Tennis Match Prediction using Machine Learning

Ajay Krishna Amudan

ajkrish@Stanford.edu

Problem Statement

Tennis is a highly unpredictable sport ripe with upsets abound. The goal of this problem statement is to predict the outcome of a match before it starts and use it for betting. We predict the outcome of men's singles professional matches from the time period of 1997-2019. We then want to use the confidence of our predictions to decide which matches to bet on and how much to bet on each such match to maximize our profit.

Dataset

The dataset that is used for extracting features is from Jeff Sackmann's Github repo which has the set of matches played in ATP level tennis, challengers and futures – specifically we use the period from 1997 to 2019 because the matches from this period have extensive match level statistics available. We include challenger level matches and qualifiers from 2010 to ensure that we have more data on younger players to negate any negative recency bias. We also use http://tennis-data.co.uk/ for betting data.

Data Sanitation

I ignored all matches pre 1997 because they did not contain as many features – there were around 25 features missing. I also ignored all matches that had even a single column with unexpected data – because there is a high possibility of it being noise. Furthermore I ran several sanity checks on each feature to ensure that we have expected values for each column. Once clean data was generated, I also ran aggregated sanity checks to ensure that we got expected results when we ran known queries on this data – as an example – I counted the number of aces served overall by Roger Federer and it came out to 10995 – which is exactly what is expected.

Feature Extraction

I initially started out with a number of in-match statistics which were averaged out throughout the career of the player till before he played the match and then the difference in value of this statistic for the two players was chosen as the feature. Specifically the initial model had 9 features which were all extracted directly from the given dataset – as the last 18 columns of data after averaging and taking difference.

The second set of features I added to the model were out-of-match statistics which were again present as columns in the dataset – this did not require any averaging – for example difference in heights, ranks and ranking points. This was a set of 6 features.

The third set of features were more derivative in nature and includes the following:

- Surface level historical form
- Recent form of players
- Serve vs Returns stats contrast
- Head to Head
- Fatigue – recent number of matches
- Seasonal form – using form during specific months over previous years
- Common Opponent Features – how well each player has performed against common opponents

Results

Logistic Regression – L2 – Accuracy – 71%

Logistic Regression – L1 – Accuracy – 70%

Random Forests – Accuracy – 68%

SVM – SVC – 65%

Logistic Regression model generalized very well as the training and test accuracy using 23 fold validation – I chose 1 year for prediction at a time – were both almost the same – around 70-75% prediction

Conclusion and Future Work

Logistic Regression was the only feasible model I could run with the resources but surprisingly it showed pretty high accuracy and generalization. The benchmark that I set out to beat was the accuracy attained by predicting every match to be won by the higher ranked player – this was 66%. The logistic regression beat this simple model by 5% is quite an achievement. Secondly logistic regression was very helpful because of the low turnaround time to decide which features to add and which ones were not very useful. Because the final set of features were close to 50, this operation was not prohibitively expensive. The quality of the features – I believe – is the most important part of this project. Once the right set of features are chosen – with more powerful models like neural networks I would expect higher accuracy.

Before publishing the report, I will also show how to use these predictions to bet effectively. As future improvements to the model itself, I intend to use more powerful machines for a longer period to get better results.

References

1. http://cs229.stanford.edu/proj2017/final-reports/5242116.pdf
2. http://cs229.stanford.edu/proj2017/final-reports/5243744.pdf
3. https://www.doc.ic.ac.uk/teaching/distinguished-projects/2015/m.sipko.pdf
4. A. Somboonphokkaphan, S. Phimoltares, and C. Lursinsap. Tennis Winner Prediction based on Time- Series History with Neural Modeling.IMECS 2009: International Multi-Conference of Engineers and Computer Scientists, Vols I and II, I:127132, 2009.
5. https://github.com/JeffSackmann/tennis%20atp%20,%20Jeff%20Sackmann
6. http://www.tennis-data.co.uk/alldata.php

https://www.youtube.com/watch?v=PD-0k1kBWJA&feature=youtu.be