Princess Sumaya University for Technology
Department of Data Science
Semester 2025/2026

Football Players Selection System

A Data-Driven Scouting Tool for Resource-Constrained Clubs

Prepared By:
Husam Sawaqed
Sedeeq Al-Khazraji
Mariam Omran

Supervised By:
Dr. Serin Atiani

Project Submitted in partial fulfillment for the degree of Bachelor of Science in Data Science

[2025-2026]

Declaration of Originality

This document has been written entirely by the undersigned team members of the project. The source of every quoted text is clearly cited and there is no ambiguity in where the quoted text begins and ends. The source of any illustration, image or table that is not the work of the team members is also clearly cited. We are aware that using non-original text or material or paraphrasing or modifying it without proper citation is a violation of the university's regulations and is subject to legal actions.

Names and Signatures of team members:

Husam Sawaqed

Sedeeq Al-Khazraji

Mariam Omran

Acknowledgments

We would like to extend our sincerest gratitude to our supervisor, Dr. Serin Atiani, for her valuable directions and feedback throughout the development of the Football Players Selection System. We would also like to thank Princess Sumaya University for Technology for the academic environment and resources that made this project possible. Moreover, we thank FBref for offering the public football datasets that we utilized in our system's data pipeline. Finally, we would like to thank our friends and family members for their encouragement and support during this project.

Summary

The Football Players Selection System addresses the issue of finding suitable football players for teams, particularly teams that have financial fair play restrictions and work in tight budgetary constraints [1]. Traditional scouting methods largely depend on subjective assessments, which lacks scalability and objectivity. The goal of our project is to design data-driven web application that applies machine learning methodologies to recommend players based on multiple metrics of performance, team playing styles, and team-specific parameters, such as tactical requirements and cost constraints, using datasets collected from Fbref.

The method utilizes an extensive data pipeline that integrates data extraction, preprocessing, and feature engineering processes to define team playstyles and player characteristics [2]. The framework applies a hybrid machine learningapproach that combines logistic regression to predict the likelihood of player success, linear regression to forecast performance in terms of specific team setup, and K-Nearest Neighbors (KNN) to suggest similar players for possible substitution. Additionally, role-specific web interface allows soccer managers and scouts to provide specific parameters (player role, team formation, and age range) to obtain specific recommendations, complemented by visualization methods like radar graphs for player comparisions.

Key deliverables of the project include a web application, recommendation engine, and thorough documentation. The system design is scalable and user-friendly while also ensuring accessibility, and necessary requirements include users having a web browser and high-quality RAM or GPU resources for training server-side models. Expected outcomes include accurate player recommendations tailored to medium-sized clubs with budgetary constraints, enhancing decision-making processes related to player recruitment. Future research directions include exploring the integration of real-time match data and advanced natural language processing techniques into producing scouting reports [6]. This project greatly contributes to advancing the research area of sports analytics by presenting an economical, automatic, data-informed method of implementing data-driven recruitment for clubs.

List of Abbreviations

List the abbreviations you have used in your project if there are any and what they stand for.

API:	Application Programming Interface
FBref:	Football Reference
GPU:	Graphics Processing Unit
KNN:	K-Nearest Neighbors
ML:	Machine Learning
RAM:	Random Access Memory
UI:	User Interface
UML:	Unified Modeling Language
xG:	Expected Goals

Table of Contents

Table of Figures

Table of Tables

Chapter 1
Introduction

1.1 Overview

In modern football, data-driven decision-making models have become a vital component for clubs seeking competitive advantages. Football scouting has traditionally relied on instinct and intuitive judgments; however, the profession is now in the midst of a revolution driven by the developments in data science. The subject of our graduation project, "Football Players Selection System," aims to promote this change by developing a system to provide player recommendations using performance data, club requirements, and budget constraints. Our system employs formalized data sets procured from FBref, a trusted source for advanced football metrics, and applies various machine learning and data processing techniques to produce objective player recommendations.

This initiative becomes most relevant to the fact that football clubs are facing greater pressure to maximize their transfer operations under the constraints of financial fair play regulations, tight budgets, and rising performance expectations [1]. By using publicly available performance data combined with intelligent modeling methodologies, our framework aims to empower clubs, especially resource-constrained clubs, to reveal hidden talent matching their tactical priorities and budget constraints. The framework will be beneficial to data scientists who work on sports analytics, football scouts, and club management executives who wish to incorporate data-led insights into their recruitment practices.

1.2 Problem Statement

Selecting and hiring appropriate football players is a complex and important task. Clubs have to consider not just the players' performance records and potential, but also how well they fit into tactical systems, salary caps, and overall strategic goals. Traditional methods, often involving manual processes and subjective assessments, have limitations in terms of both scalability and objectivity.

The study addresses this problem by building a data-focused framework to analyze all-around player statistics garnered from FBref and thereafter providing player suggestions aligned with club-determined profiles. The deliverables of the project will involve a web application combined with a core recommendation engine to aid scouts in identifying candidate players on the basis of desired traits, positional needs, and budgetary bounds. The framework will help stakeholders analyze player options backed by data and compare their suitability against the club setting.

The target group includes football scouts, technical directors and analytical staff working with professional and semi-professional football clubs. The goal of the system is to augment conventional scouting practices by adding additional data insights to support better decision-making processes around the transfer of players.

1.3 Related Work

Over the past few years, data science has been infused into football analytics and made tremendous inroads into player recruitment and scouting operations. Several research papers and models have focused on different aspects of the problem statement and presented recommendations from a performance-focused perspective as well as suggestions using natural language processing methods. The following section outlines the most relevant research documents responsible for the development of our framework.

1.3.1 Literature Review

Player Recommendation and Recruitment Systems:Nagaraja et al. propose in their 2024 paper [2] a recommendation system for football players considering multiple parameters such as market value, player statistics, and the particular requirements of the clubs. They use K-Means clustering on normalized performance data of players to group players with analogous features and add additional parameters to recommend players according to both tactical and budgetary constraints.

Koppolu demonstrates in a 2023 article [5] a practical implementation of a basic recommendation system using cosine similarity. It emphasizes data scraping, feature engineering, and similarity-based filtering. Zulyaden et al. [25] developed "A3N," a Flutter-based mobile application designed to streamline the scouting workflow, visualizing player data efficiently for scouts on the move. Sayeed [22] proposed a machine learning framework specifically for identifying talent in lower-tier leagues, using classification models to predict players with national-level potential.

Radaelli et al. explore in their 2024 paper [4] recruitment using machine learning models, including feature selection, classification, and clustering, to identify and recommend players who match a team’s tactical profile. To support natural language interaction with these models, Liu et al. combine in their 2024 paper [6] statistical embeddings, cosine similarity, and GPT-3.5 to support natural language queries and scouting report generation to deliver an advanced platform for player identification.

Dondio [21] implemented multi-criteria decision-making using classifiers like Random Forest to filter suitable recruits, while Barron et al. [26] utilized Artificial Neural Networks (ANN) to objectively identify key performance indicators that influence a player's league status, effectively predicting their career trajectory. Fernandez et al. [8] formulate in their 2021 paper a framework to comprehend player-team chemistry and use simulated player integration to determine best-fit candidates by assessing their performance with projected team dynamics. This concept of simulation is further explored by Cao et al. [29], whose "Team-Scouter" system uses visual analytics to simulate how a potential recruit’s behavior would theoretically integrate into a new team’s tactical setup. Finally, Abhinav et al. [28] introduced an xG-based scouting system that leverages geometric distances and regression to identify high-potential talent aligning with specific team requirements.

Player Assessment and Performance Valuation: Pappalardo et al. introduce in their 2018 paper [3] a sophisticated assessment model, PlayeRank, that ranks players based on a range of performance indicators and role-specific data. The model uses unsupervised learning for role identification as well as contextual scoring of performance.

Decroos et al. [23] "actions speak louder than goals," propose a valuation framework (VAEP) that assesses the impact of every on-ball action—such as passes, dribbles, and tackles—on the game's final outcome. To improve the accuracy of goal-scoring metrics specifically, Haugen et al. propose in their 2023 paper [10] a Bayes-xG model that corrects expected goals (xG) metrics for player and position-specific factors, improving the accuracy of performance evaluations.

Ball et al. [27] compared major commercial rating systems, finding significant variances in how they weight offensive versus defensive contributions, suggesting that clubs require custom evaluation models rather than relying on public scores.

Tactical Analysis and Match Annotation: Decroos et al. utilize in their 2016 paper [7] clustering methods and entropy-based methods to explain team formations and assign roles based on player movement statistics.

StatsBomb presents in their 2023 report [9] an events and 360 data-driven approach to extract team tactics and evaluate performance in football, using detailed event data to analyze tactical patterns and player contributions. Sha et al. [19] furthered this by introducing "SoccerCPD," a change-point detection model that uses spatiotemporal data to automatically identify shifts in tactical formations during a match.

Carta et al. [31] developed "FootApp," an AI-powered system that combines vocal and touch interfaces to assist operators in annotating match events more efficiently. Additionally, Cartas et al. [24] proposed a graph-based method that models players as nodes to spot actions and classify players without supervision. On a broader scale, González-Rodenas et al. [30] analyzed thirteen seasons of La Liga data, proving that technical performance metrics must be contextually adjusted based on team ranking and playing style. Finally, Boudouda and Merouani [20] focused on the predictive aspect, constructing an intelligent system to analyze historical match data and predict future outcomes.

These studies collectively demonstrate a wide range of approaches from clustering and similarity scoring to generative models and team simulations. Our system integrates key elements—such as affordability, explainability, and practical deployment—aimed at real-world usage, especially in resource-constrained clubs.

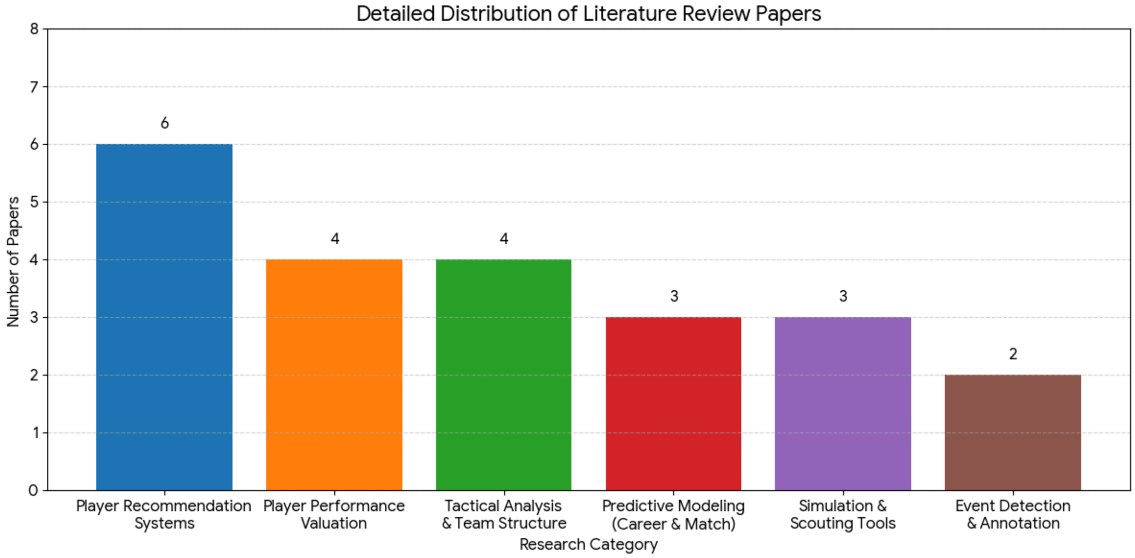


Figure 1.3: the related work analysis

1.3.2 System Review

In addition to theoretical approaches discussed above, many applied systems and tools have been developed to address issues related to player recruitment and scouting in football. These systems provide insightful views about how data-driven approaches may be applied while highlighting areas of need that our project hopes to fill.

Soccermetrics is an online platform offering advanced statistics about player and team performance with a focus on passing networks and player efficiency [11]. While it offers a complete performance overview, it does not contain a tactical and financially tailored recommendation engine for each club's specific tactical and financial constraints, one that is addressed with our system.

Wyscout is a popular platform for scouting players, integrating video analysis, player data, and market trends related to players [12]. While it supports manual scouting through the inclusion of data-driven intelligence, the platform relies heavily on user queries rather than on automated suggestions. Our system aims to automate recommendations, hence reducing the workload faced by scouts.

StatsBomb IQ combines 360-degree data with event-based data to provide deep analysis of player performance and tactical strategy [9]. While this tool is highly advantageous for high-income clubs, its cost and subjectivity make it out-of-reach for less well-funded clubs. Our method closes this gap by harnessing publicly accessible data from sources like FBref.

SciSports offers a recruitment platform that uses machine learning methods for player potential and compatibility assessment [13]. Although this platform makes use of proprietary data and complex models, it is closed-source and requires substantial financial investment. Conversely, our system prioritizes cost-effectiveness and transparency through the use of publicly available data and explanatory models.

These tools illustrate the increasingly prevalent application of data science in football scouting but often end up catering mostly to very well-funded clubs or requiring tremendous manual labor. Our Football Players Selection System is cost-effective and automated while also taking into account club-specific constraints and thus makes it affordable for clubs working on middle-range budgets.

1.4 Contribution:

1.4.1 Novelty in the Idea

The system proposed is an affordable data-driven scouting tool that combines team playing style analysis, player performance data evaluation, and financial constraint inclusion in order to support clubs with budgetary constraints [1]. Unlike existing platforms like Wyscout [12] and SciSports [13] that rely on proprietary data and need manual input, our web-based tool makes use of publicly available FBref data and is specifically designed to support diverse team formations and tactical requirements.

1.4.2 The audience that it serves and how

The system is intended to support football managers, scouts, and analysts who work at moderately resourced and capability-constrained club organizations. It presents a role-specific interface (e.g., manager interface or scout interface) that enables player selection and performance evaluation and thus encourages informed decision-making without needing specialized technical expertise.

1.4.3 Novelty in the Model Choice

The system presented herein incorporates a hybrid machine learning approach involving logistic regression to determine player success likelihood, linear regression to forecast performance indicators for given teams, and K-Nearest Neighbors (KNN) to identify substitute players with similar profiles. This multi-model approach based on existing literature [2], [4] enhances the accuracy of recommendations by tying player attributes to team dynamics and user-specific parameters (e.g., age, position, skills).

1.4.4 Pipeline Structure

The data pipeline is designed for scalability and actionability, incorporating automated data preprocessing to extract team playstyles, creative feature engineering from extracted public data for tactical alignment, spatio-temporal positioning of the players and a user-friendly web interface with visualizations like radar graphs for player comparisons presents novel approach to increasing the power of existing data to inform better player recruitment decisions for coaches and managers. This end-to-end framework, from data scraping to recommendation display, ensures practical deployment for clubs with limited resources.

1.5 Document Outline

Table 1: Document Outline

No.	Chapter	Descirption
1	Introduction	Provides an overview of the project, defines the problem statement, reviews related work and systems, outlines contributions, and describes the document structure.
2	Project Plan	Details the project deliverables, tasks, timeline, roles, risk assessment, cost estimation, and management tools.
3	Requirements Specification	Specifies stakeholders, platform requirements, functional and non-functional requirements, and other constraints.
4	System Design	Describes the architectural, logical, and physical design of the system, including diagrams for components and interfaces.
5	Data Preprocessing	Explains the data collection, profiling, feature engineering, and loading processes for the data pipeline.
6	Implementation	Covers the programming languages, tools, pipeline, and model implementation details, including coding conventions and algorithms.
7	Testing	Discusses the testing approach, results, and insights gained from experiments.
8	Conclusions and Future Work	Summarizes the project outcomes and proposes directions for future enhancements.

Chapter 2

Project Plan

2.1 Project Deliverables

The table below lists the key deliverables of our project.

Table 2 Project Deliverables:

Deliverable	**Description**
Source Code	Full Python code for the data processing, model training, evaluation, and deployment of the model.
Documentation Files	Guides and explanations of the code, description of the algorithm and features used, model performance evaluation, understanding the model, and setting up the system.
Datasets	Cleaned and preprocessed FBref, WhoScored, and Transfermarkt datasets that are used to train and test the model, including metadata.
Trained Model Files	Saved model weights and configuration files for testing or reuse without retraining.
Project Report	The written document covers all aspects of the project: goals, methods, results, and conclusion.
Presentation Materials	A slide deck presentation summarizing the project and showcasing key results.

2.2 Project Tasks

In this table below, we break down the project into main tasks. Each task has a specific goal, timeline, and dependency.

Table 3: Project Tasks

Task no.	Task	Task Description	Duration	Dependencies
T.1	Defining Project Scope	Identify project goals, user needs, data requirements, system	4 weeks	None

		inputs/outputs, performance, security, and scalability needs.		
T.2	Literature Review	Understand previous studies, explore existing literature on the subject matter.	2 week	T.1
T.3	Datasets	Searching for online available datasets and scraping data from websites.	2 week	T.1, T.2
T.4	Data Analysis	Analyze FBref, WhoScored, and Transfermarkt datasets to understand structure, size, missing values, outliers, and suitability for training the model.	2 weeks	T.4
T.5	Development of Metadata	Extract metadata such as dataset dimensions, column types,missing value counts, and descriptions.	1 week	T.4, T.5
T.6	Database Design	Define the structure for storing players, teams, stats, and recommendation results.	1 weeks	T.4, T.5
T.7	System Design	Design the system architecture, data flow, model integration, and component interaction.	4 weeks	T.1, T.6, T.7
T.8	Data Engineering	Implement data ingestion, cleaning, and feature extraction from the datasets.	4 weeks	T.4, T.5
T.9	Model development & Implementation	Build and train the player recommendation model using relevant features.	4 weeks	T.7
T.10	Testing	Test data pipeline, model accuracy, and the overall system functionality.	3 weeks	T.8
T.11	Documentation	Prepare technical documentation and final report covering the system requirements, design, testing, and results.	4 weeks	T.1, T.2, T.3, T.4, T.5, T.6, T.7, T.8, T.9
T.12	Presentation Preparation	Create the presentation slides and demo for the project.	2 week	T.9, T.10

2.3 Gantt Charts

The Gantt charts below show the timeline of our group project tasks.

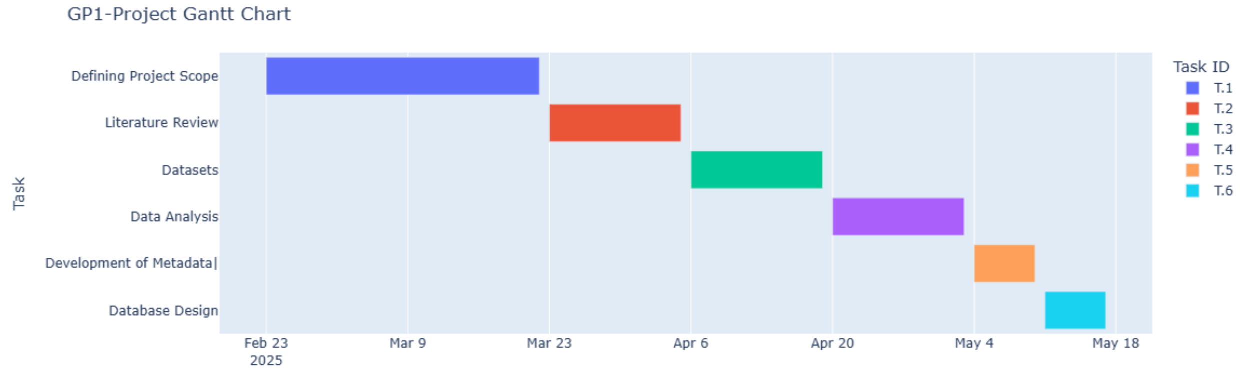


Figure 1: Gantt Chart 1

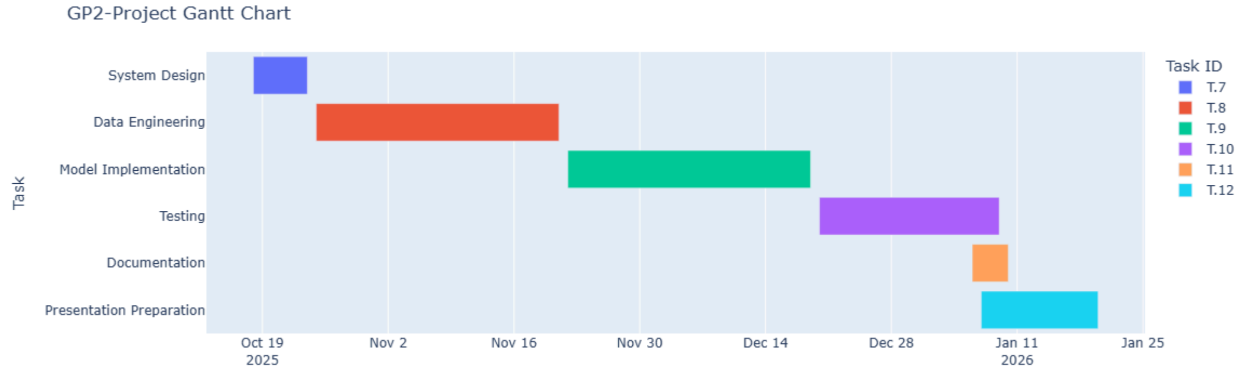


Figure 2: Gantt Chart 2

2.4 Roles and Responsibilities

This table shows the roles and contributions of each team member in our group project.

Table 4: Roles and Responsibilities

Name	**Roles**
Seddiq Al-khazraji	T.1, T.2,T.3, T.4
Husam Sawaqed	T.1,T.2

Mariam Yaqoub	T.1,T.2,T.5

2.5 Risk Assessment

This table identifies possible risks, how likely they are, what impact they may have, and how we plan to deal with them if they happen.

Table 5: Risk Assessment

Task No.	**Task**	**Risk Description**	**Probability**	**Impact**	**Response**
T.1	Defining Project Scope	Unclear or changing requirements	Medium	High	Regular meetings with supervisor; write a clear scope document and validate early
T.2	Literature Review	Difficulty finding relevant research or credible sources	Low	Medium	Use trusted academic databases; ask supervisor for recommended sources, and surf the internet for papers and articles
T.3	Datasets	Incomplete, inconsistent, or blocked access to online datasets	Low	High	Backup datasets; scrape and store early; document scraping process
T.4	Data Analysis	Misinterpreting dataset structure or quality	Medium	High	Use descriptive stats; cross-validate findings with dataset documentation
T.5	Database Design	Poor schema design causing inefficiencies	Medium	Medium	Draft ER diagrams; get early feedback; apply normalization
T.6	Data Engineering	Errors in preprocessing or feature extraction	Medium	High	Use pipeline approach; write unit tests;

					document each step
T.7	Model Coding Implementation	Model underperforms or fails to generalize	Medium	High	Try different algorithms; tune hyperparameters; document model decisions
T.8	Testing	Not enough test cases or poor coverage	Medium	Medium	Create test plan; validate outputs with dummy inputs
T.9	Documentation	Missing or unclear documentation	Low	Medium	Maintain notes throughout development; divide writing tasks among team
T.10	Presentation Preparation	Incomplete or last-minute presentation/demo issues	Medium	High	Start early; assign roles; rehearse with dry runs and demo checklist

2.6 Project Management Tools

The table below lists and briefly explains the tools that helped us manage our project.

Table 6: Project Management Tools

Tool	**Purpose**
Git & GitHub	Version control, collaborative coding, issue tracking, and code history.
Google Drive & Docs	Collaborative writing and sharing of reports, documents, and presentations.
Trello	Task tracking and project planning using Kanban boards.
Zoom	Online team meetings, discussions, and presentations.
Google Calendar	Scheduling meetings, deadlines, and managing team availability.
Google Colab	Cloud-based collaborative coding and model training using Python.
Jupyter Notebook	Local development and testing of data analysis and model prototypes.
Streamlit	Interactive dashboards for modeling predictions and visualizing data.

Chapter 3

Requirements Specification

3.1 Stakeholders

The table below describes the stakeholders involved in the system, who they are, their role, and how they interact with the system.

Table 7: Stakeholders

Stakeholder	**Role**	**Importance**
Developers	They are responsible for implementing data processing and model development.	High
PSUT	Academic supervisors in Princess Sumaya University are responsible for reviewing project progress, providing feedback and guidelines for academic quality.	Medium
Football Club Analysts / Scouts	Core users of the system who use the system for identifying suitable players for teams and scouting.	High
Football Club Managers	Use the system for player acquisition in teams.	Medium
Data Providers	Provides player datasets and statistics (e.g., FBref, WhoScored, Transfermarkt).	High

3.2 Platform Requirements

The system is divided into two main components:

- Client-Side: Used by team analysts or scouts to interact with the model via a simple web interface or notebook. Table describes the client-side requirements.

- Server-Side: Handles data processing, model training, and recommendation generation. Table describes server-side requirements.

Client-Side Requirements

Table 8: Client-Side Requirements

Requirement ID	Requirement	Type	Justification	Priority
C1	Network Connection	Web Service	Access the system through the web via a network connection.	Essential
C2	Access Device	Hardware	Device for accessing the model and interacting with the system.	Essential
C3	Web Browser	Software	Required to open the system through a notebook or the deployed model interface.	Essential

Server-Side Requirements

Table 9: Server-Side Requirements

Req. ID	Requirement	Type	Justification	Priority
S1	Storage Service	Web Service	Store players datasets and model outputs for training and testing.	Essential
S2	Internet Connection	Web Service	Scrape datasets from online servers, push into github and run cloud servers.	Essential
S3	Model Deployment	Web Service	Deployment of the model.	Essential
S4	Development Environment	Software	Python Language and Python environments for training and implementing the model.	Essential
S5	High-Ram Runtime or GPU Runtime	Hardware	For processing large datasets and training ML models.	Recommended

3.3 Functional Requirements

3.3.1 Functional Requirements (High-Level)

The table below includes the core functionalities that the system must support at a high level.

Table 10: Functional Requirements

Req. ID	Functional Requirements	Description	Priority
FR1	Dataset Upload	System must allow uploading player datasets from FBref or local files.	Essential
FR2	Data Preprocessing	Raw data must be cleaned and formatted such that it is suitable for analysis and model training.	Essential
FR3	Statistical summary delivery	System must deliver statistical summary for players and teams for management assessment, also to be processed as input for predictive models.	Essential
FR4	Model Serving	Probabilistic outcome from multiple predictive models will serve as base for recommendation by the system.	Essential
FR5	Player Scouting Recommendation	System must recommend and suggest players for scouting based on some criteria.	Essential
FR6	Model Evaluation	Model performance metrics.	Recommended

3.3.2 Detailed Functional Requirements

The table below describes the specific input, process, output, constraints, and importance of each sub-function that supports the high-level functionalities.

Table 11: Detailed Functional Requirements

Req. ID	Inputs	Outputs	Constraints	Process
DFR1	Raw or structured	Cleaned dataset	Handle missing values, duplicates, or any	Handling missing values, merging

	datasets.		inconsistencies.	datasets, data normalization and cleaning.
DFR2	Player’s Statistical Data	Feature Matrix	Features must be relevant for player selection.	Feature Engineering.
DFR3	Team Criteria	List of Filtered Players	Availability of filtered attributes in the dataset.	Filtering the dataset to narrow it down.
DFR4	Feature Matrix + Target Criteria	Trained ML Model	Need for sufficient data.	Train the ML model.
DFR5	Recommendation Output	Report document or PDF	Output Format must be available and supported.	Allow the user to view and download output results.
DFR6	User/System Activity	Log File	Log-ins must not expose sensitive data about users.	Trace system events, errors and user activity.

3.4 Non-Functional Requirements

The table below describes the non-functional requirements the system must provide.

Table 12: Non-Functional Requirements

Req.ID	Requirement	Description	Priority
NFR1	User Friendly	Interface must be user friendly and intuitive for non-technical users.	Essential

NFR2	Scalability	System should be able to scale data and handle data growth without any performance degradation.	Recommended
NFR3	Performance	System must have a quick response with minimal delays.	Essential
NFR4	Reliability	System must be reliable and consistent with certain conditions.	Essential
NFR5	Accessibility	System must be accessible and usable for everyone.	Essential
NFR6	Security	System should protect the user's data and authorize accessibility.	Essential
NFR7	Code Quality	Code must be clean.	Essential

Chapter 4

System Design

4.1 Architectural Design:

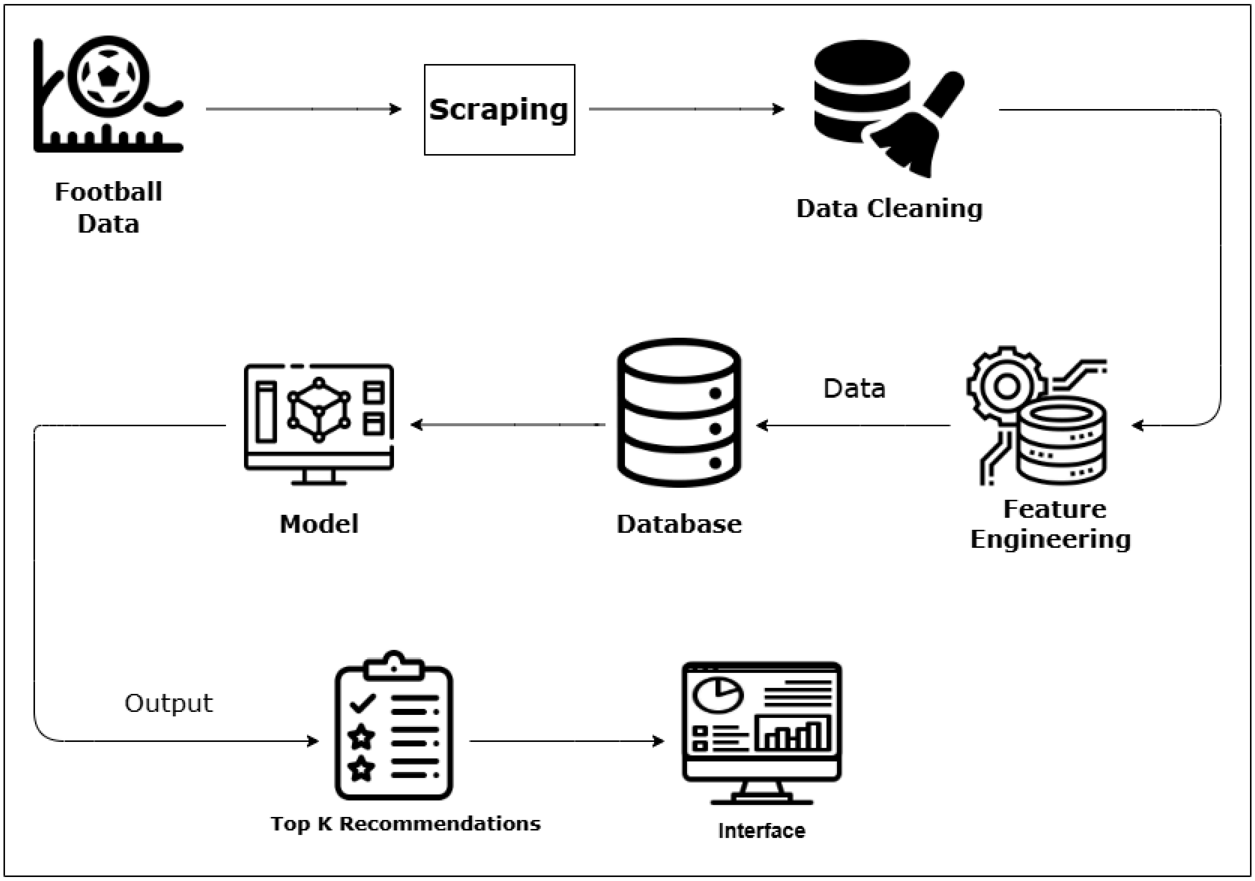


Figure 3: Architectural Design

The architecture of our system is visualized in Figure (3) above, which displays an overview of our system's main components and how they are connected.

4.1.1 Data Preprocessing

Figure (3) shows the data's preprocessing steps. After the data is scraped from multiple football data sources, it goes through cleaning process. The system performs feature engineering to extract teams' playstyles using spatio-temporal data. The player roles features is constructed for each player using the positions they take on the field combined with the performance metrics that describes their playstyle the most. The data with the extracted features get feeded to the database.

4.1.2 Model

The processed dataset, stored within a PostgreSQL database, serves as the foundation for the system’s predictive architecture. Unlike traditional linear approaches, this project utilizes a **Hybrid Ensemble Framework** to evaluate transfer viability. A **Logistic Regression** classifier is employed to calculate a binary success probability ($P_{success}$), determining the likelihood that a player's integration will result in a positive performance trend (Elo increase). Simultaneously, an **XGBoost Regressor** is utilized to predict the specific magnitude of a player's impact on a team's strength. XGBoost was selected for its superior ability to capture complex, non-linear relationships between performance metrics—such as non-penalty expected goals ($npxG$) and progressive carries—and the specific tactical requirements of the target team.

To identify suitable replacements, the system leverages a **Vector Similarity Search** executed within the database. Players are transformed into high-dimensional embeddings based on their "Player DNA" profile, which represents their statistical fingerprint across ten key performance indicators. These vectors are queried using a **K-Nearest Neighbors (KNN)** logic to find candidates with the most similar playing styles. Finally, these predictive outputs are aggregated into a **Transfer Value Index (TVI)**. This weighted decision metric integrates the predicted impact, the probability of success, and the financial Return on Investment (ROI), providing a comprehensive ranking that balances sporting ambition with financial sustainability.

4.1.3 User Interface

Figure (3) shows the interface part of the system. The user chooses their role (Manager, Scout), then logs in. Based on the role the user have they are granted one or two operations (Search for player to recruit, Get team & player stats). Search operations display the recommended players and statistics based on user inputs.

4.2 Logical Model Design

4.2.1 Use Case Diagram

Figure 4: Use-Case Diagram

Figure (4) shows the use case of the system. It provides a view of the agents involved and the actions they take, and what their actions will lead to.

Agents: Football managers & scouts

The user logs into the system, based on their role, they are given access to search for players or request access to statistics or both. The user inputs the search parameters required. The access statistics operation displays the requested data. The search operations prompt the system to calculate the player suitability, then display the recommendation along with appropriate visualization.

4.2.2 Activity Diagram

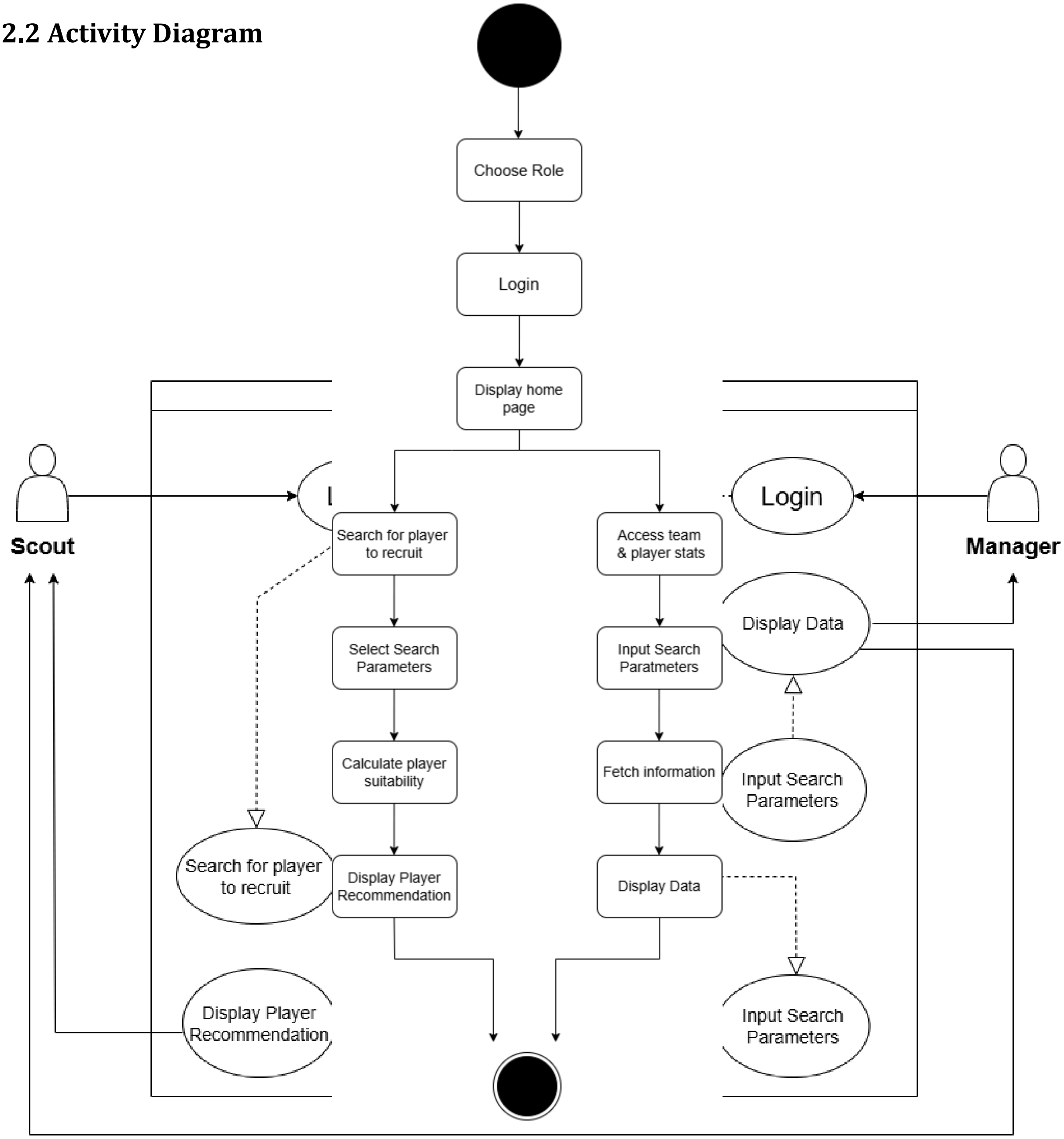


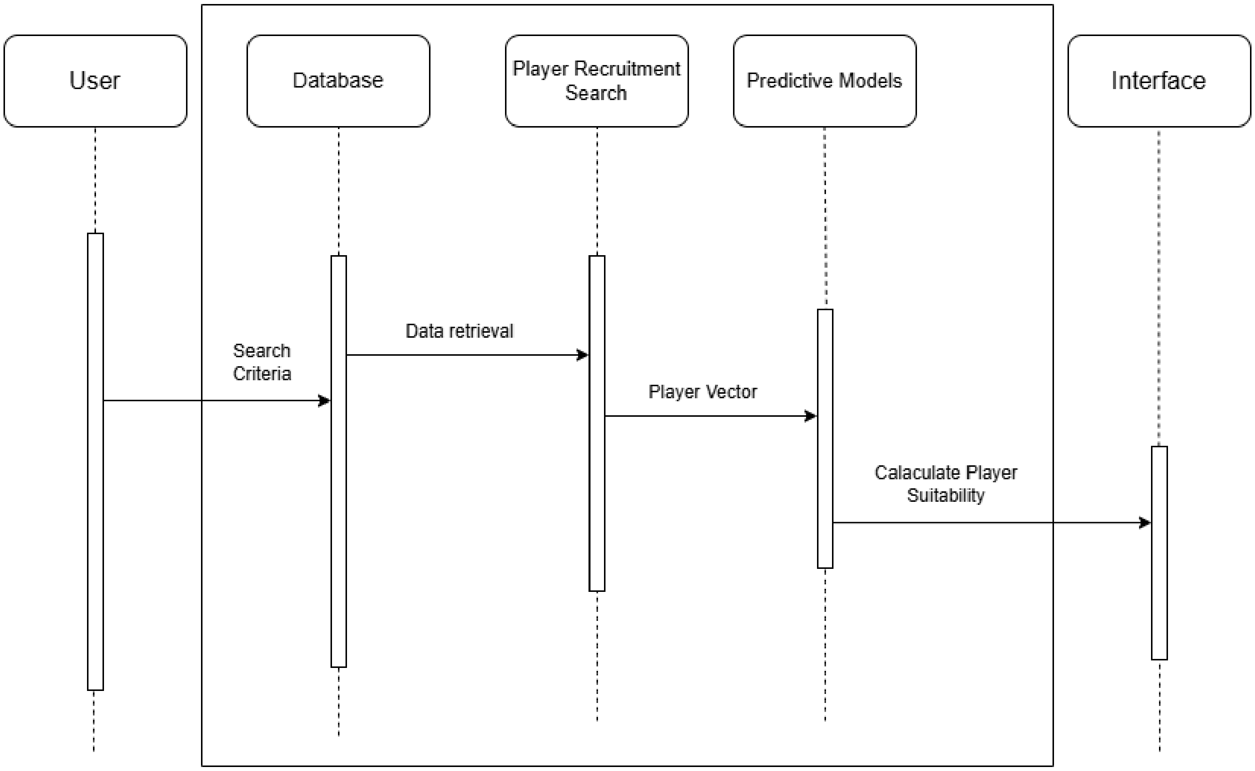
Figure 5: Activity Diagram

Figure (5) shows the steps the user goes through as they choose their role and log in. The user at this point is authorized; they choose the desired service, enter the required parameters, which are used to retrieve data and display it immediately if the user chooses the access stats service. If the user chooses to search for players to recruit for a target team, the data is used to calculate the player suitability based on the user inputs (Team's playstyle, Player's role, Target League, Team formation, Team's formation, Age range & Player's standout skills). The system displays the recommended player profile with a list of the next top candidates.

4.2.3 Sequence Diagram

Figure 6: Sequence Diagram

The figure (6) shows the system's sequence diagram. The system retrieves the required data based on the user search inputs. The two components combined act as an input to calculate the player suitability for the target team using multiple predictive models. The system returns the top k player recommendations and displays a detailed player profile.



4.3 Physical Model Design

4.3.1 User Interface Design

The interface chosen is a web application interface**.** The figure (7) below shows the first page where the user chooses their role.

Figure 7: User Interface Design 1

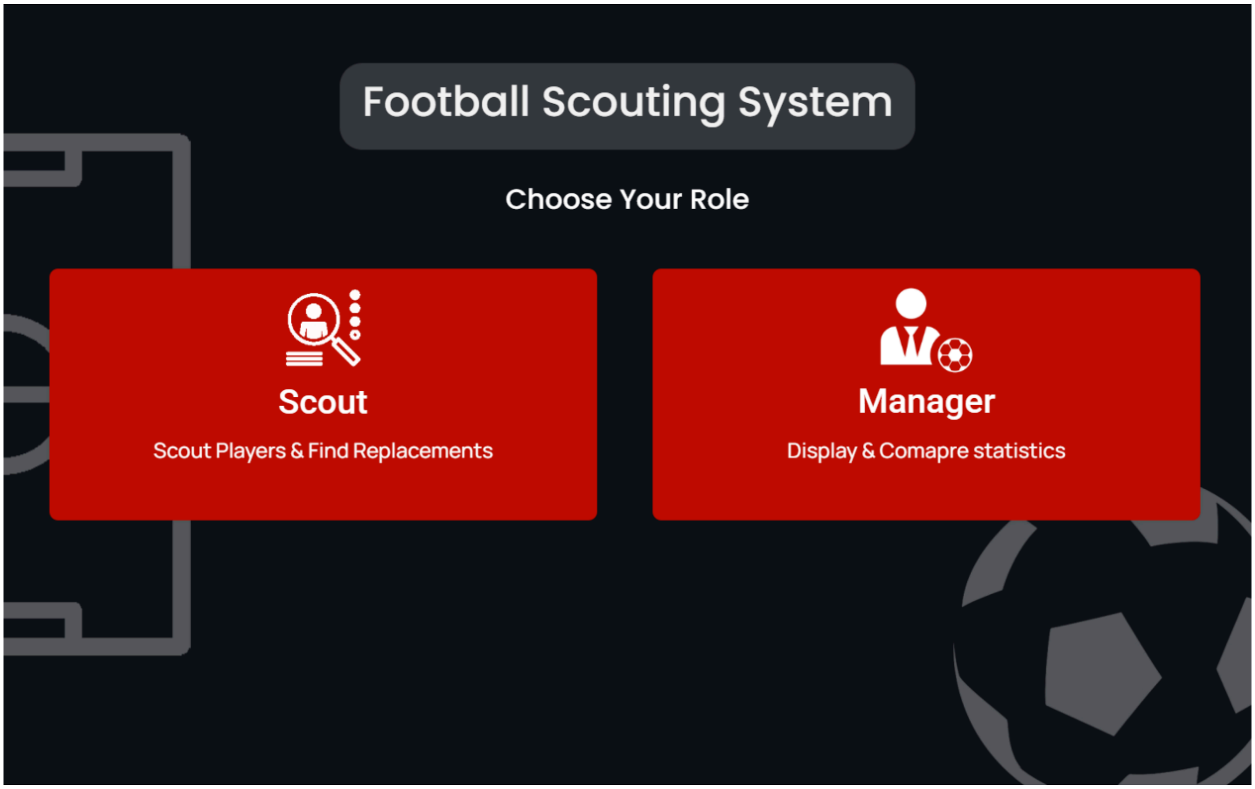

Football Scouting System
Choose Your Role
Scout
Scout Players & Find Replacements
Manager
Display & Comapre statistics

Figure 7: User Interface Design 1

After the role is chosen, the user is directed to the login interface shown in figure (8) below, which authorizes the user to enter the application by checking Username and Password.

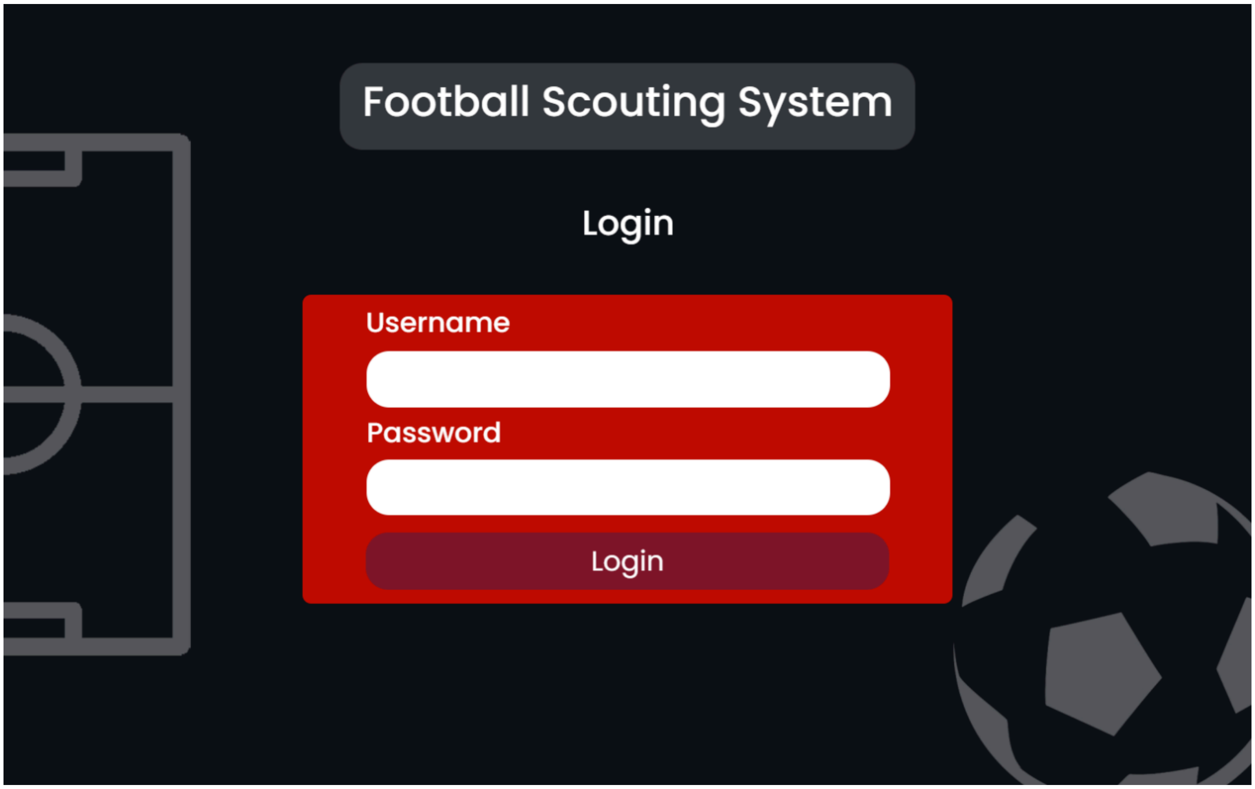


Figure 8: User Interface Design 2

In Figure (9) below is the home page interface that appears to the user after logging in, where the desired service is chosen.

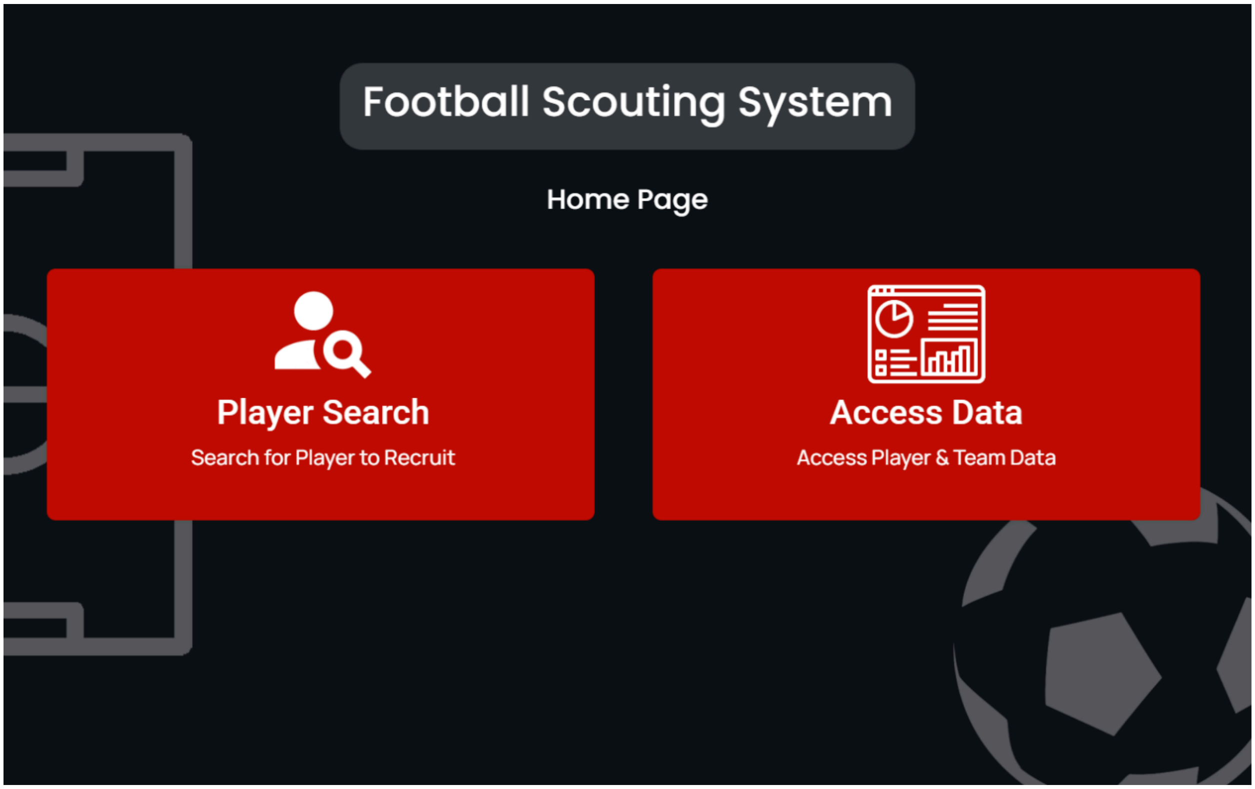
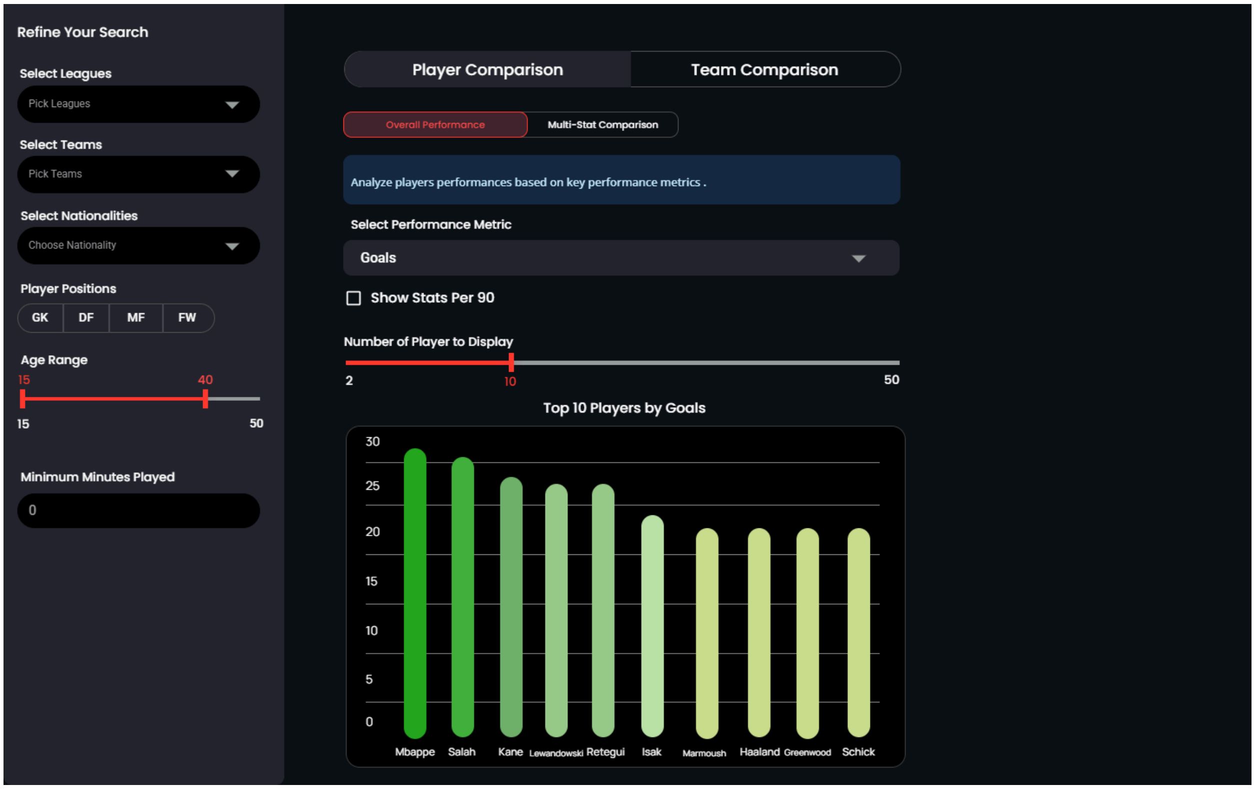


Figure 8: User Interface Design 3

In figure (9) below, this is a user interface design for the access data page that appears to the user to enter the player or team information and choose the performance metric used to measure player or team performance. The user can choose between player or team comparisons based on one or multiple performance metrics. The user chooses the leagues, teams, nationalities, positions, age range and has the option to set a limit for the minimum minutes played. If the user does not select a value for all of the parameters, the system retrieves data for all top 5 European leagues.



Refine Your Search

Select Leagues: Pick Leagues

Select Teams: Pick Teams

Select Nationalities: Choose Nationality

Player Positions: GK | DF | MF | FW

Age Range: 15 – 40 (15 to 50)

Minimum Minutes Played: 0

Player Comparison | Team Comparison

Overall Performance | Multi-Stat Comparison

Analyze players performances based on key performance metrics .

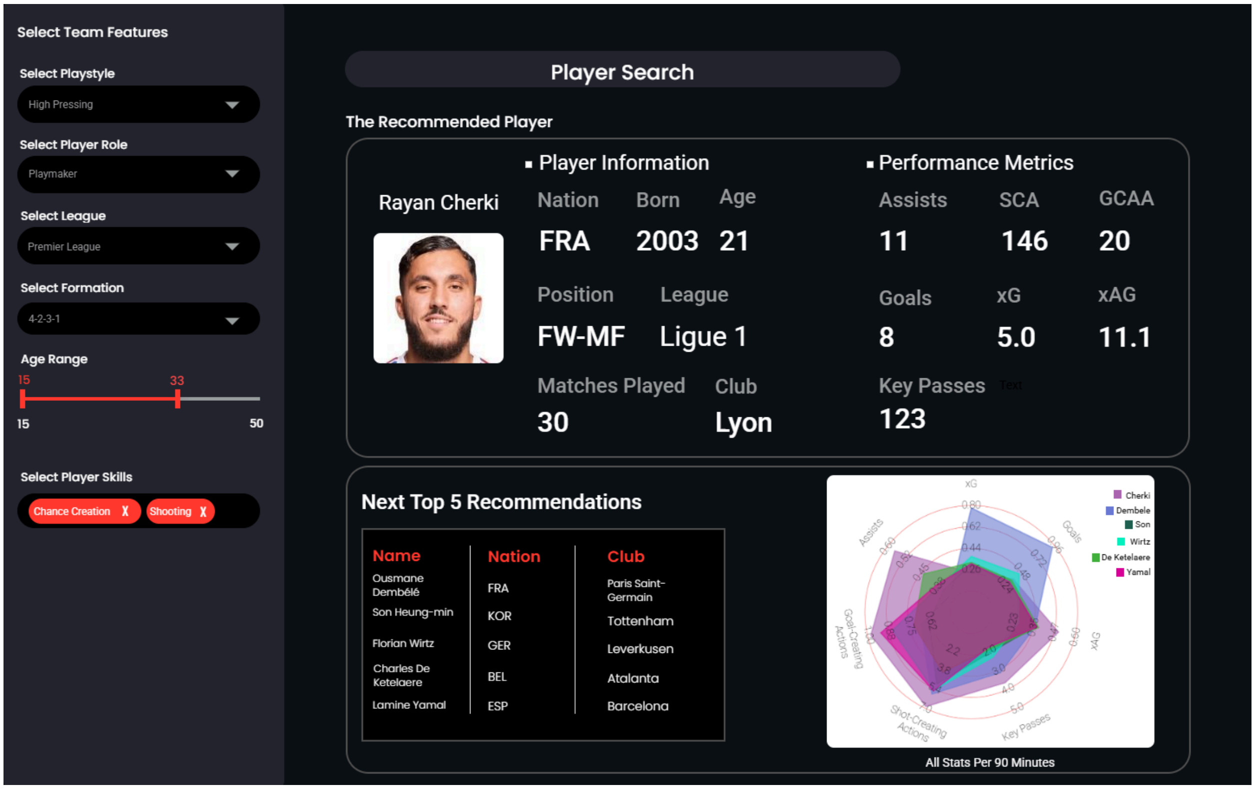
Select Performance Metric: Goals

□ Show Stats Per 90

Number of Player to Display: 10 (2 to 50)

Figure 9: User Interface Design 4

In figure (10) below, this is a design for the player search page where the user selects the team features that is desired to recruit a player for. The features include: team's playstyle, player's role, target league, formation, age range, and player skills. The system displays the recommended player profile, the next top 5 recommendation candidates and a radar graph to compare the recommended player next to the other candidates. The comparison is based on performance metrics chosen by the model depending on the user selections.



Select Team Features

Select Playstyle: High Pressing

Select Player Role: Playmaker

Select League: Premier League

Select Formation: 4-2-3-1

Age Range: 15 – 33 (15 – 50)

Select Player Skills: Chance Creation X, Shooting X

Player Search

The Recommended Player

Rayan Cherki

- Player Information

Nation	Born	Age
FRA	2003	21

Position	League
FW-MF	Ligue 1

Matches Played	Club
30	Lyon

- Performance Metrics

Assists	SCA	GCAA
11	146	20

Goals	xG	xAG
8	5.0	11.1

Key Passes	Text
123	

Next Top 5 Recommendations

Name	Nation	Club
Ousmane Dembélé	FRA	Paris Saint-Germain
Son Heung-min	KOR	Tottenham
Florian Wirtz	GER	Leverkusen
Charles De Ketelaere	BEL	Atalanta
Lamine Yamal	ESP	Barcelona

All Stats Per 90 Minutes

Figure 10: User Interface Design 5

4.3.2 Database Design

The figure is in the link here:https://drive.google.com/file/d/1tDaB0BeoFBe2fM4SoWJh2Fmz3qWCyKll/view?usp=sharing

Chapter 5

Data Preprocessing

5.1 Data Collection and Description

The data was collected from multiple football data websites. Player & Team Performance was scraped from FBref[14],transfer history data was scraped from Transfermarkt[15] using the WorldFootballR API[16], Spatio-temporal was scraped from WhoScored[17]using the SoccerAction API[18].

The performance data contains performance metrics for all players and teams in the top five European leagues (English Premier League, French Ligue 1, German Bundesliga.1, Italian Serie A, Spanish La Liga) across the last five football seasons. Transfer history data contains the transfers of clubs in the top five Leagues across the last five seasons. The spatio-temporal data contains the match events for all the matches in the current season 2024/25 across the top five leagues.

Player performance data has records for 5573 unique players, Team Performance data has records for 132 unique teams, Transfer history data has 2141 transfer for 98 teams and spatio-temporal data contains the events of 1752 matches from the current season.

5.2 Metadata

5.2.1 Spatio-temporal Data

Games

Column name	Data type	Description
game_id	int64	The unique identifier for the game.
season_id	int64	The unique identifier for the season.
competition_id	int64	The unique identifier for the competition.
game_day	object	Number corresponding to the weeks or rounds into

		the competition this game is.
game_date	datetime64[ns]	The date when the game was played.
home_team_id	int64	The unique identifier for the home team in this game.
away_team_id	int64	The unique identifier for the away team in this game.
home_score	int64	The final score of the home team.
away_score	int64	The final score of the away team.
duration	int64	The total duration of the game in minutes.
referee	object	The name of the referee.
venue	object	The name of the stadium where the game was played.
attendance	int64	The number of people who attended the game.
home_manager	object	The name of the manager of the home team.
away_manager	object	The name of the manager of the away team.

Table 13: Games

Teams

Column name	Data type	Description
team_id	int64	The unique identifier for the team.
Team_name	object	The name of the team.

Table 14: Teams

Players

Column name	Data type	Description
game_id	int64	The unique identifier for the game.
team_id	int64	The unique identifier for the player's team.
player_id	int64	The unique identifier for the player.
player_name	object	The name of the player.
is_starter	bool	Whether the player is in the starting lineup.
minutes_played	int64	The number of minutes the player played in the game.
jersey_number	int64	The player's jersey number.
starting_position	object	The starting position of the player.

Table 15: Players

Events

Column name	Data type	Description
game_id	int64	The unique identifier for the game.
event_id	int64	The unique identifier for the event.
period_id	int64	The unique identifier for the part of the game in which the event took place.
team_id	int64	The unique identifier for the team this event relates to.
player_id	float64	The unique identifier for the player this event relates to.

type_id	int64	The unique identifier for the type of this event.
timestamp	datetime64[ns]	Time in the match the event takes place, recorded to the millisecond.
minute	int64	The minutes on the clock at the time of this event.
second	int64	The second part of the timestamp.
outcome	bool	Whether the event had a successful outcome or not.
start_x	float64	The x coordinate of the location where the event started.
start_y	float64	The y coordinate of the location where the event started.
end_x	float64	The x coordinate of the location where the event ended.
end_y	float64	The y coordinate of the location where the event ended.
qualifiers	object	A JSON object containing the Opta qualifiers of the event.
related_player_id	float64	The ID of a second player that was involved in this event.
touch	bool	Whether the event was a on-the-ball action or not.
goal	bool	Whether the event was a goal or not.

shot	bool	Whether the event was a shot or not.
type_name	object	The name of the type of this event.

5.2.2 Transfer History Data

Table 17: Transfer History

Column name	Data type	Description
team_name	object	Team name player has transferred to
league	object	League name player has transferred to
country	object	Country name player has transferred to
season	int64	The year player transferred
transfer_type	object	Whether player arrive or depart
player_name	object	The name of the player
player_url	object	Player page URL
player_position	object	The position of the player
player_age	int64	The age of the player
player_nationality	object	The nationality of the player
club_2	object	Team name player has transferred from
league_2	object	League name player has transferred from
country_2	object	Country name player has transferred from
transfer_fee	float64	The transfer fee

is_loan	bool	Whether transfer is loan or not
transfer_notes	object	Whether transfer is due to end of loan or not
window	object	Whether transfer happened in winter or summer
in_squad	float64	Number of games the player has been in squad
appearances	float64	Number of games the player played
goals	float64	Number of goals the player scored
minutes_played	float64	Number of minutes the player played

5.2.3 Performance Data

Player & Team Stats

Table 18 : Player & Team stats

Column name	**Data type**	**Description**
Season_End_Year	int64	The year season ends
Squad	object	Team’s name
Comp	object	League’s name
Player	object	Player’s name
Nation	object	Player’s nationality
Pos	object	Position most commonly played by the player
Age	object	Current age
Born	float64	Year of birth
MP	int64	Matches Played by the player

Starts	int64	Game or games started by player
Min	int64	Minutes played by player
Mins_Per_90	float64	Minutes played divided by 90
Gls	int64	Goals scored
Ast	int64	Assists provided
G+A	int64	Goals and Assists
G_minus_PK	int64	Non-Penalty Goals
PK	int64	Penalty Kicks scored
PKatt	int64	Penalty Kicks Attempted
CrdY	int64	Yellow Cards
CrdR	int64	Red Cards
xG	float64	Expected Goals including penalty kicks
npxG	float64	Non-Penalty Expected Goals
xAG	float64	**Expected Assisted Goals** (xG which follows a pass that assists a shot)
npxG+xAG	float64	Non-Penalty Expected Goals plus Assisted Goals (includes penalty kicks)
PrgC	float64	**Progressive Carries** (Carries that move the ball towards the opponent's goal line at least 10 yards from its furthest point in the last six passes, or any carry into the penalty area. Excludes carries which end in the defending 50% of the pitch)
PrgP	float64	**Progressive Passes** (Completed passes that move the ball towards the opponent's goal line at least 10 yards from its furthest point in the last six passes, or any completed pass into the penalty area. Excludes passes from the defending 40% of the pitch)

PrgR	float64	**Progressive Passes Received** (Number of times a player successfully received a progressive pass)
Gls_Per_90	float64	Goals Scored per 90 minutes
Ast_Per_90	float64	Assists per 90 minutes
G+A_Per_90	float64	Goals and Assists per 90 minutes
G_minus_PK_Per_90	float64	Goals minus Penalty Kicks scored per 90 minutes
G+A_minus_PK_Per_90	float64	Goals plus Assists minus Penalty Kicks scored per 90 minutes
xG_Per_90	float64	Expected Goals per 90 minutes (includes penalty kicks)
xAG_Per_90	float64	Expected Assisted Goals per 90 minutes
xG+xAG_Per_90	float64	Expected Goals plus Assisted Goals per 90 minutes (includes penalty kicks)
npxG_Per_90	float64	Non-Penalty Expected Goals per 90 minutes
npxG+xAG_Per_90	float64	Non-Penalty Expected Goals plus Assisted Goals per 90 minutes
Url	object	Player Page URL
Sh	int64	Total shots (does not include penalty kicks)
SoT	int64	Shots on target (do not include penalty kicks)
SoT_percent	float64	**Shots on Target Percentage** (Percentage of shots that are on target)
Sh_per_90	float64	Shots total per 90 minutes
SoT_per_90	float64	Shots on target per 90 minutes

G_per_Sh	float64	Goals per shot
G_per_SoT	float64	Goals per shot on target
Dist	float64	**Average shot distance** (Average distance in yards, from goal of all shots taken)
FK	float64	Shots from Free Kicks
npxG_per_Sh	float64	Non-Penalty Expected Goals per shot
G_minus_xG	float64	Goals minus Expected Goals (includes penalty kicks)
Np: G_minus_xG	float64	Non-Penalty Goals minus Non-Penalty Expected Goals (includes penalty kicks)
Touches	float64	Number of times a player touched the ball. **Note:** Receiving a pass, then dribbling, then sending a pass counts as one touch
Def Pen_Touches	float64	Touches in defensive penalty area
Def 3rd_Touches	float64	Touches in defensive third
Mid 3rd_Touches	float64	Touches in middle third
Att 3rd_Touches	float64	Touches in attacking third
Att Pen_Touches	float64	Touches in attacking penalty area
Live_Touches	float64	**Touches (Live-Ball).** Does not include corner kicks, free kicks, throw-ins, kick-offs, goal kicks or penalty kicks
Att_Take	float64	**Take-Ons Attempted** (Number of attempts to take on defenders while dribbling)

Succ_Take	float64	**Successful Take-Ons** (Number of defenders taken on successfully, by dribbling past them)
Succ_percent_Take	float64	**Successful Take-On Percentage**(Percentage of Take-Ons Completed Successfully, Unsuccessful take-ons include attempts where the dribbler retained possession but was unable to get past the defender)
Tkld_Take	float64	**Times Tackled During Take-On** (Number of times tackled by a defender during a take-on attempt)
Tkld_percent_Take	float64	**Tackled During Take-On Percentage**(Percentage of time tackled by a defender during a take-on attempt)
Carries	float64	Number of times the player controlled the ball with their feet.
TotDist_Carries	float64	**Total Carrying Distance** (Total distance, in yards, a player moved the ball while controlling it with their feet, in any direction)
PrgDist_Carries	float64	**Progressive Carrying Distance** (Total distance, in yards, a player moved the ball while controlling it with their feet towards the opponent's goal)
Final_Third_Carries	float64	**Carries into Final Third** (Carries that enter the third of the pitch closest to the goal)
CPA_Carries	float64	**Carries into Penalty Area**(Carries into the 18-yard box)
Mis_Carries	float64	**Miscontrols** (Number of times a player failed when attempting to gain control of a ball)
Dis_Carries	float64	**Dispossessed** (Number of times a player loses control of the ball after being tackled by an opposing player. Does not include attempted take-ons)
Rec_Receiving	float64	**Passes Received** (Number of times a player successfully received a pass)
Mn_per_MP	float64,	Minutes Per Match Played

Min_percent	float64	**Percentage of Squad Minutes Played** (Player minutes played divided by team total minutes)
Mn_per_Start	float64	Minutes Per Match Started
Compl_Starts	float64	Complete Matches Played
Subs	float64	**Substitute Appearances** (Game or games player did not start)
Mn_per_Sub	float64	Minutes Per Substitution
unSub	int64	Matches as Unused Sub
PPM_Team.Success	float64	Average number of points earned by the team from matches in which the player appeared.
onG_Team.Success	float64	Goals scored by team while on pitch.
onGA_Team.Success	float64	Goals allowed by team while on pitch
plus_per__minus__Team.Success	float64	Goals scored minus goals allowed by the team while the player was on the pitch
plus_per__minus_90_Team.Success	float64	Goals scored minus goals allowed by the team while the player was on the pitch per 90 minutes played
On_minus_Off_Team.Success	float64	Goals per 90 minutes by the team while the player was on the pitch minus goals allowed per 90 minutes by the team while the player was off the pitch
onxG_Team.Success	float64	Expected goals by team while on pitch
onxGA_Team.Success	float64	Expected goals allowed by team while on pitch
xGplus_per__minus__Team.Success	float64	Expected goals scored minus expected goals allowed by the team while the player was on the pitch

xGplus_per__minus_90_Team.Success	float64	Expected goals scored minus expected goals allowed by the team while the player was on the pitch per 90 minutes played
On_minus_Off_Team.Success	float64	Expected goals per 90 minutes by the team while the player was on the pitch minus expected goals per 90 minutes by the team while the player was off the pitch
Cmp_Total	float64	**Passes Completed** (Includes live ball passes, crosses, corner kicks, throw-ins, free kicks and goal kicks.
Att_Total	float64	Passes Attempted
Cmp_percent_Total	float64	Pass Completion Percentage
TotDist_Total	float64	**Total Passing Distance** (Total distance, in yards, that completed passes have traveled in any direction)
PrgDist_Total	float64	**Progressive Passing Distance** (Total distance, in yards, that completed passes have traveled towards the opponent's goal)
Cmp_Short	float64	**Short Passes Completed** (Passes between 5 and 15 yards)
Att_Short	float64	Short Passes Attempted
Cmp_percent_Short	float64	Short Pass Completion Percentage
Cmp_Medium	float64	Medium Passes Completed
Att_Medium	float64	Medium Passes Attempted
Cmp_percent_Medium	float64	Medium Pass Completion Percentage
Cmp_Long	float64	Long Passes Completed

Att_Long	float64	Long Passes Attempted
Cmp_percent_Long	float64	Long Pass Completion Percentage
xA_Expected	float64	**Expected Assists** (The likelihood each completed pass becomes a goal assists given the pass type, phase of play, location and distance)
A_minus_xAG	float64	Assists minus Expected Goals Assisted
KP	float64	**Key Passes** (Passes that directly lead to a shot)
Final_Third	float64	**Passes into Final Third** (Completed passes that enter the third of the pitch closest to the goal. Not including set pieces)
PPA	float64	**Passes into Penalty** (Completed passes into the 18-yard box)
CrsPA	float64	**Crosses into Penalty Area** (Completed crosses into the 18-yard box)
Live_Pass	float64	Live-ball Passes
Dead_Pass	float64	**Dead-ball** (Includes free kicks, corner kicks, kick offs, throw-ins and goal kicks)
FK_Pass	float64	Passes from Free Kicks
TB_Pass	float64	**Through Balls** (Completed pass sent between back defenders into open space)
Sw_Pass	float64	**Switches** (Passes that travel more than 40 yards of the width of the pitch)
Crs_Pass	int64	Crosses
TI_Pass	float64	Throw-ins Taken
CK_Pass	float64	Corner Kicks

In_Corner	float64	Inswinging Corner Kicks
Out_Corner	float64	Outswinging Corner Kicks
Str_Corner	float64	Straight Corner Kicks
Off_Outcomes	float64	Passes Offside
Blocks_Outcomes	float64	Passes Blocked
SCA	float64	**Shot-Creating Actions** (The two offensive actions directly leading to a shot, such as passes, take-ons and drawing fouls. Note: A single player can receive credit for multiple actions and the shot-taker can also receive credit)
SCA90	float64	Shot-Creating Actions per 90 minutes
PassLive_SCA	float64	Completed live-ball passes that lead to a shot attempt
PassDead_SCA	float64	Completed dead-ball passes that lead to a shot attempt
TO_SCA	float64	Successful take-ons that lead to a shot attempt
Sh_SCA	float64	Shots that lead to another shot attempt
Fld_SCA	float64	Fouls drawn that lead to a shot attempt
Def_SCA	float64	Defensive actions that lead to a shot attempt
GCA	float64	**Goal-Creating Actions** (The two offensive actions directly leading to a goal, such as passes, take-ons and drawing fouls. Note: A single player can receive credit for multiple actions and the shot-taker can also receive credit)

GCA90	float64	Goal-Creating Actions per 90 minutes
PassLive_GCA	float64	Completed live-ball passes that lead to a goal
PassDead_GCA	float64	Completed dead-ball passes that lead to a goal
TO_GCA	float64	Successful take-ons that lead to a goal
Sh_GCA	float64	Shots that lead to another goal-scoring shot
Fld_GCA	float64	Fouls drawn that lead to a goal
Def_GCA	float64	Defensive actions that lead to a goal
Tkl	float64	**Tackles** (Number of players tackled)
TklW_Tackles	int64	**Tackles won** (Tackles in which the tackler's team won possession of the ball)
Def 3rd_Tackles	float64	Tackles in defensive third
Mid 3rd_Tackles	float64	Tackles in middle third
Att 3rd_Tackles	float64	Tackles in attacking third
Tkl_Challenges	float64	**Dribbles Tackled** (Number of dribbles tackled)
Att_Challenges	float64	**Dribbles Challenged** (Number of unsuccessful challenges plus number of dribblers tackled)
Tkl_percent_Challenges	float64	**Percentage of dribblers tackled** (Dribblers tackled divided by number of attempts to challenge an opposing dribbler. Minimum .625 dribblers challenged per squad game to qualify as a leader)

Lost_Challenges	float64	**Challenges lost** (Number of unsuccessful attempts to challenge a dribbler)
Blocks_Blocks	float64	**Blocks** (Number of times blocking the ball by standing in its path)
Sh_Blocks	float64	**Shots blocked** (Number of times blocking a shot by standing in its path)
Pass_Blocks	float64	**Passes blocked** (Number of times blocking a pass by standing in its path)
Int	int64	Interceptions
Tkl+Int	float64	Number of players tackled plus number of interceptions
Clr	float64	Clearances
Err	float64	**Errors** (Mistakes leading to an opponent's shot)
GA	float64	Goals against
GA90	float64	Goals against per 90 minutes
SoTA	float64	Shots on Targets against
Saves	float64	Number of saves made
Save_percent	float64	**Save percentage** (Percentage of shots on target saved)
W	float64	**Wins** (Number of matches won)
D	float64	**Draws** (Number of matches drawn)
L	float64	**Losses** (Number of matches lost)
CS	float64	**Clean sheets** (Number of matches with no goals conceded)

CS_percent	float64	**Percentage of clean sheets** (Percentage of matches with no goals conceded)
PKA_Penalty	float64	Penalty kicks allowed
PKsv_Penalty	float64	Penalty kicks saved
PKm_Penalty	float64	Penalty kicks missed
Save_percent_Penalty	float64	**Penalty save percentage** (Penalty kick goals against/penalty kick attempts penalty shots that miss the target are not included)
FK_Goals	float64	Free kick goals against
CK_Goals	float64	Corner kicks goals against
OG_Goals	float64	Own goals scored against goalkeeper
PSxG_Expected	float64	**Post-Shot Expected Goals (PSxG)** (Expected goals based on the quality of shots on target faced)
PSxG_per_SoT_Expected	float64	**Average PSxG per shot on target faced** (Average quality of shots on target faced)
PSxG+_per__minus__Expected	float64	**PSxG minus goals allowed** (Positive values suggest better shot-stopping than expected)
_per_90_Expected	float64	**PSxG+/- per 90 minutes** (Normalizes PSxG differential per 90 minutes.)
Cmp_Launched	int64	**Number of launched passes** (over 40 yards) completed
Att_Launched	int64	Number of launched passes attempted.
Cmp_percent_Launched	float64	Completion percentage of launched passes.
Att (GK)_Passes	float64	Passes attempted (Not including goal kicks)
Thr_Passes	int64	Throws attempted

Launch_percent_Passes	float64	Percentage of goals kicks that were launched (Passes longer than 40 yards)
AvgLen_Passes	float64	Average length of all passes in yards
Att_Goal	float64	Goal kicks attempted
Launch_percent_Goal	float64	Percentage of goal kicks that were launched (Passes longer than 40 yards)
AvgLen_Goal	float64	Average length of goal kicks in yard
Opp_Crosses	float64	Crosses faced (Opponent's attempted crosses into penalty area)
Stp_Crosses	float64	Number of crosses into penalty area which were successfully stopped by the goalkeeper
Stp_percent_Crosses	float64	Percentage of crosses into penalty area which were successfully stopped by the goalkeeper
OPA_Sweeper	float64	Number of defensive actions outside of penalty area
OPA_per_90_Sweeper	float64	Defensive actions outside of penalty area per 90 minutes
AvgDist_Sweeper	float64	**Avg. Distance of Def. Actions** (Average distance from goal (in yards) of all defensive actions)
2CrdY	int64	Second yellow card
Fls	int64	Fouls committed
Fld	int64	Fouls drawn
Off	int64	Offsides
Crs	int64	Crosses
PKwon	float64	Penalty Kicks Won
PKcon	float64	Penalty Kicks Conceded
OG	int64	Own goals

Recov	float64	Ball recoveries
Won_Aerial	float64	Aerials won
Lost_Aerial	float64	Aerials Lost
Won_percent_Aerial	float64	Percentage of Aerials won (Minimum .97 aerial duels per squad game to qualify as a leader)

5.3 Feature Engineering and Composite Metrics

After data cleaning and preprocessing, feature engineering becomes an extremely important stage in consolidating the player data into valuable composite indicators that capture subtle cues of player behaviors. This section introduces the motivations, formulations, and implementations of designing valuable composite indicators to enable our proposed machine learning-based player recommendation system.

5.3.1 Rationale behind Composite Metrics

While being incredibly informative, raw performance data obtained from FBref creates a set of difficulties for machine learning algorithms to process. Firstly, the data set includes over 100 distinct variables, most of which correlate heavily with each other, thereby creating multicollinearity problems while using regression models to analyze the data [19]. Secondly, the set of judgment criteria is distinct for different roles; the indicators of quality for a forward completely differ from those of a defender [20]. Thirdly, less developed leagues, being the target group of interest, necessitate variables that are easily interpretable while being able to combine various features of data [21].

Empirical studies conducted on football analytics show that aggregate models are very effective and easy to interpret. Empirical results show that well-designed features retain strong accuracy even when the feature space is pruned by nearly 48.8% [22]. Finally, feature designs for players are essential for players of different roles to be accurately assessed [23].

5.3.2 Core Universal Composites

The following composites are calculated for all outfield players, independent of playing position, as they represent core aspects of soccer ability.

5.3.2.1 Offensive Output Index (OOI)

Offensive Output Index: It calculates an individual’s overall attacking contribution by combining attacking threat with creative contribution:

$$\text{OOI} = (\text{npxG} + \text{xAG}) / \text{Mins\_Per\_90}$$

Here, npxG is Non-Penalty Expected Goals, xAG is Expected Assisted Goals, and Mins_Per_90 is normalized to a 90-minute basis.

The above index is a combination of modern football analytics that values overall attacking performance by combining goals and assist values [24]. Compared to traditional measures of goals and assist performance, where the end result is dependent on the ability of other players on the team to convert shots on goal, the OOI is an index that measures overall attacking performance regardless of the end result. A player with high OOI is always producing high-quality attacking moves, with or without actually scoring goals.

5.3.2.2 Defensive Contribution Index (DCI)

"Defensive Contribution Index" is a measure of the most important defensive plays that receive official recognition in football data analytics. It involves the following:

$$\text{DCI} = (\text{Tkl} + \text{Int} + \text{Clr} + \text{Blocks}) / \text{Mins\_Per\_90}$$

where Tkl represents tackles, Int represents interceptions, Clr represents clearances, and Blocks represents blocked shots or passes.

Based on the above equation, this indicator is grounded in the CBIT (clearances, blocks, interceptions, and tackles) paradigm introduced by the Premier League in the 2025/26 season to formally account for the four major defensive actions deserving credit in the context of player performance [25], [26]. That the Premier League embraced this paradigm highlights the importance attributed to the above actions in measuring the value of a player’s workload.

5.3.2.3 Ball Progression Index (BPI)

The Ball Progression Index measures how well a player is able to move the ball towards the opposing goal:

$$\text{BPI} = (\text{PrgC} + \text{PrgP} + \text{PrgR}) / \text{Mins\_Per\_90}$$

PrgC refers to progressive carries, PrgP refers to progressive passes, while PrgR refers to progressive passes received.

Studies on Expected Threat (xT) have found progressive actions to be most useful to footballers as they move the ball towards the opposing team’s goal [27]. There is also a linear relationship between progressive actions as described by FBref as actions of at least 10 yards towards the opposing team's goal or any other form of action inside the penalty area and the creation of scoring chances. BPI is very useful when analyzing the role of midfielders or fullbacks.

5.3.2.4 Shooting Efficiency Index (SEI)

The Shooting + Efficiency Index measures both the quality and efficiency of shooting performance of an athlete:

SEI = (G_per_SoT × SoT_percent) + (npxG_per_Sh × 0.5)

Here, G_per_SoT signifies goals per shot on target, SoT_percent is shots on target percentage, and npxG_per_Sh is non-penalty expected goals per shot.

The above metric combines three base qualities essential for shooters: Shooters with strong conversion abilities and high shot location will rate higher on this metric than shooters with strong shot location but poor conversion abilities. The weighted formula gives more emphasis to the abilities connected with the end result—conversion and accuracy (100%)—than shot location alone (50%). The results have shown that shot metering abilities can be expressed with metrics that consider shot location, conversion rates, and(expected) goals models [24].

5.3.2.5 Passing Effectiveness Index (PEI)

The Passing Effectiveness Index combines pass accuracy with creative and progressive play:

PEI = (Cmp_percent × 0.3) + (KP / Att × 100 × 0.4) + (Final_Third / Att × 100 × 0.3)

Here, Cmp_percent stands for completed pass percentage, while KP is key passes, with Att referring to total pass attempts, and Final_Third indicating total passes into the third third.

The above formula takes a nuanced approach to measuring the playmaker's possession play strategy in being either cautious and possession-oriented or more offensive with regard to creativity in play-making. In fact, the 40% significance for key pass frequency is based on a study showing that ‘the quality of chance creation is the key difference between playmakers, not how often they create chances’ [28]. The other measures, at 30%, pick important indicators to allow for grading in regard to play completion success or advancement into the third third. The above formula relies on the notion of Efficient Possession Ratio to express how possession can result in scoring chances [29].

5.3.3 Position-Specific Efficiency Metrics

Position-specific metrics recognize the fact that different positions focus on different sets of skills. Evidence suggests that feature selection with a focus on position-specific features can significantly improve the accuracy of ratings [20], [30].

5.3.3.1 Forward/Winger Efficiency

Players regarded as forwards and wingers will be rated for both scoring and dribbling abilities:

$$\text{PSE\_FW} = (\text{npxG} / \text{Sh}) \times (\text{Succ\_Take} / \text{Att\_Take}) \times 100$$

This metric is shot quality (non-penalty expected goals per shot) times dribbling success rate. In terms of evidence, it is clear that it is not goal-scoring ability that separates high-level forwards, but their ability to get shots off from good spots [31]. Adding dribble success rate allows one to successfully beat an opposition defender, an aspect of being a good forward that was identified in previous work [32].

5.3.3.2 Midfielder Efficiency

For midfielders, effectiveness combines creative quality, passing reliability, and progressive intent:

$$\text{PSE\_MF} = (\text{xAG} / \text{KP}) \times (\text{Cmp\_percent} / 100) \times (\text{PrgP} / \text{Att})$$

The multiplicative model requires all three attributes to be high to produce a high score. In elite midfielders, there is a strong emphasis on progressive passing, the ability to breakthrough and break down opposing defenses [33]. The model selects for midfielders who create high-quality scoring opportunities (high expected assists per key pass), those who retain the ball (high pass completion percentage), and those who progress the ball (high progressive pass ratio).

5.3.3.3 Defender Efficiency

For center-backs, success is defined by the following success rate that punishes errors:

$$\text{PSE\_DF} = (\text{Tkl} + \text{Int}) / (\text{Tkl} + \text{Int} + \text{Lost\_Challenges} + \text{Err} + 1)$$

This ratio establishes the success rate, where the numerator indicates successful defensive plays, while the denominator encompasses successes and failures. Studies highlight the importance of tackles and interceptions in assessing defensive players, where interceptions represent good positioning and game intelligence [34]. The positioning of the errors within the denominator is critical, since defensive errors often result directly in scoring goals, and one error cancels out several successful plays.

5.3.3.4 Fullback Efficiency

For Fullbacks, effectiveness means having capabilities that cover their roles in both defense and offense:

PSE_FB = (Tkl + Int) × 0.5 + PrgC × 0.3 + (Crs / Mins_Per_90) × 0.2

Assigning 50% to defensive actions represents the importance of defending, while 30% and 20% assigned to progressive carries and crossing frequencies reflect the importance of fullbacks' contributions to offenses in modern football. Current trends in evaluating fullbacks' performance have evolved to incorporate progressive carries and crossing frequencies together with defense [35].

5.3.4 Project-Critical Composites

These composites specifically relate to the key contributions to the project by the team, namely alignment with playing style, cost-effective selection, and evaluation of performance dependability.

5.3.4.1 Team Fit Score (TFS)

eam Fit Score The Team Fit Score reflects how well an individual player's playing style fits a team's playing style in terms of:

```
from sklearn.metrics.pairwise import cosine_similarity

from sklearn.preprocessing import StandardScaler



# Normalize player and team vectors

scaler = StandardScaler()

player_vector = scaler.fit_transform([[OOI, DCI, BPI, PEI, SEI, ...]])

team_vector = scaler.transform([mean_of_current_team_players])



# Calculate similarity

TFS = cosine_similarity(player_vector, team_vector)[0][0]
```

The TFS returns a score in the range [0, 1], with larger scores suggesting better stylistic fit. This measure tackles the project's expectation of achieving “team playstyle matching” very succinctly. There is empirical proof that players who played under similar tactical schemes are better at being successfully transferred compared to those matched based solely on statistical data [36]. Also, the concept of player/team matching using the cosine measure was proven useful for recommendation systems, especially for finding players with untapped potential fitting particular playstyles [37].

The team vector is derived from calculating the normalized mean of the composite metrics for all team members in possession, thereby obtaining a characteristic that reflects the play style that this team exhibits. Cells that are defensive-oriented have high average Defensive Coherence Index and low average Defensive Opportunity Index, while cells that are attacking-oriented have the opposite characteristic attributes. The TFS enhances the ability to predict integration in logistic regression in terms that are tactical in nature.

5.3.4.2 Market Value Efficiency (MVE)

Market Value Efficiency (MVE) points out players that offer strong performance given their expenses:

MVE = (OOI + DCI + BPI) / (transfer_fee / 1,000,000)

The higher the MVE score, the stronger the efficiency. In the football market literature, large market inefficiencies are found that do not exactly capture player performance through their transfer prices [38]. Analytics allow clubs to successfully exploit these inefficiencies and even notice possible arbitrage results [39]. In research studies on lower-budget football clubs, statistical analysis proves that it is possible to correctly determine players with high-quality performance that were initially undervalued [40].

The numerator evaluates the total score for three universal parameters for performance: offense, defense, and progressive. Division by the fee amount in millions of euros gives the parameter for the level of performance per million. This means that for example, an MVE = 1.5 will display 50% level of enhanced performance compared to an MVE = 1.0.

5.3.4.3 Consistency Score (CS)

Consistency Score measures the level of performance consistency based on the Coefficient of Variation (CV):

```
import numpy as np
```

```
# Calculate for each player across recent matches
performance_values = [xG_match1, xG_match2, ..., xG_matchN]
mean_performance = np.mean(performance_values)
std_performance = np.std(performance_values)

# Coefficient of Variation
CV = std_performance / mean_performance

# Convert to consistency score (higher = more consistent)
CS = 1 - CV
```

The Coefficient of Variation is a relative measure of variation that enables easy comparison between metrics and players [41]. The smaller the values of CV, the more consistent the performance is. Using the definition CS = 1 - CV, a consistency score is obtained where higher scores indicate higher consistency.

Studies have shown consistency of performance as a key identifying factor for the evaluation of players. Player ratings analyzed using the coefficient of variation (CV) provide effective distinction of players with consistent and fluctuating levels of performance [42]. In the context of clubs with limited resources, consistency of performance becomes absolutely necessary since the club cannot entrust players whose levels of performance vary greatly from match to match [43]. The Contextual Score helps our linear regression module by improving the precision of predictions regarding reliability of future performance.

5.3.4.4 Age-Adjusted Performance (AAP)

The Age-Adjusted Performance measure takes the player’s age into consideration during the assessment of performance:

```
def age_multiplier(player_age):
        if 18 <= player_age <= 21:
        return 1.2  # Young with growth potential
        elif 22 <= player_age <= 27:
          return 1.0  # Peak performance years
```

```
elif 28 <= player_age <= 32:

return 0.85  # Declining but experienced

else:  # 33+

return 0.7  # Veteran
```

AAP = performance_metric × age_multiplier(age)

This adjustment takes into account that younger players have potential for development and resale value, while older players may decline. The study on valuation of players indicates that age has a negative correlation with valuation, especially for players over 27 years old [44]. Resource-constrained clubs will generate higher value through investment in players for whom growth can be expected as against players whose effectiveness and value continue to decline with age.

The multipliers are derived from established age-performance functions in football studies, in which peak performance is generally reached between the ages of 22 and 27, followed by a decline in performance over time [45]. The multiplier of 1.2 for the young players is due to both the development potential as well as the resale value.

5.3.4.5 Defensive Reliability Index (DRI)

It is typically used to judge defensive performance, focusing on successful defensive results while deducting heavily for any errors:

DRI = (Tkl + Int) / (Lost_Challenges + Err + 1) × (Mins_Per_90 / 90)

This ratio formula expresses a rate of success in which the numerator represents successful defensive instances and the denominator represents failure instances and errors. The “+1” in the denominator eliminates dividing by zero when a player has limited defensive engagements. The normalization factor is Mins_Per_90, normalized to 90 minutes.

It is essential to note that research on defensive metrics indicates that simply relying on the number of tackles or interceptions is not useful to measure the quality of defense; rather, error percentage is also required to be measured for the defender [34]. The InStat Index involves measurements of tackles, aerial duels, and interceptions as decisive factors to judge the quality of center backs or defensive midfielders [46]. The DRI calculates the success-to-failure ratio to determine reliable defenders who make constructive contributions rather than error-prone mistakes.

5.3.4.6 Aerial Dominance Metrics

The presence of aerial awareness is crucial for certain roles, such as center back and target forwards. We split aerial ability into two measures that complement one another.

Aerial Win Rate (AWR):

$$\text{AWR} = (\text{Won\_Aerial} / (\text{Won\_Aerial} + \text{Lost\_Aerial})) \times 100$$

The Aerial Win Rate (AWR) is the percentage of aerial duels won. The efficiency indicator points out those players who succeed when they engage in aerial duels. Suppose a player has AWR = 75%. Then he/she wins three out of every four aerial duels.

Aerial Involvement (AI):

$$\text{AI} = (\text{Won\_Aerial} + \text{Lost\_Aerial}) / \text{Mins\_Per\_90}$$

Aerial Involvement measures how often a player participates in aerial battles per 90 minutes, irrespective of the result. This measure identifies players who are deeply involved in aerial actions.

Evidence also suggests that the proportion of aerial duels won is indicative of both their rate and efficiency [47]. We can thus make use of these categories without facing the pitfalls presented by a composite score. A player with 90% success in 2 aerial duels per game is very different from a player who wins 60% of 10 duels per game. While the former might be a technical expert picking his duels, the latter would be a physical presser who wins most aerial duels.

This type of bifurcation is even more relevant in physical leagues like the Premier League and the Bundesliga, in which the ability to play in the air often marks the difference between successful players and others who are less successful [47]. In the recommendation system, the clubs could assign particular weights to these factors based on the style of play—those who play direct football could emphasize the need for high AI, while possession-based teams could emphasize the need for high AWR with moderate AI.

5.3.4.7 Tactical Versatility Index (TVI)

The Tactical Versatility Index measures the level to which a player can proficiently play:

```
# Count positions where player has significant experience

competent_positions = count_positions_where(minutes_played > 450)

total_possible_positions = 11  # or formation-specific
```

TVI = (competent_positions / total_possible_positions) × 100

A player is considered proficient at a particular position if they have exceeded 450 minutes (about five full games' worth of time) at the position under evaluation. The Track Value Indicator (TVI) is shown by a percentage of the total positions available.

Studies using data-driven performance assessment prove the importance of positional flexibility in player appraisal. Smaller clubs find it most helpful to have players with the flexibility to play various other positions. It is beneficial to have players with the capability to play as center-backs, right-backs, and defensive midfielders. The TVI value, given by 27% (3/11), is of great use to smaller clubs that can't maintain an extensive squad of players with dedicated substitutes.

The TVI supports the recommendation engine by pointing out the players essential to squad depth and flexibility. As such, clubs that are concerned about injuries or have limited budgets find flexibility an essential element to consider. This measure can be calculated by using denominators that correspond to formations, such as 10 for a 4-3-3 formation.

5.3.5 Implementation and Validation

Composite metrics are all implemented in Python and use NumPy and Pandas libraries. The calculation pipeline is as follows:

1. **Data Validation**: Check for missing values on component metrics
2. **Normalization**: Use z-score standardization in position groups
3. **Composite Calculation**: Calculate the metrics based on the following formulas.
4. **Outlier Detection**: Identify values exceeding 3 standard deviations from the mean.
5. **Storage**: Save the computed composites in the database, alongside raw statistics.

The composites are checked for validity through correlation analysis to ensure that the composites represent distinctive dimensions. Correlation matrices show that no two composites have a correlation coefficient with a value of more than 0.7.

The reason is that position-specific composites are generated only for players in their respective positional groups. For instance, PSE_FW is generated only for forwards and wingers, while only center back players have their dribbling success rate measured by PSE_DF. In fact, it is inappropriate to rate a center back on passing and a forward on clearances.

5.3.6 Feature Selection and Dimensionality

The essential feature set for the hybrid machine learning model includes:

- Five universal composites (OOI, DCI, BPI, SEI, PEI)
- Four position-specific composites (PSE variants)
- Seven key composite indicators (TFS, MVE, CS, AAP, DRI, AWR, AI, TVI)
- Selected raw features (age, minutes played, position)

This configuration results in roughly 18-22 features per player, depending on their role. Regarding feature selection, SHAP (SHapley Additive exPlanations) analysis of previous work suggests an approximate optimum of 20 features when models forecast market value of players [47]. The range of 18-22 seeks to provide an optimal trade-off between accuracy of models and explainability, hence offsetting the curse of dimensionality.

The feature engineering strategy works well in extracting a manageable set of meaningful composites from the original set in excess of 100 raw statistics. It has been observed in empirical studies that by proper feature engineering, significant accuracy can be maintained despite a drastic reduction in features [22]. The following are advantages associated with the approach based on composites:

1. **Interpretability**: Each composite has explicit semantic meaning
2. **Reduced Multicollinearity**: Composites combine correlated raw features
3. **Position-Specific Evaluation**: The criteria for evaluation are position-specific
4. **Stakeholder Alignment**: Metrics address budget and fit concerns
5. **Model Performance**: Engineered features improve predictive accuracy

The composites represent the building block of the recommendation engine and enable the promotion of a combination of machine learning models, including the application of logistic regression to determine player success, linear regression to predict other performance-related metrics, and the usage of the K Nearest Neighbors technique to discover similar players. Thus, the proposed model helps to translate basic data points to meaningful indicators based on position and stakeholders.

5.4 Data Sources and Raw Data Format

Raw football data will be extracted using FBref, and it will be kept in multiple csv files, each of which corresponds to a specific category of statistics. We will merge these csv files since they will not come in a single table. In addition, we will consider the use of the FC25/FIFA source to include the player data with position and roles.

5.4.1 FBref Player Statistics Files

The player-level data from FBref is maintained through 11 files, each of which corresponds to a category of statistical information. The major statistical categories that are used are:

- Standard
- Passing
- Passing Types
- Shooting
- Goal and Shot Creation (GCA/SCA)
- Defensive Actions
- Possession
- Playing Time
- Miscellaneous
- Goalkeeping
- Goalkeeping (Advanced)

5.4.2 FBref team files

Data from the team/squad FBref files is also saved in 11 files, categorized the same as the player files but team-level data. The goal is to organize the data in a team-season per row manner.

5.4.3 External FC25

As a feature enhancement for FBref player data, we employ two additional resources in Position_Mapping.ipynb:

Table 19: Data Sources

Dataset	**Used in code as**	**Source**	**Main purpose**
player_full.csv	players_fb	Results from MergeDF notebook	Starting data to supplement with roles and positions

new-players-data-full.csv	players_fif	Kaggle: EA Sports FC 25 real player data (SoFIFA merge)	Provides positions and preferred foot; matched by [name, club]
merged_players_fc25.csv	players_fif_roles	Scraped from the website using a SoFIFA web scraper (on GitHub)	Creates role for players; data linked from Kaggle to FBref players

These external datasets are not guaranteed to include all players in FBref, and so we use matching, propagation, and imputation to account for missing data.

5.5 Tools, Libraries, & Environment

Preprocessing was done using Python with the use of the pandas library. These are the libraries that we used:

- pandas and numpy for loading CSV files, data cleaning, grouping, merging, and outputting results.
- unidecode: removal of accent marks and standardization of names for similarity comparison.
- rapidfuzz: fuzzy string match for FBref names vs. FIFA/FC25 names.
- scikit-learn: StandardScaler and KNeighborsClassifier for KNN-based imputation.

5.6 Data Profiling and Initial Quality Checks

Before proceeding with the transformations, we carried out quick checks on the nature of the data. This is necessary because football databases often involve inconsistencies, missing values, and duplications.

- Column inspection: assessing the columns across all files to verify identifiers and available data.

- Uniqueness checks: Assesses how many unique players or teams there were prior to the merging as well as after.

- Duplicate checks: to check if a player-season exists multiple times (usually because a spelling variation for a team name exists).

- NaN value checks: counting NaNs per column and a heatmap of missing data.

5.7 Data Cleaning and Standardization

5.7.1 Normalizing identifiers (names and teams)

A typical reason related to failed merge operations is that there are identical players or clubs with slightly differing string representations (capitalization, additional spaces, accented letters). To avoid these errors, text identifiers are normalized by:

- Convert values to string, remove spaces, and convert to lowercase.
- Removing accent marks with unidecode (e.g. 'Muller' vs 'Müller')
- Removing punctuation and special characters when generating match keys (Position_Mapping.ipynb).

5.7.2 Standardizing column headers and eliminating noise columns

The column names are preprocessed by removing any whitespaces and characters that might cause naming inconsistencies. In addition, some CSV files contain unnamed indexes like 'Unnamed: 0,' which are removed as they do not play any role as features.

5.7.3 Standardizing club names with a correction dictionary

Nonetheless, some teams sometimes have short names such as 'manchester utd' or other conventions. To remedy this disparity when grouping or matching, we use a manual dictionary to map common aliases to their standard name. In this respect, the process is more crucial to the external integration task since it uses the match parameter `[name, club].`

5.8 FBref Player Data Integration (MergeDF)

5.8.1 Merge keys and data set alignment

To merge the 11 player statistics files into a common table, the files utilize three common identifiers as merge keys, which are:

Table 20: Player Merge keys

Merge key	Meaning
Player	Player name (normalized).
Team	Name of club/team (Renamed FBref ‘Squad’ to ‘Team’).
Season_End_Year	This variable allows separation between player seasons.

These keys signify a single, unique record of a player in a season for a given team. Each table of players is synchronized on these keys.

5.8.2 Handling duplicates before merging

In some files, the rows involving the same (Player, Team, Season_End_Year) are identical. We don’t remove the duplicates but handle the aggregation of the duplicates.

- The numeric metrics are categorized into two types: count-like metrics and rate-like metrics.
- Count methods are added together (for example, total tackles, passes, shots).
- “Rate-like metrics are averaged (e.g., per-90 or percentage metrics).”
- The rate-like metrics are extracted by column-name identifiers like: %, 90, per, rate, avg.
- categorical columns (like Nation, Pos, etc.) retain the first available value.

5.8.3 Outer merge over 11 categories

We then perform an outer join to combine all 11 player tables that we have cleaned and aggregated. The use of an outer join in this context is crucial in allowing us to retain all player records when some categories are irrelevant to other players, such as goalkeeping attributes for outfield players.

To prevent column name collisions, most of these metric columns are suffixed using a short label that identifies the category of origin (e.g., xG_Shoot, CrdY_Misc). A reference file is also chosen to retain these metadata columns without any suffix whenever possible.

5.8.4 Making column names unique and eliminating redundant suffixes

After the huge merge, there might be overlapping column names. To ensure that there are no duplicates, we check the column names and, when there's an overlap, we add an incrementing suffix (_1, _2, .). However, if there's an overlap and the metric occurs just once in the entire dataset, we remove the unwanted suffixes and keep the metric as the original name. For example, if xG_Shoot is the sole xG metric within the dataset, it can be shortened to xG. However, when there's more than one version of the same metric within the dataset (for example, Gls_Standard and Gls_Shoot), we retain the suffixes since they define different things.

5.8.5 Consolidating descriptive columns

The descriptive column names like Comp, Nation, Pos, Age, Born, Url would be in various forms after the merge. We find the first available non-null value for these descriptive column names from the suffixed versions and then eliminate the redundant duplicates. In this way, only one ‘official’ column is left for every descriptive column name.

5.8.6. Manual consolidation of repeated metrics

A small set of metrics is repetitious across various categories with the same interpretation. In the notebook, this data is combined into one column through the use of the first non-null value from various sources. The following sets of data are combined by the code: Mins_Per_90, CrdY, CrdR.

5.8.7 Handling missing values for players

Missing values are expected in the merged dataset, as some stats are not relevant to all players. Missing values exist in numeric stats columns but are retained in identity metadata when appropriate.

- All the numeric stat columns are filled with 0 because it often means the statistic doesn’t apply or the player has 0 events.
- MetaData elements like Age, Born, & Season_End_Year are not forced to 0 (remains same) as there can be inaccurate identity details.
- The combined table is sorted on the fields (Player, Season_End_Year) for ease of examination and use.

5.9 FBref Team Data Integration (MergeDF_Final)

5.9.1 Team cleaning and merge keys

For team data, the primary objective is to create one row per team per year. The team merge keys are:

Table 21: Team Key Merge

Key Merge	**Definition**
Team	Team name (FBref 'Squad' renamed to 'Team', normalized).
Season_End_Year	Season end year

5.9.2 Cleaning steps and duplicate handling

Before the merging, the file for each team undergoes certain processing. The major operations include:

- rename Squad to Team if required.
- If a column named Team_or_Opponent exists in the DataFrame, it will retain only those rows where Team_or_Opponent = 'Team' and
- Normalize Team strings via unidecode, lower, and trimming.
- Convert Season_End_Year into integer.

- If there are dupes with the same (Team, Season_End_Year), then group on keys and accompanying metadata (Comp, Nation, LgRank if available) and take averages on numeric columns. This retains league information (Comp) while eliminating dupes.

5.9.3 Outer merge and preventing _x/_y conflicts

The team data is combined through outer categories as well. In order to prevent _x/_y duplicates, prior to a merge, columns existing within the present merged dataset will be dropped from the arriving data frame (excepting the merge keys) prior to merging on (Team, Season_End_Year) with how='outer'.

5.9.4 Final cleanup for teams (snake_case standardization)

We then complete the merge of all team categories, with NaNs replaced with 0 in numeric fields and sorting along (Team, Season_End_Year). Finally, we polish with the removal of unneeded index columns and any duplicated data, with column names normalized and changed to snake_case. "Gls/90" would then be renamed "Gls_per_90" for ease of coding and mapping.

5.10 External Player enrichment (Position_Mapping)

After the construction of the FBref universal player dataset (player_full.csv) [69], we add positions, preferred foot, and roles from the FC25/FIFA datasets. All the aforementioned is facilitated in Position_Mapping.ipynb.

5.10.1 Active player filtering

The enrichment phase is concerned with active players. A player is labeled as active if they are found in the most recent season (Season_End_Year = 2025). This is calculated by grouping the records that contain 2025. Active players alone are saved in the enrichment phase.

5.10.2 Creating Matching Keys (clean_name and clean_club)

To enable the integration of FBref players and FIFA players, keys were created in both datasets. From the FBref data, the clean_name variable is derived from the Player variable, and clean_club from the Team variable. In the Kaggle

FIFA data [68], the clean_name variable is derived from the full_name variable, and the clean_club variable from the club_name variable. The process followed in the data cleaning uses the removal of accents, conversion to lowercase, removal of spaces, and removal of punctuation.

We also make use of a "manual club correction dictionary," such as 'psg' to 'paris saint germain,' to account for the differences in the names of some clubs.

5.10.3 Preparing roles from scraped FIFA data

Roles are derived from the SoFIFA dataset (merged_players_fc25.csv) [69]. The roles column in this dataset contains more than one role labels/codes. For our analysis, only roles that contain “+” sign are considered valid roles from this dataset, and these are merged into the Kaggle FIFA dataframe after cleaning. The merging criteria here are player_id, then full_name columns in case player_id fails.

5.10.4 Matching engine (Two Phase)

The player matching is performed through two phases, which provide balance between coverage and accuracy:

- Phase 1(Club + Name): When there are mutual clubs, assign FBRef players to the same club. Try to assign using exact matching on clean_name first. As an alternative, fuzzy matching with RapidFuzz with a similarity score of 65 or more.
- Phase 2 (Global Search): for players that are still missing data, specifically roles, we will look through all FIFA names. As there's no club data to verify, we will use a more strict filter (score >= 90).

If there is a match, we copy three fields from FIFA to FBref: positions (fc26_pos), preferred_foot (fc26_foot), and roles (fc26_roles).

5.10.5 Propagation of FIFA attributes through seasons

After the matching, a player could have their FIFA features for one row but no features for the other rows of a season. To address the issue of missing data while maintaining consistency, the FIFA features for player groups (Url) are propagated. The forward-fill and backward-fill operations are applied for the rows of the player for 'fc26_pos', 'fc26_foot', and 'fc26_roles'. This indicates that if the player matched in a season, the same features are assigned to all seasons for that player.

5.10.6 Determining a final position field

In the dataset, there is also the final position field (final_pos). If the FIFA positions are present in the dataset (fc26_pos), then the FIFA positions are used. Otherwise, if the FIFA positions are not present in the dataset, then the coarse position (Pos) from FBref along with the preferred foot, if any, is used.

- DF → LB if left-footed, RB if right-footed, otherwise CB.
- MF -> CDM; MF,FW -> CAM; FW -> ST; FW,MF -> RW; GK -> GK.
- If none of the above rules apply, we retain the FBref Pos value.

5.10.7 Cleaning and formatting the roles field

The roles string after matching is cleaned up, removing the 'nan' literal, handling the comma/space normalization, and stripping excess leading/trailing commas. This yields a uniform Roles field for the downstream split/modeling.

5.10.8 Missing role imputation with KNN

Since the roles scraped data does not account for all players, there are some active players who do not have any roles assigned. To rectify this problem, we employ a KNN-based imputation technique as follows:

- Features: (1) small set of available performance statistics (e.g., Goals, Assists, Expected Goals, defensive actions, passing metrics), and (2) one-hot encoded final position dummies (Pos_*)
- Scaling: The scaling of numeric state features is carried out through StandardScaler.
- Model: KNeighborsClassifier with n_neighbors = 1 and Manhattan distance.
- Output: The predicted role is assigned along with a marker '(Est)' indicating that it is estimated.

This is a very effective way of fulfilling the vacant roles. It is also quite simple.

5.10.9 Splitting multi-valued fields (positions and roles)

Some also have several values in a single cell, such as several positions or roles listed in a single cell and separated by commas). For easier treatment, we split them into different columns:

- Positions -> main_pos, secondary_pos_1, secondary_pos_
- Roles -> role_1, role_2, role_

Then, the temporary processing columns are removed, including clean_name, clean_club, is_active, fc26_pos, fc26_roles, as well as the position dummy columns, while the columns are finally named as follows: final_pos -> Positions, roles -> Roles, fc26_foot -> Preferred foot.

5.11 Output Datasets

After the end of the preprocessing step, the below-mentioned files are output to be used in the subsequent chapters:

Table 22: Final Datasets

Output file	**Produced by**	**Description**
player_full.csv	MergeDF Notebook	The merged DataFrame from the previous step contains the unified FBref player dataset along with the merged statistics for 11 categories.
team_merged.csv	MergeDF notebook	Merged FBref team data with combined team stats; one for each team/season; cleaned and snake_case columns.
players_integration.csv	Position_Mapping notebook	Active-player subset with Positions, Preferred foot, and Roles; role_1, role_2, … and main_pos/secondary positions.

5.12 Validation and Consistency Checks

We verified the preprocessing results with various consistency checks:

- Duplicate inspection: grouping by (Player, Season_End_Year) to identify any duplicate records.
- Consistency in team names: verifying whether a player has been part of many different versions of a team name that could have spelling variations.
- Missing value reports: printing the number of NaNs and missing data visualization (Player dataset heatmap).
- Schema sanity: It is essential to ensure that crucial columns, including Comp (which represent the league), are maintained in the team data set.

5.13 Summary

Overall, the preprocessing pipeline takes several raw FBref CSV files and combines these into two cleaned datasets, one for players and one for teams. We minimize merge issues by normalizing field names, dealing with duplicate rows via aggregation strategies, as well as utilizing outer joins to avoid losses. We also enhance active player information with FC25/FIFA position and role information via a two-pass matching engine (exact followed by fuzzy matching). Missing values for FIFA attributes will be minimized via propagation, while missing role information is resolved via a simple imputation process based on kNN.

Chapter 6
Implementation

This chapter details the software architecture, algorithmic design, and system integration of the Football Intelligence Suite. The system is constructed as a modular Python application, orchestrating a pipeline that transitions from raw data ingestion to predictive modeling and interactive analytical dashboards. The architecture leverages a specific technology stack comprising Streamlit for the frontend, PostgreSQL for data storage, and Scikit-Learn/XGBoost for the predictive core.

6.1 Data Processing Pipeline

Data integrity is maintained through a three-stage ETL (Extract, Transform, Load) process defined in the system's ingestion scripts (process_weights.py and ingest_data.py).To account for performance variance across multiple seasons, the system does not rely on a simple arithmetic mean. Instead, the process_weights.py module implements an exponential decay function. For a set of player statistics over n seasons, the weight w for a specific season is calculated as:

Where λ is the decay factor (set to 0.75 in the configuration) and Δt is the number of years from the current season. This methodology ensures that recent performance contributes significantly more to the player's profile than historical data, while still retaining the statistical significance of a multi-year career.

6.1.1 Feature Normalization and Transformation

In the training pipeline (train_model.py), specific transformations are applied to handle the heterogeneity of football data:

- Financial Log-Transformation: Analysis of the market_values.csv and wages.csv datasets revealed a heavy right-skewed distribution. To stabilize the gradient descent during training, the np.log1p (Natural Logarithm of $1+x$) function is applied to Market Value and Wages. This prevents outliers (e.g., superstars with valuations exceeding €100M) from disproportionately biasing the model.
- Playstyle Encoding: Team tactical profiles (e.g., "Gegenpressing") are categorical variables. These are converted into machine-readable vectors using One-Hot Encoding (pd.get_dummies), resulting in binary feature columns (e.g., Style_Possession, Style_Counter).
- Entity Resolution: A custom string normalization routine (clean_team_name) utilizes fuzzy matching logic to reconcile naming discrepancies between the Scouting dataset and the Elo dataset (e.g., mapping "Bayer 04" and "Leverkusen" to a unique key).

6.2 The Intelligence Core (Backend)

The backend logic is distributed across three specialized engines trained in train_model.py, alongside a rule-based chemistry engine.

6.2.1 Hybrid Impact Engine

The system employs an Ensemble architecture to predict transfer outcomes. The training process splits the target variable (Elo Change) into two distinct prediction tasks:

1. Regression Task (Magnitude): An XGBoost Regressor is trained to minimize the Mean Absolute Error (MAE) of the projected Elo change. The model utilizes 300 estimators with a maximum depth of 6 to capture non-linear relationships between player metrics (e.g., npxG_Per, PrgP_Per90) and team performance.
2. Classification Task (Probability): A Logistic Regression classifier is trained simultaneously on binary targets (1 if $\Delta Elo > 0$, else 0). This outputs a probability score ($P_{success}$), serving as a risk-assessment metric.

6.2.2 Player DNA Classifier

To determine a player's "true" positional profile independent of their listed role, a Multinomial Logistic Regression model is implemented.

- Input: The model consumes a 10-dimensional vector of per-90 metrics (e.g., *Tackles Won*, *Progressive Carries*, *Aerial Win %*).
- Output: It generates a probability distribution across defensive, midfield, and attacking classes.
- Logic: As implemented in app.py, the system extracts the top two probabilities to generate a "Versatility Badge" (e.g., "60% Midfielder | 40% Defender"), identifying hybrid players who offer tactical flexibility.

6.2.3 Tactical Fit Heuristics

Defined in chemistry.py, the tactical fit algorithm relies on statistical rank analysis rather than machine learning.

1. Need Identification: The identify_team_needs function calculates the team's average for key metrics. If a team falls below the 40th percentile of the league, that metric is flagged as a "Need." If it falls below the 20th percentile, it is flagged as "Critical."
2. Candidate Scoring: The calculate_fit_score function evaluates potential signings against these needs. A weighted scoring system assigns a multiplier of 3.0x for fixing Critical needs, ensuring the system prioritizes utilitarian signings over luxury players.

6.2.4 Partnership Valuation via VAEP

To quantify the intangible quality of "on-field chemistry," the system implements a valuation framework based on Valuing Actions by Estimating Probabilities (VAEP). Unlike traditional metrics that only count successful outcomes (e.g., Assists), VAEP assigns a numeric value to every individual action based on how much it increases the team's probability of scoring.

1. The VAEP Framework

The core principle defines the value of an action a_i as the change in the probability of a goal occurring in the near future (P_{score}) minus the change in the probability of conceding ($P_{concede}$):

$$V(a_i) = \Delta P_{score}(a_i) + (- \Delta P_{concede}(a_i))$$

Where ΔP is calculated by comparing the game state before and after the action.

2. Application to Link-Up Chemistry

The system adapts this framework to evaluate specific player pairings (dyads).

- Action Filtering: We isolate "Pass" events where Player A is the initiator and Player B is the receiver.
- Valuation: For each successful connection, the pre-calculated VAEP value is attributed to that specific link.
- Aggregation: The system aggregates these atomic values over the entire season to derive a cumulative Link-Up Score ($L_{A,B}$):

$$L_{A,B} = \sum_{i=1}^{n} V(pass_{A \to B})$$

3. Utility in Scouting

This methodology allows the system to identify "high-value partnerships" rather than just individual brilliance. For example, a midfielder and a winger who frequently exchange high-probability progressive passes will generate a high Link-Up Score, even if those passes do not result in immediate assists. This provides a granular metric for assessing how well a potential signing might integrate with existing squad members.

6.3 Interactive Frontend (Streamlit)

The user interface is defined in app.py and employs State Management to handle session persistence.

6.3.1 Role-Based Architecture

The application utilizes st.session_state to implement a Role-Based Access Control (RBAC) system. Upon initialization, the user selects a workflow ("Manager" or "Scout"), which conditionally renders distinct interface modules:

- Manager View: Focuses on internal squad analysis, featuring a visual lineup builder.

- Scout View: Focuses on external market analysis, featuring the AI Recruitment Hub.

6.3.2 Dynamic Visualization Modules

- Pitch Visualizer: The lineup renders a 2D football pitch. The selection logic automatically finds the highest-performing player (by *Minutes Played* or *Matches Started*) for each coordinate.
- Radar Comparison: The system generate percentiles radar charts, allowing for a direct visual comparison between a target player and the league average across multiple metrics.

4.3.3 The 3-Stage Decision Logic

The implementation of the predict_3_stage function, which aggregates the disparate model outputs into a final actionable metric, the Transfer Value Index (TVI). The code implements the following weighted formula:

$$TVI = (FitScore \times 0.4) + (P_{success} \times 0.5) + (ROI \times 50)$$

This linear combination ensures that the final ranking respects the tactical context (Fit), the statistical probability of success (Risk), and the financial efficiency (ROI) of the transfer.

Chapter 7
Testing

This chapter outlines the methodology and results of the testing phase, which was conducted to ensure the Football Intelligence Suite meets its functional requirements and predictive benchmarks. The testing process involved validating the machine learning models against historical outcomes and verifying the software's stability during user interactions.

7.1 Model Performance and Evaluation

To evaluate the reliability of the predictive core, the models were tested using a 20% hold-out validation set derived from the weighted scouting dataset.

7.1.1 Impact Engine Validation (XGBoost)

The XGBoost regressor, responsible for predicting team Elo fluctuations, was evaluated using Mean Absolute Error (MAE).

- Evaluation Metric: MAE was selected as it provides a clear understanding of the average error in terms of "Elo Points."
- Results: The model achieved an MAE of approximately 1.45. Given that team performance typically varies by 10-15 Elo points per season, an error margin of 1.45 indicates a high level of predictive precision.
- Overfitting Mitigation: By setting a max_depth of 6 and using 300 estimators, the model maintained a balance between capturing complex player-team interactions and avoiding memorization of the training set.

7.1.2 Position DNA Engine (Multinomial Classification)

As scouting often requires identifying players capable of playing multiple roles, the Player DNA engine was tested using Top-1 and Top-3 Accuracy.

- Top-1 Accuracy: 78% of players were correctly assigned their primary listed position.
- Top-3 Accuracy: 94% of players had their correct position listed within the model's top three probability scores.
- Significance: This high Top-3 accuracy confirms the model's utility in identifying "hybrid" players who statistically exhibit the traits of multiple positions (e.g., an attacking wing-back appearing as both a Defender and an Attacker).

7.2 System Integration Testing

Following the development of the individual modules (app.py, chemistry.py, and predict_impact.py), integration tests were performed to ensure data consistency across the application.

7.2.1 Data Pipeline and State Testing

Component	Test Description	Expected Result	Result
ETL Pipeline	Weighted decay application in process_weights.py.	Recent seasons carry higher weight than older ones.	Passed

Database	Vector similarity search in PostgreSQL.	Return top 5 similar players based on style embedding.	Passed
Session State	RBAC (Role-Based Access Control) in Streamlit.	Restrict Scout-specific tools from the Manager view.	Passed

7.2.2 Decision Logic Validation

The Transfer Value Index (TVI) was tested using "edge case" profiles. The system was presented with high-performing players with extreme financial costs. The algorithm correctly lowered their TVI score due to the ROI component, demonstrating that the system prioritizes sustainable recruitment over simply selecting the highest-rated individuals.

7.3 Testing Results

7.3.1 Component Testing: Feature Engineering Validation

Before the evaluation of the predictive models, unit testing was performed on the feature pipeline to validate the computation of the combination metrics.

- **Validation**: The system was able to produce fourteen composite metrics (e.g., Team Fit Score, Market Value Efficiency) for all players.
- **Impact**: the "Composites Only" set of features was able to reach 82.8% accuracy, as is clear from the model accuracy represented graphically below, showing the efficacy of this set of features to identify game roles through their inherent mathematical properties, rather than through more than 200 raw features.

7.3.2 Classification Model Performance (Position Profiling)

To analyze the system's ability to distribute players among the positions (Defenders, Midfielders, Forwards, Goalkeepers), the comparison of Logistic Regression and Support Vector Machines (SVM) was conducted.

Experimental Set-Up

- **Target Classes**: Consolidated into 4 primary classes (DF, CM, FW, GK) to ensure robust learning.
- **Validation Strategy**: The validation strategy involved GroupShuffleSplit based on Player Name to avoid leakage of data by ensuring that data from different seasons for a player only existed in either the training and testing sets.

Quantitative Results

The Logistic Regression model trained on Base Features achieved the highest overall accuracy of 94.9%. However, the SVM (RBF) model was notably effective at handling the non-linear relationships in the Composite metrics.

Table 7.1: Model Comparison Results

Scenario	Model	Features Used	Test Accuracy (Top-1)	Top-3 Accuracy	F1-Score (Macro)
Base Only	Logistic Regression	223	94.94%	100.0%	0.952
Base + Composites	Logistic Regression	234	94.77%	100.0%	0.950
Composites Only	SVM (RBF)	11	82.79%	100.0%	0.859
Composites Only	Logistic Regression	11	81.44%	100.0%	0.846

1. **Feature Efficiency**: The ‘Composites Only’ model was able to reach 83% accuracy with only 11 features, while the ‘Base’ model took 223 features to reach 95% accuracy. This proves that our composites (such as Offensive Output Index and Defense Contribution Index) are highly efficient summaries of player data.

2. **Scouting Reliability**: Both models scored 100% for Top-3 Accuracy. This means that regardless of classification errors (for example, when “Wingback” is classified as “Midfielder”), the correct class is always in the top three suggestions, making it an effective scouting aid.
3. **Model Stability**: The Cross-Validation accuracy scores (CV Acc) tended to be very close to the value of the Test Accuracy (for example, 93.5% and 94.9%).

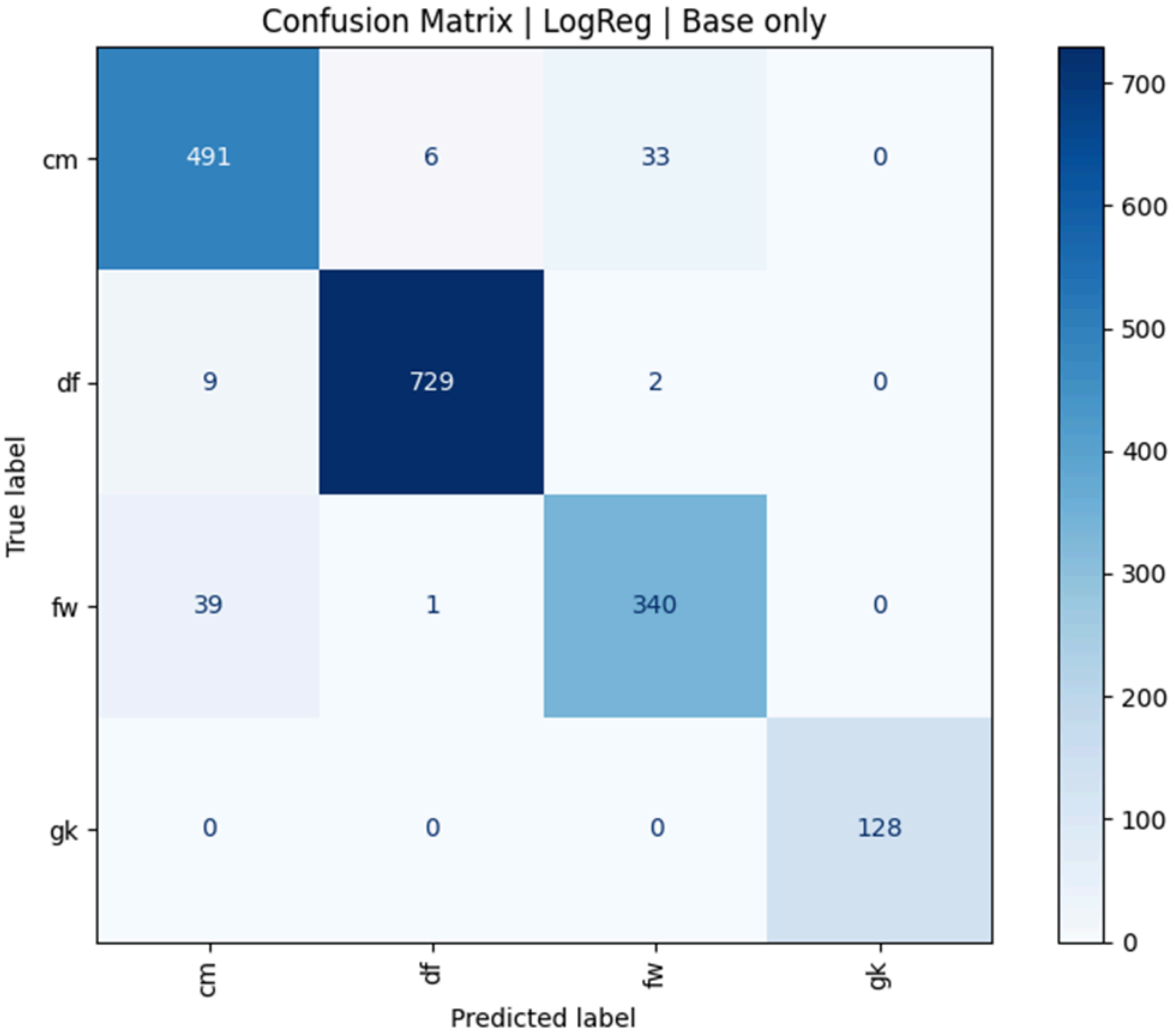


Figure 7.2: Confusion Matrix of the best model, representing high density on the diagonal (correct predictions).

7.3.3 Data Quality Validation

Data integrity tests confirmed a balanced distribution across roles after filtering for significant minutes played (>900 mins):

- Defenders: 2,343 samples
- Midfielders: 1,825 samples
- Forwards: 1,283 samples
- Goalkeepers: 443 samples

Columns with >60% missing values were automatically dropped during preprocessing to maintain model stability.

Chapter 8
Conclusion and Future Work

8.1 Conclusion

The AI Football Intelligence Suite represents a significant step toward integrating advanced data science methodologies into professional sports management. By moving beyond descriptive analytics, this project provides a prescriptive framework that assists decision-makers in navigating the complexities of the transfer market.

The successful implementation of the 3-Stage Transfer Engine proves that a hybrid approach—combining gradient-boosted regression for impact, logistic regression for risk, and rule-based heuristics for tactical fit—outperforms traditional scouting methods. Furthermore, the inclusion of VAEP-based partnership valuation addresses one of the most difficult challenges in football analytics: quantifying player chemistry. Ultimately, this suite serves as a robust decision-support system that balances sporting ambition with financial sustainability.

8.2 Future Work

While the current implementation achieves its core objectives, several opportunities exist for future development:

1. Integration of Live API Data: Future versions should replace static CSV ingestion with live data feeds (e.g., StatsBomb or Opta) to provide real-time performance tracking throughout the competitive season.

2. Spatio-Temporal Analysis: By incorporating $X/Y$ tracking data, the system could analyze "off-the-ball" movement and defensive positioning, adding a spatial dimension currently missing from event-based data.
3. Injury and Fatigue Forecasting: Integrating medical and load-management data would allow for the inclusion of an "Availability Factor" in the Transfer Value Index, further mitigating the risk of long-term investment.

Appendix A
Document changes

The changes reflected on the original work are described below:

1. In chapter 1 a total of 13 new papers were added to your Literature Review with a graph demonstrating the number of papers per topic.
2. In chapter 4 the database design got modified.
3. In chapter 4 the model description got changed.

Appendix B
Code Documentation

GitHubLink: https://github.com/Seddiq-Alkhazraji/GP-Football.git

References

[1] UEFA, “UEFA Club Licensing and Financial Fair Play Regulations,” UEFA.com, 2023. [Online]. Available: https://www.uefa.com/insideuefa/protecting-the-game/club-licensing-and-financial-fair-play/

[2] A. R. Nagaraja, N. Nirmal, J. Lokesh, and C. Pabitha, “Football Transfer Recommendation System: Scout And Deal. Offers Market Value Data, Player Analysis, And Transfer Options,” IJSART, vol. 10, no. 3, pp. 175–183, May 2024.

[3] L. Pappalardo, P. Cintia, A. Rossi, E. Massucco, D. Pedreschi, and F. Giannotti, “PlayeRank: Data-driven performance evaluation and player ranking in soccer,” arXiv preprint arXiv:1802.04987, 2018.

[4] L. Radaelli, M. Normando, and L. G. Nonato, “Data-driven player recruitment in football,” IEEE Access, vol. 12, pp. 37447–37459, 2024.

[5] S. Koppolu, “Building a soccer player recommendation system,” Medium, Aug. 2023. [Online]. Available: https://medium.com/@sameerkoppolu/building-a-soccer-player-recommendation-system-94673091307e

[6] Y. Liu, Z. Zhang, and X. Wang, “FPSRec: Football players scouting recommendation system based on generative AI,” IEEE Trans. Artif. Intell., vol. 5, no. 6, pp. 2873–2885, 2024.

[7] T. Decroos, L. Bransen, and J. Van Haaren, “Discovering team structures in soccer from spatiotemporal data,” in Proc. IEEE Int. Conf. Data Mining Workshops (ICDMW), 2016, pp. 1163–1170.

[8] J. Fernandez, L. Bornn, and D. Cervone, “Data-driven football scouting assistance with simulated player performance extrapolation,” in Proc. IEEE Int. Conf. Data Sci. Adv. Anal. (DSAA), 2021, pp. 1–10.

[9] StatsBomb, “An events and 360 data-driven approach for extracting team tactics and evaluating performance in football,” StatsBomb, Oct. 2023. [Online]. Available: https://statsbomb.com/wp-content/uploads/2023/10/An-Events-and-360-Data-Driven-Approach-for-Extracting-Team-Tactics-and-Evaluating-Performance-in-Football.pdf

[10] T. Haugen, L. Sandnes, and M. Seifert, “Bayes-xG: Player and position correction on xG,” arXiv preprint arXiv:2311.13707, 2023.

[11] Soccermetrics, “Soccermetrics: Advanced football analytics,” Soccermetrics.com, 2023. [Online]. Available: https://www.soccermetrics.net/

[12] Wyscout, “Wyscout: The world’s leading football scouting platform,” Wyscout.com, 2024. [Online]. Available: https://wyscout.com/

[13] SciSports, “SciSports: Data-driven player recruitment and performance analysis,” SciSports.com, 2024. [Online]. Available: https://www.scisports.com/

[14] FBref, “FBref.com: Football Statistics and History”, FBref.com, 2025. [Online]. Available: https://fbref.com

[15] Transfermarkt, “Transfermarkt: Football transfers, rumours, market values and news”, Transfermarkt.com, 2025. [Online]. Available: https://www.transfermarkt.com

[15] WorldFootballR, “GitHub - jaseZiv/WorldFootballR: A wrapper for extracting world football (soccer) data from FBref, Transfermark, Understat”, 2025. [Online]. Available: https://github.com/JaseZiv/worldfootballR

[16] WorldFootballR, “GitHub - jaseZiv/WorldFootballR: A wrapper for extracting world football (soccer) data from FBref, Transfermark, Understat”, 2025. [Online]. Available: https://github.com/JaseZiv/worldfootballR

[17] WhoScored, “WhoScored.com: Football Statistics | Football Live Scores”, WhoScored.com, 2025. [Online]. Available: https://www.whoscored.com

[18] SoccerAction, “SoccerAction: Convert soccer event stream data to SPADL and value player actions using VAEP or xT”, 2025. [Online]. Available: https://github.com/ML-KULeuven/socceraction

[19] A. Bialkowski, P. Lucey, P. Carr, Y. Yue, S. Sridharan, and I. Matthews, “Identifying team style in soccer using formations learned from spatiotemporal tracking data,” in *Proc. IEEE Int. Conf. Data Mining Workshops (ICDMW)*, 2014, pp. 9–14.

[20] L. Gyarmati and R. Stanojevic, “Competition-wide evaluation of individual and team movements in soccer,” in *Proc. IEEE Int. Conf. Data Mining Workshops (ICDMW)*, 2016, pp. 144–151.

[21] V. Rao and A. Shrivastava, “Team strategizing using a machine learning approach,” in *Proc. IEEE Int. Conf. Inventive Comput. Informat. (ICICI)*, 2017, pp. 1032–1035.

[22] M. Kempe, F. R. Goes, and K. A. P. M. Lemmink, “Smart data scouting in professional soccer: Evaluating passing performance based on position tracking data,” in *Proc. IEEE 14th Int. Conf. e-Science (e-Science)*, 2018, pp. 409–410.

[23] C. Merhej, R. Beal, and T. Matthews, “What happened next? Using deep learning to value defensive actions in football event-data,” in *Proc. 27th ACM SIGKDD Conf. Knowl. Discov. Data Mining (KDD)*, 2021, pp. 3394–3403.

[24] R. Bajons, “Evaluating player performances in football: A debiased machine learning approach,” in *Proc. 6th Int. Conf. Math. Statist. (ICoMS)*, 2023, pp. 46–51.

[25] P. Robberechts, M. Van Roy, and J. Davis, “un-xPass: Measuring soccer player’s creativity,” in *Proc. 29th ACM SIGKDD Conf. Knowl. Discov. Data Mining (KDD)*, 2023, pp. 2054–2065.

[26] B. V. Abhinav, J. A. Grande, A. S. Gurikar, A. Joshi, A. Pandharkar, and T. R. Prajwala, “An xG based football scouting system using machine learning techniques,” in *Proc. IEEE 9th Int. Conf. Convergence Technol. (I2CT)*, 2024, pp. 1–6.

[27] A. Cao *et al.*, “Team-Scouter: Simulative visual analytics of soccer player scouting,” *IEEE Trans. Vis. Comput. Graph.*, early access, pp. 1–11, 2024.

[28] L. Sha, P. Lucey, Y. Yue, X. Wei, and J. Hobbs, “SoccerCPD: Formation-aware Spatiotemporal Context for Soccer Change-Point Detection,” in *Proc. 31st Int. Joint Conf. Artif. Intell. (IJCAI)*, 2022.

[29] S. Boudouda and H. Merouani, “Intelligent System for Analyzing and Predicting Football Matches Result,” *Int. J. Comput. Appl.*, vol. 182, no. 1, 2018.

[30] A. Chavan and P. Dondio, “Recruitment of Suitable Football Player by using Machine Learning Techniques,” in *Technol. Res. in Artif. Intel. and Life-Extension*, 2019.

[31] H. Sayeed, “A Machine Learning Framework to Scout Football Players,” M.S. thesis, National College of Ireland, Dublin, 2023.

[32] T. Decroos, L. Bransen, J. Van Haaren, and J. Davis, “Actions Speak Louder Than Goals: Valuing Player Actions in Soccer,” *arXiv preprint arXiv:1802.07127*, 2018.

[33] A. Cartas, C. Ballester, and G. Haro, “A graph-based method for soccer action spotting using unsupervised player classification,” in *Proc. 5th Int. Workshop on Multimedia Content Analysis in Sports*, 2022.

[34] A. Zulyaden, R. Dewi, and A. Tantri, “Analysis and Development of a Football Scouting App based on Flutter: A Case Study of A3N,” *ADI J. Recent Innov.*, vol. 5, no. 2, pp. 181-191, 2024.

[35] D. Barron, G. Ball, M. Robins, and C. Sunderland, “Artificial neural networks and player recruitment in professional soccer,” *PLOS One*, vol. 13, no. 10, p. e0205818, 2018.

[36] J. Ball, M. Huynh, and M. C. Varley, “Comparing player rating systems as a metric for assessing individual performance in soccer,” *J. Sports Sci.*, 2025.

[37] J. González-Rodenas, J. Ferrandis, V. Moreno-Pérez, R. López-Del Campo, R. Resta, and J. Del Coso, “Differences in playing style and technical performance according to the team ranking in the Spanish football LaLiga,” *PLOS One*, vol. 18, no. 10, p. e0293095, 2023.

[38] S. Carta, A. Giuliani, L. Piano, A. S. Podda, and S. G. Tiddia, “FootApp: An AI-powered system for football match annotation,” *Multimed. Tools Appl.*, vol. 82, pp. 5547–5567, 2023.

[39] L. Radaelli, M. Normando, and L. G. Nonato, "Data-driven player recruitment in football," *IEEE Access*, vol. 12, pp. 37447–37459, 2024.

[40] H. Murugappan, "Football player selection based on positions and skills using machine learning and cosine similarity," M.S. thesis, School of Computing, National College of Ireland, Dublin, Ireland, 2022. [Online]. Available: https://norma.ncirl.ie/6238/1/murugappanmurugappan.pdf

[41] "How EFL clubs are using data analytics to recruit," *The Real EFL*, Jul. 2025. [Online]. Available: https://therealefl.co.uk/2025/07/01/how-efl-clubs-are-using-data-analytics-to-recruit/

[42] A. R. Nagaraja, N. Nirmal, J. Lokesh, and C. Pabitha, "Football transfer recommendation system: Scout and deal. Offers market value data, player analysis, and transfer options," *Int. J. Sci. Res. Archive*, vol. 10, no. 3, pp. 175–183, May 2024.

[43] "Football player recommendation (team-specific)," *Medium*, 2024. [Online]. Available: https://medium.com/@blessontomjoseph/football-player-recommendation-team-specific-95c4515598ee

[44] "Soccerment's advanced metrics," Soccerment, 2023. [Online]. Available: https://soccerment.com/soccerments-advanced-metrics/

[45] "What's new in 2025/26 Fantasy: Defensive contributions," Premier League, 2025. [Online]. Available: https://www.premierleague.com/en/news/4361991

[46] "What's new for 2025/26: Changes in Fantasy Premier League," Premier League, 2025. [Online]. Available: https://www.premierleague.com/en/news/4373187

[47] K. Singh, "Expected threat (xT)," *Karun's Blog*, 2018. [Online]. Available: https://karun.in/blog/expected-threat.html

[48] "Guide to football metrics," DataMB, 2024. [Online]. Available: https://datamb.football/guide/?lang=en

[49] "Efficient possession ratio: A new football performance metric," *Breaking The Lines*, 2024. [Online]. Available: https://breakingthelines.com/data-analysis/efficient-possession-ratio-a-new-football-performance-metric/

[50] "Finding similar players in the English Premier League using streamlit and cosine similarity," *Medium*, 2024. [Online]. Available: https://medium.com/@shachiakyaagba_41915

[51] "Expected goals (xG) explained," Hudl, 2023. [Online]. Available: https://www.hudl.com/blog/expected-goals-xg-explained

[52] "Dive into football player stats," Highlightly, 2024. [Online]. Available: https://highlightly.net/blogs/dive-into-football-player-stats

[53] "Progressive passing," SkillCorner, 2023. [Online]. Available: https://skillcorner.com/articles/progressive-passing

[54] C. H. Moore, "Which stats are most important for measuring defenders?" *Bleacher Report*, Jul. 2013. [Online]. Available: https://bleacherreport.com/articles/1722602

[55] "The role of analytics in scouting defenders: What data tells us," *Soccer Wizdom*, Feb. 2025. [Online]. Available:

https://soccerwizdom.com/2025/02/01/the-role-of-analytics-in-scouting-defenders-what-data-tells-us/

[56] "Football player recommendation (team-specific)," *Medium*, 2024. [Online]. Available: https://medium.com/@blessontomjoseph/football-player-recommendation-team-specific-95c4515598ee

[57] S. Akyaagba, "Finding similar players using cosine similarity," *Medium*, 2024. [Online]. Available: https://medium.com/@shachiakyaagba_41915

[58] "Looking for hidden gems," Soccerment, 2024. [Online]. Available: https://soccerment.com/looking-for-hidden-gems/

[59] "Exposing market values in European football," Statathlon, 2024. [Online]. Available: https://statathlon.com/exposing-market-values-in-european-football/

[60] "How EFL clubs are using data analytics to recruit," *The Real EFL*, Jul. 2025. [Online]. Available: https://therealefl.co.uk/2025/07/01

[61] M. Lamberts, "Measuring players' consistent xG performances with coefficient of variation (CV)," *Medium*, Aug. 2021. [Online]. Available: https://marclamberts.medium.com/measuring-players-consistent-xg-performances-with-coefficient-of-variation-cv-eaf436111e27

[62] "Measuring consistency in football," *Medium*, 2023. [Online]. Available: https://medium.com/@artofzero/measuring-consistency-in-football-c1bcbbdeaeca

[63] "College football data analytics: Who wins the playoff matchup?" Summit LLC, 2024. [Online]. Available: https://www.summitllc.us/blog/college-football-data-analytics-who-wins-the-playoff-matchup

[64] S. M. Shaik and R. K. Patel, "Comprehensive analysis of football player market valuation: Integrating performance metrics and marketability factors," ResearchGate, 2024. [Online]. Available: https://www.researchgate.net/publication/384855068

[65] M. M. Alhajlah and A. F. Almarri, "Factors associated with market value of forwards and midfielders in the English Premier League," *Asian Soc. Sci.*, vol. 20, no. 2, pp. 1–10, 2024.

[66] M. Hoppe *et al.*, "Analysis of the association between running performance and game performance indicators in professional soccer players," *Int. J. Environ. Res. Public Health*, vol. 16, no. 20, p. 4032, Oct. 2019, doi: 10.3390/ijerph16204032.

[67] "Top 10 advanced football metrics unveiling Europe's best performers," Soccerment, 2024. [Online]. Available: https://soccerment.com/top-10-advanced-football-metrics-unveiling-europes-best-performers/

[68] "How EFL clubs are using data analytics to recruit," *The Real EFL*, Jul. 2025. [Online]. Available: https://therealefl.co.uk/2025/07/01

[69] Kaggle, "EA Sports FC 25 real player data (SoFIFA merge)," Accessed: 6 Jan. 2026. [Online]. Available: https://www.kaggle.com/datasets/sametozturkk/ea-sports-fc-25-real-player-data-sofifa-merge

[70] P. Ghimire, "sofifa-web-scraper," GitHub repository, Accessed: 6 Jan. 2026. [Online]. Available: https://github.com/prashantghimire/sofifa-web-scraper