University College Cork

Department of Computer Science

Using Machine Learning to Predict the Success of Football Transfers

Robbie O' Sullivan

Supervisor: Cathal Hoare

Submitted: April, 2024

Final Year Project

BSc Computer Science (Software Entrepreneurship)

Abstract

In this project, we attempt to use machine learning techniques to predict the success of football players upon transferring between clubs. Aimed at enhancing the decision-making processes for football clubs, this project combines statistical analysis, predictive modelling, and user-friendly web technology to forecast player performance in new club environments.

Therefore, The European Transfer Prediction System has been created. The system evaluates players based on a number of parameters, including age, playing minutes, goals, assists, and historical transfer values, to estimate their potential impact at a new club. These parameters were selected through data exploration and trend analysis, underlining key features and their correlation to the metrics we sought to predict. A player's success is quantified by a 'Success Score' rating, calculated through an algorithm that considers both on-field performance metrics and market dynamics. Upon evaluating a transfer, the system presents the key metrics and predictions, enabling users to make data-driven decisions. The system's effectiveness has been proven on several transfers from the transfer window of January 2024, and

The project highlights the power of data analytics in sports management, demonstrating the system's potential for future applicability in a wider sports context.

Declaration of Originality

In signing this declaration, you are confirming, in writing, that the submitted work is entirely your own original work, except where clearly attributed otherwise, and that it has not been submitted partly or wholly for any other educational award. I hereby declare that:

- this is all my own work, unless clearly indicated otherwise, with full and proper accreditation;
- with respect to my own work: none of it has been submitted at any educational institution contributing in any way to an educational award;
- with respect to another's work: all text, diagrams, code, or ideas, whether verbatim, paraphrased or otherwise modified or adapted, have been duly attributed to the source in a scholarly manner, whether from books, papers, lecture notes or any other student's work, whether published or unpublished, electronically or in print.

Signed: Robbie O' Sullivan

Date: 24/04/2024

Acknowledgements

I want to thank my supervisor, Professor Cathal Hoare, whose encouragement and insightful suggestions kept me focused throughout the project and whose contribution has been invaluable.

Contents

List of Figures

Chapter 1

Introduction

With the cost of running a professional football team skyrocketing – clubs are on the constant hunt to lessen their risk when it comes to spending large amounts of money on statement signings. In this project, machine learning models will be trained on historical data to predict player metrics and determine whether players will succeed at their new clubs, and potentially save football teams millions of euros by avoiding poor investments.

The role of data is becoming more important in all aspects of professional sport and as such, football has increasingly become data-driven. Predictive models leverage vast amounts of historical and real-time data to make informed decisions to give clubs competitive advantage in the market, and will lead to direct results on the pitch. Each and every professional club has a team of data analysts – tasked to improve the decision making process [2]. The difference in quality of analysis shown between clubs in the market is staggering, with some clubs consistently recruiting very effectively, while others waste massive budgets on poorly informed decisions.

Measuring the success of a transfer is difficult to do, with many factors to consider when judging the performance of a new player, such as appearances, goals scored, games won, competitions won, awards won, and the rise in the players market value to give a few examples. As hard as it is to measure a successful transfer, predicting a successful transfer, is even harder. In this project, machine learning models will be built and trained that consider these factors, and when prompted by the user through a web application, return a potential successful transfer list for the club and position in question, or rate a specific transfer requested by the user. The predictive models are based on data from *'Transfermarkt'*, a dataset available online. This dataset has everything needed to complete this prediction project, and as such, is extremely large in size. Therefore, preprocessing of the dataset will be conducted, including handling missing values, encoding categorical variables, and scaling features. *Exploratory data*

analysis and *feature selection* will be essential to understand the dataset and to select the most relevant features for transfer prediction.

The European Transfer Prediction System strives to go further on usability than any other football transfer prediction system. When considering the past projects and literature, the lack of clear, understandable results and a consistent process was apparent. A key expectation in this project is to establish a transparent and repeatable process that offers actionable insights while acknowledging the intrinsic uncertainty of the football landscape. While the current project focuses on football, the underlying principles could be applied to other sports, paving the way for a future *scalability* and *customisation* of the project.

Chapter 2

Analysis

2.1 Background Research

This section will outline a sample of the key findings from academic papers and studies in the area of research, or that is useful to consider when researching the usage of AI in football. It will also give a breakdown of how the process of these projects would differ using the dataset chosen to train my model on 'Football Data from Transfermarkt' from kaggle.com. These papers were discovered through research in the area of study, using repositories such as the "ACM Digital Library".

2.1.1 'PlayeRank': Data-driven Performance Evaluation and Player Ranking in Soccer via a Machine Learning Approach - L. Pappalardo et al.

This paper, written by L. Pappalardo et al.(2019) focuses on the data-driven evaluation of football players via a machine learning approach. 'PlayeRank' outperformed existing approaches in being significantly more concordant with professional soccer scouts. Moreover, their experiments showed several interesting results, shedding light on novel patterns that characterise the performance of soccer players. 'PlayeRank' models the performance of a soccer player in a match as a multidimensional vector of features extracted from soccer logs, currently produced by several sports analytics companies (i.e., Wyscout, Opta, Stats). Pappalardo's system would work perfectly using the data set chosen in this project, as the data it is using is formatted very similarly to the 'Transfermarkt' data set. The key metrics used, such as appearances and performance over a season are very applicable, but the system does not use transfer value as a metric, which will be a heavy focus of this project [12].

2.1.2 Using machine learning to identify high-value football transfer targets - Tattersall

The system built by Jack Tattersall (2021) trains an algorithm "to identify players who will perform well in the coming season. Further, which players would likely perform well at a club if we signed them" [14]. This is structured as a supervised machine learning problem using the xgboost decision tree algorithm, with a continuous target variable relating to future player performance, which will be predicted in advance. Tattersall uses the FIFA 20 dataset, which is used in a popular video game, and trains his model with a feature of the video game's 'potential' system, which is a prediction made by the developers as to how the player will grow throughout the year. This represents a notable limitation of the system, as the 'potential' data used is purely fictional. The model trained in this project will be based on real data, and yield real results. Therefore, Tattersall's project would not work with the 'Transfermarkt' data set, as a key metric of this model's training is the 'potential' system from the FIFA video game, which is unreliable and is based on a video game developer's view of the current market. This project is not going in this direction.

2.1.3 Jeremy Doku and using AI to predict the success of transfers - Melvang et al.

This article written by Melvang et al. (2021) looks into specific players, using AI to predict how successful they will become if transferred to a new club. The predictions are based on the role that they play in their current club, and how that will be suited to a potential new club [8]. The first thing that was very apparent in this article is that the key player that the authors are highlighting, Jeremy Doku, was playing for the football club Rennes F.C. when this article was published. He has since transferred to Manchester City F.C. in the summer of 2023, arguably the best football club in the world, and is performing very well there. The model is trained on past transfers of players from Rennes, and analyses if the role that the player performs has had successful transfers in the past. The results provided in this article were hugely successful, with the model predicting a transfer of the Brazilian Raphina to Leeds United F.C. would be successful, and it was, with Leeds being very competitive in the English Premier League, and Raphina since moving on to one of the world's biggest football clubs, F.C. Barcelona.

This project uses a "multi-head neural network model that predicts 19 metrics that are aggregated per 90 minutes - shots per 90, xG per 90, passes in the final third per 90 and so on". The details of how the model was trained are not given, but the complexity of the individual player predictions here is not needed in the European

Transfer Prediction System project, as it judges transfers with both performance and financial data, which is not considered by the authors. This complexity could be achieved in the project with more development.

2.1.4 Finding strikers to replace Aguero at Manchester City in 2016 - TYSONIKE

This project from kaggle user TYSONIKE, (2023) uses the 'Football Data from Transfermarkt' dataset, which is the dataset targeted for use in this project [13].

This project uses an interesting metric that is specific to its target, finding a striker to replace Sergio Aguero, the 'SKORE' rating. The code calculates a metric called 'SKORE' for each player, which represents their potential as a striker. The 'SKORE' value is calculated based on various factors, including goals scored per minute played, height, age, market value, current club's domestic competition, and foot preference. This project has had a large influence on the European Transfer Predicition System. The 'SKORE' rating of a striker is the main inspiration in the creation of the "Success Score" rating system for transfers, based on research and findings from other projects in the field. The use of a specific position to find the best players was a factor that influenced the creation of the stretch goal in this project.

2.1.5 Measuring Transfer Success Through Minutes Played - Analytics FC

An interesting challenge with this subject matter is determining what exactly is a measure of a successful transfer? Speaking at a conference at Stamford Bridge, Liverpool Football Club's Head of Research, Dr Ian Graham, made the claim that, statistically speaking, over 50 percent of "big transfers", defined in this case as those over £10m in terms of fee — fail.

His definition of failure was based around a percentage of minutes played. If a player failed to play 50 percent of the available minutes, then that transfer was considered to be a failure [1]. This has inspired the 'Success Score' algorithm. Minutes played for a new club will have a big impact on the transfer's final rating.

2.1.6 A Multiple Linear Regression Approach For Estimating the Market Value of Football Players in Forward Position - Kologlu et al.

This study uses multiple linear regression to estimate the market values of football players, specifically forwards, based on physical and performance data from the 2017-2018 season across Europe's top four leagues. The model assesses the impact of various factors on player valuation and tests for homoscedasticity (varience in the reaiduals is constant) to ensure the robustness of the regression results. The paper identifies which factors most significantly affect players' market values and provides a detailed analysis of how these factors are quantitatively related to value estimations.

This research is extremely useful for my project. The research done by Kologlu et al. has influenced the machine learning algorithm selection and testing in my project [7]. It provides a helpful benchmark and point of comparison to assess the effectiveness of the models used in the European Transfer Prediction System.

2.1.7 AI-Driven Predictive Models in Football: Evaluating Player Performance and Transfer Fees - Sulimov

This study by Sulimov, investigates the application of machine learning techniques to predict football player performance and transfer fees. It primarily focuses on the use of expected goals (xG) models and the Linear Regression, Random Forest, and CatBoost Classifier algorithms, which are employed to analyse match event data, such as shooting angles and distances from the goal. The research highlights how these models manage imbalanced datasets, a common issue in sports analytics, by integrating categorical features directly, which is a noted advantage of CatBoost. These models are evaluated on their precision, recall, F1-score, and AUC score to ensure the reliability of their predictions in real-world scenarios, such as predicting the outcome of matches and the potential transfer fees of players [16].

The importance of this study lies in its comprehensive analysis of machine learning's potential to transform sports predictions by providing accurate assessments of a player's potential performance in new settings.

2.2 Project Objectives

The primary aim of this project is to create a predictive system that accurately evaluates the potential success of football player transfers based on historical data from 2018 to 2023. Utilising the comprehensive 'Football Data From Transfermarkt' dataset, this system will leverage player and club statistics to simulate outcomes of past and hypothetical future transfers. By predicting a player's percentage of total minutes played at a new club and their future transfer value, the system will calculate a 'Success Score' for each transfer. This scoring mechanism, grounded in data-driven insights, aims to assist football club decision-makers in optimising their recruitment strategies. Ultimately, the project seeks to not only validate the accuracy of its predictions against known outcomes but also to serve as a tool for suggesting strategic player signings within predefined budgets.

An essential objective of this project is to design and implement a user-friendly interface that allows users to interact seamlessly with the predictive system. This interface will not only facilitate the input of transfer scenarios but also present predictive outcomes in an accessible and informative manner. By doing so, the system aims to simplify access to advanced data analytics in football, enabling club managers, scouts, and even fans to make well-informed decisions based on robust statistical evidence. The interface will highlight key metrics and insights derived from the model's predictions, ensuring users can understand the reasoning behind each 'Success Score' and its implications for transfer strategy. In doing so, the project hopes to bridge the gap between complex data analysis and practical, actionable insights within the football community.

Implementation wise, the main goal of the project is to successfully develop a web application that allows users to pick a player and club from the database to judge if the transfer would be successful. This should be implemented in a user friendly way so that very little football knowledge is needed to use the system. The project also has a stretch goal of allowing the user to choose a club, position and budget for a new transfer, and the model will run through all players from that position, returning the best suited players with the highest success score.

2.3 Exploring the Dataset

To determine the most important features to a football player's successful transfer, a comprehensive analysis of past data must be carried out. A past analysis was conducted by Kaggle user 'DAVIDCOXON', titled 'Football Transfer Market EDA and Basic Modelling', and this section will take inspiration from this breakdown of the dataset, and then build upon it to look further into the specific features that transfer success depends on.

2.3.1 Overview of the Dataset

The 'Football Data From Transfermarkt' dataset includes 30,000+ players and 400+ clubs, with 400,000+ player valuations. This dataset is a rich source of information for statistical analysis and predictive modelling in football, offering insights into players' performances across different seasons and leagues. This is an extremely large dataset, and for the models to be efficient, preprocessing of the data was crucial. Figure 2.1 is a class diagram of the complete database [5].

2.3.2 Data Cleaning and Preprocessing

All data cleaning and preprocessing was done in Python with Pandas. Certain sections of this dataset were not applicable to this project, such as 'game events', 'club games' and 'competitions', and therfore were excluded. Along with this, the other elements needed to be combined, for ease of use and more efficient model training.

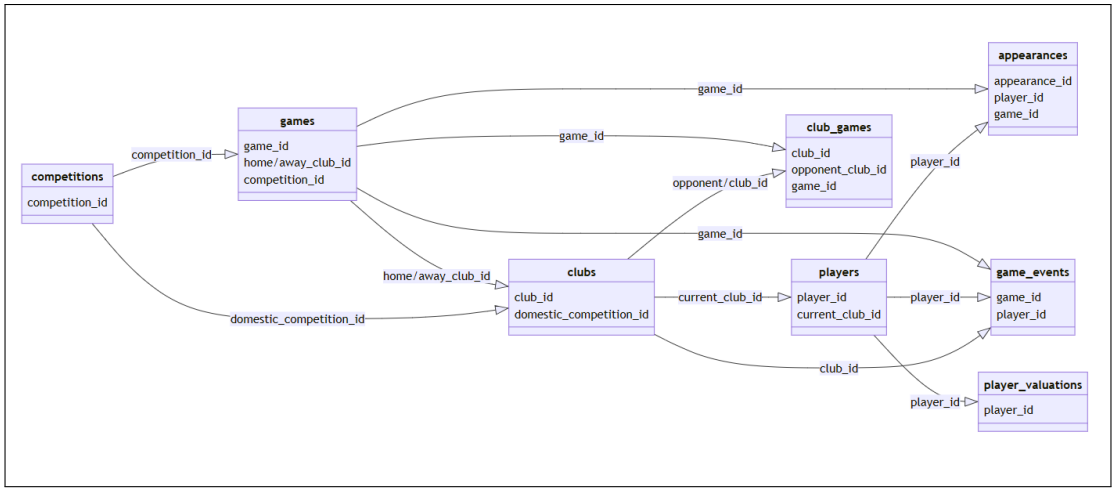

Figure 2.1: 'Football Data From Transfermarkt' Class Diagram

```
# Load player and player valuation CSV files into DataFrames
players_df = pd.real_csv('input/players.csv')
player_valuations_df = pd.read_csv('input/player_valuations.csv')
# Merge player and player valuation DataFrames based on player_id
merged_df = pd.merge(player_valuations_df, players_df, on='player_id',
    how='left')
# Export the merged DataFrame to a CSV file
merged_df.to_csv('merged_player_valuations.csv', index=False)

# Calculate the age of each player
merged_df['date_of_birth'] = pd.to_datetime(merged_df['date_of_birth'])
# Drop players with no date of birth
merged_df = merged_df[merged_df['date_of_birth'].isnull() == False]
now = datetime.now()
merged_df['age'] = (now - merged_df['date_of_birth']).apply(lambda x: x
    .days) / 365.25
merged_df['age'] = merged_df['age'].round().astype(int)
```

Each player had statistics to be counted in other .csv files, as player statistics were not originally in the players table. Retired players were also not included in this project. Each player's percentage of minutes played is also calculated here for each season. This code was modified from Cordiero, L.G.'s 'Market Value EDA' project [10].

```
appearances = pd.read_csv('input3/appearances.csv')
games = pd.read_csv('input3/games.csv')

games_and_apps = appearances.merge(games, on=['game_id'], how='left')
def player_stats(player_id, season, df):

df = df[df['player_id'] == player_id]
df = df[df['season'] == season]

    df["goals_for"] = df.apply(lambda row: row['home_club_goals'] if
    row['home_club_id'] == row['player_club_id']
                else row['away_club_goals'] if row['away_club_id'] ==
    row['player_club_id']
                else np.nan, axis=1)
    df["goals_against"] = df.apply(lambda row: row['away_club_goals']
    if row['home_club_id'] == row['player_club_id']
                else row['home_club_goals'] if row['away_club_id'] ==
    row['player_club_id']
                else np.nan, axis=1)
    df['clean_sheet'] = df.apply(lambda row: 1 if row['goals_against']
    == 0
```

```
                else 0 if row['goals_against'] > 0
                else np.nan, axis=1)

    df = df.groupby(['player_id',"season"],as_index=False).agg({'goals'
    : 'sum', 'game_id': 'nunique', 'assists': 'sum','minutes_played' :
    'sum', 'goals_for' : 'sum','goals_against' : 'sum', 'clean_sheet' :
     'sum'})
    out_df = df.rename(columns={'game_id': 'games'})

    return out_df

season = 2022
for index in players_df.index:
    id = players_df.loc[index][0]
    name = players_df.loc[index][1]
    stats = player_stats(id, season, games_and_apps)
    players_df.at[index,'goals_{}'.format(season)]= stats['goals'][0]
    players_df.at[index,'games_{}'.format(season)]= stats['games'][0]
    players_df.at[index,'assists_{}'.format(season)]= stats['assists'
    ][0]
    players_df.at[index,'minutes_played_{}'.format(season)]= stats['
    minutes_played'][0]
    players_df.at[index,'goals_for_{}'.format(season)]= stats['
    goals_for'][0]
    players_df.at[index,'goals_against_{}'.format(season)]= stats['
    goals_against'][0]
    players_df.at[index,'clean_sheet_{}'.format(season)]= stats['
    clean_sheet'][0]

total_team_mins = players['games'] * 90
players['minutes_percentage'] = (players['minutes_played'] /
    total_team_mins)
```

This calculates all relevant statistics for each player. To calculate for other seasons, the 'season' variable can be changed to the target season and the code cell can be run again. Once this is done, each season can be saved in separate files and concatenated.

2.4 Feature Selection

In order to train models to predict minutes percentages and transfer values, the most correlated features must be selected from the data. This section will investigate the features for the two target values below.

2.4.1 Percentage of Minutes Played

A correlation matrix is the most efficient process to establish the most important features in the target variable. To run a correlation matrix on 'minutes_percentage':

```
numeric_df1 = players_df.apply(pd.to_numeric, errors='coerce')

# Drop columns with all non-numeric values
numeric_df1 = numeric_df1.dropna(axis=1, how='all')

correlation_matrix1 = numeric_df1.corr()

# Find the most correlated columns with 'value'
most_correlated = correlation_matrix1['minutes_percentage'].sort_values
    (ascending=False)

N = 10  # Set N to the desired number of top correlated columns
print(f"Top {N} most correlated columns with 'minutes_percentage'")
print(most_correlated.head(N))
```

This code produces the output below:

```
Top 10 most correlated columns with 'minutes_percentage'
minutes_percentage              1.000000
avg_minutes_percentage          0.909475
minutes_played                  0.799460
games                           0.746637
goals_against                   0.725215
goals_for                       0.651056
clean_sheet                     0.627632
assists                         0.382361
goals                           0.324994
MP                              0.319880
```

A correlation value close to 1 indicates a strong positive relationship. The highest correlation is with 'avg_minutes_percentage', suggesting consistency between average and actual minutes played. Other significant correlations are with 'minutes_played' and 'games', indicating that more playing time and appearances contribute to a higher minutes percentage. Defensive metrics like 'goals_against' and 'clean_sheet' also show strong correlations, implying defensive roles may influence playing time. Offensive contributions ('goals_for', 'assists', 'goals') are less correlated but still relevant. This proves that team-wide statistics have more correlation with a players percentage of minutes played than individual statistics. This is an interesting study, and very valuable when choosing features.

2.4.2 Future Transfer Value

A correlation matrix was also run for the target variable 'value' to gain insight on the features that can be used to predict a player's value. The matrix output is printed below:

```
Top 15 most correlated columns with 'value'
value                          1.000000
previous_value                 0.675398
highest_market_value           0.611192
goals_for                      0.548794
clean_sheet                    0.514755
highest_market_value_in_eur    0.510165
minutes_played                 0.475118
goals                          0.471312
assists                        0.469956
games                          0.467289
goals_against                  0.360583
MP                             0.319439
avg_minutes_percentage         0.299495
minutes_percentage             0.291307
```

The correlation matrix reveals how a player's market value is related to various factors. A strong positive correlation with 'previous_value' and 'highest_market_value' suggests a player's past and peak market values are significant indicators of their current value. Performance metrics such as 'goals_for', 'clean_sheet', 'goals', and 'assists' also show notable correlations, and show which on-field contributions play a crucial role in determining a player's worth. The correlation with 'minutes_played' and 'games' underscores the importance of consistent participation.

Chapter 3

Design

This section will provide a **high-level overview** of the design of The European Transfer Prediction System.

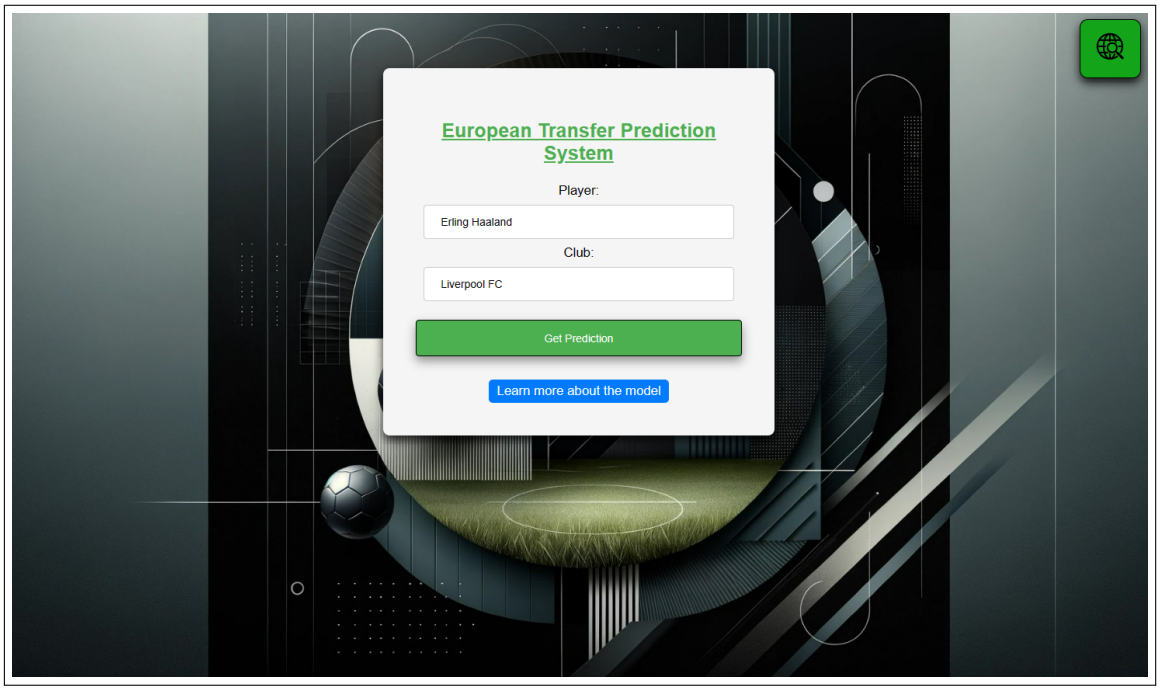

Figure 3.1: Representation of Club and Player Picking Interface

3.1 System Architecture

Figure 3.1 is a representation of the system's main functionality that the user can interact with. The proposed product is a web application that can interact with the

trained models and success score algorithm calculation. The web application allows users to select a player from an auto-completing list sourced from the 'Transfermarkt' database and to specify a transfer club. The model then calculates and returns a success score for the proposed transfer, as depicted in Figure 3.2. The system also provides detailed feedback on key factors influencing the transfer's potential success or failure. This insight enables users to either reconsider their choice or proceed with confidence in their decision. Achieving this interactive and informative functionality was a primary objective of the project.

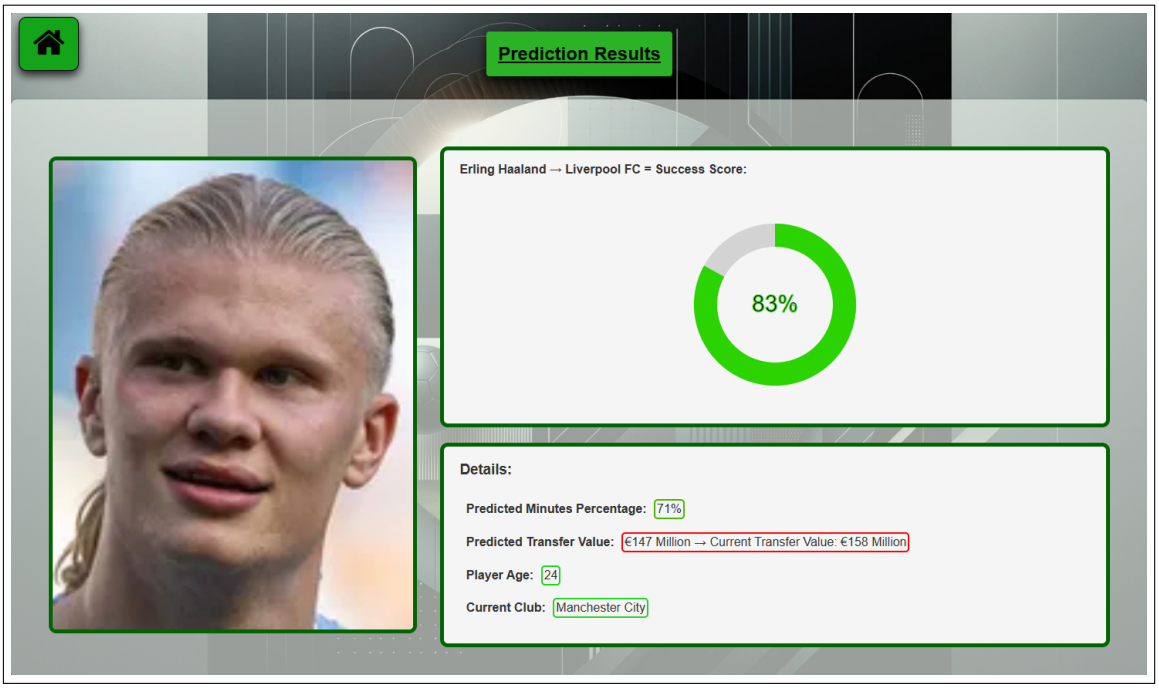

Figure 3.2: Representation of Success Rating Output

A stretch goal of the project is a feature on the web app which will allow the user to pick a club, budget and position (role on the team) they would like to fill with a new transfer. This would allow users to start their search of for a player on the system, as well as confirming their choices. The Input form is seen below in Figure 3.3.

Once the user has input their chosen club, position and budget, the system provides the user with a list of suggested players to sign sorted by their success rating, amount of games played and age. The returned list is shown in Figure 3.4.

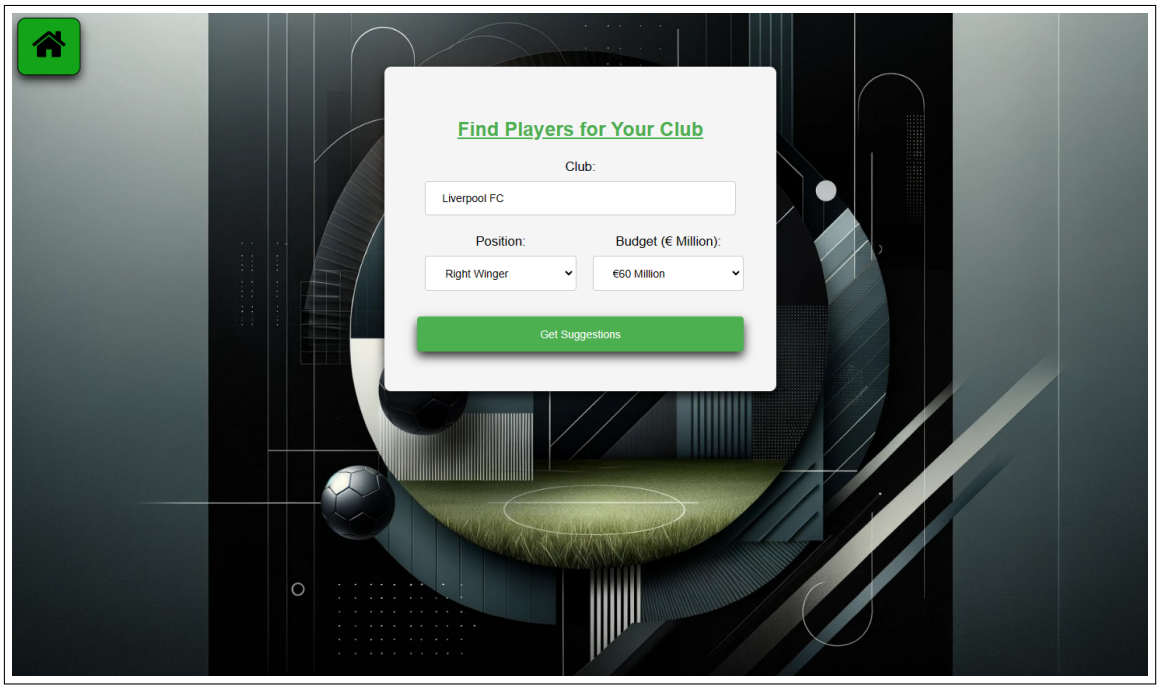

Figure 3.3: Representation of Player Suggestion Feature

This list provides quick details about each player, giving the user clear indication as to who the algorithm thinks would have to most potential at their chosen club.

The below sections will give further description of the major components of the system, the role that each plays and the interrelationships between them.

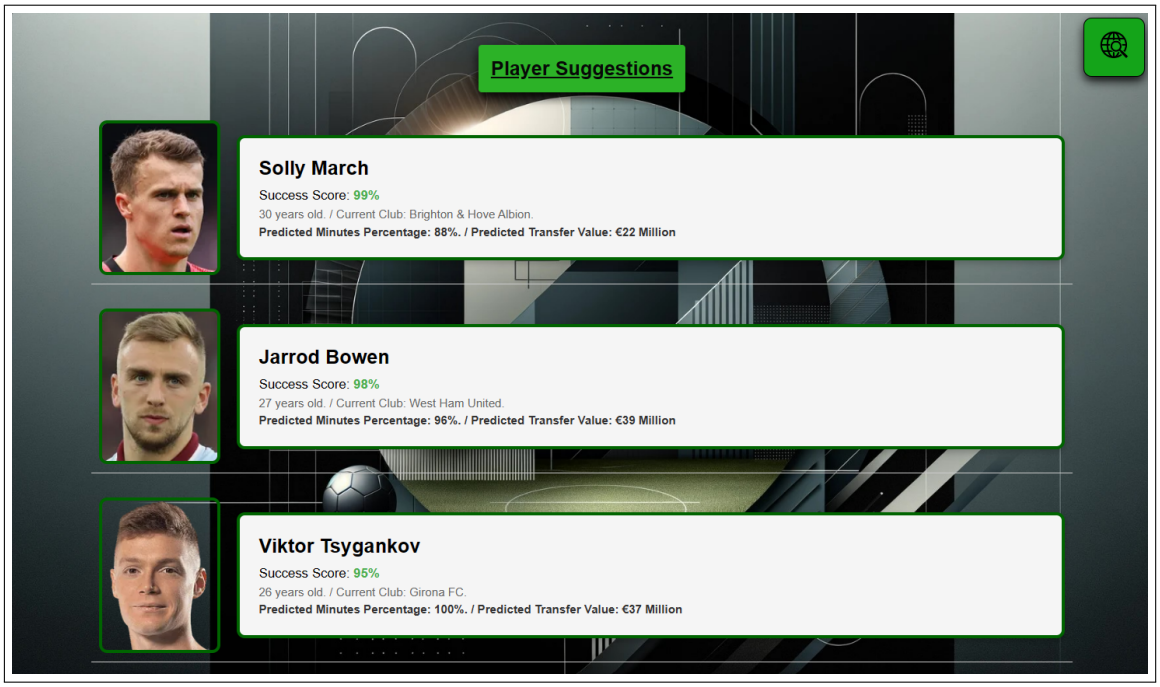

Figure 3.4: Representation of Player Suggestion Output

3.2 Predictive Model Design

The core of the system is its machine learning models. These models were trained on the prepoccessed dataset, and saved to .pkl files. These models are downloadable, along with the full code, on the project GitHub linked in the appendix. Two models operate in tandem:

The first model is designed to forecast the percentage of playing minutes a player is likely to receive at a new club, which is a direct indicator of the player's integration and utility in the new team. The second model estimates the player's future market value based on current performance metrics and historical data, providing an economic perspective on the player's potential financial impact.

Both of these are 'Linear Regression' models. After examining algorithms used in the existing literature, such as decisions trees and neural networks, the decision to use 'Linear Regressions' to predict relatively straight-forward figures seemed obvious.

The advantages of using 'Linear Regression' models are:

- Simplicity and Transparency: Linear regression provides a clear and understandable relationship between inputs and outputs, making it easier for stakeholders to interpret how player attributes influence predicted outcomes.
- Efficiency: These models are computationally inexpensive, ensuring that the web application remains responsive and can deliver predictions efficiently.
- Data Suitability: The nature of the dataset, with its linear correlations between features like playing minutes, goals, assists, and market values, lends itself well to linear modelling.
- Learning Curve: Great starting point to develop knowledge of machine learning principles.

The models are integrated into the 'Success Score' algorithm, which is detailed below in section 3.3.1, and the system's back-end, where models retrieve player and club data. This predictive model setup ensures that users receive immediate and relevant feedback on potential transfers.

The main work to be completed on the machine learning side in the project was the data preprocessing and feature selection. Performing this extensive process allowed for these efficient models to quickly predict targets. The specifics of building and training the model will be discussed below in the Implementation section.

3.3 Determining Success in a Transfer

A major goal for this project is designing an intuitive and comprehensive measurement of a transfer's success. After conducting the literature review above, and applying previous knowledge of the game, the project proposes a new *'Success Score'* algorithm for determining a transfer's success. This algorithm can be used to judge transfers in the past by using existing data and, given some estimated parameters from the machine learning models, give an accurate prediction as to whether the transfer will be a success or not.

3.3.1 Defining the Algorithm

The algorithm takes two major parameters, the players' percentage of total available minutes played, and the player's transfer value in the next transfer period. These each have a 47.5% weight in the algorithm. Alongside this, if the player is transferring to a club in the same league, they will gain 5%.

If a player is over the age of 32, they will get a maximum of 30% from each weighted section of the algorithm, and if the player is 20 or under, they will have a maximum of 30% weight in the playtime section of the algorithm. This is required for accurate scores, as the models have trouble determining age drop off. This weighting calculation helps to solve this issue. Studies show that players usually start rapidly declining after the age of 31, as seen by Figure 3.5 and players typically do not get much playtime with big clubs before the age of 21 [9].

3.3.2 Calculating the Algorithm

To calculate the total playtime half of the algorithm:

$$47.5 \times Percentage\ of\ Minutes\ Played$$

To calculate the transfer value half of the algorithm:
If the player's market value has risen, the transfer gets the full value section of 47.5 percent.
If the player's value has decreased, this equation is run:

$$47.5 \times \left(\frac{Future\ Player\ Value}{Player\ Value\ at\ Transfer} \right)$$

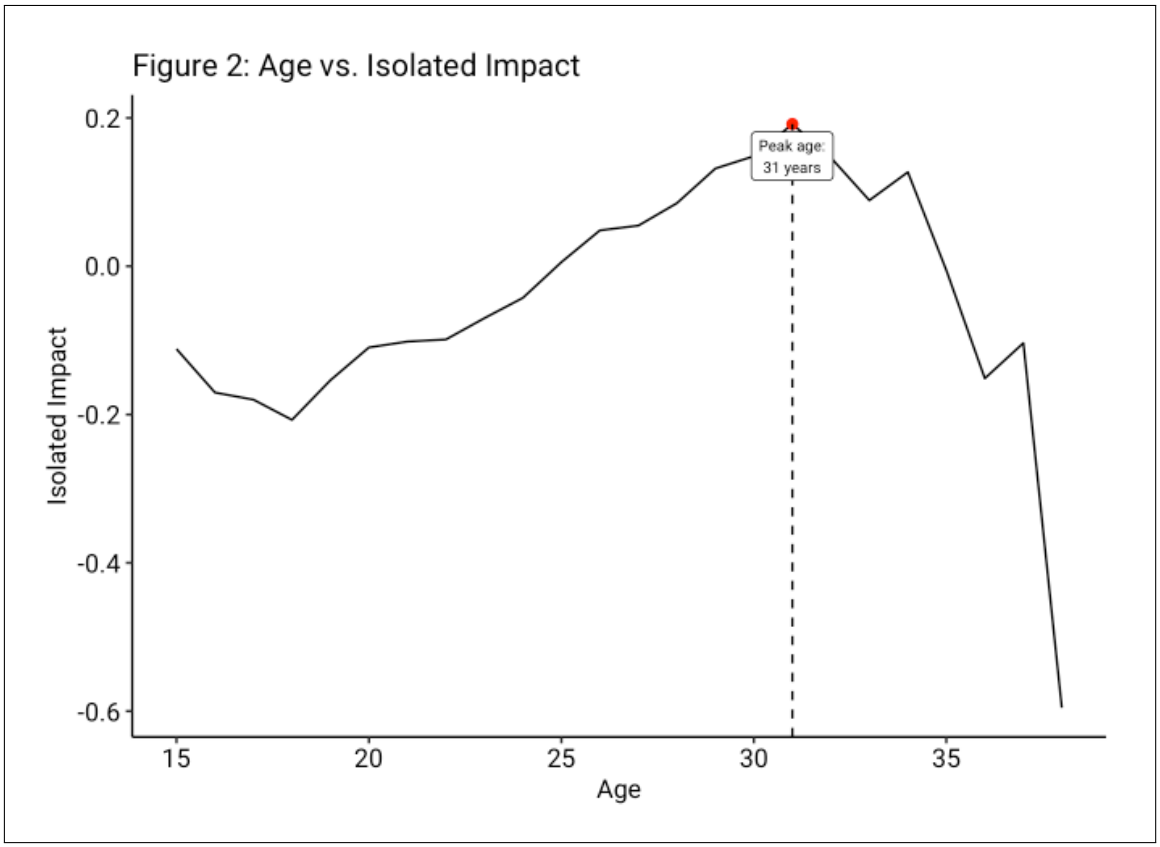

Figure 3.5: Player Age Comparison

When this calculation is complete, the system adds 5 to the value *if the player currently plays in the same league*. These complete calculations added together give the 'Success Score' of a transfer. Testing of the algorithm will be carried out in the next section.

3.4 Testing the Algorithm

This section will provide two examples of algorithm testing on past transfers, and will prove that this method of defining success in a transfer is robust, giving the user a comprehensive appraisal of a player's performance after transferring to a new club.

Two examples of past transfers are detailed below. The first transfer, Virgil van Dijk transferring from Southampton F.C. to Liverpool F.C. was an extremely successful transfer. The second is Eden Hazard's surprisingly poor transfer from Chelsea F.C. to Real Madrid C.F.

3.4.1 Virgil van Dijk *Southampton* → *Liverpool*, 2018

Virgil van Dijk transferred from Southampton F.C. to Liverpool F.C. in January of 2018, and is now widely recognised as one of the world's best defenders, with his transfer being a watershed moment for Liverpool, bringing the club major honour after major honour, including the UEFA Champions League, and English Premier League titles [15].

Virgil van Dijk is given a 'Success Score' of **99%** from the algorithm. This is because he played 98% of available minutes, stayed in the English Premier League, and his value went from €50 million euro to €90 million euro.

3.4.2 Eden Hazard *Chelsea* → *Real Madrid*, 2019

After being a Chelsea F.C. mainstay for many years, Eden Hazard transferred to Spanish giants Real Madrid C.F., and is now known as one of Madrid's worst ever signings. Hazard is now retired from football altogether, with the player announcing the end of his Real Madrid career in 2023 [6]. This is an example of promising transfer that sounded bulletproof on paper quickly turning sour.

Eden Hazard is given a 'Success Score' of **40%** from the algorithm. This is because he played 31% of available minutes, and his value depreciated 53%, from €150 million to €80 million. He also moved from the English Premier League to the Spanish La Liga.

Transfers like these were the inspiration for the European Transfer Prediction System. Football clubs need to make decisions like these every six months, and a failed transfer costing €150 million euro could have serious consequences for a club's future success.

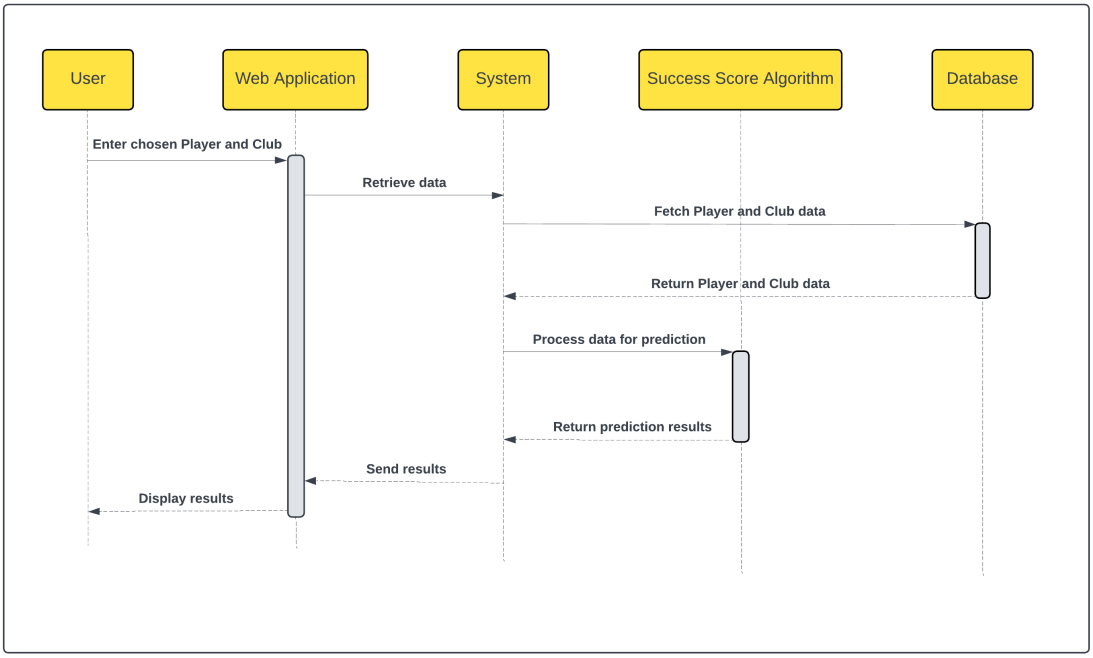

Figure 3.6: UML Sequence Diagram of Transfer Prediction Feature

3.5 System Sequence

The design of each major component of the system is explained in the above subsections. To explain the interaction of these components, the UML Sequence Diagrams Figure 3.6 and Figure 3.7 were created.

3.5.1 Transfer Prediction Feature

Figure 3.6 below is the sequence of interactions when the user chooses a transfer to rank by picking a player and club on the system's home page.

- **User Interaction**: The process begins with the user entering their chosen player and club into the web application.
- **Web Application to System**: The web application takes the input and feeds it to the system's back-end.
- **Database Interaction**: Once the chosen player and club is received by the system, the database receives a query to fetch player and club data. The database processes this query and returns the requested data to the system.
- **System to Success Score Algorithm**: The system now has all player and club data, and the prediction process is started. The two models predict the

player's playtime percentage and future transfer value. These values are used with other statistics to return a 'Success Score' of the transfer to the system.

- **Results Delivery**: After the prediction is made, the results are sent back through the system to the web application.
- **Display Results**: Finally, the web application displays the results to the user, completing the interaction loop.

3.5.2 Find Players Feature

Figure 3.7 is the sequence of interactions in the 'Find Player for Your Club' feature. The user inputs their club, specifies a position they would like to fill and their budget.

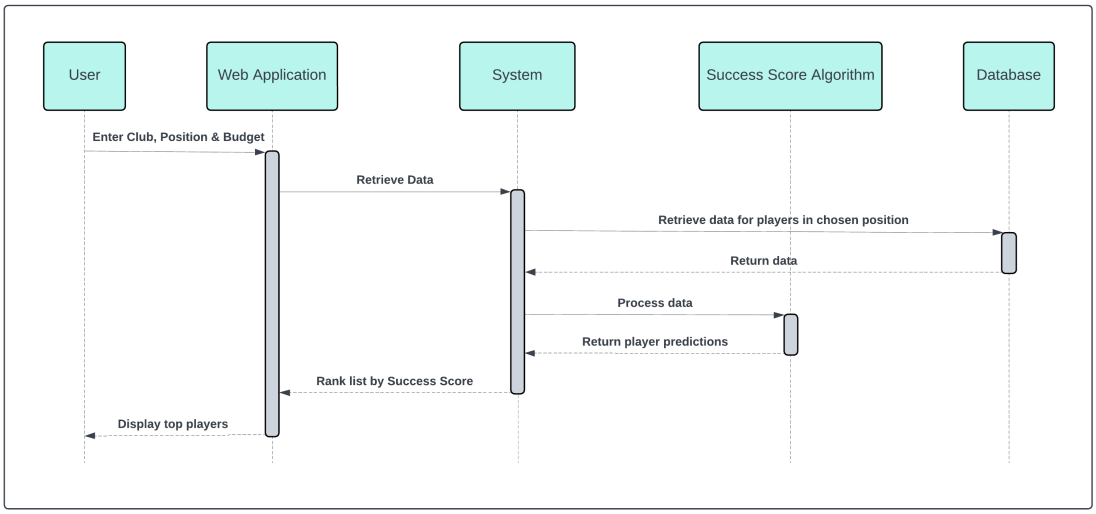

Figure 3.7: UML Sequence Diagram of Find Players Feature

- **User Interaction**: The user interacts with the web application by entering the details of their chosen club, position, and budget.
- **Web Application to System**: The input data is passed on from the front-end to the system's back-end.
- **Database Interaction**: Once the chosen player and club is received by the system, a query is sent to the database to retrieve data for players that match the position specified by the user.
- **Data Processing**: The system then ranks each of the players returned by the database separately, applying the predictions models and 'Success Score' algorithm to each to determine their transfer score.

- **Ranked Results**: The system then ranks the players according to the 'Success Score', age and other relevant factors.
- **User Display**: Finally, the web application displays the top players as a ranked list back to the user, allowing them to observe results.

Chapter 4

Implementation

This section details the key technical components of the European Transfer Prediction System and the challenges encountered during development.

4.1 Implementing the Web Application Front-End

The front-end of the European Transfer Prediction System's web application serves as the interface between the user and the predictive models. Figure 4.1 is a basic UML system architecture diagram detailing the flow of the system.

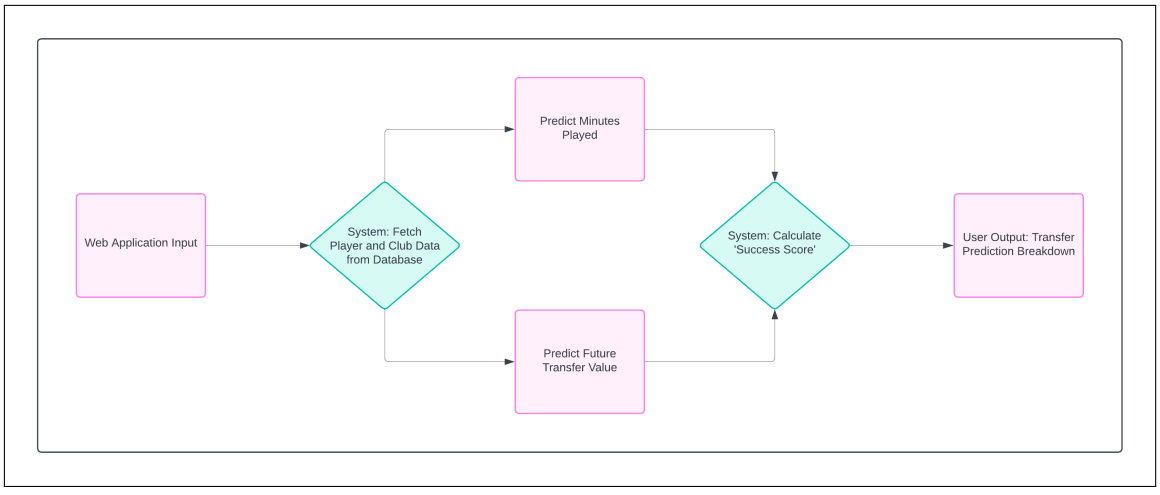

Figure 4.1: Representation of System Architecture

The user-interface was crafted with a focus on usability and aesthetics. The starting goal for the front-end development was that no prior knowledge of the current football landscape is required to use the system. The system aids the user with the auto-complete feature when choosing players and clubs. Figure 4.2 is a screen capture of the user view when typing "lion". This accessibility feature is vital for usability, as names are sometimes difficult to spell even for people with

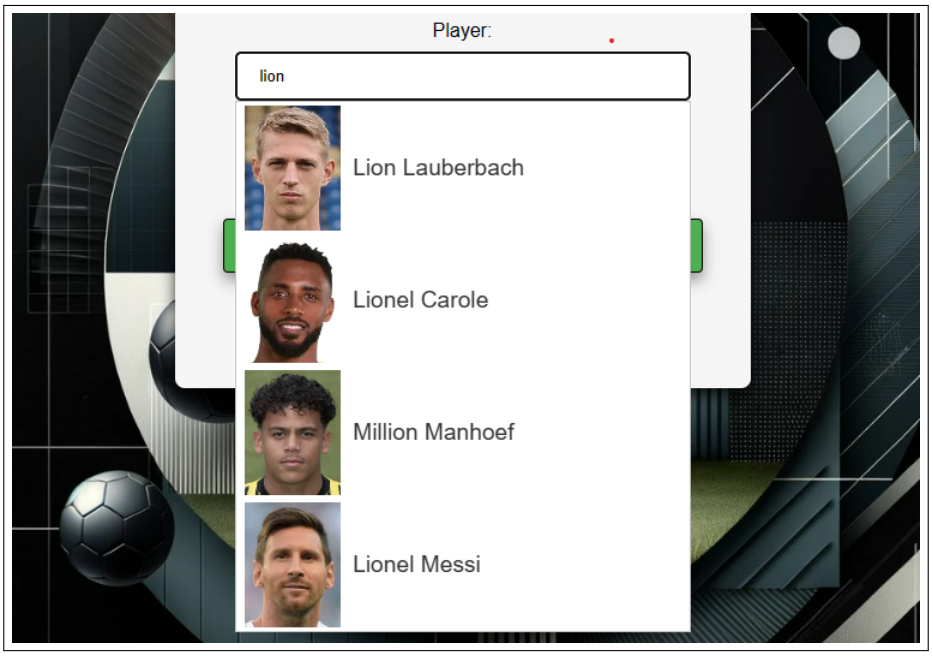

Figure 4.2: Front-End Player Auto-Complete

extensive knowledge of the football industry.

Technically, this is implemented with app routes in the Python program. The python app routes are defined as:

```
@app.route('/suggest/<string:partial_name>')
def suggest(partial_name):
    current_players = stat[stat['season'] == 2022]
    suggestions = current_players[current_players['name'].str.contains(partial_name, case=False, na=False)]
    suggestions = suggestions[['name', 'image_url']].drop_duplicates().to_dict(orient='records')
    return jsonify(suggestions)

@app.route('/suggest_club/<string:partial_name>')
def suggest_club(partial_name):
    club_suggestions = club_data[club_data['club_name'].str.contains(partial_name, case=False, na=False)]['club_name'].drop_duplicates().tolist()
    return jsonify({'suggestions':club_suggestions})
```

The auto-complete is then initialised in a JavaScript function:

```
    $(document).ready(function() {
        $("#player").autocomplete({
            source: function(request, response) {
                $.ajax({
                    url: suggestPlayerURL + request.term,
```

```
                    dataType: "json",
                    success: function(data) {
                        response($.map(data, function(item) {
                            return {
                                label: item.name,
                                value: item.name,
                                image: item.image_url
                            };
                        }));
                    }
                });
            },
            minLength: 2, // Trigger the autocomplete with at least two characters.
            select: function(event, ui) {
                // This event is triggered when a selection is made.
            }
        }).autocomplete('instance')._renderItem = function(ul, item) {
            return $("<li>")
                .append("<div><img src='" + item.image + "' style='width: 80px; height: auto; margin-right: 10px;'><span>" + item.label + "</span></div>")
                .appendTo(ul);
        };
```

Implementing the attached player images to the auto-complete list was a small challenge but adds a large impact to the user experience. A similar auto-completion occurs when a club is inputted in the club field. Unfortunately, club logos were not available in the data. If the project had a longer time-span, the addition of club logos in the web application's input and output pages would be a great addition to the aesthetics of the system.

The front-end of the web application is made up entirely of HTML, CSS and JavaScript. This decision made to make the web application as lightweight as possible. Any dynamic variable such as 'Success Score', player image and details of the prediction models are unpacked through the flask main app route. The success score circle colour is calculated on a gradient from bright green to bright red. Figure 4.3 shows how the colour changes depending on score.

This representation of the score gives the user a clear indication of how successful or unsuccessful the system rates the transfer at a glance.

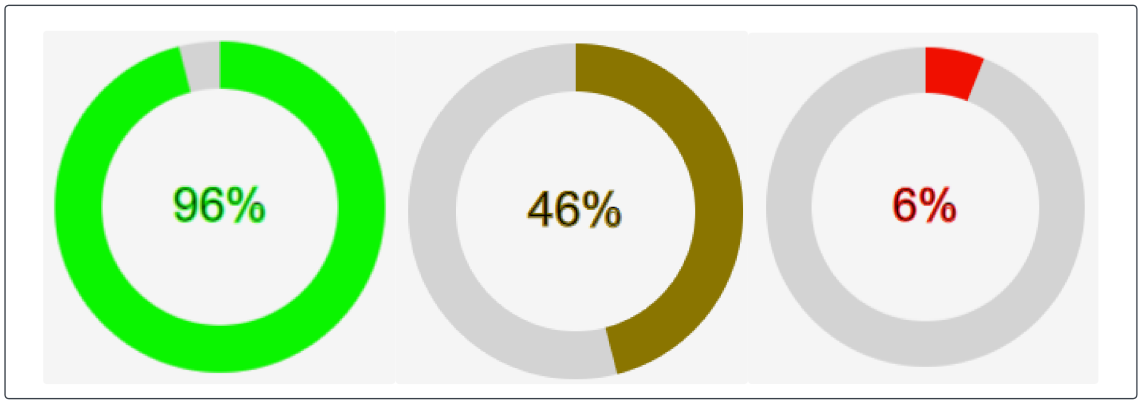

Figure 4.3: Representation of Transfer Success Score Colour Change

Along with this, each key metric passed to the front-end is highlighted by a colour, representing a positive or negative impact on the transfer's success score. This is shown in Figure 4.4.

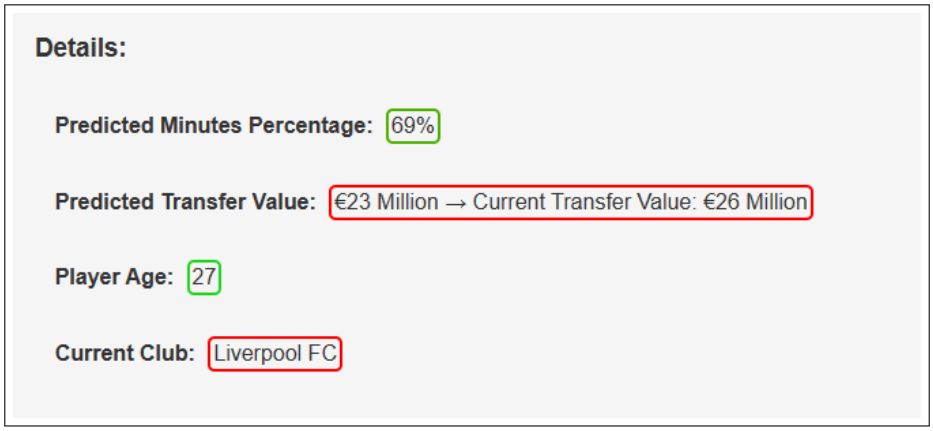

Figure 4.4: Representation of Dynamic Key Metrics

The dynamic highlighting of metrics is implemented in a JavaScript function. The colour variables are defined in the 'styles.css' file. *Predicted Transfer Value* will be used as an example to show the code functionality.

```
var valueAtTransfer = {{ value_at_transfer | tojson }};

window.onload = function() {
                var valueChangeClass = "";
                if (playerPredictedValueRounded > valueAtTransfer) {
                    valueChangeClass = "value-risen";
                } else if (playerPredictedValueRounded <
    valueAtTransfer) {
                    valueChangeClass = "value-fallen";
                } else {
                    valueChangeClass = "value-maintained";
```

```
11                  }
12
13                  document.getElementById('transferValue').classList.add(
    valueChangeClass);
```

The Python variables are first loaded into the JavaScript file. The function creates an empty variable called 'valueChangeClass', and checks if the player's value is predicted to rise, fallen or maintain. It then applies a class name to the empty variable, which is loaded into the HTML div with the corresponding id. When this is completed, the CSS file automatically applies styling.

Errors handling is implemented in the web-application for these cases:

- Player does not exist in the data.
- Club does not exist in the data.
- Player is already at the chosen club.

For each of these cases, a custom message is sent to the user, informing them of the problem. This error handling allows the user to have the complete knowledge of why their transfer did not get a rank, and allows them to return to the form to correct it. The player specific errors are automatically bypassed in the "Find Players" feature, as the system will filter out players with no data or that already play for the chosen club.

4.2 Implementing the Models

This section will detail how the models were trained and detail how these models make their predictions. The models were implemented and tested using *scikit-learn*. The results of the model testing are examined in the "Evaluation" section.

4.2.1 Building the Models

First, the features were selected after finding from the above literature review and correlation matrices. The model is then defined as a 'LinearRegression()'. The data was then split into a train-test split using the imported scikit-learn module. We will look at the process used to build the 'minModel', as the training process was similarly conducted for both models.

```
from sklearn.model_selection import train_test_split

# Feature selection
features = ['age', 'avg_minutes_percentage', 'minutes_played', 'MP', '
    games', 'goals_against', 'goals', 'goals_for','assists','
    clean_sheet','highest_market_value_in_eur','previous_value']

X = stat[features]
y = stat['minutes_percentage']

X_train, X_test, y_train, y_test = train_test_split(X_scaled, y,
    test_size=0.2, random_state=42)
```

After some processing, such as dropping rows with no value, the final training data and the target variable are fit to the model. A prediction of the target variable is then made on each row of the test dataset.

```
minModel.fit(X_train, y_train)

y_pred = minModel.predict(X_test)
```

4.2.2 Using the Prediction Models

The process of making predictions in the system involves several steps, executed through the *prediction_function*. This function begins by filtering data for the specified player and club from the dataset, and confirms the selected player is still active. If valid data is found, the function retrieves detailed information about the player and the prospective club, including the club's average match participation and the league in which the club competes.

```
def prediction_function(player, club):
    # Filter stat for the selected player and club
    player_data = stat[(stat['name'] == player) & (stat['season'] ==
    2022)]
    club_df = club_data[club_data['club_name'] == club]

    features = ['age', 'avg_minutes_percentage', 'minutes_played', 'MP'
    , 'games', 'goals_against', 'goals', 'goals_for','assists','
    clean_sheet','highest_market_value_in_eur','previous_value']

    # Find details for new club
    average_MP_for_club = club_df['club_average_matches'].iloc[0]
    club_league = club_df['domestic_competition_id'].iloc[0]
```

Next steps are to preparing the data for prediction. This includes extracting and scaling relevant features such as age, goals scored, assists, and previous market values. The scaled data is then fed into the two models.

```
    # Update 'MP' for the player
    player_data['MP'] = average_MP_for_club

    # Prepare data for prediction
    X_player = player_data[features]
    X_player_scaled = scaler.transform(X_player)

    # Predict minutes percentage and transfer value
    player_predicted_minutes_percentage = minModel.predict(X_player_scaled)[0]
    player_predicted_value = valueModel.predict(X_player_scaled)[0]
    player_predicted_value_rounded = np.round(player_predicted_value / 1_000_000) * 1_000_000
```

After the prediction is made and saved to a variable, the function does two things. First, it uses the output of these predictions in the calculation of the final success score. This will be detailed in the section following this one.

Secondly, the function formats each prediction in a user-friendly way, allowing this to be sent to the front-end and displayed to the user. The formatting is done as follows:

```
    player_predicted_minutes_percentage = min(player_predicted_minutes_percentage, 1.0)
    formatted_minutes_percentage = "{:.0f}%".format(player_predicted_minutes_percentage * 100)
    predicted_minutes_color = calculate_color(player_predicted_minutes_percentage * 100)

    # Format transfer value in millions
    formatted_transfer_value = "{:.0f} Million".format(player_predicted_value_rounded / 1_000_000)
    formatted_previous_value = "{:.0f} Million".format(value_at_transfer / 1_000_000)

    # Prepare results string using the formatted values
    prediction_results = f"{player} to {club} = Success Score:"
```

This gives useful variables that are formatted strings, and can be used in the front-end when passing data through. This makes the process streamlined.

4.2.3 Implementing 'Success Score' Calculation

The innovative design of a transfer's *Success Score* was the most challenging aspect of this project. The process used to calculate Success Score is the based on the research conducted in this project, and knowledge of the game.

First, the system designates the weighting of the algorithm. The algorithm assigns base weights of 47.5% each to the player's predicted market value (weight_value) and playing minutes percentage (weight_minutes). These weights are adjusted for players above 32 years or below 21 years of age due to their typically varied performance trajectories. The Design section of this report explains the reasoning behind these choices.

```
1
2 def calculate_success_score(predicted_value, value_at_transfer,
    predicted_minutes_percentage, player_age, league_weight):
3     weight_minutes = 0.475
4     weight_value = 0.475
5
6     # Adjust weight_minutes based on player's age
7     if player_age > 32 or player_age <= 20:
8         weight_minutes = 0.3
9
10    if player_age > 32:
11        weight_value = 0.3
```

The system then carries out the necessary calculations with the prediction model outputs. It computes the ratio of the predicted transfer value to the previous transfer value, capped at 1. The Success Score is then calculated by multiplying the value ratio and the minutes percentage by their respective weights, adding the two products together along with any league-specific weight adjustments.

```
1
2     if value_at_transfer > 0:
3             value_ratio = predicted_value / value_at_transfer if
    predicted_value < value_at_transfer else 1
4         else:
5             value_ratio = 0
6         valueX = value_ratio * weight_value * 100
7         valueY = min(predicted_minutes_percentage, 1) * weight_minutes
    * 100
8         valueY = min(valueY, 47.5)
9
10        success_score = valueX + valueY + league_weight
```

This total is then rounded and capped at 100 to produce a final success score.

```
    success_score = round(success_score)
    success_score = min(success_score, 100)
    return success_score
```

Chapter 5

Evaluation

This section of the report provides a comprehensive evaluation of the European Transfer Prediction System, assessing its effectiveness and efficiency in fulfilling the project's objectives. It reviews the system's performance through various testing stages, outlines known issues, and measures the system's success against its initial goals. The evaluation aims to verify the reliability of the software, its functionality, and its user experience to ensure that the system is fit for its intended purpose.

5.1 Model Testing

This section evaluates the accuracy and robustness of the machine learning models developed for the system. Comparisons with other models in the existing body of work will be highlighted in the Conclusions section. The model testing was done by using the models to predict all player's *percentage of total minutes played* and *transfer value* for the current season. These predictions were then tested using the scikit-learn imported testing modules.

```
from sklearn.metrics import mean_squared_error, mean_absolute_error,
    r2_score

print("Mean Absolute Error (MAE):", mean_absolute_error(y_test, y_pred)
    )
print(f'Root Mean Squared Error:', mean_squared_error(y_test, y_pred,
    squared=False))
print("R-squared:", r2_score(y_test, y_pred))
```

The three tests conducted on the models are explained below. These statistics prove that the models are accurate, non-random and fair in their predictions.

1. **Mean Absolute Error (MAE)**: This is the average of the absolute errors between the predicted values and the actual values. An error here refers to the difference between the observed value and the model's predicted value. A lower MAE suggests a model with better performance.

2. **Root Mean Squared Error (RMSE)**: This is the square root of the average of the squares of the errors. The error is the amount by which the prediction of the model differs from the actual value of the target variable. A smaller RMSE value indicates a model that predicts more closely to the actual data, while a larger RMSE value indicates larger errors in prediction.

3. **R-squared ($\mathbf{R^2}$)** : This is the coefficient of determination and measures the proportion of variance in the dependent variable that is predictable from the independent variables. R^2 values range from 0 to 1, where 0 means the model does not explain any of the variability of the response data around its mean, and 1 means it explains all the variability.

5.1.1 'minModel' Testing

```
1    Mean Absolute Error : 0.0946
2    Root Mean Squared Error: 0.1437
3    R-squared : 0.8611
```

The above are the results of the 'minModel' evaluation. Overall, these results are strong, indicating the model is accurate, and the data is a good fit for the model.

- MAE : On average, the predictions of the the model deviate from the true values by about **0.0949**, or approximately **9%**. This means that the errors are quite low on average, showing the robustness of the model.

- RMSE : The RMSE is a measure of the average magnitude of the errors. A RMSE of **0.1437** means that typically, the model's predictions are off by about **14%** from the actual data points.

- R^2 : An R^2 of **0.861** suggests that approximately **86.1%** of the variability of the dependent variable has been accounted for by the model. This implies a strong explanatory power, indicating that the model fits the data well.

5.1.2 'valueModel' Testing

The above are the results of the 'valueModel' evaluation. Overall, these results are weaker than 'minModel', as it has been proven that features have a stronger correlation with playtime than value, and there is an element of opinion that must be considered when predicting value. The results are still strong however, with good indications of a player's value consistantly predicted.

```
1  Mean Absolute Error : 3763992.38
2  Root Mean Squared Error: 6650703.94
3  R-squared : 0.6976
```

- MAE : An MAE of **3,763,992.38** means that, on average, the model's predictions are about **€3.76 million** away from the actual player values, which is quite low.
- RMSE : An RMSE of **6,650,703.94** suggests that there's about **€6.6 million** of difference between prediction and actual values when emphasising the outliers. This is also quite low, which suggests low variation in the model's ability to predict the player values accurately.
- R^2 : An R^2 of **0.6976** indicates that approximately **70%** of the variability in player values can be explained by the model. While not a perfect fit, a value above 0.5 is considered well fit with the inherent variability in the data.

5.2 System Testing

The objective of the system testing phase was to evaluate the predictive accuracy of the European Transfer Prediction System against real-world transfers. As the data used for this project was capped at October of 2023, it has opened the opportunity to test the system using the most recent January transfer window. For this testing phase, three of January's highest profile transfers was selected based on their relevance. This selection aims to test the robustness of the system across these transfers.

5.2.1 Eric Dier *Tottenham Hotspur* → *Bayern Munich*

Eric Dier was transferred from Tottenham Hotspur F.C. to F.C. Bayern Munich in January. His transfer announcement was met with a mixed reception, as Dier was not performing to a great standard during the final season of his Tottenham career. However, Dier's transfer has turned out to be a great success, as his signing contract

until the end of the 2023/24 season has been extended for a further season after his good performances [4]. F.C. Bayern Munich have since reached the semi-final of Europe's elite club competition, the UEFA Champions League, beating top teams such as Lazio and Arsenal. Eric Dier has featured in all these games, conceding only three goals in these four knockout legs.

This transfer was rated **94%** by the system. This is an impressive prediction by the system, as most opinion on the transfer had been negative, such as articles like *"Eric Dier to Bayern Munich: How the hell did that happen?"* by Walsh [17] and many more.

5.2.2 Kalvin Phillips *Manchester City → West Ham United*

Kalvin Phillips was transferred from Manchester City F.C. to West Ham United on loan in January until the end of the season. This transfer was initially well received by West Ham United media and supporters, but since his transfer, has struggled to make an impact [11].

This transfer was rated **52%** by the system. This reflects the transfer accurately. Phillips has featured in many of West Ham United's matches, but has produced a series of sub-par displays that have hampered his side. His poor injury record was also considered by the predictive models, as he is predicted to feature in 44% of West Ham United's minutes. Currently, he has played in 48% of available minutes for West Ham United, which proves this prediction to be quite accurate.

5.2.3 Radu Drăgușin *Genoa CFC → Tottenham Hotspur*

Radu Drăgușin's January transfer from Genoa CFC to Tottenham Hotspur was greeted with skepticism due to his limited appearances in Serie A and unproven status at the top level. Despite his potential, Drăgușin struggled to find regular playing time at Genoa, making his transfer a considerable gamble for Spurs. [3].

This transfer was rated **8%** by the system. Drăgușin is primarily seen as a developmental player rather than an immediate first-team starter. This brings the score down. The system always predicts a player to a first choice player in their new club, so players with little history of consistent playtime will not be rated highly by the system.

5.3 Evaluation of Project Goals

This section evaluates the extent to which the initial objectives of the European Transfer Prediction System project have been met. The primary goal was to develop a predictive system capable of accurately evaluating the potential success of football player transfers using historical data, with a focus on usability and user-friendliness. This has been achieved. A stretch goal of a feature to find the best players for your club has also been met, and the system performs this function above expectations. These goals are explored below:

5.3.1 Development of Predictive Models

Robust machine learning models were required to achieve the primary objective of this project. Two models were successfully built and trained using the 'Transfermarkt' dataset, focusing on features such as player age, previous performance metrics, and historical transfer values. The models achieved a high degree of predictive accuracy, as evidenced by statistical validations which showed promising results for both mean absolute error and R-squared values.

5.3.2 'Success Score' Algorithm Evaluation

The system successfully calculates a success score for input transfers, based on the two machine learning models and other metrics such as age and league. The system's accuracy when calculating success score was validated against historical transfer from up to January 2024, demonstrating its reliability and effectiveness in forecasting transfer successes. The system not only presents a transfer's predicted success, but also educates its users by highlighting key metrics and insights derived from the model's predictions. This transparency helps users understand the reasoning behind each 'Success Score'.

5.3.3 Development of 'Find Players' Feature

In addition to the main objectives, the project included a stretch goal of allowing users to specify a club, position, and budget to find the best-suited players for transfer. This feature has been implemented, enhancing the system's utility and aligning with the project's vision of providing comprehensive support in player recruitment.

5.4 Evaluation of User-Experience

The implementation of this system within a user-friendly web application has allowed users to input transfer scenarios and receive insightful predictions, thus simplifying the access to advanced data analytics in football. During the University College Cork Computer Science Open Day 2024, visitors had the chance to use the system, and opinions on the system's usability were collected. The feedback collected was overwhelmingly positive, with all users able to use and understand both features offered by the system without any explanation. Both users with football knowledge and without were asked for feedback, and the users with little knowledge of the industry had no problem using the system. This positive feedback confirms that the system is not only functional and robust but also aligns well with the needs and expectations of its intended audience.

5.5 Comparisons to Established Systems

Comparing results of this project to existing systems that offer contrasting approaches and insights is needed for a thorough evaluation of the product. This comparison will be carried out on three of the systems that inspired the European Transfer Prediction System: PlayeRank, Tattersall's model, and Melvang et al.'s AI predictions. Benefits and drawbacks will be given for the European Transfer Prediction System against each of these below.

5.5.1 Comparison with PlayeRank

PlayeRank by Pappalardo et al. (2019) offers a robust analysis of player performance using data-driven features. While it provides a detailed evaluation of players' actions during matches, it does not factor in the economic aspects of player transfers, which are crucial for assessing the financial impact and success of transfers [12].

Pros of European Transfer Prediction System:

- Uses real, historical data from Transfermarkt, ensuring predictions are based on actual player performances and not simulated potentials.
- Likely more reliable for professional use in sports management due to its grounding in real-world data.

Cons of European Transfer Prediction System:

- Might be less intuitive or engaging for users familiar with the FIFA gaming metrics, potentially limiting its appeal to a broader audience. The system is targeted to real world application.

5.5.2 Comparison with Tattersall's System

Tattersall's Model uses the xgboost algorithm based on the FIFA video game's 'potential' ratings, which may not always reflect real-world scenarios accurately. This model's reliance on simulated data might limit its applicability in actual transfer decisions [14].

Pros of European Transfer Prediction System:

- Incorporates economic metrics such as transfer values, providing a more comprehensive tool for clubs to assess both performance and financial implications.
- Extends utility by aiding financial decision-making alongside performance evaluation.

Cons of European Transfer Prediction System:

- Require more complex data handling to integrate both financial and performance data effectively compared to PlayeRank's singular focus on performance.

5.5.3 Comparison with Melvang et al.'s A.I. Predictions

Melvang et al.'s AI Predictions focus on the suitability of players for new clubs based on their roles in previous teams. While insightful, this approach may not fully consider the financial aspects of player transfers.

Pros of European Transfer Prediction System:

- Combines role suitability with financial impact analysis, offering a dual perspective that can more comprehensively guide transfer decisions.
- Provides a 'Success Score' that encapsulates both performance potential and economic viability.

Cons of European Transfer Prediction System:

- Individual player statistics are predicted in a more complex model, leading to more accurate predictions.

Chapter 6

Conclusions

6.1 Project Summary

The primary accomplishment of this project was to develop a predictive system, the **European Transfer Prediction System**, that evaluates the potential success of football player's transfer to another club. The system can also suggest transfers that would be successful for a club, given certain parameters. By using the 'Football Data From Transfermarkt' dataset, this system uses player and club statistics to simulate the outcomes of future transfers. Central to the system is the 'Success Score', a data-driven algorithm that quantifies the likelihood of a transfer's success by using machine learning models that predict a player's total minutes played at a new club and their prospective transfer value. This algorithm has been designed with extensive research into what makes a transfer successful, and what factors introduce risk.

This system is designed to support football club decision-makers in refining their recruitment strategies, helping to maximise returns on player investments and reduce financial risks. Testing has been carried out on the system, validating the accuracy of its predictions against known outcomes, such as key transfers in the January 2024 transfer window. The project was executed through the development of a web application that allows users to select a player and club from the database and assess the potential success of the transfer. The application was designed to require minimal football knowledge for its use, enhancing its accessibility, but knowledge of the current football landscape allows users to get useful predictions from the system to use in the football transfer market.

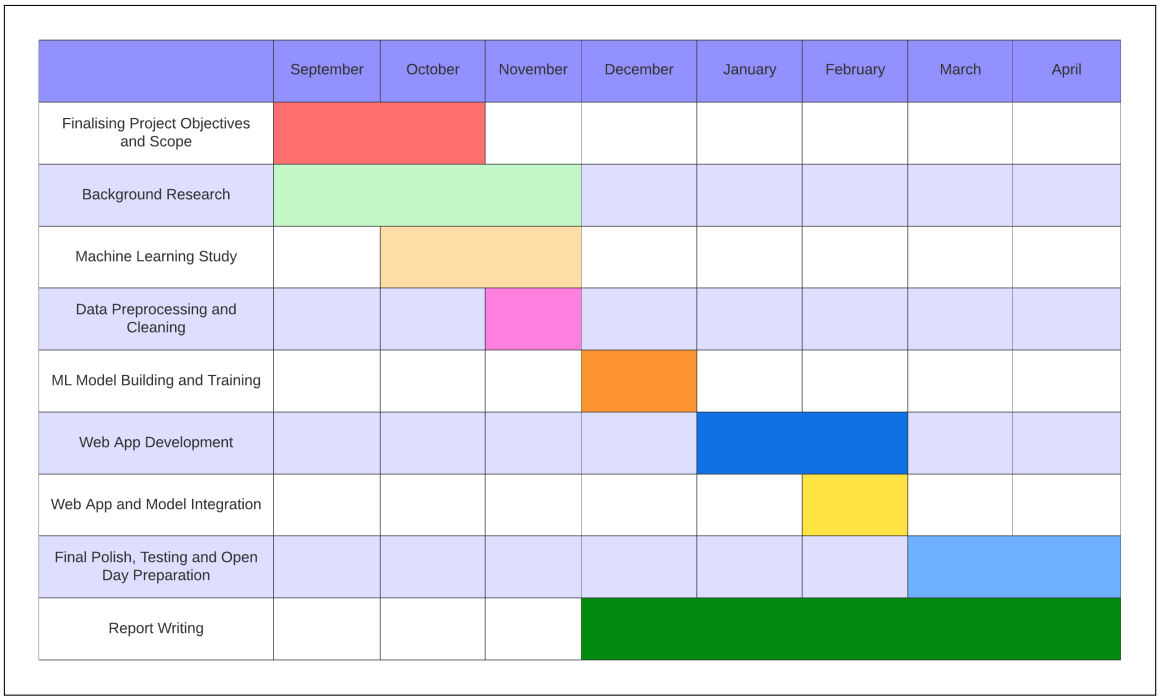

Figure 6.1: Gantt Chart of Project Timeline

The project was conducted in phases, starting with a stage of defining the scope and objectives of the project. This was joined with a research phase, which helped defining goals to fill gaps in the body of existing knowledge. After this phase, learning ML principals was vital to achieve objectives, followed by data preprocessing, building and training the models, developing the web app and implementing the integration of the model in the web app. This report was written concurrently with the phases from an early stage. Each phase was carefully planned; however, some tasks took longer than anticipated, particularly finalising the appearance of the interface, which impacted the subsequent phases. Figure 6.1 is a Gantt chart illustrating the project timeline.

6.2 Contributions to Knowledge

The project contributes to the field of sports analytics by merging performance data with monetary factors to predict the success of football transfers. This approach helps to fill a gap in existing transfer prediction models, which often focus solely on player performance or market value. The European Transfer Prediction System advances the application of machine learning in sports, providing actionable insights with a level of accessibility that is not commonly found in existing systems.

Unlike other analytical tools that often require deep technical knowledge or a strong background in data science, this system is designed with user-friendliness in

mind. It enables users with minimal football knowledge or analytical expertise to interact with the system. One of the standout features of this system compared to others in the field is the clarity and understandability of its output. The 'Success Score' metric, a central element of the system, is presented in a way that users can easily comprehend what the scores signify and how they are calculated. This contrasts sharply with some existing models, which either do not prioritize user experience or present results in formats that non-specialists find challenging to interpret. The project contributes to bridging the gap between statistical analyses and practical insights within the football community.

6.3 Future Developments

This project represents a meeting of data analytics and football transfer strategy. However, there are a vast number of options for future development on a prediction system like this. Future developments could take several forms:

1. **Algorithmic Refinement**: Continuous improvement of the predictive algorithms through the incorporation of deep learning or more specialised machine learning models could provide even more accurate predictions.

2. **New Leagues and Regions**: Currently, the system uses data from Europe's top seven leagues, rated by 'Transfermarkt'. The systems database could be expanded to every league in Europe, or every regions in the world.

3. **Cross-sport Application**: The 'Success Score' algorithm was specially designed for football, but the algorithm could be adapted for use other sports with significant transfer markets, such as basketball or baseball. In theory, even applications outside of sport that with usable data and features that can be predicted would be possible to adapt the system too, such as hiring of employees at businesses.

4. **Market Trends Analysis**: Integrating a feature to analyse and predict market trends could provide clubs with strategic insights into when to buy or sell players for maximum financial gain. This feature would allow to system to cover the entire football transfer needs for a club.

6.4 Personal Reflections

Throughout this project, I have gained invaluable experience in data preprocessing, machine learning, and web application development. As stated above, this project was my introduction to machine learning, so a large amount of self-learning helped me to gain new skills and develop the ones gained in my undergraduate degree.

One significant reflection of mine is the importance of research in a large project like this. Conducting a large literature review allowed me to clearly know what I can and cannot achieve with this project, and what makes my project stand out from the existing body of work.

Time management and planning were rewarding but challenging, particularly when aligning the project milestones with academic deadlines. I set out a rough list of tasks to be completed by certain deadlines, represented in the Gantt chart Figure 6.1 above, and these tasks were completed by sticking to the plan.

Upon reflection, I really enjoyed designing and developing the European Transfer Prediction System. My passion for football, along with the challenge that the system posed provided the motivation for me to choose to offer a solution to the transfer prediction problem. I am especially proud of the work done to create the 'Success Score' algorithm, which in my opinion, is the best transfer evaluation metric on the market.

Bibliography

[1] ANALYTICS FC, 2023. Measuring transfer success through minutes played, Analytics FC. Available at: https://analyticsfc.co.uk/blog/2021/10/18/measuring-transfer-success-through-minutes-played/.

[2] Brandsen Sports, 2024. Impact of Data Analytics on Football Team Performance. LinkedIn. Available at: https://www.linkedin.com/pulse/impact-data-analytics-football-team-performance-brandsen-sports/.

[3] Clasper, B., 2024. Problems: Radu Drăgușin's agent speaks out again as player awaits first start at Tottenham. The Spurs News. Available at: https://www.thespurs.news/news/problems-radu-dragusins-agent-speaks-out-again-as-player-awaits-first-start-at-tottenham/.

[4] Ornstein, D., 2024. 'Eric Dier's permanent transfer to Bayern Munich'. Available at: https://theathletic.com/5308278/2024/02/29/eric-dier-bayern-munich-transfer-permanent.

[5] Cariboo, D., 2024. Football Data from Transfermarkt, Kaggle.com. Available at: https://www.kaggle.com/datasets/davidcariboo/player-scores.

[6] Hindle, T., 2023. Eden Hazard: From Chelsea icon to Real Madrid's worst-ever signing and early retirement at 32, Goal.com. Available at: https://www.goal.com/en/lists/demise-eden-hazard-chelsea-icon-real-madrid-worst-ever-signing.

[7] Kologlu, Y. et al., 2018. A Multiple Linear Regression Approach For Estimating the Market Value of Football Players in Forward Position. Available at: https://arxiv.org/abs/1807.01104.

[8] Melvang, J., 2021. Jeremy Doku and using AI to predict the success of transfers, Training Ground Guru. Available at: https://trainingground.guru/articles/jeremy-doku-and-using-ai-to-predict-the-success-of-transfers.

[9] Macro Football, 2024. When Do Football Players Peak? Available at: https://macro-football.com/other/aging/.

[10] Cordeiro, L.G., 2023. Market Value EDA. Available at: https://www.kaggle.com/code/luisgasparcordeiro/market-value-eda.

[11] Pearson, S., 2024. Game over: Kalvin Phillips West Ham talk as incredible Leeds Utd return heats up. TeamTalk. Available at: https://www.teamtalk.com/news/game-over-kalvin-phillips-west-ham-talk-incredible-leeds-utd-return-heats-up.

[12] Pappalardo, L. et al., 2019. PlayeRank: Data-Driven Performance Evaluation and player ranking in soccer via a machine learning approach: ACM Transactions on Intelligent Systems and Technology: Vol 10, no 5. Available at: https://dl.acm.org/doi/10.1145/3343172.

[13] Tysonike, 2023. Finding strikers to replace Aguero at MC in 2016, Kaggle. Available at: https://www.kaggle.com/code/tysonike/finding-strikers-to-replace-aguero-at-mc-in-2016/notebook.

[14] Tattersall, J., 2021. Using machine learning to identify high-value football transfer targets, Medium. Available at: https://towardsdatascience.com/using-machine-learning-to-identify-high-value-football-transfer-targets-d4151a7ffcac.

[15] LFC.com, 2024. Virgil van Dijk - Liverpool FC - Homepage. Available at: https://www.liverpoolfc.com/team/mens/player/virgil-van-dijk.

[16] Sulimov, D., 2024. Performance Insights-based AI-driven Football Transfer Fee Prediction. arXiv preprint. Available at: https://arxiv.org/abs/2401.16795.

[17] Walsh, S., 2024. 'Eric Dier to Bayern Munich: How the Hell Did That Happen'. 90min. Available at: https://www.90min.com/posts/eric-dier-to-bayern-munich-how-the-hell-did-that-happen.

Appendix

Glossary of Football Terms

Football Transfer : The action of a football player moving from one club to another, involving negotiation of a transfer fee and the player's personal terms.

Transfer Window : A designated period during the year when football clubs are allowed to buy or sell players. Typically occurs during the summer and mid-season (January).

Market Value : The estimated financial value of a football player, often determined by their performance, age, contract duration, and market demand.

Success Score : A new metric used to evaluate the potential success of a player's transfer to a new club, incorporating various performance metrics.

Playtime Percentage : A measurement that quantifies the amount of playing time a player receives at a club, expressed as a percentage of the total available minutes.

Transfermarkt : An online database and website that tracks football player transfers, values, and other career statistics.

Predictive Model : Statistical models used in football to forecast outcomes such as a player's future performance or the success of a transfer.

Minutes Played : The actual amount of game time a player has participated in, used as a key indicator of a player's involvement and impact at a club.

Transfer Fee : The amount of money paid by one club to another to secure the services of a football player.

League Compatibility : A term discussing a player's ability to adapt and perform in the specific competitive environment of a new league.

Figure 6.2: Project GitHub

The project's source code and additional resources can be found on the GitHub repository at https://github.com/ros6ucc/Using-Machine-Learning-to-Predict-Successful-Football-Transfers, or by scanning the QR code: