Submitted in part fulfilment of the requirements for the degree
of
Master of Science in Business Analytics

League of Legends Victory Prediction Analysis

By

Calum Palmer

Surrey Business School

Faculty of Arts and Social Sciences

University of Surrey

September 2022

Word Count: 15,000

© Calum Palmer, 2022

Executive Summary

(1000 words; 10% of marks)

Set out on its own immediately after the title page. This often takes the form of a series of summary statements, ordered under similar headings to those used within the Dissertation. These summarise the key information or findings. The Executive summary should be written for an intelligent layman. An example of an Executive summary can be found in SurreyLearn.

Declaration of Originality

I hereby declare that this thesis has been composed by myself and has not been presented or accepted in any previous application for a degree. The work, of which this is a record, has been carried out by myself unless otherwise stated and where the work is mine, it reflects personal views and values. All quotations have been distinguished by quotation marks and all sources of information have been acknowledged by means of references including those of the Internet. ***I agree that the University has the right to submit my work to the plagiarism detection sources for originality checks.***

Name: Calum Palmer

Signature:

Date: 07/09/2022

Contents

Executive Summary **ii**

Declaration of Originality **iii**

Table of Contents **iv**

List of Figures **v**

List of Tables **vi**

Acronyms **viii**

Glossary **ix**

1 Introduction **1**

1.1 Context 1

1.2 League of Legends 3

1.2.1 General Information 3

1.2.2 Champion Selection 4

1.3 Structure, Aims, and Objectives 6

2 Literature Review **7**

2.1 Predictive Analytics in Traditional Sports 7

2.2 Predictive Analytics in Esports 8

2.2.1 Match Outcome Prediction 8

2.2.2 Champion Selection Prediction 10

2.3 Esports Betting 12

2.4 Summary 13

3 Methodology **15**

3.1 Data Pre-processing 15

3.1.1 Data Overview 15

3.1.2 Data Preparation 17

3.2 Data Modelling 20

3.3 Summary 23

4 Results **24**

4.1 Performance Metrics 24

4.2 Model Results 27

4.2.1 10 Minute Model 27

4.2.2 15 Minute Model 28

4.2.3 20 Minute Model 29

5 Conclusion **31**

5.1 Summary of Works 32

5.2 Research Limitations 32

5.3 Further Research 32

A Python Code **38**

B Data Description **39**

C Feature Correlations **40**

D Idk **41**

List of Figures

1.1 Map of League of Legends 3
1.2 A depiction of the Champion Selection phase 5

2.1 A graph showing the steps of a Monte-Carlo Tree Search . . . 11

3.1 A graph showing the target balance of the dataset 20
3.2 A diagram showing how the Random Forest algorithm classifies 21

4.1 A confusion matrix of a Logistic Regression Model trained on the 10 minute dataset 25
4.2 A graph showing the imp 10 minute dataset 29

C.1 A matrix of correlations between features in the dataset 40

List of Tables

3.1 An excerpt from the dataset. 16

3.2 A table describing the models that were built 23

4.1 A table showing the model results for the 10 minute dataset . 27

4.2 A table showing the mean metrics after 10-fold cross-validation 28

4.3 A table showing the model results for the 15 minute dataset . 30

4.4 A table showing the model results for the 20 minute dataset . 30

Acronyms

AUC Area Under the ROC curve

FN False Negative

FP False Positive

FPR False Positive Rate

MCTS Monte Carlo Tree Search

MOBA Multiplayer Online Battle Arena

NLP Natural Language Processing

RNN Recurrent Neural Networks

ROC Receiver Operating Characteristic

TN True Negative

TP True Positive

Glossary

baron A neutral monster that spawns after the 20-minute mark that will give a powerful buff when slain. 4

champion A unique player-controlled character possessing a distinct set of abilities and attributes. 3, 5

dragon A neutral monster that spawns every 5 minutes that will give a moderate team-wide buff when slain. 4

gank When a surprise attack is made upon a player, often made by a jungler or support. 4

jungle A section of the map where neutral monsters spawn that can be slain for gold, experience and buffs. 3, 4

meta The most effective strategy for winning. 2

minion A unit that periodically spawns from the Nexus, advances along a lane towards the enemy Nexus and engages with any enemy they encounter. 3

nexus A structure that serves as the primary objective of the game. When the enemy Nexus is destroyed, victory is achieved. 4

patch A version of the game with a set of changes made to the game to update, improve or balance. 5, 6

rift herald A neutral monster that spawns between the 8 and 20-minute mark that can be used as a powerful tower sieging tool when slain. 4

tower A structure that deals damage to enemies that come into its radius and must be destroyed in order to reach the Nexus. 4

ward A deployable unit that grants vision of the surrounding area for a duration, they are typically used to gain valuable information on the enemy. 3, 4

Chapter 1

Introduction

1.1 Context

Esports is a form of competition using video games where participants will compete either individually or in a team for a chance at victory. These competitions attract millions of viewers, with estimates of 532 million spectators by the end of 2022, and this value is expected to grow annually at a value of roughly 8.7% (Newzoo 2022). The rapid growth in Esports has led to the industry becoming professional, with hundreds of players contracted on full-time contracts competing for prize pools of up to $40 million (Esports Earnings n.d.). According to Newzoo (2022) this viewership will help the industry generate over $1.38 billion in revenue by the end of 2022. As the Esports industry continues to grow, so does the importance on teams to win and remain relevant in the industry.

In traditional sports, analytics has become an extremely popular field with teams investing heavily in some form of analytics. These analytics can be used from evaluating opposing teams, to individual player forecasting and even used to decide signings or team selection (Sarlis & Tjortjis 2020, Apostolou & Tjortjis 2019). Apostolou & Tjortjis (2019), Sarlis & Tjortjis (2020) shows that these analytics can be applied for each athlete, giving an accurate estimation of key metrics such as goals scored per season or the number of shots attempted in a given match. This same methodology could be applied to Esports, using these machine learning techniques could highlight specific factors both pre-game and in-game, helping analysts and coaches refine strategies within the game.

The ease of data collection coming from each match has led to a rise in Esports analytics. In-depth analysis of matches, teams and pre-game factors become key techniques for teams to gain this advantage over their competitors, with teams being required by their leagues to have at least one dedicated coach and analyst similar to traditional sports teams (LoLEsports 2022). These coaches and analysts use predictive analytics to maximise their team's likelihood of winning by altering numerous features related to pre-game and in-game strategies, current meta analysis and common patterns of their competition (Kokkinakis et al. 2021). However, this analysis is often completed manually by watching key highlights of matches using the analyst's intuition and using rudimentary analysis of in-game factors.

If matches can be accurately predicted using machine learning techniques, then analysts can provide new opportunities to optimise player strategies and can lead their teams to better outcomes. Applying the same findings found in Gray & Wert-Gray (2012), it can be seen that the overall performance and fan satisfaction with a sports team's performance has a measurable impact on revenue via fan attendance and their media response. Esports fans also appear to increasingly demand skillful performances especially from players that are deemed as *'superstars'*, with these players being more likely to attract new viewers, thus increasing the economic gain of the market (Mangeloja 2019, Ward & Harmon 2019). It would then be in the interest of both teams and individual players to maximise their abilities and career longevity using these advanced analytics, so they can fully realise their potential; especially when the volatility of a players job security results in only the top 10% of players having lasting, stable careers (Ward & Harmon 2019).

1.2 League of Legends

1.2.1 General Information

League of Legends is a Multiplayer Online Battle Arena (MOBA) game developed by Riot Games released in 2009, it is one of most popular esports games in the world with over 180 million monthly players and a peak of 73.8 million concurrent viewers (Riot Games 2021, McLaughlin 2021). A MOBA is fusion genre of real-time strategy, role-playing and action games in which two sets of teams will compete in a known arena. The objective of each game is to defeat the opposition by destroying the enemy's base. Each player will select and control a unique champion with their own set of distinct abilities, this champion will be selected before the game starts and cannot be changed until the game has ended - this will be covered further in Section 1.2.2. Players can strengthen their champions by gaining experience and gold, this can be done by slaying enemy minions, jungle monsters, enemy structures or enemy champions. This gold can be spent in the shop allowing players to purchase items that enhance the attributes of their champion, as well as various utility items such as wards.

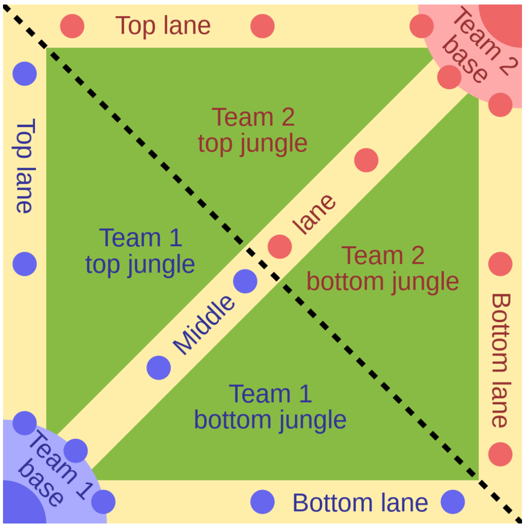


Figure 1.1: Map of League of Legends

A map of League of Legends can be seen in Figure 1.1. There are three lanes, Top, Middle and Bottom, with the jungle filling the space between these lanes. Typically, a player will be assigned to each of these lanes including the jungle, the exception being two players assigned to the bottom lane. The roles are typically labeled as follows:

- Top laner - Starts in the top-lane, often a champion who has larger health and resistance to damage with the ability disrupt enemy players.
- Jungler - Roams in the jungle, they help their laners whenever possible often using surprise attacks on enemy laners commonly referred to as a gank.
- Mid laner - Starts in the mid-lane, often a champion who is a spell-caster that can cause magic damage over a wide area or has high single-target burst damage.
- Attack Damage Carry (ADC) - Starts in the bottom-lane, often a long-ranged champion that requires gold in order to deal massive damage towards the latter stages of the game.
- Support - Also starts in the bottom-lane, often a champion that can provide aid to the ADC throughout the game via protective abilities and utility such as wards.

Each coloured dot represents a tower that must be taken in order to reach the enemy Nexus. A river separates the territories between the Blue (Team 1) and the Red team (Team 2) along the dotted black line seen in Figure 1.1. In this river you can find Baron or Rift herald in top-side and the Dragon in the bottom-side, they are key objectives that will often be contested.

1.2.2 Champion Selection

Champion Selection plays an important part in every game of League of Legends. Certain champions have inherent synergies with one another, meaning they are beneficial to be picked with each other. Likewise, some champions are considered counter matchups when they are good at stopping another champion. This means that picking a good mixture of champions that are solid synergistically, whilst also ensuring the opponents champions do not counter yours is vital. These ideas are the fundamentals of champion selection, and they are what professional coaches and analysts attempt to solve each week. Factors such as player champion experience, the current game balance patch or a champion's ability to be flexible across different lanes will change champion select from game to game.

As seen in Figure 1.2, the current draft phase works as follows:

- Ban Phase 1 begins with the Blue team, in turn each team bans three champions from the pool.
- Pick Phase 1 begins with a singular pick from the Blue side, followed by two picks from the Red side. Blue side will get two more picks, followed by a singular pick from Red side for three picks each.
- It will then enter Ban Phase 2. Here both teams will ban two more champions in turn, with Red side starting.
- Pick Phase 2 will begin. Here Red side get their fourth champion pick, followed by the final two picks from Blue side and finally Red side pick their final champion.

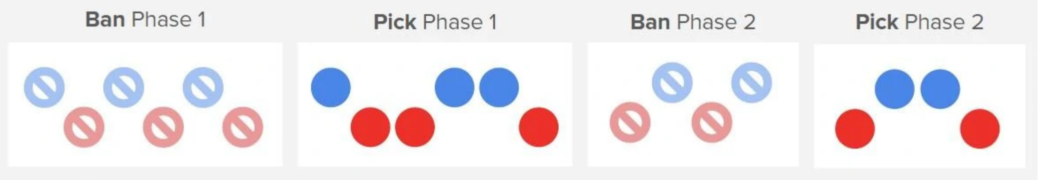


Figure 1.2: A depiction of the Champion Selection phase

This champion selection structure leads to clear opportunities for teams to ban out champions that are deemed too strong in Ban Phase 1. Blue side getting the first pick gives them a chance to pick any champion that is deemed too strong that still remains after Ban Phase 1. Whilst Red side getting the last pick gives a defined opportunity to pick a counter match-up to any given lane. These factors can give one side the edge based on the Patch, leading to varying strength levels of Blue side vs Red side and can make side selection important.

1.3 Structure, Aims, and Objectives

The following section explores an overview of this dissertation:

The next chapter, Chapter 2, discusses the role of...

Chapter 3 follows the techniques and tools used for predicting the effect of pre-game choices on the outcome of the match. Beginning with...

Chapter 4 then proceeds with a discussion of the work carried out and presents the outputs of the model created. An evaluation of...

Finally, chapter 5 will conclude the dissertation giving an overall summary of the work completed, as well as any further opportunities for research.

Having pre-established the landscape of Esports and its relationship with analytics, it is clear that refinement in the way that this industry uses its highly available data is needed. Many academics have predicted the outcomes of matches in Esports titles such as Silva et al. (2018). However, performing these studies, few academics addressed the implications of the champion select phase on the overall outcome on a given League of Legends match. Often it is put into the model as a singular feature defined as champion or ban, without giving much implication on how an individual champion effects a game more than another. In contrast to other studies, this study uses a much larger, updated dataset and will concentrate much more on the overall effects of the pre-game choices that a team will make inside champion select. This includes the effects of each individual champion on the likelihood of winning a match. Therefore, the research question that will be addressed is as follows:

Can the outcome of a League of Legends match be predicted?

Chapter 2

Literature Review

This section aims to review previous literature of prediction models in the esports industry that was explored during in Chapter 1. These literatures will help create a better working understanding of how to properly model, extract and gain knowledge from the datasets. It will cover papers from traditional sports where prediction models are widespread and lesser researched esports models, to research on the betting industry with its new relation to esports.

2.1 Predictive Analytics in Traditional Sports

In the mid to late 2000s the use of prediction modelling exploded academically, being applied to a multitude of fields from biology and medicine to political sciences or sports. This academic surge of interest caused thousands of studies all using various forms of predictive modelling. Not only did this evolve many practises across these fields, but it also helped develop the techniques of predictive modelling today. This is section we will explore literature related to how predictive analytics are used in sports, and how insights can be gained from this type of modelling. Sarlis & Tjortjis (2020) studied the impact of how various performance evaluation metrics can be used to identify dominant attributes that will help predict candidates for the Most Valuable Player (MVP) award, as well as Defender of the Year. They concluded that using their Propagation Neural Networks with 8 years of training data, that their model was able to accurately predict the MVP of the season with 100% accuracy, and could also predict the Defender of the Year. Similar studies have been performed with football such as Pantzalis & Tjortjis (2020), where they analysed 4 top football leagues in Europe and predicted the final standings with up to 70% accuracy. Scelles et al. (2021) shows us that transferring this level of analysis from a traditional sports team to an esports team is highly supported, with claims that 'esports... could provide some insights about the future development of sport'. This suggests that prediction modelling is likely to work when used to predict both player performances, whilst potentially being able to predict standing of esports leagues globally.

2.2 Predictive Analytics in Esports

2.2.1 Match Outcome Prediction

tThe work of predictive modelling in esports only started in the mid 2010s, with one of the original papers by Lin (2016) exploring match outcomes in League of Legends. This work focused on the relationship between the potential feature-sets that a game such as League of Legends can create, and how they impact the predicted match outcome both pre-game and in-game. The data was collected using the Riot API from matches with average ranked players, with the in-game data being extracted from the statistics published at the end of a match. It becomes apparent that the data used in this initial study appears to be highly correlated with the match result, and this is reflected by the 95% success rate on prediction using in-game data.

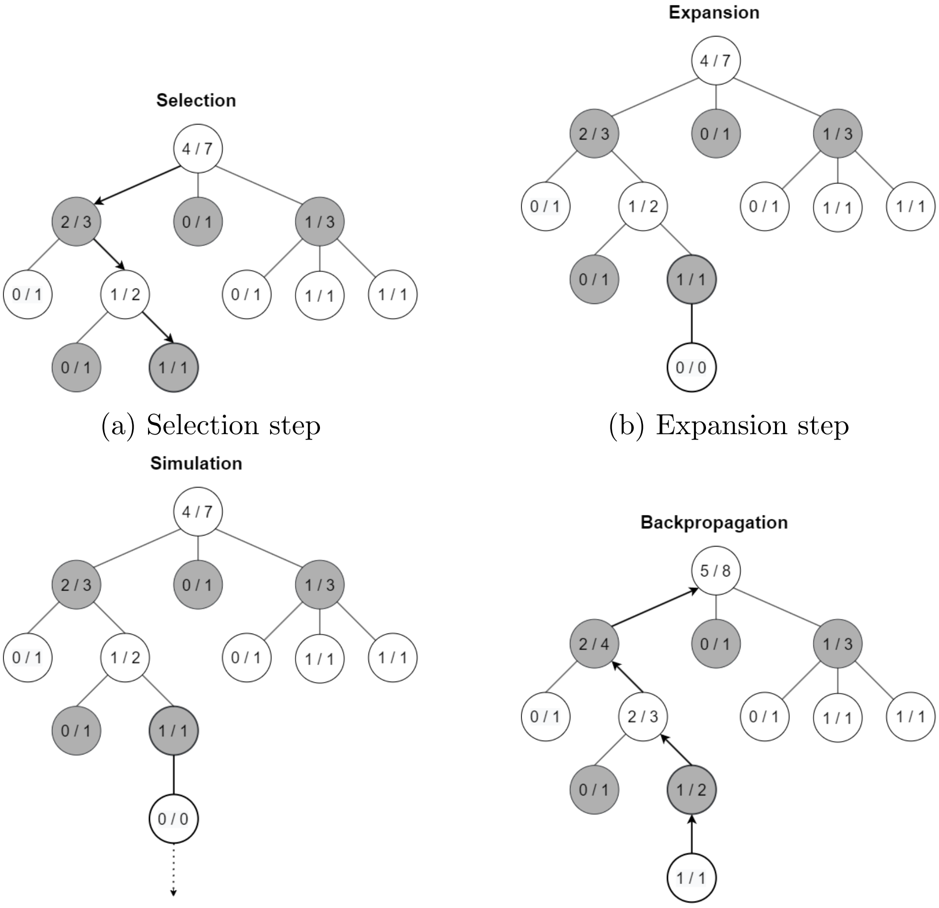
Two studies built upon the previous research just two years later. Silva et al. (2018) introduces the use of Recurrent Neural Networks (RNN) to predict esports matches through time intervals between the 0 and 25 minute mark, and uses data from matches between 2015 to early 2018. They concluded that these RNNs were capable of obtaining prediction accuracy of between 63.91% to 83.54%, depending on the time interval in-game. Their study presented evidence of the predictability of each match and its positive correlation with the in-game timer, showing that the accuracy of prediction becomes stronger as matches progress, reflecting the snowballing nature of the game itself. However, it should be noted that the study ignores the presentation of each feature's weighting and thus does not provide any insight into what features are most influential in these victories.

Gaina & Nordmoen (2018) built upon the study from Lin (2016), analysing the features that predict match outcomes at the 10-minute interval. Moreover, the correlation between the impact of early performance on the match result for each individual player in all 5 roles was found to be ‘a medium correlation‘, with ‘a weaker one‘ when overall team performance is compared. Other scholars such as Ani et al. (2019) have presented findings that suggest that prediction model accuracy can reach performance levels of 99.75%, with a combination of pre-game and in-game predictors using the Random Forest algorithm. An interesting note about Ani et al. (2019) is whilst using Adaboost, Gradient Boosting and Extreme Gradient Boosting they were only able to achieve an accuracy of 57.22% to 65.67% using only pre-game features which is drastically lower than the Random Forest algorithm. In the literature, both the Random Forest algorithm and Gradient Boosted Trees are commonly used machine-learning strategies that are used inside prediction models. These methods are both decision tree based in which a target variable is predicted based on a number of decision rules inferred from the feature-set, with the main downside is potential overfitting of the data (scikit-learn n.d.). This level of accuracy from pre-game features is more in line with other studies and will be discussed further in Section 2.2.2.

Lee et al. (2020) also produced a match outcome prediction model, using data from the Riot API of high level players from the ranked ladder. Similar to their predecessors of Silva et al. (2018), they tested their dataset across game time intervals, as well as the importance of each feature in the dataset. They achieved a prediction accuracy of between 62.25% and 96.08%, which is an improvement over the RNN used in previous studies. Gold difference and the number of Towers taken appear to consistently be the two most important features across most studies referenced, with gold difference having a relative feature importance calculated up to 43.08% at the 10-minute mark and turrets having up to a 22% importance (Lee et al. 2020, Ani et al. 2019, Gaina & Nordmoen 2018). Another interesting study comes from Novak et al. (2020), here they apply a coach-centred approach to modelling performances seen at the 2018 League of Legends World Championship. Three coaches rated the proposed feature set using a correlation scale of 1 to 10, with median ratings for features equal or over 6 being retained for modelling. After multicollinearity checks, only 14 predicting features remained and the strongest fixed effect was 'Tower Percentage', closely followed by 'Inhibitors Taken' and allowed the model to achieve prediction accuracy of 95.8% - which remains consistent with previous studies.

2.2.2 Champion Selection Prediction

As described earlier in Section 1.2.2, champion selection is a key part of any MOBA game and has the ability to give your team an inherent advantage in-game before the game even begins. Previous works have explored the ideas of character recommendation systems with two key methodologies in mind - models based upon historical win rates of each character and an association rule model based on common selection frequencies. Chen et al. (2018) proposed a Monte Carlo Tree Search (MCTS) recommendation model in another popular MOBA game - DOTA 2, that suggests characters to add to the team in the draft phase that will maximise the team's victory probability. The draft phase is approximated to a combinatorial, sequential, zero-sum game with perfect information and deterministic rewards. Therefore, an optimal pick at each stage will be the highest predicted win-rate character at a given stage, when both teams behave optimally in the process. Due to the large branching factor of the draft process, the decision-making scales exponentially larger as the draft process progresses and the computational power required does too. The MCTS is a tree search algorithm that is commonly used to solve deterministic games such as Chess, Tic Tac Toe and Go (Duck-



(a) Selection step

(b) Expansion step

(c) Simulation step

(d) Backpropagation step

Figure 2.1: A graph showing the steps of a Monte-Carlo Tree Search

ett 2016). As seen in Figure 2.1, this algorithm explores the choice tree from root to leaf, selecting child nodes that represent the best winning outcomes.

If the leaf node does not terminate the game, it will create the next step with more child nodes throughout the tree, and if this simulated child node gives optimistic result probabilities it will update and back-propagate up the tree towards the root. At its core, the MCTS chooses the optimal move from the current state of a game's tree with the help of reinforcement learning. Using this system, Chen et al. (2018) were able to achieve a win-rate of up to 88.0% versus a standard association rule draft recommendation system used in studies such as Hanke & Chaimowicz (2017).

Recently, Shen et al. (2022) initiated studies into the champion recommendation using a mix of Natural Language Processing (NLP) techniques and Bidirectional long-short term memory models. They found that their proposed recommendation mechanisms were effective in both the coverage of champions and user satisfaction Whilst this paper reiterated the fact that the champion selection phase can be predicted, it gives a unique perspective of adding a level of subjectivity to model by using user feedback and offers a wider-coverage of champions who are deemed less popular picks. But this lack of objective measure gives it no quantifiable metric onto which this study can extract valuable information from.

2.3 Esports Betting

As esports has grown, so has the ability of bookmakers to capitalise on a new developing market and create revenue. According to Absolute Reports (2022), the global esports betting market has been estimated to be worth up to $10 Billion in 2021, and is forecasted to double by the year 2028 with a compound annual growth rate of 13.1%. A new focus of esports has grown throughout traditional bookmakers, with an increasingly larger number of esports titles to bet on and some bookmakers even sponsoring some events (Byrne 2019). With the large variety of titles comes the challenge of creating odds that are representative of the outcomes that will occur. There are numerous studies showing how gambling markets in most traditional sports can change how a given team is evaluated, often with regard to sentiment bias (Feddersen et al. 2018, Na et al. 2019). This means that odds-makers must create predictive models that can accurately replicate the true likelihood of match outcome in order to ensure money is continually being made. According to the efficient-market hypothesis, sport bets should be subject to all available information that may be publicly available and this information will be reflected in the odds themselves(Even & Noble 1992). The betting market is therefore thought of as a fair and efficient market in which match outcomes can be accurately predicted.

Betting within esports has taken on many forms. Money line bets and proposition bets are the most common types of bets. Money line bets are bets that are placed on the outcome of a specific match, with payouts based on the odds that are created by the odds-makers using their internal prediction models. Proposition bets are bets made based on whether a specific event will take place within the game itself while the match is ongoing. A common proposition bet is whether a team would achieve the first kill of the game - commonly referred to as First Blood. All these types of bets require highly calculated odds ensuring the bookmakers will make money. With esports betting being in its early form, there are no regulatory structures put in place to effectively to keep match integrity in place similar to those found within most sports (Dos Reis 2017). This lack of match integrity causes potential match-fixing scandals which threatens the competitive integrity of both the league and the gambling market, as well as ruining games for spectators alike. Whilst studies such as Abarbanel & Johnson (2019) claim that esports spectators aren't deeply concerned about potential match-fixing, with most spectators being willing to forgive infractions that have occurred previously. Cases of match-fixing have already been investigated, with a Chinese player- 'Bo' being suspended after being coerced into match-fixing in the Chinese academy leagues, subsequently causing a league-wide large-scale investigation (Dot Esports 2021). This caused a call for harsher punishments and stricter measures to ensure infractions like these would become highly disincentivised, however no regulatory structures apart from the league's punishment systems currently exist.

2.4 Summary

This chapter provided an overview of key literature in predictive analytics and its relation to both sports and esports. Starting with a background on predictive analytics in sports and the various uses scholars have modelled with. This was followed with studies showing how these predictive analytics can be applied to various esport titles. The various methodologies that have been used previously were explained such as the Monte-Carlo Tree Search and the Random Forest classifier. The betting industry was then explored, showing its relation to the world of esports and how they use prediction modelling for their business. With the information gained throughout this literature review, the following research hypotheses have been created:

> *Hypothesis 1: League of Legends matches are snowballing in nature, therefore prediction accuracy will increase as in-game time increases.*
>
> *Hypothesis 2: League of Legends prediction models will naturally suffer from some level of multicollinearity.*
>
> *Hypothesis 3: A decision tree algorithm will likely be the optimal machine learning technique for prediction.*
>
> *Hypothesis 4: Esports betting indicates that highly accurate prediction models for match outcomes are feasible outside of academia and provide value to a business.*

Chapter 3

Methodology

In this chapter, the methodology that is used to research the topic is presented. This study attempts to produce a predictive analytics model that uses a machine learning approach that can predict the match outcomes of a League of Legends match. Firstly, an explanation of the pre-processing data-pipeline - how the data was collected, a description of the dataset, and how the data is prepared and cleansed for modelling. The subsequent section then follows the dataset through the prediction methodology, and ends with how the predictive capability of the model is tested and assessed.

3.1 Data Pre-processing

3.1.1 Data Overview

The data used in this project is based upon the match data collected from all the League of Legends esports leagues found globally in 2021. Oracle's Elixir is a website that collects all the esports match data provided by Riot Games; the publisher of League of Legends, and aggregates them into datasets that are freely downloadable and offered to coaches, analysts and fans alike (Oracle's Elixir n.d.). These datasets are updated daily, and go back until 2014. When the dataset is downloaded, it is given in a .csv format that can be opened Microsoft Excel to get an easier perspective of the data. Once opened, it can be seen that it is a huge dataset contained within one spreadsheet with

149,496 rows and 123 columns, and will have to be prepared in order to be ready for modelling. The rows contain data from a unique player within a given match, this starts off with the Blue side Top Lane player and continues serially until the Red side Support player is reached, it will then contain a row for each team's collective data. This means there are 12 rows for each unique match before reaching data for the next match, this can be seen in Table 3.1.

Table 3.1: An excerpt from the dataset.

Participant	Side	Position	TeamName
1	Blue	top	DWG KIA
2	Blue	jng	DWG KIA
3	Blue	mid	DWG KIA
4	Blue	bot	DWG KIA
5	Blue	sup	DWG KIA
6	Red	top	Nongshim RedForce
7	Red	jng	Nongshim RedForce
8	Red	mid	Nongshim RedForce
9	Red	bot	Nongshim RedForce
10	Red	sup	Nongshim RedForce
100	Blue	team	DWG KIA
200	Red	team	Nongshim RedForce

The target variable is the intended prediction variable, and is found at the eighteenth column named 'Result'. It is a binary value and simply denotes whether a team wins or loses using 1 and 0 respectively. Preceding this column are the match descriptor variables, here lies the unique match id, the league in which the game is played, the date, the game number for best of 5 series, and other information such as match length and those found in Table 3.1. One of the key columns that will be used is the 'datacompleteness' variable, this contains 4 options of 'complete', 'found', 'partial' and 'reparse', and describes the state of the data in a given row. Following these 18 match descriptor variables, are the 105 in-game statistic variables that have been recorded. The values in these columns will be the key predictors in our match-outcome prediction model, however the usefulness of each variable will be decided later in Section 3.1.2. These predictors range from simple statistics such as the total number of kills or deaths in a given match, to more advanced statistics such as GSPD - The average gold spent difference between teams. Many of these statistics also come in the form such as ‘goldat10’ and ‘goldat15’, which is simply the amount of gold obtained by the 10 and 15-minute mark in-game respectively. These time-based statistics will be important in factoring how much the match outcome varies by the in-game time. Another note about the data, is the fact that it tracks the opponent statistics for all of these statistics inside each row, so this should allow the use of data from only one side whilst maintaining all the statistics from both teams.

3.1.2 Data Preparation

It is vital for the data gathered to be both reliable and of high validity, this is ensured by a rigorous data preparation stage. In order for the data to be prepared for modelling, coding is required. The coding language used in this study is Python, it is a well-rounded language that is opensource and allows access to great analytical libraries, visualisations and is well documented online whilst being the main analytics language used in industry (TIOBE n.d.). Jupyter Notebooks are used as a personal preference, but any IDE could be used.

The CSV file of all the match data for 2021 is read into the notebook as a dataframe. The data-types of the dataframe are checked to ensure that they are as intended. Once again the data is inspected and a large proportion of missing data becomes visible, many features such as ‘turretplates’ and ‘elementaldrakes’ had well over 100,000 null values and were not fully complete. Features like these will likely be removed during feature selection due to their lack of data, otherwise it would ultimately harm the modelling process - the feature selection process will be covered later. Firstly, the dataframe is sorted for the feature ‘datacompleteness’ and opts to reject any row that is not ‘complete’, this massively helps reduce the number of null values found in the data. Additionally, another filter is used on the ‘position’ feature, reducing our data to only rows with team information instead of a mixture of individual and team data. This will help focus on the overall impact of the team’s actions in response to an outcome of a match. Now the dataframe should only consist of team-base data, but will contain rows for both the blue-side and the red-side teams for each individual ‘gameid’. The problem here is that only one side of data is required to predict with, here it is chosen to predict the outcome of a blue-side team using blue-side match data. This issue is easily subverted by dropping any duplicate ‘gameid’ rows, leaving the dataframe with only team-based data from the blue-side team. As the dataset is now made-up of only team-based data, no duplicate rows exist and contains no null values, descriptive statistics are used to further inspect the data to search for potential outliers that could corrupt our model. This is completed using the ‘describe’ function and manually searching for data minimums and maximums that look suspicious. It can be seen that the ‘gamelength’ feature has an unusually high maximum value when compared to both the median value and the 75th percentile value. This is then removed by removing any rows that exist above the 99th percentile for ‘gamelength’, reducing the maximum gamelength from over 40,700 minutes, down to a maximum of 47.2 minutes; a gamelength that is as expected. After these steps are complete, the dataset has been properly cleansed and now 11,145 rows remain.

Then comes the step of feature selection that aims to reduce the overall dimensionality of the dataset, increase the computational cost of modelling and to improve the performance of the model (Brownlee 2019). All the features that include opposition team data are removed as they will cause issues with multicollinearity later, as well as prediction from opposition statistics being deemed redundant for understanding how a given team could improve. After removing many features, a correlation matrix is created between the remaining features in the data subset. Finding the correlation allows for the measure of how features share an interdependence between one another. Here the Kendall correlation is calculated between features using the ‘corr()’

function and visualised using the ‘heatmap()’ function from the ‘Seaborn’ package. Correlation values between the target and themselves are deemed better predictors when this value moves closer to 1 or -1. Therefore, those features with values close to zero should be removed from the dataset as they likely provide no benefit to the model, whilst adding further complexity. On the contrary, features with high correlation values with other predicting features should be removed or reworked due to the issue of multicollinearity (Alin 2010).

The correlation matrix can be found at Appendix C.1. Here it can be observed that the correlations with our target - ‘result’, range anywhere between an absolute minimum value of 0.17 from the ‘firstdragon’ and ‘assistsat10’ features, to 0.66 from the ‘firstbaron’ variable. These correlations suggest that most features have a weak-moderate, with a few stronger predictors such as ‘firstbaron’ and ‘firstthreetowers’. When the correlation between features is examined, features containing assists and kills at the 10 and 15 minute mark are highly positively correlated with a values of 0.80 and 0.78. When correlations are high between features, these features should be either be ignored or transformed. Here it is chosen to transform these features, as they both are deemed useful to the model due to their correlation to the target variable. These four features will simply be combined to create two new features called ‘KillParat10’ and ‘KillParat15’, and were constructed as such:

$$Kill\ Participations = Kills + Assists$$

Checking the correlation of these new features now shows that the relationship between the target remains stable, whilst removing the problematic large correlations between predictor features in the dataset. Another correlation issue is seen between recurrent features at different minute intervals in-game. This is fixed by splitting these features into different datasets which in-turn will create separate models based on the statistics from the 10-minute mark and the 15-minute mark.

Balancing the data was deemed unnecessary due to the target variable - 'result', having a 52.8 - 47.2% win to loss balance as seen in Figure 3.1. This means that the dataset does not require any balancing techniques such as SMOTE to be applied, as it already should give reliable and decisive outputs.

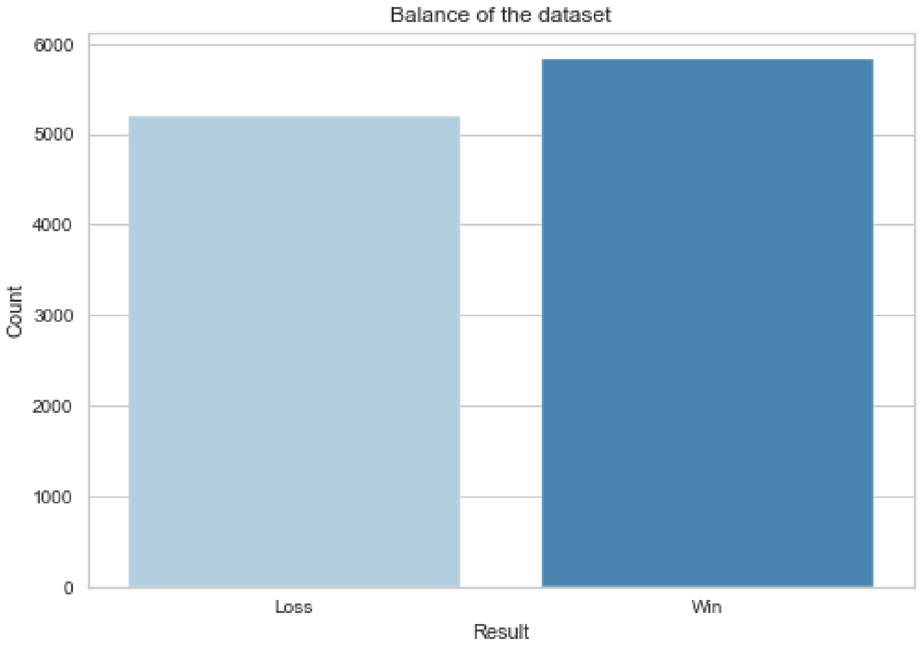


Figure 3.1: A graph showing the target balance of the dataset

Finally, the data is put through the normalization process, re-scaling the data in each column between the values of 0 and 1. This step is completed to reduce the effects of the magnitude of feature scale on the model, potentially creating faster converging and better performing models (Jayant Verma n.d.).

3.2 Data Modelling

Now that the dataset has been properly collected, cleansed and processed, the data can enter the data modelling process. With this data the target is a binary classification problem, meaning that a classification algorithm is required to help generate a probability for our target variable based on our features. As mentioned in Section 2, previous works have used a variety of methods such as Naïve Bayes classifiers, the Random Forest algorithm, Recurrent Neural Networks and Gradient Boosting. For this instance, it is chosen to use the Python Libraries - 'scikit-learn' and 'pycaret', they both offer a wide range of machine learning techniques After reviewing the possible models, it was chosen that the Random Forest algorithm would be used to create the baseline model. Then the PyCaret library would be used to test a larger pool of techniques and ultimately tune the model until the final model is chosen. The Random Forest algorithm is a commonly used supervised machine learning algorithm that creates an ensemble of decision trees and uses a bagging, majority voting system to help classify our problem - see Figure 3.2. This technique ensures that the chance of overfitting is kept to a minimum, as well as allowing for easy determination of importance from any given feature in the model.

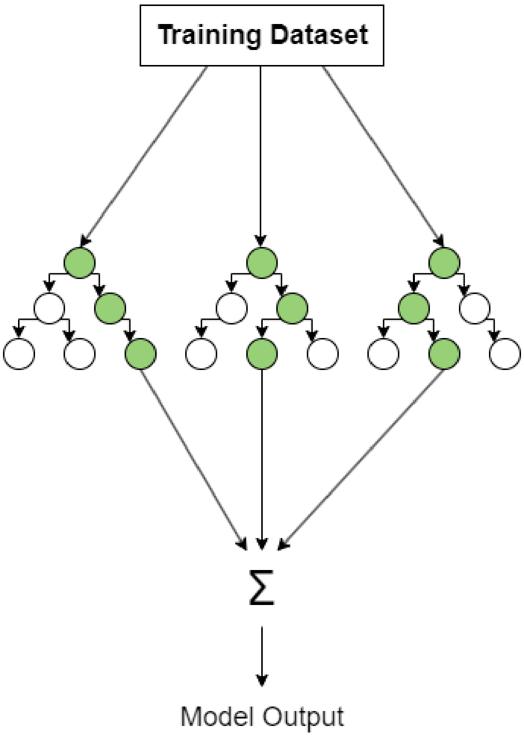


Figure 3.2: A diagram showing how the Random Forest algorithm classifies

A binary logistic regression model was also developed following the Random Forest Model. Like the previous model, a logistic regression classifier can estimate the probability of a target binary response variable using a set of predictor variables. This machine learning technique is based upon the natural logarithm of the odds; commonly referred to as a logit or log-odds, system for each event with a logistic function converting these log-odds into a probability. This logistic function is a sigmoid function that takes inputs between the bound of zero and one, it is defined as follows:

$$p(x) \ = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

Here β_0 is the intercept from the linear regression equation which is solved using numerical methods, and $\beta_1 x$ is the regression coefficient. With x being a combination of values from the predictor variables. Once solved for $p(x)$, it will provide the probability for the Blue-side team winning a given match.

The data then needs to be split into training data and testing data. Here a subset of the total dataset is split purely to train the machine learning model, before it is deployed to unseen training data. This help guarantees the model ensuring that there is no overfitting to the seen data, and validates the quality of the model. The data split ratio is chosen at a 70:30 train-test data split. This allows a larger split to train the model, and the test data will be tested on a more robust, accurate model because of this. A technique called Stratified K-Fold cross validation is used to get an accurate representation of the overall population balance from the target variable - reflecting the 52.8 - 47.2% win-loss balance, as well as evaluating a truer accuracy by summarising models built upon K subsets of data. This leads to a model that is less biased overall and more realistic in its prediction accuracies.

Table 3.2 shows the four models which were built for analysis purposes. The first three models were built using the Random Forest Algorithm with the 10, 15 and 20 minute datasets, and the following three models were built using Logistic Regression also using the datasets from the 10, 15 and 20 minute marks.

Table 3.2: A table describing the models that were built

Model	Technique	Dataset
1	Random Forest	10-minute
2	Random Forest	15-minute
3	Random Forest	20-minute
4	Logistic Regression	10-minute
5	Logistic Regression	15-minute
6	Logistic Regression	20-minute

3.3 Summary

In this Chapter, the data extracted from Oracle's Elixir was explored and described in a data overview. The methodology behind the data preparation were rationalised during the data cleansing and pre-processing subsections, highlighting key techniques used to ensure the data modelling process would work correctly. This helped reduce the dataset from over 140,000 rows with 100+ features to a more succinct and useful 11,000 rows with less than 20 features. This was then followed by the data modelling subsection, that explained the machine learning techniques used to create the predictive modelling system. Following this chapter is Chapter 4, which will present and analyse the results of this work.

Chapter 4

Results

In this chapter, the performance of the prediction model is analysed, and the discussion of the results are explored. The machine learning algorithms are evaluated to determine the best model at predicting on unknown data.

4.1 Performance Metrics

The performance of any algorithm is evaluated by a few factors that are measured on the test dataset, where the trained model attempts to correctly predict the target field and this performance is then compared to its true outcome. This assessment method is the norm on any binary classification problem and often will be visualised using a confusion matrix. The confusion matrix is a simple visualisation in which the four possible permutations of predictions are displayed. As seen in Figure 4.1, it compares the number of predictions of wins and losses made by the model verses the number of real wins and losses found in the dataset.

On the y-axis the true class of the result can be seen, and on the x-axis there is the model's predicted class. These four possible permutations are commonly referred to as follows:

- True Positive (TP) is seen in the bottom right of the matrix, this is when the predicted win actually reflects the true win.

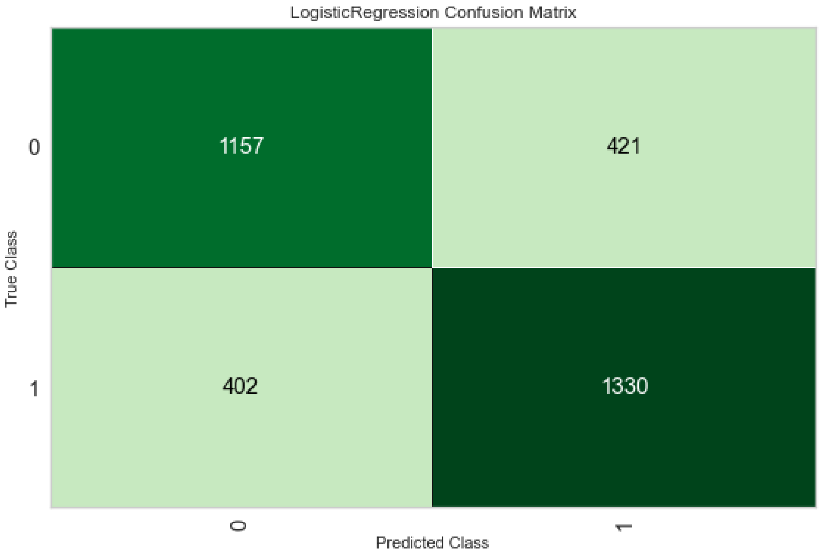


Figure 4.1: A confusion matrix of a Logistic Regression Model trained on the 10 minute dataset

- True Negative (TN) is seen in the top left of the matrix, this is when the predicted loss actually reflects the true loss.
- False Positive (FP) is seen in the top right of the matrix, this is when the predicted win is incorrect and the true class is a loss.
- False Negative (FN) is seen in the bottom left of the matrix, this is when the predicted loss is incorrect and the true class is a win.

By collecting these outcome values, the effectiveness of each model is then able to be assessed via many performance metrics that are calculated using these values. These measures each give an understanding of a model, with different implications depending on the objective of the project being assessed. In this study, the focus is on assessing the correct classifications and there is little to no consequences for misclassification. Therefore, the main criteria needed are the proportion of true classifications - aka the number of TPs and TNs compared to the whole prediction group. This calculation is also known as the Accuracy of the prediction, and is calculated as follows:

$$Accuracy = \frac{\sum TP + \sum TN}{\sum TP + \sum TN + \sum FP + \sum FN}$$

This is considered the key metric for this problem, as our dataset is pretty balanced so the other metrics such as Precision and Recall are not as important. Here Accuracy can help decide how well a given model is predicting the correct outcomes of an esports League of Legends match. The Accuracy of a model is only an effective metric if higher than the balance of the dataset, thus must be greater than 52.8% to show any improvement over guessing the balance of the data. Other metrics such as AUC, Precision, Recall are also calculated and evaluated, these metrics are calculated as such:

$$Precision = \frac{\sum TP}{\sum TP + \sum FP}$$
$$Recall = \frac{\sum TP}{\sum TP + \sum FN}$$
$$FPR = \frac{\sum FP}{\sum FP + \sum TN}$$

A Receiver Operating Characteristic (ROC) curve is a graph that plots the Recall verses the False Positive Rate (FPR), it plots these performance metrics between the bounds of zero and one. Area Under the ROC curve (AUC), is a measure that understandably calculates the area under the ROC curve which provides a measure of performance across all possible classification thresholds (Google n.d.). Essentially, AUC is the probability that a model can distinguish between a positive and negative class - in this study it represents the ability to distinguish wins from losses. Therefore, to conclusively accept a model as a feasible option, the Accuracy and AUC are the two core metrics that a model will have to score sufficiently on and will ultimately be the performance indicators to evaluate with.

4.2 Model Results

4.2.1 10 Minute Model

As seen in Table 4.1, most of the machine learning algorithms score quite similarly. The four metrics of Accuracy, AUC, Recall and Precision all are within a 5% range for every algorithm tested. Here the best performing machine learning algorithm is the Logistic Regression model, scoring an accuracy of 75.06% and an AUC of 82.30%. This is closely followed by the Linear Discriminant Analysis at an identical accuracy, but with an AUC of 82.28%.

Model	**Accuracy**	**AUC**	**Recall**	**Precision**
Logistic Regression	0.7506	0.8230	0.7798	0.7574
Linear Discriminant Analysis	0.7506	0.8228	0.7813	0.7566
Ada Boost Classifier	0.7469	0.8181	0.7747	0.7546
Naive Bayes	0.7458	0.8128	0.7613	0.7599
Gradient Boosting Classifier	0.7458	0.8407	0.7759	0.7529
Quadratic Discriminant Analysis	0.7456	0.8363	0.7601	0.7604
SVM - Linear Kernel	0.7436	0.0000	0.7769	0.7492
Light Gradient Boosting Machine	0.7357	0.8316	0.7630	0.7453
Random Forest Classifier	0.7316	0.8230	0.7498	0.7462
Extra Trees Classifier	0.7248	0.8116	0.7459	0.7385
K Neighbors Classifier	0.7061	0.7794	0.7276	0.7212
Decision Tree Classifier	0.6804	0.6794	0.6969	0.6998

Table 4.1: A table showing the model results for the 10 minute dataset

Interestingly enough the decision tree based algorithms that were commonly used in previous studies such as Random Forest, appear to perform slightly worse than algorithms such as Logistic Regression or Linear Discriminant Analysis. These techniques are techniques based on linear regression analysis that opt to find a linear combination of features that cause separation in a problem. These results then may indicate that the problem of match outcomes in League of Legends is linearly separable. The model for the 10-minute mark was chosen to be built upon a logistic regression technique.

Firstly, the model is tested across the data split by the stratified k-fold cross-validation and tuned afterwards. The resultant performance metrics can be seen in Table 4.2.

	Accuracy	**AUC**	**Recall**	**Precision**
Mean	0.7516	0.8226	0.7618	0.7682
Std	0.0150	0.0169	0.0285	0.0116

Table 4.2: A table showing the mean metrics after 10-fold cross-validation

Table 4.2 shows very similar results to the previous stated values, meaning that the model tuning had very little effect on improving the model's performance. It also gives a standard deviation on the core metrics of under 2%. During the modelling, the feature importance of the features included in the dataset were calculated. Figure 4.2 shows the average relative importance of each feature when using the 10-minute dataset. The top 10 features by feature importance are presented down the y-axis in decreasing order according to their relative importance values. The most important features are those related to the player performance inside their lanes such as 'csdiffat10', 'deathsat10' and 'KillParat10', with more objective team-based features having lower importance in predicting the resultant outcome of a given match. At first this seems counter-intuitive, especially when given that teams must destroy towers in order to eventually win the game. However, it is likely that at this stage of the game there is very low likelihood that teams can effectively destroy towers, so the focus should be on progressing their individual leads inside their lane. These leads that they garner are reflected in the three features mentioned previously, and should translate into further advantages in more objective oriented features in a later game-state.

4.2.2 15 Minute Model

Table 4.3 highlights that once again that all the machine learning algorithms score within a 5% range of the four key metrics tracked. The two best performing machine learning algorithm remain the Logistic Regression model

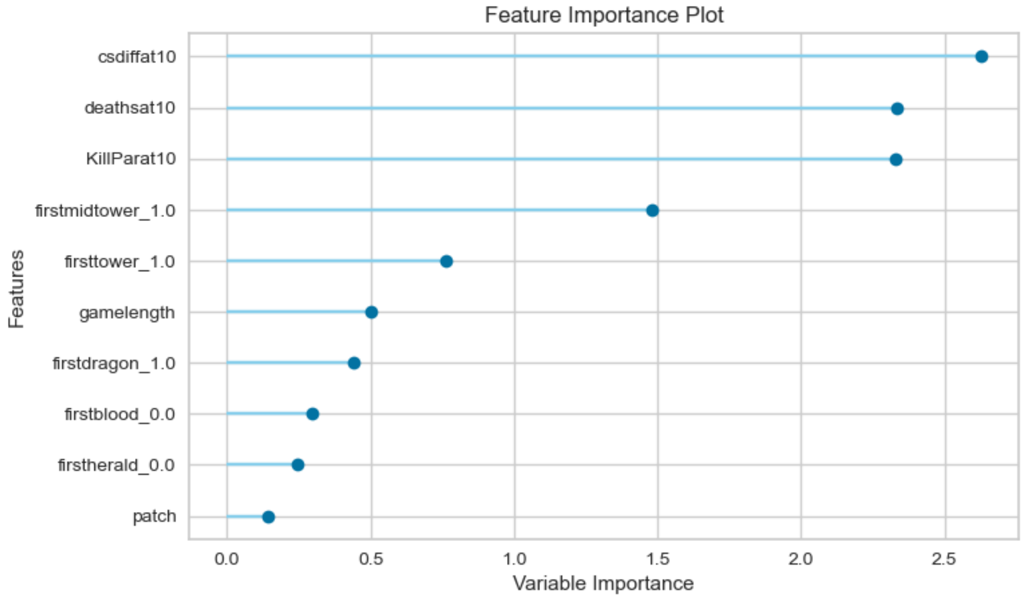


Figure 4.2: A graph showing the imp 10 minute dataset

and the Linear Discriminant Analysis model, scoring an accuracy of 76.84% and 76.87%, and AUC values of 84.74% and 84.53% respectively.

4.2.3 20 Minute Model

As seen in Table 4.4, most of the machine learning algorithms score quite similarly. The four metrics of Accuracy, AUC, Recall and Precision all are within a 5% range for every algorithm tested. Here the best performing machine learning algorithm is the Logistic Regression model, scoring an accuracy of 75.06% and an AUC of 82.30%. This is closely followed by the Linear Discriminant Analysis at an identical accuracy, but with an AUC of 82.28%.

Model	**Accuracy**	**AUC**	**Recall**	**Precision**
Linear Discriminant Analysis	0.7687	0.8453	0.7994	0.7727
Logistic Regression	0.7684	0.8474	0.7972	0.7733
Ridge Classifier	0.7681	0.0000	0.7986	0.7721
Ada Boost Classifier	0.7662	0.8418	0.7899	0.7741
Quadratic Discriminant Analysis	0.7643	0.8578	0.7794	0.7772
Gradient Boosting Classifier	0.7638	0.8601	0.7935	0.7690
Naive Bayes	0.7589	0.8362	0.7694	0.7746
Light Gradient Boosting Machine	0.7577	0.8551	0.7877	0.7635
SVM - Linear Kernel	0.7562	0.0000	0.7835	0.7642
Random Forest Classifier	0.7519	0.8483	0.7745	0.7623
Extra Trees Classifier	0.7431	0.8359	0.7652	0.7545
K Neighbors Classifier	0.7307	0.8067	0.7449	0.7470
Decision Tree Classifier	0.6870	0.6854	0.7125	0.7021

Table 4.3: A table showing the model results for the 15 minute dataset

Model	**Accuracy**	**AUC**	**Recall**	**Precision**
Gradient Boosting Classifier	0.8585	0.9433	0.8782	0.8586
Logistic Regression	0.8583	0.9266	0.8648	0.8679
Linear Discriminant Analysis	0.8583	0.9257	0.8513	0.8780
Quadratic Discriminant Analysis	0.8582	0.9406	0.8706	0.8634
Ridge Classifier	0.8581	0.0000	0.8511	0.8778
Ada Boost Classifier	0.8559	0.9245	0.8623	0.8655
Random Forest Classifier	0.8525	0.9387	0.8653	0.8581
SVM - Linear Kernel	0.8521	0.0000	0.8504	0.8701
Light Gradient Boosting Machine	0.8494	0.9392	0.8645	0.8538
Extra Trees Classifier	0.8451	0.9315	0.8584	0.8512
K Neighbors Classifier	0.8367	0.9057	0.8433	0.8482
Naive Bayes	0.8284	0.9122	0.8509	0.8300
Decision Tree Classifier	0.8024	0.8017	0.8140	0.8137

Table 4.4: A table showing the model results for the 20 minute dataset

Chapter 5

Conclusion

1500 words

It can be hard to know which section to discuss your results – this or the preceding one – and you may decide to combine these two sections into one or more chapters based on theme, depending on your topic and your supervisor's views. However, what is vital is that your Dissertation contains sufficient analytical discussion in addition to the more descriptive 'scene setting' material of the literature review sections, and presentation of results. It is here that you will compare and contrast your findings with those already reported in the literature.

Here you need to answer the "So what?" question. What significance do your research findings have? For whom? Why? and How? In this chapter you link the research problem with literature review and findings, stating what you can conclude based on the work conducted. Based on your conclusions you should comment on managerial implications, the limitations of the research, suggest further work and better ways to resolve the problem.

5.1 Summary of Works

5.2 Research Limitations

5.3 Further Research

Bibliography

Abarbanel, B. & Johnson, M. R. (2019), ‘Esports consumer perspectives on match-fixing: implications for gambling awareness and game integrity’, *International Gambling Studies* **19**(2), 296–311.

Absolute Reports (2022), ‘Global esports betting market’.

Alin, A. (2010), ‘Multicollinearity’, *Wiley interdisciplinary reviews: computational statistics* **2**(3), 370–374.

Ani, R., Harikumar, V., Devan, A. K. & Deepa, O. (2019), Victory prediction in league of legends using feature selection and ensemble methods, *in* ‘2019 International Conference on Intelligent Computing and Control Systems (ICCS)’, IEEE, pp. 74–77.

Apostolou, K. & Tjortjis, C. (2019), Sports analytics algorithms for performance prediction, *in* ‘2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA)’, IEEE, pp. 1–4.

Brownlee, J. (2019), ‘How to choose a feature selection method for machine learning’, *Machine Learning Mastery* **10**.

Byrne, L. (2019), ‘Betway agrees to a one-year sponsorship deal for BLAST Pro Series’, *Esports Insider* . Accessed: 20-07-2022.
URL: *https://esportsinsider.com/2019/03/betway-agrees-to-a-one-year-sponsorship-deal-for-blast-pro-series/*

Chen, Z., Nguyen, T.-H. D., Xu, Y., Amato, C., Cooper, S., Sun, Y. & El-Nasr, M. S. (2018), The art of drafting: a team-oriented hero recommendation system for multiplayer online battle arena games, *in* ‘Proceedings of the 12th ACM Conference on Recommender Systems’, pp. 200–208.

Dos Reis, V. (2017), ‘Q&a: The rise of esports betting and the challenges the industry faces’, *Gaming Law Review* **21**(8), 630–633.

Dot Esports (2021), ‘FPX’s Bo handed 4-month ban for match-fixing in Chinese academy league’. Accessed: 23-07-2022.
URL: *https://dotesports.com/league-of-legends/news/fpx-bo-handed-four-month-ban-match-fixing*

Duckett, C. (2016), ‘Google AlphaGo AI clean sweeps European Go champion’, *ZDNet* . Accessed: 24-07-2022.
URL: *https://www.zdnet.com/article/google-alphago-ai-clean-sweeps-european-go-champion/*

Esports Earnings (n.d.), ‘Largest Individual Tournament Prize Pools’. Accessed: 16-07-2022.
URL: *https://www.esportsearnings.com/tournaments/largest-team-tournaments*

Even, W. E. & Noble, N. R. (1992), ‘Testing efficiency in gambling markets’, *Applied Economics* **24**(1), 85–88.

Feddersen, A., Humphreys, B. R. & Soebbing, B. P. (2018), ‘Sentiment bias in national basketball association betting’, *Journal of Sports Economics* **19**(4), 455–472.

Gaina, R. & Nordmoen, C. (2018), ‘League of legends: A study of early game impact’.

Google (n.d.), ‘Classification: ROC Curve and AUC’. Accessed: 08-08-2022.
URL: *https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc*

Gray, G. T. & Wert-Gray, S. (2012), ‘Customer retention in sports organization marketing: examining the impact of team identification and satisfaction with team performance', *International Journal of Consumer Studies* **36**(3), 275–281.

Hanke, L. & Chaimowicz, L. (2017), A recommender system for hero line-ups in moba games, *in* 'Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainmen', pp. 43–49.

Jayant Verma (n.d.), '2 Easy Ways to Normalize data in Python'. Accessed: 05-08-2022.
URL: *https://www.digitalocean.com/community/tutorials/normalize-data-in-python*

Kokkinakis, A., York, P., Patra, M., Robertson, J., Kirman, B., Coates, A., Pedrassoli Chitayat, A., Demediuk, S. P., Drachen, A., Hook, J. D. et al. (2021), 'Metagaming and metagames in esports', *International Journal of Esports* .

Lee, S.-K., Hong, S.-J. & Yang, S.-I. (2020), Predicting game outcome in multiplayer online battle arena games, *in* '2020 International Conference on Information and Communication Technology Convergence (ICTC)', IEEE, pp. 1261–1263.

Lin, L. (2016), 'League of legends match outcome prediction', *Comput. Sci. Dept., Univ. Stanford, Stanford, CA, USA, Rep* .

LoLEsports (2022), 2022 LCS Rule Set, Technical report, Riot Games.
URL: *https://lolesports.com/article/2021-lcs-rule-set-and-penalty-index/bltd4266fc4777c19a9*

Mangeloja, E. (2019), 'Economics of esports', *Electronic Journal of Business Ethics and Organization Studies* **24**(2).

McLaughlin, D. (2021), 'Worlds 2021 final draws 73.8 million peak concurrent viewers, Riot reports', *Upcomer* . Accessed: 18-07-2022.
URL: *https://upcomer.com/worlds-2021-final-draws-73-8-million-peak-concurrent-viewers-riot-reports*

Na, S., Su, Y. & Kunkel, T. (2019), 'Do not bet on your favourite football team: the influence of fan identity-based biases and sport context knowledge on game prediction accuracy', *European Sport Management Quarterly* **19**(3), 396–418.

Newzoo (2022), Global Esports & Live Streaming Market Report, Technical report, Newzoo.

Novak, A. R., Bennett, K. J., Pluss, M. A. & Fransen, J. (2020), 'Performance analysis in esports: modelling performance at the 2018 league of legends world championship', *International Journal of Sports Science & Coaching* **15**(5-6), 809–817.

Oracle's Elixir (n.d.), 'Match Data Downloads'. Accessed: 06-06-2022.
URL: *https://oracleselixir.com/tools/downloads*

Pantzalis, V. C. & Tjortjis, C. (2020), Sports analytics for football league table and player performance prediction, *in* '2020 11th International Conference on Information, Intelligence, Systems and Applications (IISA', IEEE, pp. 1–8.

Riot Games (2021), 'Total League of Legends Playercount'. Accessed: 18-07-2022.
URL: *https://twitter.com/riotgames/status/1455172784938651649?s=20-&t=dydJavkVHnrR5YnceUFnMw*

Sarlis, V. & Tjortjis, C. (2020), 'Sports analytics—evaluation of basketball players and team performance', *Information Systems* **93**, 101562.

Scelles, N., Peng, Q. & Valenti, M. (2021), 'Do the peculiar economics of professional team sports apply to esports? sequential snowballing literature reviews and implications', *Economies* **9**(1), 31.

scikit-learn (n.d.), '1.10. Decision Trees'. Accessed: 24-07-2022.
URL: *https://scikit-learn.org/stable/modules/tree.html*

Shen, Y., Zhou, J., Lin, W. & Feng, Z. (2022), A deep learning supported sequential recommendation mechanism for ban-pick in moba games, *in* '2022 IEEE 2nd International Conference on Software Engineering and Artificial Intelligence (SEAI)', IEEE, pp. 259–265.

Silva, A. L. C., Pappa, G. L. & Chaimowicz, L. (2018), 'Continuous outcome prediction of league of legends competitive matches using recurrent neural networks', *SBC-Proceedings of SBCGames* pp. 2179–2259.

TIOBE (n.d.), 'TIOBE Index for August 2022'. Accessed: 28-07-2022.
URL: *https://www.tiobe.com/tiobe-index/*

Ward, M. R. & Harmon, A. D. (2019), 'Esport superstars', *Journal of Sports Economics* **20**(8), 987–1013.

Appendix A

Python Code

Code goes here

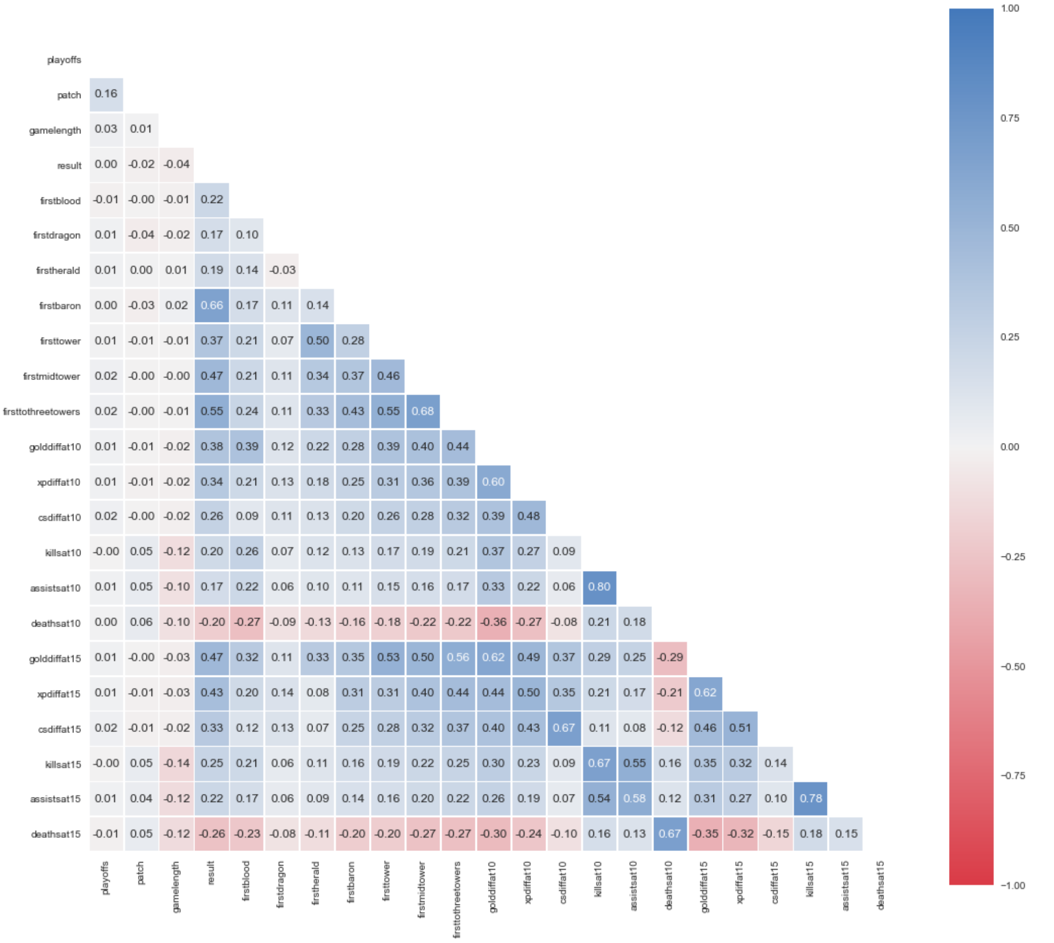
Appendix B

Data Description

Data description table

Appendix C

Feature Correlations



	playoffs	patch	gamelength	result	firstblood	firstdragon	firstherald	firstbaron	firsttower	firstmidtower	firsttothreetowers	golddiffat10	xpdiffat10	csdiffat10	killsat10	assistsat10	deathsat10	golddiffat15	xpdiffat15	csdiffat15	killsat15	assistsat15	deathsat15
playoffs																							
patch	0.16																						
gamelength	0.03	0.01																					
result	0.00	-0.02	-0.04																				
firstblood	-0.01	-0.00	-0.01	0.22																			
firstdragon	0.01	-0.04	-0.02	0.17	0.10																		
firstherald	0.01	0.00	0.01	0.19	0.14	-0.03																	
firstbaron	0.00	-0.03	0.02	0.66	0.17	0.11	0.14																
firsttower	0.01	-0.01	-0.01	0.37	0.21	0.07	0.50	0.28															
firstmidtower	0.02	-0.00	-0.00	0.47	0.21	0.11	0.34	0.37	0.46														
firsttothreetowers	0.02	-0.00	-0.01	0.55	0.24	0.11	0.33	0.43	0.55	0.68													
golddiffat10	0.01	-0.01	-0.02	0.38	0.39	0.12	0.22	0.28	0.39	0.40	0.44												
xpdiffat10	0.01	-0.01	-0.02	0.34	0.21	0.13	0.18	0.25	0.31	0.36	0.39	0.60											
csdiffat10	0.02	-0.00	-0.02	0.26	0.09	0.11	0.13	0.20	0.26	0.28	0.32	0.39	0.48										
killsat10	-0.00	0.05	-0.12	0.20	0.26	0.07	0.12	0.13	0.17	0.19	0.21	0.37	0.27	0.09									
assistsat10	0.01	0.05	-0.10	0.17	0.22	0.06	0.10	0.11	0.15	0.16	0.17	0.33	0.22	0.06	0.80								
deathsat10	0.00	0.06	-0.10	-0.20	-0.27	-0.09	-0.13	-0.16	-0.18	-0.22	-0.22	-0.36	-0.27	-0.08	0.21	0.18							
golddiffat15	0.01	-0.00	-0.03	0.47	0.32	0.11	0.33	0.35	0.53	0.50	0.56	0.62	0.49	0.37	0.29	0.25	-0.29						
xpdiffat15	0.01	-0.01	-0.03	0.43	0.20	0.14	0.08	0.31	0.31	0.40	0.44	0.44	0.50	0.35	0.21	0.17	-0.21	0.62					
csdiffat15	0.02	-0.01	-0.02	0.33	0.12	0.13	0.07	0.25	0.28	0.32	0.37	0.40	0.43	0.67	0.11	0.08	-0.12	0.46	0.51				
killsat15	-0.00	0.05	-0.14	0.25	0.21	0.06	0.11	0.16	0.19	0.22	0.25	0.30	0.23	0.09	0.67	0.55	0.16	0.35	0.32	0.14			
assistsat15	0.01	0.04	-0.12	0.22	0.17	0.06	0.09	0.14	0.16	0.20	0.22	0.26	0.19	0.07	0.54	0.58	0.12	0.31	0.27	0.10	0.78		
deathsat15	-0.01	0.05	-0.12	-0.26	-0.23	-0.08	-0.11	-0.20	-0.20	-0.27	-0.27	-0.30	-0.24	-0.10	0.16	0.13	0.67	-0.35	-0.32	-0.15	0.18	0.15	

Figure C.1: A matrix of correlations between features in the dataset

Appendix D

Idk

Put other stuff here