Predicting Counter-Strike Matches

Using Machine Learning Models

Erik Broms & William Nordansjö

Bachelor thesis in Statistics

Department of Statstics
Lund University School of Economics and Management
Sweden

January 2024

Supervisor: Jonas Wallin

Abstract

Sports betting is a widespread industry where predictive modeling play a big role. The goal of this thesis is to explore the possibilities of machine learning within the realm of e-sport prediction. The data used for this thesis is publicly available data was recorded over a three year period. The chosen variables are defined as the difference in player performance between two teams in order to create conditional probabilities. The paper focuses on two machine learning models for evaluating predictability within the data, Logistic regression with parameter regularization and Random Forest. Both models were optimised with cross-validation and their effectiveness is compared to a benchmark which in this case is the betting odds of multiple bookmakers. Measures such as accuracy, Log-Loss and the κ-parameter are our main points of comparison. The findings suggest that our models were capable of achieving an accuracy exceeding 50/50 on the test data, implying a certain level of predictability. The models were also applied to a professional tournament, and although a small sample size with large standard errors had an influence, we conclude that the evaluated models did not surpass the performance of the benchmarks.

Contents

Glossary

ADR Average damage per round. 11, 40

APR Average number of assists on teammates kills. 11, 40

Counter-Strike 2 The sequel and successor to Counter-Strike: Global Offensive. 8, 27

Counter-Strike: Global Offensive A multiplayer first person shooter game. 8

DPR Average number of deaths per round. 11, 40

Impact Overall impact of player performance, correlated with decisive kills and assists. 11, 40

Kaggle An online data science platform and community. 10

KDR The ratio between player kills and player death. 11, 40

mtry Number of variables tested at each split of a decision tree. 14, 23

Rating Metric of player quality. 11, 40

SPR Average number of saved teammates per round. 11, 40

Chapter 1

Introduction

Sport is a field that attracts interest from both companies and researchers. As it is a billion-dollar industry, and according to Wunderlich & Memmert (2021), being able to accurately forecast the outcome is a fundamental aspect of sports betting for both bookmakers as bettors. Modern sport prediction and team composition rely on data, which must be meticulously recorded. The quality of this data is contingent upon the accuracy of the information collected and the efficiency of data processing techniques. Instead of competing on a physical field, imagine the sport taking place in a virtual arena, each and every player action, position and situation can be measured. This could then be evaluated and used to train machine learning algorithms.

The purpose of this paper is to widen the academic understanding of specific machine learning models performance on publicly available Counter-Strike player-performance data. This objective will be reached by evaluating metrics such as accuracy and κ on model performance. Our research question is: How accurately can logistic and random forest models predict professional Counter-Strike matches based on historical player performance?

This paper looks at the online multiplayer first person shooter *Counter-Strike*. The *Background* chapter conveys the games evolution from a fan made modification to a celebrated tactical shooter and a giant in the E-sport industry. Following this, a literature overview is given, which describes the relationship between sport prediction and machine learning as well as giving an overview of related studies and their methods and results. The *Data* chapter describes the dataset and its structure and variables, it also describes how the data was transformed and adapted in order to fit our analysis. The *Empiric* chapter is made up of a method section where we discuss how to evaluate and compare the models as well as describing cross-validation which is applicable on both models. The

model section gives a description of the models used, Random Forest and Logistic Regression. The last section of the empiric chapter contain both the result of our model training, cross-validation as well as how our models performed when applied to a real world data set. Finally, in the *Analysis* chapter, the result is discussed analysed. Following this we provide our conclusion and suggestions for future research.

Chapter 2

Background

Section 2.1 will give an overview of what Counter-Strike is and the transition to Counter-Strike 2. While section 2.2 will provide an overview of previous research and literature related to machine learning and E-sport.

2.1 Counter-Strike

Counter-Strike is a online multiplayer first person shooter, with an emphasis on competitive gameplay. Two teams compete in multiple rounds of objective based game modes with the aim of winning the most matches. The main game mode is called competitive and it pits two teams of five against each other. It involves the terrorist team planting a bomb while the Counter Terrorist team attempts to stop them. After each round, players are rewarded based on individual and team performance. This reward shows itself as in-game currency and can be spent on weapons and utilities ("Counter-Strike: Global Offensive", 2023).

Counter-Strike saw its beginnings as a popular modification of the game Half-Life, but was later recognised as an official product of the Valve Corporation, and has since seen multiple later releases including Counter-Strike 1.6, Counter-Strike: Source and Counter-Strike: Global Offensive (Llewellyn, 2018).

Professional play began in 2012 where popularity and its scale has according to Ferguson (2018) grown ever since. CS:GO Major events like ESL One occur a couple of times each year. Professional teams compete for prize funds totaling over $1,000,000. Tournaments have audiences of thousands at live events and millions through online streaming services.

2.1.1 Transition to Counter-Strike 2

Counters Strike: Global Offensive has been the primary version in recent years. On Wednesday the 27th of september 2023 the successor was released. The term successor instead of sequel is apt in this case as Global Offensive was made unavailable upon the the release of Counter-Strike 2, this was done in order to combat a potential schism like what happened between previous iterations (Stubbs, 2023). The biggest change in the latest version is the new game engine. The new engine overhauls the games graphics, character models and sounds. Another major change was the transition to sub tick servers which slightly change the spray rate of different weapons (Stanton, 2023). This means that while the data our models was trained on originates from Counter-Strike: Global Offensive, the application and analysis will be performed on Counter-Strike 2.

2.2 Literature overview

2.2.1 Sport prediction and machine learning

Cojocariu (2022) provides a historical perspective on the evolution of sport analysis. In the 1970s attempts were initially made to predict the profitability of sporting events, it has since transitioned into being primarily interested in predicting match outcomes with machine learning models. The author focuses on the importance and availability of predictive variables, also called features, and underlines that technological advances such as cameras and sensors has enabled more variety in potential features. Bunker & Susnjak (2022) contributes to this narrative by examining the diverse application of machine learning models in predicting outcomes across different sports, emphasizing variations in feature selection and their impact on prediction accuracy. Additionally, Wilkens (2021) looks at the case of tennis and finds large variation within the sport, both in terms of which features were used with different models and which accuracy they achieved, with predicted accuracy ranging from 67% to 99%.

2.2.2 E-sports and machine learning

In the realm of e-sports the opportunity to utilize more data as features is increased as every player action theoretically can be recorded and used to train a model. Hodge et al. (2019) present a case study focusing on live-betting in the context of Multiplayer Online Battle Arenas (MOBAs), and differentiate three types of data *pregame*, *in-game* and

post-game, where *in-game* data can be collected at a particularly low level, for example the topological features of the individual team members positions. Makarov et al. (2018) contribute to the understanding of e-sport prediction by attempting to examine both individual and team level ratings. This was done by applying Microsofts *TrueSkill* system (which is a form of ELO) to Dota 2 and CSGO, and then analysing game replays in order to determine player and team ratings. Xenopoulos et al. (2020) also attempts to predict outcomes in Counter-Strike: Global Offensive (CS:GO) by using player metrics extracted from game demos. The authors focused on how to assign value to player actions and developed a framework called Win Probability Added or WPA which attempts to dynamically assign a change to win probability based on the action of a given player. In a similar work to ours, Švec (2022) applied different machine learning models to scraped HLTV.org data, and achieved the best accuracy with an ELO-based model. A recurring theme in the literature is the emphasis on feature selection as a crucial factor of increasing the accuracy of predictions of match outcomes, a summary of model performance of previous studies can be found in table 2.1.

Table 2.1: Accuracy metrics of previous studies (note: only the studies top performing models were included)

Model	Accuracy	AUC	Log-Loss
Makarov et al. (2018)	0.62	–	0.675
Xenopoulos et al. (2020)	–	0.791	0.535
Švec (2022)	0.64	–	–

Chapter 3

Data

The following chapter will overview the data used in the thesis. Section 3.1, describes the original data, its structure and its variables. Section 3.2 will discuss how we condition the data as the performance difference between the two competing teams. Section 3.2 also details how the data transformed into the final data set used for our models. Finally, section 3.3 describes the data we collected from the professional tournament BLAST Premier Fall Final 2023.

3.1 Original data set

The data used in this thesis is a Kaggle data set created by Gabriel Tardochi (2020) using data on professional matches between 2017 and 2020, scraped from HLTV.org. The data is organised by match basis, and contains information on team and player level. The team level data contains values such as team-name and match date while the player level data contains statistics of players one to five for both teams. Kaggle data sets are free and open source, and this data set falls under the licensing "CC BY-NC-SA 4.0", which enables us to share and adapt the data (Creative commons (n.d.)).

3.1.1 Data Structure and Variables

The original structure of the data file is compromised of 3788 rows and 170 columns. Every row represents one professional CS:GO game spanning the time-frame 18/12/2016 - 10/01/2020. The first 10 columns are game specific and the rest are player specific. Each player has 16 columns and is structured as following: The first 16 columns are dedicated to player one in team one. The following 16 columns are player two in team one and so on

until every player in team one is covered. The data then switches to player one in team two and continues the same way until every player in their team is included.

Not all of the 16 variables can be used as covariates simply because they are not available to be extracted as HLTV player statistics. Because of this the input variables that will be used as covariates are Average number of deaths per round (DPR), Average number of saved teammates per round (SPR), The ratio between player kills and player death (KDR), Metric of player quality (Rating), Overall impact of player performance, correlated with decisive kills and assists (Impact), Average damage per round (ADR) and Average number of assists on teammates kills (APR). The output variable will be the binary *Win team 1*, or positive match outcome from the perspective of an arbitrarily chosen team in the data. In other words, we are regressing on whether or not the team that is classified as team 1, wins or loses given a competing team.

Datapoints that represent performance, such as KDR or SPR, are recorded within a specific game. Meanwhile Rating and Impact are assigned to players at the time of the game. When it comes to predicting future game outcomes, the input variables will be acquired from HLTVs website, these are the player average stats for their entire professional career.

3.2 Final data set

3.2.1 Conditional data

The ultimate objective of our models is to estimate the probability of victory for a team when facing a specific opponent. To achieve this, the model needs to be trained on data that is conditioned in relation to the opposing team. This was done by aggregating the statistics of players from Team 1 and subtracting the aggregated statistics of Team 2, resulting in variables representing a suggested performance difference between the two teams. This enables us to interpret our output variable as *win* if > 0.5 and *loss* < 0.5.

3.2.2 Transformations and final data set

In order to make the data useful for our purposes, some data wrangling had to be performed. First of all we divided the data into player data and match data. This division was made possible by assigning each row an individual number which we defined as match id as well as creating an additional data set called index, containing match date, match id

as well as the binary variable win team 1. This index enabled us to link the two data sets while still keeping them separate. At this point we have a data set where each row is a match and the columns are the player specific statistics of player one to ten, additionally, we have a data set containing redundant match data with variables such as team name, these will be disregarded going forward. In order to compare the two teams, we divided the player data further and created ten player specific data sets, once again linked by indexing. At this point we also assigned the players with the binary variable ist1, this was to be able to differentiate between team one and two. Players 1-5 are described as 1, while players 6-10 are 0.

The player specific data sets could now be combined, and by stacking them on top of each other we achieved a complete data set where each row is an individual player and each column a separate variable. Then we aggregated the data by calculating the mean of our variables, grouped by match and team which resulted in a data set where each row is a match played, and the columns are the average stats of the two teams which played in that specific match. From this data set we calculated the difference of average performance statistics between the two teams, which later will be used in the machine learning models. The final data set was randomly divided into thirds. Two thirds of the data was used to train the models, this is referred to in the text as *the training data* and the final third were used to test the model and is refer to as *the test data* in the text.

3.3 BLAST Premier Fall Final 2023

The comparison and testing of accuracy between models and bookmakers will be conducted during one of the bigger Counter-Strike tournaments of the year. The BLAST Premier Fall Final 2023 Tournament located in Denmark. Eight teams competed for a total prize pool of 425,000 dollars. Bookmakers odds were captured at the day of the match. Individual player statistics for the players competing were collected from HLTV.org. These statistics reflect the players average stats for their entire career as a professional player.

Chapter 4

Empiric

Section 4.1 will go over cross-validation, κ and Log-Loss. The parameters used for evaluating the models. The following chapter 4.2 will overview the models and how they work. Section 4.3 will go through the results generated in the training, test and validation stage.

4.1 Method

The models we have chosen are two *Logistic regression* models and a *Random forest* model. All models will use the same data-set discussed in the previous chapter. These will be compared to each other as well as to various betting sites as a benchmark of performance.

These models will provide a probability of success for each team competing with probabilities ranging from 0 - 1, where a larger number indicate a higher probability of success. The creation and evaluation of model performance will be conducted in three stages. The first stage is training and it will be conducted on two thirds of the input data. The subsequent phase involves testing, where we assess the models capability to predict the outcomes of the unseen final third of the input data. In this context accuracy and the κ-parameter will be examined. The last stage compares the models to the benchmark models with the BLAST dataset, here the predictions will be evaluated and rated using accuracy, κ and Log-loss.

4.1.1 Cross-validation

Cross-validation will be used on the models. It is a statistical method for evaluating and comparing learning algorithms as described by Refaeilzadeh et. al (2009). It does this by dividing the data into two categories, training data used for teaching the model, and

validation data used for validating the model. Usually in cross-validation, training and validation sets cross-over in successive rounds, this ensures that each data point is being trained on and validated against. The most basic form, and the one being used, is K-fold cross validation.

K-fold cross-validation divides the data into K equally sized segments. Subsequently, K iterations of training and validation are performed with each iteration $K-1$ sets of the data is used for training and the remaining set of data is used for validation. The performance of each learning instance can and will be tracked using the metric *accuracy*. The final accuracy metric will be the average of the K accuracy iterations. A K of 10 will be used for evaluating the data.

Cross-validation will be used for finding the optimal λ value in the Ridge - and LASSO logistic regression. It will also be used for finding the optimal mtry value for Random forest. What λ and mtry means will be expanded upon in following sections.

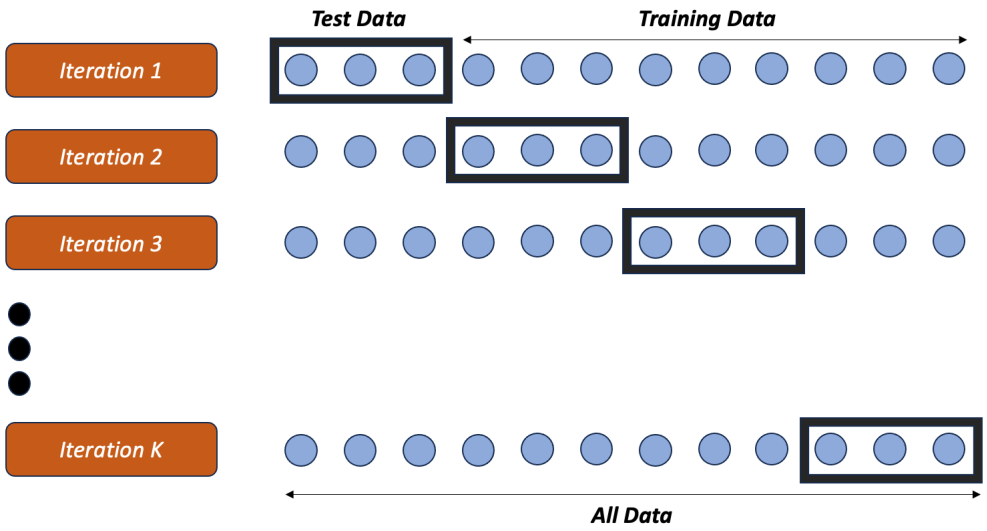

Figure 4.1: K-fold cross-validation

4.1.2 Model significance, the κ-parameter

While less intuitive than accuracy, κ or Cohen´s kappa as described by McHugh ML (2012) is a measure of whether or not the predictive capabilities of the models could be attributed to chance. It is a form of correlation coefficient that ranges form (-1 to 1). κ is usually discussed in terms of agreement between the categorical data sets (in our case predicted and true match outcomes). Absolute values close to $|1|$ suggest a relationship

between the data sets, while values close to zero attributes the relationship to chance. In general, absolute values < 0.2 is considered as slight agreement, between 0.2 and 0.4 as moderate, < 0.6 is considered as substantial and > 0.8 as almost perfect agreement. κ can be expressed mathematically as

$$\kappa = \frac{P(x) - P(z)}{1 - P(z)} \tag{4.1}$$

where P(x) is the observed agreement, or the number of agreements divided by number of matches. And P(z) is the expected number of chance agreements, or the sum of rows total times columns total divided by the total number of matches squared.

4.1.3 Log-Loss

To gauge the performance of our models in comparison to the benchmark beyond accuracy and κ, we will employ the log-loss function for result comparisons which is defined by Dembla (2020).

A game will be considered to be correctly predicted when the team with the highest percentage of success wins the match. For instance, if team one has a calculated probability of victory greater than 50 percent and wins, the model has successfully predicted the game. Even if two models successfully predicted the game, they did not necessarily do so with the same confidence. The log-loss function then allows us to evaluate and compare multiple models that made the correct prediction. The general formula of the log-loss function is presented in (4.2) where y_i is the true result of the i:th match, p_i is the calculated probability of victory and ln is the natural logarithm. The log-loss function is usually expressed as being negative in order to frame it as a minimization problem, expressed as

$$Logloss_i = -[y_i ln p_i + (1 - y_i) ln(1 - p_i)]. \tag{4.2}$$

The log-loss function can also be expanded to any given amount of matches by a modification of expression 4.2, where the log-loss value of a certain match is summed up and divided by the total amount of matches evaluated. Thus, the average log-loss is given by

$$AVGLogloss = -\frac{1}{N} \sum_{i=1}^{N} [y_i ln p_i + (1 - y_i) ln(1 - p_i)]. \tag{4.3}$$

4.1.4 Benchmarking with bookmakers

Since there are no publicly available prediction models built on CS:GO data, we have selected various betting sites, or bookmakers, as our benchmark on model performance. There are a plethora of online betting sites available, and although we do not have access to their calculations behind their predictions, they attribute professional CS:GO games an odds ratio. That the odds ratio reflect the house prediction on the match outcome and Tillman (2023) showcase that they can easily be converted into percentage of probability by dividing 1 with the attributed odds, expressed as

$$Probability_{teamA} = P(a) = \frac{1}{Odds}. \tag{4.4}$$

This calculated probability will most likely not be a 1:1 representation of the house prediction but a very close one. If one would to add their probabilities, they will not sum up to 100 percent but rather approximately 100 to 105 percent. Sohail (2023) attribute this is to the house adding a small percentage to the total probability and thereby increasing their expected value of the match's outcome.

Due to this, the odds will be normalized to adhere to a 0-100% structure, ensuring they collectively sum up to 1. This normalization process involves dividing each inverted odds value by the sum of all inverted odds, expressed mathematically as

$$\frac{P(a)}{P(a) + P(b)}. \tag{4.5}$$

The same percentage of correctly predicted matches will be calculated for the betting sites. If their percentage of probability for team one exceeds 50 percent and the team wins, the site has correctly predicted the game outcome. Once acquired, we can compare the calculated benchmark probability and the resulting log-loss value, to the results of our models.

4.2 Models

4.2.1 Logistic regression

Logistic regression belongs to the family of supervised machine learning models and use predictive analysis by estimating the probability of an event occurring. Because of this, the dependant variable is bounded to be between 0 and 1 (IBM (n.d. (1)).

We will employ the maximum likelihood method for parameter estimation in logistic regression, although various methods are available for this purpose. Akalin (2020) explains that the objective of the model is to maximize the likelihood that the sampled data accurately represents a distribution of interest. If y_i represents our observed value, constrained to only 0 and 1, we can conceptualize y_i as a random variable with probabilities p_i and $1 - p_i$ for the values 0 and 1, respectively. The model, designed to meet the boundary requirement of 0 and 1, is articulated as

$$p_i = \frac{e^{(\beta_0+\beta_1 x_i+....+\beta_n x_i)}}{1 + e^{(\beta_0+\beta_1 x_i+....+\beta_n x_i)}}. \tag{4.6}$$

The equation can be linearized as

$$logit(p_i) = ln(\frac{p_i}{1-p_i}) = \beta_0 + \beta_1 x_i + ... + \beta_n x_i, \tag{4.7}$$

where the left side of the equation termed "logit" stands for logistic unit, also known as log odds. The model will now produce values on the log scale. With the equation above, values can be transformed to the 0-1 scale. We can now with maximum likelihood estimate the parameters that maximizes the likelihood the statistical model actually produced the observed data.

The response variable follows the Bernoulli distribution, which is a special case of the Binomial Distribution. The model now needs to find the parameter p_i that best fit our data. The maximum log-likelihood function of the binary response variable is

$$ln(L) = \sum_{i=1}^{N} [ln(1 - p_i) + y_i ln(\frac{p_i}{1-p_i})] \tag{4.8}$$

and the log-likelihood can then be rewritten in the same way as the log-loss function

$$L_{log} = -ln(L) = -\sum_{i=1}^{N} [-ln(1 + e^{(\beta_0+\beta_1 x_i+....+\beta_n x_i)}) + y_i(\beta_0 + \beta_1 x_i + ... + \beta_n x_i)]. \tag{4.9}$$

By using multiple iterations of different values for beta the model optimize the best fit for log odds. When log-odds has been maximized the best parameter estimates has been found. Once the coefficients has been found, conditional probabilities can be calculated (IBM, n.d.(1))

LASSO- and Ridge regularisation

Akalin 2020 highlight that it is possible to limit flexibility of the model and help with performance on unseen data. This is called regularization and introduces bias to the model in order to decrease variance. This is achieved by adding a penalty term to the loss function that shrinks the estimates of the coefficients. This shrinkage can be performed by adding ether the Ridge or LASSO penalty term to the function . They can be expressed as

$$Ridge = L_{log} + \lambda \sum_{j=1}^{p} \beta_j^2 \tag{4.10}$$

and

$$LASSO = L_{log} + \lambda \sum_{j=1}^{p} |\beta_j|. \tag{4.11}$$

The only way to keep the function at its minimum is to assign smaller values to the coefficients. The lambda, or λ parameter controls how much emphasis is given to the penalty term. The larger value of λ the more the coefficients will be pushed towards zero. However, ridge regression coefficients will never reach zero, which means that all variables will have an impact on the output variable. This is called a *dense* solution and is not desirable if we want the model to only select the important variables. Therefore, with a small modification to the equation, we can create a LASSO model. LASSO on the other hand has the ability to shrink coefficients to zero creating what is called a *sparse* solution.

4.2.2 Random forest

Random forest is a supervised machine learning algorithm that combines the output of multiple decision trees as one result (IBM, n.d.(2)), and as such, in order to be able to properly introduce the random forest algorithm one first need to explain decision trees. The section starts to introduce decision trees in general, then explaining Entropy and Information Gain as a way of optimising the trees. Finally the random fores model is introduced.

Decision Trees

A decision tree can be used for both classification and regression tasks. The decision tree is built by using nodes, leaf nodes and branches. Nodes denotes a test on an attribute, branches represents the outcome, and leaf nodes, also known as terminal nodes, classify the outcome (Geeks for Geeks, 2023).

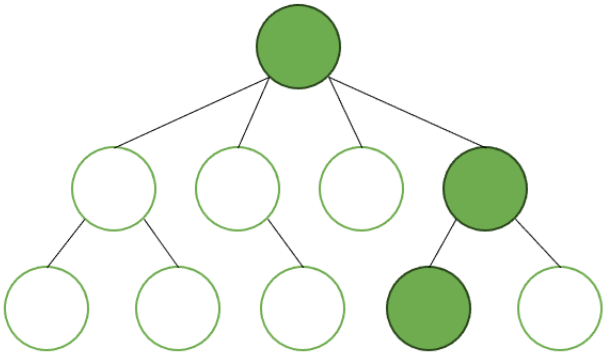

Figure 4.2: Decision Tree

The tree is constructed by repeatedly splitting the training data into subsets until a criterion is met. The criterion could be a maximum tree depth, minimum number of samples required to split a node or when splitting no longer adds value to the predictions. When training the model the algorithm selects the best split using metrics such as entropy or Gini impurity, these measure the level of impurity or randomness in the the subsets. The goal of each split is to create the most homogeneous subset of data and thereby maximizing the information gain. In our case, entropy will be used for each split. Which will be described in the following section (Geeks for Geeks, 2023).

Entropy and Information Gain

Binary cross entropy is another way of saying log-loss, and is used to evaluate the quality of a split at the different nodes in the decision tree. The goal is to minimize the entropy in the resulting subsets of data. Entropy measure the degree of randomness or uncertainty. In the case of classifications, randomness is based on the distribution of class labels within the data set. Entropy is at its lowest value, 0, when the data is completely homogeneous, this indicates no uncertainty in the data set. On other end, entropy reaches its maximum value when the data set is equally divided between the classes, indicating maximum uncertainty (Geeks for Geeks, 2023).

The entropy for a resulting split of the original data set is defined as

$$Entropy = -\frac{1}{N}\sum_{i=1}^{N}[y_i ln p_i + (1 - y_i) ln(1 - p_i)] \tag{4.12}$$

where p_i is the probability of element i in the data(T, 2019).

Tyagi, (2021) expresses that we can calculate information gain by using entropy. Information gain is a metric that describes the decrease in entropy by splitting the node. The feature that maximize the information gain for each split will then be chosen, this will in turn reach the least amount of impurity and the node will be optimized .

$$Information\ Gain = Entropy\ before\ splitting - Entropy\ after\ splitting \tag{4.13}$$

Random Forest

The random forest algorithm is comprised out of multiple decision trees. Each tree is trained with different data that has been selected using random sampling with replacement, called a bootstrap sample. The goal of introducing randomness is to reduce the variance of the model and combat overfitting. This leads to a small increase in bias and some loss of interpretability but an overall boost in performance explained by Akalin (2020).

The algorithm tries to decorrelate the multiple decision tress by giving them different sets of data, by doing it this way, they learn different things. For instance, if one or a few predictor variables are very strong, they will be selected in many of the trees, causing them to be correlated. By randomly sampling the independent variables, we can give the model a chance of learning other features of the data. Bootstrap resampling brings the advantage of using out of the bag (OOB) error prediction. The variables not picked to be trained upon in the model can now be used for estimation. OOB estimation can be used as a substitute to cross-validation estimated errors.

4.3 Results

4.3.1 Training and Cross-Validation

Training and cross-validation was conducted similarly for both the two logistic regression models and the Random Forest model. The models were trained on the training data and then the models predictive capabilities were tested on the test data. This generated confusion matrices that compares the predicted outcomes to the reference, along with various statistics such as our main parameters of interest, accuracy and κ. Additionally in the appendix, plots of model coefficients for LASSO and Ridge was generated, along with an error plot for the Random Forest model. Finally we included tables of variable importance for the Random forest model. Note that the outcome variables *loss* or *win* are defined as the match outcome from the perspective of an arbitrarily chosen team.

Logistic regression

Figures 4.3 and 4.4 show how the cross-validation algorithm optimized the model on the training data, and how different regularization parameters of λ affect the overall accuracy. Tables 7.3 and 7.4 show the optimal λ values for LASSO and Ridge. Note that while λ was optimized to maximise model accuracy, the κ parameter were also included. The optimal λ value on the training data for the Ridge-model was 3.636, while the λ value for the LASSO model was very close to zero, which functionally makes it a conventional logistic regression model.

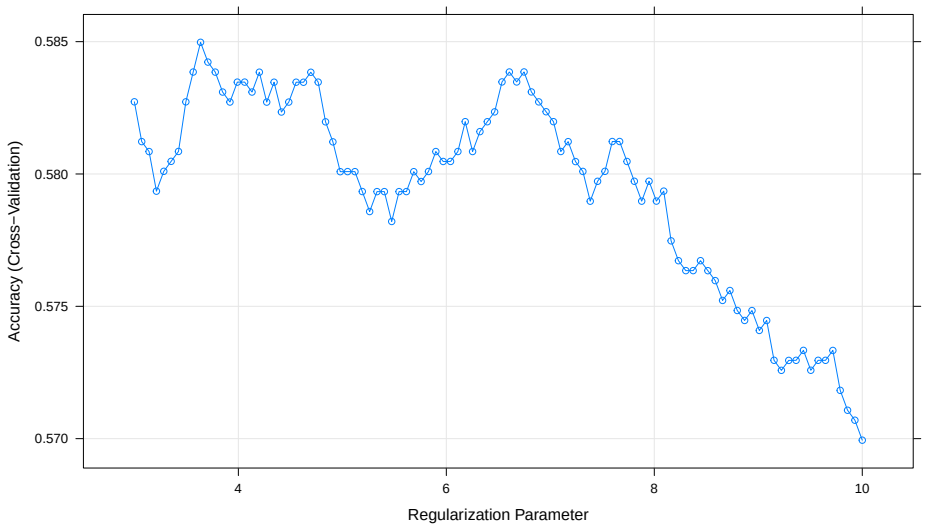

Figure 4.3: Optimal Regularization Parameter λ for Ridge

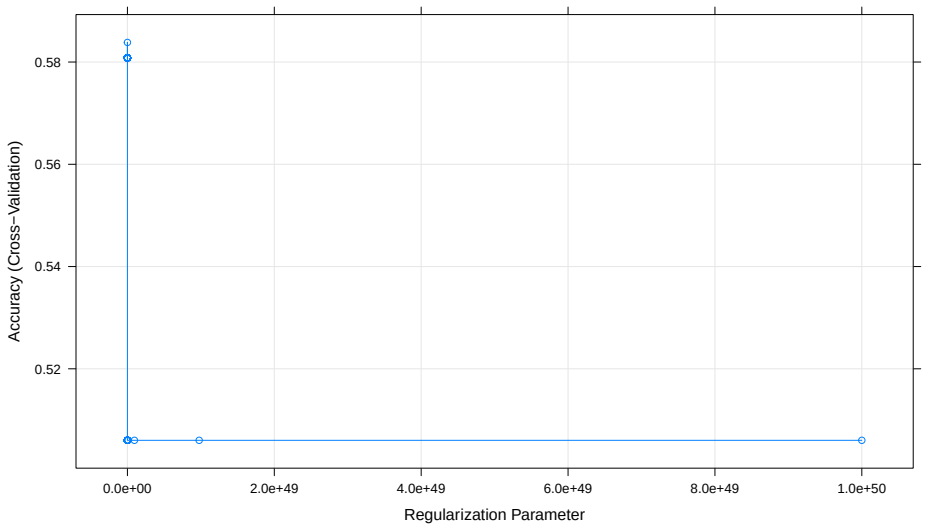

Figure 4.4: Optimal Regularization Parameter λ for LASSO

Tables 7.6 contains the predictions made on the test data and table 4.1 is made up of the key metrics derived from the predictions. From the confusion matrices we find that LASSO made 382 correct win predictions while Ridge made 423, LASSO made more correct loss predictions at 300 compared to Ridges 251. This translates to an estimated accuracy of around 60% for both models with similar confidence intervals that hovered

around $\pm$ 0.06. LASSO however outperformed Ridge slightly both in terms of accuracy and κ-value with 0.21 and 0.19 respectively. Another point of interest is the *No Information Rate* or NIR, which tells us how successful a model would have been if it only predicted the same outcome on all games. NIR also tells us the distribution of the test data set and potential skewness is adjusted by the *balanced accuracy* metric. As both models were trained and tested on the same data sets, they share a NIR of 0.510 which indicates that the data set is to a large degree even. This results in a *balanced accuracy* similar to the original accuracy metric.

Table 4.1: Statistics LASSO and Ridge

Statistic	LASSO	Ridge
Accuracy	0.605	0.598
95% CI	(0.576, 0.633)	(0.567, 0.627)
No Information Rate	0.510	0.510
κ	0.208	0.191
Balanced Accuracy	0.604	0.595

Random Forest

Similar to optimizing the logistic regression, the random forest underwent refinement through 10-fold cross-validation on the training data set. Cross-validation was preformed in order to find the optimal mtry, or number of variables tested at each split. Figure 4.5 and table 7.5 show how accuracy declined when increasing mtry up to four variables per split. Although a slight increase of accuracy and kappa occurred at seven variables, it did not exceed the values achieved at two variables per split. The model where then tested on the test data. As seen in the confusion matrix in table 7.6, the Random Forest model accurately predicted 338 wins and 302 losses and thus achieved an estimated accuracy of close to 57% at a narrow 95% confidence interval range. The model also achieved a κ value of almost 0.14, and with the same data set as the logistic regression model, a *NIR* of 0.51 did not alter the *balanced accuracy* greatly.

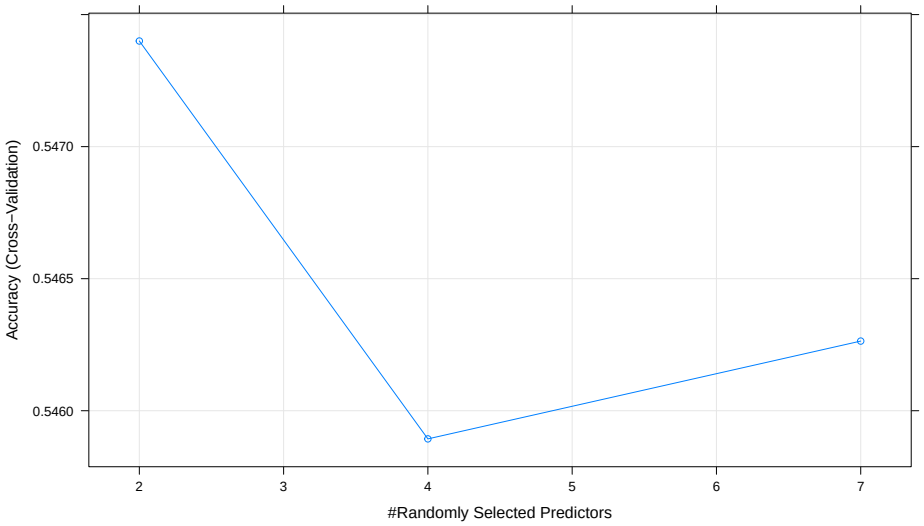

Figure 4.5: Optimal mtry for Random forest

Table 4.2: Statistics Random Forest

Statistic	Random Forest
Accuracy	0.568
95% CI	(0.538, 0.597)
No Information Rate	0.510
κ	0.135
Balanced Accuracy	0.568

4.3.2 BLAST Premier Fall Final 2023

In this section, our finished models were applied to 9 professional games played during the tournament BLAST Premier Fall Final 2023, and the first part show the accuracy achieved by the models, how they classified the data sets and the probabilities assigned to each match outcome. In the second part we compare our models performance to the benchmark.

Accuracy

Table 4.4 shows the prediction accuracy achieved on the BLAST data set. As seen in the confusion matrix, 4.3, all three models correctly predicted five matches but failed to predict the correct outcome of four matches. The results in 4.4 are heavily influenced by the small sample size which relates to a wide confidence interval range for accuracy and a *kappa* value of 0.053. The data set was somewhat skewed with a *No Information Rate* of 0.778 which punished the *balanced accuracy* down to 0.53.

The calculated probabilities of individual match outcomes can be seen in 4.5, while all three models drew the same conclusion on the outcomes of the matches, the calculated probabilities were drastically different between the logistic regression models and the Random Forest model. LASSO and Ridge assigned their outcomes with extreme values of either close to one or close to zero, while the Random Forest model was more restrictive.

Table 4.3: Confusionmatrices for the final predictions of LASSO, Ridge and Random Forest

	LASSO		Ridge		Random Forest	
Prediction / Reference	Loss	Win	Loss	Win	Loss	Win
Loss	1	3	1	3	1	3
Win	1	4	1	4	1	4

Benchmark and log-loss comparison

Below is table 4.6 depicting the average log-loss and accuracy for both bookmakers and our models. Worth to note is the large difference between our models log-loss and accuracy compared to the bookmakers.

Table 4.4: Statistics of the final predictions of LASSO, Ridge and Random Forest

Statistic	LASSO	Ridge	Random Forest
Accuracy	0.556	0.556	0.556
95% CI	(0.212, 0.863)	(0.212, 0.863)	(0.212, 0.863)
No Information Rate	0.778	0.778	0.778
κ	0.053	0.053	0.053
Balanced Accuracy	0.536	0.536	0.536

Table 4.5: Probabilities of the final predicitions of LASSO, Ridge and Random Forest

	LASSO		Ridge		Random Forest	
Game	Loss	Win	Loss	Win	Loss	Win
1	0.000	1.000	0.000	0.999	0.064	0.936
2	1.000	0.000	0.998	0.001	0.568	0.432
3	1.000	0.000	0.988	0.002	0.502	0.498
4	0.999	0.000	0.867	0.133	0.546	0.454
5	0.000	0.999	0.208	0.791	0.372	0.628
6	0.000	1.000	0.007	0.993	0.436	0.564
7	1.000	0.000	0.990	0.009	0.500	0.500
8	0.000	1.000	0.001	0.999	0.332	0.668
9	0.000	1.000	0.000	0.999	0.336	0.664

Table 4.6: Average Log-loss

Model	Average log-loss	Accuracy
Betway	0.322	0.778
THUNDERPICK	0.304	0.778
LOOT BET	0.302	0.778
X1X BET	0.307	0.667
PINNACLE	0.330	0.778
X22BET	0.305	0.667
PARI MATCH	0.299	0.778
N1 BET	0.307	0.778
ROOBET	0.307	0.778
CSGO EMPIRE	0.300	0.778
Ridge	1.756	0.556
LASSO	1.756	0.556
Random Forest	0.232	0.556

Chapter 5

Analysis

We will in the following chapter discuss how successful our models has been in generating accurate predictions on professional game outcomes, this chapters structure is similar to the results section above. We differentiate between the metrics achieved on the training data and the cross-validation process in section 5.1.2 with the predictions made on the tournament in section 5.1.3. In section 5.1.1 we discuss the possible implications of the transition to Counter-Strike 2, and in section 5.1.4 we give our suggestions for future research. In section 5.2 we summarise the discussion and come to a conclusion.

5.1 Discussion

A fundamental consideration in prediction generation and machine learning is understanding the actual distribution of outcomes. This delves into whether the winners of these games achieve victory through luck, randomness, or if there is another underlying factor that determines the winner, perhaps something intangible. If luck or randomness heavily influences game results, it poses a significant challenge for machine learning algorithms to generate good results. This train of thought will be prevalent in the subsequent discussion and shape the conclusion we draw.

5.1.1 Transition to Counter-Strike 2

The Counter-Strike e-sport scene has gone through a transition as the new version of the game was launched. Despite the implementation of some technical adjustments and minor gameplay alterations, we maintain that the core characteristics of the professional players have remained unaltered, and as our variables are designed to measure precisely

these traits, it leads us to conclude that the relevance of our training data remains largely intact.

5.1.2 Training and validation data

The initial assessment of our models were, as previously discussed, performed with the original data set. This data set had been partitioned into thirds. Two thirds were allocated for training the model, and the other third, unseen to the trained model, used for accuracy testing. As shown, we reached a 60.51% accuracy on LASSO, 59.8% on Ridge and a 56.79% on Random forest. The metrics had a statistical significance above 50% accompanied by small confidence-intervals.

While these results might not stand out as particularly impressive, we argue that they demonstrate a degree of predictability within the data. The question whether or not an accuracy of 60% is good hinges on the true distribution and probabilities of game outcomes. Consider a scenario where a team has a true 55% chance of winning, and our model predicts that accurately - in this case, our models would prove highly effective, even if our overall accuracy seems low. Conversely, if a team with a 95% chance of winning is predicted at 55%, it signifies a subpar model. This true probability is of course not available leaving us with the task of estimating it as best as we can.

Another metric of interest is the κ parameter, which was calculated for all of our models. It revolved around 0.2 for our logistic regression models and 0.13 for the Random Forest model, these can only be considered a slight or moderate agreement. However this statistic suffer from the same symptoms as accuracy, as it is limited by the real world distribution of outcomes on the professional level.

In our literature overview, we examined a few related articles and also made a note of their predicted accuracies. As seen in table 2.1, different measures of model effectiveness was used across the literature which makes direct comparisons a challenge. Generally however the results achieved by all models were quite similar, with accuracies and Log-Loss values around 0.6. While our models had lower accuracy than the ones found in the literature, we also achieved the lowest Log-Loss value. The existence of multiple models achieving a similar accuracy of 60% could be evidence that highest level of predictability has been reached. In other word, it might not be possible to generate a better model.

5.1.3 BLAST Premier Fall Final 2023

Our benchmark comparison reached some interesting results. Firstly, nine games is a low number for generating truly accurate and trustworthy metrics. Optimal metric reliability is more likely achieved with a data set comprising hundreds of games.

Despite the potential limitations of accuracy in our metrics and the likelihood of a relatively large standard deviation, we believe valuable insights can be observed from the trends and patterns within the data. The first observation one might make is the difference in accuracy for the bookmakers and our models. The bookmakers consistently outperformed our models with the most common accuracy being 77.8% while we achieved 55.6%. This is quite a significant difference and shows us that our models might not perform as well. However, it is crucial to acknowledge that the small sample size could play a significant role in the accuracy metric, making it premature to definitively label our models as subpar.

The more interesting result, and a result we suspect might persist even if we increase the sample size, is the log-loss metric. Notably, our Ridge and LASSO models exhibited the highest log-loss while having the lowest accuracy. This elevated log-loss is a consequence of the models expressing exceptional confidence in predicting game outcomes, particularly assigning close to a 100% probability to the winning team. We can see this in Table 4.5. Therefore, because their accuracy was poor, the extreme confidence exhibited by these models significantly inflates their log-loss.

In comparing the predicted probabilities of LASSO and Ridge to those of Random Forest and the bookmakers, a distinct difference emerges. Both the bookmakers and the Random Forest model exhibit greater restraint in predicting large probabilities, remaining closer to a 50/50 outcome, something we believe better reflects the true probabilities. And again, even though accuracy for the Random forest model where lower than for the bookmakers, its log-loss remained roughly the same.

5.1.4 Future research

The primary focus of future improvements in our research and model building revolves around the fundamental philosophy of "bad data in, bad data out." First and foremost, the quantity and quality of data is the cornerstone when it comes to accurate predictions within machine learning. Ideally, we would want more extensive and more recent data, especially since our current data set extends only up to 2020. In writing this thesis we have discovered a dissonance between data level and data availability, where low-level

data set are few and far between. This means that we have been unable to fully harness the potential of the virtual arena where every player action is measured, recorded and evaluated as described in the introduction.

Another point of improvement, and a change we think would drastically increase our predictability on upcoming events has to do with how we incorporate player statistics into our models. We are currently utilizing the average stats for a players entire career. This approach fails to capture the sudden change in stats following a player drastically improving or worsening right before a game. To address this limitation, we propose incorporating more recent player performance data, specifically considering the average of the previous five games.

Additionally, the data we currently leverage predominantly reflects individual players core skills, neglecting critical aspects of their environment, such as team composition, coaching ability, and overall team confidence. Something we believe might play a huge part in whether or not a team triumph. And a way to incorporate this into a model as a numerical value, we argue, can be done by using a system such as ELO. This kind of system is mainly used in chess but we believe could be used to simulate a teams environment and their most recent performance. However, it is crucial to acknowledge a counterpoint to this argument — our models successfully predicted the grand final, outperforming benchmark models that did not achieve this feat. This success implies that our more general models may exhibit greater robustness to factors such as hype and other intangibles, challenging the notion that a more detailed environmental simulation is necessary for accurate predictions.

5.2 Conclusion

Firstly, we have been successful in our mission of creating two different machine-learning models trained on professional Counter-Strike data. The question of machine performance is although up for debate. We believe that when it comes to the training data, the models performance was fair, neither exemplary nor subpar. It is essential to note that the efficacy of a model is contingent on real-world probabilities and its ability to accurately simulate them. This underscores the significance of benchmarking as a critical aspect of the evaluation process. It is also important to keep in mind that the achieved accuracy of roughly 60% is similar to the referenced research. This gives evidence to a maximum predictability for professional CS:GO games and that the real-world probabilities might

have been reached.

While the validation results on the test data may be subject to interpretation, upon closer examination of the benchmark, we assert that the performance of our models falls below the desired standard. Again, although the sample size is not enough to derive accurate results, it is enough to give some sort of insight into how our models fare in comparison to the benchmark.

With an accuracy closer to 50/50 it is immediately apparent that we lack some kind of predictability compared to the benchmark. Moreover, the abnormally large log-loss observed in our Ridge and LASSO models, as depicted in Figure 7.5, raises significant concerns. This stems from the models assigning extremely large probabilities to the match outcomes, a representation that appears inconsistent with reality. In a scenario where a model, such as ours, claims 100% certainty about the winner and accurately models real-world probabilities, it would not make incorrect predictions. However, since our model does make incorrect predictions, it deviates from accurately representing real-world probabilities. The random forest model comes closer in what looks like real world probabilities, but as we can see, falls on its ability to make accurate predictions, once again falling short compared to the benchmark.

In conclusion, our models fail to deliver results equivalent to the benchmark, and we do not attribute this discrepancy solely to the large standard error resulting from the small sample size. We are although hopeful, that future research and an implementation of the suggested improvements can create a better and more accurate model.

Chapter 6

References

Akalin, A. (2020). Computational Genomics with r. CRC Press.

Brewer, G., Demediuk, S., Drachen, A., Block, F., & Jackson, T. (2022). Creating Well Calibrated and Refined Win Prediction Models. Available at SSRN 4054211.

Bunker, R., & Susnjak, T. (2022). The application of machine learning techniques for predicting match results in team sport: A review. Journal of Artificial Intelligence Research, 73, 1285-1322.

Cojocariu, I. C. (2022). PREDICTIVE MODELS APPLIED IN SPORTS MANAGEMENT–LITERATURE REVIEW ON RESEARCH TRENDS. Journal of Public Administration, Finance and Law, 11(23), 148-153.

Creative commons (n.d.). CC BY-NC-SA 4.0 LEGAL CODE Attribution-NonCommercial-ShareAlike 4.0 International. Creative Commons. https://creativecommons.org/licenses/by-nc-sa/4.0/

Dembla, G. (2020, November 17). Intuition behind Log-loss score. Towards Data Science. Retrieved November 24, 2023, from https://towardsdatascience.com/intuition-behind-log-loss-score-4e0c9979680a

Ferguson, M. (2018, September 19). Esports Essentials: The Impact of the Counter-Strike Majors. The Esports Observer. Retrieved October 12, 2023, from

https://archive.esportsobserver.com/esports-essentials-counter-strike-majors/

Hodge, V. J., Devlin, S., Sephton, N., Block, F., Cowling, P. I., & Drachen, A. (2019). Win prediction in multiplayer esports: Live professional match prediction. IEEE Transactions on Games. 13(4), 368-379.

IBM, (n.d.) (1). What is logistic regression? IBM. https://www.ibm.com/topics/logistic-regression

IBM, (n.d.) (2). What is random forest? IBM. https://www.ibm.com/topics/random-forest

Llewellyn, T. (2018, September 17). THE RISE OF AN ESPORTS PHENOMENON. Science and Media Museum. Retrieved October 12, 2023, from https://blog.scienceandmediamuseum.org.uk/counter-strike-esports/

Makarov, I., Savostyanov, D., Litvyakov, B., & Ignatov, D. I. (2018). Predicting winning team and probabilistic ratings in "Dota 2" and "Counter-Strike: Global Offensive" video games. In Analysis of Images, Social Networks and Texts: 6th International Conference, AIST 2017, Moscow, Russia, July 27–29, 2017, Revised Selected Papers 6 (pp. 183-196). Springer International Publishing.

McHugh, M. L. (2012). Interrater reliability: the kappa statistic. Biochemia medica, 22(3), 276-282.

Refaeilzadeh, P., Tang, L., & Liu, H. (2009). Cross-validation. Encyclopedia of database systems, 532-538.

Švec, O. (2022). Predicting Counter-Strike Game Outcomes with Machine Learning.

Stanton, R. (2023, October 27). Exclusive interview: Valve on the future of Counter-Strike 2. Retrieved October 13, 2023, from https://www.pcgamer.com/counter-strike-2-interview/

Stubbs, M. (2023, March 5). 'Counter-Strike 2' Reportedly Launching This Month.

Forbes.com. Retrieved October 12, 2023, from https://www.forbes.com/sites/mikestubbs/2023/03/05/counter-strike-2-reportedly-launching-this-month/

Sohail, S. (2023, September 10). The Math Behind Betting Odds and Gambling. Investopedia. https://www.investopedia.com/articles/dictionary/042215/understand-math-behind-betting-odds-gambling.asp

T, S. (2019, January 11). Entropy: How Decision Trees Make Decisions. Towards Data Science. https://towardsdatascience.com/entropy-how-decision-trees-make-decisions-2946b9c18c8

Tyagi, N. (2021, March 22). What is Information Gain and Gini Index in Decision Trees? Analytics Steps. https://www.analyticssteps.com/blogs/what-gini-index-and-information-gain-decision-trees

Tillman, J. (2023, September 17). IMPLIED PROBABILITY - WHAT DOES IT MEAN FOR NOVICE BETTORS. LINES. https://www.lines.com/guides/what-is-implied-probability-betting/1550

Wilkens, S. (2021). Sports prediction and betting models in the machine learning age: The case of tennis. Journal of Sports Analytics, 7(2), 99-117.

Wunderlich, F., & Memmert, D. (2021). Forecasting the outcomes of sports events: A review. European Journal of Sport Science, 21(7), 944-957.

Xenopoulos, P., Doraiswamy, H., & Silva, C. (2020). Valuing player actions in Counter-Strike: Global Offensive. In 2020 IEEE international conference on big data (big data) (pp. 1283-1292). IEEE.

Chapter 7

Appendix

7.1 Descriptive statistics

Table 7.1: Descriptive statistics part 1

	avg rating diff	avg impact diff	avg kdr diff	avg adr diff
Min.	-0.250	-0.324	-0.366	-0.205
1st Qu.	-0.036	-0.036	-0.052	-0.019
Median	0.004	0.002	0.004	0.002
Mean	0.003	0.003	0.006	0.002
3rd Qu.	0.042	0.042	0.062	0.023
Max.	0.250	0.282	0.366	0.158

Table 7.2: Descriptive statistics part 2

	avg apr diff	avg dpr diff	avg spr diff
Min.	-0.046	-0.134	-0.090
1st Qu.	-0.008	-0.022	-0.008
Median	0.00	-0.002	0.000
Mean	0.001	-0.002	-0.001
3rd Qu.	0.010	0.018	0.008
Max.	0.050	0.134	0.070

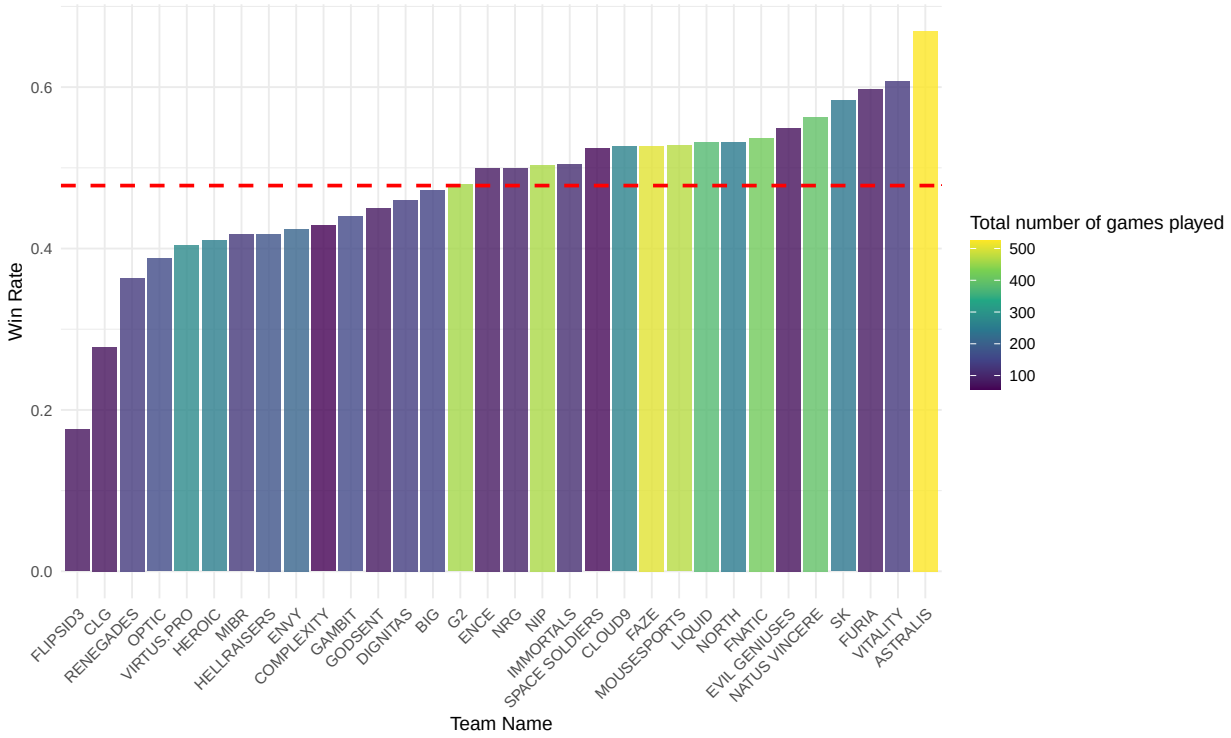

Figure 7.1: Win rate of the teams in our data set with more than 50 games played, red dotted line is average win rate

7.2 Results

7.2.1 Cross-validation

Table 7.3: Ridge-regression λ optimisation

λ	Accuracy	κ
...	...	...
3.566	0.584	0.166
3.636	0.585	0.168
3.707	0.584	0.166
...	...	...

Table 7.4: LASSO-regression λ optimisation

λ	Accuracy	κ
...	...	...
-0.3030	0.581	0.161
-0.101	0.581	0.161
0.101	0.5060	0.000
...	...	...

Table 7.5: Random Forest Cross-Validation

mtry	Accuracy	κ
2	0.547	0.0946
4	0.546	0.091
7	0.546	0.092

7.2.2 Coefficient plots for LASSO and Ridge

Figures 7.2 and 7.3 shows the regularization process of the LASSO and Ridge models. The y-axis shows the magnitude of the features coefficients with $\log(\lambda)$ along the bottom x-axis. A notable difference between the models is that while LASSO continually excludes

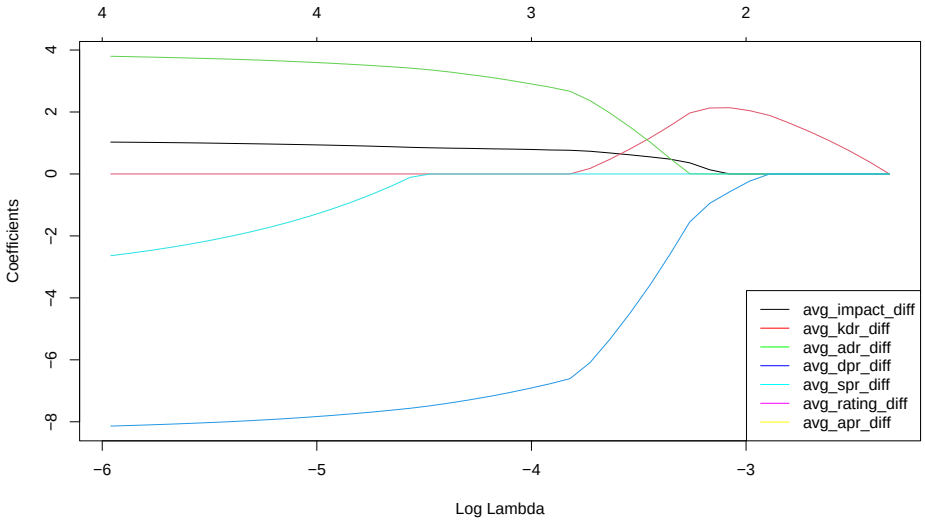

Figure 7.2: Coefficient plot of the LASSO model

features as lambda increases, the ridge model keeps all seven features (as seen on the x-axis above the plots). Note that $\log(\lambda_{optimal})$ for the ridge model was ≈ 1.3 while $\lambda_{optimal} = 0$.

7.2.3 Confusion matricies

Table 7.6: Confusion matrices for LASSO, Ridge and Random forest models

	LASSO		Ridge		Random Forest	
Prediction / Reference	Loss	Win	Loss	Win	Loss	Win
Loss	300	193	251	152	302	237
Win	252	382	301	423	250	338

7.2.4 Random Forest

Gini Impurity

Gini Impurity evaluates the splits accuracy among the classified groups. The Gini Impurity score range between 0 and 1. Zero when all observations belong in the same class, and 1 when the elements are randomly distributed within the classes. The goal is to have a

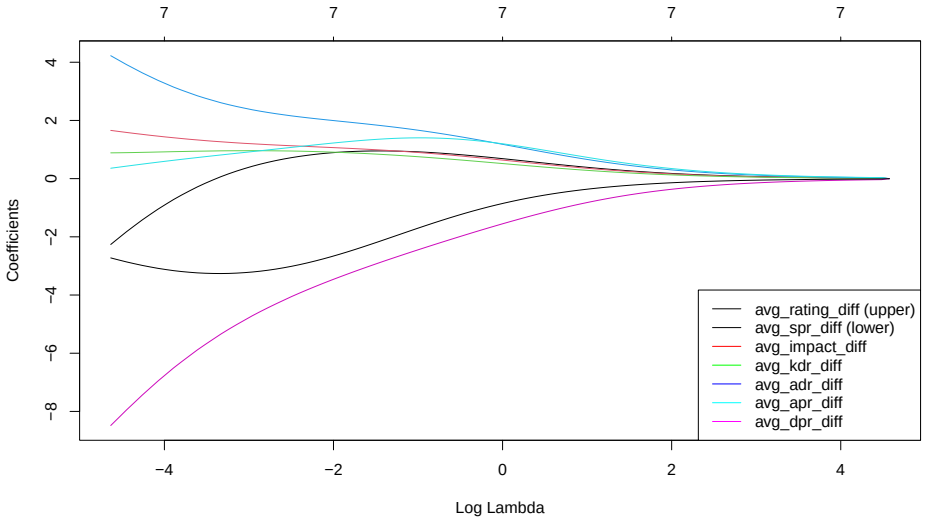

Figure 7.3: Coefficient plot of the Ridge model

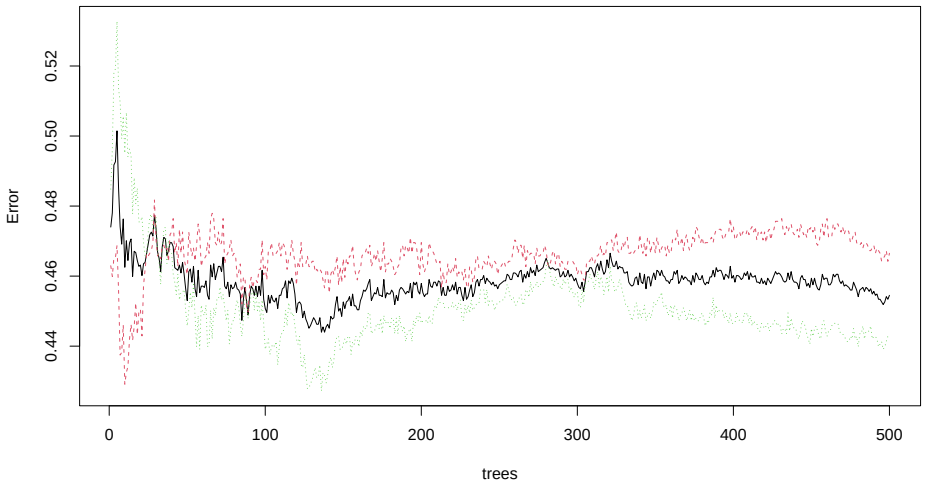

Figure 7.4: Final Random Forest model

Gini index score as low as possible. In our case, we will be using mean gini decrease as a measure of variable importance for the Random Forest model (Geeks for Geeks, 2023).

$$Gini\ Impurity = 1 - \sum p_i^2 \tag{7.1}$$

p_i is represents the proportions of elements in the set that belongs to the i category.

Table 7.7 describe the importance of the variables used in the Random Forest model, defined by their *Mean Gini Decrease*. Average damage per round achieved the highest decrease and is thus defined as most important, other important variables were KDR, Impact and Rating. In the lower echelons of importance we find Average number of assists on teammates kills (APR) and Average number of saved teammates per round (SPR). Average number of deaths per round had the overall lowest Gini decrease and was therefore considered the least important variable.

Table 7.7: Variable importance for Random Forest

Variable	Mean Gini Decrease
avg adr diff	222.052
avg kdr diff	198.466
avg impact diff	196.138
avg rating diff	191.979
avg apr diff	179.347
avg spr diff	175.516
avg dpr diff	162.998

7.2.5 Model comparison

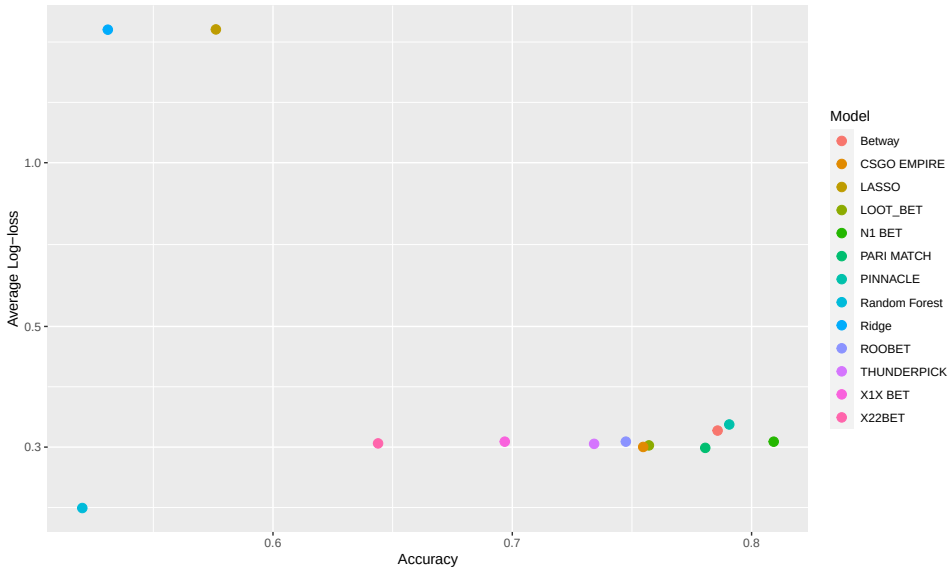

Figure 7.5: Log-loss and accuracy per model (with a small jiggle-effect on the x-axis and a logarithmic y-xis)