Problem Chosen	2024 MCM/ICM Summary Sheet	Team Control Number
C		2403500

A Final for the Ages: Momentum or Random Effect?

Summary

It's a remarkable battle in the 2023 Wimbledon Gentlemen's final. Many fans deemed it as the rarest phenomenon that Djokovic lost tiebreaker in the final, while some believed that the winning of Alcaraz was due to his courage and resolution, often referred to as momentum. This report aims to unravel whether this intangible force played a pivotal role in the match's outcome using statistical models.

After scrutinizing the raw data, we address missing values by employing both mode and mean imputation methods based on the data type. Then, we choose random logistic regression approach to select more important features from the dataset and find several quite significant features such as "server", "break point", "winner", "points", "games", "rally count" and "victor".

First, we focus on the quantification of momentum and explore its correlation with player's success. We define momentum as the probability of a player winning the next point and construct new, well-suited features based on the ones selected from the dataset. Then, we quantify momentum based on logistic regression. The fitting result of the model shows 100% for this binary classification problem, indicating that our idea is reasonable and the classification model is accurate. To delve into the correlation between the quantified momentum and player's success, we calculate the PCCs and the result value 0.7471 indicates a strong correlation between the difference in momentum among two players and their probability of winning, which strongly suggests that Alcaraz's success was not a random effect.

Next, we explore the swings of momentum and quantify it by calculating the standard deviation of the momentum curve. To better model it, we construct new features of percentage based our original features. Using a random forest approach, we predict the swings of momentum and find the high fitness between the model and the actual. To further evaluate it, we calculate the MSE value of our model, which is 0.0056, indicating the model's excellent predictive capability regarding the swings of momentum.

Then, through the importance scores of the four main features obtained in random forest model, we know the main factors that influence the swings of momentum and come up with some advice after analyzing the factors. These factors include the percentage of successful serves, the success rate in converting break points into actual breaks, the percentage of winner points and unforced errors and Success rate in winning point at long rallies.

Finally, to verify the established models' stability and accuracy and further generalize the models, we collect relevant dataset of 1,446 matches in women's singles at the 2023 U.S. Open tennis tournament. Although it lacks of the data about "break point", the relevant findings based on our test set demonstrate the highly similar outcomes with the predicting results and the actual. For example, based on the new dataset, the PCCs of 0.7328 also indicates a strong correlation between the difference in momentum among two players and their probability of winning. What's more, the sensitivity analysis of the model also shows the good robustness of our models.

Keywords: Logistic Regression; Random Forest; Tennis; Momentum;

Contents

1 Introduction

1.1 Background

In the illustrious Wimbledon Gentlemen's final of 2023, a remarkable match demonstrated the essence of the sport's unpredictability and the shifting sands of momentum. Carlos Alcaraz, a 20-year-old rising star from Spain, defeated 36-year-old Novak Djokovic. The victory ended the unbeaten streak of Djokovic, one of the tennis triumvirate, at Wimbledon since 2013 and marked a significant shift in tennis dynamics. Using data from Wimbledon 2023 given to us by the COMPA officials, we will develop models to capture the flow of the match, quantify the elusive concept of momentum and predict swings in the match and so on, which help the players and tutors to know the magic of momentum and how to utilize it.

1.2 Restatement of the Problem

Based on the provided context and requirements, we are tasked with addressing the following challenges:

- **Model Construction & Momentum Quantitation**: Construct a model to capture the flow of play as points unfold across matches. This model should indicate which player is performing better at any given moment and the extent of their superiority. Based on the established model or metrics, analyze if the shifts in play and consecutive successes are random or if momentum significantly influences the game.
- **Finding Key Indicators**: Identify and define crucial indicators for measuring match momentum and flow, analyzing how these indicators evolve over time to reflect player performance and the flow of the play.
- **Testing Model and Generalization**: Apply the constructed model to additional matches to evaluate its predictive accuracy regarding match dynamics and swings. Assess the model's performance and the need for integrating new factors for improvement. Explore the model's applicability across different matches.

1.3 Our Work

Considering the background and realistic problems, our work mainly includes the following:

- Through data processing and feature engineering, we selected main features from dataset by using the Randomized Logistic Regression in Python, and constructed several well-suited features for the next problems to be addressed.
- Then, we utilized the logistic regression model to explore, analyze and predict the quantified vogue concept—momentum, which is defined as the probability of the player winning the next point. And explored the correlation between momentum and a player's success.
- Next, we explored the swings of momentum with random forest model and relevant plots. In this process, we also found the main factors that influence the swings of momentum and given some meaningful advice to players based on our findings.
- Finally, we tested the established models using open source dataset in Github and the test outcomes verified our models' robustness, demonstrating consistent performance across ordinary scenarios.

The whole modeling process can be simplified to a flow chart as follows:

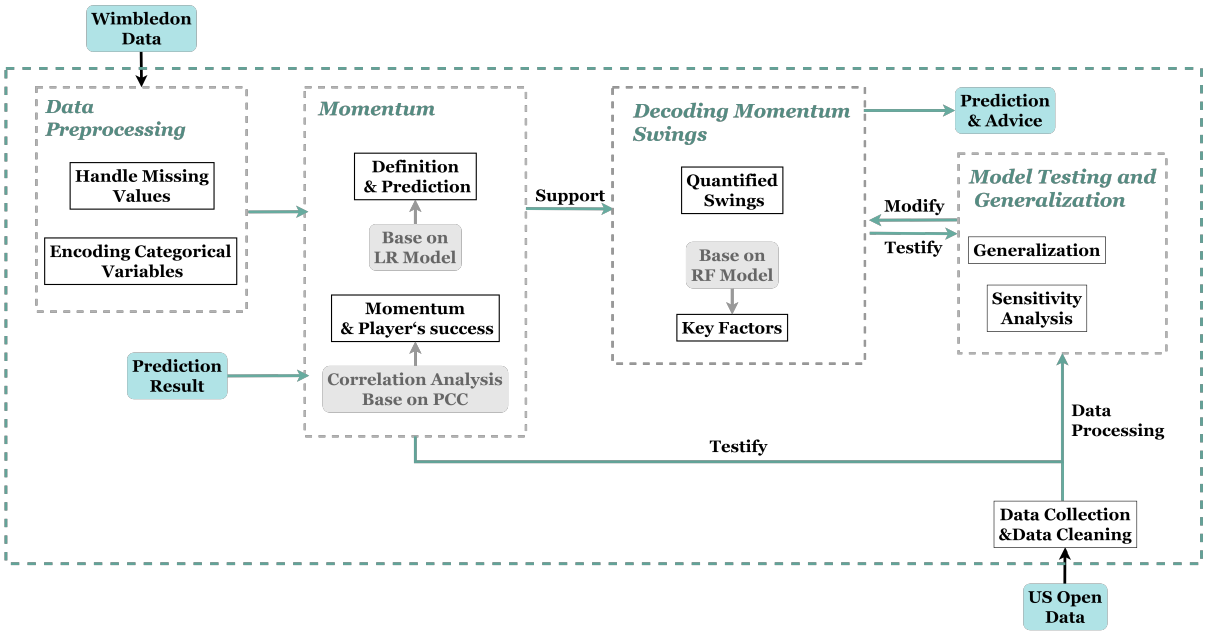

Figure 1: The Workflow of Our Approach

2 Assumptions and Notations

To simplify the problem, we make the following basic assumptions, each of which is properly justified.

Assumption **1:** **Momentum exists and can be quantified, i.e., in sports, the advantage of one side in the score can affect the performance and outcome in the game.**
Justification: This assumption is based on a common phenomenon in sports, where one team or player demonstrates a relative advantage over a period of time, which tends to have a knock-on effect in the game, affecting the form and outcome of both sides.

Assumption **2:** **There is a linear relationship between the feature variables and the log odds of the probability of an event occurring.**
Justification: The momentum is obviously affected by the many factors in sport match. In fact, we extract the main features from the dataset through feature selecting, but in order to simplify the model, we assume the correlation between them.

Assumption **3:** **Swings in the match are correlated with a player's success.**
Justification: This assumption is based on the directly observed connection between fluctuations in sports matches, indicating that significant swings in player often influence their ultimate success.

3 Data Processing and Feature Engineering

3.1 Data Preprocessing

1. **Handling Missing Values**

Table 1: Notations used in this paper

Symbol	Description
S_w	Success rate in winning points while serving
R_w	Success rate in winning point when returning serves
BP_w	Success rate in converting break points into actual breaks
N_w	Success rate in winning point at the net
LR_w	Success rate in winning point at long rallies
S	The percentage of successful serves
W_u	The percentage of winner points and unforced errors
D_f	The percentage of double faults

*There are some variables that are not listed here and will be discussed in detail in each section.

After an initial examination of the datasets, it turns out that there were some missing values but no outlier. Considering that speed is a continuous variable, we use the mean to fill in the missing values; while the other variables are discrete, we use the mode to fill in the vacant values. The missing values' details are shown in the table below:

Variables	speed_mph	serve_width	serve_depth	return_depth
Count	752	54	54	1309

Table 2: Statistics on the Number and Type of Missing Values

2. **Encoding Categorical Variables**

To facilitate model computation, we transform the categorical string variables "serve_width", "serve_depth" and "return_depth" into numerical codes. This encoding process involves mapping each unique string label to a numerical identifier, which allows us to integrate this non-numeric data into our logistic regression model as quantitative factors. We use Pandas in Python and then One-Hot Encoding method to make this conversion.

Variables	winner_shot_type	serve_depth	return_depth	serve_width
Encoding	F→1	CLT→1	D→1	B→1
	B→0	NCLT→0	ND→0	W→5

Table 3: Encoding String Variables in Dataset

3.2 Feature Selection

To select main relevant features for the model, we utilize Random Logistic Regression (RLR) in Python. This approach involves randomly sampling subsets of features and fitting a logistic regression model to each subset. By repeatedly performing this procedure, we can assess the stability and significance of each feature can be determined based on its contribution to the model's predictive accuracy. Features that are consistently chosen across multiple random subsets are deemed more important and are included in the final model. We screen these key features from the dataset, including "server", "serve_no","point_victor", "point_no", "game_victor", "game_no","p1 & p2_break_pt " and "p1 & p2 _break_pt_won".

4 Quantifying Momentum: A Logistic Regression Approach

4.1 Defining Momentum

Defining momentum in sport can be quite a challenge due to the intricate nature of the sport. Generally, momentum in tennis refers to the advantageous psychological and physiological boost that a player experiences. A psychological boost entails positive changes in cognition, such as increased self-efficacy, motivation, and attention. On the other hand, a physiological boost involves positive changes in behavior, including activity level, pace, posture, and frequency. Therefore, we adopt the concept of momentum in tennis, which encompasses both psychological and physiological effects. Our definition aligns with previous definitions of momentum in sports, as outlined in the multidimensional model of momentum by Taylor and Demick(1994) [1]. Therefore, we formulate momentum as a concept that encompasses both psychological and physiological effects.

We define momentum as the probability of winning the next point and apply the Logistic Regression (LR) model to predict the probability. Based on previous assumptions, the selected feature variables directly reflect the performance of a player at a given moment of the match, i.e., there is a linear relationship with the log odds of the probability of the event occurring. As a binary outcome, a player's win or loss of a point must coincide with the total points changing. Considering LR model's fast training time and being less prone to over-fitting compared to more complex models, we choose LR model to quantify the dynamics of momentum by translating these real-time performances into the win rate of point and to determine the relative relevance of the match data that contribute to victory.

4.2 Quantifying Momentum based on Logistic Regression

Logistic Regression [2] is a statistical machine learning model primarily used for binary classification tasks. It falls under the category of supervised learning algorithms and aims to predict the probability of a binary outcome based on one or more independent input variables (features). The fundamental concept behind LR is to establish a relationship between the input features and the binary outcome by utilizing the logistic sigmoid function (the equation in formula 1). This function transforms a linear combination of the features into a probability value [6], ensuring that the predicted probabilities range from 0 to 1 (as shown in figure 2).

$$\sigma(z) = \frac{1}{1+e^{-z}} \tag{1}$$

Indeed, the sigmoid function is used to predict the probability of a binary outcome, where $z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n$, β_0 is the intercept, $\beta_1, \beta_2, \ldots, \beta_n$ are the regression coefficients, and $x_1, x_2, \ldots, x_n$ are the input feature variables.

Based on the above understanding of logistic regression, we apply it to capture the flow of play as points occur by quantitative momentum and identify which player is performing better at a given time in the match. The detailed steps and results analysis are presented as follows:

Step 1 Defining Variables & Constructing Feature Variables

First of all, we define the model's feature variables X_i (independent variables) and target variables Y (dependent variables). X_i refer to the feature variables we would construct based on the extracted feature variables from the original dataset in Chapter 3 Feature Engineering part. Y refers to whether or not the player wins on the next scoring point; if the player win, Y equals 1, otherwise Y is 0.

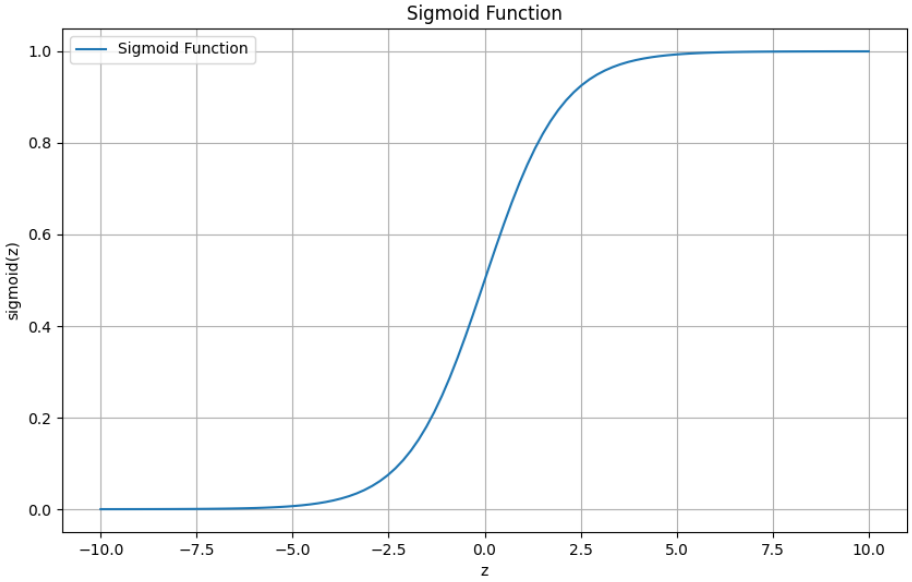

Figure 2: Sigmoid Function Graph

Then, to more accurately portray the match's momentum, we construct four feature variables X_i (i=1,2,3,4), including serve win rate, return win rate, break point win rate, and whether the player is serving. These factors are pivotal in influencing the player's probability of winning the next point in a match.

Step 2 Feature Standardization

To ensure a balanced weighting of feature influences in the logistic regression model, we employed feature standardization. This step involves subtracting the mean and dividing by the standard deviation to transform the features into a standard normal distribution with a mean of 0 and a standard deviation of 1.

Standardization helps eliminate scale differences among features, preventing certain features from having an excessive impact on the model. It also ensures a fair and accurate allocation of weights to each feature. This process improves the performance and stability of the model, allowing it to effectively learn the relationships between features and make predictions.

Step 3 Splitting Dataset

For the subsequent testing of the model, we divide the dataset into two subsets, the training set and the test set. And we use simple random split approach to allocate 80% to the training set and 20% to the test set. The model is then trained on the training set and evaluated on the test set, providing a clear measure of its predictive performance.

Step 4 Constructing Model

The equation for the conditional probability P, which represents the probability that the outcome (event) Y is 1 (means the player wins the next point), would be as follows:

$$P(Y=1) = \frac{1}{1+\exp\left(-(\beta_0+\beta_1 S_w+\beta_2 R_w+\beta_3 BP_w+\beta_4 S)\right)} \tag{2}$$

where

- $Y=1$ represents the player wins the next point in a tennis match.
- β_0 is the intercept, indicating the log-odds of the outcome when all input features are zero.

- β_1, β_2, β_3, β_4 are model parameters that correspond to the coefficients of the feature variables S_w, R_w, BP_w, S respectively. These coefficients indicate the degree of influence of the features on the output probability.

Step 5 Model Fitting and Testing

We fit the logistic regression model using the training set to learn the parameters of the model. After the completion of the model fitting, the model is able to predict the probability of a player winning the next point in a match based on their feature inputs.

In this binary classification problem, we use 0.5 as a threshold to determine the positive and negative categories, i.e., the probability that a player will win the next point in a match. If this probability is greater than the threshold, the model predicts a positive category (the player wins the next point), otherwise a negative category (the player loses the next point).

To evaluate the performance of the model on unknown data, we validate it using a test set and calculate the accuracy of the model on the test set. Accuracy is a commonly used metric for evaluating classification models, which indicates the ratio of the number of samples correctly predicted by the model to the total number of samples.

During model testing, we achieved a 100% accuracy, indicating that the model successfully classified all samples in the test set without any misclassifications. This high accuracy may be attributed to the effective feature selection and model parameter tuning. The binary classification distribution model, combined with our approach, facilitated accurate classification on this specific dataset.

Step 6 Visualizing the Dynamic Change of Momentum

In order to visualize the dynamic momentum change of both players in a match, we chose the momentum difference of the two players to present the change of the flow in a match, which also points the successful break points to visualize the relationship between momentum and critical moments, as shown in the figure below.

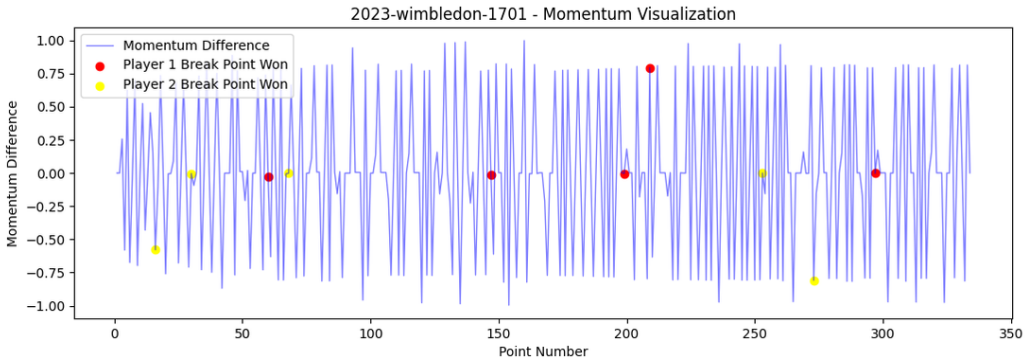

Figure 3: The Change Flow of Momentum Difference with Point Number

4.3 Correlation between Momentum and Player's Success

Although we explored the relationship between momentum and points in the previous subsection, we found that their correlation is not enough to find the association between momentum and player success by calculation. By searching for relevant studies, we found the perspective that the momentum effect from prior points seems to have a short memory. Therefore, we adopt game as new research object to explore its relationship with the difference of the predicted probability. The detailed steps and results analysis are presented as follows:

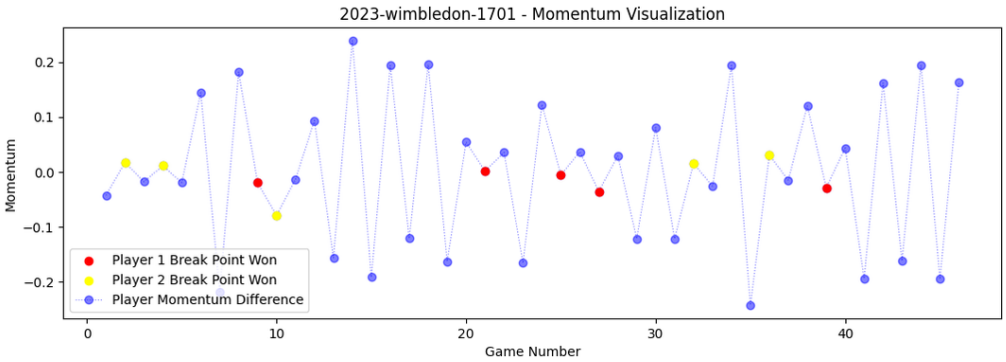

Figure 4: The Change Flow of Momentum Difference with Game Number

Step 1 Data Processing

The change of the momentum is determined by the players and many related factors, while the success or failure involves both sides. So the momentum of total points of each game for two players is averaged to represent the momentum of this game, and then the difference is made to judge the correlation with game victor.

Step 2 Correlation Analysis

Through correlation analysis, we found a strong correlation (Pearson Correlation Coefficient of 0.7471) between the difference in momentum among two players and their probability of winning. This result indicates that a player's success is not a random occurrence but significantly associated with their momentum.

Step 3 Result Visualization and Interpretation

To accurately interpret our results, we utilize a violin plot and a decision region plot. The violin plot synthesizes the density distribution of numerical data and box plot, with its width representing the data density. This visualization underscores that positive momentum differences, where *player*1 surpasses *player*2, generally correspond to a Game Victor outcome of 1. In contrast, negative momentum differences favor *player*1 with a Game Victor outcome of 2.

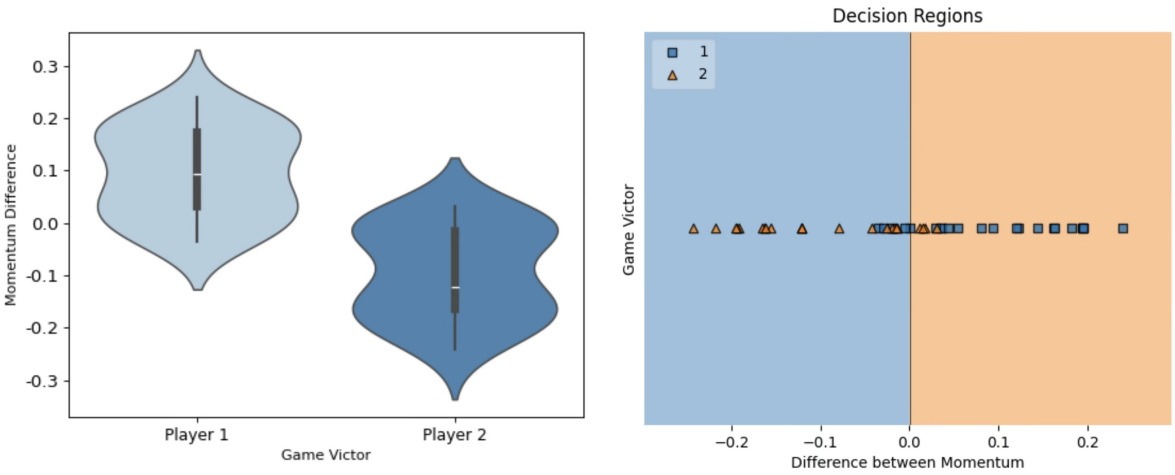

Figure 5: Plots of Violin and Decision Region

The decision region plot offers detailed insights, marking each player's wins against momentum differences. Distinct clusters of squares for *player*1 and triangles for *player*2 illustrate who's ahead, while colored areas predict winners by momentum.

Our findings, illustrated by these plots, highlight the intricate balance of power within a match and how momentum, while elusive, is a discernible factor influencing the game's tide.

5 Decoding Momentum Swings: A Random Forest Approach

In the previous section, we quantified momentum with the probability of winning the next point using logistic regression model and demonstrated that the success of a player is not a random effect but rather correlated with their momentum. In this section, we want to explore the factors that influence the swings of momentum so that to help players improve their performance and the probability of winning in match.

5.1 Quantifying the Swings of Momentum

We quantify the swings of the momentum by training a random forest model to predict the fluctuation of the momentum in a particular match. From the trained random forest model, the importance of each selected feature in influencing the fluctuation of momentum can be obtained. It helps players to make rational and effective decisions when facing different opponents in a match.

Taking a step further, we quantify momentum swings by calculating the standard deviation of the momentum curve. The standard deviation of the momentum curve reflects the degree of dispersion in momentum changes. A larger standard deviation indicates more significant fluctuations in momentum during a match, and vice versa. The detailed steps and results analysis are presented below:

Step 1 Data Processing

We sorted the data chronologically and grouped it by match identifiers. After selecting a specific match, we processed the cumulative statistical data for two players, including S, BP_w, LR_w, N_w, W_u and D_f, which are explained in previous chapter.

Step 2 Feature Selecting

We use the processed data as model features. By analyzing the importance of the features, we retained the features that are crucial to the prediction of potential fluctuations and screened out the features that have little impact on the prediction of potential fluctuations. In the end, the four most important features were selected: S, BP_w, W_u and LR_w.

Step 3 Model Selecting

The Random Forest model [3] is a collection of multiple decision trees. A larger number of trees tends to improve accuracy in this model, which can be utilized for both classification and regression problems while avoiding overfitting. Swings in momentum can be affected by multiple factors, and random forests excel at capturing non-linear relationships, making them well-suited to handle complex data patterns.

The Random Forest model incorporates two key concepts beyond merely averaging prediction trees. The first concept involves random sampling of training data when constructing trees. The second concept is the selection of a random subset of features [7] during node splitting. The partial workflow chart of the RF model is shown in the above Figure 7.

Step 4 Splitting Dataset & Model Training

For the subsequent testing of the model, we divide the dataset into two subsets, the training

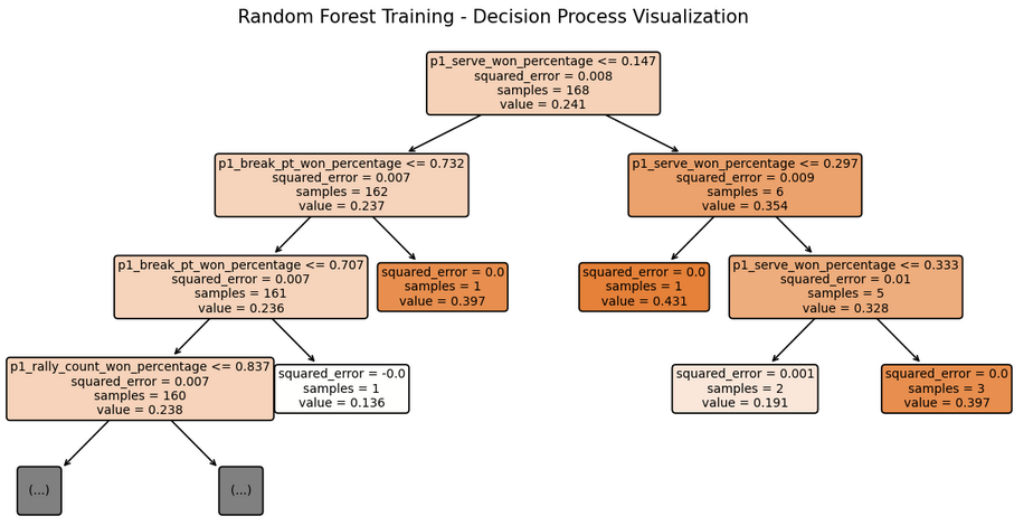

Figure 6: The First Decision Tree in Random Forest

set and the test set. And we use simple random split approach to allocate 80% to the training set and 20% to the test set.

In the previous step, we selected four distinct features to train our model. By employing the approach that combines Bootstrap sampling with random feature selection, we developed an ensemble of decision trees. Each tree is trained on varied subsamples and subsets of features, collectively contributing to the predictive model that forecasts swings in momentum.

Step 5 Model Evaluation

1. MSE Analysis

   In the evaluation process, we used the test set to verify the performance of the model. We calculated the Mean Square Error (MSE), a measure of the model's prediction error. A smaller MSE value indicates better performance. The MSE value of our model is 0.0056373334 in the data of this selected match, which indicates that our model achieves good results in predicting the swings of the momentum, i.e., the predicted value of the model is very close to the true value.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(P_i - \hat{P}_i)^2 \tag{3}$$

2. Residual Analysis

   To get a clear view of the model's performance on the test set, we generated a residual plot. This plot visualizes the variance between the standard deviation of the actual momentum fluctuations and the model's predictions, helping us to identify whether the model has large prediction errors at certain points. The plot reveals that the residuals are randomly scattered around zero and no obvious trend, signifying a strong fit of our model.

3. Visualization and Result Analysis

   We visualize the standard deviation of the actual potential fluctuations versus the standard deviation of the model's predicted potential fluctuations, as well as the significance bars of the features. In the figure below, it is evident that the model's predicted fluctuations closely align with the actual ones, indicating that the model can predict the swings of the momentum well. Thus, our model could provide some meaningful statistics for players and analysts making informed an advanced preparation and choices for future matches.

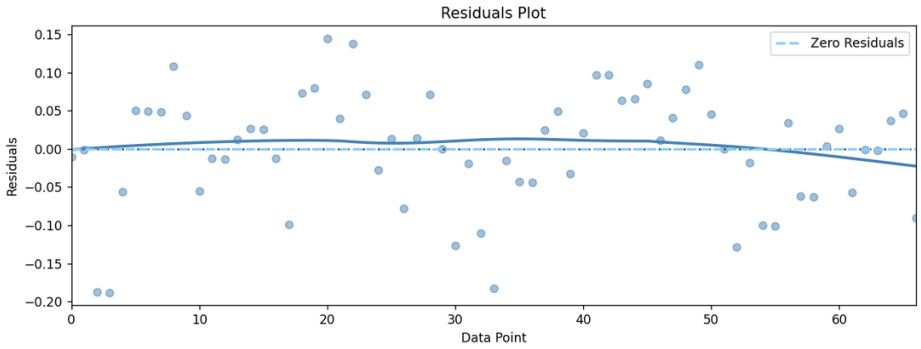

Figure 7: Residual Plot

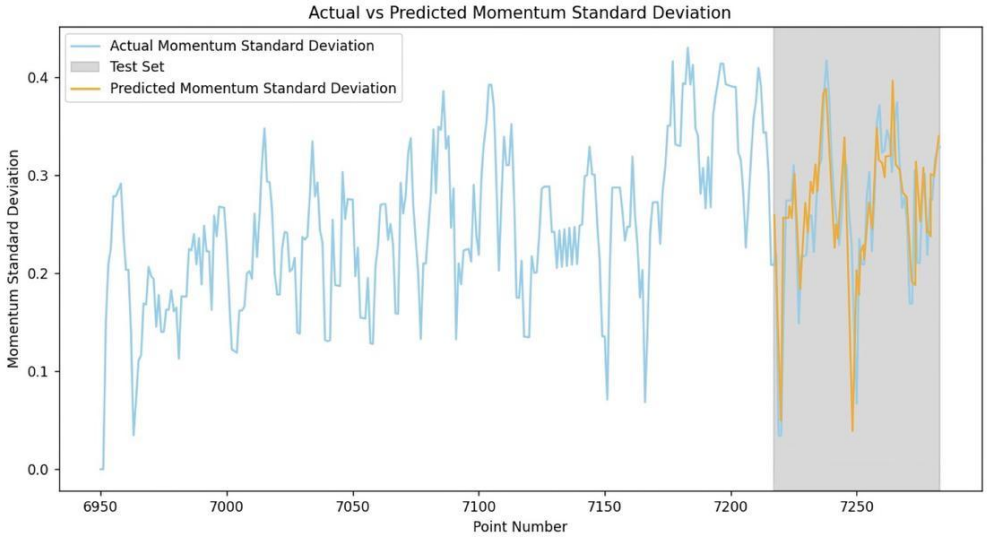

Figure 8: Model Fitting and Predicting Plot

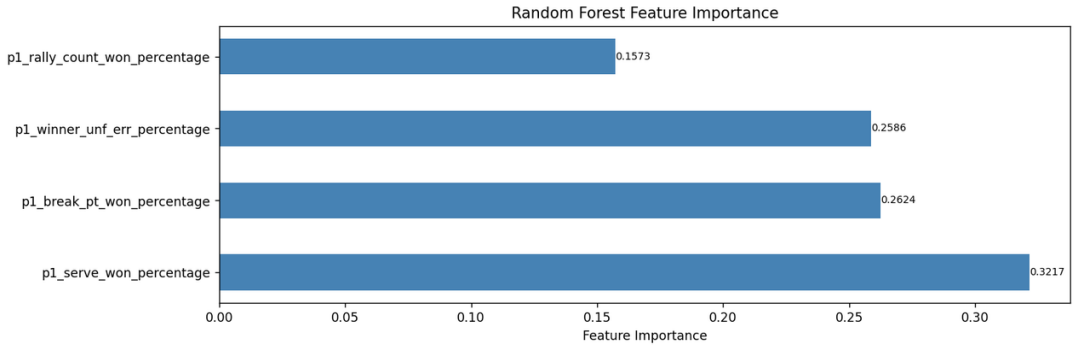

Figure 9: Feature Importance Degree

5.2 Factors Influencing Momentum Swings Analysis

From the findings of the previous subsection, we've got the main factors (features) that are most related with the swings of momentum. The importance scores of the four main features in this model can be seen from the feature importance bar chart above (Figure 9). The specific analysis of the factors is as follows:

Factor 1: The percentage of successful serves (S)

As evident from the graph, this factor stands out as the mst crucial. Strong and

precise serves have the potential to secure direct points or create advantageous court positions, thereby exerting pressure on the opponent both mentally and in terms of scoring. Players proficient in serving typically have better control over the match's tempo.

Factor 2: Success rate in converting break points into actual breaks (BP_w)
It is an important indicator of a player's ability to capitalize on opportunities at critical moments. Players with a high break point success rate can win points on their opponent's serve, which can be crucial in changing each other's momentum when the match is on the line[4].

Factor 3: The percentage of winner points and unforced errors (W_u)
This factor encompasses the player's ability to consistently execute winning shots while minimizing unforced errors. A higher ratio of winners to unforced errors suggests that the player is not only playing aggressively but also maintaining a level of control that prevents giving away easy points. This balance is indicative of a player's ability to dominate play and can significantly swing the momentum in their favor.

Factor 4: Success rate in winning point at long rallies (LR_w)
This factor evaluates a player's endurance, consistency, and baseline play during extended exchanges. A high success rate in long rally points indicates an advantage in attritional battles, which is particularly crucial during tense or pivotal stages of a match. Such a capability not only tests the physical stamina of the player but also their mental fortitude, as winning these protracted exchanges can significantly boost a player's confidence and disrupt the opponent's rhythm, contributing to shifts in momentum.

To sum up, each factor plays a critical role in shifting the dynamics of a match, whether through direct scoring, exploiting crucial opportunities, maintaining aggressive yet controlled play, or enduring and outlasting the opponent in long exchanges. These elements together shape the ebb and flow of the game, marking the difference between winning and losing.

5.3 Strategic Tips for Tennis Players: Mastering Momentum Swings

Following the analysis of the factors influencing momentum swings, we provide the following advice to players in tennis match:

1. **Serve Mastery:** Invest in honing your serve technique. A powerful and accurate serve not only secures points but also imposes psychological pressure on your opponent. It serves as a strategic tool to dictate the match's pace and can often be the deciding factor in crucial games.

2. **Strategic Focus:** Develop the ability to concentrate during key points, such as break points or tie-breaks. This mental acuity can lead to winning important points and shifts in momentum which could alter the course of the match.

3. **Balanced Offense and Defense:** Strive for a harmonious blend of aggressive shots and controlled play. By reducing unforced errors while maintaining an assertive stance, you can force opponents into difficult positions without risking easy giveaways.

4. **Baseline Endurance Training:** Enhance your baseline rally skills and patience. Long rally success is vital, especially during high-stress periods of the match. Endurance, both physical and mental, is tested, and winning these rallies can be a significant psychological boost.

5. **Data-Driven Match Preparation:** Utilize our model to analyze match data and identify the key techniques and strengths for scoring. By understanding your own game as well as studying the opponent's strategies with our model, you can approach the match with a tailored plan to exploit their weaknesses and reinforce your advantages.

6. **Random Forest Analysis for Specific Matches:** Apply match data to our model to derive a feature importance score using Random Forest analysis. This insight can inform targeted improvements and tactical decisions that focus on the key strengths that contribute to scoring in that particular match.

In conclusion, we genuinely hope that our strategies will greatly boost players' performance and provide them with a competitive advantage in their matches.

6 Models Testing and Generalization

In this section, we delve into the testing and application of our logistic regression model nd random forest model, especially focusing on its performance in the context of missing break point feature and its predictive capabilities for other match scenarios, including a 2023 women's tennis match analysis.

6.1 Data Processing

We collected the test set for our models from Github website [5], which includes 1,446 matches in women's singles at the 2023 U.S. Open tennis tournament. After the initial checking of it, we found that the dataset is lack of "p1_break_pt_won" and some relatively unimportant features. Then we transformed the useful data types into the same types of the title given dataset to make a foundation for testing our models.

Visualizations created in code Part3, including line graphs and comparative analyses with official data, illustrated the similarity in momentum fluctuations and their increased frequency as matches intensify. However, due to missing data fields from the official dataset, visualizations lacked insight into the relationship between winning break points and momentum shifts.

6.2 Models Testing and Generalization

1. **Model Performance Test for LR Model**

   (a) Accuracy of the model:
   - Player 1's accuracy: 0.9512195121951219
   - Player 2's accuracy: 1.0
   - Overall prediction accuracy: 95.1

   This indicates that the model performs well on the new data, even without the break point feature. The accuracy scores demonstrate a satisfactory fit of the model.

(b) Correlation between momentum and match success:

- The Pearson Correlation Coefficient is 0.7328.
- It indicates a strong correlation between the difference in momentum and match success, affirming the non-random nature of this relationship.

(c) Visualization of momentum:

- Violin plot and decision region plot (figure 11) further validate our model's conclusions, showcasing a concentrated distribution of victories when momentum differences favor player 1, with the violin plot's width indicating high data density.
- The prediction of swings in momentum exhibit similarities with the original official data (figure 13).
- However, the visualization lacks the relationship between winning break points and momentum due to missing data fields.

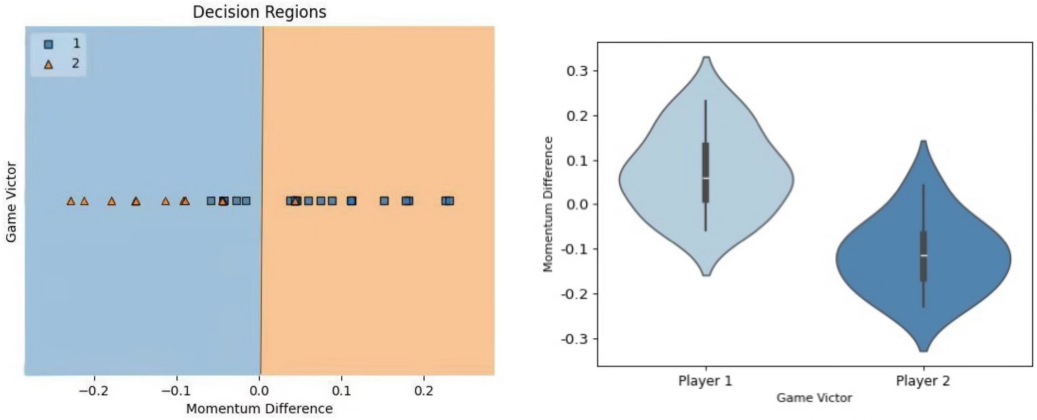

Figure 10: Plots of Decision Region and Violin

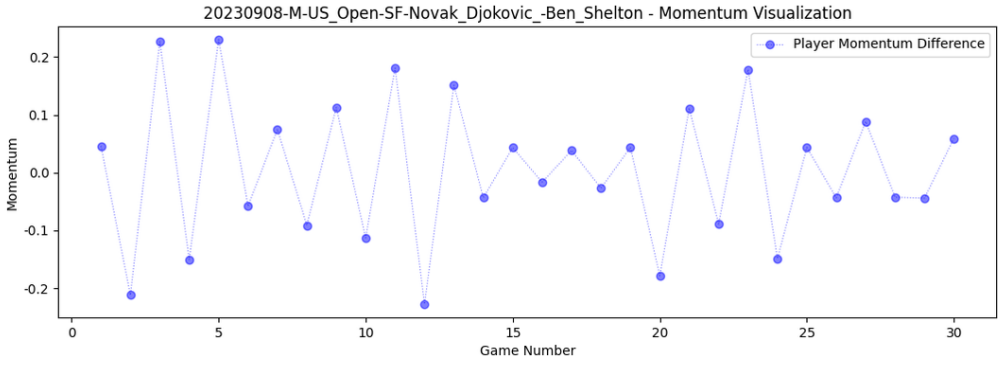

Figure 11: The Change Flow of Momentum Difference with Game Number

2. **Model Performance Test for BF Model**

- Since the newly collected data does not include any break point related information, the training of the new model only incorporates three features: serve success rate, ratio of winners to forced errors, and percentage of points won in long rallies.
- The trained model achieves an MSE of 0.006516292653929047, indicating excellent performance on the new data. The residual plot (figure 12) generated from the

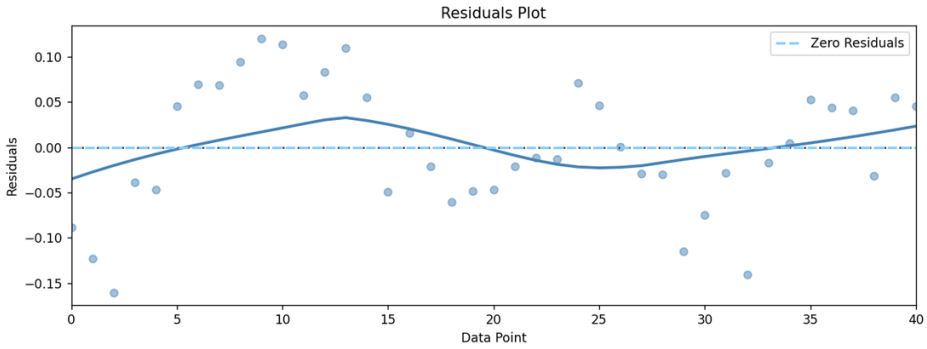

Figure 12: Residual Plot

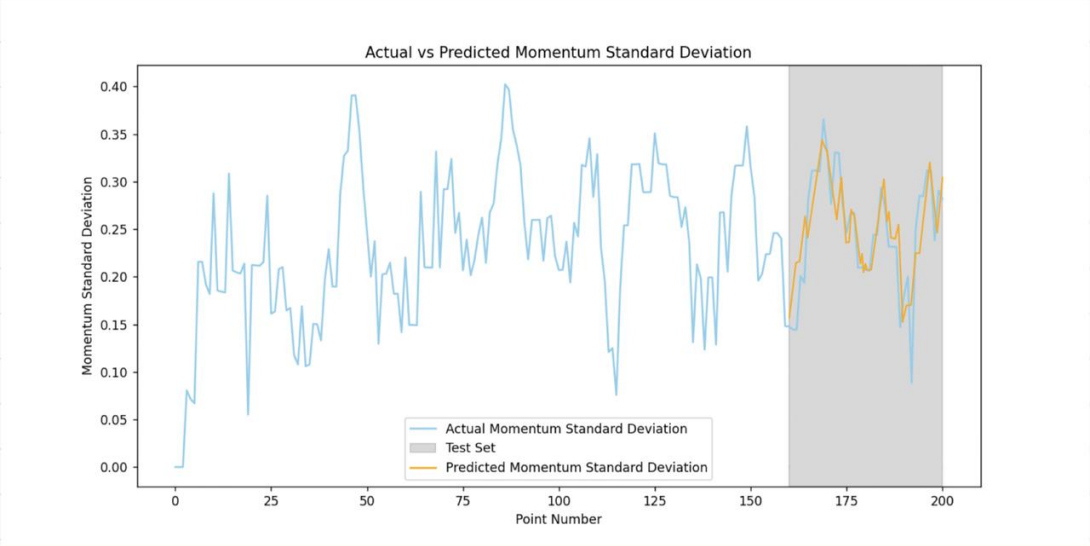

Figure 13: Model Fitting and Predicting Plot

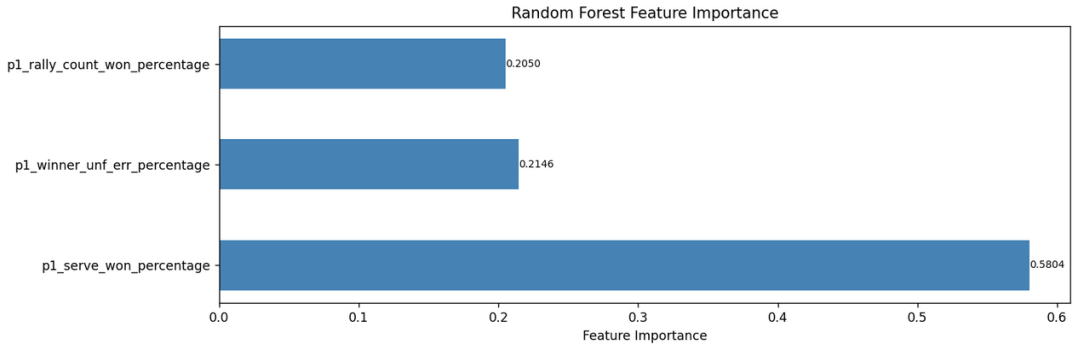

Figure 14: Feature Importance Degree

model shows that the residual points are randomly distributed around zero without any discernible trend. This further confirms the good fit of our model.

- The importance bar chart (figure 14) of features reveals the significance scores of serve success rate, ratio of winners to forced errors, and percentage of points won in long rallies in this model.
- Overall, the model's performance on the new data is excellent, highlighting its robustness and ability to predict momentum fluctuations effectively.

6.3 Sensitivity Analysis

In the application of the model for the women's singles at the 2023 U.S. Open tennis tournament, we had to remove the feature of break point conversion rate from both of our constructed models due to a lack of data related to break points.

After removing this feature, the LR model's accuracy in predicting player 1's momentum decreased from 1.0 to 0.9512. Similarly, the RF model's mean squared error increased from 0.0056 to 0.0065. These results indicate that the break point conversion rate plays a significant role in both models for accurately predicting momentum and capturing fluctuations in momentum. This validates the rationale behind our feature selection during model construction.

Furthermore, the fact that both models still perform well even after removing such an important feature demonstrates their robustness. It shows that our designed models can effectively make predictions even when certain key features are missing.

7 Model Assessment

7.1 Strengths and Limitations

1. **Strengths**
    - **High Correlation**: The model not only shows a strong correlation between momentum and success, but also smoothly measures the momentum of players, captures changes in the flow of the match, and effectively predicts the success of a player.
    - **Capture Main Features**: Key features (factors) influencing momentum shifts are unveiled through feature importance analysis within the random forest model, guiding strategic focus areas.
    - **Accuracy**: The model's accuracy is underscored by its capacity to predict fluctuations in momentum throughout the tennis match, providing strategic insights into the match's progression.
    - **Visualization**: The model provides visualizations of momentum swings and correlations between momentum and players' success enhance intuitive comprehension of the model's fit and the relationships between variables.
2. **Limitations**
    - **Linear Assumption**: Logistic regression presupposes a linear relationship between the features and the log odds of the outcomes, which might not always be valid.
    - **Feature Selection**: The features chosen might not fully capture all the relevant factors [5] affecting momentum and its fluctuations.
    - **Risk of Overfitting**: There is a potential risk of the model overfitting, particularly when the data available is limited.

7.2 Further Improvement

- **Data Augmentation**: Expanding the dataset through data augmentation techniques or by introducing additional relevant features can enhance the model's ability to generalize to new, unseen data, thereby improving its predictive accuracy.

- **Feature Engineering**: Investigating new features or combinations of existing features can enhance the model's ability to capture the dynamics of momentum, potentially revealing deeper insights into the underlying patterns of the game.
- **Regularization**: Applying regularization techniques helps in reducing overfitting and increasing the model's generalization capabilities by penalizing model complexity and encouraging simpler models that perform better on new data.

8 Conclusion

In conclusion, our study has made significant strides in quantifying momentum through logistic regression models, offering valuable insights into player performance over time. The proposition that 'the success of a player is not a random effect but rather correlated with their momentum' has been verified, reinforcing the robustness of our model.

Additionally, the successful application of the Random Forest model in predicting momentum swings and analyzing key influencing factors has not only yielded noteworthy predictive results on new data but also provided actionable decision-making suggestions for players. As we look forward, the model holds the potential for widespread application across diverse game scenarios, establishing a solid foundation for future research and practical implementation.

Memo

To: Coaching Team
From: MCM Team #2403500
Date: Feb, 5th, 2023
Subject: Insights into Momentum in Tennis and Strategic Advancements

---

Dear Esteemed Coaches,

In the journey of our recent exploration into the heart of tennis, we have embarked on a quest not just of numbers and analysis, but of understanding the soul of the game we cherish. Our study, a labor of love and intellect, delves into the concept of momentum, that elusive force that ebbs and flows like the tide, influencing the course of a match with its invisible hand.

Through the lens of logistic regression and random forest models, we've sought to quantify this phenomenon, to capture its essence and distill it into wisdom that can guide our players through the stormiest of matches. Momentum, as we've defined it, is the probability of winning the next point—a beacon that shines light on who holds the sway in the battle of wills and skills on the court.

Our findings weave a narrative that goes beyond mere statistics. They reveal the intricate ballet of mental resilience, the power of a well-placed serve, and the critical moments that can turn the tide of a match. It's a tale of human endeavor, of strategy and heart, and of the silent strength that resides within our athletes.

To you, the architects of champions, we offer these insights not just as data points, but as tools to sculpt the minds and talents of those under your guidance. Let them serve as a compass in your coaching, helping you to navigate the complexities of competition and to harness the whispering currents of momentum in your favor.

Our analysis reveals several key strategies for leveraging momentum in competitive play:

- Psychological Resilience: Training should include focus on psychological endurance, preparing players to withstand and capitalize on momentum shifts.
- Tactical Acumen: Coaches are advised to develop player acumen for recognizing and creating momentum shifts through strategic play, especially in serving and return games.
- Physical Conditioning: Emphasis on conditioning to sustain performance during momentum shifts can significantly impact match outcomes.
- Data-Driven Preparation: Employing our model's insights for opponent analysis and strategy formulation can provide players with a competitive edge.

We see this not just as a report, but as an invitation—to dialogue, to reflection, and to a shared journey towards excellence. Our research is a stepping stone, but it is your wisdom and experience that will bring these insights to life on the tennis court.

With deepest respect and in shared passion for the game.

Sincerely,
Teams #2403500

References

[1] Taylor, Jim, and Andrew Demick. "A Multidimensional Model of Momentum in Sports." Journal of Applied Sport Psychology, vol. 6, no. 1, 1994, pp. 51-70.

[2] Murphy, Kevin P. Machine Learning: A Probabilistic Perspective. MIT Press, 2012.

[3] Biau, Gérard, and Erwan Scornet. "A Random Forest Guided Tour." Test, vol. 25, 2016, pp. 197-227.

[4] Meier, Philippe, et al. "Separating Psychological Momentum from Strategic Momentum: Evidence from Men's Professional Tennis." Journal of Economic Psychology, vol. 78, June 2020, p. 102269.

[5] Noel, Jordan Truman Paul, et al. "A Comprehensive Data Pipeline for Comparing the Effects of Momentum on Sports Leagues." Data, vol. 9, no. 2, Feb. 2024, p. 29.

[6] Prasetio, D., and Harlili, D. "Predicting Football Match Results with Logistic Regression." 2016 International Conference On Advanced Informatics: Concepts, Theory And Application (ICAICTA), 2016, pp. 1-5.

[7] Smithies, T.D., Campbell, M.J., Ramsbottom, N., et al. "A Random Forest Approach to Identify Metrics That Best Predict Match Outcome and Player Ranking in the Esport Rocket League." Sci Rep, vol. 11, 19285, 2021.

[8] https://github.com/JeffSackmann/tennis_MatchChartingProject

Appendix

Code Block

Listing 1: Core Code

```
def calculate_metrics(df):
    # process useful data for the logic model
    df.sort_values(by=['match_id', 'point_no'], inplace=True)

    df['p1_serve_won_total'] = df[(df['point_victor'] == 1) & (df['server'
        ↪ ] == 1)].groupby('match_id')[
        'server'].cumcount()
    df['p1_serve_total'] = df[df['server'] == 1].groupby('match_id')['
        ↪ server'].cumcount() + 1
    df['p1_unserve_won_total'] = df[(df['point_victor'] == 1) & (df['
        ↪ server'] == 2)].groupby('match_id')[
        'server'].cumcount()
    df['p1_unserve_total'] = df[df['server'] == 2].groupby('match_id')['
        ↪ server'].cumcount() + 1
    df['p1_break_pt_won_total'] = df.groupby('match_id')['p1_break_pt_won'
        ↪ ].cumsum()
    df['p1_break_pt_total'] = df.groupby('match_id')['p1_break_pt'].cumsum
        ↪ ()
```

```
    # similar data processing for the player2:
    # df['p2_serve_won_total'] = df[(df['point_victor'] == 2) & (df['
        ↪ server'] == 2)].groupby('match_id')[.....

    df.replace([np.inf, -np.inf], np.nan, inplace=True)
    df.fillna(0, inplace=True)
    return df

def plot_match_momentum(df, selected_match_id):
    df['diff_momentum'] = df['p1_momentum'] - df['p2_momentum']

    # draw change of momentum in Point
    plt.figure(figsize=(40, 4))
    plt.plot(df['point_no'], df['diff_momentum'], label='Momentum
        ↪ Difference',
             color='blue', linestyle='-', alpha=0.5, linewidth=1)
    break_point_won_points_p1 = df[df['p1_break_pt_won'] == 1]['point_no']
    plt.scatter(break_point_won_points_p1, df.loc[df['point_no'].isin(
        ↪ break_point_won_points_p1), 'diff_momentum'],
                color='red', marker='o', label='Player 1 Break Point Won')
    break_point_won_points_p2 = df[df['p2_break_pt_won'] == 1]['point_no']
    plt.scatter(break_point_won_points_p2, df.loc[df['point_no'].isin(
        ↪ break_point_won_points_p2), 'diff_momentum'],
                color='yellow', marker='o', label='Player 2 Break Point Won
                    ↪ ')

    plt.xlabel('Point Number')
    plt.ylabel('Momentum Difference')
    plt.title(f'{selected_match_id} - Momentum Visualization')
    plt.legend()
    plt.show()

# encoding Non-Numeric Variables with Unique Hot Coding
def encode_categorical_features(df):
    encoder = OneHotEncoder(drop='first')
    categorical_columns = ['serve_width', 'serve_depth', 'return_depth', '
        ↪ winner_shot_type']
    df_encoded = pd.get_dummies(df, columns=categorical_columns,
        ↪ drop_first=True)
    return df_encoded

# Calculate the standard deviation of momentum swings over a rolling
    ↪ window
def quantify_momentum_swings(df):
    df['momentum_std'] = df['p1_momentum'].rolling(window=5, min_periods
        ↪ =1).std()

# Visualize a decision tree using matplotlib
def visualize_decision_tree(tree, feature_names, class_names):
```

```
    plt.figure(figsize=(12, 8))
    plot_tree(tree, feature_names=feature_names, filled=True, rounded=True
        ↪ , fontsize=8,
              class_names=class_names, max_depth=3)
    plt.title("Random Forest Training - Decision Process Visualization")
    plt.show()

# Correlation between Momentum Difference and Game Victor
def diff_momentum_and_game(df):
    # data processing...
    # grouped = df.groupby(['match_id', 'set_no', 'game_no'], as_index=
        ↪ False).agg({...})
    grouped['diff_momentum'] = grouped['p1_momentum'] - grouped['
        ↪ p2_momentum']
    grouped['game_p1'] = grouped.groupby(['match_id'], as_index=False)['
        ↪ game_victor'].transform(
        lambda x: (x == 1).cumsum())
    correlation1 = grouped['diff_momentum'].corr(grouped['game_victor']) #
        ↪  0.7472
    print(f'Correlation between Momentum Difference and Game Victor : {
        ↪ correlation1}')

    temp = grouped.loc[:, ['diff_momentum', 'game_victor']]
    X = grouped['diff_momentum'].to_numpy().reshape((-1, 1))
    y = grouped['game_victor'].to_numpy()
    # train a classifier...
    # draw decision regions...
    # draw violin plot...
    return grouped

# prediction of potential fluctuations using a random forest model
def predict_momentum_fluctuations(df):
    """ Random Forest Model """
    # data processing...
    # transform dataset to get 'p1_winner_unf_err_total' and so on...
    # df['p1_winner_unf_err_total'] = df[(df['p1_winner'] == 1) | (df['
        ↪ p1_unf_err'] == '1')].groupby......

    feature_columns = ['p1_serve_won_percentage', '
        ↪ p1_break_pt_won_percentage', 'p1_winner_unf_err_percentage',
                       'p1_rally_count_won_percentage']

    X = df[feature_columns]
    y = df['momentum_std']
    imputer = SimpleImputer(strategy='mean')
    y = imputer.fit_transform(y.values.reshape(-1, 1)).ravel()
    train_size = int(0.8 * len(X))
    X_train, X_test, y_train, y_test = X[:train_size], X[train_size:], y[:
        ↪ train_size], y[train_size:]
```

```
    # train Random Forest Regression Models
    model = RandomForestRegressor(n_estimators=200, max_depth=10,
        ↪ min_samples_split=5, random_state=42)
    model.fit(X_train, y_train)
    predictions = model.predict(X_test)
    tree = model.estimators_[0]
    visualize_decision_tree(tree, feature_names=feature_columns,
        ↪ class_names=['No Fluctuation', 'Fluctuation'])
    mse = mean_squared_error(y_test, predictions)
    print(f"MSE: {mse}")
    plot_residuals(y_test, predictions)

    # get feature importance and print
    feature_importances = model.feature_importances_
    feature_names = X.columns
    feature_importance_dict = dict(zip(feature_names, feature_importances)
        ↪ )
    sorted_feature_importances = sorted(feature_importance_dict.items(),
        ↪ key=lambda x: x[1], reverse=True)
    print("Feature Importance Scores:")

# prediction of momentum using a logistic regression model
def train_and_evaluate_model(df, selected_match_id):
    """ Logistic Regression Model """
    df = calculate_serve_total(df)
    df = calculate_metrics(df)
    selected_match_df = df[df['match_id'] == selected_match_id].copy()

    # For Player 1 (p1)
    p1_feature_columns = ['serve_win_percentage', 'return_win_percentage',
        ↪  'break_point_win_percentage', 'server']
    X_p1 = selected_match_df[p1_feature_columns]
    y_p1 = selected_match_df['point_victor'] == 1

    X_train_p1, X_test_p1, y_train_p1, y_test_p1 = train_test_split(X_p1,
        ↪ y_p1, test_size=0.2, random_state=42)

    scaler_p1 = StandardScaler()
    X_train_scaled_p1 = scaler_p1.fit_transform(X_train_p1)
    X_test_scaled_p1 = scaler_p1.transform(X_test_p1)

    model_p1 = LogisticRegression()
    model_p1.fit(X_train_scaled_p1, y_train_p1)

    predictions_p1 = model_p1.predict(X_test_scaled_p1)
    accuracy_p1 = accuracy_score(y_test_p1, predictions_p1)
    print(f"Player1's accuracy: {accuracy_p1}")

    selected_match_df['predicted_probabilities'] = model_p1.predict_proba(
        ↪ selected_match_df[p1_feature_columns])[:, 1]
```

```
# For Player 2 (p2)... similar to Player 1's process
selected_match_df['p1_momentum'] = selected_match_df['
    ↪ predicted_probabilities'].diff().fillna(0)
selected_match_df['p2_momentum'] = selected_match_df['
    ↪ p2_predicted_probabilities'].diff().fillna(0)

n_features = len(model_p1.coef_[0])
formula1 = f"Probability(Y=1) = 1 / (1 + exp(-({model_p1.intercept_
    ↪ [0]} + {' + '.join([f'{model_p1.coef_[0][i]}*X{i + 1}' for i in
    ↪  range(n_features)])}))) "
print(f"model mathematical formula{formula1}")
n_features = len(model_p2.coef_[0])
formula2 = f"Probability(Y=1) = 1 / (1 + exp(-({model_p2.intercept_
    ↪ [0]} + {' + '.join([f'{model_p2.coef_[0][i]}*X{i + 1}' for i in
    ↪  range(n_features)])}))) "
print(f"model mathematical formula{formula2}")

diff_momentum_and_game(selected_match_df)
plot_match_momentum(selected_match_df, selected_match_id)

# quantify and forecast momentum's volatility standard deviation
selected_match_df['momentum_std'] = selected_match_df['p1_momentum'].
    ↪ rolling(window=5, min_periods=1).std()
predict_momentum_fluctuations(selected_match_df)
```