



University of London

6CCS3PRJ Final Year Prediction of Football Game Results using Kernel Methods

Final Project Report

Author: Maten Rehim

Supervisor: Dr Zoran Cvetkovic

Student ID: 1638595

April 7, 2019

Abstract

Bookmakers produce risk when offering football bets available to the public. This paper explores new ways of predicting football events based on prior knowledge. Both previously observed data, such as the features from the Manchester City Analytics program and new features such as FIFA data will be considered. From this data, prediction models are created and trained using Support Vector Machines with k-fold cross-validation. The resulting models are then used strategically in combination with several different betting strategies in order to maximise the expected profit per match day. Furthermore, the effectiveness of the whole process is evaluated in empirical simulations using data for the 2012 to 2013 Premier League season.

Originality Avowal

I verify that I am the sole author of this report, except where explicitly stated to the contrary. I grant the right to King's College London to make paper and electronic copies of the submitted work for purposes of marking, plagiarism detection and archival, and to upload a copy of the work to Turnitin or another trusted plagiarism detection service. I confirm this report does not exceed 25,000 words.

Maten Rehim

April 7, 2019

Acknowledgements

I would like to thank my supervisor, Prof. Zoran Cvetokovic for aiding and supervising me throughout this individual project. I would also like to thank Dino Oglic for giving me helpful advice about previous work in this area.

Contents

1	Introduction	3
2	Motivation	5
3	Background	7
3.1	Chance and Predictions	7
3.2	Poisson Distribution	7
3.3	Neural Networks	9
3.4	Logistic Regression	12
4	Data Collection	13
4.1	Group Data	13
4.2	FIFA Data	13
4.3	Venue Data	15
4.4	Team Data	16
4.5	Bookmaker Data	16
4.6	Form Data	16
5	Model	17
5.1	One Against All	18
5.2	All Pairs	18
5.3	C parameter	18
5.4	Kernel Functions	21
5.5	Gamma	22
5.6	K-fold Cross-Validation	23
6	Betting Strategies	24
6.1	Naive Approach	24
6.2	Matched Betting	24
6.3	Kelly Criterion	25
6.4	Mutually Exclusive Kelly Betting	27
6.5	My Approach	28
7	Experiments	29
7.1	K=2	30
7.2	K=3	31

7.3	K=4	32
7.4	K=5	33
7.5	Graphs	34
8	Legal, Social, Ethical and Professional Issues	38
9	Conclusion and Future Work	39
	Bibliography	43
A	Extra Information	44
	A.1 Data Collection	44
B	User Guide	47
	B.1 Instructions	47
C	Source Code	48
	C.1 Instructions	48

Chapter 1

Introduction

This paper will explore ways of using game data to accurately predict football outcomes and will use these predictions to bet on odds from Bookmakers to maximise profit. This process involves collecting data, modelling the data and using the model to predict games. These predictions are used in the betting strategy to place calculated bets to maximise profit.

Football is a world-wide sport with a colossal fan base in every continent of the world. It has continued to grow and develop in many countries to promote team work, leadership among other fundamental skills. One of the many reasons why it is so popular may be due to the fact that it is so simple to play as it only requires a football. This makes it much cheaper to play than sports such as Tennis or Badminton where every player is required to have a piece of equipment if they want to play together. Due to the popularity of football, large sums of money are invested into the sport for new stadiums, sport facilities and advertisement promoting more growth. All Bookmakers thrive on this growth and expand their businesses which has led to a healthy competition between them. As competition increases, Bookmakers start producing riskier bets to encourage people to make bets with them rather than other Bookmakers which leaves a selection of bets to exploit. If money is placed on bets strategically, based on an accurate and highly confident predictive model, then one can potentially make huge profits.

Predicting whether a football match ends in a win, draw or loss is a machine learning problem that falls into a class of supervised learning tasks. Supervised learning problems can be solved in different ways such as using Support Vector Machines, Logistic Regression and Neural Networks. Neural Networks typically do not work well with small amounts of data because they require large amounts of data to optimise the parameters such as weights and biases. In addition, theoretical properties of these methods are not well understood and there

is a substantial risk of overfitting on small sample problems. In Support Vector Machines, overfitting is typically reduced as the generalisation properties are improved by using k-fold cross-validation when selecting relevant hyper parameters.

Supervised learning algorithms are given pairs of inputs consisting of an input object such as a feature vector and an output object called a class label. The model finds a function to distinguish between the different classes which predicts a class label for a new feature vector accurately generalising from training examples to unseen test samples. For example, a feature vector could contain home goals scored and away goals scored then the class label could be 0 for a home win, 1 for an away win and 2 for a draw. The model would train on training data and predict the outcome of new feature vectors. The prediction of the result is used in a betting strategy to maximise profit.

The data used for the feature vector was broken down into six main categories: Group data (section 4.1), FIFA data (section 4.2), Venue data (section 4.3), Team data (section 4.4), Bookmaker data (section 4.5) and Form data (section 4.6). Features that have an effect on the game such as shots taken for strikers and goals conceded will be recorded to predict the outcome of the game. These features are combined to produce a feature vector.

The characteristic property of SVMs is effectiveness in binary classification problems on small samples. There are three main approaches to binary classification problems which are directly predicting class labels with a single SVM, using multiple one against all SVMs (section 5.1) and having a different SVM for each unique pair of class labels (section 5.2). Our problem is a multi-class one and we follow a standard practice to turn this problem into three binary classification problems by considering classifiers for home wins, draws and away wins. The one against all strategy was preferred to directly predicting class labels with a single SVM as it simplifies the problem because we have to predict if a game is won or not won as opposed to predicting a win, draw or loss. In addition, one against all uses more training samples than all pairs. Predicting starting match days is difficult enough as there is very little training data therefore any way to maximise the number of samples as quickly as possible is ideal.

Different approaches for making a betting strategy were considered such as a naive approach (section 6.1), matched betting (section 6.2) and the Kelly Criterion (section 6.3). Also, a successful betting strategy was broken down and analysed to see what factors it considered to maximise profit and minimise risk which were attempted to be incorporated into a new betting strategy.

Chapter 2

Motivation

Football is the largest sport in the world with an estimated following of over 4 billion fans, 1.5 billion more than the second most popular sport Cricket [26]. There are professional football leagues all over the world. For example: the Premier League in England, Brasileirão in Brazil and K-League 1 in South Korea. In each country, there are multiple leagues in a hierarchical structure, where the top-tier teams can be promoted into the league above and the bottom-tier teams can be demoted into the league below at the end of the season. Each team in every league play each other team twice, one at home and one away which makes it balanced due to home team advantage. Studies show that home advantage was greatest in football compared to other sports, with the home team obtaining roughly 64% of all points gained in the Premier League [28]. Home teams have more crowd support due to season tickets which allows fans to watch all home games for a discounted price. In addition, travel fatigue is far less for the home team and they are very familiar with the pitch conditions as opposed to the away team.

There are currently 211 national football teams with official FIFA memberships [16]. The best 32 in-form countries take part in the world cup which is the largest sport event in the world, with over 3.5 billion watchers [14]. As a result, football bets have dominated other sports in the betting industry with roughly 70% of the bets every year being football related with large competitions like the World Cup being the most lucrative time to bet [20]. Over £2.5 billion was spent on betting in the 2018 world cup in the UK [20]. Bookmakers produce bets for a certain event to happen, if you pick random events than you should win 50% of the time on average. However if you can pick events strategically, then you can potentially win above 50% in which case you make profit.

The Bookmakers have a different house margin depending on the country. On average,

Bookmakers have a house margin of around 0-12%, whereas state-owned Bookmakers have a much higher house margin, such as Germany having 25% [33]. Free market Bookmakers have better deals in general because they are competing with each other, compromising attraction and profit, hence overall lowering house margins. This leaves a gap for people to exploit if they make smart bets in different Bookmakers in certain countries. A common approach used is matched betting which is a technique used by individuals to profit from the free bets and incentives offered by Bookmakers. The individual puts a bet for an event to occur and against the same event guaranteeing success so they benefit from bonus deals such as having a free £50 bet after making your first bet. On average, matched bettors make around £500 per month. Individuals who learn more complex matched betting techniques can earn up to £1000 a month [18].

Algorithms are used to search for bets in Bookmakers that have a high outcome for good profit. They observe the previous data and predict the outcome of the game and continue to do this to train the models in the algorithms. The features of the previous data are all factors that the creator of the algorithm think have an effect on what they are trying to predict, such as the final score of a football match. Prior data becomes increasingly important to help predict football matches as new factors considered could increase the accuracy of the algorithm. The Sports Analytics market is expected to grow by nearly 5 times from 2016 to 2021 at a compound annual growth rate of 37.9% [13].

Chapter 3

Background

3.1 Chance and Predictions

Previous statistics play a large role in determining the outcome of games which is why data is needed for a computer model to predict the outcome of a football match. But is it necessary to have a computer do the predictions for us? Experienced football pundit Mark Lawrenson, predicted the 2012-2013 Premier League season with great accuracy, correctly predicting 200 out of 379 results in the process [21]. If a £1 bet was put on all 379 results on Mark Lawrenson's predictions then you would have made roughly £18.60 from the Bookmakers [21]. However, QPR was predicted to be last place by Lawrenson and ended up finishing 8th exceeding most expectations showing that football will always have an unpredictable side to the game [21].

3.2 Poisson Distribution

The simplest approach to predict football scores is to use the Poisson Distribution like MJ Moroney in 1956 [27]. This model is used to predict the expected number of goals for both the home team and away team respectively. The expected number of goals for the home team was calculated by multiplying the home team's attack strength with the away team's defence strength and the average number of home goals in the season. The home team's attack strength is the ratio between the home team's average number of goals scored at home and the average number of goals scored at home for all teams. The away team's defence strength is the ratio between the away team's average number of goals conceded away and the average number of goals conceded by all teams.

Selecting a representative data range is crucial when calculating attack strength and defence strength as a large data sample will include games from too long ago which might give inaccurate values [23]. Having a short data sample will increase the likelihood of outliers skewing the data [23]. Normally, a sample size of a season is used as it is a good representation of a team's current ability however this does not take into account players brought in the summer transfer window before the start of the current season. The expected number of goals are usually decimals and in football no game ends 1.62-0.34 therefore the probability of scoring a certain amount of goals must be calculated for each team playing.

$$P(X = x) = \frac{\lambda^x e^{-\lambda}}{x!}$$

- x: exact number of successes
- e: Constant equal to approximately 2.71828
- λ: mean of successes in the given time interval

Figure 3.1: The Poisson Distribution Formula
[12]

Poisson Distribution, a formula created by Simeon Denis Poisson (as seen above in figure 3.1), allows us to use the expected average number of goals to find the probability of a team scoring a certain amount of goals [19]. These probabilities are used to predict certain scores that the Bookmaker is offering. The probability is not always accurate as many factors effect the outcome of a game which the Poisson Distribution does not take into consideration such as weather, travel fatigue and difficult fixtures [27].

A key assumption of the Poisson Distribution is that the number of events is independent of time [31]. In a game with two goals, the second goal has no relation to the first goal which is not always the case in football as team's with very poor defence are more likely to concede more goals after conceding one. In addition, when a team scores first than the opposing team is forced to play more offensive so the defence suffers especially towards the end of a game. Opta collected data on when 21,871 goals were scored in the Premier League and the most common time period was 81-90 minutes [35].

The Poisson Distribution solves a harder problem than necessary as it predicts the probability of each possible result whereas SVMs simply tell us if a team wins or not. This is more relevant to our problem as we want to know whether an outcome occurs or not and bet accord-

ingly. In addition, the Poisson Distribution does not express the many factors that effect the result of a football match as the model is not very complex. SVMs can produce a hyperplane in hyperspace therefore allowing any number of factors to be considered such as team budget and season ticket prices.

3.3 Neural Networks

Neural Networks are supervised learning methods which process information similarly to the human brain. Each node in a Neural Network is connected to other nodes. The nodes represent nerve cells (which is shown in figure 3.2 below) called neurones and the links connected between them represent the synapse. In the brain, neurones receive an electrical impulse from other neurones, sums the incoming electrical impulses to generate input in the cell body. If the sum reaches a threshold value, the neurone fires an electrical impulse down the axon to the other neurones. The point of connection of one neurone and another neurone can vary in strength which results in connections being inhibitory or excitatory. This results in another neurone potentially receiving an electric impulse and the whole process is repeated through various neurones.

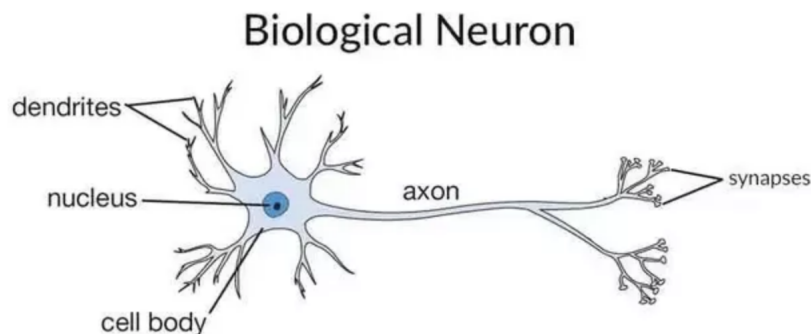


Figure 3.2: Biological Neurone
[4]

Certain Neural Networks can be feed-forward networks which means that the same node can not be revisited again. These types of Neural Networks suffer from a problem called vanishing gradient or exploding gradient. Vanishing gradient occurs when the gradient becomes too small and the network is not learning whereas exploding gradient occurs when the gradient becomes too large and tends to infinity. The main cause is not properly setting parameters and hyper-parameters of the network. Parameters can consist of weights and biases while hyper-parameters could be the learning rate, the number of times you iterate over the entire dataset

or the number of batches used during training. Due to this, the Neural Network may take a longer time to train and learn from the data. This can lead to less or no convergence of the Neural Network parameters. If convergence is achieved there is a good chance the accuracy will be very low. Fortunately, there are some solutions to the vanishing gradient problem such as Long Short Term Memory (LSTM) networks and Gated Recurrent Unit (GRU) networks.

3.3.1 Long Short-Term Memory

Long Short-Term Memory networks is a type of recurrent Neural Network. Recurrent Neural Networks contain feedback loops which allow information to be held like memory until it is decided to be used in the network. LSTM networks was created to store inputs based on their importance. If the LSTM network considers an input as important early on in the network, then it will store it for a long amount of time however if deemed not important it will be used in the network quickly or not at all. The LSTM network solves the vanishing gradient and exploding gradient problems. There are two factors that affect the magnitude of gradients which is the weights and the activation functions (or more importantly the derivative). If either of these factors are smaller than 1 then the gradient may vanish over time whereas if the factors are greater than 1, then the gradient may explode to infinity. For example, the tanh derivative is always less than 1 except for 0 as input. The sigmoid derivative is even worse as it is always less than or equal to 0.25 therefore the sigmoid activation function is much more susceptible to the vanishing gradient problem than the tanh activation function. In the LSTM network, the activation function is the identity function with a derivative of 1. So, the back propagated gradient does not vanish nor explode when the input passes through.

3.3.2 Gated Recurrent Unit

In 1997, LSTM networks solved the long-standing problem of vanishing gradient [22]. Multiple studies revealed that some of these components in the hidden layer of the LSTM network does not hurt the performance so much and in 2014 Cho et al proposed GRUs [25]. Over this time, weights were reduced which made training the Neural Network much faster. GRUs do not maintain an internal memory state unlike LSTM networks which slows down performance and overall efficiency of the network. However, LSTM networks remember longer sequences than GRUs and outperform them in longer duration tasks. A study was done to find out the best performing Neural Network from GRU and LSTM networks and it was determined that neither was strictly better than the other [22]. The best approach would be to try both and compare

the accuracy of the predictions on new data [22].

3.3.3 Dropout

A problem both LSTM networks and GRUs suffer from is overfitting due to a large amount of parameters. This problem is worse for LSTM networks as they require more parameters than GRUs. A model that overfits the training data will result in low accuracy and it can be minimised using Dropout. The idea of dropout is to randomly cuts the connections between nodes to prevent the units from adapting too much to the training data. In a study, dropout improves the performance of Neural Networks on supervised learning tasks such as vision, speech recognition, document classification and computational biology [34].

The problem Neural Networks, such as LSTM networks and GRUs, solve is non-convex which means that it finds a local optima which could potentially perform badly. In contrast, SVMs solve a convex optimisation problem which guarantees a globally optimal solution. SVMs are theoretically well understood and can give more of an insight of how the data is being manipulated whereas Neural Networks reveal little to no information during the data manipulation.

3.3.4 Negative Empirical Results

Neural Networks can be used effectively to predict football matches but there are some key flaws. For one they require a lot of training data. Daniel Petterson and Robert Nyquist produced a paper on predicting football scores using LSTM networks. They initially started with a prediction accuracy of 33.35% for many-to-one and 43.96% for many-to-many which is very low as there was a lack of training data at the start of the season. Training data in the previous season could have been used with a lower weighting than recent games to produce more training data. However in the end, the accuracy went to 98.63% for many-to-one and 88.68% for many-to-many. This is really high but if you start placing bets at the start of a season with an accuracy of 30%-40% then statistically you will not have much money to place for the remainder of the season. The Neural Networks are black boxed due to the hidden layer which provides very little insight into what these models really do. Other supervised learning such as Logistic Regression gives much more information about the correlation between the input and output of the model.

3.4 Logistic Regression

Logistic Regression can be used for binary supervised learning problems where it takes a feature vector of size x and outputs the predicted label. The feature vector contains features that we assume has an effect on the prediction such as shots taken for predicting a football match. Each feature can be checked to see if it has an effect on the prediction. If the feature does not have a significant effect on the prediction then it can be removed to increase the efficiency of the model as less variables will have to be considered. For example, trainer brand may not have an effect on the players performance during a game so may be taken out of the feature vector. Logistic Regression was used by Darwin Prasetyo and Dra Harlili to predict games in the 2015/2016 Premier League season [29]. The training data used was the 2011/2012 Premier League season until the 2015/2016 Premier League season [29]. The variables considered was home offence, home defence, away offence and away defence [29]. They yielded a prediction accuracy of 69.5% which was an improved accuracy to Snyder [32].

An issue with these models is that on a large dimensional feature representation with a huge number of highly correlated variables the approach does not perform very well. In addition, the computational complexity of the approach is cubic in the size of the feature representation. This can be problematic for representations with thousands of features, which is what happens in our problem. SVMs have a cubic computational cost in the number of samples and there is only a small set of examples, less than 1000 games. This makes the approach based on SVMs more efficient than Logistic Regression for this particular problem.

Chapter 4

Data Collection

4.1 Group Data

Previous feature data is essential when predicting games as there could be correlations between the variables used and the final score. The first feature that was collected was player data which came from Manchester City's analytics program for the Premier League in 2011-2012. There are hundreds of features recorded for each game and for each player for both the home and away side including shots on target, blocked shots and corners taken. The player data was collected for each team and separated into goalkeeper data, defender data, midfielder data and striker data. Each group was split and averaged to produce data for each respectively. However, not all players played the whole game therefore each player in the group was weighted by their time played divided by the time played for the whole group. The result was summed throughout the group to produce group data representative of the whole game. This data was found for both the home and away side for the previous k game where k is a hyper parameter of the problem.

4.2 FIFA Data

FIFA data was collected for each team by using FIFA cards which give a summary of stats, generated by experts reviewing player's abilities when watching them play. The player's league and manager also have an impact on the ratings. Mueller-Moehring stated "When you look at passing completion, if you play for Bayern Munich or if you play for Manchester City or if you play for Pep Guardiola, if your system is based on possession, you will have more successful passes than other players" which indicates the level of depth of analysis used by FIFA [30].

FIFA ratings also factor experience which is of high importance when measuring the strength of a team. Standardly, most teams have a mix of experienced players and youngsters so the experienced players can help them become better players.

The features taken for all players, except goalkeepers, are overall rating, pace, shooting, passing, dribbling, defending and heading [11]. Each feature is split into subcategories. For example, shooting is split into heading, shot power, finishing, long shots, curve, free kick accuracy, penalties and volleys [11]. Dribbling is an important factor to consider because if one team has better dribbling abilities than the opposing team then they can get past one or two players. This then forces other players to go towards the player with the ball leaving their man unmarked creating gaps in the defence of the opposing team which can be exploited to push closer to the opposing goal. To highlight how important dribbling is Lionel Messi is considered to have one of the best dribbling abilities, if not the best, and is considered to be the best player in the world. He has obtained 5 ballon d'Ors [2] which is given out to the best individual player in each football season which lasts for a year. This record is only matched by one other player called Cristiano Ronaldo.

Pace is an important feature when predicting the football score as it gives an offensive and defensive benefit. For example, the faster player can sprint away from the player marking him which could allow a team mate to play an open pass to them. This gives a risk free way to move up the pitch. While the player sprints away from the defender it gives the player more time to think and consider the best thing to do next with the ball. In addition, the faster player has an edge to track back to a player if he is in a lot of space and sprinting towards your goal. The faster player would get goal side of the opposition to force them to move side ways or backwards which gives more time for your team to get back and defend. Also, having a high pace stat would mean that they have more power on their shots as they would get more momentum which would make it more likely for them to score if their shot was on target.

Shooting is a necessary feature to take into account as it results in a higher conversion rate of goals given key chances. Taking key chances in a football match can determine the outcome of the final game and sometimes is more important than many other statistics recorded in a football match. For example in the Champions League, a tournament between top European clubs, Barcelona lost to Celtic 2-1 despite having 78% possession and 789 more total passes [3]. This shows how an inferior team can beat a superior team through taking advantage of their minimal chances that they produce on the night of the game.

A skill that is sometimes overlooked by many people who watch professional football is

defending. Defenders do not always get the same amount of credit as attackers as they do not usually get the goals and celebrate with the fans. However, defending is extremely important when deciding the winner of a football match. The best players in football need the ball to be able to get the best out of them. For example, if the best player on your team has really good shooting skills then he can not shoot if they do not have the ball. This is where defending comes in as it allows the key players to take advantage of their abilities to more effect. In addition, it is important to consider that a team with good defensive capabilities are more likely to have higher possession during the game. A top football coach called Pep Guardiola implemented a football tactic which involved being very close to every opposing player when losing the ball to regain possession. Then when they had possession they would keep the ball and try to open up the opposition defensively. Guardiola required high work rates from all players to regain the ball as it is much less effective when a single player is chasing the ball due to the opposition just passing the ball around them. In the 2008/2009 season, Guardiola won the treble with Barcelona through this strategy. They won the Copa del Ray, La Liga and UEFA Champions League which are all the trophies available [1].

In FIFA, goalkeepers have the attributes: rating, diving, kicking, reflex, speed and positioning which are not subdivided [11]. Goalkeeper speed is very crucial in a game as it can sometimes prevent the strikers of the opposing team from getting to the ball. This could reduce the number of goal scoring chances of the opposition. Also, if a team knew that the opposing keeper was slow of his goal line then you have more space in front of the keeper to play a pass decreasing the risk of the pass. Goalkeepers occasionally have to perform two saves in succession. If the keeper can get up quickly after saving the first shot then they can get close to the ball for the second shot. This narrows down the angle of the shot which makes it more difficult for the striker to score. In addition, a common way strikers score is by going around the keeper with the ball to score in an empty net which can be prevented if the keeper quickly gets to the ball before the striker.

4.3 Venue Data

Venue data was obtained for the home team and away team. Stadium capacity data is collected as football teams with more fans have a higher budget, expectations and support which can be put into the team to improve their performance. In addition, distance to the stadium is calculated for the away team as this acts as an indication of their travel fatigue before the match starts. Google maps was used to find the longitude and latitude of each stadium in the

Premier League. The Haversine formula was used to calculate the distance travelled from the difference from latitude and longitude as shown in figure 4.1 below.

$$d = 2r \sin^{-1} \left(\sqrt{\sin^2 \left(\frac{\phi_2 - \phi_1}{2} \right) + \cos(\phi_1) \cos(\phi_2) \sin^2 \left(\frac{\psi_2 - \psi_1}{2} \right)} \right)$$

Figure 4.1: The Haversine Formula
[10]

4.4 Team Data

Team data was gathered in the form of season wages, points gained and final league standing in the previous season. Not all teams in the previous season are part of the new season as 3 teams are relegated. The 3 teams promoted into the next season will store the values for the 3 teams relegated in the prior season. Season wages are a good indication of the transfer potential of a club which can have a huge impact in team performance if they can get some new better players.

4.5 Bookmaker Data

The Bookmaker data contained odds from various Bookmakers, for each game, including Bet365, Ladbrokes and WilliamHill [15]. For each betting company, three odds were calculated: home win odds, away win odds and draw win odds. Betbrain collects betting odds from many Bookmakers and the average was calculated for certain odds like average of more than 2.5 goals per game, maximum over 2.5 goals and average under 2.5 goals.

4.6 Form Data

The previous k games of stats were calculated for the home team and away team. The stats included are shot on target, fouls committed and corners among others. This data is gathered for the previous k home games, away games and combined than the average of each is calculated. These factors are important to include as there are many links between attributes like shots on target and predicting whether a team wins because the team with more shots is more likely to score.

Chapter 5

Model

Support Vector Machines are a supervised learning model which can be used to predict the output given a feature vector of arbitrary size x . The feature vector is normalised for reasons of numerical stability and faster convergence to the optimal solution. It finds correlations between the variables and produces a hyperplane in $(x-1)$ -dimensional space (as seen below in figure 5.1) after being trained with data. In the Binary Support Vector Machine model, if the feature vector lies on one side of the hyperplane then it will be labelled as the first class otherwise if it lies on the other side it will be labelled as not the first class. All Support Vector Machines are inherently binary however they can be used to predict multi-class problems. The two common ways this can be done is one against all or all pairs.

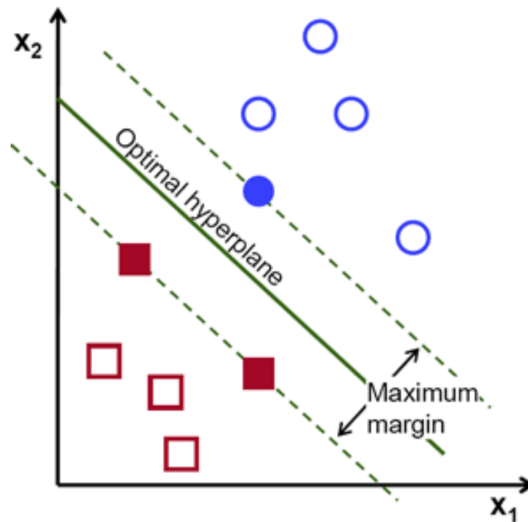


Figure 5.1: Hyperplane example
[17]

5.1 One Against All

One against all involves creating a SVM model for each class and predicting if the feature vector is in the class or not. If the feature vector is in more than one class then the class with the highest likelihood is picked. However, if the feature vector is predicted to not be in any class than it picks the class with the smallest distance between the feature vector and the class hyperplane. In theory, if you have k classes then you will have to produce k SVM models which will increase time consumption and might reduce efficiency as you have to consider the error of each SVM model. In addition the highest likelihood class could be effected if only one SVM comes up with a wrong prediction. One Against All will be used for the SVM in this paper as it solves an easier problem for predicting games as it simply has to find out whether the game is a win or not rather than if a game is a win, draw or loss. After every 10 match days are predicted, these will be included in the next training set when predicting the next 10 games.

5.2 All Pairs

The multi-class labels are all split into pairs so for k classes you will have $k(k-1)/2$ SVM models. Each SVM model will be trained with feature vectors that are labelled as one of the two classes the SVM predicts. As a result, there is considerably less training data for each SVM whereas One Against All uses all training data for each SVM. Therefore One Against All will be used in my experiment to maximise the training data which will likely increase the accuracy of the predictions. Training data is vital for the first match days predicted as there will not be much data. So maximising training data is essential to ensure that any model gives accurate prediction including our problem. In addition, the training time for All Pairs is considerably greater than One Against All because there are fewer SVMs. For example, if there is a problem with 1000 classes then All Pairs uses $1000(999)/2=499500$ SVMs however One Against All uses 1000 SVMs. Decreasing the number of SVMs allows us to do k -fold cross-validation with larger C and γ ranges which could improve our model accuracy.

5.3 C parameter

If a feature vector is very far into one side of the hyperplane then it is easy to classify the point with high confidence. However, if the feature vector is close to the hyperplane then it makes it more difficult to predict. The hyperplane has a margin which is on the boundary of the two

closest data points from each class. The ideal scenario is to maximise the distance between the hyperplane and the margin while reducing the penalising slack variables. To explain the hyper parameter C in more depth we first have to consider how to find the shortest distance from a point to a hyperplane (plane used for simplicity).

5.3.1 Shortest Distance from a point to a plane

We want to find the shortest distance between a point on the hyperplane P and a point Q which is not on the hyperplane (as shown below in figure 5.2). In addition, \vec{n} is the normal to the hyperplane. The difficulty in the problem is that we do not know where P lies. If we did know we could find the euclidean distance between the two points to calculate the shortest distance. We will use the notation $||x||$ which is the magnitude of x and $|x|$ which is the absolute value of x. As a result, we do the following to obtain the distance:



Figure 5.2: Point and a hyperplane
[5]

$$\text{Shortest Distance} = \text{Projection of } \vec{PQ} \text{ onto } \vec{n} \quad (5.1)$$

$$\text{Shortest Distance} = \left\| \frac{\vec{PQ} \cdot \vec{n}}{||\vec{n}||^2} \cdot \vec{n} \right\| \quad (5.2)$$

$$\text{Shortest Distance} = \frac{|\vec{PQ} \cdot \vec{n}|}{||\vec{n}||^2} \cdot ||\vec{n}|| \quad (5.3)$$

$$\text{Shortest Distance} = \frac{|\vec{PQ} \cdot \vec{n}|}{||\vec{n}||} \quad (5.4)$$

In the equations above, \vec{n} is the same as w in the general formula of a hyperplane ($w \cdot x + c = 0$). To optimise the margin of the hyperplane, we need to maximise the distance. This problem can be converted to a minimisation problem as we can minimise $||w||$ to maximise the distance.

The minimisation problem can be categorised as an optimisation problem.

5.3.2 Optimisation Problem

The Optimisation problem we have currently is the following:

$$\min_w ||w||, \text{ subject to } y_i(W^T x_i + c) \geq 1, i = 1, 2, \dots, N \quad (5.5)$$

The equation above is a convex optimisation problem which has the theoretical property that there is more than one optimal point. However, strictly convex problems guarantees the existence of a single optimal point. Let us say that our hypothesis is $f(x)$. A convex problem would satisfy the following constraint $f''(x) \geq 0$ whereas a strict convex problem would satisfy $f''(x) > 0$ which makes it a simpler problem. Therefore, we can change it from convex to strictly convex by integrating with respect to w . Therefore we get the following problem.

$$\min_w \frac{1}{2} ||w||^2, \text{ subject to } y_i(W^T x_i + c) \geq 1, i = 1, 2, \dots, N \quad (5.6)$$

This equation above seems like a good way of maximising the margin however it does not take noise into consideration. For example, one point may have more noise than another point and it may cause the data points from the two respective classes to not be linearly separable by a hyperplane. In these cases, you have to accept that some number of points will be incorrectly classified and slack variables (ζ) are used to ensure that you reduce this to a minimum. We use ζ to satisfy a new constraint even if the example does not meet the original constraint above. A problem arises when we can choose large values of ζ so that $y_i(W^T x_i + c) \geq 1 - \zeta_i$ will easily be satisfied. We can use regularisation to minimise ζ .

5.3.3 Regularised Optimisation Problem

The regularised optimisation problem we have currently is the following:

$$\min_{w, \zeta} \frac{1}{2} ||w||^2 + \sum_{i=1}^N \zeta_i, \text{ subject to } y_i(W^T x_i + c) \geq 1 - \zeta_i, i = 1, 2, \dots, N \quad (5.7)$$

We want to make sure that we do not minimise the objective function by choosing negative ζ_i values. Therefore we need to add the constraint $\zeta_i \geq 0$. A regularisation parameter C can be inserted into the equation to give us more control for deciding how much we want to avoid misclassifying each training example. When we introduce the constraint and the regularisation

parameter C we get a new equation as seen below in figure 5.8.

$$\min_{w, \zeta} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \zeta_i, \text{ subject to } y_i(W^T x_i + c) \geq 1 - \zeta_i, \zeta \geq 0, i = 1, 2, \dots, N \quad (5.8)$$

Large values of C implies that the minimisation will focus on reducing the number of incorrectly classified training examples and enforcing the inequality constraints. As a result, the margin which is given as $\frac{1}{2} \|w\|^2$ might not be minimised as it is not a priority and could be narrow (the optimisation will not focus on decreasing $\|w\|$ to zero). This could then result in overfitting because the hypothesis could be overly reliable on training samples and not generalise to unseen feature vectors. If we reduce the value of C , then the focus of minimisations turns to $\|w\|$ and consequently this maximises the margin. As a result, there might be a number of examples which are incorrectly classified because the focus of optimisation is not on pushing the slack variables to zero and thus reducing the number of incorrectly classified examples. A good choice of C is typically determined via cross-validation to ensure good generalisation properties and reduce the chance of overfitting and under-fitting. In figure 5.4 below, the circled exemplars are support vectors. Support vectors are feature vectors inside the margin where $y_i(W^T x_i + c) \leq 1$.

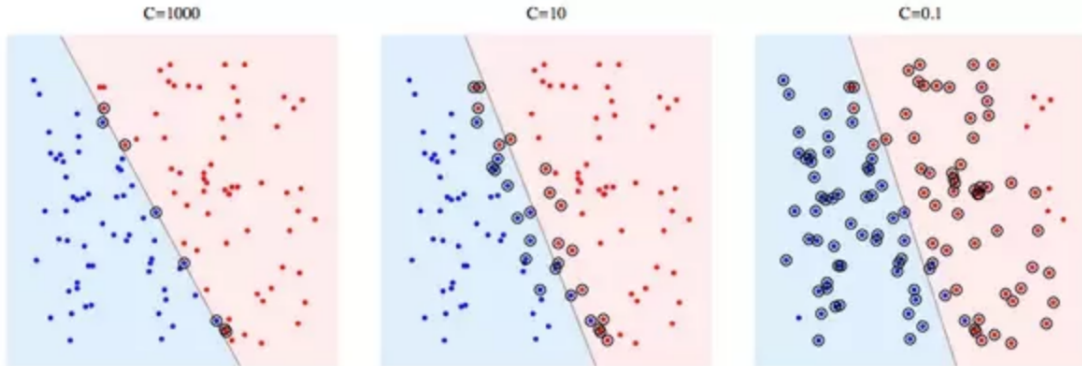


Figure 5.3: Hyperplane examples
[8]

5.4 Kernel Functions

Every support vector machine uses a kernel function to measure the similarity between two feature vectors. The most common types of kernel functions are:

- **Linear:** $K(x, y) = x^T y$

- **Polynomial:** $K(x,y) = (x^T y + 1)^d$
- **RBF:** $K(x,y) = \exp(-\gamma ||x - y||^2)$

Any of these functions can be used to give an accurate model however depending on the nature of the problem some can excel over others. The RBF kernel uses euclidean distance whereas the other 2 functions use the dot product. These are both very similar in nature as a higher value means that the points are further away and more dissimilar. The main difference is if the two feature vectors are on the opposite side of the origin than the euclidean distance will be higher whereas the dot product will be a lower. In addition, all feature vectors can be translated by the same amount for the RBF kernel and it would still produce the same result.

5.5 Gamma

Gamma, also known as γ , is a hyper parameter for RBF kernels when comparing two feature vectors. This hyper parameter defines how far the influence of a single training example reaches. For small values of gamma the model is too constrained and cannot capture the shape of the data [9]. The region of influence of a support vector would include the whole training set. As a result, it will perform similarly to a linear SVM model. Conversely, if gamma is too large the radius of the area of influence of a support vector will only include itself and changing the value of C will not prevent overfitting (as seen in figure 5.4 below) [9]. Overfitting occurs when the model is well trained on the training data and does not generalise well when you predict new outcomes. In practice, k-fold cross-validation is used to avoid overfitting when picking hyper parameters which produce models with the best accuracy. In figure 5.4 below, the red region is the decision region for red points and blue regions is the decision region for the blue points.

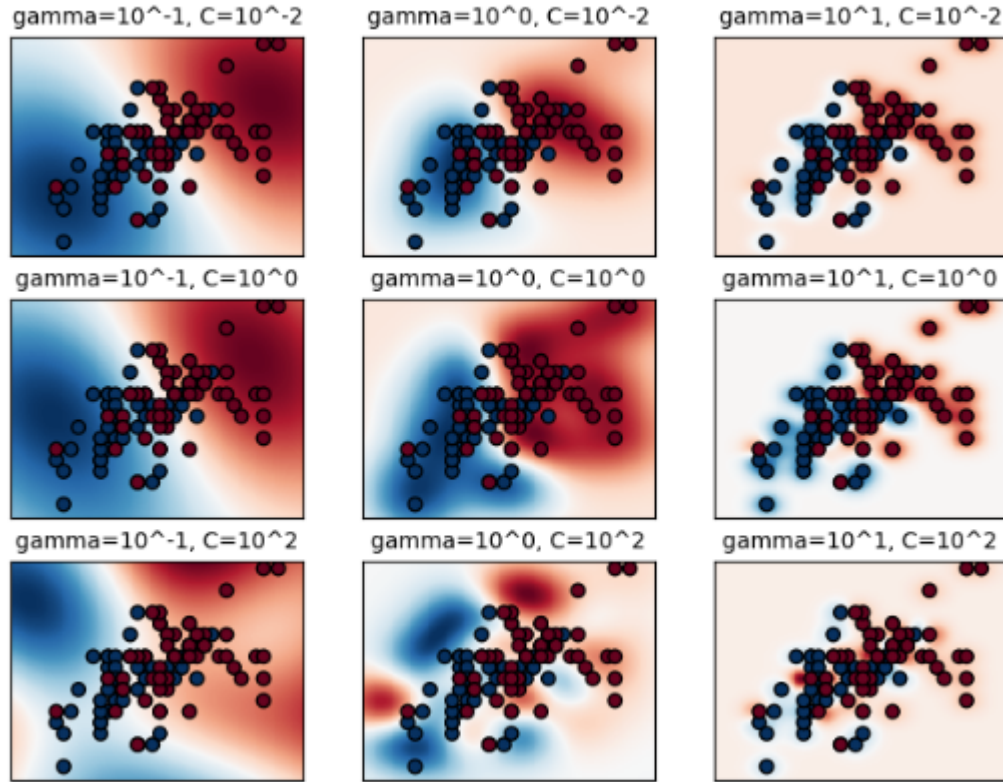


Figure 5.4: Hyperplane examples
[9]

5.6 K-fold Cross-Validation

K-fold cross-validation is a process where the data is split into k folds. The model trains on $k-1$ folds and the last fold is used to test the model. This process is repeated k times so that each fold is used to test which avoids overfitting as it is picking the model with the highest accuracy when predicting new results. K-fold cross-validation can be used for picking both γ and C in the Support Vector Machine model, potentially producing different optimal parameters for the 3 Support Vector Machines. When using k -fold cross-validation, initially it produced very low values for both γ and C which could be caused by an insufficient amount of data. The model aims to avoid this by skipping a match day so the training dataset is larger.

Chapter 6

Betting Strategies

6.1 Naive Approach

The naive approach would be to place a bet on the most probable outcome predicted by the model. The amount placed on each of these outcomes could be constant or vary depending on the probability of the prediction. For example, say the probability of predicting a win from the model was 80% and not winning was 20%, the probability for a draw was 20% and not drawing was 80%, and the probability of a loss was 30% and not losing was 70% then you would place more money on the higher percentage bet as it has a lower risk of loss. Therefore you could decide to only bet when the model predicts above 60% for an outcome. The percentage could change depending on how much risk you want to take. This approach is very simplistic but can be improved in different ways to maximise profit. A way it can be improved upon is by using Matched Betting.

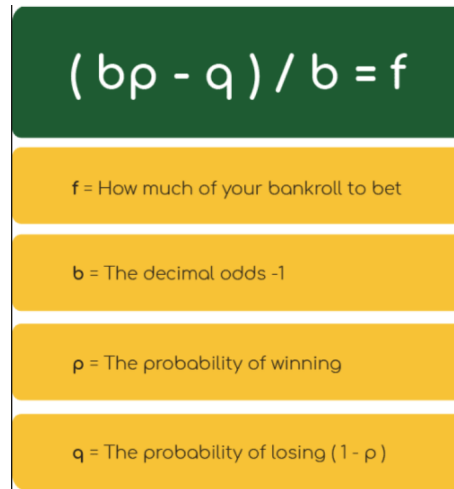
6.2 Matched Betting

Matched Betting is a technique used by betters to ensure that they profit from the free bets and incentives offered by Bookmakers. Usually Bookmakers encourage players to bet by giving them a free bet after their first bet. For example, "Bet 10 pounds today and get 30 pounds on the next bet". Therefore, if you bet for an outcome to happen then use the free bet to bet against the same outcome then you are ensured to always make a profit. The same Bookmaker can be used to do matched betting however you can use different Bookmakers. Different Bookmakers may provide different odds so you can maximise the profit by matched betting by exploiting

the competition between them. The term "matched" is given because the two bets you have made are matching each other. Most people exploit this more thoroughly if their account on a Bookmaker has just been made as there are better deals to encourage people to bet more. Matched Betting is paired with the Kelly Criterion. The Kelly Criterion is used to calculate how much money of your initial budget you should put on each bet.

6.3 Kelly Criterion

The Kelly Criterion is a formula (as shown below in figure 6.1) used to calculate the fraction of the current total to wager which maximises profit as the number of bets tends towards infinity. Although this seems like a relatively straight forward task, it is more complex than many may believe. In a study [24] [6], each person was given \$25 and asked to bet on a biased coin that would land on heads 60% of the time. Surprisingly, 28% of the people lost all the money, and the average total at the end was only \$91. Only 21% of the participants reached the maximum of \$250. 18 of the 61 people taking part bet everything on one toss which might have been lowered if they were betting with their own money rather than someone else's. Remarkably, two-thirds gambled on tails at some point in the experiment which does not make much sense intuitively given the nature of this task.



$$(bp - q) / b = f$$

f = How much of your bankroll to bet

b = The decimal odds -1

p = The probability of winning

q = The probability of losing (1 - p)

Figure 6.1: Kelly Criterion
[7]

The formula above may be simplified where π_i is the probability of outcome i and O_i is the decimal odds:

$$f_i = \frac{(b * p - q)}{b} \quad (6.1)$$

$$f_i = \frac{(O_i - 1)\pi_i - (1 - \pi_i)}{O_i - 1} \quad (6.2)$$

$$f_i = \frac{O_i\pi_i - \pi_i - 1 + \pi_i}{O_i - 1} \quad (6.3)$$

$$f_i = \frac{O_i\pi_i - 1}{O_i - 1} \quad (6.4)$$

The Kelly Criterion finds a theoretical optimum of a single bet by maximising the expected value of $\log_e(C)$ where C represents the budget. It does this by not placing any bets when f_i is negative as this represents bets where you are expected to lose money. A negative f value is produced when $E[ROI]_i < 1$. $E[ROI]_i$ represents the expected return of investment which can be calculated by doing $O_i * \pi_i$ where i is a bet. In addition, if the odds of i is less than 1 then f can also be negative. If both of these events happen simultaneously then f_i will be positive but you would never make a bet where the expected return of investment is less than 1. So it can be simplified to $E[ROI]_i < 1$. For example, if $\pi = \{ \pi_H = 1, \pi_D = 0, \pi_A = 0 \}$, $O = \{ O_H = 5, O_D = 1, O_A = 7 \}$ then $\pi * O = \{ E[ROI_H] = 5, E[ROI_D] = 0, E[ROI_A] = 0 \}$. After using the Kelly Criterion strategy we get $f_H = 1$ which indicates that we should invest all our budget on the home win to maximise our budget.

A problem arises when we use the Kelly Criterion to predict football outcomes as it does not generalise well for multiple interdependent results. An example of multiple interdependent results in football can be home wins, draws and away wins. They have an effect on each other therefore the Kelly Criterion may not always produce an optimum value when betting on a single football game. In certain scenarios, it might be more beneficial to bet when $E[ROI]_i < 1$ which the Kelly Criterion does not allow. For example, if $\pi = \{ \pi_H = 0.3, \pi_D = 0.7, \pi_A = 0 \}$, $O = \{ O_H = 5, O_D = 1, O_A = 2 \}$ then $\pi * O = \{ E[ROI_H] = 1.5, E[ROI_D] = 0.7, E[ROI_A] = 0 \}$. After using the Kelly optimum strategy we get $f_H = 0.1$ and f_D is not used as $E[ROI_D] < 1$. Therefore a tenth of our budget should be spent on the home odds and leaves us with nine tenths of our budget. This can be represented by this equation $E[C] = 3/10*(5*0.1C) + 7/10*(9/10)C = 0.78C$. However this expected value of the budget is sub optimal as we can increase $E[C]$ by betting half of our budget on the home win and the other half of our budget on a draw. This would give us $E[C] = 0.3*(5*0.5C) + 0.7*(0.5C) = 1.1C$. The new expected value of C is higher than the previous despite us using the draw odds which has an $E[ROI_D] < 1$. As the Kelly Criterion does not allow this, it suggests that the Kelly Criterion may have to

be altered to maximise $E[C]$ if used. An example of an altered Kelly Criterion is the Mutually Exclusive Kelly Betting algorithm proposed by Snyder [32].

6.4 Mutually Exclusive Kelly Betting

The Mutually Exclusive Kelly Betting algorithm is a betting approach which has correctly predicted 169 out of 331 bets correctly and produced £65.98 of profit per game [32]. The algorithm uses a different way to get the fraction to put on each bet. It uses S^* which represents the set of all bets that should be made. S^* is initially empty but increases as you insert bets from best to worst $E[ROI_i]$. The bet is only inserted to S^* if the $E[ROI_i] > R(S^*)$ where $R(S^*)$ is the reserve rate of S^* . This indicates that making the bet increases the total budget more than if the bet was not made (S^* is unchanged). The formula to produce $R(S)$ is shown below [32].

$$R(S) = \frac{1 - \sum_{i \in S^*} \pi_i}{1 - \sum_{i \in S^*} \frac{1}{O_i}} \quad (6.5)$$

6.4.1 Pseudocode

Input: A vector of odds \mathbf{O} which contains O_H , O_D and O_A . A vectors of predicted probabilities $\boldsymbol{\pi}$ which contains π_H , π_D and π_A .

Output: A vector of fractions \mathbf{f} which contains f_H , f_D and f_A . These represent how much of the budget should be placed on each bet.

Method M-E KELLY BETTING($\mathbf{O}, \boldsymbol{\pi}$) [32]

Local variables:

$\mathbf{S}^* = \emptyset$, \mathbf{S}^* represents the optimal odds to bet on.

$R(\mathbf{S}^*) = 1$, $R(\mathbf{S}^*)$ represents the reserve rate of \mathbf{S}^* .

$E[\mathbf{ROI}] = \boldsymbol{\pi} * \mathbf{O}$, $E[\mathbf{ROI}]$ represents the expected return of investment.

1. Sort $E[\mathbf{ROI}]$ in decreasing order so the best bets are placed at the start and it progressively gets worse.

2. For each bet $\in E[\mathbf{ROI}]$ do:

If $E[ROI_{bet}] > R(\mathbf{S}^*)$ then

$\mathbf{S}^* = \mathbf{S}^* \cup \text{bet}$

$$R(S) = \frac{1 - \sum_{i \in S^*} \pi_i}{1 - \sum_{i \in S^*} \frac{1}{O_i}}$$

3. For each possible bet do:

If $\text{bet} \in S^*$ then

$$f_{\text{bet}} = \pi_{\text{bet}} - \frac{R(S^*)}{O_i}$$

else

$$f_i = 0$$

return f

6.5 My Approach

My approach uses matched betting to distribute the initial budget. The predictions for the feature vector of the outcome is calculated and it bets a tenth of the current budget. For example, say that Arsenal play Chelsea and it predicts a win and a draw. A £10 bet would be placed on a win and a draw to cover both predictions. In addition, if the prediction for a win, draw and loss was false for a single game then no bets would be made as the model did not manage to come up with a sensible conclusion. Therefore it would be safer to not place a bet at all then to risk placing a bet on an outcome with low probability.

The betting strategy used only the predict value of the SVM which was either 1 or 0. This limited the betting strategy as most other betting strategies, such as M-E Kelly Betting, used the probability of the prediction to maximise profit. Initially, the predict_proba function was used to get the probability of the prediction for the SVM however after doing some checks it turned out that the probabilities from the predict_proba function was incorrect for the training data. Either the predict_proba function or the predict function could have been chosen. Ultimately, the predict function was chosen as it correctly predicts the training data.

Chapter 7

Experiments

An experiment was carried out to find which K value produced the most consistent model where K represented the number of games prior of k-form data that would be stored for each feature vector. The 4 values of K chosen was 2, 3, 4 and 5. The K value could not be too large otherwise the predictions would start later in the season as the first match day predicted would be K+1. In addition, K can not be too low or you will not have much training data as you start with K*10 training games. For example, if K=2 then you only have 10 training games for the first day predicted. The first match day for K=2 in the experiment was K+3 to ensure that there was sufficient training data for each SVM used.

To produce the optimal hyper parameters grid search cross-validation was used for python. It trains a model on all combinations of hyper parameter values and uses the model with the optimal accuracy on the test data. The C range for my experiment was 1,10,100 and 1000 and the γ range was 2^{-5} , 2^{-3} , 2^{-1} and 2^1 . Gamma and C were continuously modified to find good values but the grid search cross-validation usually kept picking the lowest value for both. Low values of C indicate that the hyperplane margin is large but there are a lot of points correctly classified but on the wrong side of the hyperplane. Therefore, the likelihood of poor generalisation and under-fitting is increased. Low values of gamma indicate that the model is too constrained and cannot capture the shape of the data (similar to figure 7.1 shown below).

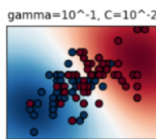


Figure 7.1: Hyperplane example
[8]

7.1 K=2

Matchday	SVMhome			SVMdraw			SVMaway			Total
	C	gamma	Accuracy	C	gamma	Accuracy	C	gamma	Accuracy	
5	1	0.125	30	1	0.5	90	1	0.03125	80	100
6	10	0.125	60	10	0.03125	70	1	0.03125	90	121.70
7	100	0.125	60	10	0.03125	90	1	0.03125	60	148.10
8	1	0.125	70	10	0.03125	70	1	0.03125	80	133.29
9	1	0.5	40	10	0.03125	80	1	0.03125	50	119.96
10	10	0.125	80	10	0.03125	80	1	0.03125	60	144.08
11	10	0.03125	50	10	0.125	90	1	0.03125	70	144.08
12	10	0.03125	80	10	0.03125	70	1	0.03125	60	144.08
13	10	0.03125	70	1	0.03125	70	1	0.03125	60	144.08
14	100	0.125	50	100	0.03125	90	1	0.03125	70	129.67
15	100	0.03125	60	100	0.03125	80	1	0.03125	70	129.67
16	10	0.125	80	1	0.03125	70	100	0.125	60	129.67
17	1	0.125	90	1	0.03125	70	100	0.125	70	129.67
18	10	0.125	50	1	0.03125	30	100	0.03125	90	155.86
19	10	0.125	60	1	0.03125	60	100	0.03125	50	140.28
20	10	0.125	70	1	0.03125	100	100	0.03125	70	168.47
21	10	0.125	40	1	0.03125	70	1	0.03125	80	168.47
22	10	0.125	50	1	0.03125	80	1	0.03125	70	201.33
23	10	0.125	70	10	0.125	50	100	0.03125	60	181.19
24	100	0.03125	80	1	0.03125	70	1	0.03125	70	181.19
25	10	0.125	60	1	0.03125	100	1	0.03125	50	181.19
26	10	0.03125	70	1	0.03125	70	10	0.125	70	181.19
27	10	0.03125	60	1	0.03125	80	1	0.03125	70	181.19
28	10	0.03125	50	1	0.03125	90	1	0.03125	90	181.19
29	100	0.03125	70	1	0.03125	80	1	0.03125	70	213.81
30	10	0.125	50	1	0.03125	80	1	0.03125	60	213.81
31	10	0.03125	40	1	0.03125	90	1	0.5	70	192.43
32	10	0.03125	70	1	0.03125	60	1	0.03125	90	192.43
33	1	0.125	70	1	0.03125	90	1	0.03125	60	192.43
34	10	0.03125	40	1	0.03125	60	1	0.03125	90	192.43
35	1	0.125	80	1	0.03125	70	1	0.03125	60	192.43
36	10	0.125	40	1	0.03125	70	10	0.03125	70	173.18
37	10	0.125	50	1	0.03125	60	10	0.03125	70	173.18
38	10	0.125	60	1	0.5	90	100	0.03125	70	173.18

Table 7.1: K = 2 was used to produce these results

7.2 K=3

Matchday	SVMhome			SVMdraw			SVMaway			Total
	C	gamma	Accuracy	C	gamma	Accuracy	C	gamma	Accuracy	
6	1	0.125	40	1	0.125	70	1	0.03125	90	100
7	10	0.03125	70	1	0.5	90	1	0.03125	60	121.80
8	10	0.125	60	1	0.5	70	1	0.03125	80	109.62
9	1	0.5	60	1	0.5	80	1	0.03125	50	109.62
10	10	0.03125	90	1	0.5	80	1	0.03125	60	131.65
11	10	0.03125	40	1	0.03125	90	1	0.03125	70	131.65
12	10	0.03125	70	1	0.03125	70	1	0.03125	60	131.65
13	10	0.03125	70	1	0.03125	70	1	0.03125	60	118.48
14	100	0.03125	60	100	0.03125	80	1	0.03125	70	106.63
15	10	0.125	50	1	0.03125	100	1	0.03125	70	106.63
16	1	0.125	90	1	0.03125	70	10	0.125	50	106.63
17	1	0.125	80	1	0.03125	70	1	0.03125	50	106.63
18	10	0.125	50	1	0.03125	30	100	0.03125	100	128.18
19	10	0.125	70	1	0.03125	60	100	0.03125	30	128.18
20	10	0.125	70	1	0.03125	100	100	0.03125	60	153.94
21	10	0.125	40	1	0.03125	70	1	0.03125	80	153.94
22	10	0.125	50	1	0.03125	80	1	0.03125	70	183.96
23	10	0.125	70	1	0.03125	60	10	2	80	183.96
24	10	0.03125	90	1	0.03125	70	100	0.03125	60	183.96
25	10	0.03125	50	1	0.03125	100	1	0.03125	50	183.96
26	10	0.125	80	1	0.03125	70	100	0.03125	80	183.96
27	10	0.03125	50	1	0.03125	80	100	0.03125	60	165.56
28	100	0.125	50	1	0.03125	90	100	0.03125	80	165.56
29	1	0.125	60	1	0.03125	80	1	0.03125	70	165.56
30	10	0.125	60	1	0.03125	80	1	0.5	60	165.56
31	1	0.125	30	1	0.03125	90	10	2	70	165.56
32	10	0.125	80	1	0.03125	60	1	0.03125	90	165.56
33	10	0.125	60	1	0.03125	90	1	0.03125	60	165.56
34	1	0.125	50	1	0.03125	60	10	0.125	90	202.15
35	10	0.125	80	1	0.03125	70	100	0.03125	70	202.15
36	10	0.125	50	1	0.03125	70	100	0.03125	70	181.94
37	10	0.125	60	1	0.03125	60	100	0.03125	70	181.94
38	10	0.125	90	1	0.5	90	100	0.03125	90	218.51

Table 7.2: K = 3 was used to produce these results

7.3 K=4

Matchday	SVMhome			SVMdraw			SVMaway			Total
	C	gamma	Accuracy	C	gamma	Accuracy	C	gamma	Accuracy	
7	100	0.03125	60	1	0.03125	90	1	0.03125	60	121.80
8	100	0.03125	50	1	0.03125	70	1	0.03125	80	109.62
9	1	0.03125	30	1	0.03125	80	1	0.03125	50	98.65
10	10	0.125	70	1	0.03125	80	1	0.03125	60	118.48
11	10	0.03125	50	1	0.03125	90	1	0.03125	70	118.48
12	10	0.03125	60	1	0.03125	70	1	0.03125	60	118.48
13	10	0.03125	50	1	0.03125	70	1	0.03125	60	106.63
14	10	0.125	50	1	0.03125	90	1	0.03125	70	106.63
15	100	0.03125	30	1	0.03125	100	100	0.03125	60	106.63
16	1	0.03125	90	1	0.03125	70	100	0.03125	80	147.16
17	1	0.125	90	1	0.03125	70	100	0.03125	70	147.16
18	10	0.125	60	1	0.03125	30	100	0.03125	90	176.88
19	10	0.125	80	1	0.03125	60	100	0.03125	50	176.88
20	10	0.03125	80	1	0.03125	100	100	0.03215	50	212.44
21	10	0.03125	40	1	0.03125	70	100	0.03125	70	255.57
22	10	0.03125	60	1	0.03125	80	1	0.03125	70	305.40
23	10	0.03125	80	1	0.03125	60	100	0.03125	70	305.40
24	10	0.125	70	1	0.03125	70	1	0.03125	70	305.40
25	10	0.03125	50	1	0.03125	100	10	0.03125	50	305.40
26	10	0.125	70	1	0.03125	70	10	0.03125	90	305.40
27	10	0.125	50	1	0.03125	80	10	0.03125	60	305.40
28	100	0.125	50	100	0.03125	90	10	0.125	90	365.87
29	10	0.03125	60	1	0.03125	80	10	0.125	80	431.73
30	1000	0.03125	60	1	0.03125	80	10	0.125	60	431.73
31	10	0.125	30	10	0.03125	90	10	0.125	80	431.73
32	10	0.03125	70	10	0.03125	60	10	0.125	90	431.73
33	10	0.03125	60	10	0.03125	90	10	0.03125	60	431.73
34	100	0.03125	30	10	0.03125	60	100	0.03125	70	431.73
35	100	0.03125	60	1	0.03125	70	1000	0.03125	70	431.73
36	10	0.03125	50	1	0.03125	70	100	0.03125	60	431.73
37	10	0.125	60	1	0.03125	60	1	0.03125	80	431.73
38	10	0.03125	80	1	0.03125	90	100	0.03125	90	518.51

Table 7.3: K = 4 was used to produce these results

7.4 K=5

Matchday	SVMhome			SVMdraw			SVMaway			Total
	C	gamma	Accuracy	C	gamma	Accuracy	C	gamma	Accuracy	
8	10	0.03125	40	1	0.03125	70	1	0.03125	80	90
9	100	0.03125	30	1	0.03125	80	1	0.03125	50	81
10	10	0.03125	70	1	0.03125	80	1	0.03125	60	81
11	100	0.03125	50	1	0.03125	90	1	0.03125	70	81
12	10	0.03125	70	1	0.03125	70	1	0.03125	60	81
13	100	0.03125	50	1	0.03125	70	1	0.03125	60	72.9
14	100	0.03125	50	1	0.03125	90	100	0.03125	80	72.9
15	10	0.125	30	1	0.03125	100	100	0.03125	60	72.9
16	1	0.125	90	1	0.03125	70	1	0.03125	60	72.9
17	10	0.03125	70	1	0.03125	70	1	0.03125	50	65.61
18	10	0.125	70	1	0.03125	30	100	0.03125	90	65.61
19	10	0.03125	80	1	0.03125	60	100	0.03125	60	65.61
20	10	0.03125	60	1	0.03125	100	100	0.03125	50	78.79
21	10	0.03125	50	1	0.03125	70	100	0.03125	80	94.79
22	10	0.03125	60	1	0.03125	80	100	0.03125	70	113.27
23	10	0.03125	80	1	0.03125	60	100	0.03125	80	113.27
24	10	0.03125	70	1	0.03125	70	10	0.03125	70	113.27
25	10	0.03125	50	1	0.03125	100	100	0.03125	60	148.39
26	10	0.03125	70	1	0.03125	70	100	0.03125	80	148.39
27	10	0.03125	50	1	0.03125	80	100	0.03125	70	148.39
28	10	0.03125	70	1	0.03125	90	100	0.03125	90	177.77
29	10	0.03125	60	1	0.03125	80	1	0.125	70	209.77
30	10	0.03125	50	1	0.03125	80	100	0.03125	60	209.77
31	10	0.125	30	1	0.03125	90	100	0.03125	70	209.77
32	100	0.125	60	1	0.03125	60	100	0.03125	90	209.77
33	10	0.125	50	1	0.03125	90	1	0.03125	60	209.77
34	100	0.03125	20	1	0.03125	60	10	0.125	90	209.77
35	100	0.03125	60	1	0.03125	70	10	0.125	60	209.77
36	10	0.03125	90	1	0.03125	70	1	0.03125	70	209.77
37	10	0.03125	70	1	0.5	60	1	0.03125	80	209.77
38	10	0.03125	70	1	0.03125	90	1	0.03125	80	209.77

Table 7.4: K = 5 was used to produce these results

7.5 Graphs

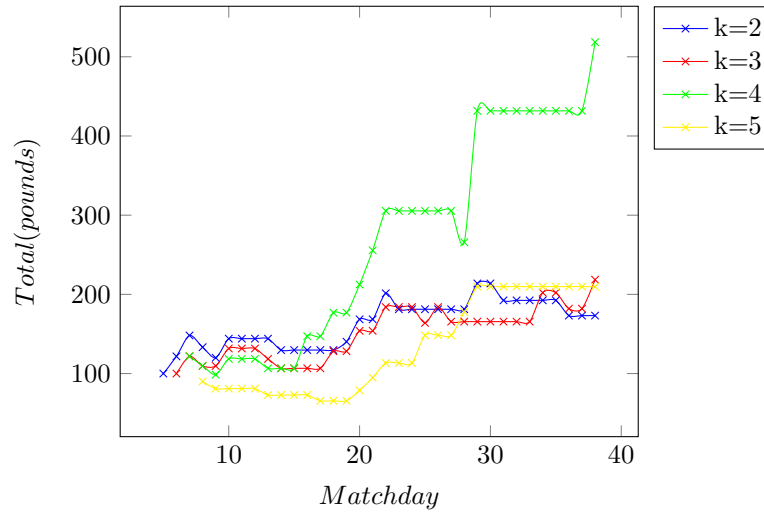


Figure 7.2: Budget for various K values

From the graph, it shows that the bets placed initially were not as accurate as the bets placed further on in the season. This could be due to the fact that there was less data at the lower match days. The first match day prediction could have been delayed to increase the probability of success however it still made profitable bets over the first 5 match days which suggest that it was accurate enough to be included. For $k=5$ all bets taken before match day 18 lost money. The cause of the loss of money could be caused by a lack of training data as the total started going up as training data increased.

A low risk betting strategy was used which can be seen from the graph as there are many match days where no money was invested at all such as match day 28 to 37 for $k=4$. In addition, every k value increased the initial budget from the betting strategy which shows that big risks do not have to be taken to profit by the end of the football season.

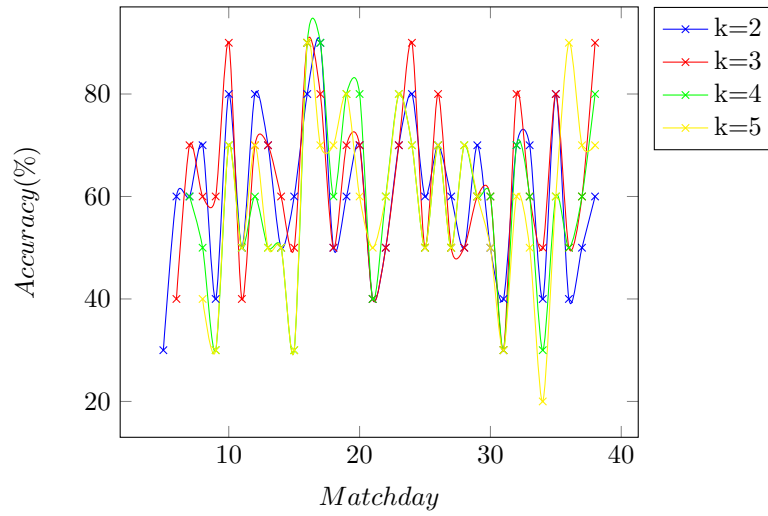


Figure 7.3: Accuracy over match days for SVM home

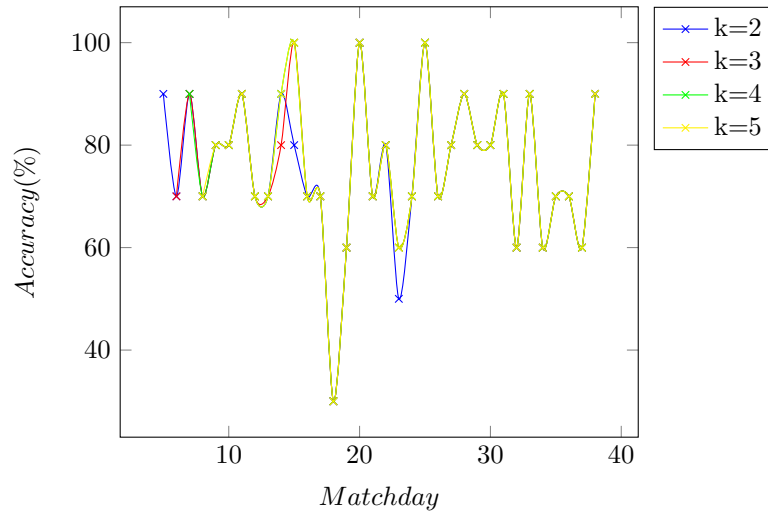


Figure 7.4: Accuracy over match days for SVM draw

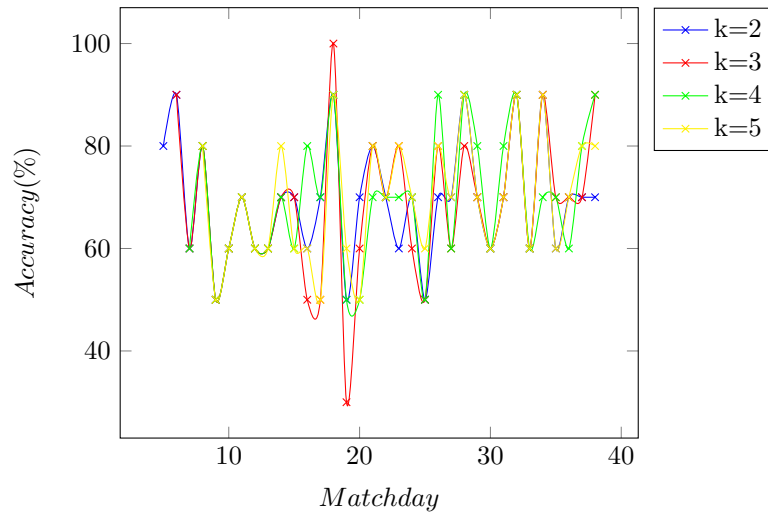


Figure 7.5: Accuracy over match days for SVM away win

The SVM for predicting draws had an accuracy of over 60% for the majority of the match days which dropped on match day 18 to just 30%. This could be due to the random nature of football where the better teams do not win depending on many factors such as a bad conversion rate by the key players in the losing teams. In addition, underfitting could have occurred in the model as the parameter values were always pretty lowest for both C and γ when compared to the SVM for home and away wins.

The SVM draw was much more accurate and stable then the two other SVMs. Surprisingly, the value of k did not make much of a difference for SVM draw with all games above 23 having the same accuracy for all values of k which indicates that under-fitting occurred. In addition, the low values of C show that the model might be under-fitting by choosing a flat hypothesis which resembles a linear SVM. It might also mean that the used kernel could be too expensive, especially given the fact that the dimensions of our problem is large.

The highest accuracy for SVM home was $k=3$, highest accuracy for SVM draw was $k=4$ and the highest accuracy for SVM away was $k=5$ (as shown below in figure 7.5). This suggests that depending on the type of result being predicted, different k values are required to get the optimal prediction accuracy from the SVM. However, $k=3$ had the greatest average accuracy of all 3 SVMs.

k	SVM home accuracy	SVM draw accuracy	SVM away accuracy	average
2	60.29	75.59	69.41	68.43
3	62.73*	75.76	68.48	68.99*
4	58.75	76.25*	69.38	68.13
5	58.71	75.81	69.68*	68.07

Table 7.5: Average of experiments

7.5.1 Testing

The code for data collection, modelling and betting strategy were all tested indirectly as it is extremely unlikely that profit would have made if there was any major mistake. Certain checks were made throughout the code. For example, the feature vector length was measured to ensure that they were the same size to prevent certain attributes from not being included. Once the code had to be modified when it was spotted that Sunderland played with no strikers against Arsenal. Therefore, Sunderland had no attacker stats for their players so average default values were given to minimise the factor from playing a key role in the modelling phase. If this consideration was not made then the values in the feature vector would represent different factors for the Arsenal vs Sunderland game which could have effected the position of

the hyperplane. Resulting in worse experiment results as the One Against All SVMs use every SVM to make a prediction so if the feature vector for the Arsenal vs Sunderland game was a support vector it would have likely changed the output greatly.

Chapter 8

Legal, Social, Ethical and Professional Issues

The Code of Conduct and Code of Good Practice produced by the British Computer Society have principles that should be followed by all members. The principles were followed where appropriate throughout the project. However, not many principles were relevant to this project due to the small scale.

Chapter 9

Conclusion and Future Work

This project has taught me that the data collection phase was very difficult because I could not include all the factors I wanted to consider. Initially, I wanted to include football fantasy data which is a game that lets you pick a squad from a budget and each player gets points for their performance after every week in the Premier League. The points for each player could be a good indication of their performance in their previous season which could have potentially helped the model to produce more accurate results. The data was unavailable online unfortunately so I could not use it during modelling. Some websites were selling the fantasy football data online which showcased how important people value data. This importance could be due to the prizes in fantasy football which are made up of holidays and football tickets. In future, data should be collected on football fantasy for the previous season and stored in a csv file therefore no money needs to be spent and it could have improved the accuracy of the model.

Making a model showed me that picking the hyper parameters can be troublesome which makes it harder to optimise the prediction accuracy of the Support Vector Machine. I used the grid search cross-validation from a python library to pick all combinations of parameters from C and γ to produce the model with the highest accuracy. The difficulty was that the C and γ range could not be too long as it would increase the computation time exponentially. Assuming the C and gamma range had the same number of elements k , the number of combinations tried for each SVM would be k^2 . In future an alternative method to trying all combinations would be to pick random elements from both the C and γ range and pick the local optimum. This method can use longer ranges in each respective hyper parameter and can be scaled up and down to cover more or less of each range.

Implementing the betting strategy was intriguing as it showed me how profit could be made

from a simple naive betting strategy where the only inputs were odds and predictions. In future, I could have used the distance between the hyperplane and the feature vector to find a probability for each outcome and used it in a new betting strategy. Similar to M-E Kelly betting strategy, the odds of each outcome could also be used to encourage bets with high odds. The predictions in this paper were only made possible due to the advancement in machine learning algorithms. This is extremely exciting as there is always new algorithms being produced and already developed algorithms being improved upon which can give us more accurate predictions in the future. An addition to this project could be to predict a future season in the Premier League to see how a betting strategy would perform and produce empirical results.

References

- [1] 2008–09 fc barcelona season.
- [2] Ballon d’or winners list.
- [3] Celtic record famous 2-1 win over barcelona in the uefa champions league.
- [4] Cs231n convolutional neural networks for visual recognition.
- [5] Distance between point and plane.
- [6] Irrational tossers.
- [7] The kelly criterion.
- [8] What are c and gamma with regards to a support vector machine?
- [9] Wrbf svm parameters.
- [10] Haversine formula, 2011.
- [11] Fifa index, 2012.
- [12] Anomaly detection with the poisson distribution, 2015.
- [13] Sports analytics market by type (solutions and services), by applications (player analysis, team performance analysis, health assessment, video analysis, data interpretation and analysis, fan engagement), by deployment type and by region - global forecast to 2021, 2016.
- [14] 10 most-watched sport events in the history of television, 2018.
- [15] Data files: England, 2018.
- [16] Fifa/coca-cola world ranking, 2018.

- [17] Haversine picture, 2018.
- [18] Matched betting 2018 – how much can you make from matched betting?, 2018.
- [19] Poisson distribution, 2018.
- [20] Raian Ali, Emily Arden-Close, John McAlaney, and Keith Phalp. World cup online betting is the highest it’s ever been, 2018.
- [21] Ben Carter. How good are lawro’s predictions?, 2013.
- [22] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.
- [23] Benjamin Cronin. Poisson distribution: Predict the score in soccer betting, 2017.
- [24] Victor Haghani and Richard Dewey. Rational decision-making under uncertainty: Observed betting patterns on a biased coin. *Available at SSRN 2856963*, 2016.
- [25] Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. An empirical exploration of recurrent network architectures. In *International Conference on Machine Learning*, pages 2342–2350, 2015.
- [26] MultiMedia LLC. MS Windows NT kernel description, 1999.
- [27] Michael Joseph Moroney. *Facts from figures*. Penguin books, 1962.
- [28] Richard Pollard. Home advantage in soccer: A retrospective analysis. *Journal of sports sciences*, 4(3):237–248, 1986.
- [29] D. Prasetyo and D. Harlili. Predicting football match results with logistic regression. In *2016 International Conference On Advanced Informatics: Concepts, Theory And Application (ICAICTA)*, pages 1–5, Aug 2016.
- [30] Sherif Saed. Ea explains how fifa player ratings are calculated, 2016.
- [31] David Sheehan. Predicting football results with statistical modelling, 2018.
- [32] Jeffrey Alan Logan Snyder. What actually wins soccer matches: Prediction of the 2011-2012 premier league for fun and profit. *University of Washington*, 2013.
- [33] Martin Spann and Bernd Skiera. Sports forecasting: a comparison of the forecast accuracy of prediction markets, betting odds and tipsters. *Journal of Forecasting*, 28(1):55–72, 2009.

- [34] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.
- [35] Georgina Turner. The most prolific minute in premier league history?, 2013.