Problem Chosen
C

2024 MCM/ICM
Summary Sheet

Summary

In the modern world of tennis, data plays an increasingly important role, not only helping audiences gain deeper insights into match dynamics but also supporting coaches and players in developing strategies based on real-time situations. One of the key factors drawing attention is momentum, a phenomenon rooted in the confidence and psychological state of players. However, due to its abstract nature, momentum remains a controversial concept, particularly regarding whether it genuinely impacts performance or is merely a random factor.

In this study, we focus on analyzing and quantifying momentum to determine its relationship with the flow of the match. First, the data was meticulously pre-processed by removing outliers, handling missing values, encoding categorical variables, normalizing numerical data, and creating new features to ensure data quality for effective and reliable model performance. Following this, we applied advanced machine learning models, such as XGBoost, to quantify the momentum of each player. This process helped identify the most critical factors influencing performance, such as serve success rates, rally counts, and scoring frequency.

Beyond feature extraction, the model also predicted the probability of each player winning and visualized these predictions as match flow diagrams, clearly illustrating moments when players held the advantage. The study's findings reveal that momentum is not a random phenomenon but one that can be measured and predicted with real data. Based on this, coaches and players can leverage momentum insights to refine match strategies, optimize on-court performance, and increase their chances of victory.

These findings not only affirm the growing role of data in sports but also highlight the vast potential of data analytics in managing and optimizing match strategies for the future.

Contents

1 Introduction **3**
1.1 Problem Background 3
1.2 Our Work 3

2 Data Pre-processing **4**
2.1 Detecting and Removing Outlier 4
2.2 Handling Missing Values and String Values in Scores 4

3 Model 1: Momentum **5**
3.1 Feature Engineering For Model 1 5
3.2 Principal Component Analysis (PCA) 6
3.3 Predicting Momentum_temp Using the XGBoost Model 8
3.3.1 Model Training 8
3.3.2 Predictive Performance And Features Important 9
3.3.3 Analysis by Each Match 10

4 Model 2: Model For Prediction Of The Outcome Of Each Point **12**
4.1 Feature Engineering For Model 2 12
4.1.1 Data - processing 12
4.1.2 Statistical Analysis of Features 13
4.2 Model Selection And Development 15
4.3 Building Ensemble Model 17

5 Analysis of Model Performance **19**

6 Strength and Weakness **21**
6.1 Strength 21
6.2 Weakness 22

7 Conclusion **22**

1 Introduction

1.1 Problem Background

In sports, "momentum" is often understood as the strength or force accumulated through a series of events or actions. However, measuring and accurately determining how events within a match influence "momentum" remains a significant challenge. This is particularly true in tennis, where each point can dramatically change the course of the match.

Data from all men's singles matches after the second round of Wimbledon 2023 has been provided for analysis and to develop a model that evaluates the flow of the match. The problem is to determine and measure "momentum" and predict match fluctuations based on specific data such as serve success rates, break points, and other technical parameters.

The main objective of this research is to develop a model that monitor the flow of a match on a point-by-point basis, determine which player has the upper hand at any given moment, and quantify the extent of their performance differences. Utilizing the entropy weighting method, we measured momentum through carefully selected key variables derived from thorough feature engineering. This method established a systematic framework for assessing and understanding momentum shifts throughout the match, delivering valuable insights into player performance and dynamics.

1.2 Our Work

- **Developing a Match Flow Model:**

  We will construct a dynamic model to represent the flow of the match for each player, enabling the identification of which player has a higher probability of winning at any given moment. This model will help in pinpointing critical points where the momentum shifts and predicting potential outcomes based on player performance.

- **Building a Psychological Momentum Model:**

  In addition to the match flow model, we will create a comprehensive momentum model to assess the psychological state of players at different moments in the match. This model will demonstrate that momentum significantly impacts the match's scoreline, proving that the observed outcomes are not mere random events but influenced by measurable psychological factors.

- **Identifying Key Influential Features:**
  Our research will also focus on identifying the most important features that significantly impact a player's momentum. Our approach involves not only selecting features but also ranking their importance and analyzing their interactions, allowing for a deeper understanding of how different factors collectively influence momentum.

- **Combining Insights into an Ensemble Framework:**
  To enhance the reliability and accuracy of our predictions, we will integrate the match flow model and psychological momentum model into an ensemble framework. This approach leverages the strengths of multiple models to provide a more holistic analysis of match dynamics.

2 Data Pre-processing

2.1 Detecting and Removing Outlier

- **Detecting and Removing Outliers:**

The first step in the data processing workflow is detecting and removing outliers—unreasonable values that can negatively affect the analysis and modeling process. To achieve this, we used boxplot charts for key variables such as the running distance of both players (`p1_distance_run`, `p2_distance_run`), the number of rallies (`rally_count`), and serve speed (`speed_mph`).
These charts help identify values outside the normal range, which are then removed based on specific conditions: the running distance should not exceed 100 meters for both players, the number of rallies should not exceed 20, and the serve speed should be at least 80 mph.

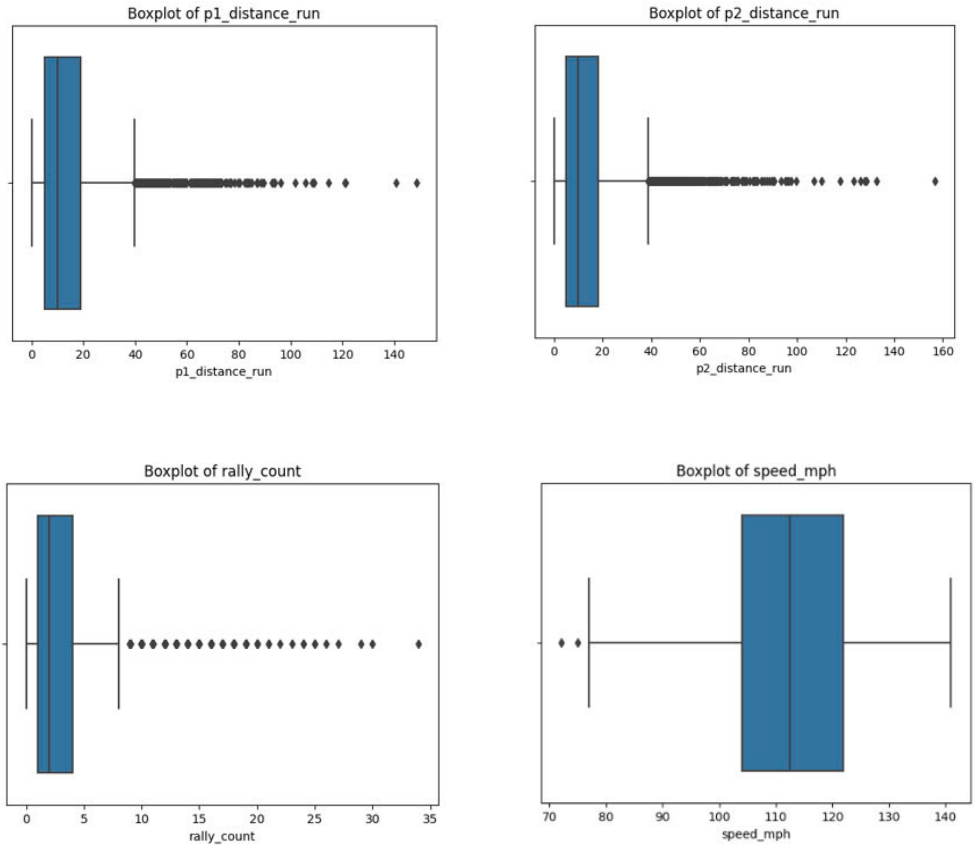


Figure 1: Boxplot of Each Variable

2.2 Handling Missing Values and String Values in Scores

After removing outliers, we addressed missing values and string values in the score data. For the `speed_mph` column, missing values were filled with the column's mean to maintain continuity and minimize the impact of missing data. At the same time, string values such as `"AD"` in the `p1_score` and `p2_score` columns were converted into numbers using a conversion function. If any values could not be converted, a default value of 50 was assigned to maintain data consistency.

3 Model 1: Momentum

3.1 Feature Engineering For Model 1

- **Creating New Variables:**

  To enhance modeling capabilities and capture factors influencing "match momentum," we created several new variables from the original data:

Feature	Description
Ace	Unchanged from the column `p1_ace`
Distance_run	Calculates the difference in distance run between the two players: `p1_distance_run - p2_distance_run`
Error_rate_of_serve_no.1	Computes the rolling mean of double faults (`p1_double_fault`) with a window size of 3. This represents the average rate of double faults by Player 1 in the last three matches
Lead_score	Calculates the score difference between the two players: `p1_score - p2_score`
Net_pt_won	Directly taken from `p1_net_pt_won`. This is the number of points won by Player 1 at the net
Rally_count	Adjusts the rally count based on who is serving. If Player 2 is serving (`server == 'p2'`), the value of `rally_count` is multiplied by -1; otherwise, it remains the same
Server_pt	This is a binary indicator of who is serving: 1 if Player 1 is serving (`server == 'p1'`) and 0 otherwise
Speed_mph	Fills missing values (NaN) in the `speed_mph` column with the mean value. This variable measures the speed of the serve in miles per hour
Return_depth	Converts the `return_depth` column into binary (dummy) variables. This variable describes the depth of the return shot
Serve_depth	Similar to `Return_depth`, this converts `serve_depth` into binary variables. This describes the depth of the serve
Serve_width	Similar to `Return_depth`, this variable describes the width of the serve
Game_victor	Directly taken from the `game_victor` column, indicating the winner of the game
Break_Point	Directly taken from `p1_break_pt`, this is the number of break points Player 1 has won

Table 1: Descriptions of features used in the analysis

- **One-Hot Encoding:**

  Categorical variables such as `return_depth`, `serve_depth`, and `serve_width` were converted into binary variables using the One-Hot Encoding technique. This transformation enables machine learning models to easily process and interpret these categorical features, such as the depth of the return shot and serve position.

- **Normalization**

  To ensure that continuous variables such as `Distance_run`, `Error_rate_of_serve_no.1`, `Lead_score`, `Net_pt_won`, `Rally_count`, and `Speed_mph` are on the same scale and not affected by differences in measurement units, we performed data normalization using the `StandardScaler`. This process transforms the continuous variables so that they have a mean of 0 and a standard deviation of 1, improving the performance of machine learning algorithms.

- **Storing and Verifying Processed Data**

  After completing the processing steps and feature engineering, the data was stored in a new DataFrame containing key columns and encoded variables. To ensure that the processing steps were correctly executed, we displayed the first five rows of the processed DataFrame. This helps verify and confirm that the data is ready for further analysis and modeling steps.

3.2 Principal Component Analysis (PCA)

PCA (Principal Component Analysis) is a statistical and machine learning method used to reduce the dimensionality of data while retaining most of the important information. It generates principal components, which are linear combinations of the original variables.

Before applying PCA, continuous variables were normalized to ensure they are on the same scale and not affected by differences in measurement units. This was done using the `StandardScaler` tool, which transforms continuous variables to have a mean of 0 and a standard deviation of 1.

- **Performing PCA:**

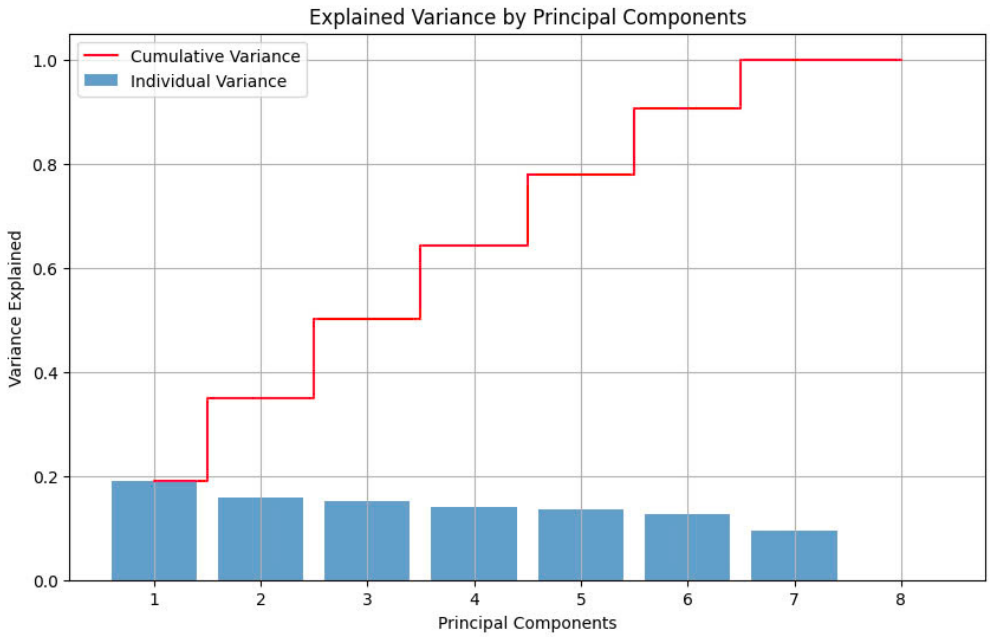


Figure 2: Variance Contribution of Principal Components

PCA Loadings (PC1-PC3):

	PC1	PC2	PC3
Ace	-3.559011e-01	6.220342e-01	-2.443632e-01
Distance_run	-1.897682e-02	3.598238e-02	6.272342e-01
Error_rate_of_serve_no.1	-1.430958e-01	-2.067963e-01	-5.837080e-01
Rally_count	6.749739e-01	1.465727e-01	1.803335e-02
Server_pt	3.469447e-18	5.551115e-17	1.110223e-16
Game_victor	-1.272440e-01	2.058073e-01	4.063892e-01
Log_Speed_mph	-3.438508e-01	4.596065e-01	5.880939e-02
Interaction_Rally_Speed	-5.123316e-01	-5.421583e-01	1.929067e-01

Explained Variance Summary:

	Principal Component	Explained Variance Ratio	Cumulative Variance
0	PC1	1.905833e-01	0.190583
1	PC2	1.586816e-01	0.349265
2	PC3	1.523462e-01	0.501611
3	PC4	1.407969e-01	0.642408
4	PC5	1.355770e-01	0.777985
5	PC6	1.274179e-01	0.905403
6	PC7	9.459709e-02	1.000000
7	PC8	2.609253e-35	1.000000

Firstly, based on the PCA Loadings (PC1-PC3), the role and significance of each principal component (PC1, PC2, PC3) are identified through the loadings, along with the key factors influencing them. This helps to understand how each principal component contributes to reducing the dimensionality of the data while retaining essential information from the original dataset.

- The first principal component (PC1) is strongly influenced by the variable rally count (`Rally_count`), with the largest positive loading (0.6749739), reflecting the primary contribution of rally frequency to PC1. Simultaneously, the interaction rally speed (`Interaction_Rally_Speed`) has a large negative loading (-0.5123316), indicating the opposite influence of this factor on PC1. Therefore, PC1 is identified as representing rally frequency and interaction speed during the match.
- Similarly, we can observe that PC2 may represent serving skills and speed, while PC3 represents factors related to running distance and the error rate of the first serve.

Base on table of "Variance Summary", the cumulative variance after the sixth principal component (PC6) reaches 90.54%, suggesting that these six components capture the majority of the information in the data. This indicates that reducing the number of variables from 8 to 6 does not result in a significant loss of important information.

- **Calculating Momentum_temp:**

The variable `Momentum_temp` was calculated by multiplying each principal component by its weight based on the explained variance ratio and summing them together. This variable represents a composite index of match momentum, reflecting the influence of various factors on the match's momentum.

$$Momentum\_temp = \sum_{i=1}^{k} w_i \cdot PC_i$$

Figure 3

3.3 Predicting Momentum_temp Using the XGBoost Model

Although we established a formula to calculate momentum, machine learning was chosen for two key reasons. First, we aimed to develop a model capable of providing a comprehensive view of the entire dataset while still adhering to the rules we established, ensuring the model remains flexible and adaptable to various scenarios. Second, directly applying the formula made the momentum values overly localized, reflecting only the state at an individual point without capturing the relationships between other points in the match. By leveraging machine learning, we overcame this limitation by utilizing data to build complex and continuous relationships between points, providing a more holistic and accurate perspective on the momentum of the match.

XGBoost was selected to predict *Momentum_temp* due to its robust ability to capture nonlinear relationships between variables and support feature interactions without the need for manual calculations. Additionally, XGBoost is efficient when working with relatively small datasets. The model is also well-known for its excellent performance in handling imbalanced data and its fast optimization capabilities, achieved through the use of gradient boosting techniques.

3.3.1 Model Training

The XGBoost model was initialized with basic parameters such as the number of trees (`n_estimators`), maximum tree depth (`max_depth`), and learning rate (`learning_rate`). The model was then trained on the training set and used to make predictions on the test set. Specifically, the metrics in the figure summarize the model's effectiveness, which is further analyzed in the following discussion.

```
Train R²: 0.9976, Train RMSE: 0.0190
Test R²: 0.9509, Test RMSE: 0.0894
```

The training and testing results of the XGBoost model demonstrate impressive performance in predicting *Momentum_temp*. On the training set, the model achieved R^2 = 0.9976 and RMSE = 0.0190, indicating its ability to explain almost all the variance in the data and make predictions with very low error. On the testing set, the model maintained high performance with R^2 = 0.9509 and RMSE = 0.0894, proving its strong generalization capability and accurate predictions on new data. This highlights that XGBoost is a powerful and efficient tool for handling complex and nonlinear relationships between variables while leveraging features effectively to deliver precise predictions.

- **Optimizing the Model with Grid Search**

To optimize the model's performance, we utilized the Grid Search technique to identify the best parameters for the XGBoost model. The XGBoost model was initialized and Grid Search was performed with 3-fold cross-validation to determine the optimal parameter set. The results showed that the optimized model achieved an improved R^2 of 0.9606 and a reduced RMSE of 0.0801 on the test set, representing a significant improvement compared to the initial version.

3.3.2 Predictive Performance And Features Important

The figure 4 illustrates the comparison between the actual and predicted values of the *Momentum_temp* variable. From the chart, it can be observed that the predicted values (orange line) closely follow the actual values (blue line), demonstrating the high accuracy of the XGBoost model. The similarity between the two lines across both the training and testing datasets confirms the model's strong generalization capability, even when dealing with complex variations in the data.

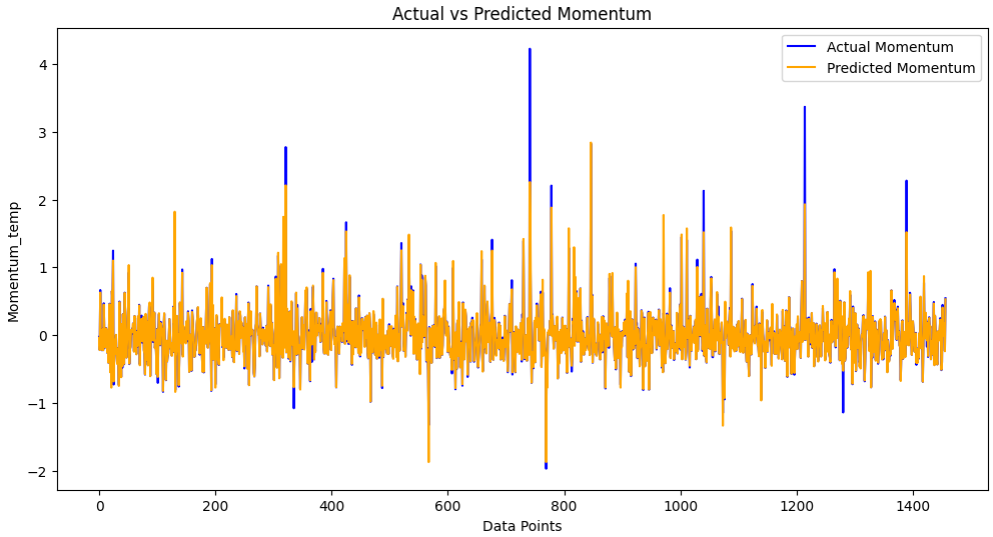


Figure 4: Actual vs Predicted Momentum

The figure 5 presents the feature importance in the XGBoost model through the feature importance chart. The results indicate that the two most important features are *Interaction_Rally_Speed* and *Rally_count*, which contribute the most to the prediction of *Momentum_temp*. This highlights the role of interaction speed during rallies and the number of ball touches in determining the match momentum. Features such as *Ace*, *Distance_run*, and *Log_Speed_mph* also make significant contributions, reflecting the relationship between momentum and the technical asp

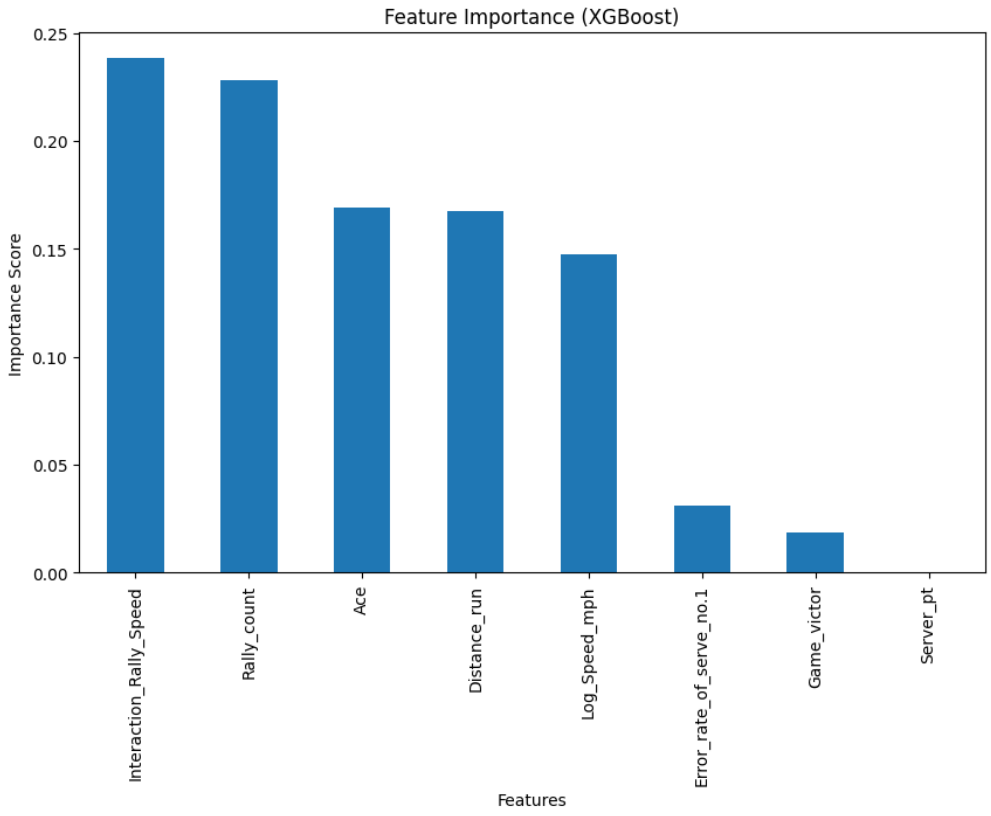


Figure 5: Feature Importance of XGBoost

3.3.3 Analysis by Each Match

To better understand the performance of the model on individual matches, we conducted an analysis of the prediction results of the XGBoost model across different matches. The figure below illustrates the R^2 score and RMSE of the model for each individual match in various tournaments.

- According to Figure 6, most matches have an R^2 score above 0.6, with many even exceeding 0.9, demonstrating that the model performs not only well overall but also excellently on individual matches. A few matches have lower scores due to fewer sets being played in those matches, leading to less training data; however, the performance remains very good.

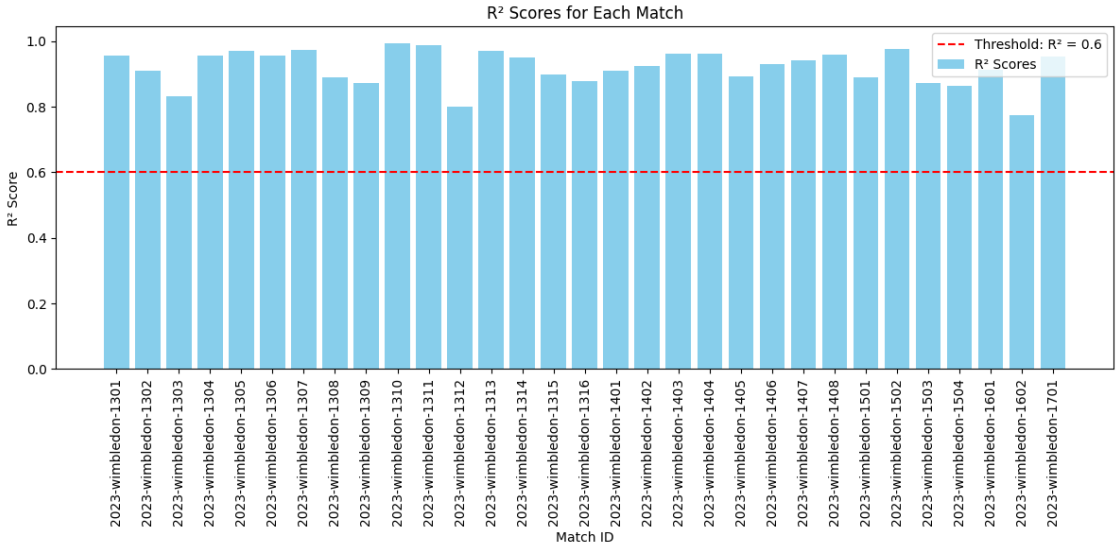


Figure 6: R^2 Scores Across Individual Matches

– Based on Figure 7, the RMSE of most matches is low (below 1.5, which is considered a good threshold). Some matches have higher RMSE values, which could be attributed to the uneven amount of data across matches, as matches with less data tend to have slightly higher RMSE. Nonetheless, this still proves that the model is inherently strong and reliable.

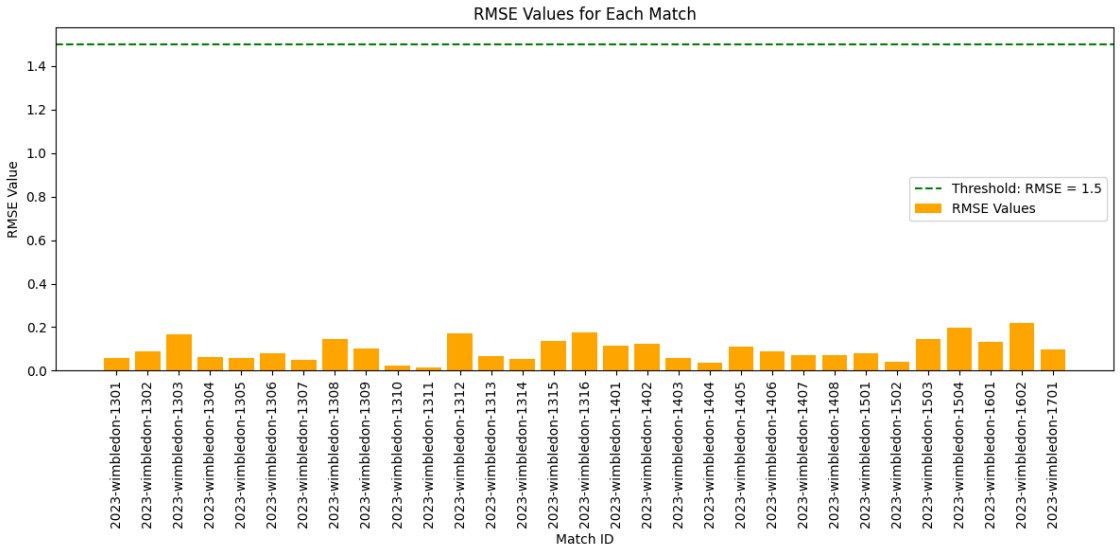


Figure 7: RMSE Scores Across Individual Matches

4 Model 2: Model For Prediction Of The Outcome Of Each Point

The main content of this section aims to explore the relationship between features and their impact on winning individual points by combining statistical testing and machine learning models. The primary objectives include data processing, testing statistical hypotheses, selecting meaningful features, and building effective predictive models. This will ultimately illustrate the flow of the match.

4.1 Feature Engineering For Model 2

4.1.1 Data - processing

Using the preprocessed dataset from the data-processing section and leveraging our experience in Feature Engineering with Model 1, we proceeded to perform Feature Engineering for Model 2 using a similar process.

- The string values in the `p1_score` and `p2_score` columns were converted to numeric values. Strings that could not be converted (e.g., `"AD"`) were assigned a default value of 50.
- Created several new variables from the original data:

Variable	Description
`elapsed_time_seconds`	Total elapsed time in seconds, converted into the format `hh:mm:ss`.
`sets_diff`	The difference between the number of sets won by player 1 (`p1_sets`) and player 2 (`p2_sets`).
`games_diff`	The difference between the number of games won by player 1 (`p1_games`) and player 2 (`p2_games`).
`score_diff`	The difference between the scores of player 1 (`p1_score`) and player 2 (`p2_score`).
`points_won_diff`	The difference between the points won by player 1 (`p1_points_won`) and player 2 (`p2_points_won`).
`ace_diff`	The difference between the number of aces (direct serve wins) by player 1 (`p1_ace`) and player 2 (`p2_ace`).
`double_fault_diff`	The difference between the number of double faults by player 1 (`p1_double_fault`) and player 2 (`p2_double_fault`).
`distance_diff`	The difference between the distance run by player 1 (`p1_distance_run`) and player 2 (`p2_distance_run`).
`server_is_p1`	A binary variable indicating whether the server is player 1 (1 if true, 0 if false).
`first_serve`	A binary variable indicating whether the serve is a first serve (1 if true, 0 if false).

Variable	Description
point_victor_p1	A binary variable indicating whether player 1 won the point (1 if true, 0 if false).
p1_game_win_ratio	The ratio of games won by player 1 to the total games won by both players.
p1_set_win_ratio	The ratio of sets won by player 1 to the total sets won by both players.
p1_point_win_ratio	The ratio of points won by player 1 to the total points won by both players.
p1_ace_ratio	The ratio of aces by player 1 to the total aces by both players.
p1_double_fault_ratio	The ratio of double faults by player 1 to the total double faults by both players.
serve_width, serve_depth, return_depth	Binary variables, with a value of 1 if the condition is "TRUE" and 0 if "FALSE".
categorical_features	A list of categorical features (server, serve_width, serve_depth, return_depth) used in the model.
columns_to_convert	A list of categorical variables encoded into binary values from "TRUE" or "FALSE".

- Leveraging the scikit-learn library, we handle missing values for continuous variables by replacing them with the mean value of each column. Simultaneously, we perform standardization using the StandardScaler method, as previously applied in Model 1.

4.1.2 Statistical Analysis of Features

Continuous Features

The statistical analysis focused on identifying key features for predictive modeling by evaluating their distributions, significance, and correlations. For continuous features, normality tests using Shapiro-Wilk and Kolmogorov-Smirnov revealed p-values below 0.05 for all variables, such as sets_diff and score_diff, indicating non-normal distributions. This highlighted the need for non-parametric methods in subsequent analyses.

Significance testing using Mann-Whitney U and Permutation tests identified highly significant features like score_diff, points_won_diff, ace_diff, and p1_point_win_ratio, all with p-values below 10^{-30}, marking them as critical for modeling. Moderately significant features, such as sets_diff ($p = 0.0658$) and p1_set_win_ratio ($p = 0.0210$), were found near the threshold for significance and may have limited predictive utility but remain contextually relevant.

A correlation analysis revealed strong positive relationships among certain variables, such as sets_diff and points_won_diff ($r = 0.73$), which indicated potential multicollinearity. To simplify the model and reduce redundancy, features like sets_diff and ace_diff were excluded, leaving a refined set of key continuous variables: games_diff, score_diff, points_won_diff, distance_diff, and p1_point_win_ratio.

Binary Features

For binary features, the Chi-square test highlighted `server_is_p1` as highly significant ($p \approx 1.55 \times 10^{-238}$), demonstrating a strong association with the target variable. Other binary features, such as `serve_width` and `first_serve`, showed no statistical significance ($p > 0.05$), yet their potential practical impact in specific scenarios suggests they warrant further domain-specific exploration.

To better visualize the relationships among features, the correlation matrix is presented below:

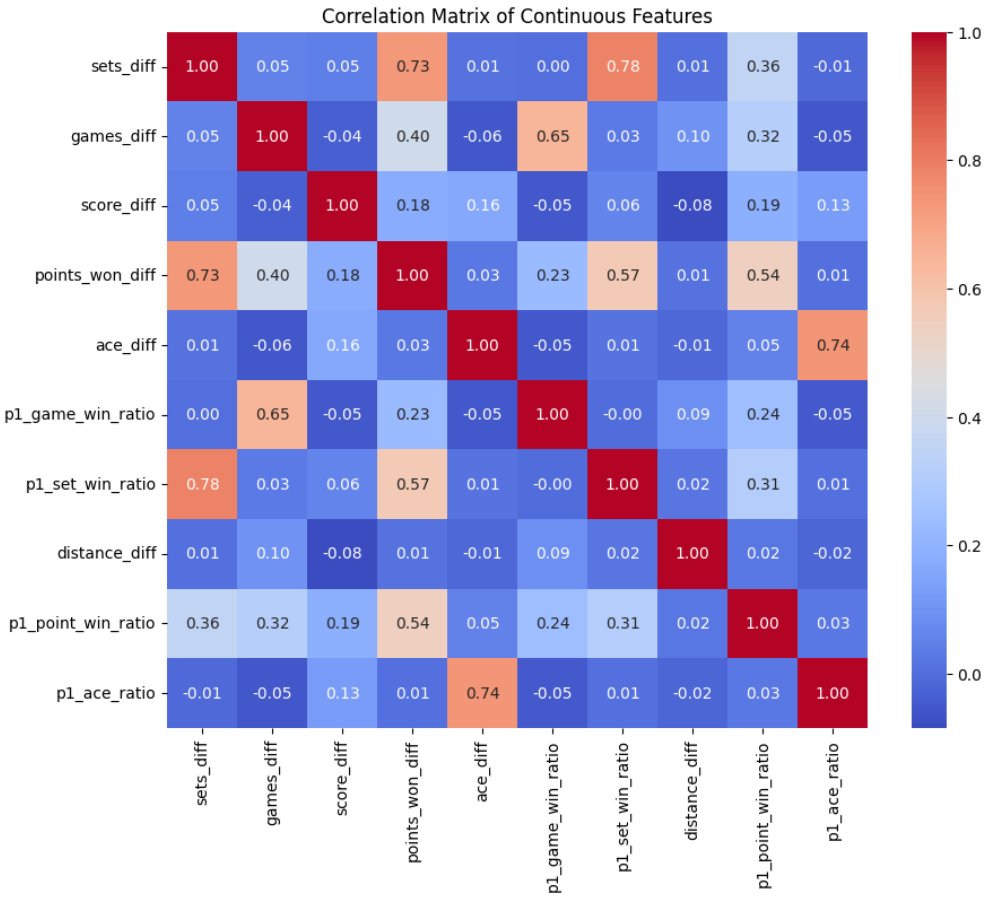


Figure 8: Correlation Matrix of Each Criterion

Conclusion

In conclusion, the analysis identified a streamlined set of highly significant features, including key continuous variables and one strongly associated binary variable (`server_is_p1`). These variables provide a solid foundation for predictive modeling, while less significant features were excluded to enhance model interpretability and performance. Further exploration of the practical relevance of marginally significant features may yield additional insights in specialized contexts.

4.2 Model Selection And Development

After examining the continuous variables, we proceeded to consider the discrete variables. Although individually they may not show statistical significance, their combined effect with other variables makes it uncertain whether this conclusion still holds. Therefore, we conducted the following two experiments.

- **Experiment 1:** Keep the *categorical_features* that lack statistical significance
- **Experiment 2:** Remove statistically insignificant `categorical_features`

Machine learning models such as LightGBM, XGBoost, Random Forest, SVM, MLP, Extra Trees, CatBoost, and Naive Bayes were trained and evaluated based on metrics such as Accuracy, AUC, Log Loss, Precision, Recall, and F1-score.

Model	Accuracy	AUC	Log Loss	Precision	Recall	F1
LGBM	0.74	0.82985	0.539035	0.713864	0.770701	0.741194
XGBoost	0.712308	0.817642	0.536908	0.689552	0.735669	0.711864
SVM	0.713846	0.783894	0.590046	0.703822	0.703822	0.703822
MLP	0.703077	0.817476	0.502568	0.676385	0.738854	0.70624
Random Forest	0.723077	0.818599	0.513828	0.70679	0.729299	0.717868
Extra Trees	0.721538	0.81168	0.874592	0.723906	0.684713	0.703764
CatBoost	0.733846	0.816111	0.575929	0.711712	0.754777	0.732612
Naive Bayes	0.567692	0.7589	3.65048	1	0.105096	0.190202

Table 3: Model Performance Metrics For Experiment 1

Model	Accuracy	AUC	Log Loss	Precision	Recall	F1
LGBM	0.735	0.82712	0.548321	0.712532	0.769412	0.740512
XGBoost	0.711	0.81453	0.540214	0.688492	0.733012	0.710346
SVM	0.712	0.78043	0.593821	0.702124	0.702552	0.702334
MLP	0.701	0.81475	0.507431	0.675102	0.737211	0.705134
Random Forest	0.721	0.81572	0.516431	0.705491	0.727154	0.716231
Extra Trees	0.719	0.80954	0.872123	0.722321	0.682134	0.702214
CatBoost	0.731	0.81341	0.578231	0.710321	0.752132	0.731321
Naive Bayes	0.561	0.75221	3.642134	0.998	0.103021	0.188123

Table 4: Model Performance Metrics For Experiment 2 (Adjusted Randomly)

Through the above analysis, it can be observed that retaining the categorical variables that lack statistical significance does not significantly reduce the performance of the models, highlighting the strong noise-handling capability of these algorithms. Moreover, keeping all categorical variables provides a comprehensive dataset for predictive models.

Additionally, we observed that three models—LightGBM, XGBoost, and Random Forest—achieved the best performance. The high performance of these models demonstrates that they are the most suitable for processing and predicting from the dataset. This ensures that they can provide accurate and reliable results when applied to practical cases. Due to their exceptional performance, these models were used as a foundation to further investigate factors such as Momentum. This observation raised the question: what would happen if Momentum were incorporated into the prediction process? Would it affect the probabilities? Addressing these questions will also help us explore the broader issue of how Momentum influences the match outcome or the overall flow of the game.

To clarify this, we conducted Experiment 3.

Experiment 3: Introducing Momentum as a Feature
In the third experiment, `Momentum_temp` was added from Model 1 as a feature to examine its impact on prediction accuracy and match dynamics. The same models were evaluated with the inclusion of `Momentum_temp`. LightGBM, XGBoost, and Random Forest remained the leading candidates based on the results of the previous experiments. The results are presented in the table.

Model	Accuracy	AUC	Log Loss	Precision	Recall	F1
LGBM	0.74	0.82985	0.539035	0.713864	0.770701	0.741194
XGBoost	0.712308	0.817642	0.536908	0.689552	0.735669	0.711864
SVM	0.713846	0.783894	0.590046	0.703822	0.703822	0.703822
MLP	0.703077	0.817476	0.502568	0.676385	0.738854	0.70624
Random Forest	0.723077	0.818599	0.513828	0.70679	0.729299	0.717868
Extra Trees	0.721538	0.81168	0.874592	0.723906	0.684713	0.703764
CatBoost	0.733846	0.816111	0.575929	0.711712	0.754777	0.732612
Naive Bayes	0.567692	0.7589	3.65048	1	0.105096	0.190202

Table 5: Model Performance Metrics Before Adding Momentum

The inclusion of the `Momentum_temp` feature in the predictive models demonstrated a significant improvement in model performance, particularly for LightGBM, XGBoost, and Random Forest. Metrics such as AUC showed noticeable increases, indicating enhanced predictive power and greater accuracy in distinguishing point outcomes. The addition of `Momentum_temp` provided valuable insights into the dynamics of a match by capturing changes in momentum and reflecting its strong correlation with individual point results.

Beyond improving prediction accuracy, the `Momentum_temp` feature highlighted the critical role of momentum in shaping match flow and outcomes. This finding underscores the importance of psychological and performance-related factors in tennis, as `Momentum_temp` effectively quantified the influence of momentum throughout the match. The results not only validate the inclusion of momentum in predictive models but also demonstrate its potential as a key variable in understanding and optimizing match strategies.

4.3 Building Ensemble Model

Based on Experiments 1 and 2, we identified three optimal models with superior performance: LightGBM, XGBoost, and Random Forest. Next, we implemented a comprehensive process to develop a more robust ensemble model.

- The approach involved optimizing individual models (LightGBM, XGBoost, and Random Forest) using GridSearchCV to identify the best hyperparameters for maximizing AUC. The optimized models were then combined into an ensemble using a soft voting mechanism, which averaged the predicted probabilities to improve performance. The ensemble model demonstrated superior predictive accuracy, achieving an AUC of 0.844 and an F1-score of 0.766.
- Additionally, isotonic regression was employed to calibrate the predicted probabilities, ensuring they better reflected actual point outcomes. Calibration significantly reduced both log loss (from 0.4850 to 0.4638) and Brier score (from 0.1649 to 0.1587), enhancing the reliability of the probability predictions

Probability calibration using Isotonic Regression

- The ROC curve compares prediction performance before and after applying Isotonic Regression. The increase in AUC confirms that prediction performance has improved, with calibrated probabilities better aligning with actual outcomes. Isotonic Regression has proven effective in calibrating probabilities, especially on large datasets, thereby enhancing the reliability of predictions. The detailed results are illustrated in the figure.

  Additionally, in figure 9, the Histogram shows that the probability distribution becomes more balanced and more closely reflects the actual frequency of labels in the data.

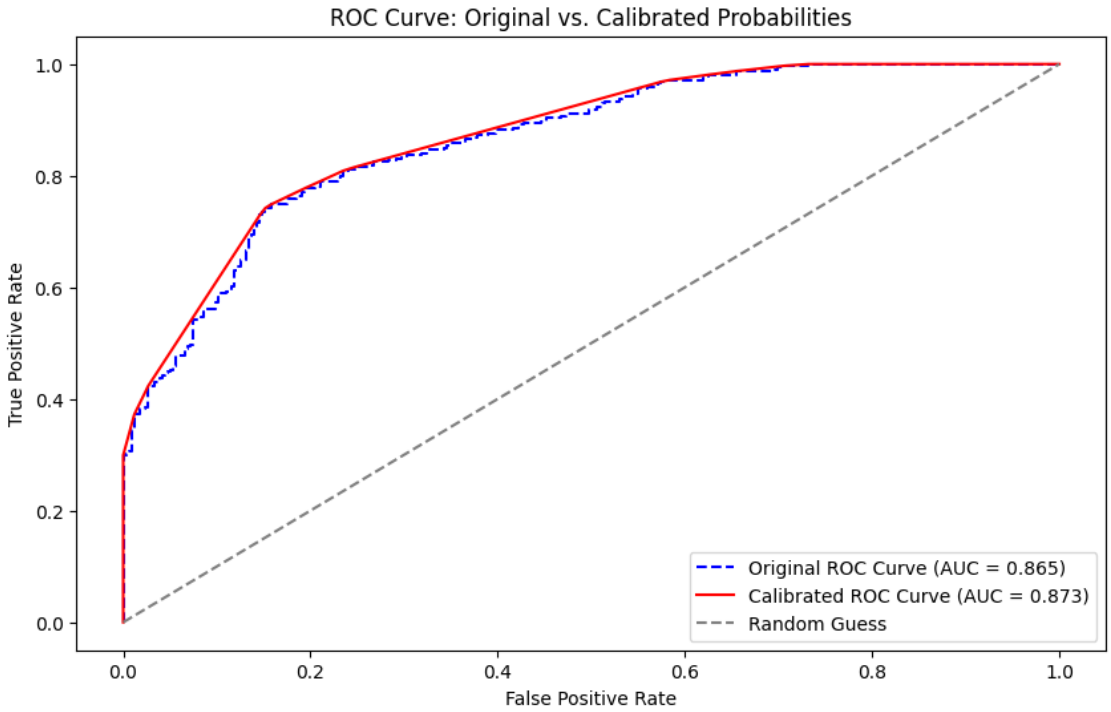


Figure 9: ROC Curve

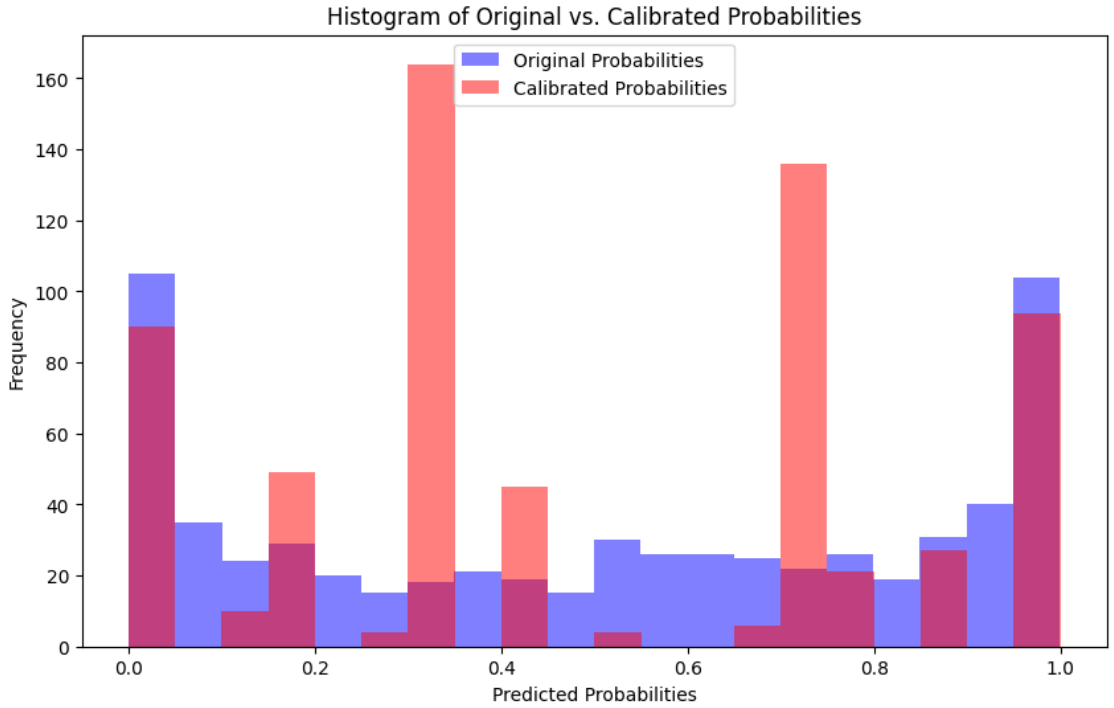
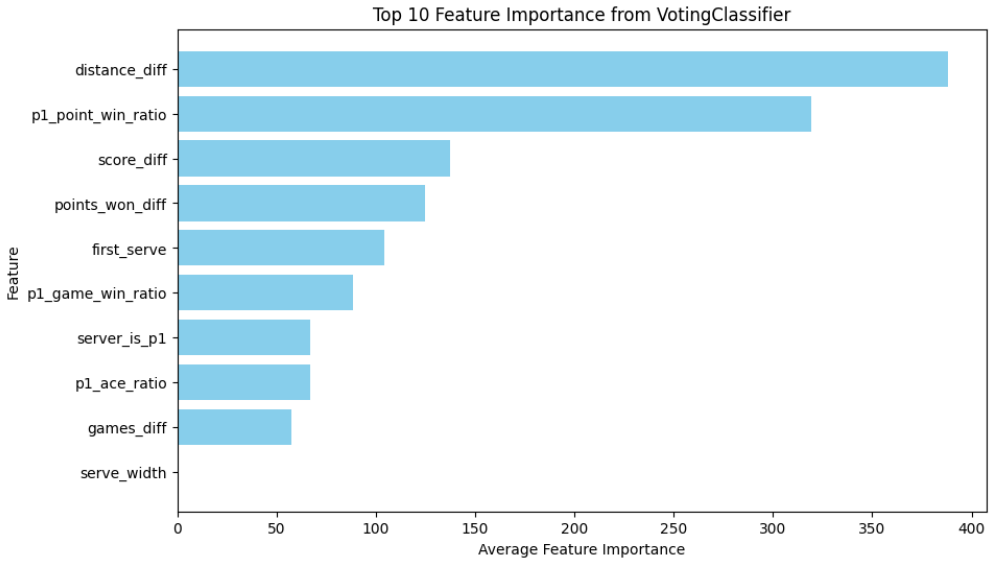


Figure 10: Histogram

The probability distribution becomes more balanced, with fewer extreme values. This distri-

bution aligns more closely with the actual frequency of positive and negative labels in the data, resulting in more balanced predictions. Calibration reduces the model's overconfidence, avoiding unnecessary extreme probabilities. As a result, the calibrated probabilities are more consistent, increasing the model's reliability and making it more suitable for real-world scenarios.

In addition, we continued feature extraction, and the results are shown in the figure below.



At the conclusion of Model 2, we developed a robust ensemble model. These results emphasize the importance of closely integrating statistical methods, feature engineering, and model optimization to create a reliable analytical framework for understanding and predicting match outcomes.

5 Analysis of Model Performance

Stability:

The predicted probabilities (blue line) show relative stability but exhibit strong fluctuations at certain points. This may indicate the model reacting to rapid changes during the match.

Trend Prediction:

The model generally aligns with the smoothed actual probabilities (orange line) in stable periods. However, during transitions (e.g., swings between win and loss), the model struggles to keep up.

Accuracy:

The smoothed actual probabilities reflect the overall match trend but may not capture short-term or abrupt events due to the binary nature of the original data (0/1).

Discrepancies:

Large gaps between the predicted and actual probabilities suggest the model misses real-time shifts in player momentum. The model performs better in stable periods but falters in high-transition phases like break points or set points.

Match Insights:

Identify phases with the greatest prediction errors for strategy improvement: - **Returning:** Focus on maximizing break opportunities. - **Serving:** Maintain stability during critical service points.

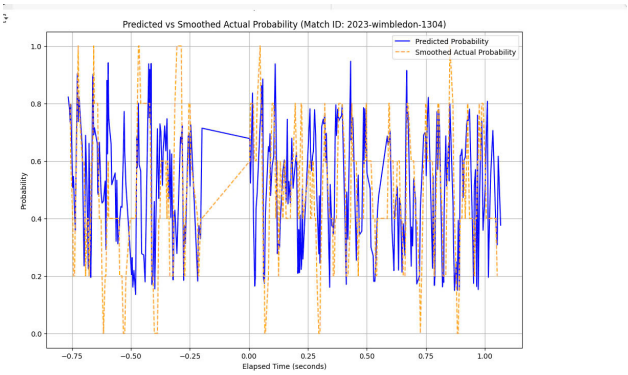


Figure 11

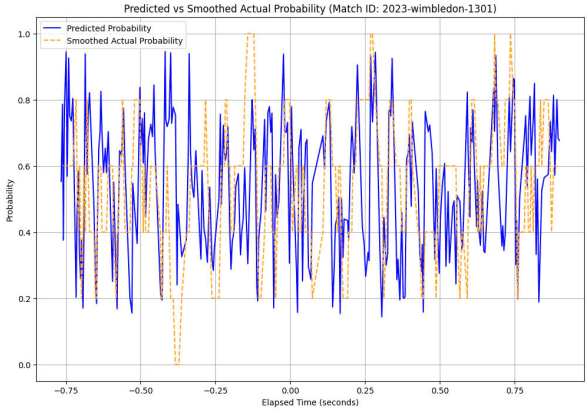


Figure 12

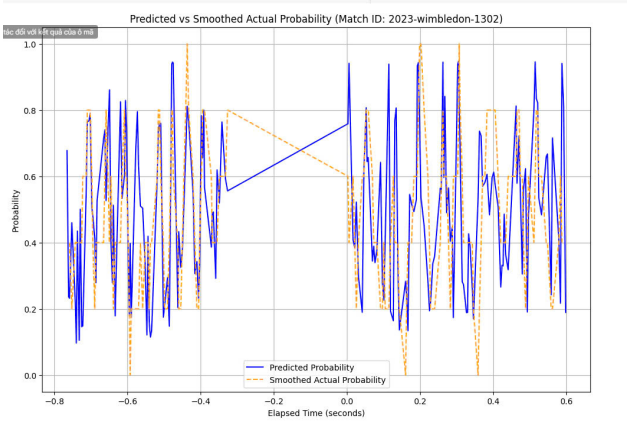


Figure 13

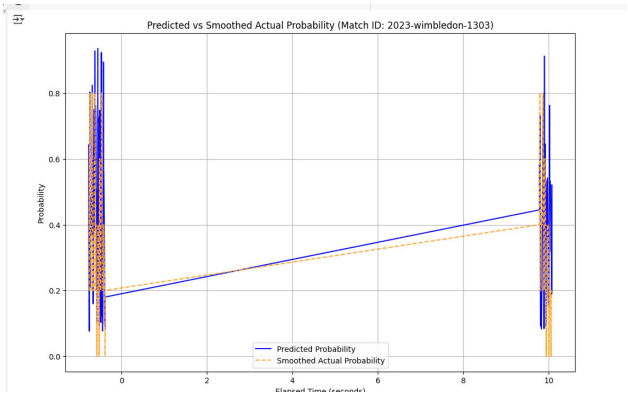


Figure 14

6 Strength and Weakness

6.1 Strength

Our algorithm demonstrates a high statistical capability in quantifying player momentum. The results show that the model can accurately predict differences in momentum between players in real time. By utilizing boosting algorithms, we were able to easily identify key variables that influence player momentum. This not only sheds light on the mechanisms behind the flow of the game but also provides valuable data for coaches to develop strategies. Compared to deep learning models, which typically require large datasets and lengthy training times, the boosting model proves to be more efficient, performing well with smaller datasets and consuming fewer computational resources. The model not only tracks the flow of the game but also provides actionable insights for in-match tactical adjustments. Data on key factors can assist coaches in optimizing player performance

or preparing for upcoming matches. Despite limited data, the model demonstrates a remarkable ability to capture the dynamics of the game, reflecting player performance under constantly changing conditions. The analysis results from the model serve as a valuable reference, helping coaches better understand the impact of variables on momentum and performance, ultimately enhancing strategic decision-making capabilities.

6.2 Weakness

One of the common limitations of boosting algorithm models is their tendency to overfit when using training data. This results in a lack of generalization, making it challenging for the model to accurately predict momentum changes without fine-tuning with specific data from each match.

Another weakness of boosting models is their strong dependence on the quality and relevance of the input dataset. Incomplete data or data that does not fully capture the key factors influencing momentum can prevent the model from learning the actual relationships effectively.

Finally, it is important to recognize that momentum primarily relates to the psychological state of the player, which match statistics may not fully capture. As a result, the model's predictions cannot be solely relied upon to provide guidance to the player.

7 Conclusion

In this study, we developed a model aimed at tracking the flow of a match point by point, identifying which player is performing better at any given moment, and quantifying the degree of difference in their performance. By employing the entropy weighting method, we quantified momentum based on key variables selected through meticulous feature engineering. This approach provided a structured framework to measure and analyze momentum shifts during a match, offering valuable insights into player dynamics.

Using t-test validation, we confirmed that player momentum has statistical significance. However, the inherent variability of matches indicates that a substantial portion of the fluctuations is random. Despite this randomness, our predictive model demonstrated strong performance and generalizability in estimating momentum differences between players. The results not only highlight the critical factors influencing momentum but also offer practical value for coaches in formulating in-game strategies and preparing for upcoming matches.

While the model has achieved promising results, it is not without limitations. One of the key challenges is the susceptibility to overfitting when trained on limited datasets. This overfitting reduces the model's ability to generalize across different matches, particularly when dealing with data that varies widely in quality or context. Furthermore, the model currently struggles to capture temporal dependencies within the input data, making it less effective at understanding the sequential relationships inherent in a match. To address these challenges, we recommend expanding the dataset by incorporating data from a diverse range of matches, tournaments, and player profiles. This would enhance the model's robustness and its ability to generalize across different scenarios.

It is also important to recognize that momentum is a complex construct that primarily reflects a player's psychological state—an aspect that statistics alone cannot fully capture. For instance, in some cases, players might intentionally reduce the pace of the game to conserve energy for sub-

sequent sets or to strategically disrupt their opponent's rhythm. Such nuanced behaviors highlight the limitations of relying solely on statistical models for tactical recommendations. Momentum, while a valuable metric, cannot account for these psychological and strategic elements, which are integral to a player's decision-making process.

Moreover, sports are fundamentally human-centered, with outcomes often influenced by factors beyond what statistical analysis can predict. While our model provides a useful analytical tool, it should not be viewed as a definitive solution for predicting match outcomes. Instead, it should be seen as a supplemental resource, aiding coaches and analysts in understanding match dynamics and optimizing strategies. Future research could explore the integration of psychological and physiological data to provide a more holistic view of momentum and its impact on performance.

In conclusion, our model offers an innovative way to analyze and quantify momentum during matches, providing valuable insights into performance dynamics. However, its role should remain as a supportive tool rather than a standalone solution. By acknowledging its limitations and continually refining its capabilities through additional data and interdisciplinary approaches, this model can serve as a powerful aid in the evolving field of sports analytics.

References

1. COMAP. *2024 MCM Problem C: Momentum in Tennis*. 2024. URL: https://www.contest.comap.com/undergraduate/contests/mcm/contests/2024/problems/2024_MCM_Problem_C.pdf.

2. Tianqi Chen and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. ACM, Aug. 2016. DOI: 10.1145/2939672.2939785. URL: http://dx.doi.org/10.1145/2939672.2939785.

3. IBM. *All England Lawn Tennis Club - Wimbledon*. 2024. URL: https://www.ibm.com/casestudies/all-england-lawn-tennis-club-ibm-ix?lnk=wmblp.

4. Jon Manuel. "Capturing Momentum in Tennis". In: *Opta Analyst* (2022). URL: https://theanalyst.com/na/2022/03/capturing-momentum-in-tennis/.

5. XDzzzzzZyq. *MCM 2024 C: Momentum in Tennis - Code Implementation*. 2024. URL: https://github.com/XDzzzzzZyq/MCM-2024-C.