Tennis Match Winner Predictions

1: Abanoub Abdelmalak (abanoub_abdelmalak@gatech.edu) - OMSCS
2: Sheikh Jalaluddin Mohammad (jalal@gatech.edu) - OMSA
3: Michael S Rivera (mrivera63@gatech.edu) - OMSA
4: Mohamad A Elsayed (melsayed31@gatech.edu) - OMSCS

1 Introduction

Tennis is one of the most popular sports in the world and its popularity has aroused a great interest in the betting market in the recent years. However, in tennis, unlike other popular sports as soccer, analytics is still in a very early stage. This fact presents a great opportunity to make the difference in the betting market especially that the sports betting market size is experiencing a tremendous growth of $167.66 billion by 2029, estimating a CAGR (Compound Annual Growth Rate) of 10.26% [1].

2 Problem Definition

As tennis is a 1v1 sport with no time-frame, it is blessed with a massive amount of data across the years that can be utilized for different applications. while there has been a lot of effort into predicting and analyzing the tennis match on a micro level (e.g. Analyzing motion through matches to predict what kind of motion is best predictor for winning matches), the effort looking at the macro analysis has been limited mainly to interesting statistical data-points, with no cohesive vision to collect all this information and present it in a way that provides deeper insights. These insights can be used for several applications, such as building a more cohesive betting strategy while suggesting the best features that greatly impact the match results.

Utilizing the literature survey and our domain knowledge, we create a web application that provides an easy and efficient methodology for predicting the winner for professional men tennis matches. The application will simultaneously show important parameters and statistics for comparison. This implies the development of a machine learning model that allows us to predict the winners of the matches, which can potentially open the way for improved betting odds.[1] [2].

3 Literature Survey

[3] Diverse techniques have been historically used to analyze tennis matches, this includes statistically driven approaches focusing as well as Markov-chain methodology approaches. data obtained during a match has also been used [4]. Lately, machine learning driven approaches have become more used. straight-forward regression based algorithms, and more complex neural network approach were used, while some used constantly updating probabilistic model[4]. Paper [5] uses probability models on a combination of players past performance, a players physical characteristics, and match characteristics. Models are evaluated on Brier scores. This provides us with a baseline model. We can try to improve upon this model by incorporating various elo ratings & rankings and human judgements. Paper [6] shows how data combined with judgement can significantly improve the accuracy of models predicting outcome of tennis matches. We will explore adding parameters not used in this paper. Paper [7] and [8] utilize the fact that rankings are derived from a points rating and uses these difference in ATP rankings of players to predict the winner of a match and subsequently the tournament. We Plan to add other parameters to this and also explore using elo ratings and rankings. Paper [9] offers a technique of considering non-deterministic behaviours by adding a probabilistic component to the player activities, and applying that on the Markov Decision Processes. While interesting, it's more focused on play-by-play predictions, rather than an overall match winning, which is not the scope of implementation. Paper [10] offers a strategy of real-time betting based on match results. while we don't intend to consider real-time betting, this can be a good seed for future platform development. Paper [4] and [11] show how apart from data obtained before a match, data such as serving statistics obtained during a match as it progresses can be used to predict the outcome. Access to data during a match is not accessible to us as of now hence we wont be using it in our project. Paper [2] combines statistics of the two players to find the probability of each one of them winning their services and responses. Then those odds are fed into a Markov Chain model to predict the winner of the match. The whole model is based on how good the players are to serve/respond, leaving other crucial aspects of the game out of analysis, however, some techniques for

[1] Source: Data Bridge Market Research

feature design could be useful for our analysis. Papers [2] and [12] use multiple machine learning models (logistic regression, logistic regression with interaction features, and ANN are some of them) on a player attribute dataset. [2] found a profitable betting strategy and [12] founxd no evidence for the case of inefficiencies in the betting market. New features could be included in order to improve the results seen in both cases.

4 Proposed Method

4.1 Intuition

Currently, there is no holistic approach to provide tennis matches data and prediction (i.e, visualizations are mostly isolated visualizations) without offering a simple way of combining all of this into a unified environment. Additionally, any predictions are usually locked to betting platforms, which don't necessarily offer a visualization aspect. This combination of visualizing analytic, and adding layer of prediction should provide a clear added value to the existing state of the art.

4.2 Approaches

We are developing an interactive web platform that allow the user to select the two players that will compete against each other and show the winning percentages accordingly. The technologies used are:

- Data-sets are analyzed using publicly available data. as Such, multiple features data-sets are generated to properly study the impact of features on Match winning performance
- A PostgreSQL database stored in a Docker image is used for pulling the necessary data. This data provided an easy and accessible way of analyzing the data, generating insights, and performing feature engineering. Afterwards, final data-sets are stored as simple CSV files, and simply imported within the app.
- Web platform visualization using plotly and dash. The Web-app is hosted on the render platform, and is directly available at https://tennis-app-v2.onrender.com/
- Interactive plots created that help, not only providing a win/lose answer also provide a strong overview about historic players performance
- Machine learning Training and deployment is done through the use of sklearn open source library. Machine Learning Evaluation is done separately through an extensive and robust methodology. afterwards, it is stored and deployed directly within the application, where it is used for prediction.

In the next Experiments section, we will be discussing the flow that we have used in the code development along with the iteration we went through to apply the corresponding data filtering and get the final prediction results.

5 Experiments/Evaluation

The section is divided into 4 main sections: data-set selection, data engineering and analysis, machine learning model for prediction, and finally our visualization platform. Dataset Selection section, where we go through the different databases examined and how did we collect our final in use database schema. Data Analysis & Feature Egnineering section where we discuss the data features we have examined and used and how did that affect the final quality of the machine learning model prediction and applying the necessary filtration that fits the required objective. Machine Learning Algorithms Tuning And Analysis section where we examine the Machine learning model accuracy. Final Machine Learning Model Review section, where we review final Machine learning results, and our final selection. Visualization and Web App methodology section where we show our experimentation for the techniques that are being used for interactive visualization of the data features and match winner prediction.

5.1 Dataset Selection

- **Internet Scrapping:** At this point we tried gathering data from different source from the internet. This helped finding different rich sources of data that are much better organized that leads to the next two trials.
- **Docker PostgreSQL:** Finding this database helped a lot building an interaction around it for a database module that retrieve all the database table in pandas dataframe fashion. The only drawback was finding some gaps in the data that leads to finally deciding on using it in accordance with the below methodology as well.

- **Tennis API:** The drawbacks is the premium access has only a month of free trial that will act as an extra cost.

5.2 Data Analysis & Feature Engineering

After settling on the defined data source, we started looking into the data in a more detailed fashion trying to find out what are the best/correct features that we can use. This is what we will discuss in the next points. The term "best" should be associated by a numerical metrics that can tell us how effective a specific feature on the final machine learning model accuracy.

5.2.1 Match Database Processing

The matches database by default stores the information of who won or lost the specific match selected. Accordingly, we scrambled winner and loser players to player 1 and player 2. This processing prevents biasing the model based on player order. Additionally, statistical calculations relating player1 and player2 data were done to ensure model is order agnostic.

5.2.2 Data Cleaning Methodology

Initially, 182964 matches were pulled from the database. We applied filtering to the matches based on:

- Active players: Both players must be currently active.
- Not sufficient data: Any match that does not have enough data features that we are need already for prediction has been filtered out.

Final count of data-points used for Calibration and testing is 16049 data points.

5.2.3 Data Features Summary

The main Objective of the Machine learning model is to predict the winner of a match between two players under certain conditions. To decide this, we extensively reviewed the data-set, aided by the existing literature survey, and the domain knowledge of the team. We've observed that relevant data-points using simple entropy & information gain to define the highest correlation factors. And finally, we've settled on the collection of data data-points that can be categorized as follows:

- Match-based Features (match format, surface, indoor/outdoor)
- Unique Player Features (height, age, One backhand or two)
- Current Ranking (official and ELO ranking)
- Player Win % (General win %, win % under same match format, on same tournament surface, and in this format)

in Total, final features data-set included 12 Features.

5.3 Machine Learning Algorithms Tuning And Analysis

in this section, we detail the machine learning models that were tried and tested to assess the best performing model. We review the types of Models that were assessed, the Hyper-parameter tuning methodology, and the scoring methodology

5.3.1 Types of Models assessed

We have settled on approaching this as a purely classification problem. As such, we've tried multiple types of classification models, with varying levels of complexity. Initially, we reviewed a simple Decision Tree model as a starting point, with varying degrees of hyper-parameters tuning(such as Tree-depth, leaf size), we've then added a degree of complexity by using a Boosted Decision Trees, such as Adaboost and GradientBoost. Alternatively, we also considered different approaches, such as K-nearest Neighbor algorithm, and finally we tested using a more complex Neural Network algorithm.

5.3.2 Scoring Methodology

We settled on F1-score as our scoring methodology for our Machine Learning model, as it provides a good weight balance between in-sample and out-of-sample scoring. The scoring methodology was applied on all model types, and both the training set and the test-set scoring. it was also used to qualify the quality of our hyper-parameter tuning

5.3.3 Hyper-parameter tuning

A robust methodology was performed to ensure Hyper-parameter tuning. This methodology focuses on both extracting the best possible parameters data-set, as well as test and judge the model quality, and whether it over-fits/under-fits the data.

- Split And Scale: Data-set is split into 70% for Training, and 30% for validation. If needed for a certain Model type, data is scaled using a standard scaler
- Grid Search: Based on the type of model, set of parameters are targeted for tuning(e.g. No of Estimators for a boosted learners, number of stages for a Neural Network model). Then grid-search is performed across the targeted parameters to arrive at the best parameters combination
- Generate Validation Curves: as a secondary to grid-search, generate validation curves of targeted hyper-parameters, to analyze and review it's performance, and recognize regions where parameter starts over-fitting to existing data5.3.3
- Review Features Importance for Final Model: to qualify our model in relation to the input data, feature importance is calculated and impact of each feature is understood and reviewed for any outliers or inconsistencies with expectations
- Generate Learning Curve for Final Model: to qualify how our data input scales with model learning, Learning curves are generated to measure how well the model is improving due to additional data points.5.3.3

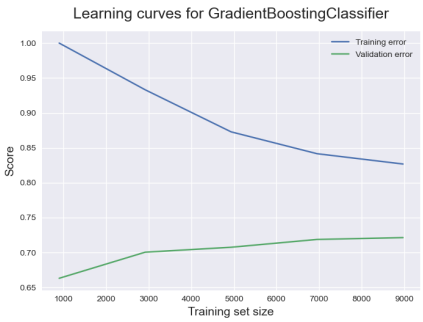


Figure 1: Learning Curve for Gradient Boost Classifier

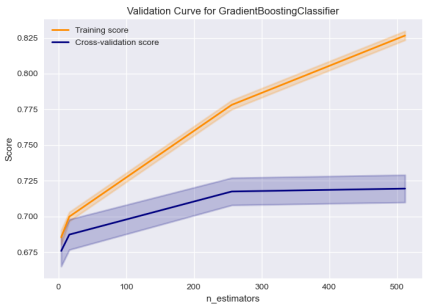


Figure 2: Validation Curve for Gradient Boost Classifier

5.4 Final Machine Learning Model Review

The three features data-set, in addition to the Multiple machine learning models mentioned, were analyzed and compared. Below table highlights the Training and Testing accuracy of results. Also, the plot highlights the confidence interval when using each Features dataset, across all model types.

ML Model	Train F1 Score	Test F1 Score
Decision Tree	0.712383	0.676517
Gradient Boosting	0.812047	0.733461
Neural Network	0.712053	0.705958
Ada Boost	1.0	0.730786
K Nearest Neighbors	0.719273	0.684189

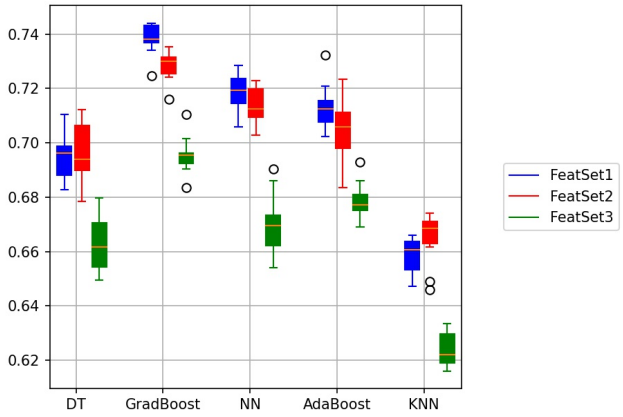


Figure 3: ML Model F1 Score comparison for different feature sets selected

- **Feature Set 1:** 'surface', 'best_of', 'indoor', 'elo_rating', 'rank', 'age', 'height', 'surface_win_p', 'indoor_p', 'best_of_win_p', 'matches_won_p', 'backhand'
- **Feature Set 2:** 'surface', 'best_of', 'indoor', 'elo_rating', 'rank', 'surface_win_p', 'indoor_p', 'best_of_win_p', 'matches_won_p'
- **Feature Set 3:** 'surface', 'best_of', 'indoor', 'elo_rating', 'rank', 'age', 'height'

Consequently. it is decided that Gradient boosting, using Feature Set 1 provides the best balance of accuracy, performance, and model fitness.

5.5 Visualization and Web App

Our goal was to provide a web app where user will have the option to select two players and variables for the ML model like, type of surface, indoor or outdoor and best of three or best of five, output would be the match winner based on our Machine Learning models results, as well as a set of relevant statistics and graphics for each player, like a radar chart to show their overall winning percentage on different type of court, line graph to show season-wise tournaments won, current ranking, to make the app appealing we have added visuals like photograph of selected players, flags of the selected players country, a map showing the country selected players belong to.

5.5.1 App Framework

For creating the visualization web app we have used the Dash framework, it is a python framework created by plotly for creating interactive web applications. Dash is written on the top of Flask, Plotly.js and React.js. Dash is open source and the application build using this framework can be viewed on the web browser.

5.5.2 Visualization

For creating the visualizations within the app we have used "plotly" which is an interactive, open-source plotting library for python that supports over 40 unique chart types covering a wide range of statistical, financial, geographic, scientific, and 3-dimensional use-cases.

5.5.3 Web App Deployment

The app was deployed on Render, it is a unified cloud to build and run apps and websites with free TLS certificates, a global CDN, DDoS protection, private networks, and auto deploys from Git.

5.5.4 Data for Web App

Our primary data source for visualization was a Docker-PostgreSQL database from which we extracted the relevant data to our visualization as a csv file. Player ID was the common identifier across charts and the ML model. since storing image of all players and flags would have taken a lot of space we complemented the data from csv file with API provided by api-tennis.com for realtime rankings and statsr, for flags of countries we have used the API provided by countryflagsapi.com. Both the APIs used were either free versions or free.

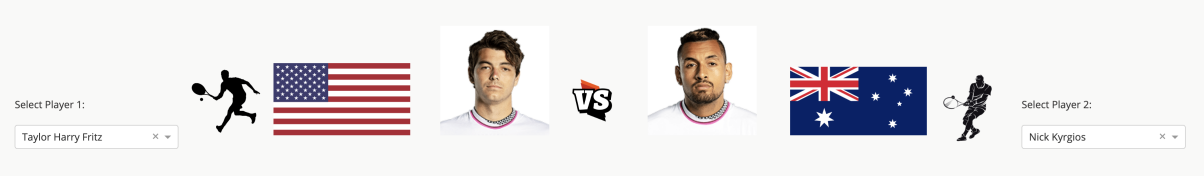


Figure 4: Interactive Player Selection

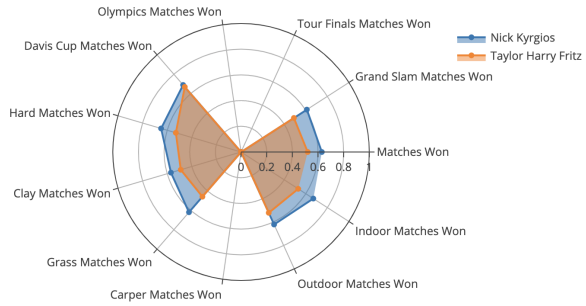


Figure 5: Radar Chart

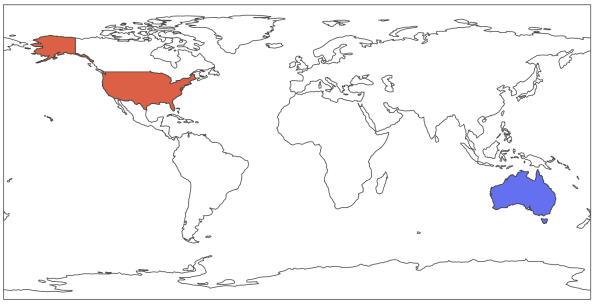


Figure 6: Choropleth map

Interactive visualizations added to the web app include a Radar Chart which is a graphical method of displaying multivariate data in the form of a two-dimensional chart of three or more quantitative variables represented on axes starting from the same point. The relative position and angle of the axes is typically uninformative however the points on the axis show comparison of one variable compared to other which in our case is ratio of matches won on various surfaces and tournaments. This gives us a quick glimpse of relative performance of one player over another.

6 Conclusions and discussion

This project aimed at building an algorithm for Match prediction & Analytics using existing datasets. A flow was defined to collect input data-sets via a docker+postGreSQL, select critical features, train and evaluate models using a robust Modeling criteria, Build an analytics and visualization interface using Dash & plotly, and publish the App via Render.

Both Machine Learning & Data analytics played critical role in this project, a Model with more than 80% accuracy was developed, and successfully linked to a web-application interface with various analytics to provide a holistic analysis between player-pairs and build an intuitive methodology of predicting winners.

Potential Future extensions can include:

1. Build a mechanism for live-data feeding
2. build a mechanism to continuously train and fit model results to recent data, instead of the Build-once-then-predict current methodology
3. use model Prediction confidence as a mechanism of providing optimum betting odds on match, and compare that with existing market betting odds to uncover efficiencies.

All team members have contributed a similar amount of effort.

References

[1] S. Lyocsa and T. Vyrost, "To bet or not to bet: a reality check for tennis betting market efficiency," *Applied Economics*, vol. 50, pp. 1–22, 10 2017. [Online]. Available: https://doi.org/10.1080/00036846.2017.1394973

[2] M. Sipko, "Machine learning for the prediction of professional tennis matches," *Imperial College London, MEng Computing Final year project*, 2015. [Online]. Available: https://www.doc.ic.ac.uk/teaching/distinguished-projects/2015/m.sipko.pdf

[3] A. Sarcevic, M. Vranic, D. Pintar, and A. Krajna, "Predictive modeling of tennis matches: a review," in *2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO)*, 2022, pp. 1099–1104. [Online]. Available: https://ieeexplore.ieee.org/document/9803645

[4] F. J. Klaassen and J. R. Magnus, "Forecasting the winner of a tennis match," *European Journal of Operational Research*, vol. 148, no. 2, pp. 257–267, 2003, sport and Computers. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221702006823

[5] J. del Corral and J. Prieto-Rodrguez, "Are differences in ranks good predictors for grand slam tennis matches?" *International Journal of Forecasting*, vol. 26, no. 3, pp. 551–563, 2010, sports Forecasting. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169207009002076

[6] W. Gu and T. L. Saaty, "Predicting the outcome of a tennis tournament: Based on both data and judgments," *Journal of Systems Science and Systems Engineering*, vol. 28, no. 3, pp. 317–343, 2019. [Online]. Available: https://doi.org/10.1007/s11518-018-5395-3

[7] S. R. Clarke and D. Dyte, "Using official ratings to simulate major tennis tournaments," *International Transactions in Operational Research*, vol. 7, no. 6, pp. 585–594, 2000. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0969601600000368

[8] B. L. Boulier and H. O. Stekler, "Are sports seedings good predictors?: an evaluation," *International Journal of Forecasting*, vol. 15, no. 1, pp. 83–91, 1999. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169207098000673

[9] J. S. Dong, L. Shi, L. V. N. Chuong, K. Jiang, and J. Sun, "Sports strategy analytics using probabilistic reasoning," in *2015 20th International Conference on Engineering of Complex Computer Systems (ICECCS)*, 2015, pp. 182–185. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7384244?casa_token=MDPajVlBpKYAAAAA:9YcOP9aCOM3elNAITeegwajwRZf9kB8O5VswHQco3BPyJ-krFfwXFb4Rw9qV103oswM3no3giXE

[10] C.-M. Rosca and A. Stancu, "Real-time betting algorithm for tennis matches," in *2022 14th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, 2022, pp. 1–4. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9847483

[11] T. Barnett and S. R. Clarke, "Combining player statistics to predict outcomes of tennis matches," *IMA Journal of Management Mathematics*, vol. 16, no. 2, pp. 113–120, 2005. [Online]. Available: https://ieeexplore.ieee.org/document/8132379

[12] S. Wilkens, "Sports prediction and betting models in the machine learning age: The case of tennis," *Journal of Sports Analytics*, vol. 7, no. 1, pp. 99–117, 2021. [Online]. Available: https://content.iospress.com/articles/journal-of-sports-analytics/jsa200463