Dylan Webb Tennis Data Project

December 9, 2020

1 Introduction

Tennis is a popular sport worldwide, with tournaments held throughout the year to test the strength of the world's best players. Four tournaments in particular, known as the Grand Slams, determine the greatest players. Three male players–Roger Federer, Rafael Nadal, and Novak Djokovic–have monopolized recent Grand Slams, collectively winning 50 of the 59 Grand Slam men's championships in the past 15 years. Nevertheless, these three players are aging, and a pertinent question in the tennis world is who will be the next greatest male players?

1.1 Research Question

The aim of this project is to predict which current male tennis players are most likely to win Grand Slams in the absence of Roger Federer, Rafael Nadal and Novak Djokovic. It is a popular topic among tennis fans and many conjectures have been made on observation alone. Miller (2020), a sportswriter for the New York Times, collected the sentiments of tennis experts and retired players. He reports that Dominic Thiem, Stefanos Tsitsipas, Daniil Medvedev, Andrey Rublev and Alexander Zverev are most likely to topple the current great players. Analytic attempts have also been made to answer this question. Kovalchic (2020), a data science researcher, compares the most promising younger players' trends in current ranking to those of Federer, Nadal and Djokovic at the same age. Like Miller, she proposes that Dominic Thiem, Stefanos Tsitsipas, Daniil Medvedev and Andrey Rublev show promise, with the addition of Matteo Berrettini. Indeed, the six aforementioned players currently hold 3rd, 4th, 6th, 7th, 8th and 9th place in the international Association of Tennis Professionals (ATP) rankings, with Djokovic at 1st, Nadal at 2nd and Federer at 5th (due to injury).

There have been multiple studies predicting, not the greatest future tennis players, but the winners of tennis matches using machine learning techniques. Most notably, the random forest model by Gao and Kowalczyk (2019) boasts 83% accuracy. Another type of predictive model, an analytic network process, was used by Gu and Saaty (2019) to predict winners with 85% accuracy. Due to the efficacy of random forests in tennis match prediction demonstrated by Gao and Kowalczyk, I consider the prediction of future dominating players answerable via machine learning techniques with adequate data. To answer my research question, I created my own random forest prediction model and optimized its accuracy on existing data. I then fabricated matches between the top 30 ranked players. With a highly accurate random forest model, one would expect the players with the most potential to be correctly predicted as the winners of these fabricated matches.

2 Data

The data I used for this project was compiled by Jeff Sackmann, an author and software developer who enjoys tennis analytics. The data was scraped directly from the ATP website and formatted into csv files found on https://github.com/JeffSackmann/tennis_atp. The ATP is the worldwide governing body on men's professional tournaments and collects standard statistics for all matches. Sackmann's csv files include the ATP statistics on each match for the larger tournaments over 48 years. Because it contains valuable statistics on the serve, considered the most influential stroke in the game of tennis, I expected this data to be applicable to my prediction model. Analyzing the data revealed which features were most conducive to winning a match.

2.1 Data Cleaning

To prepare the data for analysis, I combined Sackmann's 2003-2019 data into one large data frame. Because some tournaments are smaller than others, the player statistics are not present for every recorded match in the data set. For this reason, I kept only the data from the four Grand Slam tournaments: the Australian Open, Roland Garros, the US Open, and Wimbledon. Among the remaining data, there were still a few missing entries, so I removed all rows with any blank entries.

Of the many statistics provided by Sackmann, I used the following nine numerical statistics for both winners and losers: total number of aces, double faults, points served, first serves in, first serves won, second serves won, serve games, break points saved, and break points faced. Combining these 18 with year, tournament name, court surface, winner name and loser name, I began with 23 features. Problematically, these features are totals rather than proportions, making players who win longer matches appear disproportionately skilled. For this reason, most of my engineered features are proportions.

3 Feature Engineering

3.1 Individual Match Features

There are many different racket movements, or strokes, in the game of tennis. Each time a player hits the ball, their stroke is largely determined by the previous stroke made by their opponent. Therefore, the only stroke entirely under a player's control is the first stroke of the game—the serve—making it an accurate predictor of a player's offensive capabilities. Likewise, the second stroke in the game, the return, is an accurate gauge of a player's defensive capabilities.

From the work of Gao and Kowalczyk (2019), the serve is suggested as the most important feature for a random forest model, so I engineered the following six features by combining existing ones: number of second serves in, proportion of serves in which were first serves, proportion of serves in which were second serves, proportion of first serves won, proportion of second serves won, and proportion of all serves won.

Following the suggestions of Gu and Saaty (2019), who found the return to be the most important feature in their model, I engineered three additional features: the proportion of returns won on the opponent's first serve, the proportion of returns won on the opponent's second serve, and the proportion of all returns won.

Combining the above engineered features for both winners and losers, I had 41 total features describing an individual match: 18 quantitative features for both winner and loser and 5 categorical features describing the match. As a reminder, the 5 categorical features are as follows: year, tournament name, court surface (which I label encoded), winner name and loser name. The 5 categorical features and the proportion of returns won on the opponent's first serve (1stRnWon%), the proportion of returns won on the opponent's second serve (2ndRnWon%, and the proportion of serves won (svWon%) are stored in tennis.csv, generated in Appendix A. These last three features are present for both the match winner and loser, with respective prefixes w and l.

```
[ ]: pd.set_option('display.max_columns', None)
     pd.read_csv("tennis.csv").tail()
```

```
[ ]:       year tourney_name  surface        winner_name         loser_name  \
     8597  2019      US Open      1.0  Matteo Berrettini        Gael Monfils
     8598  2019      US Open      1.0       Rafael Nadal  Diego Schwartzman
     8599  2019      US Open      1.0    Daniil Medvedev    Grigor Dimitrov
     8600  2019      US Open      1.0       Rafael Nadal  Matteo Berrettini
     8601  2019      US Open      1.0       Rafael Nadal    Daniil Medvedev

           w_svWon%  w_1stRnWon%  w_2ndRnWon%  l_svWon%  l_1stRnWon%  l_2ndRnWon%
     8597  0.629630     0.285714     0.546875  0.611111     0.306818     0.375000
     8598  0.626374     0.428571     0.540541  0.526882     0.298246     0.298246
     8599  0.606838     0.372881     0.526316  0.567010     0.254545     0.581818
     8600  0.826667     0.341772     0.564103  0.584746     0.097561     0.219512
     8601  0.666667     0.350427     0.461538  0.609890     0.225806     0.344086
```

3.2 Five-Year Match Features

The aim of this project is to predict who will become the next greatest men's tennis players. Because I answer this question through simulated tennis matches, an important goal of the data is to predict the outcome of future matches. However, my initial 18 quantitative features (from the above section) describe the ability of a player *after* the outcome of a given match. I therefore needed quantitative features which would describe a player's ability against a specific opponent before the match occurred. I will hereafter refer to the 18 quantitative features describing a specific match as “individual match” features, and the features summarizing ability prior to this match “five-year match” features.

The function generateStats from Appendix A takes as input a single individual match feature from the 18 quantitative individual match features described above, as well as the names of two players. It then calculates the following five-year match features on the given individual match feature for both players: average value of the feature over five years in all matches won, in all matches lost, in matches won against common opponents, in matches lost against common opponents; and variance of the value of the individual match feature over five years in all matches won, in all matches lost, in matches won against common opponents, and in matches lost against common opponents.

Common opponents are those which both input players had matches with in the last five years. This is an important metric because if the two input players have not participated in many other Grand Slams, they have a miniscule chance of having played each other in a Grand Slam in the past five years. However, there is a greater chance of the input players having both played the same opponent during this time.

Rather than summarize a player's ability over their entire career, I chose to summarize it over five years (through the averages and variances described above) because player ability can fluctuate over time; data from more recent years is more indicative of present success. I first summarized ability over three years but found that five years raised model accuracy.

From a single inputted individual match feature I could now output 16 five-year match features describing two players' past ability in general and with respect to common opponents. In the final version of my model, I predict past performance for three individual match features, making 48 total five-year match features. These 48 features are stored in stats.csv, generated in Appendix A. The file is too large to preview in this report; the lengthy output can be found in Appendix I.

4 Data Visualization and Analysis

I now turn to the creation of my random forest tennis prediction model. All accuracies in this section are benchmarks performed by testing the model on matches with a recorded outcome.

4.1 Random Forest Classifier

I first used a random forest classifier to test the effectiveness of my individual match features and determine their relative importance in predicting the outcome of a tennis match. Appendix C contains the code for this initial classifier utilizing permutation importance. I split the individual match statistics into training data and testing data. The random forest takes as input all the individual match features shared by winner and loser along with court surface (which I label encoded), making 19 total. The forest is then trained on whether the features came from a winner or a loser. Thus, when tested on a player's 19 individual match features, the forest predicts whether the player won or lost the given match and is tested against the actual outcome of the match. The feature permutation importance and accuracy is displayed below:

```
Ranking All Features:           Ranking Top Three Features:
2ndRnWon%                       2ndRnWon%
svWon%                          svWon%
1stRnWon%                       1stRnWon%
rnWon%                          Accuracy = 0.9827586206896551
1stWon%
2ndWon%
bpFaced
1stWon
ace
2ndIn
2ndWon
svpt
df
SvGms
bpSaved
surface
2ndIn%
1stIn%
1stIn
Accuracy = 0.9845021309569935
```

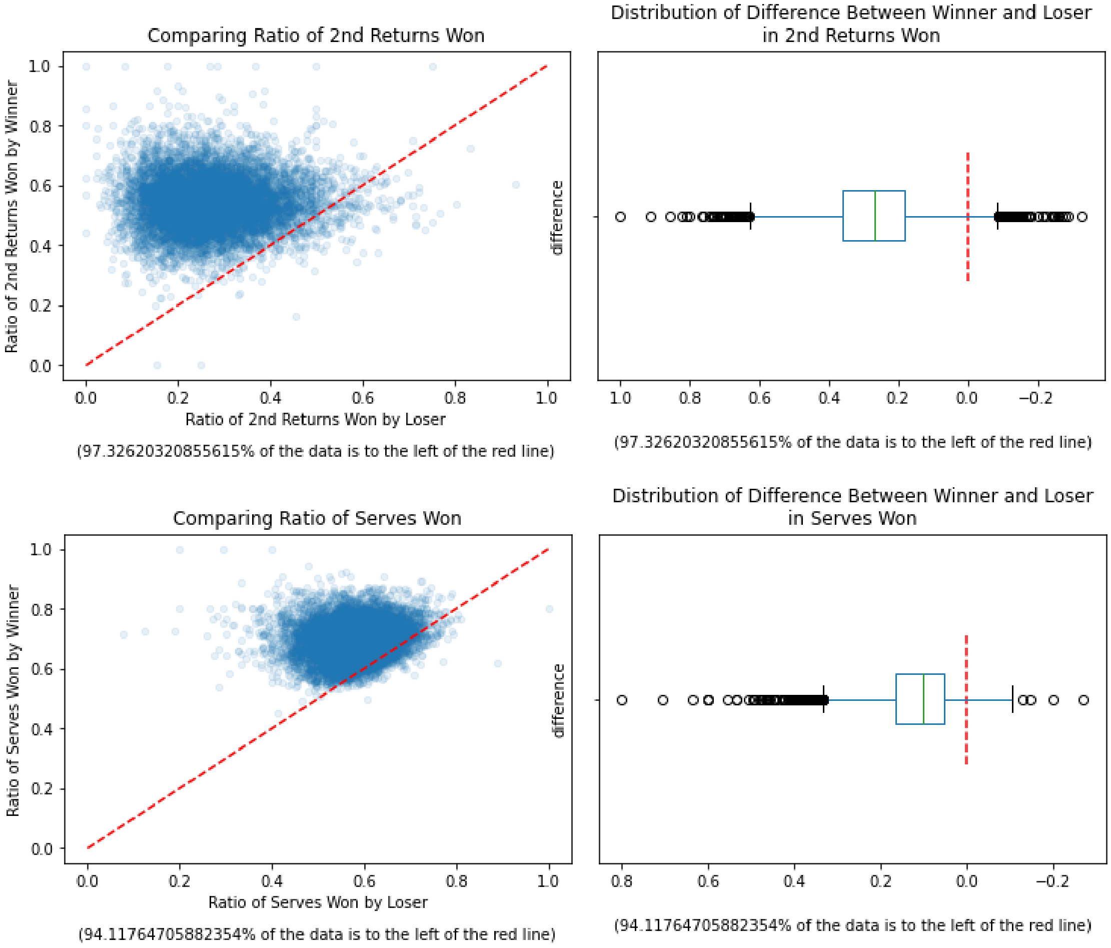
The ranking reveals which three individual match features are most important for predicting the winner of a match: the proportion of returns won on the opponent's second serve, the proportion of serves won, and the proportion of returns won on the opponent's first serve. When all other features are removed, the prediction model drops only slightly in prediction accuracy–about 98% of its predicted outcomes were correct when compared with actual match outcomes. Running a principle component analysis on the same set of individual match features confirms that only three features convey most of the information needed to predict the winner. The following result is obtained using the code in Appendix F:

```
Variance Explained by Top Three Features: 0.9792233054518066
```

We can infer that the top three features in the principle component analysis are the same as those returned by the random forest feature ranking: proportion of returns won on the opponent's second serve, proportion of serves won, and proportion of returns won on the opponent's first serve. The following visualizations–generated in Appendix D–confirm the efficacy of these three individual match features in predicting the outcome of a match. Each graphic compares one of the three features between the winner and loser of a match. The red, dashed line is at equilibrium, so points to the left of the line in each graphic represent matches where the winner outperformed the loser in the depicted feature. Most points lie to the left of the line in each graphic, which is why these three individual match features are good predictors of winning.

```
[ ]: Image("2ndRnWon%_svWon%.png")
```

[ ]:



(97.32620320855615% of the data is to the left of the red line)

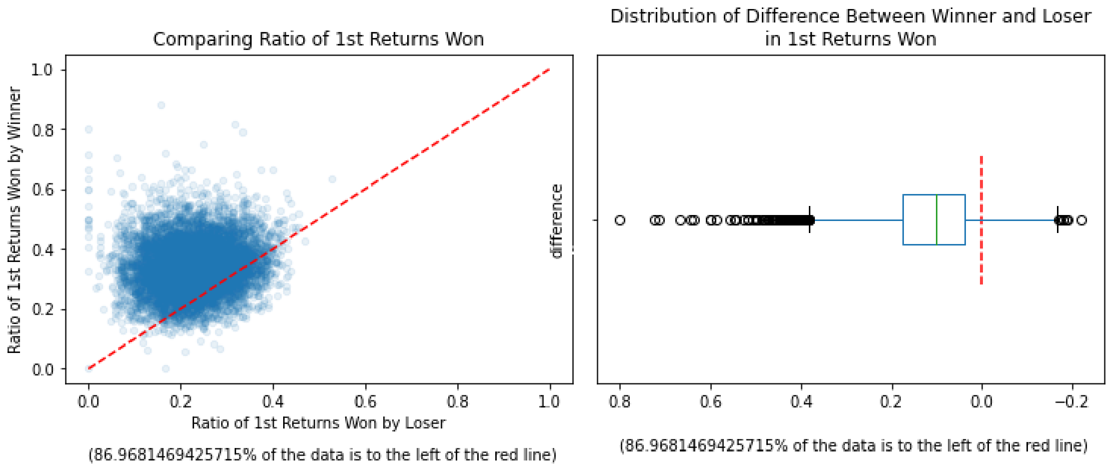
(97.32620320855615% of the data is to the left of the red line)

(94.11764705882354% of the data is to the left of the red line)

(94.11764705882354% of the data is to the left of the red line)

```
[ ]: Image("1stRnWon%.png")
```

[ ]:



With the three individual match features pictured above, my random forest classifier predicts match outcome with 98% accuracy, so I discarded the remainder of the quantitative individual match features. Consequently, the tennis.csv file generated in Appendix A has only 11 features: five categorical individual match features and the proportion of returns won on the opponent's second serve, the proportion of serves won, and the proportion of returns won on the opponent's first serve for both players. These three individual match features do not exist, however, if a match has yet to occur. To use the forest classifier to predict the outcome of a future match between two players, I needed to first predict the values of the three individual match features.

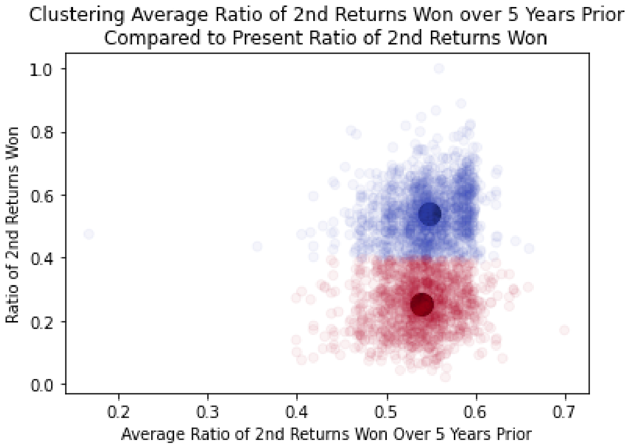
4.2 Random Forest Regressor

To predict the three individual match features, I created a random forest regressor which uses the 48 five-year match features from my feature engineering to predict the 3 individual match features needed for the random forest classifier. The regressor is trained on the stats.csv file generated in Appendix A, which contains 54 features: the 3 individual match features needed for the classifier, the 48 five-year match features, the year, the court surface, and the outcome of the match. stats.csv contains these 54 features for every Grand Slam match from 2008 to 2019. There was at least one blank entry for about half the matches (likely due to players not having participated in many Grand Slams); I removed these matches for more accurate training of the regressor. The optimal values for minimum samples per leaf (6), number of estimators (200), and maximum tree depth (150) were then obtained using a gridsearch.

To test the regressor's accuracy, I first split the stats.csv file generated in Appendix A into training and testing data, and trained the random forest regressor to use five-year match features to predict individual match features. I then input the testing data to predict individual match features, which I in turn inputted into the classifier described in the previous section to obtain the predicted match outcomes from the testing data. Comparing these outcomes with the actual match outcomes, the combined regressor and classifier achieved an average score of 69% accuracy, usually in the range of 68% to 70% (Appendix G contains the code for this result). The 48 five-year match features are therefore only adequate predictors of the 3 individual match features. The reason why is captured in the below graphic–generated in Appendix D–of a k-means clustering run on the five-year match feature, "average ratio of 2nd returns won over five years prior," compared to the individual match feature "ratio of 2nd returns won"–a more succinct way of saying "proportion of returns won on the opponent's second serve."

```
[ ]: Image("Clustering.png")
```

[ ]:



We observe two clusters separating higher and lower ratios of 2nd returns won. These two clusters represent winners and losers, respectively, to a high degree of accuracy, as seen in the earlier graphic "Comparing Ratio of 2nd Returns Won," where the winner almost always had a higher proportion of second returns won than the loser. The cluster centers–represented by the darker gray circles–are only slightly offset, however, making it difficult to predict the proportion of returns won on the opponent's second serve (an individual match statistic) based on the average proportion of returns won on the opponent's second serve over five years prior (a five-year match statistic). There is very little linear correlation between any of the five-year match features and the corresponding individual match features, explaining the random forest regressor's lower accuracy. In turn, these inaccurate predictions lower the effectiveness of the random forest classifier.

As observed, using two random forests in succession compounds each forest's independent inaccuracies. To increase accuracy, I attempted to implement a single random forest classifier which predicts the match outcome directly from the five-year match features. In the resulting model, however, overall accuracy was lower. This is likely because the random forest regressor outputs predicted individual match features which, even when inaccurate, can fall within an acceptable range of possible values to be correctly classified by the random forest classifier. This flexibility is lost when the entire prediction is performed in one inaccurate classifier. If stronger five-year match features were added, however, re-attempting a single random forest classifier would be recommended.

4.3 Clustering

The k-means clustering above separated the data into two distinct groups resembling winners and losers, suggesting that k-means could be a more accurate predictor of match outcome than the random forest model. However, this correct clustering only occurred because the proportion of second returns won is included in the visualization–in predicting a future match, this individual match statistic is unknown. Nevertheless, it was worth exploring because clustering returns a prediction for match outcome in only one step, whereas my random forest model takes two.

To test the potential of k-means clustering, I again split the stats.csv file generated in Appendix A into training and testing data and clustered the five-year match features into two clusters. Comparing these outcomes with the actual match outcomes, k-means clustering achieved an average accuracy of just above 50%–no better than guessing. I ran the same test using spectral clustering with a Gaussian kernel, but there was no observable improvement. Appendix H contains the code for these results.

4.4 Approximate Features Algorithm

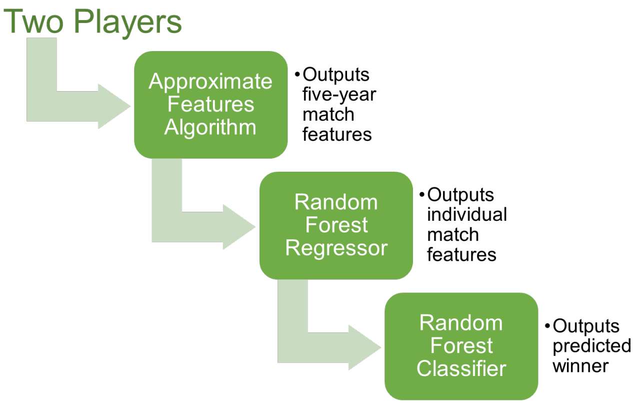
After attempting clustering, I continued developing my double random forest model: a random forest regressor feeding into a random forest classifier. As described in the Random Forest Regressor subsection, I initially trained and tested my model on the stats.csv file generated in Appendix A with 69% accuracy. As a reminder, I removed blank entries from the stats.csv file for more accurate training; these blank entries represented players with missing five-year match features and had therefore played in few Grand Slams. Hence, my model was only tested on players with no missing five-year match features.

To test the model on all players, I needed to fill in the missing features. I modified the generateStats function found in Appendix A to become the approximateFeatures function found in Appendix B. Rather than removing missing five-year match features, it replaced them with the average corresponding feature value for players with no missing data–averaging down the column in stats.csv for the missing five-year match feature.

The double forest prediction model now functioned as follows: the approximate features algorithm calculated the five-year match features for two players (with gaps replaced by global averages). These were input into the random forest regressor, which ran three times to predict each of the three individual match features needed in the classifier. The predicted individual match features were then input into the random forest classifier, which returned the predicted winner of the match. The following graphic visualizes the model hierarchy:

```
[ ]: Image("Model Flowchart.png")
```

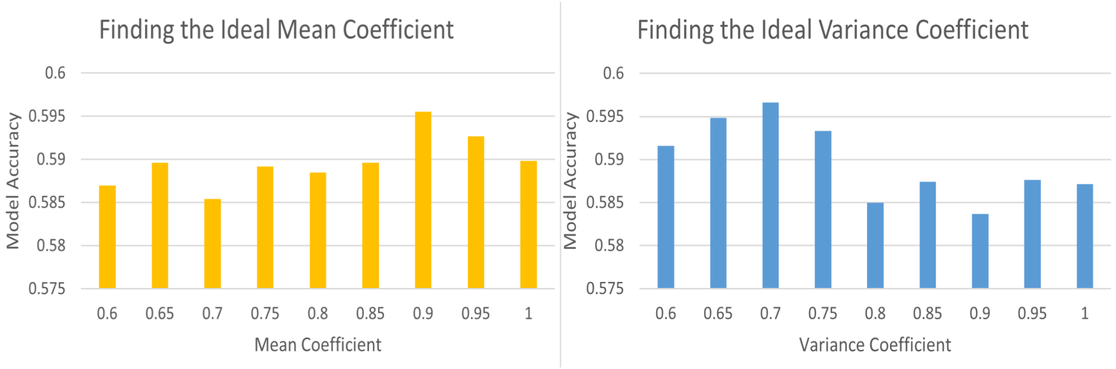
[ ]:


Two Players
Approximate Features Algorithm
•Outputs five-year match features
Random Forest Regressor
•Outputs individual match features
Random Forest Classifier
•Outputs predicted winner

When using this model to predict the outcomes of every Grand Slam match in 2019, the double random forest model achieved only 57% accuracy, significantly lower than the 69% accuracy when tested on the stats.csv file. This drop occurs because the players with missing data typically have less skill (because they qualify for fewer Grand Slams) than the average player with no missing data. Hence, the global average values were overestimating the missing values. To correct for overestimation, I multiplied the average means and variances by coefficients less than one. I ran a grid search in the form of two nested for-loops on the random forest model to find the values for these coefficients which optimized model accuracy. The following charts were created using Microsoft Excel and the results of the grid search (found in Appendix J).

```
[ ]: Image("Optimal Coefficients.png")
```

[ ]:



My grid search showed that the best mean coefficient was 0.9 and the best variance coefficient was 0.7. After substituting missing values with global averages multiplied by these coefficients, the double random forest model boasted up to 64% accuracy (with a usual range of 60% to 63%). This is the final form of the prediction model, as found in Appendix B.

The following result is obtained using the code in Appendix E which runs the prediction model on all men's singles matches played in Grand Slams during 2019:

```
Proportion of Accurate Predictions: 0.6403162055335968
```

Although my research question centers on the next greatest male tennis players, I wanted my model to be applicable to female players as well. However, when using the same code, average accuracy dropped by about 5%. This is likely because the mean and variance coefficients found above are optimal for men's tennis. More improvements must be made to obtain equal model accuracy among all players.

The following result is obtained by modifying the code in Appendix E to run the prediction model on all women's singles matches played in Grand Slams during 2019:

```
Proportion of Accurate Predictions: 0.5900990099009901
```

Despite the double random forest model's moderate success at predicting match outcome for both men and women, it did not reach the same level of accuracy as when tested on the stats.csv file.

This indicates that the contents of stats.csv are not well-suited as training data. A recommended avenue for future improvement is exploring better predictive five-year match features.

5 Applying the Double Random Forest Model

I now return to my research question–predicting the next greatest male tennis players–by applying my double random forest model. I also test the hypotheses of Miller (2020) and Kovalchik (2020), who proposed the following players as candidates for next greatest: Dominic Thiem, Stefanos Tsitsipas, Daniil Medvedev, Andrey Rublev, Alexander Zverev, and Matteo Berrettini. All six of these players are currently ranked among the top 30 by the Association of Tennis Players, so using the completed prediction model, I tested the relative skill of the top 30 ranked players by simulating a round robin tournament. This is a tournament where each player plays every other player exactly once, which is better than a bracket for comparing the relative performance of every player.

I performed this simulation using the roundRobin function in Appendix B, assuming the environment of the Australian Open during the year 2020. I selected the Australian Open, a hard court tournament, because some players perform better on clay or grass, so hard court is a more neutral playing field. The Australian is also the first Grand Slam of the year, and thus most directly affected by player performance in 2019. Although it would have been preferable to simulate the round robin in 2021, Jeff Sackman has yet to update his database with all the player statistics for 2020. The results of the simulation follow, with the left column displaying ATP ranking and the right column totalling the number of simulated matches won.

```
[ ]: roundRobin(pd.read_csv("Top30.csv"), 2020, "Australian Open")
```

```
Round Robin                                  Round Robin Continued
ATP ranking           name  score            ATP ranking           name  score
14            Milos Raonic     28            9         Matteo Berrettini     15
1            Novak Djokovic    27            18            Stan Wawrinka     13
11             Gael Monfils    26            16        Pablo Carreno Busta   12
2              Rafael Nadal    26            17            Fabio Fognini     12
5             Roger Federer    26            13    Roberto Bautista Agut     12
25               John Isner    23            21    Felix Auger-Aliassime      8
12         Denis Shapovalov    21            23           Alex De Minaur      8
19          Grigor Dimitrov    21            22           Cristian Garin      7
3             Dominic Thiem    20            24              Borna Coric      7
6        Stefanos Tsitsipas    19            20          Karen Khachanov      6
15             David Goffin    19            29             Taylor Fritz      5
4           Daniil Medvedev    18            26           Dusan Lajovic      4
7          Alexander Zverev    17            27              Casper Ruud      3
10        Diego Schwartzman    15            28             Benoit Paire      1
8             Andrey Rublev    15            30              Ugo Humbert      1
```

Interestingly, Milos Raonic is predicted as the winner of the round robin, despite being ranked 14th. Although Raonic often plays in the Grand Slams, he has never won, advancing to the final round only once. One reason for his high ranking could be the model's preference for serve ability–Raonic is known to have the third highest serve win ratio of all time at 91%. Excluding Djokovic, Nadal and Federer, the current three greatest players (Federer is ranked fifth for having missed recent tournaments due to an injury), the prediction model finds Milos Raonic, Gael Monfils, John

Isner, Denis Shapovalov and Grigor Dimitrov to have the greatest potential (in decreasing order) of becoming the next greatest players. None of the players highlighted by Miller (2020) and Kovalchik (2020) were ranked in the top five. Because these players are relatively new to Grand Slams and the model predicts based on five years of previous data, one would expect the prediction model to be slightly biased against them.

An important additional finding was the potential of the random forest model to predict the winner of an entire tournament from only the first round of contestants. This is achieved by making a prediction on the first round, and then putting those round winners in a new bracket (with the same progression as defined by the tournament) and predicting again, repeating this process until one player remains. This function is defined as predictTournament in Appendix B. We see below that by applying this process to the first round of the Roland Garros tournament of 2020, Rafael Nadal is ultimately projected to play Novak Djokovic for the final round and win (the first three predicted rounds were removed for brevity). This is exactly what happened in the final round of the recent tournament. The predictions for the progression of other players, however, are not as accurate, again reflecting the preference of the tennis prediction model for players who play most often in Grand Slams. Because the predictTournament function iterates on only moderately accurate predictions, it amplifies the prediction model's existing inaccuracy and produces inaccurate predictions for most players.

```
[ ]: predictTournament(pd.read_csv("round1.csv"), 2020, "Roland Garros")
```

```
Round 4
     predicted_winner  likelihood
0       Novak Djokovic    1.000000
1   Jan Lennard Struff    0.600000
2          Sam Querrey    1.000000
3   Stefanos Tsitsipas    0.533333
4         Gael Monfils    0.866667
5          Andy Murray    1.000000
6         David Goffin    0.933333
7         Rafael Nadal    1.000000

Round 5
  predicted_winner  likelihood
0   Novak Djokovic    1.000000
1      Sam Querrey    1.000000
2     Gael Monfils    0.666667
3     Rafael Nadal    1.000000

Round 6
  predicted_winner  likelihood
0   Novak Djokovic         1.0
1     Rafael Nadal         0.8

Final Round
  predicted_winner  likelihood
0     Rafael Nadal    0.866667
```

6 Conclusion

The final prediction model relies on the following values averaged over five years: the proportion of returns won on the opponent's second serve, the proportion of serves won, and the proportion of returns won on the opponent's first serve. The model hierarchy comprises three sections–the approximate features algorithm, followed by a random forest regressor, followed by a random forest classifier–and achieves an overall maximum estimated accuracy of 64% when predicting the outcomes of tennis matches.

When simulating a round robin among the current top 30 tennis players, the prediction model failed to corroborate the hypotheses of Miller (2020) and Kovalchik (2020). It predicted instead the future dominance of Milos Raonic, Gael Monfils, John Isner, Denis Shapovalov and Grigor Dimitrov, likely due to a preference for players with more experience in Grand Slams and high serve strength (in the case of Milos Raonic). The lack of data from 2020 also skewed the results away from newer players.

When simulating an entire tournament from the first round alone, the prediction model was successful in predicting the two finalists and champion. The projected success of other players, however, was highly inaccurate. This is again because of the model's preference for participation in Grand Slams.

Despite its shortcomings, the double random forest model is generally successful. Through future improvements to the five-year match feature engineering, overall accuracy could likely be improved.

6.1 Ethical Implications

The ethical dangers posed by my research question and prediction model are minimal. The data collected is public domain and represents only player performance statistics and not any personal information. The success of the tournament-level prediction function in correctly predicting Rafael Nadal as the ultimate winner should not be misconstrued to represent model accuracy–this is only representative of an overall bias toward players with more Grand Slam experience. The conclusion of the round robin is likewise misleading; experts would agree it is unlikely that Milos Raonic becomes the next greatest player. Again, this result only reveals a bias toward players with a particularly high serve ability. Apart from these potential misunderstandings, the prediction model is a transparent system posing no danger of malicious feedback loops and no threat to the careers of tennis players.

7 References

Gao, Z., & Kowalczyk, A. (2019). Random forest model identifies serve strength as a key predictor of tennis match outcome. *Arxiv.* https://arxiv.org/ftp/arxiv/papers/1910/1910.03203.pdf

Gu, W., & Saaty, T. (2019). Predicting the outcome of a tennis tournament: Based on both data and judgments. *Journal of Systems Science and Systems Engineering, 28*(3), 317-343. https://doi.org/10.1007/s11518-018-5395-3

Kovalchik, S. (2020, January 19). Who can break up the 'Big 3' monopoly on men's tennis? Here's what the numbers say. *The Conversation.* https://theconversation.com/who-can-break-up-the-big-3-monopoly-on-mens-tennis-heres-what-the-numbers-say-127991.

Miller, S. (2020, January 19). Rising Tennis Stars for the New Decade.*The New York Times.* https://www.nytimes.com/2020/01/19/sports/tennis/rising-stars.html.