Live Tennis Odds Prediction
CSE 519 - Data Science Fundamentals
Project Report

Submitted by

Charuta Pethe	**111424850**
Deven Shah	**111482331**
Neha Mane	**111491083**

December 6, 2017

1 Introduction

Tennis is a favorite sport for betting around the world. The fundamental reason behind this is that at any point of time, there is a tennis match being played somewhere in the world. There are a large number of variables that can be considered in a tennis match, making it an interesting sport to bet on. To maximize the profits, we need a model which predicts the outcome of each match as accurately as possible.

The objective of this project is to build a model to predict the probabilities of the outcomes of a tennis match. This is to be achieved using previously available data about each player, as well as changes in the game state during game play. The subsequent sections describe this process, our work, and the results obtained using our prediction models.

2 Data

2.1 Description

This dataset contains point-by-point data for Grand Slam matches since 2011 in 4 major slam tournaments - Wimbledon, US Open, Australian Open and French Open. It consists of 56 files of two types - 'matches' files consisting of records of matches, and 'points' files consisting of point-by-point records of each match. The point-by-point data files are composed of records for each point in each match. (However, the records did not include the current number of sets won by each player, or the winner of the match.)

2.2 Preprocessing

We performed the following preprocessing on the dataset:

1. We computed the current score at each point in the match and added it to the points dataset.
2. We computed the winner based on the score in the last point, and added it to the matches dataset.

3. We removed the matches in which a player retired, as these matches were outliers for our model.

4. In order to get all the player rankings, we scraped the player rankings data from the website *www.atpworldtour.com*, which is the official ATP Ranking website [2].

5. We mapped the ranks scraped from the website to the dataset, and dropped those records in which either of the player's rank was missing.

6. For each match, we calculated the difference between the ranks of the two players, and stored it for initial probability calculation.

7. We created an additional field to store whether the higher ranked player won the match (1 = True, 0 = False) for the baseline model.

8. Further, for each set of matches with the same difference between the ranks of the two players, we calculated the probability of the higher ranked player winning, and stored it separately.

9. We converted and split categorical variables into binary variables for each player, as explained in Section 5.

3 Overview

3.1 Model structure

Our probability prediction model follows a sequence of processing as shown in Figure 1.

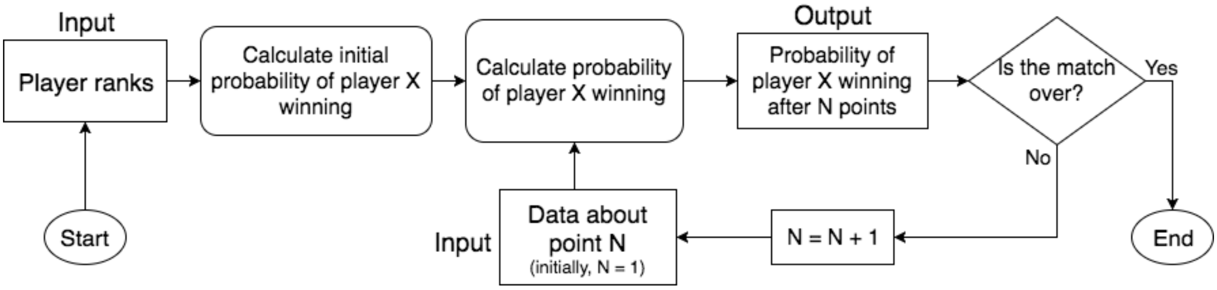
Input
Player ranks
Start
Calculate initial probability of player X winning
Calculate probability of player X winning
Output
Probability of player X winning after N points
Is the match over?
Yes
No
End
N = N + 1
Input
Data about point N
(initially, N = 1)

Figure 1: Data flow

We predict the probabilities of either of the players winning throughout the match. First, we calculate the initial probability of the players winning based on their ranks. As the match progresses, we take the initial probability and the current state of the match to predict the winning probability at each point in the match. Finally, the model outputs a series of values, which are the probabilities of the player winning at each point in the match.

3.2 Training and Testing Data

In order to train the model, we have taken the matches from the four Grand slams played during the years 2011-2016.

We have tested our model on all the matches played in the four Grand slams in 2017.

3.3 Evaluation Environment

We have developed an environment to evaluate our prediction model. The environment works for a single player as follows:

1. Get all the matches played by a player.
2. For each match, find the probability of the player winning at each point in the match, using the prediction model.
3. Create a list of buckets of numbers from 0 to 1 at intervals of 0.0125.
4. For each value in the list, check if any probability in the bucket has been predicted in the match. If yes, increment the won/lost count accordingly, based on whether the player has won or lost that match. Repeat for all matches.
5. For each value in the list of buckets, calculate the ratio of number of matches in which this value was observed and the player won, to the total number of matches in which this value was observed.
6. Plot a graph of the observed probability of a player winning a match vs the predicted probability of a player winning a match.

The ideal graph in this case would be a straight line from (0, 0) to (1, 1). In order to obtain a more general and concrete numeric estimate of model performance, we have also calculated the mean of the mean squared error over all players, i.e. deviation from the ideal line.

4 Baseline Model

4.1 Initial Probability Calculation - Partitioned Linear Regression

The initial probability prediction depends on the difference in the ranks of the two players. Due to the wide variety of differences in ranks, the regression varies over different ranges. This is why we have divided the difference in ranks in 3 groups.

The 3 groups that we have chosen are as follows: The first group consists of differences from 0-30 (Figure 2), in which a rapid change in the probability is observed in a small range. The second group consists of matches with differences between 31-75 (Figure 3), which shows a gradual increase in the probability. The last group consists of differences between 76-1000 (Figure 4) which shows an almost negligible increase in the probability. Based on the difference in ranks we have then applied linear regression to each group, and the cumulative result is shown in Figure 5.

We also tried polynomial regression for initial probability. Due to the scattered nature of the probabilities for higher differences in ranks i.e. matches with difference greater than

75 up to 1000, we did not get the desired result and hence the partitioned linear regression works better in this case.

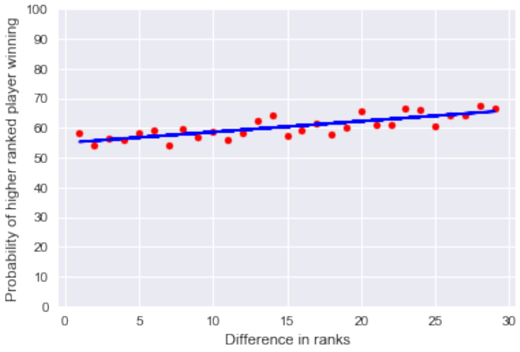

Figure 2: Set 0-30

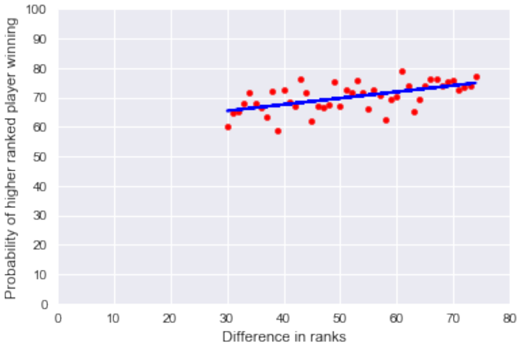

Figure 3: Set 31-75

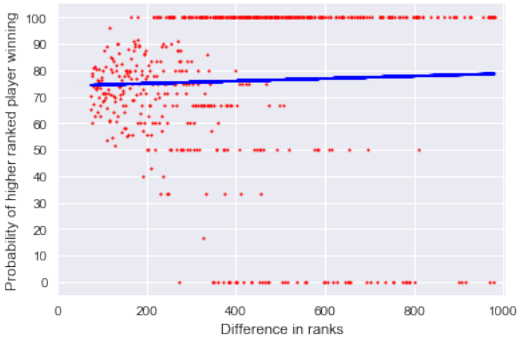

Figure 4: Set 76-1000

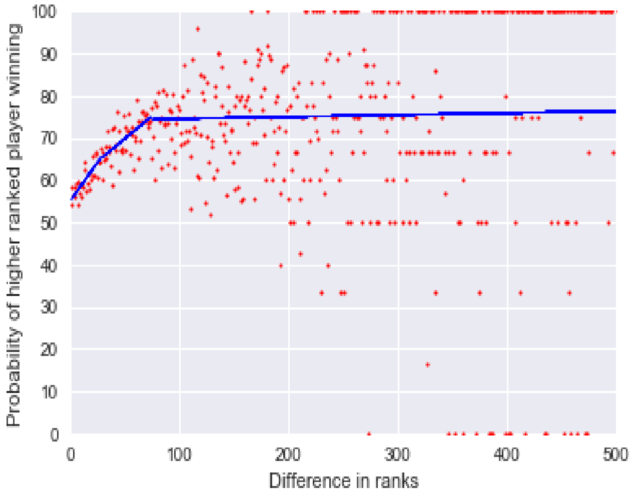

Figure 5: Linear Regression for all Rank Differences

4.2 Calculation of probability for each point in the match

In our baseline model:

- The higher ranked player's probability of winning increases linearly from the initial probability to 1 irrespective of who wins the match, as shown in Figure 6.
- The lower ranked player's probability of winning decreases linearly from the initial probability to 0 irrespective of who wins the match, as shown in Figure 7.

According to this model, the higher ranked player is never expected to lose, and the lower ranked player is never expected to win.

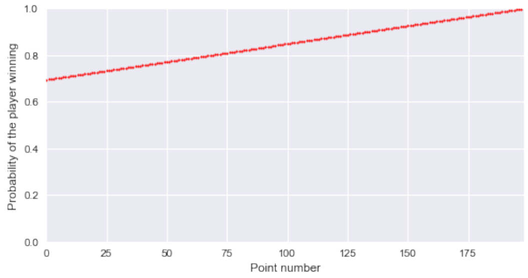

Figure 6: Probability of the higher ranked player winning over the course of a match

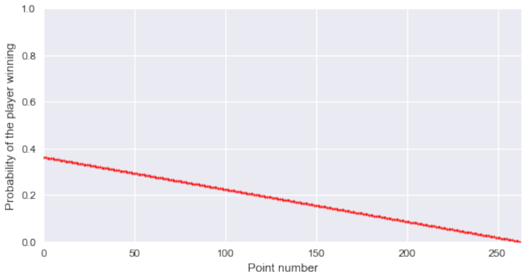

Figure 7: Probability of the lower ranked player winning over the course of a match

4.3 Performance of Baseline Model

We evaluated the baseline model for all matches of various players throughout the four Grand slams. The following figures show the predicted probability vs the observed probability of a player winning the match over the four Grand Slam tournaments for the years 2011-2017.

As seen from Figure 8 and Figure 9, the graph starts with the initial probability and constantly increases up to 1.0, as the players won almost all the matches where they were predicted to win.

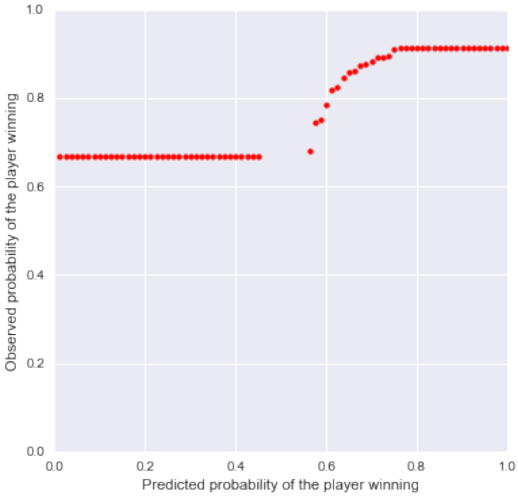

Figure 8: Randomly selected player1 - 151 matches

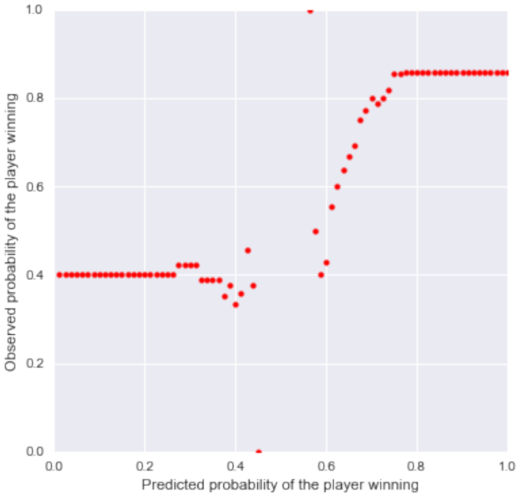

Figure 9: Randomly selected player2- 76 matches

On evaluating the performance of the model for each player, and calculating the mean of the RMS errors, the baseline model gives a mean RMS error of 0.1026. This means that the mean deviation of the data points from the ideal line is 0.1026.

5 Improved Model

5.1 Data Preprocessing

1. Besides the preprocessing done before developing the baseline model, we have converted the categorical column match_winner into a new column p1_match_winner, which is 1 if player 1 has won the match, and 0 if player 1 has lost the match.

2. We calculated the cumulative sets won by both players over the course of the match and stored them in two new columns p1_sets_won and p2_sets_won.

5.2 Feature Selection

1. The features we have used to build our prediction model are: number of sets, games and points won so far in the match by Player 1 and Player 2 respectively, the ranks of the two players, the initial probability of Player 1 winning the match, and a variable indicating whether Player 1 has won the match (p1_match_winner).

2. The idea behind choosing these features is that the points, games and sets depict the entire state of the game at any given point.

3. The ranks of the players help us calculate the initial probability using our baseline model, as mentioned in Section 4.1

4. The dependent variable here is the p1_match_winner which is the basis of our prediction

5.3 Model

As we have used p1_match_winner to train our model, in our model we are predicting the probability of player 1 winning in any match and as probabilities are complimentary we also get the probability of player 2 winning.

We have performed logistic regression on the above mentioned features to predict the winning probability of both the players.

We have used logistic regression as it gives us the probability of winning based on the values of the independent variables(features).

This model gave a score of 0.81, which is an improvement from our baseline model that gave a score of 0.66.

5.4 Results

5.4.1 Output

The improved model gives the output for various cases as follows:

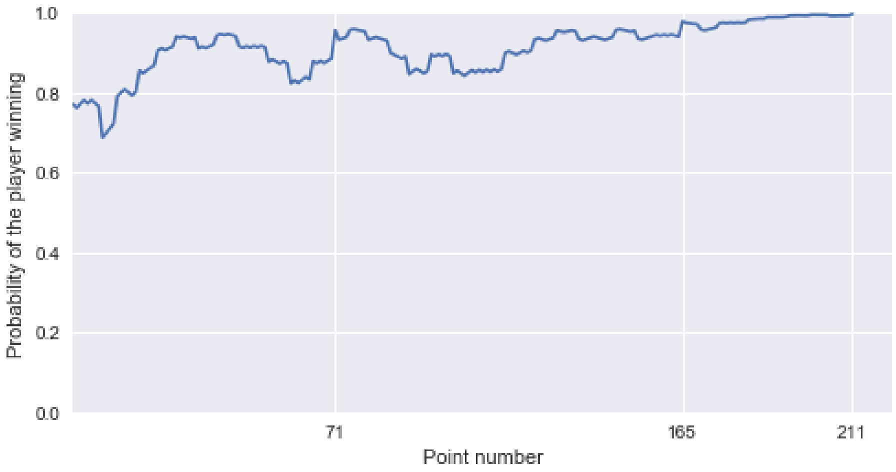

Figure 10: Probability output for the higher ranked player winning the match (2017-ausopen-1101)

1. **Higher ranked player winning the match:** In this match, Rank 1 played against Rank 93. As shown in Figure 10, the initial probability of the higher ranked player winning was 0.77. As he won the match in three straight sets, his probability of winning kept increasing throughout the match, with a few variations caused due to the opponent winning a few points in between.

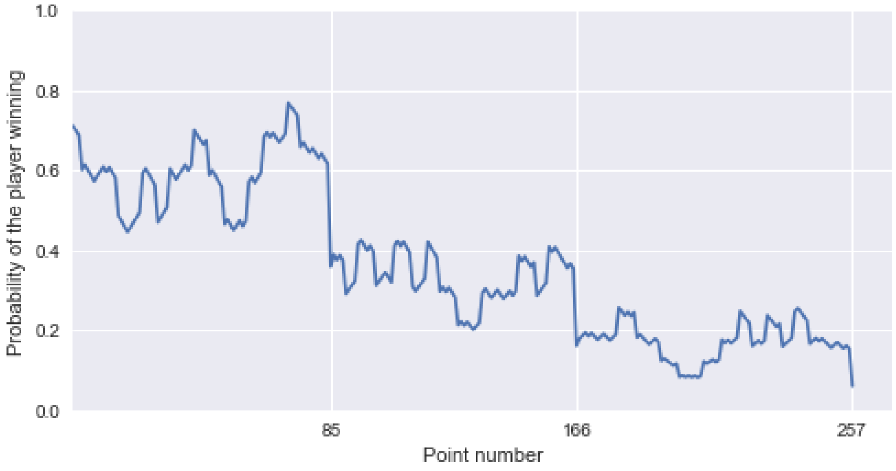

Figure 11: Probability output for the higher ranked player losing the match (2017-frenchopen-1223)

2. **Higher ranked player losing the match:** In this match, Rank 71 played against Rank 35, and the higher ranked player lost the match. As shown in Figure 11, the initial probability of the higher ranked player winning was 0.71. The first set (up to point number 85) was a tie-breaker, and was won by the lower ranked player, which

caused a severe drop in the graph. Subsequently, the higher ranked player went on to lose two more sets and hence lost the match, resulting in further lowering of the probability of winning throughout.

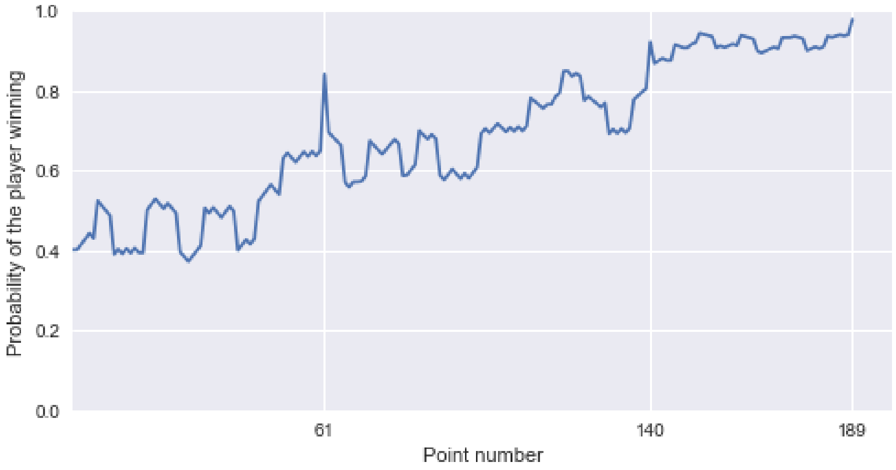

Figure 12: Probability output for the lower ranked player winning the match (2017-usopen-1152)

3. **Lower ranked player winning the match:** In this match, Rank 45 played against Rank 36, and the lower ranked player won the match. As shown in Figure 12, the initial probability of the lower ranked player winning was 0.4. The first set was won by the lower ranked player at the 61st point, up to which both the players were winning the games alternatively and hence the graph shows variations. The lower ranked player went on to win the match by winning the next two sets as well. The upward nature of the graph captures this well.

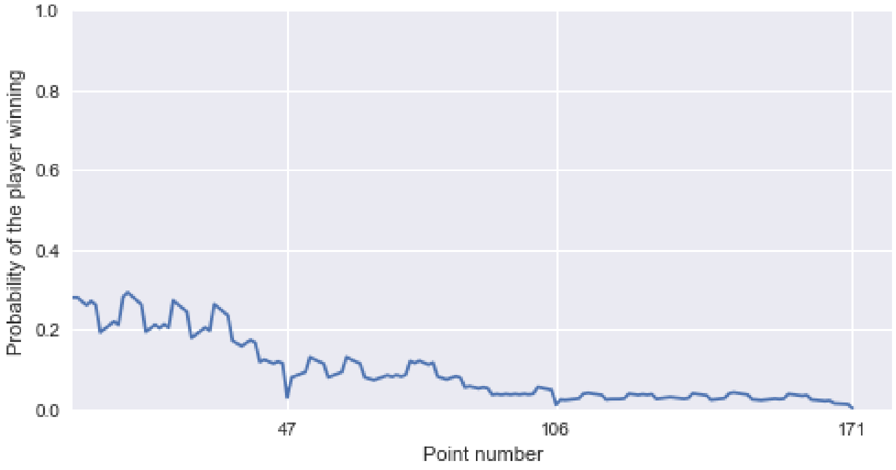

Figure 13: Probability output for the lower ranked player losing the match (2017-usopen-1220)

4. **Lower ranked player losing the match:** In this match, Rank 49 played against Rank 14, and lost the match. As shown in Figure 13, the initial probability of the lower ranked player winning was 0.4. As the opponent won the match in three

straight sets, the probability of the lower ranked player winning kept decreasing throughout the match, with a few variations caused due to winning a few points in between.

5.4.2 Performance of Improved Model

We evaluated the improved model for all the matches played by various players across the four Grand slams for the years 2011-2017.

Figures 14 and 15 show the predicted versus observed probability of the player winning across all matches.

On evaluating the performance of the model for all matches for each player, and calculating the mean of the mean squared errors, our improved model gives a mean mean-squared error of 0.0124.

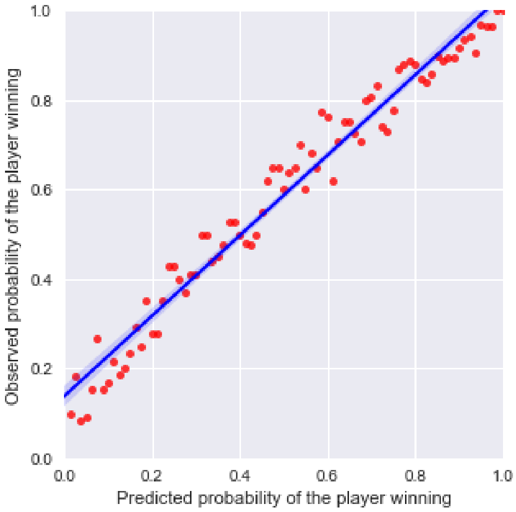

Figure 14: Random Player 1

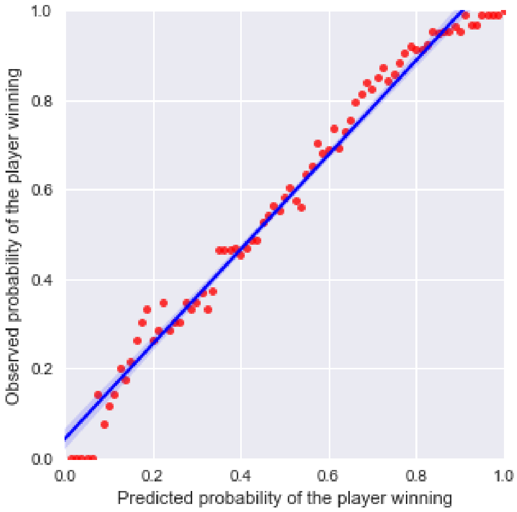

Figure 15: Random Player 2

5.4.3 Comparison with Baseline Model

The following figures show the predicted vs the observed probability for our baseline and improved models.

As seen from the figures 16, 17 and 18, 19, our improved model is able to predict the probability of winning much better than the baseline model as it now takes the ranks as initial probability and the game state at every point in the match as opposed to just the difference in the ranks used in baseline model. For the improved model, the mean deviation of the data points from the ideal line is 0.0124, which is very low and is an extreme improvement over our baseline model that gave a mean deviation of 0.1026.

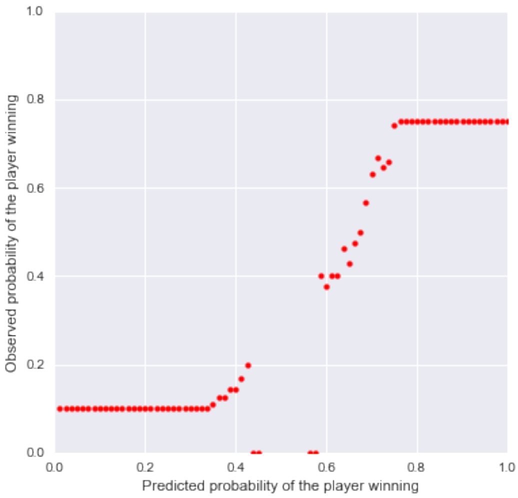

Figure 16: Random Player1 - Baseline Model

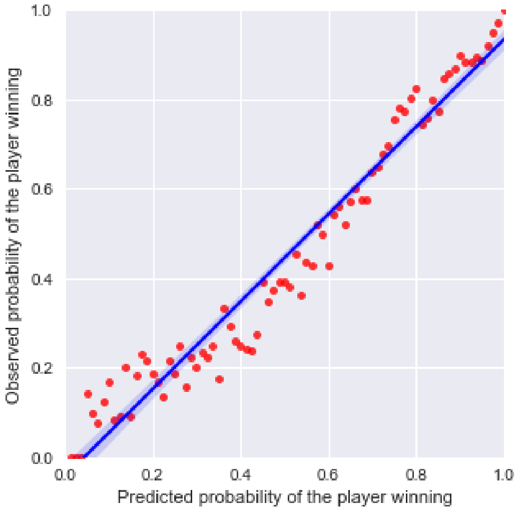

Figure 17: Random Player1 - Improved Model

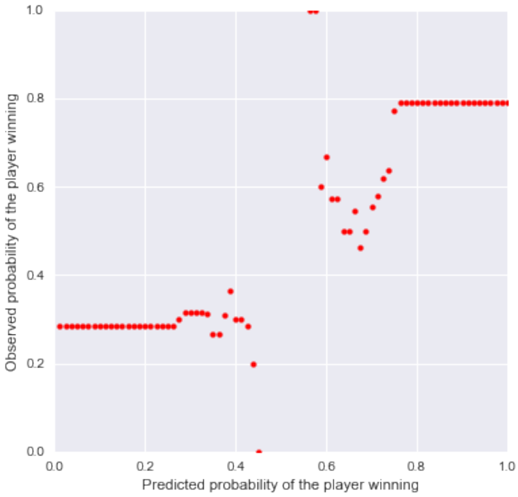

Figure 18: Random Player2 - Baseline Model

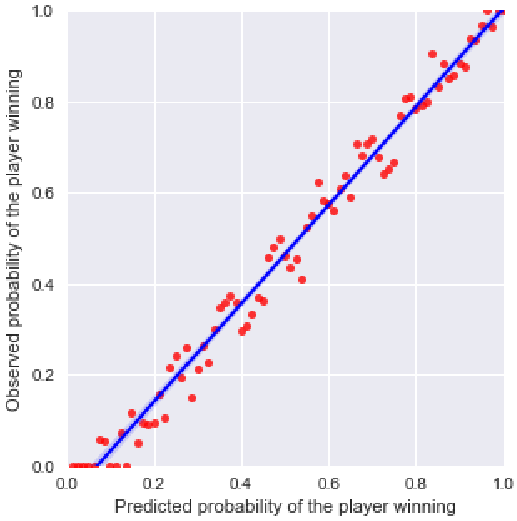

Figure 19: Random Player2 - Improved Model

6 Conclusion

In this report, we have described our model which predicts the probability of a player winning a tennis match, throughout the match, with a score of 0.81 (the closer the score is to 1, the better is the model). We were able to improve the predictions by a wide margin, from our baseline model to our final model.

A major challenge that we faced while implementing our model was with our data. Although there was a lot of data available about tennis matches, it was scattered and hence required a lot of scraping from various websites. Furthermore, the scrapped data wasn't in the format we needed so that required a lot of preprocessing before the model was implemented.

Finally, our model was able to predict the outcomes of a match based on the initial probability(ranks of the players) and taking into account the game state at every point(points, games and sets status).

References

[1] Grand Slam Point-by-Point Data, 2011-17, *https://github.com/JeffSackmann/tennis_slam_pointbypoint*.

[2] ATP World Tour Rankings, *http://www.atpworldtour.com/en/rankings/singles*.