**A Machine Learning Approach to Men’s Tennis Prediction**

Ishan Dhanani

Department of Economics, Texas A&M University

ECON 675: Capstone

Dr. Li Gan

November 1, 2022

**A Machine Learning Approach to Men’s Tennis Prediction**

**Introduction**

Sports have always been prevalent in our society. From early variations of boxing to the rising sport of spike ball, we have always emphasized these physical activities. With the rise of modern technology, many aspects of sports have become datasets allowing for the intersection of sports and data analytics. One of the most popular sports to harness these data-driven insights is tennis. With the growth of technologies like UTR and Tennis Recruiting, players are constantly looking for ways to use techniques including machine learning and data analytics to augment their game. In addition, predicting the win/loss binary outcomes has long been the goal of various sports betting firms and players.

Tennis is one of the largest international sports. Tennis players come from various backgrounds and countries and compete in the Association of Tennis Professionals circuit. This year-long circuit hosts different levels of tournaments worldwide on three different surfaces. Each match produces a large amount of data, and numerous variables come into play before and during the match. As a lifelong tennis player and fan, I decided to model a tennis match solely based on pre-match data and predict its outcome.

One of the simplest methods to tackle the problem of binary classification is logistic regression. Its advantages include computational simplicity and an interpretable result. Recently, new machine learning techniques including extreme gradient boosting (XGBoost) and neural networks are also being utilized to analyze sports data. These methods are extremely powerful and can model the complex and high-dimensional data found in sports. This paper attempts to use these complex models to develop a prediction model for ATP tennis matches. This paper will use logistic regression, random forest, gradient-boosted trees, and a simple neural network. **Literature Review**

Neural networks have been repeatedly used in conjunction with sports data to gain actionable insight across many sports. A relevant example is another A&M student who used Artificial Neural Networks to predict NFL player positions (Manage 2019). Although its use in tennis is not as prevalent, there has been literature that has used Artificial Neural Networks to predict binary outcomes in tennis. However, due to the hierarchal nature of tennis (points turn into games which turn into sets), the primary models used to predict tennis have been statistical in nature and use Markov Chains. Although some aspects of this paper are mentioned in other papers, this is a unique contribution to using only pre-match-engineered features and a new confidence metric to predict win/loss.

As mentioned earlier, various methods have been used to predict tennis matches. One of the earliest models comes from Klaasen and Magnus which uses the Markov Model mentioned above. This model takes the probability that each player wins a point if serving and then creates a tree of outcomes using this probability. As the match goes on, the probability updates with new information that harnesses a tennis match's constantly evolving nature (Klassen 2003). In 2016, renowned tennis data analyst Stephanie Kovalchik used regression and paired comparison using metrics including ELO rating. Her prediction results varied between 59% and 72%. Most recently, Jacob Gollub explored these hierarchal models and added novel methods to analyze these models in his work Forecasting Serve Performance in Professional Tennis Matches (Gollub 2017). The most recent addition has been the use of the ‘Glicko Model” to predict tennis matches. Developed by Jack Yue and Elizabeth Chou, the ‘Glicko Model” pulls from Bayesian models primarily used in chess and applies it to various pre-match and during-match metrics. This paper also used different ML techniques including logistic regression and light gradient boosting to evaluate its results (Chou et. al. 2022)

Research works that use neural networks for tennis include Michal Sipko’s Machine Learning for the Prediction of Professional Tennis Matches (Sipko 2015) and DeepTennis: Mid-Match Tennis Predictions (Lerner 2019). Sipko’s work focused on developing and extracting novel features including fatigue and injury from existing data sets and using them as predictors for the model. He outperformed industry standard stochastic modeling algorithms using logistic regression and ANN's. These works experiment with different neural network architectures including Long Short-Term Models (LSTM) and Multi-Layer Perceptron models. Another approach combining different ML models comes from Andre Cornman’s work on Machine Learning for Professional Tennis Match Prediction and Betting. This paper uses similar models but has a different set of input features. Overall, Cornman found that the ANN performed better on the training set but slightly worse on the validation sets when compared to the Random Forest (Cornman et. al 2019). Finally, another Stanford paper attempts to predict tennis matches using a similar set of input features but only uses logistic regression as its predictive model (Amudan 2019).

**Data**

Data for this project was used from the open-source tennis database hosted by Jeff Sackmann on GitHub. This dataset includes matches from 1968 for all three tiers of the ATP tour. Pre-match data includes ranking, age, height, right/left hand, tournament, and surface. I decided to use tier 1 matches starting in 2000 to account for differences in playstyles and technology. Tennis changed significantly when racquet’s started using composite technology and players started adopting a more baseline strategy. For example, when Lleyton Hewitt won Wimbledon in 2002, he did not serve and volley even once. This is in strict contrast to previous winners and the overall playstyle of aggressive tennis. The modern game has followed suit with most players relying on an aggressive or defensive baseline strategy and very little reliance on winning points at the net.

Although the dataset was mostly complete, there was initially a lot of sparsity in earlier years due to matches that needed to be appropriately recorded or completed. To solve for many of these missing values, I decided to impute as much as possible from my own research but eventually removed the rest of the missing rows.

The dataset was initially labeled by the winner and loser of the tennis match. To avoid bias for each labeled player, I assigned each player to “Player1” and “Player2” based on reverse alphabetical order. Along with the existing features, I developed some metrics that are commonly considered when comparing two players in tennis. The first metric is “head-to-head”. This is a moving sum of wins against a specific player. An example is shown below.

**Table 1**

*The Head-To-Head Feature*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Player 1** | **Player 2** | **Winner** | *H2H\_p1* | *H2H\_p2* |
| *Roger Federer* | *Rafael Nadal* | *Roger Federer* | *0* | *0* |
| *Rafael Nadal* | *Roger Federer* | *Roger Federer* | *0* | *1* |
| *Roger Federer* | *Rafael Nadal* | *Rafael Nadal* | *2* | *0* |
| *Roger Federer* | *Rafael Nadal* | *Rafael Nadal* | *2* | *1* |

The following features developed were cumulative win percentages for different surface and level types. Different tennis surfaces cater to different playstyles. For example, Rafael Nadal has a career grass win percentage of 78.35% but has a career clay win rate of 90%. The different level types correspond to tournament tiers in the ATP. In the calendar year, there are 4 Grand Slam tournaments which take the highest ranked players. The next tier consists of 250-, 500-, and 1000-point tournaments. A vast majority of players have different performances at these events. For example, David Ferrer was a consistent top-10 player for over seven years. He has won numerous titles at different levels but never a grand slam.

The final feature I developed was an indicator of recent form. Although ranking is very deterministic of how a player performs, it typically cannot account for short-term form of a player. For this feature, I added a positive penalty if the player had won more than 10 matches over the last three months. This is equivalent to a player getting to the second or third round at most tournaments for three months. I used an exponential function that qualitatively matches a slow rise in confidence bounded on [0, infinity). The equation is shown below.

**Figure 1**

*Graph of Roger Federer and Novak Djokovic’s 200 Match Recent Form Metric*

*Chart

Description automatically generated*

*Note: When Djokovic entered the ATP tour, his wins were much more sporadic than Federer. That is shown by large peaks and valleys. In contrast, Federer has an upward trend in his confidence metric. Federer’s data starts in 2001 and Djokovic’s starts in 2004*

After the feature engineering, a lot of the early features were zero due to no previous history. To solve the issue of matrix sparsity, I subset the data to include players who had played over 100 matches. This still left me with 22068 observations in the final dataset.

**Models and Results**

To develop a baseline comparison, I used two models. The first model relied only on ranking at time of the match and stated that whichever player had the higher ranking was predicted to win. The second model that I used was a logistic regression that used features that came with the data. The results are shown below.

**Table 2**

*Preliminary model accuracy*

|  |  |  |
| --- | --- | --- |
| Model | Accuracy | Precision |
| Ranking model | 0.656 | - |
| Logistic Regression | 0.61 | 0.609 |

All predictors used for the logistic regression were statistically significant at the 5% level. However, the ranking model prevailed with an accuracy of about 65%. This is the baseline accuracy.

After building the features, I used four different models to predict the matches: Random Forest, XGBoost, Logistic Regression, and an Artificial Neural Network. All these models except for the logistic regression are non-parametric models that can capture complex non-linear relationships within the data. A brief description of each model is provided along with a description on how it was trained and its results. Due to the balanced nature of the dataset (player1 won 11426 times and player2 won 10642), we use accuracy and precision to evaluate the model. Precision is a metric that takes type1 error into account. In this case, that measures what proportion of wins that the model identified was actual correct. I used 10-fold cross validation when training each model.

**Model 1: Logistic Regression**

*Description*

Logistic regression is a binary classification algorithm that predicts the probability of an event occurring. To bound the probability between [0,1], it uses a sigmoid function. Logistic regression can be thought of as a linear classifier wrapped in a sigmoid function.

*Results*

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Train Accuracy | 10-fold CV | CV-Precision |
| Logistic Regression | 0.706 | 0.703 | 0.699 |

The logistic regression beat the accuracy of the ranking model by 5 percentage points and had a higher precision than the original logit model.

**Model 2: Random Forest**

*Description*

Random forests are decision tree-based algorithms and are used to solve for individual decision tree overfitting. Instead of fitting a single tree, a Random Forest combines multiple trees together and selects a random selection of predictors at each split. This is a powerful algorithm but suffers from computational complexity during tuning. The main hyperparameters in a random forest are the number of trees, the number of features considered, max tree depth, and minimum samples needed to split a node. I initialized the model using the randomized cross validation implementation of sci-kit learn totaling 12 fits.

*Results*

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Train Accuracy | 10-fold CV | CV-Precision |
| Random Forest | 0.715 | 0.702 | 0.695 |

Chart, bar chart

Description automatically generatedThe Random Forest model showed a slight improvement of the logistic regression in training accuracy but had a similar cross validation score and accuracy. However, due to computational complexity and limits on my computer, I was not able to tune the random forest for more than 12 iterations. The randomized cross validation was not able to check each grid value which could have contributed to a weaker performance. Below is a graph displaying feature importance for the model

Per the graph, the engineered features were important to the Random Forest model when conducting its splits. Most important was the surface win percentage followed by the level win percentage. This follows my preliminary hypothesis that surface and tournament level play a large role in a match outcome.

**Model 3: XGBoost**

*Description*

XGBoost is another decision tree-based algorithm. However, instead of growing trees in parallel, boosting algorithms grow trees sequentially working on increasing the power of each weak learner. XGBoost has been used widely in industry for its power however it also requires heavy hyperparameter tuning.

*Results*

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Train Accuracy | 10-fold CV | CV-Precision |
| XGBoost | 0.875 | 0.698 | 0.692 |

Due to the limitations of my machine, I was not able to tune the model for as long resulting in overfitting and poor performance on the validation set. This model overall performed similarly to the Random Forest and had similar feature importance with surface and level win percentage deemed the most important by the split criterion.

**Model 4: Artificial Neural Network**

*Description*

A neural network is composed of neurons and layers. The input layer takes in the predictors and the following layers take in the output of the previous layer. Each neuron applies a linear transformation (consisting of a multiplicative weight and added bias) and then bounds it with an activation function (like the sigmoid discussed in logistic regression).

Neural networks perform best when they can “learn” the weight and bias matrices. This is done through a process called back-propagation which updates each matrix until the loss function is minimized.

*Results*

|  |  |  |
| --- | --- | --- |
| Model | Train Accuracy | CV-Precision |
| ANN (input layer, hidden layer with 10 neurons, hidden layer with 5 neurons, output layer) | 0.591 | 0.51 |

These results were surprising as I expected the model to have a higher training accuracy.

**Model Comparison**

The best performing model ended up being the random forest which had the highest test accuracy out of the four models. Additionally, we can see that the features developed ended being statistically significant and increased the performance of all models when compared to the baseline ranking and logit model.

**Analysis of Confidence Metric**

We can see the recent form metric is an important feature from its statistical significance in the logistic regression model and its importance in the Random Forest model. For further analysis, I re-ran the logistic regression model without the recent form metric. The regression results are shown in Appendix A. Below are the accuracy scores from both models

*Results*

|  |  |
| --- | --- |
| Model | Accuracy |
| Including confidence | 0.7098 |
| Without confidence | 0.7092 |

Without the confidence metric, the logistic regression’s accuracy is not affected. For future studies, more work can be done on the confidence metric and a different functional form can be found. Alternate penalties could include the square root function or a logarithmic function. However, it seems like the “confidence” effect is mostly captured by the ranking predictor. This makes sense because the ranking predictor is updated for each player every tournament.

**Conclusion**

With the growth of big data technologies, tennis players and coaches alike have turned to machine learning to find an edge during each match. This paper attempts to use four of the most common and robust ML models to predict the winner of ATP level matches. In this study, we find that the random forest performed the best on the testing dataset closely followed by the XGBoost and the Logistic Regression. Although this paper was not able to reach the levels that other papers have reached in terms of accuracy and predictive power, it is still a good first step in applying these models.

Based on the models, it seems like the input features are the most important step to increasing the model power as opposed to using a stronger model. Even though Logistic Regression does not have nearly as much flexibility as a Random Forest, its performance was not that far off. Overall, the engineered features and the models were able to beat the baseline set by ranking and the original logistic regression.

**Limitations of Existing Research**

The limitations in this research were mostly due to lack of hardware. My Macbook Pro was not able to handle the tuning and training of the XGBoost and the Artificial Neural Network resulting in potential poor performance. Additionally, the feature space was limited to the data provided within Jeff Sackmann’s dataset.

**Directions for Future Research**

I believe future research should break down each incorrectly predicted match and try to understand why. Although upsets are rampant in tennis and sometimes a player with lower stats might be the better player during the day, work should be done to figure out what features did the statistically inferior player have over the other player. I believe that more granular data would highly increase the power of each model. For example, historical serve accuracy and break point save statistically are a strong indicator of how a player performs under pressure. Although this data is not easily accessible, adding it to the models might increase accuracy by a significant amount.

**References**

Courtesy Jeff Sackman: <https://github.com/JeffSackmann/tennis_slam_pointbypoint>

Ajay, Amudan K. “Tennis Match Prediction Using Machine Learning - Stanford University.” *Tennis Match Prediction Using Machine Learning*, https://cs229.stanford.edu/proj2019aut/data/assignment\_308832\_raw/

Andre, Cornman, et al. “Machine Learning for Professional Tennis Matches.” *CS229*, https://cs229.stanford.edu/proj2017/final-reports/5242116.pdf.

Gollub, Jacob. “Forecasting Serve Performance in Professional Tennis Matches”

Journal of Sports Analytics 7.4 (2021): 223-233.

Klaassen, Franc JGM, and Jan R. Magnus. "Forecasting the winner of a tennis match."

European Journal of Operational Research 148.2 (2003): 257-267.

Kovalchik, Stephanie Ann. "Searching for the GOAT of tennis win prediction." Journal of

Quantitative Analysis in Sports 12.3 (2016): 127-138.

Sipko, Michal, and William Knottenbelt. "Machine learning for the prediction of

professional tennis matches." MEng computing-final year project, Imperial College London (2015).

Lerner, Sven and Badri, Dipika and Kevin Monogue. “DeepTennis: Mid-Match Tennis

Predictions” CS230-Fall 2019, Stanford (2019).

Manage, Sithija. “Neural Network Approach to NFL Position Classification.” *SIAM Undergraduate Research Online*, vol. 15, 2022, <https://doi.org/10.1137/21s1444485>.

Yue JC, Chou EP, Hsieh M-H, Hsiao L-C (2022) A study of forecasting tennis matches via the Glicko model. PLoS ONE 17(4): e0266838. https://doi.org/10.1371/journal.pone.0266838

**Appendix A**

Below are the results for the logistic regression model with and without the recent form metric

*Logistic Regression with the form metric*

**A picture containing table

Description automatically generated**

*Logistic Regression without the form metric*

**A picture containing table

Description automatically generated**