Applying Various Machine Learning Regression Models to Predict Future Performance in a Fantasy Football Environment

By Joshua Cesare Placidi
910252

Submitted to Swansea University in fulfilment
of the requirements for the Degree of Bachelor of Science

Swansea University
Prifysgol Abertawe

Department of Computer Science
Swansea University

May 7, 2020

Declaration

This work has not been previously accepted in substance for any degree and is not being con-currently submitted in candidature for any degree.

Signed ..... Joshua C Placidi ..... (candidate)

Date ..... 4/05/2020 .....

Statement 1

This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Signed ..... Joshua C Placidi ..... (candidate)

Date ..... 4/05/2020 .....

Statement 2

I hereby give my consent for my thesis, if accepted, to be made available for photocopying and inter-library loan, and for the title and summary to be made available to outside organisations.

Signed ..... Joshua C Placidi ..... (candidate)

Date ..... 4/05/2020 .....

Abstract

This project explores employing a purely statistical approach to selecting fantasy football teams. Various regression based algorithms are compared and contrasted, ultimately a linear regression approach is utilised to project individual player performances. A linear optimisation algorithm is then executed on player predictions calculating the optimal legal Fantasy Premier League team such that the sum of player points are maximised. Results generated by prediction models and the calculated optimal team for each gameweek are then published to a web application to be used to guide decision making by Fantasy Premier League users.

Acknowledgements

My deepest thanks to my family, particularly my mother and father without whom this project would not have been possible. I would also like to thank my tutors and lectures at Swansea University who have encouraged, guided and supported me throughout my studies.

Table of Contents

Chapter 1: Introduction

1.1 Background

Football is the most watched sport in the world. The game originated in England in the 12th century, with the establishment of formal leagues in the 1800s. Through the 20th century the sport became more commercial and avid supporters and enthusiasts of the game, determined to capitalise on their supposed unique knowledge, wagered money on the outcomes of games and the likelihood of individual events occurring. This proved to be lucrative for the betting industry and soon other businesses were keen to get in on the action. Teams and eventually players developed sponsorships with large corporations selling their images to the likes of car companies, banks and food producers [2].

Across the Atlantic in the United States of America, basketball was taking the country by storm. The first examples of fantasy sports appeared in the 1980s where individuals would coordinate their own independent basketball fantasy leagues. Fantasy sports allow participants to manage a team of players each of whom earn points when their real-life counterparts perform well. By the early 90s fantasy sports had become a global franchise with many newspaper companies creating their own privately run leagues. This new industry reached Europe and soon most major British publications featured a football version of the game [3]. In 1992 the English First Division became the English Premier League and in 2002 released their own modern digitised official version of a fantasy football system. Users could log in on a website at any time and monitor player performance and make changes to their team. With the end of each season the Fantasy Premier League (here on referred to as the FPL) introduced deeper interaction and became more sophisticated, user numbers rapidly increased and soon all other presences of fantasy football in England became obsolete to the official Premier Leagues application. In the 2018/2019 season the FPL reported a record-breaking 7 million participants in the league [1].

1.2 Fantasy Premier League

The modern rules of FPL are as such: users pick a team consisting of 15 players from a list of current premier league players each with an associated price tag, with the basic notion that better players who score more fantasy points have a higher price. Selected teams must abide by budgetary and formation restrictions preventing a team made up solely of the highest performing players in each position being selected, therefore encouraging careful planning and strategy. An FPL season mirrors the Premier League season and consists of 38 gameweeks, a gameweek simply refers to a short time period in which all teams player their respective fixtures, typically over a long weekend.

Figure 1: Example of a selected legal FPL team [1]

Conscious and unconscious biases play a huge part in team selection decisions, often users will overlook statistical data in favour of 'popular' assets or team allegiance [4]. For example devoted Everton F.C. fan will find it more difficult to bring in a Liverpool F.C. player due to the teams historic rivalry, despite Liverpool's recent dominance and clear statistical advantage. These biases ultimately have a negative effect on player selection and the resulting team will generally earn fewer fantasy points. The underlying motivation for this project is to minimise these human-prone biases and create a system which picks a high performing team based purely on statistical data.

1.3 Project Aims

Many approaches to tackling the influence of biases on decision making were considered, a system built on machine learning principles was adopted to respond effectively to the fundamental problem of identifying patterns and trends in larger datasets. More insight into the specifics of the design and implementation are detailed in their respective chapters accompanied by justification of decisions made and alternative methods that were considered. To ensure the stated problem was tackled effectively and in its entirety the following project aims were established:

I. From an input of raw player performance data, projections about future performance are made which satisfy a given accuracy measure.

Fulfilment of this first aim ensured a system was created in which predictions of player performance are possible. This aim also states that predictions should satisfy some arbitrary accuracy measure, this subjects the system to fall within a threshold enforcing predictions to be reliable to some degree.

II. From a list of players and predicted fantasy points approximate a high scoring team of players which obeys all FPL restrictions

The second aim of this project acts as a bridge between the first and third aim. The aims input is taken from the result of the first aim and output passed to the third. Satisfaction of this aim ensures implementation of a system that can convert a list of players and predicted performance into a FPL legal format. Satisfaction of aim II allows for direct comparison between the systems selection of players and that of a human user, success of this is vital to the evaluation of the system.

III. Create a public web application, allowing users to view the results generated by the systems defined in Aim I and II.

The final aim is not strictly system-based but instead involves the community aspect of fantasy sports. Since the web-based FPL version of the game, many online communities have been created and devoted to the discussion and sharing of tactics. All referenced projects directly tied to FPL have created their work with public access in mind and have used this to great success in gaining meaningful feedback. Aim III will be accomplished by creating a simple web application allowing interaction with the data generated through the development of this project.

Referring back to the original stated problem of "creating a system to select high performing FPL legal teams in which human bias is minimised" the satisfaction of the above aims have bounded the final system to each address individual parts of the problem: Aim I ensures a system which can select players with some underlying accuracy and logic is possible, Aim II guarantees a legal FPL team is selected and aim III while not directly tackling an aspect of the problem allows the results of the prior aims to be interacted with by FPL participants.

Chapter 2: Related Work

Extensive research was conducted throughout the development of this project, possible approaches and their results were gathered from a range of sources including but not limited to academic papers, books and self-published articles. These sources provided great insight into different ways to execute areas of the project to achieve meaningful results. Fortunately, there is a range of papers and articles devoted to analysing past sport data to predict future performance, many of these utilise fantasy sport environments to quantify performance and measure accuracy. In this chapter, research that has directly influenced this project, by either direct inclusion or just consideration of their methodology, is discussed in detail. Each subsection concerns itself with an individual area of focus, discussing the significant relevant material.

2.1 Similar Projects

Initial reading was first focused on analysing possible ways to approach a sport data projection system. A plethora of publicly available projects, both personal and academic in nature, have applied many different approaches in an attempt to produce a system capable of analysing sport data to identify trends and patterns. Research was confined to material specifically targeting fantasy sport applications. The benefits of individual designs and implementations are discussed in detail.

2.1.1 "Interactive Tools for Fantasy Football Analytics and Predictions using Machine Learning" by Neena Parikh [5]

Parikh's 2014 paper aims to predict the fantasy score of players in the NFL by analysing their performance data. A focus is applied to micro-level statistics as apposed to the typical macro data analysis which at the time was used by many existing statistical tools. A range of machine learning techniques are explored to seek the optimal performing model which produces the least error in predictions. The focus of the paper is to generate data that can be displayed in an interactive tool to assist fantasy football participants in team management.

A classification approach was taken using both k-means clustering and support vector machines. Both models were trained to group players into similarly performing clusters. The player base was split by position such that individual models could be trained on players with similar play styles, with the idea that a model trained on just quarterbacks would produce more accurate predictions than a general model trained on all players. This method allows for the optimisation of models on specified subsets of players by fine tuning parameters and supplied features.

Results were compared using established popular fantasy prediction providers as benchmarks. Parikh's aim was to evaluate each of the models individually, attempting to out-perform their positional benchmarks. Support vector machine models were found to outperform their k-means positional equivalents, achieving better prediction performance than their relevant benchmarks for all positions but quarterbacks.

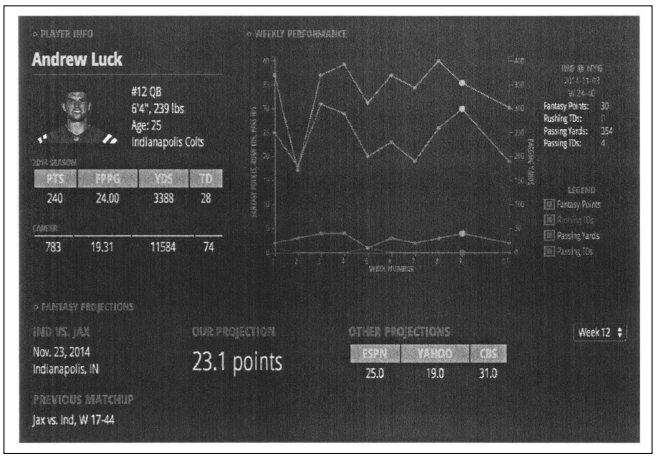


Figure 2: Figure taken from Neena Parikh's 2014 paper showing an example of how individual player performance is displayed in her developed web application.

A web application was developed using the Django web framework and the D3.js JavaScript library. Model results were used to populate the web application with detailed performance statistics for individual players, allowing results to be viewed and compared. Calculated performance projections are displayed in combination with projections from other leading fantasy analytic tools.

2.1.2 "Time Series Modeling for Dream Team in Fantasy Premier League" by Akhil Gupta [6]

Recurrent Neural Networks are used by Akhil Gupta in his 2017 paper to create time series predictions of FPL player points in the 2016/17 Premier League season. Once player predictions had been made, linear programming is employed to find the optimal arrangement of players such that the sum of their points is maximised while obeying team restriction rules. Gupta experimented with a non-linear approach to modelling player performance and was able to achieve consistent results. Data from different seasons was combined to create detailed player profiles, players who featured in multiple seasons had all their associated data grouped so, in theory, more accurate time series predictions could be made.

The developed system was divided into three broad phases, first historical data was aggregated and formatted, then time series models were used to project points for individual players. Finally linear optimisation was implemented to generate a projected high performing FPL legal team. Team constraints were modelled as linear relationships, the objective function can then be defined as the sum of the selected players points. The PuLP library in Python 2.7 was used to find a solution which maximised the objective function while obeying the defined constraints. Gupta was able to consistently find the optimal team configuration, celebrating the efficiency of the algorithm. The final system produced a team for each gameweek of the 2016/17 season exhibiting an error of only 87 points over 38 selections. The errors in predictions were largely due to a selected players unexpected absence from a fixture resulting in 0 points earned.

2.1.3 "Predicting Optimal Game Day Fantasy Football Teams" by Glenn Sugar and Travis Swenson [11]

Sugar and Swenson conducted an experimental study comparing different regression based machine learning algorithms to determine an optimal model to predict NFL fantasy points. They compared ridge regression, bayesian ridge regression and elastic net regression, each minimise a cost function to plot a prediction plane through multi-dimensional data. A similar data formatting strategy to Parikh's was incorporated, splitting the player base into subsets to which individual models would be applied [5]. Using the predefined positional split imposed by Pro-Football-Reference (an NFL data gathering website) they were able to achieve results which rivalled projections from the multi-billion dollar company Yahoo.

An efficient data structure was used to split features into three basic categories: career features, current features and recent history. Career features recorded the running averages of the player over their entire career. Current features gave information about the current game they were attempting to predict and consisted of metrics such as opposition difficulty, whether the game was at home or away etc. Recent history was built from statistics from the n most recent games. The value of n was experimented with through trial and error to determine the optimal representation of recent games.

Comparing the root mean squared error (RMSE) of each model on each of the positional subsets they determined the optimal model for each position to implement into the final system. FanDuel, the fantasy platform they tested their system on, applies team restrictions similar to the FPL. After generating predictions they found the optimal team by solving the Markowitz Portfolio Optimisation Problem, generating a team with minimal variance.

They compared the results of their models to the NFL support tool produce by Yahoo, one of the most used fantasy tools at the time, and were able to better their predictions in 2 of the 5 recognised player positions with the other 3 being within 4% of Yahoo's predicted score. In future works Sugar and Swenson discuss alternative methods of clustering players, grouping by similar play-styles as apposed to straight positional classification. This would, when provided with enough data, create more accurate models as the play-styles in the positional classification can vary greatly resulting in different relationships between features and predictions.

2.1.4 “Competing with Humans at Fantasy Football: Team Formation in Large Partially-Observable Domains” by Tim Matthews, Sarvapali D. Ramchurn and Georgios Chalkiadakins [8]

Matthews, Ramchurn and Chalkiadakins paper, featured at the 26th annual AAAI conference, explores the application of applying bayesian reinforcement learning to football player characteristics with the aim of building a competitive fully automated fantasy football manager. Their system competed over the full 2010/11 season achieving a final rank in the top one percentile of the 2.5 million participants.

Using a belief-state Markov Decision Process (MDP) to model the FPL sequential team formation problem, they captured uncertainty in player contributions. Performable actions were defined as the selection of a valid team such that each selected player is a subset of the available player base and the team formation of selected players obeys the FPL restrictions. After thorough experimentation and evaluation a bayesian q-learning algorithm was used to handle the generated uncertainty as they found it to outperform other uncertainty-agnostic approaches on the 2010/11 FPL dataset, using mean final score to characterise performance. Their final system placed in the top 1% of users on average and in its best case, where 2222 points were obtained, it placed within the top 500.

2.1.5 Machine Learning Applications in Fantasy Basketball By Eric Hermann and Adebia Ntoso [7]

In their 2015 paper Eric Hermann and Adebia Ntoso assess various linear regression machine learning techniques in their ability to model fantasy basketball player predictions. Implementing a beam search team selection algorithm they were able to calculate highly performing teams consistently achieving more points than the average player.

A large focus of their study is on model data presentation, they employ a similar approach to Sugar and Swenson where aggregated statistics from previous games are used as features of the prediction algorithms. They found using the cumulative summed relevant statistics from the previous 5 games resulted in the highest model accuracy. They also found increasing the size of the feature set reduced test error significantly, and opted to even include features outside of the relevant actions a player earns points through. They combined recent player statistical features with game specific data such as the opponent’s recent win percentage and whether the game was at home or away.

After player predictions were calculated, they passed data to a beam search algorithm tasked with assembling the optimal selection of player such that their predicted score was maximised while obeying the team restrictions of their chosen fantasy basketball application. Although they did find success with the algorithm they noted that it sometimes took up to several minutes for selections to be made due to the $O(n(kb)log(kb))$order of time complexity, where b is the branching factor and k is a score threshold parameter used to vary the size of the search space.

2.1.6 “Multi-stream Data Analytics for Enhanced Performance Prediction in Fantasy Football” By Nicholas Bonella, Joeran Beel, Seamus Lawless and Jeremy Debattista [9]

Bonella, Beel, Lawless and Debattista’s 2019 paper, featured in the 27th AIAI Irish conference, explores a different approach to fantasy football predictions than has previously been discussed. The study aims to solve the problems present when calculating predictions using just historical statistical data by incorporating human feedback into their model. A novel approach is presented using a combination of historical data, betting market analysis and opinions of the general public and experts gathered via web articles and forum posts. When tested on the English Premier League 2018/19 season they managed to achieve a rank of 30,000 out of 6.5 million, ranking within the top 0.5% of all participants.

Implementations of many different machine learning algorithms were considered and tested but ultimately they concluded a gradient boosting machine model produced the best results. A combined dataset consisting of betting odds, human-centred data and historical statistics was constructed, models were then trained and optimised on this data.

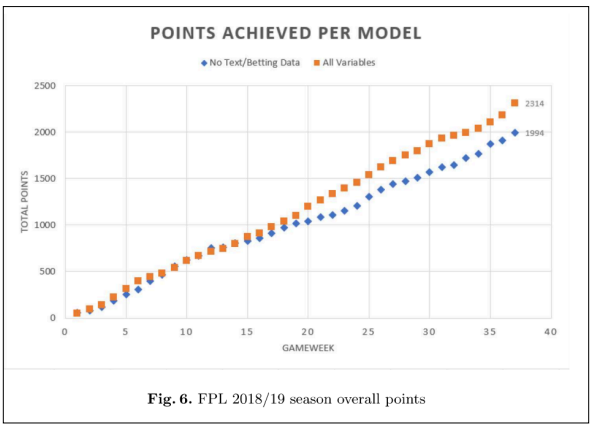


Figure 3: Figure taken from a 2019 paper by Nicholas Bonella, Joeran Beel, Seamus Lawless and Jeremy Debattista. Visualises the difference in points earned between a purely statistical model (blue) and a model also using betting and human-centred data (orange)

Figure 3 (taken from their paper) shows a comparison of results between the purely statistical model and their optimised final system. A significant improvement was found with an average gain of 10 points per gameweek over the statistical model.

2.2 Prediction Models

2.2.1 Viable Algorithms

Each of the papers discussed previously offers a different approach to predicting player performance each with varying degrees of success. Further research into prediction models was conducted to determine the designs that should be considered for this project. A published study[1] by Zefeng Zhang, Donny Chen, Eric Lehman and Philip Rotella compares various regression, random forest and gradient boosting models analysing the accuracy in their predictions when applied to NFL fantasy data. They individually tested models on different defined player positions and found ridge regression on average to outperform the other models. A article by Paul Solomon[2] describes a similar application he developed using neural networks with which he was able to rank in the top 1% in three of the past seven FPL seasons, he comfortably achieved top 10% in the other four. Considering all research conducted into predicting player performance, it became apparent that a machine learning approach should be employed due to the previous success others have achieved with their implementations. Machine learning algorithms aim to identify trends and patterns in large datasets, this applies perfectly to our system where patterns were attempted to be found between a player performing well and their previous performances.

Both supervised and unsupervised algorithms were appropriate for this project. The underlying aim of supervised machine learning algorithms when training a model is to identify patterns and trends in data to find a consistent transformation accurately mapping an input to its known output. This mapping can then be further applied to inputs with unknown/undisclosed outputs in an attempt to predict its true output. An unsupervised algorithm takes a set of data and searches for similarities in which the data elements can be grouped [12]. Both methods have previously been applied successfully. Neena Parikh used the unsupervised algorithms k-means clustering and SVMs to successfully predict optimal NFL team selections, achieving results with accuracy close to that of leading fantasy experts [5]. Glenn Sugar and Travis Swenson used supervised regression algorithms to compete against the score predictions generated by fantasy analysis titan Yahoo [11]. Ultimately the decision was made to explore the viability of linear regression based algorithms in a similar manner to that of Sugar and Swenson, and Hermann and Ntoso [7].

[1] Project published by GitHub user zzhangusf and available to be viewed at: https://github.com/zzhangusf/Predicting-Fantasy-Football-Points-Using-Machine-Learning

[2] Article written by Paul Solomon available at: https://medium.com/@sol.paul/how-to-win-at-fantasy-premier-league-using-data-part-1-forecasting-with-deep-learning-bf121f38643a

2.2.2 Data Preparation

The accuracy of a prediction model is directly proportional to the quality of data provided to it, even the best prediction model run on low quality data will produce low quality predictions. In the data sourcing chapter of Neena Parikh's 2014 paper adequate ways of presenting data are discussed, the approach taken involved splitting the player data by field position i.e quarter backs, wing back, running backs etc are grouped separately [5]. Models trained and tested on this split data should then in theory be more accurate due to the different play styles inherit to each position.

Sugar and Swenson experiment with different presentations of previous performance statistics. They used data from *n* previous games as features to predict a performance for a given gameweek. Changing the value of *n* produced different model accuracy with they tested with through trail and error until an optimal value was found. The context of the data presented to the model is also important, consider a player, *P1* who scores 3 goals against the worst team in the league and another player, *P2* who scores 2 goals against the best team. Disregarding the opponent, *P1* will be interpreted to have performed better but in actuality *P2* has accomplished a more difficult feat and so his performance should be weighted higher. This is the system integrated in Akhil Gupta's, amongst others, projects where attempts are made to give statistics some form of context. Outlier statistics must also be accounted for, if measuring goals as goals per minute, a substitute who comes on in the last 5 minutes of a game and scores will have a hyper inflated metric not because they score a large quantity of goals but because they play fewer minutes. These and many other data formatting considerations were made and are discussed in the design and implementation chapters of this document.

2.3 Team Selection

FPL imposes restrictions on the way teams can be formatted, it is therefore necessary to create an algorithm that from a list of players and their predicted points can generate a high performing FPL legal team. There are different ways to produce a legal team from a selection of players. Tim Mathews, Sarvapali D. Ramchurn and Gerogios Chalkiadakins, in work featured at the 26th AAAI conference, suggested modelling team selection as a multidimensional knapsack problem, where players are items with their costs as modelled weight, for each constraint that needs to be enforced a new dimension is added to the algorithm [8]. Hermann and Ntuso implemented a backtracking and beam search method, however they encountered long runtimes since their solution was of time complexity $O(n^x)$ where x represents the number of available player positions in the team. They applied their algorithm to a fantasy basketball application where a team consist of 8 players, using a player pool size of 240 players they found it took several minutes for solutions to be found. Akhil Gupta found success in his linear programming based design, enabling him to consistently find optimal teams in relatively small timeframes. Both a knapsack solution and linear programming were researched further, assessing their viability for this project and their performances.

2.3.1 Knapsack Solution

David Pisinger's "Algorithms for Knapsack Problems" supplies detailed information and solutions to a large variety of knapsack problems. The problem of subset of players subject to contains from a list of players and their associated performances can be modelled as a 1-0 knapsack problem. Players are modelled as items which can either be selected (1) or not selected (0), fractions of a player are not possible. The knapsack problem is NP-complete, it can be solved through brute force search but there isn't a known algorithm that can correctly solve the problem in polynomial time. However a pseudo-polynomial solution can be achieved using dynamic programming. Dynamic programming is the process of sub diving a problem into smaller sub-problems recursively and then combining the solutions of those sub-problems to solve the larger problem. A multi-dimensional solution is needed to enforce all the different restrictions the FPL enforces on team structure.

2.3.2 Linear Programming

Gupta Akhil uses linear programming to find the optimal fantasy team from a set of players and their projected performances. Linear programming uses a simplex methodology to find the best outcome of a mathematical model whose constraints are represented by linear relationships [13]. It aims to maximise (or minimise) an objective function while abiding by a set of defined constraints. This can be applied to team selection by having the sum of the players points as the objective function which needs to be maximised and

then apply constraints as linear relationships in which the outcome must fall within a set threshold. Linear programming will consistently find the best solution from a list of players such that the produced team obeys all FPL restrictions.

Chapter 3: Design

3.1 System Design

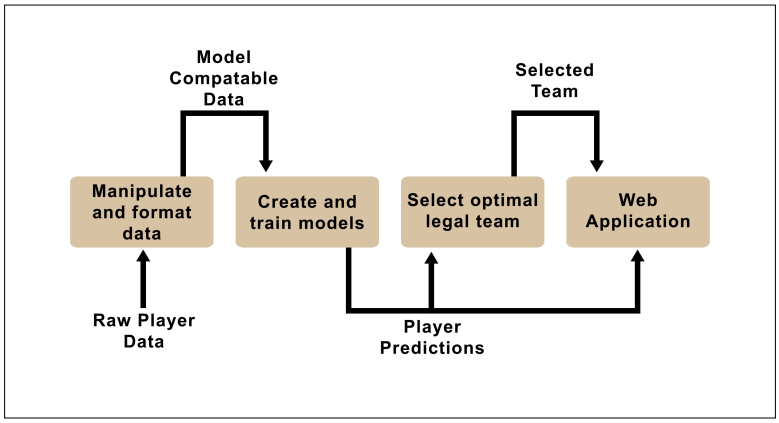


Figure 4: Overview of the pipeline architecture of the system

The system was designed to operate as a pipeline of modules, each taking data from their predecessor and outputting to the next. Figure 4 shows a broad outline of the modular design and the flow of data through the system. Adopting a modular design architecture breaks the overarching problem of trying to build a fantasy football team selector into individual functional parts. Raw player statistics are inputted into the system and an FPL team and individual player predictions are outputted to a web application. The design of each module is discussed individually in this chapter in chronological order from data input to web application design.

3.2 Data

Raw player data will need to be formatted and manipulated into a compatible form such that the prediction models effectively be can run on them. Sugar and Swenson's rolling data structure where the *n* most recent performance statistics are used as features is incorporated in this design. Varying the value of *n* creates models that produce different predictions, the optimal value of *n* for given data can be determined through trial and error. Table 1 shows an example of how a rolling dataset using *n=2* previous games can be structured, note that in a rolling dataset the first sequential *n* entries for a player are rendered unusable by models because *n* previous games are required to calculate rolling aggregated features.

player_id	gameweek	goals	assists	prev_goals_sum	prev_assists_sum
215	1	0	1	-	-
215	2	1	1	-	-
215	3	1	0	1	2
215	4	0	3	2	1
215	5	0	0	1	3

Table 1: Example of a rolling dataset of n = 2 previous games considered

A large feature set approach is used where models are provided access to the full range of available statistics; Initially it may seem pointless to use a goals conceded metric for projecting performance for forwards since a forwards FPL points score has no direct link to the number of goals they concede, however these features are included to allow models to explore indirect relationships. An opponent metric was used to measure the difficulty of a fixture, this gives the models a further indication into expected points since a player is likely to perform better against a worse team.

3.3 Prediction Models

3.3.1 Player division

Datasets were divided by player positions, different algorithms could then be trained and tested on each positional dataset with the best performing model implemented into the final system. Similar projects such as Neena Parikh's, and Hermann and Ntuso's were able to improve model accuracy using this data split and even concluded that error could be further reduced by subdividing positional subsets into play-style subsets [5, 7]. Play-style subsets could see improvement in projection accuracy, however this approach was not adopted here as certain positions such as forwards and goalkeepers already have smaller player pools with limited entries, applying further division then just a positional split would result in less than ideal data available to sufficiently train some models. Ultimately the player base was divided into the four FPL pre-defined positions in which individual model testing of different algorithms was conducted.

3.3.2 Linear Regression

Linear regression, ridge regression, lasso regression and elastic net regression were all tested with the best performing algorithm for each position implemented. These algorithms were chosen due to the linear nature of the data and the success achieved by Hermann and Ntuso, and Sugar and Swenson in creating accurate projections utilising linear models [7, 11]. The algorithms are all forms of linear regression and behave similarly, attempting to plot a plane through data such that a cost function is minimised. Linear regression minimises the sum of the squared residuals (SSR) between projected data points and their true value. Ridge regression builds on linear regression introducing slight bias to the cost function by adding a penalty term:

$$SSR + (\lambda \times Gradient^2)$$

λ is a coefficient that controls the influence of the penalty term $Gradient^2$, larger values of λ introduce more bias to the model making projections less sensitive to the explanatory variables. Lasso regression works similarly to ridge regression, it differs by having the ability to remove all influence from explanatory variables whereas ridge can only reduce their coefficients to be asymptotically close to zero. The cost function for lasso regression is:

$$SSR + (\lambda \times |Gradient|)$$

Similarly to ridge, in lasso regression λ controls the influence of the penalty function with larger value resulting in larger bias. Elastic net combines both ridge and lasso regression with the cost function:

$$SSR + (\lambda 1 \times |Gradient|) + (\lambda 2 \times Gradient^2)$$

Values of $\lambda 1$ and $\lambda 2$ can be set individually to alter the influence of the different penalty terms. Linear regression can be susceptible to multicollinearity, where multiple explanatory variables are highly linearly related. Ridge, lasso and elastic net regression introduce slight bias in return for a reduction in variance to reduce their susceptibility to the effects of multicollinearity, their cost functions reduce the influence such linked variables have on projections.

3.4 Team Selection

3.4.1 0-1 Knapsack Solution

The problem of finding the optimal configuration of players in an FPL legal form can be modelled as a 0-1 multi-dimensional knapsack problem and has been used in previous FPL applications. An AAAI conference featured paper by Tim Matthews, Sarvapali D. Ramchurn and Georgios Chalkiadakins used a knapsack base solution to obtain optimal performing teams of FPL players [8]. The basic knapsack problem arrises when trying to calculate the optimal way to organise items in a knapsack with a given weight limit where each item has an associated weight and value, the total value of all items is to be maximised while not exceeding the weight limit. This can be applied to the team selection problem by modelling players as items with predicted points as their value and their cost as their weight. The solution takes the form of a 0-1 knapsack solution where a player can either be selected (1) or not selected (0), i.e 50% of a player cannot be selected. In order to account for all necessary constraints a multi-dimensional knapsack approach was considered, for each constraint applied another dimension must be added to the algorithm. There are many possible ways of designing a knapsack solution algorithm as discussed in great detail in David Pisgners "Algorithms for Knapsack Problems" [10]. For this project a dynamic programming approach was explored.

3.4.2 Linear Optimisation

Linear optimisation is a method for finding the optimal solution to a model with a set of constraints that must be satisfied. Requirements are represented as a set of linear relationships where values must remain within a defined threshold. Linear optimisation can be used to model FPL team selection by defining the objective function to maximise as the sum of player points in the selected team, constraints can be applied by creating linear functions that must return true. For example, the sum cost of the selected players must be equal to £100M or below. Linear optimisation has been implemented in many FPL prediction oriented projects such as Akhil Gupta's 2017 paper exploring time series modelling and Joseph O'Connor's web article[3] describing in detail his successful application of linear programming to FPL datasets [6].

3.5 Web Application

The purpose of the web application is to enable FPL users to publicly view the results of prediction models and team selections for a given gameweek. A minimalist design methodology was adopted where the interface would simply display results in a readable and understandable manner without overcomplicating the design.

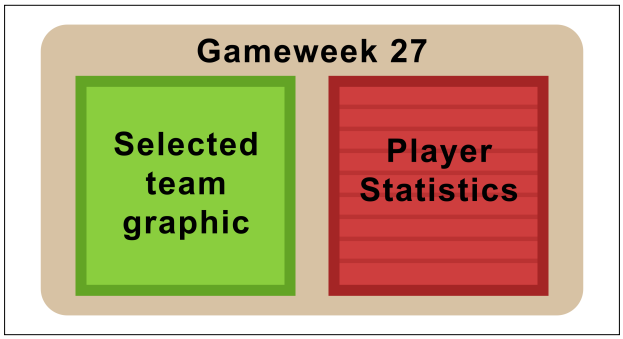


Figure 5: Simplistic interface layout designed for the web application.

Figure 5 shows the simple design layout in which data corresponding to a given gameweek is displayed. The user is able to view the selected team in an FPL familiar graphic on the left and more detailed statistics about the selected players on the right hand side. A continuous chronological blog format was designed where information from the most recent gameweeks is presented at the top of the page, the user can then scroll down to view data from past gameweeks. Each graphic is accompanied by a short paragraph describing the performance of the model for that week.

[3] Joseph O'Connor's web article can be found at: https://medium.com/@joseph.m.oconnor.88/linearly-optimising-fantasy-premier-league-teams-3b76e9694877

Chapter 4: Implementation

Individual elements of the system were implemented separately and then combined through passing data from the previous module into the next. This allowed for modular testing in which components could be individually implemented and then internally tested to ensure they worked correctly. Once each module's functionality is correct linking the system together becomes simple, and because of the structure the source of errors is easily determined.

4.1 Data

4.1.1 Data sourcing

The official FPL API allows access to a wealth of data relating to individual players, teams and fixtures[4]. A multitude of statistics quantifying player performances are updated weekly; this is useful for getting the latest information for each player, but statistics recording performance in previous gameweeks are overwritten making it impossible to build a detailed time series profile of a player. The Varstaav FPL repository[5] addresses this issue by logging FPL API data each week into conveniently formatted files. Data from all available seasons (2016/17 to 2019/20) was used, the primary files accessed were the merged gameweek and raw player data files for each season. The gameweek files contained the game by game breakdown of a player's performance which will be used to quantify performance. The raw player data files contain the necessary information about each player such as position, the team they play for, their FPL price etc. A combination of these two types of file provides all the information necessary to build the system. The Pandas library was used in python 3.0 to read the files hosted in the repository.

	name	total_points	opponent_team	assists	goals_scored	influence	threat	was_home
232	Jamie_Vardy_166	2	20	0	0	6.8	4.0	True
760	Jamie_Vardy_166	2	6	0	0	0.0	10.0	False
1289	Jamie_Vardy_166	8	15	0	1	33.6	27.0	False
1821	Jamie_Vardy_166	16	3	1	2	87.6	45.0	True
2357	Jamie_Vardy_166	2	12	0	0	1.2	4.0	False

Figure 6: Subset of statistics available for Leicester player Jamie Vardy

Figure 6 shows a subset of the features available for each gameweek. Influence and threat are examples of meta data generated by the FPL that provides insight into a players performance. These statistics quantify aspects of the game that traditional football statistics often overlook, for example threat quantifies the danger a player posed to their opposition's goal. Players can have unfortunate games where despite performing well, traditionally tracked statistics can say otherwise. These meta statistics create a broader picture of a player's performance rather than just the goals he did or didn't score. FPL user data can also provide useful insight into a players performance, for each gameweek statistics about the number of users who selected a player for their own team can give the system a social aspect. There are many influences off the pitch that can effect performance on the pitch, complications in personal life, transfer rumours and falling-out with teammates. These influences, while not reflected in the previous performance of a player, will often be considered by FPL users and can lead them to rightfully transfer in/out a player from their team. By considering data of these different natures a more detailed player profile can be created in an attempt to more closely model performance than would be possible by just considering traditional football quantifiers.

4.1.2 The FPL Points System

To predict fantasy performance, one must first understand the actions that players are awarded fantasy points for. In FPL players belong to one of four positional categories: goalkeepers (GK), defenders (DEF), midfielders (MID) or forwards (FWD). Each position is awarded points differently as described in Table 2.

[4] FPL data can be accessed directly at: https://fantasy.premierleague.com/api/bootstrap-static/

[5] GitHub repository hosted at: https://github.com/vaastav/Fantasy-Premier-League

Action	Goalkeepers (GK)	Defenders (DEF)	Midfielders (MID)	Forwards (FWD)
For playing up to 60 minutes	1	1	1	1
For playing 60 minutes or more	2	2	2	2
For each goal scored	6	6	5	4
For each goal assisted	3	3	3	3
For keeping a clean sheet	4	4	1	-
For every 3 saves made	3	-	-	-
For each penalty saved	5	-	-	-
For each penetaly missed	-2	-2	-2	-2
For every two goals conceded	-1	-	-	-
For each yellow card	-1	-1	-1	-1
For each red card	-3	-3	-3	-3
For each own goal	-2	-2	-2	-2
Bonus Point System [6]	1-3	1-3	1-3	1-3

Table 2: FPL point allocation system by position

It is important to understand the conditions in which players earn points but indirect influences should not be disregarded. By studying the points allocation system it may seem unnecessary for goals conceded to be a consideration for a forward's points, since there is no direct relationship. However it is possible that an indirect relationship exists between the two and so should be included to allow the prediction models to identify patterns if they exist.

4.1.3 Data manipulation

As well as past performance data current, fixture metrics were used, namely an indicator describing the location of the match (home or away) and an opponent difficulty rating. Home-field advantage is a highly significant influencer of the outcome of football matches and player performance, so much so that major tournaments such as the Champions League require teams to play twice at each stage, once at their stadium and once at the opponents. Much research has been conducted into the subject area and relationships to varying degrees have been proven [13]. The location of a given match is therefore an important metric to consider.

The difficulty of an opponent is another essential feature to measure. Playing a weaker team will generally produce greater performances, so an opponent difficulty variable was used to attempt to quantify the 'strength' of a team. Difficulty was set to the sum of the points the team earned in the season being considered. So, for example, throughout 2018/19 seasonal data any player facing Manchester City would have an opponent difficulty rating of 100 (the points tally Manchester City finished on) associated with that gameweek. Similarly players facing West Bromwich Albion in the same season would have an opponent difficulty rating of 31. While this system fails to account for individual spells of good and bad form all teams experience throughout a season, it provides a generalised context to the difficulty of a fixture.

Final datasets were constructed for each of the four available seasons where the number of previous games aggregated for a player's performance was varied and combined with fixture data. The format of these datasets allow for models to be trained and tested so that they may predict future player performance with only past performance and fixture information. Using rolling datasets, each aggregating statistics from a

[6] Bonus points are awarded to the player with the highest BPS (Bonus Points System) score, calculated from a wide range of metrics, generally perceived to be the best players of the pitch. The player with the highest BPS score earns 3 points, second heights earns 2 and third earns 1.

different number of performances, trial and error experiments were conducted to determine the optimal number of previous performances to consider to generate the most accurate prediction model.

Once raw files had been read and combined, manipulations were performed to get the necessary format to train prediction models. The format adopted in this project takes inspiration from Sugar and Swenson's paper where the combination of statistics from the previous *n* games is used as explanatory variables to predict performance of a future game [11]. This creates a rolling dataset of values which generates a format on which models can be trained. Rolling datasets were created with varied *n* values in the range of 1 to 9 previous performances considered. The upper limit was set at 9 as models trained using a rolling dataset of *n = 9* can only make predictions when 9 previous games have been played, just under a quarter of the Premier League's 38 game season. Requiring more than a quarter of the season to be played before any predictions can be made reduces the usefulness and application of the system so a limit of *n = 9* was set. The Pandas and NumPy libraries were used in Python 3.0 to create these rolling datasets.

```
# Set the statistics for a player for a gameweek equal to the sum of the statistics from the n_prev_gws gameweeks
df = df.groupby(['player_id']).rolling(n_prev_gws).agg({'minutes':np.sum, 'bps':np.sum, 'influence':np.sum,
                                                        'threat':np.sum,'ict_index':np.sum, 'creativity':np.sum,
                                                        'yellow_cards':np.sum, 'red_cards':np.sum, 'selected_by':np.sum,
                                                        'transfers_balance':np.sum, 'goals_scored':np.sum,
                                                        'assists':np.sum, 'points':np.sum, 'value':np.sum,
                                                        'saves':np.sum,'goals_conceded':np.sum,'clean_sheets':np.sum}).shift(1).fillna(0)
```

Figure 7: Functions used to create rolling datasets with Pandas in Python 3.0

Figure 7 shows the functions used to calculate the rolling datasets, where the variable *n_prev_games* is the number of previous performances to aggregate. Each variable is individually set to the sum of its predecessors to allow for individual alterations. For example, the minutes variable could be separately set to the median of its predecessors instead of the sum. However after experimenting with different representations a universal sum of previous values was adopted.

player_id	GW	second_name	points	opp_diff	was_home	selected_by	minutes_sum	goals_sum	assists_sum	points_sum	threat_sum	influence_sum
166	11	Vardy	6	51.10	False	2522878.0	270.0	4.0	1.0	30.0	124.0	145.8
166	12	Vardy	12	54.29	True	3661521.0	270.0	5.0	1.0	34.0	154.0	177.4
166	18	Vardy	9	77.36	False	10548327.0	270.0	3.0	1.0	23.0	204.0	125.8
166	19	Vardy	2	107.45	True	10848653.0	270.0	3.0	1.0	27.0	196.0	134.4

Figure 8: Performance statistics for Jamie Vardy in the 2019/20 season using a rolling data of n_prev_games = 3

Figure 8 shows another example of a subset of Jamie Vardy's statistics from the current 2019/20 season, this time with the rolling data function applied (*n_prev_games* = 3) and the fixture specific information added. These entries show a good form spell from Vardy, where his *goals_sum* and *points_sum* metrics were very high. Vardy returned a healthy number of points in all but gameweek 19 (GW=19) where he faced Liverpool F.C. who in the current 2019/20 season have so far had the best performance of any team in the history of the premier league. They are projected to finish the season on a record breaking 107.45 points which is shown in the *opp_diff* and perhaps explains the lower points score Vardy achieved.

4.2 Prediction Models

Four different linear regression based prediction algorithms were implemented and tested: multiple linear regression, ridge regression, lasso regression and elastic net regression. As explained in *Design: 3.1.2 Linear Regression,* all considered algorithms find a plane of best fit through dimensional data, each with a different cost function associated with it. The different cost functions lead to different coefficients being assigned to the explanatory variables. All models were implemented using the Scikit-Learn Python library using the Pandas library to structure dataframes. Scikit-Learn allows for linear models to fit to data by specifying a set of features and a target variable to predict. The models then attempt to accurately model the relationship between the set of features and the target variable, this is done by applying coefficients to each feature which weights its influence on the predicted variable. Initial experiments were conducted training the models on a combined-season rolling dataset of *n_prev_games = 5*. The coefficients each model applied to the explanatory variables was then compared to see which variables were being prioritised by each model.

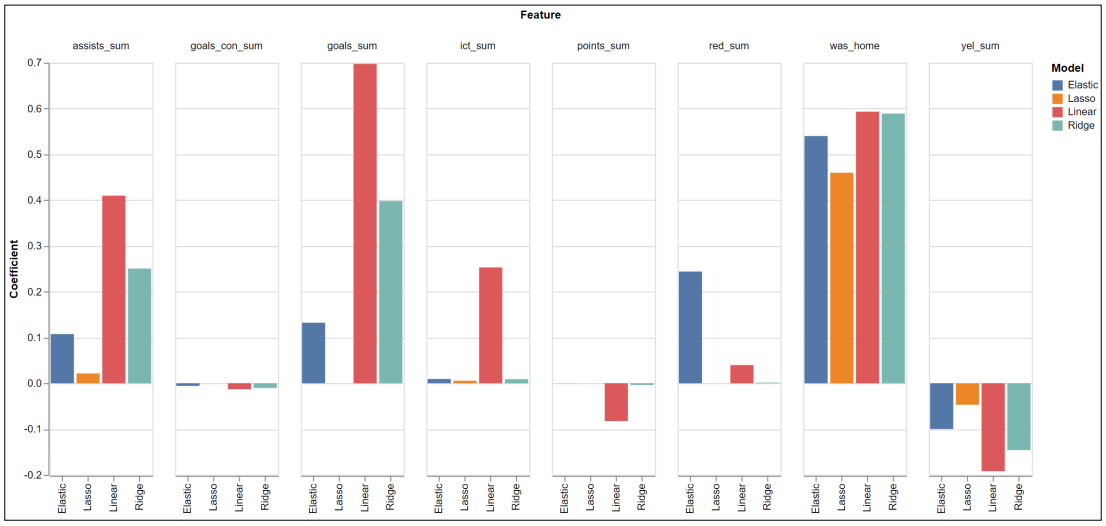


Figure 9: Comparison of a subset of calculated feature coefficients from MID rolling dataset with n_prev_games = 5

Figure 9 shows the different feature coefficients applied by each model. All models applied a similar weight to the *was_home* feature finding a positive relationship between a home performance and points earned. Vast differences can be seen for *assists_sum* and *goals_sum*, the linear model applied larger emphasise to these features whether as elastic net and lasso find little to no correlation. These results are not significant, nor were they factored into any of the design decisions but they do show that different linear based regression algorithms can find largely different ways to model the same data.

4.2.1 Comparing Models

For each FPL position the single model that produced the least error for its respective data was chosen to implement into the final system. Large scale simulation tests were run for each model on positional data with varied rolling datasets. K-fold cross validation of 9 parts training to 1 part testing was employed to randomly select unique training and testing subsets from the combined seasons master dataset. Each model then calculated points predictions for each player which were then compared to the real number of points the player earned for the given gameweek. Each simulation consisted of thousands of training and testing cycles each using newly randomised cross validated data. The mean of the errors from these cycles was then used to represent the accuracy of the model for the given input data. For each position all four models were run on 5000 cycles on each of the 9 rolling datasets for that position.

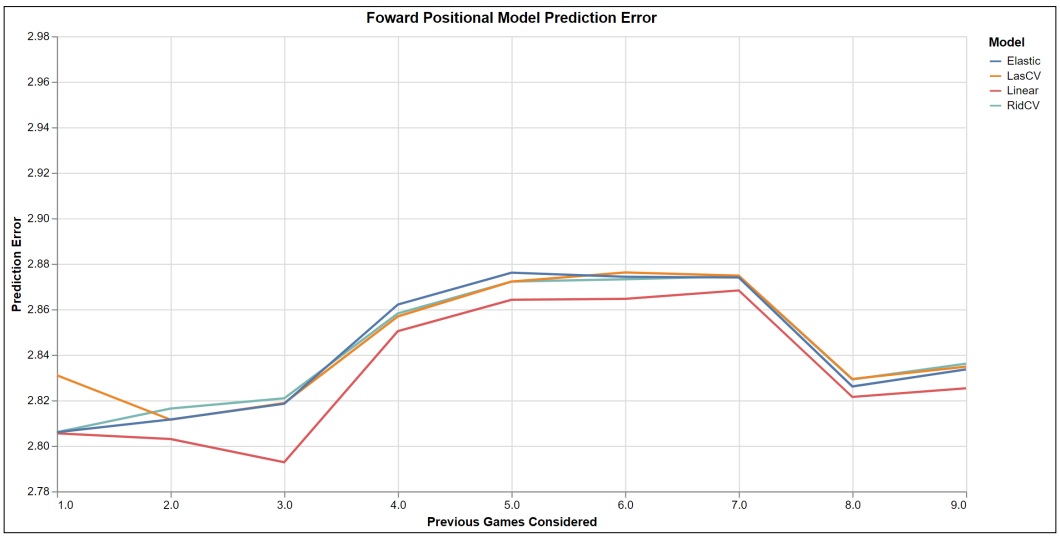


Figure 10: Calculated prediction error for each model when applied to forward positional data

Figure 10 shows the calculated error for each of the models trained and tested of varying forward rolling datasets. We can see that variation in error for all the models is similar with the linear model dominating all others in performance and therefore was chosen as the prediction model for forwards.

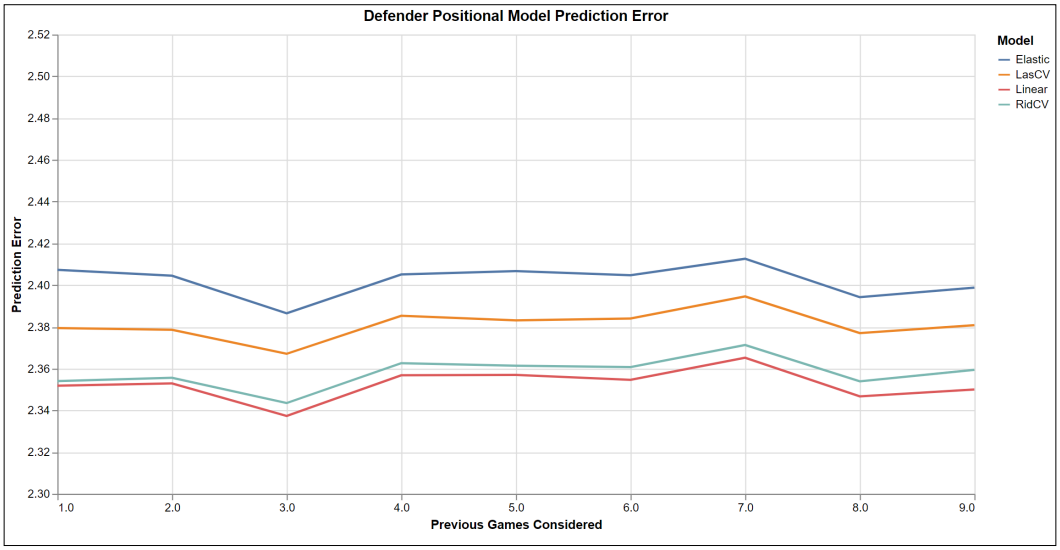


Figure 11: Calculated prediction error for each model when applied to defender positional data

Similar results were found for other positions, figure 11 shows the calculated error for defenders which follows a similar pattern to figure 10. We can see that here the linear model also dominates all others for defender data regardless of the rolling dataset used. Both the midfielder and goalkeepers models produced also linear dominate results leading to the universal implementation of linear regression for all positions.

Looking past the individual model performances we can see in figure 10 and to a less pronounced extent in figure 11 the general trend of error for models, lower errors for 1 to 3 previous games and larger for 4 to 7. This is caused by short-term form verses long-term ability, the models perform well at predicting players with good form in the previous games considered range of 1 to 3, as form is typically short term lasting only a few games. The models then fall off by continuing to predict players who had short runs of form to further perform well and when they don't the error in predictions rises. The error drops again at the 8 previous games considered mark when the algorithms can build more rounded player profiles, aggregating a greater number of performances. When considering larger amounts of game data, outliers have smaller impacts on the results so players that consistently play well are predicted with higher scores than players with a period of good form followed by a period of bad form.

4.2.2 Final Model Implementation

Final linear models were trained on seasonal data from the 2016/17 to 2018/19 seasons for each FPL position using k-fold cross validation. The models were then tested on data from the 2019/20 season and the error in their predictions was recorded.

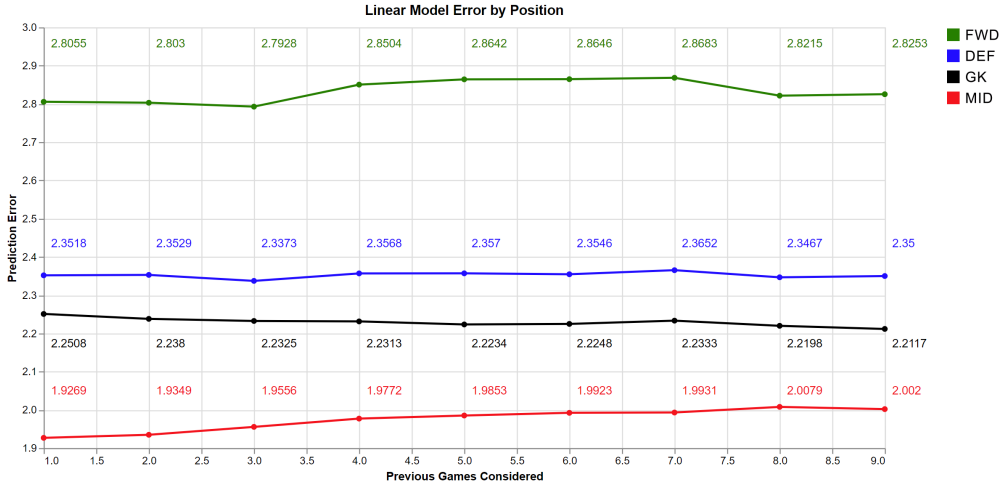


Figure 12: Error associated we each positional model when trained on different rolling datasets

Studying figure 12 we can see that despite large differences in prediction accuracy between positions, the model performs consistently when considering different numbers of previous games. The gap in positional performance could be explained through a combination of limited data and inherit position unpredictability. Forwards have the largest error in prediction, nearly a full point more than midfielders, this is due to the smaller selection of data to train from and the minimal ways in which forwards can earn points compared to midfielders. A forward is fully reliant on scoring or assisting goals, failing to do so results in poor point returns. A midfielder on the other hand can earn a clean sheet point and is more likely to win bonus points due to their higher involvement in the game. Despite having the smallest player pool, goalkeepers have the second lowest prediction error, this is because of the consistent nature of their position. A goalkeeper is part of a larger defence whom are all responsible for preventing goals, a poor performance from one player can be negated by the reliance on their fellow defenders.

4.3 Team Selection

Once performance predictions have been made for each player a subset of players must be selected to form an FPL legal team for a given gameweek. FPL imposes rules on the structure and distribution of players required to make a team to promote diverse selection and prevent all users selecting popular high performing players. The rules are as follows:

• Exactly 2 goalkeepers, 5 defenders, 5 midfielders and 3 forwards in total must be selected.
• The combined cost of all 15 players must not exceed £100M
• 11 of the 15 players must be selected for the starting line up with the remaining 4 being benched. Benched players get automatically substituted into the starting lineup should a starter not play.
• At least 3 defenders, 3 midfielders, 1 forward and exactly 1 goalkeeper must feature in the starting lineup.
• No more than 3 players from a single club may be selected.
• Exactly 1 starting player must be selected as captain (captains have a X2 multiplier applied to their scores), and exactly one starting player as vice captain, whom should the selected captain not player, assumes the role of captain.

Table 3: FPL enforced restrictions on team structure

The implemented algorithm must select a team that not only abides by all these restrictions but also maximises the total points of the combined players. Computational time is another aspect that was considered; large scale simulations involving many thousands of selections were used to test the accuracy and viability of the proposed solutions.

4.3.1 Baseline algorithms

Simple and naive selection algorithms were implemented to use as baselines to test the performance of more complex solutions. A greedy based approach and a random selection method were used to measure the more sophisticated implementations and provided baseline results which could be improved upon. The greedy method cycles through players in descending order of predicted points, selecting a player as long as their addition doesn't violate any of the defined rules. Greedy algorithms essentially make naive decisions and pick a solution quickly but without consideration of the optimal arrangement of elements, just the values of each element individually. The random selection creates a randomly selected legal team of players, the mean performance of a random selection over many iterations gives a baseline score of an algorithm with no optimisation logic. Algorithms that can constantly greatly outperform this random selection demonstrate that their logic is sound and decisions are calculated.

4.3.2 Multi-Dimensional Knapsack solution

Initial tests where conducted on implementations derived from David Pisgners paper discussing a range of knapsack problems types and their appropriate solutions [10]. Using a dynamic programming approach a multi-dimensional 1-0 knapsack solution was created in Python using Pandas for dataframe structure. The solution models each constraint as a new dimension, for n constraints n dimensions are needed. When combined with a dynamic programming approach which uses a lookup table to calculate optimal solutions the number of dimensions exponentially increases runtime.

4.3.3 Linear Optimisation

A linear optimisation solution was created in Python using the PuLP library. PuLP provides a constrained value maximisation solver that maximises an objective function subject to the conditions of the defined constraints. The sum of selected players scored was set to be the function to optimise with rule constraints being modelled as simple linear relationships.

```
#cost constraint
model += sum(decisions[i] * player_costs[i] for i in range(num_players)) <= budget
```

Figure 13: Cost constrained defined as a linear relationship.

Figure 13 shows how the cost contained can be easily applied. This constraint is forcing the sum of the costs of the select players in the model to be less than or equal to the budget supplied, the positional and club constraints are defined in the a similar manner

4.3.4 Algorithm comparison

Large scale simulations were run to compare the solutions generated by each method. Sample prediction data from the 2019/20 season was inputted to each algorithm in the gameweek range of 6 to 27, the present number of available gameweeks using a rolling dataset of $n\_prev\_games = 5$. The average points score and runtime each algorithm generated were used to compare their viability.

Algorithms	Total Predicted Points	Total Team Cost	Mean Gameweek Points	Average Runtime (seconds)
Greedy Selection	1533.693	1820.7	73.033	0.0467
Random Selection	1019.619 (Mean)	1728.3 (Mean)	52.839	0.0215
Linear Optimisation	1620.99	2087.4	77.19	0.0504
Knapsack Solution	1620.99	2087.4	77.19	5.832

Table 4: Results of running each selection algorithm on generated prediction data from the 2019/20 season

The results of the tests are shown in table 4, as evident from the mean gameweek scores the greedy selection, linear optimisation and knapsack solution were able to consistently find high performing teams, with the latter two finding the optimal solution every gameweek. The random selection performed as expected, it was very fast at finding a team but with average performance. The greedy solution performed unexpectedly well, achieving close to the optimal score each week. Upon inspection the cause of this unexpected high performance was due to the nature of club lineups in the 2019/20 season, many teams have fielded young, relatively inexperienced players. The most notable of this being Chelsea FC who have managed to achieve good results with a young rookie populated team, the biggest example of which are Mason Mount and Tammy Abraham who have achieved 103 and 131 points respectively as of gameweek 27. Naturally these younger players have smaller FPL price tags associated with them, which helps to mitigate the flaw of greedy algorithms which naturally selects the biggest-valued option in sight. This flaw is less pronounced when the biggest-value option in sight happens to also be the best due to the high performance of these low cost players.

Both knapsack and linear optimisation are appropriate for calculating a high performing team, the most notable difference between the two is their respective run times. The time complexity of the knapsack solution increases exponentially with each dimension added. Due to its significantly lower runtime a linear optimisation solution was chosen to implement into the final system.

4.3.5 Final Team Selection Algorithm

After committing to a linear optimisation solution further alterations were made to adapt the algorithm best to an FPL selection. A team of 15 players only contains 11 starters, by applying weighting the performance of the starting line up higher monetary sacrifices can be made in the bench selection in order to further improve the strength of the starting lineup. The prediction model only considers players who have recently played enough minutes for their statistics to have a level of consistency, this ensures that players selected for the team will have a high chance of playing in the upcoming game. Therefore the likelihood of a benched player being automatically substituted into the starting lineup is relatively low. The selection algorithm was altered to weight the performance of starting players higher than that of benched players, this results in a slightly riskier strategy with higher rewards where a team is more reliant on its starting 11. Captain selection was

also built into the system, where automatically the player predicted to score the most points is given the captains armband, in FPL players selected as captains have a X2 modifier applied to their scores.

4.4 Combined System

With both the prediction models and the team selection algorithm fully implemented final tests were run to determine which rolling dataset type produced the best results. Seasonal tests were conducted where player predictions were generated for a full season with models trained on data from the three other seasons, so to test the performance in the 2016/17 season, models were trained on data from the 2017/18, 2018/19 and 2019/20 seasons. An import distinction between these tests and ones run earlier is that the system was fully implemented it was optimising for maximum points return rather than prediction accuracy. Therefore the performance of the system is measured as the total actual points score its selections return rather than predicted score or prediction error.

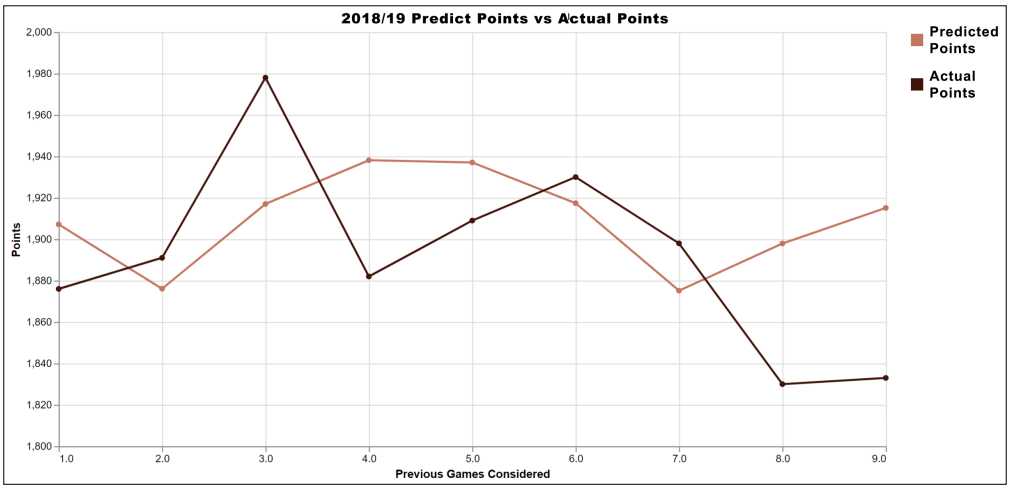


Figure 14: Predicted points plotted against actual points for a simulation run over the 2018/19 season

Figure 14 shows a graph of predicted total score and actual total score of a simulation over the full 2018/19 season. The line of predicted points represents how the system calculates its team selection will perform, the actual points line represents how well it actually performed. Each x value represents the data used to train and test the model, for example $x = 5$ shows the performance of the model when trained using rolling datasets which consider the statistics from 5 previous games. Simulations were used to attempt to identify which x values correspond to the highest points returns, for the 2018/19 season shown in figure 14 values of 3 and 6 performed best. The same simulations were run on data from each season in an attempt to find a universal data structure that produces the best results regardless of season.

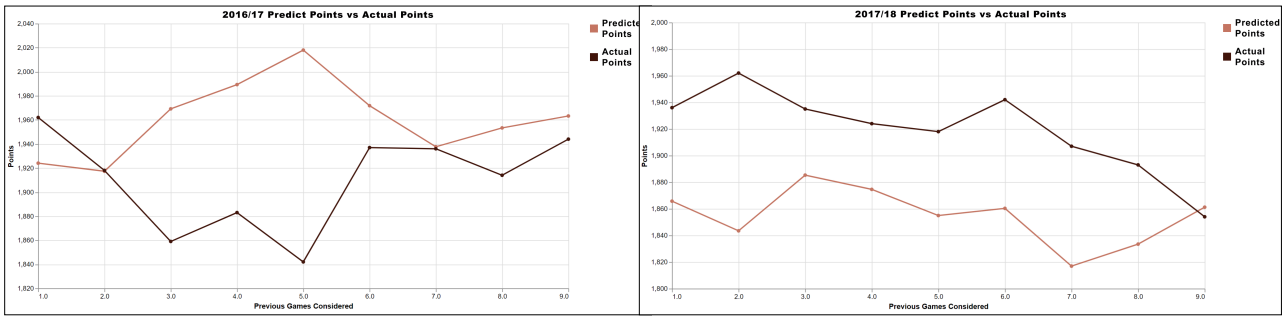


Figure 15: Left: Predicted points plotted against actual points for a simulation run over the 2016/17 season. Right: Predicted points plotted against actual points for a simulation run over the 2017/18 season

Figure 15 shows the results of simulations run on data from the 2016/17 (left) and 2017/18 (right) seasons. While the 2017/18 season loosely follows the trend of figure 14, having peaks at x values of 2 and 6, the 2016/17 seasons follows a completely different path. It was so determined that there is no optimal rolling

dataset that can be applied generally to all seasons, instead data structure should be determined on a season by season basis. To find the best dataset to use for the 2019/20 season simulations were run on the 27 games that have currently been played in the season.

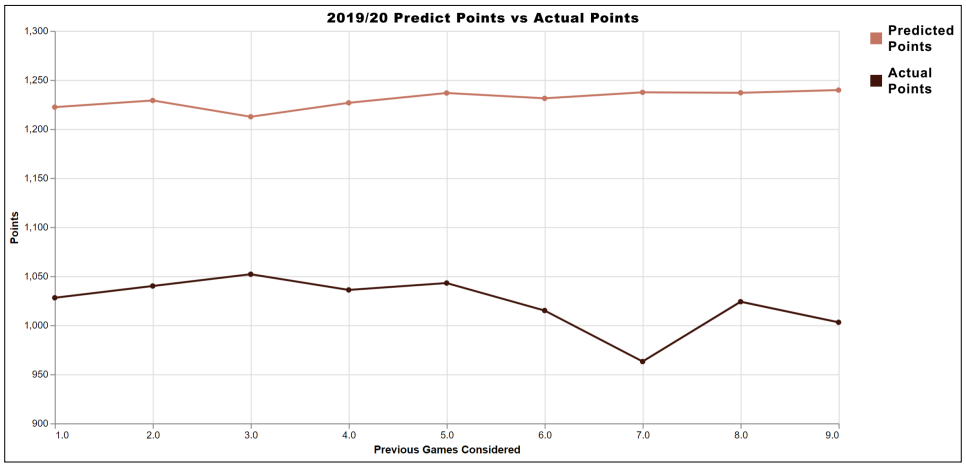


Figure 16: Predicted points plotted against actual points for a simulation run over the currently played gameweeks in 2019/20 season

The simulation results show much less variance with change in previous games considered with only a significant drop at $x=7$. The maximum points achieved was 1050 at $x=3$, therefore rolling datasets of *n_prev_games = 3* were used to make predictions in the final system.

4.5 Web Application

The web application was implemented using Google Apps Script, an environment that allows for simple JavaScript driven html applications to be created. One of the biggest advantages of using this Google Apps Scripts is that Google will host the application at no cost on their servers, allowing anyone to view it. Avoiding setting up website hosting and paying server fees was a much welcomed feature. Two pages were created, the home screen which displays data generated by the models and team selection algorithm and a short about page which provides some basic information into the project and its motivations.

Figure 17: The graphic for post gameweek 27 accompanied with a short textual description of the models performance

From figure 17 we can see gameweek 27 information, a graphic view of the selected starting line-up and bench are visible on the left hand side of the graphic with player information available of the right. A direct comparison between the models score and the average players score is also available to be viewed. This is the view for gameweeks that have already taken place where player points are now known. Figure 18 shows the graphic created for yet unplayed gameweeks.

Figure 18: Graphic created for gameweek 29 predictions accompanied with a short textual description

Users are able to view the system's selected team and also lists of players with the highest projected points and highest value, defined as points per £1M (predicted points / player cost). These lists of players give users information to help aid transfer decisions for their personal team, often users are looking for cheap players to fill the last spots in their team so they can afford a greater number of premium players (high cost, high reward). The final web application can be viewed at: https://bit.ly/2Wf4jfq.

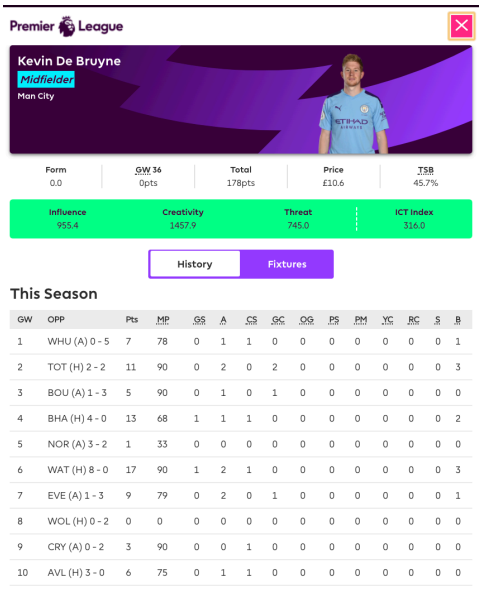
Chapter 5: Evaluation

5.1 Further System Testing

Throughout the development of this project testing was used to ensure implemented subsections of the created system were functioning as desired. In this section the further tests that each modular component of the system was subjected to are described, with the respective results achieved.

5.1.1 Data Testing

This is a data oriented project, the accuracy and format of said data formed the foundations upon which the other components were built. It was therefore imperative that the necessary precautions were taken to validate the data and the various manipulations performed on them. Functional tests were performed on the raw data read from the Vaarstav Github repository[7] to cross check its contents against the official FPL website. Random player samples from the *merged_gw.csv* and *raw_player.csv* files were checked against official recorded statistics.



GW	OPP	Pts	MP	GS	A	CS	GC	OG	PS	PM	YC	RC	S	B
1	WHU (A) 0 - 5	7	78	0	1	1	0	0	0	0	0	0	0	1
2	TOT (H) 2 - 2	11	90	0	2	0	2	0	0	0	0	0	0	3
3	BOU (A) 1 - 3	5	90	0	1	0	1	0	0	0	0	0	0	0
4	BHA (H) 4 - 0	13	68	1	1	1	0	0	0	0	0	0	0	2
5	NOR (A) 3 - 2	1	33	0	0	0	0	0	0	0	0	0	0	0
6	WAT (H) 8 - 0	17	90	1	2	1	0	0	0	0	0	0	0	3
7	EVE (A) 1 - 3	9	79	0	2	0	1	0	0	0	0	0	0	1
8	WOL (H) 0 - 2	0	0	0	0	0	0	0	0	0	0	0	0	0
9	CRY (A) 0 - 2	3	90	0	0	1	0	0	0	0	0	0	0	0
10	AVL (H) 3 - 0	6	75	0	1	1	0	0	0	0	0	0	0	0

GW	name	total_points	minutes	goals_scored	assists	clean_sheets	own_goals
1	Kevin_De Bruyne_215	7	78	0	1	1	0
2	Kevin_De Bruyne_215	11	90	0	2	0	0
3	Kevin_De Bruyne_215	5	90	0	1	0	0
4	Kevin_De Bruyne_215	13	68	1	1	1	0
5	Kevin_De Bruyne_215	1	33	0	0	0	0
6	Kevin_De Bruyne_215	17	90	1	2	1	0
7	Kevin_De Bruyne_215	9	79	0	2	0	0
8	Kevin_De Bruyne_215	0	0	0	0	0	0
9	Kevin_De Bruyne_215	3	90	0	0	1	0
10	Kevin_De Bruyne_215	6	75	0	1	1	0

Figure 19: Official FPL statistical page for Kevin De Bruyne compared to data sourced for the Vaastav repository.

Figure 19 shows the official FPL statistical page for Kevin De Bruyne compared with the data sourced from the the Vaastav repository. We can see that for each gameweek the statistics accurately match. Further player scored were cross-checked using a random sampling method. Tests concluded finding zero errors. Many of the referenced FPL applications in this project utilise the same repository so it is extremely unlikely the data is malformed or contains inaccuracies.

5.1.2 Model Testing

Prediction model validity was ensured by minimising bias, calculating random training and testing subsets of the data over thousands of iterations to calculate model accuracy. Implemented models were designed to be applicable to new data generating somewhat accurate predictions. Overfitting can be a large problem in statistical models so removing all training-testing biases was important to ensure the creation of accurate models which could be generalised to unseen data.

[7] Vaastav GitHub repository available at: https://github.com/vaastav/Fantasy-Premier-League

5.1.3 Team Selection Testing

To ensure the team selection algorithm was performing as desired, functional tests were conducted to analyse the produced and desired outputs for different inputs. Large scale simulation tests were run over hundreds of iterations where the produced output was compared to the correct desired output. A dream team selection was used in these tests where the algorithm was attempting to find the best possible combinations of players given their real points returns for each gameweek. These optimal gameweek scores are publicly available on the FPL website and so comparison of the algorithms results to that of the FPLs can test the functionality of the algorithm. Significant simulations were run concluding that the implemented algorithm did perform correctly as desired.

5.1.4 Web Application Testing

A compatibility test structure was used to assess the web applications usability and readability on different viewing platforms. Precautions were taken when implementing the application to ensure text and graphics would display correctly on desktop and mobile devices. The application was found to display accurately on desktop and laptop devices tested however some minor inconsistencies were found in mobile devices. On occasion the graphics did not display correctly requiring the refreshing of the page to prompt the correct visual interface to be shown. This is a problem that cannot easily be solved in Google Apps Scripts because of its limit accessibility to the underlying systems creating the web application. However due to the minor nature of the problem it was not a large cause for concern.

5.2 Evaluation

5.2.1 Project Development

At the start of the project a gantt chart was development to aid time management. Aims were broken down into smaller sub tasks each with a purpose to move the project forward. Tasks had an associated start and finish date which were used as indication to the accurate pace of development.

Tasks	Start date	Finish Data	Oct	Nov	Dec	Jan	Feb	Mar	Apr
Aim 1									
Implement the four different machine learning algorithms in python	20th Oct 2019	30th Oct 2019							
Identify set of explanatory variables	1st Nov 2019	3rd Nov 2019							
Train and test models and refine explanatory variables	20th Nov 2019	25th Nov 2019							
Compare different model approaches and commit to a single model	5th Jan 2020	10th Jan 2020							
Extensively test and refine positional models	10th Jan 2020	15th Jan 2020							
Aim 2									
Identify possible solution algorithms	5th Nov 2019	10th Nov 2019							
Test and compare possible solutions	10th Nov 2019	15th Nov 2019							
Commit to a single solution	20th Nov 2019	25th Nov 2019							
Add team and formation restrictions	20th Nov 2019	30th Nov 2019							
Optimise algorithm to minimise runtime	15th Dec 2019	30th Dec 2019							
Test algorithm to ensure acceptable functionality	25th Jan 2020	30th Jan 2020							
Aim 3									
Identify desired use cases	1st Dec 2019	5th Dec 2019							
Plan user interface	5th Jan 2020	10th Dec 2020							
Create web application	15th Feb 2020	25th Feb 2020							
Testing	15th Mar 2020	25th Mar 2020							
Publish web application	25th Apr 2020	-							

Figure 20: Time schedule gantt chart used to aid project time management

Despite minor set backs all tasks were accomplished. During the development cycle updates were required to be made to certain tasks and aims, for example originally a knapsack solution was designed to be used for aim 2 so its sub tasks were all knapsack related. These tasks were updated to have more general wording and to be applicable to solutions of any nature. Development in shorter sprints was found to be more favourable than long drawn out implementations phases, the majority of tasks were updated to have smaller time frames to solely focus on their development. Overall the development cycle was a smooth gradual process in which all tasks and aims were achieved to a satisfactory level.

5.2.2 Final Results

With the full system implemented and tested to ensure correctly desired behaviour, final results could be gathered. A simulation was run on the current 2019/20 Premier League season where each week the system would select an FPL legal team of players. The results of the actual performance of the selected teams were then recorded and compared to the average points earn by all FPL users and a dream team score in the same gameweek. The dream team was defined as the best possible score achievable for a given gameweek by an FPL legal selection of players.

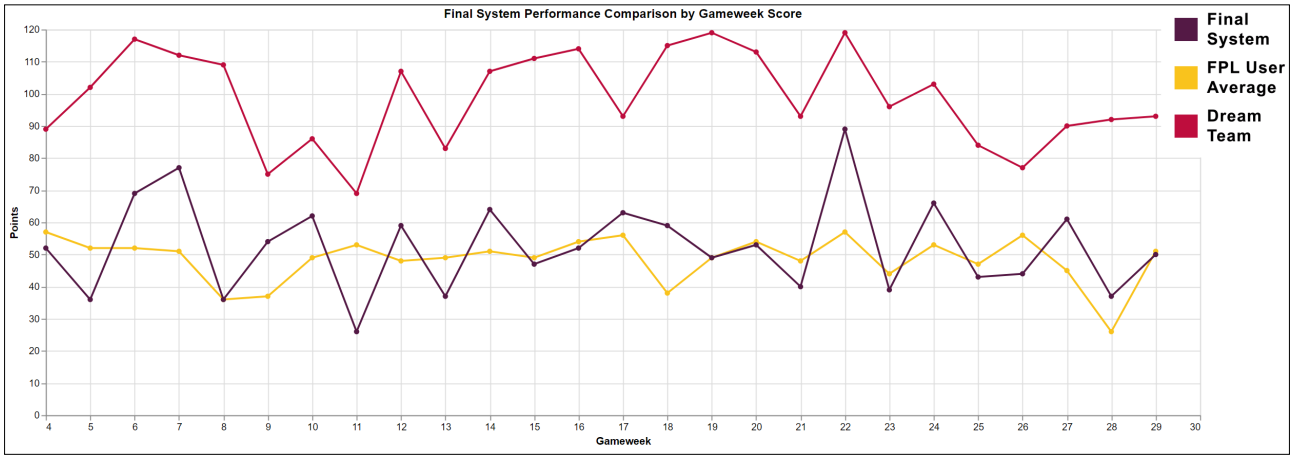


Figure 21: Comparison of the final systems performance versus FPL user average and the calculated Dream Team score by each gameweek

From figure 21 we can see direct comparisons of the performance of the team selected by the final system versus the FPL user average, the dream team gives context showing the maximum possible points that could be earned each gameweek. As expected the FPL user average has much less variance in its points earnings as its taken as the average score from almost 7.5M players, the final system has more variance in its individual scores but follows a similar trend to the average score.

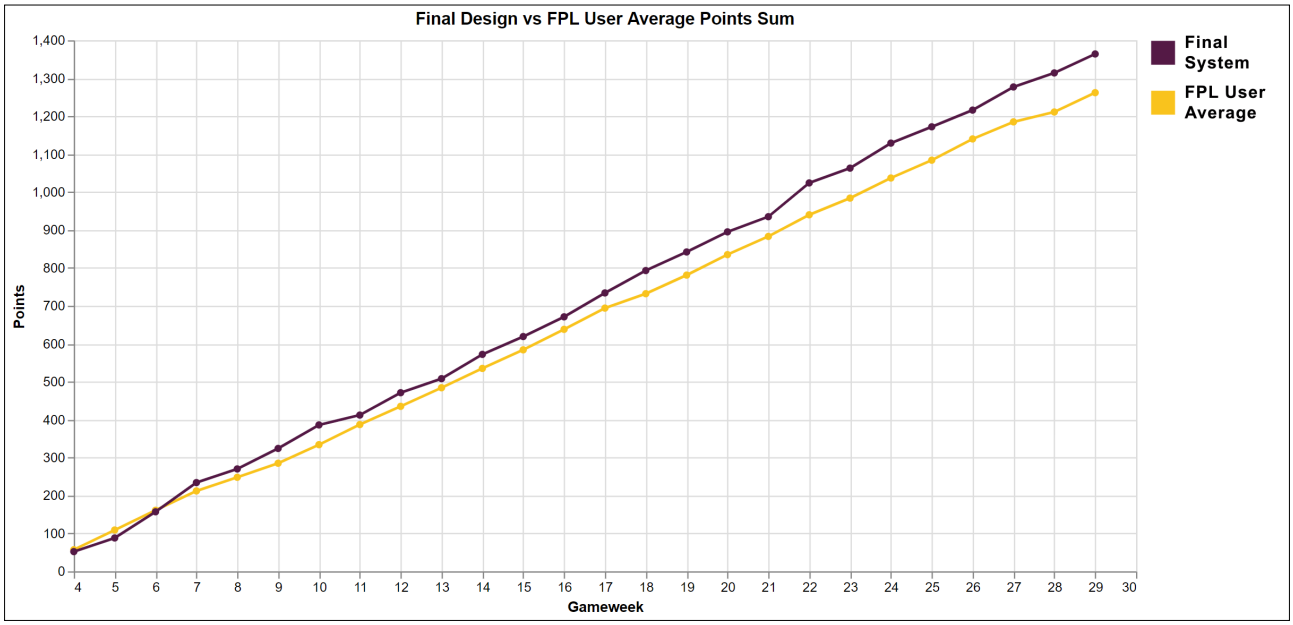


Figure 22: Cumulative sum of points earned of the Final system compared to the FPL user average

Figure 22 shows the cumulative scores from both the final system and the average of users allowing us to assess the general performance trends. The final system performs slightly better than the average score earning 1364 points in the 4 to 29 range of gameweeks, compared to the 1262 of the average score. For performance over this time span of gameweeks the system placing in top 16.66% of all users. These results

must be taken with a ‘grain of salt’ though as the final system is creating a team from scratch each week where as real FPL users must abide by weekly transfer restrictions. However this system was created to generate an optimal FPL team for each given gameweek where selections were made purely using statistical data which it achieves.

Chapter 6: Future Work

6.1 Further Design

There are a number of further alterations that could be made to improve the system presented here. This section discusses these possible improvements, how they might be implemented and their effect on the system. The most notable areas were determined to be data sourcing and prediction algorithm design.

6.1.1 Data improvement

Despite having access to player data from three and a half seasons (as the 2019/20 season remains unfinished as of now) lack of data for certain positions reduced the accuracy of the prediction models. The position which the models predicted with the lowest error was midfielders, which had the largest data pool. Both goalkeeper and forward prediction were significantly worse as there was only half as much data available on which to train models on.

As well as collecting statistics from more seasons, collection of different types of data could potentially significantly reduce prediction error. Bookmaker odds give further access to a wealth of data although the reliability of the source would need to be verified to ensure statistics were accurate. The formatting of data could be improved by further subdividing players into positional subsets, for example in the current system both attacking wingbacks and centre backs fall under the defender label despite their vastly different play styles and the way they typically earn points. Further dividing players would reduce variance in the data and allow models to fit more accurate prediction planes, though the quantity of data would need to be substantial to ensure sufficient training could be performed.

6.1.2 Model improvement

As discussed in Chapter 2 Related Works there are many approaches that can be taken to predict future player performance in a fantasy sport environment other than the linear methods explored in this project. To further improve models, other supervised non-linear algorithms could be implemented such as neural networks or regression focused support vector machines. An unsupervised approach could also be explored using classification algorithms such as k-means clustering. Many projects examine in Chapter 2 Related Works adopted different approaches and all managed to achieve successful results demonstrating the wide range of possible methods that can be applied to this problem. Akhil Gupta achieves very low prediction error with his recurrent neural network based approach, which is an approach that warrants further exploration and experimentation.

6.1.3 Team Selection improvement

Team selection is handled well in the current system, an optimal performing team is able to be found for each input and substitutions and captain picks are handled well. The current algorithm has acceptable time complexity and scales well with feasible player pool sizes. A modification that could be made would be to attempt the system to take an already selected team and suggest transfers to improved the past team that take into account transfer penalty. Each week the FPL awards a user a free (free as in no points penalty) transfer, the player has the option to make further transfers but at a -4 points penalty to their current standings. A system that could weigh up the benefits and positives of making transfers would be able to compute a legal team each week that could directly participate in the league making alterations to its existing team each week instead of creating new ones. Such a system would need to factor in a number of more complex decisions as well as transfers, in FPL a system of ‘chips’ is also available, which players can play a limit amount of to get a boost for one week, an example are the Triple Captain chip which multiples your captain points by a factor of 3 instead of 2 for one week. The system would need to account for these chips and determine the optimal times to play them. With these suggested alterations a competitive system that fully manages a team through a season is possible, with the potential to place very highly on the global leaderboard.

6.1.4 Web Application improvement

While the developed web application does function as required, allowing FPL users to view data generated by models it can be significantly improved. A more in-depth design that still embodies and simplistic layout would reduce the amount of data that can be meaningfully communicated to the viewer of the web application. Features such as the ability to search through all prediction data for specific player statistics and recording more in-depth records from past gameweeks would aid the function of the web applications. The web page created by Neena Parikh as part of her 2014 paper gives a good example of the depth and functionality the current implementation could adopt.

To create a more holistic experience, a shift to a complete web framework such as Laravel would be advised. Google Apps Sheets provides the necessary functionality for the current web application but has limitations to what can be achieved in its environment. Frameworks such as Laravel allow for full control of the web application using a Model-View-Controller architecture were each part of the application can be carefully crafted to the desired needs.

6.2 Future Applications

6.2.1 Different Fantasy Sport Application

Many of the projects discussed in Chapter 2: Related Works apply systems to fantasy leagues other than the FPL. Hermann and Ntosos paper explores fantasy application in the sport of basketball, Neena Parikh's models are applied to NFL datasets and there are many more examples. The ideas explored in this project can be applied to any fantasy sport system provided enough past data exists to sufficiently train and test models. In many cases the rolling data structure used here can be copied over to fantasy application with no major redesigns necessary. Predicting points simply becomes a matter of redefining the explanatory and target variables and then adequately testing the prediction models.

6.2.2 Predicting Real Life Performance

The scrutiny of sports data isn't unique to fantasy applications, modern sport franchises invest million of pounds in statistical analysis. Data sciences are employed to identify trends and patterning in their own and their opponents play-styles. Arsenal Football Club is one of the biggest and most successful in the world, they currently use data science techniques to analyse potential transfers and identify players who will adapt well to the teams player style [15]. The core methodology used isn't dissimilar to the work explored in this project. Past performance data is analyses to look for patterns and trends that can reveal information about future unknown metrics.

It is not unfeasible to imagine a system using explanatory variables of a similar nature to the ones used in this project, although preferably of much greater quantity, that is capable of predicting real life player performance metrics to an acceptable degree of accuracy. After all, that is the approached used here, only using the FPL as an application to simplify and quantify player performance. Future work in this area is at the cutting edge of some of the leading data science research and sport data analysis is rapidly developing as its own respected sport field of focus.

Chapter 7: Conclusion

In Chapter 1: Introduction, the problem this project was tasked with solving was to “create a system to select high performing FPL legal teams in which human bias is minimised”. Project aims were defined to ensure that with their satisfaction, a solution to the stated problem was created.

I. From an input of raw player performance data, predictions about future performance statistics are made which satisfy a given accuracy measure.

Through the development and implementation of the final system all of the defined aims were achieved to a satisfactory level. Linear regression prediction models were run on optimally formatted data, achieving a mean prediction error for selected teams of just 13.23 points when applied to data from the current 2019/20 Fantasy Premier League season. Models aimed to project individual player points within an accuracy of 2.329 points with detailed guidance on actions that could be taken to reduce this error presented in the Future Works and Evaluation Chapters.

II. From a list of players and predicted fantasy points approximate a high scoring team of players which obeys all FPL restrictions

Perhaps one of the greatest successes of this project, a functional team selection algorithm capable of considering performance statistics and FPL data from a pool of over 500 players was implemented. Linear optimisation was used to consistently find the best selection of players such that their combined sum of points was maximised while ensuring all FPL formation and selection restrictions were obeyed to their full extent. The final algorithm was able to create a team of 15 players, 11 of which were selected in the starting lineup with the 4 others being benched; extensive tests confirmed the solution was capable of consistently finding legal teams in less than 1 second.

III. Publicly publish player and team predictions to a web application, allowing FPL users to view the generated results.

A web application was successfully created and deployed featuring data generated by both the created prediction models and the team selection algorithm. Despite having a smaller area of focus than Aims I and II the creation of the application enabled generated data to be publicly visible to any whom seek it. Detailed graphs accompanied by textual descriptions allow viewers to see the latest model results as well as explore the historical data generated for previous gameweeks.

Many problems were faced and solved during the development of this project, including efficiently determining an optimal data structure, evaluating the success and flaws of different machine learning regression algorithms and the creation of optimal legal FPL teams from a long list of player data. Ultimately through combined individual satisfaction of each of the projects aims it can be stated that a solution to the problem of creating a purely statistical system that can select high performing FPL legal teams has been successfully achieved.

Bibliography

[1] Official Fantasy Premier League: https://fantasy.premierleague.com/
[2] Football History: https://www.footballhistory.org/
[3] Fantasy Sport, Encyclopaedia Britannica: https://www.britannica.com/sports/fantasy-sport
[4] The Psychology of Fantasy Football, Fantasy Football Scout: https://www.fantasyfootballscout.co.uk/2016/05/26/the-psychology-of-fantasy-football/
[5] Interactive Tools for Fantasy Football Analytics and Predictions using Machine Learning, Neena Parikh 2014: https://dspace.mit.edu/handle/1721.1/100687
[6] Time Series Modelling for Dream Team in Fantasy Premier League, Akhil Gupta 2017: https://arxiv.org/abs/1909.12938
[7] Machine Learning Applications in Fantasy Basketball, Eric Hermann and Adebia Ntuso 2015: http://cs229.stanford.edu/proj2015/104_report.pdf
[8] Competing with Humans at Fantasy Football: Team Formation at Large Partially-Observable Domains, Tim Matthews, Sarvapali D. Ramchurn, Georgios Chalkiadakis 2013: http://www.intelligence.tuc.gr/~gehalk/Papers/fantasyFootball2012cr.pdf
[9] Multi-stream Data Analytics for Enhanced Performance Prediction in Fantasy Football, Nicholas Bonello, Joeran Beel, Seamus Lawless, Jeremy Debattista 2019: https://arxiv.org/pdf/1912.07441.pdf
[10] Algorithms for Knapsack Problems, David Pisinger 1995: http://hjemmesider.diku.dk/~pisinger/95-1.pdf
[11] Predicting Optimal Game Day Fantasy Football Teams, Glenn Sugar and Travis Swenson: https://pdfs.semanticscholar.org/a575/4aaa4edfe3099afd27a6180e78fc388a6cd6.pdf
[12] Supervised Verses Unsupervised Learning: Key Differences, Guru99: https://www.guru99.com/supervised-vs-unsupervised-learning.html
[13] Linear Programming, Wikipedia: https://en.wikipedia.org/wiki/Linear_programming
[14] Home-Field Advantage, iresearchnet.com: http://psychology.iresearchnet.com/social-psychology/control/home-field-advantage/
[15] Burt, J. (2017, August 17), https://www.telegraph.co.uk/football/2017/08/17/