

Multi-stream Data Analytics for Enhanced Performance Prediction in Fantasy Football

Nicholas Bonello

A Dissertation

Presented to the University of Dublin, Trinity College

in partial fulfilment of the requirements for the degree of

Master of Science in Computer Science (Intelligent Systems)

Supervisor: Joeran Beel, Seamus Lawless

Assistant Supervisor: Jeremy Debattista

August 2019

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

Nicholas Bonello

August 9, 2019

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Nicholas Bonello

August 9, 2019

In memory of Professor Séamus Lawless

Acknowledgments

Firstly, I would like to thank my supervisor Professor Seamus Lawless and co-supervisor Dr. Jeremy Debattista whose knowledge, patience and mutual passion for fantasy football have made this an enjoyable and gratifying journey.

As importantly, I would like to thank Professor Joeran Beel for accepting to mentor and guide me towards the end of my dissertation. Providing extremely valuable insights and recommendations throughout this time that have helped push me in the right direction.

Finally, I would like to thank my family and friends for their continuous support.

NICHOLAS BONELLO

*University of Dublin, Trinity College
August 2019*

Multi-stream Data Analytics for Enhanced Performance Prediction in Fantasy Football

Nicholas Bonello, Master of Science in Computer Science
University of Dublin, Trinity College, 2019

Supervisor: Joeran Beel, Seamus Lawless

Fantasy Premier League performance predictors commonly base their datasets on purely historic statistical data. The main problem with this approach is that external factors such as injuries, managerial decisions and other tournament match statistics can never be factored into the final predictions. In this dissertation, we propose a novel approach for predicting future player performances that automatically incorporates human feedback into the predictions. By combining standard statistical measures with data from betting-markets, social media and web articles, we aim to make more informed predictions by expanding the number of factors that can be considered before making predictions. We explore the current state-of-the-art fantasy football recommender services and compare their prediction accuracy to the different algorithms we propose. When tested on the English Premier League 2018/19 season, we found that including multiple data-sources significantly improves the performance for recommender services. Our machine-learning models trained on historic statistics alone ranked within the top 800,000 players (top 13%) out of over 6.5 million FPL players. On the other hand, machine-learning models trained using multiple data-sources, namely; historic statistics, betting-markets, news-articles and blogs, achieved a rank of 20,000 (top 0.5%). This work shows the potential that multi-stream data analytics can have in any field where both statistics and external factors affect the final outcomes.

Contents

Acknowledgments	iv
Abstract	v
List of Tables	ix
List of Figures	x
Chapter 1 Introduction	1
1.1 Motivation	2
1.1.1 How can human feedback help?	3
1.2 Research Question	5
1.3 Research Aims	5
1.4 Research Challenges	6
1.5 Dissertation Outline	6
Chapter 2 Background Reading	8
2.1 Learning Algorithms	8
2.1.1 Supervised Learning	8
2.1.1.1 Support Vector Machines (SVM)	9
2.1.2 Ensemble Learning	11
2.1.2.1 Random Forests	13
2.1.2.2 Gradient Boosting Machines	14
2.1.2.2.1 Tree-Specific Parameters	15
2.1.2.2.2 Boosting Parameters	16
2.1.2.2.3 Miscellaneous Parameters	16

2.2	Natural Language Processing	17
2.2.1	Named Entity Recognition	17
2.2.2	Sentiment Analysis	18
2.2.3	Microblog Natural Language Processing	20
Chapter 3 Literature Review		22
Chapter 4 Design & Implementation		27
4.1	Application Overview	27
4.1.1	Application Architecture	28
4.2	Prediction Algorithms	29
4.2.1	Dataset	29
4.2.2	Machine Learning Algorithms	30
4.2.3	Parameter Tuning	35
4.3	Statistical Analysis	37
4.3.1	Historical Gameweek Data	38
4.3.2	Upcoming Fixture Statistics	38
4.3.3	Bookkeeper Odds	38
4.3.4	Data Collection Pipeline	40
4.4	Linguistic Analysis	41
4.4.1	News-Articles and Blogs	41
4.4.1.1	Data Collection	41
4.4.1.2	Generating Predictions	43
4.4.2	Twitter data	46
4.4.2.1	Data Collection	47
4.4.2.2	Generating Predictions	48
4.4.2.3	Issues with Tweets	49
4.5	Generating Predictions	52
4.6	Challenges	54
4.6.1	Statistical Analysis	54
4.6.2	Linguistic Analysis	54
4.7	Summary	55

Chapter 5 Results & Evaluation	56
5.1 Hypothesis	56
5.2 The Predictive Model and FPL Rules	56
5.3 Web Application	57
5.4 Evaluation Approach	59
5.4.1 Precision	60
5.4.2 Recall	60
5.4.3 Accuracy	60
5.4.4 F-Score	61
5.5 Evaluation	61
5.5.1 Comparing different Machine Learning Algorithms	61
5.5.2 Comparing Datasets	64
5.5.3 Results	66
5.6 Summary	67
Chapter 6 Conclusion	68
6.1 Objective Assessment	68
6.2 Criticisms & Limitations	69
6.2.1 Statistics	69
6.2.2 News-articles and Blogs	70
6.2.3 Twitter	70
6.3 Future Work	71
6.4 Final Outtakes	72
Bibliography	74
Appendices	78

List of Tables

4.1	Player metrics for the upcoming gameweek aggregated by isCaptain value	32
4.2	Player metrics for previous gameweek aggregated by isCaptain value . .	32
4.3	Player metrics match 2 gameweeks ago aggregated by isCaptain value .	32
4.4	FPL provided statistics aggregated by isCaptain value	33
4.5	FPL player value and betting odds for all players aggregated by isCap- tain value	33
4.6	Queries used to gather news articles and blogs	43
4.7	Aylien API vs IBM Watson performance when predicting entity-based sentiment.	46
4.8	Example of searching periods per gameweek	47
5.1	Confusion Matrix layout	59
5.2	Comparing different machine learning algorithms performance predict- ing gameweek 38	63
5.3	Detailed results for GBM predictions in FPL season 2018/19 gameweek 38	64
5.4	Comparing different datasets and algorithms	66

List of Figures

1.1	Reddit comments that correctly predict Lindelhof missing an upcoming gameweek	4
2.1	Linearly separable SVM [1]	9
2.2	Optimising hyperplanes to maximise margin between plane and nearest point	10
2.3	Bootstrapping dataset example	12
2.4	Bootstrapping dataset example	13
2.5	Random Forest structure [2]	14
4.1	Proposed high-level architecture design	28
4.2	Points scored per player per gameweek in past 3 seasons of FPL data .	30
4.3	Feature importance score for midfielder predictions	34
4.4	Correlation Matrix for all available parameters	35
4.5	Calculating AUC ROC for different values of n-estimators	37
4.6	Blog and news-article sentiment score per player per gameweek in FPL season 2018/19	44
4.7	Tweet example showing emoticons used to imply transfers	49
4.8	Tweet example showing independent player emoticons	50
4.9	Tweet showing injuries through emoticons	51
4.10	Tweet written without any correct grammatical structure	52
4.11	High level flow diagram for loading the latest data into our dataset and generating predictions for the upcoming gameweek.	53
5.1	GBM prediction for season 2018/19 gameweek 2	58

5.2	SVM prediction for season 2018/19 gameweek 2	58
5.3	Comparing SVM vs RF vs GBM performance using historical statistics, betting odds, news articles & blog data	62
5.4	Comparing different dataset performances for GBMs	65

Chapter 1

Introduction

The English Premier League is the highest-level tournament in the English football league system, consisting of 20 teams with each team playing the other 19 teams twice; once at their stadium (home), and once at the opposing teams stadium (away). These 380 fixtures are split across 38 different gameweeks which typically tend to fall on weekends, with all fixtures of the same gameweek happening within a period of 3 or 4 days. However, due to external factors such as mid-week European tournaments or international tournament qualification matches, fixtures can sometimes be rescheduled. This means that some teams would miss certain gameweeks while playing twice in others [3].

Fantasy premier League (FPL) ¹ is the official fantasy football browser game for the English premier league, with over 6.5 million players actively competing against each other every season, aiming to accumulate the highest number of points. Fantasy sports are continuously becoming more popular, with Fantasy Premier League leading the way in terms of number of players per season. Because of this rise in popularity, achieving a respectable rank is becoming increasingly difficult, much less trying to win on either the official or paid leagues. At the start of each season, footballers are each assigned a value or cost which reflects that players point-scoring potential, and a position depending on the players real-world playing position. This position is one of goalkeeper, defender, midfielder or forward. Player positions are important for two main reasons; because each position is awarded different points for the same action

¹<https://fantasy.premierleague.com>

and because FPL players only have a restricted number of player choices per position. Each FPL player commonly referred to as a manager, must select a team of 15 players from over 500 available players, whilst being aware of the maximum budget that all managers are equally assigned - £100. Each team must consist of 2 goalkeepers, 5 defenders, 5 midfielders and 3 forwards with no more than 3 players originating from the same football club.

At the start of every gameweek, managers are required to select 11 of these 15 players to form part of that teams starting lineup. These players earn points for the team based on their individual contributions to their real-world matches, gaining points by simply being in the real-world starting lineup and gaining additional points through scoring goals, assists, clean sheets and many more factors. For each match in the gameweek, the best players on the pitch are even given bonus points for their exceptional performances. However, players can also lose points if their team concedes many goals, are given a card, miss a penalty, or even score an own goal. If the player spends the duration of the game as a substitute on the bench, no points are awarded [4].

Between gameweeks, managers are then allowed to make transfers - buying and selling different players to use in the upcoming weeks. However, managers are penalised for making too many transfers, with only a single unpenalized transfer being allowed every week. Once a team is created, players can compete against friends through leagues, or can even compete against others in prize-eligible leagues where players go head-to-head against each other competing for large prizes. For each week, players are also allowed to pick a "captain" - a player that is granted a double-points multiplier. The goal of this game is to maximize the total number of points achieved throughout the entire 38 gameweek season, by selecting players that are likely to earn large points in the upcoming weeks, whilst also considering the many different constraints and the overall unpredictability of football.

1.1 Motivation

Most players tend to haphazardly pick their starting teams and make weekly transfers without any in-depth consideration of previous form, upcoming fixtures and external factors. In addition, most players would not be capable of, or have the required knowl-

edge on data science techniques that could help achieve a higher score through analysis of publicly available data. Several recommender systems that can provide insights on which players to pick for the upcoming gameweeks already exist ^{2 3 4}. However, all these systems are disposed to only consider statistical data. This means that the models would only consider past performances, injuries and upcoming fixture difficulty amongst other significant statistical patterns. The main issue with these models is that potentially important non-statistical data can never be factored into consideration; such as for instance a team playing an additional mid-week non-premier league game (e.g. Champions League). While there seems to be no significant changes in performance in games after a mid-week match [5], squad rotation is a very important factor to consider for fantasy premier league players. While it may sound counter-intuitive, Cox explains the importance of lineup rotation and how it benefits teams to have weaker fully-rested set of players playing rather than have the same full first team play full midweek and weekend matches [6]. Since fantasy football is player-oriented, understanding and predicting these lineups is a crucial factor that players must consider when picking their starting lineups. Having an expensive player not play a significant amount of time and thus not gain many points would be considered a significant loss. Unexpected events such as upsets or even players getting lucky and scoring a large point haul in a previous gameweek would also have a huge effect on predictions of a strictly-statistical model; likely predicting that that same player should be chosen next week due to the previous weeks unexpectedly good result.

1.1.1 How can human feedback help?

By combining important statistical measures with human feedback through news-articles, blogs and social-media data, we hope to create a model that can provide better predictions by also considering external factors.

²<https://www.fantasyfootballfix.com/>

³<https://fantasyoverlord.com/FPL>

⁴<http://www.fplstatistics.co.uk/>

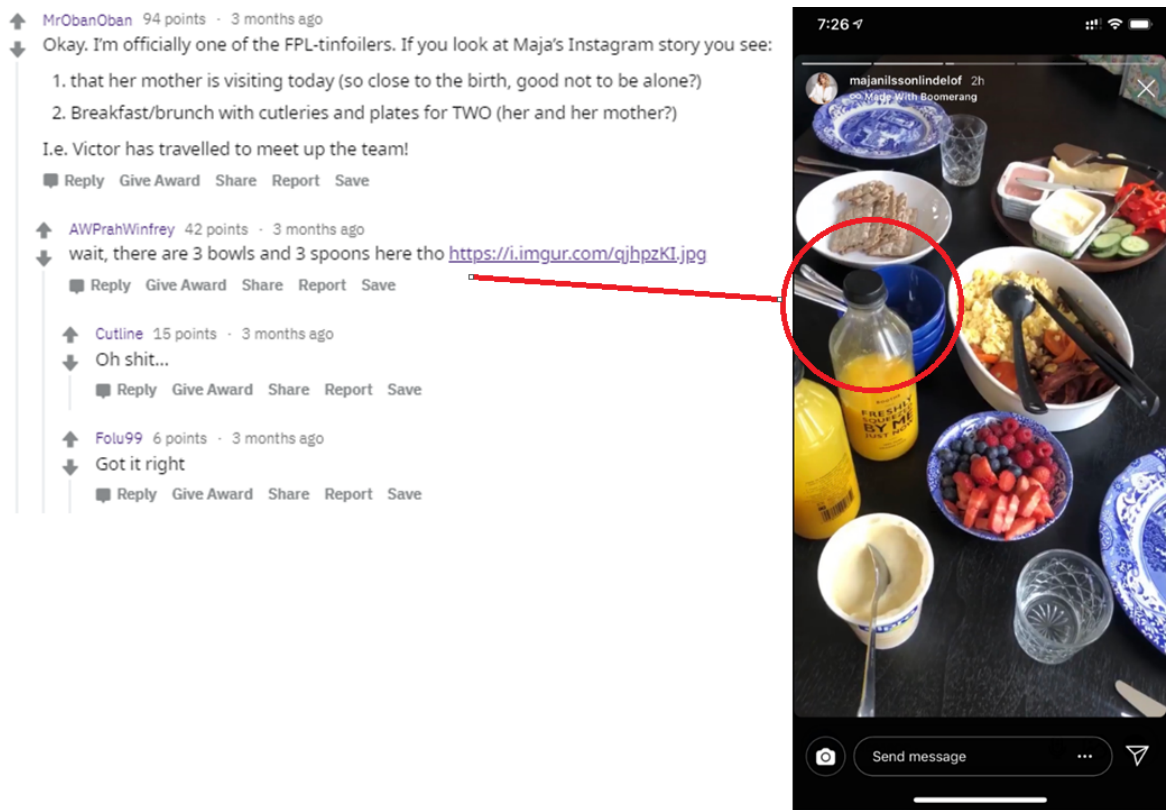


Figure 1.1: Reddit comments that correctly predict Lindelhof missing an upcoming gameweek

Figure 1.1 represents a post on an FPL-specific Reddit thread⁵, discussing whether Manchester United defender Lindelhof would play in the upcoming gameweek. After users discovered that his wife was expected to give birth that week, rumours rose claiming that Lindelhof would be missing the next game to spend time with family. Looking at social-media pictures posted by his wife, Reddit users argued that because there are 3 bowls in the picture, then Lindelhof must be with his wife and her mother - instead of training with the team. This would then imply that he would not be playing in the upcoming gameweek. Once more users saw this comment, additional posts were dedicated to this topic with even more detailed analysis⁶.

These claims were correct and Lindelhof missed the gameweek. FPL recommender systems focused entirely on statistics would have missed this, and might have even

⁵https://www.reddit.com/r/FantasyPL/comments/b79ekw/lindelof_theory/

⁶https://www.reddit.com/r/FantasyPL/comments/b4yzcl/maja_nilsson_lindel

recommended buying the player because he was in great form in the previous matches. We hope that by automatically including the opinions of experts and the general public, we can cater for such events.

1.2 Research Question

The main research question that will be examined throughout this dissertation is;

Can fantasy football prediction models be improved when incorporating statistical match day data with external non-statistical data?

1.3 Research Aims

By understanding the problems with statistics-based predictors, we aim to solve these pitfalls by incorporating data from different sources. News-articles, FPL specific blogs, betting market odds, and even tweets using FPL specific hashtags will be considered for their viability in this project. These external data sources allow the automatic feedback inclusion from experts and fans alike; which will then be used as additional parameters incorporated into our previously obtained statistical performance prediction per player. This would allow our model to incorporate external factors such as midweek game lineups and performances, managerial decisions, rotations and injuries together with unexpectedly good or bad performances in previous weeks. Ultimately, we aim to determine to what extent fantasy football predictors can be improved when statistical data is combined with information obtained through both betting data and public opinions.

In order to be able to consider text-based information sources such as tweets and news-articles, state of the art natural language processing (NLP) techniques such as named entity recognition and entity-based sentiment analysis must be incorporated. The main aim here is to investigate whether standard classifiers can classify text correctly in terms of finding an optimal player for the upcoming gameweek or whether a special FPL classifier would need to be developed to optimally solve this problem. Given a sentence discussing a players long-term injury should ideally result in a negative sentiment, while a player being in phenomenal form should result in a positive

score.

1.4 Research Challenges

When answering the research question posed below, several different challenges were encountered across all the different data types;

- a. The Twitter API is very restricted in terms of maximum number of calls per month and how many tweets were able to be extracted per call. Because of this, testing to find the optimal query parameters was extremely difficult.
- b. Tweet contents are very informal and often use shorthand or emojis, causing standard semantic analysis classifiers to misinterpret content.
- c. Similarly, player names are often shortened (eg. TAA Trent Alexander Arnold), or even misspelt. Many of these more common observations could be catered for through techniques described below.
- d. Collecting sufficient historical data and organizing it in a manner that could be useful for predictions was very time consuming due to the lack of available data and different sources that needed to be combined.
- e. Creating a robust system that can automatically gather the latest data required lots of time handling all the corner cases.
- f. Gathering odds required depending on a set of APIs that were unreliable at the start of the season and often did not provide the full odds per fixture initially.
- g. Comparing the different learning algorithms and their performances was extremely time consuming. Even then, optimizing the parameters to achieve optimal results depended heavily on the constantly updating test set.

1.5 Dissertation Outline

Chapter 2 discusses all the relevant literature that was reviewed in preparation for this dissertation, most importantly delving into detail on different learning algorithms, sentiment analysis techniques, language modelling and information aggregation methods.

Chapter 3 presents the current state-of-the-art in fantasy football recommender systems and other areas that use both statistics and human sentiment data to make predictions for future events.

Chapter 4 builds on the literature review, outlining the application architecture and implementation together with all the design decisions made throughout development together with all the challenges faced.

Chapter 4 discusses the results together with an in-depth evaluation of these results. The applications usability, performance and prediction accuracy are also assessed.

Chapter 2

Background Reading

2.1 Learning Algorithms

Machine learning algorithms are continuously becoming more advanced; improving their methods, accuracy and ability to solve increasingly harder research questions. Due to the nature of this dissertation, we explore various algorithms within the supervised and ensemble learning methodologies. Given that our historical seasonal performance dataset also contains the number of points achieved in that gameweek, our dataset is automatically labelled and thus becomes a supervised learning problem. This is because our research goal can be rephrased to trying to find a full squad of players that will score above a certain number of points in the upcoming gameweek. Therefore, unsupervised learning techniques are deemed out of scope for this thesis.

2.1.1 Supervised Learning

Supervised learning refers to the group of machine learning algorithms that generate a function by analysing a set of labelled training data. This function is then applied to previously unseen data, predicting the classification labels by generalizing an optimal algorithm from the training data to unseen scenarios [7]. In this project we explored the performance and accuracy of Support Vector Machines (SVM) as a baseline model. SVMs were chosen because of their reported success in Thapliyas captaincy prediction algorithm [8].

2.1.1.1 Support Vector Machines (SVM)

Support Vector Machines are a learning algorithm that are generally used for classification and regression problems [9]. A separating hyperplane is used to separate all the data into the different classification classes [10].

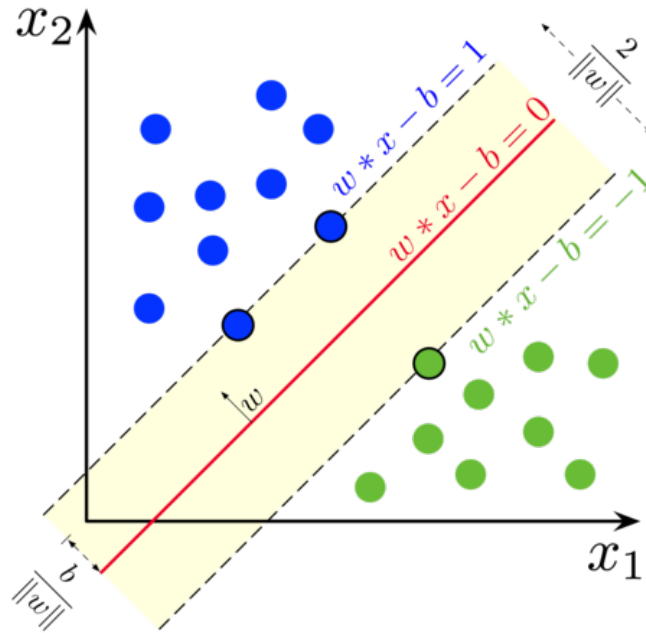


Figure 2.1: Linearly separable SVM [1]

The algorithm takes a weighted feature vector x_i and label y_i pair as parameters, where x_i is the data point and y_i is the points label. The goal of the algorithm is to find an appropriate decision surface that can correctly separate the different variable classes. A separating hyperplane is defined attempting to find the maximum possible margin; meaning that the distance from the nearest point on both sides of the hyperplane is maximised. Two supporting lines are also added, one per side of the separating hyperplane; defining the distance between the nearest point on that side and the hyperplane. By maximising these support distances, support vector machines will effectively reduce their classification error due to the larger margin for error between the different classifications [9].

Optimal hyperplanes are decided by minimising $|w|^2$ w is a vector perpendicular to the hyperplane [1]. Through this formula, the hyperplane orientation is deduced and whereas b decides the hyperplanes position. Through iteration of all the samples

in the test set, the SVM updates the values for w and b , learning about the optimal hyperplane orientation for all the data points in the test set.

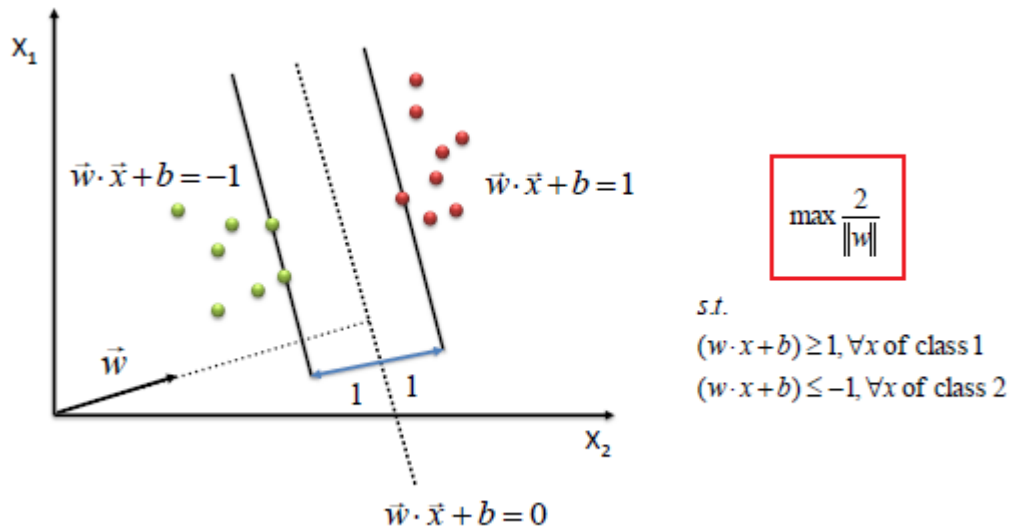


Figure 2.2: Optimising hyperplanes to maximise margin between plane and nearest point

Once the SVM is fully trained, predictions for all the previously unseen data points are made using the decision function $f(x) = w \cdot x - b$ the decision function outputs a positive number, the data point is labelled as a positive class, while if the decision function outputs a negative number, the point is labelled as a negative class [1].

SVMs are a useful tool for handling non-regular data or data with unknown distributions, capable of robust predictions. By making use of the regularisation parameter, the algorithm can effectively help reduce the risk of over-fitting to the testing data. In addition, by applying the kernel trick mapping non-linear separable data into a higher dimensional space to find a hyperplane that can separate the samples effectively [11], expert knowledge can be added into the model [12]. SVMs are defined as a convex optimisation problem which means that there is always a unique solution since there are no local minima. As discussed by Joachims, SVMs are extremely powerful when handling text data through techniques such as the bag-of-words algorithm which generates a large number of unique features [13].

2.1.2 Ensemble Learning

When considering important decisions that can have significant implications, it is considered good practice to seek a second and even a third opinion before deciding. Through this method, we typically weigh the gathered individual opinions and compare them before reaching an informed decision. The benefits of implementing such a technique for decision making systems has been recognised and is known under various names including multiple classifier systems or ensemble based systems [14].

Ensemble learning methods follow this approach of gathering several predictions, using a combination of different learning algorithms to obtain better predictive performance than could be obtained from any single algorithm individually [15]. They are particularly useful when dealing with both small and large datasets. When the amount of available training data is too large for a single classifier, the data could also be partitioned into smaller subsets. Each subset being used to train a separate classifier which can ultimately be combined through a prespecified combination rule. On the other hand, if there is too little data to sufficiently train a model, through various techniques such as bootstrapping, boosting and bagging, the use of the available dataset will be maximised and used to its full potential. On the other hand, techniques such as bootstrapping, boosting and bagging, can be used in scenarios where only small datasets are available to train a model.

Bootstrapping is the practice of using random dataset sampling with replacement. By doing so, we can better understand the variance of a dataset. Each bootstrap-sample consists of a random data sample drawn with replacement and treated completely independently from all the other bootstrap-samples [16]. While this technique is generally used for small subsets of training data to help provide an understanding of the standard deviation of the dataset, this method can also be used on whole datasets when sufficiently small [17]. This is demonstrated in figure 2.3, where the super population is being split into three subsamples with replacement. Each sub-model in the ensemble classifier would then be trained on these different subsamples to optimise the results, even on small datasets.



Figure 2.3: Bootstrapping dataset example

Bagging, or bootstrap aggregation, is an ensemble technique is the procedure used to generate trees in a high-variance format ensuring that each tree in an ensemble method is sufficiently different. In a repeated fashion, a subset of all the data and features within the dataset are used to create different models, with the final prediction being an aggregation of all the different predictions by each unique model.

Finally, boosting refers to algorithms that utilize weighted averages to transform weak learners into stronger learners. Unlike the technique used in bagging methods, where each tree is run independently and their predictions aggregated at the end, boosting runs its models one at a time. At each iteration, the previous model specifies which features the next model should focus on. In figure 2.4 we see that the weak learners represented by Box 1, Box 2 and Box 3, all misclassify a few results. The stronger learner that is the aggregated results of each of these boxes is then much more successful in understanding the dataset as reflected by its correct predictions [17].

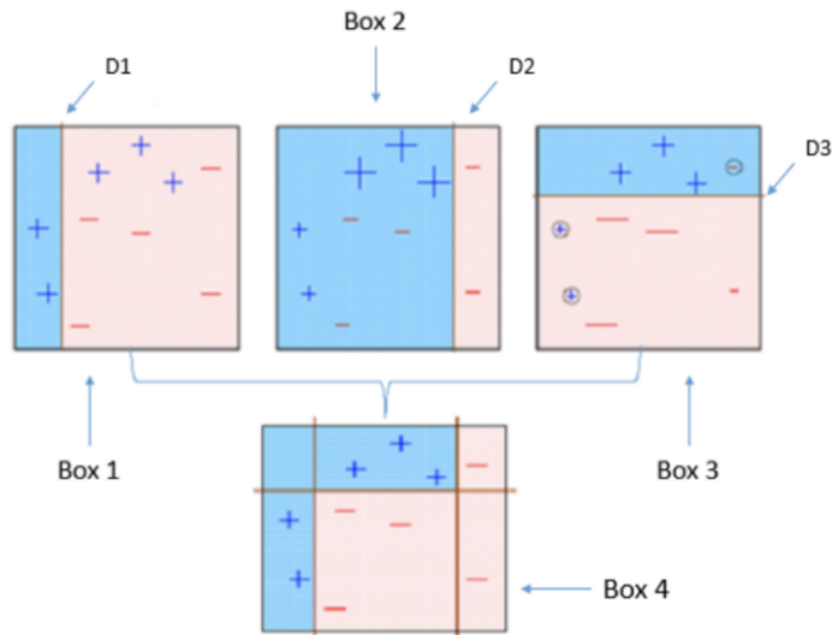


Figure 2.4: Bootstrapping dataset example

2.1.2.1 Random Forests

Random forests are an ensemble learning model using a combination of tree predictors such that each tree depends on the values of a random vector sampled independently. Random forests are predominantly used for classification and regression problems. A forest of different decision trees is generated by selecting data and applying random and independent vectors [18].

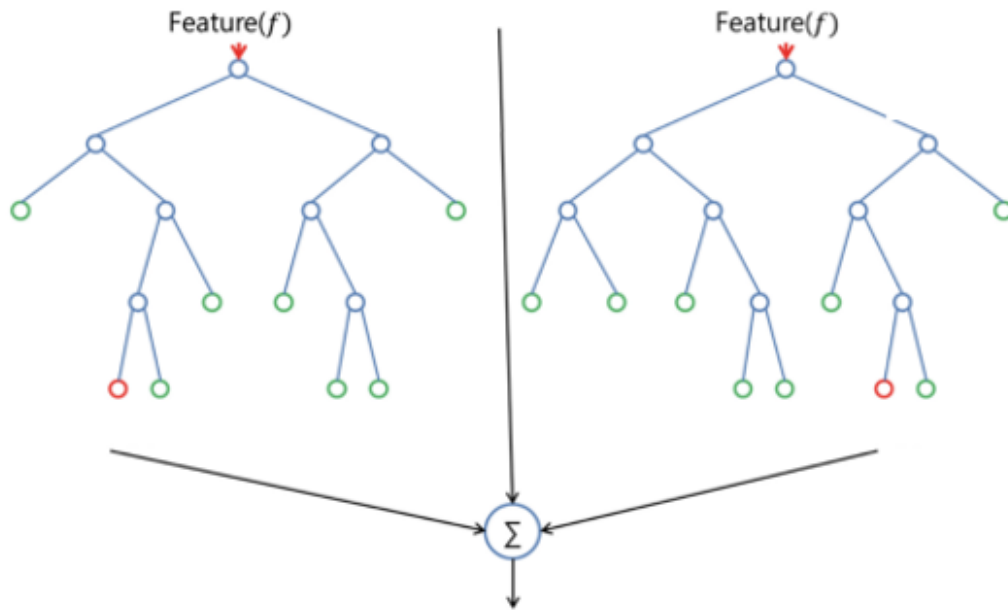


Figure 2.5: Random Forest structure [2]

Single decision trees are a popular method for tackling different machine learning problems, but because they tend to overfit data, might not commonly be the best choice for more complex models. By combining and averaging the results of different decision trees tackling different subsamples of a dataset, we overcome the overfitting problem faced by individual trees. The method for applying random forests makes use of bootstrap-aggregating or bagging. Given a training set and their responses, the bagging algorithm repeatedly selects a random sample (with replacement) of the training set and fits trees strictly to those samples [18]. By using a large enough forest we are confident that overfitting is not likely due to their independent training techniques.

2.1.2.2 Gradient Boosting Machines

Gradient boosting is another form of ensemble learning model that is used to solve regression and classification problems, producing a model in the form of weak prediction models such as decision trees. Similar to other boosting models, gradient boosting machines build the model in a stage-wise fashion, allowing optimization through a differentiable loss function [19].

Similar to linear regression models which assume that the sum of its residuals is 0, gradient boosting models repeatedly leverage patterns in residuals and strengthen the model through several weak predictions, effectively making the base model repeatedly stronger. Gradient boosting machines work by first fitting a regular decision tree on the data and based on the error residuals calculated (actual target value minus the predicted target value), a new model based on these error residuals as the target variable with the same input variables is created. This process is repeated until the sum of residuals converges or overfitting is observed. GBMs come with a number of hyper-parameters that allow for fine-tuning the model specifically to fit the dataset in question. Through parameter tuning, we can prevent both under and overfitting [20].

Soon after the introduction of the gradient boosting algorithm, Friedman proposed that at each iteration of the algorithm the learning models should be specifically fit to a subsample of the training-set selected at random. The proposed technique is comparable to the bootstrap-aggregation method described in section 2.1.2. This modification saw a significant improvement in the gradient boosting models accuracy [21].

Most of GBMs strengths lie in their flexibility and capability to adapt to change due to its many parameters. These can be divided into 3 general categories, which we will discuss in the upcoming sub-sections [22] [23] [24];

1. **Tree-Specific Parameters:** affect each independent sub-tree in the model;
2. **Boosting Parameters:** affect the boosting methods in the model;
3. **Other Parameters:** affect various functions within the model.

2.1.2.2.1 Tree-Specific Parameters

The following parameters alter the way all the trees within the model work [23] [24];

1. **Min_samples_split:** Defines the minimum number of observations which are required in a node to be considered for splitting. This parameter is used to control over-fitting, higher values can prevent a model from learning relations that may be highly specific to a particular sample [25].

2. **Min_samples_leaf:** Defines the minimum number of observations required in a leaf node. This parameter is used to control over-fitting in a similar fashion to `min_samples_split` [25].
3. **Max_depth:** Defines the maximum depth of any tree. Can be used to control over-fitting since larger depths allow models to learn very specific sample relations [25].
4. **Max_features:** The number of features that are considered when search for a most optimal tree split.

2.1.2.2.2 Boosting Parameters

Once the tree parameters have been defined, the boosting parameters are used to specify how the bootstrap-aggregation method will work [23] [24].

1. **Learning_rate:** Determines the impact of each sub-tree on the final prediction outcome. An initial estimate is defined which is repeatedly updated through the outputs of every tree. This parameter defines the magnitude by which this change occurs.
2. **N_estimators:** The number of sequential trees that are modelled. Overfitting may occur if the number of trees is excessively high.
3. **Subsample:** The fraction of observations that will be selected for every tree through random sampling.

2.1.2.2.3 Miscellaneous Parameters

1. **Loss:** Defines the loss function that is minimized in every split.
2. **Random_state:** The random seed number ensures that random numbers will be generated every time.

2.2 Natural Language Processing

2.2.1 Named Entity Recognition

Named entities are defined as phrases that contain a number of names identifying people, places, organisations, products and many more. For example;

U.N official Ekeus heads for Baghdad.

[ORG U.N.] official [PER Ekeus] heads for [LOC Baghdad].

The above sentence contains three named entities; the organisation U.N, the person Ekeus and the location Baghdad. Named Entity Recognition (NER) is the subtask of information extraction whose aim is to tag proper nouns in unstructured text with predefined classifications. Early models made use of primitive handcrafted rule-based algorithms, but modern focuses have shifted focus to automated methods through machine learning models [26].

Due to the nature of this task, supervised learning techniques that study examples of named entities across a large dataset requires an often unrealistically large annotated corpus. Asahara and Matsumoto combined a statistical Part-Of-Speech (POS) tagger with an SVM-based model that could extract named entities using these annotated features achieving an F-score measure of 87.21 [27]. As is common with supervised NER models, the performance of the system heavily depends on vocabulary transfer; the proportion of words without repetition that appear in both the training and testing corpora. Because of this problem, it is evident that language factors and textual genres are also an important factor to consider when reporting accuracies of models. Poibeau reported that systems tested on different collections from their testing corpus would report a drop in performance and recall between 20% to 40% [28]. This is clearly a significant problem because it implies that each model is closely tied to the training dataset and may lose accuracy in previously unseen examples should there be new words.

To solve the lack of large training data available, clustering and other unsupervised learning models have recently become the state-of-the-art in NER. Etzioni et al. investigate the viability of Pointwise Mutual Information and Information Retrieval (PMI-IR) as a feature that could be used to classify a named entity as a given type [29]. Developed by Turney [30], PMI-IR determines the measure of dependence between

to expressions through web queries. A high PMI-IR value indicates that expressions tend to co-occur, implying a high correlation probability. Through this method, Etzioni could create features for every located entity and automatically generate discriminatory phrases per entity. For example, for a located entity such as London, phrases like is a city or nation of would be generated which can then be used to correctly classify entities. This method handles the primary issues outlined in supervised learning methods outlined above, where unseen words would not be classified. Using this example, if a new city name is mentioned in the sentence, Tokyo is a city in Japan; even though Tokyo was not seen in the training data, the model could infer that it is a city in the country Japan.

2.2.2 Sentiment Analysis

Sentiment analysis aims to methodically identify, extract and score sentimental ratings and distinct information from entities within textual data. In its most primitive form, the task of sentiment analysis is to classify the overall polarity of a text document understanding whether the opinions expressed within the document are positive, negative or neutral. Pang and Lee investigated whether standard topic-based categorization techniques would be sufficiently suitable for sentiment classification or whether special sentiment-categorization techniques would need to be developed [31]. The dataset used was made up of a number of English movie reviews together with their overall rating thumbs up or thumbs down. To do this, three standard models were compared; Nave Bayes classification, maximum entropy classification and Support-Vector Machines. Interestingly, they reported that the standard machine learning algorithms incorporated with natural language processing techniques provided accuracies of up to 82.9% when only considering unigrams within the text. When compared with a set of human-generated baselines that gave a sentiment score for every word in the dataset, the algorithms outperformed them by 13.9%. While these results look promising, they were not able to achieve new highest accuracies, but argued that this was because of the structure of their dataset, where reviewers would set up a deliberate contrast to earlier discussions causing bag-of-words algorithms to misclassify such instances while humans would have no problem identifying the true sentiment of the review. To solve this issue, where the whole is not necessarily the sum of its parts, Turney suggests that

some discourse analysis would be required that makes use of more advanced technology than the current positional features used prior [30].

In practice, labelled documents are very rare, and the general trend is to work with unlabelled data. Because of this, understanding how to exploit unlabelled data for text classification has become a very active research problem [32] [33] [34] [35]. Nigam et al. proposed an Expectation Maximization (EM) algorithm combined with a Nave Bayes classifier, which makes use of EM to dynamically produce labels for unlabelled data whilst simultaneously improving the system [33]. This allows the Nave Bayes algorithm to incorporate unlabelled data into supervised learning methods. The algorithm works by first training the model on all the available labelled documents, and after assigns probabilistic labels on all the unlabelled documents. A classifier is then trained using the new labels, showing results that reduce classification error on unlabelled data by up to 33% on real-world tasks.

Similarly, Joachims implemented a modified SVM that could exploit unlabelled data sources; commonly referred as the Transductive Support Vector Machine [34]. Given insufficient labelled data, the SVM may incorrectly learn from the dataset and mistake noise to be an important factor. The idea behind TSVMs is that a segment of the test-set is added into the training data, attempting to make training the model less ambiguous by increasingly populating denser areas while optimally leaving the ideal maximum-margin solution as a relatively low-density area.

Both Dumais et al. [35] and Joachims [34] show that SVMs are extremely well suited for text classification problems because of the following characteristics;

1. Large number of features Each stemmed word is considered its own feature
2. Document vectors are sparse For each document, the equivalent document vector contains very few non-zero entities.
3. Irrelevant features are sparse Joachims suggests that most features are relevant and should not be removed [34].

TSVMs inherit most of these benefits from SVMs but can also take advantage of string co-occurrence patterns; words that are likely to occur in conjunction with each other, due to their prior knowledge on the learning task [34].

2.2.3 Microblog Natural Language Processing

Microblogs such as tweets provide new challenges in natural language processing, mostly related to information extraction and sentiment analysis. Microblogs represent an important source of information through which named entities occur repeatedly with very distinct contexts per blog due to the individual opinions expressed independently. Twitter has around 300 million active users posting over 500 million unique tweets every single day; the fastest growing network in terms of activity [36]. Because of this large volume together with the posting of independent user opinions, harvesting sentiments through tweets could provide extremely useful for companies looking to understand how users react to certain products or events.

As discussed earlier, regular NER and NEL systems are typically developed and evaluated specifically to work on lengthier, carefully written texts such as news articles and blogs. Applying natural language processing techniques to tweets and microblogs in general is a challenging rising research area as shown in [29] [37] [38] [39]. The 280-character limit imposed on tweets makes users tend to avoid grammar and proper spelling, whilst also making use of emoticons, abbreviations and hashtags all of which could have an important role on the meaning and sentiment of the tweet [37]. Because of their short, noisy and unstructured nature, information extraction from tweets is generally performed through a pipeline using different methods involving tokenisation, part-of-speech tagging, named entity recognition and entity disambiguation [37].

State of the art named-entity recognition methods that would generally have an accuracy of 85-90% on standard text articles were reported to only achieve 30-50% when analysing tweets [38] [40]. To attempt to solve all the above problems, current research targeting microblog specific information retrieval techniques have used implementations involving Conditional Random Fields (CRFs) [38]. Together with CRFs, a special focus is being dedicated to text normalisation methods as a way of removing extra linguistic noise prior to introducing the natural language processing pipeline [41]. Through these methods, Ritter et al. was able to prove that standard NER systems perform poorly when applied to tweets but can be improved by building tools that were trained on unlabelled, domain-specific and non-specific data showing considerably improved results over standard news-trained models. Their individually trained POS tagger outperformed the Stanford POS tagger, reducing error rate by 41% [38].

Once named entities have been correctly located within microblogs, Go et al. compared integrating different n-gram features together with their respective POS tags into the training of supervised models including Support Vector-Machines, Maximum Entropy (MaxEnt) and Nave Bayes [42]. Their results showed that the MaxEnt model trained through a combination of unigrams and bigrams outperformed all other models that used POS tags by almost 3%, reaching a maximum accuracy of 82.7%. Part-of-Speech tags were used as features because different contexts can cause a word to have a different meaning, such as the word over being used as a verb could imply negative sentiment, while when used as a noun may simply refer to the Cricket terminology. Pak & Paroubek reported contradictory findings, stating that adding POS tagged features into his n-gram models improved sentiment accuracy on tweets, arguing that they provided a strong indicator of emotional texts within tweets [39]. This is likely very dependent on the dataset and objective of the model in question where syntax and common phrases can be easily predicted. Similarly, Barbosa and Feng stated that the use of n-grams on tweet data may negatively affect classification performance because of the large number of infrequently used words together with the different vocabularies used by each individual Twitter user [43]. Instead, they proposed a framework that made use of Twitter specific features such as retweets, hashtags, replies, punctuation and even emoticon usages. By training SVMs using these features, the sentiment accuracy improved by 2.2% when compared to standard SVM models trained only on unigrams.

Extending on these previous works, Saif et al. investigated a different approach to sentiment analysis through tweets; namely applying semantic features and concepts across several tweets containing that entity occurrence as an additional feature to the overall sentiment prediction [44]. The features compared in their report were unigram, POS-based, Sentiment-Topic and their new Semantic-based models. They state that through their methods, the increases in precision, recall and F-score rose by at least 6% throughout all their tests on different datasets.

Chapter 3

Literature Review

In this section, the current state of the art in fantasy football predictions will be analysed and discussed. Any techniques will be analysed and compared to methods we propose, together with why they differ. Further, we analyse other works that combined different data-sources to make improved predictions in different areas. The general inclination observed in previous works has been to use historical statistical data in combination with machine learning methods to predict future scores.

Strictly using previous gameweek performances together with a Gaussian Nave Bayes algorithm, Thapliya was able to predict future performances with a reported accuracy of 86% [8]. The model aimed to a set of 3 players that would score 6 or more points in the upcoming gameweek, essentially predicting the best captaincy picks. As discussed in the FPL rules section, each team is allowed to pick a single player per week to be the team captain. This player is then given a double points multiplier, making it especially important to pick a good captain consistently. While this method acts as a good baseline method, the final implementation is still very lacklustre, with injuries and additional factors such as one-off performances not being included in the final predictive model. By building on this technique, we can extend the algorithm to pick the best full team of players for the upcoming gameweek. In addition, by incorporating a better statistics dataset together with all the other proposed data-sources we hope to improve on both accuracy and the average weekly points achieved.

Bonomo et. Al derived a mathematical model attempting to predict ideal line-ups per gameweek in the Argentinian first division [45]. Very similar to our goal

of predicting the optimal weekly line-ups by incorporating previous information with knowledge gained from press-reports and manager updates before games. The aim of this model is to build an index that can produce reasonable representations of what will happen in future gameweeks. They concluded that simply analysing the players point average in recent matches alone is not a good indicator due to the lack of key match characteristics that are considered. Most importantly the opponent, the matches home or away status and the current performance of the team. To improve the model Bonomo extended the points averaging system by applying a weighting strategy based on the following three factors;

1. Home or away status of the clubs next match. Weight of 1.05 applied for home games, 0.95 for away games.
2. League table position of next opponent. Weight of 1-1.05 for bottom five teams, 0.95-1 if in the top five.
3. Current performance or situations of player or club. Up to 5% if the player is on a winning or scoring streak, up to -5% if on a scoreless or winless streak, but even applied after recent external tournament matches.

Initial teams were selected using two separate techniques. First, initial teams were selected based on the computed index to maximise the teams whole-season number of points. The second initial team was built in a non-myopic fashion, only aiming to consider the upcoming gameweek. Interestingly, the non-myopic model outscored the myopic initial team across the entire season regardless after transfers are considered. They confidently state that the longer the season goes on, the better the model can perform based on more relevant data to learn from.

Bonomo also investigated a posterior model that had all the available seasonal information to investigate factors that could help future models. This model calculated points obtained by the optimal team and a comparison of points between different available formations.

Match-outcome prediction models have many potential applications, most commonly in models attempting to beat betting markets. However, we can incorporate similar methods into our own system to improve predictions. Similar to Bonomos work converting expected results into fantasy football recommendations, we can assign each

player with both an attacking and defending returns probability. Owen states that most previously published work focused on Generalized Linear Models (GLMs), which assumes that all the model parameters representing the underlying statistics and abilities per team remain constant over time [46]. To improve on the works, a Dynamic GLM is introduced that evolve based on past matches to better predict future matches. Predictive probabilities for every fixture result would be derived based solely on the results gathered prior to that round within the season, adapting after each gameweek results are updated.

In more sophisticated methods, Matthews presented an innovative fantasy football predictor that consisted of belief-state MDP algorithms combined with Bayesian Q-Learning to train models on the past five years of football data [3]. Through this technique, expert knowledge was combined with statistical player data in many different forms based on the machine in question. Their most successful model used a state of the art Bayesian Q-learning model to handle the uncertainty, achieving a score of 2222 points, placing the machine within the top 500 players. The original model was naive; acting myopically only considering the single next gameweek. Even with all these restrictions, this model still achieved a very respectable score of 1981, effectively ranking it at position 113,921. By extending this model to also look back on data from the previous season, the model leaves out all the players who did not appear but still improved performance slightly, upping the score to 2021 with a rank of 60,633. While the scope of this dissertation is different, extending the work to compete with the real FPL leader board while following the official set of rules would be an important next step to fully understand exactly how important the different data sources are in terms of rank improvements.

Paid subscription models such as Fantasy Football Fix offer a similar statistically based model that works by analysing the same statistical trends that will be used in this paper, namely all the transfer data and the points predictions using historical season data ¹. While their model has successfully been able to predict player price changes with an accuracy of 98.7%, predicting future gameweek performance had much weaker accuracy due to their sole focus on statistical data. The application states that its primary goal is to optimize a players team by providing recommended transfers to the individuals team; but since this is only done through statistical analysis and informa-

¹<https://www.fantasyfootballfix.com/>

tion on the upcoming EPL fixtures, the model has no rationalization and cannot predict injuries, rotations, managerial decisions, tactical changes and the unpredictability of football. While our application cannot solve the unpredictability of football, by incorporating automated human, and more importantly, expert feedback into our model, we will effectively be able to also consider all the other external factors that have been hindering previous models so far in an automated way.

Godin et al. proposed a technique that involved combining historical statistical data with large numbers of tweets to predict football match results in order to try and beat bookmaker accuracies [47]. They show that combining multi-stream data proves to be useful when attempting to predict match outcomes. By considering several different factors, including sentiment analysis, tweet volume for specific hashtags, and even unigrams and bigrams, they were able to produce the following results;

1. a. Considering the previous 5 games of statistical data was able to correctly predict 62% of upcoming match results;
2. Twitter volume for different hashtags and keywords is not a good metric, only correctly predicting 50% of all results;
3. Sentiment analysis was also not suitable for microblog analysis, predicting just 52% of all results correctly.

By only considering tweets that mentioned just one team or their nicknames together with a predicted score, the algorithms accuracy improved to 61%. Once the individual feature vectors of both statistics and tweets were used to create a single feature vector, the prediction accuracy rose to 68%. Across this 14 gameweek timeframe, the bookmaker prediction accuracy was 67%. While these results dont significantly outperform bookmaker predictions, it is clear that incorporating human feedback has significance. While this method of tweet understanding might be viable for FPL predictions, the number of different scenarios is much larger in our domain. The assumption that a team being mentioned in a tweet implies that they are expected to win cannot be translated to FPL. Firstly, there is a significant drop in FPL specific tweet volume, and secondly, it is not safe to assume that mentions mean that the player is expected to perform well. Our task of predicting a full team of 11 players that will perform well

in the next gameweek is much harder to predict using twitter and alternate methods would need to be explored.

Bahrami et al. presented a technique to predict upcoming public protests by using both statistical data and twitter posts [48]. Specific to protests held after the US 2017 elections, their model would iteratively search for tweets containing words such as protest. Once a predetermined threshold is exceeded, the most common hashtags are collected to help provide an understanding of what the protest is about. Additional tweet details such as the polarity of tweets calculated through unigrams, the number of violent words per tweet and the state in which the tweet is published are used to determine whether a protest is likely. External statistics used included the percentage of Trumps vote in each state; providing a clear understanding of the general sentiment towards Trump per state. Bahrami et al. report that combining statistics with social media content has very powerful predictive possibilities, reporting an accuracy of 85.5% across 6 months of data. As expected, the accuracy increases significantly with more data having an average accuracy of 62% in the first month, increasing to 78% by the second. By iteratively filtering tweets depending on event-specific hashtags we can incorporate real time tweets into our predictions. In our case, the events would be the specific gameweeks, with twitter users posting their predictions, possible rumours or even their current team line-ups using the hashtag #GWxx; where xx represents the current gameweek in the season.

Chapter 4

Design & Implementation

This project requires interactions with large amounts of raw data retrieved from a variety of different data sources to achieve its task. In this chapter, an overview of the application will be presented; discussing design choices, tools and technologies used. We will also analyse several methods used to handle the different data sources and produce a set of predictions that accurately reflect on the data contents.

4.1 Application Overview

The aim of this dissertation is to generate predictions for future FPL scores through various data sources. By combining all the different predictions to form a single prediction, we expect to have more powerful predictive power than achievable by any of the individual data sources. This research is based on the principle that statistical factors combined with human expertise and the opinions of large crowds could improve the model by automatically factoring in non-statistical criteria. To do this, individual components responsible for all the necessary steps required to get a prediction from their respective data source are developed. Once all components are developed, a final aggregator is used to gather and combine the results from all the data sources. Through a series of tests on the English premier league 2018-2019 season, we identified the most optimal weighting strategy that produced the most accurate results based on how good each data source was at predicting player performance.

4.1.1 Application Architecture

As discussed above, the application is divided into multiple core components that are each responsible for the gathering and handling of data from their data source and using this data to make a prediction. The components discussed below are:

1. Statistical analysis;
2. Twitter analysis;
3. Blog and news-article analysis.

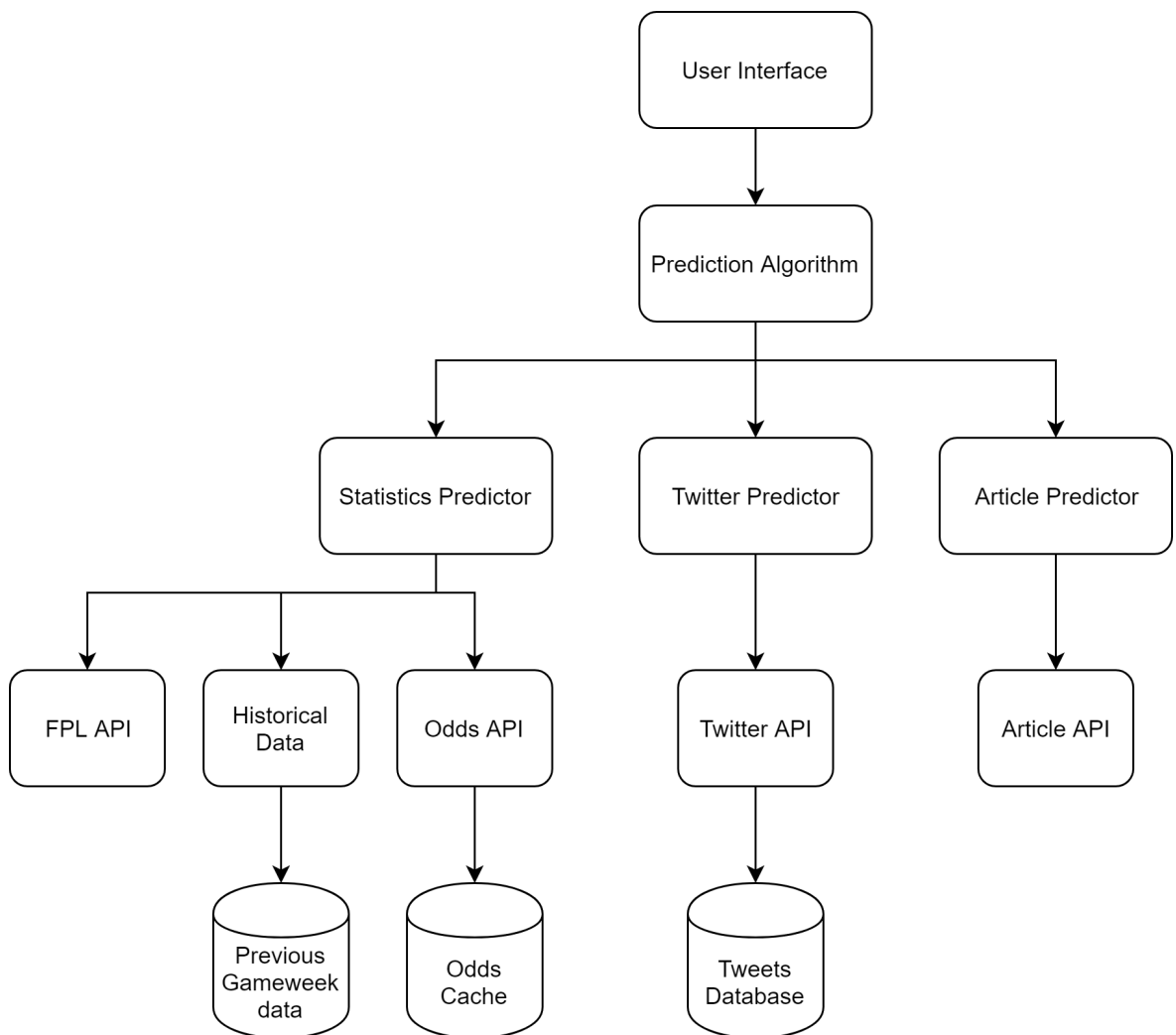


Figure 4.1: Proposed high-level architecture design

4.2 Prediction Algorithms

4.2.1 Dataset

Our dataset is a combination of different datasources containing all the latest available FPL specific data for the current season. As a result, the dataset is very varied; being extremely limited during the first gameweeks of the season, and being very large towards the end. For every gameweek within the season there are over 500 players, each with unique statistical measures that must be gathered. At the early stages of an FPL season we would only have limited training data, meaning that any algorithm used would need to be suitable to handle such datasets effectively. Similarly, towards the end of a season, the amount of data available will be increasingly huge. For example, by the 38th and final gameweek, we would have 37 gameweeks of statistical data for over 500 players per week. Added to this, we would also have the results obtained from around 300 different news articles and blogs together with an average of 400 tweets per gameweek.

Further, our dataset is an extremely unbalanced one. Our goal of predicting players expected to perform well in upcoming gameweeks can be rephrased to predicting players who will earn 6 or more points in the next gameweek. Due to the way points are earned in FPL, 6 points is a good baseline that should be expected for each player on a team in order to be successful. All players score 1 point for playing up to 60 minutes in a game, and 2 points for any additional minutes. Added to this, goalkeepers and defenders get 4 points for a clean sheet, and forwards get an extra 3 points together with a bonus point for scoring one goal. Midfielders get 1 point per clean sheet and get 4 points per goal, and 3 points per assist. This makes scoring 6 points very achievable for any player expected to perform well. However, as seen in figure 4.2, the number of players that achieve 6 or more points is extremely small when compared to the overall set of players. Out of the 16,384 players achieving points in every gameweek for the past 3 years of FPL data, 9533 players earned 0 points (58.18%), while only 1399 players earned 6 or more points in the gameweek (8.54%).

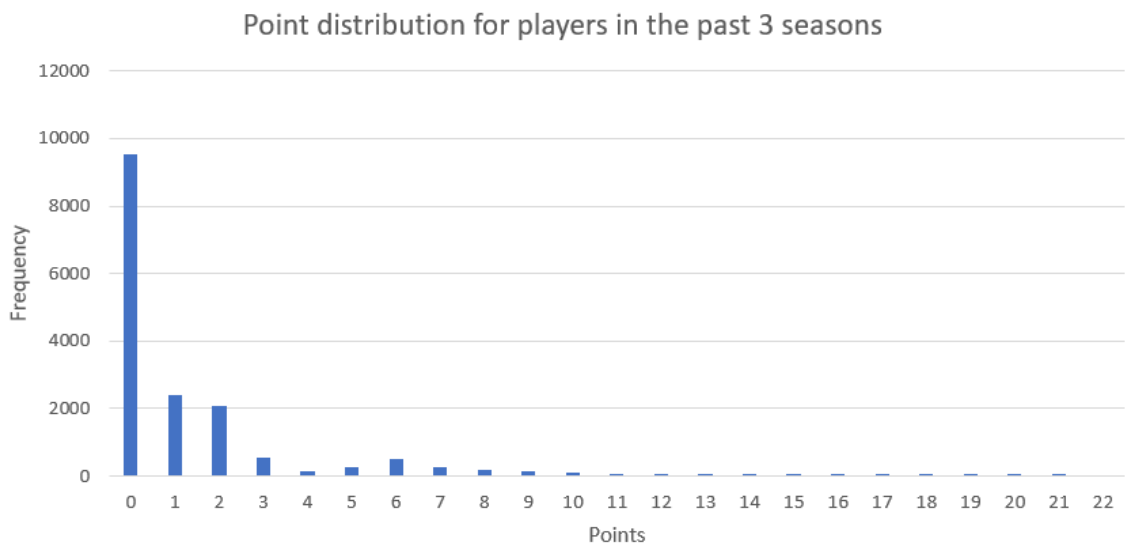


Figure 4.2: Points scored per player per gameweek in past 3 seasons of FPL data

4.2.2 Machine Learning Algorithms

It is clear that we needed to incorporate an algorithm that could deal with both small and large unbalanced datasets. We considered 3 primary algorithms; SVMs, RFs and GBMs. SVMs provided a very good baseline, especially during the early stages of the season - however proved to be unusable in the later stages of the season due to their excessively long training times when making use of large datasets. We also found that they were not very effective at dealing with unbalanced datasets in general, meaning that they were generally very weak at predicting the rarer outcomes and generally focused on extremely standard team selection templates. However, the main downside with SVMs was that they are not able to specify their probabilistic confidence values for any selections meaning that there is no effective way of selecting the most confident picks. Additionally, SVMs are extremely susceptible to both over and under fitting the dataset and as a result are very dependent on hyper parameter and kernel function values. Because our dataset changes drastically per week we would need to manually fine-tune the models repeatedly every week to achieve optimal results.

Due to these constraints brought about by the dataset, ensemble methods were expected to provide more benefits over supervised learning algorithms. Both RFs and

GBMs provided exceptional predictive power and both worked extremely well with both small and large unbalanced datasets - making both of these algorithms ideal candidates for our task. Ultimately, because GBMs follow the boosting technique explained above, we found that by learning from previous iterations of weak-learners, GBMs were more able to correctly predict unlikely outcomes; such as underdog teams unexpectedly winning games. Additionally, in the gameweek after such an unexpected event occurs, GBMs were less likely to fall for the trap of recommending those players again off the back of an unexpectedly large points haul. On the other hand, because of RFs bagging technique, any data picked from unlikely events will have a larger than ideal impact on the final predictions; often recommending players simply due to recent one-off good performances. While this can be negated by increasing the number of random trees within an RF, we still found that GBMs outperformed them throughout most gameweeks in a season.

GBMs were eventually chosen as the prediction algorithm of choice for this task. Their flexibility and ability to handle both large and small amounts of data effectively meant that the algorithm will work effectively throughout all stages of the current season. The proposed plan for the application is that once developed, is expected to run automatically; retrieving all the necessary data, making the predictions, and deciding on the final prediction without any human interventions.

Optimising the algorithm to correctly fit the data without over or under-fitting was a crucial task that could severely impact the predictive power of the algorithm. Finding the optimal set of variables required for the task requires expertise and extensive trial and error techniques to fully understand the problem at hand.

The first step was to identify how well the different variables correlate to players scoring large point hauls in the next gameweek. To do this, another variable was added to the dataset called `isCaptain`. This is a boolean value with value 1 for players that score 6 or more points and 0 otherwise. This effectively makes the problem at hand attempting to correctly predict the value of this parameter for the upcoming gameweek. By using this variable, our model now becomes a more clearly defined classification problem rather than trying to predict the exact number of points that a player is expected to score which would lead to high variance and an extremely low accuracy in our testing.

To demonstrate the variance in all our parameters in reference to the `isCaptain`

value for that specific player and gameweek, the dataset is split into 2 rows. The first row represents the mean values for all variables when isCaptain is 0, and the second column is used as the mean values for all rows where isCaptain is 1. The final table of features together with the mean values for isCaptain is shown in tables 4.1 - 4.5

Table 4.1: Player metrics for the upcoming gameweek aggregated by isCaptain value

isCaptain	Player	Opponent	Opponent_FDR	isHome	Points	Minutes	Round	ElementID
0	...	10.564	2.965	0.486	1.323	56.539	16.616	2.675
1	...	10.068	2.552	0.566	8.218	85.714	18.523	2.4855

Table 4.2: Player metrics for previous gameweek aggregated by isCaptain value

isCaptain	Opponent_PrevWeek	Opponent_FDR_PrevWeek	isHome_PrevWeek	Points_PrevWeek
0	9.7115	2.703	0.479	2.643
1	10.633	2.879	0.445	3.571

Table 4.3: Player metrics match 2 gameweeks ago aggregated by isCaptain value

isCaptain	Opponent_2PrevWeek	Opponent_FDR_2PrevWeek	isHome_2PrevWeek	Points_2PrevWeek
0	9.405	2.616	0.448	2.301
1	10.317	2.788	0.473	2.973

By understanding these values, it becomes clear that variables such as bonus points (BPS), Transfers-Balance and Influence play an important role in predicting future performance. Meanwhile, factors such as opponent, round or even elementID seem to have no correlation with the player performance since their mean values seem to be very similar for both values of "isCaptain". To prevent the classifier from learning wrongly, these irrelevant values are removed before learning. While it may seem like factors such as the opponent 2 weeks ago has no difference, these features were kept in, because it implicitly affected the number of points scored in that week. To expand the available feature space, we then create interaction terms; variables multiplied together to produce extra variables. One example of this is creating a new variable called "OpponentFDR-Points-PrevWeek", which is the result of multiplying the fixture difficulty of the previous weeks match together with the points achieved. These variables allow us to get a better understanding of how factors such as match difficulty affected points earned in previous gameweeks.

Table 4.4: FPL provided statistics aggregated by isCaptain value

isCaptain	ICT_index	Threat	Influence	Transfers_Balance
0	2.618	8.402	10.935	2196.895
1	7.301	23.081	35.035	11971.045

Table 4.5: FPL player value and betting odds for all players aggregated by isCaptain value

isCaptain	Value	BPS	DefenseOdds	OffenceOdds
0	53.875	7.589	59.705	26.616
1	59.778	29.513	46.619	19.210

It also seems like DefenseOdds and OffenseOdds values extrapolated from the betting data - representing the likelihood of a team to keep a clean sheet and score respectively, are insignificant. This is because the table represents the mean values across all players regardless of their position. While goalkeepers, defenders and midfielders benefit from clean sheet points, forwards do not. This means that when considering forwards, clean sheets are likely useless except for showing a teams expected dominance in a game. On the other hand, goalkeepers and defenders rarely benefit from any attacking returns through goals and assists, making goal probabilities effectively useless for them too. To further understand how significant each variable is when attempting to predict future data, the dataset is split for every player position. Next, different GBMs are trained per dataset. When predicting unseen data in the test-set for midfielders, we observe the variable importance scores shown in figure 4.3. As expected, influence and ICT-index remain extremely important, but elementID is useless; matching our expectations from tables 4.1 and 4.4.

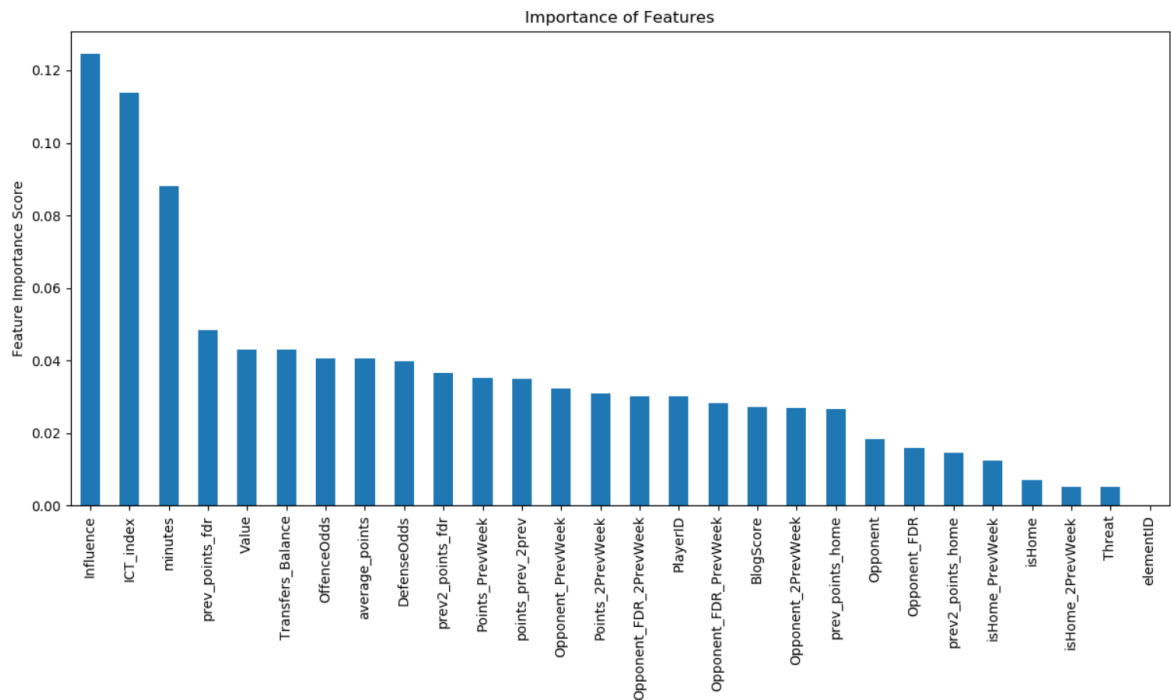


Figure 4.3: Feature importance score for midfielder predictions

Similarly, from the correlation matrix in figure 4.4, we can see the relevance of certain variables when compared to each other. Each cell indicates the relevance of the variable in its column to the variable in the current row. For example, when observing the correlations for Opponent-FDR-PrevWeek - the combination of the opponent ID and the fixture difficulty rating of the previous week, we clearly see that there is no strong relation with any other value. This means that if kept in our prediction model, this value could even affect the outcomes for the worse. On the other hand, when looking at the isCaptain value, we see that there are high correlations with the isHome, Threat, ICT-index and several others. This matches our hypothesis drawn from the mean-values table and variable importance figure above, where high variance indicates an important feature. Since "isCaptain" is the variable we are attempting to predict, through this correlation matrix, we can easily identify which variables will be important to our model.

Different gameweeks are then selected and all the previous gameweek data is used as training data - with the current gameweek used as test data. By selecting a wide range of different gameweeks from various phases of the season, we can ensure that our

so that optimal M-values for it are unstable. ... the qualitative nature of these results is fairly universal [21].

The recommended approach is to begin by setting a large value for M and tune the shrinkage parameter to achieve the most optimal results. Several further suggestions are mentioned across Friedmans papers, recommending subsampling without replacement with a value around 50%, the number of terminal node per tree ranging between 2 and 8 and the number of trees in a GBM to be within the range of 100 to 500 [21] [49].

The approach used in this paper involved a similar method of starting the model with all the default values and finetuning them each individually. By understanding the trends and estimated values that are recommended for each parameter, an array of several possible values are defined [22]. Through a 5-fold cross validation technique, models are repeatedly tested using all the different array values, and predictive performance is measured by measuring the accuracy as well as the AUC-ROC (Area Under Curve Receiver Operating Characteristics) curve. Through the AUC-ROC, we can identify how well our model distinguishes different classes by comparing the True Positive rate on the y-axis against the False Positive rate on the x-axis. The bigger the area covered (the higher the True positive rate converges), the better the model is at predicting correct classes.

In figure 4.5 below, we see an example of this technique, where the n-estimators parameter was assigned various values and the AUC score was measured for each instance. For each different parameter value, 5-fold cross validation is used to ensure that the results are accurate and not a one-off result skewing the results. The average AUC score across all cross-validation results is then picked and compared to the rest of the values.

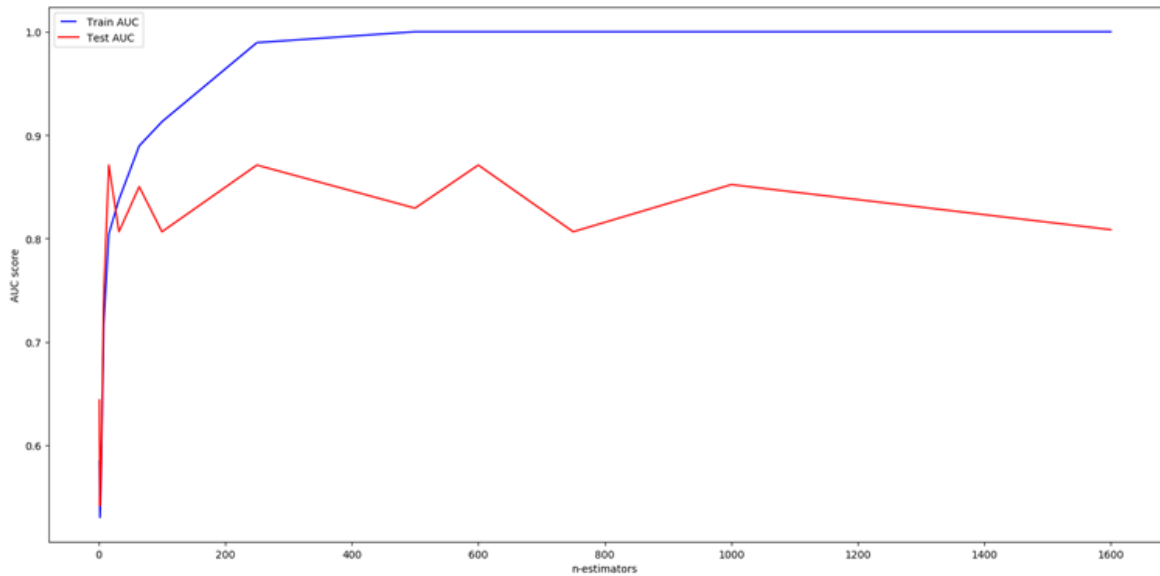


Figure 4.5: Calculating AUC ROC for different values of n-estimators

It is clearly seen that after a value of around 300, the values seem to converge for the training set, while for the test set the value stays similar even decreasing slightly. For this reason, we can confidently pick the right value for n-estimators without having to be concerned about over-fitting our model. This technique is then repeated for all the different parameters that the GBM uses, and once complete our statistical model is ready to be used for future gameweeks.

4.3 Statistical Analysis

The statistical analysis component is responsible for all possible statistical factors that could be used to predict a players future performance including;

1. Historical gameweek data;
2. Upcoming fixture statistics;
3. Bookkeeper odds.

4.3.1 Historical Gameweek Data

This component refers to both player and gameweek specific data for every previous round within the current season. For every fixture played, important statistical measures such as the players form, threat, goals, assists, clean sheets etc. are combined with information about that fixture such as whether the game was played at home, the fixture difficulty rating and the number of fantasy football users that both transferred in and transferred out the player in that gameweek. By training a model on all this historic data we can create models that use information about upcoming statistics together with the latest player form data to make more accurate predictions.

4.3.2 Upcoming Fixture Statistics

The Fantasy Premier League website directly provides free unlimited access to their API that provides player specific information for the upcoming fixture ¹. They calculate the players current form, expected points, influence, threat and creativity as well as providing a detailed list of statistics of their season so far.

The main problem with using these measures individually is that factors such as expected points and form are heavily influenced by the previous performance. This means that if an unusual event such as a substitute defender scoring a hat-trick occurs, these variables will be heavily biased and predict that the expected points for the upcoming game will be much higher than they really should be. This indicates that while these statistics are helpful, they must not be too heavily relied on.

4.3.3 Bookkeeper Odds

By also factoring in bookkeeper odds, we can obtain freely available information about the upcoming fixtures that statistics alone cannot provide. Bookkeepers odds are extremely accurate, taking into consideration all the possible factors that could affect the game; both statistical in terms of previous match performance as well as external factors such as injuries, importance of the match for both teams and even the rest days between matches from different competitions.

¹<https://fantasy.premierleague.com/api/bootstrap-static/>

By converting the odds into percentage probabilities, we can use the Result and both teams to score market to infer the likelihood of teams winning, scoring goals and keeping a clean sheet. These team probabilities are then assigned to all players within the team. The possible outcomes for the market are as follows;

1. Team A win and Yes (Team A wins and both teams score)
2. Team A win and No (Team A wins and both teams do not score - meaning only Team A scores)
3. Draw and Yes (Draw and both teams score)
4. Draw and No (Draw and both teams do not score - implies a 0-0 draw)
5. Team B win and Yes (Team B wins and both teams score)
6. Team B win and No (Team B wins and both teams do not score - meaning only Team B scores)

Through this information we can infer the probabilities of Team A scoring by combining the probabilities of;

1. Team A win and Yes
2. Team A win and No
3. Draw and Yes

Similarly, the probabilities of Team A keeping a clean sheet are calculated through the combination of the following result probabilities;

1. Team A win and No
2. Draw and No

Through this technique, we can add additional metrics to our predictions, by assigning both teams a scoring and clean sheet likelihood into our player dataset. Each player within the team is then assigned these values.

4.3.4 Data Collection Pipeline

Gathering and preparing all the required data in an automated fashion proved to be very challenging, mainly due to the sparsity of information and the lack of standardized information across the different data sources. The Fantasy-Premier-League Scraper application developed by Vaastav provides extensive statistics for each player in every game all the way back to the start of the 2016/17 season [50]. The application provides CSV formatted files containing all the required statistical information from all the previous gameweeks such as for example goals scored, assists, clean sheets, transfer data and even availability in terms of the players injury status. All the data gathered through this application was used as the core of the statistical prediction model.

Once all the required data was gathered, the FPL API was used to insert the additional features for the upcoming gameweek. This was essential to give the machine learning algorithm the required data to work with when attempting to predict the player performances for every round. Factors such as the next opponent, whether the game is home or away, and the number of fantasy players that transferred in and out the player as of the time of running the application were added.

Additional features such as the fixture-difficulty ratings were also used. Based on an algorithm developed by calculating a set of formulae together with the teams form across the past 6 matches, each team is assigned a rank for the perceived difficulty of their opponents. The values can range between 1-5, with 1 implying that the match should be relatively easy while 5 suggests that the opponent is favoured in the match [51]. Once calculated, these values are published for use directly by the FPL team and may update throughout the season should their calculations change due to performance throughout the season.

Betting data was also added through the Result and Both teams to score market. The API used provides data for the entire season and returns a set of odds given a requested fixture ². This allows our model to also store clean sheet and scoring odds for each fixture, and by identifying which player belongs to which team, these probabilities can be assigned accordingly. By doing so our machine learning algorithm now also can automatically make use of betting data; a mathematical, highly accurate prediction parameter.

²<https://www.api-football.com/>

Once all the columns were chosen and populated, the dataset was cleaned by also removing players that are injured and have less than 50% probability of playing in the next gameweek. Additionally, players who had 0 minutes in the previous gameweek were also removed from the dataset because this drastically increased the chances of false predictions, while also helping to balance out the dataset as recommended by Thapaliya [8].

4.4 Linguistic Analysis

In our dissertation, linguistic analysis is used in two separate components, namely for news-articles and blog posts, and for twitter data. The goal of linguistic analysis within these components is to understand who specific text documents are writing about, understanding their sentiments and assigning mentioned players with a sentiment rating per document. Through this method we aim to predict future FPL performances based on the recommendations of domain-experts and fan opinions alike.

4.4.1 News-Articles and Blogs

Because FPL is such a huge topic that generates popular discussions every week, many news articles and blogs exist that post their personal recommendations, tips, and premier-league related news. These are carefully written and often contain insider-information by domain experts including rumoured line-ups, squad rotations and even potential unconfirmed injuries. It is common for serious FPL players to sift through these blogs weekly, trying to optimise their teams based on these recommendations. Through NLP techniques including named entity recognition and extraction, and sentiment analysis, we aim to automatically derive the entity-based sentiment per blog and player. This way we can automatically incorporate blog and news-article opinions and recommendations into our predictive model.

4.4.1.1 Data Collection

Different approaches to this task were considered. Initially, the naive approach of manually selecting the best FPL specific blog sites and scraping their latest articles for new data was proposed. This approach is the generally recommended method

when scraping data from news articles because it reduces the amount of noise [52] [53]. By specifying the websites from which articles are scraped, we can ensure a certain standard and can even judge how accurately that source predicts future events based on their previous predictions. By understanding their individual accuracies, we could assign different weightings to each author.

The main issue with this approach, however, is that each blog tends to only give a recommendation for a select few players. Figure 4.6 demonstrates the player sentiment results for all players throughout all the gameweeks in the FPL 2018/19 season when over 300 blogs are analysed per gameweek. It is observed that most players have 0 points, which means that they are never even mentioned in blogs. This makes sense because just like we saw in the points distribution graph, most players don't make any appearances in games, making it extremely unlikely that they would be mentioned in FPL recommendation blogs. The next majority of players have just a few points, both on the negative and positive side. Similarly, this indicates they've only been mentioned on a small number of blogs in the over 300 considered. This means that if we used the standard approach of hand-picking the top blogs and only relying on them the result would be a small subset of players who are recommended just a few times each, while the vast majority are unmentioned. As discussed in the method below, this would not be enough for our model because most players never get mentioned meaning that this would not be effective in predicting future results. This would also mean that any incorrect predictions by any single blog would have a very big impact on our overall model when only considering a select few blogs to consider.

To solve this issue, we propose a method through repeated scraping that involves selecting the top 100 web search results for several different FPL specific search queries. In addition, we also use a single, constantly updated injury specific blog ³ to remove all the currently confirmed injured players from our dataset. While most players are still never mentioned, a select few who are expected to perform well are repeatedly recommended by several different blogs. This will result in a very high score that should provide a very powerful indicator to the recommender systems predictions. As more time in the season passes, the model is also expected to improve because it will learn to understand that large blog-score points are generally a very important indicator.

³<https://www.fantasyfootballscout.co.uk/fantasy-football-injuries/>

During the testing and evaluation phase, it was important to ensure that all retrieved results were written and uploaded prior to the gameweeks start. Otherwise, by including post-gameweek articles that discuss the actual performances, our work would be demonstrating NLP techniques rather than providing an FPL recommender system. To ensure this, a date range is included with each search query, filtering all documents that are not written between 1-5 days before the gameweeks start.

Table 4.6: Queries used to gather news articles and blogs

QueryID	Query
1	Fantasy Premier League gameweek "GWxx"
2	FPL gameweek "GWxx"
3	FPL budget "GWxx"
4	FPL injury "GWxx"
5	FPL tips "GWxx"

Table 4.6 shows the current queries that are used to gather the web-documents to be analysed. By using different keywords across the searches, we can maximise the range of blogs found while also covering a range of different topics, ranging from the standard favourites, to injured players and budget non-conventional picks. This technique also allows us to expand our search range while still only considering the documents considered to be the top-rated results for the given query, effectively giving us more valuable results than simply searching for more documents within a single query. Any duplicate results and links to videos are removed.

4.4.1.2 Generating Predictions

Once all the blogs have been collected, the next step is to parse and understand the sentiment behind their texts. A local CSV file ⁴ containing all the player names, surnames, common names and nicknames is created, that will be matched against each recognised entity in every document. Through named-entity-recognition and extraction methods discussed above, entities extracted from the web documents are compared to entities in the database. If matches are found, the entity-based sentiment score is

⁴<https://github.com/bonellon/MoneyBall/blob/master/Blogs/players.csv>

calculated, and the respective player is given a rating. This process is repeated for every crawled website resulting in a large web of player scores. As shown in figure 4.6, most players are never mentioned in any of the crawled blogs, while players expected to perform well are repeatedly mentioned. This finding correlates with the average number of points players achieve, with the vast majority scoring less than or equal to 2 points per gameweek. It is logical that for most of these players most of whom dont even make an appearance in the game, there will be no discussion on news articles and blogs. We also see that injuries and weak performances in games are given negative sentiments and when discussed.

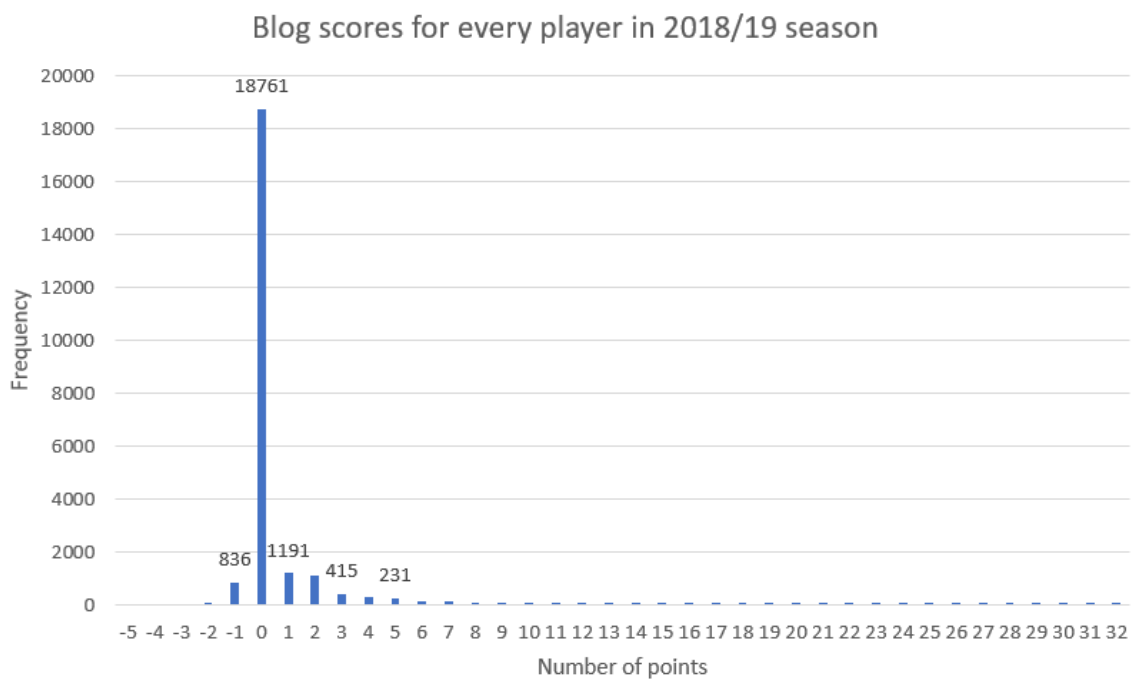


Figure 4.6: Blog and news-article sentiment score per player per gameweek in FPL season 2018/19

To complete this task, several different tools were required. Scraping the web for the different queries was done using the python googlesearch library ⁵. By providing the different queries as parameters, the library would search and return a list of URLs returned by the query.

⁵<https://github.com/MarioVilas/googlesearch>

All the text in each document is then extracted and passed on to the text-analysis API. Initially, IBM Watsons Natural Language Understanding toolkit ⁶ was used; however it was found that the Aylien Text-Analysis API ⁷ provided much better functionality. Ayliens text-analysis API provides inbuilt entity-based sentiment analysis, meaning that while text is being processed, all the found entities are extracted, and their individual sentiment scores can be calculated. On the other hand, IBM Watson provides entity extraction, but does not give separate sentiment analysis scores per entity. Instead, giving a single sentence long sentiment score - resulting in sentences containing two or more different subjects being given the same score. For example, given the following sentence:

With Danny Ings suffering a hamstring injury that will potentially see him sidelined for 2 weeks, the Saints forward has made way for the in-form Marcus Rashford, whos bagged 3 goals and 2 assists in the 4 games under Ole Gunnar Solskjaer.

The results from the two different text-analysis APIs are shown in table 4.7. Because of this primary benefit of the Aylien API it was selected for the rest of the project implementation. A human reader of the above sentence would easily be able to recognise that Danny Ings (player) is now injured and should not be in an FPL team for the next week. The blog then recommends Marcus Rashford as a replacement. Its important that the API used can replicate this human behaviour, extracting names through named-entity recognition and extraction, and afterwards determining the sentiment around the entity. The APIs also extract Ole Gunnar Solskjaer, the Manchester United manager. When comparing this name to our aforementioned player database, no match is found meaning that this entity will be ignored. Similarly, Saints - which is the nickname for Premier league team Southampton is also ignored for similar reasons.





Once the scores are retrieved from the Aylien API, theyre converted to scores that we can use to aggregate the total score across all the documents we consider. Initially, a player total starting from 0 points is kept for each player. When we detect a positive mention, we add the sentiment score to the current player score, and similarly subtract when a negative mention is detected.

⁶<https://www.ibm.com/watson/services/natural-language-understanding/>

⁷<https://aylien.com/text-api/>

Table 4.7: Aylien API vs IBM Watson performance when predicting entity-based sentiment.

Aylien API	Entity	Overall Sentiment	Type	Mentions
	Saints	Positive 0.71	Organization	1
	Danny Ings	Negative 0.5	Person	1
	Marcus Rashford	Positive 0.56	Person	1
	Ole Gunnar Solskjaer	Positive 0.54	Person	1

IBM Watson	Name	Type	Score
	Danny Ings	Person	 0.33
	Saints	Organization	 0.33
	Marcus Rashford	Person	 0.33
	Ole Gunnar Solskjaer	Person	 0.33

4.4.2 Twitter data

Similar to the arguments made for news-articles and blogs, twitter data is another essential source of the latest FPL information. Rather than expertly written articles, the focus in this domain is to aggregate the wisdom of crowds principle by compiling the list of tweets using FPL specific hashtags prior to every single gameweek. Twitter and social media in general are important sources because of their sheer volume and also because all the insider information, possible rumours and varied discussions and opinions generally tend to be first shared here. This means that social media discussions

on twitter and other FPL specific content sites generate lots of valuable discussions that human readers and text analysers could use for improved predictions.

4.4.2.1 Data Collection

In a similar technique to the news-article and blog collection, Python libraries are used to take full advantage of Twitter’s advanced search functionality ⁸. Twitter also provides their own set of developer APIs ⁹, offering a range of different functionalities including historical tweet searches. They also provide several different libraries such as Tweepy ¹⁰ which has all the API calls ready allowing the user to simply modify the parameters and begin using the API. However, the free Twitter API versions are very restricted both in terms of API-call allowances as well as the number of tweets returned per API call. This allowance would be suitable during the upcoming seasons, where 100 calls per month each returning a maximum of 100 tweets, would be sufficient. This is because these calls can be evenly split across the 4-5 gameweeks per month. During implementation and testing however, all the historic seasonal tweets need to be gathered for every gameweek in the previous season, together with different experimental search queries to examine which query provides the best results. When gathering data for the previous season for testing purposes, it was important to ensure that no post-game tweets get into the dataset possibly skewing our results. For this reason, we would only search for tweets for any given gameweek up to 1 day before the first match.

Table 4.8: Example of searching periods per gameweek

Gameweek	Gameweek start	Gameweek end	Query start	Query end
1	10th August 2018	12th August 2018	3rd August 2018	9th August 2018
2	18th August 2018	20th August 2018	13th August 2018	17th August 2018
3	25th August 2018	27th August 2018	21st August 2018	24th August 2018
4	1st September 2018	2nd September 2018	28th August 2018	31st August 2018

⁸<https://twitter.com/search-advanced>

⁹<https://developer.twitter.com/en/docs/tweets/search/api-reference/premium-search>

¹⁰<https://tweepy.readthedocs.io/en/latest>

4.4.2.2 Generating Predictions

Once a method for gathering tweets was settled, the next step was to analyse and infer player sentiments from each tweet. Again, the Aylie API proved to be the most promising because it could correctly identify named entities and provide their individual sentiments within lengthier blogs and articles. However, this proved to not be as effective as with correctly structured English texts as seen in news articles and other web documents for reasons discussed in detail below.

We investigated different options with varied success. Initially we tried to replicate the method used for news-articles and blogs however this method was found to be extremely ineffective. The brevity of tweets combined with the lack of grammar and proper syntax means that even if the named entities can be extracted, their sentiment is generally inaccurately labelled. Additionally, many tweets contained lots of emoticons that replaced recommended actions, or even described a player. For example arrow emoticons would explain who is replacing who, ticks imply transfers in, and an ambulance emoticon was even seen representing an injured player.

Another alternative method was to use sentiment analysers specifically trained and targeting tweets. However, the same problem of misreading both named entities and their sentiments was again observed. This problem is seen because of the FPL-specific language used in such tweets. Emoticons are used to represent transfers in, transfers out, a players team, anger, injuries and many more. In addition, lack of grammar and proper sentence structure also causes additional problems that further add to the issues in predictions. Our final recommendation for future methods looking to incorporating tweets is to create a custom corpus with historical domain-specific data. In our case, extensive data is available from all the previous years of FPL meaning that this task boils down to manually classifying all positive and negative tweets while removing all player and gameweek specific information. Improving on this recommendation, we could keep track of twitter users and assign them their own individual reliability scores, increasing for correct predictions in previous weeks and vice versa. This would allow us to increasingly accurately identify domain-experts and ITK (in the knows) people claiming to have insider information from specific teams that often release vital information such as rumoured line-ups.

4.4.2.3 Issues with Tweets

Unfortunately, gathering and analysing tweets was found to be a very problematic area causing it to be neglected in the final models predictions. Tweets are a form of microblog with a maximum limit of 280 characters per tweet. They are notorious for their common issues including bad spelling and grammar, lack of punctuation and even emoticon usages to represent ideas and sentiment. As shown in figures 4.7 - 4.10 below, it becomes clear why standard non-FPL specific classifiers cannot perform well on tweets and social media in general.

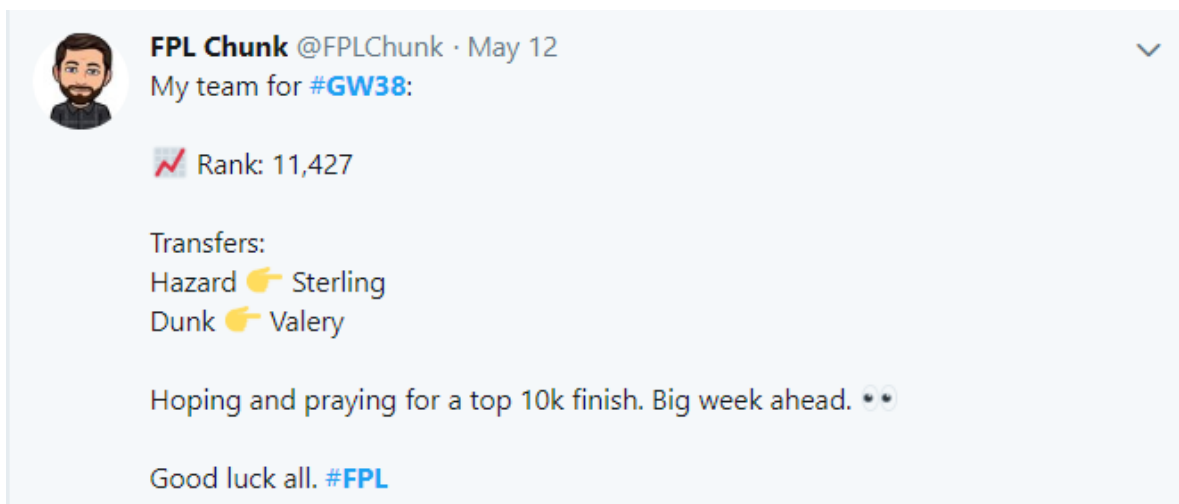


Figure 4.7: Tweet example showing emoticons used to imply transfers

The tweet shown in figure 4.7 shows a user who plans to transfer out Hazard and Dunk, replacing them with Sterling and Valery respectively. In this case the pointing finger emoji tells us who will be replacing which player. Players being pointed to would need to have positive sentiment, while the rest should have a negative sentiment.



Figure 4.8: Tweet example showing independent player emoticons

Similarly, in figure 4.8, players Son and Jimenez are being removed and replaced with Fraser and Vardy. Additionally, He plans to captain Aguero in the upcoming gameweek. In this example, these actions are represented through emoticons. If a model was not trained using historical FPL data which had such instances in the training data, there is no way of accurately understanding the sentiment implied by these emoticons.



Figure 4.9: Tweet showing injuries through emoticons

In figure 4.9, we find out that De Bruyne is injured as seen by the ambulance emoticon. This player recommends replacing him with Redmond from Southampton nicknamed Saints. Both players Firmino and Mane are part of Liverpool, commonly nicknamed the reds. While certain emoticons such as an ambulance and a middle-finger could easily be used to extract the implied sentiment, team-specific emoticons such as the saint or the red circle are highly specific to our domain.



Figure 4.10: Tweet written without any correct grammatical structure

In figure 4.10, the user recommends replacing Hazard with Salah. While this is easily understood for human readers, sentiment analysers might get confused by the lack of grammatical structure, unnecessary punctuation and incorrect capitalization. Further down in the tweet, the user mentions that he also intends to "TC" - meaning triple captain; which is a token that allows FPL players to gain a triple multiplier from their captain for one gameweek in the season.

4.5 Generating Predictions

Before we can make any predictions for the upcoming gameweek, all the data for the previous gameweeks must be loaded and cleaned. Through the methods described in the above sections, all the final data for each previous gameweek is gathered per data-source. Similarly, all the currently available data for the upcoming gameweek is retrieved. Because of the nature of tweets, betting-odds, news articles and blogs, different predictions will be made depending on how long before a gameweek the recommender service is run. Naturally, both twitter data and news-articles will be much more sparse the further away a gameweek is. To retrieve the data that most accurately reflects on the likely outcomes for the gameweek, running the predictor as close as possible to the gameweek start is recommended.

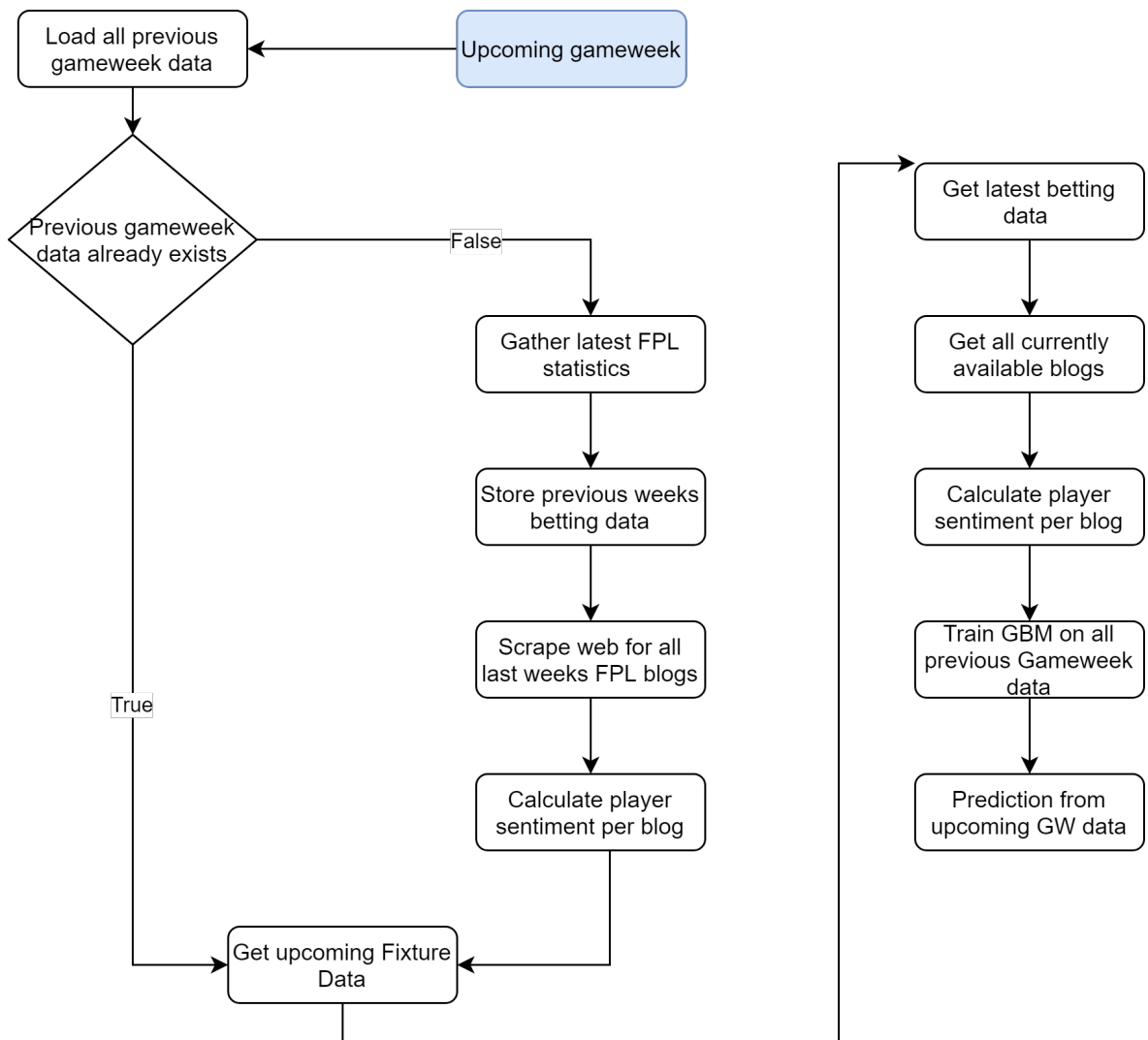


Figure 4.11: High level flow diagram for loading the latest data into our dataset and generating predictions for the upcoming gameweek.

After a gameweek passes, the application does not automatically gather all match-day statistics and the final set of news-articles, blogs, Twitter data and betting odds. This means that once the predictions for the next gameweek are requested, the application must first check if all the final data for the previous gameweek already exists in the dataset. Figure 4.11 highlights the flow diagram in gathering all the data before any predictions can be made. If the latest data for the previous gameweek was never updated (meaning that the recommender service was never run after the gameweek passed), each data-source specific component is run to retrieve the latest data for the

passed gameweek, ensuring that any retrieved non-statistical data stems from before the gameweek started.

Once all the data is prepared, all previous gameweeks are dedicated to training data to maximise the amount of available data for our models. By understanding the importance of all the variables within our dataset (figure 4.3), the irrelevant metrics are removed. Once training is complete, the trained models are then given all the currently available data for the upcoming gameweek in order to generate predictions. Through this method, we can reliably make predictions for consecutive gameweeks using the latest available data without any human interventions.

4.6 Challenges

4.6.1 Statistical Analysis

Dealing with all the publicly available data and free APIs proved to be unexpectedly challenging. The official FPL API totally overhauled their endpoints in between seasons, while the betting API would often not have all the odds available for each fixture both upcoming and historic. Additionally, because we are gathering data from various different sources, it was important to ensure that they are all gathered, cleaned and prepared individually. Any errors that could cause data to not be retrieved could result in a weak dataset that would need to be populated using default values.

4.6.2 Linguistic Analysis

Due to the number of challenges encountered when attempting to identify sentiment correctly through tweets, we deemed it not suitable to be incorporated into the project. A custom corpus would have been needed that was FPL specific based on all the previous seasons of tweets. While this has some potential, it would only be successful for tweets predicting single players in sentences. In our case, as seen in figure 4.7 it wouldn't be possible to split that specific example tweet into positive and negative sentiment training examples because there is simply no way of splitting it up into a positive and negative training example for our corpus.

4.7 Summary

In this chapter we discuss the methods used to search, collect, prepare and make predictions for all the different data-sources considered. We compare different techniques along with analysis of what methods are most effective and why. Finally, we discuss the challenges with each data-source.

Chapter 5

Results & Evaluation

The following chapter will present the findings of this dissertation and how they help answer the research questions. Firstly, I outline the application built and the differences to real-world FPL users in terms of rules followed. I then discuss the different approaches and how they perform against each other and the official FPL leader board.

5.1 Hypothesis

Our hypothesis is that combining different data-sources to traditional FPL prediction algorithms will increase their predictive power and performance. Using a baseline model focused on a purely historical statistics dataset, we expect to see a rise in average points achieved per gameweek when incorporating data from additional non-statistical sources into our dataset. While statistical measures provide a good indicator of what is expected to happen in future FPL events, external events such as squad rotations, injuries, and mid-week international matches can never be automatically considered.

5.2 The Predictive Model and FPL Rules

Our final model is a recommender system for FPL, helping users by recommending a set of 11 players that are expected to perform well in the upcoming gameweek. Because of this, our model does not follow the official rules, and this must be taken into consideration when comparing our model with the FPL 2018/19 seasonal rankings

below. The primary advantage that our model has over regular players, is that we don't consider transfer penalty fees between gameweeks. This means that we can effectively totally overhaul our team every single week without any penalties. In the official game, each player is given a single free transfer per gameweek, with additional transfers costing a deduction of 4 points.

On the other hand, our model only picks 11 players per week which prevents us from having any substitute players in our squad. FPL players have the benefit of owning 15 players, so that if a player in their starting squad does not make an appearance in their real matches, they will be automatically replaced in the FPL lineup. Additionally, FPL allows for each team to have a captain a player whose given a points multiplier of 2. While we could have easily allowed our model to pick 15 players and field the 11 that are most confidently expected to perform well, we decided that in order to try and balance out the advantages and disadvantages that our model had over real players, the model would only be allowed to pick 11 players without any captain.

5.3 Web Application

In order to demonstrate the functionality of the application created as part of this dissertation, a demo website was created. The website would allow users to view both upcoming and historical predicted teams per machine learning algorithm. Because of the results discussed below, no options to switch datasets were provided only providing predictions for the superior multi-stream dataset we created. For all the previous gameweeks, each player is also shown along with their achieved points and the total points that team would have achieved in the gameweek. The historical screens shown for the FPL 2018/19 season gameweek 2 are seen in figure 5.1 and 5.2 below.

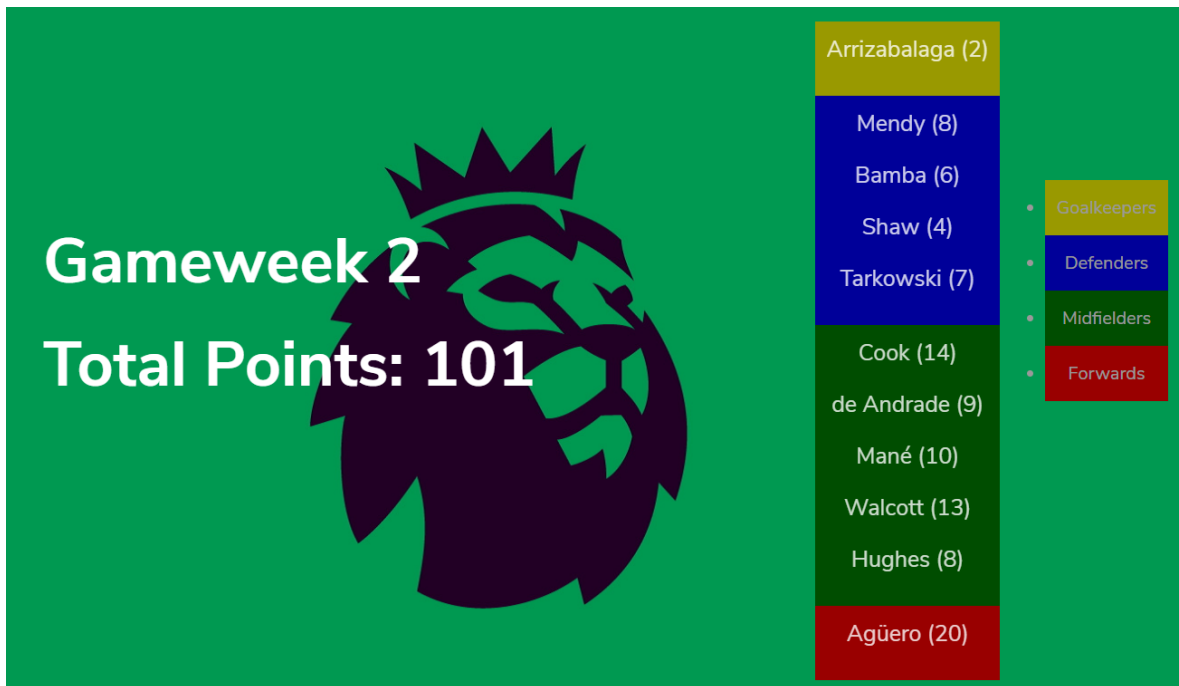


Figure 5.1: GBM prediction for season 2018/19 gameweek 2

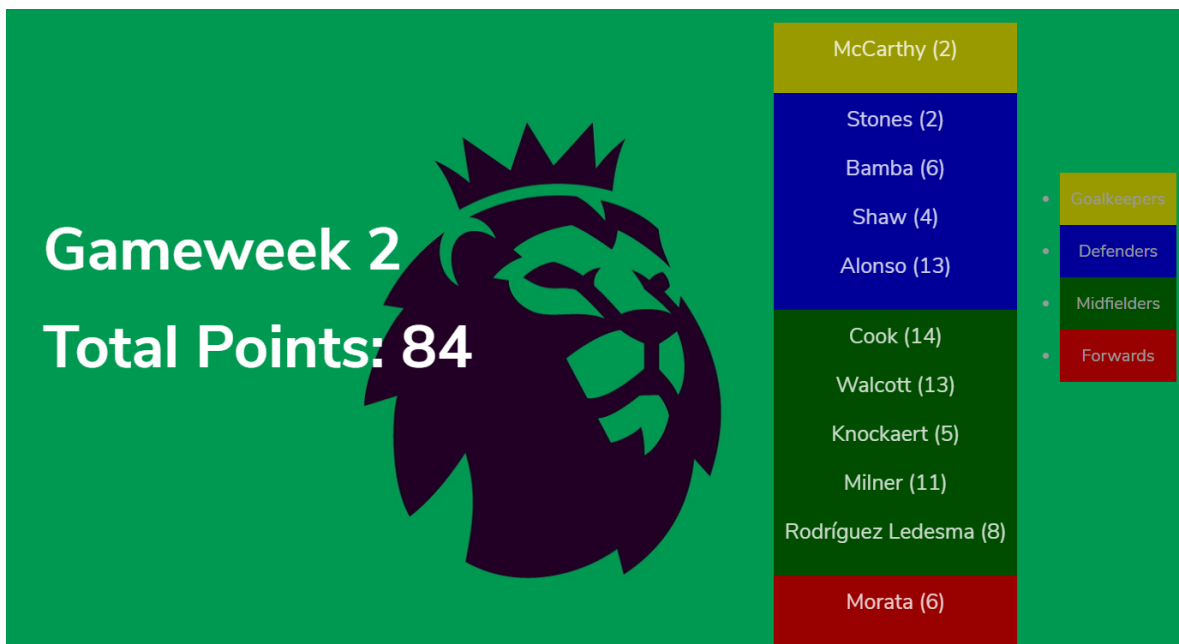


Figure 5.2: SVM prediction for season 2018/19 gameweek 2

Since this website was created as a demo to demonstrate the current capabilities of the application it was built to show very simple teams and their point scores

highlighting the differences in different algorithms. There are a number of proposed improvements both in terms of functionality and usability that will be discussed in the future work section below.

5.4 Evaluation Approach

In order to evaluate the different algorithms and datasets, we tested them on the entire FPL 2018/19 season. All algorithms were used to predict gameweeks 2-38 three times each. Once a team is predicted, the corresponding real score achieved by that team is calculated. For each algorithm, the average score per gameweek is repeatedly calculated 3 times to ensure that the final result is an accurate representation of the predictions. This process is then repeated for the remaining gameweeks in the 2018/19 season and the final scores are compared to each other and to the official FPL leader boards. To correctly evaluate the performances of different machine learning models, a confusion matrix is used. A confusion matrix is a table used to understand the performance of a classification model. After classification is complete, the confusion matrix is made up of the following states;

1. **True Positives (TP)** The true positive rate represents the number of positive cases in the data that are correctly predicted to be true
2. **False Positives (FP)** The false positive rate represents the number of positive cases in the data that are incorrectly predicted to be false.
3. **True Negatives (TN)** The true negative score represents the number of negative cases that have been correctly classified to be false.
4. **False Negatives (FN)** The false negative score represents the number of negative cases that have been incorrectly classified to be true.

Table 5.1: Confusion Matrix layout

	Predicted Positive	Predicted Negative
Actual Positive	TP	FP
Actual Negative	FN	TN

5.4.1 Precision

The precision score is the percentage of correctly predicted true results when considering all positively predicted results. High precision scores imply that there is a high probability that retrieved results will be accurate to the task. In our case, having a high precision would mean that most players picked by an algorithm will have 6 or more points. On the other hand, having a low precision score results in poor retrieval accuracy and a low FPL score. Precision is calculated as follows;

$$Precision = \frac{TruePositiveRate}{TruePositiveRate + FalsePositiveRate} \quad (5.1)$$

5.4.2 Recall

On the other hand, recall measures the percentage of correctly predicted positive results when considering all the positive classifications within the whole dataset. In every FPL gameweek, there is an unknown and constantly varying number of players that achieve more than 6 points. While recall is an important metric to understand how many potential options were missed, having a full team of correct predictions is far more important meaning that the precision of our selections is far more important.

$$Recall = \frac{TruePositiveRate}{TruePositiveRate + FalseNegativeRate} \quad (5.2)$$

5.4.3 Accuracy

Accuracy measures the percentage of correct predictions when considering all the predictions. Accuracy is an important metric for us to consider, however because of the small number of players who achieve 6 or more points in any given gameweek, accuracy is not the most important metric since it can be heavily skewed by the accuracy of non-captain player predictions. For example, in table 5.3 we see that in FPL season 2018/19 gameweek 38, only 19 midfielders scored 6 or more points out of 121 midfielders.

$$Accuracy = \frac{TruePositiveRate + TrueNegativeRate}{TotalPredictions} \quad (5.3)$$

5.4.4 F-Score

By taking the mean values of both the precision and recall scores, the F-Score provides an accuracy measurement. Considering both the precision and recall, the F1 score is the harmonic average achieving an optimal value at 1. By altering the importance of the precision and recall in the equation, the F-Score can be modified to properly suit the needs of the problem at hand.

$$FScore = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (5.4)$$

5.5 Evaluation

5.5.1 Comparing different Machine Learning Algorithms

We evaluated the viability of 3 different machine learning models; namely **SVMs**, **RFs** and **GBMs**. As discussed, all the different algorithms were compared by comparing their average scores across three different prediction iterations per gameweek in the 2018/19 FPL season. Three iterations per gameweek were used to ensure that one-off predictions are averaged out as much as possible. Through figure 5.3 we can see that while all the different models performed exceptionally well, the GBM slightly outperforms the competition consistently throughout the season. These results are obtained by training each model on all the statistical, betting, and linguistic data available for all the previous gameweeks in the current FPL season. This score would have effectively placed the GBM prediction model within the top 0.5% of FPL players. A more detailed breakdown of the results and rankings is shown in table 5.2 below. While the advantage of being allowed to create a new team every gameweek definitely is a large advantage, the ability to pick correct predictions consistently is what would be expected from a recommender service that provides precise weekly recommendations.

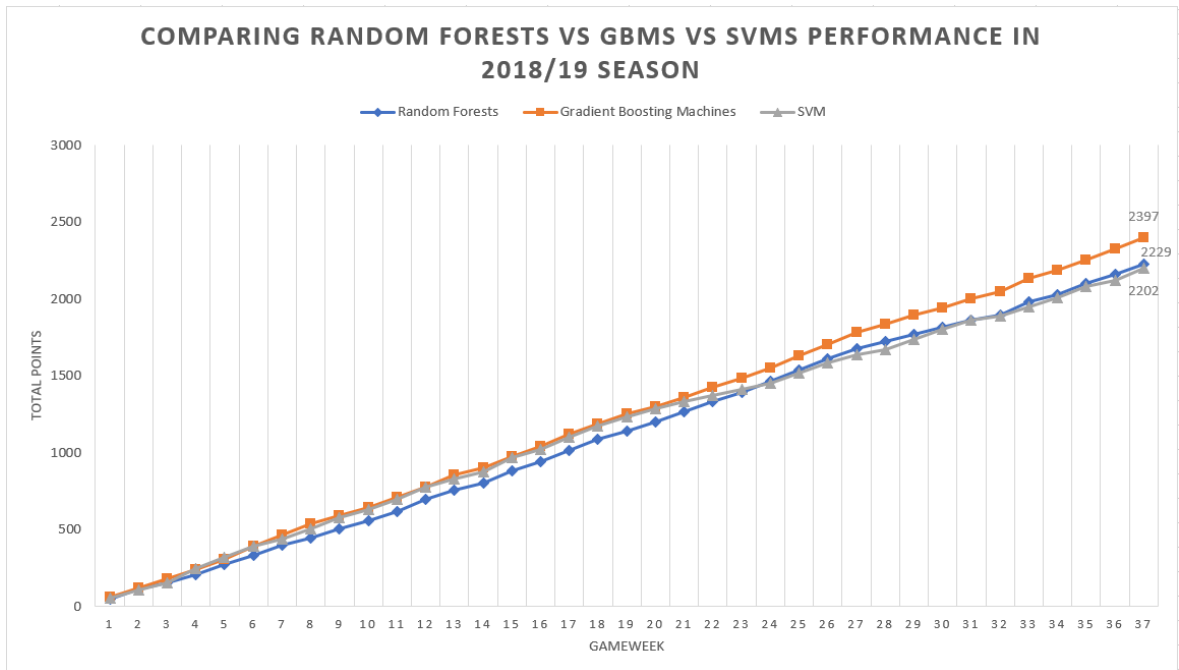


Figure 5.3: Comparing SVM vs RF vs GBM performance using historical statistics, betting odds, news articles & blog data

Aside from the points differential, it's also important to note that SVMs aren't as suitable for larger datasets due to their training time. When predicting the optimal team for the final gameweek of the season, the amount of time each algorithm took is seen in table 5.2. Because we consider 4 different datasets (1 per position), the time taken to generate a team prediction involves training 4 different algorithms. The performance metrics shown for the different models represent the average values across the 4 datasets. Additionally, within each dataset, the performance metrics are calculated by taking an average value across the two classification classes; players predicted to achieve 6 or more points, and players predicted to achieve less. The full results achieved by a GBM in FPL season 2018/19 gameweek 38 is shown in table 5.3. Even though the predictive power of the GBM is extremely weak for correctly predicting valuable goalkeepers (precision of 0.25), because of the ability to correctly eliminate goalkeepers who will not perform well, the average precision score remains 0.76. When calculated across the entire player dataset, the precision rises to 0.897 due to the powerful predictions in more forward positions.

While optimisations such as simultaneously training the 4 models were imple-

mented, SVMs take considerably longer than GBMs and RFs, making them a weaker candidate for a possible future user-friendly web-application. While it may still be suitable when making generic predictions, SVMs would be unusable when making user-specific team recommendations that would involve both training and prediction in real time. Taking around 40 minutes to produce the final set of predictions is simply not feasible, especially when compared to the other algorithms that both take less than 30 seconds each.

Table 5.2: Comparing different machine learning algorithms performance predicting gameweek 38

Model	Dataset	Precision	Recall	F-Score	Average Points/Gameweek	Season total points	Season ranking	Training + Prediction time/s ¹
SVM	Historical statistics + Betting odds + Blogs	0.824	0.830	0.850	59.41	2202	350,000	2512.22
RF	Historical statistics + Betting odds + Blogs	0.869	0.891	0.875	60.24	2229	300,000	7.14
GBM	Historical statistics + Betting odds + Blogs	0.897	0.903	0.896	61.89	2397	20,000	24.96

Table 5.3: Detailed results for GBM predictions in FPL season 2018/19 gameweek 38

		Precision	Recall	F-Score	Number of Players
Goalkeepers	0	0.86	0.86	0.86	21
	1	0.25	0.25	0.25	4
	Average	0.76	0.76	0.76	25
Defenders	0	0.82	0.91	0.86	74
	1	0.22	0.12	0.15	17
	Average	0.71	0.76	0.73	91
Midfielders	0	0.98	0.99	0.99	102
	1	0.94	0.89	0.92	19
	Average	0.97	0.98	0.97	121
Forwards	0	0.99	0.99	0.99	28
	1	0.98	0.96	0.94	9
	Average	0.99	0.98	0.98	37

5.5.2 Comparing Datasets

Similar to our methods to investigating which machine learning algorithms performed best, we repeated the process for the different data-sources; namely purely statistics, betting-markets and news-articles & blogs. The goal of this test was to understand whether using the same machine learning algorithms with different datasets shows significantly different results. As seen in figure 5.4, the model that used all the available data (statistics, betting odds, news articles and blogs) showed a significant improvement over the purely statistical model. An in-depth comparison of the results can be seen in 5.4.

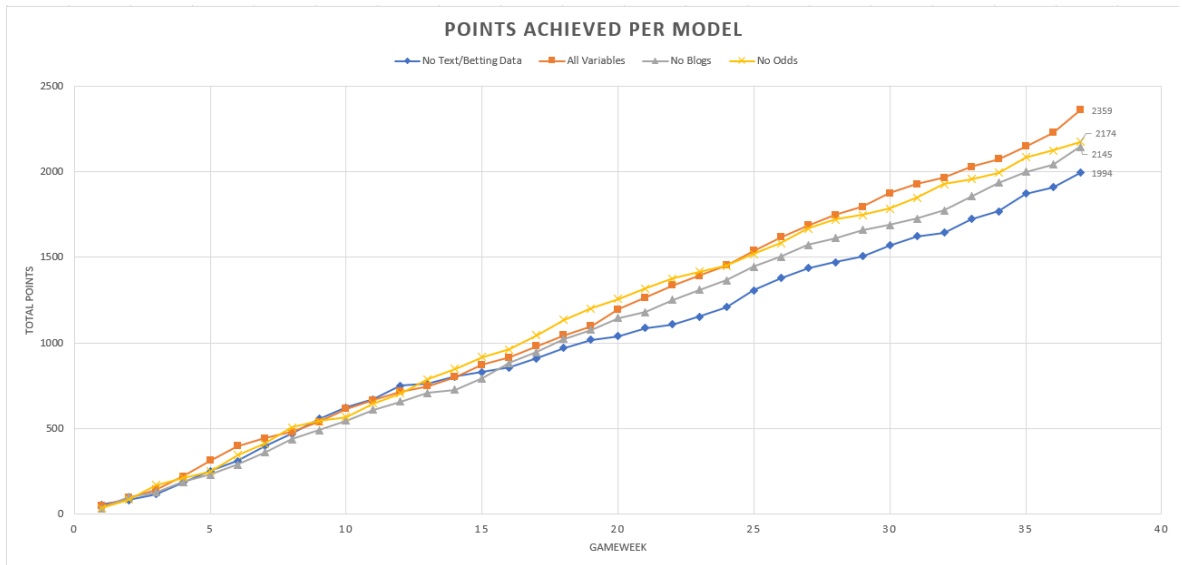


Figure 5.4: Comparing different dataset performances for GBMs

In-line with our hypothesis, the dataset containing all the information from the various different data sources significantly outperforms the other variants; improving the overall ranking on the FPL leaderboards from the top 13% to top 0.5%. In fact, adding just one single source of additional data to the baseline model also sees a slight improvement in performance. Removing the news-articles and blogs data showed the biggest drop in performance, indicating that this is one of the more important data-sources to consider.

Table 5.4: Comparing different datasets and algorithms

Model	Dataset	Average points	Total points	Ranking	Accuracy
SVM	Historical statistics + Betting odds + Blogs	59.41	2202	350,000	0.85
RF	Historical statistics + Betting odds + Blogs	60.24	2229	300,000	0.88
GBM	Historical statistics	53.89	1994	800,000	0.91
GBM	Historical statistics + Betting odds	57.97	2145	600,000	0.89
GBM	Historical statistics + Blogs	58.76	2174	600,000	0.90
GBM	Historical statistics + Betting odds + Blogs	63.76	2359	30,000	0.89

5.5.3 Results

Our best model – a GBM trained using all the available data-sources significantly outperformed the baseline purely historical statistics model. As shown in figure 5.4, the models started the season at a very similar total points haul, but the multi-stream dataset model consistently achieved more points every single week. With an average score of 63.76 points per gameweek, the multi-stream dataset model achieves an average of 10 additional points per gameweek. While this may not seem significant at first, observing the difference in ranking because of these points means dropping from the top 0.5% of players to the top 14%.

Across all multi-stream datasets, a slight decrease in prediction accuracy is seen (from 0.91 to 0.89). We find that while multi-stream methods are much stronger when predicting midfield and forward positions, they were less precise when predicting goalkeepers. This is likely because goalkeepers are very dependent on their form and previous statistics making them very consistent. Because of this, one of the more common FPL strategies is to acquire a set-and-forget goalkeeper; meaning buying

a goalkeeper that is expected to perform well throughout the whole season. Barring injuries or other unlikely events, this goalkeeper is expected to play for the entire season without having to spend any free transfers to replace him. Goalkeepers generally tend to have a relatively low points ceiling; 2 points for playing 90 minutes, and an extra 4 points for a clean sheet. Additional points are sometimes awarded for making a number of saves, or saving a penalty but this is not common. On the other hand, both midfielders and forwards have a much higher point ceiling; getting additional points for goal contributions and additional bonus points depending on their performances. For this reason, a slight drop in accuracy when predicting goalkeepers is considered to be worth the trade-off for better predictions in other roles. In fact, this trade-off results in an average increase of 10 points per gameweek.

5.6 Summary

In conclusion, we find that incorporating different data sources proves to be a highly effective method for improving the standard, purely statistical FPL prediction techniques. These findings are in line with expectations, because for any FPL fanatic it is clear that lots of research is required before every single gameweek to truly optimise a team. This research generally comes in two different forms, firstly by looking at the statistics and secondly by reading blogs, news articles and lots of social media FPL content to gather many different opinions. In this work we proved that this research can be done automatically by recognising the entities and their sentiments within online text documents to significantly improve predictive power.

Seeing this large improvement through the incorporation of news-articles and blogs further brings about the question of how much of an improvement we could see through twitter if a corpus was built. Amidst all the noise lie rumours, insider information and lots of valuable predictions that are typically first spread through twitter and other social media. Extracting all this valuable information would surely be beneficial to any multi-stream prediction algorithm.

Chapter 6

Conclusion

This chapter summarises the work undertaken as part of this dissertation. We also discuss the primary weaknesses and recommended improvements together with recommended future works and the final outtakes from this work.

6.1 Objective Assessment

The primary research aim of this dissertation was to investigate whether combining data from different data sources would help in increasing the accuracy and overall performance of predictions. Ultimately, we found that while the accuracy of predictions suffers slightly (from 91% to 89%), we found that the average points achieved per game-week increases significantly from 54 to 64 points per gameweek. This is a significant points-increase, resulting in a rise in FPL leader board rankings from rank 800,000 (top 13%) to rank 30,000 (top 0.5%) out of the 6.5million total players. By properly understanding the strengths and weaknesses of each data-source, it becomes easy to solve such issues by simply applying more weight importance to statistical parameters when considering goalkeepers.

In this dissertation we aimed to combine statistical metrics with betting odds, news-articles & blogs, and Twitter data to improve FPL recommender services. We found that the inclusion of both betting data and news-articles & blogs individually, significantly improves on the power of predictions made by statistical predictors. However, by including both sets of data simultaneously, the predictions become considerably

more powerful than their individual counterparts. This indicates that the incorporation of any relevant additional data would have positive effects on the predictive power of such recommender systems. While we also investigated adding social media data through Twitter, we faced many challenges leading to inaccurate predictions affecting the overall predictive power for the worse. By creating a custom corpus based on the previous years of FPL data, FPL-specific social media could have a significantly beneficial impact on our model. This is because of the vast amount of discussions and independent opinions that are spread through such sites. Another alternative could have also been the FPL-specific Reddit discussion board ¹.

6.2 Criticisms & Limitations

While the recommender system performed extremely well, there are several limitations across all areas that hindered the overall predictive power. Firstly, we believe that by increasing the number of data-sources, the prediction accuracy and average seasonal score can increase even further. This is based on the fact that when individual data-sources were added to the statistical dataset, the prediction performance always increased. However, when multiple data-sources were combined; in our case, betting markets and news-articles & blogs, the average points achieved per season rose much more significantly than the individual parts.

In the subsections below, we discuss the limitations faced for each of the individual data-sources.

6.2.1 Statistics

Our work on gathering both historic and upcoming statistical data was severely hindered by the amount of available data. While there are many FPL-specific metrics that summarize the performance of players in matches, having more detailed statistics such as the number of shots on goal, shots off goal, pass accuracy, number of dribbles etc. would help us paint a clearer picture of who is and should continue to perform exceptionally well. Opta-Stats ² for example, provide the most in-depth analysis for the

¹<https://www.reddit.com/r/FantasyPL/>

²<https://www.optasports.com/sports/football/uk-football/>

English Premier League, which is also where the official FPL statistics come from. Similarly, betting companies have large amounts of data used to create accurately priced markets. By understanding not only the current prices, but also the price changes leading up to the fixture, additional analysis could be possible. The free API used in this dissertation only provided a limited amount of markets, but if we could extend this to also include player-specific betting markets it could help us understand the expected contributions per player. For this dissertation, the only market that was considered was the Result and both teams to score market, which provides the offensive and defensive probabilities within a single market. By looking at different odds such as the team possession percentages, goal lines, or even the number of corners; we could identify the expected dominance of a team and use this information to make more informed predictions.

6.2.2 News-articles and Blogs

When considering news-articles and blogs, only articles written in the English language were considered. By extending this to a wider range of articles we could effectively widen our search-space without affecting the quality of the blogs, since we would still only be considering the top articles per language.

Additionally, only the raw text was considered within the documents. No additional importance was given if text was a title, subtitle, a footnote or even a user comment reply. Many blogs also post images, either of the most noteworthy players being mentioned, or of their recommended FPL team for the week. While pictures of players would be harder for an algorithm to understand, images representing a recommended FPL team are typically in the same format, making it more possible to extract information through such images.

6.2.3 Twitter

The main limitation with the progress of this dissertation was the inability to incorporate microblog data. We noticed a significant increase in accuracy when even a single different data source is incorporated with historical statistics, leading us to believe that this would only increase further through additional sources such as microblogs. By combining our current methods with new datasets composed of FPL focused social

media discussions, the latest information could also be considered. However, social media discussions could be about many different topics; from predictions to possible rumours or even asking for recommendations. Since these topics are widely different to those seen in news articles and blogs, it is important to ensure proper noise removal before performing the NER and sentiment analysis.

6.3 Future Work

Apart from extending the model based on the limitations outlined above, our project could be improved significantly by enforcing all the traditional FPL rules. At the current state, our application can only be used as a recommender system that highlights players expected to perform exceptionally well in the upcoming gameweek. The problem here is that since users do not have unlimited free transfers per week, they can only compare the recommended team to their own and make their own decisions on who they should transfer.

This could be expanded in two separate directions. The first option would be to act as a form of team-specific recommendation service. Users would provide a link to their team and our service would look at their current squad, the available funds and whether they have any free transfers available. Through all this information, our service could examine weaknesses in their squad and provide a number of recommended transfers specifically catered towards the current user.

The second option is to modify the model to follow the official FPL rules and compete against the real players. By taking this approach we can observe predictions in real-time and compare the different models to actual scores achieved across future seasons. To achieve this, the most significant change is the addition of an accurate method for predicting the first gameweek. In our current model, the first gameweek is skipped because of a lack of any recent statistical information that could be used. Once a first team is created, the model then needs to assign a captain and use a single free transfer per week. The captaincy could be given to the player most confidently picked to perform well, while the transfer would be replacing the player within the squad expected to perform worst.

6.4 Final Outtakes

The availability of various different data sources has long been underused in automated FPL predictors. Through this dissertation, we show that making use of multi-stream data has huge potential in FPL predictors as well as unrelated areas that are affected by both statistical and external factors. There is a significant improvement seen when additional data-sources are combined with historical statistics, increasing the final leaderboard ranking from the top 13% to the top 0.5% of players. Naturally, there is a lot of work that can be done to expand on the work shown in this dissertation, either by increasing the functionality or by adding additional data sources to further improve the accuracy of such recommender systems.

Abbreviations

- a. FPL - Fantasy Premier League
- b. EPL - English Premier League
- c. GW - Gameweek
- d. SVM - Support vector machines
- e. RF - Random forests
- f. GBM - Gradient boosting machines
- g. NLP - Natural language processing
- h. NER - Named entity recognition
- i. NEE - Named entity extraction

Bibliography

- [1] A. Sun, E. P. Lim, and Y. Liu, “On strategies for imbalanced text classification using SVM: A comparative study,” *Decision Support Systems*, 2009.
- [2] Niklas Donges, “The Random Forest Algorithm,” 2018.
- [3] T. Matthews, S. D. Ramchurn, and G. Chalkiadakis, “Competing with Humans at Fantasy Football: Team Formation in Large Partially-Observable Domains,” *AAAI*, 2012.
- [4] “Fantasy Premier League, Official Fantasy Football Game of the Premier League,” 2019.
- [5] BBC, “Mythbusting: Do teams play worse after European games? - BBC Sport.”
- [6] Michael Cox, “Criticising Liverpool for rotation is silly as it arguably makes them great.”
- [7] S. J. S. J. Russell, P. Norvig, and E. Davis, *Artificial intelligence : a modern approach*.
- [8] R. Thapaliya, “Using Machine Learning to Predict high- performing Players in Fantasy Premier League,” 2017.
- [9] C. Cortes, C. Cortes, and V. Vapnik, “Support-Vector Networks,” *MACHINE LEARNING*, vol. 20, pp. 273—297, 1995.
- [10] Y. Peng, G. Kou, G. Wang, and Y. Shi, “FAMCDM: A fusion approach of MCDM methods to rank multiclass classification algorithms,” *Omega*, 2011.
- [11] Saptashwa, “Support Vector Machine: Kernel Trick; Mercer’s Theorem,” 2018.

- [12] L. Auria, R. A. M. Berlin, and R. A. Moro, “Support Vector Machines (SVM) as a Technique for Solvency Analysis,” 2008.
- [13] T. Joachims, “Text Categorization with Support Vector Machines: Learning with Many Relevant Features,” tech. rep.
- [14] L. Rokach, “Ensemble-based classifiers,” *Artificial Intelligence Review*, vol. 33, pp. 1–39, feb 2010.
- [15] R. Polikar, “Ensemble based systems in decision making,” *IEEE Circuits and Systems Magazine*, vol. 6, no. 3, pp. 21–45, 2006.
- [16] T. G. Dietterich, “Ensemble Methods in Machine Learning,” pp. 1–15, Springer, Berlin, Heidelberg, 2000.
- [17] S. Jinde, “Ensemble Learning Bagging and Boosting - Becoming Human: Artificial Intelligence Magazine.”
- [18] L. Breiman, “Random Forests,” tech. rep., 2001.
- [19] L. Breiman, “ARCING THE EDGE,” tech. rep., 1997.
- [20] F. Thomas, “Parameter Tuning in Gradient Boosting (GBM) with Python — datacareer.ch,” 2018.
- [21] J. H. Friedman, “Greedy Function Approximation: A Gradient Boosting Machine,” tech. rep., 1999.
- [22] J. Brownlee, “How to Configure the Gradient Boosting Algorithm,” 2016.
- [23] Agarwal Nityesh, “Getting started with Gradient Boosting Machines using XGBoost and LightGBM parameters,” 2019.
- [24] M. Ben Fraj, “In Depth: Parameter tuning for Gradient Boosting,” 2017.
- [25] A. Singh, “A Comprehensive Guide to Ensemble Learning (with Python codes),” 2018.
- [26] E. F. Tjong, K. Sang, and F. De Meulder, “Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition,” tech. rep., 2003.

- [27] M. Asahara and Y. Matsumoto, “Japanese Named Entity Extraction with Redundant Morphological Analysis,” in *HLT-NAAC*, (Edmonton), pp. 8–15, 2003.
- [28] T. Poibeau, “The Multilingual Named Entity Recognition Framework,” tech. rep.
- [29] O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates, “Unsupervised Named-Entity Extraction from the Web: An Experimental Study,” tech. rep., 2005.
- [30] P. D. Turney and P. D., “Thumbs up or thumbs down?,” in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*, (Morristown, NJ, USA), p. 417, Association for Computational Linguistics, 2001.
- [31] B. Pang, . £, L. Lee, . , and . £, “Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales,” tech. rep.
- [32] H. Xu, S. Tan, X. Cheng, and Y. Wang, “Adapting Naive Bayes to Domain Adaptation for Sentiment Analysis,” *LNCS*, vol. 5478, pp. 337–349, 2009.
- [33] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell, “Learning to Classify Text from Labeled and Unlabeled Documents,” tech. rep., 1998.
- [34] T. Joachims, “Transductive Inference for Text Classification using Support Vector Machines,” tech. rep.
- [35] S. Dumais, J. Platt, M. Sahami, and D. Heckerman, “Inductive Learning Algorithms and Representations for Text Categorization,” tech. rep.
- [36] K. Young, “The Biggest Social Media Trends Shaping 2018 - GlobalWebIndex,” 2017.
- [37] L. Derczynski, D. Maynard, G. Rizzo, M. Van Erp, G. Gorrell, R. Troncy, J. Petrak, and K. Bontcheva, “Analysis of named entity recognition and linking for tweets,” *INFORMATION PROCESSING AND MANAGEMENT*, vol. 51, pp. 32–49, 2015.
- [38] A. Ritter, S. Clark, and O. Etzioni, “Named Entity Recognition in Tweets: An Experimental Study,” tech. rep.

- [39] A. Pak and P. Paroubek, “Twitter as a Corpus for Sentiment Analysis and Opinion Mining,” tech. rep.
- [40] X. Liu, M. Zhou, F. Wei, Z. Fu, and X. Zhou, “Joint Inference of Named Entity Recognition and Normalization for Tweets,” tech. rep., 2012.
- [41] B. Han, P. Cook, and T. Baldwin, “Automatically Constructing a Normalisation Dictionary for Microblogs,” tech. rep., 2012.
- [42] A. Go, R. Bhayani, and L. Huang, “Twitter Sentiment Classification using Distant Supervision,” tech. rep.
- [43] L. Barbosa and J. Feng, “Robust Sentiment Detection on Twitter from Biased and Noisy Data,” tech. rep., 2010.
- [44] H. Saif, Y. He, and H. Alani, “Semantic Sentiment Analysis of Twitter,” tech. rep., 2012.
- [45] F. Bonomo, G. Durán, and J. Marengo, “Mathematical programming as a tool for virtual soccer coaches: a case study of a fantasy sport game,” *International Transactions in Operational Research*, vol. 21, pp. 399–414, may 2014.
- [46] A. Owen, “Dynamic bayesian forecasting models of football match outcomes with estimation of the evolution variance parameter,” *IMA Journal of Management Mathematics*, vol. 22, pp. 99–113, 2011.
- [47] F. Godin, J. Zuallaert, B. Vandersmissen, W. De Neve, and R. Van De Walle, “Beating the Bookmakers: Leveraging Statistics and Twitter Microposts for Predicting Soccer Results,” tech. rep.
- [48] M. Bahrami, Y. Findik, B. Bozkaya, and S. Balcisoy, “Twitter Reveals: Using Twitter Analytics to Predict Public Protests,” tech. rep.
- [49] J. Brownlee, “How to Configure the Gradient Boosting Algorithm,” 2016.
- [50] A. Vaastav, “Fantasy-Premier-League,” 2018.
- [51] “FPL Fixture Difficulty Rankings,” 2019.

- [52] D. Gallegos and A. Hau, “Predicting Stock Prices through Textual Analysis of Web News,” tech. rep., 2015.
- [53] H. Patel, “How Web Scraping is Transforming the World with its Applications,” 2018.