TILBURG UNIVERSITY

GAME, SET, AND BET?

A STRUCTURAL ANALYSIS OF MACHINE LEARNING FOR MALE TENNIS PREDICTION

THOMAS WILLEKES

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY
AT THE SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
OF TILBURG UNIVERSITY

STUDENT NUMBER

2085959

COMMITTEE

dr. Drew Hendrickson
dr. Elisabeth Huis in 't Veld

LOCATION

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science &
Artificial Intelligence
Tilburg, The Netherlands

DATE

June 24, 2022

ACKNOWLEDGMENTS

First and foremost, I would like to sincerely thank Dr. Drew Hendrickson for helping me through the final part of my Master. His enthusiasm and empathetic demeanour motivated and inspired me throughout the Thesis. Moreover, I would like to thank Jeff Sackmann and the people of http://www.tennis-data.co.uk/ for having a high quality open source database. Finally, I would like to thank my thesis group for giving valuable feedback and input.

GAME, SET, AND BET?

A STRUCTURAL ANALYSIS OF MACHINE LEARNING FOR MALE TENNIS PREDICTION

THOMAS WILLEKES

(8449 words)

Abstract

Forecasting sports outcomes is a fundamental occupation of all that follow and/or analyse a sport. Tennis is no exception in this regard. However, machine learning/deep learning is relatively novel in this domain. Due to this novelty, a strong benchmark concerning features, algorithms, and time period seems to lack. Hence, we ask ourselves the question how various machine learning/deep learning algorithms utilize point statistics, player rankings, contextual information, and betting odds when predicting ATP matches. The performance is compared to the state-of-the-art Elo model and a simplified bookmaker consensus model. The construction of the features is enabled because of the data sources from Jeff Sackmann and tennis-data.co.uk. Through the unique combination of the out-of-sample time period, the number of features used, and the number of algorithms used, this thesis contributes by constructing a structural review of all important dimensions in this domain. It is found that the multilayer perceptron has the best overall performance, with the betting odds as the main driver of this performance. Furthermore, the random forest closely follows the multilayer perceptron in terms of performance and performs the most consistent when ranking-based subsets are investigated. Overall, the state-of-the-art statistical baseline is beaten by all trained models, indicating a strong argument for using machine learning in this domain.

DATA SOURCE/CODE/ETHICS

This thesis did not involve collecting data from human participants or animals. The author of this thesis does not claim any ownership of the data obtained from Jeff Sackmann or tennis-data.co.uk. All code used in this thesis was generated by the author itself, albeit with inspiration from Stack Overflow. However, this inspiration did not include outright copying of code from Stack Overflow or any other sources. The code in this thesis is available under the following link: https://github.com/ThWillekes/MasterThesisTW, however only for the supervisors of this thesis or upon request. All images used in this thesis are produced by the author itself.

1 INTRODUCTION

Forecasting sport outcomes is a fundamental occupation of all that follow and/or analyse a sport. Sports analysts use it in their preview, fans have broad discussions about who is going to win, and athletes and coaches use knowledge about forecasting to adjust strategy and their training. Moreover, researchers and hobbyists have extensively tried to use statistics and machine learning in sports such as football and cricket to get an edge over the betting markets (Goddard & Asimakopoulos, 2004; Kampakis & Thomas, 2015). This does not restrict itself to just predicting the outcome, but also predicting the goal difference (Wheatcroft, 2020). Tennis is no exception in this regard.

However, predicting tennis matches and generating a profitable strategy is rather difficult. Kovalchik (2016) in her extensive overview of statistical models, notes that all approaches fail to beat the bookmaker consensus model by Leitner, Zeileis, and Hornik (2009). Moreover, Wilkens (2021) in an analogous overview only mentions eight papers that base their approach on machine learning. Furthermore, she tests five algorithms and could not include a conclusive answer as to which algorithm suits tennis prediction best. These findings indicate two interesting properties in tennis prediction. First, machine learning is still a relatively novel approach and a conclusive answer regarding algorithm usage is absent. Second, the failure to consistently achieve good results or beat the bookmakers, indicates that a strong foundation/benchmark is missing. The second property is in line with the fact that tennis prediction models achieve consistently better performance for higher-ranked - than for lower-ranked players (Candila & Palazzo, 2020; Kovalchik, 2016). A combination of a robust investigation of feature importance with a good performing model could potentially unravel the discrepancy of performance between these groups and contribute to a stronger argument for the usage of machine learning/deep learning.

Hence, the goal of this thesis is twofold. In the first place, we draw upon the most used and important features found in the literature and try to improve the predictive performance of male tennis matches upon a statistical state-of-the-art model and a simplified interpretation of the bookmaker consensus model of Leitner et al. (2009). The process of improvement involves the comparison of several algorithms. Subsequently, we investigate the best-performing model using SHAPley values as a robust measure of feature importance. This leads to the following main question:

MQ *How do various machine learning/deep learning models utilize contextual information, betting odds, player rankings, and point statistics while being compared to a set of baselines to predict the outcome of ATP tennis matches between 2005 and 2021?*

To answer the main question, the following subquestions are derived and necessarily need to be answered:

SQ1 *To what extent do a random forest, a multilayer perceptron (MLP), and a support vector machine (SVM) affect the predictive performance in terms of return on investment, log loss, and accuracy?*

SQ2 *How does the predictive performance vary across player ranking-based categorized groups?*

SQ3 *According to SHAPley values, which features of the best-performing model are of the greatest importance for all players, and which features are of the greatest importance for the player ranking-based groups?*

The scope of the research and the feature importance used, make the approach unique. To the best of our knowledge, this is the first study that uses an out-of-sample dataset with such a wide time frame, the feature set used is one of the biggest (only second to Candila and Palazzo (2020)), and this is the first study that incorporates the SHAPley values in this domain. Accordingly, it is believed that such a systematic investigation can act as the foundation for improvements in future research. Furthermore, it is believed that the knowledge generated could exploit potential inefficiencies in the betting market. Unravelling inefficiencies in the betting market is a signal for the bookmakers to adjust their methods and steer towards an efficient betting market. It has already been noted by Leitner et al. (2009) that the bookmakers do not always act in their own best interest.

We find that the overal performance of the MLP is best, however, this depends on which risk scenario is taken. The SVM lends itself better to more conservative strategies. The insight that different algorithms function dissimilar based on the risk appetite, opens a gap for future research where the risk factor is a hyper-parameter as well. The random forest performed almost as good as the MLP in terms of log loss and accuracy, however, it had only half the ROI in what was considered to be the most robust strategy for both models. Although the overall predictions of the random forest were not as good as the MLP, it was more consistent when the metrics were grouped by ranking. Furthermore, the feature importance of the MLP was constructed and the betting odds were almost exclusively responsible for its performance. It is assumed that bookmakers use similar information as is captured by the constructed features.

2 RELATED WORK

Before the related work is investigated, the scope of tennis match prediction has to be demarcated. A tennis match has the property of being a zero-sum game. It means that independent of any occurrences there is always a loser and a winner. Thus, the outcome of this situation is binary with no other possibilities. This study focuses on the binary outcome of a tennis match and so do the investigated papers in this section. Therefore, the prediction can either be a probability of win or lose or it can be a dummy that indicates win or lose.

2.1 *Early Work*

Although there are notable differences, statistical modeling lays the foundation of the features used in machine learning/deep learning approaches. Kovalchik (2016) in her comparison of statistical approaches categorizes previous research based on the type of models, but implicitly also features used. She distinguishes paired comparison - , Point-based - , and Regression-based models. Paired comparison models calculate a probability of a player winning based on the relative strength of its opponent. A famous and relatively simple method is the Elo approach which was first used to rate chess players Elo (1978). The Elo approach achieved the best overall result and has gained further traction through the adjustment of Angelini, Candila, and De Angelis (2022). To construct the needed rankings of the Elo method, they weight the ranking by how many games were won in the previous match. Point-based models assume that winning a point is independent and identically distributed. Given this logic, the probability of winning a match can be derived from the probability of winning a point. Lastly, Regression-based models use a-priori information about the players, their opponent, and match characteristics to predict the outcome.

The aforementioned methods/models have all proven their worth in tennis prediction and have provided future research with useful features.

The focus of statistical approaches on only one category stems from the fact that statistics is built around inference. Including more variables complicates models and inference becomes increasingly difficult (Bzdok, Altman, & Krzywinski, 2018). The use of deep learning/machine learning eases this restriction and makes it possible to model more complex relationships through the usage of a wider scope of features.

2.2 *Machine Learning/Deep Learning approaches*

Wilkens (2021) in her comparison of five machine learning algorithms gives an overview of machine learning studies for predicting tennis matches. Eight approaches have been investigated. Two of the eight studies were not published. Gosh, Sadhu, Biswas, Sarkar, and Sarkar (2019) report an average accuracy of their decision tree model of 99.14%, however, it seems they use results of the match they are predicting as predictors (e.g. number of games won), consequently, creating information leakage. The five remaining approaches all utilize the strength of machine learning/deep learning and include at least two of the three categories found in statistical studies (Candila & Palazzo, 2020; Cornman, Spellman, & Wright, 2017; Gao & Kowalczyk, 2019; Sipko & Knottenbelt, 2015; Somboonphokkaphan, Phimoltares, & Lursinsap, 2009).

Although the first to use an artificial neural network (ANN), the focus of Somboonphokkaphan et al. (2009) lies on grand slam tournaments only. Furthermore, a comparison of models is based solely on accuracy. Despite the fact that correct predictions are important, avoiding high confidence in incorrect predictions is deemed to be of higher importance. However, they do show promising results by outperforming previous statistical approaches.

An assessment of the five remaining approaches (including Wilkens (2021)) has been carried out and aside from the features used, the performance of a model depends on different scopes. First, the confidence/threshold needed to bet on a match affects the number of bets placed and the ROI. Generally, the higher the constraint of needed confidence in a bet, the higher the ROI and the lower the number of bets placed (Candila & Palazzo, 2020; Sipko & Knottenbelt, 2015). Second, the scope of the out-of-sample forecasts influences the predictive performance of a model. When using a subset of all matches with the constraint that at least one high-skilled player is involved, predictions tend to be more precise (Candila & Palazzo, 2020; Cornman et al., 2017; Kovalchik, 2016). Thirdly, the strategy (closely linked to the confidence) highly affects the ROI. The strategy can be seen from two perspectives. One where the capital of the bettor is disregarded and the prime objective is to maximize the ROI based on a set threshold of confidence. The kelly criterion optimizes the strategy based on the bettor's capital (Kelly Jr., 1956). This criterion provides solid results in two studies and takes into account real-life situations (Sipko & Knottenbelt, 2015; Wilkens, 2021). Lastly, the different distribution of probability outputs that algorithms have, has a significant influence on getting an edge over the bookmakers (Gao & Kowalczyk, 2019).

In regard to algorithm usage, Wilkens (2021) is considered to be the benchmark. She finds no notable differences between algorithms. However, it is to be noted that her feature set is scarce in comparison to the other studies. In addition to her findings, the other studies report conflicting messages. Although conflicting, the MLP, random forest, and the SVM are consistently used and achieve good results (Cornman et al., 2017; Gao & Kowalczyk, 2019; Sipko & Knottenbelt, 2015)

One common thread throughout all investigated studies is the use of conventional measures of feature importance. This leads to varying reports of what the most important feature is. Gao and Kowalczyk (2019) & Sipko and Knottenbelt (2015) identify the serve strength of a player as most important, Wilkens (2021) & Cornman et al. (2017) identify the betting odds as most important, and Candila and Palazzo (2020) report the paired comparison feature to be of the highest importance.

2.3 *Feature Importance*

Knowing if an action has the desired effect is the backbone of solid decision-making. Analogous, an understanding of the functioning of a model, can be valuable for improving it (Arrieta et al., 2019). Intending to tackle the problem of explainable artificial intelligence (XAI), Lundberg and Lee (2017) developed a local agnostic method called Shapley Additive exPlanations (SHAP). SHAP stands out among other feature importance techniques, because of three useful properties. Firstly, SHAP calculates the feature importance per prediction whereas global methods do not. Thus, offering more in-depth information about the varying influence of differing magnitudes of a feature. Secondly, using human explanation as a reference, Lundberg and Lee (2017) compare SHAP to other local explanation methods and find that SHAP is better aligned with human understanding. Finally, as opposed to other feature importance methods, SHAP ensures local accuracy. For example, permutation importance relies on randomly shuffling the feature. Each time a permutation importance iteration is repeated, the importance will likely vary.

It is thought that SHAP could be a solution to the conflicting reports of feature importance.

2.4 *Literary Gap*

Based on the literary research, the following gap can be identified. First of all, machine learning is relatively novel in this domain. It is believed that this is the cause of conflicting results as to which algorithm and which features are most important. Furthermore, none of the studies have compared themselves to the statistical state-of-the-art Elo model as identified by Kovalchik (2016). Hence, this thesis addresses these points by doing the following:

1. investigate three of the most used and successful classification algorithms and compare them to the state-of-the-art statistical model.
2. utilize the most used/best-performing features found in the literature,
3. establish which of these features are most important through SHAPley values.

3 METHODS

3.1 *Algorithms*

The algorithms are chosen based on the classification of Arrieta et al. (2019). Generally, all algorithms are considered to be more blackbox than transparent. Although given the aforementioned similarity, the workings of each algorithm is profoundly different. Hence, two important criteria are combined. First, blackbox algorithms are chosen based on their association with better performance. Second, choosing these algorithms provides a wide variety of algorithms with different workings. Due to the fact that it is often difficult to determine a-priori which algorithm fits a problem best, this approach is deemed appropriate.

3.1.1 *Random Forest*

The first classification algorithm that will be used is a random forest. A random forest is a non-parametric ensemble method with useful properties that make it a great classifier. For each tree, a random sample of the total dataset is taken with replacement and at each node, only a random subset of all features is considered (Biau & Scornet, 2016). This process is based on an optimization of the balance between the individual strength of a tree and the decorrelation of individual trees (Breiman, 2001). A well-implemented hyper-parameter tuning process should guide the optimization.

Of interest for the hyper-parameter tuning process is the depth of an individual tree, the number of features per split, the number of observations required for a split, and the number of observations required at a leaf node. For a classification model it is advised to set the number of features per split to a logarithmic scale base two of all features (Biau & Scornet, 2016). This setting prevents trees from always splitting on the strongest feature. The remaining hyper-parameters are used to fine-tune the depth of a tree.

The reduction of overfitting in comparison to the decision tree and its empirical success over other Machine Learning/Deep Learning methods in different domains such as accident detection in traffic and brain stroke classification (Dogru & Subasi, 2018; Subudhi, Dash, & Sabut, 2020), but also tennis prediction (Gao & Kowalczyk, 2019), make the random forest a more than suitable algorithm.

3.1.2 *Support Vector Machine*

An SVM is dubbed for its discriminative power, strong theoretical foundation, and great generalization capabilities (Cervantes, Garcia-Lamont, Rodríguez-Mazahua, & Lopez, 2020). It achieves this by fitting a hyperplane that maximizes the margin between the hyperplane and the classes. As with all algorithms, there is a trade-off between bias and variance. For the SVM, this trade-off is optimized by controlling the margin between the hyperplane and the classes. The regularization parameter C adjusts the margin, consequently, controlling the number of misclassified instances when training.

Furthermore, not all problems are linearly separable. The SVM counters this problem by mapping the instances to a higher dimension using the kernel trick. Previous studies have shown that the linear kernel performed best (Cornman et al., 2017; Wilkens, 2021). The radial basis function (RBF) is dubbed the most reasonable choice (e.g. standard setting of the Scikit Learn library). Thus, both the linear - and RBF kernel are investigated. When using the RBF kernel, the curvature of the hyperplane becomes another setting that influences the performance of an SVM.

A caveat of the RBF kernel is its computation time. The RBF kernel depends on the number of observations used. As a result, computation time grows exponentially/quadratic (depending on the regularization value) when including more observations. Hence, it is often advised to sample data instead of using the entire dataset.

3.1.3 *Feedforward Artificial Neural Network*

An artificial neural network (ANN) is constructed to simulate the learning process we as humans are capable of. By doing so ANN's are suited to model complex non-linear relationships. It is shown by Candila and Palazzo (2020) based on the linearity test of Teräsvirta, Chien-Fu, and Clive (1993) that the relationship to be modeled is indeed strongly non-linear.

The ANN this thesis uses is called a multilayer perceptron (MLP). An MLP is a feedforward ANN. The architecture of an ANN consists of neurons that make up the different layers, which in the case of an MLP consist of an input -, a hidden layer(s), and an output layer.

The output of a node is the transformed weighted sum of the input. This transformation is done through activation functions. Four activation functions will be investigated namely, Tanh, ReLU, LeakyReLU, and Sigmoid. The Sigmoid activation will be used for the output layer as this outputs a probability which is needed for the evaluation. However, the Sigmoid activation function causes the gradient to vanish when used in the hidden layer. This causes slow learning which the ReLU activation avoids and the Tanh reduces. However, the ReLU activation faces the problem of dead neurons due to the gradient of negative inputs being zero. Maas, Hannun, and Ng (2013) addressed this problem with their adaption of the ReLU activation, the LeakyReLU. The LeakyReLU has a small slope for negative values hence, avoiding the problem of dying nodes.

Through backpropagation the MLP learns the correct weights. In essence, backpropagation is the assessment of the prediction of one iteration. After assessing how far off the prediction is, the weights are adjusted depending on the learning rate settings and the chosen optimizer. The Adam optimizer will be used to find the correct weights. This optimizer combines the useful characteristics of the RMSProp and AdaGrad (Kingma & Ba, 2014).

The flexibility an MLP has is also a cause for concern. Considering that an MLP can learn complex patterns from a dataset, its predictions can become too specific and the learned patterns hard to generalize. Common solutions to overfitting include early stopping of the number of iterations, regularizing the learned weights, and randomly dropping nodes.

3.2 *Baselines*

Two baselines are computed. The predicted probability of the Elo method is the first baseline. The Elo probability mentioned by Kovalchik (2016) can be calculated as follows:

$$\pi_{i_j}(t) = (1 + 10^{\frac{(E_j(t) - E_i(t)))}{400}})^{-1} \tag{1}$$

Here π_{i_j} denotes the probability that player i wins against player j at time t. E_j and E_i are the Elo points and denote a player its capability. The calculation of the Elo points for player i resulting from the match at time t is given by the following equation:

$$E_i(t+1) = E_j(t) + K_i(t)(\pi_{i_j}(t) - W_i(t)) \tag{2}$$

K is a player dependent scale factor that can be used as a way to account for certain properties (e.g how many matches a player has played). The approach of Angelini et al. (2022) is followed.[1] W_i is an indicator if player i won or not.

Furthermore, the average bookmaker's odds are used as a measure of consensus. The logic behind a prediction of this baseline is as follows: if the average odds of all bookmakers for player i in match j is lower than the average odds of all bookmakers of its opposing player, then player i is predicted to win. Therefore, only binary

[1] $K_i(t) = \frac{250}{(N_i(t)+5)^{0.5}}$ $N_i(t)$ denotes the number of matches of player i at time t

predictions are obtained. This leads to an adjustment of the betting strategy. In this case, a bet is placed on all players that are believed to win. Throughout the thesis this baseline will be referred to as the consensus baseline.

3.3 *Evaluation*

To assess the model its generalization abilities, out-of-sample forecasts and evaluation metrics are needed. The out-of-sample observations are created through a stratified k-fold cross-validation and a stratified hold-out validation. The strategy to be taken is to create a separate train -, validation - , and test set for each fold. The initial dataset is split based on a 25% to 75% proportion for the test - and train set respectfully. The train set for each fold is then split in the same fashion to create the validation set.

The training set is used to optimize the hyper-parameters based on the log loss score on the validation set. Bergstra and Bengio (2012) find that to achieve equal performance, the random search needs just a fraction of the iterations a grid search needs. Given that a great reduction in the iterations is achieved, a randomized search better addresses the optimization problem of the curse of dimensionality and being stuck in local optima. Hence, randomized search is preferred over the grid search.

The predictions on the out-of-sample observations are assessed through the computation of the log loss score, accuracy score, and return on investment (ROI). Log loss is used as it is said to converge to the Kelly criterion (Kovalchik, 2016) and because it penalizes high deviations from the target. This in turn is advantageous for the ROI. The ROI of one match can be explained by the following equation:

$$ROI_{i_j} = \frac{o_{i_j} * s_j}{s_j} * 100\% \tag{3}$$

In this case i stands for the player and j for the match. The odd o is multiplied by the stake s placed and by dividing it through the placed stake s and multiplying it by 100%, the ROI of one placed bet is calculated. The ROI of a model is the mean of all matches and is dependent on the model its performance and the strategy used. Based on the confidence in an outcome and the betting odds, a strategy tells us when to bet on a match. The used strategy can be shown by the following simple rule $\frac{1}{p_{i_j}} * r < o_{i_j}$, whereas p is the predicted probability of player i in match j, r is a risk factor, and o denotes the betting odd of player i in match j. Setting r higher denotes a risk-averse strategy, a lower setting of r would be a risk-friendly strategy. If a player is predicted to win with a probability of 0.5 and r is 1 (neutral strategy), a bet would be placed if the odd offered on the market is higher than 2. The risk factor of 1, 1.25, and 1.5 will be used to indicate different risk appetites. Accuracy is used to investigate the difference between just predicting correctly and predicting correctly with high confidence. Hence, the comparison of accuracy with log loss and ROI gives better insight into the functioning of a model. Furthermore, the second baseline does not provide a prediction of the probability. Thus, accuracy is needed as an evaluation metric.

Moreover, the consensus in the literature is that the performance of predictions on matches with higher-ranked players is higher. Hence, the evaluation is also done on three different subsets. On top of the evaluation of the entire out-of-sample subsets, the performance metrics are also grouped by player rankings. The categorization of the three subsets are a player ranking > 30, 30 > player ranking >= 50, or player ranking < 50. Only log loss and accuracy are evaluated on the subsets as it is suspected that only few bets on the top 30 subset will be placed. Consequently, the ROI of these subsets is hard to compare.

The models will be compared based on their overall performance. Afterward, the best-performing model will be further evaluated with the SHAP approach.

4 EXPERIMENTAL SETUP

In the following section the experimental setup will be explained. On top of the textual explanation, a flowchart with a visualization of the process can be found in Figure 1.

4.1 *Software*

All data handling and model building is done in Jupyter Notebook (Kluyver et al., 2016) using Python version 3.9.7 (Van Rossum & Drake Jr, 1995). The model building is done using Scikit Learn (Pedregosa et al., 2011) and Keras (Chollet et al., 2015). Data processing, data cleaning, and feature engineering are done using Pandas (McKinney et al., 2010) and Numpy (Harris et al., 2020). Finally, visualizations are made using Seaborn (Waskom, 2021), Matplotlib (Hunter, 2007), and SHAP (Lundberg & Lee, 2017).

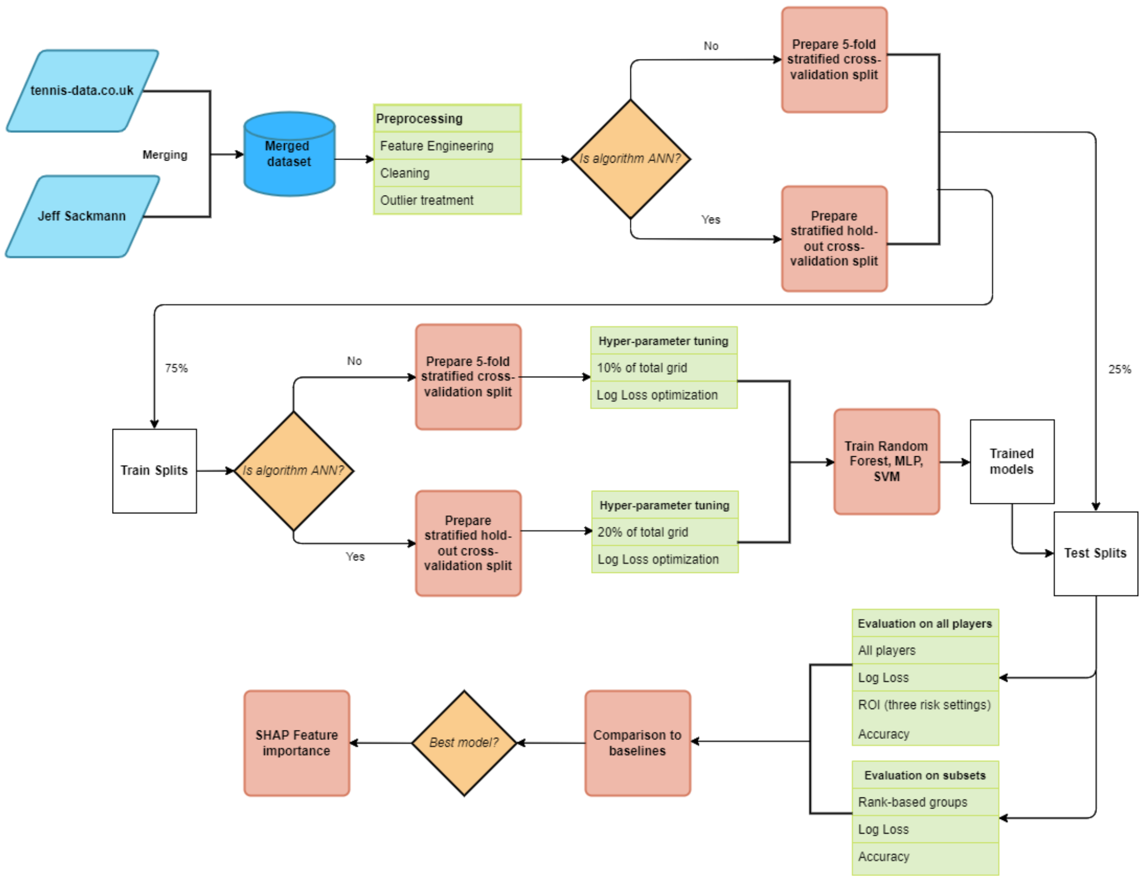
4.2 *Data Structure, Preprocessing, and Cleaning*

4.2.1 *Merging*

To conduct the research two data sources were utilized. The first source stems from the GitHub of Jeff Sackmann.[2] He captured point statistics, outcomes, tournament information, rankings, and the length of a match of 62.515 matches from 2001 to 2021. The second source is tennis-data.co.uk which provides an overlap in statistics of 55.700 matches from 2001 to 2021, but also contains betting odds of various betting agencies for each player. By combining these sources 45.299 unique matches were obtained. After removing useless and duplicate columns, 71 columns remain. A table with all columns and an explanation of their meaning can be found in Table A.1.

[2] https://github.com/JeffSackmann/tennis_atp

Figure 1:
Flowchart of the experimental setup of the thesis



4.2.2 *Missing Values*

There were 1.642 matches which are not finished hence, they were removed from the dataset. Furthermore, there were 745 matches where no odds were recorded, however, 610 were played before 2005. The missing values before 2005 did not affect the training, testing, or feature extraction. The choice to start training after 2004 is elaborated in section 4.2.4. Hence, the 135 matches after 2004 were deleted after the feature extraction. Lastly, the matches without point statistics were deleted. In total 127 matches without point statistics were deleted, however, only 42 were after 2004. List-wise deletion was chosen because the magnitude of the lost observations was rather small and imputation rather difficult. Imputation is rather difficult because odds and point statistics are highly dependent on individual players, opposing players, and the length of a match. Thus, a simple mean/median imputation would not have been correct.

Some matches contained players with an unknown hand preference. It was investigated if these players have known values for the hand preference in other matches, but this was not the case. We decided to keep the unknown values due to the suspicion that the hand preference will not be a very strong predictor.

For the feature extraction, 43.657 matches remained and for training and testing 33.849 matches after 2004 remained.

4.2.3 *Outliers*

A thorough investigation of outliers in the variables was conducted. Despite the fact that the boxplot method identified outliers in the point statistics, these were deemed to be reasonable (i.e. the longest match recorded, also recorded the most aces). It could be possible that treating these outliers increases overall prediction performance, but decreases the performance on high-ranked players. It was chosen to keep the values as is to best capture reality. Furthermore, most of the engineered features were based on weighted averages over a long period. This mitigates the influence of outliers due to the Law of Large Numbers.

However, for the betting odds there were some very unreasonable values. These were replaced by the 75% percentile (In the case of positive values) or the 25% percentile (In the case of negative values).

4.2.4 *Feature Engineering*

The inspiration for the features was taken mainly from Candila and Palazzo (2020) & Wilkens (2021) & Sipko and Knottenbelt (2015). For most features, the ideas are similar to what these authors did. For example, a strong serve is often dubbed as a must for successful players. Hence, double faults, aces, and winning points on serve games are used as features. The inclusion of the other features followed the same thought process.

As the nature of a tennis match is such that the outcome always has one player winning and one losing, a symmetric model was needed. As a second condition, the features need to be known beforehand to avoid information leakage. The total aces made by a player in the current match cannot be used for the construction of a feature of the current match.

First, each row was split into a row per player. The strategy was to construct features that were the same (in the case of categorical features) or were the difference between two opposing players. Consequently, adhering to the condition of a symmetric model. The features can be categorized based on how they were constructed. Most of the features are proportional point statistics and were calculated as follows:

$$X_i(t) = \frac{1}{N} \sum_{i=1}^{n} a_i(t-1) + a_i(t-2)... + a_i(t-n) \tag{4}$$

Where X denotes the feature at time t of player i. N is the number of matches played. a denotes the proportion of a point statistic and is calculated as follows:

$$a_i(t) = \frac{b_i(t)}{c_i(t)} \tag{5}$$

For example, when one calculates the proportion of aces relative to the total service points, b would be the number of aces and c would be the number of service points. The average of point statistics can be denoted by equation (4) as well. Whereas a in the case of aces would be the number of aces for player i at time t. X would then be the weighted sum of the average number of aces at time t for player i.

Categorical features either have the same input (e.g. surface type) or were manipulated such that the input is the same. For example, the feature preferred hand was transformed into both left, both right, or one left one right. All categories of the categorical features can be found in Table A.3

The Elo probability was used as a paired comparison input feature.[3]

All numerical features were scaled to a range of -1 to 1 using the min-max normalization. The extracted features and their description can be found in Table A.2. Not all features have been discussed as some are deemed self-explanatory.

In the previous section, it was mentioned that training and testing is done based on matches after 2004. Since most features are time-variant, maturing of the features is required. To visualize the variability, the standard deviations of the proportional features and average features over time were plotted in Figure B.1 and Figure B.2, respectively. It can be seen that after 2004 the standard deviation pretty much stabilizes (Except for 2019 and higher which could be due to an influx of many new players).

4.3 *Out-of-Sample split*

After the dataset was prepared, a 5-fold stratified test split was created. In the case of the MLP, a stratified hold-out set was created. Splits are stratified such

[3] See equations (1) and (2) for reference

that each out-of-sample - and in-sample set contains proportionally the same number of matches per year. Moreover, a dummy variable was created for the three ranking-based groups. These were used to indicate which player falls in which group.

4.4 *Hyper-Parameter tuning*

The randomized search was used to conduct the hyper-parameter tuning process. For the SVM and random forest, a 5-fold cross-validation was used and for the MLP a hold-out cross-validation was used. The decision for a hold-out cross-validation is based upon the computation time an MLP needs. Although it is shown that a randomized search has a probability of 95% to find the top 5% settings in just 60 iterations, a threshold of 10% of the total grid in the case of SVM and random forest and 20% of the total grid in the case of the MLP were chosen. A higher percentage for the MLP was chosen as it is believed that this grid has been the least well defined. The log loss metric was used as a validation score.

4.4.1 *Random Forest*

To define the grid of the random forest a loop over the hyper-parameters which, according to the literature are most important, was done. To reduce the computation time of the randomized search the number of features per split based on the recommendation in the literature and the splitting criterion based on empirical findings were fixated and the number of trees was set to 100. The hyper-parameter number of trees in training will be 1000 since the out-of-sample performance converges to a maximum with unlimited trees. However, due to the computation time, this was not deemed feasible for the randomized search.[4] The settings of the grid can be seen in Table A.4. The final parameters for the random forest can be seen in Table A.5

4.4.2 *Support-Vector Machine*

For the SVM two kernels were trained. For the RBF kernel the curvature, regularization, and a stopping tolerance were tuned. For the linear kernel only the regularization and a stopping tolerance are tuned. The recommendations from Hsu, Chih-Chung, and Chih-Jen (2003) were followed. As mentioned in the methods, the RBF kernel its computation can quickly become infeasible when a lot of observations are used. Instead of lowering the k-fold cross-validation, it was decided to randomly sample 5000 observations of each training split. Furthermore, the stopping tolerance speeds up the optimization as well, hence, a range of values has been tried. The same approach was used for the linear kernel since the computation time was also rather long. The output of a SVM does not lend itself to obtaining probabilities. To get the log loss of each iteration, a calibrated classifier was used to execute the method proposed by Platt (1999). He addressed the problem through a

[4] The randomized search took 8 hours with 100 trees

sigmoid transformation of the decision boundaries an SVM outputs. The grid of the RBF - and linear kernel can be seen in Table A.6. The final hyper-parameters for the linear - and RBF kernel can be seen in Table A.7 and Table A.8, respectively.

4.4.3 *Multilayer Perceptron*

The hyper-parameters of an ANN are more intertwined and harder to set than the previous two algorithms. Although, limiting the depth of a tree automatically increases the minimum samples needed for a leaf node, the choice of values and hyper-parameters for an ANN is broader. Therefore, a sequential approach was taken. First, we tried to determine the number of epochs for the randomized search. This was done by creating a standard ANN based on recommendations the recommendation of Heaton (2008) and then evaluating the train versus the test loss. Heaton (2008) recommends using one hidden layer and less than half of the input dimensions as the nodes in the hidden layer. Hence, the standard setup consisted of one hidden layer with 19 nodes and used a Adam optimizer with the standard learning rate. The epochs for the randomized search were set to 20 based on a visual assessment of Figure B.3.

After choosing the epochs, a grid was determined. The smoothing parameter of the l2 regularization and the dropout proportion were tuned to reduce potential overfitting. The number of nodes in the hidden layer was built around the recommendation of Heaton (2008). Moreover, the LeakyReLU, ReLU, and Tanh were the possible hidden layer activation functions. Lastly, a range of extra values around the default setting of the Adam learning rate was tried. The full grid can be seen in Table A.9. After the tuning of the parameters in the grid, a loop over the batch size with a early stopping patience for the epochs was done. The final model specifications can be seen in Table A.10.

4.5 *Feature Importance*

As previously stated, SHAP offers a good alternative to conventional feature importance methods such as the permutation feature importance or a tree-based feature importance. For this method the SHAP library of Lundberg and Lee (2017) was used. Initially, it was planned to use the DeepExplainer, however, as the results were evaluated it became apparent that the newest version of Tensorflow was not yet supported by the DeepExplainer. Therefore, an adjustment of the approach was taken and the Explainer was used. According to the documentation, the auto option should make the best choice as to which algorithm fits the model best. The auto algorithm used a permutation explainer with 10 iterations. The iterations allow the explainer to get good SHAPley values for models with higher-order interactions. As opposed to the conventional permutation importance, the SHAP permutation explainer ensures local accuracy.

5 RESULTS

To answer the research question the results are presented in three parts. First, the overall (global) predictions are evaluated using all features described in section 4.2.4. Second, using the same features, the prediction on a subset of the data is evaluated. These subsets consist of players grouped by their rankings. For the predictions, the log loss is considered to be the most important measure. Last, the feature importance of SHAP for both of the previous evaluations is investigated.

5.1 *Global Predictions*

The evaluation metrics of the out-of-sample predictions are summarized in Table 1. The log loss and accuracy are straightforward to interpret, however, for the ROI more insight into the specifics of the ROI is needed. The different ROIs are based on the risk appetite explained in section 3.3. The models are compared to a simplified consensus - and a Elo baseline. Comparing the performance to the consensus baseline indicates if enough information is gathered to have an edge over the betting market. The Elo baseline is regarded in the literature as a high standard statistical model. Improving upon the Elo baseline would indicate a successful model.

First, with regard to the accuracy, it can be seen that all trained models outperform the consensus - and the Elo baseline. The models improve upon the Elo - and consensus baseline from 1.5% to 4.1% and 2.8% to 4.1%, respectively. Both the MLP and the random forest perform best with 70.3% of all matches correct.

However, the importance lies in the difference in log loss. Due to the nature of the log loss, a lower (better) value indicates a model that avoids high confidence in false positives or false negatives. Unfortunately, the consensus baseline does not output a probability, hence, no log loss is recorded. Still, the Elo baseline is beat convincingly by all models. The increase ranges from 0.02 - 0.04. The log loss of 0.562 of the MLP is the best. However, just narrowly beating the random forest by a value of 0.002. Based on accuracy and log loss, the random forest and the MLP are considered to have the best performance.

As said before, more information for the evaluation of the ROI is needed. Therefore, in Table 1 the number of bets placed depending on the strategy and model are reported below the ROI values. It can be seen that the MLP and random forest outperform the other models based on the ROI itself. However, due to the fact that for the middle strategy the MLP and random forest place 25 and 11 bets, respectively and for the most risk-averse strategy 7 and 3, respectively, these values are considered to be not robust. In Table 2 the profit when betting one euro per bet can be seen. When combining all results it becomes apparent that the overall performance of the MLP is the best. It achieves a profit of 83.11 euro with 886 bets for the neutral strategy. Due to the high profit, a considerable number of bets, and the highest ROI, this is considered to be the best combination of model and strategy. Even so, the MLP in combination with the two risk-averse strategies is to be avoided. One upset highly affects the ROI and the profit of the MLP. In this

Table 1: Out-of-sample prediction performance for ATP matches played between 2005 and 2021

Models	Log Loss	Accuracy	ROI (r = 1)	ROI (r = 1.25)	ROI (r = 1.5)
Elo	0.604	0.672	-4.59 (3779)	-2.93 (1015)	0.92 (421)
Consensus	N/A	0.660	-11.72* (8342)	N/A	N/A
Random Forest	0.564	**0.703**	4.88 (1130)	**477.88** (11)	**1702.00** (3)
MLP	**0.562**	**0.703**	**9.38** (886)	355.96 (25)	1180.29 (7)
Linear Kernel SVM	0.579	0.688	-2.35 (2524)	4.12 (461)	26.66 (145)
RBF Kernel SVM	0.583	0.687	-2.05 (2680)	5.68 (550)	36.57 (195)

Notes: The asterisk denotes the fact that the consensus baseline follows a different strategy when betting. The best score for each metric is in bold. The differing *r* values of the ROI regard the risk factor (for reference see section 3.3). The values between parentheses represent the number of bets placed.

experiment the capital of a bettor is disregarded, consequently, this high variability can become detrimental in real-life with bettors without infinite capital. Hence, for the risk-averse strategy, it would be wiser to follow the two SVM models. These models tend to have a lot of matches where the edge over the betting market is substantial. Furthermore, the RBF kernel achieves a profit of 71.31 with 195 bets for the most risk-averse strategy. Due to the combination of profit, ROI, and the number of bets, the SVM with RBF could be considered the second-best model in regard to ROI. The log loss of the random forest is almost identical to that of the MLP, but this does not show itself in the ROI. Strangely, it seems that the random forest bets on more false positives which due to the betting rule should be avoided if the log loss is of a high standard.

5.2 *Ranking-Based Predictions*

The summary of the results ranking-based groups can be seen in Table 3. It is clear that highest-ranked players are easiest to predict. For the top 30, the MLP is the

Table 2: Profit in terms of €

Models	**ROI**	**ROI**	**ROI**
	(r = 1)	(r = 1.25)	(r = 1.5)
Elo	-173.45	-24.25	3.87
Consensus	-977.68*	N/A	N/A
Random Forest	55.14	41.57	48.06
MLP	**83.11**	**63.99**	**75.62**
Linear Kernel SVM	-59.31	18.99	38.67
RBF Kernel SVM	-54.94	31.24	71.31

Notes: This table presents the theoretical profit that would have been made if in the period from 2005 - 2021 the different betting strategies were followed. The best score for each strategy is in bold. The asterisks denotes the fact that the consensus model bases its strategy on another rule than the other algorithms.

best model. Although the accuracy is almost identical to that of the random forest, the log loss is undoubtedly better than the random forest. Remarkably, the Elo also performs very well in the top 30 group. Its log loss is the second-best of all models. This could be due to the fact that for the Elo points and consequently the predicted probability to be robust, a lot of games by one player have to be played. Generally, the top players play the most matches which could explain this good performance.

Furthermore, whereas the deterioration of performance between the top 30 and top 30-50 group is notable, the reduction from the top 30-50 group to the over 50 group is non-existent. The Elo has the biggest reduction in performance when comparing these groups, which is only a 0.8% decrease in accuracy and a 0.004 increase in log loss. The MLP and the random forest predict the over 50 group even better than the top 30-50 group.

The lowest accuracy difference in comparison to the top 30 group is that of the over 50 group for the MLP and the random forest. These models both see a decrease of almost 5% in accuracy. The lowest log loss increase of 0.014 is that of the random forest for the groups top 30 and over 50. Overall the random forest seems to be the most consistent model.

5.3 *Feature Importance*

As mentioned before, the MLP had the best overall performance. Hence, the MLP its performance is further investigated. The mean SHAP values of the ten most important features can be seen in Figure 2. It can be seen that the difference

Table 3: Out-of-sample prediction performance grouped by player rankings for ATP matches played between 2005 and 2021

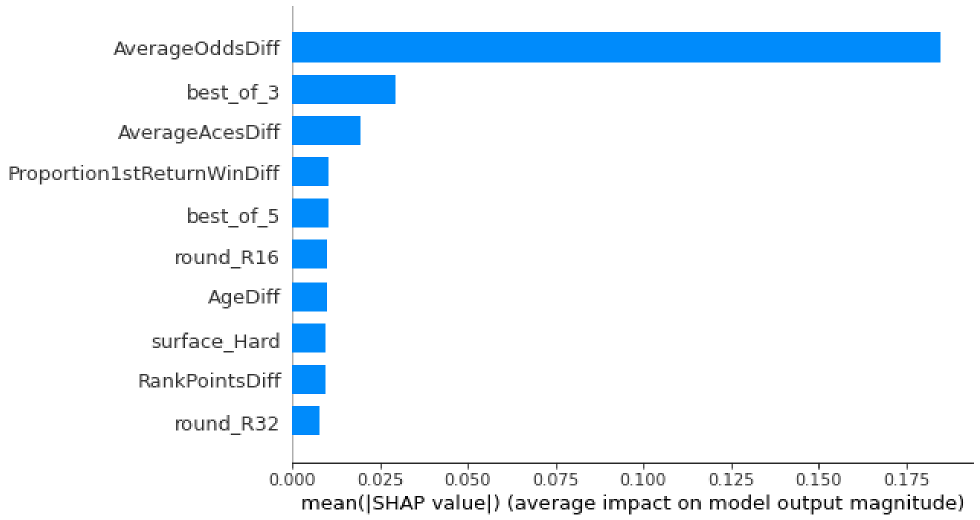
Models	**Top 30**		**Top 30-50**		**Over 50**	
	Accuracy	**Log Loss**	**Accuracy**	**Log Loss**	**Accuracy**	**Log Loss**
Elo	0.727	0.549	0.652	0.628	0.644	0.632
Consensus	0.697	N/A	0.642	N/A	0.644	N/A
Random Forest	0.738	0.567	**0.679**	0.591	0.690	0.581
MLP	**0.739**	**0.521**	0.672	**0.588**	**0.691**	**0.578**
Linear Kernel SVM	0.728	0.585	0.660	0.618	0.655	0.618
RBF Kernel SVM	0.728	0.585	0.658	0.618	0.655	0.618

Notes: This table displays the prediction performance on different subsets that are grouped by player ranking. The bold values denote the best metric for that group.

in the average odd is by far the most important feature. is further investigated. Strangely, the constructed point statistics, but also the ranking features are rather insignificant. One could only imagine that the betting odds already contain a lot of the information that is generated by these features. The dummy best of 3 and the difference in average aces are the next best features. After those two all features contribute roughly the same.

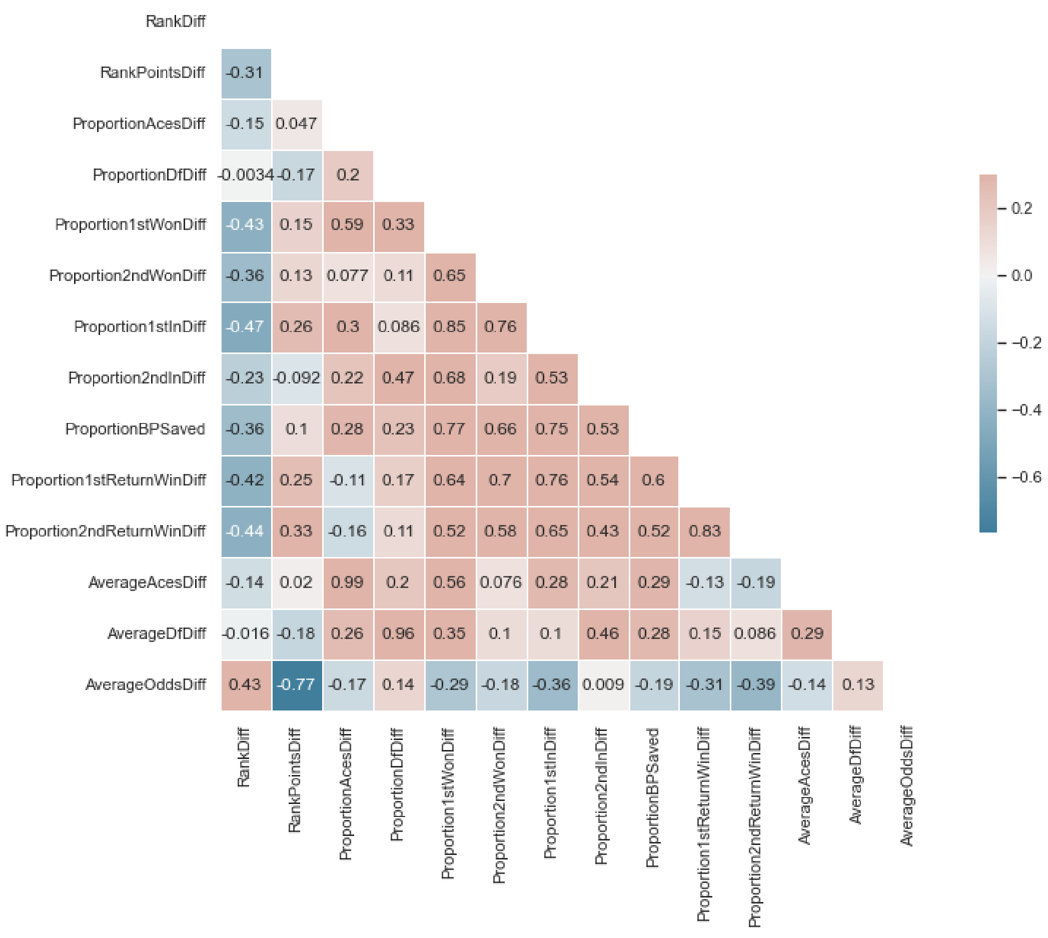
A correlation plot was made to see if the multicollinearity could be a cause of the importance of betting odds. The correlation plot can be seen in Figure 3. The general picture is that the point statistics are positively correlated with each other. This is pretty logical, a player who proportionally wins more points probably has a positive statistic for the 1st - and 2nd service points won. This would overall indicate that this player is better than the opposing player. The only surprising point statistic correlation is the AverageDfDiff feature with the other point statistics. Except for the return point statistics, it is positively correlated with the point statistics. One would expect that better players have fewer double faults, however, this does not seem to be the case. The AverageOddsDiff is especially correlated with the ATP rankings and the ATP ranking points. This makes it seem that betting agencies base their odds strongly on the ATP rankings and the ATP ranking points. However, an observation of just the correlation plot is not enough to indicate that the point statistics are captured by the betting odds.

Figure 2:
Global mean SHAPley values of the 10 best features



The different subsets of section 5.2 have also been investigated using the same barplot. However, the overall picture does not differ. In fact, the 10 most important features stay roughly the same (Only occasionally another feature such as difference in Elo probability is in the top 10). These figures can be seen in the appendix for the top 30, 30-50, and over 50 in Figure B.4, Figure B.5, and Figure B.6 respectively.

Figure 3:
Correlation matrix of all point statistics -, rankings -, and odds features



	RankDiff	RankPointsDiff	ProportionAcesDiff	ProportionDfDiff	Proportion1stWonDiff	Proportion2ndWonDiff	Proportion1stInDiff	Proportion2ndInDiff	ProportionBPSaved	Proportion1stReturnWinDiff	Proportion2ndReturnWinDiff	AverageAcesDiff	AverageDfDiff	AverageOddsDiff
RankDiff														
RankPointsDiff	-0.31													
ProportionAcesDiff	-0.15	0.047												
ProportionDfDiff	-0.0034	-0.17	0.2											
Proportion1stWonDiff	-0.43	0.15	0.59	0.33										
Proportion2ndWonDiff	-0.36	0.13	0.077	0.11	0.65									
Proportion1stInDiff	-0.47	0.26	0.3	0.086	0.85	0.76								
Proportion2ndInDiff	-0.23	-0.092	0.22	0.47	0.68	0.19	0.53							
ProportionBPSaved	-0.36	0.1	0.28	0.23	0.77	0.66	0.75	0.53						
Proportion1stReturnWinDiff	-0.42	0.25	-0.11	0.17	0.64	0.7	0.76	0.54	0.6					
Proportion2ndReturnWinDiff	-0.44	0.33	-0.16	0.11	0.52	0.58	0.65	0.43	0.52	0.83				
AverageAcesDiff	-0.14	0.02	0.99	0.2	0.56	0.076	0.28	0.21	0.29	-0.13	-0.19			
AverageDfDiff	-0.016	-0.18	0.26	0.96	0.35	0.1	0.1	0.46	0.28	0.15	0.086	0.29		
AverageOddsDiff	0.43	-0.77	-0.17	0.14	-0.29	-0.18	-0.36	0.009	-0.19	-0.31	-0.39	-0.14	0.13	

6 DISCUSSION

The goal of this study is to critically assess the performance and workings of machine learning/deep learning algorithms when using the most used and successful features found in the literature. An assessment is only useful if put into perspective, hence, two baselines were constructed that have already proven their value in this domain. Furthermore, findings in the literature indicate a difference between the predictive performance of higher-ranked - and lower-ranked players which was further investigated.

The ROI as a metric takes into account the information of the bookmakers. A positive ROI is very difficult to obtain due to the fact that bookmakers are aiming for profit. We deem the neutral ROI of the MLP to be the best possible outcome of all possible combinations of models and strategies. However, the SVMs perform well when more risk-averse strategies are followed. The risk factor was used to imitate different risk appetites. The different values were chosen somewhat at random (Except for r=1 which can be seen as having an edge over the betting market). Although, we believe the addition of a risk factor is fruitful, its usage can be improved. Given that different algorithms behave differently with decreasing risk, one could tune the risk parameter more extensively. Generally, it seems that even with well-performing models and a relative improvement upon two competitive baselines, betting on tennis matches is not remarkably lucrative. As opposed to conventional investment strategies (e.g stocks or bonds), in betting, losing means immediately losing the entirety of your staked capital. This high variability is detrimental for individuals without enormous amounts of capital.

As can be seen in the literature and this study, higher-ranked players are easier to predict. This could be due to the features used. A lot of the features are time-dependent and player-dependent. Therefore, players that play more matches have more robust features. Generally speaking, high-ranked players play more matches which would generate more robust statistics and partially explain this difference. In comparison to Kovalchik (2016), the difference between the different groups is not as pronounced as in her study. Accordingly, we were able to reduce this variability and better capture the uncertainty of the lower-ranked players. Furthermore, the MLP has a very low log loss for the top 30. This could partially explain the ROI it achieves in comparison to the random forest.

It is believed that this study convincingly achieved its goal in terms of performance. Especially the log loss has been increased a lot when compared to the Elo model. The random forest and MLP perform particularly well. The MLP has an edge over the random forest in terms of betting. Apparently, the MLP better utilizes information in matches that according to the bookmakers are comparatively uncertain. This could be because ANNs are capable of modeling complex and unnoticed associations. By and large, the MLP had the best performance, although closely followed by the random forest. Hence, a conclusive answer on algorithm usage is difficult to give. This conclusion is in line with Wilkens (2021). However, not with Gao and Kowalczyk (2019) as they report an accuracy of 83% for the random forest. This could be because they performed feature selection and mentioned that the random forest improved the most while reducing the number of features. Furthermore, this is the third study that compares an ANN to other algorithms, but the first that reports the best results for this algorithm.

Unfortunately, the feature importance results were unexpected. According to the permutation explainer, the average odds had a feature importance of approximately six times more than the next best feature. Hence, the interpretation of this result is tough as most conclusions would be that the average odd alone would suffice as a model. Two insights can be drawn from this result. First, given the correlation matrix, a more conservative approach in regard to feature inclusion is a good alternative to the current approach. For example, computing an overall measure of a player its capabilities as implemented by Sipko and Knottenbelt (2015) & Candila and Palazzo (2020). All point statistics could then be reduced to one feature or just a few features. Second, the average odds should function as a benchmark. Beating this benchmark would indicate an information gain over the information found on the betting market.

7 CONCLUSION

The aim of this study is to provide insights into the workings of machine learning/deep learning in the tennis prediction domain. To evaluate this, two baselines inspired by the literature were used to give us a starting point. The goal was translated into a main question which is broken down into three subquestions. The main question and three subquestions are answered chronologically.

SQ1 *To what extent do a random forest, a multilayer perceptron, and a support vector machine affect the predictive performance in terms of return on investment, log loss, and accuracy?*

We find that the MLP has the best overall performance. Based on the results, a case could be made for the random forest, however, the ROI of the neutral strategy gave the MLP its edge over the random forest. Furthermore, the SVMs both perform well when a very high confidence is needed. Combining these insights could be a gap for future research. Instead of optimizing the log loss, an optimization of the ROI could take place, thus, opening the possibility to tune the risk factor as well. The stake placed was disregarded, varying the stake based on the edge one has over the betting market could very well improve the ROI.

SQ2 *How does the predictive performance vary across player ranking-based categorized groups?*

The accuracy - and log loss scores grouped by ranking do not vary as much as was expected. A striking result was the fact that the performance of the random forest and the MLP for the middle group was worse than for the lowest group. This could indicate the capabilities of these algorithms are better suited to pick up matches that are regarded as uncertain. One could argue that it would be better to train different models for different groups of players. However, this would put at risk the needed symmetry of a model. One future gap could be to investigate the lower regions of tennis only. Though, difficulties in gathering the necessary data would arise. The data source of Jeff Sackmann is generated by manually charting the matches. Hence, gathering data such as point statistics is an expensive process.

SQ3 *According to SHAPley values, which features of the best-performing model are of the greatest importance for all players, and which features are of the greatest importance for the player ranking-based groups?*

Unfortunately, the results of the feature importance were unforeseen. The average odds were by far the most important feature. The bookmakers likely use similar (if not more extensive) information to determine the betting odds. This is a limitation of the current approach. Moreover, there were no notable differences in regard to feature importance between the different groups. Which is less surprising as the difference between performance was not so pronounced. It would have been an improvement if we accounted for multicollinearity instead of focusing primarily on performance. Possible ventures are the aggregation of statistics, avoiding the dummy trap (i.e. one dummy for the 'best of' feature), and feature selection.

MQ *How do various machine learning/deep learning models utilize contextual information, betting odds, player rankings, and point statistics while being compared to a set of baselines to predict the outcome of ATP tennis matches between 2005 and 2021?*

We conclude that the overall performance is significantly better than the two literature-inspired baselines. The MLP is regarded to be the best, however, context is not to be neglected. Furthermore, the time frame used provided us with a model that is robust to time variance. The features were of less importance. More extensive research in this regard would be beneficial for this domain. However, when analysing other papers there seems to be an upper bound to what can be achieved.

It is believed that by conducting one of the widest studies (in terms of features, algorithms, and time frame) into the modeling of tennis prediction, this study can be used as a benchmark. The knowledge generated by the feature importance gives some extra insight into the features that can be used (i.e. aggregating point statistics and exploiting the bookmakers' information). Moreover, by showing that it is possible to partially exploit the inefficiencies of the betting market, bookmakers could take this study as a signal to counter these inefficiencies.

REFERENCES

Angelini, G., Candila, V., & De Angelis, L. (2022). Weighted elo rating for tennis match predictions. *European Journal of Operational Research*, *297*(1), 120–132.

Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., ... Herrera, F. (2019). Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, *58*, 82–115.

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of machine learning research*, *13*(2), 281–305.

Biau, G. B., & Scornet, E. (2016). A random forest guided tour. *Test*, *25*(2), 197–227.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Bzdok, D., Altman, N., & Krzywinski, M. (2018). Statistics versus machine learning. *Nature America*, *15*(4), 233–234.

Candila, V., & Palazzo, L. (2020). Neural networks and betting strategies for tennis. *Risks*, *8*(3).

Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., & Lopez, A. (2020). A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*, *408*, 189–215.

Chollet, F., et al. (2015). *Keras.* GitHub. Retrieved from `https://github.com/fchollet/keras`

Cornman, A., Spellman, G., & Wright, D. (2017). Machine learning for professional tennis match prediction and betting. *Stanford University*.

Dogru, N., & Subasi, A. (2018). Traffic accident detection using random forest classifier. *2018 15th learning and technology conference (L&T)*, 40–45.

Elo, A. E. (1978). The rating of chessplayers, past and present. *BT Batsford limited*.

Gao, Z., & Kowalczyk, A. (2019). Random forest model identifies serve strength as a key predictor of tennis match outcome. *Journal of Sports Analytics*, 1–8.

Goddard, J., & Asimakopoulos, I. (2004). Forecasting football results and the efficiency of fixed-odds betting. *Journal of Forecasting*, *23*(1), 51–66.

Gosh, S., Sadhu, S., Biswas, S., Sarkar, D., & Sarkar, P. P. (2019). A comparison between different classifiers for tennis match result prediction. *Malysian Journal of Computer Science*, *32*(2), 97–111.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... Oliphant, T. E. (2020, September). Array programming with NumPy. *Nature*, *585*(7825), 357–362.

Heaton, J. (2008). *Introduction to neural networks with java.* Heaton Research, Inc.

Hsu, C.-W., Chih-Chung, C., & Chih-Jen, L. (2003). A practical guide to support vector classification. *Department of Computer Science National Taiwan University*.

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95.

Kampakis, S., & Thomas, W. (2015). Using machine learning to predict the outcome of english county twenty over cricket matches. *arXiv preprint arXiv*, *1511.05837*.

Kelly Jr., J. L. (1956). A new interpretation of information rate. *The Kelly capital* *growth investment criterion: theory and practice*, 25–34.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization.

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., … Willing, C. (2016). Jupyter notebooks – a publishing format for reproducible computational workflows. In F. Loizides & B. Schmidt (Eds.), *Positioning and power in academic publishing: Players, agents and agendas* (p. 87 - 90).

Kovalchik, S. A. (2016). Searching for the goat of tennis win prediction. *Journal of Quantitative Analysis in Sports*, *12*(3), 127–138.

Leitner, C., Zeileis, A., & Hornik, K. (2009). Is federer stronger in a tournamentwithout nadal? an evaluation of odds and seedings for wimbledon 2009. *Austrian Journal of Statistics*, *38*(4), 277–286.

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems*, *31*, 4768–4777.

Maas, A. L., Hannun, A. Y., & Ng, A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models. *Proc. icml.*, *30*(1), 3.

McKinney, W., et al. (2010). Data structures for statistical computing in python. In *Proceedings of the 9th python in science conference* (Vol. 445, pp. 51–56).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Platt, J. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, *10*(3), 61–74.

Sipko, M., & Knottenbelt, W. (2015). Machine learning for the prediction of professional tennis matches. *MEng computing-final year project, Imperial College London*.

Somboonphokkaphan, A., Phimoltares, S., & Lursinsap, C. (2009). Tennis winner prediction based on time-series history with neural modeling. *Proceedings of the International MultiConference of Engineers and Computer Scientists*, *1*, 18–20.

Subudhi, A., Dash, M., & Sabut, S. (2020). Automated segmentation and classification of brain stroke using expectation-maximization and random forest classifier. *Biocybernetics and Biomedical Engineering*, *40*(1), 277–289.

Teräsvirta, T., Chien-Fu, L., & Clive, W. G. (1993). Power of the neural network linearity test. *Journal of time series analysis*, *14*(2), 209–220.

Van Rossum, G., & Drake Jr, F. L. (1995). *Python tutorial*. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands.

Waskom, M. L. (2021). seaborn: statistical data visualization. *Journal of Open Source Software*, *6*(60), 3021.

Wheatcroft, E. (2020). A profitable model for predicting the over/under market in football. *International Journal of Forecasting*, *36*(3), 916–932.

Wilkens, S. (2021). Sports prediction and betting models in the machine learning age: The case of tennis. *Journal of Sports Analytics*, *7*(2), 1–19.

APPENDIX A

All code necessary to replicate this study can be found on GitHub:

https://github.com/ThWillekes/MasterThesisTW

Table A.1: Columns of merged datasets after deleting duplicate - and useless columns

Column Name	Meaning of column
Avg*	Average odd of player
B&W*	Bet&Win odds
B365*	Bet365 odds
CB*	Centrebet odds
EX*	Expekt odds
GB*	Gamebookers odds
IW*	Interwetten odds
LB*	Ladbrokes odds
Max*	Maximum odds
PS*	Pinnacles Sports odds
SB*	Sportingbet odds
SJ*	Stan James odds
UB*	Unibet odds
Series	Name of ATP tennis series
Tournament	Name of tournament
Pts*	ATP Ranking points
sets*	Number of sets won
tourney_id	Unique tournament identifier
surface	Surface type
tourney_level	Tournament level (i.e. Grand Slam)
id*	Match identifier
name*	Player name
hand*	Preferred playing hand
ht*	Player height
age*	Player age
minutes	Duration of match in minutes
round	round of the tournament
Date	Date of the match
df*	Number of double faults
svpt*	Number of total service points
1stIn*	Number of 1st service points in
1stWon*	Number of 1st service points won
2ndWon*	Number of 2nd service points won
SvGms*	Number of service games won
bpSaved*	Number of break points saved
bpSaved*	Number of break points faced
rank*	Ranking
best_of	Best of 5 or 3 sets to be won
cleaned_score	Score without tie-break score

Table A.2: Extracted Features from the raw data. The features ending with 'Diff' are captured as the difference of that statistic between opposing players. Features denoted by an asterisks are time-variant.

Feature	Meaning of Feature	Values
HeightDiff	Height in cm	Float
AgeDiff	Age in years	Float
RankDiff	ATP Ranking	Float
RankPointsDiff	ATP Ranking Points	Float
ProportionAcesDiff*	proportion of Aces to total Service points	Float
ProportionDfDiff*	proportion of Df to total Service points	Float
Proportion1stWonDiff*	proportion of 1st Service won to 1st Service in	Float
Proportion2ndWonDiff*	proportion of 2nd Service won to 2nd Service in	Float
Proportion1stInDiff*	proportion of 1st Service in to 1st Service points	Float
Proportion2ndInDiff*	proportion of 2nd Service in to 2nd Service points	Float
ProportionBPSaved*	Proportion of break points saved versus break points faced	Float
Proportion1stReturnWinDiff*	Proportion of won points on opponents 1st Serve	Float
Proportion2ndReturnWinDiff*	Proportion of won points on opponents 2nd Serve	Float
AverageAcesDiff*	Average aces	Float
AverageDfDiff*	Average double faults	Float
AverageOddsDiff	Average of odds	Float
SpreadOddsDiff	Difference between higest and lowest odd	Float
EloProbDiff	Elo probability	Float
PreferredHand	Captures which hand both of the players prefer	Three categories
Surface	Surface type of the match	Three categories
Round	Round of the tournament	Eight categories
TournamentLevel	Level of the tournament	Four categories
BestOf	Indicates how many sets have to be won	Two categories

Table A.3: Categories of the categorical features

Feature	Categories
Surface	Grass, Hard, Carpet, and Clay
Round	Round of 128, Round of 64, Round of 32, Round of 16, Quarter-final, Semi-final, Final, or Round Robin
TournamentLevel	Grand Slam, Masters, Tour Finals, or Others
BestOf	Best of 5 or Best of 3
PreferredHand	Both left, Both right, or one left one right

Table A.4: The grid for the random search of the random forest. All other settings of the RandomForestClassifier of Scikit Learn were set to the default settings.

Hyper-parameter	Values
Splitting criterion	Entropy
Maximum number of features considered per split	log2
Maximum depth of individual tree	Range from 2 - 30 with steps of 2
Minimum samples needed for a leaf node	Range from 2 - 60 with steps of 3
Minimum samples needed for a split	Range from 2 - 60 with steps of 3
Number of iterations of the random search	560

Table A.5: The settings for the five different splits of the random forest.

Hyper-parameters	Split one	Split two	Split three	Split four	Split five
Number of trees in a forest	1000	1000	1000	1000	1000
Splitting criterion	Entropy	Entropy	Entropy	Entropy	Entropy
Number of features considered per split	log2	log2	log2	log2	log2
Maximum depth of individual tree	16	28	16	18	14
Minimum samples needed for a leaf node	5	14	17	17	5
Minimum samples needed for a split	38	2	32	59	41

Table A.6: The grid for the random search of the SVM. All other settings of the svc of Scikit Learn were set to the default settings. Gamma is only relevant for the RBF kernel, hence the asterisks.

Hyper-parameter	Values
Curvature gamma*	Range from 2^{-15} - 2^{3} with 10 exponential steps
Regularization C	Range from 2^{-5} - 2^{15} with 11 exponential steps
Stopping tolerance	Range from 0.0001 - 0.001 with steps of 0.00001
Number of iterations of the random search RBF kernel	1000
Number of iterations of the random search linear kernel	100

Table A.7: The settings for the five different splits of the linear kernel SVM. These settings were used to train the models. Only the hyper-parameters that deviate from the default settings are displayed. The values are rounded to five numbers after the decimal.

Hyper-parameters	Split one	Split two	Split three	Split four	Split five
Regularization C	0.5	0.03125	0.125	0.03125	0.03125
Stopping tolerance	0.00045	0.00065	0.00095	0.00024	0.00075

Table A.8: The settings for the five different splits of the RBF kernel SVM. These settings were used to train the models. Only the hyper-parameters that deviate from the default settings are displayed. The values are rounded to five numbers after the decimal.

Hyper-parameters	Split one	Split two	Split three	Split four	Split five
Regularization C	128	32	2048	128	128
Stopping tolerance	0.00045	0.00065	0.00095	0.00024	0.00075
Curvature	0.00781	0.00012	3.05e-05	3.05e-05	3.05e-05

Table A.9: The grid for the random search of the MLP.

Hyper-parameter	Values
Activation function	ReLU, LeakyReLU, or Tanh
Nodes of hidden layer	Range from 3 - 41 with step size of 2
Dropout rate	0.5, 0.25, or 0
Learning rate Adam	0.0001, 0.0003, 0.0005, 0.0007, 0.0009, 0.001, 0.003, or 0.005
L2 regularization smoothing	0.0001, 0.001, 0.01, 0.1, or 0.0

Table A.10: Final model specifications of the train split of the MLP after the randomized - and manual search.

Hyper-parameter	Values
Activation function	LeakyReLU
Nodes of hidden layer	19
Dropout rate	0
Learning rate Adam	0.003
L2 regularization smoothing	0.0
Epochs	20
Batch size	32

APPENDIX B

Figure B.1:
Standard deviation of proportional point statistics plotted over time in years

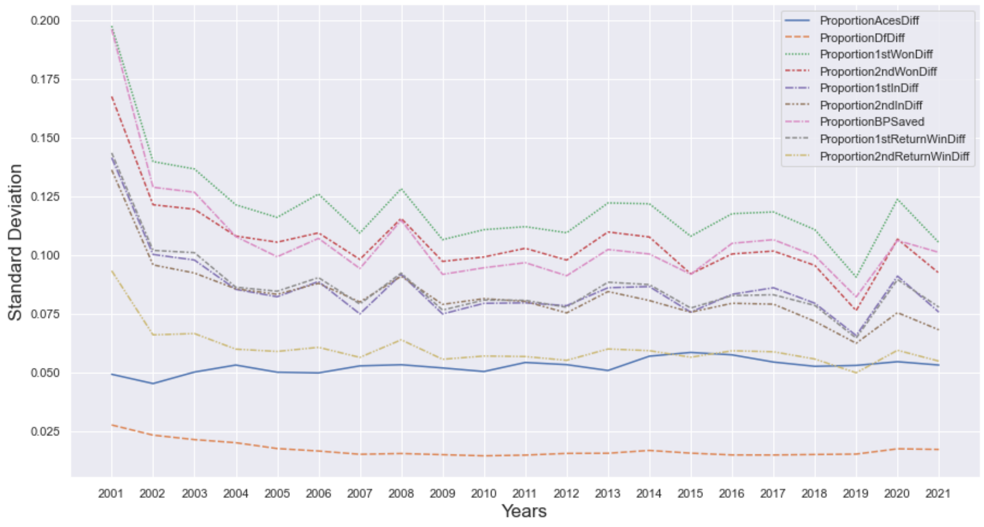


Figure B.2:
Standard deviation of average point statistics plotted over time in years

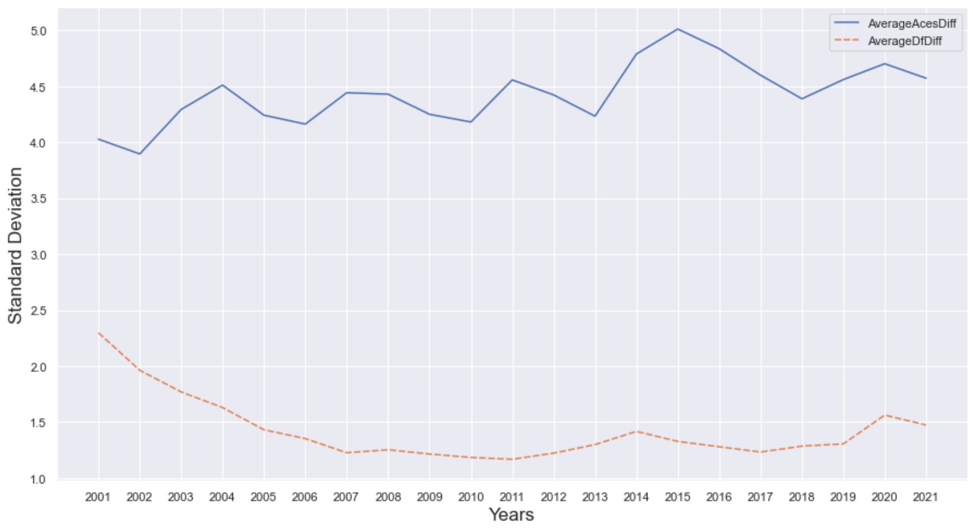


Figure B.3:
Train - versus test loss of initial model to determine number of epochs for randomized search

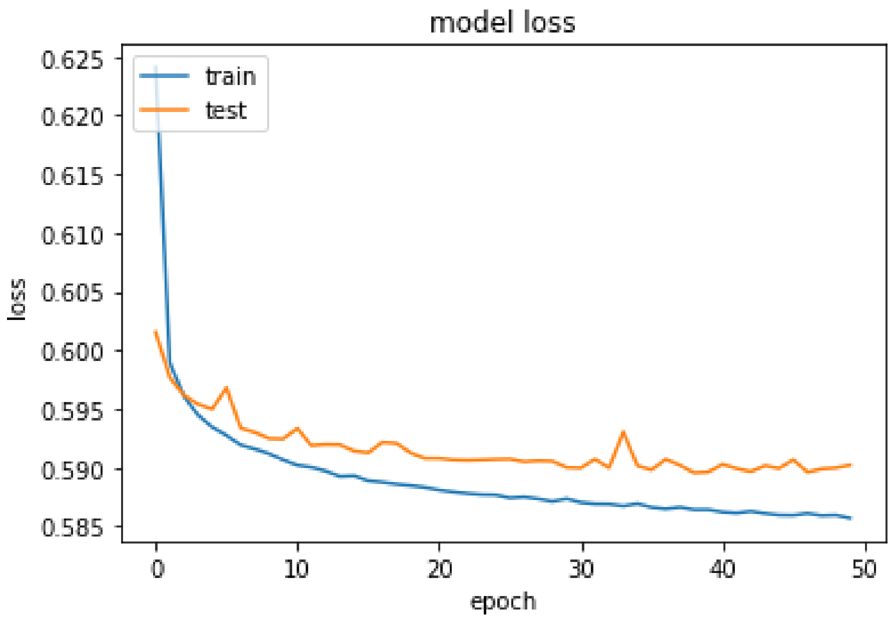


Figure B.4:
Mean SHAPley values of the 10 best features for the subset with top 30 players

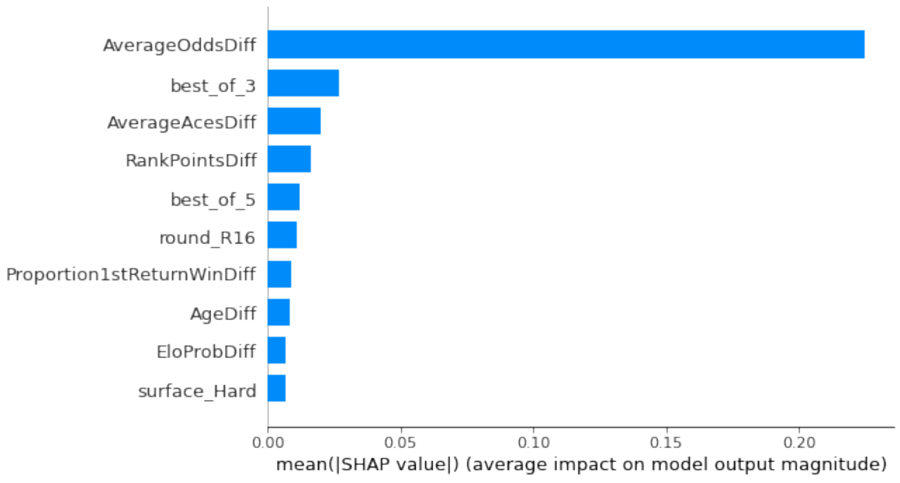


Figure B.5:
Mean SHAPley values of the 10 best features for the subset with top 30 players to top 50 players

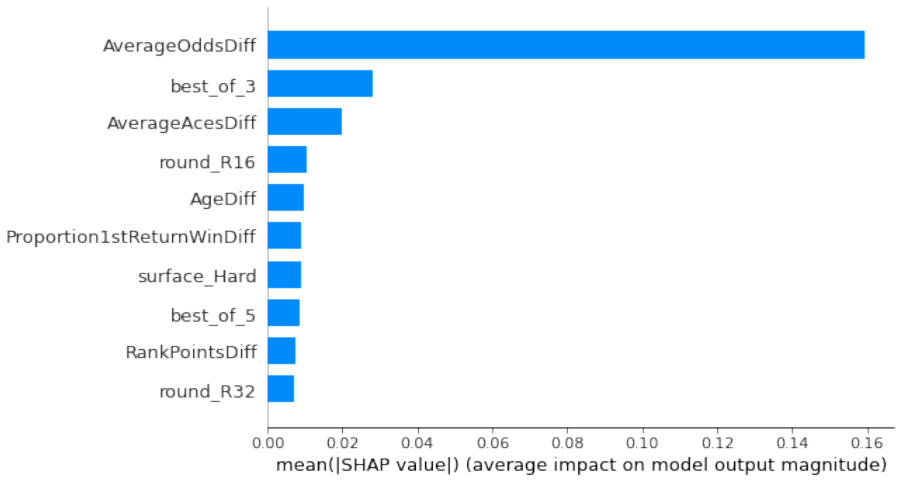
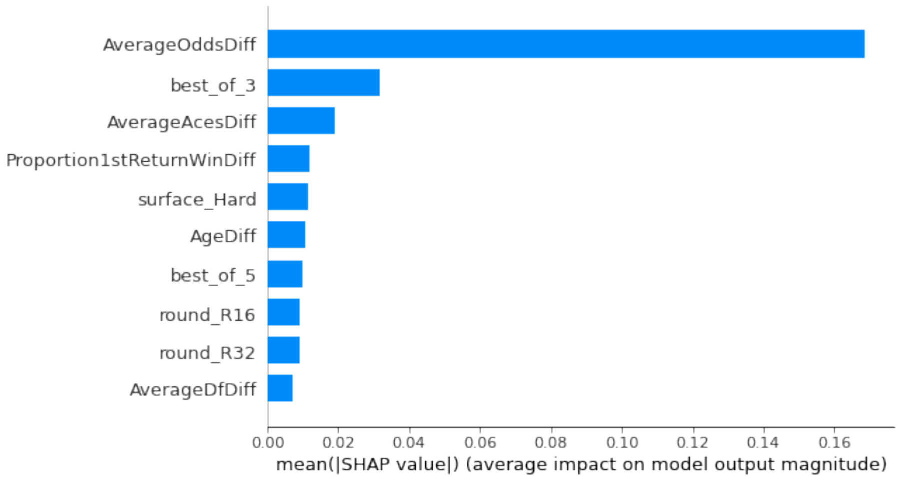


Figure B.6:
Mean SHAPley values of the 10 best features for the subset with players ranked over 50


AverageOddsDiff
best_of_3
AverageAcesDiff
Proportion1stReturnWinDiff
surface_Hard
AgeDiff
best_of_5
round_R16
round_R32
AverageDfDiff
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0.16
mean(|SHAP value|) (average impact on model output magnitude)