse

Database design mainly involves the design of the database schema. The entity-relationship (E-R) data model is a widely used data model for database design as it provides a convenient graphical representation to view data, relationships, and constraints.

E-R data models are made of two main components: entities and relationships. An *entity* is an object that exists in the real world and is distinguishable from other objects. This distinction is expressed by associating with each entity a set of attributes that describes the object. A *relationship* is an association among several entities. A *relationship set* is a collection of relationships of the same type, and an *entity set* is a collection of entities of the same type. For each entity set and for each relationship set in the database, there is a unique relation schema that is assigned the name of the corresponding entity set or relationship set. This forms the basis for deriving a relational database design from an E-R diagram.

The design process of a database can be splitted in three main design steps: *conceptual*, *logical* and *physical*.

The conceptual design provides a detailed overview of the solution. It specifies the entities that are represented in the database, the attributes of the entities, the relationships among the entities, and constraints on the entities and relationships.

Typically, the conceptual-design phase results in the creation of an E-R diagram that provides a graphic representation of the schema.

In the logical-design phase, the high-level conceptual schema is mapped onto the implementation data model of the database system that will be used. The implementation data model is typically the relational data model, and this step usually consists of mapping the conceptual schema defined using the E-R model into a relation schema.

Finally, the resulting system-specific database schema is used in the subsequent physical-design phase, in which the physical features of the database are specified. These features include the form of file organization and choice of index structures.

The physical schema of a database can be changed relatively easily after an application has been built. However, changes to the logical schema are usually harder to carry out, since they may affect a number of queries and updates scattered across the application code.

Conceptual Design

The E-R data model employs three basic concepts: entity sets, relationship sets, and attributes. It also has an associated diagrammatic representation, the E-R diagram:

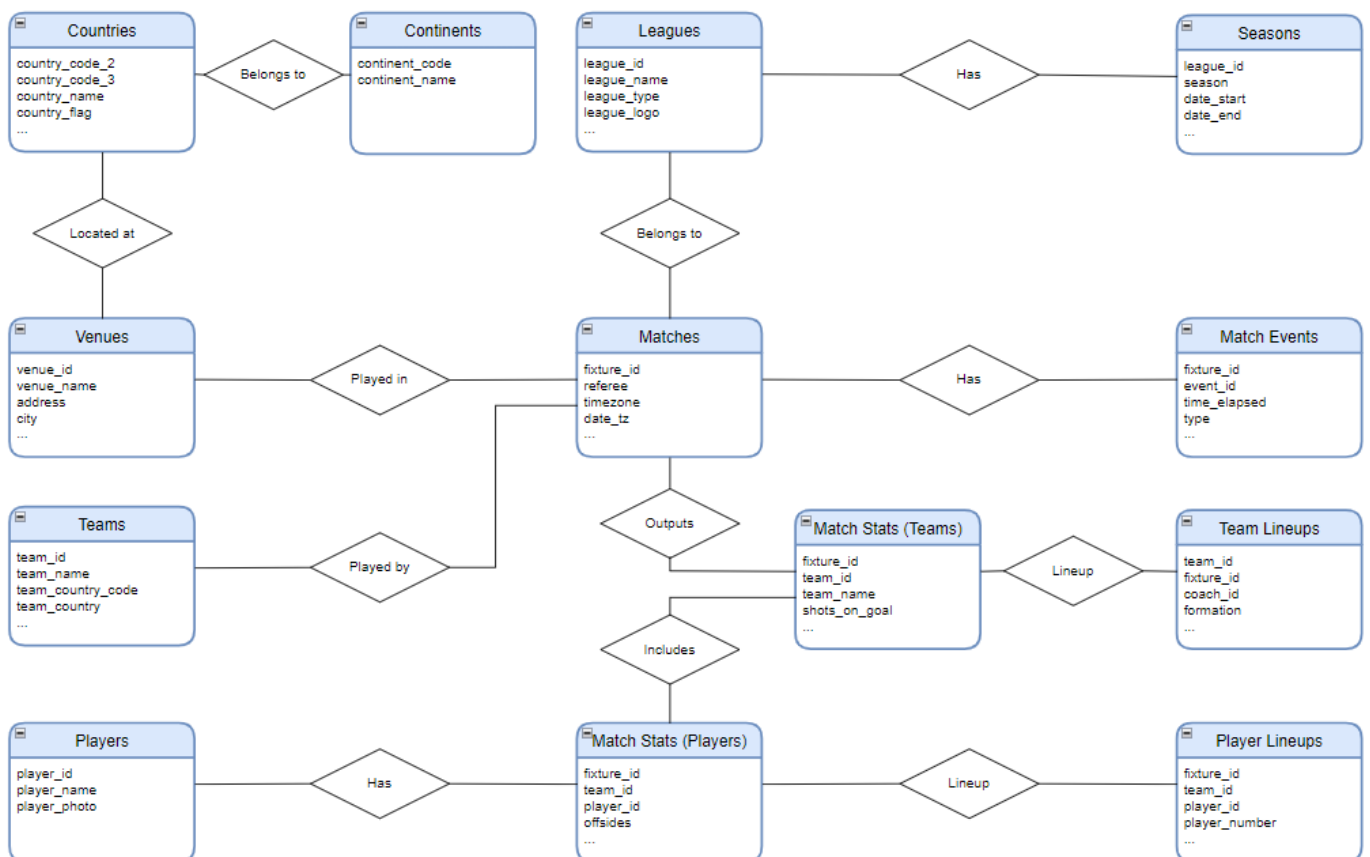


Figure 3.2: Datawarehouse conceptual design

Logical Design

In general, the goal of relational database design is to generate a set of relation schemas that allows us to store information without unnecessary redundancy, yet also allows us to retrieve information easily. This is accomplished by designing schemas that are in an appropriate normal form.

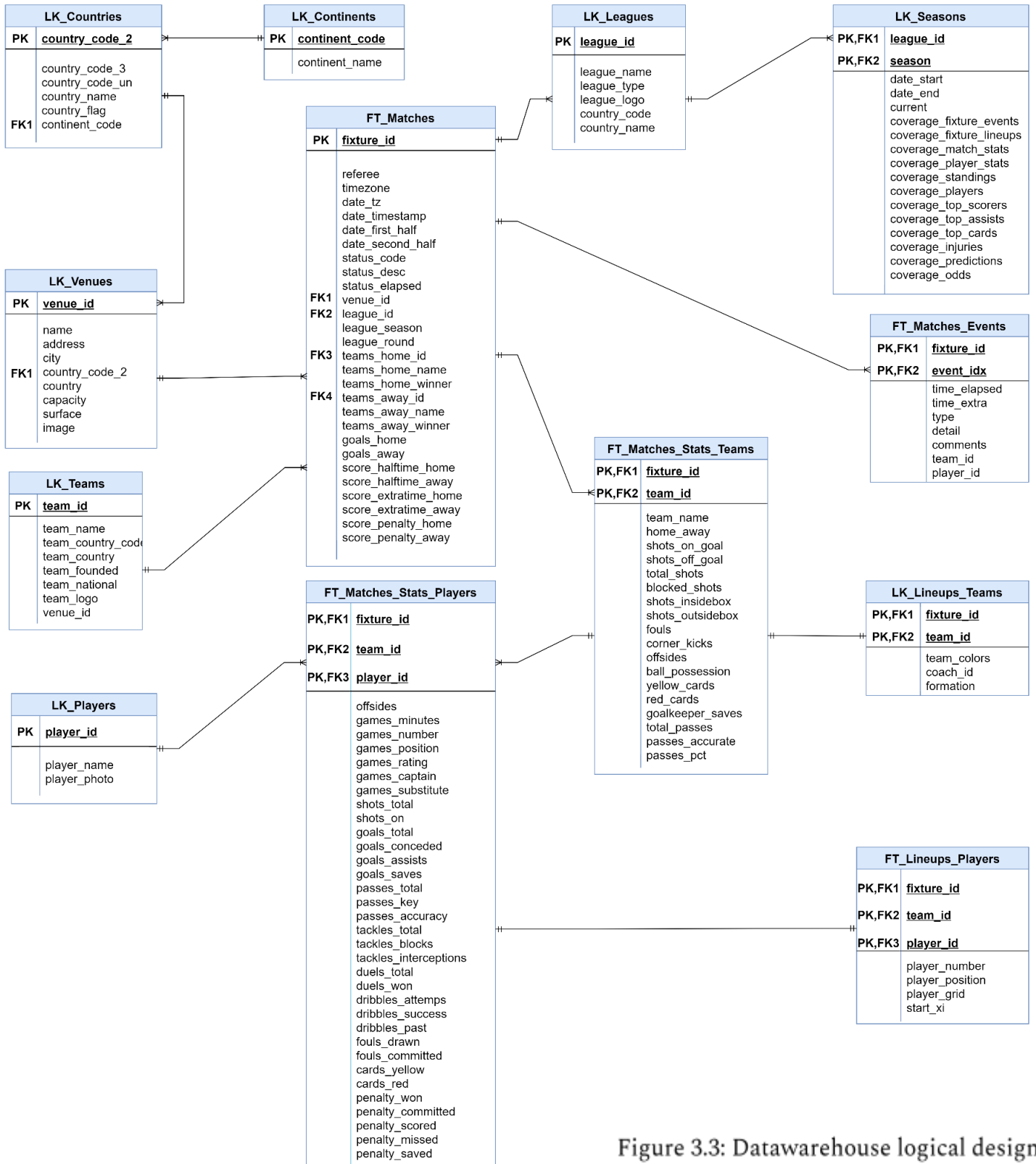
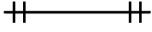

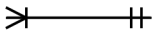


Figure 3.3: Datawarehouse logical design

Mapping cardinalities, or cardinality ratios, express the number of entities to which another entity can be associated via a relationship set. The cardinalities of the relationships between the entities sets depicted in our logical design are of three possible types:

- 
- **One-to-One:** An entity in A is associated with at most one entity in B, and an entity in B is associated with at most one entity in A.
- 
- **One-to-Many:** An entity in A is associated with any number (zero or more) of entities in B. An entity in B, however, can be associated with at most one entity in A.
- 
- **Many-to-One:** An entity in A is associated with at most one entity in B. An entity in B, however, can be associated with any number (zero or more) of entities in A.

Physical Design

The objective of physical database design is to specify how database records are stored, accessed, and related in order to ensure adequate performance of a database application. Physical database design is related to query processing, physical data organization, indexing, transaction processing, and concurrency management, among other characteristics. Since the size of the database for this project is not big enough to raise concerns about query performance, and the focus is on the analytical use of the data, this aspect of the design was out of scope.

3.2.2 ELT vs ETL vs EtLT

Data pipelines are sets of processes that move and transform data from various sources to a destination where new value can be derived. They are the foundation of analytics, reporting, and machine learning capabilities.

Both the acronyms ELT and ETL stand for the *Extract*, *Transform* and *Load* phases of a usual data pipeline. The difference between the two is the order of their final two steps (transform and load), but the design implications in choosing between them are substantial.

The *Extract* step gathers data from various sources in preparation for loading and transforming.

The *Load* step brings either the raw data (in the case of ELT) or the fully transformed data (in the case of ETL) into the final destination. Either way, the end result is loading data into the data warehouse.

The *Transform* step is where the raw data from each source system is combined and formatted in such a way that it's useful to analysts, visualization tools, or whatever use case the pipeline is serving.

Over the past years ELT has become the most common pattern for pipelines built for data analysis, data science, and data products. This pattern reduces the need to predict exactly what analysts will do with the data at the time of building extract and load processes. Though understanding the general use case is required to extract and load the proper data, saving the transform step for later gives analysts more options and flexibility.

When ELT emerged as the dominant pattern, it became clear that doing some transformation after extraction, but before loading, was still beneficial. However, instead of transformation involving business logic or data modelling, this type of transformation is more limited in scope. This blended pattern is usually referred to as EtLT. Some examples of the transformations represented by the *lowercase t* are:

- Parse JSON data sources into tabular form
- Deduplicate records in a table
- Essential data cleaning and missing values
- Any transformation addressing data quality issues

For the purpose of this project the EtLT pattern was used to create the data pipeline that feeds the data warehouse, as illustrated in Figure 3.4.

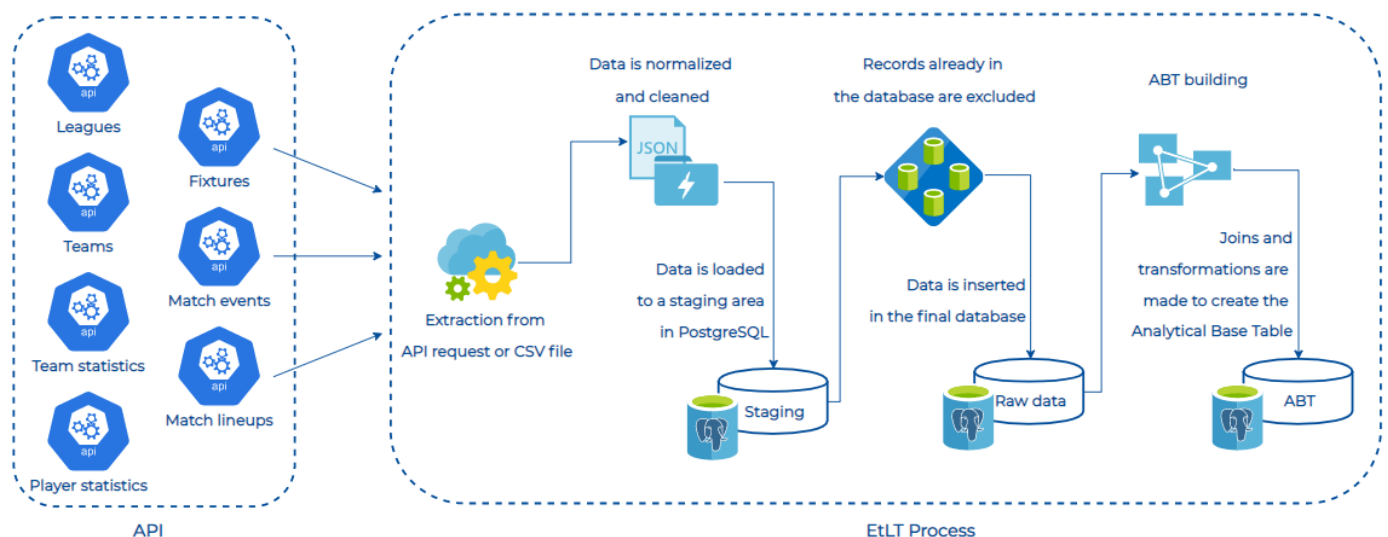


Figure 3.4: EtLT process implemented for the data pipeline

Implementation

In order to develop and implement the EtLT data pipeline that would extract the data from the API, perform quality related transformations and load it to the data warehouse, an *etl_config* JSON file was created with all the necessary parameters to

run the process, such as paths, PostgreSQL and API connection parameters, and available API endpoints. Four classes were defined to build the pipeline:

class Extract

This class is aimed at enclosing attributes and methods related to extracting data from sources. The class is initialized by providing the *etl_config* file, and has two methods.

The method *getCSVdata()* loads a CSV file, given the name of the file and the separator. The path is specified in the config file and should be the same for all CSV inputs.

The method *getAPIFootballData()* extracts the data from the API. Its parameters are the endpoint of the API to which to make the request, and the specific query to add in the request, if applicable.

class Transformation

The purpose of this class is to address data quality issues and minimal standardization, namely the *lowercase t* of the EtLT pattern. The class is initialized by providing the response of the API request and has 14 methods, one for each of the possible data sources (both possible CSV files and API endpoints).

In general, the procedure of these methods is to first convert the JSON response into tabular form. Next the column names of the resulting dataframe are harmonized to keep a uniform structure. All column names are in lowercase and spaces are denoted with an underscore.

For the endpoints in which a query is necessary, the query parameters are added to the table. For example, match and player statistics are retrieved by requesting the data for each match, and the *fixture_id* used as parameter is not included in the response. Therefore, it is added to the dataframe before loading it into the data warehouse.

Other transformations include adjusting data format (from string to decimal, and viceversa) or, in the case of percentages that are presented as strings, delete first the “%” symbol and then convert them to decimal.

class Load

This class contains all necessary methods to connect to the PostgreSQL database and load the extracted data. It is initialized by providing the *etl_config* file, and has three main methods.

The method *load_into_staging()* receives as input the Pandas dataframe to load and the name of the table in the datawarehouse. It drops the old table in the Staging area

and creates a new empty one, saves the dataframe into a memory buffer and finally copies the data from the buffer to the new table.

The method `insert_from_staging()` takes as input the name of the table in the datawarehouse. It inserts the data from the table in the Staging area to the final table, excluding the records that are already present.

Finally, the method `load_batch()` encapsulates the two described methods and executes them sequentially.

class ETL_Engine

This is the main class that guides the whole process by making use of the previously defined classes in the different steps of the process. It is initialized by providing the `etl_config` file, and has 13 methods, one for each of the entity sets in the datawarehouse.

Each method extracts the data, performs the necessary transformations, loads it to the staging area, and then to the final table.

3.3 Analytical Base Table

An Analytical Database is a common set of base tables (ABTs) that can be used across multiple analytical projects. In general its more common use cases are predictive models, but it can be used to support any analytical task. ABTs are used by smart analytically driven enterprises to get the best results from their investment in analytics.

For the purpose of this project, an ABT was built to summarize a player's performance in a given league and season. Each row of the table has the tuple `player_id-season` as primary key, qualitative information such as position, number and team; and 30 features describing the performance of the player in the season.

league_season integer	player_id integer	player_name text	player_preferred_position character varying (2)	player_preferred_number integer	team_id integer	player_minutes bigint	shots_total bigint	shots_on_goal bigint	goals_total bigint
2021	2455	Fernando Aristeguieta	F	9	2291	516	13	8	1
2021	2557	Pablo Hernandez	M	16	2320	1034	15	8	2
2021	1634	Cristian Zapata	D	3	460	1050	7	4	1
2021	2453	Luis Manuel Seijas	M	20	1139	22	1	1	0
2021	2502	Gustavo Gomez	D	15	121	1719	14	3	1
2021	2556	Sebastian Vegas	D	20	2282	988	6	2	0
2021	576	Eduardo Salvio	M	11	451	152	3	3	0
2021	1619	Franco Soldano	F	27	451	347	3	3	2
2021	2398	Alejandro Chumacero	M	3	2321	995	16	6	2
2021	2448	Arquimedes Figuera	M	8	2541	1674	8	2	0
2021	2482	Camilo Vargas	G	12	2283	1549	0	0	0
2021	2501	Juan Escobar	D	24	2295	1213	9	2	0
2021	2518	Carlos Gonzalez	F	32	2279	871	23	6	3
2021	2555	Oscar Opazo	D	16	2315	2486	7	5	0
2021	195	Joao Miranda	D	22	126	1900	4	2	0

Table 3.1: Analytical Base Table sample

A sample is provided in Table 3.1. The full list of features used is included in Appendix C.

Regarding the position, number and team for each player in each season, for the sake of simplicity the value with more minutes of game played per season was assigned, since a player may switch teams during the course of a season, as well as play in different positions and with different numbers.

In order to normalize the data and make it comparable across different players and positions, some typical transformations in the field of football analysis were made, such as *Per 90* and *Possession-adjusted* stats.

Per 90 Stats

It is tempting to compare players directly on raw, basic numbers such as the sum of shots, goals made and passes in a season. These are useful numbers to have to gain a basic understanding of how a player performed in any given game or season. However, the problem with these is that they do not factor the individual's minutes played, leading in many cases to wrong conclusions when comparing players' performances.

To solve this issue, all statistics are normalized to a per 90 minutes basis, meaning that they are divided by the sum of minutes in the field for the player, and then multiplied by 90. For example, if a player made 10 goals in a season and played 600 minutes, then the metric *Goals P90* would be $10/600*90=1.5$ (goals per 90 minutes played). These metrics provide a more fair way of comparing players with different time spent on the field.

Possession-adjusted

As Wyscout [20] defines it, Possession-adjusted (PAdj) is a method to calculate defensive statistics to take possession values into account. The logic for adjusting defensive statistics is the following: players can only make defensive contributions when they're not in possession of the ball. Therefore, when high defensive values are looked at, normally the defenders of the lower teams in the league would only be encountered: their defenders, being dominated by a possession-leading team, are forced to make more actions (defensive duels, interception, tackles, etc.). The defenders of possession-based teams are naturally making less actions. Adjusting these values to the possession (as if the match was played with a 50%/50% possession) gives further insight to the frequency of defensive actions.

For example, let's suppose that in a given match team A had 80% of possession, whereas team B had 20%. Player 1, who played the whole match for team B, had 10 interceptions. His *PAdj interceptions* value would be $10/0.8*0.5=6.25$. Player 2, who played the whole match for team A, had 5 interceptions. His *PAdj interceptions* value

would be $5/0.2 \times 0.5 = 12.5$. Therefore, while Player 2 made only half of the interceptions of Player 1, the huge difference in possession makes his possession-adjusted value twice higher.

3.4 Statistical Analysis

The ultimate goal of this project is to have a robust indicator to search and compare players, in order to make smart decisions regarding player scouting. The main reason for this objective is that it often seems easier for people to interpret composite indicators than to identify common trends across many separate indicators.

In general terms, an indicator is a quantitative or a qualitative measure derived from a series of observed facts that can reveal relative positions in a given area. When evaluated at regular intervals, an indicator can point out the direction of change across different units and through time. A composite indicator is formed when individual indicators are compiled into a single index on the basis of an underlying model.

In order to build a robust composite index for measuring players' performance, guidelines presented in the *Handbook on Constructing Composite Indicators*, by the Organization for Economic Co-operation and Development (OECD) were followed. The main aim of this Handbook is to provide builders of composite indicators with a set of recommendations on how to design, develop and disseminate a composite indicator. These recommendations are presented in an “ideal sequence” of ten steps, summarized in Table 3.2. Each step is important, but coherence in the whole process is equally vital.

Step	Description	Why it is needed
Theoretical framework	Provides the basis for the selection and combination of variables into a meaningful composite indicator under a fitness-for-purpose principle.	<ul style="list-style-type: none"> • To get a clear understanding and definition of the multidimensional phenomenon to be measured. • To structure the various sub-groups of the phenomenon (if needed). • To compile a list of selection criteria for the underlying variables, e.g., input, output, process.
Data selection	Should be based on the analytical soundness, measurability, coverage, and relevance of the indicators to the phenomenon being measured and relationship to each other. The use of proxy variables should be considered when data is scarce.	<ul style="list-style-type: none"> • To check the quality of the available indicators. • To discuss the strengths and weaknesses of each selected indicator. • To create a summary table on data characteristics, e.g., availability, source, type.

Imputation of missing data	Is needed in order to provide a complete dataset (e.g. by means of single or multiple imputation).	<ul style="list-style-type: none"> • To estimate missing values. • To provide a measure of the reliability of each imputed value, so as to assess the impact of the imputation on the composite indicator results. • To discuss the presence of outliers in the dataset.
Multivariate analysis	Should be used to study the overall structure of the dataset, assess its suitability, and guide subsequent methodological choices (e.g., weighting, aggregation).	<ul style="list-style-type: none"> • To check the underlying structure of the data along the two main dimensions, namely individual indicators and countries. • To identify groups of indicators or groups of countries that are statistically “similar” and provide an interpretation of the results. • To compare the statistically-determined structure of the data set to the theoretical framework and discuss possible differences.
Normalization	Should be carried out to render the variables comparable.	<ul style="list-style-type: none"> • To select suitable normalization procedure(s) that respect both the theoretical framework and the data properties. • To discuss the presence of outliers in the dataset as they may become unintended benchmarks. • To make scale adjustments, if necessary. • To transform highly skewed indicators, if necessary.
Weighting and aggregation	Should be done along the lines of the underlying theoretical framework.	<ul style="list-style-type: none"> • To select appropriate weighting and aggregation procedure(s) that respect both the theoretical framework and the data properties. • To discuss whether correlation issues among indicators should be accounted for. • To discuss whether compensability among indicators should be allowed.
Uncertainty and sensitivity analysis	Should be undertaken to assess the robustness of the composite indicator in terms of e.g., the mechanism for including or excluding an indicator, the normalization scheme, the imputation of missing data, the choice of weights, the aggregation method.	<ul style="list-style-type: none"> • To consider a multi-modelling approach to build the composite indicator, and if available, alternative conceptual scenarios for the selection of the underlying indicators. • To identify all possible sources of uncertainty in the development of the composite indicator and accompany the composite scores and ranks with uncertainty bounds. • To conduct sensitivity analysis of the inference (assumptions) and determine what sources of uncertainty are more influential in the scores and/or ranks.
Back to the data	Is needed to reveal the main drivers for an overall good or bad	<ul style="list-style-type: none"> • To profile country performance at the

	performance. Transparency is primordial to good analysis and policy-making.	indicator level so as to reveal what is driving the composite indicator results. <ul style="list-style-type: none"> • To check for correlation and causality (if possible). • To identify if the composite indicator results are overly dominated by few indicators and to explain the relative importance of the sub-components of the composite indicator.
Links to other indicators	Should be made to correlate the composite indicator (or its dimensions) with existing (simple or composite) indicators as well as to identify linkages through regressions.	<ul style="list-style-type: none"> • To correlate the composite indicator with other relevant measures, taking into consideration the results of sensitivity analysis. • To develop data-driven narratives based on the results.
Visualization of the results	Should receive proper attention, given that the visualization can influence (or help to enhance) interpretability	<ul style="list-style-type: none"> • To identify a coherent set of presentational tools for the targeted audience. • To select the visualization technique which communicates the most information. • To present the composite indicator results in a clear and accurate manner.

Table 3.2: Steps for constructing a Composite Indicator (OECD)

It is worth noting that the guidelines presented in this handbook are mainly intended for constructing economic composite indicators for comparing countries and implementing governments' policies. Therefore, much of the examples and directions in which the guidelines are written are towards those goals. The overall framework proposed will be followed as much as possible to ensure robustness of the results, but in some cases the recommendations presented don't apply in the context of football players' performance measurement. In fact, for the purpose of this project, these ten steps were reduced and/or aggregated into four.

Steps 1, 2 and 3: Theoretical framework, Data selection and Imputation of missing data

The basis for the selection and combination of variables was taken from leading football analytics platforms such as StatsBomb, Soccerment, Wyscout and StatsPerform. All these platforms assess players' performance by selecting and combining specific variables for each position in the football pitch.

The final variable selection for this project was structured based on these benchmarks and on the API coverage, availability and measurability of the different aspects of the game. Table 3.3 contains the variables selected for each role.

Role	Variable
Goalkeeper	Saves per 90 minutes Goals conceded per 90 minutes Goals conceded ratio Total passes per 90 minutes Passing accuracy Fouls drawn per 90 minutes Fouls committed per 90 minutes Duels per 90 minutes Duels success ratio Tackles per 90 minutes Interceptions per 90 minutes Penalties committed per 90 minutes
Defender	Total passes per 90 minutes Passing accuracy Key passes per 90 minutes Scoring contribution (Goals P90 + Assists P90) Fouls drawn per 90 minutes Fouls committed per 90 minutes Dribbles past per 90 minutes Duels per 90 minutes Duels success ratio Tackles per 90 minutes Blocks per 90 minutes Interceptions per 90 minutes
Midfielder	Total passes per 90 minutes Passing accuracy Key passes per 90 minutes Scoring contribution (Goals P90 + Assists P90) Dribble success ratio Fouls drawn per 90 minutes Fouls committed per 90 minutes Dribbles past per 90 minutes Tackles per 90 minutes Interceptions per 90 minutes
Forward	Non-penalty goals per 90 minutes Shots per 90 minutes Shooting accuracy Non-penalty goal conversion Passing accuracy Assists per 90 minutes Key passes per 90 minutes Dribbles per 90 minutes Dribble success ratio Tackles per 90 minutes

Table 3.3: Variables used to describe players' performance by role

The imputation of missing data in most of these variables is relatively straightforward: a null value would imply that the player didn't perform any of the action described by the variable (e.g. a null value in goals would mean that the player didn't score any goals).

Many of the players in the database have not played enough minutes in a given season. Therefore, to ensure having data with statistical relevance but also not missing out possible outperformers with not much playing time, a threshold of 270 minutes (three matches) in a season was set to filter out under-represented players.

Steps 4, 5 and 6: Multivariate analysis, Normalization and Weighting and aggregation

Principal Component Analysis

Different analytical approaches, such as principal components analysis, can be used to explore whether the dimensions of the phenomenon are statistically well-balanced in the composite indicator. The goal of principal components analysis (PCA) is to reveal how different variables change in relation to each other and how they are associated. This is achieved by transforming correlated variables into a new set of uncorrelated variables using a covariance matrix or its standardized form – the correlation matrix.

The objective is to explain the variance of the observed data through a few linear combinations of the original data. Even though there are Q variables x_1, x_2, \dots, x_Q , much of the data's variation can often be accounted for by a small number of variables – principal components, or linear relations of the original data, Z_1, Z_2, \dots, Z_Q that are uncorrelated. At this point there are still Q principal components, i.e., as many as there are variables. The next step is to select the first, e.g., $P < Q$ principal components that preserve a “high” amount of the cumulative variance of the original data.

$$\begin{aligned}
 Z_1 &= a_{11}x_1 + a_{12}x_2 + \dots + a_{1Q}x_Q \\
 Z_2 &= a_{21}x_1 + a_{22}x_2 + \dots + a_{2Q}x_Q \\
 &\dots \\
 Z_Q &= a_{Q1}x_1 + a_{Q2}x_2 + \dots + a_{QQ}x_Q
 \end{aligned}
 \tag{1}$$

A lack of correlation in the principal components is a useful property. It indicates that the principal components are measuring different “statistical dimensions” in the data. When the objective of the analysis is to present a huge data set using a few variables, some degree of economy can be achieved by applying Principal Components Analysis (PCA) if the variation in the Q original x variables can be accounted for by a small number of Z variables. The weights a_{ij} (also called

component or factor loadings) applied to the variables x_j in equation (1) are chosen so that the principal components Z_i satisfy the following conditions:

- (i) they are uncorrelated (orthogonal);
- (ii) the first principal component accounts for the maximum possible proportion of the variance of the set of x s, the second principal component accounts for the maximum of the remaining variance, and so on until the last of the principal components absorbs all the remaining variance not accounted for by the preceding components.

PCA involves finding the eigenvalues λ_j , $j=1,\dots,Q$, of the sample covariance matrix CM, where the diagonal element cm_{ii} is the variance of x_i and cm_{ij} is the covariance of variables x_i and x_j . The eigenvalues of the matrix CM are the variances of the principal components and can be found by solving the characteristic equation $|CM - \lambda I| = 0$, where I is the identity matrix with the same order as CM and λ is the vector of eigenvalues.

Factor Analysis

Factor analysis (FA) is similar to PCA. It aims to describe a set of Q variables x_1, x_2, \dots, x_Q in terms of a smaller number of m factors and to highlight the relationship between these variables. However, while PCA is based simply on linear data combinations, FA is based on a rather special model. Contrary to the PCA, the FA model assumes that the data is based on the underlying factors of the model, and that the data variance can be decomposed into that accounted for by common and unique factors.

The model is given by:

$$\begin{aligned}
 x_1 &= \alpha_{11}F_1 + \alpha_{12}F_2 + \dots + \alpha_{1m}F_m + e_1 \\
 x_2 &= \alpha_{21}F_1 + \alpha_{22}F_2 + \dots + \alpha_{2m}F_m + e_2 \\
 &\dots \\
 x_Q &= \alpha_{Q1}F_1 + \alpha_{Q2}F_2 + \dots + \alpha_{Qm}F_m + e_Q
 \end{aligned}
 \tag{2}$$

where x_i ($i=1,\dots,Q$) represents the original variables but standardized with zero mean and unit variance; $\alpha_{i1}, \alpha_{i2}, \dots, \alpha_{im}$ are the factor loadings related to the variable X_i ; F_1, F_2, \dots, F_m are m uncorrelated common factors, each with zero mean and unit variance; and e_i are the Q specific factors' errors supposedly independently and identically distributed (i.i.d.) with zero mean.

The most common approach to dealing with the model given in equation (2) is the use of PCA to extract the first m principal components and to consider them as factors, neglecting those remaining. Principal components factor analysis is most

preferred in the development of composite indicators as it has the virtue of simplicity and allows for the construction of weights representing the information content of individual indicators.

On the issue of how many factors should be retained in the analysis without losing too much information, methodologists are divided. The decision on when to stop extracting factors depends basically on when there is only very little “random” variability left, and is rather arbitrary. However, various guidelines (“stopping rules”) have been developed, roughly in the order of frequency of their use in social science (Dunteman, 1989: 22-3):

Stopping Rule	Methodology
Kaiser criterion	Drop all factors with eigenvalues below 1.0. The simplest justification for this is that it makes no sense to add a factor that explains less variance than is contained in one individual indicator.
Scree plot	Plots the successive eigenvalues, which drop sharply and then level off. It suggests retaining all eigenvalues in the sharp descent before the first one on the line where they start to level off.
Variance explained criteria	Some researchers simply use the rule of keeping enough factors to account for 70% (sometimes 60%) of the variation.
Joliffe criterion	Drop all factors with eigenvalues under 0.70. This rule may result in twice as many factors as the Kaiser criterion produces and is less often used.
Comprehensibility	Though not a strictly mathematical criterion, there is much to be said for limiting the number of factors to those whose dimension of meaning is readily comprehensible. Often this means the first two or three.

Table 3.4: Stopping rules to decide when to stop extracting factors in PCA

After choosing the number of factors to keep, it is standard practice to perform rotation so as to enhance the interpretability of the results. The sum of eigenvalues is not affected by rotation (the analytical solutions obtained ex-ante and ex-post the rotation is unchanged), but changing the axes will alter the eigenvalues of particular factors and will change the factor loadings. The idea behind transforming the factorial axes is to obtain a “simpler structure” of the factors (ideally a structure in which each indicator is loaded exclusively on one of the retained factors). The most common rotation method is the “varimax rotation”.

Summarizing the steps of the PCA/FA exploratory analysis method:

1. Calculate the covariance/correlation matrix: if the correlation between individual indicators is small, it is unlikely that they share common factors.
2. Identify the number of factors necessary to represent the data and the method for calculating them.
3. Rotate factors to enhance their interpretability (by maximizing loading of individual indicators on individual factors).

Normalization

Different normalization methods will produce different results for the composite indicator. Three of the most common techniques for normalizing the data are standardization, robust standardization and min-max.

In the standardization (or z-score) method, for each individual indicator x , the average across players \bar{x} and the standard deviation across players σ_x are calculated.

The normalization formula is

$$X_i = \frac{x_i - \bar{x}}{\sigma_x} \quad (3)$$

so that all X_i have similar dispersion across players. The drawback of this method is that, as it is based on the average, it is very sensitive to outliers.

An alternative to standardization is using a robust approach, substituting the mean with the median, and the standard deviation with the interquartile range. The resulting formula for robust normalization is:

$$X_i = \frac{x_i - x_{Median}}{(x_{75} - x_{25})} \quad (4)$$

This approach is less sensitive to outliers as it is based on more robust statistics.

Finally, the min-max approach calculates the standardized values as:

$$X_i = \frac{x_i - \min_x}{(\max_x - \min_x)} \quad (5)$$

In this way, the normalized indicators X_i have values lying between 0 (laggard, $X_i = \min_x$), and 1 (leader, $X_i = \max_x$). However, this transformation is not stable when data for a new time point becomes available. This implies an adjustment of the analysis period T , which may in turn affect the minimum and the maximum for some individual indicators and hence the values of X_i . To maintain comparability between

the existing and the new data, the composite indicator for the existing data must be re-calculated.

Weighting and aggregation

According to PCA/FA, weighting intervenes only to correct for overlapping information between two or more correlated indicators and is not a measure of the theoretical importance of the associated indicator. If no correlation between indicators is found, then weights cannot be estimated with this method.

The last step in PCA/FA deals with the construction of the weights from the matrix of factor loadings after rotation, given that the square of factor loadings represents the proportion of the total unit variance of the indicator which is explained by the factor. The approach used by Nicoletti et al., (2000) is that of grouping the individual indicators with the highest factor loadings into intermediate composite indicators. The final intermediate composites are aggregated by assigning a weight to each one of them equal to the proportion of the explained variance in the dataset. Note that different methods for the extraction of principal components imply different weights, hence different scores for the composite.

Steps 7, 8 and 9: Uncertainty and sensitivity analysis, Back to the data and Links to other indicators

Composite indicator development involves stages where subjective judgements have to be made: the selection of individual indicators, the treatment of missing values, the choice of aggregation model, the weights of the indicators, etc. All these subjective choices are the bones of the composite indicator and, together with the information provided by the numbers themselves, shape the message communicated by the composite indicator.

Sensitivity analysis is the study of how the variation in the output can be apportioned, qualitatively or quantitatively, to different sources of variation in the assumptions, and of how the given composite indicator depends upon the information fed into it. Sensitivity analysis is thus closely related to uncertainty analysis, which aims to quantify the overall uncertainty in players' performance index as a result of the uncertainties in the model input. A combination of uncertainty and sensitivity analysis can help to gauge the robustness of the composite indicator, to increase its transparency, to identify which players are favored or weakened under certain assumptions and to help frame a debate around the index.

Typically the most common approach is to hold all the attributes at their mean value while varying just one of the inputs to assess the effect of changing just one variable. More advanced analyses could include varying multiple inputs at the same time to study the combined effect of multiple variables.

For this project, we will vary one attribute at a time at multiple steps to assess the overall sensitivity of each variable. Adding half a step means we are using the midpoint value between the mean and maximum for a single variable and adding a full step means we will use the maximum value for that variable. Figure 3.5 illustrates the concept behind the stepwise increasing and decreasing variable value from the mean to the maximum or minimum:

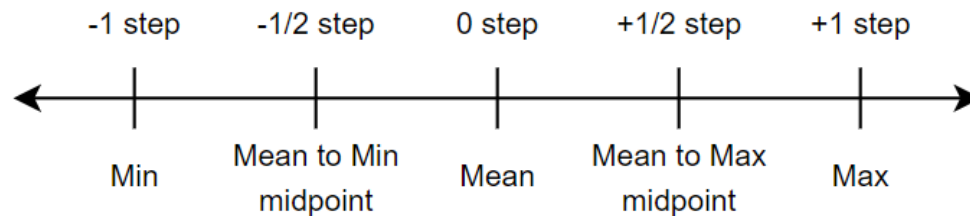


Figure 3.5: stepwise variation from the mean for sensitivity analysis

After iterating through all the attributes and varying single variables by increasing and decreasing their value by our predefined steps, we can plot the effect each input had on the output.

Step 10: Visualization of the results

Data visualization reveals data features that statistics and models may miss: unusual distributions of data, local patterns, clusterings, gaps, missing values, evidence of rounding or heaping, implicit boundaries, outliers, and so on. Graphics raise questions that stimulate research and suggest ideas.

While training the model for calculating the performance index, a combination of plots will be used to assess its resulting distribution, as well as its benchmark against the API's player rating.

Moreover, data visualization will be a key aspect of the final web application. It is essential for the scouting tool to be easy to use and with clear plots and graphs that portray the picture of a player's performance.

Implementation

In order to develop the process for calculating the performance index and assessing the results, the *EDA_PerformanceIndex.ipynb* Jupyter notebook was used to prototype and test different approaches. This notebook deals with the preprocessing of the data (feature engineering, normalization and variable selection), factor analysis to calculate the index, assessment of results and sensitivity analysis.

Once the process had been tested, four classes were implemented in PyCharm for implementing the process in a more robust way.

class PI_Preprocessing

This class is aimed at performing all the necessary steps for preparing the data to calculate the performance index.

The method *filter_and_data_engineering()* filters the players with less than 270 minutes played, deals with missing values and creates the per 90 minutes variables, as well as other calculated features.

The method *normalize_by_position()* takes as a parameter the type of scaling to perform (Standard, Robust or MinMax) and applies it to the data, splitted by position, to normalize the features.

class PI_FactorAnalysis

This class performs the factor analysis to extract the factor loadings, normalize them, calculate the weights and aggregate them into the performance index. It also scores, i.e. calculates the performance index, for a dataset given as an input.

The method *get_factors()* applies the factor analysis, extracting the factor loadings and the variances, normalizes them and calculates the weights for each variable. It returns a table with all the factor loadings and weights per variable.

The method *create_index()* encapsulates the previous method and uses the weights to calculate the performance index. Since the objective is to obtain a metric between 0 and 100%, the resulting score is bounded to [0, 1] by normalizing the scores with a MinMax scaler.

The method *score_index()* is used to score a new set of data. It takes as an input the dataframe with the data and a dictionary with all the necessary objects for calculating the performance index (scalers, weights, etc.).

class PI_Main

This class utilizes the two previous classes for training the model that scores and persisting all necessary objects for calculating the performance index. These objects include a dictionary with the variables used for each position, scalers objects, and a dictionary with all the weights for each variable and position. All these objects are encapsulated in a dictionary that is exported as a pickle object.

class PI_Scoring

This class calculates the performance index for a given set of data. It takes a dataset and the pickle with all necessary objects, created in the previous class, and scores the data.

3.5 Web Application

There are several frameworks for developing a web application in python. Regardless on which one is used, the app should allow the user to:

- **Analyze:** manipulate and summarize data
- **Visualize:** display plots and graphs of the data to improve interpretability.
- **Interact:** slice and dice the data to analyze specific segments.

Three popular frameworks were considered for developing the app, for which some attributes are summarized in Table 3.5:

	Simplicity	Maturity	Flexibility	Primary Use
Dash	B	B	B	Dashboards
Streamlit	A	C	B	Dashboards
Flask	C	A	A	Web Interfaces

Table 3.5: Comparison between visualization frameworks

Though complex to use, Flask is by far the best option for developing a production level web application due to its flexibility to build a highly customized solution from the ground up. The downside is that it not only requires Python knowledge, but also HTML, and that the focus is more on serving pages and structuring the web framework rather than on visualizing data. Therefore, given that the intention of this project is to develop a prototype of a web application, without getting into unnecessary complexity, this option was discarded.

Streamlit and Dash are very similar libraries. They are both full dashboarding solutions built with Python, and both include components for data analysis, visualization, user interaction, and serving. Streamlit is more structured and focused on simplicity. It only supports Python-based data analysis and has a limited set of widgets (for example, sliders) to choose from. On the other hand, Dash is more adaptable. Although it's built with Python and pushes users towards its own plotting library (Plotly), it's also compatible with other plotting libraries and even other languages, such as R or Julia. In addition, it is built on top of Flask and uses Flask as its web routing component.

Despite their similarities, the fact that Dash uses Flask framework in the background represents an advantage. This is because an eventual migration from the prototype to a production level web application would be more straightforward, as they both share the same technology underneath. Therefore, the solution developed in this project will be based on Dash.

Chapter 4

Results

4.1 Player Performance Index

For the purpose of this project, PCA/FA analysis was used to develop the players' performance index, using a combination of the Kaiser criterion, the Scree plot, and the Variance explained criteria as stopping rules for extracting the most relevant factors, and varimax rotation to enhance their interpretability.

Regarding normalization, the three methods were used to assess their impact on the outcomes. The best results for preprocessing before applying PCA/FA were achieved using robust standardization. Finally, for scaling the resulting performance index to a range between 0 and 1, a MinMax approach was used.

4.1.1 Forwards

Analyzing the scree plot in figure 4.1, the method would suggest retaining the first three factors. In contrast, the Kaiser criterion would suggest retaining four or five.

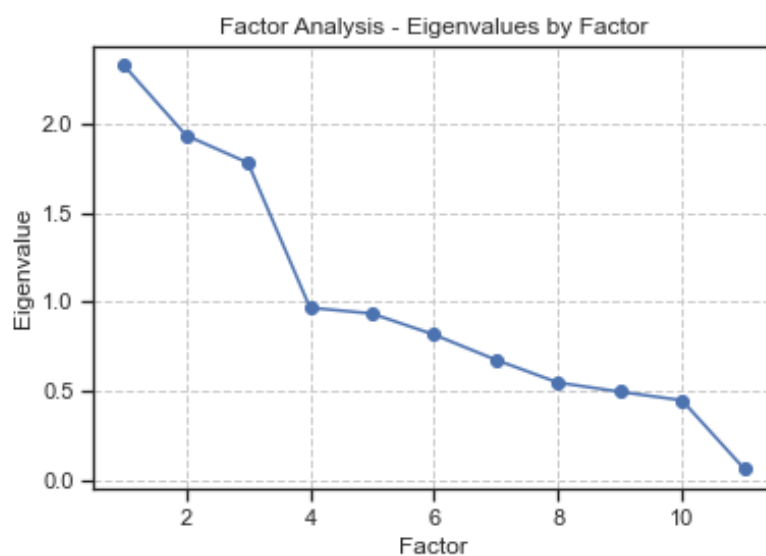


Figure 4.1: Scree plot for forwards

The standard practice recommended in the *OECD Handbook for Constructing Composite Indicators* is to choose factors that:

- (i) have associated eigenvalues larger than one;
- (ii) contribute individually to the explanation of overall variance by more than 10%;

(iii) contribute cumulatively to the explanation of the overall variance by more than 60%.

The use of the aforementioned practice resulted in retaining the first five factors, as shown in table 4.1.

	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Communalities	Sq Norm Factor 1	Sq Norm Factor 2	Sq Norm Factor 3	Sq Norm Factor 4	Sq Norm Factor 5	PC_Weight
np_goals_p90	0.89	0.03	0.37	0.02	-0.07	0.93	0.35	0.00	0.08	0.00	0.00	0.12
shots_p90	0.22	0.09	0.77	0.10	-0.08	0.67	0.02	0.01	0.36	0.01	0.00	0.08
shooting_accuracy	0.70	0.07	-0.36	0.01	-0.06	0.63	0.22	0.00	0.08	0.00	0.00	0.08
goal_conversion_np	0.91	-0.02	-0.05	-0.04	-0.05	0.83	0.37	0.00	0.00	0.00	0.00	0.10
passing_accuracy	-0.23	0.03	0.78	-0.05	0.02	0.67	0.02	0.00	0.36	0.00	0.00	0.08
assists_p90	0.05	0.88	-0.00	-0.03	0.00	0.79	0.00	0.51	0.00	0.00	0.00	0.10
key_passes_p90	-0.00	0.80	0.11	0.11	0.25	0.72	0.00	0.42	0.01	0.01	0.04	0.09
dribbles_p90	-0.05	0.29	0.09	0.16	0.67	0.57	0.00	0.06	0.01	0.03	0.30	0.07
dribbles_success_ratio	-0.02	0.07	0.01	0.96	0.05	0.93	0.00	0.00	0.00	0.90	0.00	0.12
tackles_p90	0.06	-0.02	-0.42	0.15	0.60	0.56	0.00	0.00	0.11	0.02	0.24	0.07
interceptions_p90	-0.13	0.04	0.00	-0.15	0.78	0.65	0.01	0.00	0.00	0.02	0.41	0.08
Variance	2.23	1.53	1.68	1.03	1.50	-	-	-	-	-	-	-
Proportional Variance (%)	0.20	0.14	0.15	0.09	0.14	-	-	-	-	-	-	-
Cummulative (%)	0.20	0.34	0.49	0.59	0.72	-	-	-	-	-	-	-
Expl.Var/Tot (%)	0.28	0.19	0.21	0.13	0.19	-	-	-	-	-	-	-

Table 4.1: Factor analysis for forwards

Table 4.1 contains the results for the factor analysis of the forwards' performance. The last four rows contain the variance explained by each of the factors. *Proportional variance (%)* is the variance explained by the factor and *Expl./Tot (%)* is the explained variance divided by the total variance of the five factors.

The rotation is used to minimize the number of individual indicators that have a high loading on the same factor, thus enhancing interpretability. The last step deals with the construction of the weights from the matrix of factor loadings after rotation, given that the square of factor loadings represents the proportion of the total unit variance of the indicator which is explained by the factor. The approach used by Nicoletti et al., (2000) is that of grouping the individual indicators with the highest factor loadings into intermediate composite indicators. The five intermediate composites are aggregated by assigning a weight to each one of them equal to the proportion of explained variance in the dataset.

For example, for the first variable *np_goals_p90* the *Squared Norm Factor 1* would be calculated as:

$$SqNorm\ Factor\ 1 = \frac{(Factor\ loading)^2}{Explained\ variance} = \frac{0.89^2}{2.23} = 0.35$$

The final weight for the variable *np_goals_p90* is obtained by aggregating the five intermediate composites and weighting them by the relative explained variance:

$$\text{Weight} = \sum(\text{SqNorm Factor} * \text{Expl. Var}/\text{Tot.})$$

$$\text{Weight} = 0.35 * 0.28 + 0.00 * 0.19 + 0.08 * 0.21 + 0.00 * 0.13 + 0.00 * 0.19$$

$$\text{Weight} = 0.12$$

Judging by the final vector of weights obtained, it would seem that the most relevant variables for summarizing a forward's performance are *np_goals_p90* (non-penalty goals per 90 minutes), *dribble_success_ratio*, *assists_p90* and *goal_conversion_np* (non-penalty goal conversion ratio). These four variables look like an accurate set of metrics for describing the performance of a forward.

Finally, to obtain the performance index of a player, the describing variables are aggregated and weighted by the resulting *PC_Weight* vector of weights.

The resulting distribution of the calculated indexes can be assessed in figure 4.2.

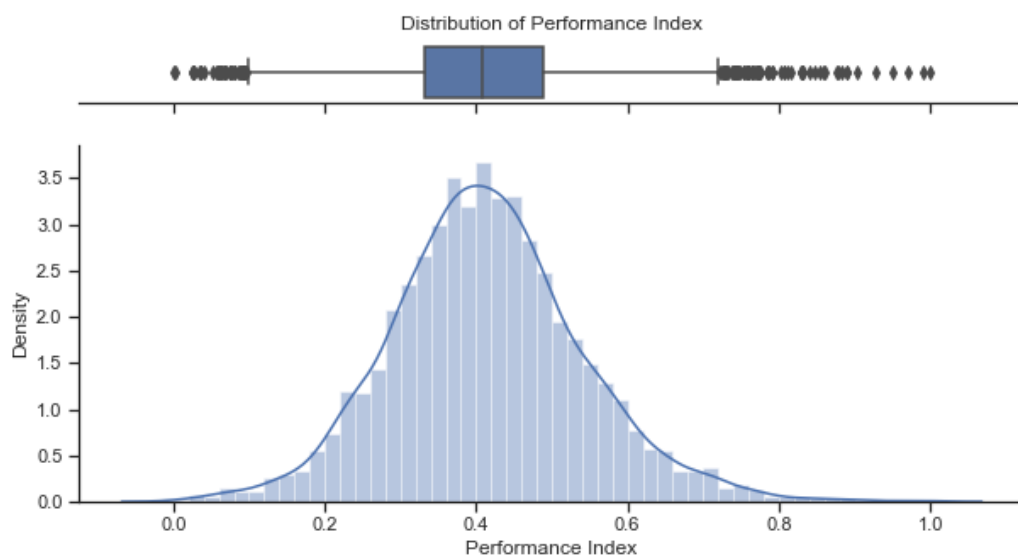


Figure 4.2: Distribution of Performance Index of forwards

Based on the boxplot it would seem that players with a performance index above 0.75 are outliers, namely outstanding forwards. The equivalent could be stated for players with an index below 0.15 as being the exceptionally bad forwards.

In order to assess these results, it was used as a benchmark the average player's rating for the season retrieved from the API. These ratings refer to the qualification given by the official broadcaster of the league, and are based more on a qualitative rather than a quantitative assessment. Although this metric is not available for all players in all matches (circa 25% of null values), the judgment of experts in the

domain, albeit subjective, constitutes a reliable benchmark and helps gain confidence in the results obtained. If the developed performance index shows a strong correlation with these ratings, it would be possible to extend the scoring to all players in which the API's rating is not available, and more importantly to assess players in a systematic and quantitative way.

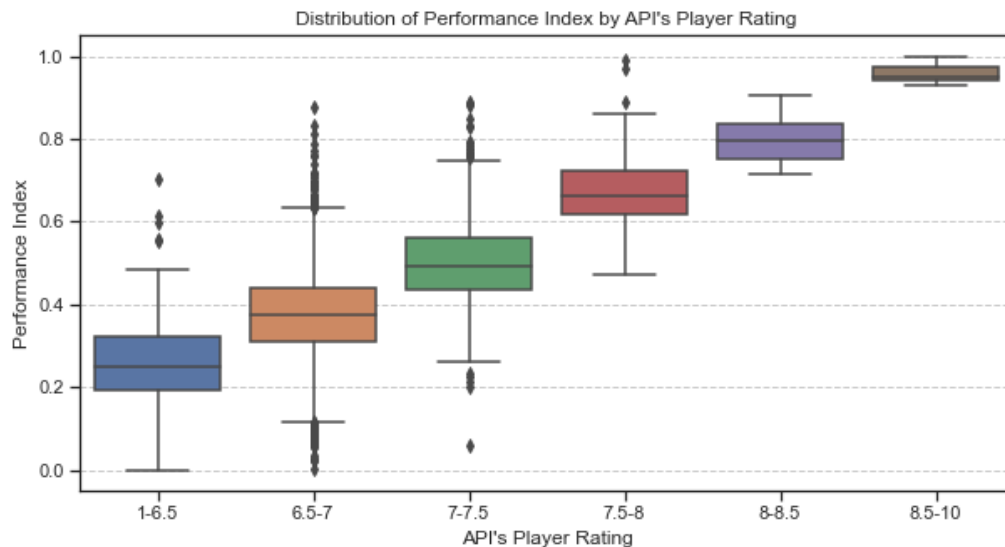


Figure 4.3: Comparison of distributions of Performance Index vs API's Player Rating

It looks like both the calculated performance index and the player's rating follow the same trend. This is more evident and conclusive for higher ratings.

The results of the sensitivity analysis illustrated in Figure 4.4 indicate that the variables that mostly affect the resulting index are *interceptions*, *key passes*, *tackles* and *assists*. This is an interesting insight, as one would probably expect a greater relevance of metrics such as *goals* and *goal conversion rate* for a forward. However, though these are important and also have a high impact in the index, the former are in line with what modern forwards are required nowadays and with what makes a difference in the quality of a player. The ability to actively contribute to the non-possession phase and to the scoring contribution is becoming more and more required for forwards, especially in European leagues.

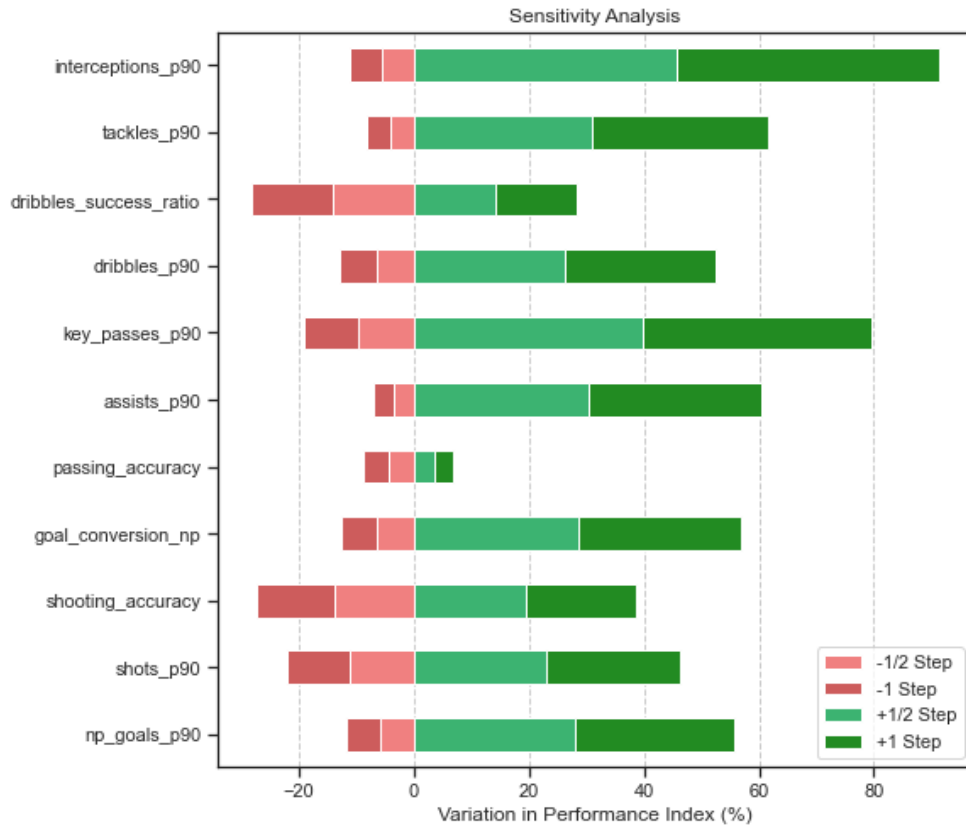


Figure 4.4: Sensitivity analysis for forwards' performance index

The variables that move the index down are *dribble success ratio*, *shooting accuracy* and *shots*.

4.1.2 Midfielders

In the case of the midfielders, both the scree plot in Figure 4.5 and the Kaiser criterion would suggest retaining the first four factors.

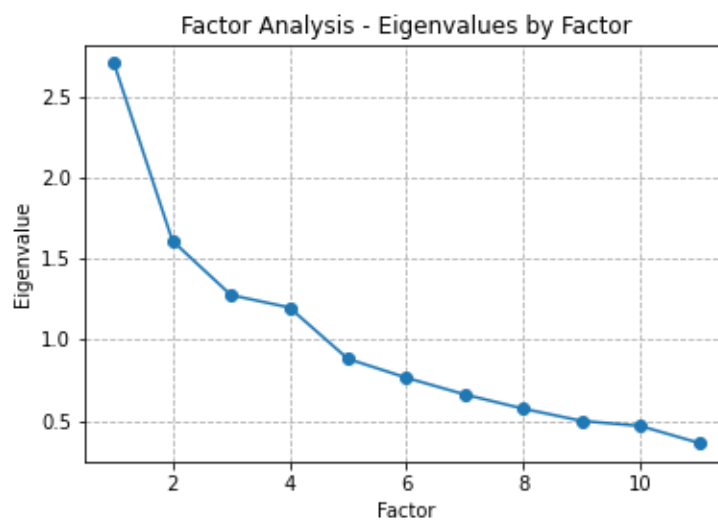


Figure 4.5: Scree plot for midfielders

However, following the OECD’s recommended standard practice would suggest retaining the first five factors, accounting for 70% of the variance.

	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Communalities	Sq Norm Factor 1	Sq Norm Factor 2	Sq Norm Factor 3	Sq Norm Factor 4	Sq Norm Factor 5	PC_Weight
passes_p90	0.03	0.50	-0.15	-0.29	0.57	0.68	0.00	0.15	0.02	0.06	0.26	0.09
passing_accuracy	-0.00	0.14	0.88	0.01	0.02	0.79	0.00	0.01	0.56	0.00	0.00	0.10
key_passes_p90	0.86	0.01	0.09	0.05	0.08	0.75	0.36	0.00	0.01	0.00	0.01	0.10
scoring_contribution	0.79	0.02	0.05	0.07	-0.13	0.65	0.30	0.00	0.00	0.00	0.01	0.08
dribbles_p90	0.54	-0.11	-0.07	0.53	-0.09	0.60	0.14	0.01	0.00	0.21	0.01	0.08
dribbles_success_ratio	-0.10	0.01	0.03	0.05	0.92	0.87	0.01	0.00	0.00	0.00	0.68	0.11
fouls_drawn_p90	0.16	-0.05	0.04	0.83	0.01	0.73	0.01	0.00	0.00	0.51	0.00	0.09
fouls_committed_p90	0.35	-0.58	-0.05	-0.46	0.07	0.67	0.06	0.20	0.00	0.16	0.00	0.09
dribbles_past_p90	-0.09	-0.80	0.09	0.06	-0.07	0.67	0.00	0.39	0.01	0.00	0.00	0.09
tackles_p90	-0.10	0.41	-0.73	-0.02	0.06	0.71	0.01	0.10	0.39	0.00	0.00	0.09
interceptions_p90	-0.49	0.46	0.13	-0.25	0.18	0.56	0.12	0.13	0.01	0.05	0.03	0.07
Variance	2.06	1.64	1.36	1.35	1.25	-	-	-	-	-	-	-
Proportional Variance (%)	0.19	0.15	0.12	0.12	0.11	-	-	-	-	-	-	-
Cummulative (%)	0.19	0.34	0.46	0.58	0.70	-	-	-	-	-	-	-
Expl.Var/Tot (%)	0.27	0.21	0.18	0.18	0.16	-	-	-	-	-	-	-

Table 4.2: Factor analysis for midfielders

In this case, the final vector of weights obtained indicates that the most relevant variables for summarizing the performance are *dribble_success_ratio*, *key_passes_p90* and *passing_accuracy*. These three variables make sense as being the most relevant ones for describing a midfielder's game. All other variables are very close in terms of their weights. This is probably because classifying midfielders in a single category is very broad, as there are more defensive and more offensive players for which their attributes and skills could vary greatly.

The resulting distribution of the calculated indexes can be assessed in figure 4.6.

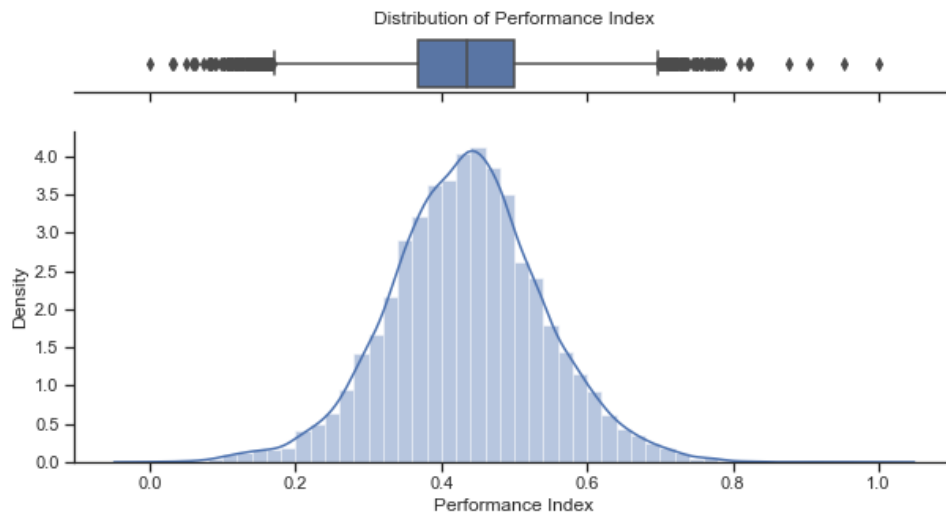


Figure 4.6: Distribution of Performance Index of midfielders

Based on the boxplot it would seem that players with a performance index above 0.75 are outliers, namely outstanding midfielders. The equivalent could be stated for players with an index below 0.15 as being the exceptionally bad midfielders.

In order to assess these results, it was also used as a benchmark the average player's rating for the season retrieved from the API.

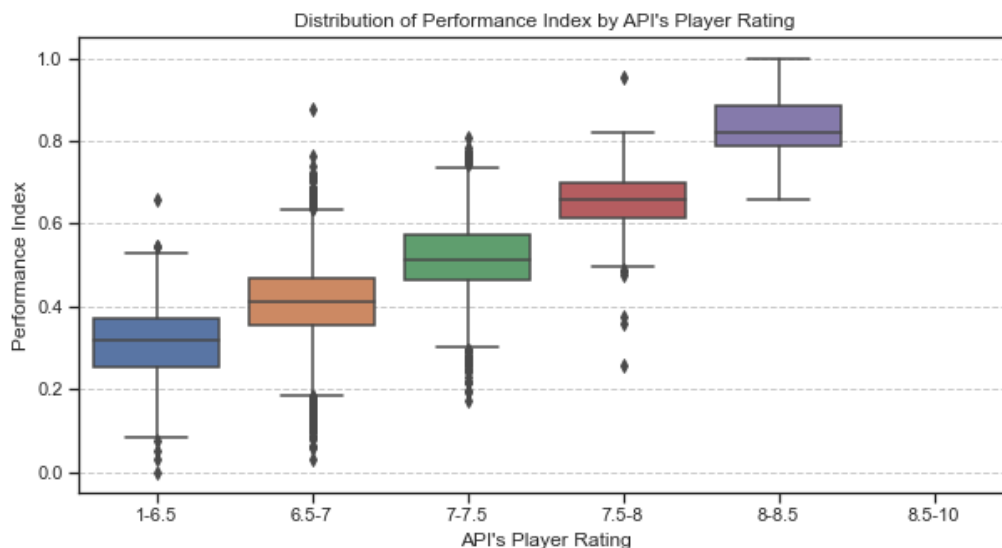


Figure 4.7: Comparison of distributions of Performance Index vs API's Player Rating

It looks like both the calculated performance index and the player's rating follow the same trend. This is more evident and conclusive for higher ratings.

The results of the sensitivity analysis illustrated in Figure 4.8 indicate that the variables that mostly affect positively the resulting index are the *scoring_contribution*

and *key_passes*, followed by *fouls_drawn_p90* and *interceptions_p90*. The midfielder position is a very broad one, as it can go from a defensive midfielder to an offensive one, with all the different shades in the middle. Although they all share some traits in common, their function in the football pitch is very different, and therefore the variables that will affect their performance will be different. Unfortunately, it is not possible in this dataset to determine the sub-role of each player. However, the defensive variables (like *interceptions_p90*) that affect the Performance Index are probably more related to the defensive midfielders, while the attacking ones (like *scoring_contribution* and *key_passes_p90*) are more related to attacking midfielders.

On the other hand, *fouls_committed_p90* and *dribbles_past_p90* are the two most relevant for moving the index down. These two variables are more associated with a defensive aspect of the game.

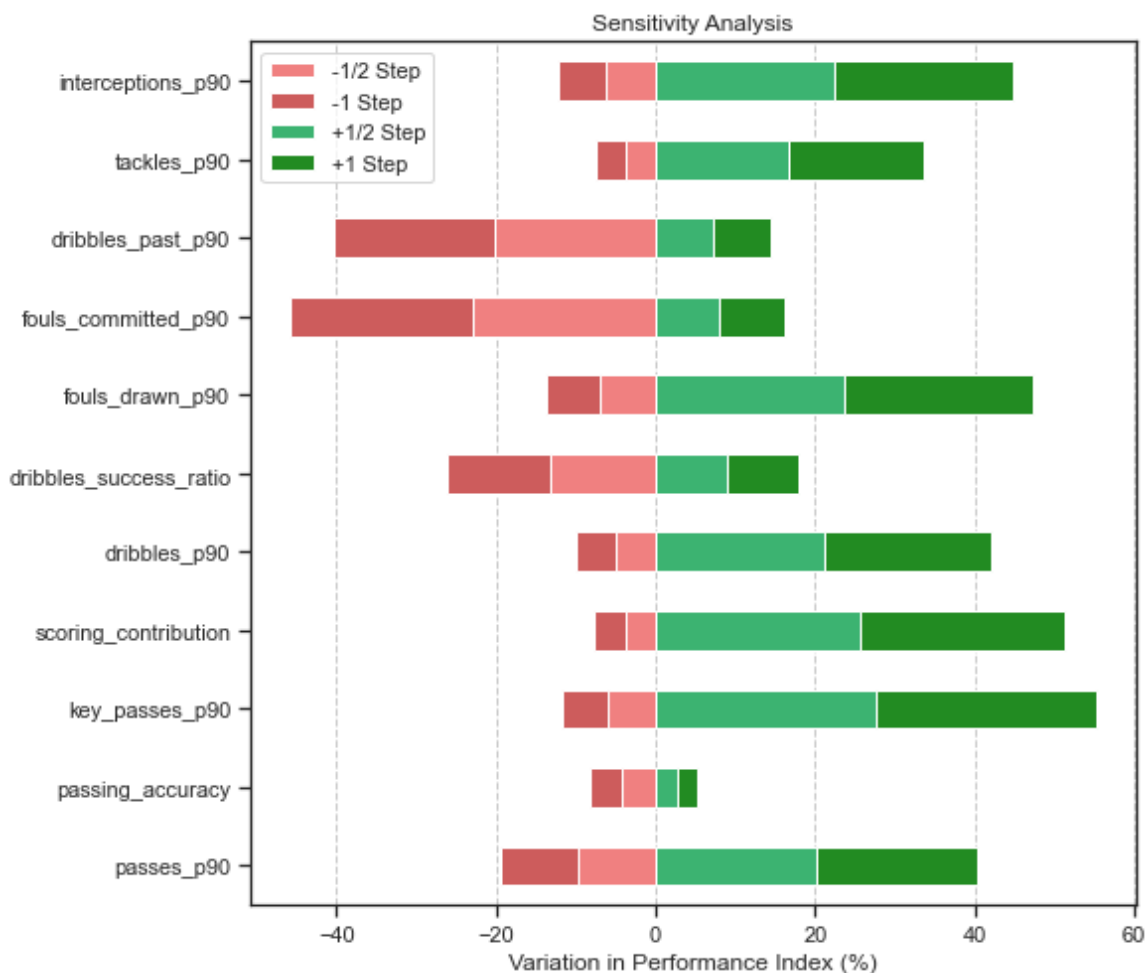


Figure 4.8: Sensitivity analysis for midfielders' performance index

4.1.3 Defenders

In the case of the defenders, the Kaiser criterion would suggest retaining the first four or five factors, whereas the scree plot is not conclusive as the curve does not level off after a specific eigenvalue.

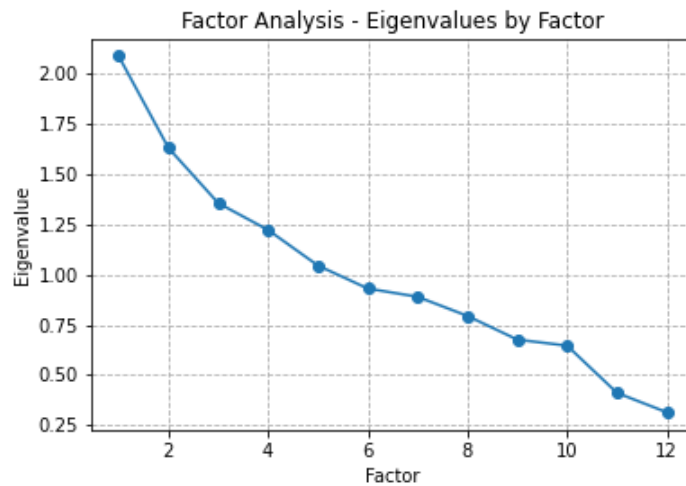


Figure 4.9: Scree plot for defenders

Again, following OECD's standard practice, five factors were retained, accounting for 61% of the variance.

	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Communalities	Sq Norm Factor 1	Sq Norm Factor 2	Sq Norm Factor 3	Sq Norm Factor 4	Sq Norm Factor 5	PC_Weight
passes_p90	0.64	0.26	0.31	0.18	0.01	0.61	0.31	0.05	0.06	0.02	0.00	0.08
passing_accuracy	0.00	-0.83	-0.22	0.11	0.06	0.76	0.00	0.48	0.03	0.01	0.00	0.10
fouls_drawn_p90	-0.48	0.14	0.08	0.23	0.47	0.53	0.17	0.01	0.00	0.04	0.15	0.07
fouls_committed_p90	0.10	-0.02	-0.04	-0.06	-0.82	0.70	0.01	0.00	0.00	0.00	0.46	0.09
dribbles_past_adj	0.10	-0.01	0.08	0.75	0.10	0.59	0.01	0.00	0.00	0.41	0.01	0.08
scoring_contribution	-0.09	0.02	-0.03	0.64	-0.00	0.42	0.01	0.00	0.00	0.30	0.00	0.06
duels_p90	-0.12	0.12	0.76	0.20	0.33	0.76	0.01	0.01	0.33	0.03	0.08	0.10
duels_success_ratio	-0.07	0.04	0.82	-0.10	-0.20	0.72	0.00	0.00	0.39	0.01	0.03	0.10
tackles_p90	0.32	0.67	-0.17	0.24	0.26	0.71	0.07	0.31	0.02	0.04	0.05	0.10
blocks_p90	0.69	0.01	-0.27	-0.12	-0.03	0.57	0.36	0.00	0.04	0.01	0.00	0.08
interceptions_p90	0.18	-0.43	0.47	0.01	0.36	0.56	0.03	0.13	0.13	0.00	0.09	0.08
penalty_committed_p90	-0.21	0.04	0.03	0.41	-0.45	0.42	0.03	0.00	0.00	0.12	0.14	0.06
Variance	1.34	1.43	1.72	1.38	1.47	-	-	-	-	-	-	-
Proportional Variance (%)	0.11	0.12	0.14	0.11	0.12	-	-	-	-	-	-	-
Cumulative (%)	0.11	0.23	0.37	0.49	0.61	-	-	-	-	-	-	-
Expl.Var/Tot (%)	0.18	0.20	0.23	0.19	0.20	-	-	-	-	-	-	-

Table 4.3: Factor analysis for defenders

The final vector of weights obtained indicates that the most relevant variables for summarizing the performance of defenders are *duels_p90*, *duels_success_ratio*, *tackles_p90*, and *fouls_committed_p90*. These four variables make sense as being the

most relevant ones for describing a defender's game, as they all represent defensive aspects.

The resulting distribution of the calculated indexes can be assessed in figure 4.10.

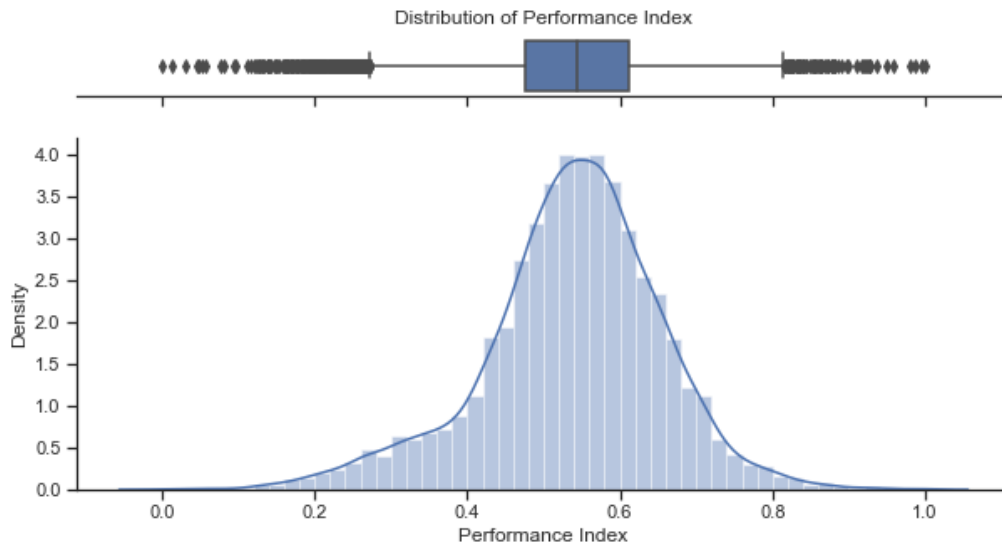


Figure 4.10: Distribution of Performance Index of defenders

For the case of defenders, it looks like the distribution is much narrower than for forwards or midfielders. This suggests that it is difficult for a player to stand out. In fact, the longer tail at the left of the distribution would suggest that it is easier to stand out from the rest due to a bad performance rather than to a good one.

Based on the boxplot it would seem that players with a performance index above 0.82 are outliers, namely outstanding defenders. The equivalent could be stated for players with an index below 0.45 as being the exceptionally bad defenders.

In order to assess these results, it was also used as a benchmark the average player's rating for the season retrieved from the API.

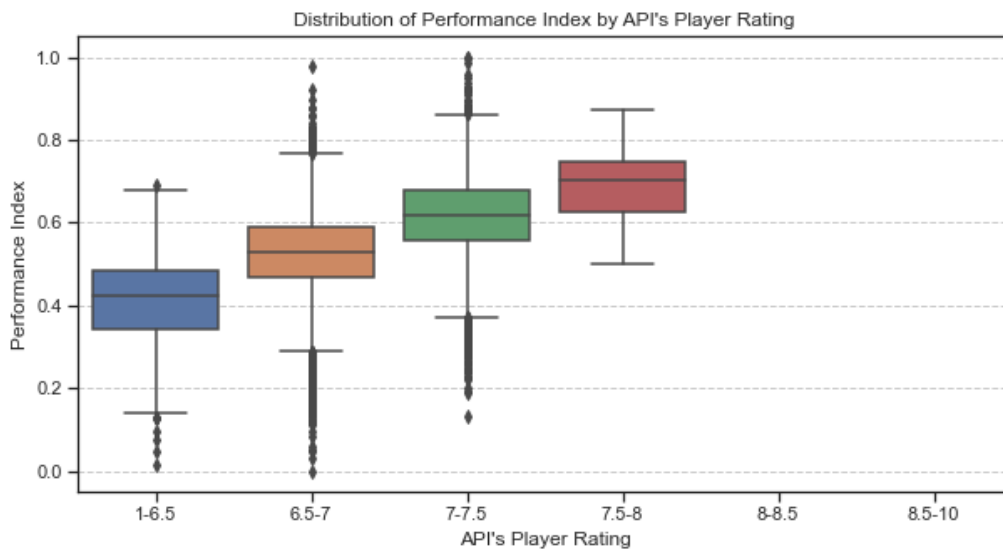


Figure 4.11: Comparison of distributions of Performance Index vs API's Player Rating

In line with the leptokurtic distribution previously observed in the histogram, the rating available for the players in the API is at maximum 8 points out of 10, meaning that the performance of defenders is much more stable than for the previous roles. There seems to be a good correlation between the constructed performance index and the different players' ratings.

The results of the sensitivity analysis illustrated in figure 4.12 indicate that *penalty_committed_p90*, *dribbles_past_p90* and *fouls_committed_p90* are the variables that mostly negatively affect the resulting index. In contrast, *scoring_contribution* is the most relevant one for moving the index up. This makes sense as the former are aspects that greatly affect the team (and player) results when done poorly (committing fouls, penalties, or not being able to stop attackers), whereas the latter makes a defender stand out from the rest for being able to score goals and give key passes.

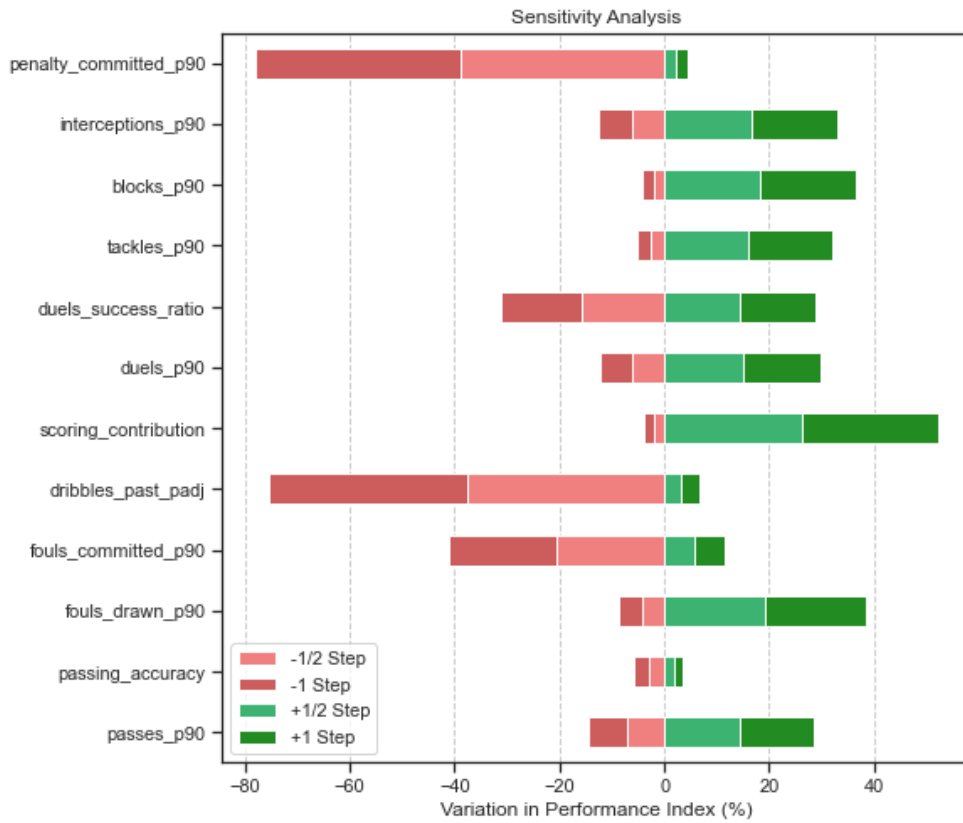


Figure 4.12: Sensitivity analysis for defenders' performance index

4.1.4 Goalkeepers

In the case of the goalkeepers, the Kaiser criterion would suggest retaining the first five or six factors, whereas the scree plot is not conclusive as the curve does not level off after a specific eigenvalue.

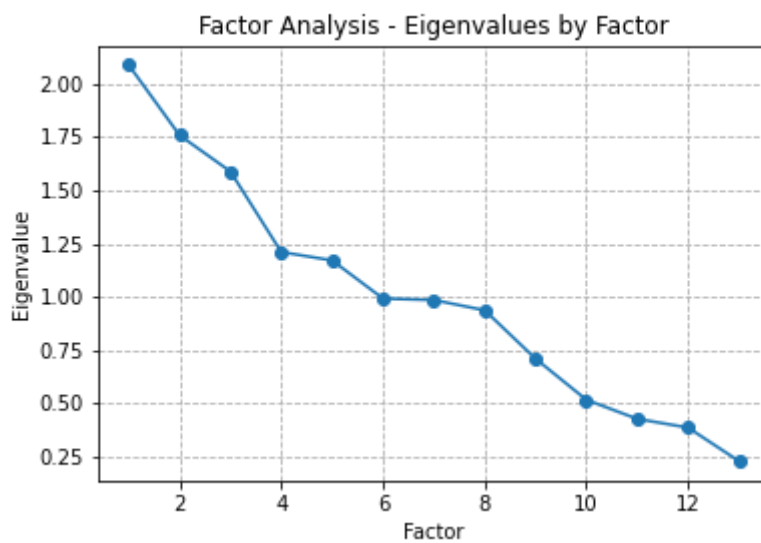


Figure 4.13: Scree plot for goalkeepers

Following OECD's standard practice, five factors were retained, accounting for 60% of the variance.

	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Communalities	Sq Norm Factor 1	Sq Norm Factor 2	Sq Norm Factor 3	Sq Norm Factor 4	Sq Norm Factor 5	PC_Weight
saves_p90	-0.04	-0.00	0.02	0.04	0.95	0.91	0.00	0.00	0.00	0.00	0.77	0.12
goals_conceded_p90	0.16	0.11	-0.88	0.12	-0.20	0.86	0.01	0.01	0.51	0.01	0.03	0.11
goals_conceded_ratio	0.33	-0.05	0.80	0.03	-0.34	0.87	0.06	0.00	0.42	0.00	0.10	0.11
passes_p90	-0.83	-0.06	-0.01	-0.07	0.11	0.71	0.38	0.00	0.00	0.00	0.01	0.09
passing_accuracy	0.82	-0.04	0.18	0.15	0.17	0.75	0.37	0.00	0.02	0.01	0.02	0.10
fouls_drawn_p90	0.13	0.69	-0.04	-0.13	0.06	0.51	0.01	0.27	0.00	0.01	0.00	0.07
fouls_committed_p90	-0.11	-0.04	-0.08	0.86	-0.07	0.76	0.01	0.00	0.00	0.49	0.00	0.10
duels_p90	-0.08	0.83	0.11	-0.17	0.09	0.75	0.00	0.39	0.01	0.02	0.01	0.10
duels_success_ratio	-0.12	0.74	-0.03	0.26	-0.10	0.65	0.01	0.31	0.00	0.04	0.01	0.08
tackles_p90	-0.34	0.11	0.12	0.01	0.06	0.14	0.06	0.01	0.01	0.00	0.00	0.02
blocks_p90	0.04	-0.13	-0.19	-0.04	-0.01	0.06	0.00	0.01	0.02	0.00	0.00	0.01
interceptions_p90	0.24	0.14	-0.03	-0.11	0.20	0.13	0.03	0.01	0.00	0.01	0.03	0.02
penalty_committed_p90	0.29	-0.02	0.08	0.78	0.08	0.70	0.05	0.00	0.00	0.40	0.01	0.09
Variance	1.80	1.80	1.52	1.51	1.18	-	-	-	-	-	-	-
Proportional Variance (%)	0.14	0.14	0.12	0.12	0.09	-	-	-	-	-	-	-
Cummulative (%)	0.14	0.28	0.39	0.51	0.60	-	-	-	-	-	-	-
Expl.Var/Tot (%)	0.23	0.23	0.19	0.19	0.15	-	-	-	-	-	-	-

Table 4.4: Factor analysis for goalkeepers

The final vector of weights obtained indicates that the most relevant variables for summarizing the performance of defenders are *saves_p90*, *goals_conceded_p90* and *goals_conceded_ratio*. These three variables make sense as being the most relevant ones for describing a goalkeepers' game.

The resulting distribution of the calculated indexes can be assessed in figure 4.14.

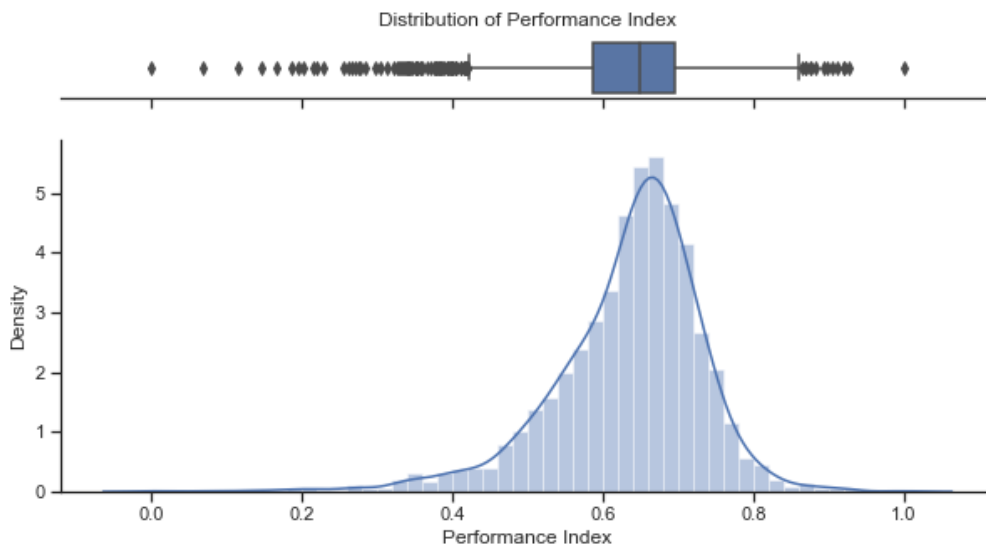


Figure 4.14: Distribution of Performance Index of goalkeepers

The distribution of the performance index for goalkeepers is even narrower than for defenders. In this case, however, there seems to be more outstanding performances, both good and bad. It would seem that players with a performance index above 0.82 are outliers, namely outstanding goalkeepers. The equivalent could be stated for players with an index below 0.42 as being the exceptionally bad goalkeepers.

In order to assess these results, it was also used as a benchmark the average player's rating for the season retrieved from the API.

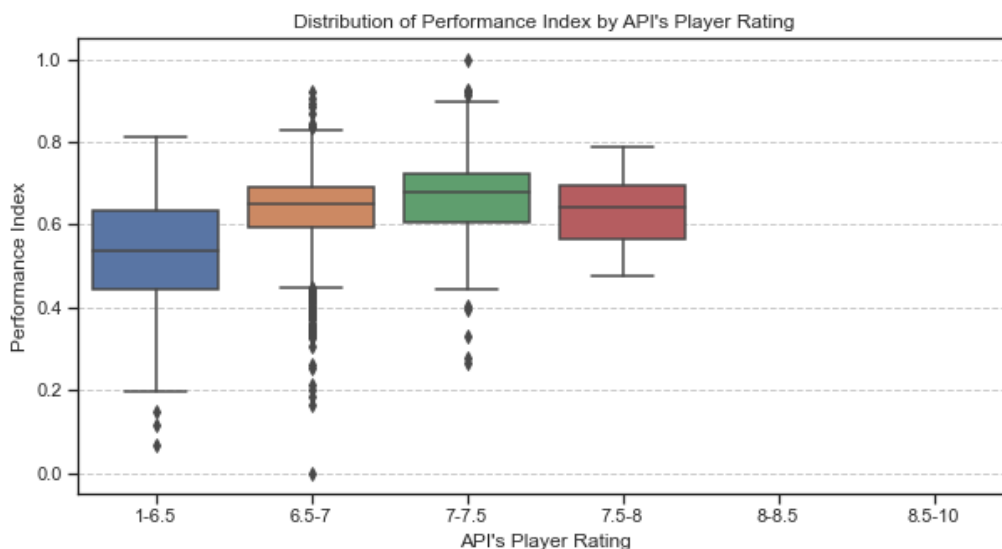


Figure 4.15: Comparison of distributions of Performance Index vs API's Player Rating

Similarly to the defenders' case, the rating available for the players in the API is at maximum 8 points out of 10, meaning that the performance of goalkeepers is much more stable than for the previous roles. However, contrary to defenders, it would seem that the constructed performance index does not discriminate very well between the different players' ratings, as the boxplots are somewhat overlapping. The variables available in the API to describe a goalkeeper's game are very few to be able to discriminate effectively their performances. Top platforms like StatsPerform contemplate other metrics like *exits*, *aerial duels*, and even state-of-the-art ones like *expected goals on target* (xGOT) [27], which help to better describe the performance of a goalkeeper.

According to the sensitivity analysis illustrated in figure 4.16, *fouls_committed_p90* and *goals_conceded_p90* are the two variables that mostly negatively affect the resulting index. These two metrics make sense as being the ones that move the index down. In contrast, *tackles_p90* and *interceptions_p90* are the two most relevant for moving the index up.

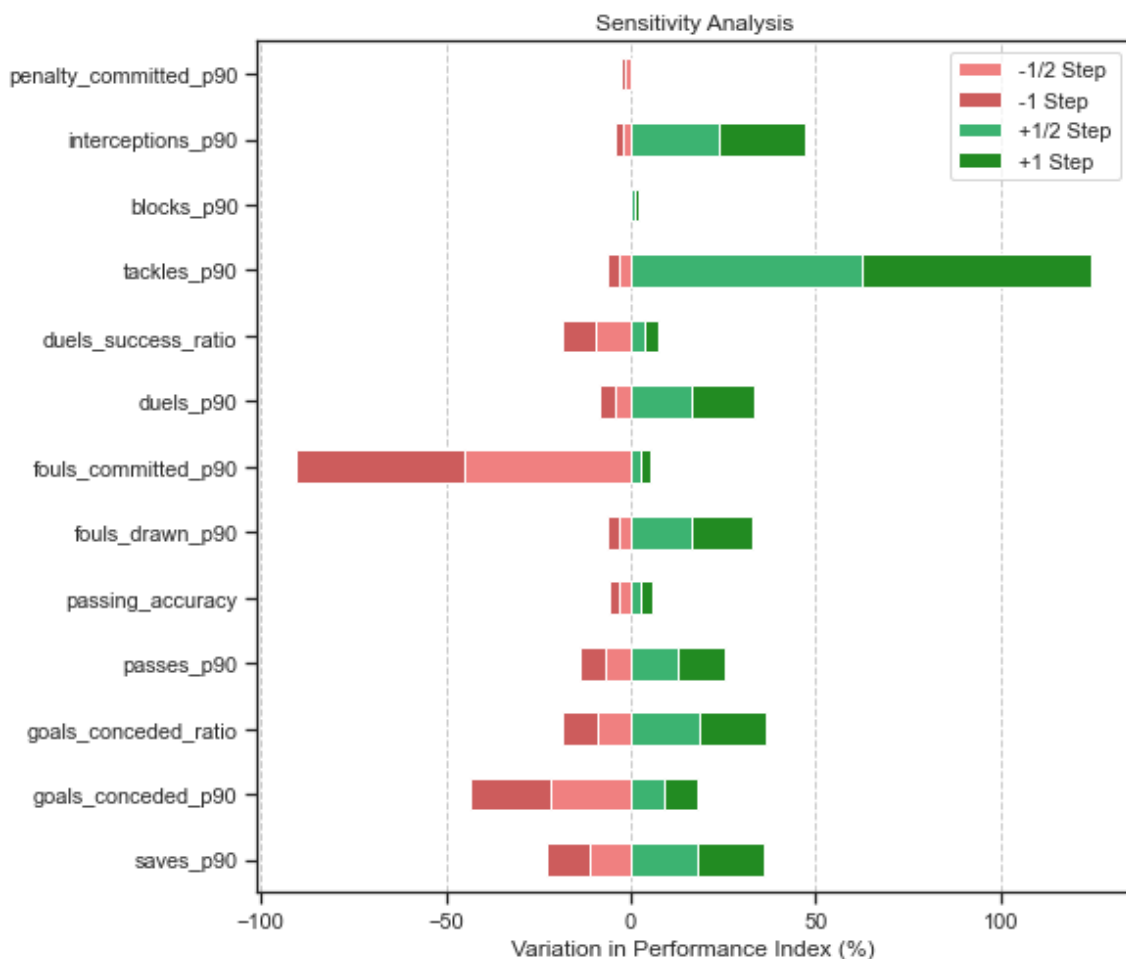


Figure 4.16: Sensitivity analysis for goalkeepers' performance index

4.1.5 General results

The robustness of the constructed performance index for each position can be assessed by looking at three elements:

- Distribution of the resulting Performance Index (Histogram)
- Comparison to player ratings extracted from the API (Boxplot)
- Sensitivity Analysis

The histogram of the calculated performance index gives an intuition of the shape of the metric's distribution, showing if it is normally distributed, if it is bimodal, if it is skewed towards one tail, etc. The fact that the histograms for all positions follow a bell-shaped curve suggests that the calculated index is normally distributed, with no apparent abnormal behavior. In the case of defenders and goalkeepers, it can be seen that the shape is more leptokurtic, indicating that the performance of the players in these positions is more similar with each other, and that is difficult to stand out from the rest. It can also mean that there are variables missing that could help discriminate better the nuances in the performance.

The boxplots comparing the distribution of the performance index for each interval of the API's player rating is probably one of the strongest elements to assess the robustness of the calculated metric. Even though the ratings are not available for all players, the fact that the quantitative approach developed in this project correlates with the judgment of subject experts suggests that the analytical process is correctly capturing the latent aspects of the game and weighting them appropriately. This is true especially for forwards, midfielders and, to a lesser extent, defenders. For the goalkeepers it would seem that a better job could be done by adding more specific variables that provide a deeper discrimination of the performance.

Finally, the sensitivity analysis helps to gauge the robustness of the calculated index and to increase its transparency, by evaluating how it depends upon the information fed into it. While forwards and midfielders do not rely heavily on one or a few specific variables, it can be seen that for defenders and goalkeepers the performance index depends greatly on just two or three. In the case of defenders, only *penalty_committed_p90* or *dribbles_past_p90* can drive the index down as much as 75% to 80%. For goalkeepers, *tackles_p90* can move the index up in over a 100%, while *fouls_committed_p90* can do the opposite also in almost 100%. This behavior indicates that the index could be unstable and experience considerable variations when these variables change. Even though these variables are important to the game and their impact on the index makes sense, the lack of other relevant metrics (*exits*, *aerial duels*, *expected goals on target*, etc.) are probably making the calculated index rely too much on the aforementioned ones.

All in all, it would seem that the calculated performance index is more robust for forwards and midfielders, while it could be improved for defenders and especially for

goalkeepers. It is fair to note, though, that there are more variables available in the dataset for describing the performance of the former positions than the latter. The inclusion of metrics better suited for describing defenders' and goalkeepers' game would probably improve the robustness of the index.

4.2 Football Analytics Web Application

The main goal of this project is to build a tool that helps scouts optimize the process of finding and assessing players, reducing time and costs involved. Having built the analysis pipeline with solid statistical foundations, with which every player can be assessed and its performance index calculated, the next step is to present the information in a user-friendly way. Two views were built in order to achieve this goal:

- **App Scouting:** view with all the players of Latin America, in which a scout can tailor the search by means of different filters.
- **App Players:** view by individual player presenting in-depth statistics about their game by season played, as well as its evolution.

The focus has been set on creating a robust performance index and its presentation through a working prototype. Therefore, minimum attention was set on UX/UI and frontend development, as it would have exceeded the scope.

4.2.1 App Scouting

The scouting view has the objective of presenting all available players to the scout, sorted in descending order by their performance index, and the possibility of filtering the database dynamically by different variables such as league, season, position in the pitch, and performance index, among others. An example can be seen in Figure 4.17, in which midfielders with a performance index between 0 and 45 for season 2018/19 in Argentina have been filtered.



Figure 4.17: Scouting view. Example of filtering.

The resulting table is the first step into spotting potential interesting players that may be worth looking at their in-depth statistics and eventually invite to a trial for evaluation. As an example, based on this list a scout in 2017 who is looking for a player for the next season, might choose to perform a deeper analysis on Guillermo Fernandez from Godoy Cruz, whose performance index is of 43. A low-scored player with potential to grow is a great prospect as an investment, as the club can profit from paying a low value and then selling the player at a higher price.

4.2.2 App Player

The player view shows all available statistics for a particular player. The scout can filter players for every team in a particular league and season.

For each player, the first table shows general information about the player, such as its position, minutes played in the season, average rating, goals and assists. Accompanying this table is the evolution of the player's performance index. This plot, along with the radar chart, should be the two main pieces of information a scout uses to assess a player.

Finally, three other sections describing a player's skills are presented: Attack, Build-up and Defense. Each of these shows a table with all available statistics for the season, along with the evolution of the main ones. These sections complement the evolution of the player's performance, completing the player's profile for a detailed evaluation.

Following the example presented in the App Scouting, Figure 4.18 shows the player view of Guillermo Fernandez for the season 2017/18 in Godoy Cruz (Argentina). Standing in 2017/18, the player was presenting a performance with a growth of 30% in the overall performance with respect to the previous season. This could be an indication of a promising player, depending of course on its market value and age.

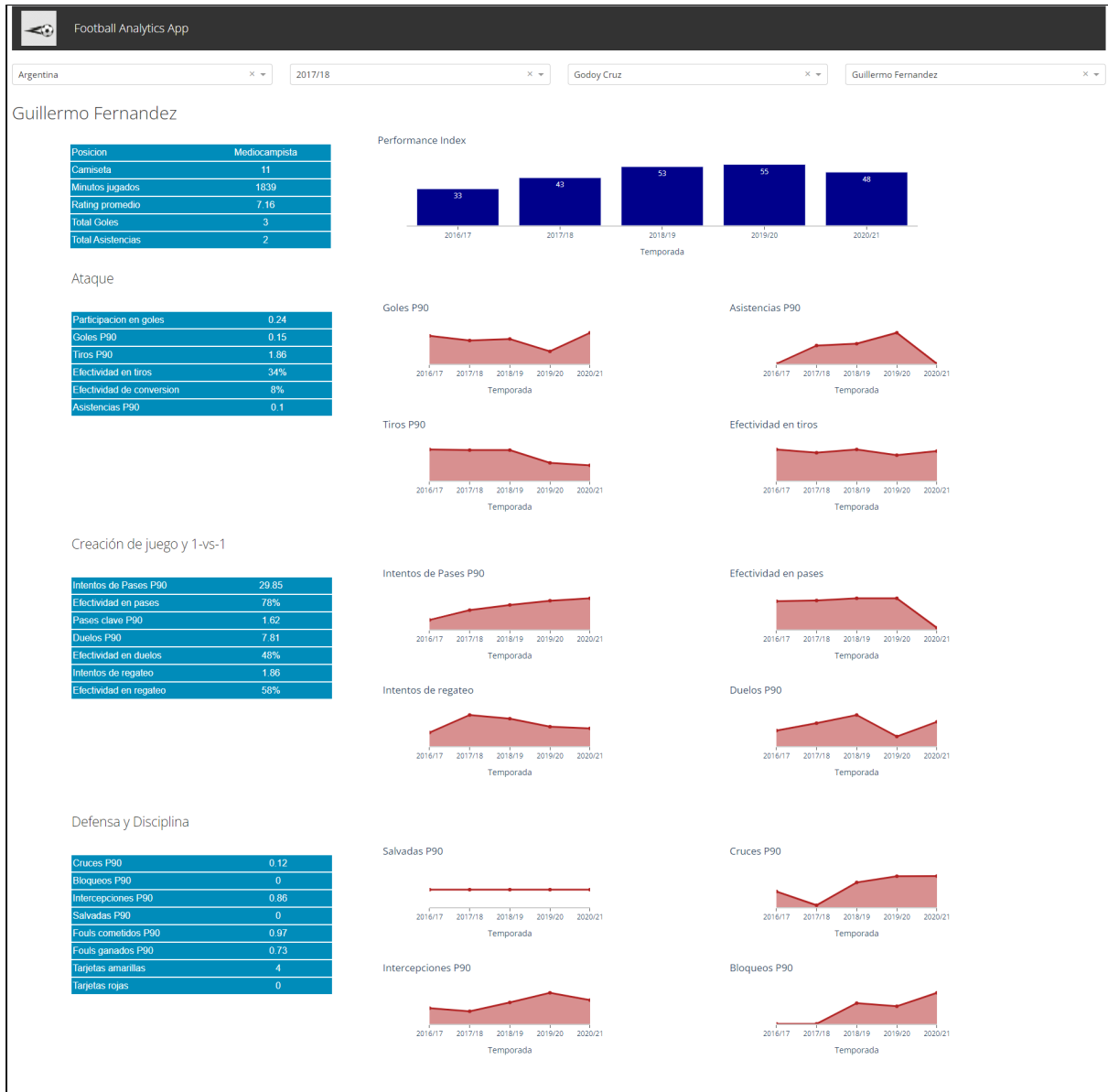


Figure 4.18: App Player. Example of stats for Guillermo Fernandez profile

Unfortunately the API does not provide neither the players' market value nor the age. Therefore, for this example the information was retrieved from the leading platform Transfermarkt. In 2018, Guillermo Fernandez was 27 years old and worth 1.3 million euros. Racing Club, also from Argentina, spotted the opportunity and bought his rights. After a year, Fernandez continued to improve his performance to an index of 53 in season 2018/19.

As a result of this improvement, Racing Club was able to sell Fernandez to Cruz Azul (Mexico) for a value of 4.43 million euros, thus obtaining an outstanding return over the investment of 3.41 (+241%).

In a very productive transfer market, Racing Club had also made a very similar investment for defender Renzo Saravia, a player from Belgrano (Argentina) bought

for 1.15 million euros after season 2016/17 and later sold in 2018/19 to Porto (Portugal) for 5.5 million euros, obtaining a massive ROI of 4.78 (+378%).

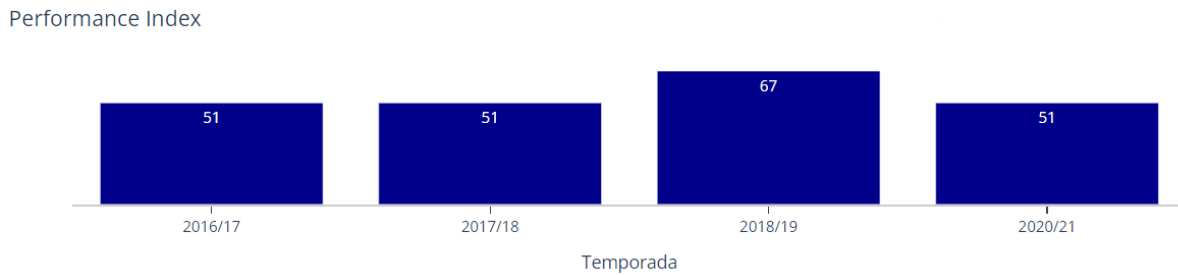


Figure 4.19: Renzo Saravia's performance improvement from season 2016/17 to 2018/19, period in which he played for Racing Club.

On the other end is Club America (Mexico), a club that could have benefited from the tool in the transfer market of 2018, when they bought the Chilean forward Nicolas Castillo from Benfica (Portugal) for the vast amount of 7.13 million euros, becoming the best-paid player in the history of the Mexican league. Castillo had been playing in Europe since season 2016/17, after winning Copa America with his national team and being two times champion of the Chilean tournament with Universidad de Chile, having him as the maximum scorer of the championship in both opportunities.

Castillo's track record indicated that this transfer was going to be a big success for Club America. However, the forward's performance was far from living up to expectations with just 9 goals in 26 matches. He lost continuity in the team and eventually was loaned to Juventude (Brazil) until his contract was over at the beginning of 2022. Even with a much lower market value of 1.4 million euros, Club America was unable to sell Castillo, constituting one of the worst investments of the club and the league.

If we take a look at Nicolas Castillo's profile in the scouting tool (Figure 4.20), it can be seen that his performance had been dropping over the past seasons before Club America bought him. The overall performance over the years indicated that this acquisition might not have been that fruitful. Had the team had this tool back then, they might have reevaluated their decision, or at least negotiated a better contract rather than the best-paid one in Mexican history. The money lost in this sole investment is already sufficient to justify the use of a smart scouting tool as the one developed in this project.

Posicion	Delantero
Camiseta	15
Minutos jugados	1236
Rating promedio	7
Total Goles	5
Total Asistencias	2

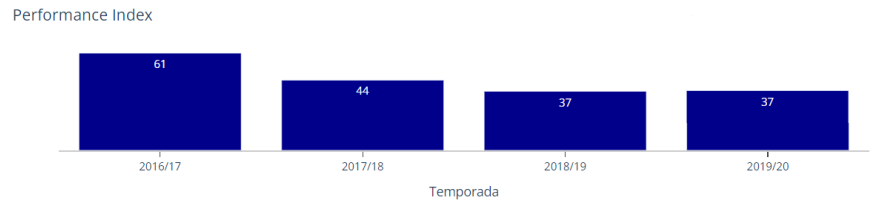


Figure 4.20: Nicolas Castillo’s performance evolution over time

These stories are just some examples of what can be achieved by combining a scout's expertise with a smart tool. By being able to slice and dice the data according to the needs of the team, the scout can speed up the process of searching candidates for trials. In addition, it also reduces the unnecessary trips made to evaluate players that could have been discarded by previously assessing them in the tool, thus optimizing costs.

Chapter 5

Conclusions

5.1 Achievements

Throughout the current project a data-driven scouting tool was developed to search and evaluate football players in Latin America, reducing time and costs involved in the process. This was achieved by designing, developing and implementing all stages of the pipeline, from the extraction of the data and storage in a datawarehouse, to the statistical analysis and graphical representation in a user-friendly web application.

A data warehouse's logical and conceptual designs were created based on the available data in the API. Data was extracted from 14 endpoints of the API and processed with a custom EtLT pipeline that fed the different tables of the data model. Finally, an Analytical Base Table was created with all necessary features to describe a player's performance in a given league and season.

Through descriptive exploration and factor analysis the performance of football players was summarized in one single index, making it easier to compare between them, regardless of their team, league, and role in the football pitch. In addition, sub-indexes were created to have a more detailed assessment. By comparing a player's profile in terms of these stats to an average player in the same position, a scout can have an instant overview of the player's skills.

A web application with two views was created in order to optimize the scouting funnel. Firstly, the scout is able to scan the entire database of 6,000 players from Argentina, Brazil, Chile, Peru, Colombia and Mexico. By means of several dynamic filters, the scout can tailor the search to the team's specific needs. Secondly, once players have been shortlisted, the scout can evaluate each one of them more thoroughly in a dedicated individual view containing all relevant stats and comparisons.

Finally, three concrete use cases were presented that demonstrate the strengths of the tool and how it can be used to make better decisions in the scouting process.

5.2 Limitations and potential improvements

There are some limitations that were encountered when working on this project.

To begin with, even though the data provided by the API is rich enough to develop an index, it is not as comprehensive as what other data feed providers such as Opta and StatsBomb offer. Basic features like age, height, and weight would be useful to have a better understanding of a player's profile. Moreover, these suppliers offer more advanced statistics such as expected goals and expected assists, which are the

state-of-the-art metrics used to evaluate a player's performance [15]. These stats, along with other dozens of extra features that they provide, would constitute an important improvement in the elaboration of the performance index.

Finally, being able to retrieve the market value of the players in each season would be a great way to validate the calculated performance index by looking for correlations.

Appendix A

Samples for each of the API's endpoints used in the project

A.1 Endpoint: Leagues

Data retrieved: leagues and its corresponding seasons

```
{'get': 'leagues',
 'parameters': {'id': '128'},
 'errors': [],
 'results': 1,
 'paging': {'current': 1, 'total': 1},
 'response': [{'league': {'id': 128,
 'name': 'Primera Division',
 'type': 'League',
 'logo': 'https://media.api-sports.io/football/leagues/128.png'},
 'country': {'name': 'Argentina',
 'code': 'AR',
 'flag': 'https://media.api-sports.io/flags/ar.svg'},
 'seasons': [{ 'year': 2015,
 'start': '2015-02-13',
 'end': '2015-12-07',
 'current': False,
 'coverage': {'fixtures': {'events': True,
 'lineups': True,
 'statistics_fixtures': True,
 'statistics_players': False},
 'standings': True,
 'players': True,
 'top_scorers': True,
 'top_assists': True,
 'top_cards': True,
 'injuries': False,
 'predictions': True,
 'odds': False}},
 { 'year': 2016,
 'start': '2016-08-26',
 'end': '2017-06-27',
 'current': False,
 'coverage': {'fixtures': {'events': True,
 'lineups': True,
 'statistics_fixtures': True,
 'statistics_players': True},
 'standings': True,
 'players': True,
 'top_scorers': True,
 'top_assists': True
```

A.2 Endpoint: Teams

Data retrieved: teams of each league in each season and its corresponding venue.

```
{'get': 'teams',
 'parameters': {'league': '128', 'season': '2021'},
 'errors': [],
 'results': 26,
 'paging': {'current': 1, 'total': 1},
 'response': [{'team': {'id': 434,
  'name': 'Gimnasia L.P.',
  'country': 'Argentina',
  'founded': 1887,
  'national': False,
  'logo': 'https://media.api-sports.io/football/teams/434.png'},
 'venue': {'id': 77,
  'name': 'Estadio Juan Carmelo Zerillo',
  'address': 'Avenida 60 y 118',
  'city': 'La Plata, Provincia de Buenos Aires',
  'capacity': 24544,
  'surface': 'grass',
  'image': 'https://media.api-sports.io/football/venues/77.png'}},
 {'team': {'id': 435,
  'name': 'River Plate',
  'country': 'Argentina',
  'founded': 1901,
  'national': False,
  'logo': 'https://media.api-sports.io/football/teams/435.png'},
 'venue': {'id': 31,
  'name': 'Estadio Monumental Antonio Vespucio Liberti',
  'address': 'Avenida Presidente José Figueroa Alcorta 7597, Núñez',
  'city': 'Capital Federal, Ciudad de Buenos Aires',
  'capacity': 65645,
  'surface': 'grass',
  'image': 'https://media.api-sports.io/football/venues/31.png'}},
 {'team': {'id': 436,
  'name': 'Racing Club',
  'country': 'Argentina',
  'founded': 1903,
  'national': False,
  'logo': 'https://media.api-sports.io/football/teams/436.png'},
 'venue': {'id': 99
```

A.3 Endpoint: Fixtures

Data retrieved: aggregated information related to each match of each season in each league.

```
{'get': 'fixtures',
 'parameters': {'league': '128', 'season': '2021'},
 'errors': [],
 'results': 501,
 'paging': {'current': 1, 'total': 1},
 'response': [{'fixture': {'id': 674018,
  'referee': 'F. Tello',
  'timezone': 'UTC',
  'date': '2021-02-15T00:30:00+00:00',
  'timestamp': 1613349000,
  'periods': {'first': 1613349000, 'second': 1613352600},
  'venue': {'id': None,
  'name': 'Estadio Jorge Luis Hirschi',
  'city': 'La Plata, Provincia de Buenos Aires'},
  'status': {'long': 'Match Finished', 'short': 'FT', 'elapsed': 90}},
 'league': {'id': 128,
  'name': 'Primera Division',
  'country': 'Argentina',
  'logo': 'https://media.api-sports.io/football/leagues/128.png',
  'flag': 'https://media.api-sports.io/flags/ar.svg',
  'season': 2021,
  'round': '1st Phase - 1'},
 'teams': {'home': {'id': 450,
  'name': 'Estudiantes L.P.',
  'logo': 'https://media.api-sports.io/football/teams/450.png',
  'winner': True},
  'away': {'id': 435,
  'name': 'River Plate',
  'logo': 'https://media.api-sports.io/football/teams/435.png',
  'winner': False}},
 'goals': {'home': 2, 'away': 1},
 'score': {'halftime': {'home': 0, 'away': 0},
  'fulltime': {'home': 2, 'away': 1},
  'extratime': {'home': None, 'away': None},
  'penalty': {'home': None, 'away': None}}},
 {'fixture': {'id': 674019,
  'referee': 'Ariel Penel, Argentina',
  'timezone': 'UTC',
  'date': '2021-02-13T20:10:00+00:00',
  'timestamp': 1613247000,
  'periods': {'first': 1613247000, 'second': 1613250600},
  'venue': {'id': 33,
```

A.4 Endpoint: Match events

Data retrieved: events occurred in a match. Event types available: Goals (normal, own goal, penalty, missed penalty), Cards (yellow, red), Substitution, VAR (goal canceled, penalty confirmed).

```
{'get': 'fixtures/events',
 'parameters': {'fixture': '215550'},
 'errors': [],
 'results': 17,
 'paging': {'current': 1, 'total': 1},
 'response': [{'time': {'elapsed': 5, 'extra': None},
  'team': {'id': 460,
  'name': 'San Lorenzo',
  'logo': 'https://media.api-sports.io/football/teams/460.png'},
  'player': {'id': 6613, 'name': 'C. Barrios'},
  'assist': {'id': 6617, 'name': 'A. Díaz'},
  'type': 'Goal',
  'detail': 'Normal Goal',
  'comments': None},
 {'time': {'elapsed': 25, 'extra': None},
  'team': {'id': 439,
  'name': 'Godoy Cruz',
  'logo': 'https://media.api-sports.io/football/teams/439.png'},
  'player': {'id': 5799, 'name': 'Leandro Vella'},
  'assist': {'id': None, 'name': None},
  'type': 'Card',
  'detail': 'Yellow Card',
  'comments': None},
 {'time': {'elapsed': 30, 'extra': None},
  'team': {'id': 439,
  'name': 'Godoy Cruz',
  'logo': 'https://media.api-sports.io/football/teams/439.png'},
  'player': {'id': 6320, 'name': 'Richard Prieto'},
  'assist': {'id': None, 'name': None},
  'type': 'Card',
  'detail': 'Yellow Card',
  'comments': None},
 {'time': {'elapsed': 46, 'extra': None},
  'team': {'id': 439,
  'name': 'Godoy Cruz',
  'logo': 'https://media.api-sports.io/football/teams/439.png'},
  'player': {'id': 6303, 'name': 'M. Rouzies'},
  'assist': {'id': 6295, 'name': 'A. Aleo'},
  'type': 'subst',
  'detail': 'Substitution 1',
  'comments': None},
 {'time': {'elapsed': 46, 'extra': None}
```

A.5 Endpoint: Match Lineups

Data retrieved: teams' lineup for each match (formation, start XI, substitutes).

```
{'get': 'fixtures/lineups',
 'parameters': {'fixture': '215550'},
 'errors': [],
 'results': 2,
 'paging': {'current': 1, 'total': 1},
 'response': [{ 'team': {'id': 460,
  'name': 'San Lorenzo',
  'logo': 'https://media.api-sports.io/football/teams/460.png',
  'colors': None},
  'coach': {'id': 1691,
  'name': 'Pizzi',
  'photo': 'https://media.api-sports.io/football/coachs/1691.png'},
  'formation': '4-2-3-1',
  'startXI': [{ 'player': {'id': 35961,
  'name': 'N. Navarro',
  'number': 22,
  'pos': 'G',
  'grid': '1:1'}},
  { 'player': {'id': 50228,
  'name': 'S. Vergini',
  'number': 27,
  'pos': 'D',
  'grid': '2:4'}},
  { 'player': {'id': 6606,
  'name': 'A. Díaz',
  'number': 33,
  'pos': 'F',
  'grid': '5:1'}}},
  'substitutes': [{ 'player': {'id': 6618,
  'name': 'M. Insaurralde',
  'number': 36,
  'pos': 'M',
  'grid': None}},
  { 'player': {'id': 6095,
  'name': 'E. Cerutti',
  'number': 11,
  'pos': 'M',
  'grid': None}},
  { 'player': {'id': 6627,
  'name': 'N. Blandi',
  'number': 9,
  'pos': 'F',
  'grid': None}},
  { 'player': {'id': 6152,
```

A.6 Endpoint: Team Statistics

Data retrieved: aggregated statistics for each team in each match.

```
{'get': 'fixtures/statistics',
 'parameters': {'fixture': '215550'},
 'errors': [],
 'results': 2,
 'paging': {'current': 1, 'total': 1},
 'response': [{'team': {'id': 460,
  'name': 'San Lorenzo',
  'logo': 'https://media.api-sports.io/football/teams/460.png'},
 'statistics': [{'type': 'Shots on Goal', 'value': 6},
 {'type': 'Shots off Goal', 'value': 11},
 {'type': 'Total Shots', 'value': 25},
 {'type': 'Blocked Shots', 'value': 8},
 {'type': 'Shots insidebox', 'value': 11},
 {'type': 'Shots outsidebox', 'value': 14},
 {'type': 'Fouls', 'value': 6},
 {'type': 'Corner Kicks', 'value': 7},
 {'type': 'Offsides', 'value': None},
 {'type': 'Ball Possession', 'value': '73%'},
 {'type': 'Yellow Cards', 'value': None},
 {'type': 'Red Cards', 'value': None},
 {'type': 'Goalkeeper Saves', 'value': 2},
 {'type': 'Total passes', 'value': 632},
 {'type': 'Passes accurate', 'value': 542},
 {'type': 'Passes %', 'value': '86%'}]}],
 {'team': {'id': 439,
  'name': 'Godoy Cruz',
  'logo': 'https://media.api-sports.io/football/teams/439.png'},
 'statistics': [{'type': 'Shots on Goal', 'value': 4},
 {'type': 'Shots off Goal', 'value': 2},
 {'type': 'Total Shots', 'value': 8},
 {'type': 'Blocked Shots', 'value': 2},
 {'type': 'Shots insidebox', 'value': 5},
 {'type': 'Shots outsidebox', 'value': 3},
 {'type': 'Fouls', 'value': 21},
 {'type': 'Corner Kicks', 'value': 2},
 {'type': 'Offsides', 'value': None},
 {'type': 'Ball Possession', 'value': '27%'},
 {'type': 'Yellow Cards', 'value': 6},
 {'type': 'Red Cards', 'value': None},
 {'type': 'Goalkeeper Saves', 'value': 3},
 {'type': 'Total passes', 'value': 225},
 {'type': 'Passes accurate', 'value': 138},
 {'type': 'Passes %', 'value': '61%'}]}]}
```


A.7 Endpoint: Player Statistics

Data retrieved: aggregated statistics for each player in each match.

```
{'get': 'fixtures/players',
  'parameters': {'fixture': '215550'},
  'errors': [],
  'results': 2,
  'paging': {'current': 1, 'total': 1},
  'response': [{'team': {'id': 460,
    'name': 'San Lorenzo',
    'logo': 'https://media.api-sports.io/football/teams/460.png',
    'update': '2020-05-13T18:18:04+00:00'},
    'players': [{'player': {'id': 35961,
      'name': 'Nicolas Navarro',
      'photo': 'https://media.api-sports.io/football/players/35961.png'},
      'statistics': [{'games': {'minutes': 90,
        'number': 22,
        'position': 'G',
        'rating': '6.3',
        'captain': True,
        'substitute': False},
        'offsides': None,
        'shots': {'total': 0, 'on': 0},
        'goals': {'total': None, 'conceded': 2, 'assists': None, 'saves': 2},
        'passes': {'total': 11, 'key': 0, 'accuracy': '100%'},
        'tackles': {'total': None, 'blocks': 0, 'interceptions': 0},
        'duels': {'total': 0, 'won': 0},
        'dribbles': {'attempts': 0, 'success': 0, 'past': None},
        'fouls': {'drawn': None, 'committed': None},
        'cards': {'yellow': 0, 'red': 0},
        'penalty': {'won': None,
          'committed': None,
          'scored': 0,
          'missed': 0,
          'saved': 0}}]}],
      {'player': {'id': 6606,
        'name': 'Gino Peruzzi',
        'photo': 'https://media.api-sports.io/football/players/6606.png'},
        'statistics': [{'games': {'minutes': 90,
          'number': 4,
          'position': 'D',
          'rating': '8.0',
          'captain': False,
          'substitute': False},
          'offsides': None,
          'shots': {'total': 1, 'on': 1},
```

Appendix B

Github repository contents

Path: https://github.com/fpretto/MiM_Analytics_Tesis

B.1 - data_etls

This folder contains all necessary scripts for the EtLT process that retrieves the data from the API and stores it in the PostgreSQL database. The code files contain the classes described in section [3.2.2 ELT vs ETL vs EtLT](#).

B.2 - data_queries

This folder contains all SQL queries used for creating the Analytical Base Table presented in section [3.3 Analytical Base Table](#) and described in [Appendix C](#).

B.3 - performance_index

This folder contains a jupyter notebook used for developing the players' performance index. Its sections include the preprocessing of the variables (P90, Possession-adjusted), normalization, factor analysis, validation, graphical representation and sensitivity analysis. This process is then modularized in the four scripts *PI_Preprocessing*, *PI_FactorAnalysis*, *PI_Scoring* and *PI_Main*, all described in section [3.4 Statistical Analysis](#).

B.4 - dash_app

This folder contains the scripts for the two views of the web application (Scouting and Player) presented in section [4.2 Football Analytics Web Application](#). By running these scripts the user can interact with the data and search and evaluate players in the different leagues, seasons and teams. It also includes a script for preprocessing the data.

B.5 - datasets

This folder contains the final Analytical Base Table exported into a CSV file. This dataset is used for feeding the web application.

B.6 - plots_and_tables

This folder contains all the plots and tables presented throughout the present thesis. They were all generated in one of the different stages of the pipeline.

B.7 - sandbox

This folder contains scripts and notebooks with practice code and different approaches that were tried during the elaboration of the present thesis.

Appendix C

List of variables in Analytical Base Table

Variable	Definition
league_season	Season of the tournament
player_id	ID of the player
player_name	Name of the player
player_preferred_position	Position of the player with more minutes throughout the season
player_preferred_number	Shirt number of the player with more minutes throughout the season
team_id	ID of the team
avg_team_position	Team's average position throughout the season
player_minutes	Minutes played in the season
wavg_player_rating	Average rating of the player, weighted by minutes played
offsides	Number of offsides in the season
shots_total	Number of total shots in the season
shots_on_goal	Number of shots on goal in the season
goals_total	Number of total goals in the season
goals_conceded_padj	Possession-adjusted number of goals conceded in the season (goalkeepers metric)
goals_assists	Number of assists in the season
goals_saves_padj	Possession-adjusted number of goals saved in the season (goalkeepers metric)
passes_total	Number of total passes in the season
passes_key	Number of passes in the season that lead to a goal attempt, but does not result in a goal.
passes_completed	Number of passes completed in the season
tackles_total_padj	Possession-adjusted total tackles in the season

tackles_blocks_padj	Possession-adjusted total blocks in the season
tackles_interceptions_padj	Possession-adjusted total interceptions in the season
duels_total_padj	Possession-adjusted total duels in the season
duels_won_padj	Possession-adjusted duels won in the season
dribbles_attemps	Number of total dribbles attempts in the season
dribbles_success	Number of successful dribbles in the season
dribbles_past_padj	Possession-adjusted dribbles past in the season
fouls_drawn	Number of fouls drawn in the season
fouls_committed_padj	Possession-adjusted fouls committed in the season
cards_yellow	Number of yellow cards received in the season
cards_red	Number of red cards received in the season
penalty_won	Number of penalties won in the season
penalty_committed_padj	Possession-adjusted number of penalties committed in the season
penalty_scored	Number of penalties scored in the season
penalty_missed	Number of penalties missed in the season
penalty_saved	Number of penalties saved in the season

Bibliography

- [1] Perelman, L.; Barret, E. and Paradis, J. (1997). The Mayfield Electronic Handbook of Technical and Scientific Writing. USA: Mayfield Publishing Company. [<http://web.mit.edu/course/21/21.guide/home.htm>]
- [2] List of professional sports leagues by revenue. [https://en.wikipedia.org/wiki/List_of_professional_sports_leagues_by_revenue]
- [3] Harper, J. (2021). Data experts are becoming football's best signings. BBC News. [<https://www.bbc.com/news/business-56164159>].
- [4] Soccerment Research (2021). The growing importance of Football Analytics. Soccerment. [<https://soccerment.com/the-importance-of-football-analytics/>]
- [5] Herbinet, C. (2018). Predicting Football Results Using Machine Learning Techniques. Imperial College London.
- [6] Ulmer, B. and Fernandez, M. (2014). Predicting Soccer Match Results in the English Premier League. Stanford University
- [7] Yezus, A. (2014). Predicting the outcome of soccer matches using machine learning. Saint-Petersburg State University
- [8] Buursma, D. (2011). Predicting sports events from past results: Towards effective betting on football matches. University of Twente
- [9] Hvattum, L. M. and Arntzen, H. (2010). Using ELO ratings for match result prediction in association football. International Journal of Forecasting 26, 460-470
- [10] Constantinou, A. C.; Fenton, N. E. and Neil, M. (2012). pi-football: A bayesian network model for forecasting association football match outcomes. Knowledge-Based Systems 36, 322-339
- [11] Spearman, W. (2018). Beyond Expected Goals. MIT Sloan Sports Analytics Conference.
- [12] Bunker, R. and Thabtah, F. (2019). A machine learning framework for sport result prediction. Applied Computing and Informatics 15, 27-33
- [13] Castellano, J.; Casamichana, D. and Lago, C. (2012). The Use of Match Statistics that Discriminate Between Successful and Unsuccessful Soccer Teams. Journal of Human Kinetics Volume 31, 139-147
- [14] Tax, M. and Joustra, Y. (2015). Predicting The Dutch Football Competition Using Public Data: A Machine Learning Approach. Transactions on Knowledge and Data Engineering.

- [15] Tippett, J. (2019). The Expected Goals Philosophy: A Game-changing Way of Analyzing Football. Independently published.
- [16] Knutson, T. (2016). Understanding Football Radars For Mugs and Muggles. <https://statsbomb.com/2016/04/understand-football-radars-for-mugs-and-muggles/>
- [17] Stats Perform (2021). Objective Player Recruitment. <https://www.statsperform.com/team-performance/football-performance/player-recruitment/>
- [18] Soccerment Analytics (2021). Soccerment Performance Rating. <https://soccerment.com/faq/>
- [19] InStat (2021). Objective Performance Rating for Teams & Players. https://instatsport.com/football/instat_index
- [20] Wyscout (2021). Possession-adjusted statistics. https://dataglossary.wyscout.com/p_adj/
- [20] Hoffmann, A.; Giovannini, E.; Nardo, M.; Saisana, M.; Saltelli, A.; and Tarantola, S. (2018). Handbook on Constructing Composite Indicators: Methodology and User Guide. Organization for Economic Co-operation and Development (OECD) and Joint Research Centre (JRC) of the European Commission.
- [21] Martin, L. (2016). Sports Performance Measurement and Analytics. Pearson Education, Inc.
- [22] Sumpter, D. (2016). Soccermaths: Mathematical Adventures in the Beautiful Game. Bloomsbury
- [23] Biermann, C. (2019). Football Hackers: The Science and Art of a Data Revolution. Blink Publishing
- [24] Bonde, A. (2013). Thinking Small: Bringing the Power of Big Data to the Masses. Digital Clarity Group
- [25] Densmore, J. (2021). Data Pipelines Pocket Reference: Moving and Processing Data for Analytics. O'Reilly Media
- [26] Silberschatz, A.; Korth, H. and Sudarshan, S. (2019). Database System Concepts. McGraw-Hill Education
- [27] Whitmore, Jonny (2021). Introducing Expected Goals on Target. StatsPerform. <https://www.statsperform.com/resource/introducing-expected-goals-on-target-xgot/>