BUSS2505-03-Machine Learning

Project Report

Course code	BUSS2505-03
Course name	Machine Learning
Team members	DANG HOANG ANH QUAN JASON CHOW
Lecturer	李成璋

- **Disclosure on the usage of generative AI:**

It is a matter of fact that Generative AI such as DeepSeek played an important role for us in completing this project. The task which AI played the biggest role in is the coding. More specifically, we used AI to help us in the feature engineering task and model building, acknowledging there could be mistake, we always tried to verify the code provided by AI. Additionally, AI was occasionally used to review grammatical issue, but not much.

I. Introduction

Throughout the history of sports, the ability to predict match outcomes has captivated audiences and fueled the growth of a multi-billion-dollar betting industry. Tennis, as one of the most popular sports globally, presents unique challenges and interest for prediction due to its individual nature and the multitude of factors influencing performance. This project aims to build a pre-match tennis prediction by leveraging machine learning algorithms to analyze the impact of player statistics, historical performance, and factors such as court surface and fatigue, on the match outcome. Using a comprehensive dataset of professional ATP matches from 1991 to 2024, we engineer features like surface-specific ELO ratings, head-to-head records, and recent form indicators to build predictive models.

Machine Learning Models in the same problem with the same scope of matches (all levels, all surfaces men individuals) has the accuracy ranges from 65% to 70% (De Seranno 2020; Lisi & Zanella 2017; Pham & Bufi 2023; Mateus 2017). Our goal is to try to reach the same level of prediction capabilities and even higher if capable. Practical applications in the betting market is also a consideration for this project, leading us to choose models that can have probabilistic results rather than just classification task. However, try to determine a profitable strategy based on the model is out of the scope of our project.

II. Data and features

1. Tennis match dataset

A. Overview

The dataset and its quality can be considered one of the most vital part in building a machine learning model. In our case, it is fortunate that there exist some excellent open-source datasets for tennis.

The ATP (Association of Tennis Professionals) provides information about every official ATP men's tennis match since 1968. Luckily, Jeff Sackmann (2025) has already collected all of the raw tennis match data from open sources and published them on GitHub. Each dataset contains all tour-level matches from Grand Slams, ATP Masters 1000 tournaments, and ATP

500/250 series tournaments starting of each year from 1968 up to the end of 2024. However, only matches starting from 1991 have more in-match statistics related to serves, breakpoints,...

For each row in the dataset from 1991 onwards, which corresponds to one real tennis match, there are statistics regarding to the winners and losers, including each of their age, rank, handedness, height, and match statistics such as the score, percentage of first serves won, percentages of break points won,... These matches are sorted by date and contain information on the tournament and surface they played.

Although data of tournaments for younger players like the ATP Challengers and ATP futures is also available, it is beyond the scope of this prediction, and such data is not as useful as well since the characteristics of these tours is different, and only a fraction of the young futures players will make it as a professional tennis player in the end.

B. Data cleaning

As stated, only matches from after 1991 have in-match statistics, therefore, we only use the data from 1991 and later for the models.

Matches that were stopped before completion is removed from the dataset. It is not useful that the model learns from these matches, as they are usually the result of walkovers, injuries or disqualification. In such cases , the injured or disqualified player's strength is not properly reflected in the result of the match and is thus not useful as learning data for the model. Moreover, in-match statistics of such matches is also not available due to the match's incompletion. Several matches with missing serving stats will also be removed.

Matches from Davids Cup and Laver Cup is also removed. Firstly, the characteristics of Davids Cup is team competition, and Laver Cup is exhibition-style team event, which can make the players being less serious, and not useful in reflecting players' performance. Furthermore, in-match and ranking statistics are also missed in such matches.

This data cleaning removes 3531 matches from the dataset mostly come from the Davids Cup matches, the number of matches left in the data is 94999.

C. Statistical exploratory analysis

To better understand what factors in the matches that can make a player more likely to win, we will explore the difference between the performance of the winners and the losers in each match of the data.

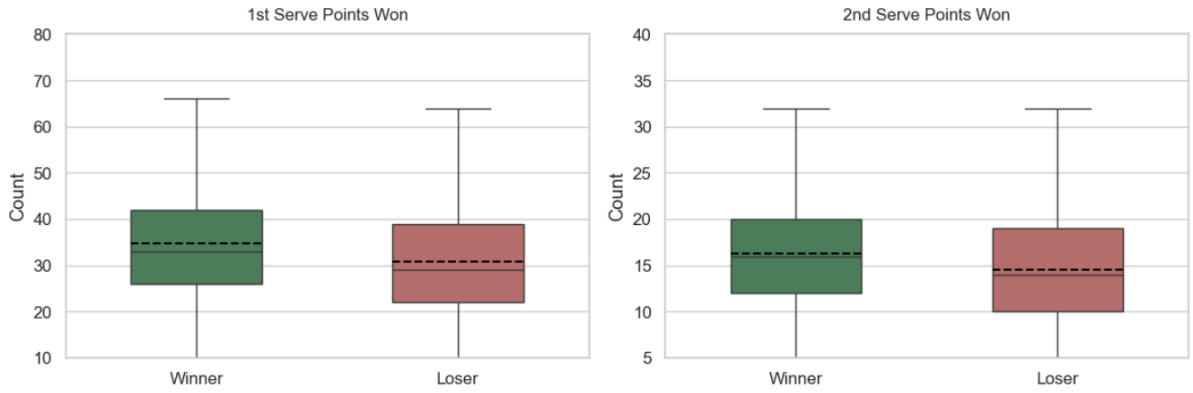

Figure 1: *Winners' versus Losers' performance in Serve Points result (Outliers removed)*

The box plots overall show that winners maintain slightly higher counts for both 1st and 2nd serve points won compared to losers. Although, the differences are modest in magnitude (few points per match), these marginal gains are significant in tennis, for some legendary players, for example Roger Federer, who has a career win percentage of 86%, only won 54% of his played points (ATP Tour 2024).

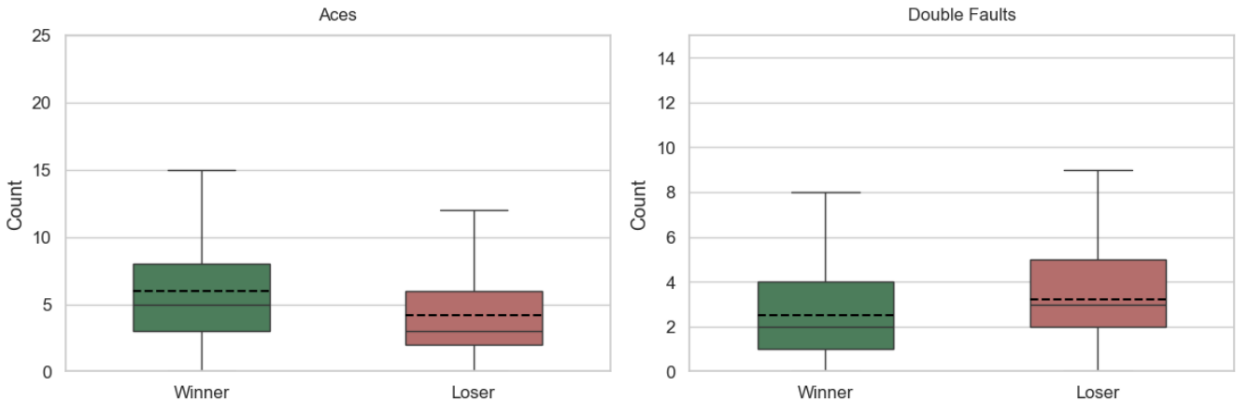

Figure 2: *Winners' versus Losers' Aces and Double Faults each match (Outliers removed)*

From the box plot (**Figure 2**), it is clear winners hit more aces and commit fewer double faults than losers. Firstly, in tennis, this advantage is critical because aces represent free,

uncontested points, while fewer double faults minimize unforced errors that gift points to opponents. Furthermore, these metrics also demonstrate better serve efficiency of winners over the loser.

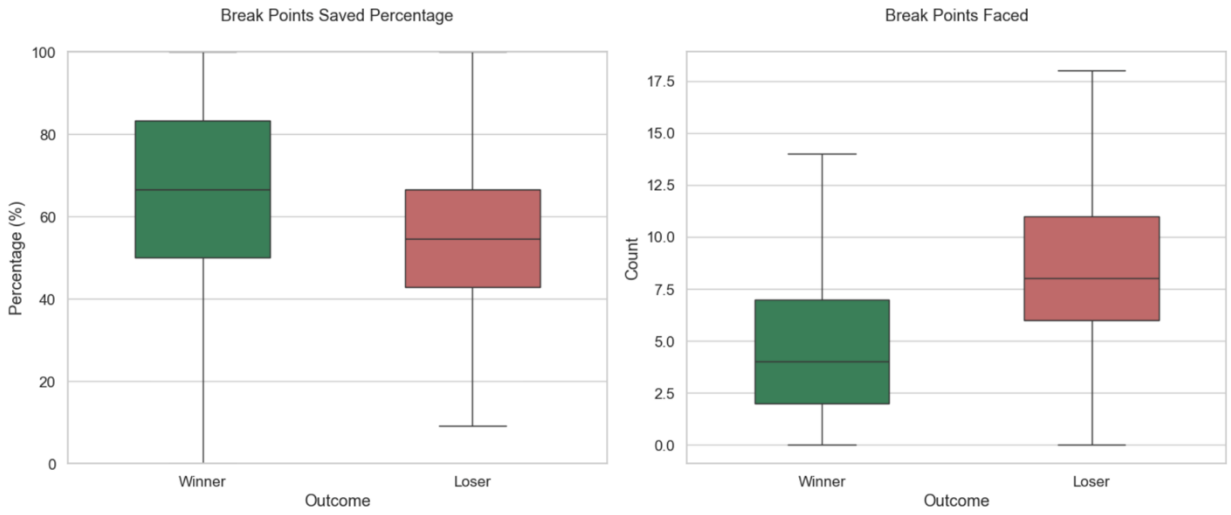

Figure 3: *Winners' versus Losers' break point statistics (Outliers removed)*

This plot show that winners outperform in break points statistics—they save a higher percentage and face fewer overall. This is the result of both better serve games performance (less break points faced) and better clutch, mentality when pressured (outperform in saving break points faced). Overall, this show that the winner is better in minimizing opponents' chances of forcing a breakpoint and also perform better when facing the pressure.

2. Feature extraction:

To plug into the machine learning model, the initial dataset needs to be mapped into an informative set of features, where each data record represents one tennis match in time, with the desired target value (win or loss).

A. Target:

From the original dataset, we have the information of the tennis matches in terms of winners' stats and losers' stats. With this setup, there is not yet a desired target value of whether a player win or not. To create a target value, firstly, the winners' and losers' features need to

be modeled into separate features for player 1 and for player 2, and the target variable would be whether Player 1 win or not:

$$Target = \begin{cases} 1 \; if \; player \; 1 \; win \\ 0 \; if \; player \; 1 \; lose \end{cases}$$

To transform from winners' and losers' data to feature of player 1 and player 2, there are several ways, one including transform one match into two different but equivalent versions where the winner will be player 1 in the first versions and player 2 in the other, for example:

Player 1 name	***Player 2 name***	***Surface***	***Target***
Roger Federer	Novak Djokovic	Hard	1
Or			
Player 1 name	***Player 2 name***	***Surface***	***Target***
Novak Djokovic	Roger Federer	Hard	0

Table: *One match can be represented in 2 different but equivalent ways*

This way will ensure that the problem of class imbalance will be avoided since the two targets (classes) will always be equal. However, this way will double the numbers of observation, may making the computation longer. Therefore, we chose to randomly assign 50% of the winners to become player 1 and the rest become player 2, the losers will be the remaining player. Doing this will not double the data size and ensure avoiding classes imbalance.

B. Previous in-match performance features

From the statistical exploratory analysis previously, we know that the winners outperform the losers in the metrics related to in-match performance. But such statistics is post-match information, and it cannot be used when predicting match outcomes. Because in reality, when we make a prediction, we have to use the information available before the match.

However, it can be argued that players who have such metrics performed better in his previous matches is whether a better player or currently in better form, and more likely to win this current match. Therefore, we will extract players' previous matches in-match performance metrics into the features. For example, for a specific performance metrics, we

extract that metric of a player in his previous match into the current match "last1" features. We also use larger time spans for a specific metric like the average of that metric in the last 3 and 5 (or larger) games and denote them as "last3" and "last5".

There are two major considerations for this method, firstly is how to deal with cases where the number of games played is not enough (players played only 2 games but the features is of the last 5 games), and secondly is the case of a player first game when previous data is not available. For the first case, we just calculate the average of the matches available, for a player played 2 games, his average performance of the last five games will only be the average of his 2 played games. Regarding the second case, we impute the average of all players' statistic for player who appears for the first time. To make it easier to interpret, we show first serve percentage features of one player in his first several matches below:

match_num	current	last1	last3
1	0.698113	0.601478	0.601478
2	0.676923	0.698113	0.698113
3	0.719298	0.676923	0.687518
4	0.758065	0.719298	0.698112
5	0.590909	0.758065	0.718095
6	0.467742	0.590909	0.689424
7	0.638298	0.467742	0.605572

***Figure 4**: Roger Federer's first 7 games first serve percentage and its past performance*

From the ***figure***, the "last1" feature is always the feature of the previous match, and the "last3" is the average of three most current previous matches. This is the way we do with all the in-match performance features that were proved to be indicative of the winner from the statistical exploratory analysis (serve points won, break points, ace, double faults...).

C. Win and loss

One of the most basic player features to capture a player recent performance is to use the number of wins he had in previous matches. One of the way to model this is to use the winning percentage in previous games. However, it is a bad feature for players with a low number of matches played. For example, a player who has played and won only one single match. Even though this player has a winning percentage of 100%, it is very likely that this

player is not as good as someone with an 80%-win rate over many more matches. For this reason, this feature will be simply modeled by the absolute number of wins.

To better capture players' recent performance, we only create features to count players' number of win in their last 20 games at maximum. If not limit the time span, we might capture the wins so long ago that are irrelevant to players' current form, and older players who played more matches will have the advantage. Lastly, we have ensured that the current match results will not be included in the win count, since this will create data leakage and unrealistically inflate the accuracy (*Illustration in Appendix 1*).

Similarly, we also create the feature to capture the number of wins, but against top players (ATP ranking) when a player faced them. Our reasoning is that more win against top players is indicative of a better form and a better player , and thus more likely to win.

D. Overall ranking and Surface Specific ELO ranking

The ATP have its ranking system, in which ranking points are awarded to a player according to how far they get in each tournament, with more prestigious tournaments worth more points. These earned ranking points are dropped 52 weeks after the tournament took place. Therefore, a player's ATP-ranking points represent his performance in the last 1 year. The higher rank points will result in a better rank with rank 1 is the highest. The rank points and ranks in the original dataset are those prior to the match, therefore can be used without any modification.

Although these two features are good indicators of players' level, there are still some drawbacks. Firstly, ranking points are awarded irrespective of the opponents while the ideal case is that the level of the opponent should also be considered in assessing the value in a win or loss. Secondly, it does not account for different court surfaces, which have a massive influence on the expected outcome of a match, due to the surface' difference in characteristics and better facilitating specific play styles. Therefore, players will have their performance varies in different surface. To compensate the features that the ATP ranking systems lack, we additionally create a surface specific ELO rating features, similar to chess ELO calculation. This feature ensure that opponent's level is taken into account, and each player will have different surface ELO for three kinds of surface Hard, Clay, Grass. The formula for Elo updating is:

$$R_1' = R_1 + K\left(S_1 - \frac{1}{1+10^{\frac{R_2-R_1}{400}}}\right); R_2' = R_2 + K\left(S_2 - \frac{1}{1+10^{\frac{R_1-R_2}{400}}}\right),$$

In which:

$$R = Elo\ rating;\ S = 1\ if\ win\ or = 0\ if\ lose; K = 32$$

For explanation, every player starts with 400 ELO points, after every match the winning player takes a number of Elo points from the losing one proportional to the Elo difference before the match. When creating this feature, it is important to ensure not using the current match result to update the current ELO, the ELO rating in every observation is the ELO of that player prior to that match (illustration in *Appendix 2*). Without this, we are using the match result itself to predict the match result.

Figure *5* show that this surface specific Elo can well capture different surface performance. Considering two players with same level, Rafael Nadal, who strength is the clay court has his clay court ELO dominant compared to Roger Federer. On the other hand, Roger Federer, considered the best grass court player, has his grass court ELO outperform Rafael Nadal.

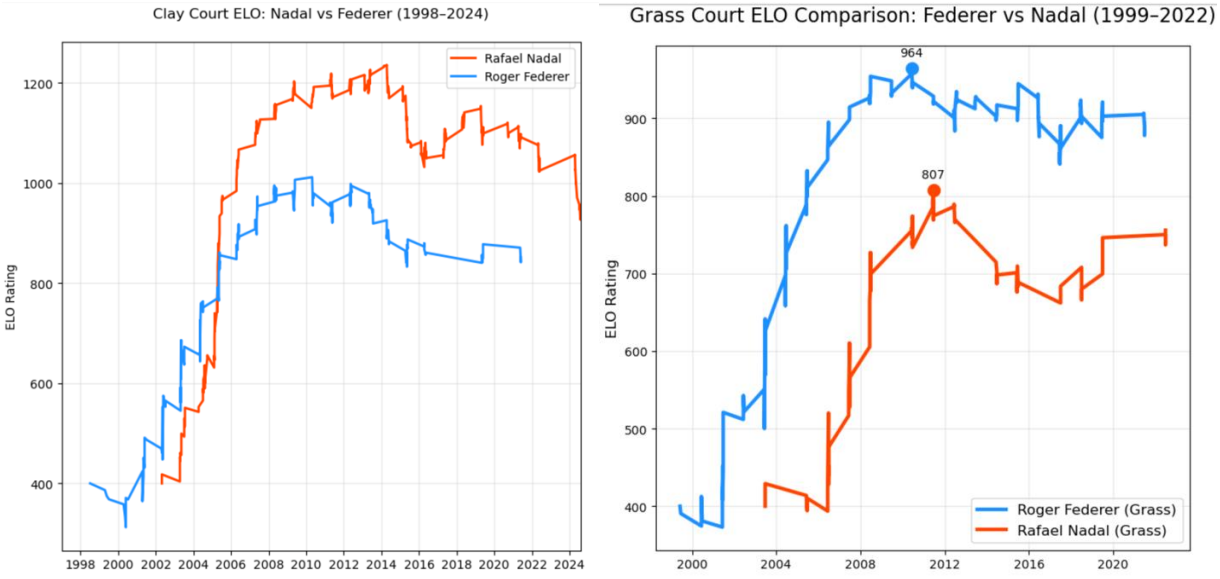

Figure 5*: Federer's and Nadal's ELO comparison in Clay court (left) and Grass court (Right)*

E. Head-to-Head

An indicative feature in any sport, is the head-to-head of the competing players. A particular playing style of a player can have an inherent advantage over some other playing style. Consequently, it makes sense that past performances against an opponent can predict future performances. Furthermore, a player's player play style may also have a specific advantage over the other's on different court surface, therefore, surface specific head-to-head result is also added. Once again, the current match result will be ensured to not be counted in the head-to-head features of the current match (Appendix 3).

F. Fatigue features

Tennis tournaments are held in short period of times, usually, two weeks. In such two weeks, players play with a high frequency as they only have one to two days rest before the next match. Therefore, stamina is an important factor to each player's performance. To capture this, we use the number of hours played by a player's previous match (in the same tournament) to indicate the player's fatigue. The reasoning is that, since players mostly have the same number of days rest before their next match, player who played more hours in his last match has his stamina more affected, thus being more fatigue. Tennis matches can last for 1 hour up to 5 hours. This feature can also be unintentionally indicative in a different way, which is that player win with less hour in his previous match is a sign of his dominance in that match, thus, possibly in a better form (player played in this match show that he won his last match).

G. Age

Age difference between players can have an effect on match outcomes. Old players near his retirement can have many disadvantage compared to a younger player in his prime. In particular, del Corral & Prieto-Rodríguez (2010) shows that the probability of a higher ranked player's victory decreases as this player competes against a younger player. Therefore, the age of each player is also added as a feature for the model.

H. Overall

To capture the relative performance of a player to each other, we create variables to capture the difference in the features of player 1 compared to player 2 (player 1's features minus player 2's features, since the target is whether player 1 win). It is similar to comparing two player's previous performance with each other when making prediction.

III. Machine Learning Models

1. Testing strategy

Both the Logistic Regression and Neural Network models employ a rigorous, time-based testing strategy to ensure robust and unbiased performance estimates, preventing data leakage by simulating real-world deployment. Initially, the chronologically sorted tennis match data is consistently split into three distinct sets: a training set (matches before 2018-12-31), a validation set (matches between 2018-12-31 and 2021-12-31), and a final, unseen test set (matches after 2021-12-31). This temporal partitioning ensures models are always trained on past data and evaluated on future data. For Logistic Regression, an inner loop performs grid search over regularization strengths (C values) on the training data, with the best model selected based on its performance (AUC-ROC) on the validation set. Similarly, for Neural Networks, the validation set is crucial for monitoring training progress and implementing early stopping, which implicitly tunes the optimal number of epochs and restores the best weights to prevent overfitting. In both cases, after the models are optimized using their respective training and validation sets, their final generalization performance is assessed only once on the completely independent and previously untouched test set, providing a reliable measure of how they would perform on truly new data.

2. Logistic Regression

a. Model description

Logistic regression is selected for its simplicity, interpretability, resistance to overfitting, and its effectiveness in binary classification tasks. In this context, it is used to predict the outcome of tennis matches, specifically determining whether Player 1 will win or lose. The model computes the probability of Player 1 winning by applying a logistic function to a linear

combination of features, such as Elo rating differences, rank points, and various performance statistics from past matches.

Approach 1 employs a direct and efficient strategy. After feature scaling with StandardScaler to normalize the input data, the model is trained once on all available historical matches up to 31st December 2018. Its performance is then immediately assessed on a single, future segment of unseen data – matches after 31st December 2021. This provides a straightforward, first-pass evaluation of how well the model generalized to new, real-world scenarios without any further tuning.

Approach 2 introduces a more robust and sophisticated validation strategy to refine the model's performance. The dataset is chronologically divided into three distinct part as in Approach 1, where it's crucial for hyperparameter tuning, specifically for optimizing the model's regularization strength (C). Throught a systematic grid search, various C values are tested, and the model's performance is meticulously evaluated on the validation set. The C value that yields the highest AUC on this validation data is selected, ensuring the model is fine-tuned for generalization without peeking at the ultimate test data. This method provides a more reliable and unbiased performance estimate when finally assessed on the completely unseen test set.

b. Results

Approach 1 (Direct Method) achieved a test accuracy of 0.657 and an AUC of 0.721. The confusion matrix revealed 2701 true negatives, 2680 true positives, 1354 false positives, and 1460 false negatives. An accuracy of 65.7% is noticeably better than random chance, indicating that the model is moderately effective in predicting match outcomes. The AUC score of 0.721 suggests that the model's probability estimates are reasonably well-calibrated and able to rank predictions fairly well. However, since this approach does not involve hyperparameter tuning, it serves primarily as a baseline. As such, it's difficult to determine whether its performance reflects the model's full potential or if it's limited by default settings and suboptimal regularization.

Approach 2, which included hyperparameter tuning, selected C=0.01 as the optimal regularization strength based on a validation AUC of 0.7085. The test accuracy of 0.656 and test AUC of 0.721 were nearly identical to those of Approach 1. Similarly, the confusion matrix and classification report showed very comparable patterns in correct and incorrect predictions. This suggests that either the dataset and feature set are relatively robust to variations in C, or the optimal value was close to the default used in Approach 1. While the predictive results were similar, Approach 2's key advantage lies in its diagnostic depth and model reliability. The tuning process enables more trustworthy generalization, reducing the risk of overfitting the test set. Moreover, the feature importance plot adds valuable interpretability by identifying key drivers of match outcomes, such as surface_elo_diff, rank_diff, and rank_points_diff. This shows that while the current ATP ranking is a good indicator of players' level and winning capability, our newly created ELO features, due to its several discussed advantages, can also be a very reliable measures and predictors.

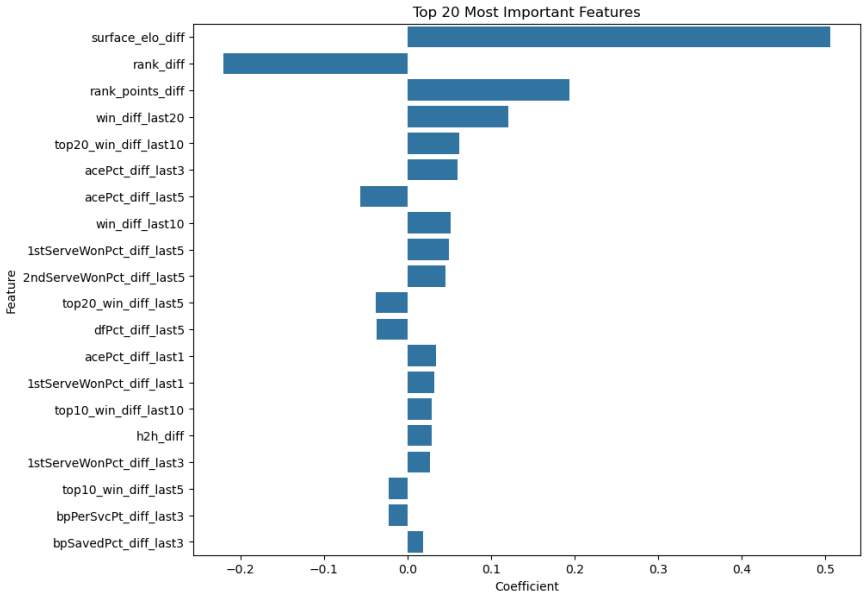

***Figure 6**: Top 20 Features that significantly affect the prediction.*

In conclusion, both Logistic Regression approaches produced similar predictive performance on the unseen test data, with accuracies around 65.6–65.7% and ROC AUC scores of approximately 0.721. This implies that the model's current feature set may already be pushing against a performance ceiling. However, despite the numerical similarities, Approach 2 is methodologically stronger. Its systematic tuning process offers a more robust

foundation for evaluating generalization performance. Additionally, its interpretability tools, such as feature importance visualizations, provide essential insights for future feature engineering and model refinement. Thus, while both approaches perform comparably, Approach 2 ultimately offers a deeper and more actionable understanding of the model's behavior.

3. Neural Network

a. Network Design

To conduct a thorough search of the best Neural Network structure on the dataset, we intended to try multiple approaches to find the most appropriate network design, starting from simpler structure, we can add into it additional hidden layers, and apply tools to search for the best hyper parameters.

For Approach 1, The network architecture for this model consists of an input layer, and only one hidden layer with 128 neurons, before the output layer. The input layer takes in the chosen features, corresponding to various match statistics and player metrics that we created, with Rectified Linear Unit (ReLU) activation functions, allow the model to learn nonlinear interactions. A dropout is also set between this layer and the output layer, to prevent overfitting by randomly dropping connections between neurons during training. The output layer produces a single prediction: the probability of Player 1 winning the match, using sigmoid function. The model is compiled using the Adam optimizer, which adapts the learning rate during training for better convergence. The model is trained using binary cross-entropy as the loss function, since it is a binary classification task (win or loss). Early stopping is employed as callbacks to prevent overfitting during training. It stops the training if the validation loss does not improve for twenty consecutive epochs (patience=20), ensuring that the model does not continue learning irrelevant noise in the data.

For the second approach, we add one more additional hidden layers with 64 neurons compared to the first approach, the aim is to potentially catch more complicated patterns in the data set. Dropout rates are also deployed between each hidden layers, the other characteristic also remained.

For the third approach, we deploy 'KerasTuner', which is a general-purpose hyperparameter tuning library (Invernizzi et al. 2019), to search for optimal hyperparameters. The Keras Tuner automates hyperparameter optimization by testing different configurations (like layer sizes, layers numbers, dropout rates, and learning rates) to maximize a specified objective metric, in our case, we used the AUC on validation set. More specifically, the search space for our KerasTuner deployment is: 64 to 512 nodes for each hidden layer, a dropout rate from 0.2 to 0.5 between the layers, the learning rates (1e-3, 5e-4, 1e-4), and number of hidden layers from 1 to 5. Through iterative trials (We set the maximum trials to 10 trials), it identifies which combinations perform best (evaluated by validation AUC) and returns the top-performing model architecture. With this search, the model structure that has the best performance in validation AUC is the one with firstly an input layer, 1 hidden layers (256 nodes) and 1 additional hidden layers (320 nodes), and an output layer. The dropout rate is 0.4 and 0.2 respectively, the learning rate is 0.001.

Approach 1	Approach 2	Approach 3
• Input layer • Hidden layer (128 nodes) Dropout (0.45) • Output layer	• Input layer • Hidden layer (128 nodes) Dropout (0.3) • Hidden layer (64 nodes) Dropout (0.3) • Output layer	• Input layer • Hidden layer (256 nodes) Dropout (0.4) • Hidden layer (320 nodes) Dropout (0.2) • Output layer

***Table 1** : Summarize of Neural Network Structures*

b. Results

The Artificial Neural Network (ANN) models demonstrated progressively stronger performance across three developmental approaches, consistently outperforming the Logistic Regression baseline. Approach 1, a foundational ANN with a single hidden layer, achieved a notable Test Accuracy of approximately 0.706 and an ROC AUC of 0.785, indicating a robust initial ability to predict tennis match outcomes. Building on this, Approach 2 introduced a deeper architecture, resulting in a marginal but consistent improvement, with a Test Accuracy of around 0.710 and an ROC AUC of 0.794. This suggests

that increasing the network's depth allowed it to capture slightly more intricate patterns within the data, leading to better discrimination between win and loss probabilities.

The final iteration, Approach 3, further expanded the network's capacity with wider layers, yielding the highest performance among the ANNs, achieving a Test Accuracy of approximately 0.712 and an ROC AUC of 0.796. While the gains from Approach 2 to Approach 3 were incremental, they confirm that even subtle architectural enhancements can contribute to better predictive power. Across all ANN approaches, the consistent use of early stopping based on validation loss proved effective in preventing overfitting, ensuring that the models generalized well to unseen data. Furthermore, the calibration curves for the more advanced ANNs demonstrated good alignment between predicted probabilities and empirical outcomes, enhancing the reliability of their probability estimates. Overall, the ANNs consistently provided superior predictive capabilities compared to Logistic Regression, with the deeper and wider architectures offering the most refined performance.

4. Comparison

Across all approaches, the ANN models consistently and significantly outperformed the Logistic Regression models in terms of both accuracy and AUC on the test data. From **Table 2**, it showed that ANN was significantly more effective at learning complex patterns within the tennis match data, leading to a higher overall predictive accuracy and better discrimination between winning and losing outcomes. The ANN's higher AUC suggests it is better at ranking positive instances higher than negative instances, making it a more powerful classifier.

***Table 2** : Summarize of Test Accuracy and AUC of Logistic Regression and ANN*

	Logistic Regression		**Neural Network**		
	Approach 1	**Approach 2**	**Approach 1**	**Approach 2**	**Approach 3**
Accuracy	0.6566	0.6563	0.7056	0.7101	0.7148
AUC	0.7207	0.7208	0.7852	0.7936	0.7957

A calibration curve assesses how well a model's predicted probabilities align with actual outcomes; a perfectly calibrated model predicting a 70% chance of an event means that event occurs 70% of the time among all such predictions. This metric is crucial for interpreting

results as it determines the trustworthiness of probability estimates, which is vital for informed decision-making beyond mere classification accuracy.

Comparing the best-performing Logistic Regression and Artificial Neural Network models reveals distinct strengths. The LR model (Approach 2) achieved a Test Accuracy of 0.656 and an ROC AUC of 0.721, demonstrating decent overall performance. In contrast, the ANN model (Approach 3) significantly outperformed LR in predictive power, yielding a Test Accuracy of 0.712 and an ROC AUC of 0.796, indicating superior performance (***Figure 7***).

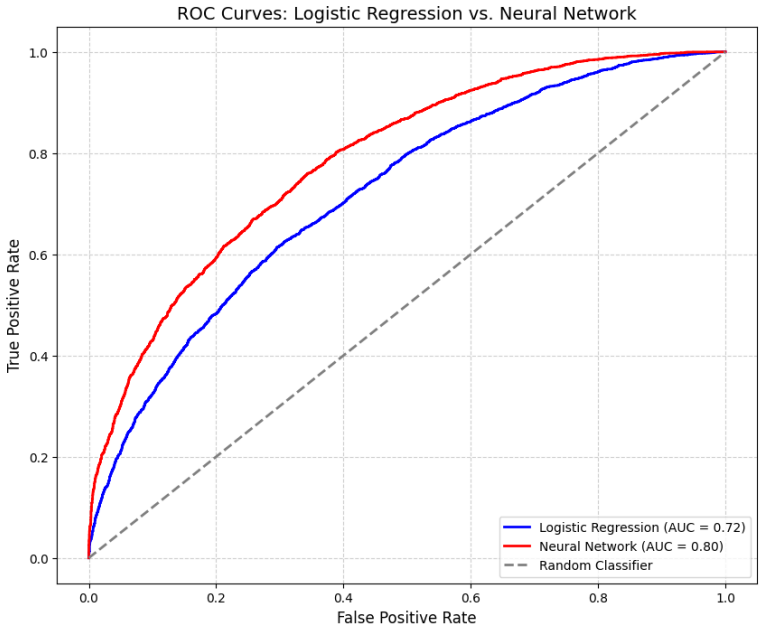

Figure 7*: ROC curves for the logistic regression and neural network*

Despite being outperformed in terms of AUC and accuracy, LR performs well regarding to calibration curve, having a similarly good alignment with the true probability line compared to the NN (***Figure 8***). Both models' good performance in calibration curve shows that not only they can predict the outcome, but its predicted probability is also reliable, which can be useful many applications.

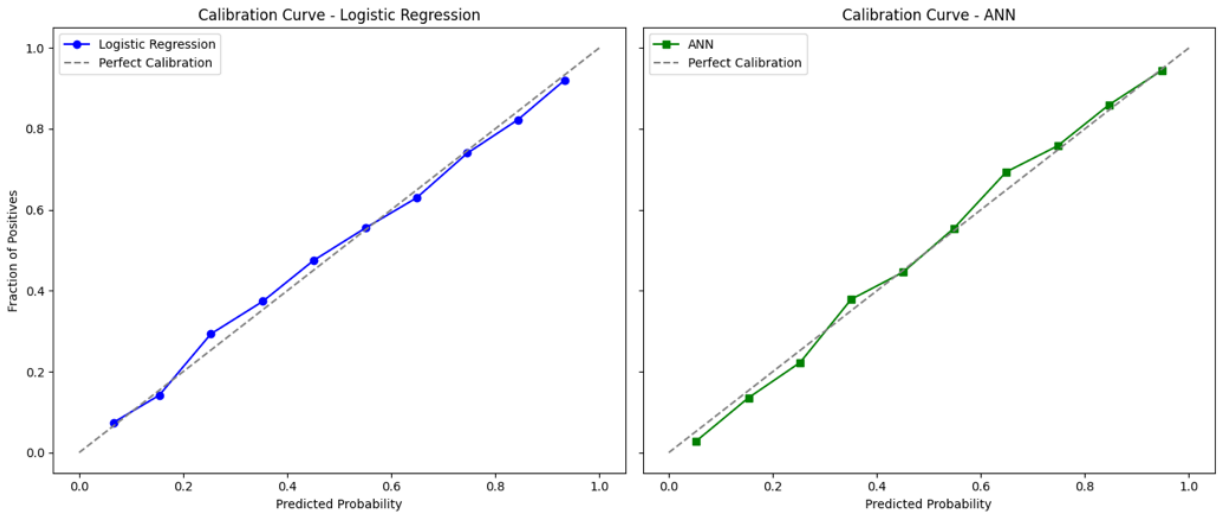

***Figure 8**: Calibration curves for the logistic regression and neural network*

IV. Application (Betting market)

The overall sport betting market has been increasing rapidly in recent years, and tennis betting is not an exception. Tennis is the third most popular sport in the sport betting market, ranking by total monetary value (PWC 2024).

Betting is usually placed through a bookmaker, before the matches, the bookmaker assigns odds to each possible outcomes. These odds represent how probable different outcomes of a bet are. If an outcome is expected to be unlikely to occur, it will give higher payout. In contrast, the more expected outcomes will have a lower payout.

We can look at the odds of the Australian Open Finals in 2025 to have a better understand. At the start of the match, the odds for Jannik Sinner to win is 1.33, meaning that a dollar bet on Jannik Sinner will result in a profit of 0.33 dollar if he wins. While the odd for Alexander Zverev is 3.4, thus if he wins, a one-dollar bet will have a return of 3.4 dollars (2.4 dollars of profit). This betting odds can be very useful if we convert it into the implied winning probability of a player, which can be done by taking the inverse of the odds. The results probability then represents the expected probability of an outcome occurring according to the bookmaker. For example, regarding to the match we mentioned, the implied probability of winning for Jannik Sinner is 75.19% and for Alexander Zverev is 29.41%. The add-up of these exceeds 100% is usually called the bookmaker margin.

In both Machine Learning Models we built, the final output came from sigmoid function, thus, it represents the probability of Player 1 win. This leads to an application of the machine learning models in the betting market, in which, if the model's prediction of winning probability is higher than the implied probability of the odds, it means that the model suggest that this bet is profitable. And if a model can somehow know for a fact an outcome has a higher probability than its implied probability, it is always profitable to bet on that outcome in the long run. Although, our best model, have a calibration curve closely resembles the true probability curve, which means that the model's predicted probabilities accurately reflect the true likelihood of events, it is certain that it cannot achieve the state of "know for a fact" the true probability, a simple strategy of just placing a bet when the model's probability is higher than the implied probability will likely leads to a lost.

Therefore, a profitable application of the model into betting strategy should not bet on every match that is suggested by the model to be profitable. A better strategy and application should only bet or bet more on matches where there is more confidence (the model's predicted probability is significantly higher than the odds implied one, etc.) and the opposite when the uncertainty is high. However, determine a profitable strategy is out of the scope of this project.

V. Conclusion

Through rigorous feature engineering, including surface-specific ELO ratings, head-to-head records, fatigue indicators…, we achieved test accuracies of 65.7% with logistic regression and 71.48% with neural networks, surpassing the typical 65–70% for this problem. The neural network, with its ability to capture nonlinear relationships, demonstrated superior predictive power. Not only the accuracy, other metrics like AUC and calibration curve also exhibits positive results, with the best AUC of 0.8, and a calibration curve closely represents the true likelihood line. The models showed potential for betting market applications, due to their well performed calibration curve. However, further refinement is needed to translate probabilities into profitable strategies, which is out of the scope of our project and might be an interesting problem in a future work.

VI. Reference

De Seranno, A 2020, *Predicting Tennis Matches Using Machine Learning*, viewed 10 May 2025, <https://libstore.ugent.be/fulltxt/RUG01/002/945/727/RUG01-002945727_2021_0001_AC.pdf>.

Del Corral, J & Prieto-Rodríguez, J 2010, 'Are differences in ranks good predictors for Grand Slam tennis matches?', *International Journal of Forecasting*, vol. 26, no. 3, pp. 551–563, viewed 3 May 2025, <https://www.sciencedirect.com/science/article/abs/pii/S0169207009002076>.

Invernizzi, L, Long, J, Chollet, F, O'Malley, T & Jin, HJ 2019, *Keras documentation: Getting started with KerasTuner*, Keras.io, viewed 22 May 2025, <https://keras.io/keras_tuner/getting_started/>.

Lisi, F & Zanella, G 2017, 'Tennis betting: Can statistics beat bookmakers?', *Electronic Journal of Applied Statistical Analysis*, vol. 00, Universita del Salento, no. 00, pp. 1–35, viewed 27 May 2025, <https://www.researchgate.net/publication/310774506_Tennis_betting_Can_statistics_beat_bookmakers?__cf_chl_rt_tk=mSLSsnS3zJrZ1etlcW.aGogeenyPKPnmTuvT5XDaI0A-1748329413-1.0.1.1-EFj19iDcTtlocoJHbwdWj5ajL2Svj0WjRfRFZ354plY>.

Mateus 2017, 'Using Soft Computing Techniques for Prediction of Winners in Tennis Matches', *Machine Learning Research*, vol. 2, Science Publishing Group, no. 3, p. 86, viewed 27 May 2025, <10.11648/j.mlr.20170203.12>.

Pham, C & Bufi, K 2023, *Predicting Tennis Match Results Using Classification Methods*, viewed 27 May 2025, <https://lup.lub.lu.se/luur/download?func=downloadFile&recordOId=9121180&fileOId=9121181>.

PWC 2024, *Sports betting handle by sport worldwide 2023| Statista*, Statista, viewed 10 May 2025, <https://www.statista.com/statistics/1534873/leading-sports-handle-betting-worldwide/>.

Sackmann, J 2025, *GitHub - JeffSackmann/tennis_atp: ATP Tennis Rankings, Results, and Stats*, GitHub, viewed 15 May 2025, <https://github.com/JeffSackmann/tennis_atp>.

VII. Appendix

	player1_name	player1_win_last5	player1_win_last10	player1_win_last20
0	Emilio Sanchez	0	0	0
1	Ivan Lendl	0	0	0
2	Steve Guy	0	0	0
3	Horst Skoff	0	0	0
4	Jean Philippe Fleurian	0	0	0
5	Martin Jaite	0	0	0
6	Eric Jelen	0	0	0
7	David Wheaton	0	0	0
8	Richard Fromberg	0	0	0

Appendix 1: *First several matches of the dataset, none of the players played a match, therefore win count is zero since we ensure not to count the current match*

surface	player1_name	player1_surface_elo	player2_name	player2_surface_elo	surface_elo_diff
Hard	Emilio Sanchez	400.0	Renzo Furlan	400.0	0.0
Hard	Ivan Lendl	400.0	Wally Masur	400.0	0.0
Hard	Steve Guy	400.0	Malivai Washington	400.0	0.0
Hard	Horst Skoff	400.0	Derrick Rostagno	400.0	0.0
Hard	Jean Philippe Fleurian	400.0	Brett Steven	400.0	0.0
Hard	Martin Jaite	400.0	Guillaume Raoux	400.0	0.0
Hard	Eric Jelen	400.0	Gilad Bloom	400.0	0.0
Hard	David Wheaton	400.0	Andrei Cherkasov	400.0	0.0

Appendix 2: *First several matches of the dataset, none of the players played a match, therefore ELO is all 400 since we ensure this is the pre-match ELO*

Head-to-Head Features for Federer (ID:104925) vs Nadal (ID:104745):

	tourney_date	tourney_name	surface	round	winner_id	h2h_diff_fed	h2h_surface_diff_fed
46863	2006-05-29	Roland Garros	Clay	QF	104745	0	0
49108	2007-03-05	Indian Wells Masters	Hard	F	104745	-1	0
49199	2007-03-19	Miami Masters	Hard	QF	104925	-2	-1
49517	2007-05-07	Rome Masters	Clay	QF	104745	-1	-1
49747	2007-05-28	Roland Garros	Clay	SF	104745	-2	-2
50014	2007-06-25	Wimbledon	Grass	SF	104745	-3	0
50439	2007-08-05	Canada Masters	Hard	SF	104925	-4	0
51149	2007-11-12	Masters Cup	Hard	RR	104745	-3	1
51900	2008-03-13	Indian Wells Masters	Hard	SF	104925	-4	0
52337	2008-05-11	Hamburg Masters	Clay	SF	104745	-3	-3

Appendix 3: *First several matches of the Federer and Nadal pair, in their first matches, the result head to head difference is 0 since we ensure not to include the current match*