



Transfer Learner: Assessing the qualities of a good football team to make optimal transfer decisions using machine learning and optimisation

CS344 Discrete Mathematics Project

Author: Joshua Uwaifo

Supervisor: Dr Theo Damoulas

Department of Computer Science
University of Warwick

Academic Year: 2016/2017

Word Count: 16,926

Initial Roster			Updated Roster		
	Player name	Position		Player name	Position
●	S Given	goalkeeper	●	S Given	goalkeeper
●	K Schmeichel	goalkeeper	●	K Schmeichel	goalkeeper
●	J Hart	goalkeeper	●	J Reina	goalkeeper
●	M Ball	defender	●	J Bosingwa	defender
●	W Bridge	defender	●	M Ball	defender
●	N Onuoha	defender	●	W Bridge	defender
●	P Zabaleta	defender	●	N Onuoha	defender
●	M Richards	defender	●	P Zabaleta	defender
●	V Kompany	defender	●	V Kompany	defender
●	J Garrido	defender	●	J Garrido	defender
●	R Dunne	defender	●	R Dunne	defender
●	M Johnson	midfielder	●	M Johnson	midfielder
●	S Wright-Phillips	midfielder	●	S Ireland	midfielder
●	G Fernandes	midfielder	●	Elano	midfielder
●	Elano	midfielder	●	F Lampard	midfielder
●	M Petrov	midfielder	●	D Hamann	midfielder
●	S Ireland	midfielder	●	G Fernandes	midfielder
●	D Hamann	midfielder	●	N de Jong	midfielder
●	N de Jong	midfielder	●	S Wright-Phillips	midfielder
●	K Etuhu	forward	●	E Hadji Diouf	forward
Swaps Made:					
Swap J Hart with J Reina from liverpool					
Swap M Richards with J Bosingwa from chelsea					
Swap M Petrov with F Lampard from chelsea					
Swap K Etuhu with E Hadji Diouf from blackburn					
Effects on:					
Budget:			Value:		
£0.5 M (Fantasy Premier League Millions)			Added value on the team after suggested swaps: 55.4029454539		

Abstract

This project aims to automatically generate transfer decisions a football team should make in order to increase their current capabilities with/without a specified budget. Transfer decisions are a list of players a team should buy from other teams and/or sell to other teams for a given price. The project will use machine learning techniques to model what makes a successful team, past and present. The datasets obtained were extracted from the necessary websites using a Python web scraper. Optimisation techniques are used to maximise the value of a team after the transfer decisions have been made given constraints. These constraints include the total cost of the transfer decisions being cost effective or not. In addition, a web interface has been made to add user engagement and as an added extra to the project.

Keywords

Transfer Learner, Football, English Premier League, Supervised learning, Regression, Transfer decisions, Optimisation, Linear program

Acknowledgements

I would first like to thank my supervisor and as a result of this experience, friend, Dr Theo Damaoulas. Thank you for taking me under your wings and helping me to complete this project. He provided the needed expertise at many points over the duration of this project. He allowed me to do a project in an area that means a lot to me, football, in a fun yet very technical and academic atmosphere.

I would like to acknowledge a few (of the many) students that have helped me over the course of the project. I would like to thank Rhiannon Micheltmore and Alen Buhanec (4th year Computer Science), for their technical input over the project, especially when it came to the Machine Learning and Shell scripting aspect of the project. I would also like to acknowledge Joseph Sigbeku and Daniel Barovbe (2nd year Computer Science) for their late night encouragement in DCS when it came to creating a web scraper to extract the information needed. Lastly, I want to thank Daniel Namu-Fetha (3rd year Computer Science) for his technical help and patience during the creation of the web interface made for this project. I also want to thank him for the technical conversations that have helped build my technical skills over the last year.

To conclude and wholeheartedly, I'd like to thank God, my friends and family (you all know who you are). This project is not just for me, it is for you all, emphasising the beauty of teamwork, hope, faith and love.

List of Figures

3.1	Image of positions in Football: Goalkeeper(s), Defender(s), Midfielder(s) and Forward(s) ^[56]	19
5.1	Dependencies between objectives	29
7.1	Transfer Learner's Architcture	32
7.2	Transfer Learner's High Level Overview	33
7.3	Example of WhoScored data ^[54]	35
7.4	Image of Premier League website showing 1992/93 season option for teams ^[43] . .	36
7.5	Image of premier league points total for the 2015/16 Premier League Season ^[55] .	37
7.6	Image of premier league prize money for best 4 teams in 2015/16 ^[36]	38
7.7	Example image of Positive and Negative correlation ^[57]	40
7.8	Components of a supervised learning system ^[10]	45
7.9	Visual description of overfitting and underfitting ^[46]	48
7.10	Image of a (convex and smooth) squared error loss function ^[58]	50
7.11	Example of a possible Decision Tree constructed by model	57
8.1	Python code for the cost effective transfer decision	73
8.2	Portal Screen to web interface	75
8.3	User selection of Birmingham in the 2009/10 Premier League season	75
8.4	Loading screen before transfer decisions are outputted	76
8.5	Cost effective and greatest value transfer decision example	77
8.6	Modal showing the cost effective transfer decision for Birmingham in the 2009/10 EPL season	78

8.7	Modal showing best valued transfer decision for Birmingham in the 2009/10 EPL season	79
9.1	Example of original set of team attributes from the Premier League	81
9.2	9 Final attributes used in the final machine learning model	81
9.4	Transfer Learner's Ridge regression results, R2 score	82
9.3	Transfer Learner's Ordinary Least Squares results, R2 score	82
9.5	Transfer Learner's Lasso regression results, R2 score	82
9.6	Transfer Learner's Decision Tree regression results, Standard deviation reduction	83
9.7	Lasso-based greatest value transfer decision	85
9.8	Ridge-based greatest value transfer decision	86

List of Tables

6.1	System Specifications	30
9.1	Premier League Teams and Seasons with transfer decisions from Transfer Learner available	84

Contents

Abstract	2
Keywords	2
Acknowledgements	3
List of Figures	5
List of Tables	6
1 Introduction	11
1.1 Football Problems - Transfer Decisions	11
1.2 Problem Statement	12
1.3 Project Stakeholders	12
1.4 Author's Background	13
2 Report Structure	14
2.1 Research	14
2.2 Methodology	14
2.3 Analysis	15
3 Research	16
3.1 Machine Learning - ML	16
3.1.1 Examples of Machine Learning applications	17
3.1.2 Machine Learning Paradigms	17

3.2	Optimisation	17
3.3	Football and Transfer Decisions	18
3.3.1	(English) Premier League - EPL	18
3.3.2	Transfer Decisions	19
3.3.3	Example of a “bad” transfer decision	20
3.4	Existing Solutions and Literature Review	22
4	Legal, Social, Ethical, and Professional Issues	24
4.1	Legal Issues	24
4.2	Social Issues	25
4.3	Ethical Issues	25
4.4	Professional Issues	25
5	Objectives	27
6	Technical Constraints	30
7	Design	31
7.1	Transfer Learner’s Architecture	31
7.2	Data Extraction	33
7.2.1	Player and Team Seasonal Statistics	34
7.2.2	Team Seasonal Valuations	36
7.2.3	Player seasonal valuations	38
7.2.4	Summary of Final choices for data sources	39
7.3	Exploratory Data Analysis and Data Cleaning	40
7.3.1	Summary of data cleaning	41
7.4	Data integration (merging)	43
7.5	Machine Learning models	45
7.5.1	Supervised Learning - Regression	45
7.5.2	Overfitting and Underfitting	47
7.5.3	Standardisation of samples	48
7.5.4	Models	49

7.5.5	Choosing the Final Model for Transfer Learner	60
7.5.6	Identifying important team features	61
7.6	Optimisation	62
7.6.1	Making Transfer Decisions	62
7.6.2	Best Cost Effective Transfer Decision	64
7.6.3	Greatest value Transfer Decision	64
7.7	Web Interface	65
8	Implementation	66
8.1	Data Extraction	66
8.2	Exploratory Data Analysis and Data Cleaning	68
8.3	Machine Learning	69
8.4	Optimisation of Transfer Decisions	71
8.5	Web Interface	74
9	Results and Analysis	80
9.1	Results	80
9.1.1	Assessing the qualities of a football team	80
9.1.2	Making transfer decisions	84
9.1.3	Analysing the success of the transfer decisions made	85
9.2	Analysis of Objectives	86
10	Project Management	87
11	Author's Assessment of the Project	88
12	Conclusion	92
12.1	Summary	92
12.2	Further Work	92
12.2.1	Scalability	92
12.2.2	Integration into the world of Football	93
12.2.3	Build upon transfer decision evaluation	93

13 References	94
References	94

Chapter 1

Introduction

Football, known as Soccer^[1] in other parts of the world, is one of the world's most popular sports^[2]. The appeal towards the game is huge. For example, in the 2016/17 Premier League season, the Manchester derby generated around a billion registered viewers. The Manchester derby is a rival fixture between Manchester United and Manchester City. These 2 teams tend to fight for the league title, and with both teams having 2 of the best managers in the world in Mourinho and Guardiola, respectively, it was expected to be a classic. According to the Telegraph, the viewing for the game was expected to easily eclipse 901 million viewers, not taking into account "people watching in pubs and clubs"^[3]. Furthermore, the appeal from this sport can be seen in newspaper, television shows and amongst other media streams across the World. As you can see, Football is a beautiful sport.

1.1 Football Problems - Transfer Decisions

With the appeal generated by football, comes levels of extremism. These extremisms manifest themselves in different avenues like hooliganism^[4] and improper financial conducts^[5]. In this paper, the focus is on providing a solution to one of the financial problems in football. Currently, large sums of money is being spent on players with no real justification^[6]. For example, in the summer of 2016, John Stones, a 21 year old defender with no major history of success was signed by a financially strong football team, Manchester City, for £47.5 million. To put this

in context, in 2001, Zinedine Zidane, a winner of the World Cup and European championship, and regarded as one of the greatest footballers in his generation^[7], was signed by Real Madrid for £46 million. This emphasises the money problem in football and its impact on transfer decisions. The solution, this paper provides, is one that hopefully rebalances the power play. This is done by enabling any team, weak or strong financially, to have the ability to make transfer decisions that improve their team, in a quantifiable manner.

1.2 Problem Statement

With increased financial strength due to broadcasting rights^[8], football clubs have been able to spend obscene amounts of money on players the clubs believe will improve their team. These beliefs have caused clubs over the years to make bad transfer decisions and in hindsight, these decisions should not have been made and would have saved the team a large amount of money^[9]. The goal of this project is to use machine learning and optimisation to encourage the notion of a level playing field in football transfers. This will be achieved by creating a model that evaluates the attributes of football teams and optimising transfer decisions using constraints like the team's budget. The program will output a list of transfer decisions for a selected team, that would improve the teams value on a season by season basis.

1.3 Project Stakeholders

Since the project's inception a number of individuals have vested personal interest within the project and can be considered as stakeholders. Firstly, the project supervisor, Dr. Theo Damoulas, has played a major role offering valuable insight, suggestions and feedback regarding all aspects of the project. In addition to the project's author, Joshua Uwaifo, who has devoted his time and effort to make this project a success, other individuals have vested personal interest like (but not limited to) Daniel Namu-Fetha, Rhiannon Micheltmore, Joseph Sigbeku, Aaron Conway, Alen Buhanec and Daniel Barovbe. These individuals amongst others, have been so crucial in their support at various stages of the project. Their collective contributions/involvement in this project amplifies how valuable this project is.

1.4 Author's Background

The author's background in the area before taking on this project included 10+ years experience of playing and researching football, and just under 6 terms of a Discrete Maths degree, given their late switch from Maths at the end of first term of his first year.

In order to improve the understanding of Machine Learning topics, technically and theoretically, the author participated in a Data Science bootcamp after the exam season of 2nd year. In addition, due to the huge appreciation for the subject as a result of the boot camp, the optional modules chosen in the 3rd year of study were Data Science focused, to reach the required expertise for the project.

Chapter 2

Report Structure

This report is a combination of research, academic investigation, technical development and project management. Therefore, the report can be broken down in three main areas: research, methodology, and analysis.

2.1 Research

Chapters 3 to 4 focus on the preliminary research and considerations that took place before the software program and web interface development. Chapter 3 gives a detailed breakdown of the field of machine learning, linear programs in optimisation and football as a sport. This is important as the project is an interaction of the three. In addition, Chapter 3 focuses on the existing solutions that may or may not do what this project aims to accomplish. It also focuses on the paper that serves as the foundation behind this thesis in a literature review. The following chapter, Chapter 4, highlights the possible legal, social, ethical and professional considerations of this project and how Transfer Learner accomodates these issues.

2.2 Methodology

The creation of the software relating to Transfer Learner will be the main focus of this thesis. This section has both academic and technical themes. Chapter 5 outlines the objectives that

the project aims to solve and Chapter 6 highlights the constraints system-wise this project was under. Chapter 7 focuses on the design of the data science and optimisation workflow of the project. Chapter 8 discusses the implementation of the design considerations made in Chapter 7.

2.3 Analysis

The final section of this thesis focuses on the analysis of the project as a whole. Chapter 9 discusses the results and analysis of the Design and Implementation segments. Chapter 10 investigates the project management side to the project as a whole and the issues that were encountered. Chapter 11 involves the author's assessment of the project with Chapter 12 summarising the project and discussing future work.

Transfer Learner aims to merge technical, academic and quantitative evaluations to be a tool that aids effective transfer decisions in football, thereby, directly solving a huge part of the financial problem in football.

Chapter 3

Research

Before work on Transfer Learner could begin it was important to consider the subject areas relating to this project alongside the existing solutions for transfer suggestions in football. This project comprises of machine learning, optimisation and football.

3.1 Machine Learning - ML

Machine Learning is the study of systems and algorithms that learn from data^[10]. The typical process of machine learning is one where a computer program learns from experience E (the data). It learns by considering some tasks T which has a performance metric P . The beauty being that its performance improves with more experience (data).

Machine Learning is described and notable applications are mentioned in the following extract from Tom M. Mitchell's book, Machine Learning:

The field of machine learning is concerned with the question of how to construct computer programs that automatically improve with experience. In recent years many successful machine learning applications have been developed ranging from data-mining programs that learn to detect fraudulent credit card transactions, to information-filtering systems that learn user's reading preferences, to autonomous vehicles that learn to drive on public highways.

3.1.1 Examples of Machine Learning applications

Machine learning has been thriving in many areas. Netflix's success in suggesting what film a user would want to watch is based on recommendation systems^[11], a direct application of ML. In addition, Microsoft's Xbox Kinect, object recognition software is based on Machine Learning as is Facebook's Blind Alternative Text reader^[12], which uses ML to give blind readers the ability to "see" images on facebook through image classification and pattern recognition.

3.1.2 Machine Learning Paradigms

Machine Learning has three main subfields; supervised learning, unsupervised learning and reinforcement learning.

Supervised learning is when the computer is given data that has a target and learns to improve its ability to predict the label of a new datapoint. Unsupervised learning is a Machine Learning paradigm that learns the structure of unlabelled data, for instance, a computer can identify clusters within a set of datapoints. Finally, Reinforcement learning is where the computer learns to improve at a task by statistical reward system.

The task of predicting how good a team, given a set of team specific attributes, is a supervised learning task. The model will be used to extract the important features and their weighting to accurately gauge the value of a team. This will be crucial in evaluating the success of each transfer decision made. Furthermore, these transfer decisions will be subjected to constraints in order to output solution to optimisation problems, this emphasises the need for optimisation.

3.2 Optimisation

Optimisation is one of four main themes in the field of Operational Research. Operational research deals with making decisions using a scientific approach^[13]. Essentially, operational research is concerned with quantitative models and producing optimal solutions. An optimal solution is a feasible solution that is generated by the objective function. An objective function is a function that maximises or minimises the solution to a decision problem, for example, maximising the profit or minimising the loss.

A linear program is an example of an optimisation problem^[14]. The general structure of a linear program can be expressed as the following:

$$\max z = f(x), \text{ with the constraint that } g(x) R b$$

where $g(x)$ is the linear function expressing reality

R is a comparison or equality operator, for example, \leq and \geq

b is the benchmark value (the threshold value)

In an optimisation problem, P is the problem, $f(x)$ is the objective function which is to be maximised or minimised (i.e $\min z = f(x)$) and z is the value of the objective function associated with the current solution x . Z can be maximum profit, minimum cost etc. Z is then optimised by taking into account the constraints, $g(x) R b$. The constraints are the restrictions that have been imposed by the problem itself. These constraints cannot be violated and must be achieved.

This gives a gentle introduction to optimisation and will be further explored in the Optimisation section of the Design and Implementation chapters. Machine learning and optimisation will be applied in this project in a football context. Transfer decisions and the Premier League are the domain of this project, both of which will be broken down in the following section.

3.3 Football and Transfer Decisions

Football is a sport where 2 teams of 11 players compete to score more goals than the other team. Tactically, the positions in football are split into 4 player positions; goalkeepers, defenders, midfielders and forwards. The positional breakdown can be seen in the image below.

3.3.1 (English) Premier League - EPL

The Premier League (formerly known as the Barclays Premier League)^[15] is the football league this paper focuses on. The Premier League is the highest ranked league in the English Football



Figure 3.1: Image of positions in Football: Goalkeeper(s), Defender(s), Midfielder(s) and Forward(s)^[56]

League System^[16]. The EPL consists of 20 football teams^[17] who play each other twice over the course of the season, one game would be at their stadium and the other game would be at the oppositions stadium. This results in a total of 38 games played by each team over the course of the season. In each football game, there are 3 possible scenarios, assuming the game reaches its conclusion in 90 or so minutes. These scenarios are either a win, draw or loss. A win occurs when a team scores more goals than the opposition over the course of the entire game. A loss is the opposition scenario to a win. A draw occurs when both teams have scored the same number of goals at the end of the match. The premier league season typically runs from August to May of the following year. For example, the season 2016/17 starts on the 13th of August 2016 and is scheduled to end on the 21st of May 2017.

3.3.2 Transfer Decisions

A transfer in football refers to the buying and/or selling of players between football teams. These dealings have been done in football for over 100 years. The claim is that the first ever transfer for a professional footballer occurred in 1893. This transfer was Jack Southworth being moved from Blackburn Rovers to Everton for 400 pounds^[18].

In the Premier League and across European Leagues, there are two main occasions on which transfers can occur. These are either at the end of the season or halfway through it. The unofficial term for these two occasions is the transfer window^[19].

The transfers that occur at the end of the season coincide with the Summer months, hence, why it is called the summer transfer window. By football regulations, this period last for at most 12 weeks. These 12 weeks are decided by the national football association for the given country. For example, the Football Association (FA) - the national football association for England - decided in 2015/16 that the transfer window runs from the 9th of June 2016 to the 31st of August 2016. Other countries have different time periods but they tend to coincide with the English league system. This is done by making sure that they all close on the same day. Similarly, there is a 4 week window in January, at the midpoint of the season, for transfers to occur. This is known as the winter transfer window.

It is important to clarify that these transfer decisions can occur between any team. However the data used in this paper is limited to teams within the Premier League. It is also important to mention that the end of the summer transfer window and the whole of the winter transfer window both occur whilst the league season running.

Transfer decisions require choosing a player to buy or sell from a list of possible players from other teams. Legally, all 20 teams in the Premier League are limited to 25 players^[20] roster. These players include all 4 player position types.

The sheer volume of possible transfer decisions can lead to bad transfer decisions if there was no sound logic involved in the exchange. For the purpose of this paper a bad transfer decision is one where the addition of a player(s) into a team's roster does not directly improve the value of the team (in a way that would be expected given the price of the player(s)).

3.3.3 Example of a “bad” transfer decision

Fernando Torres at Liverpool was referred to as one of the best strikers in the World. He scored 65 goals in 102 matches played for Liverpool in the Premier League^[21]. However, in the winter transfer window, he was bought by Chelsea for a huge fee of £50 million. To add

context to this, Chelsea had a good strike force before Torres was introduced ut the appeal of such a "great" striker, caused Chelsea to get Torres with any means necessary. Unfortunately, the transfer decision was not successful for Chelsea, as seen in the 110 games play where only 20 goals were scored by Torres. This transfer decision shows how difficult it is to make good transfer decisions based on the naive knowledge fuelled by the success of the player in other teams.

3.4 Existing Solutions and Literature Review

Although there are no existing programs that provide an answer to the problem described above, there are applications that claim to have solutions that involve data collection and exploratory analysis of the data.

The following is a list of solutions that help certain decision making processes in football; Scout7, DataScout and Prozone.

Scout7 helps 138 clubs in 30 leagues around the world by summarising each football match into 30 to 45 minute detailed video analyses^[22]. In addition, Scout7 provides a football database of 135,000 + players that are currently playing football. This provides a generic but effective tool that gives football clubs more information to make strategic decisions on and off the football pitch. Transfer Learner uses a similar method of data analysis to make transfer decisions using measurable metrics.

DataScout^[23], like Scout7, also provides data analysis. It provides a comparative analysis of players using over 400 player statistics across 10 or so years of their careers. The analysis is based on a detailed study of over 30,000 matches. This allows users of the software to search, compare and explore players' abilities. This comparative analysis is similar to what Transfer Learner aims to provide when it comes to making transfer decisions. In Transfer Learner the comparative analysis will be used to determine transfer decisions. Transfer decisions will be swaps, where a player is bought from a club and exchanged with another player with or without an added financial cost.

Prozone similarly tracks and delivers innovative performance analysis of all sports^[24], rather than solely football. This makes it a successful generalising tool but like the previous solutions it does not focus on generating optimal transfer decisions for a football team. This is what Transfer Learner aims to do. The foundation is based on these existing solutions, having quality data and the possibility of data analysis to generate optimal transfer decisions.

Given that prior research shows that no software is currently available to solve the transfer decision problem in a football context, this solution could bring benefits to football teams

when it comes to optimising the money spent during transfers . The challenge of being able to accurately model players and generate insights has vast possibilities, for example this project could be further developed personally or as a contribution towards research involving machine learning and sport in general.

Finally, this paper is primarily built upon the paper “Machine Learning for Soccer Analytics” by Gunjan Kumar^[25]. This paper used football match data to accomplish four goals. One of the goals was to identify the important attributes that best represent a football player’s ability. The paper utilised an OPTA^[26] dataset for the 2011/12 Premier League season. The goal was achieved by splitting the dataset into four player groups and finding the optimal set of performance attribute for each position. Transfer Learner parallels this paper with respect to it’s goal: to identify the important attributes that determine a team’s performance.

The solution in this paper describes a process of splitting the original player dataset into four subsets: goalkeepers, defenders, midfielders and forwards. Transfer Learner builds upon this by using a machine learning model to select the important features which will be used to predict the value of a team. Furthermore, these important attributes are then used to optimise transfer decisions based on player positions. The output of this project is a solution to an optimisation problem. This optimisation problem involves collating a list of transfer swaps a team should make with regards to their goalkeeper, defender, midfielder and forwards.

To summarise, the main difference between “Machine Learning for Soccer Analytics” and Transfer Learner stems in the goal of the projects. "Machine Learning for Soccer Analytics" was trying to predict the output of football matches using match and player statistics for only one season. Transfer Learner not only tries to predict the quality of a football team using team and player statistics from a variety of football seasons (limited to the Premier League), it also has the additional goal of generating optimal transfer decisions for football teams present in the Premier League on a season by season basis.

Chapter 4

Legal, Social, Ethical, and Professional Issues

With software products, it is important to make sure that the products follow guidelines that ensure it can be trusted by users of the product. This section of the report will discuss the possible issues that could arise during the development of Transfer Learner. The possible issues are broken down into the four segments: legal, social, ethical and professional issues. These issues and their detailed breakdown were inspired by the British Computing Society Code of Practise^[27]. The issues discussed are not exhaustive but depict possible pitfalls that could occur in the project.

4.1 Legal Issues

Due to the nature of the system, Transfer Learner will be processing and displaying data from external sources. The concern here relates to the theme of intellectual property. Intellectual property will be preserved in a number of ways. The main concern is the data from the Premier League’s official website^[28]. This website emphasises that if data is extracted, it cannot be made publically available. As a result, any team and player information extracted from the Premier League has been stored locally. This ensures it is in accordance with the “private and personal use”^[29] clause. In addition, images like the Premier League logo used in the web interface have

been referenced. The use of official APIs make sure that the method of processing will be in line with what the terms of accessibility defined by the provider. Legal issues are an important topic in all products, and have been carefully considered. This ensures that the Intellectual Property is used responsibly.

4.2 Social Issues

Social issues are issues that might make the user feel victimised in one way or the other. Transfer Learner will thrive for inclusivity, ensuring that cultural, social, gender and disability themes with regards to the information presented is taken into account. The project is of a descriptive nature as it offers transfer suggestions. Social issues tend to relate to projects where user interaction is being stored. The only user interaction in Transfer Learner is the choice of a football team in a given season and this interaction is not being stored. This means that cultural, social and gender issues do not become a problem here. In terms of disability, future iterations of the project will ensure that the system takes into account colour blind and disabled users, by providing multiple colour schemes and having detailed enough descriptions in the alt tags relating to images and the web interface functionalities.

4.3 Ethical Issues

The key ethical concerns with web interfaces is data privacy. However, similar to the social issues considered, no user data is being stored, eradicating the issue of data privacy. In addition, the nature of the data present is not of an explicit nature, as the data available relates to player or team names. This is an important consideration as data presented should not be of an uncensored type.

4.4 Professional Issues

Potential users of this project could be **professional** football clubs. It is therefore important to create a sense of worth for the platform. As previously mentioned, all the data provided

is offered as a service with no user information being stored. The developer will abide by the aforementioned BCS code of conduct by showing professional competence and integrity and duty to relevant authority^[27].

Regarding development practises, as this is an interdisciplinary project, a professional approach will be taken ensuring that code is well documented and follows the design pattern of each specific field.

Following the background research, the objectives are then defined. To do so, the author constructed a set of objectives which highlights all the functionalities (core and additional) this project aims to meet.

Chapter 5

Objectives

The goals of the project were broken down into 14 objectives. These objectives are either categorised as core or additional objectives, and their dependencies. The objectives are also presented in the project specification^[30] and progress report^[31] with slight modifications.

1. Gain permission from PremierLeague.com to scrape data from their website (This objective was achieved given the terms and conditions of the website) (core - independent)
2. Extract the dataset of Premier League (EPL) team statistics since the 1992/93 season (core - dependent on objective 1)
3. Extract dataset of EPL teams' seasonal prize-money/value (core - independent)
4. Extract dataset of EPL player statistics since the 1992/93 season (core -dependent on objective 1)
5. Extract dataset of Premier League player values using Fantasy Premier League costs (core - independent)
6. Cleanse and pre-process data (core - dependent on objectives 2, 3, 4 and 5)
7. Merge team specific dataset (core - dependent on objectives 2 and 3)
8. Merge player specific dataset (core - dependent on objectives 4 and 5)

9. Build and evaluate regression-based supervised machine learning models using datasets (core - dependent on objectives 7 and 8)
10. Validate model and choose best model based on validation scores (core - dependent on objective 9)
11. Refine model (core - dependent on objective 10)
12. Perform the transfer swaps (core - dependent on objective 11)
13. Optimise transfer swaps (core - dependent on objective 12)
14. Create a web interface for user interaction with Transfer Learner (additional - dependant on objectives 11 and 13)

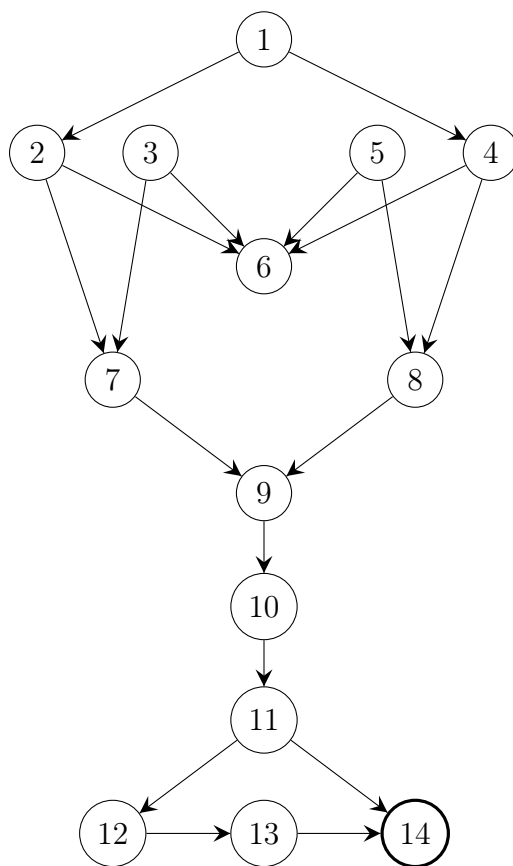


Figure 5.1: Dependencies between objectives

Chapter 6

Technical Constraints

Name	Warwick University DCS Computer
Operating System	Linux - RedHat
Processor	Quad Core
Memory	16GB
Disk Space	50GB + 1TB External Hard Drive
Graphics Card	Nvidia GTX-750

Table 6.1: System Specifications

Web interfaces and software programs with high computational activities like Machine Learning, Data Analysis and iterative searches involving a big search space, are restricted by the hardware it is deployed on. Due to the lack of ownership of a personal computer, the primary computer used was the machines provided by the Department of Computer Science (DCS). Table 6.1 shows the hardware specification of the deployed environment. The machine learning and optimisation part of the program will place a huge strain computationally, so good code health^[27] needs will be ensured.

Following a clear definition of the objectives and a thorough background research, design of the program and web interface can occur. The objectives and preliminary research form the foundation that ensures the work is focused with well-defined goals.

Chapter 7

Design

Transfer Learner proposes an architecture to the financial problems involved in football transfer decisions. This section will describe the design of the project and its workflow. Transfer Learner begins by first extracting data, cleaning that data and integrating the cleaned data. Following this will be the machine learning section, whose main purpose is to assess and predict how the value of a team (which measures how good a team as). Subsequently, the Optimisation segment is used to output optimal transfer decisions to the optimisation problems. Finally, the design of the web interface serves as a demonstration of the capabilities of Transfer Learner to the user.

7.1 Transfer Learner's Architecture

The breakdown of Transfer Learner's architecture can be seen in Figures 7.1 and 7.2. Figure 7.1 provides a depiction of the interaction in Transfer Learner between the data gathering, data processing, machine learning, optimisation stages and the web interface components. Figure 7.2 gives a high level overview between technologies used at each stage of the project. The 6 stages of the project are: data extraction, data cleaning, data integration, machine learning, optimisation and the demonstration. Data extraction will be responsible for retrieving the data needed for this problem domain. Data cleaning is responsible for correcting any inconsistencies present in the extracted data. Data cleaning also includes the exploratory data analysis in order

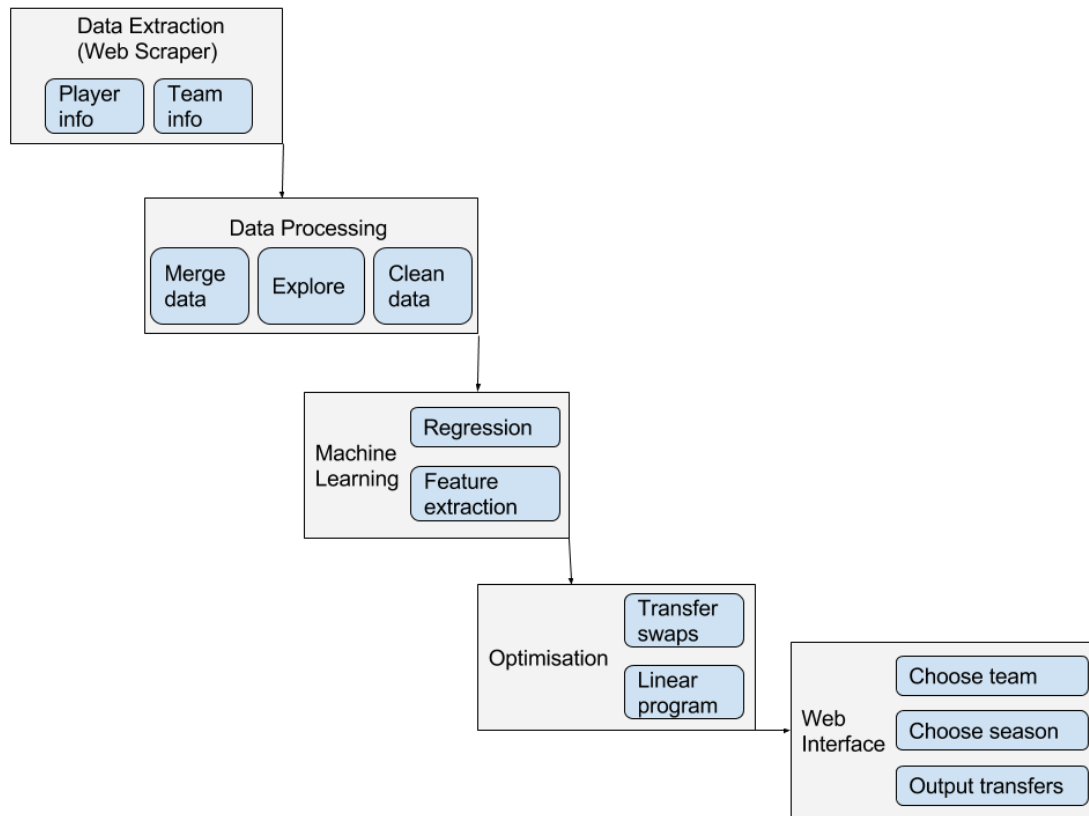


Figure 7.1: Transfer Learner's Architecture

to investigate the statistical characteristics of the datasets. Data integration merges the cleaned datasets used in the project. Machine learning deals with selecting the “best” model that does well on the integrated data and will also be used to reduce the dimension of the integrated datasets. The dimension of a dataset is the number of attributes the dataset contains. This process in the machine learning section can be summarised as attribute selection. Optimisation deals with exchanging players based on the attributes selected and optimising these exchanges under certain constraints, resulting in the optimal transfer decisions. These optimal transfer decisions are then presented to the user via a web interface.

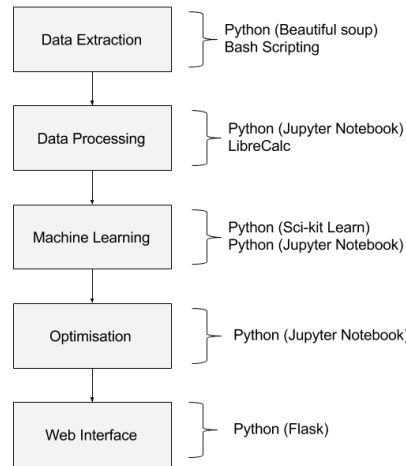


Figure 7.2: Transfer Learner's High Level Overview

7.2 Data Extraction

Data needs to be relevant to the domain (problem area) it is linked to. The data being extracted must be carefully considered. This will ensure that the system, Transfer Learner, has the appropriate data to solve the required problem. This data serves as the information required to do the machine learning and optimisation asks.

These four channels of data needed for this project are:

1. Player seasonal statistics
2. Team seasonal statistics
3. Player seasonal valuations
4. Team seasonal valuations

For any data science workflow, the selection of the data source is important. With the huge volume of data available on the web, there are considerations to be made. For example, data should be rich enough to allow investigation to occur. Very small dataset(s) may not contain

enough information needed for the problem and too large a dataset may be difficult to work with, when one takes into account the technical limitations of the computer in use, as seen in the Technical Constraints section. In addition, the dataset should have many attributes of different types, encouraging variation in the dataset. For example, name, age and sex are a list of attributes that do not infer much and therefore would not yield interesting results. The ideal goal is to allow a combination of different datasets and therefore a combination of different kinds of attributes. Fundamentally, the data sources must be related in some way. For example, in transfer learner, the player and team statistics will come from the same data source.

Finally, when selecting data source or sources, a number of factors need to be considered:

Is the content provider popular?

Is there an existing flexible API that provides information needed?

Will the content provider's inclusion generate popularity?

7.2.1 Player and Team Seasonal Statistics

Initial Data Source - WhoScored

Given, these questions and considerations, the choice for the source of team and player statistics was initially WhoScored^[32]. From prior investigation, it was clear that WhoScored, had the most publicly visible team and player information from available. WhoScored covered 500 tournaments/leagues across the world, 15,000 teams and 250,000 players^[33]. In addition, SkySports, the dominant subscription television sports brand in the United Kingdom and Ireland^[34], frequently linked to WhoScored during their football analysis segments. Finally, the fact that WhoScored was constantly updated added an extra appeal as one could do further investigation in the future and compare results.

Before delving deeper into WhoScored, it must be mentioned that there was no API (Application Programming Interface) that offered the solutions needed with no added financial cost. The only option was OPTA who although popular, provided datasets at huge costs, which was not feasible for a student. As a result, it became clear that the only solution was to implement a web scraper that generates raw data from websites.

Champions League Final Stage Player Statistics






Summary												
Defensive												
Offensive												
Passing												
Detailed												
View: Overall Home Away						Filter: Minimum apps All players						
R	Player	Apps	Mins	Goals	Assists	Yel	Red	SpG	PS%	AerialsWon	MotM	Rating
1	 Neymar Barcelona, 25, AM(CLR),FW	9	797	4	8	5	-	2.2	74.8	0.6	3	8.24
2	 Lionel Messi Barcelona, 29, AM(CR),FW	9	810	11	2	-	-	4.2	81.4	0.1	2	8.24
3	 Cristiano Ronaldo Real Madrid, 32, M(L),FW	10	930	7	5	1	-	5.6	86.5	1.5	2	7.99
4	 José Giménez Atletico Madrid, 22, D(C)	5	450	-	-	2	-	0.6	78.7	5	1	7.99
5	 Marco Reus Borussia Dortmund, 27, M(CLR),FW	3(1)	291	4	1	-	-	3	77	0.3	1	7.83

Figure 7.3: Example of WhoScored data^[54]

Web scraper

A web scraper is a technical tool that is employed to websites with large amounts of data. The data is then extracted and saved to a local file on the computer in preferably csv format^[35]. A csv file, which is an abbreviation to comma separated values, represents the information (rows, columns and data entries) in the datafile but separates them by commas. Typically, a web scraper takes a long time to be adapted to the website the information is needed from. For example, the whole journey of web scraping for transfer learner, as a student with no prior experience, took approximately 10 weeks, which is probably a month or so longer than it would have been for someone with prior experience.

Coming back to WhoScored, during the extraction process, it became clear that a web scraper was not allowed. This was due to the website having deep security measures that prevented any web scraper from accessing their website. Although unsuccessful with regards to WhoScored, the time taken (around 6 weeks or so) and the skill learnt from implementing a web scraper (a technical challenge in itself) became useful in the final data extraction.

Contingency data source - English Premier League

After the unsuccessful attempt with WhoScored, the next option chosen was the English Premier League website. This website is the official website of the English Premier League mentioned in the EPL section of the Research chapter. The website contains team and player statistics from the Premier League, alone, since the 1992/93 season, when the Premier League started.

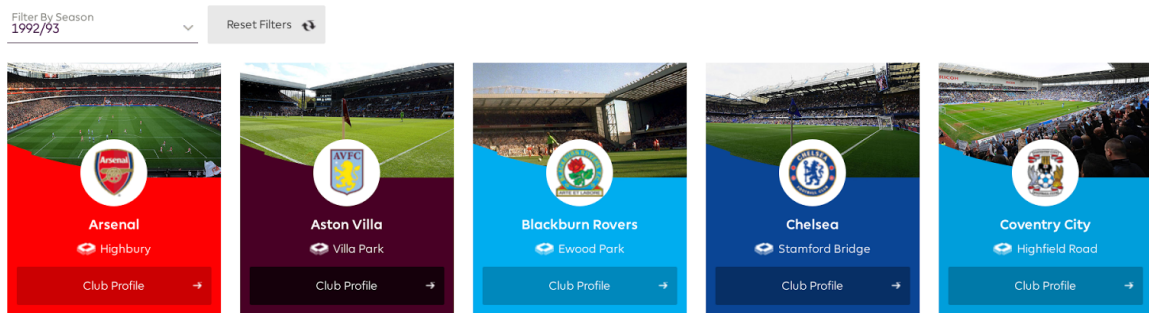


Figure 7.4: Image of Premier League website showing 1992/93 season option for teams^[43]

Although, the EPL website does not have statistics from other World Football Leagues, it has a well confined body of statistics ranging over 25 seasons in the Premier League. Transfer Learner extracted player and team information from the Premier League website using 2 web scrapers (see Implementation - Extraction)ctively.

The choice of the player and team valuation are also important considerations, however, they will come from different sources as these pieces of information are very subjective and there is no source where both player and team values are widely accepted.

7.2.2 Team Seasonal Valuations

Given that the teams present in this project are seasonal Premier League teams it is important to have valuations that link to their Premier League status on a season by season basis. Initially, the choice was to use their premier league points total at the end of a season. The points

generated for each team relate to the total points received from their performance in each match over the course of the season.

Position	Club	Played	Won	Drawn	Lost	GF	GA	GD	Points	Form
1 ●	 Leicester City	38	23	12	3	68	36	32	81	D W D W D
2 ▲	 Arsenal	38	20	11	7	65	36	29	71	W D W D W
3 ▼	 Tottenham Hotspur	38	19	13	6	69	35	34	70	W D D L L
4 ●	 Manchester City	38	19	9	10	71	41	30	66	D W L D D

Figure 7.5: Image of premier league points total for the 2015/16 Premier League Season^[55]

This valuation may seem indicative of the success of teams in a given season, however, football has other considerations and the designated value for the purpose of this project was financial as the end goal/reward was intended to be monetary. In addition, the author made sure that the financial valuation of a team on a seasonal basis relates to the team's seasonal achievements as well. This led to the choice of using the Official Premier League prize money reward (seasonally) as the team valuation. This reward system takes into account three factors^[36]:

- **Equal tv rights share for each team** - a proportion of money equally distributed to all Premier League teams
- **Merit money** - the amount each team gets due to their final league position
- **Facility fee** - based on how frequently each team is shown live on tv in the United Kingdom

An example of the reward system can be seen below.

The Official Prize money rewards of the Premier League (aka team valuation) extraction had a complication. No website existed that contained information about all the seasons and the prize money awarded by teams. Some information was stored as images in PDFs, other's were images in certain websites. As a result, these values were manually recorded from the seasons available. The seasons available were from 2006/07. The sources where the information was retrieved from are:

NO#	CLUB	EQUAL SHARE	MERIT BASED	FACILITY FEE	TOTAL
1	Leicester City	£55.5m	£24.7m	£12.8m (15 matches)	£93m
2	Arsenal	£55.5m	£23.5m	£21.8m (27 matches)	£100.8m
3	Tottenham	£55.5m	£22.2m	£17.3m (21 matches)	£95m
4	Man City	£55.5m	£21m	£20.3m (25 matches)	£96.8m

Figure 7.6: Image of premier league prize money for best 4 teams in 2015/16^[36]

- Premier League official Annual Report^[37]
- Sporting Intelligence^[38]
- The Telegraph^[39]
- Sports Lens^[40]
- TotalSportek^[36]
- Daily Mail^[42]

7.2.3 Player seasonal valuations

As indicated previously, valuing a player is difficult to do. Prior research did not show any measurable and publically available real life player valuation. As a result, the Fantasy Premier League Player costs were used. This is widely utilised by sports fans across the World as it gives a normalised way of comparing players in a season based on their teams merits from previous seasons ^[44]. Regarding the extraction of player valuations, there was no issue in being able to retrieve player costs from Ffocities, a Fantasy Premier League player valuation website, using a web scraper.

7.2.4 Summary of Final choices for data sources

The data extraction process led to the following datasets being obtained (prior to cleaning):

1. Player features from the **English Premier League**^[42]
2. Team features from the **English Premier League**^[43]
3. Player valuations from **Fantasy Football Central**^[44]
4. Team valuations using the **official English Premier League seasonal financial rewards**^[36–41]

7.3 Exploratory Data Analysis and Data Cleaning

Exploratory data analysis (EDA) is a crucial step in the data science workflow. It serves to identify statistical distribution of attributes, analyse the attributes individually and also with respect to other attributes too. This is useful in identifying issues present and thereby correcting them in the data cleaning process. The considerations made and answered by EDA include:

Are certain attributes totally correlated with each other?

Correlation determine how two attributes behave with respect to each other. Correlation is measured through a metric called Pearson product-moment correlation coefficient (PMCC). This is a function of the attributes covariance (measure of the magnitude of the values in each attribute) normalised using the standard deviation of each attribute. This obtains a measure of how strongly correlated two attributes are. The PMCC of two attributes X and Y is calculated in the following way:

$$PMCC(X, Y) = \frac{Covariance(X, Y)}{\sigma(X)\sigma(Y)} \quad (7.1)$$

If two attributes are totally correlated, their PMCC value is 1. This implies the duplication of attributes and can be cleaned by removing one of the two. This helps in reducing the dimension of the dataset (number of attributes present).

Correlation can also be deduced by the scatter plot representation of the 2 attributes against each other.

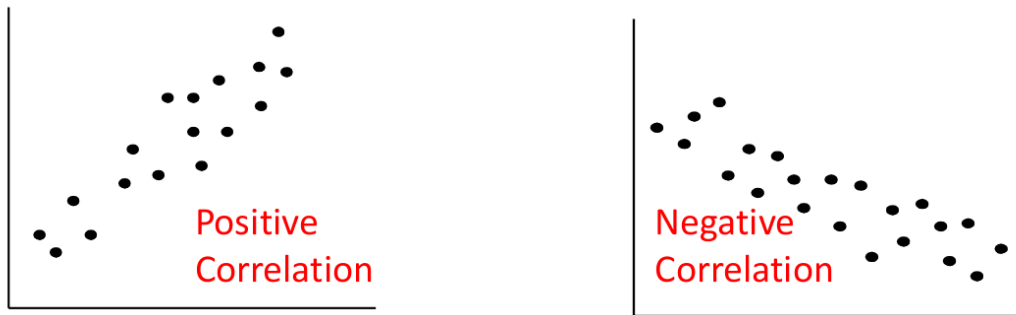


Figure 7.7: Example image of Positive and Negative correlation^[57]

Is the data sparse? Are there missing values?

Missing values are common in data. Data is said to be sparse when missing values exist. Missing values can be due to the data in the dataset not being correctly measured (may be as a result of human error or problems with the tools measuring the data), some data being lost and/or transcribed incorrectly. Handling missing values is crucial. With transfer learner's data cleaning process, the intention is to fill in the missing values manually as much as possible. Another solution would be to drop samples/observation with missing data, but this will serve as a contingency plan. This needs to be considered as the Premier League website may have some player/team profiles existing in the library that have dead links. A dead link refers to a redirection to a web page that has no information and appears blank.

Is the Data Noisy?

Noisy data is another concern in the data cleaning process. Noisy data values are those that are due to error in the measurement. Noise can be due to translation/mismatching errors in the data extraction process. For example, if the web scraper pattern matches a specific player ID that was incorrectly transcribed in the PL website, this could cause incorrect values to be present in the dataset. This is why EDA and data cleaning is vital.

7.3.1 Summary of data cleaning

At the end of the data cleaning process, the matrix below shows how the player features (player name and player attributes) from the English Premier League website will be represented.

$$\left[\begin{array}{c|cccc} player_1 name & player_1 attribute_1 & player_1 attribute_2 & \dots & player_1 attribute_n \\ player_2 name & player_2 attribute_1 & \dots & \dots & player_2 attribute_n \\ \dots & \dots & \dots & \dots & \dots \\ player_m name & \dots & \dots & \dots & player_m attribute_n \end{array} \right] \quad (7.2)$$

Similarly for teams, the matrix below shows how the team features (team name and team attributes) from the English Premier League website will be organised.

$$\left[\begin{array}{c|cccc} team_1name & team_1attribute_1 & team_1attribute_2 & \dots & team_1attribute_n \\ team_2name & team_2attribute_1 & & \dots & team_2attribute_n \\ \dots & \dots & \dots & \dots & \dots \\ team_mname & \dots & \dots & \dots & team_mattribute_n \end{array} \right] \quad (7.3)$$

The player cost/value is an important attribute that will be augmented to the existing player matrix, seen in equation (7.2). The following matrix shows how the cleaned player values from Fantasy Football Central will look.

$$\left[\begin{array}{c|c} player_1name & player_1value \\ player_2name & player_2value \\ \dots & \dots \\ player_mname & player_mvalue \end{array} \right] \quad (7.4)$$

Finally, the following matrix shows how the cleaned team values from the official EPL seasonal financial rewards will look. The team value is what the machine learning models will learn from and aim to predict on unseen team attributes.

$$\left[\begin{array}{c|c} team_1name & team_1value \\ team_2name & team_2value \\ \dots & \dots \\ team_mname & team_mvalue \end{array} \right] \quad (7.5)$$

7.4 Data integration (merging)

The results of the data cleaning stage will be the four data sources (with no inconsistencies):

1. **Cleaned Player features** from the English Premier League
2. **Cleaned Team features** from the English Premier League
3. **Cleaned Player valuations** from Fantasy Football Central
4. **Cleaned Team valuations** using the official English Premier League seasonal financial rewards

Data integration refers to combining data from two sources into one. This is a challenging problem faced by many organisations, for example, if two companies merge they need to combine their datasets. However, the datasets may be inconsistent, for example, the team names may differ across different sources. This can be seen in the case of *Tottenham*, where it is represented as *Tottenham* in one source and *Tottenham Hotspurs* in other sources.

In Transfer Learner, data integration combines the four data streams above into two datasets, one for the team and one for the player information.

The player statistics and the player valuations merge into a single player dataset where the features are the player statistics and the player valuations. This is done by matching player names, their team name in a given season from both datasets to obtain the corresponding player features. This results in the following player dataset (investigated further in Implementation).

$$\begin{aligned} Dataset_{player} &= (X_{player}) \\ X_{player} &= \begin{bmatrix} player_1name & player_1attribute_1 & \dots & player_1attribute_n & player_1value \\ player_2name & player_2attribute_1 & \dots & player_2attribute_n & player_2value \\ \dots & \dots & \dots & \dots & \dots \\ player_mname & \dots & \dots & player_mattribute_n & player_mvalue \end{bmatrix} \end{aligned} \quad (7.6)$$

A similar procedure is performed for the team statistics and team valuations, resulting in the following team dataset.

$$Dataset_{team} = (\mathbf{X}_{team}, \mathbf{t}_{team})$$

$$\mathbf{X}_{team} = \begin{bmatrix} team_1name & team_1attribute_1 & team_1attribute_2 & \dots & team_1attribute_n \\ team_2name & team_2attribute_1 & \dots & \dots & team_2attribute_n \\ \dots & \dots & \dots & \dots & \dots \\ team_mname & \dots & \dots & \dots & team_mattribute_n \end{bmatrix} \quad (7.7)$$

$$\mathbf{t}_{team} = \begin{bmatrix} team_1value \\ team_2value \\ \dots \\ team_mvalue \end{bmatrix} \quad (7.8)$$

It is important to note that from football domain knowledge, certain \mathbf{X}_{team} attributes are linear functions of certain \mathbf{X}_{player} attributes. For example, the total goals scored by a football team in a given season is the sum total of the goals scored by all the players in that given team in that season. This serves as the reason behind the assumption of a linear relationship between player attributes and the value of a team via the team's attributes (accumulation of player attributes). These features will be predicted in the machine learning model.

7.5 Machine Learning models

The goal of this section is to design a model that is good at predicting the value of a team. The models use team and player information from the dating back from the 2006/07 season. The idea is to understand what team and player attributes affect the team valuation. This gives a measurable way to define what makes a good team, thereby, solving the first part of the problem statement: “Assessing the quality of a good football team”.

7.5.1 Supervised Learning - Regression

A supervised learning process is as follows:

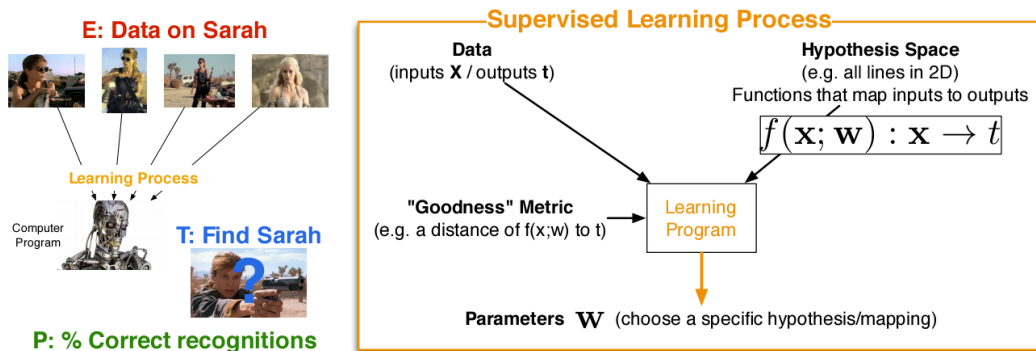


Figure 7.8: Components of a supervised learning system^[10]

Here the learning program, for example linear regression model, is given data X and a possible hypothesis space over w . w is a vector of parameters. This vector is over the real space R^D where D refers to the number of attributes in X . The learning program uses a “goodness” metric over the possible vectors w , where the model finds best w that describes the data.

In Transfer Learner, the target is the team value. The team value being numeric and not categorical means that the supervised learning paradigm that this task falls under is regression as opposed to classification. The prediction from a regression model is a numeric value, in this case, the value of a football team in a given season.

There are some further questions that need to be considered^[10]:

- Does the input data \mathbf{X} need to be encoded?
- What hypothesis space function $f(\mathbf{X}; \mathbf{w})$ should be fitted?
- What should be the performance metric?
- What knowledge from the domain is useful?

To recap, the data presented for this problem is the following:

$$Dataset_{team} = (\mathbf{X}_{team}, \mathbf{t}_{team})$$

$$\mathbf{X}_{team} = \begin{bmatrix} team_1name & team_1attribute_1 & team_1attribute_2 & \dots & team_1attribute_n \\ team_2name & team_2attribute_1 & \dots & \dots & team_2attribute_n \\ \dots & \dots & \dots & \dots & \dots \\ team_mname & \dots & \dots & \dots & team_mattribute_n \end{bmatrix} \quad (7.9)$$

$$\mathbf{t}_{team} = \begin{bmatrix} team_1value \\ team_2value \\ \dots \\ team_mvalue \end{bmatrix} \quad (7.10)$$

$$Dataset_{player} = (X_{player})$$

$$X_{player} = \begin{bmatrix} player_1name & player_1attribute_1 & \dots & player_1attribute_n & player_1value \\ player_2name & player_2attribute_1 & \dots & player_2attribute_n & player_2value \\ \dots & \dots & \dots & \dots & \dots \\ player_mname & \dots & \dots & player_mattribute_n & player_mvalue \end{bmatrix} \quad (7.11)$$

The hypothesis present for \mathbf{X}_{team} and \mathbf{t}_{team} (team value) is that there is a linear relationship between the two. To clarify, \mathbf{X}_{team} is a matrix over the real space $R^{N_{team} \times D_{team}}$. N_{team} is the number of team samples, 200 team samples for transfer learner. Moreover, D_{team} refers to the number of team attributes present, which for Transfer Learner was initially 30 but then was refined to 9 attributes using the machine learning models. Furthermore, \mathbf{t}_{team} is a vector over the real space $R^{N_{team}}$. Finally, as highlighted earlier, a linear relationship exist between \mathbf{X}_{player} and \mathbf{t}_{team} , through \mathbf{X}_{team} .

Note that, \mathbf{X}_{player} is a matrix over the real space $R^{N_{player} \times D_{player}}$, where N_{player} and D_{player} are the number of player attributes and player samples respectively. In Transfer Learner, there were originally 30 possible player attributes and around 5000 player samples. This can be mathematically expressed in the following way:

$$\begin{aligned}\mathbf{t}_{team} &= f(\mathbf{X}_{team} ; \mathbf{w}_{team}) \\ &= \mathbf{w}_{team}[\text{attribute}_0] + \mathbf{X}_{team}[\text{attribute}_1]\mathbf{w}_{team}[\text{attribute}_1] + \mathbf{X}_{team}[\text{attribute}_2]\mathbf{w}_{team}[\text{attribute}_2] \\ &+ \dots\end{aligned}$$

$$\mathbf{X}_{team} = g(\mathbf{X}_{player})$$

$$\mathbf{t}_{team} = f(g(\mathbf{X}_{player}) ; \mathbf{w}_{team})$$

The machine learning regression-based models deduce the best parameters \mathbf{w}_{team} in order to accurately predict \mathbf{t}_{team} from \mathbf{X}_{team} ^[45]. To reiterate, the hypothesis of a regression problem states that there exists a linear relationship between the input and the target.

7.5.2 Overfitting and Underfitting

The goal therefore in regression problems is to find the line that best models the problem by learning from the data made available. When learning from data, one needs to ensure that the model does not overfit or underfit.

Underfitting is when the model does a poor job in generalising the characteristics of a dataset, whereas overfitting does extremely well on the training data, by memorising its details, but

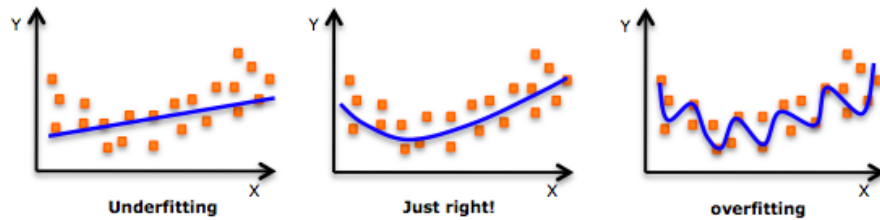


Figure 7.9: Visual description of overfitting and underfitting^[46]

does a poor job generalising to new datasets. As mentioned previously, the goal is to choose the line that best models the regression problem. This is the line that is as close to possible to the datapoints without being subject to overfitting or underfitting.

7.5.3 Standardisation of samples

In general it is highly recommended to standardise data before using machine learning models. Standardisation is useful when the attributes measured have differing range of values. For example, consider 2 attributes; mass in grams and length in metres. The length tends to be a smaller value whereas the mass would be quite large. Standardisation rescales the data so that each attribute has a mean of 0 and a variance of 1. This helps solve problems where different attributes are of varying scales.

Mathematical Descriptions

The mean (μ) of a set of values (X) is the average of the values.

$$\mu(X) = \sum_{i=1}^N \frac{x_i}{N}$$

where N is the number of values and x_i is a value in the set of X .

Standard deviation (SD) measures how dispersed the data values are. A low SD value means the data is not really varied, whereas a high SD implies a varied dataset.

SD (σ) is the square root of the variance.

The variance of a set of values (X) = $\mu(X^2) - (\mu(X))^2$

Algorithm 1 Standardisation algorithm

Input \leftarrow data X

for each $X[\textit{attribute}]$ in X **do**

$X[\textit{attribute}] = X[\textit{attribute}] - \mu(X[\textit{attribute}])$

$X[\textit{attribute}] = X[\textit{attribute}] / \sigma(X[\textit{attribute}])$

end for

Output \leftarrow standardised X

7.5.4 Models

The optimisation problem presented in this report is a linear program. This makes it appropriate to use linear (as well as piecewise linear) models. This means non-linear models like Neural Networks are not considered as they relate to non-linear optimisation problems. As a result, the focus was on OLS, OLS with regularisation and Decision Trees.

Ordinary Least Squares - OLS

The idea in machine learning is to get the right parameter. The ‘goodness’ of a parameter is dependent on the loss function. The loss function measures the difference between the actual label and the predicted label of a specific observation/datapoint/sample. When dealing with loss functions, and due to the data being in vector-matrix form, it is better to express these problems using vector-matrix notation.

Loss function: $L_n(\mathbf{t}_n, \hat{\mathbf{t}}_n)$

where for each observation $(\mathbf{x}_n, \mathbf{t}_n)$:

\mathbf{x}_n - vector of attributes for the observation/datapoint/sample

\mathbf{t}_n - actual label for the observation

$\hat{\mathbf{t}}_n$ - predicted label for the observation

And for each linear model f (i.e. f_{OLS} , $f_{OLSwithRegularization}$, $f_{DecisionTreeRegressor}$)

$L_n(\mathbf{t}_n, \hat{\mathbf{t}}_n) = L_n(\mathbf{t}_n, f(\mathbf{x}_n ; \mathbf{w}))$

Each model generates a vector of weights (\mathbf{w}) such that:

$\mathbf{x}_n \mathbf{w} = \hat{\mathbf{t}}_n$ subject to $\hat{\mathbf{t}}_n \approx \mathbf{t}_n$ without overfitting

In OLS the loss function is the squared error loss function.

$$L_n(\mathbf{t}_n, f_{OLS}(\mathbf{x}_n ; \mathbf{w})) = (\mathbf{t}_n - \hat{\mathbf{t}}_n)^2$$

However, L_n is for a single observation. In order to get the total loss, L , for all the inputs, one calculates the mean of the loss function for all samples.

$$L = \frac{1}{N} \sum_{n=1}^N L_n$$

The OLS solution for \mathbf{w} comes from minimising the loss function, L .

$$\begin{aligned} \mathbf{w}_{OLS} &= \min_w L = \min_w \frac{1}{N} \sum_{n=1}^N L_n(\mathbf{t}_n, f_{OLS}(\mathbf{x}_n ; \mathbf{w})) \\ &= \min_w \frac{1}{N} \sum_{n=1}^N (\mathbf{t}_n - \hat{\mathbf{t}}_n)^2 = \min_w \frac{1}{N} \sum_{n=1}^N (\mathbf{t}_n - \mathbf{x}_n \mathbf{w})^2 \end{aligned}$$

The choice of the loss squared error function is because it is convex and smooth. This ensures that there is a single global minima, implying a specific unique “best” solution to the problem.

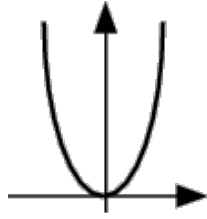


Figure 7.10: Image of a (convex and smooth) squared error loss function^[58]

OLS for Transfer Learner The aim is the find the solution that minimises the loss function with respect to \mathbf{t}_{team} . This would come in the form of parameters, $\widehat{\mathbf{w}_{OLS}}$

$$\begin{aligned} \mathbf{t}_{team} &= f(\mathbf{X}_{team} ; \widehat{\mathbf{w}_{OLS}}) = \widehat{\mathbf{w}_{OLS}}[attribute_0] + \mathbf{X}_{team}[attribute_1] \widehat{\mathbf{w}_{OLS}}[attribute_1] + \mathbf{X}_{team}[attribute_2] \widehat{\mathbf{w}_{OLS}}[attribute_2] \\ &+ \dots \end{aligned}$$

OLS, in the transfer learner setting, learns the values $\widehat{\mathbf{w}_{OLS}}$ using the training data such that:

$$\mathbf{t}_{team} = f(\mathbf{X}_{team} ; \widehat{\mathbf{w}_{OLS}})$$

In order to do prediction on a new datapoint, $\mathbf{X}_{newteam}$ replace \mathbf{X}_{team} with the new datapoint so that:

$$\mathbf{t}_{newteam} = f(\mathbf{X}_{newteam} ; \widehat{\mathbf{w}_{OLS}})$$

There is a slight difference with the OLS solution to above. The difference is that \mathbf{X}_{team} (training, testing or a new one) is transformed by including a column of 1s to the left of \mathbf{X}_{team}

$$\mathbf{X}_{team} = \begin{bmatrix} team_1name & team_1attribute_1 & team_1attribute_2 & \dots & team_1attribute_n \\ team_2name & team_2attribute_1 & \dots & \dots & team_2attribute_n \\ \dots & \dots & \dots & \dots & \dots \\ team_mname & \dots & \dots & \dots & team_mattribute_n \end{bmatrix} \quad (7.12)$$

becomes

$$\mathbf{X}_{team} = \begin{bmatrix} 1 & team_1name & team_1attribute_1 & team_1attribute_2 & \dots & team_1attribute_n \\ 1 & team_2name & team_2attribute_1 & \dots & \dots & team_2attribute_n \\ 1 & \dots & \dots & \dots & \dots & \dots \\ 1 & team_mname & \dots & \dots & \dots & team_mattribute_n \end{bmatrix} \quad (7.13)$$

Similarly, like highlighted above $f(\mathbf{x}_n ; \mathbf{w}) = \mathbf{x}_n \mathbf{w}$. So for all observation the linear model becomes: $\mathbf{X}\mathbf{w}$

In addition, the squared error loss function becomes $L = \frac{1}{N}(\mathbf{t} - \mathbf{X}\mathbf{w})^T(\mathbf{t} - \mathbf{X}\mathbf{w})$

Note that: $(\mathbf{t} - \mathbf{X}\mathbf{w})^T$ implies a transpose. The effect of a transpose on $(\mathbf{t} - \mathbf{X}\mathbf{w})$, is that the columns of $(\mathbf{t} - \mathbf{X}\mathbf{w})$ become the rows in $(\mathbf{t} - \mathbf{X}\mathbf{w})^T$.

To minimise L , it is required to differentiate L with respect to the parameters w and set it to 0.

The following lines describe the steps to derive the minimiser:

$$\frac{dL}{dw} = \frac{d}{dw}(L) = \frac{d}{dw}\left(\frac{1}{N}(\mathbf{t} - \mathbf{X}\mathbf{w})^T(\mathbf{t} - \mathbf{X}\mathbf{w})\right)$$

$$= \frac{d}{dw}\left(\frac{1}{N}(\mathbf{t}^T - \mathbf{w}^T \mathbf{X}^T)(\mathbf{t} - \mathbf{X}\mathbf{w})\right)$$

This implies that:

$$\frac{d}{dw} \left(\frac{1}{N} (\mathbf{t}^T \mathbf{t} - \mathbf{w}^T \mathbf{X}^T \mathbf{t} - \mathbf{t}^T \mathbf{t} \mathbf{X} \mathbf{w} + \mathbf{w}^T \mathbf{X}^T \mathbf{t} \mathbf{X} \mathbf{w}) \right) = 0$$

Resulting in the following by the rules of matrix differentiation

$$-2\mathbf{X}^T \mathbf{t} + 2\mathbf{X}^T \mathbf{X} \mathbf{w} = 0$$

Therefore $\widehat{\mathbf{w}}_{OLS}$ the solution to the problem (by making \mathbf{w} the subject of the formula) implies:

$$\widehat{\mathbf{w}}_{OLS} = (\mathbf{X}^T \mathbf{X})^{-1} (\mathbf{X}^T \mathbf{t})$$

Which for Transfer Learner means: $\widehat{\mathbf{w}}_{OLS} = (\mathbf{X}_{team}^T \mathbf{X}_{team})^{-1} (\mathbf{X}_{team}^T \mathbf{t}_{team})$

Algorithm 2 OLS algorithm

To train OLS model:(investigated further in Implementation)

Input $\mathbf{X}_{team}, \mathbf{t}_{team}$

Pre-process \mathbf{X}_{team} by Standardisation

Compute OLS parameters: $\widehat{\mathbf{w}}_{OLS} = (\mathbf{X}_{team}^T \mathbf{X}_{team})^{-1} (\mathbf{X}_{team}^T \mathbf{t}_{team})$

To predict:

Input new observation $\mathbf{X}_{teamnew}$

Pre-process $\mathbf{X}_{teamnew}$ with same standardisation as training

Compute prediction of the value of $\mathbf{X}_{teamnew}$:

$$\mathbf{t}_{teamnew} = \mathbf{X}_{teamnew} \widehat{\mathbf{w}}_{OLS}$$

Computational complexity: $O(D^2 N)$ ^[47]

D^2 comes from the matrix multiplication of $(\mathbf{X}_{team}^T \mathbf{X}_{team})^{-1}$ which considers the D dimensions of \mathbf{X}_{team} twice.

N comes from the consideration of all observations in \mathbf{X}_{team}

OLS with regularisation

Regularisation controls how complex the linear model (like Ordinary Least Squares) is by restricting the magnitude of its parameters. Intuitively, this can be thought of as a competition between the parameters to obtain the best combination of parameter values.

There are two regularisation techniques; ridge regression and lasso regression.

Ridge Regression for Transfer Learner Ridge regression's competition with regards to the parameters, involve keeping the parameters of \mathbf{w} as low as possible:

$$Ridge_{regulariser} = \sum_{i=1}^N (w_i^2) = \mathbf{w}^T \mathbf{w} \leq l$$

Where N is the number of entries in the vector \mathbf{w} , $\mathbf{w}^T \mathbf{w}$ is in vector matrix form and l is a specified limit.

This updates the loss function seen in OLS in the following way:

$$L_{ridge} = L + Ridge_{regulariser} = L + \lambda \mathbf{w}^T \mathbf{w}$$

λ indicates how strong the regularisation is. λ equal to 0 (no regularisation) takes ridge regression back to OLS regression.

In a similar way to the derivation above for OLS, minimising the loss function for ridge regression leads to the optimal parameter value of \mathbf{w}_{ridge} :

$$\widehat{\mathbf{w}_{ridge}} = (\mathbf{X}^T \mathbf{X} + N\lambda I)^{-1} (\mathbf{X}^T \mathbf{t})$$

Which for Transfer Learner means: $\widehat{\mathbf{w}_{ridge}} = (\mathbf{X}_{team}^T \mathbf{X}_{team} + N\lambda I)^{-1} (\mathbf{X}_{team}^T \mathbf{t}_{team})$

Algorithm 3 Ridge Regression algorithm

To train Ridge model:

Input $\mathbf{X}_{team}, \mathbf{t}_{team}$

Pre-process \mathbf{X}_{team} by Standardisation

Compute Ridge parameters: $\widehat{\mathbf{w}}_{Ridge} = (\mathbf{X}_{team}^T \mathbf{X}_{team} + N\lambda I)^{-1}(\mathbf{X}_{team}^T \mathbf{t}_{team})$

To predict:

Input new observation $\mathbf{X}_{teamnew}$

Pre-process $\mathbf{X}_{teamnew}$ with same standardisation as training

Compute prediction of the value of $\mathbf{X}_{teamnew}$:

$$\mathbf{t}_{teamnew} = \mathbf{X}_{teamnew} \widehat{\mathbf{w}}_{Ridge}$$

Computational complexity: $O(D^2 N)$ ^[47]

D^2 comes from the matrix multiplication of $(\mathbf{X}_{team}^T \mathbf{X}_{team})^{-1}$ which considers the D dimensions of \mathbf{X}_{team} twice.

N comes from the consideration of all observations in \mathbf{X}_{team}

Lasso Regression for Transfer Learner Lasso Regression is a similar regularisation technique to Ridge regression but with a different regulariser and therefore uses a different loss function.

$$Lasso_{regulariser} = \sum_{i=1}^N (|w_i|) \leq l$$

This implies that:

$$L_{lasso} = L + Lasso_{regulariser} = L + \lambda \sum_{i=1}^N \sum_{i=1}^N (|w_i|)$$

However, unlike Ridge and OLS, there is no closed form solution for $\widehat{\mathbf{w}}_{lasso}$, but there are software libraries that have approximate solutions for lasso by implementing a quadratic program solver to minimise the lasso loss function (L_{lasso}). For now, the solution will simply be referred to as $\widehat{\mathbf{w}}_{lasso}$. As a result of no proper analytical solution, the algorithm for Lasso regression will be more general.

Algorithm 4 Lasso Regression algorithm

To train Lasso model:

Input $\mathbf{X}_{team}, \mathbf{t}_{team}$

Pre-process \mathbf{X}_{team} by Standardisation

Lasso parameters: $\widehat{\mathbf{w}}_{Lasso}$

To predict:

Input new observation $\mathbf{X}_{teamnew}$

Pre-process $\mathbf{X}_{teamnew}$ with same standardisation as training

Compute prediction of the value of $\mathbf{X}_{teamnew}$:

$$\mathbf{t}_{teamnew} = \mathbf{X}_{teamnew} \widehat{\mathbf{w}}_{Lasso}$$

Decision Tree - DT

The fourth model that will be used in solving the Transfer Learner regression problem is the decision tree model. Decision tree builds regression models in the form of tree with decision nodes and leaf nodes^[48]. It does so by splitting the training data through the decision tree.

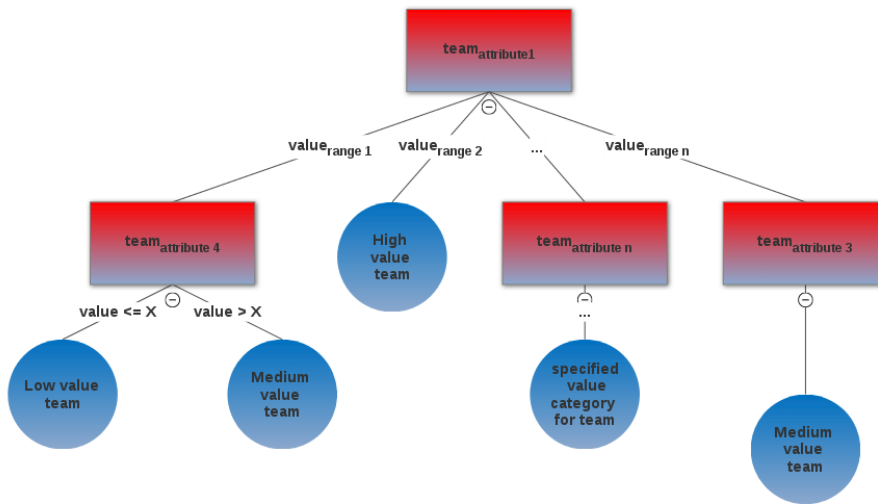


Figure 7.11: Example of a possible Decision Tree constructed by model

A decision node (e.g **team_{attribute1}**) has 1 or more branches (**value_{range1}**, **value_{range2}**, etc.) each representing the values of the attribute being tested. The leaf nodes represent the possible values of any one team, in Transfer Learner. The decision node at the top of the tree represents the best attribute in predicting the value of the team. This decision node is called the root node. Generally, the better attributes are higher up in the tree. The branches at each node represent the choices on how to further split the training data. In a regression problem, the splitting criteria is done by Standard Deviation Reduction.

Standard deviation reduction

Standard deviation reduction^[49] works by choosing attributes that result in the highest standard deviation reduction when split on specific attribute values. If the standard deviation, when split

on attributes, is greater than 0, more splitting is required. If the value is equal to 0 it will be a leaf node.

Algorithm For Decision Tree Regressor in Transfer Learner

The main algorithm behind building decision trees is ID3 by J.R. Quinlan. This algorithm works by going from the top of the tree (the root) downwards choosing attributes that have the highest standard deviation reduction at each stage. The Decision Tree Regressor algorithm for Transfer Learner can be seen in Algorithm 5.

A decision tree overfits when it grows deeper, becoming more complex. As a result the tree becomes prone to fitting the noise present in the data more so than the actual trend, thereby overfitting.

There are 2 approaches in controlling overfitting in decision trees:

1. Stop growing the tree earlier before it starts fitting the noise present in the training data
2. Allow the tree to completely grow (overfitting the training data) and then post-prune it (trim it off by removing parts that are more complex / simplifying the decision tree)

Transfer Learner for each model when implemented will take into account the possibilities of overfitting and act accordingly. In addition, Transfer Learner makes use of a tool, optimised by a community of experts, to perform data analysis. This validates the reliability of the code. Finally, the reason for the detailed algorithmic breakdown of each model demonstrates the author's understanding of the inner workings of each model.

Algorithm 5 Decision Tree Regression model

To train Decision Tree Regressor:

Input \mathbf{X}_{team} , \mathbf{t}_{team}

Calculate the Standard Deviation Reduction for $\mathbf{X}_{team}[attribute]$ in \mathbf{X}_{team}

Choose attribute $\mathbf{X}_{team}[attribute_{best}]$ with the highest standard deviation reduction as the root node

// Expand the tree

Loop: Until all observations have been considered

Extend branches of $\mathbf{X}_{team}[attribute_{best}]$

Eliminate $\mathbf{X}_{team}[attribute_{best}]$ from list of available $\mathbf{X}_{team}[attribute]$

for each available $\mathbf{X}_{team}[attribute]$ **in** \mathbf{X}_{team} **do**

 Calculate the Standard Deviation Reduction

end for

Choose next node based on highest standard deviation reduction

Return tree

To predict:

Input new observations $\mathbf{X}_{teamnew}$

for each observation in $\mathbf{X}_{teamnew}$ **do**

 Simulate observation through decision tree

end for

Output: results of simulations

7.5.5 Choosing the Final Model for Transfer Learner

Occam's razor states that if two explanations (models in Transfer Learner) are good at doing a task, the simpler one is usually better^[50].

As a result, when comparing between models implemented, choosing a simple but effective model is at the forefront of the decision making process. To make the models simpler, regularisation techniques are used for OLS and the decision tree will be pruned effectively.

Coefficient of determination - R^2

To determine the success of a regression model, the coefficient of determination (R^2) is used^[51]. R^2 measures how much of the original uncertainty in the data can be explained by the regression model created. Intuitively, this can be thought of as the following:

How much benefit is gained in using the regression model?

The value of the coefficient of determination ranges between 0 and 1. An R^2 value of 0 means that there is no benefit in using the regression model. On the other hand, an R^2 value of 1 means that the regression model gives a perfect description of what is going on in the data.

Validation of Models

An effective model will be judged by its validation ability. The validation ability of a model informs how well it does in generalising to unseen data. To see how good a model performs, one checks its prediction on a validation set. A validation dataset is a portion of the original dataset that has not been used during training.

K-fold Cross-validation

Cross-validation is a type of validation that samples and replaces the validation set many times^[47]. K-fold cross validation works by splitting the training dataset into k folds (subsets of the dataset containing some observations) and taking one fold at a time, training it with the other k-1 folds prior to predicting upon the remaining fold.

The final grading of the model will be based on the average performance across the k folds. When k is n (n -folds), this becomes an extreme case of cross validation called Leave One Out Cross-Validation (LOOCV)^[47]. Here, each subset (fold) is a single observation and as a result gives a better indication of the model's ability. The downside to such a high value of k , n , leads to the model being more computationally expensive to run^[47]. Transfer Learner considers 10 seasons of which there are 20 teams in each. Therefore, 10-fold cross validation is used to represent a season of teams (subset of all the teams in all the seasons) and the R^2 score generated will be compared across the models to choose the best one for Transfer Learner. In addition from research^[59], it shows that 10-fold cross validation is an optimal number for cross validation.

7.5.6 Identifying important team features

The model chosen has capabilities to determine which attributes make up a good team. For example, in Decision trees, the attribute at the root node is the greatest predictor. What this means for Transfer Learner is that the root node attribute is the most important in predicting the quality of a football team. Also, in order to reduce the number of attributes present in the primary dataset, an iterative removal of non-predictive attributes (which are suggested by the models) will occur. This will result in a subset of team attributes that are strong predictors of the makings of a good team. This answers the first part of the problem statement (assessing the quality of a football team).

7.6 Optimisation

7.6.1 Making Transfer Decisions

Transfer Learner's idea of transfer decisions involves the idea of exchanging players. This means that a transfer decision involves buying a player from another team but at the same time selling a player to that team. Each transfer would either cause the team to gain money, lose money or stay the same financially. To make transfer decisions, Transfer Learner for each team, will choose the optimal combination of goalkeeper, defender, midfielder and forward swaps that the team should make. Given that all the 4 player position types are combined, this makes the time to compute the transfer swaps $O(gdmf)$ where g is the number of possible goalkeeper swaps, d is the number of possible defender swaps, m is the total number of midfielder swaps and f is the total number of forward swaps. These swaps are further investigated in the Implementation section. In addition, as the time taken to sort through a list of predicted values for transfer decisions is $O(n \log n)$, where n is the size of the list. Transfer Learner therefore has a total time complexity of $O(n \log n) + O(gdmf)$. As the combination of possible swaps in all the transfer decisions is larger than the number of transfer decisions made. This gives transfer learner a time complexity of $O(gdmf)$.

The solutions to the problem of optimising transfer decisions will be of 2 types.

1. Best Cost effective
2. Greatest value

The best cost effective transfer decision for a team will be a collection of swaps that does not cause the team to spend money after the transfer decision. This set of swaps represents the best cost effective transfer decision possible, based on the machine learning model's prediction on the relative effect of the transfer swaps in total. The greatest value transfer decision looks at the best transfer decision a team can make without taking into consideration its effect on the team's budget.

The choice of an optimisation technique depends on the structure of the problem. The following serves as a reason behind Transfer Learner's optimisation being a linear program.

Linear Program

For linear programs, there are a few assumptions that need to be satisfied^[14]:

1. Deterministic property: All parameters for the problem are certain (known beforehand).
A parameter is any coefficient for the functions stated, for example in the function $f(x) = 2x$, the parameter is 2 and the variable is x
2. Divisibility: The variables can take any rational value. A rational value is any number that can be expressed as the fraction p/q of two integers, p and q . Note that q has to be non zero
3. Linearity: The objective function and the constraints mentioned are linear. For example:
 $f_1(x) = 6x_1 + 14x_2^2$ is nonlinear $f_2(x) = 3x_1 + \sqrt{7}x_2$ is linear

Transfer Learner as a Mathematical Programming problem

The problems stated below are the optimisation problems to be solved.

1. Choosing the greatest valued cost effective transfer decision (problem 1: P_1)
2. Choosing the greatest valued transfer decision with no financial restrictions (problem 2: P_2)

For the 2 optimisation problems below, $X = [x_1, x_2, x_3, x_4]$

where:

x_1 : *goalkeeper swap_{value}*

x_2 : *defender swap_{value}*

x_3 : *midfielder swap_{value}*

x_4 : *forward swap_{value}*

In addition, for P_1 , $Y = [y_1, y_2, y_3, y_4]$

where:

y_1 : *goalkeeper swap_{cost}*

y_2 : *defender swap_{cost}*

y_3 : *midfielder swap_{cost}*

y_4 : *forward swap_{cost}*

7.6.2 Best Cost Effective Transfer Decision

P_1 : $\max f(X) = \text{goalkeeper swap}_{value} + \text{defender swap}_{value}$
 $+ \text{midfielder swap}_{value} + \text{forward swap}_{value}$

Subject to the constraints that: $g(Y) \leq 0$

where: $g(Y) = \text{goalkeeper swap}_{cost} + \text{defender swap}_{cost} + \text{midfielder swap}_{cost} + \text{forward swap}_{cost}$

7.6.3 Greatest value Transfer Decision

P_2 : $\max f(X) = \text{goalkeeper swap}_{value} + \text{defender swap}_{value}$
 $+ \text{midfielder swap}_{value} + \text{forward swap}_{value}$

Subject to the no (financial) constraints

Are the Linear Program Assumptions fulfilled

The assumptions for a linear program has been satisfied by the Transfer Learner assumption as seen below:

1. Deterministic property: For both problems all the parameters present have the value of 1 and is known beforehand. For example in problem 1. $f(X) = 1 * x_1 + 1 * x_2 + 1 * x_3 + 1 * x_4$ and similarly for $g(Y)$. Therefore this assumption has been satisfied
2. Divisibility: Each position dependent swap value for P_1 and P_2 (e.g *goalkeeper swap_{value}*) are rational values as they are predictions from the machine learning model for the positional exchange
3. Linearity: Trivially, $f(X)$ for P_1 and P_2 are linear functions

7.7 Web Interface

In Transfer Learner, the web interface must provide a platform for users to view the transfer suggestions in a way that is engaging and easy to consume. Design techniques emphasise the beauty of minimalism (simplicity)^[52]. To confine to these guidelines, the web interface does the following tasks, simply but effectively design-wise:

- (a) User picks a season
- (b) User picks a team in the selected season
- (c) Two options provided to user for the selected team:
 - i. Cost effective transfer decisions
 - ii. Best (in terms of predicted value) transfer decisions
- (d) User picks any of (3)
- (e) Display results

Depending on the users choice, the following output is displayed:

- Current team
- Transfer decisions suggested
- Updated team
- Effect of transfer decisions on budget
- Effect of transfer decisions on the team value

It is important to mention that the web interface is a demonstration added to give a visual solution to the user. It's use is to be a simple but effective demo of how transfer learner works.

Chapter 8

Implementation

The Transfer Learner is implemented in the Python^[59]. Python is a popular general-purpose programming language with an expansive library of tools for data analysis, web frameworks and other useful functionality. The tools for data analysis especially made Python a suitable language for this project, because this project involves extracting useful data from various sources and then processing and cleaning that data. The following tools are used to implement Transfer Learner:

8.1 Data Extraction

The first component of the Transfer Learner is data extraction. Data extraction is the process of retrieving useful data from sources that are usually poorly structured, and then performing some processing on this retrieved data. In Transfer Learner, data extraction is performed with the use of a web scraper and is instructed to extract the following data (from their respective sources):

- Team attributes
- Player attributes

The web scraper is implemented in Python using a library called *Beautiful Soup*^[58]. This

library is designed to navigate documents (in this case, web pages) and constructing a tree of elements that the document consists of. This tree of elements contains a collection of data of which only the useful data is required. To extract only useful data, the web scraper navigates the tree of data and stores the useful data in appropriate Python variables for later use. In this project, Transfer Learner uses a web scraper to collect data on football teams and players in the Premier League. The useful information collected by Transfer Learner includes the team names, the player names, the number of goals scored/conceded and other attributes that can be used when training the machine learning models. This information was located by studying the general structure of the web page (the source of the data) and compiling functions that extracted the required information from the required locations of the web page.

The collection of useful attributes can be split into two parts: team attributes and player attributes. The team attributes refer to features that characterise a given team. The player attributes refer to features that characterise a given player. Once all the necessary features have been extracted and stored in a Python object, they are then written to a CSV (comma separated values) file for further processing.

Although the web scraper helped in extracting useful data, it does not ensure that the data is formatted correctly. To accommodate this, the CSV file is subjected to a sequence of Linux commands. These Linux commands are specially selected to iterate through the data in the CSV file, modify that data appropriately and then output the modified data into a new CSV file. The following modifications were applied to the extracted data:

- (a) Removing occurrences of the term **None**
- (b) Replacing instances of a newline
- (c) Replacing exclamation marks with a newline
- (d) Removing occurrences of commas at the start of each line
- (e) Removing occurrences of commas at the end of each line

Due to the poorly structured web pages, from where the data was extracted, the web scraper may have found no data at a location where it was expecting useful data. This resulted in the web scraper storing a `None`, to represent "nothing".

In the CSV file, attributes corresponding to a specific player/team were separated by a newline. In CSV, each occurrence of a newline indicates a new row of data, however the attributes separated by a newline belonged to the same "row of data". To group these attributes together with the correct corresponding player/team, the instances of a newline were replaced by a comma.

In the CSV file, each set of player/team attributes were separated by an exclamation mark. However, the use of an exclamation mark to separate different "rows of data" was incorrect syntax. The correct syntax was using a newline, therefore the occurrence of an exclamation mark is replaced with a newline.

In some rows of data, in the CSV file, there would sometimes be occurrences of multiple contiguous commas at the beginning of a row and at the end of a row. This was a result of replacing occurrences of newline with a comma. In CSV, multiple contiguous commas empty column data. To remove this incorrect information, the occurrences of commas at the beginning and end of a row were removed.

8.2 Exploratory Data Analysis and Data Cleaning

To analyse the relationship between different attributes, a program called Weka was used. Weka is a program that contains a variety of machine learning algorithms. One useful functionality provided by Weka is a graph that visually represented the set of data contained in the CSV file. In particular, this graph visually conveyed any correlations that existed between the different attributes of the players/teams. This graph was very useful in analysing the relationships that existed between the various attributes. In addition to this, it also made it possible to identify different attributes that represented the same

piece of information, for example 'Goals Scored' and 'Goals'. These two pieces of duplicate information occurred as a result of the same information appearing in multiple places on the same web page. Removing these duplicates was necessary to extract the relevant data (unique and required information) from the large collection of useful data (information that was deemed useful during the extraction process).

To remove the duplicate information, a program called LibreCalc was used. LibreCalc is a spreadsheet program available for Linux machines that offers the ability to edit CSV files. Using Weka, the duplicate attributes were located and then using LibreCalc, they were removed from the CSV file. During the data extraction, it was mentioned that some data may have been incorrectly formatted. However, some of these incorrect formats were unique to each row of data. This required the use of LibreCalc to manually investigate the rows of data within the CSV file and locate any errors. These errors would have to be verified against the corresponding data on the Premier League website, and then updated to represent the correct information. This process of finding and resolving errors within the CSV file is an ongoing process that will persist throughout the development of Transfer Learner. Fortunately, with each correction made, the CSV file better represents the information it is intended to.

8.3 Machine Learning

At the core of Transfer Learner is the machine learning model that takes a set of team attributes (composed from a set of player attributes), and predicts the value of that particular team roster. Transfer Learner is designed to solve a linear regression problem, whereby a machine learning model tries to model the relationship between a given team's attributes and the value of that given team. This constitutes to a linear optimisation, therefore linear regression models were considered for Transfer Learner. Non-linear models imply a substantially more complex optimisation problem. The implementation of the machine learning component of Transfer Learner can be broken down into the following objectives:

- (a) Loading team statistics from CSV files
- (b) Partitioning and standardising team statistics
- (c) Comparing performance of machine learning models
- (d) Choosing the best performing machine learning model

After applying the necessary modifications to the CSV file of player and team data, it is then appropriate to use this data to train a machine learning model(s) to be effective at predicting a team's value. Data that is read from the CSV file(s) is stored in a **DataFrame**. A **DataFrame** is a two-dimensional tabular data structure. This data structure enables the Python code to easily extract the required piece of information from the structured set of data retrieved from the necessary CSV file. Using the specified team name and season number, two sets of information are retrieved from two CSV files: the statistics of all teams, and the statistics of all players. These statistics include the team's value and the player's value respectively.

Following the paradigm for regression models, the team names and season numbers are encoded with a value between 0 and the total number of team names or season numbers (respectively).

The team statistics and the corresponding team values are stored separately, in separate **DataFrames**, in preparation for training the machine learning models in the subsequent stage.

Before the set of team statistics (attributes) are used to train the machine learning model(s), a function from the a Python data analysis tool is used to standardise the attributes of the teams to help the model(s) more effectively discover relationships between the different attributes.

To decide upon which regression model would be used in Transfer Learner, the following models were evaluated and compared:

- Lasso Regression
- Ridge Regression
- Decision Tree

- Ordinary Least Squares (OLS)

The comparison of the models involved using K-Fold Cross Validation to obtain a cross validation score, more specifically the R^2 score, of each model being evaluated. The performance of each model was then compared using its evaluated R^2 score.

Within each performance test, the set of team statistics, on which the models were trained, was gradually reduced. Not all of a team's attributes were important when establishing the relationship between a team's attributes and its value. To remove these "unnecessary" features, the models R^2 scores were evaluated over several rounds of cross validation. Each round involved removing a different set of attributes from the team statistics. Each time a new set of attributes were removed from the team statistics, the R^2 score would be calculated (for each model) and compared. After 9 rounds of evaluation, the set of remaining attributes were considered as the "important" attributes necessary for a model to effectively predict the value of a team, based on this set of attributes.

From the models listed above, the Lasso and Ridge models were seen to be the *best* models at predicting a team's value. Although the Ridge model was seen to perform slightly better than the Lasso model, the Lasso model is currently used in Transfer Learner due to it being less computationally expensive. Due to the stronger penalising

8.4 Optimisation of Transfer Decisions

Using the chosen regression model, Transfer Learner generates a list of all possible transfer decisions that can take place using the specified team (and season) as the point of reference. The formation of the list works by initially partitioning the set of player statistics into 4 subsets (representing a distinct player position): a set of goalkeepers, a set of defenders, a set of midfielders and a set of strikers.

Assume team A is team specified in the parameters for Transfer Learner. For each position in team A , Transfer Learner would accumulate a list of *swaps* performed on players in team A with players of the same position that are not in team A . The *swaps* lead to a

change in team *A*'s team roster, and in turn this has an effect on team *A*'s value and budget. These effects are stored in conjunction with each *swap* made in a **DataFrame**. The relative change in the values of team *A*'s attributes, are the values that Transfer Learner uses to predict the new value of team *A* when a *swap* of players has taken place. The output of this process is four **DataFrames** that represent the transfer decisions (*swaps*) made for each player, in team *A*'s roster, within their respective player position.

The next step is to combine all four sets of transfer decisions to obtain all the possible combinations of transfer decisions that can be made in team *A*. These combinations would represent all the different permutations of the four sets of transfer decisions generated in an earlier stage. The important attribute that is recorded with each permutation, is the collective effect (on team *A*'s budget and value) that is accumulated from a particular combination of transfer decisions (*swaps*). This collection of permutations (of transfer decisions) is then stored in another **DataFrame** called *teamSwaps*.

As in previous stages, the attribute values within *teamSwaps* are standardised to aid the machine learning model in identifying the relationship between the attributes. As noted earlier, the machine learning model that is currently used in Transfer Learner is the Lasso Regression model. After standardising the attributes, the permutations of transfer decisions are evaluated by the model to predict the new team value of team *A*.

To solve the two optimisation problems - the best-cost-effective and the greatest-value transfer decisions - Transfer Learner manipulates these permutations to obtain a set of *swaps* that represent the solution to these problems.

To obtain the solution to the greatest-value transfer decision problem, Transfer Learner begins by ordering the set of *swaps*. Using the effect of a *swap* on a team's value, Transfer Learner orders the *swaps* in a descending order of that effect. That way, the *swap* that represents a single permutation of transfers that have the best positive effect on a team's value, will be at the top of the list of *swaps*. This *swap* is considered as the transfer decision with the "greatest value", as determined by Transfer Learner.

```

# best cost effective decision
best = swaps_predictions.argsort()[::-1] # [::-1] reverses the array but sorted by the best values

team = np.array(team_subset)

stop_after = 500

counter = 0
for index in best:
    counter += 1
    print "gone through", counter, "transfer decisions"

    if counter <= stop_after:
        # transfer decision to be made
        # name of players
        team_list_init = np.copy(team[:,0]) # initialising array again need to copy by value not reference
        swap = np.array(swaps_dataframe)[index,:]
        # checking that this affects the budget positively (should not have to spend extra)
        cost_of_transfer_decision = swap[-2]
        if(cost_of_transfer_decision <= 0):

            print "\nSeason number:"
            print season_num
            print "\nTeam name:"
            print team_name
            print "\nInitial roster: "

            team_player_names = [player[0] for player in team]
            team_player_positions = [player[2] for player in team]
            for player in range(len(team_player_names)):
                print team_player_names[player], team_player_positions[player]

            print "\nBest Transfer decisions:"

            for swap_num in range(len(swap[0])):
                print "swap", swap[0][swap_num], "with", swap[2][swap_num], "from", swap[3][swap_num]

            print "\nCost on the teams budget (negative - good, positive - bad):"
            print "£", cost_of_transfer_decision, "M (Fantasy Premier League Millions)" # Cost on the budget
            print "\nAdded value on the team after suggested swaps:"
            print swaps_predictions[index]

            size = len(swap[0])

            for val in range(size):

                index_current_player_team_list = np.where(team_list_init == swap[0][val])
                # replace player with swapped player in the updated roster
                team_list_init[index_current_player_team_list] = np.array(swaps_dataframe)[index,2][val]

                new_team_player_names = [player_name for player_name in team_list_init]

                new_team_player_positions = team_player_positions
                # player positions for the swaps will remain the same as we are swapping position wise

                print "\nRoster after suggested transfer swaps:"
                for player in range(len(new_team_player_names)):
                    print new_team_player_names[player], new_team_player_positions[player]

                break

            else:
                print "no cost effective transfer decisions"
                print "keep original team"
                for player in range(len(team_player_names)):
                    print team_player_names[player], team_player_positions[player]

# no cost effective solution requires them spending a bit of money
# more considerations like the player wanting to leave

```

Figure 8.1: Python code for the cost effective transfer decision

The solution to the best-cost-effective transfer decision problem is based on the solution to the greatest-value transfer decision problem. This time, the effect that a *swap* has on a team's budget is also taken into consideration. The definition of the "best cost effective" transfer decision is a *swap* where the effect of the *swap* on a team's budget is zero or less. This means the team's budget either remained the same or increased with the *swap*. The solution to this problem involved iterating through the ordered list of *swaps*, from the previous stage, and locating the first *swaps* whose effect fits the criteria. This *swap* is considered the "best cost effective" transfer decision determined by Transfer Learner.

8.5 Web Interface

To provide an interface for users to interact with Transfer Learner, a Python web interface was developed. The web interface is split into two parts: the server-side and the client-side. The server-side represents the Python files that are responsible for processing user selections and making the necessary function calls to Transfer Learner. The client-side represents the files that are responsible for presenting a visual interface for users to interact with the options available in Transfer Learner. Some of the files in the client-side are written in another programming language called JavaScript. JavaScript is used in this web interface to manipulate the elements that are present in the web page (whether invisible/visible).

The general structure of the web interface consists a logo of the Premier League, some selection boxes (for users to choose their desired team and season number), a loading screen (in the form of dynamic text) and two images used to represent the two optimised results.

To deliver an simple yet effective user interface, the colour palette was chosen to consist of only three colours: black (background), white (text) and green (highlights/accents). The high contrast between the background and the text ensures the text can be easily read by all users.

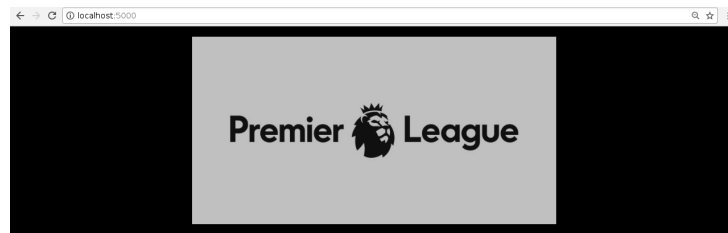


Figure 8.2: Portal Screen to web interface

The web interface begins at a simple screen with the Premier League logo positioned in the middle of the screen. Initially the logo is presented in greyscale, however by hovering over the logo, its colour is restored. By not hovering over the logo, it returns to its greyscale colour palette. This transition is used to imply that the logo requires some form of interactivity, much like a hyperlink in a web page. This screen symbolises a portal screen, whereby clicking on the logo transports the user to a dynamic web page that contains everything associated with Transfer Learner. The logo acts as a way for the user to reset their decisions and restart the web interface. The dynamic transitions within this web interface are achieved through the use of JavaScript. Upon detecting a button click, some JavaScript code repositions the logo and makes the first selection box visible to the user. A selection box represents a drop-down menu from which a user can select one item (in this web interface).

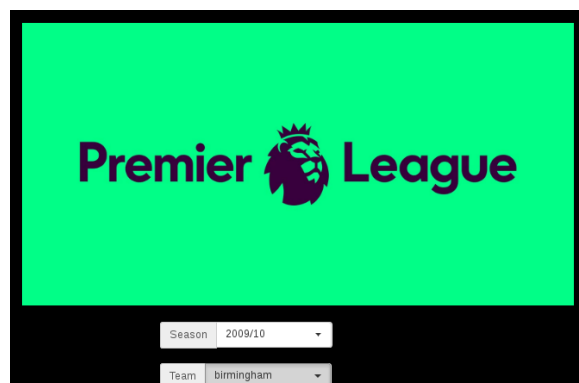


Figure 8.3: User selection of Birmingham in the 2009/10 Premier League season

The selection box contains a list of season numbers that can be selected by the user.

Upon selecting a season number, another selection box appears displaying the names of the teams available for that season number. Similarly to the first selection, the user selects a team name to proceed through the web interface. The selection boxes provided an easy way for users to navigate through all the possible options, and reduced the chances of a user inputting incorrect data. In addition to this, by not receiving raw input from the user, the web interface does not necessary having to clean or filter the user input (to get rid of special characters and/or malicious code).

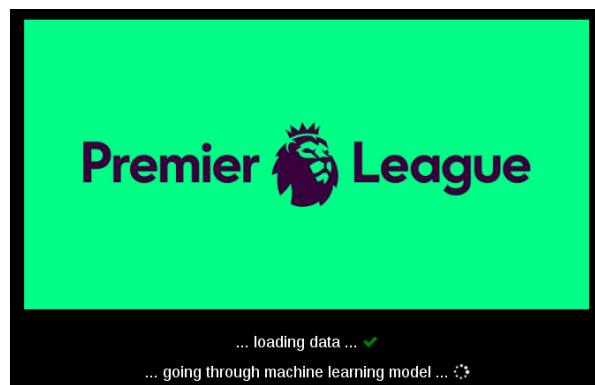


Figure 8.4: Loading screen before transfer decisions are outputted

When the user has selected their desired season number and team, the selection boxes are replaced with a loading screen. The aim of this loading screen is to notify the user, in real time, of the current tasks being performed by Transfer Learner. This is done by displaying appropriate messages (and status icons) upon the start of each function, in Transfer Learner, that corresponds to a particular task. These tasks include reading in the data from the CSV file, processing that data, running a machine learning model and collating a team of players.

To dynamically update the messages (without a page refresh), a method using AJAX is used to communicate between the client-side files and the server-side files. An AJAX call is sent from a JavaScript file (client-side) and is directed at a Python file (server-side). The Python file then makes a function call that corresponds to the execution of some task(s). Within the processing file for Transfer Learner, there are *checkpoint* that

mark the completion of a significant task. Each AJAX call is used to begin the execution of a task. When the task is completed (and/or has reached a *checkpoint*), the JavaScript file receives a response (to the AJAX call it made) and updates the required message. After the message has been updated, the JavaScript issues another AJAX call to execute the subsequent task. This process is repeated until all tasks have been completed, and no more message updates are necessary.

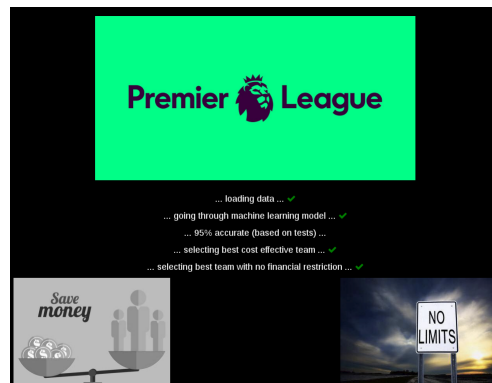


Figure 8.5: Cost effective and greatest value transfer decision example

The response of the last AJAX call is a collection of results for the two optimisation problems: the best-cost-effective transfer decision and the greatest-value transfer decision. This indicates that all the necessary tasks have been completed by Transfer Learner. At this stage, some JavaScript code is responsible for presenting two images that each represent one set of results. Similarly to the logo, the 2 images alternate between greyscale and colour when the user moves their mouse on/off the images.

Upon clicking on one of the images, the user is presented with a modal containing the results related to the optimisation solution selected. A modal is pop-up window that is displayed on top of the content of a web page. In this web interface, a modal is used to present the results of solutions to the two optimisation problems. The results include a table that visually depicts the the old team roster and the new team roster; a section highlighting the transfer decisions that were made; and a section describing the effect of those transfers on the chosen team's budget and value. The table featuring the team

compositions included a colour coded indicators, that helped the user identify the position of a particular player in a team roster. One reason for using the modal to display these results, was that user could close the modal and conveniently choose between displaying the results for either of the optimisations.

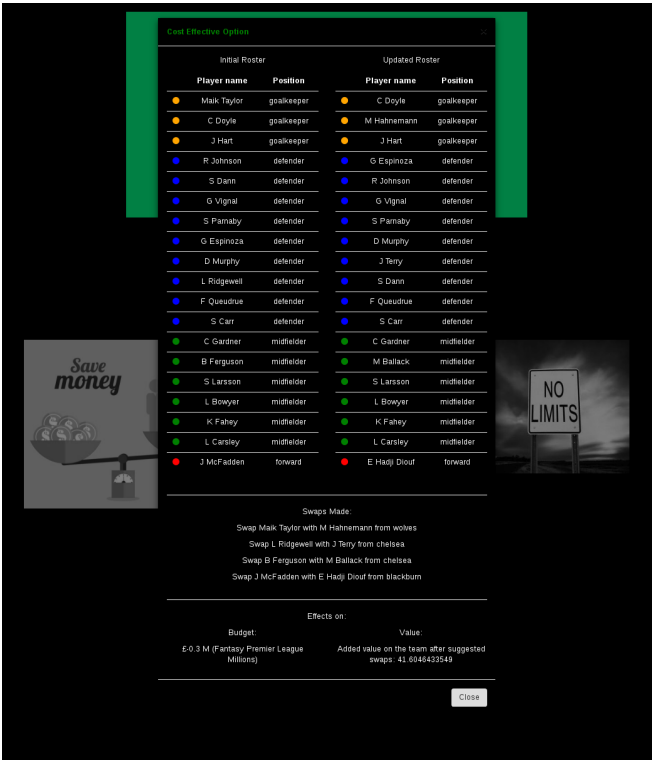


Figure 8.6: Modal showing the cost effective transfer decision for Birmingham in the 2009/10 EPL season

Best Value Options

Initial Roster			Updated Roster		
Player name	Position		Player name	Position	
Maik Taylor	goalkeeper		C Doyle	goalkeeper	
C Doyle	goalkeeper		P Cech	goalkeeper	
J Hart	goalkeeper		J Hart	goalkeeper	
R Johnson	defender		G Espinoza	defender	
S Dann	defender		R Johnson	defender	
G Vignal	defender		S Parnaby	defender	
S Parnaby	defender		G Vignal	defender	
G Espinoza	defender		D Murphy	defender	
D Murphy	defender		J Terry	defender	
L Ridgwell	defender		S Dann	defender	
F Quendruue	defender		F Quendruue	defender	
S Carr	defender		S Carr	defender	
C Gardner	midfielder		M Ballack	midfielder	
B Ferguson	midfielder		S Larsson	midfielder	
S Larsson	midfielder		C Gardner	midfielder	
L Bowyer	midfielder		L Bowyer	midfielder	
K Fahey	midfielder		K Fahey	midfielder	
L Carsley	midfielder		L Carsley	midfielder	
J McFadden	forward		E Hadji Diouf	forward	

Swaps Made:

- Swap Maik Taylor with P Cech from chelsea
- Swap L Ridgwell with J Terry from chelsea
- Swap B Ferguson with M Ballack from chelsea
- Swap J McFadden with E Hadji Diouf from blackburn

Effects on:

Budget:	Value:
£8.8 M (Fantasy Premier League Millions)	Added value on the team after suggested swaps: 51.8862485025

Close

Figure 8.7: Modal showing best valued transfer decision for Birmingham in the 2009/10 EPL season

Chapter 9

Results and Analysis

9.1 Results

This section describes the results of the main goals for Transfer Learner. The main goals have been split into 3 sections: assessing the qualities of a football team, making transfer decisions and analysing how successful the transfer decisions were.

9.1.1 Assessing the qualities of a football team

There were 30 attributes present in the original team data extraction. The possible attributes can be seen below.

Premier League		Fantasy	Stats	Video	Communities	More	Sign in	
Home	Fixtures	Results	Tables	Broadcast	Tickets	Clubs	Players	More
Goals	1,685	Passes	219,540	Clean Sheets	377	Yellow Cards	1,406	
Goals Per Match	1.76	Passes Per Match	229.40	Goals Conceded	909	Red Cards	83	
Shots	6,662	Pass Accuracy %	84%	Goals Conceded Per Match	0.95	Fouls	715	
Shots On Target	2,382	Crosses	9,606	Saves	445	Offsides	975	
Shooting Accuracy %	36%	Cross Accuracy %	21%	Tackles	8,597			
Penalties Scored	48			Tackle Success %	75%			
Big Chances Created	521			Blocked Shots	1,713			
Hit Woodwork	186			Interceptions	6,904			
				Clearances	11,911			
				Headed Clearance	4,038			
				Aerial Battles/Duels Won	29,598			
				Errors Leading To Goal	82			
				Own Goals	39			

Figure 9.1: Example of original set of team attributes from the Premier League

In each stage of the data cleaning process, the list of attributes were reduced to the following 9 player/team attributes:

```
[ 'season_number', 'wins', 'losses', 'goals', 'big_chances_created', 'passes', 'saves', 'headed_clearance', 'fouls' ]
```

Figure 9.2: 9 Final attributes used in the final machine learning model

The machine learning models used had the following results:

```

In [18]: # Ridge regression

# R2 score
# rgr means regressor
ridge_rgr = make_pipeline(preprocessing.StandardScaler(), RidgeCV())
# 200 samples
score = cross_validation.cross_val_score(ridge_rgr, teams, teams_label, c
print("Ridge R2 score is", "mean :", score.mean(), "std:", score.std())

('Ridge R2 score is', 'mean :', 0.94973704014776406, 'std:', 0.0179480
18910425051)

```

Figure 9.4: Transfer Learner's Ridge regression results, R2 score

```

In [17]: # OLS

# R2 score
# rgr means regressor
ols_rgr = make_pipeline(preprocessing.StandardScaler(), LinearRegression)
# 200 samples
score = cross_validation.cross_val_score(ols_rgr, teams, teams_label, c
print("OLS R2 is", "mean:", score.mean(), "std:", score.std()) # ma

('OLS R2 is', 'mean:', 0.94935729619344433, 'std:', 0.0182915982410289
56)

```

Figure 9.3: Transfer Learner's Ordinary Least Squares results, R2 score

```

In [19]: # Lasso regression

# R2 score
# rgr means regressor
lasso_rgr = make_pipeline(preprocessing.StandardScaler(), LassoCV())
# 200 samples
score = cross_validation.cross_val_score(lasso_rgr, teams, teams_label,
print("Lasso R2 score is", "mean:", score.mean(), "std:", score.std())

('Lasso R2 score is', 'mean:', 0.94937845135488286, 'std:', 0.01814307
9337179902)

```

Figure 9.5: Transfer Learner's Lasso regression results, R2 score

As mentioned earlier, Decision Tree Regressor is a piecewise linear function. This means it has no direct linear function representation. The term piecewise refers to the path from the root node attribute to the leaf node. The root node is the attribute with the greatest

standard deviation reduction and the leaf node is a target value. In Transfer Learner, the Decision Tree Regressor generated an R^2 score of 0.96.

```
In [20]: # Investigating important features
# Standardisation
scaler_train = preprocessing.StandardScaler().fit(teams)
scaled_teams = scaler_train.transform(teams)

# USE DECISION TREE regressor to see the features that are 'most important'
# Based on the Mean Square Error impurity - standard deviation reduction
dt_rgr_mse = DecisionTreeRegressor(criterion='mse')
dt_rgr_mse.fit(scaled_teams, teams_label)

mse = pd.DataFrame({
    'Attribute': teams.columns,
    'Importance (MSE)': dt_rgr_mse.feature_importances_
}).sort_values(by='Importance (MSE)', ascending=False).reset_index(drop=True)

mse
```

```
Out[20]:
```

	Attribute	Importance (MSE)
0	saves	0.728845
1	headed_clearance	0.096874
2	losses	0.086480
3	passes	0.046374
4	wins	0.015661
5	big_chances_created	0.014234
6	goals	0.009490
7	season_number	0.001756
8	fouls	0.000286

Figure 9.6: Transfer Learner's Decision Tree regression results, Standard deviation reduction

From the models chosen, the decision tree regressor had the best R^2 score. However, due to the decision tree regressor being a piecewise linear model, this made it unsuitable to be used in the the linear program. Although, the decision tree regressor was not used, it gave an added dimension for analysis. However, the decision tree regression model attribute importance shows that when it comes to predictive the value of a team, it believes that the number of saves team makes is crucial in judging how good a team will be. Qualitatively, this implies that a team having a great goalkeeper is a huge indicator in determining how good a team is as they would generate more saves for the team. In addition, as seen in the OLS and OLS regularisation models, their R^2 score was impressive. This emphasises the success of each model used in Transfer Learner.

Season number	Teams available
2006/07	Blackburn, Everton, Liverpool, Newcastle, Reading, Watford, West Ham, Wigan
2007/08	Aston Villa, Birmingham, Chelsea, Fulham, Liverpool, Newcastle, Portsmouth
2008/09	Blackburn, Bolton, Chelsea, Hull, Manchester City Newcastle, Portsmouth
2009/10	Birmingham, Blackburn, Burnley, Fulham, Portsmouth, West Ham, Wigan, Wolves
2010/11	Birmingham, Bolton, Liverpool, Wigan, Wolves
2011/12	Fulham, Wigan
2012/13	Everton, Fulham, QPR
2013/14	Fulham, Hull, Liverpool
2014/15	none
2015/16	none
2016/17	none

Table 9.1: Premier League Teams and Seasons with transfer decisions from Transfer Learner available

9.1.2 Making transfer decisions

Through the optimisation methodology mentioned in the design and implementation stage. There were transfer decisions suggested for the teams seen in Table 9.1.

The teams that were not present are due to some errors in the original datasource. For example, names like Mesut Özil were recorded incorrectly as Mesut Odzil in the extracted dataset instead of Mesut Ozil. One effective way to solve this transcription error was to manually compare and make the required changes. To evaluate all 5000 players (worst-case scenario) is time inefficient, therefore remains an ongoing task. However, for the given teams in the table above, the cost effective and best valued transfer decision suggestions are generated. These suggestions were from the Lasso and Ridge (seen in the later section) regression model.

9.1.3 Analysing the success of the transfer decisions made

With the Decision Tree Regressor not being considered for the Optimisation segment, this left the OLS and Linear regression regularisation models (Ridge and Lasso). Given that OLS is prone to overfit (or underfit), the choice was made to make use of the Regularisation Linear Regression models. To analyse how successful the transfer decisions for each team were, it was important to ensure that both of the regularisation models agreed with each other. The transfer decisions predicted by both models were identical, however, the effect on the value of the team differed slightly. This is because of the slight difference in the values chosen for the coefficient of each linear model.

For example, this is the greatest-value transfer decision determined by Lasso for Aston Villa in 2007/08

```
Best Transfer decisions:
swap S Taylor with E van der Sar from manchester utd
swap O Mellberg with G Clichy from arsenal
swap G Barry with M Flamini from arsenal
swap S Maloney with D Kuyt from liverpool

Cost on the teams budget (negative - good, positive - bad):
£ 0.6 M (Fantasy Premier League Millions)

Added value on the team after suggested swaps:
50.5567083881
```

Figure 9.7: Lasso-based greatest value transfer decision

Here is the Ridge regression equivalent

```
Best Transfer decisions:
swap S Taylor with E van der Sar from manchester utd
swap O Mellberg with G Clichy from arsenal
swap G Barry with M Flamini from arsenal
swap S Maloney with D Kuyt from liverpool

Cost on the teams budget (negative - good, positive - bad):
£ 0.6 M (Fantasy Premier League Millions)

Added value on the team after suggested swaps:
46.5802431644
```

Figure 9.8: Ridge-based greatest value transfer decision

This inspires confidence that the transfer decisions suggested are good transfer decisions.

9.2 Analysis of Objectives

In Transfer Learner, no objective was marked with a fail but there are still avenues for improvement. For example, the full data source integration of players had problems due to player names being incorrectly recorded in the data source. In future work, it would be interesting to further investigate methods for automating this integration, so that Transfer Learner could become an automated program that collects player information on a season by season basis.

Chapter 10

Project Management

This project progressed to the completion of the specified objectives, primarily because of the invaluable and consistent meetings with Dr Theo Damoulas. All the objectives were successfully completed and for any encountered issues, alternative solutions were put in place and incorporated to mitigate the issues.

The original timetable constructed in the Gantt Chart for the project specification, differed slightly to that of the progress report, due to the issue with original datasource. As mentioned in the Design section, there were deep security measures placed in the original choice for the data source, WhoScored. In addition, due to the lack of prior experience with extracting a dataset or implementing a webscraper, the timeline was expected to differ. However, due to the desire to make this lofty goal of a project successful, this issue was resolved. Each task in the Gantt Chart were given suitable contingency time to ensure no significant delay.

Chapter 11

Author's Assessment of the Project

What is the technical contribution for this project?

This project combines technical content from a number of areas to provide a useful transfer decision making tool, namely: web scraping, data cleaning, data integration specifically data merging, machine learning, optimisation and web development. Implementing a web scraper to extract information from a big web interface like the Premier League's official website, is not an easy feat. Neither was the data cleaning process, however, this reflects the nature of data science projects. This is due to most of the time spent in the data science workflow being the act of extracting and cleaning up the data to its required form. In addition, three different machine learning model types are then applied to this data. The first is OLS, the second is OLS with regularisation and the final model is a Decision Tree Regressor. Most projects tend to end when the machine learning models are applied and evaluated. However, this project goes two steps further in using the subject area of optimisation to create real life decisions in the form of transfer decisions and also creating a web interface to make it more engaging to the user. The use of external API's, research into the areas of machine learning and optimisation, working with unfamiliar frameworks and the design and analysis of algorithms make this a highly technical

project.

Why should this contribution be considered relevant and important for the subject of your degree?

The project utilises different aspects of Discrete Mathematics. Firstly, an organisational element is present not only in the project planning and execution but also in the deployment and preliminary understanding of a machine learning workflow. The ability to research and develop a machine learning project with the added theme of optimisation, is a valuable skill both in academia and in industry as a discrete mathematician. Equally, the research and design section of this thesis, emphasises the research needed for a discrete mathematics thesis. Technically challenging and exciting areas of machine learning and optimisation are also investigated here. Furthermore, developmental work with web technologies, web scraping and data processing were utilised. The production of a system implementing established computer science academic work highlights the skills needed for a Computer Science thesis. Finally, the project utilises algorithms, computational complexity and concepts in machine learning and optimisation, making it worthwhile as a Discrete Mathematics project.

How can others make use of the work in this project?

The resulting body of work serves as a strong introduction to how transfers can be done more efficiently and with a higher awareness of the finances involved. Also, Transfer Learner could be used as the background for other researchers that choose to delve deeper into what appears to be a new investigation. This could therefore accelerate progress into offering solutions to financial problems in Football from a transfer perspective. Similarly, football teams could use Transfer Learner to inspire transfer decisions that improve their own teams. In addition, this project and the idea behind it could be applied to a Fantasy Football setting to enable users to get optimal teams that give them a better

chance of success. Finally, for sport lovers and for machine learning and optimisation enthusiasts, this serves as a well confined body of work that could instigate discussion and hopefully more ideas.

Why should this project be considered an achievement?

As mentioned previously, this project appears to be the first of its kind that specifically applies machine learning and optimisation to the problem of transfer decisions in Football. This could generate huge possibilities with regards to transfer decisions and assessing how good a team is, in a way not seen before. Especially considering the machine learning model's high coefficient of determination score. Furthermore, the huge technical difficulty that comes with machine learning, optimisation and data extraction is difficult in isolation, not to mention when they are merged to one body of work as seen in Transfer Learner. Finally, given the limited technical ability of the author prior to the project (e.g. no previous experience with web scrapers), and to the point the project is at now, this project is an achievement. This project should be considered as an achievement as it completed the objective of suggesting transfers based on the important features of a team derived by the machine learning model. The degree of the achievement is further amplified when one considers the limited technical ability of the author prior to the project (e.g. no previous experience with web scrapers).

What are the limitations of this project?

By no means does the author consider this project perfect. As mentioned previously, it is a strong start, but nonetheless, still a start with many areas for improvement. A possible limitation could be Transfer Learner's applicability across football leagues. Transfer Learner currently uses only Premier League data. So, it will be interesting to see if it is as successful when using data from other leagues. In addition, a model is an abstraction of real life. This implies

that there are parts of real life that are unquantifiable. For example, Transfer Learner's dataset on Team and Player statistics does not currently measure non-quantitative metrics like Team Chemistry. These limitations mentioned could easily be investigated and hopefully solved in future work.

Chapter 12

Conclusion

12.1 Summary

Transfer Learner met all its proposed objectives, as seen in the results section. This technically challenging project demonstrates that access to quality data (via web scraping and manual inputting of data) and the use of machine learning and optimisation techniques generated promising results in football transfer decisions. Transfer Learner produced model that can predict the value of a team much better than just taking a guess, as seen in the 0.94+ seen in the R^2 scores for all the models used in Transfer Learner.

12.2 Further Work

There is great scope for improvement. A few of these ideas are discussed below, opening up the possibilities of further work in the future on Transfer Learner.

12.2.1 Scalability

The original dataset planned had security measures that prevented a web scraper from being used. This dataset had player and team information from leagues across the world.

A possible scope for development would be to get data from the football leagues around the world. Following this extraction, it would be interesting to see how well the machine learning model works in predicting the value of a team, assessing worldwide team attributes and using it to make transfer decisions.

12.2.2 Integration into the world of Football

Transfer Learner is currently a project that although generating promising results, needs to be utilised in the real world to see Transfer Learner's effect on real world transfer decisions. This will inspire further discussion and constructive feedback and further ways to improve the model (like researching a way to incorporate qualitative descriptors like team chemistry into the dataset).

12.2.3 Build upon transfer decision evaluation

Currently, the transfer decisions are extrapolations of the successful team evaluator model. This comes with the assumption that a successful team comes from the same underlying distribution that represents a successful transfer decision. A way to further improve this model, could be to generate a model that gets a collection of real life transfer decisions and creates a way to evaluate how good they are. This evaluation could then be adapted to Transfer Learner to better improve the way it evaluates how good a transfer decision is.

To conclude, thank you for taking time out to read about Transfer Learner. A machine learning and optimisation football program, that assesses how good a football team is and uses the information gathered to generate optimal transfer decisions.

Joshua Uwaifo

Grace and peace.

Chapter 13

References

- [1] Kevin Bliss, Live Strong. (2011). What Is Soccer?
- [2] Biggest Global Sports. A statistics-based analysis of the world's most popular sports.
- [3] Ben Rumsby, The Telegraph. (2016). Manchester derby: Match on course to become the most watched match in top-flight history.
- [4] James Rowland, University of Exeter. (2001). Policing European Football Hooliganism.
- [5] Louise Taylor, The Guardian. (2016). LMA demands 'full disclosure' of newspaper's corruption investigation.
- [6] Dipo Faloyin, Vice Sports. (2016). Rethinking the Transfer Window: Why players should be traded not transferred.
- [7] Jules Delay, Nouse. (2013). Zidane the greatest of all time.
- [8] Vivek Chaudhary, ESPN. (2016). How the Premier League's record TV deal will impact football in England.
- [9] Murad Ahmed and John Burn-Murdoch, Financial Times. (2017). Football's smartest spenders in the multibillion-euro transfer market.

- [10] Machine Learning, Introduction. (2017). Warwick University. Dr Theo Damoulas.
- [11] Recommender Systems. (2016). CS910 Foundations of Data Analytics. Prof Florin Ciucu.
- [12] Garcia, Paluri and Wu, Facebook. (2016). Under the hood: Building accessibility tools for the visually impaired on Facebook.
- [13] Eiselt and Sandblom. (2012). Operations Research: A Model-Based Approach.
- [14] Operational Research and Optimisation. (2016). Warwick University. Prof Alexander Tiskin.
- [15] About, Premier League. (2017).
- [16] English football league system, Wikipedia. (2017).
- [17] News, Sky Sports. (2016). Football League proposals for new League Three and 20 team league structure explained.
- [18] History of the Football Transfer System, Spartacus Educational. (1997-2016). Transfer System.
- [18] Transfer Window, Wikipedia. (2017).
- [19] Squads for 2016/17 Premier League confirmed, Premier League. (2016).
- [20] Player Profile, LFCHistory. Liverpool career stats for Fernando Torres.
- [21] Scout7 Changing the Game, Intel. (2015). Football scout and analysts take decision-making to the next level with big data, Scout7 and Intel technology.
- [22] DataScout, Opta Sports Pro. (2017). The full online football player recruitment tool.
- [23] Leo Chan, Football Performance Analysis. What are Performance Analysis and Match Analysis?

- [24] Gunjan Kumar. Machine Learning for Soccer Analytics.
- [25] Soccermetrics, GitHub. English Premier League 2011/2012
- [26] Code of Good Practice, British Computer Society.
- [27] Stats, Premier League. (2017).
- [28] Terms and Conditions, Premier League. (2017).
- [29] Joshua Uwaifo, Project Specification. (2016). See Appendix.
- [30] Joshua Uwaifo, Progress Report. (2017). See Appendix.
- [31] Football Statistics, WhoScored. (2017).
- [32] About Us, WhoScored. (2017).
- [33] Sky Sports Presentation, Presentation Archive. (2014 - 2017).
- [34] BigCommerce Support. (2003 - 2017). What is a CSV file and how do I save my spreadsheet as one?
- [35] Total Spotek 2. (2016). Premier League Prize Money Table 2015/16 Season (Confirmed).
- [36] The F.A. Premier League. Annual Report, season 2004/05.
- [37] Sporting Intelligence. (2016). Where the money went: Arsenal top PL prize cash table with £101m.
- [38] Telegraph. (2015). Premier League prize money - how much each club earns in 2014-15.
- [39] Sports Lens. (2008). 2007/2008 Premier League TV Revenue.
- [40] Daily Mail. (2013). Premier League prize money table for the final game of the 2013 season.
- [41] Premier League Players, Premier League. (2017).
- [42] Premier League Clubs, Premier League. (2017).

- [43] Fantasy Football (FF) Central, Neocities. (2017)
- [44] Tom M. Mitchell, Machine Learning. (1997).
- [45] Tavish Srivastava, Analytics Vidhya. (2015). How to avoid Over-fitting using Regularization?
- [46] Machine Learning, Linear Regression (OLS), Generalization, Overfitting and Model Selection. (2017). Warwick University. Dr Theo Damoulas.
- [47] Machine Learning, Decision Tree Learning: ID3. (2017). Warwick University. Dr Theo Damoulas.
- [48] Decision Tree - Regression, Saed Sayed. Standard Deviation Reduction
- [49] Hal Daume 3, A Course in Machine Learning. Learning Theory.
- [50] Metrics, Kaggle. (2017). Error Metrics for Regression Problems.
- [51] Ada Ivanoff, Sitepoint. (2000 - 2017). Design Minimalism: What, Why and How
- [52] Stack Overflow. (2017). Seamless Page Changing with AJAX.
- [53] Champions League Final Stage Player Statistics, WhoScored. (2017).
- [54] Premier League Table, Form Guide and Season Archives, Premier League. (2017).
- [55] Positions Guide: who is in a team?, BBC Sport. (2017).
- [55] Foundations of Data Analytics, Data Basics. (2017). Warwick University. Dr Florin Ciucu.
- [56] Wolfram Alpha. (2017). Convex Function.
- [57] Ron Kohavi, Stanford University. (1995). A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection.
- [58] Crummy. (2017). Beautiful Soup.
- [59] Full Stack Python. (2017). What is Python?