Game, Set, Learn: A Machine Learning Approach to Tennis Match Prediction

Nikhil Shenoy Claire Wang Abhinav Malik

***Abstract*—Tennis is an international sport that people enjoy watching for the variety of style, strategies against different players when playing on different surfaces, and the globe-trotting part of the ATP World Tour. With an increasingly talented group of players at the top of the game, a strong incentive exists, in the betting and purely academic arenas, to predict the outcome of matches. In this project, we seek to use machine learning methods to build a model of a match-up between two players based on statistics about their playing style, and predict the winner of the match. We also aim to construct a model of which statistics are most important in determining the winner of a match. We limit the scope of our model to the ATP World Tour, which is the top level of the mens game.**

I. Introduction

Models for match prediction are useful for a variety of parties. Coaches can use the information to determine where their player is weak and improve those aspects of their game. Even the players themselves can use such a model to assess a matchup with an upcoming opponent, and what their chance is to get to a certain round in the draw. In the betting world, predictions can be used to place bets on favorite players for big cash payoffs. All of these stakeholders can also use the model to analyze what went wrong in matches that were considered upsets, such as Roger Federer's loss to Sergiy Stakohvsky in the second round of Wimbledon 2013, after Federer had won the tournament seven times. Our goal in this project is to create a model that can account for all of these situations and provide useful information to all interested parties.

II. Previous Work

Interestingly, a variety of work has been done on tennis prediction, but those analyses were restricted only to statistical methods. Machine learning methods have only begun to be applied within the last two to three years. We provide a summary of the contributions that have led to the current state of the art.

Madurska [1] used a Hierarchical Markov Model in order to represent a match by taking advantage of the hierarchical scoring in tennis. Since games are composed of points, sets are composed of games, and matches are composed of sets, Madurska could build a Markov Model for each level and tune the the model by specifying the probability of winning one point. Each scoring unit (game, set, match) has a finite set of possible states, and a set of transitions between those states that make the graph sparse. This was the best model until Knottenbelt and Sipko.

Knottenbelt took a step closer to machine learning methods by analyzing matches on a feature basis using his Common Opponent Model [2]. Given the two participating players and a match statistic like the number of aces by each player, Knottenbelt finds the set of common opponents between the two players and generates means for the statistic against each player in the set. He then uses a difference variable between the two resulting machines to generate a single value for the "aces" feature in that match. This method set the stage for match prediction based on feature representations.

Sipko [3] utilized the Common Opponent Model to provide the first significant work in applying machine learning methods to tennis match prediction. He focused on logistic regression as his model, and performed extensive analysis on feature selection, noise removal, and hyper-parameter optimization, among other aspects of his model. We use Sipko's work as the starting point for our project.

III. Methodology

Our experiments seek to answer three questions:

1) Are machine learning models better at predicting match results than a casual watcher using basic rules of thumb?
2) Which machine learning model is the best for prediction?
3) Which are the most important features to consider when predicting a match?

In the first question, we assess whether machine learning models will actually do better than a casual fan using his own instincts. It is very common for a fan to say "this player is ranked higher than his opponent, so I think he will win", and use no other data for his prediction. We turn these statements into baselines, or rules of thumb, to simulate the performance of a "fan classifier". We compare the results against machine learning models to determine whether advanced analysis of tennis data is really useful.

We expect that machine learning methods will do better than a simple baseline, so the next question we consider is which model to choose. Many classifiers have been created over the years, and we compare the support vector machine, neural networks, and logistic regression to try and find the best model that suits this particular problem. We will compare the accuracy of each model on a test data set to determine the best model.

Finally, once we have the best model, we attempt to discover the most important features of a match. Such information is useful to the stakeholders because it simplifies the problem

of predictions into a handful of statistics that they can use to improve their game or maximize their winnings. To do this, we perform an ablation study to analyze the effect of removing certain features from the classification.

We decided to use tennis tournament data from a repository created by Jeff Sackmann [4], which contains all data in ATP matches from 1968 to 2017. In each match, a number of statistics such as the number of aces, double faults, and break points saved, are detailed for each player. These statistics provide ample information about the match that our classifiers can train on.

IV. Data Preprocessing

We used a variety of techniques to prepare our data for input into the algorithm. First we checked for the unique values for each feature in the raw data, and removed examples where the value would be unusable in our feature extraction phase. An example of this would be finding "nan" values in the numerical features; performing any operations on data with this value would result in a value that our classifiers would not be able to process. We considered techniques on how best to replace the value, but we eventually decided that the schemes we had come up with could still alter the outcome of the classification. Thus, we decided to drop the inconsistent examples entirely.

We also restricted the data that we considered to the years 2000-2017. Sackmann's repository contained a wealth of tennis data going back to the inception of the ATP World Tour in 1968, but we decided not to use all of the data because of how the game has evolved since then. There are a number of issues that would have cropped up if we had used the full data set:

- **Difference in technology:** The racquets, shoes, court surface, and ball, among other items, have changed drastically over the years, allowing some players to excel.For example, racquets have allowed players in the last 20 years to hit harder and with more spin, which was not possible before. To avoid this, we limit the years considered so that the issue of varied sports technology is relatively constant.
- **Difference in playing style:** From the year 1968 to around the middle of the 1990s, the playing style for most tennis players was "serve-and-volley". After that period of time, players began to finish more points at the baseline, resulting in the playing style we have today. This difference prioritizes certain statistics, like serves and volleys, more than others such as statistics. We eliminate this particular difference by choosing an era with consistent playing style.

Another decision was made to remove all the Davis Cup matches from the data set. The Davis Cup format requires players to be on a team and represent their country, while countries face each other in a tournament for the final prize. This format has different rules than normal singles play, as each team has a captain who can structure their lineup based on the opposing country.. Since these matches can be arranged differently than a standard tournament draw, we consider it as noise and remove all non-ATP matches.

Finally, we turned the original data set into a binary classification problem. The raw data set had the two players' statistics and the name of the winner, which isn't an easy format to work with. Instead, we labeled the players "Player 1" and "Player 2", and then added a label column such that the value would be 1 if "Player 1" won the match and zero otherwise. By doing this, the question our classifiers seek to answer is "what are the weights of the features that will help Player 1 win the match?", which is an easier problem to solve.

V. Feature Extraction

We used four different methods to extract features for our problem. After all the features are extracted, we create a feature vector with a length equal to the number of features, and have the aforementioned Player 1 win/loss value as our label. We provide a summary of our feature extraction methods here.

A. Symmetry of Features

In the original data, we are provided with two values per statistic in every example. For example, the "aces" statistic is recorded for both Player 1 and Player 2. This representation provides us with specific information for each player, but it also has a drawback; if we exchanged the labels for Player 1 and Player 2, then our classifier could assign different weights to the individual feature and predict a different outcome. To avoid this, we create a single feature with the raw statistics as shown in Equation 1.

$$\text{Feature}_i = \text{Raw}_{1,i} - \text{Raw}_{2,i} \tag{1}$$

where i is the example index, RAW is the raw feature from the original data set, and Feature is our newly generated feature. Previous work by Clark and Dyte [5] have shown that this representation is effective, so we transform all of our raw features in this way.

B. Common Opponent Model

We also use Knottenbelt's Common Opponent Model [2]. In this model, we consider the set of common opponents that both Player 1 and Player 2 have played against. For each of Player 1 and 2, we find the mean of the statistic against each player in the set of common opponents. Finally, we create a difference variable using the two means.

The idea behind this model is that comparing statistics between two players becomes more meaningful when judged against a standard baseline. We get a better idea of the true value of the statistic for each Player since we take into account his performance against a variety of opponents. All of our numerical features were calculated in this way.

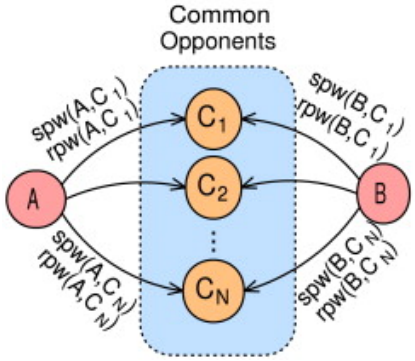

Fig. 1. The Common Opponent Model

C. Derived Features

We created derived features as a function of other features that we have. One noteable example is the Completeness feature. In debates about who the best player of all time is, commentators require that a player have a good serve, good groundstrokes, good returns, and good movement, among other factors. Someone who has all of these factors is considered a "complete" player. We try to simulate completeness in Equation 2 by taking the product of the first serves won, the second serves won, and the fraction of break points won. Unfortunately, our data set did not have information about groundstrokes and court movement, so we limited the definition to just three component features.

$$\text{Completeness}_i = \text{1stWon}_i * \text{2ndWon}_i * \frac{\text{bpSaved}_i}{\text{bpFaced}_i} \quad (2)$$

By multiplying the component features, we generate a derived feature that requires a player to be competent in all areas for the feature to be significant. But if any is lacking, then the feature value will decrease and potentially not have as much impact on the final prediction. We feel that such features provide additional information about the problem based on domain-specific knowledge and can bolster the accuracy of our models.

VI. Experiments

We briefly describe the models that we use, and then detail the results of our experiments.

A. Prediction Procedure

When doing our analyses, we trained using the preprocessed training data from the Common Opponent Model. To tune our models, we evaluated performance on the validation data. Naturally, we could not predict on validation data that was preprocessed with the same methods as the training data, as this would provide the classifier with information about a match that has not happened yet. Our goal is to give our classifier two names, and have it output a binary result about wither Player 1 will win. To simulate a feature vector, we keep track of the average values of each feature for each player in a dictionary using *only the training data*. When we get an example from the validation or test set, we look up the feature value for each player in our dictionary and create a difference variable from the two values, which is then put into the feature vector. In cases where the player has not been encountered before, the averages for the features are set to 0 since we have no other information about the player. With this averaging procedure, we provide the classifier with the means of all features for each player, which gives an estimate of each player's ability at the time of the match. We also ensure that the classifier does not see any future data and that the result is truly a prediction.

B. Models

1) Support Vector Machines: We chose to use a Support Vector Machine (SVM) for classification for two main reasons; first, no previous published work has utilized the SVM for this problem, and we saw an opportunity to examine how effective this classifier is. Also, the SVM does not encounter local minima during the calculation, which provides an advantage over classifiers like neural networks. We used the Soft SVM formulation in Equation 3 to build the classifier, where w is the weight vector, y_i is the label of example i, x_i is the i-th example, and C is a hyperparameter that scales the penalty of performing misclassifications.

$$\min_w \frac{1}{2} w^T w + C \sum_i \max(0, 1 - y_i w^T x_i) \quad (3)$$

To find the optimal hyperparameters for the SVM, we used a grid search over three kernels (linear, polynomial, and the radial basis function) and the scaling factor C. We trained on our training set and tested on the validation set, giving the results in Table I.

Kernel	C	Accuracy
Linear	0.25	.686
Linear	0.5	.708
Linear	1.0	.726
Linear	10	.717
Polynomial	0.25	.691
Polynomial	0.5	.623
Polynomial	1.0	.663
Polynomial	10	.719
RBF	0.25	.710
RBF	0.5	.734
RBF	1.0	.729
RBF	10	.712

TABLE I
Results from Tuning SVM Hyperparameters

From this grid search experiment, we found that the RBF Kernel with $C = 0.5$ was the optimal pair of hyperparameters for our model.

2) Logistic Regression: We decided to use logistic regression as our third model because that was what Sipko used in his initial paper, and we wanted to see whether we could replicate his results.

$$\sigma(z) = \frac{1}{1 + e^{-(w^T x + b)}} \quad (4)$$

One of the big advantages in using logistic regression is the ability to determine the output of the classifier as a probability; the sigmoid function, shown in Equation 4, squashes all values it encounters between 0 and 1, and then uses a tuneable threshold value to make decisions. Since the values of each feature can vary greatly, such as 113 aces John Isner hit in his famous Wimbledon 2010 match versus a very low first percentage that any player may encounter, the value of the prediction can fluctuate greatly and cause us to impose additional conditions on the problem in order to make a decision. Logistic regression gives us a neat way to avoid this issue.

3) Multi-Layer Perceptron: A multi-layer perceptron neural network is chosen as one of the three machine learning models because we learned about perceptron in class, and different from online model, multi-layer perceptron has a non-linear activation layer that is more suited for high dimensionality and linearly inseparable features. The multi-layer perceptron we used is implemented in sklearn, and has one input layer, one hidden layer with a hundred units and one output layer. We used ReLU as the activation function, and Adam as the solver.

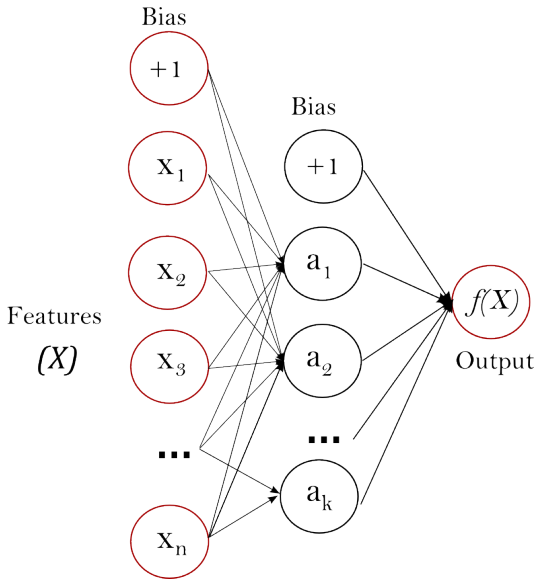

Fig. 2. Multi-Layer Perceptron Model

C. Comparison to Baselines

In Figure 3, we can see the difference in performance between the machine learning methods and the baselines. We see that the classifiers do beat out the baselines in all cases, but by not as much of a margin as we expected. The SVM classifier did the best, recording an improvement of about 7% and an accuracy around 73%. While this is a good improvement, the model could still be improved to get higher accuracy. On the other hand, the performance from the baselines was expected; if a coin-toss prediction gives about 50% accuracy on matches, then we expect that a baseline with one type of comparison to do slightly better, and this was reflected in our results. Our models performed well, but not significantly better than the best baseline. Regardless, this is an important step in our project because it justifies the use and tuning of machine learning methods to get better performance.

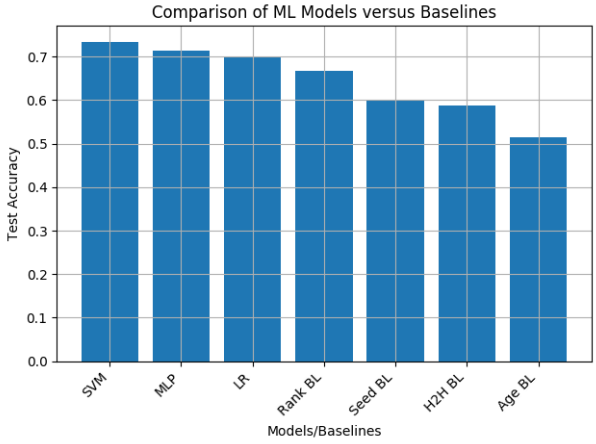

Fig. 3. Comparison of Models and Baselines

D. Comparison of Models

Figure 3 also shows us the performance between the three classifiers we trained. On the whole, the classifiers had approximately the same performance, with a difference in test accuracy of about 5%. We had originally expected the multi-layer perceptron to perform the best due to its property of being able to approximate any continuous function (using certain conditions) up to some tolerance ϵ using the three-layer architecture that we used. There are a variety of reasons for why such a result may occur. While we did do parameter tuning to get the best configuration for each classifier, it may be that in this case the SVM's ability to separate the data with a thicker margin allowed for less misclassifications than the multi-layer perceptron did for this particular data set. An avenue for future exploration would be to develop the MLP further using deep learning methods to see whether the accuracy will increase. In the case of logistic regression, we would need to invest additional effort in optimizing the classification threshold as well as feature extraction to improve the model. In all cases, the lower accuracy could be due to the split between the training, validation, and test data. Since the training data consists only of matches from 2000-2008, any changes in player performance after that time could significantly change the probability of him winning a match, and that information is not available to the classifer. This is a realistic challenge, as there is no way to know in advance how a player's performance will fluctuate throughout the year. Additional considerations have to be made for this effect in order to improve the model. In the end, our results in this step singled out SVM as the best classifier for our project, but also showed that other models are almost equally performant.

E. Ablation Study

To find out what the most important features in determining match outcome are, we perform an ablation study. In such a study, we first get the accuracy of the classifier with all features included. Then, we remove features one at a time, to examine the contribution each feature has to the final accuracy of the model. We performed this technique on all of our models, with results from the ablation study on our SVM shown in Figure 4.

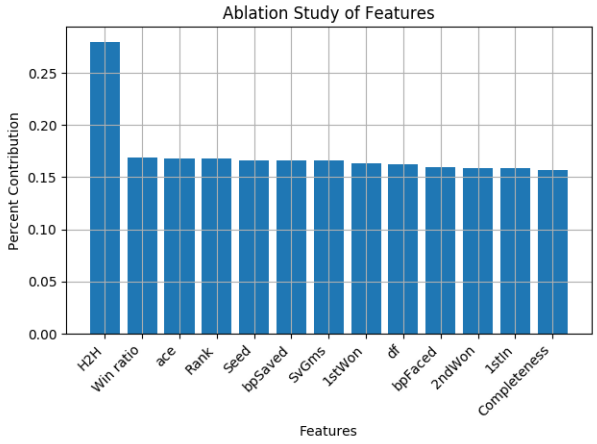

Fig. 4. Ablation Study with SVM

The results from this study are interesting because of the signifiance placed on certain features. We observed that the Head-to-Head feature had the most significant contribution to the test accuracy, boosting it by about 11%. The implication of this is that the history of the pair's past matches outweigh any of the statistics gathered during the current match. The next two most significant features were the win ratio and the aces. Intuitively, the win ratio, which indicates how much a player was won against all players and not just his current opponent, would be a fairly significant measure of his chances to win. But the more interesting trend is that the remainder of the features that we used either did not contribute as much in comparison to Head-to-Head, and that those contributions were nearly uniform. From this, a conclusion is that our model provides no information about which match statistic, such as a certain first serve percentage or break point percentage, a player should prioritize in order to win, implying that so long as the player can win, he can play whichever way he likes. The most notable of the remaining features is completeness, which is one of the derived features that we created to simulate a player's ability in multiple aspects of the game. The implication from the study is that a player training to be good at each type of shot is actually not a good indicator of success, allowing one to conclude that that a player should only focus on improving certain shots, perhaps those he has a natural talent for, to increase his chances of winning. This is a very interesting result because it invalidates many of the discussions that commentators have about how a player should rise to the top of the game and maintain dominance. We had expected the ablation study to show that each of the match statistics contributed some positive percentage to the test accuracy and suggest what areas a player should focus on improving. The final results, however, was that the most important feature in determining match outcome is the past Head-to-Head history. As a way of providing useful information to potential stakeholders in the result of this project (players, coaches, bettors), we summarize the relevance of the top 5 features in Table II

Feature	Percent Contribution
Head-to-Head	.28
Win Ratio	.169
Aces	.168
Rank	.168
Seed	.166

TABLE II
TOP 5 RELEVANT FEATURES FOR WINNING A MATCH

VII. FUTURE WORK

There are endless avenues of future work for this topic. One idea we would like to explore is to change the way the test feature vectors are generated. Currently, we average each feature across all previous years, with no regard to relevance. We would improve uponthe averaging scheme by using a weighted average which includes time discounting. In other words, we discount the value of a feature based on how many years have passed since that value was measured. The idea here is to emphasize a player's recent performance over performance in years past, with the expectation that recent performance gives a better idea of a player's winning chances than if his entire career is weighted equally.

Another very interesting avenue would be to detect examples of match fixing. In recent years, the ATP has found and prosecuted cases of players who purposely lose matches in order to get a payment from a hidden benefactor that is beyond what the tournament would pay. Such matches are very likely to be included in our data, and can add significant noise to our model. Detecting and removing such matches would be useful in order to create a more accurate model of a match.

Finally, we would like to compare the performance of our classifiers against the Hierarchical Markov Models proposed in the literature. We limited the scope of our project to just analyzing machine learning methods, but comparing against previous statistical models would provide some insight into better ways to model the game of tennis.

VIII. CONCLUSION

In this project, our goal was to take the real-world problem of predicting the winner of tennis matches and improve upon the mathematical models suggested in the past by applying machine learning methods instead. Through our experiments, we were able to confirm that the use of machine learning methods is indeed justified and does provide performance benefits over a casual fan's predictions. We performed feature extraction of match data that we felt best represented a player's ability, and compared the results of each classifer to find the

best model. Finally, we conducted analyses that assess which features carry the most weight in determining match outcome. Our results are consistent with the literature in this area, and we believe that our work advances the application of machine learning methods to tennis prediction in a significant way. For the stakeholders in this research, we provide information that will help them maximize their objectives, whether it be winning more matches or getting a bigger payoff from a bet. We believe that the results we found during the course of our project would help those interested in predicting the outcome of tennis matches, and provide valuable insight for those looking to extend this work.

IX. Feature List

This is the full list of features that we used:

- **Head-to-Head:** Percentage of matches won from all meetings between two players
- **Win Ratio:** Player's win percentage against all opponents
- **Ace:** The number of aces a player hit during the match
- **Rank:** The player's ATP World Tour rank at the time of the match
- **Seed:** The player's seed at the tournament where the match took place
- **SvGms:** The number of service games that a player has won
- **1stWon:** The number of points won by a player when the point was started with a first serve.
- **1stIn:** The number of non-fault first serves by a player
- **2ndWon:** The number of points won by a player when the point was started with a second serve.
- **df:** The number of double faults committed by a player during the match
- **bpSaved:** The number of break points saved
- **bpFaced:** The number of break points that player faced
- **Completeness:** derived metric assessing player's skill at variety of statistics. See Data Preprocessing section

Note

This paper has the most update version of our results. The poster has an old version of our results.

Acknowledgments

We would like to thank Professor Roth and the course staff for their help with this project. Their feedback was invaluable in creating the final product.

We would also like to thank Jeff Sackmann, who compiled the excellent database of tennis match statistics that we used to train our models. His data made this project possible.

References

[1] A. M. Madurska, "A set-by-set analyssis method for predicting the outcome of professional singles tennis matches," Master's thesis, Imperial College London, June 2012.

[2] W. J. Knottenbelt, D. Spanias, and A. M. Madurska, "A common-opponent stochastic model for predicting the outcome of professional tennis matches," *Computers and Mathematics with Applications*, vol. 64, pp. 3820–3827, 2012.

[3] M. Sipko, "Machine learning for the prediction of professional tennis matches," Master's thesis, Imperial College London, June 2015.

[4] J. Sackmann, "Tennis atp." https://github.com/JeffSackmann/tennis_atp, 2018.

[5] S. Clarke and D. Dyte, "Using official ratings to simulate major tennis tournaments," *International Transactions in operational Research*, vol. 7, pp. 585–594, 2000.

[6] A. Somboonphokkaphan, S. Phimoltares, and C. Lursinsap, "Tennis winner prediction based on time-series history with neural modeling," *International MultiConference of Engineers and Computer Scientists*, vol. 1, March 2009.

[7] J. G. M. Klaassen and J. R. Magnus, "Are points in tennis independent and identically distributed? evidence from a dynamic binary panel data model," *Journal of the American Statistical Association*, vol. 96, pp. 500–509, June 2001.

[8] S. Ma, C.Liu, and Y. Tan, "Winning matches in grand slam men's singles: an analysis of player performance-related variables from 1991 to 2008," *Journal of Sports Sciences*, vol. 31, pp. 1147–1155, 2013.