ML Final Project Paper

Pranet Allu

Oliver Peralta

Pieter Alley

Abstract

Predicting tennis matches requires a deep understanding of several factors in tennis. There is also a lot of data to be recorded from tennis matches that can be used to predict the outcome of these matches. Through machine learning, we have decided to create a model that attempts to predict the outcome of tennis matches with data used from previous matches from 2021 to 2023. With this data, we are able to use our predictions towards responsible gambling on the outcome of these tennis matches.

Background

To start off, the main objective of the project is to create a model that is used to predict the winner between two tennis players based on the player's profile and their performance in previous matches. The predictions are based on the ATP datasets from the year 2021 to 2023 since the previous years lacked crucial information, rendering them unsuitable for training and testing. The main features that are essential for the training of the model would include the player's hand, age, height, number of aces, rank, rank points, number of wins / losses, number of matches played, and total number of minutes played.

Data Preprocessing

For the data preprocessing, we decided on creating profiles for each of the players. We used a dataset we found on a public GitHub Repository by JeffSackman, and we decided to simplify the model by only using the values from 2021 to 2023. This is because with time, players will retire, and we did not want to make profiles of players that were already retired because we want to predict future games and that data would not help us. For each profile, each player has their name, hand they play with, height, country, age, matches, total aces, total double faults, their rank, rank points, number of wins, and total minutes played. To get into more detail. Double faults are the amount of times that a point is lost by missing both serves, an ace is when you win on the serve, and rank points are the points that players accumulate to raise their rank. That was the data that we decided to go along with since we believed it was data that could really help us determine how players would play against each other.

Architecture

The architecture here is based on the logistic regression model that is usually known to be used for binary classification purposes, which means that it can predict an outcome out of two possible scenarios. Features of both Player 1 and Player 2 would pass in the array which then predicts a value of either 1 (Player 1 is the winner) or 2 (Player 2 is the winner).

Training

To train the model, we started to go through the matches that each player played and kept track of who won in each match. We stored this information through an array where each index was a match with a player 1 and player 2, and each player had their attributes added to the value of the array. The values in the array also had a numerical value 1 and 2 to determine whether player 1 or 2 won. Along with that, we also added a win rate and BMI to each of the players in this process. Once we had the winners of the matches, we had the X and Y arrays to be able to make the training and test data. After, we used a logistic regression model to fit our data. We used logistic regression because it was the best model to use for binary classification, and it is pretty clear that we are just trying to see if a player wins or loses. There are no ties in tennis.

Steps Taken to Improve the Model

Feature engineering played a vital role in this case since some features were directly passed in while formulating others as indices in order to better predict an accurate outcome. Moreover, the training and testing data were split into a 80-20 ratio so that the model can learn about patterns from larger dataset while making sure that it can generalize well to unseen / testing data.

To improve the efficiency of the model, three parameters were passed in which include penalty, max_iter, and random_state. Penalty was set to "l2" to prevent overfitting in order to generalize well to unseen data and impose a ceiling on the loss function. In addition, a "max_iter" of 100 helps the model narrow down and converge to a solution. Lastly, a "random_state" of 42 involves shuffling of data during the training in order to ensure that the model is performing consistently good on multiple iterations.

Results

Two accuracy indicators were being used to measure the performance of the model which include the "accuacy_score" and "crossValScores". The "accuracy_score" compares the predicted and true labels / values with each other in order to output the number of predictions that the model has gotten right. In this case, the score was nearly 69%. On the other hand, “crossValScores” performs a cross validation test on the model over 10 folds and records the accuracy for each one of them. Finally, the average is being calculated to get the final accuracy score which is nearly 67%. As a matter of fact, a 67% accuracy on test data and 69% cross validation accuracy indicates there is no significant gap between them and also means that the model is not overfitting. Moreover, it generalizes well to unseen / testing data and is less sensitive to outliers and noisy data.

While other machine learning models in different fields have higher accuracies, for example, above 85%, here in the tennis realm, our percentages are good considering there are also a lot of factors that are unseen. This is due to a high irreducible error that is derived from various factors leading up to and during a tennis match. There are moments where players sometimes play extremely well or extremely bad, players go through mental blocks, have outbursts of anger, are being cheered against or for, and more. Nonetheless, it is necessary to take into account the amount of irreducible error there is in tennis and how well the model does against it.

Practical Applications

With this model, it can be used to predict future tennis matches. By predicting future tennis matches, one can bet on the outcome of matches, in a legal and responsible setting. Bettors can use this machine learning model to help them place their bets, with the hope of making profit. While tennis players also have their own coaches, the model can be used to help prepare against players they may lose to. Along with that, the same model can be used to bet on other sports. Depending on the sport, it can be used on table tennis or badminton, which are sports where there are usually only two people playing against each other, as opposed to other more complicated sports like basketball and football that have over 10 players on the field at a time and are playing at home or away.

Conclusion

The current Logistic model serves as a binary classification approach towards predicting the winner between two players. Indeed, it accomplishes this by basing its predictions on the player's profile, their statistics, and the outcome of their previous matches. Although the model yields an accuracy score of nearly 69%, it can further be refined with the inclusion of features / characteristics pertaining to tennis and the optimization of parameters within the model.

References

- https://sportsbook.draftkings.com/help/how-to-bet/tennis-betting-guide
- https://sportsbook.draftkings.com/sports/tennis
- https://tennistonic.com/
- https://libstore.ugent.be/fulltxt/RUG01/002/945/727/RUG01-002945727_2021_0001_AC.pdf
- https://github.com/JeffSackmann/tennis_atp