Analyzing the Importance of the Serve in Professional Tennis

Josh Marvald Dr. Matt Higham

10 July 2020

Contents

Abstract

This project is an attempt to provide an analysis of the serve in professional tennis and its association with win probability. We completed extensive data exploration and wrangling to both find important trends and to shape the data into usable forms to build models. The final model is a Bradley-Terry model that focuses on how first serve percentage (percent of times the first serve is made) is associated with win probability for different players. After completing the model building and testing, we built a `Shiny` app that allows users to plot the fitted Bradley-Terry model lines for players of interest and specific opponents. The app allows users to examine how the association between the probability of winning a match and first serve percentage changes for different players and first serve percentages.

Introduction

Before describing the work done in this project it is important to first understand the motivation. As tennis players and people who watch tennis matches whenever they are televised, Dr. Higham and I noted that oftentimes, tennis commentators provide statistics and comments that are not very meaningful. They will mention that one player needs to make sure to maintain a high first and second serve percentage if they want to win the match. Comments like these are not insightful. They apply to all players and should be obvious to any person familiar with tennis. We believed that is was possible to generate more penetrating statistics that would give viewers a better understanding of what a particular player should focus on regarding his or her serve. In particular, we believed that player-specific statistics would be much more valuable than the

broad, obvious statements commentators often make. We also hoped that these statistics would shed light on the importance of the serve for different professional tennis players.

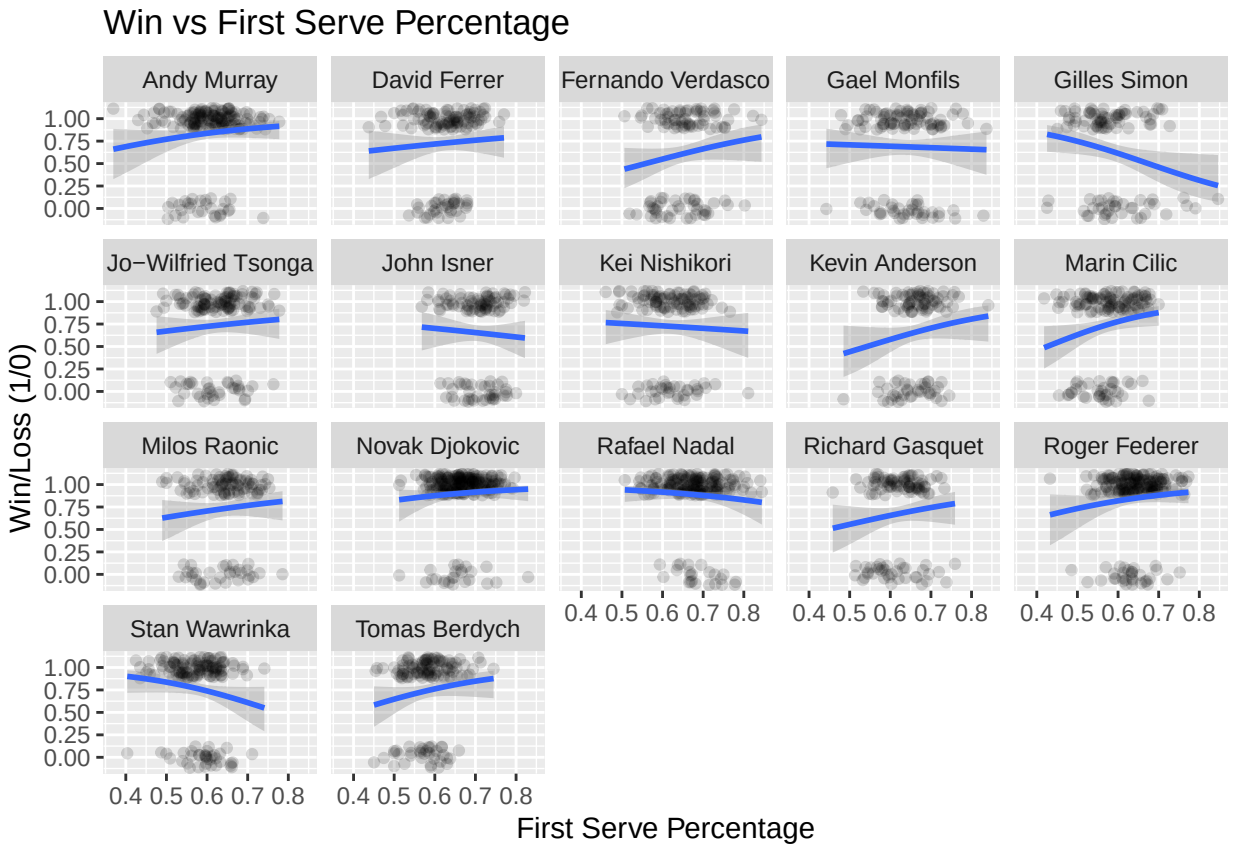
Description of Match Data

To begin the project, we started by exploring match data. In particular, we used match data from the past decade of Grand Slam matches for the ATP and WTA that was sourced from Jeff Sackmann's ATP and WTA tennis GitHub repositories (Sackmann, 2020) to investigate how serve variables were associated with win probability. The match data includes summary statistics from every Grand Slam match. Grand Slam tennis tournaments are the four biggest tennis tournaments every year with the most prize money and the highest influence on professional tennis players' rankings. The statistics most useful for this project from the match data set are first and second serve percentages and the ranking of each player in the match.

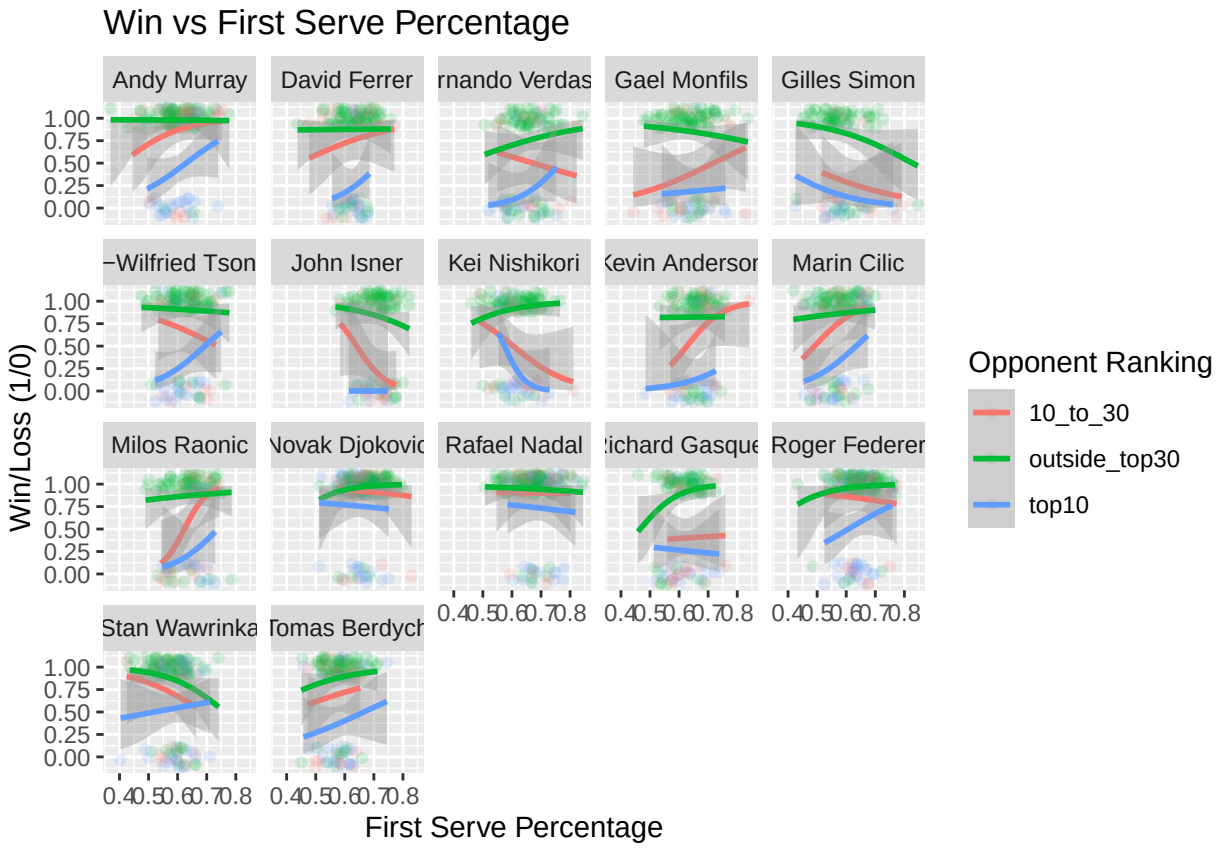
Data Exploration

Exploring this data consisted of looking for any trends in the data. It is important to note that this entire project was exploratory in nature, not confirmatory. Therefore, we do not use confidence intervals or p-values for any modeling done since these are invalid. Specifically, any trends that were related to the serve in some way were of particular interest. To find these trends, we had to tidy the data sets into more usable forms.

Once some wrangling had been done with the match data we were able to look for any relationships between first and second serve and the probability of a given player winning a match. Surprisingly, we found patterns that indicated that for some players, higher first serve percentages corresponded with lower chances of winning a match. This is shown in the graphs below that plot an indicator variable for whether the player won the match against their first serve percentage.



At this point, we realized that there were other factors in play that must be contributing to this trend. Given a single match, it doesn't make much sense that a higher first serve percentage would decrease a player's chance of winning. Our first theory was that it was possible that players were serving differently against lower and higher ranked players. Specifically, our theory was that they might serve faster and less accurately against lower ranked players and slower and more accurately against higher ranked players. This could account for having more matches won with lower first serve percentages and more matches lost with higher first serve percentages. To see if this was the case we made a categorical variable for the opponent's ranking.



As we can see from the plots above, this did not help simplify the matter. Different players have different trends and most players don't have the same trend for each category of opponent ranking. Seeing that looking at opponent ranking wasn't enough, we realized that a more opponent specific model would be needed. This is what led us to decide on a Bradley-Terry model.

Match Data Modeling

Bradley-Terry Model

Following the data exploration and manipulation process, we moved on to building models. First, we started by building a Bradley-Terry model for the match data. A Bradley-Terry model is useful for competition data, where "players" are paired and there is only one winner per competition. This type of model takes a given player and looks at the matches he or she played against all opponents and returns different lines for different player match ups.

Bradley-Terry models do this by returning an intercept and slope for each player for a given predictor. With these parameters and a given value for the predictor, a player's "ability" can be calculated for each player involved in a match. The two players' abilities are then back-transformed using a logit formula to return a predicted probability that one player will win the match given values of first serve percentages for each

player. The formula for this is provided below:

$$log(\frac{P_{ij}}{1-P_{ij}}) = (\alpha i + \beta_i * firstserve) - (\alpha j + \beta_j * firstserve)$$

Where P_{ij} is the probability that player i beats player j, αn is the intercept for each players' ability function, β_n is the slope for each players' ability function, and $firstserve$ is the predictor for first serve percentage. To actually calculate predicted probabilities a little bit of manipulation from the original Bradley-Terry model formula was necessary. If we replace $(\alpha_i + \beta_i * firstserve_i) - (\alpha_j + \beta_j * firstserve_j)$ with $(ab_i - ab_j)$ where each ab_n represents each player's ability, the following transformations allow us to calculate predicted match win probability:

$$log(\frac{P_{ij}}{1-P_{ij}}) = (ab_i - ab_j)$$

$$\frac{P_{ij}}{1-P_{ij}} = exp(ab_i - ab_j)$$

$$P_{ij} = 1 - P_{ij} * (exp(ab_i - ab_j))$$

$$P_{ij} = exp(ab_i - ab_j) - P_{ij} * (exp(ab_i - ab_j))$$

$$P_{ij} + P_{ij} * (exp(ab_i - ab_j)) = exp(ab_i - ab_j)$$

$$P_{ij} * (1 + exp(ab_i - ab_j)) = exp(ab_i - ab_j)$$

$$P_{ij} = \frac{exp(ab_i - ab_j)}{(1 + exp(ab_i - ab_j))}$$

This type of model allows for possibilities that less specific models would not reveal. Specifically, it is possible for a given player to have an overall negative relationship between their chance of winning a match and first serve percentage for all opponents while still having a positive relationship between chance of winning and first serve percent against individual opponents. While this may not be the case for all players, it is a possibility that may not be accounted for with other types of models.

Before building the model, we needed to decide how to split the data into training and testing data sets. We used a k-fold cross validation process, using matches in a single year as the 11 different folds. This process involves setting aside a single year of matches for our testing data set and using the other 10 years for our training data set. We then build the model with the training data set and test its predictive ability by comparing its predictions against the actual outcomes from the testing data set. This process is then repeated 10 more times using each year as a testing data set with the other years as the training data set. This results in 11 different iterations of a Bradley-Terry model with 11 different sets of predictions.

Seeing as our model returns a probability, it was difficult to directly test the model predictions. We know who won in each match, but we never knew the true match win probability for a player in a match. Therefore, we used a calibration test to check the predicted probabilities. The calibration test grouped the data into bins with size equal to 0.1 in predicted match win probability. For example, one of the bins included all entries with predicted match win probabilities of 0.5 - 0.6. For this example if our model predicts a group of players to have match win probabilities of 0.5 - 0.6, then we would expect that 50-60% of those players would win their matches. This is what we then checked for our calibration test for each iteration of the Bradley-Terry model. Upon completion we had 11 tables with proportions of matches won for each range of predicted match win probability.

As we suspected, the models that used years nearer the center of the year range for the testing data set seemed to perform better than those with testing data sets from years closer to 2010 or 2020. We also found that the Bradley-Terry model performed better for ATP matches than for WTA matches. We suspect that this could be due to several reasons. One reason is that WTA Grand Slam matches are best 2 out of 3 sets while ATP Grand Slam matches are best 3 out of 5 sets. With longer-format matches, it is likely that upsets

occur less frequently. If there are fewer upsets in ATP matches, then it makes sense that any model would be better at predicting ATP match winners than predicting WTA match winners. Another possible reason the Bradley-Terry model performs better for ATP matches is that the serve is more of a determining factor in men's tennis than women's tennis. It is possible that due to serves being faster in ATP matches they have a larger influence on match outcomes than in WTA matches. Overall though, our models seemed to perform reasonably well. Most of the proportions for matches won matched with the ranges for predicted match win probability. This indicated that our model was performing as we hoped. In our final model that is used for the `Shiny` app, we used all 11 years to build the model.

Plotting

Once we decided on focusing our app around the Bradley-Terry model, we were ready to move onto constructing plots for our Bradley-Terry models and building the `Shiny` app that would display these plots. First, we decided to construct a plot for a single player of interest. In order to do this, we needed to use our model to calculate individual match win probabilities for a range of predictor values. The following paragraphs explain how we calculated the win probabilities for Roger Federer for different opponents at different first serve percentages. However, the explanations apply for any particular player of interest. We started by looking at Roger Federer with first serve percentage as our predictor. Federer had a first serve percentage range of 0.52 to 0.77 in his 10 years of Grand Slam matches. We split this range 30 times to give us 30 first serve percentage values that we would use to calculate match win probabilities. Since Federer was the player of interest we averaged each opponent's first serve percentages to give us values to calculate their abilities.

We already had an intercept and slope from the Bradley-Terry model for each player so we used the 30 first serve percentage values to calculate Federer's ability at each first serve percentage value. To find his opponents' abilities, we used their average first serve percentage to calculate an average ability for each opponent. Finally, we calculated Federer's ability at each first serve percentage value and his opponents' average ability. Back-transforming the difference in abilities with the steps provided in the model building section, we obtained predicted match win probabilities for Federer at each first serve percentage value. This gave us paired value points with a first serve percentage and a predicted match win probability for each of Federer's opponents. Using these points, we could then create predicted match win probability lines for Federer against each opponent.

App Building

At this point, we had a fitted Bradley-Terry model and had established how we would use the model to make plots for each player of interest so we were ready to build the app. To do this we used the `Shiny`. `Shiny` is an `R` package that can be used to build applications viewed in a web browser. Shiny does this by using common `R` language and then converting the code to HTML behind the scenes. This makes it very easy to build smooth, fast applications that anyone with a browser can view.

`Shiny` code is split between a User Interface (UI) function and a Server function. Basically, the UI function defines what the app looks like and how the user can interact with it. The Server function can take the information provided by the user and feed it into any code within the server. The Server function also defines any plots, tables, or text that is first specified in the UI. In our case, the UI would need to allow the user to select either the ATP (men's tennis) or WTA (women's tennis), a player of interest, and any number of opponents. The server side of our app takes the info provided by the user and feeds it into the code to build plots for our Bradley-Terry model. Since we had already written code that built a plot for Federer, we were able to generalize that code to work for the player of interest provided by the user.

While much of the code for Federer was able to be reused, most of the work to build the app involved generalizing it to work for the user-provided player of interest. This proved to be the biggest challenge in making the app. The code needed to be responsive to any changes the user makes with selecting players or switching between ATP and WTA. We also found that labeling the lines plotted for individual opponents was difficult. A legend was helpful but was overwhelming if the user selected more than 5 or 6 opponents. In the

end, we decided to include 2 plots in our app. The first plot used individual lines for each opponent with a label pointing directly to the line. This was a helpful plot when looking at a small number of opponents. The second plot displayed lines for all of the player of interest's opponents at once without any labels. This plot was useful to show the user the overall trends against all opponents for the player of interest. In addition, the second plot was not reactive to changes in selected opponents while the first plot was reactive.

In the process of data manipulation and app building we learned several things about different professional tennis players and serving in general. One interesting and surprising result is that for certain players, their plotted match win probability lines trend downwards with first serve percentage. This was a result we saw before we made our model player-opponent specific that we expected the Bradley-Terry model to be able to account. We were surprised then, when these trends still appeared for some players after fitting the Bradley-Terry model. One theory we have is that for specific player-opponent match ups the player of interest is more successful when they are more aggressive on their serve. This could involve either serving faster or serving closer to the lines. If this were the case, then their first serve percentage would be lower but their win percentage against that opponent would be higher. This could potentially explain the downward trend that some players exhibit.

App link: https://jdmarv17.shinyapps.io/tennis_app/

Point Importance Introduction

Our secondary goal of the project was to investigate point importance (described later) and its influence on the serve. This, in combination with player specific statistics, would make for more in-depth, insightful observations that would give tennis fans a look into professional tennis players' strategies. Since tennis has a unique scoring system with different scoring units (points, games, sets) different points will influence the match outcome more than others. Therefore, we suspected that point importance could play a role in influencing players' serves.

The data set utilized for this portion of this project is point-by-point data from 2016 and 2017 Grand Slam matches that was sourced from Jeff Sackmann's point-by-point GitHub repository (Sackmann, 2020). This data has information on each point played in every Grand Slam from 2016 and 2017. Some of the variables include serve speed, serve placement, rally count, and the current score of the point.

The final data set used is a data set of point importance for each possible score in tennis that was sourced from Stephanie Kovalchik's `deuce` `R` package (Kovalchik, 2019). Importance values for a point give the average change in the probability of the winner of the point winning the match. Point importance would play an important role by being used as a predictor of different serve measurements in one of the models built for the project. During the data cleaning period, the importance data set was merged by score with the Grand Slam point-by-point data to assign importance values to each point played in Grand Slams.

Point Importance Data

Moving on to the point data, the exploration process was not as involved. The major relationship we were looking at was point importance versus serve speed and serve placement. The more time consuming process turned out to be manipulating the importance data and the point-by-point data to merge correctly. The importance data set was calculated using the assumption that a tiebreaker would be played at 6-6 in a fifth set; however, in some Grand Slam tournaments no tiebreaker is played for the fifth set. Instead of a tiebreaker, play continues until one player leads the other by two games. Since this was the case, upon merging the point-by-point with the importance data there were points in some matches that were not assigned importance values due to the fact that no fifth set tiebreaker was played. To account for this, we created a grouping variable that accounted for the difference in players' game scores in the fifth set once the score of 6-6 was reached. Once this was done, we grouped the merged data by point score and the grouping variable and filled in the missing importance values based on the groupings.

At this point it is important to understand how point importance is calculated. Peter O'Donoghue, who created the point importance measurement, started by looking at a large data set of points from ATP and WTA matches. O'Donoghue then took each occurrence of a particular point and calculated the average difference between the probability that the winner of the point wins the match and the probability that that same player wins the match if they lose the point. For example, if a point has an importance value of 0.05, then on average, the point winner's probability of winning the match is 5% higher than if they lost the point. Obviously, this isn't an exact difference in match win probability for matches that reach that particular point but the original data set was large enough that the change in match win probability is approximately 5% (O'Donoghue). O'Donoghue used this large set of matches to verify that importance values calculated algebraically based on the scoring structure system of tennis and a constant server win probability for each server were correct. For our project we used point importance data that was entered by Kovalchik (2019) as part of the `deuce` `R` package.

Point Data Modeling

Mixed Effects Model

For the point level data, we initially decided to use a mixed effects model with the serving player and point importance as predictors for serve speed. This type of model is useful because it allows for fixed and random effects in the model. In this type of model random effects mean that a variable of this type can vary across the group of players and a fixed effect means that a variable has the same impact on all players in the group.

We built several different models of this type. The first model used just the serving player variable for a random effect with random intercepts. The second model used point importance for a fixed effect and serving player for a random effect with random intercepts. The last model used importance for a fixed effect and serving player for a random effect with random intercepts and slopes. By comparing these three models we hoped to be able to establish how important the point importance values were in predicting serve speed. We found that all three of these models performed very similarly. Most striking was the fact that while the importance predictor was statistically significant, it barely improved serve speed prediction. Seeing as this was the case, we believe that point importance is associated with serve speed, but not enough to drastically change predictions of serve speed.

Neural Networks

After finding out that point importance did not play a large role in predicting serve speed, we decided to try building a neural network for serve prediction. Neural networks are systems of algorithms that are loosely based on the structure of neural networks in human brains and are able to improve themselves through an intensive training process (McCulloch & Pitts, 1943). They are composed of layers of neurons that accept signals from preceding layers of neurons. The first layer is called the input layer, which consists of one neuron for each predictor. The next layer is called the hidden layer and has the number of neurons set by the person building the network. Each of the neurons in the hidden layer receives a signal from each of the neurons in the input layer. The neurons in the hidden layer assign weights to the signals received before sending a signal to the final layer, the output layer. The output layer has one neuron if the variable being predicted is numerical or many neurons if the variable being predicted is categorical. At the output layer, the neurons combine all the signals from each of the hidden layer neurons to calculate a final prediction.

When a neural network runs training data for the first time, it randomly assigns weights to all of the hidden layer neurons and all of the connections between all of the neurons. The network then undergoes a process called backpropagation. Backpropagation consists of defining an error function that models the residuals of its predictions. After assigning random weights, the network computes partial derivatives of the error function with respect to each weight assigned to the network's neurons and connections. This is done in an attempt to find a local minimum of the error function. At the error function's minimum, the partial derivatives with respect to the weights will be 0. Thus, after each iteration of the network assigning weights and evaluating the error function, the network re-assigns weights to the neurons to see if the partial derivatives got closer to

0. This is repeated until a certain threshold for the error function is reached or until a specified number of iterations is completed (Nielsen, 2019).

One drawback that neural networks experience is a lack of interpretability. Since each predictor is sent to each of the hidden layer neurons, it is difficult to say which predictors are contributing to the results the network gives. In the case of this project, a lack of interpretability is a significant drawback. A large motivating factor of this portion of the project is a desire to see if point importance plays a role in serve speed and if so, how large a role. Neural networks make this difficult so we only intended to include the model in our app if its predictive abilities were much better than the mixed effects model's predictive abilities.

To build our neural network, we used the `neuralnet` `R` package (Fritsch et al., 2019). Our first neural network used point importance and serving player as predictors to predict serve speed. Unfortunately, this neural network was only marginally better at predicting serve speed than the mixed effects model we previously built. In addition, our neural network did not help us understand how point importance plays a role in serve speed. Due to this, we decided to attempt to build a different neural network that would predict serve location instead of serve speed.

The goal of this portion of the project was to predict the placement of a serve as best as possible. Placement was categorized into three categories (`W` for a serve that is hit out wide, `B` for a serve hit to the opponent's body, and `C` for a serve that is hit in the center of the court). This new neural network was different in that it was predicting a categorical variable rather than a numerical variable. It also included many more predictors than the previous network. These include a variable for whether the serve is a first or second serve, a variable for the side of the court being served on, a variable for whether the server is ahead in the game being played, a variable for the dominant hand of the serving player, a variable for the height of the returner, and a variable for whether the match is being played on hard court or a different surface. Like the previous neural network this new network also included a predictor for point importance and indicator variables for the serving player.

Once our new neural network was trained, we needed to decide how to evaluate its predictive abilities. Our first goal was to have it be better than random chance at predicting serve location, a very low bar. Since the categorical response variable had three options we obviously needed the network to be able to predict serve location correctly more than one third of the time. We also wanted the network to predict serve location correctly more often than a basic proportion model. The proportion model we compared the neural network to looked at the training data set to see which serve location appeared most often and predicted this serve location every time for the test data set. Using this method, we found that a basic proportion model would predict correctly between 45-47% of the time depending on the sampling for the training and test data sets. In comparison, our neural network was able to predict serve location correctly between 50-52% of the time.

Considering that the test data set included 8000 points, a difference of 5% in predictive abilities is about 400 points. However, we came to the conclusion that this difference was not enough to warrant including the neural network model in our app without further exploration. We also felt that the loss in ability to interpret how the model was making the predictions it was returning was a significant drawback for this project in particular. We expect to dig deeper into this throughout the coming school year.

Conclusion

Overall, our app performs as we hoped. It is fast and responsive as well as intuitive. It reacts how the user would expect and plots lines that depict valuable information. It is unfortunate that the point importance data could not be utilized more than it was but we felt that the neural network and mixed effects models did not add enough value to serve analysis to warrant including them in our app.

We expect to continue researching this throughout the coming semester. Potential areas we will look at further include trying different neural network packages in hopes of cutting down on network training time, finding new predictors that will enhance the neural network's predictions, and exploring new variables to build different Bradley-Terry models. We also hope to look into the downward trends that appear for some players in the Bradley-Terry model. Finding direct evidence of why these patterns arise would be valuable in interpreting the results the model returns.

Works Cited

Fritsch, Stefan & Guenther, Frauke & Wright, Marvin N. (2019). `neuralnet`: Training of Neural Networks. `R` package version 1.44.2. https://CRAN.R-project.org/package=neuralnet

Kovalchik, S., & Ingram, M. (2016). Hot heads, cool heads, and tacticians: Measuring the mental game in tennis (ID: 1464). In MIT Sloan Sports Analytics Conference.

Kovalchik, Stephanie (2019). `deuce`: Resources for Analysis of Professional Tennis Data. `R` package version 1.3

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics, 5(4), 115-133.

Nielsen, M. (2019, December). Neural Networks and Deep Learning. Retrieved July 07, 2020, from http://neuralnetworksanddeeplearning.com/chap2.html

O'Donoghue, P. G. (2001). The most important points in grand slam singles tennis. Research quarterly for exercise and sport, 72(2),125-131.

Sackmann, Jeff. tennis_wta, (2020), GitHub Repository, https://github.com/JeffSackmann/tennis_wta

Sackmann, Jeff. tennis_slam_pointbypoint, (2020), GitHub Repository, https://github.com/JeffSackmann/tennis_slam_pointbypoint

Turner, Heather & Firth David (2020). Bradley-Terry Models in `R`: The `BradleyTerry2` Package. Journal of Statistical Software, 48(9), 1-21. URL https://www.jstatsoft.org/v48/i09/.

Turner, Heather & Firth David (2020). Bradley-Terry Models in `R`: The `BradleyTerry2` Package. https://cran.r-project.org/web/packages/BradleyTerry2/vignettes/BradleyTerry.pdf