

Sports Data Analytics: Machine Learning Applied to European Football

Thomas Gibbs

Submitted for the Degree of Master of Science in

MSc Machine Learning



Department of Computer Science
Royal Holloway University of London
Egham, Surrey TW20 0EX, UK

June 16, 2014

Declaration

This report has been prepared on the basis of my own work. Where other published and unpublished source materials have been used, these have been acknowledged.

Word Count: 17,153

Student Name: Thomas Gibbs

Date of Submission: 30.08.2023

Signature: T.A.G.

Abstract

This project focuses on data analysis in European football. The goal was to design and implement a multi-purpose scouting tool that could be used by football clubs, scouts or pundits to analyse players from Europe's traditional top five leagues: the Premier League (England); La Liga (Spain); Ligue 1 (France); Bundesliga (Germany); Serie A (Italy). A combination of machine learning methods were employed, alongside full-stack development, to deliver "TAGscout" - a web-application whose interface allows those without prior coding or data analysis skills to interact with the data and its conclusions. TAGscout has two main functions: value prediction and similar player search. Value prediction will use a player's stats from a given season to predict their value the following season. Similar player search will take an input player and return players with similar stats, using a range of parameters to filter the search - these results will then be visualised to give some context. Similar player search shows promise as an exploratory analysis tool, whilst TAGscout's value prediction achieves a test set R^2 of 0.8647. However, further analysis will highlight the drawbacks of evaluation metrics and call into the question the validity of crowd-sourced data.

Contents

| | | |
|----------|--|-----------|
| 1 | Introduction | 1 |
| 1.1 | Motivation..... | 1 |
| 1.2 | Aims of the project..... | 1 |
| 1.3 | Justification of the chosen functionality..... | 2 |
| 2 | Background Research..... | 4 |
| 2.1 | Overview of data analysis in football..... | 4 |
| 2.2 | General techniques in player performance analysis | 4 |
| 2.3 | Techniques specifically in value prediction | 6 |
| 2.4 | Summary of research..... | 8 |
| 3 | Building the dataset..... | 9 |
| 3.1 | Determining the appropriate coding language | 9 |
| 3.2 | Data selection..... | 9 |
| 3.3 | Web-scraping..... | 12 |
| 3.4 | Formatting..... | 14 |
| 4 | Experiemental Results - Value Prediction..... | 15 |
| 4.1 | Overview | 15 |
| 4.2 | Data restructuring..... | 15 |
| 4.3 | Model selection and results | 16 |
| 4.4 | Validity and usefulness..... | 18 |
| 4.5 | Efforts to improve the model | 21 |
| 4.5.1 | Checking the residuals | 22 |
| 4.5.2 | Applying a log transformation to the labels..... | 22 |
| 4.5.3 | Outlier removal and subsets of the data | 23 |
| 4.5.4 | PCA for noise reduction..... | 23 |
| 4.6 | Conclusions from value prediction | 23 |
| 5 | Experiemental Results - Similar Player Search | 26 |
| 5.1 | Methods..... | 26 |
| 5.2 | Implementation | 28 |
| 5.3 | Test cases & justifying the use of visualisation | 29 |
| 5.4 | Conclusions from similar player search..... | 32 |
| 6 | TAGscout - Front-End | 33 |

| | | |
|----------|--|-----------|
| 6.1 | Overview | 33 |
| 6.2 | Low-fidelity wireframes..... | 33 |
| 6.3 | High-fidelity prototype | 35 |
| 6.4 | Integration of Flask with React | 37 |
| 6.5 | Test cases and final implementation | 38 |
| 6.6 | Front-end conclusions | 39 |
| 7 | Professional Issues | 40 |
| 7.1 | General issues | 40 |
| 7.2 | Data-literacy in football..... | 40 |
| 7.3 | Fundamental issues with value prediction | 42 |
| 7.4 | Legal..... | 43 |
| 7.5 | Strategic homogeneity | 43 |
| 7.6 | Summary of professional issues..... | 44 |
| 8 | Conclusion | 45 |
| 8.1 | Project review | 45 |
| 8.1.1 | Amibtion and scope..... | 45 |
| 8.1.2 | Alternative data sources..... | 45 |
| 8.1.3 | Front-end improvements | 46 |
| 8.1.4 | Evaluation of the similar player search function..... | 47 |
| 8.1.5 | Limitations of the value prediction model | 47 |
| 8.1.6 | Strengths of the project..... | 47 |
| 8.2 | Summary | 48 |
| | References | 49 |
| | Appendix A: StatsBomb's NSxG model | 51 |
| | Appendix B: Further visualisations for Antony | 52 |
| | Appendix C: High-fidelity prototypes..... | 53 |
| | Appendix D: User Manual | 54 |

1 Introduction

1.1 Motivation

European football is becoming an ever more lucrative business, with consistent growth in market size only temporarily hampered by the COVID-19 pandemic [1]. Hence, stakeholders of various descriptions are becoming increasingly interested in ways to analyse the game - with the goal being to gain an advantage over competitors. As well as the rapid growth of the European football economy, the world's most popular sport has seen another growth: the growth in the prevalence of data. Many clubs are starting to employ data-driven approaches to transfers, coaching and opposition analysis, with many finding great success. Brighton and Hove Albion and Brentford FC are two clubs that have embraced data and consequently are both punching well above their financial weight, with league positions that run in contrast to their wage bills and net transfer spends [2] [3]. It is therefore not hard to see the benefit of employing the use of machine learning algorithms if this could provide a systematic edge in the modern game. The aim of this project is both to simultaneously prove the usefulness of machine learning in football, as well as provide an entry-level data analysis tool that uses widely available data.

1.2 Aims of the project

From the outset, the project has had three aims. The first is to utilise freely available data. Due to the lucrative nature of European football, a lot of the data on it is seen as extremely valuable and is hence locked behind a paywall. This somewhat acts as a barrier to the growth of data analysis in the modern game - even if it is only by deterring the general public from investigating data-driven approaches more. Hence, no paid for APIs or data packages will be used in this project.

The second aim is that the project needs to be dynamic. The footballing world moves fast and to gain an advantage in it, teams, players and scouts must move even faster. Hence, there is little point in embarking on a project that will be of little use in one or two season's time. This means that the code needs to be written in a way such that it can be updated for current players and statistics with minimal hassle. Furthermore, dynamic code will improve the scalability of the project in the event that this is desired in the future.

The third and final goal is that the project has to be as object-oriented as reasonably practicable. Besides generally making for more structured code, this is important because it allows for as little friction as possible if

the dataset were to be expanded to leagues outside of Europe's traditional top five in the future. Furthermore, it is important for my future career aspirations that the inner workings of TAGscout can be improved and updated easily. Whilst some of the techniques employed in the analysis are rather complex, there are teams of data scientists being paid six or seven figure salaries at major football clubs to perform such analyses. Therefore, TAGscout currently serves as an introductory tool, perhaps one that can be used for less affluent clubs and individuals. However, there is still room for improvement if it were to compete with the top data-science departments in the industry. Hence, the ability to upscale the platform in the future is crucial. On a personal level, my career ambition is to work in sports data analytics, doing exactly this kind of analysis for football teams. It is my hope that the project aids my pursuit of this and for this, a scalable product is not only best-practice, but perhaps will one-day ease my transition into the sports data analytics world.

1.3 Justification of the chosen functionality

In terms of the two functions of TAGscout, each has a unique justification. The flagship feature of the platform is the value prediction model. This employs advanced machine learning techniques to predict the value of a player one season from now by using their current stats.

When you consider that in the 2021-22 season, clubs from the traditional top five European leagues spent €3,855,910,000 on transfers, the benefit of value prediction for clubs becomes fairly self-evident [4]. If a club can accurately predict a player's value one season in advance, this may allow them to obtain high-value players for a discount rate. Furthermore, it can inform a club's transfer strategy and allow them to plan for transfer windows further in advance - without having all their hard work made redundant because a player's value has grown unexpectedly and now falls outside of their budget. This thinking can be applied to numerous problems on the demand side of the transfer market (not overpaying for players, how high to table an opening bid, etc.).

Equally, value prediction can be utilised in the supply side of the transfer market. The algorithm may inform a team's decisions about selling a player that they already own. Why sell a player now if in one season's time his value is expected to rise drastically? Perhaps it may incentivise a club to let a player go for below the current market value, to ensure he leaves before his value drops substantially in the following season. In a market where the sums exchanged are often in the range of millions, foresight around future player value could make all the difference.

The similar player search function of TAGscout provides a slightly more intuitive form of analysis. This has wide ranging implications: in the

transfer market (trying to find replacements for players who have departed); in scouting (trying to find players of a certain profile; in oppositional analysis (you aren't familiar with a certain player but can get a feel for how he plays based on who he is similar to); in punditry (a player is perhaps overrated or underrated); and much more. The filters provide a more customisable approach and the addition of a graphic (alongside returning the names and information of similar players) can help to create an intuitive sense of how significant these similarities are. For example, one player may be similar to another player but these players exist in a large cluster where the similarities may not be that unique. Equally, two players may be shown as similar to a third player, but one may be significantly closer than the other. The visual aid in TAGscout allows more for this kind of assessment.

2 Background Research

2.1 Overview of data analysis in football

Data analysis in football has grown in the last few decades but is still somewhat in its infancy. John Coulson, formerly of Opta, states that “there are still maybe a lot of teams that view data as a threat rather than as a tool”, suggesting that not only is there a need to expand the technical capabilities of data analytics in football, but also a need to justify its use at all [5]. Many of the popular works on data analytics in football to date focus on this latter aim - the justification of data-driven approaches to football. Books such as *Soccernomics*, *The xG Philosophy* and *The Numbers Game: Why Everything You Know About Soccer Is Wrong* dedicate much of their focus to proving that not all conclusions in football are intuitive - and that data can help to uncover these less intuitive facts about the game.

More practical implementations of data analysis and machine learning in football are unfortunately harder to observe due to the lucrative nature of their insights. Whole teams of performance-focused data scientists have been established at football clubs, all with the goal of optimising the club’s on-pitch success, through coaching, training and player acquisition. These departments aim to give their team a competitive edge and hence, don’t freely publish much of their methodology and findings. Whilst some papers have been published from these clubs, such as the research that has been released by FC Barcelona’s “Barça Innovation Hub”, these don’t often focus on data analytics and machine learning methods in football. For this we have to turn to some smaller scale papers published on the subject, detailing more isolated attempts to improve the use of data in the game. Many of these come from independent data enthusiasts, not operating on behalf of the top European football clubs and hence, some of the key concepts that have taken root in football data analytics have started on independent blogs (see expected threat in section 2.2).

2.2 General techniques in player performance analysis

Many attempts have been made to quantify performance. These attempts are trying to address the underlying issue that football is a dynamic game with “events” that are less well defined than in something like baseball (where the Moneyball revolution famously began).

For example, StatsBomb, a leader in football data methods, have begun exploring the use of Markov Models to produce “non-shot expected goals (NSxG)” metrics [6].¹ The model serves as an interesting development in quantifying player performance and even managed to identify some of the

¹ For more details on the methodology used, see Appendix A.

best players in the world by ranking each player's "contribution per game".² However, this method has a drawback. Outside of model specification issues - such as the fact that Markov Models assume previous states have no impact on the current state, an assumption that almost certainly does not hold for a football match - the model also only provides context on how good a certain player is, not how much to pay for them. Of course, value could somewhat be derived from this, but it is not the model's primary function. Whilst the model seems to identify top players adequately, there is also no guarantee that the model will continue to accurately identify quality when less exceptional players are being considered.

Various xG spinoffs that resemble the NSxG model have also been made to quantify performance. Karun Singh introduced "expected threat (xThreat)" in 2019 [7], whilst Nils Mackay suggested "xG added" [8]. But all of these models differ from this project in their end goal. They serve as metrics to be analysed; not direct predictions of value produced by machine learning algorithms. They are interesting in their methodology and will no doubt help to shape football data analytics as a field, but they are not necessarily the methods with the most application to the task at hand in this project.

An alternative methodology that is popular in the research space is that of neural networks. They are a popular choice due to their ability to handle non-linearity, their versatility, and their reduced emphasis on feature engineering. The complicated nature of football as a game means that there are a whole host of metrics that can be analysed and/or fed into a model. Gaining insight into which of these features are useful can provide a significant challenge and hence, manual feature engineering can prove problematic. Deep neural networks can learn the most relevant features of the data independently which is a huge advantage of their use. For example, Fernández et al. used a range of methods including deep neural networks and convolutional neural networks to determine a variety of intermediate-metrics that were used to calculate their *expected possession value* metric. Their model was incredibly complex and evaluated the full spatio-temporal characteristics of the 22 players and the ball among other things [9].

Building models that focus on capturing spatio-temporal complexities is an intriguing (and cutting edge) idea that has been perpetuated elsewhere such as in Gonçalves et al.'s 2019 paper titled "Extracting spatial-temporal features that describe a team match demands when considering the effects of the quality of opposition in elite football" [10].

² To obtain contribution per game, contribution was scaled by the number of matches played. Contribution was defined as: $contribution_i = \sum Pr(Goal | State_{t+1}) - Pr(Goal | State_t) \cdot I(action \text{ by player } i)$

Liu et al. also had a similar focus in their 2020 paper which utilised deep reinforcement learning techniques (notably LSTMs) to design a *Goal Impact Metric (GIM)*. From this they then ranked players in the *English Football League Championship*. However, there are issues with assessing the validity of the ranking provided by Liu et al., and indeed any potential ranking. As the authors themselves state, the main issue is that “there is no ground truth for player ranking” [11].

Due to this project’s goals that include, not only the development of a useful scouting tool, but also the justification of machine learning’s use in football, it is more optimal to consider methods that can prove their accuracy. It is feasible that spatio-temporal data could be repurposed for use in a value prediction mode but unfortunately, this sort of data is hard to come by and violates one of the other key aims of the project: to use free and widely available data. However, the use of deep neural networks in the research, as a general methodology, is noteworthy. In fact, neural networks have also been deployed on less complex data such as the “recommendation engine for opponent scouting” devised by Ruiz et al. They started by training and testing four single-hidden-layer neural network regression models to demonstrate that a teams’ strategy can help to predict the number of shots and goals that they will produce. They then extend this method to estimate the expected shot production (broken down by shot type) that teams will generate in an upcoming game [12].

2.3 Techniques specifically in value prediction

As we’ve seen, many attempts have been made to quantify player performance more generally. However, there have also been several attempts to directly predict market value. Various methods have been used but most of these are based on data from the popular EA Sports video game series *FIFA*.

For example, in 2017 Sourya Dey conducted an analysis using data from the *FIFA 17* video game [13]. Dey constructs a multilayer perceptron neural network with 41 input features which is then used to predict one of the game’s 119 the pricing categories for each player. The overall model accuracy was 87.2%. However, it is important to note two issues with this study. The first is that FIFA assigns one of these 119 values to each player - making the task more of a classification analysis than a regression. This means that EA’s simulation of the transfer market doesn’t fully reflect the nature of the real-world transfer market - something that may cause an issue for any clubs hoping to inform their transfer strategy with this model.

The second issue is one of potential data snooping. The value that is being predicted is EA’s value, not necessarily the player’s true market value.

Given that the data Dey uses for the prediction also comes from EA, this may introduce a systematic bias into the model. It is likely that EA has a dataset that is used to produce player ratings, the individual attributes of each player and their valuation in the game. Therefore, using the game's data for each player to predict the game's value for each player threatens to cause some unintentional data snooping issues.

However, this is not the only paper that uses video game data. The paper "Predicting the Football Players' Market Value Using Neural Network Model: A Data-Driven Approach," published in 2022, also used neural networks to predict a player's market value. The model was based on data from EA Sports' FIFA 19 video game and the final incarnation of the model managed to achieve impressive values of 0.96 and 0.95 as the training and test set R^2 respectively [14].

In the same year, Al-Asadu and Tasdemir took a range of approaches to tackling a similar question, using data from the FIFA 20 video game (obtained via sofifa.com) to predict market values. They used four different models: linear regression, multiple linear regression, decision trees and random forests. The random forest technique outperformed the others, producing a test set R^2 value of 0.95 [15].

These are all interesting studies into transfer value prediction, particularly due to the range of techniques that they employ. However, their consistent use of video game data raises an interesting question about the bias that this will introduce into the models.

Stanojevic and Gyarmati's 2016 paper followed a slightly different methodology. They used a "gradient boosting trees" regression and grid-search (for hyper-parameter selection) to develop their "performance driven market value" based on data from InStat. They then compared this to Transfermarkt's estimated market value and found that the median and mean difference were around 34% and 60% respectively. They outlined several reasons for this discrepancy such as the vast range of values that the analysis spanned and the lack of ability to include commercialisation capacity or injury-proneness, which reflect the difficulty of the task at hand. [16]

There does exist a somewhat more direct competitor to TAGScout. SciSports is a football analysis software that aims to provide data-driven insights [17]. Within the platform there are multiple functions including a value prediction model. However, TAGScout differs from SciSports in several important ways. Firstly, it is not clear if SciSports offers the ability to predict future market value, the examples that SciSports showcases are for predicting the value of a player in the current season. Furthermore, the techniques used to calculate this value are not published because, as SciSports offers a software product, that would jeopardise their business model. This makes it hard to assess the accuracy of their models or

scrutinise the techniques used to generate a predicted value - something that prevents them from meaningfully contributing to the research space. However, despite all of this, the most significant difference between TAGscout and SciSports is that TAGscout has the very explicit goal of utilising freely available data in an attempt to expand the use of data within European football. SciSports' cheapest recruitment package costs €899 per month. This price tag dictates that SciSports' products are not ones which encourage an increase in data-literacy amongst fans and pundits. Hence, whilst SciSports and TAGscout do share a lot of functions, their fundamental aims are slightly different - with TAGscout serving more as a research tool than a packaged and priced software product.

2.4 Summary of the research

The current state of the research can be summarised by the following categories:

- Works that attempt to justify the use of data in football.
- Works that focus on improving performance through coaching and ensuring optimal fitness.
- Efforts to improve the ability to quantify player performance but not in a way that can be directly and comprehensively scaled and applied to a large dataset of players in various positions.
- Value prediction algorithms that mainly appear to utilise video game data instead of real-world data.

This leaves a notable gap in the research. Firstly, the research space that focuses on value prediction is not as extensive as the space dedicated purely to the derivation of new performance metrics. Furthermore, what research there is rarely utilises real world data. This leaves room for further research into the application of machine learning to football market value prediction problems using real world data.

The second gap in the research is more subtle. Essentially all papers on market value prediction in football focus on predicting a player's market value in a given season, using data from that same season. This has some practical uses but predominantly serves as a demonstration of the capability of machine learning and data analytics in football. However, it does not have the same use case as the tool that this project seeks to design. Predicting a player's value based on their statistics in the same season does not necessarily enable clubs to plan for the future as much as the model that this project describes, which uses data from previous seasons to predict a future value. So far, there does not appear to have been a meaningful attempt to predict the growth (or decline) in a player's market value across seasons. This is the gap that this project will aim to fill alongside the broader aim of producing a practical scouting tool and further justifying the use of machine learning methods in elite football.

3 Building The Dataset

This section will cover the approach to determining the appropriate tools, finding data that satisfies the requirement of being freely available, web-scraping the data and formatting it in an object-oriented way.

3.1 Determining the appropriate coding language

The first decision of the project was to settle on an appropriate coding language to work with. The two most popular languages in data analysis are R and Python so the search was narrowed to these two. In the end, Python was deemed to be favourable for a few reasons:

1. Python is usually more suitable for machine learning and deep learning due to vast libraries such as scikit-learn and PyTorch - both of which I had plans to utilise.
2. Python is more commonly viewed as the industry standard, and hence for my future career, but also for the future compatibility of TAGscout, Python appeared more optimal.
3. R's GPU support is less reliable than Python's - something that could prove to be limiting if and when deep learning techniques came to be utilised.
4. Finally, and perhaps most importantly, the ultimate goal of the project was to create an intuitive and accessible way to interact with the data. Most of the key decision makers in football do not have a background in statistics and data analysis and hence, for TAGscout to be of any real use in industry, it would need a suitable front-end. With regards to front-end integration Python is often better supported than R, for example, frameworks such as Flask are very reliable and well maintained for Python.

3.2 Data selection

After settling on a coding language, the next challenge step was to build a dataset with freely available data. For this, three components were needed: general information about a player; data about a player's performance over the season; and a player's transfer value.³

Most sources of player performance data also include basic information about the player, so this didn't need separate consideration.

In terms of the player performance data, this needed to be extensive - more than simply goals, assists, clean sheets, minutes etc. FBRef was chosen as the most appropriate source of these metrics, providing one hundred

³ General information includes things like name, age, club, nationality, etc.

different on-pitch metrics per player across five different web pages.⁴ Furthermore, in terms of players appearing on all five pages, FBRef provided information on: 2723 players in the 2022-23 season; 2790 players in the 2021-22 season; and 2704 players in the 2020-21 season. This covered players ranging from first-team players all the way down to youth players with minimal appearance time. Unfortunately, FBRef does not offer a convenient way to obtain more granular per-game data. This could have been used for more in-depth analysis including how players perform against top teams. However, the only way to obtain this data would have been to navigate to each player's personal profile and then to their match logs for each set of statistics and web-scrape this. This would have made the program very computationally expensive as this would need to be done for at least all 2723 players in the 2022-23 season and all 2790 players from the 2021-22 season, across five attribute categories, using an advanced (but slower) web-scraper such as Selenium to interact with the pages dynamically. Furthermore, to truly implement any meaningful analysis on how players have performed against top teams, the league tables or form tables would need to have been web-scraped as well. Without this, generalisations would have to be made about "top teams" which may vary season to season - for example, Chelsea would likely be considered a top team but finished 12th in the 2022-23 season. In all, this was deemed to be outside the scope of the current project but remains an interesting avenue to explore should TAGscout be expanded in the future.

SoFifa and Sofascore were both considered as alternatives to FBRef. Sofascore is a betting odds site with some player data as well, however, it was deemed to be less comprehensive than FBRef in terms of the player statistics that it offered. On the other hand, SoFifa is a site that stores data about players from EA Sports' FIFA games. The advantage of this is that it contains pre-standardised data (all attributes are given a score between 1 and 99) and it also contains data on certain variables that are harder to approximate with FBRef's data - such as acceleration, sprint speed and strength. However, FBRef was eventually deemed more suitable due to its use of real-world data. It isn't clear how EA calculates each player's in-game statistics and relying on video game data would likely introduce unnecessary bias into the model. Furthermore, FBRef obtains its data from Opta, who are an industry leader in football data, having been used by football clubs and broadcasting organisations alike [18]. This ensures that the quality of the data being analysed in this project will be of a sufficient standard - something that is crucial when dealing with the fine margins of sport.

⁴ "On-pitch metrics" is being used here to describe data about a player's on-field performance. For example, this would include goals, assists, tackles, minutes played, passes made, etc. However, it would exclude information such as age and transfer value.

The data gathered covered Europe's traditional top five leagues, which was deemed an extensive enough dataset as this covers most of the elite level football in Europe. Furthermore, these five leagues are generally the most affluent leagues in the world as well as the highest in quality.⁵ Partially for these reasons, players across these five leagues are mostly well-known to clubs and fans alike. Hence, restricting the initial analysis to these leagues can help to minimise substantial variations in league quality and the impact of commercial factors on value - which could, in theory, skew the results of the project if not accounted for.

One of the early decisions of the project was whether to include Champions League statistics in the data as well. I decided against doing this for a myriad of reasons. Firstly, the standard of the Champions League is, by definition, much higher than in domestic leagues and not every team plays in the Champions League. This would potentially have two effects, neither of which are desirable for our analysis. One effect would be a potentially unfavourable influence on per-90 statistics for a Champions League player due to the increased quality. The other effect would be that the extra game time may positively influence a player's aggregate statistics - more playing time is more time to score goals, play passes and make tackles. Both effects would somewhat skew the data.

However, neither of these are the main reason that Champions League statistics have been omitted from this analysis. The main reason is that the format of the Champions League is more variable. One team may be drawn in a Champions League group with teams from comparatively weaker footballing leagues, whilst another may find themselves in what is commonly referred to as "the group of death", where all teams are deemed to be strong. Furthermore, how many games a team plays in the Champions League (and in theory the strength of opposition) will depend on their progression through knockout rounds. It is foreseeable that a player may have inferior per-90 statistics simply because his team reached the Champions League final (or even won the competition) and hence, was faced with a greater standard of opposition. Alternatively, a player whose team got knocked out early by the eventual winners of the competition may see his aggregate numbers suffer comparatively.

In contrast, in all of Europe's top five domestic leagues, each team plays every other team twice. This creates a consistent backdrop on which to analyse the data. In all, inclusion of Champions League statistics was deemed to be an unnecessary exercise that would only introduce more

⁵ It is worth noting that European football has somewhat been thrown into disarray by the recent emergence of the Saudi Pro League. However, this is unprecedented in the sport, and it would be near impossible to draw meaningful conclusions involving the Saudi Pro League at this stage - particularly as any limits to their growth, spending and talent acquisition are currently unclear.

randomness into the analysis - and with a dataset that was already sufficiently large, this did not seem like a trade-off worth making.

Regarding obtaining player transfer values, this was a slightly more complicated task. Ideally an exact transfer value would have been found for each player as this would provide the highest accuracy. However, this would have reduced the size of the dataset drastically for several reasons. Firstly, not all players have publicly available transfer fees. Secondly, transfer fees are multifactorial. Some players have lower transfer fees due to release clauses, high wages, high agents' fees, expiring contracts, personal circumstances with the club, forcing a transfer, image rights and much more. Thirdly, not all players have ever transferred clubs, some are academy prospects who have remained at their club and progressed into the first team. Finally, of those players who have transferred, most last transferred to a new club a few seasons ago at best, so their transfer value would now be outdated.

With all this considered, Transfermarkt's estimated value had many advantages and avoided many of the issues that an exact transfer value had. Transfermarkt's value is crowd-sourced which provides an interesting lens through which to analyse this issue and differentiates this project from previous projects in the literature space. Crowd-sourced values should in theory provide an accurate reflection of a player's true market value due to a popular concept introduced by James Surowiecki known as "the wisdom of the crowd" [19]. This states that large groups tend to more accurately estimate a value (by averaging all estimates) than individuals or a small group of people. This adds an interesting dimension to the analysis. The number of people involved in transfer negotiations or establishing a player's price on FIFA is likely to be relatively small compared to the openly accessible crowd-sourcing efforts of Transfermarkt. Hence, this could in theory provide a more accurate reflection of market value than an actual transfer value that has been influenced by the negotiation tactics of two competing clubs. Overall, a combination of its novelty, availability, completeness (there is data for nearly every player unlike with actual transfer fees) and the notable lack of potential for data snooping issues (such as those caused by using video game data), Transfermarkt's estimated transfer value was deemed the most suitable fit for the project.

3.3 Web-scraping

As mentioned in the introduction (see section 1.2), the aim was for the code to be dynamic and object-oriented. An essential component of this was the construction of multiple web-scraping functions that automated the web-scraping process to retrieve the required data.

Firstly, the FBRef web-scraper was built. The scraping was performed using the Selenium driver as issues arose when using more basic packages such as BeautifulSoup - these packages struggled to pick up the complex nature of the webpages.⁶ There were two main functions written for the web-scraping of FBRef: the *find_data* function and the *fbref_top5_web-scraper* function. The data needed for the project was spread out across various web pages and hence, *fbref_top5_web-scraper* focuses mainly on constructing the relevant URLs, retrieving the HTML code using Selenium and identifying the relevant table within the HTML code. This function then calls the *find_data* function which undertakes the actual data extraction and formatting. The output of *find_data* is a dictionary where each key references the webpage that the data came from, and the corresponding value is a list of dictionaries of player data. As part of the data formatting process a function *try_convert_to_float* was also written to avoid individual errors when running the code over a large dataset. This function primarily helped to deal with instances of missing values (as these would return an error if the usual python *float()* function was used).

The next step was to construct the Transfermarkt web-scraper. This consisted of only one main function (though there were other small helper functions), *transfermarkt_scraper*, which takes a player object as an input and builds the relevant URL to take advantage of Transfermarkt's player search function. Unfortunately, Transfermarkt does not maintain a table of all player estimated transfer values in the same way that FBRef does, so the individual querying was the next most effective way to scrape the site for estimated values. The function starts by calling another function, *remove_accents*, which converts each letter of the player's name to its closest ASCII equivalent. This is because FBRef is more specific with its use of accents and non-standard characters than Transfermarkt, so some of the player names would not be recognised by Transfermarkt without this step.

Selenium is then used to interact with the page in a more dynamic way, obtaining details about the specific player (from a list of players with similar names) that allows for the construction of a second URL, leading directly to the player's estimated transfer value page. The function then inspects the JSON objects that are present on this new page and returns a list of dictionaries with various information about estimated transfer values and the dates that those values correspond to.

Crucially, the data from both the FBRef web-scraper and the Transfermarkt web-scraper is then stored by using the *pickle* package. This allows for more convenient access to the data as the web-scraping efforts are fairly time-

⁶ It is worth noting that BeautifulSoup was used later in '*fbref_top5_webscraper*' to structure the data obtained by Selenium.

consuming due to the high volume of HTML requests and the requirement to use a more powerful, though consequently less efficient, web-scraping package such as Selenium.

3.4 Formatting

The next stage of the data cleaning process was to improve the format in which the data was stored because the current “dictionary-of-lists-of-dictionaries” format was not as elegant and accessible as it could have been. For this a custom *Player* class was defined. This was perhaps the most important section of code as it provided the foundation for the rest of the project. A large range of attributes were defined within the class to store various aspects of a player’s individual data. Additionally, getter and setter functions were defined (in line with software engineering best practices) to set and retrieve different attributes. A variety of dunder methods were initialised alongside these to aid in player comparison and the presentation of information. For example, `__str__`, was used to ensure that when `print()` is called on a *Player* object, it displays some identifying information about the player in question.

To initialise the instances of these *Player* class variables, three functions were defined. The first, `combine()`, focuses on combining the stats from the five different FBRef pages so that they are not stored separately for each page. The second, `combine_players`, deals with the formation of each instance of the *Player* class. As part of this, it ensures that those stats that share a name across pages but represent different metrics are renamed so that they don’t overwrite each other. The final function, `combine_duplicates`, handles special cases where a player has played for more than one club within a season, or occasionally that two players share the same name but are not the same.

The final formatting requirement was to include each player’s estimated transfer value (and their historic estimated transfer values). For this another function, `update_values`, was written that took advantage of the setter methods within the *Player* class itself, though represented an easier way to update them all at the same time.

The collection of functions outlined above ensure that the data is in a convenient and accessible format for the rest of the analysis. In other sections of the analysis, further refinements are made such as various iterations of feature engineering and converting the statistics to per-90 statistics. However, these are done within the specific analysis itself. In the initial set up, it was important to have multi-purpose, flexible *Player* variables that could serve the needs of each individual analysis.

4 Experimental Results - Value Prediction

This section will cover the set up for producing a value prediction model, the various machine learning methodologies employed, their accuracies and an evaluation of the validity of the predictions.

4.1 Overview

For TAGscout's value prediction feature, a regression model is required. Furthermore, to fulfil one of the project's key aims - justifying the use of machine learning in football - the model needs to provide a certain degree of accuracy. As previously seen, much of the literature on value prediction utilises neural networks, so this project will also explore the use of the popular deep learning technique. However, an SVM regressor and a basic Lasso regression will also be implemented to act as a point of comparison. We will see that a combination of dropout and L2 regularisation in a neural network will provide the best evaluation statistics, but further analysis will reveal that these may be misleading.

4.2 Data restructuring

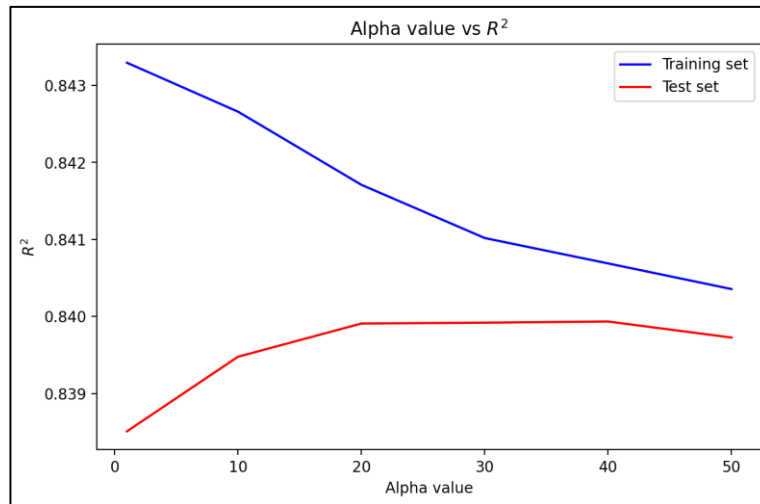
The first step, before implementing any models, is to restructure the data to ensure it is in the appropriate format. This process takes many stages but will largely be implemented through the custom defined *restructure_data* function. This function will: load the relevant data; call the *keep_attributes* helper function (to assist with feature selection and ensure the appropriate statistics are in their per-90 form); initialise some other useful variables such as dummy variables for which league a player currently plays in; and construct an array of features and an array of labels such that these can be passed into a model.⁷ To expand the size of the training data and best optimise the model's performance, on-field and market value data will be taken from two different seasons. More specifically, data from the 2020-21 season will be used to predict market value in early-to-mid 2022 and data from the 2021-22 season will be used to predict market value in early-to-mid 2023. These two analyses will be combined into one dataset that will be split into a training and test set - and a validation set, where appropriate. The data will then also be normalised so as not to skew the results of the model - crucially, this is implemented in a way that avoids data snooping.⁸ Data from seasons prior to the 2020-21 season could have been used to expand the dataset further, however, football is notorious for experiencing quite rapid and severe inflation in the transfer market. Hence, it does not appear useful to expand the dataset further in this manner.

⁷ Game time will also be passed as an input to the models. Hence, it is best if this is controlled for in the other features to avoid highly correlated variables.

⁸ The Transfermarkt values were also unanimously divided by 1000, to ensure that the results were more readable on the axes of graphs.

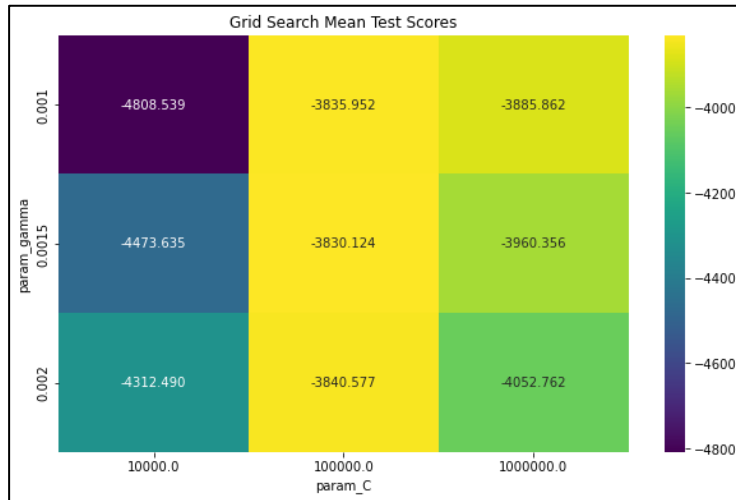
4.3 Model selection and results

The first model that we will establish will be a basic Lasso regression. Using a range of alpha values (to control the L1 regularisation strength) and plotting these against the R^2 we can see that an alpha value of 40 seems to maximise the test R^2 and reduce overfitting.



This model gives us a training set R^2 of 0.8407 and a test set R^2 of 0.8399. The evaluation statistics for this are fairly positive and indicate that future, more complex models should also perform well.

Next, we can implement a slightly more sophisticated model in the form of an SVM regressor. To find the optimal parameter values, grid search can be used. This can be trained on a subset of the original training set and tested on a different subset of the original training set, we will refer to these as the training set proper and the validation set, respectively. The results of the grid search can be visualised to ensure that there are not obvious directions in which the parameters could move to become more suitable. This visualisation is shown below and combined with printed outputs of the function, suggests that the optimal parameter values are 0.0015 for gamma, 100000 for C and 100 for epsilon. Various kernels were also tried, and RBF was deemed most favourable.



The optimal parameters applied to the original training set and test set data result in a training R^2 of 0.846 and a test R^2 of 0.837 - similar to our Lasso regression.

Having assessed two more straight forward models, we can then move onto deep learning methods. We will test various neural network architectures across four broad categories: no regularisation; regularised via the inclusion of dropout layers; regularisation via L2 regularisation; regularisation through the combined use of dropout layers and L2 regularisation. The neural networks will all utilise the ADAM optimiser as this was found to be the most optimal. Furthermore, learning rate schedules will be used to speed up model convergence and improve performance.

The best architecture for each category of regularisation provided the evaluation statistics shown below.⁹

| <i>Regularisation type</i> | <i>Training set MAE</i> | <i>Test set MAE</i> | <i>Training set R^2</i> | <i>Test set R^2</i> |
|----------------------------|-------------------------|---------------------|--------------------------------------|----------------------------------|
| <i>No regularisation</i> | 2884.86 | 3768.49 | 0.9168 | 0.8617 |
| <i>L2 regularisation</i> | 3289.05 | 3590.91 | 0.8572 | 0.8531 |
| <i>Dropout</i> | 2820.92 | 3557.63 | 0.8970 | 0.8636 |
| <i>Dropout & L2</i> | 3048.14 | 3487.40 | 0.8793 | 0.8647 |

⁹ Note that architectures involving dropout will vary in their evaluation statistics each time the model is re-trained due to the inherent randomness involved in dropout layers. The evaluation statistics shown are those provided on the last training that was run but similar values can be replicated.

From this we can see that the most effective regularisation was a combination of dropout and L2. This provided both the lowest test MAE and the higher test R^2 , though it is worth noting that all models performed within a similar range. The best architecture used for the dropout and L2 model involved three fully connected hidden layers and an output layer; all hidden layers used a ReLU activation function (though others were tried). Two dropout layers were used, the first with a probability of 70% and the other with a probability of 40%. The implementation of the architecture can be seen below.

```
class NN_combined(nn.Module):
    def __init__(self):
        super(NN_combined, self).__init__()
        self.fc1 = nn.Linear(X_train_scaled.shape[1], 1000) # fully connected layer with 1000 neurons
        self.dropout1 = nn.Dropout(0.7) # dropout layer with 70% probability
        self.fc2 = nn.Linear(1000, 250) # fully connected layer with 250 neurons
        self.dropout2 = nn.Dropout(0.4) # dropout layer with 40% probability
        self.fc3 = nn.Linear(250, 100) # fully connected layer with 100 neurons
        self.fc4 = nn.Linear(100, 1) # output layer

    def forward(self, x):
        x = torch.relu(self.fc1(x)) # ReLU activation function
        x = self.dropout1(x) # apply dropout after ReLU activation
        x = torch.relu(self.fc2(x)) # ReLU activation function
        x = self.dropout2(x) # apply dropout
        x = torch.relu(self.fc3(x)) # ReLU activation function
        return self.fc4(x)

# run the neural network with L2 regularisation of 0.1
run_NN(NN_combined(), X_train_scaled, X_test_scaled, y_train, y_test, 0.1)

Train Mean Absolute Error: 3048.144775390625
Test Mean Absolute Error: 3487.39892578125
Training Median Absolute Error: 1196.62841796875
Test Median Absolute Error: 1659.962646484375
Training R^2 Score: 0.8793136153299966
Test R^2 Score: 0.8647409115377125
```

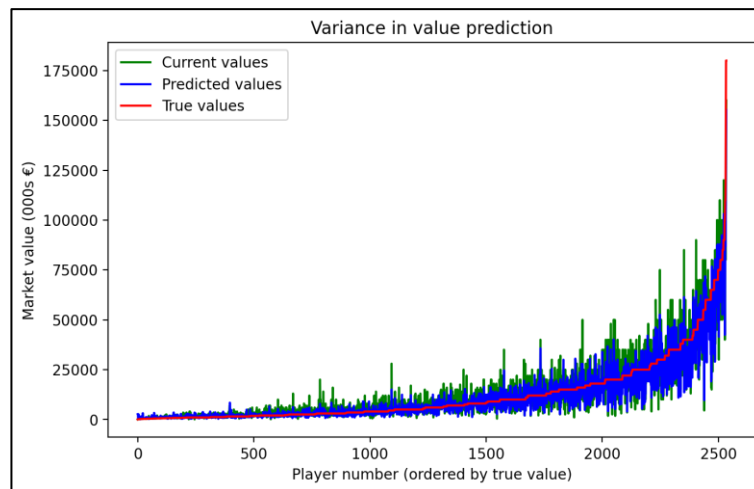
We can also see from the results that the median absolute error is significantly lower for both the training and test sets. This indicates that perhaps there are some outliers where the discrepancy between the model's prediction and the true value is large. Hence, save for a few outliers, we would expect the model to be predicting with an average variation from the true value of about €1.7m - this seems to be a fairly reasonable error given the magnitude of transfer values in football.

It is important to note, that for all the models seen above, the accuracy benefited greatly from the inclusion of a player's "current value" as a feature. This was important as it allowed for the model to somewhat factor in conditions outside of the acquired data such as a player's popularity or media presence. Furthermore, the inclusion of this feature does not constitute data snooping as it fits with the data that would be available in our use-case. If the goal is to predict future value, one would still have access to a player's current Transfermarkt value.

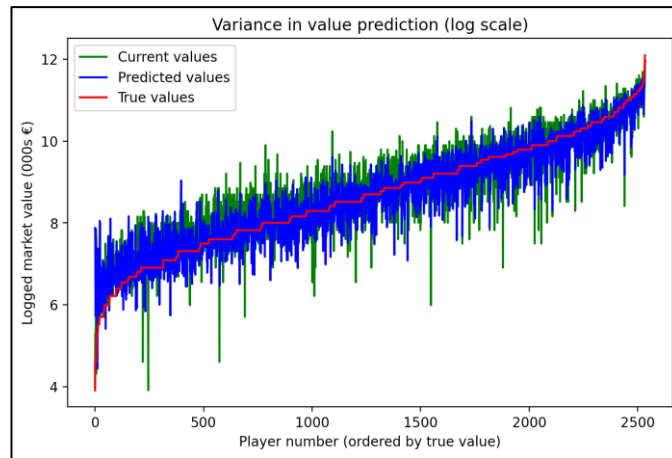
4.4 Validity and usefulness

Now that we have obtained a model with promising evaluation statistics, we can begin to explore how valid and useful this model is. By manually

testing the model on well-known players we can judge if the model is outputting reasonable suggestions. Taking Marcus Rashford as a test player, we find that the value predicted is around half of his true value. This is quite a large discrepancy for a model that has such promising evaluation statistics. Further test cases on Bukayo Saka, William Saliba and Ben White reveal a similar trend. To further investigate what is occurring here we can order players in the test set by their true values and then plot their current values (the value that is passed to the model as a feature), predicted values and true values. The results reveal the core of the problem.

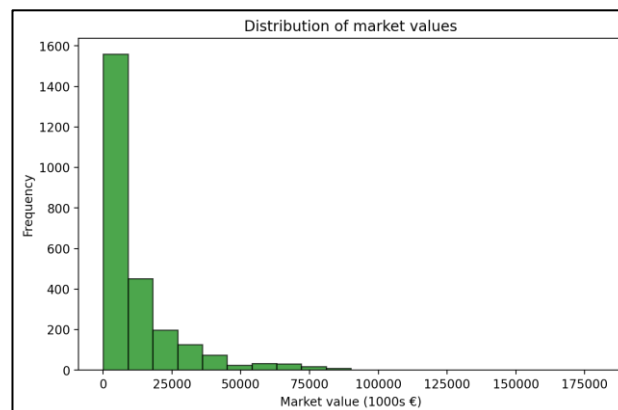


We can see that there is a huge amount of variance in the data. It is worth noting that the model does seem to perform better than simply using the player's current value - which can be seen by the reduced variance seen in the predicted values when compared with the current values. However, this variance is still fairly large, spanning tens of millions of pounds in some instances. The data seems to follow an exponential distribution as well. Hence, to better see whether the relative variance is changing over time we can plot the log-transformed market values.



From this we can see that there is significant relative variance in the predictions regardless of the magnitude of the value that they are predicting.

These findings raise the question of how a model with such high variance could obtain such positive evaluation statistics. For this we need to consider what the evaluation statistics are actually representing. The loss function during the training phase of the neural network is the mean absolute error. This means that the model is seeking to minimise the average deviation from the true value - which sounds like what we are looking for. However, due to the vast scale over which the market values are distributed, this could cause an issue. The player with the lowest value in the training data has a market value of €50,000, whilst the most valuable player has a market value of €180,000,000. This is a difference of €179,950,000 - quite a substantial variation. Furthermore, there are 1560 players in the training data that are worth less than €10,000,000 and only 864 that are worth more than €10,000,000. We can visualise the distribution of values using a histogram.



This distribution may explain a portion of the issue we're experiencing. Given that most of the players in the training set cluster at the lower end of our market value spectrum, the model may provide what seems like a low mean average error across the whole dataset but really is still fairly large if most of the players cost under €10,000,000. Furthermore, the high R^2 values may be the byproduct of the distribution as well. Consider the formula for calculating R^2 values:

$$R^2 = 1 - \frac{RSS}{TSS} = 1 - \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{\sum_{i=1}^N (y_i - \bar{y})^2}$$

As our TSS grows, the R^2 value will increase and as long as the RSS grows at a slower rate, the R^2 will remain positive - the model simply needs to be better than just predicting the mean.¹⁰ Models trained on labels with a large spread around the mean can therefore obtain a relatively high R^2 value, even if the model is not all that accurate in absolute terms. Given that the training data has an average label of €12,442,209.07, there will be a huge spread of data around the mean, with the highest market values differing by over €100,000,000. Furthermore, for the smallest sample in the training set, that has a label of €50,000, the model could predict a value of €1,000,000, giving it an residual of €950,000, compared with a difference from the mean of €12,392,209.07. The model would view this very favourably in terms of producing an R^2 value, the sample would increase the TSS significantly more than it would increase the RSS. However, the prediction for the player would be 20 times larger than the actual value and in the practical deployment of a value prediction model this would be an unacceptable difference. In fact, when one analyses the data further we can see that the prediction for the player in question (with the lowest label) was actually €2,097,234.90, which adds credence to the theory that the R^2 value is being skewed by such cases.

These findings pose quite a large issue to the model's validity and usefulness, so it is worth exploring ways to resolve this.

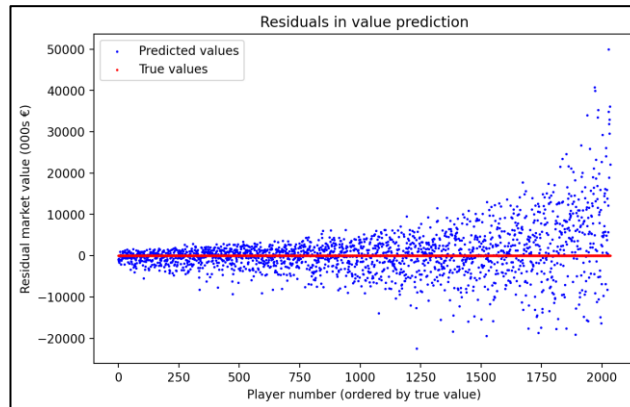
4.5 Efforts to improve the model

In an effort to improve the model there are a number of avenues to pursue. We will explore a few here.

¹⁰ TSS essentially measures the distance of each sample from the mean, and squares this. RSS measures the distance of each sample from our predicted value for that sample, and squares this. Both are calculated as the sum of these squares over all samples.

4.5.1 Checking the residuals

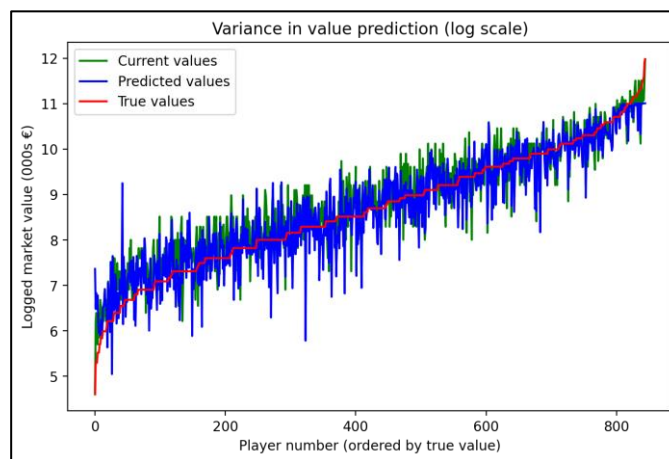
We can plot the residuals to check for systematic bias. From this we see the residuals seem to follow a pattern, with the residuals growing as the true player value grows.



However, we'd expect this pattern given the vastly different magnitudes of the labels. A further investigation into the residual relative to the magnitude of the label, reveals no obvious pattern.

4.5.2 Applying a log transformation to the labels

Given the exponential distribution of the data, applying a log transformation to the labels during training may theoretically help to improve model performance. However, in practice this results in no significant benefit when applied to the test set, as seen below.

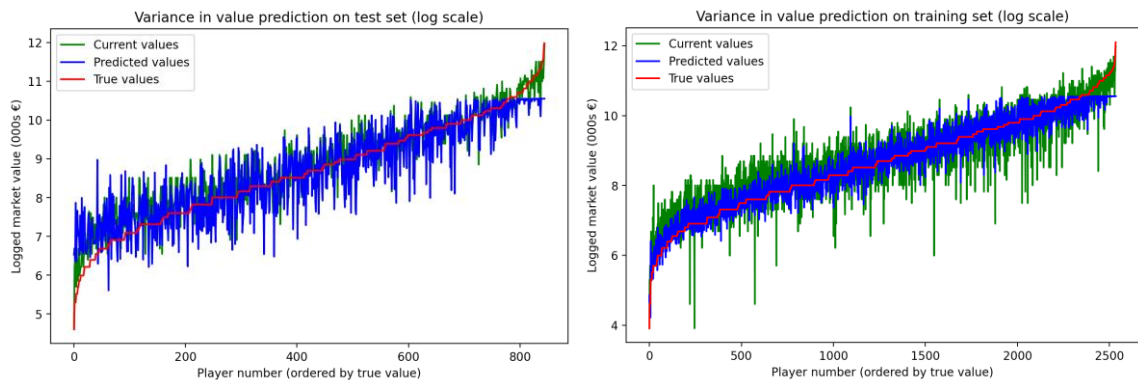


4.5.3 Outlier removal and subsets of the data

Given the issues caused by the large range over which the labels are distributed, outlier removal and taking subsets of the data are other possibilities. However, the model struggles on these subsets of the data as well. The only notable change is that on certain subsets, due to the decreased variance in the data, the R^2 values drop and there is not an obvious way to improve them. Whilst this does not improve our model, it does support our earlier hypothesis that the large spread of data from the mean was contributing to misleading R^2 values.

4.5.4 PCA for noise reduction

Another possibility is the use of PCA for noise reduction (applied to the features before they are fed into the model for training). For this the previously excluded attributes were added back in, and it was found that the first 50 principal components explain over 98% of the variation in the data. Training a model using these principal components we can see that the training set results show some signs of a slight reduction in variance. However, the test set variance is excessive and doesn't suggest that this is a suitable solution to the issues that we are faced with.



Different numbers of principal components were tried, such as the first 30, which explain over 93% of the variation in the data. However, they were similarly ineffective at delivering substantial improvements to the model.

4.6 Conclusions from value prediction

The value prediction model looks promising on the surface, the evaluation statistics appear very reasonable for the dataset that we are working with. The R^2 values are only slightly below those obtained by the studies referenced in section 2 - which is respectable given that we can expect the data in this project to be much less structured and noisier than the video game data used in the aforementioned studies. However, a more extensive investigation into the model's predictions highlights the ways in which

evaluation statistics can be misleading. Whilst, this was not the intended function of the value prediction model, the results do call attention to the need to avoid an overdependence on evaluation statistics in regression analyses such as this one.

Unfortunately, model performance does not improve much in any of the iterations implemented in this analysis. The reasons for this are not fully clear. There is always the question of whether alternative neural network architectures, with alternative parameter choices, could boost training and test set accuracy in tandem. This project failed to produce this finding, but neural network architecture selection is more of an art than a science and hence, it is possible that some variation of the network's architecture exists that would see better performance.

However, based on the findings of this analysis, it is possible that the issue is with Transfermarkt's market value estimates and more generally, crowd sourced data in football. It is unlikely that, on average, true market values fluctuate massively for one player within one or two seasons. Hence, it can be interpreted as a criticism of crowd-sourcing market values that we see such a large discrepancy between a player's "current value" and their true "future value". There are many reasons that this could be the case, perceived market value is not an isolated metric that purely reflects on field performance. For example, a player's popularity may affect their perceived market value. Equally, their media presence may have a similar effect, with players who invest a lot into their brand value obtaining higher Transfermarkt values. Even outside of a player's personal media use, the amount of media coverage that player receives, for one reason or another may affect their perceived value. It is possible that a player suddenly falls into or out of the spotlight and hence, the number of people estimating their value on Transfermarkt drastically changes. The effect this would have on their value is unknown. Regardless, all of these factors are more volatile than how talented a player is at football and could explain why Transfermarkt values are somewhat unstable. To some extent these factors may influence real-world market value as well, but perhaps the market valuations given by professionals working in football are less sensitive to these off-the-pitch influences and hence, would be more predictable. Perhaps transfer values themselves are the problem and are too volatile. Something less elastic such as a player's contract value could be a better proxy for footballing prowess - though this would come with its own drawbacks.

Another possible cause of the volatility seen in market values could be the impact of COVID-19. The finances of European football clubs were significantly impacted by the pandemic and the halting of football, followed by a return but with empty stadiums. Deloitte's *Annual Review of Football Finance 2021* found that "the European football market contracted

by 13% in 2019/20, as overall revenues fell by €3.7 billion to €25.2 billion. This is the first reduction in revenues since the impact of the global financial crisis was felt in 2008/09" [20]. This is quite a substantial shock and some of the data included in the analysis (the 2020-21 season's data) would have been directly impacted by this, whilst even the 2022-23 season could be experiencing the unknown long-term effects. The pandemic equally could have had an impact on player performance due to varying player mental health and playing without a crowd - which did appear to cause a reduction in the performance of the home team [21]. There are also other factors that have an unknown effect on crowd-sourced values such as the increase in social media usage during the pandemic [22]. Perhaps there were more people contributing to Transfermarkt's crowd-sourced values; perhaps the increased time at home meant that fans could watch more football and develop more informed assessments of players; perhaps the increased use of social media amplified the importance of brand value; perhaps fans watched highlights of players on social media more than actual games and hence have a skewed perception of a player. There is a large array of possibilities, and unfortunately, the impact of COVID-19 on European football is not yet fully understood.

With regard to TAGscout, the value prediction will continue to be implemented as a proof of concept in the hope that future models- perhaps trained on more reliable data, trained on more data or trained with alternative methodologies - will be able to make more useful predictions for TAGscout. The model that will be used for value prediction in this proof of concept will be the dropout and L2 regularised neural network shown in section 4.2 as this remains the most accurate model in this analysis, despite its limitations.

Overall, due to the underwhelmingly usefulness of the value prediction model, it is hard to argue that it justifies the use of data analytics in European football. Models based on more concrete outcomes such as goals, assists and results may better serve as an advert for the power of machine learning applied to sport - particularly as there is less uncertainty around the validity of the labels used in the training data.

5 Experimental Results – Similar Player Search

This section will explore the implementation of the similar player search function in TAGscout. It will cover methods for identifying similar players, dimensionality reduction and an analysis of some test cases.

5.1 Methods

The “similar player search” function of TAGscout has two main requirements. Firstly, the algorithm needs to find players with similar attributes to a specific inputted player. Secondly, we need to visualise this in the simplest way possible to give the user some context.

For the identification of similar players, a version of a simple K-Nearest-Neighbours (KNN) algorithm appears to be the most appropriate - easily allowing the user to input how many similar players they wish to identify. This method has a few notable strengths. Firstly, KNN is a non-parametric algorithm, meaning that it does not make any explicit assumptions about the functional form of the data. This is important for exploratory analysis such as the kind that we are trying to implement here as our main goal is to understand the basic structure of the data and uncover some basic features of the data (such as players who possess similar traits). Furthermore, KNN is a relatively straight-forward algorithm which doesn't take a great deal of data-science literacy to understand; this is desirable for the end-users of TAGscout with a limited background in data science. Given that there is no means by which to check accuracy for this function, it makes sense to keep the algorithms involved as simple as possible to ensure interpretability.

Due to the simplistic nature of the algorithm used to implement this feature of TAGscout, the algorithm will unfortunately be somewhat sensitive to irrelevant features. To best mitigate this and avoid some of the issues that can be caused by data with excessively high dimensionality, it is important to manually select features that appear to have some relevance. For this, several categories of attributes can be assembled, each only containing attributes that appear relevant at face-value. The categories that have been assembled are:

- Passing
- Progressive passing
- Short passing
- Long passing
- Defending
- Ball carrying

- Goal scoring
- Chance creation

These categories consider a variety of traits and try to avoid highly correlated variables where possible. For example, the *Passing* category includes short, medium and long pass completion percentage, as well as the number of each of these pass-types attempted. However, the category does not include the total number of completed passes for each of these pass-types. This is because passes attempted and pass completion percentage essentially cover this anyway and keeping the number of passes completed would simply add in a variable that is already highly correlated with other variables in the analysis - which could potentially skew the analysis. On top of this, TAGscout automatically restricts the analysis to only players in the same position (or one of the same positions if the player has multiple). This ensures that players operating in similar roles within a team are compared.

Further optionality was added to the similar player search function of TAGscout by adding the option to use per-90 stats in the comparison, so that how many minutes a player has played does not benefit them in the analysis. This was simply done by dividing the relevant raw statistics by the number of 90s that player played. Additionally, the user can set minimum and maximum age requirements along with the ability to specify a minimum percentage of their club's available minutes that a player must have played last season. The age requirements allow potential scouts to ensure that the players they are comparing are relevant to the age-profile requirements of their squad. It also allows the user to, for example, find a younger player with similar attributes to a more established player whose style of play is more known. The "minimum percentage of club minutes played" filter is useful partially because it allows users to only analyse players who are integral to their teams if required. However, the main benefit of this filter is that it removes players who may have skewed per-90 statistics due to playing very limited minutes. Consider a youth player at a top club who plays only the last ten minutes of the last game of the season and scores. Whilst this is somewhat impressive for the youth player, a club may not want to sign him on the back of his "goals-per-90" statistic as it is almost guaranteed that he will not keep up a rate of scoring nine goals a game! The more minutes a player has played, the more the effect of outliers, chance and luck can be smoothed out and hence this issue somewhat disappears.

Once we have obtained the similar players from our algorithm, the second task is to visualise this in a way that can provide more context. To visualise this in the simplest way possible, we'll need to reduce the dimensionality of the data down to 2 dimensions - an x and a y axis. Techniques to include

more dimensions, such as the use of colour are not appropriate here as colour will be used to highlight the inputted player and the similar players that have been identified. Similarly, whilst the use of shape could be included for categorical variables, this would make the visualisation more complicated and cluttered without really providing a huge benefit and hence, make it harder for the user to derive useful insights.

With all this in mind, the most applicable dimensionality reduction technique appears to be Principal Component Analysis (PCA). PCA has several advantages. Firstly, PCA easily allows for the reduction of the data down to two dimensions. Furthermore, these two dimensions will represent the highest possible variance in the data, which ensures that we retain the most information possible in our visualisation. Another strength of PCA is that the principal components it returns are orthogonal, which means that they are uncorrelated when we plot them. This ensures a few things. It ensures that we avoid including redundant information and it also makes the visualisation more straightforward to interpret in that there is no correlation between the two axes. Additionally, PCA can help filter out noise and hence reduce the impact of irrelevant variables. Compared with other dimensionality reduction techniques, PCA has a few more beneficial traits. Firstly, it is an unsupervised method unlike LDA. Given that there are no labels in our given problem, this is essential. Secondly, unlike t-SNE, PCA is deterministic - meaning that the algorithm will always produce the same results. To ensure users can form a consistent analysis of the data, it is more useful for the similar player search function to give the same results every time.

Most importantly, the two methods chosen here are somewhat complimentary. Our KNN algorithm looks at the whole high-dimensional subspace, whilst PCA creates a new subspace upon which to project the data. Whilst their primary objectives differ, both methods revolve around the concept of distance in a feature space. KNN uses distances to identify the nearest players, whilst PCA looks for directions that maximise variance. Hence, both methods share a foundational relationship with the geometric structure of the data in the feature space and it makes sense to combine the two to deliver the similar player search function of TAGscout.

5.2 Implementation

Now that the methods have been settled upon, the next step before applying either KNN or PCA is to normalise the data, so the scale of an attribute doesn't massively skew analysis. For example, the numbers involved in total passes attempted are likely to be significantly larger than the numbers involved in goals scored - attempting a pass is a great deal easier than scoring a goal in football. If these raw numbers were fed into

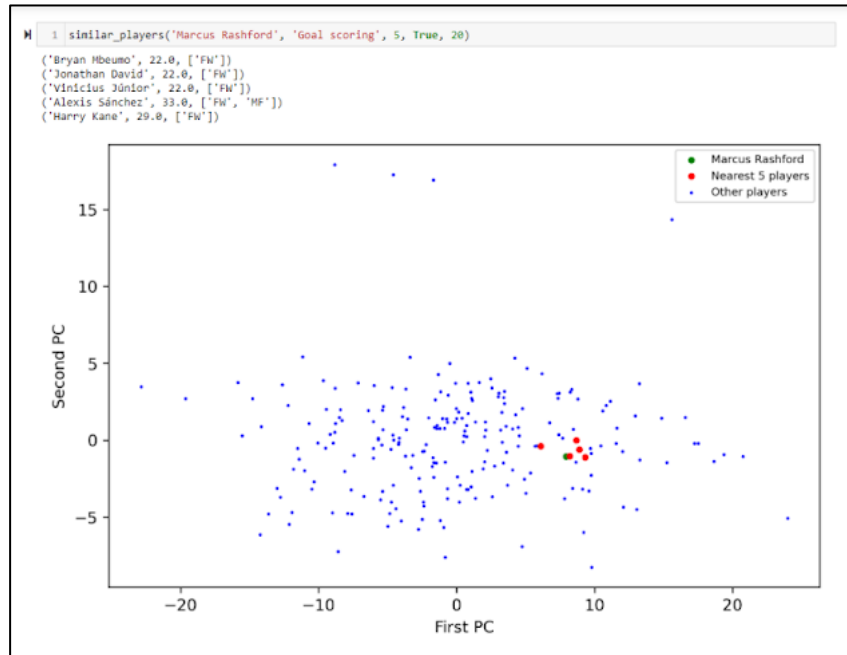
our algorithms, they would massively skew the analysis. Normalisation allows us to ensure that the scale is the same for both, whilst preserving the general structure of the data within each feature. For this, we can use Scikit-Learn's *StandardScaler()* function along with a custom built *normalise* function that will allow this normalisation to be applied to our data in its current, unique structure.

Following the normalisation process, we can refine the set of players to only those who share a position with our inputted player and apply the filters previously mentioned, including the attribute category to compare. This new subset of data can then be passed into two custom functions. Firstly, the subset of the data will be passed to *nearest_players* along with the inputted player and the desired number of similar players to return. This function will then implement our KNN algorithm and output a list of similar players (in order of most similar to least similar) whose distances to the inputted player in the subspace are the shortest. This output can then be passed to a custom defined *pca* function along with the subset of the data and the inputted player. This function will then calculate the first two principal components and visualise the results, highlighting the nearest players in a second colour and the inputted player in a third colour.

5.3 Test cases & justifying the use of visualisation

Now that the *similar_players* function is operational we can explore its results. Two different test cases have been provided that demonstrate the use of visualisation in this problem. Let's look at two players operating in similar positions at the same club: Manchester United's Marcus Rashford and Antony. We will assess a relevant attribute category for both: their goal scoring.

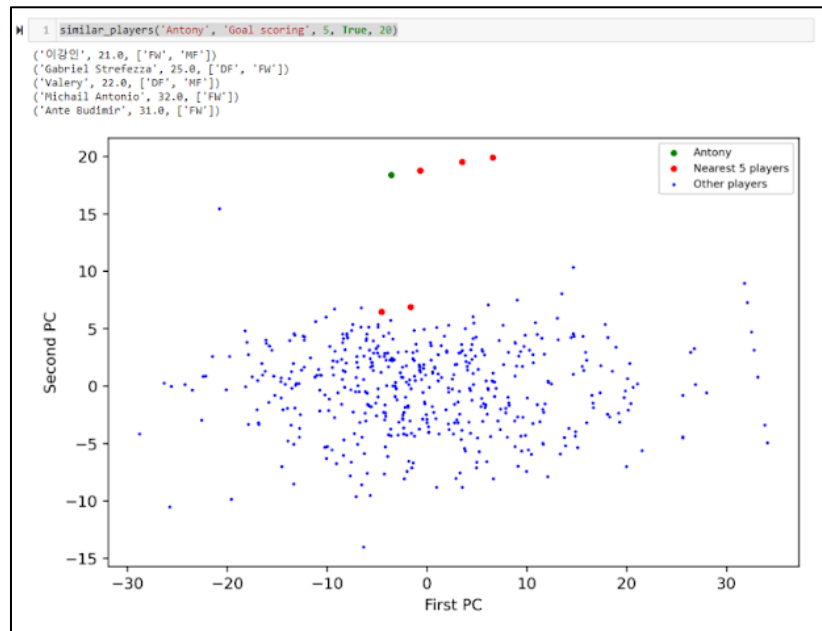
Firstly, we can provide Marcus Rashford as the input player to the function and 'Goal scoring' as the attribute category to assess. We will ask the function to provide five similar players, based on per-90 statistics, who played 20% or more of their club's available minutes in the 2022-23 season. For this test case we won't specify an age range. The outputs are shown below.



We can see that the function operates as expected. We see the summary information for five similar players to Marcus Rashford operating in the same position. The players are also highlighted appropriately on the visualisation as expected. Some background knowledge in football also indicates that the similar players suggested are fairly reasonable. Marcus Rashford is considered an elite-level, goal-scoring winger. The players provided include Vinicius Junior (another prominent elite-level, goal-scoring winger), Bryan Mbeumo (a goal-scoring winger who had something of a break-out season last year) and Harry Kane (whilst not a winger, he is an elite level goal-scorer). Whilst there is not a more robust method to test the accuracy of this function, the purpose of the function is more about exploring, understanding and interacting with the data - something that does not require accuracy testing to be quite as stringent. In all, our results seem reasonable, and we can see that all the players suggested are close to Marcus Rashford in the visualisation. However, some context that we do gain from this is that these players may not be unique in their proximity to Marcus Rashford, there are other candidates in the region that also appear to be quite similar.¹¹ The visualisation has aided in our analysis somewhat.

¹¹ It is worth noting that some of the players in the visualisation are actually closer to Marcus Rashford. This is due to the inevitable loss of some information during the PCA process. However, the visualisations still serve their purpose of providing general context to the data. Additionally, this phenomenon is expected and is **not** a sign of implementation issues within the KNN algorithm.

Next, we can move onto Antony. Passing in the same parameters as for Marcus Rashford (but with Antony as the input player) we get the following output.¹²



This provides us with a very interesting result and a strong justification for the use of visualisations in this problem. Antony appears to be a bit of an outlier from the rest of his cohort. We can see that of the five similar players returned, two of them are not that similar at all. Instead, Antony exists in a small cluster with three other players. For this reason, it doesn't make a huge amount of sense to consider Michail Antonio and Ante Budimir as similar players in the same regard as the other three. Without this visualisation the user may assume that all five players are quite similar to Antony. Running the algorithm twice more, once asking for three similar players and once asking for ten similar players highlights this result even further (see appendix B). Once again, the visualisations have aided our analysis.

5.4 Conclusions from similar player search

The similar player search function serves as a valuable tool for TAGscout. The feature pushes TAGscout closer to resembling a full-scale scouting platform as opposed to simply a prediction model with an interface. Furthermore, the customizability that is baked into the similar player search function ensures that it is flexible enough to serve a range of

¹² Note that the distribution of points for Antony is slightly different to Marcus Rashford's. This is because Antony has two registered positions: 'MF' and 'FW'. Marcus Rashford only has 'FW'. This means that more players are considered in Antony's analysis and the PCA algorithm will produce slightly different principal components to maximise the variance.

purposes. Moreover, the visualisations in TAGscout aid in ensuring that the platform can serve a wide range of users, including those who do not have an extensive background in data analysis. Additionally, the similar player search function is the most adaptive function of TAGscout, new data can easily be introduced without the need to retrain any models.

However, one of the drawbacks of this feature is that if TAGscout were to be expanded to a greater number of leagues, the process would become increasingly more computationally expensive due to the use of a KNN algorithm (distances must be calculated for all samples). Furthermore, whilst the visualisations help less data-literate users to interact with the data, the concept of PCA is fairly advanced and may pose some issues with interpretability. Therefore, this limits the visual elements of the feature to simply providing additional context and not providing accurate insights in and of themselves.

Overall, due to the inability to rigorously test the accuracy of the feature, it is difficult to say whether or not this feature justifies the use of data in European football. However, given that TAGscout uses freely available data, it seems reasonable to suggest that TAGscout's similar player search function would be of no harm to clubs, scouts and pundits - even if it only serves a starting pointing for further, more in-depth, manual scouting.

6 TAGscout – Front-End

6.1 Overview

This section will outline the design principles followed in designing the TAGscout front-end. It will cover both high and low fidelity prototypes as well as the integration of Flask with React. It is worth noting that TAGscout's web-application retains the option to predict by position - though this is not linked to a backend at present.¹³ The feature was originally present within the TAGscout backend but the implementation as it stood in this project was deemed to be somewhat trivial as FBRef's data only categorises players into goalkeepers, defenders, midfielders and forwards.¹⁴ This limited the practical use cases of a positional prediction algorithm which was originally envisioned as a tool for oppositional analysis and scouting. This remains an interesting avenue to explore as a potential future ad-on to TAGscout, and hence, the web-application set up for this has not been removed and it appears in the prototypes.

6.2 Low-fidelity wireframes

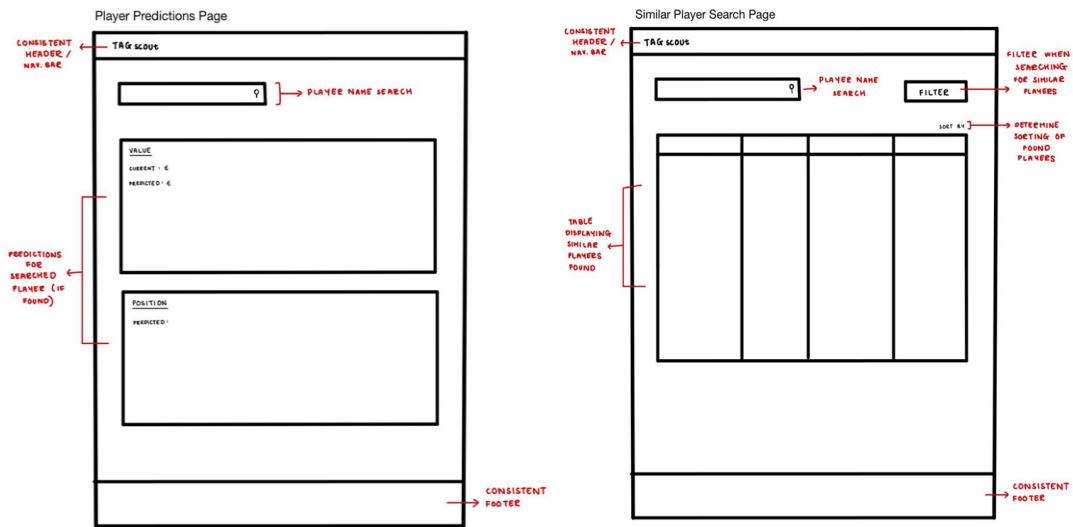
Creating a simplistic prototype was a quick and cost-effective means to explore and test different design ideas, whilst also addressing potential issues. Moreover, producing the below wireframes encouraged the focus to be on fundamental usability and layout, rather than aesthetics.

Across all wireframes presented below, the same navigation bar and footer is used. The navigation bar allows for quick identification and re-direction to the different key features of the application from any location, whilst the footer presents copyright information and could later also display identifiable links to all commonly used parts of the website.

As speakers of left-to-right languages typically focus on the left side of the page, the more important components of TAGscout were placed on this side, such as the logo, search functionality and calculated results. This makes the service more efficient and natural for the user, with further certainty provided by intuitive labelling and use of standardised images and icons.

¹³ The code for positional prediction has also still been included in the .ipynb file submitted as part of the code for this project.

¹⁴ The positional prediction model obtained an accuracy of above 85% for the test set using a strict accuracy measure that deemed the output completely incorrect if the model did not predict all positions that a player was capable of playing in.



The Player Predictions page allows users to enter a player name into the easily identifiable search bar, which upon successful submission, will display the entered player's predicted value. Similarly, the Similar Player Search page also facilitates a player name being input, but instead returns a table of similar players found, which can be filtered and sorted, as desired.

Creating the above wireframes provided an opportunity to showcase the proposed designs to a variety of potential end-users, and in turn, determine their UX-related strengths and weaknesses, along with potential solutions for the latter:

Player Predictions page:

- Found that end-users may want to know either predicted value or position, but rarely both. Solution: Make the page less cluttered and show each prediction category in their own tab or section.
- End-users appreciated the simplicity and readability of viewing current value and predicted values/positions, however, would like the option to view historical value and/or statistics, to enable easier comparison or to identify trends.

Similar Player Search page:

- Identified that end-users like having the option to easily filter and sort the similar players that are returned, however, would also like to search based on specific traits or categories, and also toggle on/off 'per 90 comparison' to provide more usable results.

- Some end-users also found it hard to read or easily compare all of the players returned in the table or commented that it ‘was not very aesthetically pleasing’.

Solution 1: Showcase the similar player results in a grid view with profile photographs (making it easier to quickly identify players by face, rather than reading through names).

Solution 2: Enable a ‘favouriting’ functionality, so users can ‘favourite’ specific players and view them in a dedicated page which allows for easy comparison. This would also allow users to create potential teams and/or ‘formations’ (rather than sift through hundreds of players and easily lose track).

Other:

- Consideration needs to be given to the handling of user searches for players that cannot be found.
- Also need to consider how to handle the user searching for a player, when multiple players have the same name.

6.3 High-fidelity prototype

Designing a high-fidelity prototype allowed for a more accurate representation of the final product, with detailed visuals and interaction, which could be tested by users and in turn, provide more reliable and actionable feedback. This helped to identify issues related to visual design and more complex interactions, whilst also facilitating potential future stakeholder buy-in/communication by delivering a tangible and polished representation of the product.

The Figma-designed prototypes for the Player Predictions page and the Similar Player Search page can be found below. See appendix C for the Landing page and Player Details page prototypes.

TRACscout

[Home](#)[Predict](#)[Similar Player Search](#)[Player Data](#)

Player Predictions

Select a player and predict their future values or most optimal position

Jordan Henderson

Age: 33 | Nationality: English | Club: Liverpool FC

By Value

By Position

Current Value (2023):
€15,000,000

Predicted Value (2024):
€13,400,000

Historic Value

| Year | Value |
|------|-------|
| 2022 | |
| 2021 | |
| 2020 | |
| 2019 | |
| 2018 | |
| 2017 | |
| 2016 | |
| 2015 | |

TRACscout

© 2023 TRACscout. All rights reserved.

TRACscout

[Home](#)[Predict](#)[Similar Player Search](#)[Player Data](#)

Similar Player Search

Find similar players given a range of metrics from a database including 1000s of players

Player Name

Jordan Henderson

of similar players wanted

20

Category

Shooting

Min. Age

18

Max. Age

30

Min. Value

€500,000

Max. Value


€1,000,000

Comparison Type

Per 90

Search

Sort by: Recommended




Josh Anderson

Age: 33

Club: Manchester Utd

Value: €




Josh Anderson

Age: 33

Club: Manchester Utd

Value: €




Josh Anderson

Age: 33

Club: Manchester Utd

Value: €




Josh Anderson

Age: 33

Club: Manchester Utd

Value: €




Josh Anderson

Age: 33

Club: Manchester Utd

Value: €




Josh Anderson

Age: 33

Club: Manchester Utd

Value: €




Josh Anderson

Age: 33

Club: Manchester Utd

Value: €




Josh Anderson

Age: 33

Club: Manchester Utd

Value: €



Josh Anderson

Age: 33

Club: Manchester Utd

Value: €

1234...20

36

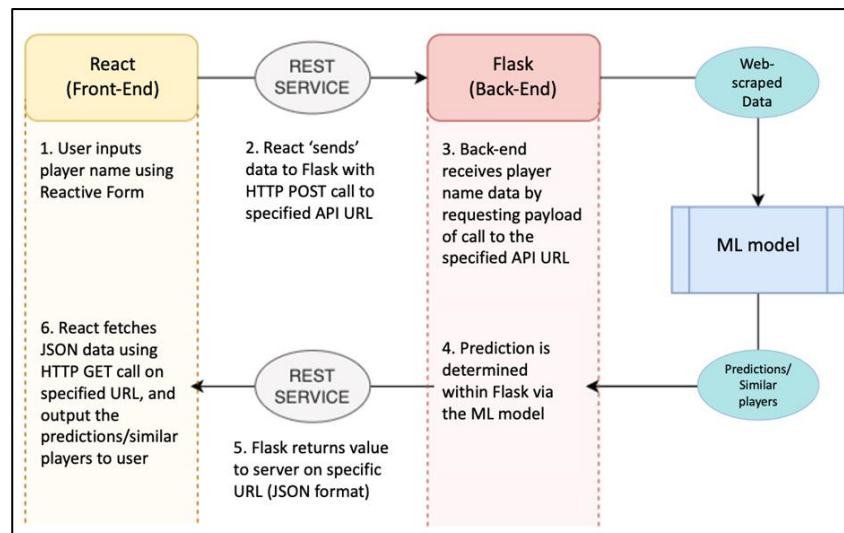
When designing the improved UI leveraging the user feedback, the standard Design Laws were considered to aid an optimal user experience. Implementing a ‘gallery’ of similar player results together in a constant form, allows users to perceive them as continuous and smooth, as recommended by the Law of Continuity [23]. Additionally, given Miller’s Law, which proposes that the average person can only keep seven items in their working memory, a ‘favouriting’ functionality was implemented to allow users to filter the displayed players and view/compare these more efficiently in a separate page [24]. In turn, users can process the information more easily and make an informed decision, focusing on the players that they care about. Similarly, displaying detailed information for each player in the ‘Player Details’ page minimises the excessive use of text in the Similar Player Search gallery, also improving readability.

6.4 Integration of Flask with React

To create the web application, the popular React JavaScript library was used for building user interfaces, whilst Flask, the lightweight Python web framework, was utilised for server-side development. This tech stack is beneficial due to its efficient and modular architecture, scalable performance, rich ecosystem, versatility, and compatibility with other third-party tools and services.

Common design systems and libraries were used both for efficiency and to ensure that the application utilises commonly accepted design standards to aid the easy use of the website, such as *bootstrap*.

A high-level overview of how the data moves between the front- and back-end, utilising the different services and APIs, to output player predictions or similar players can be seen below.



6.5 Test cases and final implementation

To demonstrate the final implementation of TAGscout's value prediction function we can use a test case. If the user inputs the name "Lewis Dunk" on the Player Predictions page, and hits enter or clicks the "Get Prediction" button, TAGscout will pass this name to the Python back-end and calculate the required values.

The screenshot shows the TAGscout website's 'Player Predictions' page. The page has a dark blue header with the TAGscout logo and navigation links: Home, Predict, Similar Player Search, and Player Data. The main heading is 'Player Predictions' with the subtitle 'Select a player and predict their future values or most optimal position'. Below this is a form with a 'Player Name' input field containing 'Lewis Dunk' and a 'Get Prediction' button. The background features a faint constellation pattern.

The second screenshot shows the results page after clicking 'Get Prediction'. It has the same header. Below the header, there are two tabs: 'Value' (selected) and 'Position'. The 'Value' tab displays the following information:

- Current Value (2023): €16000000.0
- Predicted Value (2024): €11840988.28
- Historic Value:

| Date | Value |
|--------------|-------------|
| Jun 20, 2023 | €16000000.0 |
| Nov 3, 2022 | €18000000.0 |
| Jun 15, 2022 | €20000000.0 |
| Dec 23, 2021 | €22000000.0 |
| Jun 8, 2021 | €25000000.0 |
| Mar 18, 2021 | €25000000.0 |
| Oct 13, 2020 | €27000000.0 |
| Jul 30, 2020 | €30000000.0 |
| Apr 8, 2020 | €18000000.0 |
| Dec 10, 2019 | €20000000.0 |
| Sep 12, 2019 | €20000000.0 |
| Jun 13, 2019 | €12000000.0 |
| Dec 19, 2018 | €12000000.0 |
| May 26, 2018 | €10000000.0 |
| Jan 2, 2018 | €7000000.0 |
| Jun 28, 2017 | €5000000.0 |
| Feb 27, 2017 | €5000000.0 |
| Sep 1, 2016 | €4000000.0 |
| Apr 20, 2016 | €3500000.0 |
| Sep 24, 2015 | €5000000.0 |
| Feb 24, 2015 | €1000000.0 |
| Jan 6, 2014 | €500000.0 |
| Jul 12, 2013 | €700000.0 |
| Mar 2, 2013 | €1000000.0 |
| Jun 22, 2012 | €1000000.0 |
| Feb 7, 2012 | €750000.0 |
| Oct 21, 2011 | €100000.0 |
| Jul 30, 2010 | €50000.0 |

The page will then populate with the player's (in this case, Lewis Dunk's) current value, predicted value for 2024 and all of his historic values as seen below.¹⁵

¹⁵ To reproduce these findings, please see appendix D for the user manual.

At present, the similar player search and player data pages have not been successfully linked the back-end. However, the implemented template for the similar player search page can be seen below.

6.6 Front-end conclusions

In all, the development of a web-application significantly increases the ease with which end-users can interact with the data and minimises the need for a technical background. It simplifies and streamlines the outputs from the Python backend to create an intuitive user experience, maximising the insights that can be gained from the web-scraped data.

7 Professional Issues

This section will assess various real-world issues that may stem as the result of reckless implementation of sports data analytics in football. These will range from general to legal to on-pitch issues.

7.1 General issues

Whilst this project largely advocates for the use of data analytics in European football, it is important to consider the drawbacks of this as well. It is essential that the direction that sports data analytics takes is intentional and carefully examined so as not to produce undesirable side-effects.

There are a host of specific issues with the use of data and machine learning in football but perhaps the most general is that the current analytical landscape simply isn't complex enough yet. The 'Moneyball' revolution was famously initiated in Baseball. This worked in baseball partially because baseball is a fairly static game; it is very easy in baseball to categorise certain events and analyse them in isolation. However, football suffers from being an immensely dynamic game. Football is not just about goals, passes and tackles - it is also about off the ball movement, dragging defenders away with decoy runs, positional play and so much more. Space and the exploitation of space are fundamental in football. The problem this creates is that it isn't straightforward to categorise events and analyse them. This necessitates that sports data analytics in football continues to evolve and become ever more complex. Spatio-temporal dynamics are beginning to be assessed (as discussed briefly in section 2) but these require a lot of advanced data and are computationally very intensive.

Furthermore, even the successful deployment of these advanced techniques would likely raise transparency and interpretability issues as is regularly the case in machine learning, due to the black box nature of the methods.

Football data analytics is likely in its infancy, and it is essential, not only that the methods improve and develop, but that they retain a certain degree of interpretability. Afterall, football is for the fans and a sport that becomes the domain solely of university data-science graduates and PhD candidates is likely to alienate large portions of its fans and lose its charm.

7.2 Data-literacy in football

This leads onto another issue with football data analytics: a lot of the current football world is not particularly data-literate. There are a number of immensely talented people working within and around the game who

do not have the background to fully understand, interpret and appreciate the insights provided by data analytics. This poses two issues, firstly it acts as a barrier to acceptance of data analytics within a very traditional game. This can be seen in the reluctance to accept *Expected Goals* (xG) as a valid concept. Many in the game (both fans and those closer to the decision making) have been reluctant to accept and utilise xG. The reasons for this are complicated and far reaching, however, there are some notable issues with how xG has been interpreted and marketed in the public eye. Expected goals is regularly used as a justification that “team A *should* have beaten team B”. This raises issues because an objective metric is now being used to make a subjective statement. The metric simply measures the quality of chances that a team creates, nothing more, nothing less. Critics of xG could quite justifiably argue that a team that capitalises on the limited chances it creates is more deserving of accolade and points than a team that is wasteful but creates a lot of chances. Unfortunately, the result of such lines of thought is often to criticise xG as a concept, rather than the interpretation itself. There are various solutions to this such as efforts to be more intentional and nuanced with the interpretations of xG but it highlights an overarching problem with the use of data in football: there needs to be a clear distinction between what the data is *actually* showing, and how that data is being interpreted by its user. This may go some way in easing the reluctance to embrace data analytics in football.

However, perhaps more pressingly, the lack of data-literacy also poses ethical issues. It is not hugely ethical to stand by and observe as the current wave of talented personnel in the game potentially come to see themselves replaced by data scientists. This also is not necessarily optimal; homogeneity of opinion is rarely a good thing in any context - conflicting viewpoints are often the catalyst for innovation. Therefore, it is essential that efforts are made to integrate existing staff in the football sector with the incoming wave of data scientists. This could take several forms, but there are two courses of action that would be a good starting point. The first is that time and effort needs to be dedicated to upskilling and educating the current personnel so that they have at least a rudimentary understanding of data science concepts. This could be short courses/qualifications or perhaps just the promotion of seminal works in the field such as the book *Soccernomics*. Secondly, the progression of analytical methods itself needs to bear in mind the target audience. It should remain a key aim of football data analytics research to develop efficient and effective methods that can be conveyed (at least in some capacity) to those with a less extensive grasp on data analytics. This will ease both of the concerns mentioned in this section: current personnel will be better equipped to tackle the new football landscape; and as a result there may be a more widespread acceptance of, and trust in, data methods.

7.3 Fundamental issues with value prediction

More specific issues may arise from value prediction itself. Transfer value is a partial byproduct of a player's importance to a specific club and the alternative options available. Value prediction cannot necessarily factor this in adequately. Wide-scale adoption of value prediction methods may force some clubs to sell below a value that they deem to be appropriate - or equally to buy above the odds. This is particularly problematic as the prediction will always have outliers. For example, Moisés Caicedo's Transfermarkt value jumped from €6,000,000 to €75,000,000 within one season [25]. It is unlikely that any model based on crowd-sourced values would have been able to predict this degree of change.

Alternatively, players playing for certain clubs may accrue favourable statistics due to the calibre of players around them or the system. For example, a midfielder in a possession-based team may have disproportionately positive passing statistics that would not necessarily be replicated elsewhere.

An overdependence on value prediction algorithms could also lead to forms of market manipulation if clubs learn how to influence the algorithm. Consider that clubs find total shot volume to be a factor that drives player value up. A player that a club wishes to sell could be brought on against weaker opponents or when a team is comfortably winning and told to shoot every time they touch the ball so that the algorithm subsequently boosts their value and the club has more bargaining power in negotiations with other clubs. Downstream from this there could even be an exacerbation of wealth inequality between clubs as richer and better clubs spend more time in winning positions in games and can afford to sacrifice a portion of that game to artificially boost the value of their players. There are obvious sporting integrity issues with throwing portions of a game (along with implications for the betting industry) but this also raises issues about fairness and protecting upward mobility and meritocracy in the sport.

This problem doesn't disappear if crowd sourced data is used either. Football could very easily become a popularity contest with clubs focusing on increasing their players' crowd sourced values instead of improving their tactical, technical and physical proficiencies. This could be even more problematic when considering the impact of racism in sport. Szymanski's 2000 paper on market discrimination in sport found signs of racism in the wage bills of clubs [20]. Whilst this study focused on data from 1978-93, it is no secret that racism still exists in the modern game. If the data that machine learning algorithms are being trained on included racial bias, these biases could be emphasised and somewhat immortalised. This has already been a problem with the deployment of machine learning in other sectors. Amazon famously developed an algorithm to screen job

applications which ended up heavily discriminating against women due to the male-dominant nature of previous company employees [21]. Furthermore, Szymanski's study hypothesised that racial discrepancies in wages could have been the result of clubs responding to fan prejudice. This would hence suggest that crowd sourced values could suffer from this same issue. This issue needs to be studied further to determine the implications of deploying value prediction algorithms on a large scale.

7.4 Legal

There also exist an array of legal issues that may prove problematic. Increased instances of sites providing free data being web-scraped may lead to them banning web-scraping and/or making efforts to structure their webpages in a way such that it is less conducive to being web-scraped. This could push more companies to lock their data behind paywalls and costly APIs. Much like the issue of analytical methods in the game becoming too complicated and alienating fans, increasing price barriers risks alienating the community of football data enthusiasts that currently rely on free data. Furthermore, even in the event that free data is still available, if more people use FBRef's free representation of Opta data instead of paying Opta directly, the company (and similar organisations) may struggle to afford to innovate and collect that data in the first place.

There also exists the issue of copyright and ownership. If the data is web-scraped from FBRef but then significantly altered and pre-processed, who owns this data? What about the results of any analysis using data from another company? If Opta collects data and produces insights on its own, that is fairly straightforward legally. But in the case where analysts use Opta's data to produce their own insights, the copyright and ownership issues may become more complex.

7.5 Strategic homogeneity

Football as a game has numerous schools of thought about how the game should be played and the best way to maximise the use of resources, performances and results. This is one of the many things that makes football captivating. An overdependence on data may remove this facet of the game and results in a form of "strategic homogeneity" so to speak - thinkers and actors within the game all beginning to follow the same strategy. Given that the metrics currently available aren't perfect, increased data dependency too soon could cause recruitment and training methods to angle towards the maximisation of incomplete or fundamentally flawed metrics. This in turn may become a self-fulfilling prophecy - players are picked based on these metrics and hence the data that the algorithms are trained on detects an emphasis on these metrics and perpetuates this.

This homogeneity of strategies in football may in fact stand in opposition to the “spirit of the game”. In the 2015-16 season, Leicester City pulled off what is largely considered a sporting miracle - and perhaps the greatest underdog story in the history of the sport - overcoming 5000/1 odds to win the Premier League, having been seen as favourites for relegation. To do this, they outperformed their expected points by 12.06.¹⁶ The expected points table placed them 4th. In the period spanning from the 2014-15 season to the 2022-23 season, six of the Premier League winners have placed first in the expected points table and two have placed second in the expected points table. The only exception to this trend was Leicester City in 2015-16. [22]

On the one hand, this demonstrates the validity of the expected points model. However, would Leicester have been able to achieve what they did if everyone had believed too much in expected points? Would this have influenced the sports psychology of the players, staff and fans? Would they have even tried to play a counter-attacking style of play that didn’t seek to maximise xG? If the price of a data revolution in football is that it may strip away some of the most incredible moments in the sport’s history, it is worth investigating exactly how to avoid this sort of eventuality.

7.6 Summary of professional issues

The purpose and conclusion of this section is not to temper enthusiasm about the potential of data analytics in football, nor to advocate for the halting of its progress. The key takeaway is simply that there is a balancing act to be performed and data scientists and stakeholders in football must do their due diligence and proceed with caution.

Additionally, it is worth noting that data analytics could provide a solution to many of the issues raised in this section. For example: more objective forms of analysis (such as those that can be provided by data analytics) may in fact reduce prejudicial biases in scouting; the increasing complexity of the data methods deployed in football may add a fascinating new lens through which to see the game and inspire participation from a cohort of people who may not have otherwise had an interest in football; decreased scouting costs may free up investment into the grassroots of football and scouting of local communities may become less costly and more accessible through data analytics. These are just some of the potential benefits. However, more research is needed to determine the best course of action and a great deal of care needs to be taken when ushering in any data revolution in European football.

¹⁶ *Expected points is a model that predicts the number of points a team will take from a game based on their xG created and their xG conceded.*

8 Conclusion

8.1 Project review

8.1.1 Ambition and scope

The core aim of the project was to provide a useful scouting tool for the analysis of European football, using freely available data. For this reason, the project had a more ambitious aim than many of the other published works on the subject - not to predict a player's current value, but to predict their value one season ahead. Additionally, in doing this the project aimed to move away from video game data and focus on alternative metrics, using crowd-sourced data to provide market values for each player. In hindsight, the project may have benefitted from retaining more of the core structure of previous literature on the subject. Given the already ambitious nature of predicting future value, it may have been more useful to keep this as a sole "independent variable" of sorts - keeping everything else about the project as similar as possible to previous works. This would have meant foregoing the exploration into crowd-sourced data and instead focusing on video game data as has previously been seen. In such a project, validity in the real world could be tested by utilising the results of the analysis to predict some real outcomes such as goals, assists or results.

Regarding the planning element of the project, a benefit may be felt from establishing the project goals in such a way that regardless of the outcome concrete conclusions could be drawn. This would require structuring the analysis so that even if the models do not perform as desired, this in and of itself would tell us something about how achievable the initial goal was. For example, a value prediction model that was provably effective at predicting current values but could not reliably predict future values would allow for the drawing of conclusions about how possible future value prediction is - rather than the current ambiguity around the impact of the crowd-sourced training data.

With that said, the goal of incorporating the results of the analysis into a web-application to ensure it is easier to interact with is one that I feel would remain were I to undertake the project again. As discussed in section 7.2, I believe that all and any developments in football data analytics should have a strong focus on interpretability and be accessible to those without a formal background in data science.

8.1.2 Alternative data sources

Assuming that crowd-sourced data was still used in the project, it may have been useful to incorporate some more non-performance related

metrics. For example, a more stable measure of value could have been included, such as a player's salary obtained from a site such as Capology. Alternatively, there may be a more direct way to measure popularity and a player's media reach. Perhaps, the number of social media followers a player has or Google Trends data on searches for that player could have been incorporated to factor in the more nuanced aspects of a player's crowd-sourced value. Furthermore, whilst the project used the most recent data available in an attempt to stay up to date with European transfer market inflation, older data may have been more stable as it avoids any potential impacts of COVID-19 on the analysis (as discussed in section 4.5).

8.1.3 Front-end improvements

The front end was a challenging but essential part of the project as the primary use-cases for TAGscout required that users with limited data-literacy could interact with the data. On the one hand, this was a very rewarding part of the project for me personally. I had limited experience with front-end development prior to this project and believe that the skills I've acquired through its implementation (such as competency in JavaScript and HTML) will serve me well in my future career. However, there are a few notable areas for improvement. The similar player search function is not yet implemented fully in the front-end. Furthermore, the current Python visualisations were fairly basic, and a more interactive design would improve user experience dramatically. For example, the ability to hover over a point on the graph and have the player that this corresponds to show up would have been a significant upgrade on the current implementation.

Additional improvements to the efficiency of the web-application would also have been useful. Perhaps instead of calculating a player's predicted value after their name has been entered, these values could be pre-calculated for each player and stored so that they are quicker to access. Moreover, the application would benefit from better error handling. A player's name must be entered in the exact format it appears in the back-end otherwise the values will not be calculated and displayed. Restructuring the input handling so that it is case insensitive would be of benefit and a feature to suggest corrections for small typos may be an interesting ad-on to improve user experience.

As mentioned in section 6.1, the web-application still shows the basic layout for the implementation of a positional prediction feature. Both the implementation of such an analysis (using a dataset with more in-depth player positions) and the implementation of it via the web-application would move TAGscout further in the direction of being a multi-use scouting platform.

8.1.4 Evaluation of the similar player search function

For the similar player search function, the use of PCA for dimensionality reduction to ensure a simple and intuitive visualisation remains a strength of the implementation. Whilst it does decrease interpretability, almost all dimensionality reduction techniques suffer from this issue and PCA benefits from providing minimal information loss in our analysis. Furthermore, attempts to include and visualise more dimensions would have resulted in a clunky visualisation that would make gaining insights from it more difficult. In terms of improvements, further categories of attributes to compare would never be a hindrance - equally the ability to create custom categories would be a useful feature for clubs that have very specific recruitment criteria.

8.1.5 Limitations of the value prediction model

Naturally, a key area for improvement in the project is finding ways to address the limitations of the value prediction model - most likely via the inclusion of alternative data and potential omission of crowd-sourced data. However, the attempts made to rectify these issues within this project were extensive and it is reasonable to assume that the data itself was likely more of an issue than the methods used to analyse it. Despite the suboptimal performance of the model, there are a number of positives and lessons to take away from its implementation. The investigation into the limitations of evaluation statistics is useful in highlighting the complexities in the implementation of machine learning and data analytics. It demonstrates the need for human intervention and interpretation based on a comprehensive understanding of the data and the context. More specifically, it reiterates the need for due diligence on the part of data scientists and emphasises the importance of a sound theoretical understanding in addition to practical skills. Furthermore, the results of this analysis have implications for the future perception of validity for crowd-sourced statistics. Indeed, the limitations of Transfermarkt values highlight the potential benefits of more disclosure from clubs around true transfer fees and the need for this data to be publicly accessible. On a more personal level, in trying to address the model's limitations, I have developed a strong competency across a wide range of machine learning techniques - particularly those centred in PyTorch. I have developed a more intuitive understanding of many neural network architectures, regularisation techniques, learning rate schedules, and data pre-processing and transformation methods.

8.1.6 Strengths of the project

The decision to implement multiple functions for TAGscout was well-advised. Particularly in light of the value prediction function's limitations, it is positive to be able to showcase a similar player search function that

operates as intended. Furthermore, a commitment to the use of freely available data is one that I would maintain were I to undertake the project again; in my opinion, this remains important for the growth of data-literacy in European football.

Perhaps the biggest strength of the project was in obtaining and structuring the data. The object-oriented and dynamic set up that was used was invaluable as the project progressed. For example, the original value prediction models were trained on only one season's worth of data, this provided lacklustre evaluation statistics and so more data was web-scraped. This process of acquiring additional data was incredibly straightforward due to the use of the *fbref_top5_webscraper* and *transfermarkt_scraper* functions. Moreover, during the data-collection part of the project, FBRef changed the structure of their website. Whilst this did cause issues with the code, the lack of manual data-handling meant that once the required updates were made the web-scraping functions, the data was seamlessly reintegrated into the project, without the need for more input from me. This would also be applicable to any future changes that FBRef makes to the structure of their webpage, requiring minimal changes to the code to keep it functioning.

8.2 Summary

The degree to which the project successfully adheres to the three initial constraints varies. The utilisation of freely available data had mixed results. Whilst all the data did indeed come from freely available sources, the limitations of such data were somewhat highlighted in the value prediction model. However, the project did successfully remain dynamic and object-oriented in its implementation. Future iterations of the code could look to expand on this and further decrease the friction involved in acquiring high-quality football data.

The project set out to deliver a multi-purpose scouting tool that could appeal to end-users who lack formal technical training in data analytics. In this sense, the full-stack development that saw a machine learning based back-end connected to an intuitive and aesthetically pleasing front-end succeeded in appealing to the end-user - particularly through the visual implementation of the similar player search function. The project also aspired to justify the use of machine learning in European football and prove the power of such algorithms. In this sense the project falls somewhat short due to the limitations of the value prediction algorithm. However, the project is still useful, it serves as a necessary review of evaluation statistics as well as an analysis of the validity of crowd-sourced data. Whilst TAGscout requires some improvements to deliver on the goal of being a fully-fledged scouting platform, the findings of this project can meaningfully contribute to the football data analytics conversation as the sport looks to become ever more dependent on data.

References

- [1] Deloitte, "Market size of the professional soccer market in Europe from 2010/2011 to 2021/2022 (in billion euros).", Statista, 2023.
- [2] Capology, [Online]. Available: <https://www.capology.com/uk/premier-league/payrolls/>. [Accessed 03 July 2023].
- [3] Capology. [Online]. Available: <https://www.capology.com/uk/premier-league/transfer-window/2022-2023/>. [Accessed 03 July 2023].
- [4] Capology. [Online]. Available: <https://www.capology.com/uk/premier-league/transfer-window/2022-2023/>. [Accessed 03 July 2023].
- [5] S. Kuper and S. Szymanski, in *Soccernomics: why England loses, why Germany and Brazil win, and why the U.S., Japan, Australia, Turkey - and even Iraq - are destined to become the kings of the world's most popular sport.*, New York, Nation Books, 2009, p. 3.
- [6] D. Yam, "StatsBomb," 21 February 2019. [Online]. Available: <https://statsbomb.com/2019/02/attacking-contributions-markov-models-for-football/>. [Accessed 15 July 2023].
- [7] K. Singh, "Introducing Expected Threat (xT)," February 2019. [Online]. Available: <https://karun.in/blog/expected-threat.html>. [Accessed 11 July 2023].
- [8] N. Mackay, "Improving My xG Added Model," 2017. [Online]. Available: <https://mackayanalytics.nl/2017/07/28/improving-my-xg-added-model/>. [Accessed 22 July 2023].
- [9] J. Fernández, L. Bornn and D. Cervone, "Decomposing the Immeasurable Sport: A deep learning expected possession value framework for soccer," in *MIT Sloan Sports Analytics Conference*, Boston, 2019.
- [10] B. Gonçalves, D. Coutinho, J. Exel, B. Travassos, C. Lago and J. Sampaio, "Extracting spatial-temporal features that describe a team match demands when considering the effects of the quality of opposition in elite football," *PLOS One*, vol. 14, no. 8, 2019.
- [11] G. Liu, Y. Luo, O. Schulte and T. Kharrat, "Deep soccer analytics: Learning an action-value function for evaluating soccer players.," *Data Mining and Knowledge Discovery*, vol. 34, 2020.
- [12] H. Ruiz, P. Power, X. Wei and P. Lucey, ""The Leicester City Fairytale?": Utilizing New Soccer Analytics Tools to Compare Performance in the 15/16 & 16/17 EPL Seasons.," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017.
- [13] D. Sourya, *Pricing Football Players Using Neural Networks*, 2017.

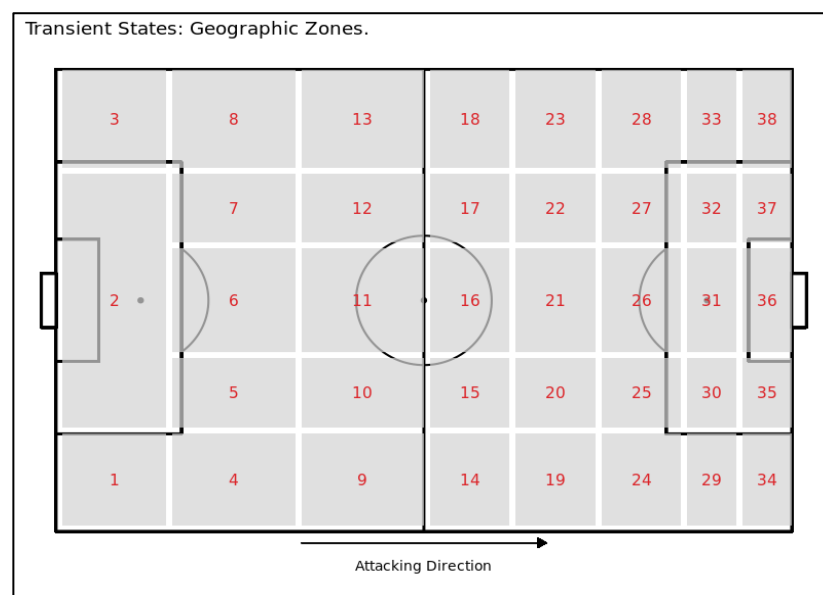
- [14] V. Steve Arrul, P. Subramanian and R. Mafas, "Predicting the Football Players' Market Value Using Neural Network Model: A Data-Driven Approach," in *2022 IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)*, Ballari, 2022.
- [15] M. A. Al-Asadi and S. Tasdemir, "Predict the Value of Football Players Using FIFA Video Game Data and Machine Learning Techniques," *IEEE Access*, vol. 10, pp. 22631-22645, 2022.
- [16] R. Stanojevic and L. Gyarmati, "Towards Data-Driven Football Player Assessment," in *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, Barcelona, 2016.
- [17] SciSports, [Online]. Available: <https://www.scisports.com/> . [Accessed 12 July 2023].
- [18] FBRef, "All About FBref.com," [Online]. Available: <https://fbref.com/en/about/>. [Accessed 10 July 2023].
- [19] J. Surowiecki, *The wisdom of crowds: Why the many are smarter than the few and how collective wisdom shapes business, economies, societies, and nations.*, New York: Doubleday & Co, 2004.
- [20] Deloitte, "Annual Review of Football Finance 2021," 2021.
- [21] D. McCarrick, M. Bilalic, N. Neave and S. Wolfson, "Home advantage during the COVID-19 pandemic: Analyses of European football leagues," *Psychology of Sport and Exercise*, vol. 56, 2021.
- [22] eMarketer, *Average daily time spent on social networks by users in the United States from 2018 to 2022*, Statista, 2021.
- [23] Medium, "Using Gestalt principles in UX design," 21 September 2019. [Online]. Available: <https://uxdesign.cc/using-gestalt-principles-in-ux-design-3fc64614d3ef#:~:text=5.,on%20the%20line%20or%20curve..> [Accessed 29 August 2023].
- [24] Laws of UX, "Miller's Law," [Online]. Available: <https://lawsofux.com/millers-law/>. [Accessed 29 August 2023].
- [25] Transfermarkt, "Moisés Caicedo," [Online]. Available: <https://www.transfermarkt.co.uk/moises-caicedo/profil/spieler/687626>. [Accessed 29 August 2023].
- [26] S. Szymanski, "A Market Test for Discrimination in the English Professional Soccer Leagues," *Journal of Political Economy*, vol. 108, no. 3, pp. 590-603, 2000.
- [27] J. Dastin, "Amazon scraps secret AI recruiting tool that showed bias against women," *Reuters*, 11 October 2018.
- [28] Understat, [Online]. Available: <https://understat.com/>. [Accessed 18 August 2023].

Appendices

Appendix A: StatsBomb's NSxG model

In order to utilise Markov Models, Yam defines two outcomes: a goal or a turnover.¹⁷ Both of these outcomes have a 0% probability of transitioning out of said state. StatsBomb then defines 84 other transient states based on contextual events (such as attacking third free kick or penalty won, etc.), geographic zones when they are under pressure and geographic zones when they are not being pressured. Following this, the expected number of plays (defined as passes, carries and shots) until an outcome is reached can be calculated using the matrices from the model.

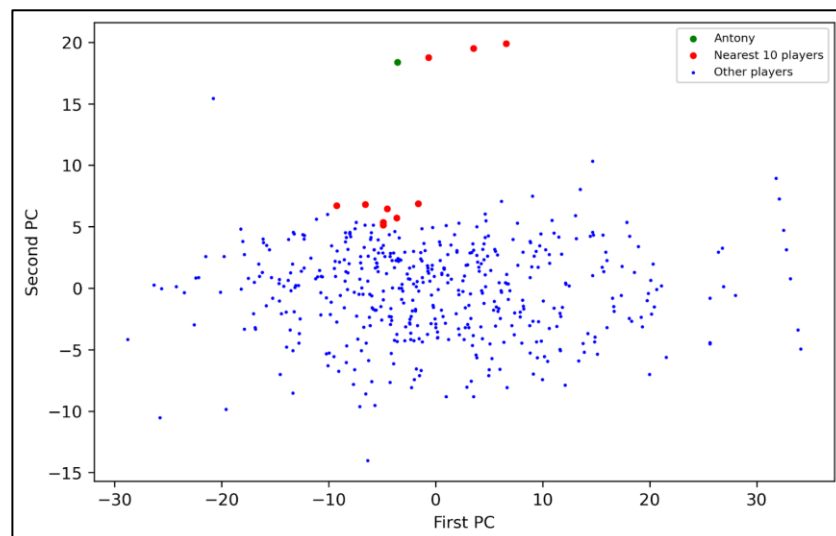
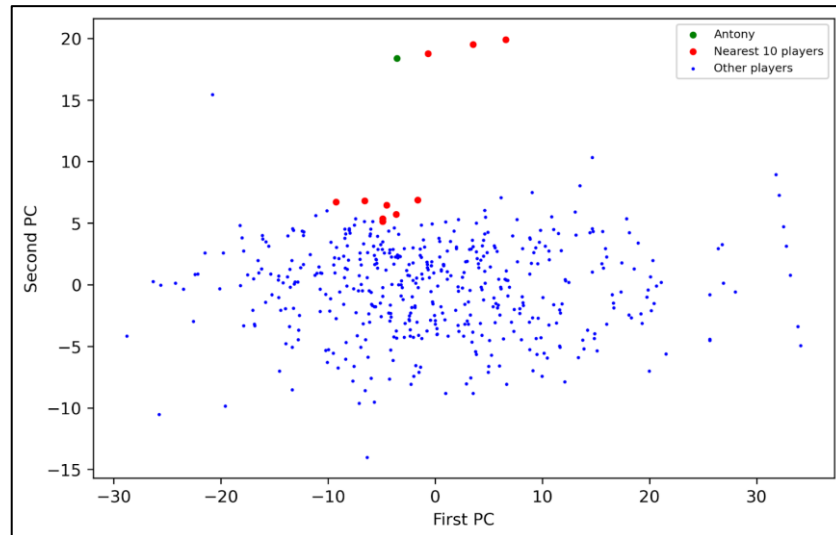
From this the model reveals that the state with the highest chance of becoming a goal is zone 36 (a state that was defined as near enough the opposition's 6-yard box) with pressure - a probability of 19.2%. Whilst the state most likely to result in a turnover of possession was zone 1 (a state that is defined as the area to the right of the team in possession's penalty area) with pressure - with a probability of 99.5%. [6] A more detailed graphic detailing StatsBomb's zone definitions can be seen below.



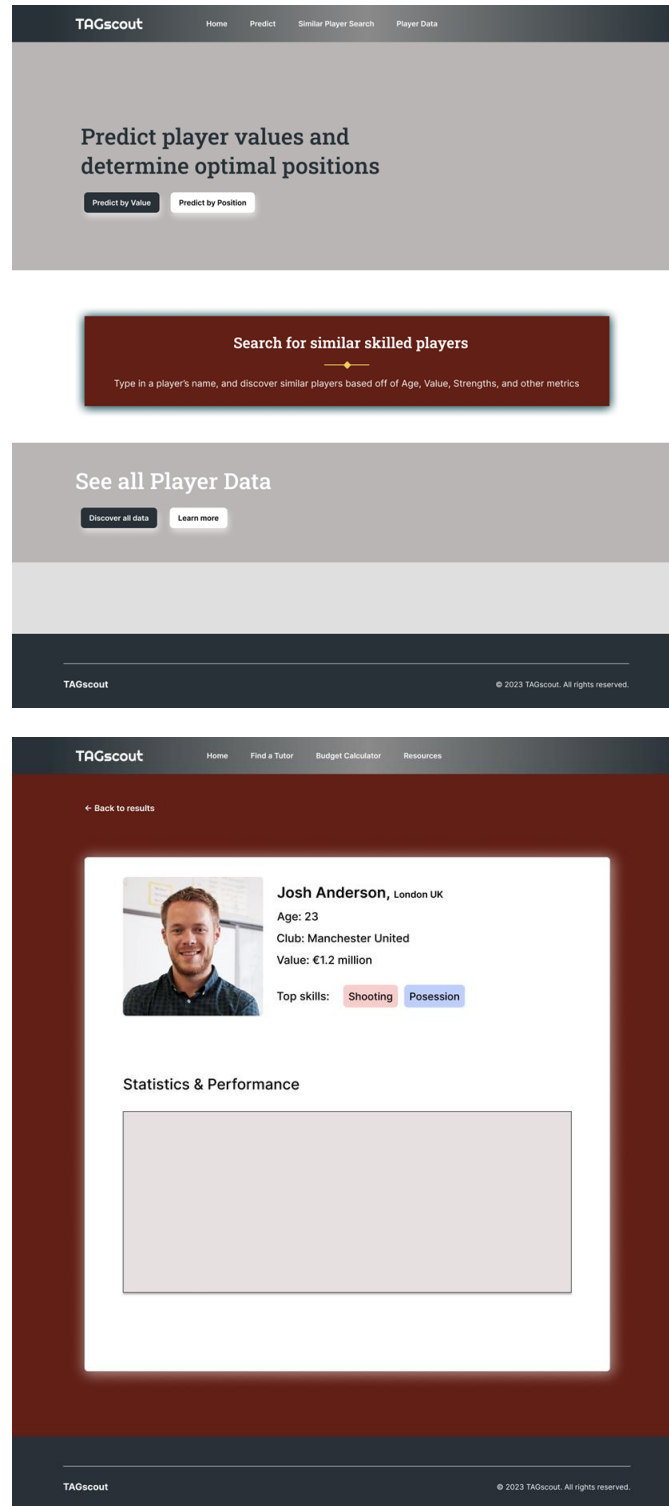
Source: Yam, D. (2019, February 21). Attacking Contributions: Markov Models for Football. StatsBomb. <https://statsbomb.com/2019/02/attacking-contributions-markov-models-for-football/>

¹⁷ The analysis was conducted by Derrick Yam on behalf of StatsBomb.

Appendix B: Further visualisations for Antony



Appendix C: High-fidelity prototypes



Appendix D: User Manual

The Final Deliverable can be found at: “tagscout-app.zip”, and is in the form of a React project, utilising a Flask back-end.

After activating a virtual environment in the tagscout-app directory, installing the application involves:

1. Installing the node programs for React using the npm package manager. To do this, run:
 - `npm install`
2. Installing the necessary Python packages. To do this, navigate to the 'backend' directory, and run:
 - `pip install -r requirements.txt`
3. Utilising the back-end, by running the Flask program. To do this, navigate to the 'backend' directory and run:
 - `flask run`
4. Running the React program in another terminal. To do this, remain in the base tagscout-app directory and run:
 - `npm start`
5. Open a web browser and navigate to:
 - `http://localhost:3000/#home`.

Contained within “100926415_Individual_Project_Code.zip” there are a number of files that were used during the testing and initial model implementation phases:

- “Data acquisition.ipynb” covers the web-scraping and data formatting section of the project. This file closely follows section 3.
- “Value prediction model testing.ipynb” covers the implementation of the value prediction algorithm along with much of the analysis that ensued. This closely follows section 4. “Value Prediction Logged Labels.ipynb” contains additional code that was used specifically to trial training the model on log transformed labels as seen in section 4.5.2.
- “Similar Players and Positional Prediction Model Testing.ipynb” covers the initial implementation of the similar player search function along with the discarded positional prediction function. This file closely follows section 5.