Who Will be the Tennis Winners?

by Mingkai Liu

2020-08

Table of Content

1. Introduction: Background and Business Problem

Sports betting industry has a big market in Australia. Nationally, the typical monthly sports betting expenditure amounted to roughly $579 million over the year [1]. Tennis is one of the sports that are favored in the jackpots, and betting winning/losing is the simplest case in betting. Nowadays, jackpots have advanced models that can provide a higher-than-average prediction accuracy and therefore can set odds that are statistically profitable to them.

In this project, we will examine the historic data of tennis games, using male singles as an example. We will develop a pipeline to analyze these data and provide some insights, for example:

(a) Who are the top players and how they performed?
(b) How has each player's performance changed over time and what are the technical statistics?
(c) Who are the top opponents of a given player and how their battle records changed over time?
(d) What factors and features are correlated with the winning odds?
(e) Can we come up with some machine learning models that can out-perform the prediction accuracy of traditional Elo rating?

2. Data Preprocessing

The historic tennis data is downloaded from https://github.com/JeffSackmann/tennis_atp.

We chose the data from the past 40 years starting from 1980 to 2020. The data come in a matrix form with 135825 rows and 48 feature columns. The features include the match information, winner/loser personal information, winner/loser technical performance.

Table I summarize the features in the original data. Most of the feature names are self-explanatory, while some are short and explained.

Match information	Winner's information	Loser's information	Winner technical performance	Loser technical performance
tourney_name	winner_id	loser_id	w_ace	l_ace
surface	winner_seed	loser_seed	w_df	l_df
draw_size	winner_entry	loser_entry	w_svpt	l_svpt
tourney_level	winner_name	loser_name	w_1stIn	l_1stIn
tourney_date	winner_hand	loser_hand	w_1stWon	l_1stWon
match_num	winner_ht	loser_ht	w_2ndWon	l_2ndWon
score	winner_ioc	loser_ioc	w_SvGms	l_SvGms
best_of	winner_age	loser_age	w_bpFaced	l_bpFaced
round	winner_rank	loser_rank	w_bpSaved	l_bpSaved
minutes	winner_rank_points	loser_rank_points		
best_of: the maximal number of games in the match, usually it's 3 or 5, *_entry: the way the player enter the tourney, for example, qualified(Q), wild card(WC), etc. *_ioc: country or region the player represents *_svpt: serviced points; *_1stIn: 1st serve in; *_1stWon: 1st serve won; *_SvGms: serviced games *_bpFaced: break point faced; *_bpSaved: break point saved;				

Tabel I: Column features in the original data sets

Although the data is in a matrix format, it includes both structured and unstructured data, typos, missing values and other artifacts that need to be taken care of before further analysis.

The *tourney_date* was recorded as np.float64, it is converted to np.datetime64 format. We further extracted the information of *year* and *month*.

The score was recorded in an unstructured text format like

'3-6 6-7 7-6 7-6 5-3 ABD',
'2-6 4-6 6-3 7-6(11) 5-2 RET',
'6-8 6-8 1-1 Played and unfinished', etc.

We converted the scores of each set into numeric format and recorded separately for winner and loser as *winner_set_N*, *loser_set_N* (N=1,2,…,5). Further we extracted the number in the bracket as the tie-break point of the set, *set_N_tb*.

Somehow there are strange scores recorded as months like 'Feb', 'Jun', 'Apr', etc, or '00'. These kind of typos are all corrected to numerics.

We leave the null value as it is now and will decide how to deal with them after exploratory data analysis.

3. Exploratory Data Analysis

3.1 Basic information of tourney and games

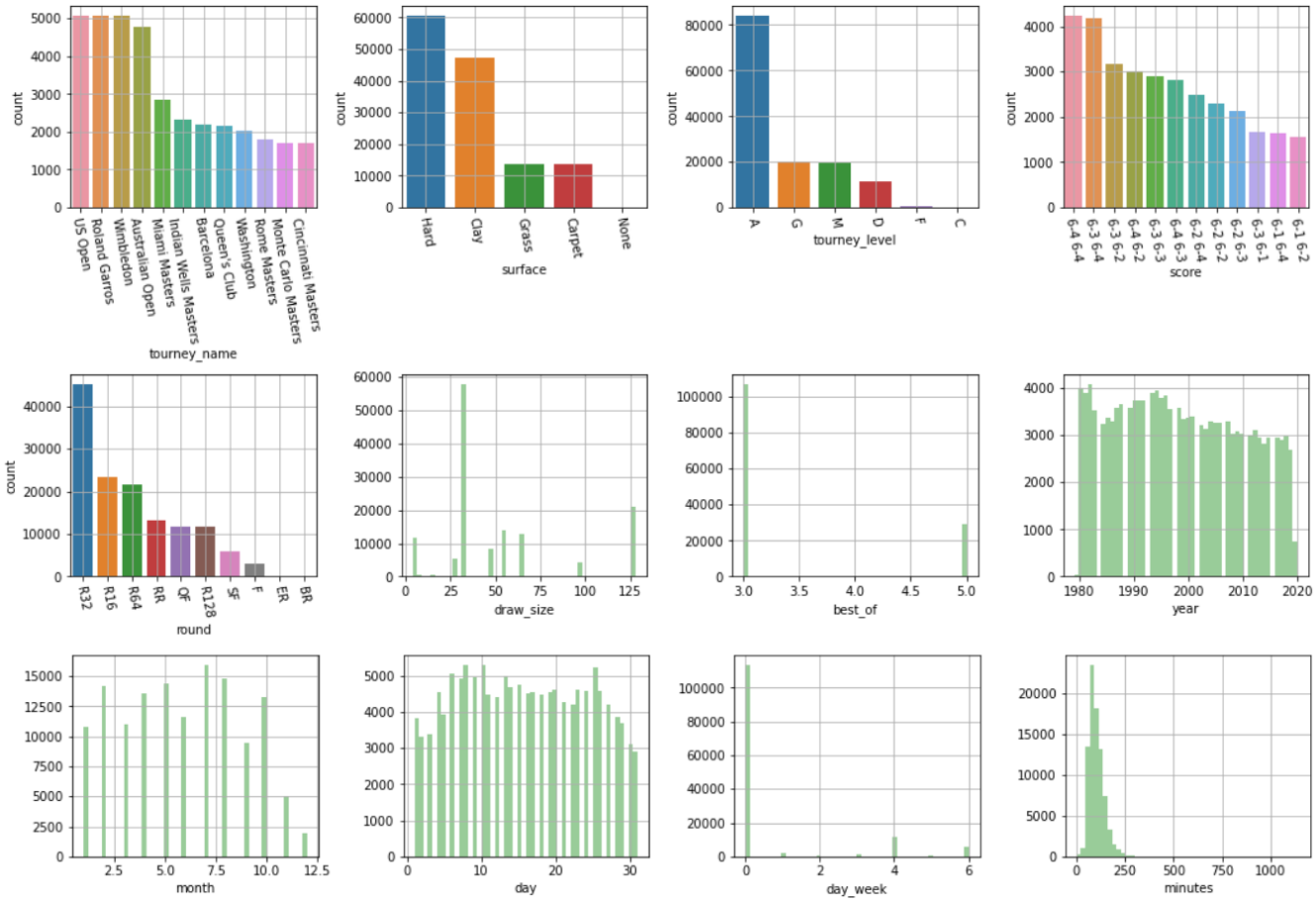
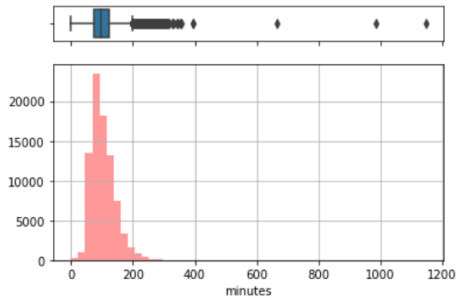


Figure 1. Distribution of the tourney information



	tourney_name	tourney_date	winner_name	loser_name	score	best_of	round	minutes
107468	Wimbledon	2010-06-21	John Isner	Nicolas Mahut	6-4 3-6 6-7(7) 7-6(3) 70-68	5.0	R128	665.0
123713	Sydney	2016-01-11	Gilles Muller	Jeremy Chardy	7-6(5) 6-3	3.0	QF	1146.0
127779	Munich	2017-05-01	Hyeon Chung	Martin Klizan	6-4 3-6 6 6-2	3.0	QF	987.0

Figure 2. Outliers of the time lengths of games

Figure 1 shows the distribution of the tourney information. For features that are categorical, we show the top 12 most frequent categories. The top 4 *most popular tourneys* are US Open, Roland Garros (French Open), Wimbledon and Australia Open. Hard surface is the most common, followed by clay. Round 32 is the most common, which makes sense since the most common draw_size is also 32. The *most common scores* in a match is (6-4 6-4), followed by (6-3 6-4); but it's surprising to see that the third most common score is not (6-4 6-3) but (6-3 6-2). In fact, (6-4 6-3) is the 6th most common score, which is quite unexpected.

Another observation is that the *total number of matches* played each year is gradually declining, does it mean tennis overall becomes less popular, or many smaller tourneys are gradually disappearing since major sponsors of the tourneys are more willing to focus their advertisements on major well-known tourneys? That's something deserves further research from the market point of view.

We also note that, while the *time lengths of games* are predominantly around 90 minutes, there were some significant outliers that last hundreds or even thousands of minutes. We filter these outliers in Figure 2. The match between John Isner and Nicolas Mahut was the most famous longest match -- look at the score of the final set!. While the other two are more likely to be paused in the middle of the game due to weather or other reasons.

3.2 Basic information of winners and losers

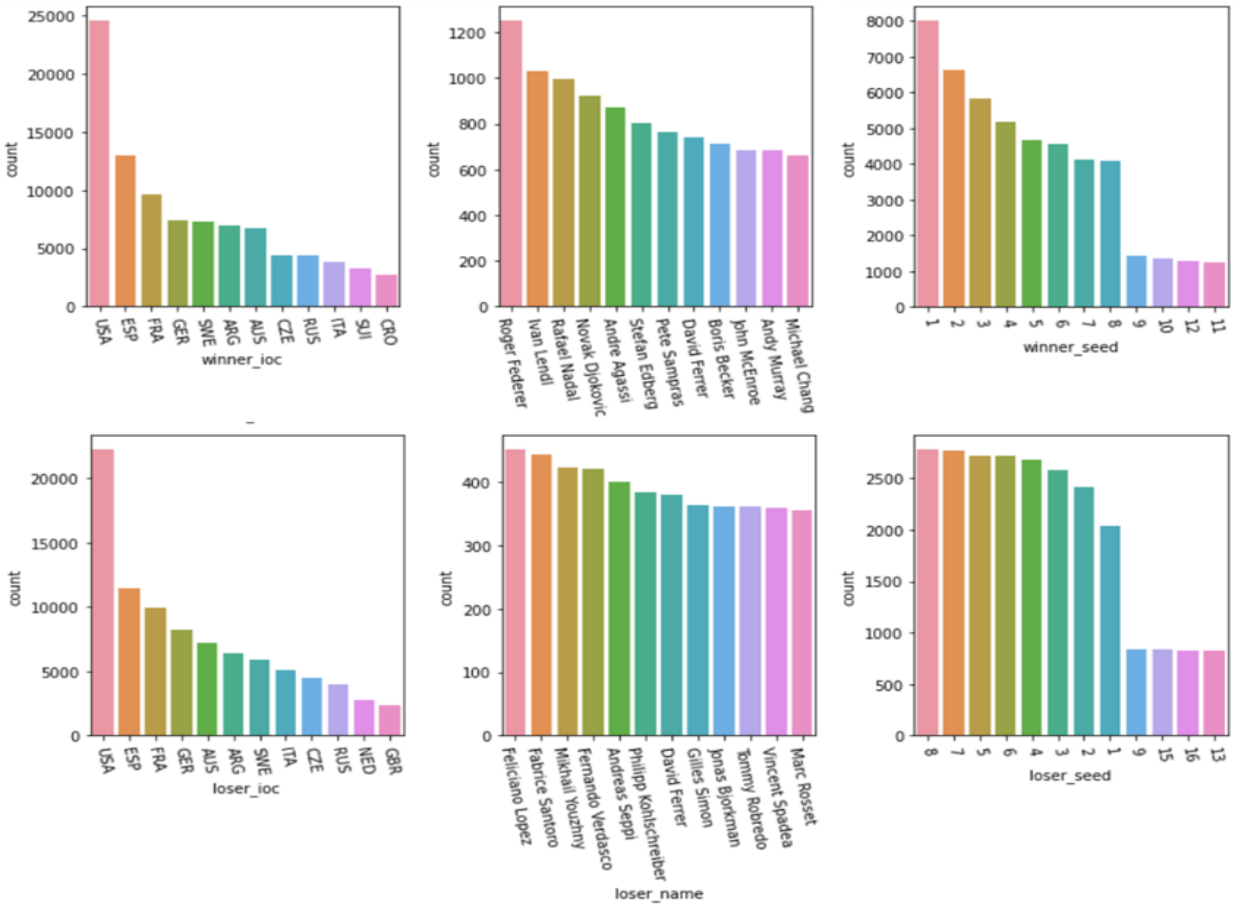


Figure 3. The most common categorical features for winners and losers

Figure 3 shows which countries most players come from, who won/lost most games and what are the most common seeds of the winners and losers. It shows that over the years (1980-2020) we studied, the majority of players come from USA, followed by Spain, France, Germany, etc. Among the players, Roger Federer won the largest number of games, followed by Ivan Lendl, Nadal, Djokovic and Agassi. Some of the most common losers are also highly ranked as the most common winners, such as David Ferrer. It is reasonable since it means they are also very good players to be qualified and competed in many games.

A very interesting observation is the seeds of winners and losers. The number of times a players with winner_seed winned a game pretty much follows the seed order from 1,2,3, ... to 12, and the number of wins have an abrupt drop from the 8th place. While the loser_seed pretty much follows an inverse order: from 8,7,... to 1, and an abrupt change also occurs at the 8th place. Basically, players with seed 1 to seed 8 dominate the top 8 numbers of winning or losing a game. It makes sense since the number of seed players is determine by

the draw size (number of participants). For the most common one with a draw size of 32 (as shown above), there will be 8 seed players, and for draw size of 64, there will be 16 seed players. It means, the seed of a player is a strong indicator of his winning potential, which is an important factor to considered in the prediction.

Figure 4 shows similar observation. Players within top 50 rank in general have won more than lost. The higher the rank, the larger the rank_point, the more games a player has won, and the difference between the number of won and lost games becomes more apparent.

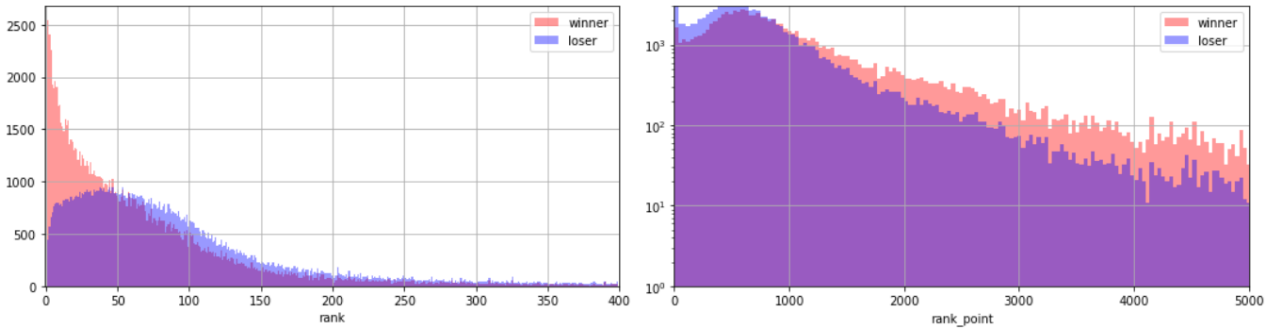


Figure 4. Distribution of ranks and rank_points of winner and loser

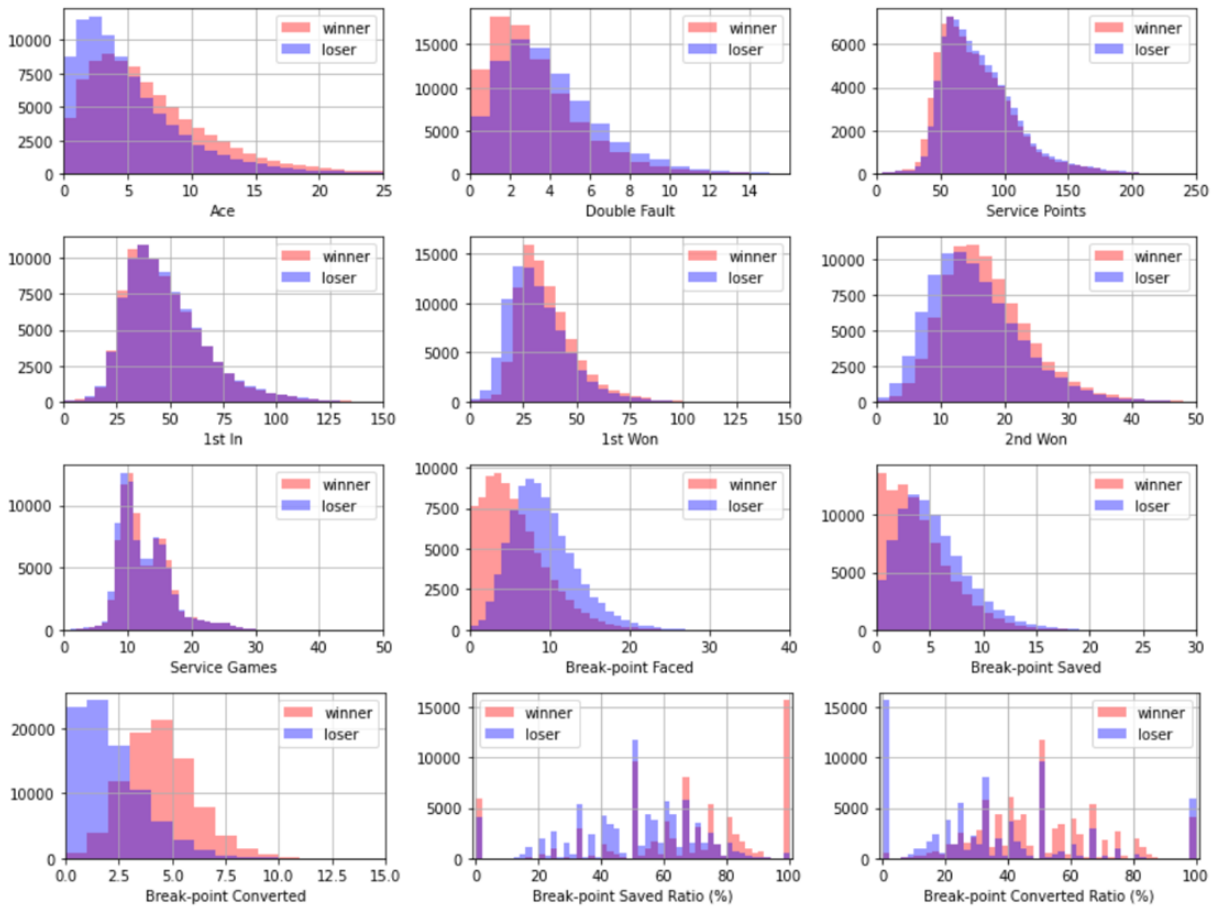


Figure 5. Distribution of technical performance of winners and losers

Apart from basic ranking information, is there any difference in terms of technical performance between winners and losers? Figure 5 summarized the findings. Winners have more ace, fewer double fault, more 1st won and 2nd won. But not much difference in terms of service points, 1st in and service games.

The most notable feature is related to the break-point performance. Apart from the break-point faced and break-point saved that have already been listed in the original data, we created new technical feature called

break-point converted. It means how many break point a player have won from his opponents when he won the set:

winner_break_point_converted = loser_break_point_faced - loser_break_point_saved,
winner_break_point_converted_ratio = winner_break_point_converted / loser_break_point_faced,

and vice versa for *loser_break_point_converted* and *loser_break_point_converted_ratio*.

From Figure 5, we can see a clear difference in the break point performance between winner and loser. Winners have faced fewer break points, and thus saved fewer break points; but have converted more break points. From the ratio we can clearly see that winners have high break-point saved ratio and higher break-point conversion ratio.

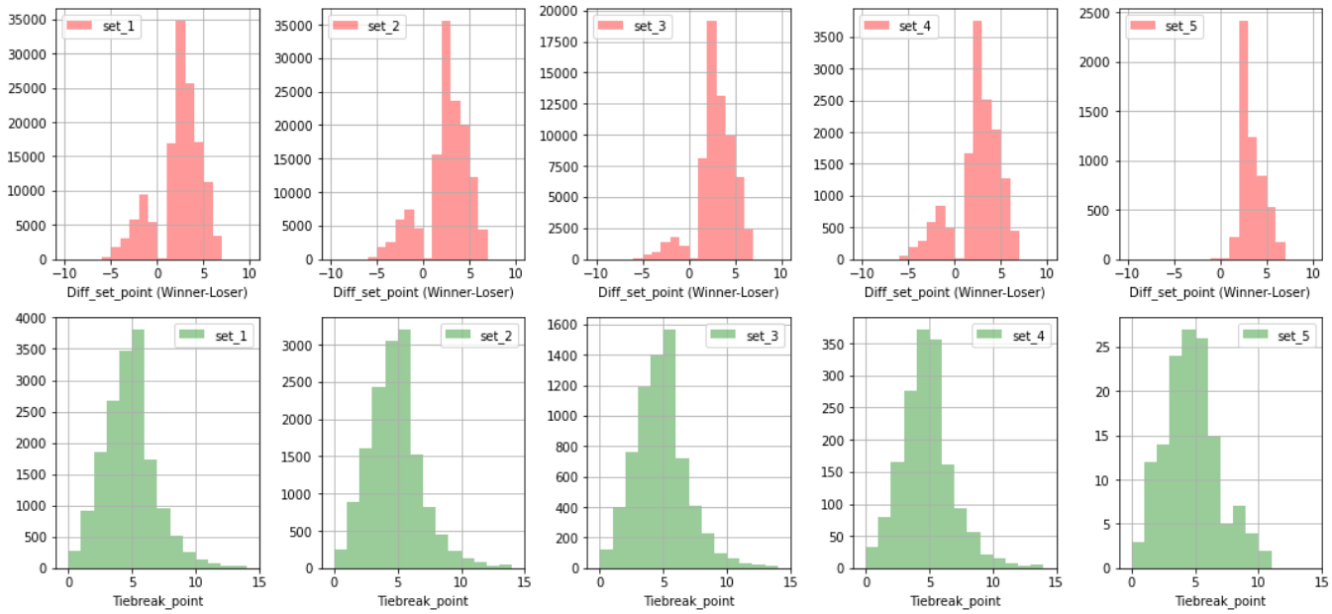


Figure 6 Distribution of set point difference (winner – loser) and the tie break points.

Since we have extracted the information of set points and tie break points, we plotted their distribution, as shown in Figure 6. We don't really see any significant difference. But it does confirm that the most common set point difference is 2. If we are only interested in the final outcomes of winning/losing, it's perhaps simpler to calculate the total set points difference as a figure to show the performance difference.

3.3 Comparing different players

While the data in the initial matrix are recorded based on winner/loser, we need to re-arrange the information as *player_1* and *player_2*. Since we aim to develop a model to predict whether a player will win the game when the opponent's information is also given, we can use the following simple convention to decide who's *player_1* within a pair of players: the one whose *player_id* is smaller numerically will be assigned as *player_1*, and the opponent will be assigned as *player_2*. For a given player, he can be *player_1* or *player_2* in different competing pairs, depending on whether his *player_id* is smaller than his opponent's.

We can now compare the evolution of the performance and ranking of different players over the years. As an example, we show the currently top 3 players: Roger Federer, Rafeal Nadal, Novak Djokovic. While most of the technical performance are very close, we still see some noticeable difference in items such as ace, 2nd Won, break-point conversion. The performance of Federer is very stable over the years, even today, 15 years after he peaked at around 2005. Federer also have higher number of ace, but lower break point converted.

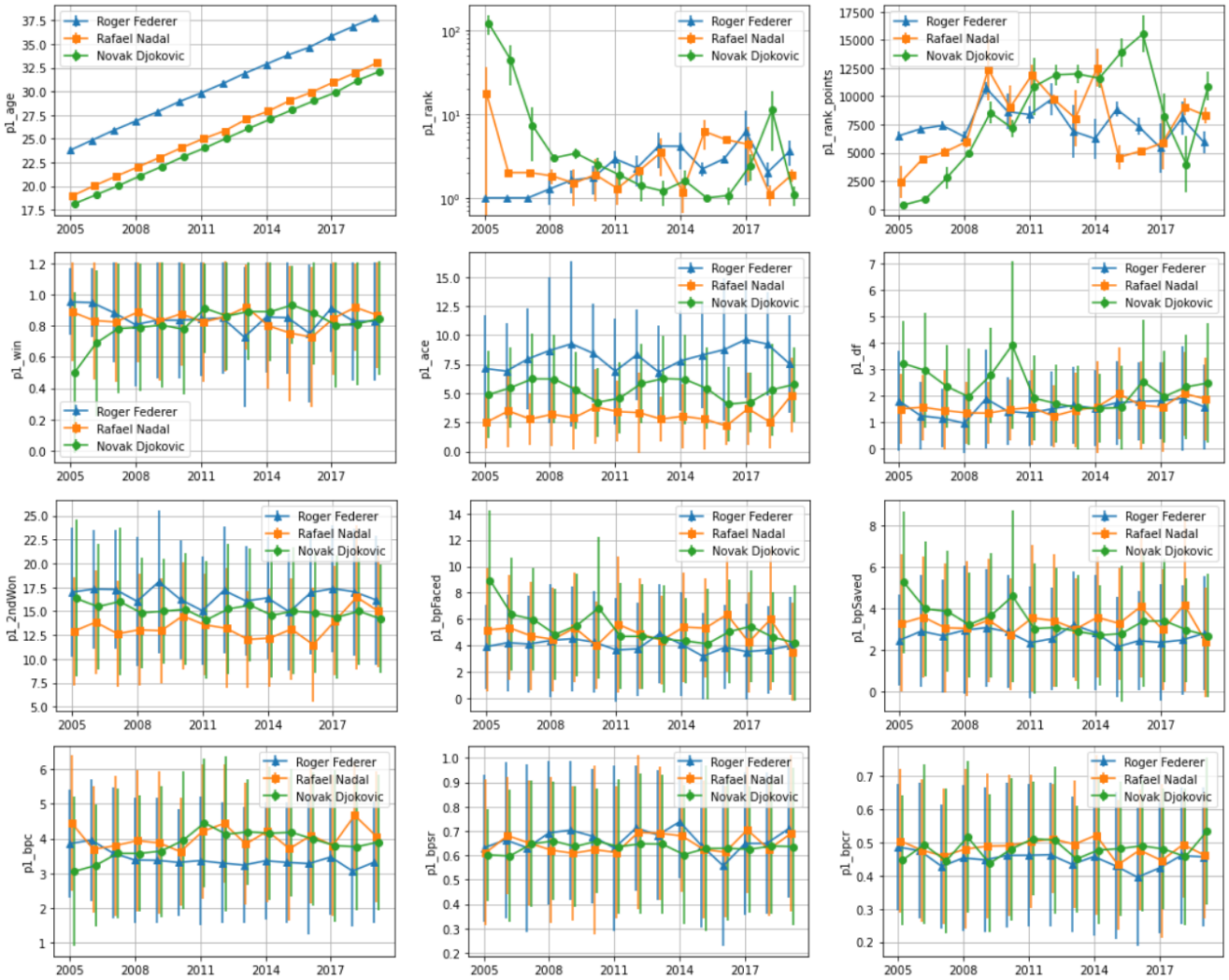


Figure 7. Performance comparison of Roger Federer, Rafeal Nadal, Novak Djokovic.

Another interesting thing to look at is who are the top opponents of each player. And how the results change over the years. Figure 8 showed the top 6 opponents of Federer and Djokovic. For clarity, we showed the moving averaged winning rate and the number of games they have competed over the past 5 years.

Djokovic was improving and rising fast and right now he's at the top of his career, after 2012, his winning ratios are all above 0.5 when facing his top 6 opponents including Nadal, Federer, Murray. Federer also kept his high performance steady. In particular, his winning ratio over his once tricky opponents Nadal and Murray increased over the years from below 0.4 to above 0.8 – probably he has adjusted his tactic when facing Nadal and Murray, but most importantly and amazingly, he managed to keep himself in very high performance even though he has way passed his prime time!

We can further summarize the conditions in which two players competed. In Figure 9, we showed the counts of tourneys, surfaces, and other factors in which Federer and Djokovic competed,. The two players were mostly competed in the round semi-final and final. Since the competition between these two top players were quite severe, the time lengths of the games were mostly in between 100 and 150 minutes, and peaked at around 130 minutes, which were 30-40 minutes longer than the average length (around 90 min) shown in Figure 2.

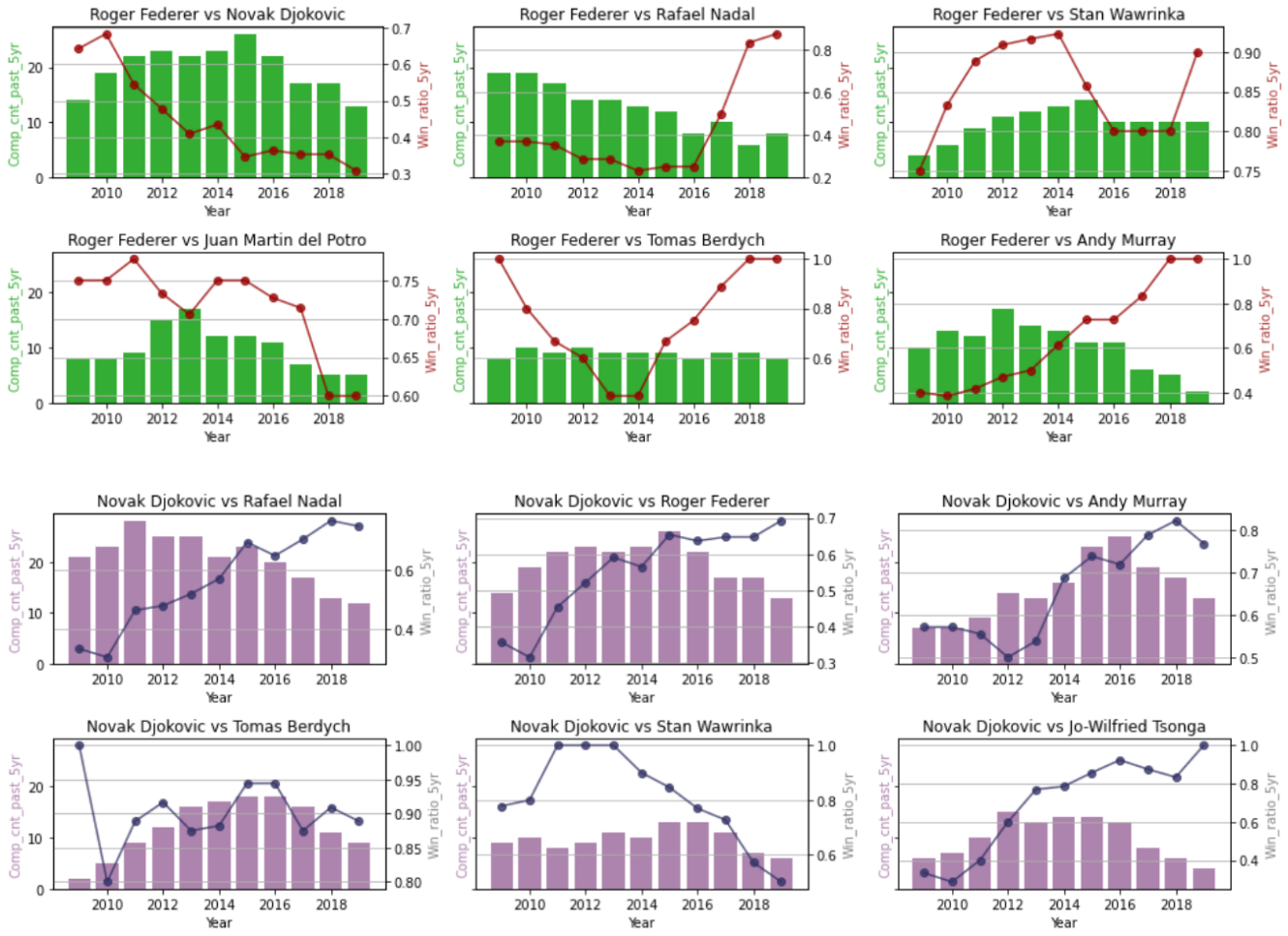


Figure 8. Performance change of Roger Federer, and Novak Djokovic over their top 6 opponents.

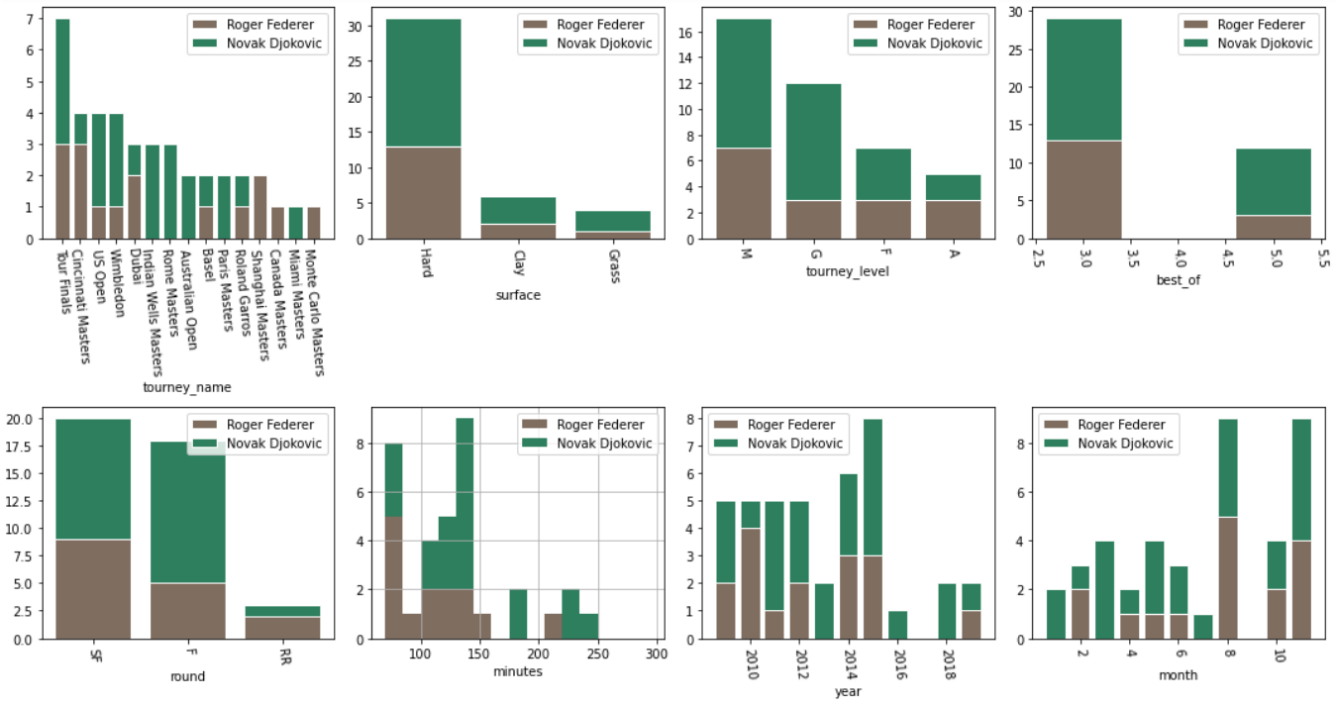


Figure 9. Roger Federer vs Novak Djokovic. Brown: Federer won, Green: Djokovic won

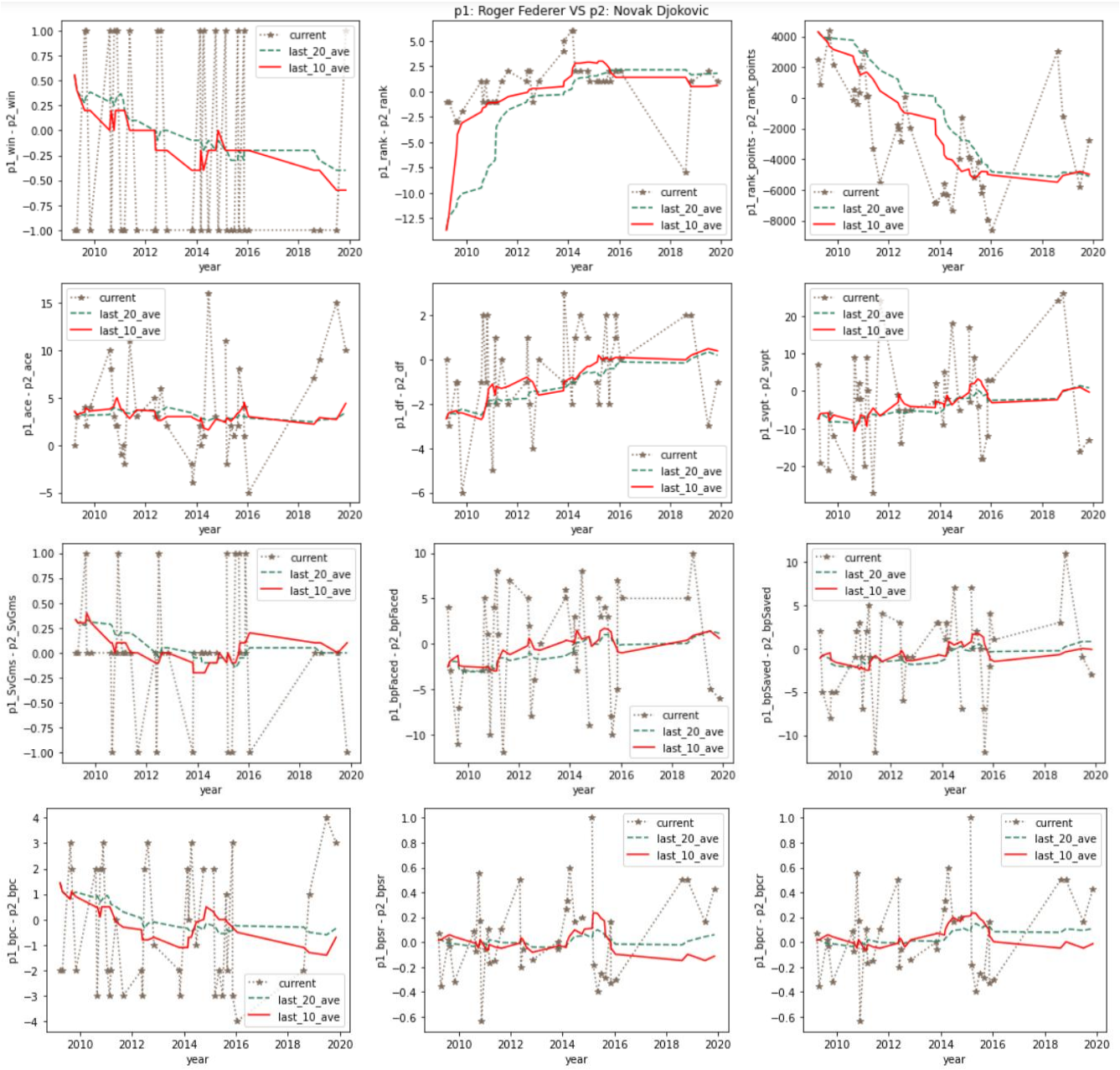


Figure 10. Performance of Roger Federer vs Novak Djokovic.

We can further analyze how the technical performance of the two players changed over time when they compete. We are only interested in their relative comparison, so we only plot the difference of Federer and Djokovic in Figure 10. The brown stars represent the difference at that particular game, while the green and red curves represents the averaged figures over their last 20 and last 10 games. We can see that the performance figures we found in Figure 5, in particular the higher break point conversion of Djokovic, is related to the higher ratio of winning. Although Federer served more ace, but it's clearly not the deterministic factor of winning the game.

4. Feature Engineering

4.1 Elo Rating

One of the most well-known models for characterizing players' ranks and the probability of winning in zero-sum games is Elo rating [2]. The rating system was introduced by physicists Arpad Elo for chess competitions and now widely applied in different types of zero-sum games including tennis, basketball, video games, etc. [3]

The basic idea of Elo rating is to incrementally adjust a player's rating based on not only the result (win/lose) of the game, but also the opponents rating. If a player wins an opponent that has a higher Elo rating, the player's Elo rating will also be increase with a larger amount.

Each player starts with an Elo rating of 1500, and player A's winning probability (based on logistic curve) when facing an opponent with an Elo rating of $\mathrm{Elo_B}$ before a match is given by

$$\mathrm{p_{A,win}} = \left[1 + 10^{\frac{\mathrm{Elo_B} - \mathrm{Elo_A}}{400}}\right]^{-1}.$$

The 400 in the formula is designed such that a difference of 200 rating points ($\mathrm{Elo_A} - \mathrm{Elo_B}$) would mean that player A has an expected winning probability of approximately 0.75.

The Elo rating of a player will be updated after the j_{th} match

$$\mathrm{Elo_{j+1}} = \mathrm{Elo_j} + \mathrm{K}\,(\mathrm{R} - \mathrm{p_{win}}),$$

where $\mathrm{R} = 1$ means the player has won and $\mathrm{R} = 0$ for losing the game.

The K factor can be a constant, but more accurate model suggest to use a decay function

$$\mathrm{K} = \frac{\mathrm{C}}{(\mathrm{M}+\mathrm{o})^{\mathrm{s}}}$$

where C is a constant, M is the number of matches in the player's dataset, o is a small offset (to avoid very large values when M is low), and s is a shape parameter which allows for more flexibility in the curve's shape.

In fivethirtyeight.com, they use the following parameters: C=250, o=5, s = 0.4.

Increasing C will increase the weight of each matches, while increasing s will decrease the weight of future games on Elo rating.

Using the procedure above, we iterated through the historic data and set the starting point of Elo calculation from 1980. To test if the parameter *s* will affect the prediction accuracy, we also calculated Elo ratings and the winning probability for different values of *s.*

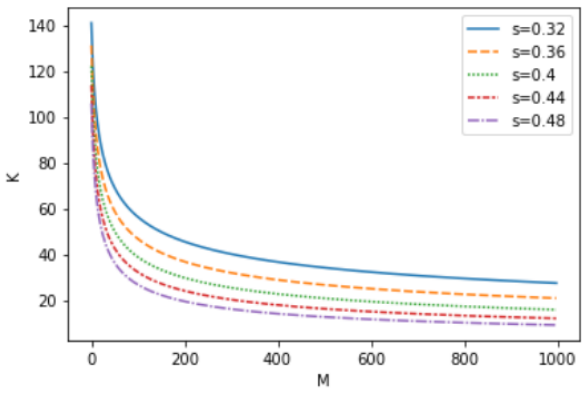


Figure 11. K coefficient vs M (number of games played) for different s values.

4.2 All features

Personal & Tourney information	Past combat records	Elo ratings	Recent competition performance
year, month, tourney_level, surface, best_of, round, pair_names, p1_entry, p2_entry, seed_code_1-2, rank_1-2, rankpoint_1-2, age_1-2, ht_1-2,	win_ratio_1-2_last_20, win_ratio_1-2_last_10, bps_1-2_last_20, bps_1-2_last_10, bpsr_1-2_last_20, bpsr_1-2_last_10, bpc_1-2_last_20, bpc_1-2_last_10, bpcr_1-2_last_20, bpcr_1-2_last_10, ace_1-2_last_20, ace_1-2_last_10, df_1-2_last_20, df_1-2_last_10, 1stWon_1-2_last_20, 1stWon_1-2_last_10, 2ndWon_1-2_last_20, 2ndWon_1-2_last_10, 1-2_sets_last_20, 1-2_sets_last_10,	Elo_sf_1-2, Elo1_1-2_sf, Elo2_1-2_sf, Elo3_1-2_sf, Elo4_1-2_sf, win_pred_sf_1-2, win_pred1_1-2_sf, win_pred2_1-2_sf, win_pred3_1-2_sf, win_pred4_1-2_sf, Elo_1-2, Elo1_1-2, Elo2_1-2, Elo3_1-2, Elo4_1-2, win_pred_1-2, win_pred1_1-2, win_pred2_1-2, win_pred3_1-2, win_pred4_1-2,	bpcr_1-2_8m, bpcr_1-2_6m, bpcr_1-2_4m, bpsr_1-2_8m, bpsr_1-2_6m, bpsr_1-2_4m, df_1-2_8m, df_1-2_6m, df_1-2_4m, ace_1-2_8m, ace_1-2_6m, ace_1-2_4m, 2ndWon_1-2_8m, 2ndWon_1-2_6m, 2ndWon_1-2_4m, 1stWon_1-2_8m, 1stWon_1-2_6m, 1stWon_1-2_4m, win_1-2_8m, win_1-2_6m, win_1-2_4m,
pair_names: 'Player_1_name / Player_2_name' 1-2_sets_last_N: total set points difference over the past N games when player 1 and 2 were competing with each others. win_1-2_Nm: difference in the number of games wonned by player 1 and player 2 during their past N months of matches (including matches when they are not competing with each others). Elo_1-2 & win_pred_1-2 : Elo rating and winning probability difference calculated on all matches, with s = 0.4. Elo_1-2_sf: Elo rating difference calculated based on different surfaces, with s = 0.4. Elo1_1-2, Elo2_1-2, Elo3_1-2, Elo4_1-2: s = 0.32, 0.36, 0.44, 0.48			

Tabel II: Column features that will be used for model training

Apart from the total Elo rating, we also calculate players' ratings for different court surfaces separately.

While Elo rating accounts for the long term historic performance, it might not be able to capture the recent performance change of the player sufficiently. For example, if a player with a high Elo rating hasn't been competing for a long time, his/her Elo rating will not change, but it will be too optimistic to assume that his/her performance will remain unchanged. To compensate that, we also calculated each player's recent performance by summing their technical data over the past 4 months, 6 months and 8 months.

Since the winning/losing depends on the relative performance of the two players, we focus on the difference of their information. Table II summarize the features that can be used for model training:

- **Personal & Tourney information** account for the initially given personal & tourney information in the raw data. Remember we showed in Figure 3 that the player's seed has a strong implication of his potential of winning. To quantify that, we convert the seed information from string ('1', '2', … '8', etc.) to numeric. For those who are not seed players, their seed_code = draw_size.

- **Past combat records** are features generated based on the historic compact records of player 1 and player 2.
- **Elo ratings** includes Elo ratings and winning probability predictions based on different values of parameter s, as well as ratings that are calculated based on different surfaces.
- **Recent competition performance** are calculated based on the sums of figures of technical performance over the past 4, 6 ,8 months. For example, to calculate bpcr_1-2_8m, we first sum up the bpcr of the player over the past 8 months' competition and then calculate the difference.

We can have a look how these features correlate with the winning of player 1. We find that indeed Elo ratings and their predictions show the highest correlation with the winning, followed by the performance in the recent months, and the compact record of the two players

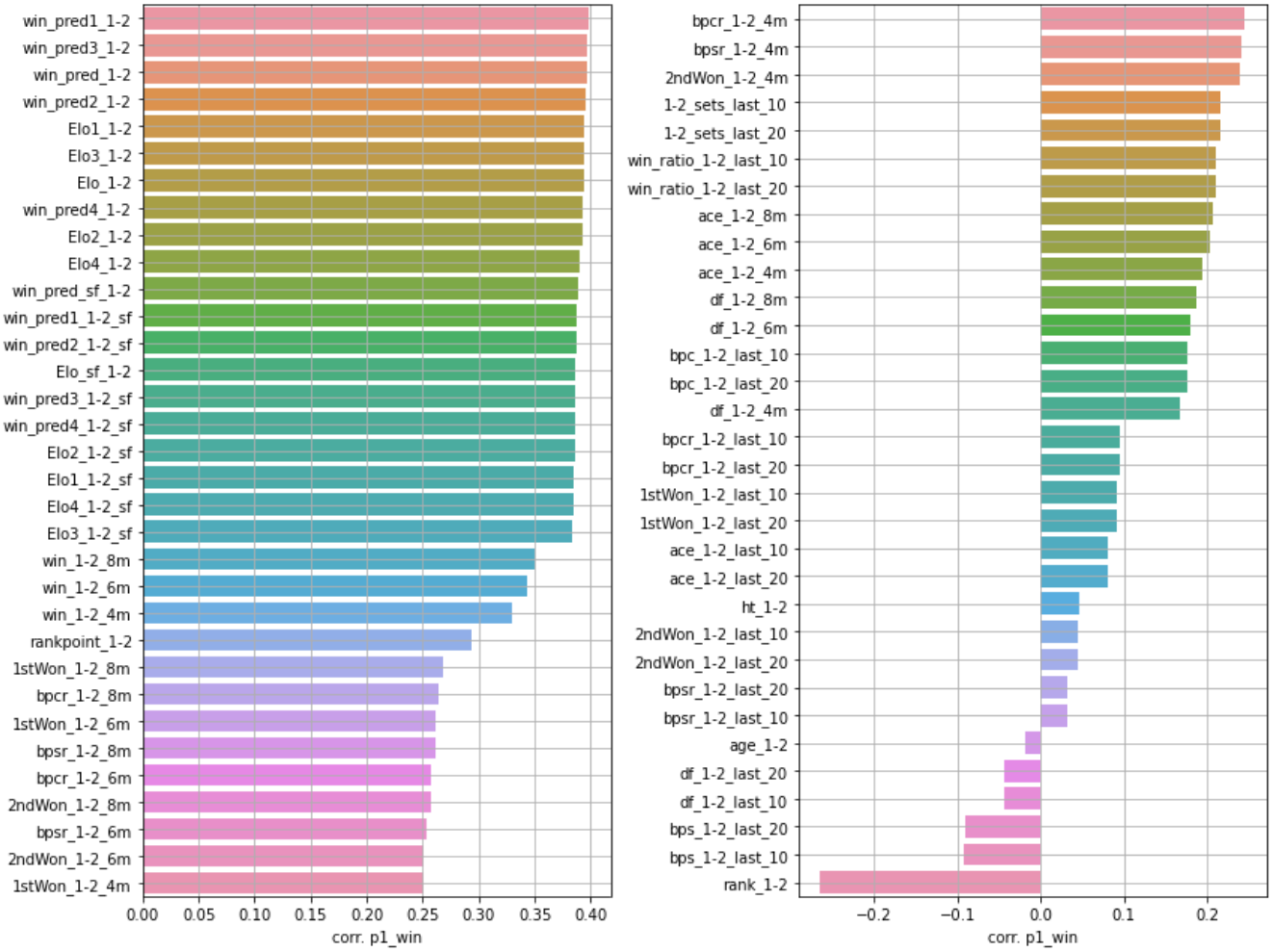


Figure 12. Pearson correlations of player 1 winning with other engineered features.

5. Machine Learning Model

5.1 Baseline models

Before we go for complicated models, we start with four different baseline models:

1. Elo rating: predict player 1 wins if ***win_pred_1-2 > 0***
2. Elo_sf rating: predict player 1 wins if ***win_pred_sf_1-2 > 0***
3. Ranking: predict player 1 wins if ***rank_1-2 < 0***
4. Recent winning: predict player 1 wins if ***win_1-2_8m > 0***
5. Winning ratio over the last 10 matches: predict player 1 wins if ***win_ratio_1-2_last_10 > 0***

We calculate the prediction accuracies of these four baseline models using data from 2010 to 2019.

The accuracy of winner prediction based on Elo is: 67.65%.
The accuracy of winner prediction based on Elo_sf is: 67.36%.
The accuracy of winner prediction based on ranking is: 65.66%.
The accuracy of winner prediction based on win_8m is: 65.30%.
The accuracy of winner prediction based on win_ratio_last_10 is: 53.40%.

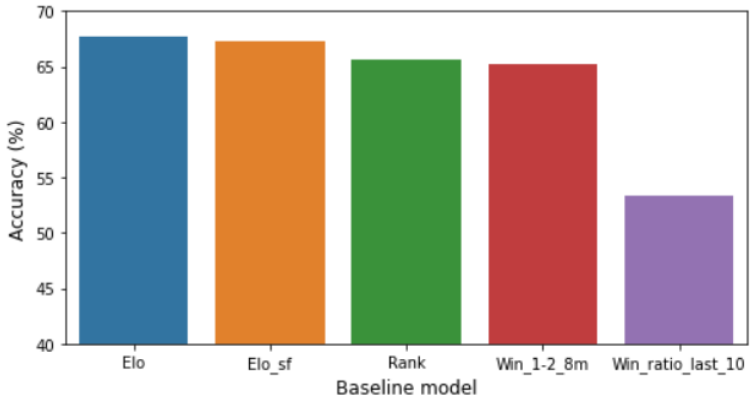


Figure 13. Accuracy of different baseline models with data from years 2010 to 2019.

5.2 Gradient-boosting tree model (Catboost)

Gradient-boosting tree models have been widely applied for supervised learning problems including regression and classification. Catboost is one of the popular open-source library, with an advantage of handling categorical features directly, which is highly valuable in many business problems.

We first split the data into three sections based on time: train set (1985 – 2000), validation set (2001 – 2009), test set (2010 – 2019). Here, we employ the CatBoostClassifier to train the train set, and apply early stopping using the validation set, finally we use the model to predict the test set and calculate the accuracy.

We apply GridSearchCV to find the optimal hyper parameters based on the accuracy on the validation set. The parameters found to be closed to optimal are given below:

optimal parameters = {'grow_policy':['SymmetricTree'],
'depth': [9],
'learning_rate' : [0.05],
'l2_leaf_reg': [5],
'random_strength':[0],
}

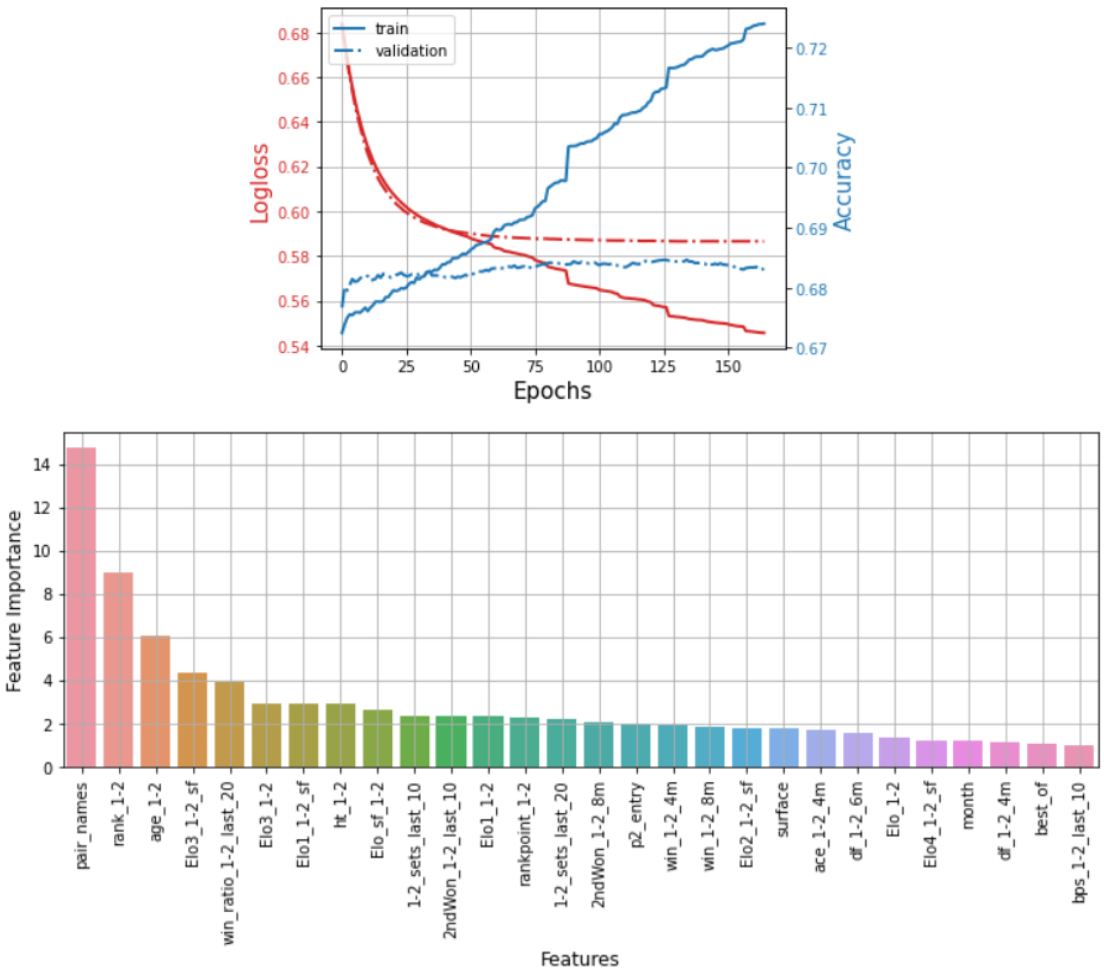


Figure 14. (Top) Training process. (buttom) Feature importance vs Features; for clarity, only features with feature importance > 1 are shown.

Figure 14 (top) shows the training process of the gradient boosting tree model using the validation set for early stopping, which gives an accuracy of 68.47% for the validation set.

The feature importance of different features is shown in Figure 14 (bottom). Clearly, the trained model captures the pair_names information and found it to be one of the most important feature. Quite surprisingly, instead of Elo rating, rank_1-2 has the more significance in the trained model. However, the feature importance of the features can have different result we the model The prediction accuracy on the test set using the best model is around 68.64%, increased by around 1% compared to the baseline model using purely Elo rating (67.65%).

Our data set has 78 numerical features. Data set with very high dimensionality not only takes up larger space and computational resources, but can lead to various problems, including data sparsity, multicollinearity, etc. Although, given the number of samples, the dimension of our data set should not lead to the so-called ‘curse of dimensionality’[4], it would be a good practice to check if there is any room for dimensionality reduction in our dataset.

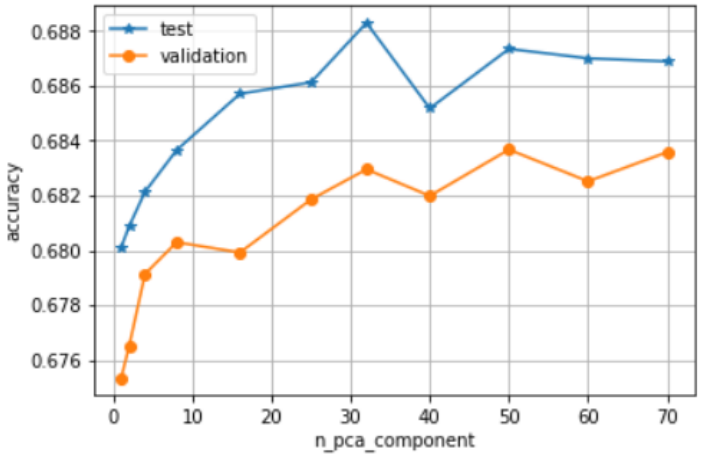


Figure 14. Model accuracy vs number of PCA components.

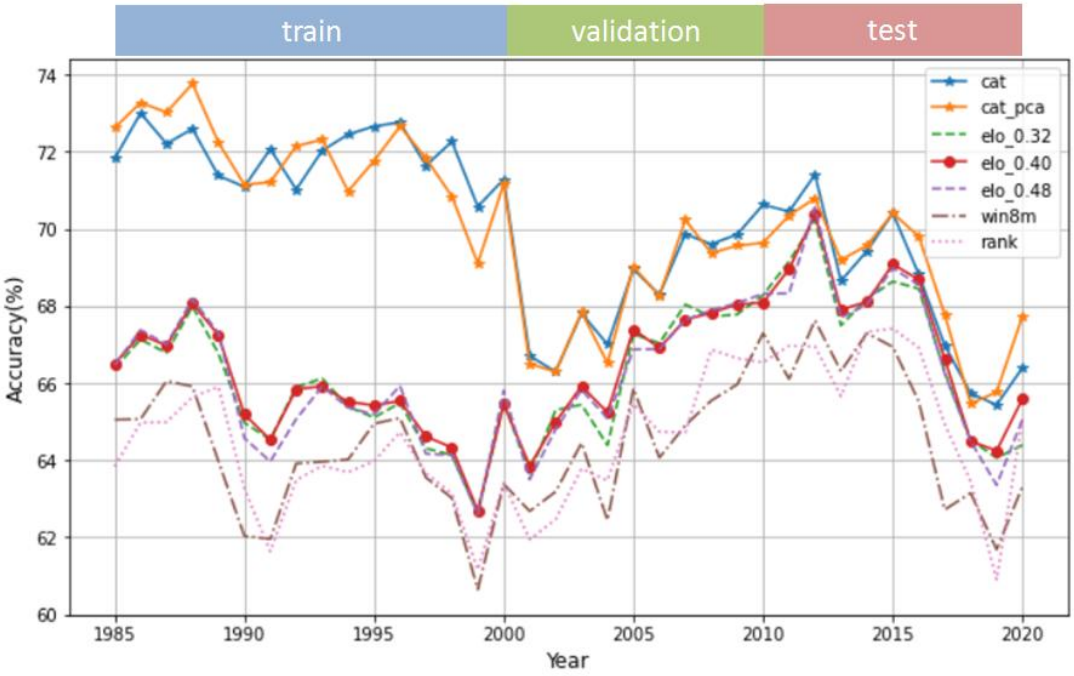


Figure 15. Model accuracy over years.

To do so, we performed principle component analysis (PCA) of the numerical features, while keeping the categorical features unchanged, and see how the prediction accuracy changes as the number of PCA components (n_pca_component) increases. The results are summarized in Figure 14. We found that the accuracy of the validation and test set is already pretty high even with a single PCA component, and the improvement becomes marginal when n_pca_component >32. Finally, we choose the model with n_pca_component = 32 to predict the whole data set.

We plot the accuracies of different models over the years, as shown in Figure 15. It's quite clear that the accuracies of the catboost models with and without PCA are comparable. Both models show much higher accuracy (~ 6%) in the training set (1985 – 2000), compared to the best baseline model based on Elo rating, but this is largely due to overfitting. The Elo models with different parameter s provide almost the same accuracy, while the baseline models based on win_8m and rank show similar accuracy.

A more proper estimation of the model accuracy is to use the test set (2011 – 2019). A comparison of the averaged accuracies of different models in the test sets are summarized in Figure 16. On average an improvement of around 1% over the Elo rating baseline model is achieved using Catboost and PCA+Catboost.

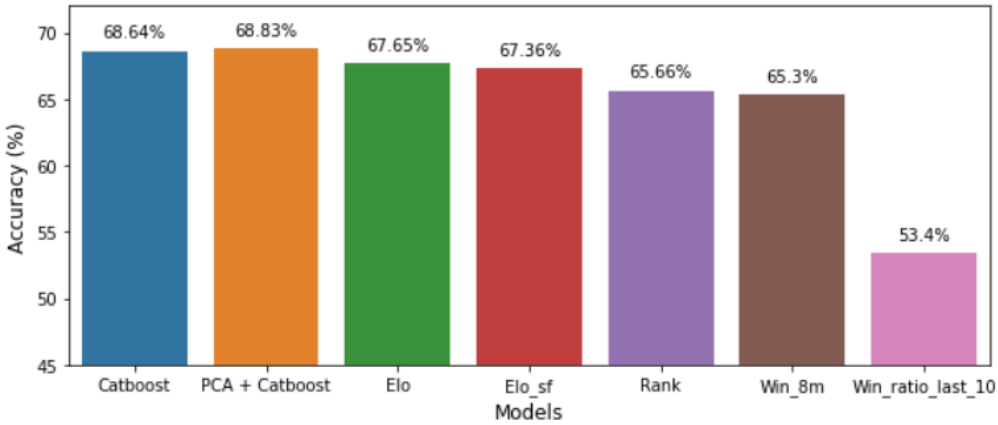


Figure 16. Accuracy of different models.

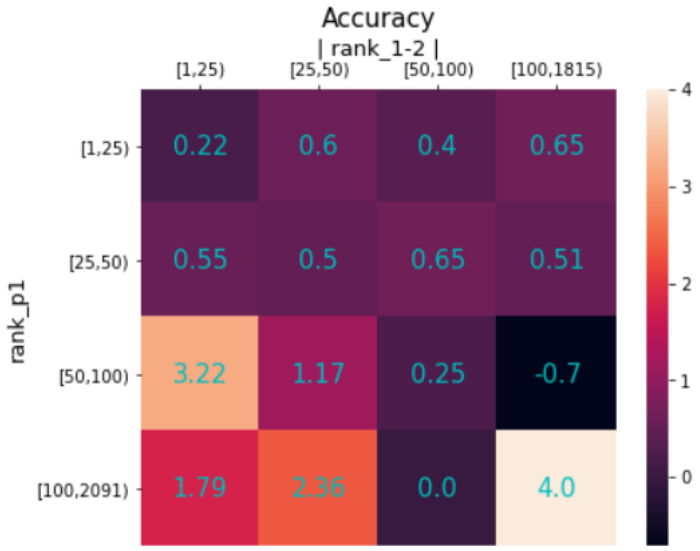


Figure 17. Accuracy difference of model PCA+Catboost over Elo rating for different rankings of player_1 and ranking difference |rank_1-2|.

We can further examine what level of improvement has been achieved in different groups of player ranking and player ranking difference. In Figure 17, we show a heat map of the accuracy difference of the model 'PCA+Catboost' and Elo rating. It shows that if both players are high ranking (rank_p1<25 & |rank_1-2|<25) , the improvement of the PCA+Catboost model over Elo_rating is not very significant. But the improvement is more noticeable (> 1%) for players with (rank_p1>50 & |rank_1-2|<50).

5.3 Neural Network model

The second type of model we built is neural networks. We choose to use a simple architect with two hidden layers, each connected to a dropout layer, as shown in Figure 19.

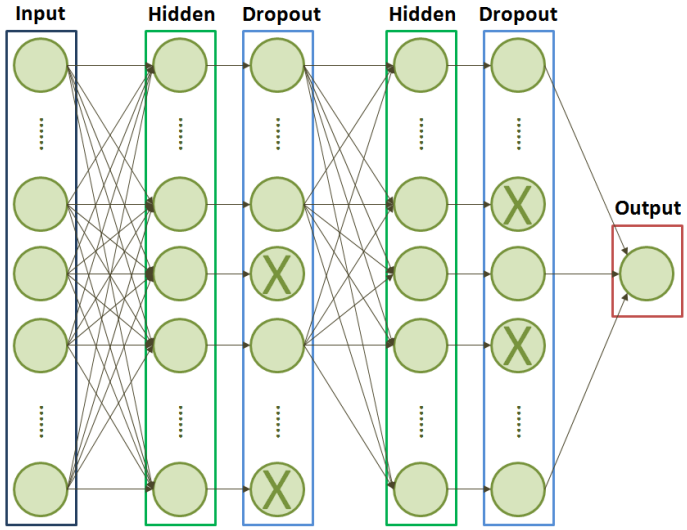


Figure 18. Schematic of the employed neural network architecture.

The features of the input data are the same as the ones used for Catboost, and principle component analysis is applied before feed in to the network. The number of principle component, the number of neurons in each hidden layer and other hyper-parameters, including the dropout coefficient, are optimized via GridSearchCV. The final optimal parameters chosen are

optimal parameters: { n_pca_component = 25
ly_hidden_1 = [25]
ly_hidden_2 = [6]
ly_final = [1]
batch_size = [50]
epochs = [200]
learning_rate = [0.00075]
drop_out = [0.1]
}

The prediction accuracy of the model on the test data set (2011-2019) is 68.20%. Figure 19 summarize the prediction accuracy improvement of different models over Elo rating over the years. We can see that model based on 'PCA + Catboost' provides the highest overall accuracy.

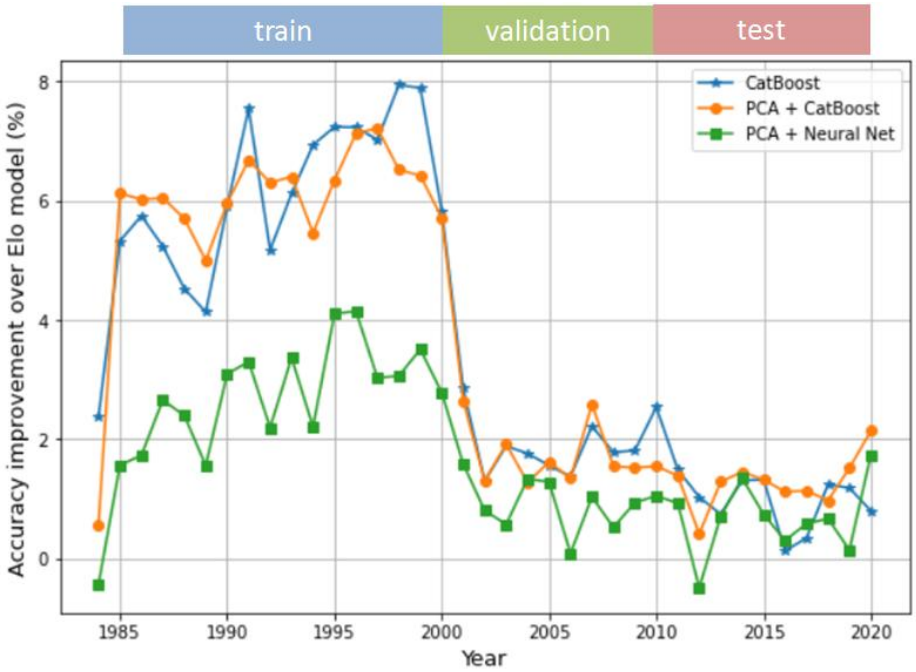


Figure 19. Prediction accuracy improvement of different models over Elo rating.

6. The Bidding Strategy

To be continued

7. Summary

- We collected, cleaned and analyzed the historic tennis competition data from ATP.
- We built a data analysis pipeline to show the statistics and the evolution of players' performance, their major opponents and their winning ratio over the historical competition.
- We found that Elo rating, player ranking, their recent winning and their record with the opponents are important baseline models for more complicated machine learning models to build on.
- Including technical data and other information of the players and the tourney can further improve the prediction accuracy.
- The best model developed is based on principle component analysis + Catboost model, which provide an accuracy improvement of around 1.2% compared to Elo rating.

8. Acknowledgements

Mingkai Liu thank Steve Siu for the valuable discussion on tennis data and Elo rating.

9. References

[1] Sports betting in Australia, Andrew Armstrong and Megan Carroll
https://aifs.gov.au/agrc/publications/sports-betting-australia#footnote-004-backlink

[2] Tennis: Elo Modelling https://www.betfair.com.au/hub/tennis-elo-modelling/

[3] Elo rating system https://en.wikipedia.org/wiki/Elo_rating_system

[4] Curse of dimensionality https://en.wikipedia.org/wiki/Curse_of_dimensionality