

**Using Neural Networks to Predict WTA Tennis Outcomes**

**Course Project- Part Two**

December 4, 2024

Mike Raucci, Derartu Werji, & Will King

BIA-678 Big Data TEchnology

Stevens Institute of Technology – Fall 2024

Table of Contents

[1 Introduction 1](#_Toc184650000)

[2 Data Collection 2](#_Toc184650001)

[2.1 Match Details 2](#_Toc184650002)

[2.2 Player Information 2](#_Toc184650003)

[2.3 Match Statistics 2](#_Toc184650004)

[2.4 Betting Information 2](#_Toc184650005)

[3 Feature Engineering 3](#_Toc184650006)

[3.1 Match Details 3](#_Toc184650007)

[3.2 Player Information 3](#_Toc184650008)

[3.3 Match Statistics 3](#_Toc184650009)

[3.4 Betting Information 3](#_Toc184650010)

[4 Model Development 4](#_Toc184650011)

[4.1 Architecture 4](#_Toc184650012)

[5 Validation and Testing 5](#_Toc184650013)

[5.1 K-Fold Cross Validation 5](#_Toc184650014)

[6 Application Development 7](#_Toc184650015)

[7 Conclusions 9](#_Toc184650016)

[8 References 9](#_Toc184650017)

# Introduction

This study details the development, training, validation, testing, and application of an artificial neural network designed to predict pre-match betting odds for WTA Tour Grand Slam tennis matches. The inspiration for this research comes from two benchmark studies[1][2] examining the use of pre-match betting odds to predict tennis match outcomes. These studies demonstrated that pre-match odds set by reputable sportsbooks contain inherent, hidden information about each matchup that can be valuable for prediction purposes. Based on this insight, we developed a neural network that processes user-input data about WTA Tour Grand Slam matchups to generate predictions for both players' pre-match betting odds.

# Data Collection

The artificial neural network was trained using data from two complementary sources listing comprehensive WTA Tour Grand Slam matchup data for the years 2016-2023. The first source, the Jeff Sackmann WTA GitHub repository[3], provided detailed match and player information. The second source, tennis-data.co.uk[4], supplied match statistics and betting information. All data points fell into four main categories, which are explained below.

## Match Details

These data points define the situation surrounding the match. Data points would include tournament location, surface tournament is played on, round of tournament matchup takes place, etc. The neural network was not designed to make predictions for these data points, they would instead be mostly used as unique match identifiers or as variables that would be input by the user into the interface of our final application.

## Player Information

These data points define the personal information of both players involved in the matchup. The Jeff Sackmann WTA repository provided data points including the name, age, height, IOC designation, WTA Rank, WTA points, and various others for each player involved in every matchup. The neural network was not designed to make predictions for these data points, they were used as unique match identifiers or as variables that would be input by the user into the interface of our final application.

## Match Statistics

These data points define how the winner and loser was able to perform in isolated matchup. Data points including minutes played, aces, double faults, first serve in-play percentage, first serve won percentage, and other key match statistics were provided by the Jeff Sackmann repository. Data points for score in each set and final match score were provided by tennis-data.co.uk. The neural network was designed to make predictions on all of these data points before making its final prediction for the pre-match betting odds. None of this data would be input by the user, the model would instead use the input data given by the user to predict these statistics for the given matchup. These statistics are predicted for every matchup; however they are not output for the user in our final application.

## Betting Information

The pre-match betting odds for both the winning player and the losing player were provided by tennis-data.co.uk. The model is designed to make predictions on these data points based on the input data given by the user and the predictions made on the match statistics discussed above. These predicted pre-match betting odds are the output variables given to the user in the interface for our final application.

# Feature Engineering

Before training, we engineered several features for the network to be trained on. These features were created using data points that were sourced from our two data sources.

## Match Details

Created a data point to represent the country each matchup is played in. This data point was used to create another feature, home factor, which we discuss in the next section. Since the study only analyzes WTA Grand Slam events, the four countries included are (1) Australia, (2) France, (3) Great Britain, and (4) the United States.

## Player Information

Features such as age difference, height difference, rank difference and others are engineered by simply taking the difference between the winning and losing player’s age, height, rank, etc. When the user inputs the data points for age, height, and rank for each player in the final application, the network will automatically develop these featured data points using the input data.

Home factor is a binary data point created to represent whether the player is playing in their home nation. By comparing the country in which the tournament is being played with the player’s IOC designation, we would indicate whether either of the two players’ have a “home field advantage”.

## Match Statistics

Features such as ace differential, double fault differential, first serve won differential were engineered simply by taking the difference between the winning and losing player’s aces, doble faults, and first serve won rate for each given match in the dataset. The neural network was trained on these data points and makes predictions for these data points; however, they are not output to the user in the final application.

## Betting Information

Features were created to represent the difference between the pre-match odds of the favorite and the odds of the underdog to win the match. As well as a binary classification to indicate whether the pre-match favorite was the player who ended up winning the match. The model makes predictions for these features, but they are not outputs for the user in the final application.

# Model DevelopmentA screenshot of a computer program Description automatically generated

The artificial neural network was designed as a deep learning model specifically tailored for predicting tennis match betting odds. The architecture was developed with several key considerations in mind:

## Input Layer Design

The network accepts multiple types of input features, including match details (like surface type and tournament round), player statistics (such as rank, points, and physical attributes), and historical performance metrics. These inputs were carefully selected to provide comprehensive match context while avoiding redundant or less predictive features.

## Hidden Layer Architecture

The network utilizes four hidden layers of decreasing size (256, 128, 64, and 32 neurons), creating a funnel-like structure. This progressive reduction in layer size helps the network learn increasingly abstract representations of the data while reducing dimensionality. Each hidden layer incorporates several important components:

* Batch normalization to standardize data flow and improve training stability
* ReLU activation functions to introduce non-linearity and prevent vanishing gradients
* Dropout layers (30% dropout in earlier layers, 20% in later layers) to prevent overfitting
* L2 regularization to further control overfitting by penalizing large weights

## Output Layer Configuration

The output layer consists of two neurons, corresponding to the predicted betting odds for both players (B365W and B365L). This dual output design allows the network to simultaneously predict odds for both players while maintaining the relationship between these predictions.

## Training Considerations

The model uses the Huber loss function instead of mean squared error, making it more robust to outliers in betting odds. The Adam optimizer was chosen for its adaptive learning rate capabilities, and various callbacks were implemented for optimal training:

* Early stopping to prevent overfitting
* Learning rate reduction when performance plateaus
* Model checkpointing to save the best performing version

The design incorporates sophisticated feature engineering, including performance ratios, log transformations, and historical metrics, allowing the network to capture complex patterns in tennis match dynamics.

# Validation and Testing

## K-Fold Cross Validation

The model's performance was evaluated using 5-fold cross-validation, a robust method to assess generalization ability. In this process, the dataset was divided into five equal parts, with each fold serving as the validation set once while the remaining folds formed the training set. This approach ensures that every data point is used for both training and validation, providing a comprehensive assessment of model performance.

## Cross Validation Results

The model demonstrated consistent performance across all five folds, with the following metrics:

Mean Squared Error (MSE):

* Fold 1: 0.156
* Fold 2: 0.149
* Fold 3: 0.158
* Fold 4: 0.153
* Fold 5: 0.155
* Average MSE: 0.1542 (±0.0034)

Mean Absolute Error (MAE):

* Fold 1: 0.248
* Fold 2: 0.242
* Fold 3: 0.251
* Fold 4: 0.246
* Fold 5: 0.249
* Average MAE: 0.2472 (±0.0034)

Root Mean Square Error (RMSE):

* Average RMSE: 0.3927 (±0.0043)

The small standard deviations in the metrics (shown in parentheses) indicate consistent performance across folds, suggesting that the model is robust and generalizes well to unseen data. This consistency is particularly important for a betting odds prediction system, as it indicates reliable performance across different subsets of tennis matches.

The RMSE of 0.3927 indicates that, on average, the model's predictions deviate from the actual betting odds by approximately 0.39 units. Given the scale of betting odds, this represents a reasonable level of accuracy for practical applications.

# Application Development

A screenshot of a computer

Description automatically generated

The application was developed as a desktop GUI using tkinter, chosen for its simplicity, native Python integration, and cross-platform compatibility. The architecture was designed to be user-friendly while maintaining robust functionality for tennis match predictions.

## Key Structural Components

1. Data Input Interface:

* Organized input fields into logical sections (Match Details, Player Details, Rankings and Points, Additional Info) to improve user experience
* Implemented dropdown menus for categorical inputs (like Country, Surface, Round, player hands) to prevent invalid entries
* Created numerical entry fields for continuous variables (like rankings, points, age, height)
* Each input field was carefully labeled and organized to guide users through the data entry process

1. Data Validation System:

* Implemented comprehensive input validation to ensure all required fields contain appropriate values
* Created error handling mechanisms to provide clear feedback when invalid data is entered
* Added type checking to ensure numerical fields contain only numbers and are within reasonable ranges
* Included safeguards against missing or incomplete data

1. Prediction Processing:

* Designed a pipeline to transform user inputs into the format required by the neural network
* Implemented the same feature engineering techniques used during training
* Created a seamless flow from raw input to preprocessed data to final prediction
* Added error handling for the prediction process

1. Results Display:

* Created a clear presentation of prediction results showing both players' predicted odds
* Included confidence assessments based on the prediction margins
* Designed an easily interpretable output format
* Added the ability to make multiple predictions without restarting the application

## Technical Considerations

The application's technical infrastructure was built with efficiency and maintainability in mind. The preprocessing components, including scalers and encoders, were saved alongside the model to ensure consistent data transformation across sessions. A modular design approach was implemented, making the application easy to update and maintain over time. Comprehensive logging functionality was integrated to track potential issues and system performance, allowing for continuous improvement and debugging when necessary.

## User Experience

The interface was designed with a strong emphasis on intuitive interaction and ease of use. Following a logical flow from input to output, the application guides users through the prediction process naturally. Clear instructions and tooltips were strategically placed throughout the interface to provide guidance when needed, while responsive design elements offer immediate feedback on user actions. The layout was carefully crafted to be accessible to users regardless of their technical expertise, ensuring that anyone interested in tennis match predictions could effectively utilize the application.

## Error Handling

A robust error management system was implemented to ensure smooth operation and user confidence. The application features comprehensive error catching and reporting mechanisms that not only identify issues but also provide clear, user-friendly messages explaining what went wrong and how to resolve it. The system was designed to recover gracefully from errors without crashing, maintaining data integrity and user progress. Additionally, real-time validation feedback was implemented to guide users toward correct input, preventing errors before they occur and ensuring a smooth user experience.

# Conclusions

In conclusion, this research successfully developed and implemented a comprehensive solution for predicting pre-match betting odds in WTA Tour Grand Slam tennis matches. Building upon previous studies that identified the predictive value of betting odds, we created a sophisticated neural network architecture that effectively captures complex patterns in tennis match data. The model's robust performance, demonstrated through 5-fold cross-validation with an average RMSE of 0.3927 (±0.0043), indicates reliable predictive capabilities across different match scenarios. The consistent performance across all validation folds suggests strong generalization ability, a crucial factor for real-world applications. The development process culminated in a user-friendly application that effectively bridges the gap between complex machine learning technology and practical user needs. Through careful attention to technical infrastructure, user experience, and error handling, the final product provides an accessible tool for tennis match analysis. This project not only validates the feasibility of using artificial neural networks for betting odds prediction but also demonstrates how such technology can be effectively deployed for practical use, contributing to the growing field of sports analytics and prediction systems.

# References

[1] Lisi, Francesco & Zanella, Germano. (2013). Tennis betting: Can statistics beat bookmakers?. Electronic Journal of Applied Statistical Analysis. 00. 1-35. 10.1285/i20705948v10n3p790.

[2] Candila, Vincenzo & Palazzo, Lucio. (2020). Neural Networks and Betting Strategies for Tennis. Risks. 8. 10.3390/risks8030068.

[3] Sackmann, Jeff. (2015). tennis\_wta. [JeffSackmann/tennis\_wta: WTA Tennis Rankings, Results, and Stats](https://github.com/JeffSackmann/tennis_wta). (2024).

[4] *Tennis-data.co.uk*. Tennis Betting, Tennis Results & Tennis Live Scores. (n.d.). http://tennis-data.co.uk/