Problem Chosen	2024 MCM/ICM Summary Sheet	Team Control Number
C		**2419212**

Uncovering Momentum Flow in Tennis: Can We Suddenly Rise Up?

Summary

Tennis events often bring tension and surprises to the audience. Comebacks from a disadvantageous position are often the most exhilarating moments. To explore whether there exists momentum in tennis that can significantly affect a player's performance, we propose and establish the **E**xponential Moving Average (EMA) Enhanced **T**ree-Based **V**oting (ETV) model to prove and quantify momentum and predict the turning points in the flow of play.

The ETV model is divided into two main stages. The first stage is **match momentum (M^2) score** extraction, aiming at proposing a new metric to measure each player's momentum. We divide momentum into three components: set factor, game factor, and motivation of a player. Specifically, we first calculate the **Spearman correlation coefficient** between the score and other features. Then, we select the **top-k** features with the highest correlation for feature fusion. Finally, the three fused features of components are weighted to obtain the match momentum score.

The second stage is match flow swing prediction, aiming to determine whether the current point is a turning point in the match. Specifically, we first perform **Exponential Moving Average** processing on M^2 to remove artificially introduced noise and obtain time series features sensitive to recent changes, which is denoted as EM^2. Then, we mark the turning points of momentum and apply four **tree-based machine learning models**, including Decision Tree, Random Forest, XGBoost, and LightGBM, for classification prediction. The final joint prediction result is obtained through a **voting method**. We evaluated the ETV model and addressed the given problems.

For Problem 1, we visualize the entire process of a match in 3, mark the number of occurrences of important factors (such as aces), and confine the proposed M^2 within [0,1]. This effectively quantified the performance differences between players.

For Problem 2, we predict whether a player would get a point using M^2 and compare the results with a random process, conducting a **Wilcoxon Signed-Rank Test**. The results showed a **p-value of 8.75e-51**, indicating that momentum plays an important role in the match.

For Problem 3, we use ETV to predict turning points in all matches of the given dataset. ETV achieves an **accuracy of 95.81%** and an **F1 score of 95.14%** on the test set, demonstrating ETV's excellent performance. We also compare the feature importance of the model, and the results show that over 60% of the features are introduced by our model (such as EM^2). In addition, we conducted an **IQR statistical analysis** on the first-order difference series of EM^2 to suggest strategies for athletes against different opponents.

For Problem 4, we test ETV on three additional datasets to verify its generalizability, including Women Tennis, Women Fencing, and 2016-World-Cup. The results demonstrate that the ETV model performs excellently, except on the World Cup dataset. We speculate this is due to the absence of individual data features within the team.

We prepare a memo summarizing our results and offering suggestions to tennis players and coaches. We also conduct a **sensitivity analysis** to reveal the sensitivity of ETV to key parameters in different working environments, summarize the advantages and disadvantages of the model, and further explore future research directions.

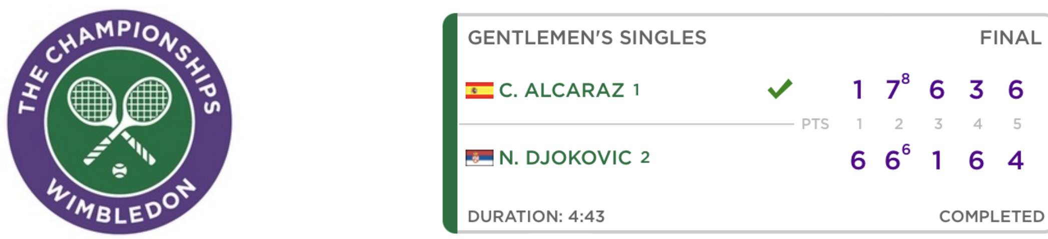
Keywords: Feature Fusion; Match Momentum Score; Exponential Moving Average; Voting

Contents

1 Introduction

1.1 Background

The Wimbledon Championships, as one of the four Grand Slam tennis tournaments globally, has held a distinguished status since its establishment in 1877, making it the oldest and most prestigious event in tennis history. [AI: 6]In the 2023 Wimbledon Men's Singles final, 20-year-old Spanish emerging talent Carlos Alcaraz defeated 36-year-old Novak Djokovic, bringing an end to Djokovic's consecutive victories at Wimbledon since 2013. The match witnessed dramatic fluctuations, especially during moments when Djokovic seemed to have the upper hand, highlighting the crucial role of **momentum** in the game.



(a) The Wimbledon Championships (b) Final score

Figure 1: Wimbledon men's singles final scores

Therefore, researching how to harness and understand player **momentum** holds significant importance. This study aids coaches in making more effective tactical adjustments, enhancing players' psychological preparedness and teamwork, and providing valuable guidance for personalized training and player development.

1.2 Restatement of the Problem

After thorough in-depth analysis and research on the background of the problem, we can specify that our article should cover the following aspects:

- Develop a model to track the progression of play during tennis matches, identifying the player's performance and considering the serving advantage. Provide a visual representation of the match flow.
- Utilize our model and metric to evaluate the significance of "momentum" in tennis matches. Assess the claim that swings in play and success runs are random, addressing the skepticism of tennis coaches.
- Develop a model to predict shifts in match flow using available data, identifying relevant factors.
- Offer advice for a player entering a new match against a different opponent based on historical momentum differentials.
- Test our developed model on additional matches to assess its predictive accuracy in anticipating match swings. Identify factors for potential improvement and evaluate the model's generalizability to various conditions.

1.3 Our Work

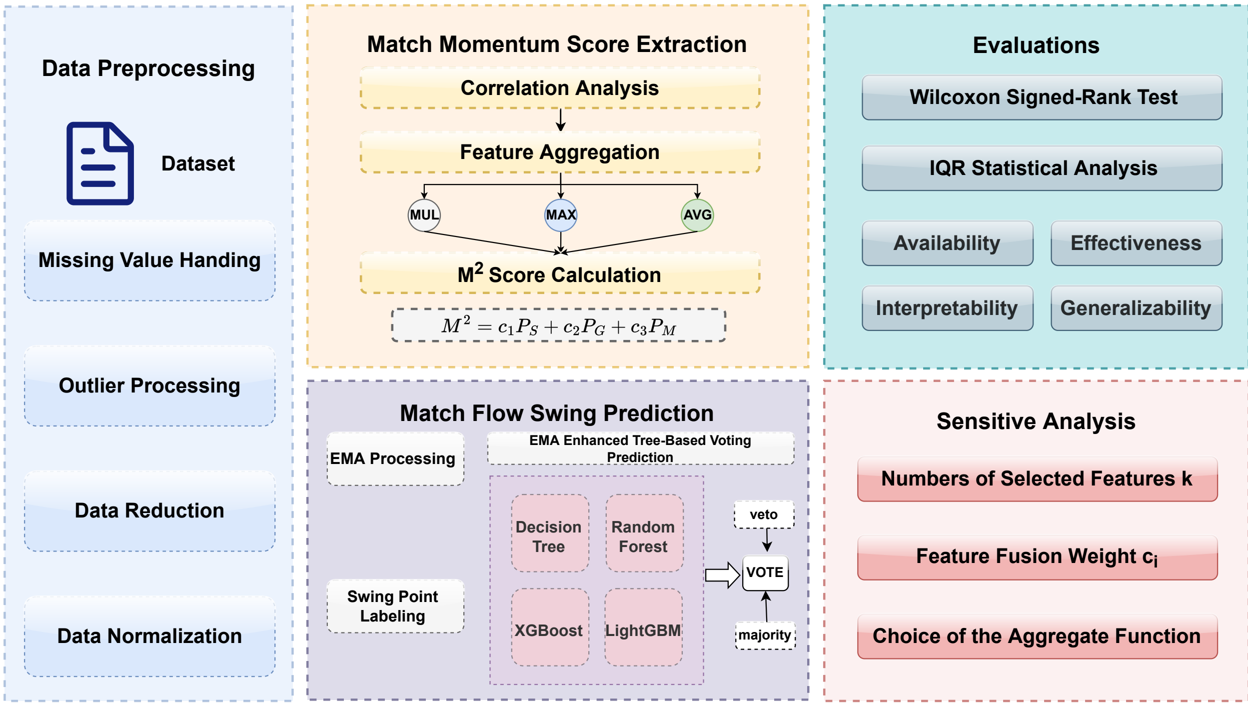


Figure 2: Our work.

2 Assumptions and Notation

In this section, we have made several overarching assumptions and explained the rationale behind these assumptions. Additionally, we have provided a list of the mathematical symbols used throughout this paper.

2.1 Assumptions and Justifications

In actual practice, many complicated conditions may affect the output of the model. In order to make the model more stable and less complex, the following assumptions and their justifications are incorporated.

- **Assuming that each data point is observed independently.** The data collection process reveals individual observations and value sampling for each factor. This sampling method helps ensure that the observed value of each data point is not affected by other factors, providing a wider application for statistical methods.

- **Assuming that the roles of player 1 and player 2 do not impact momentum.** Given the lack of documented data on individual differences, we assume that the differences between players 1 and 2 in physical and mental attributes are negligible.

- **Assuming that the change in momentum occurs after scoring.** Based on the continuous nature of momentum changes in real life. To simplify the model's handling of continuous changes, this study chooses the moment after a scoring event as the point in time for the change in momentum, thereby converting the continuous change in momentum into discrete events. [AI: 3]

2.2 Notation Description

In order to make formulas and equations more intuitive and to present our article concisely, we use these abbreviations or notations to represent different entities. As shown in Table1 below, we describe the symbols used in our formulas.

Table 1: The symbol description in the formula.

Symbol	Description
M^2	Match momentum score .
EM^2	Match momentum score process by the EMA model.
c_i	Weight of the $i-th$ component in M^2.
P	Previous factor that influences the momentum.
P_s	Component of previous sets to the current M^2.
P_G	Component of previous games to the current M^2.
P_M	Component of previous player status to the current M^2.
$\boldsymbol{X}$	Feature vector composed of the selected features.
k	Number of top correlation coefficients selected.
N	The time window size set by EMA.
α	The smoothing coefficient/weight decay coefficient set by EMA.

3 Data Preprocessing

Before data analysis, the availability of data must be guaranteed.

- **Missing Value Handling.** Given the low sensitivity of the median to outliers, we chose it to fill in missing values, thus ensuring that the gaps are effectively filled while preserving the overall trend of the data.

- **Outlier Processing.** Identify and process match time over 24 hours. Effective correction of such abnormal value minus 24 hours is carried out to keep the time data within a reasonable range. This is shown in the following Table2.

- **Data Reduction.** Map each player's score in each game to points won according to the Table2. This process preserves the original data characteristics while enhancing clarity, accuracy, and ease of processing and analysis.

- **Data Normalization.** It uses Min-Max Normalization of quantitative data according to the formula1

$$X_{\text{normalized}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}} \tag{1}$$

Table 2: Data preprocessing.

match_id	p1_score	p1_score (corrected)	p2_score	p2_score (corrected)	elapsed_time	elapsed_time (corrected)
2023-wimbledon-1303	0 (love)	**0**	15	**1**	24:57:00	**0:57:00**
2023-wimbledon-1303	30	**2**	15	**1**	24:58:22	**0:58:22**
2023-wimbledon-1303	40	**3**	15	**1**	24:58:44	**0:58:44**
2023-wimbledon-1301	AD	**4**	40	**3**	0:04:01	——
2023-wimbledon-1301	40	**4**	40	**4**	0:04:48	——
2023-wimbledon-1301	AD	**5**	40	**4**	0:05:32	——

After normalization, it is more convenient and fast. The dimensionless expression is transformed into a dimensionless expression, which is convenient for comparing and weight indexes of different units or orders of magnitude. Turning a data set with dimensions into a scalar can also simplify the calculation.

4 Model

In this section, we will introduce the construction process of our model, which includes the match momentum score extraction introduced in section 4.1 and the two-component structure discussed in section 4.2. As illustrated in Figure 3, we begin by conducting a correlation analysis of the dataset (section 4.1.1), aggregate feature values using three methods (section 4.1.2), and then calculate the match momentum score using a weighted approach (section 4.1.3). Subsequently, we smooth the match momentum score using the EMA algorithm and capture potential state information hidden in the sequence (section 4.2.1), identify the occurrence of swing points based on the intersections of the processed curves (section 4.2.2), and finally input the data into four tree-based machine learning models to predict swing points through a voting method (section 4.2.3).

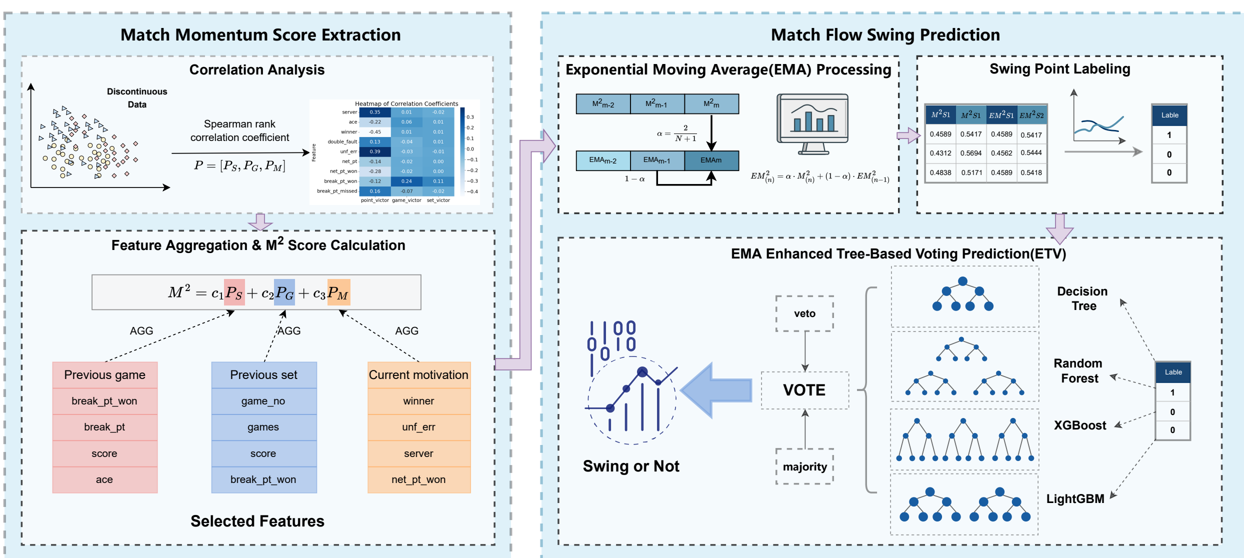


Figure 3: Overview of our model.

4.1 Match Momentum Score Extraction

4.1.1 Correlation Analysis

In tennis matches, "momentum" refers to a player benefiting from a psychological or physiological boost. Typically, when a player has high momentum, it leads to better performance, making it a valuable indicator of a player's performance.

Helmut suggests that the previous set, previous game, and current motivation of the player have a significant impact on a player's momentum [6]. Therefore, we first conduct a correlation analysis on the existing data features to understand the degree of correlation between these features and the three variables mentioned above, thereby assisting in the construction of our model.

Correlation analysis involves analyzing two or more variables that exhibit a relationship to measure the degree of their correlation. Common correlation coefficients include the Pearson correlation coefficient and the Spearman rank correlation coefficient. The Pearson coefficient is primarily used to measure the linear relationship between two continuous variables, while the Spearman coefficient is used to measure the monotonic relationship between two variables without requiring a linear relationship. Since our data mostly consists of non-continuous data, we use the Spearman rank correlation coefficient for our correlation analysis.

The formula for the Spearman rank correlation coefficient is as follows:

$$\rho = 1 - \frac{6\sum d_i^2}{n(n^2-1)} \tag{2}$$

Where d_i represents the difference in ranking between the two variables, and n represents the sample size.

Table 3 presents the results of the Spearman rank correlation coefficient, with the color shades reflecting the absolute magnitude of the correlation coefficient values. Since we assume in our study that the positions of player 1 and player 2 do not affect game performance, we have retained only the relevant data related to player 1 (p1).

Table 3: Spearman's correlation coefficient

Feature	point_victor	game_victor	set_victor	Feature	point_victor	game_victor	set_victor
set_no	0.0253	0.0012	0.0006	ace	-0.2187	0.0638	0.0071
game_no	-0.0052	-0.0130	0.1529	winner	-0.4457	0.0143	0.0080
point_no	-0.0270	0.0076	0.0453	double_fault	0.1333	-0.0355	0.0091
sets	-0.0351	-0.0060	-0.0016	unf_err	0.3882	-0.0306	-0.0055
games	0.0137	-0.0145	0.1438	net_pt	-0.1411	-0.0157	0.0027
server	0.3468	0.0145	-0.0165	net_pt_won	-0.2785	-0.0162	0.0049
serve_no	-0.0189	-0.0209	0.0041	break_pt	0.0580	0.0927	0.0539
point_victor	1.0000	0.0287	-0.0083	break_pt_won	-0.1161	0.2424	0.1146
points_won	-0.0413	0.0055	0.0445	break_pt_missed	0.1607	-0.0692	-0.0201
game_victor	0.0287	1.0000	0.2869	distance_run	0.0253	-0.0669	-0.0120
set_victor	-0.0083	0.2869	1.0000	rally_count	-0.0204	-0.0766	-0.0089
speed_mph	0.0051	0.0288	0.0024	elapsed_time	-0.0256	0.038584	0.004616

Following the correlation analysis, we employ the Top-k method[AI: 8] for feature selection, where k is an artificially set hyperparameter. Specifically, we select the top k features that exhibit the highest correlation with point_victor, game_victor, and set_victor. In this case, we set k=4. The selected features are presented in Table 4.

Table 4: Selected Features

Category	Selected Features
Previous game	break_pt_won, break_pt, score,ace
Previous set	game_no, games, score, break_pt_won
Current motivation	winner, unf_err, server, net_pt_won

4.1.2 Feature Aggregation

Before feature fusion, we first normalize the data to mitigate extreme differences between feature values, while mapping the scores to a fixed interval for subsequent comparison. We utilize the Sigmoid function to normalize each feature value:

$$\sigma(x) = \frac{1}{1 + e^{-wx}} \tag{3}$$

where w represents the correlation coefficient of the selected features with set_victor, game_victor and point_victor. To more comprehensively assess the impact of features on players, we consider the positivity of each feature when determining weights. The positivity of the weight ω depends on the benefit of the feature to the player itself. Specifically, if a feature reflects the player's own attributes or advantages, then the weight ω is positive; if the feature reflects the advantages or attributes of the opponent, then the weight ω is negative.

This method ensures that when analyzing the overall performance of a player, the contribution of each feature accurately reflects its actual impact on the outcome of the match.

In order to more fully explore the information contained in the features we selected in previous match, previous set, and current motivation three categories, we define a new metric called PreviousFactor (P) to measure the impact of factors in each category on momentum. P can also derive three categories of metrics, respectively represented as P_S, P_G, and P_M. It is defined as follows.

Definition 1: PreviousFactor (P). Assume that the feature vector of a certain category selected in section 4.1.1 is $\boldsymbol{X} = [x^{(0)}, x^{(1)}, \ldots, x^{(n)}]$, where $x^{(i)}$ represents individual elements of the feature vector. PreviousFactor can be represented as:

$$PreviousFactor = \mathcal{AGG}\left(\sigma(\boldsymbol{X})\right) \tag{4}$$

where σ is the sgmoid function defined in Equation 3, and $\mathcal{AGG}$ is the aggregation function used to fuse features. We try to utilize three aggregation functions, including average, maximum, and multiplicative, to fuse features. They aggregate different features through the following equations:

$$\mathcal{AVG}: \quad P = \frac{1}{n}\sum_{i=1}^{n} \sigma(x_i), \quad P \in \{P_g, P_m, P_s\} \tag{5}$$

$$\mathcal{MUL}: \quad P = \prod_{i=1}^{n} \sigma(x_i), \quad P \in \{P_g, P_m, P_s\} \tag{6}$$

$$\mathcal{MAX}: \quad P = \max\{\sigma(x_i)\}, \quad i \in \{0, 1, ..., n\};\ P \in \{P_g, P_m, P_s\} \tag{7}$$

PreviousFactor maps the impact of completed sets, games, and newly scored points on an athlete's performance within a match. For instance, an athlete's morale is boosted when they successfully serve an ace, whereas their morale may decrease when the opponent serves an ace. PreviousFactor quantifies the influence of these emotional values on the match outcomes, employing three distinct aggregation methods to emphasize key factors within the impact differently. The $\mathcal{AVG}$ method impartially considers each contributing feature, the $\mathcal{MUL}$ method tends to focus on features with significantly high or low impact, and the $\mathcal{MAX}$ method disregards all features of low importance.

4.1.3 M^2 Score Calculation

After quantifying the impact of the three phases of a match on a player (denoted as P_S, P_G, P_M) using the metric defined in Definition 1, we proceed to measure a player's momentum at a given moment in past matches by a new metric called Match Momentum Score (M^2).

Algorithm 1 M^2 Score Generation Algorithm

Input: $F = [f_0, f_1, ..., f_n]$ ▷ The original features in dataset
Output: M^2 ▷ The M^2 Score

$i = 0;$
$k = setThreshold();$ ▷ Top-k Threshold
$c_1, c_2, c_3 = setWeights();$ ▷ PreviousFactor Weights
while $i < n + 1$ **do**
  $f_i.corrM = getSpearmanCorrelationCoefficient(f_i, 'point\_victor');$
  $f_i.corrG = getSpearmanCorrelationCoefficient(f_i, 'game\_victor');$
  $f_i.corrS = getSpearmanCorrelationCoefficient(f_i, 'set\_victor');$
  $i = i + 1;$
end while
$j = 0;$
while $j < k$ **do**
  $F = reverseSorted(F, T.corrM);$
  $selected_M.add(f_j);$
  $F = reverseSorted(F, T.corrG);$
  $selected_G.add(f_j);$
  $F = reverseSorted(F, T.corrS);$
  $selected_S.add(f_j);$
  $j = j + 1;$
end while
$P_M = \mathcal{AGG}(\sigma(selected_M));$ ▷ Feature Fusion
$P_G = \mathcal{AGG}(\sigma(selected_G));$
$P_S = \mathcal{AGG}(\sigma(selected_S));$
$M^2 = c_1 P_S + c_2 P_G + c_3 P_M$
return M^2;

Definition 2: $\mathbf{M^2}$ **Score.** For given P_S, P_G, and P_M, we assess the impact of momentum on the player by calculating the weighted sum of the three components. M^2 is defined as follows:

$$M^2 = c_1 P_S + c_2 P_G + c_3 P_M \tag{8}$$

where c_i are artificially set hyperparameters used to control the influence of different stages of the match on the overall momentum. To facilitate the observation of the absolute magnitude of momentum, we constrain M^2 to the interval $[0, 1]$. Consequently, the hyperparameters c_i need to satisfy the following equation:

$$\sum c_i = 1 \tag{9}$$

The introduction of the M^2 Score enables the quantification of momentum. Generally speaking, for any given match process, we generate this metric through Algorithm 1.

4.2 Match Flow Swing Prediction

4.2.1 Exponential Moving Average (EMA) Processing

To minimize the noise introduced in M^2 and to incorporate the cumulative effect of time on M^2, integrating more historical state information, we employ the Exponential Moving Average (EMA) algorithm to process the M^2 series calculated in Section 4.1. The essence of EMA is to use the data of past states to obtain the weighted average of the current state, assigning higher weights to more recent data points, thereby making the measurement of momentum changes more sensitive. Given an M^2 series $\{M^2_{(1)}, M^2_{(2)}, \ldots, M^2_{(n)}\}$, the series processed by EMA is $\{EM^2_{(1)}, EM^2_{(2)}, \ldots, EM^2_{(n)}\}$, with the EMA processing as follows.

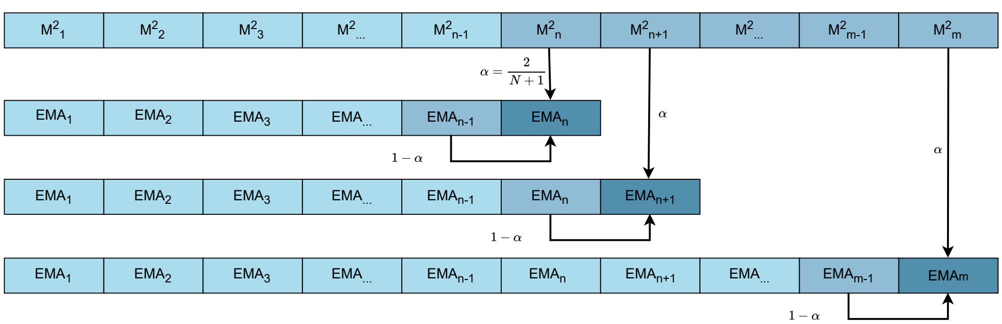


Figure 4: EMA processing process.

EMA embodies the concept of a sliding window, averaging all previous state features with exponentially decreasing weights. As illustrated in Figure 4, the weights of states farther from the current state decay exponentially after EMA processing, providing heightened sensitivity to states within the moving window. We first set the sliding window size N, then determine the decay factor α according to equation 10, and recursively calculate each state's EM^2 using equation 11.

$$\alpha = \frac{2}{N+1} \tag{10}$$

$$EM^2_{(n)} = \alpha \cdot M^2_{(n)} + (1-\alpha) \cdot EM^2_{(n-1)} \tag{11}$$

By applying the EMA algorithm, we obtain a smoothed M^2 series, which aids in identifying long-term trends and patterns in the data while reducing the impact of short-term fluctuations.

4.2.2 Swing Point Labeling

Identifying potential swing points within a match flow requires a large-scale dataset with labels. To determine whether each data point is a swing point, we label the data according to Equation 12:

$$label = \begin{cases} 0, & \frac{[EM1^2]_{(n-1)} - [EM2^2]_{(n-1)}}{[EM1^2]_{(n)} - [EM2^2]_{(n)}} > 0 \\ 1, & \frac{[EM1^2]_{(n-1)} - [EM2^2]_{(n-1)}}{[EM1^2]_{(n)} - [EM2^2]_{(n)}} < 0 \end{cases} \tag{12}$$

where $[EM1^2]_{(n)}$ represents the EM^2 of player 1 at the n-th time step, and $[EM2^2]_{(n)}$ represents the EM^2 of player 2 at the nt-h time step. As illustrated in Figure 5, we consider the possibility of a swing point occurring when the EM^2 curves of player 1 and player 2 intersect. We label such intersection points as True.

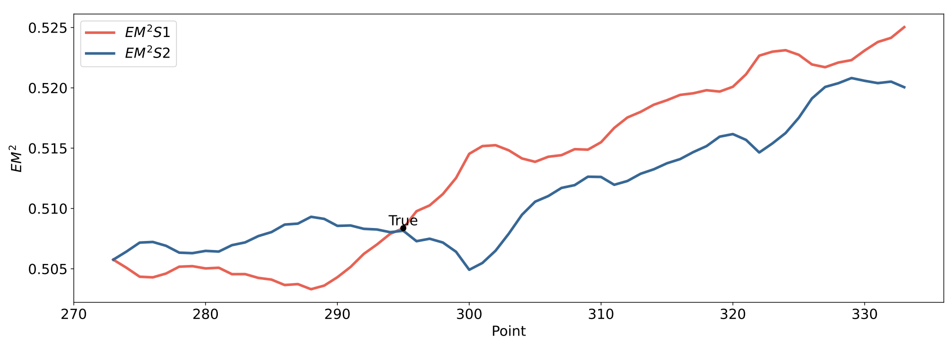


Figure 5: The data that can be labeled True.

4.2.3 EMA Enhanced Tree-Based Voting Prediction (ETV)

Tree-based machine learning models offer numerous advantages, including high adaptability to features, robustness, interpretability, and scalability, often demonstrating superior performance in classification tasks. Ensemble learning with tree-based models has gained widespread application across various domains, holding a significant position in research. In multi-model decision-making, voting is a common method where the final decision is made based on the aggregated predictions of individual models, leveraging their strengths to achieve greater robustness and lower generalization error. In our model, we integrate four tree-based models: Decision Tree, Random Forest, XGBoost, and LightGBM, and employ two voting techniques—simple majority voting and veto voting—to make decisions.

a. Tree-Based Model

Decision Tree [9]: A foundational model that uses a tree-like structure to make decisions by recursively partitioning the dataset. It selects optimal features for splitting based on information gain or Gini impurity, offering ease of understanding, interpretability, and the ability to model nonlinear relationships. Pruning strategies control model complexity and prevent overfitting.

Random Forest [4]: An ensemble model that improves accuracy and robustness by building multiple decision trees and aggregating their predictions. It introduces randomness in feature selection and data sampling, effectively handling large datasets and high-dimensional features, and is robust to outliers and missing values.[AI: 2]

XGBoost [5]: An optimized gradient boosting library that builds a series of decision trees, focusing on reducing prediction error through information gain and regularization to prevent overfitting. It is known for its efficiency, flexibility, and robustness in classification tasks.

LightGBM [7]: A histogram-based decision tree algorithm that finds optimal splitting points using histogram discrete values. It enhances efficiency and scalability with Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB), making it particularly effective for large datasets and classification tasks.

b. Voting

Simple Majority Voting: Each model votes for a class; the class with the most votes becomes the final prediction. It capitalizes on the collective decision-making of diverse models to enhance prediction accuracy.

Veto Voting: Certain models can 'veto' a class, disqualifying it as the final prediction. This method allows for nuanced decisions by leveraging the expertise of specific models in identifying incorrect classes.

To furnish the model with ample relevant information, we incorporate all metrics proposed in previous works, including P, M^2, EM^2, and the first-order differences of M^2 and EM^2, as features. This approach not only aggregates valuable information effectively but also reduces the number of features used, facilitating model simplification and efficiency.

5 Problem Solution: Implementation and Evaluation

In this section, we introduce some methods for evaluating model performance and address the problems set forth in the topic.

5.1 Experiment Set Up

We have implemented our model and conducted evaluations on the four specified tasks. To ensure the reproducibility of our results, Table 5 provides the hyperparameter settings used in the experiments.

Table 5: Hyperparameters used for training our model.

Hyperparameter	k	N	c_1	c_2	c_3	n_estimators	learning_rate	max_depth
Value	5	20	0.5	0.3	0.2	200	0.015	10

To evaluate the various components of our model, we will employ different methodologies. For the extraction of M^2, we use the Wilcoxon hypothesis test [10] to examine whether there is a significant difference between the scores predicted by M^2 and EM^2 compared to scores from random predictions. We will also utilize Interquartile Range (IQR) statistics to assess the performance levels of players. For the ETV prediction model, we will evaluate the classifier's performance using four widely used metrics: accuracy, precision, recall, and the F1 score. These metrics will help us to comprehensively understand the effectiveness of our model in classifying and predicting relevant outcomes.

Wilcoxon Signed-Rank Test. To perform the Wilcoxon signed-rank test, begin by calculating the difference d_i for each pair of data (x_i, y_i).

$$d_i = x_i - y_i \tag{13}$$

Next, disregard all pairs where the difference is zero. Sort the absolute values of the remaining differences and assign ranks R_i to them, where tied values (i.e., identical difference values) are assigned the average rank of their positions. Then, calculate the sum of ranks for positive differences and negative differences, denoted as W^+ and W^- respectively, based on the sign of the differences.

$$\begin{aligned} W^+ &= \textstyle\sum_{d_i>0} R_i \\ W^- &= \textstyle\sum_{d_i<0} R_i \end{aligned} \tag{14}$$

The test statistic W is chosen as the smaller of these two sums, i.e., $W = \min(W^+, W^-)$. Finally, compute the value of the statistic and the p-value.

IQR Statistical Analysis. The Interquartile Range (IQR) is a method to measure the dispersion of a statistical distribution, based on the quartiles of the dataset. The IQR is the difference between the upper quartile (Q3) and the lower quartile (Q1), and it describes the variability or range of the middle 50% of the data. The IQR is calculated as follows:

$$IQR = Q3 - Q1 \tag{15}$$

Data points are considered outliers if they are below $Q1 - 1.5 \times IQR$ or above $Q3 + 1.5 \times IQR$. We perform IQR statistical analysis on the first-order difference series of EM^2 to formulate player response strategies.

Metric for Prediction Model. We apply four widely recognized metrics to evaluate ETV prediction results. Accuracy represents the proportion of correctly classified samples to the total samples. Precision quantifies the proportion of samples that are predicted to be positive among the samples that are actually positive. Recall represents the proportion of correctly classified positive samples to actual positive samples. F1-Score is the harmonic mean of recall and precision, which reveals the robustness of the model. They can be calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{16}$$

$$Recall = \frac{TP}{TP + FN} \tag{17}$$

$$Precision = \frac{TP}{TP + FP} \tag{18}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{19}$$

where TP, TN, FP, FN represent the number of true positives, true negatives, false positives, and false negatives.

5.2 Match Process Visualization: Availability

Through data processing and modeling, we calculate the M^2Score of the two players when each score occurs. Since covers a variety of characteristics of a player's performance, it is a good indicator for quantifying a player's performance. The larger the indicator value, the better the player's performance. The difference in the two players' scores shows exactly how much better they are performing. In order to more clearly show the flow of the game and the performance of the players at the given time, we use four different visualization methods[AI: 7]: heat map, bar chart, pie plate chart[8], and line chart, and present the final visualization results on the poster: $TennisVis$. In figure 6, we take the 2023 Wimbledon Gentlemen's final as an example, red represents player 1 $Alcaraz$; blue represents player 2 $Djokovic$.

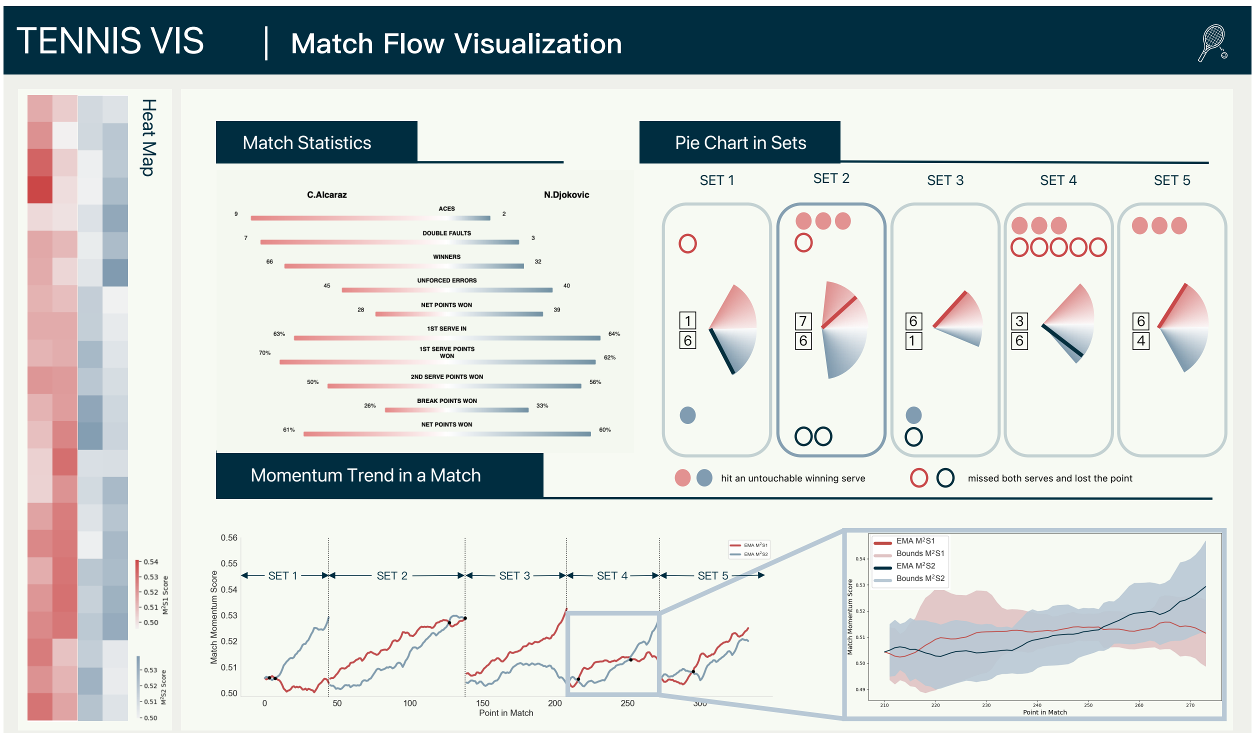


Figure 6: Tennis Visualization.

a. Heat Map

In order to see the performance of the two players in each game, we draw the M^2Score generated by the model into a heat map. Among them, each square grid represents the average value of the player M^2Score at each point in the game, and the grids are arranged vertically over time. From the color depth in the heat map, it can be seen that the red grid, that is, player 1, is significantly more transparent than player 2. In the end it was player 1 who won the game.

b. Match Statistics

In match statistics, it shows the specific indicators such as aces, double faults, and winners of the two players in this match. As can be seen from this picture, $Alcaraz$'s first smash count is higher, and it is his superb skills that enable him to win the final championship.

c. Pie Plate Chart

In order to clearly display the details of each set in a match in one picture, we use a pie plate chart. In the pie plate chart, each pie plate represents a set in the game, and the scores of the two people are recorded in the upper and lower squares; the solid circles represent hitting an untouchable winning serve; the empty circles represent the effect of the poor earth (missing both serves and losing the point) adopts a fan-shaped instrument panel. The sum of the fan-shaped angles indicates how many games the two players played in this set. The greater the angle of the pointer, the greater the score difference between the players in the game. The deepening of this is a break of serve. In the end, it can be seen that in the second game, the two were equally powerful. It took a total of 13 rounds before $Alcaraz$ narrowly won.

d. Trend Line Chart

The trend line chart can better show when the advantages of the two have changed, from their original advantages to disadvantages. We visualize the data of a match and finally find that the curve is cyclical, exactly corresponding to 5 sets, and the trend of the momentum of the two also exactly reflects the score. In the third set, player 2 shows an overwhelming victory, which is consistent with the real score of 1:6. In order to understand the most exciting reversal more clearly, we zoom in on the detailed trend of the 4th set. The intensity of the game can be seen in the repeatedly crossing curves.[AI: 4]

Answer to Problem I: The M^2 Score quantifies a player's momentum, with visual charts illustrating the entire match flow and key performances of the players. M^2 is constrained to the interval [0, 1], with values closer to 1 indicating better performance by the player.

5.3 Match Swing Explanation: Effectiveness

We conduct Wilcoxon tests on the effectiveness of M^2 and EM^2, comparing them to a random process. The random process refers to randomly guessing the scoring side at a given moment, whereas our model identifies the side with higher M^2/EM^2 as the scoring side. The confusion matrices for both processes are shown in Figure 7. Table 6 presents the statistical values and p-values from the Wilcoxon test results. It is evident that, regardless of which metric is used, the test p-values are significantly less than 0.05, indicating the effectiveness of our proposed metrics compared to a random process.

Table 6: Hyperparameters used for training our model.

Models compared	**W statistic**	**p-value**
M^2 Score v.s. Random	2534825.5	8.754e-51
EM^2 Score v.s. Random	3391520.0	0.0138

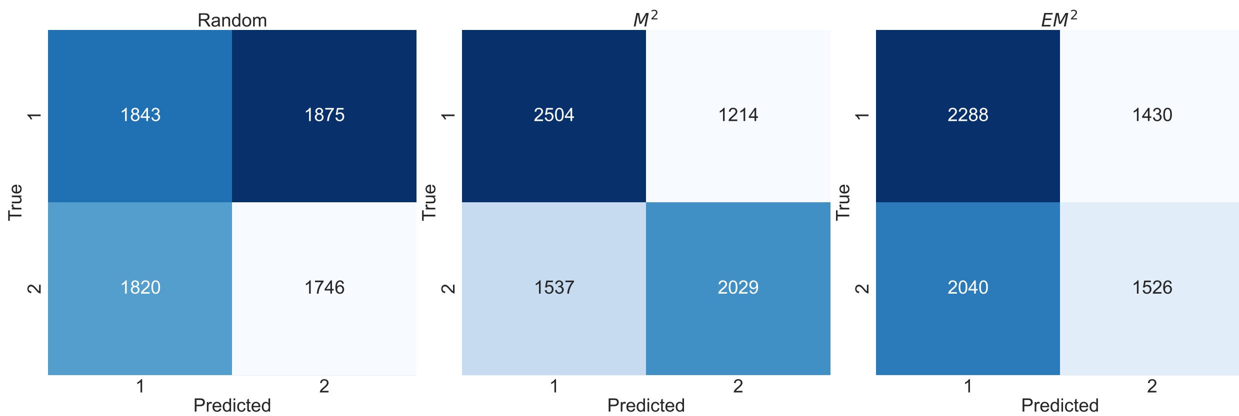


Figure 7: Confusion matrix for predicting whether a player scores using M^2, EM^2, and randomly.

Answer to Problem II: Compared to random processes, we have sufficient grounds to assert that momentum indeed exists and has a significant influence on matches. This is because the measurement of momentum (M^2) can independently predict whether a player scores at each moment of every game.

To predict the occurrence of swing points, we first compare the M^2 and EM^2 during the fourth set of the match numbered 2023-wimbledon-1701 to demonstrate the validity of the EMA algorithm. As shown in Figure 8, the EMA algorithm effectively filter out potential noise in M^2 and achieve temporal feature integration. The shaded areas indicate the boundaries of EM^2, showing that EM^2 smoothens the momentum curve, facilitating time series analysis and swing point prediction. The fourth subplot in Figure 8 illustrates the trend of EM^2 before and after the appearance of swing points.

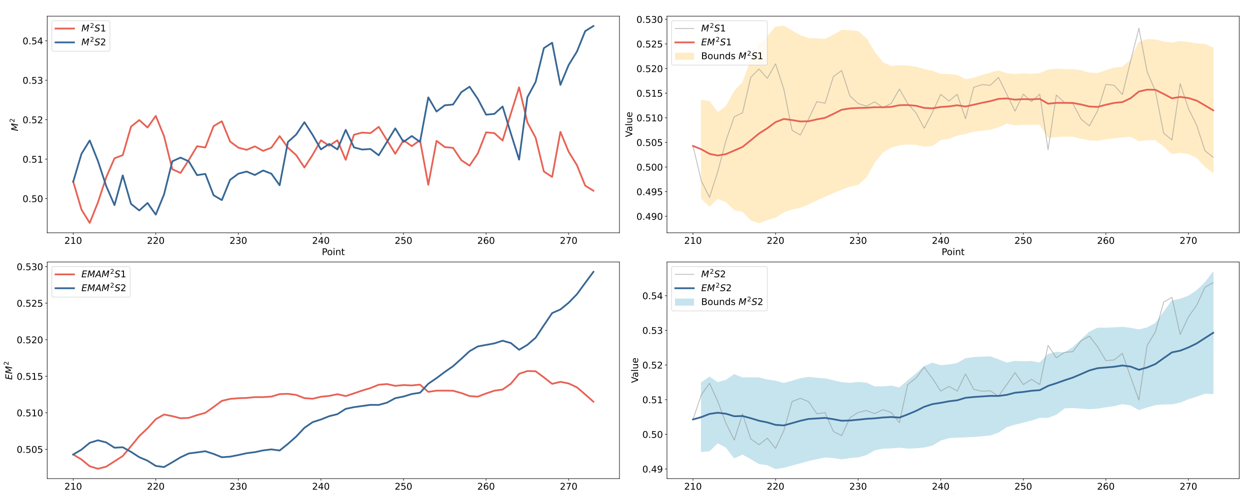


Figure 8: Comparison of M^2 before and after EMA processing.

To validate the performance of the ETV model under different conditions, we first employ undersampling techniques to randomly select an equal number of unlabeled data to mix with labeled data, creating a balanced dataset. Then, we input all data features, including those we have constructed, into the ETV model for classification predictions on both the balanced dataset and the original imbalanced dataset. The results for each model are presented in Table 7.

Table 7: Match abnormal time values modified.

Model	Balanced test set				Unbalanced test set			
	Accuracy	**Precision**	**Recall**	**F1 Score**	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Random Forest	0.709	0.7095	0.709	0.7092	0.7536	**0.9607**	0.7536	0.8269
XGBoost	0.6493	0.7074	0.6493	0.6401	0.7426	0.9585	0.7426	0.8193
Decision Tree	0.7313	0.7413	0.7313	0.7322	0.9417	0.9421	0.9417	0.9419
LightGBM	0.7313	0.7325	0.7313	0.7317	0.9574	0.9167	0.9574	0.9366
ETV-S	0.7388	0.7406	0.7388	0.7393	**0.9581**	0.9485	**0.9581**	**0.9514**
ETV-V	**0.7761**	**0.7834**	**0.7761**	**0.7706**	0.7193	0.96	0.7193	0.8029

The findings indicate that the ETV model using the veto voting method performs best on the balanced test set, while the ETV model employing simple majority voting shows the highest performance on the imbalanced dataset, surpassing all tree-based models.

We also plot the feature importance scores produced by the four tree-based models, as shown in Figure 9. [AI: 5, 9] The results reveal that EM^2 consistently ranks as the most important feature.

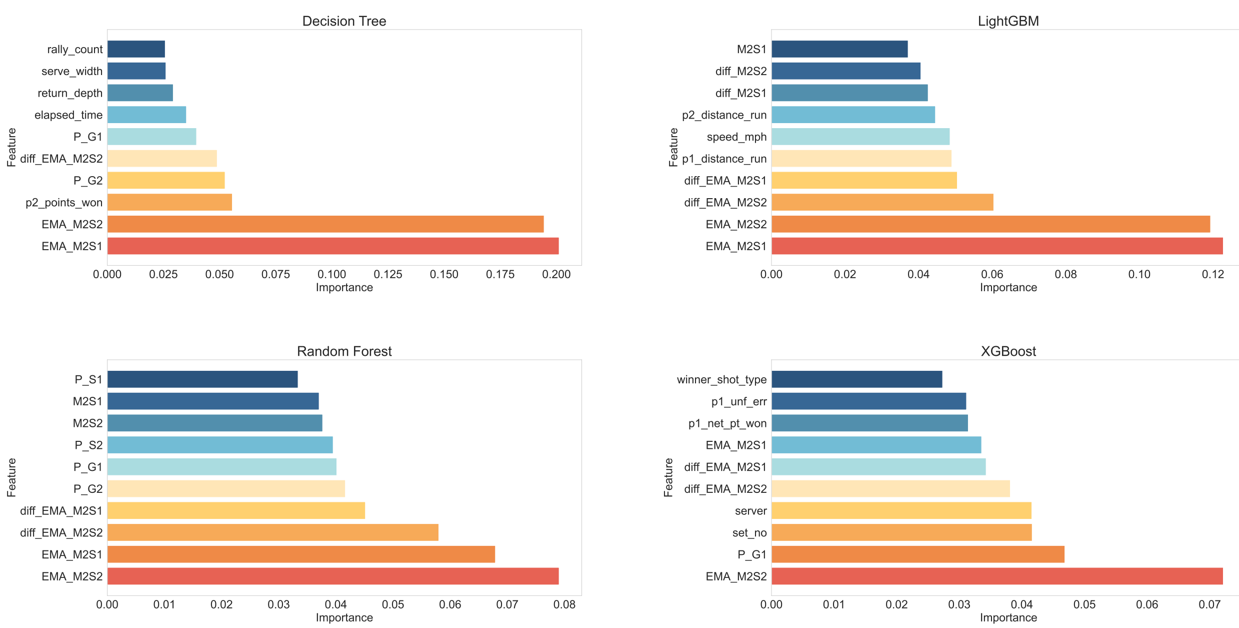


Figure 9: Top 10 feature importance across different models.

Moreover, among the top 10 important features, an average of 67.5% are those introduced by us, further validating the effectiveness of our proposed metrics from another perspective.

Answer to Problem III(a): We successfully predicted the occurrence of swing points using the ETV model, achieving an accuracy of 95.81% and an F1 score of 95.14% on the given dataset. Additionally, we demonstrated that the feature most closely related to swing points is EM^2, which represents the smoothed momentum.

5.4 Player Performance Suggestion: Interpretability

To provide more effective suggestions to coaches, we have conducted an in-depth analysis of this match. Having obtained the EMA-adjusted EM^2 scores for both players, $EM^2 1$ and $EM^2 2$, we proceeded to calculate the first-order difference. Subsequently, we performed IQR statistical analysis to pinpoint the moments of drastic changes in performance. The shaded area in Figure 10 represents the part between Q1 and Q3, and the points outside the realization are outlier outlier points.

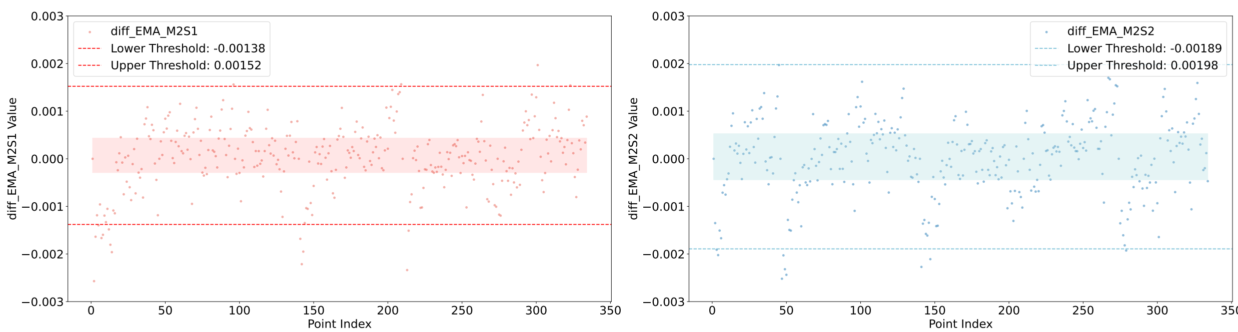


Figure 10: First-order difference scatterplot

Based on the scatter plot in Figure 10, we have identified the outlier points for player 1 and for player 2. We analyze player 2's outliers around moment 50. Near this moment, player 2 exhibited several consecutive outliers, leading us to conclude that he was in poor form. Upon examining the feature values at that time, we note that he committed double faults twice in succession, and it was the beginning of a set, suggesting that the player had not yet settled into the match.[AI: 1] Therefore, we advise that in the next game, he should focus on getting into the right mindset early on. Additionally, during the moments of double faults, the serve speed was significantly higher than at other times, indicating a potential overemphasis on speed at the expense of control. In the following game, the player should be cautious not to sacrifice control for speed and should concentrate on stability.

Answer to Problem III(b): By leveraging historical data and our recommended procedures, we've identified the following insights: A player should stabilize their mindset when consecutive double faults occur; while pursuing speed, control should not be sacrificed, ensuring stability is maintained; if unforced errors occur from consistently targeting the same serve location, changing the serve placement can help reduce mistakes.

5.5 Guidance For Other Competitions: Generalizability

We collect datasets from three additional sports, including women's tennis [3], women's fencing [1], and soccer [2], and evaluated our model on these extra datasets, with the results shown in Table X. The experimental findings indicate that the ETV model performs well on Fencing and Women Tennis but shows mediocre performance on the World Cup dataset. We attribute this discrepancy partly to the fact that fencing is an individual sport, whereas soccer is a team sport. Another reason for the decline in model performance on the World Cup dataset is the limited number of features that measure the performance of both sides during the match, such as p1_ace and p2_ace. This lack of sufficient information that could influence the momentum of both sides leads to the model's ineffectiveness.

Table 8: Match abnormal time values modified.

	Fencing				2016 World Cup				Women Tennis			
Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**Accuracy**	**Precision**	**Recall**	**F1 Score**
ETV-S	0.9835	0.9833	0.9835	0.9834	0.75	0.8181	0.75	0.6947	0.9148	0.9284	0.9148	0.9214
ETV-V	0.9914	0.9913	0.9914	0.9912	0.5833	0.4242	0.5833	0.4912	0.9601	0.9495	0.9602	0.9488

Answer to Problem IV: Our model demonstrates excellent performance in the majority of individual competitive sports, such as women's tennis and women's fencing, showcasing strong generalizability. However, the model's performance declines in team sports like soccer, due to the lack of information on individual players' abilities and the cooperative dynamics between the teams.

6 Discussion

In this section, we conduct a sensitivity analysis of the model and demonstrated the sources of our model's excellent performance. We also discuss the strengths and weaknesses of the model and proposed improvement plans.

6.1 Sensitive Analysis

In our model, the hyperparameters that influence model performance include (1) the number k of features selected using the Top-k method, (2) the weights c_i used in the weighted fusion of features to generate M^2, and (3) choices of the aggregate functions. Below, we conduct a sensitivity analysis for each of these three parameters.

a. Sensitivity analysis for k.

We select the value of k from the set 2, 3, 4, 5 and calculat the M^2 score obtained when selecting the top k features. We also plot the M^2 score values for the match with match_id 2023-wimbledon-1701, as shown in Figure 11.

As the value of k increases, the M^2 score gradually stabilizes. Therefore, we believe that the larger the value of k, the more effective information the model can obtain. However, after reaching a certain point, the model tends to converge, and increasing the number of features may introduce additional noise, potentially reducing model performance.

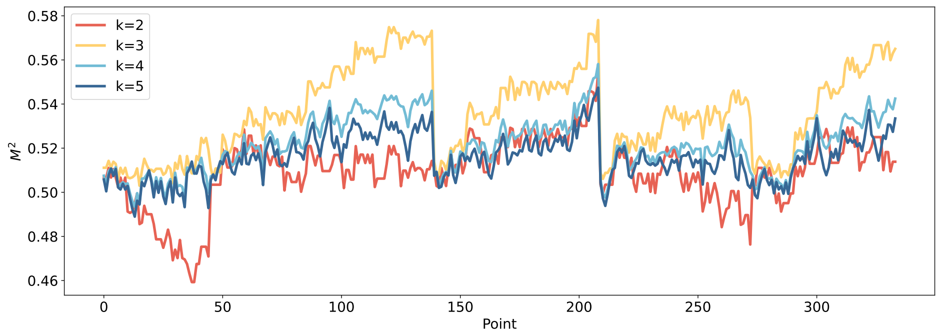


Figure 11: Sensitivity analysis for k

b. Sensitivity analysis for c_i.

We conduct ablation experiments on different combinations of c_i to explore the impact of the three hyperparameters. As shown in Figure 12, c_1 controls the fluctuation of a player's momentum throughout the entire match, exhibiting a clear strong periodicity. c_2 governs the fluctuation of a player's momentum within each set of the match, showing shorter periodicity. c_3, on the other hand, reflects the immediate performance impact on momentum, without clear periodicity. Experimental results indicate that the model is highly sensitive to c_1, while it shows insensitivity towards c_2 and c_3.

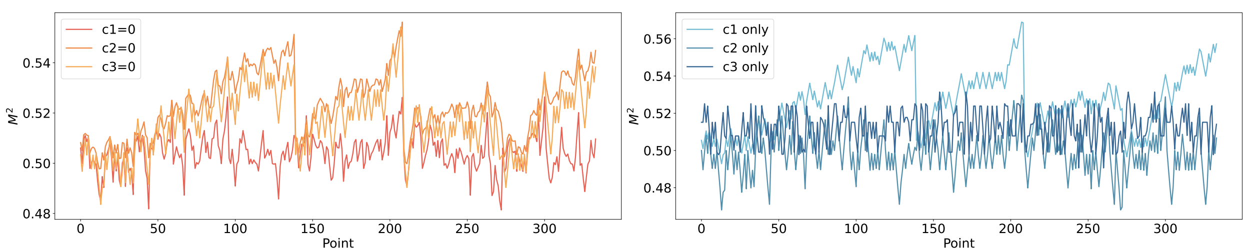


Figure 12: Sensitivity analysis for c_i

c. Choice of the aggregate function.

We test the impact of three different aggregation functions on the calculation of M^2 and plot the variation of M^2 for the same match, as shown in Figure 13.

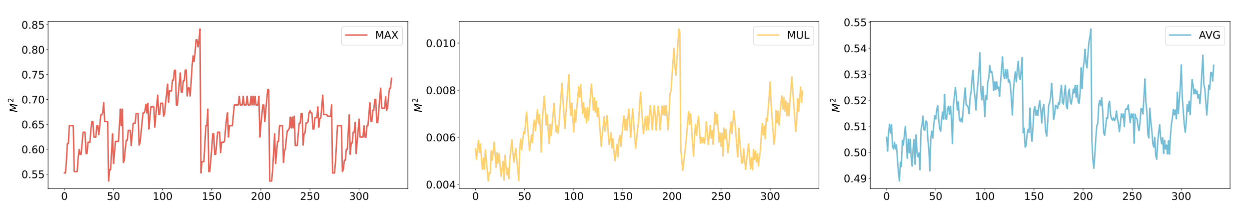


Figure 13: Sensitivity analysis for c_i

The results indicate that although the M^2 curves generated by the three aggregation methods have similar trends, the data aggregated using the $\mathcal{MAX}$ method exhibited a large number of oscillation points, and the data's absolute values become extremely small with the $\mathcal{MUL}$ aggregation, which is not conducive to highlighting differences between features. Therefore, we believe that $\mathcal{AVG}$ provides the best aggregation effect, indicating that the model is sensitive to the choice of aggregation function.

6.2 Why does Our Model Work?

Through the experiments conducted in Section 5, we have demonstrated that the proposed ETV model exhibits excellent performance across multiple datasets. Below, we will elucidate the reasons behind the effectiveness of the ETV model, starting from the two stages of the model.

In the match momentum score extraction stage, we believe the model excels in three key aspects. (1) When using the top-k feature selection, the majority of features are categorical, following a binomial distribution. Therefore, Spearman's rank correlation can accurately measure their correlation. (2) The model incorporates factors from various stages of the competition, providing more effective information grounded in practical scenarios. (3) The introduction of controllable hyperparameters facilitates fine-tuning the impact of different factors.

In the match flow swing prediction stage, we believe the model excels in two key aspects. (1) The EMA (Exponential Moving Average) algorithm smoothens data that originally exhibit strong oscillations and fluctuations, effectively removing mixed noise within the data features. (2) Collective decision-making by multiple models enhances the confidence and robustness of the model towards the outcomes.

6.3 Strength and Weakness

a. Strength

- Pearson correlation analysis is suitable for data features with various distributions, offering high interpretability and a strong theoretical foundation.
- The weighted fusion of features from different stages of the match, inspired by the concept of wave superposition in physics, allows for manually set parameters to enhance the model's generalizability.
- The EVA model can sensitively detect recent changes in momentum while removing noise from the data.
- Tree-based models are highly interpretable, efficient in computation, and adaptable to nonlinear data. The voting mechanism facilitates the integration of advantages from various models, enhancing the model's robustness.

b. Weaknesses

- The performance of the model may fluctuate across different types of matches, such as being insensitive to team sports data and unable to identify individual performances within team competitions.

- During EMA processing, the weights of earlier segments decay exponentially with the progression of time steps, which can lead to the loss of long-term memory in the data, rendering it incapable of handling long-duration data effectively.

- Our model employs machine learning techniques, which typically require extensive data to drive predictions. For competitions with limited data, our model may not be able to generate effective predictions.

6.4 Further Improvement

- During feature selection, relying solely on correlation coefficients and the top-k method to select features may not yield highly informative features. We are considering employing additional metrics for feature selection, such as information entropy and the Principal Component Analysis (PCA) algorithm.

- Capturing periodicity and long-term dependencies in the time series of a match is crucial for momentum analysis and decision-making. We plan to explore more advanced algorithms for integrating sequence features, such as Long Short-Term Memory Networks (LSTM) or Hidden Markov Chains.

7 Conclusions

In this study, we introduce a metric for quantifying momentum, referred to as the M^2 score, and develop an ETV model to predict pivotal moments within matches. With the proposed metric and model, we effectively capture the flow of matches, quantify athletes' performances, and provide visualizations. To substantiate the real presence of momentum, we conduct a Wilcoxon Signed-Rank Test, yielding a p-value of 8.75×10^{-51}, which underscores the significant role of momentum in matches. Furthermore, we employ the ETV model to anticipate when shifts in the game are likely to occur, offering strategic advice to players based on the data and model outputs. Our experiments demonstrate an accuracy of 95.81% and an F1 score of 95.14%, along with an evaluation of feature importance. To validate the generalizability of our model, we tested it on a newly collected dataset, revealing strong generalization capabilities across various sports, with the exception of team sports such as basketball and soccer. Additionally, we conducted a sensitivity analysis to understand the impact of hyperparameter variations on our results.

8 Memo

Uncovering Momentum Flow in Tennis

TO:All coaches and players

FROM:MCM Team # 2419212

DATE:February 6, 2024

Results

The Match Momentum Score quantifies a player's momentum, with visual charts illustrating the entire match flow and key performances of the players.

Compared to random processes, we have sufficient grounds to assert that momentum indeed exists and has a significant influence on matches.

We successfully predicted the occurrence of swing points using the ETV model, achieving an accuracy of 95.81% and an F1 score of 95.14% on the given dataset.

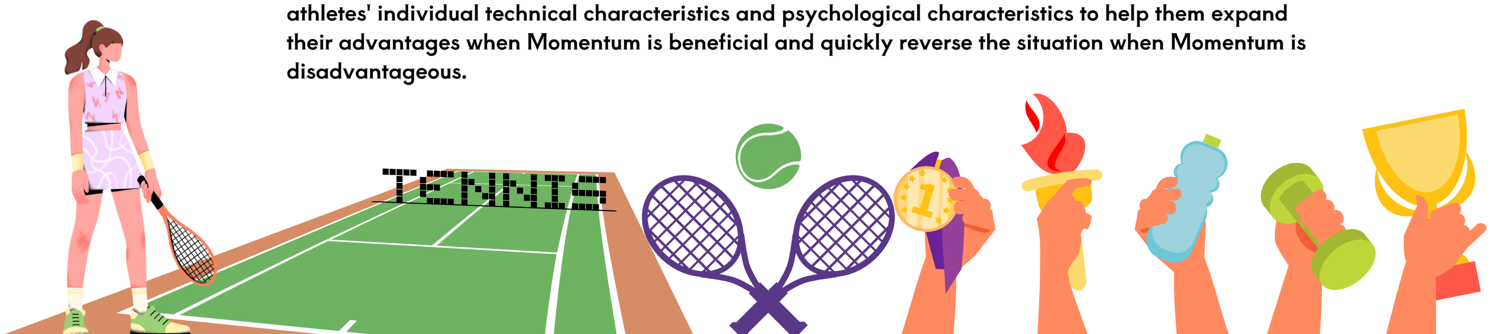
Additionally, we demonstrated that the feature most closely related to swing points is Match Momentum Score processed by EMA , which represents the smoothed momentum.

Our model demonstrates excellent performance in the majority of individual competitive sports, such as women's tennis and women's fencing, showcasing strong generalizability.

Suggestions

What should a coach do?

- Establish a Momentum tracking system: Use sensors and other technical means to monitor athletes' Momentum changes during training and competition, and use data support as the basis to more accurately understand athletes' performance fluctuations.
- Multi-dimensional adaptability training: In addition to technical and tactical training, it also simulates game situations under different situations to enhance athletes' opponent adaptability and mental toughness.
- Psychological training and emotional regulation: Psychological experts are regularly invited for team and individual consultation to teach athletes how to self-motivate and regulate emotions at critical moments.
- Formulate tactics based on Momentum: Develop personalized competition strategies based on the athletes' individual technical characteristics and psychological characteristics to help them expand their advantages when Momentum is beneficial and quickly reverse the situation when Momentum is disadvantageous.



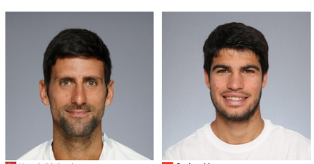
Suggestions

C. Alcaraz	MATCH STATISTICS	N. Djokovic
9	ACES	2
7	DOUBLE FAULTS	3
66	WINNERS	32
45	UNFORCED ERRORS	40
28	NET POINTS WON	39
168	TOTAL POINTS WON	166
121 MPH	AVERAGE 1ST SERVE SPEED	118 MPH
102 MPH	AVERAGE 2ND SERVE SPEED	98 MPH
132 MPH	FASTEST SERVE	127 MPH
94/150 (63%)	1ST SERVE IN	118/184 (64%)
66/94 (70%)	1ST SERVE POINTS WON	73/118 (62%)
28/56 (50%)	2ND SERVE POINTS WON	37/66 (56%)
5/19 (26%)	BREAK POINTS WON	5/15 (33%)
28/46 (61%)	NET POINTS WON	39/65 (60%)
6606.5 m	DISTANCE COVERED (M)	6195.2 m
19.8	DIST. COVERED/PT. (M)	18.5

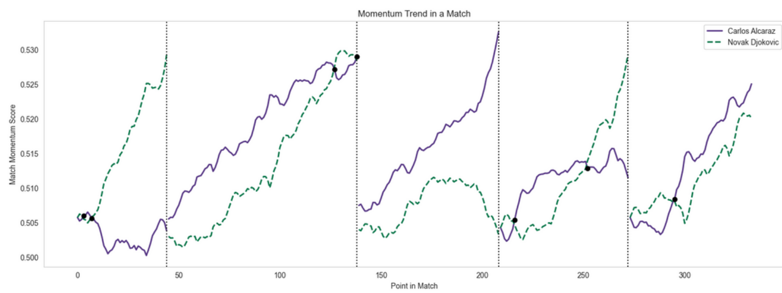
What should a player do?

- Self-regulation during the intermission period: Learn and master the skills of rapid physical and mental adjustment during the short intermission period, maintain a good mentality and stable emotions, and help yourself quickly return to the best state during the game.
- Concentration training: Train in a high-pressure environment, focus on mobilizing your own momentum, and avoid large fluctuations in status. When the momentum of both parties is improving, it is necessary to stabilize the state and make timely adjustments. Improve your ability to multi-task and respond to emergencies.
- Use data for self-analysis: Regularly review and analyze personal competition statistics, identify your own strengths and weaknesses in Momentum management, and make targeted improvements.

Case Analysis



Novak Djokovic Carlos Alcaraz



Take the 2023 Wimbledon men's singles final as an example. The game is between Carlos Alcaraz and Novak Djokovic. We can use this final to illustrate specifically how to apply the above strategies for managing and responding to Momentum:

Opening stage: Djokovic started strongly and quickly established a score advantage through active baseline play and high-quality serves. At this time, Djokovic's Momentum is on the rise, and Alcaraz needs to find opportunities to turn the situation around.

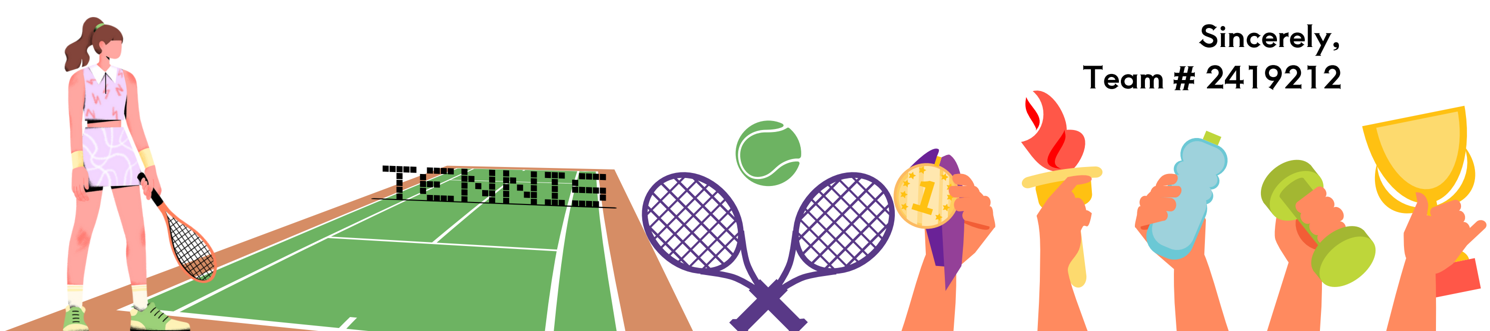
Coach: At this time, Djokovic's coach should remind him to maintain his current strategy and mentality through gestures or communication when changing sides, and at the same time be alert to Alcaraz's counterattack. Alcaraz's coach should provide suggestions for tactical adjustments, such as changing serving strategies or increasing the use of tactics at the net, in order to break Djokovic's rhythm.

Mid-game reversal: In the second and third sets, Alcaraz gradually adapted to Djokovic's style of play, and began to use his explosiveness to gradually regain the score and achieve a reversal. From the second set, Alcaraz's Momentum was on the rise. Djokovic may feel the pressure is increasing, as Momentum continued to slump in the third set after losing the tiebreaker in the second set.

Athletes' response: Djokovic needs to use the interval to make self-adjustments, such as deep breathing, positive psychological suggestions, or discussing brief tactical adjustments with the coach. Alcaraz should continue to maintain its current status while looking for opportunities to further expand its advantages.

Psychological warfare in the deciding game: In the deciding game, the Momentum of the two players may fluctuate frequently, and the outcome of each key point may have a decisive impact on the outcome of the game.

Sincerely,
Team # 2419212



References

[1] FIE Fencing Women's Foil Data. `https://www.kaggle.com/datasets/amichaelsen/fie-fencing-womens-foil-data`. Accessed: 2023-02-05.

[2] Kabaddi World Cup 2016 Dataset. `https://www.kaggle.com/datasets/parmarmanojkumar/kabaddi-world-cup-2016`. Accessed: 2023-02-05.

[3] WTA Matches Dataset. `https://www.kaggle.com/datasets/gmadevs/wta-matches`. Accessed: 2023-02-05.

[4] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[5] 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*, pages 785–794, 2016.

[6] Cornel Nesseler Helmut Dietl. Momentum in tennis: Controlling the match. 2017.

[7] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

[8] Tom Polk, Jing Yang, Yueqi Hu, and Ye Zhao. Tennivis: Visualization for tennis match analysis. *IEEE transactions on visualization and computer graphics*, 20(12):2339–2348, 2014.

[9] J Ross Quinlan. *C4. 5: programs for machine learning*. Elsevier, 2014.

[10] Frank Wilcoxon. Individual comparisons by ranking methods. In *Breakthroughs in Statistics: Methodology and Distribution*, pages 196–202. Springer, 1992.

Report on Use of AI

1. ChatGPT-4

   Query1: I would like to know some common tennis strategies

   Output: Certainly! Tennis is a strategic sport that involves not only physical skills but also mental tactics. Here are some common tennis strategies that players use to gain an advantage:

   1. Serve and Volley: This strategy involves serving aggressively and then quickly moving to the net to volley the opponent's return. It puts pressure on the opponent to pass or lob the ball, which can be challenging.

   2. Baseline Play: Many players prefer to stay at the baseline and engage in baseline rallies. This strategy involves consistent groundstrokes, with the aim of wearing down the opponent, waiting for an opportunity to hit a winner or force an error.

   3. All-Court Play: Some players are versatile and comfortable playing both at the baseline and at the net. They adapt their strategy based on the situation, mixing baseline rallies with net approaches.

   4. Aggressive Play: Aggressive players look for opportunities to hit winners and dictate the points early in the rally. They may take risks with powerful shots and well-timed approach shots to the net.

   5. Defensive Play: Defensive players focus on retrieving every ball and waiting for their opponent to make mistakes. They use slices and lobs to extend rallies and force errors.

2. ChatGPT-4

   Query1: What are the advantages and disadvantages of tree-based machine learning models?

   Output: Tree-based machine learning models, such as decision trees, random forests, and gradient boosting, are popular and widely used in various applications due to their flexibility and interpretability. However, they also come with their own set of advantages and disadvantages:

   Advantages:

   1. Interpretability: Decision trees, in particular, are highly interpretable. You can easily visualize the tree structure and understand how decisions are being made. This makes them suitable for explaining model predictions to non-technical stakeholders.

   2. Non-linearity: Tree-based models can capture non-linear relationships between features and the target variable. They are versatile and can handle both continuous and categorical data.

   3. Robust to Outliers: Decision trees and random forests are relatively robust to outliers and noisy data. They do not rely on distance metrics, making them less sensitive to extreme values.

   Disadvantages:

   1. Overfitting: Decision trees, especially deep ones, are prone to overfitting, capturing noise in the training data. Pruning and setting maximum tree depth can help mitigate this issue.

   2. Instability: Small changes in the training data can lead to significant changes in the tree structure, making decision trees unstable. This instability can be reduced with ensemble methods like random forests and gradient boosting.

3. Bias Towards Dominant Classes: Decision trees tend to favor dominant classes in imbalanced datasets, leading to poor performance on minority classes. Techniques like class weighting can be used to address this issue.

In summary, tree-based machine learning models offer valuable advantages in terms of interpretability, handling non-linearity, and robustness to outliers. However, they require careful handling to address issues like overfitting and sensitivity to data, and they may not be suitable for all types of data or extrapolation tasks.

3. ChatGPT-4
Query1: Keep the original meaning and abbreviate this paragraph:Assuming that the change in momentum occurs after scoring: Based on the continuous nature of momentum changes in real life. To simplify the model's handling of continuous changes, this study chooses the moment after a scoring event as the point in time for the change in momentum, thereby converting the continuous change in momentum into discrete events. Such an assumption allows the research to more effectively assess the direct impact of scoring on the change in momentum, while also simplifying the complexity of the model, making it easier to analyze and interpret.
Output: The assumption posits that momentum shifts occur post-scoring, a premise grounded in the continuous nature of momentum fluctuations observed in real life. In an effort to streamline the handling of these continuous changes within the analytical framework, the moment following a scoring event is designated as the juncture for momentum alteration, effectively discretizing what is inherently a continuous process. This approach facilitates a more precise evaluation of scoring's immediate impact on momentum dynamics, concurrently reducing model complexity to enhance clarity and interpretability.

4. ChatGPT-4
Query: Which command in matplotlib is used to increase the plot font size?
Output: In Matplotlib, you can increase the font size of various elements in your plot, such as axis labels, tick labels, and titles, by using the fontsize parameter when specifying text properties. Here's an example of how to increase the font size for different plot elements:

```python
import matplotlib.pyplot as plt

# Create a simple plot
plt.plot([1, 2, 3, 4], [1, 4, 9, 16])

# Increase font size for different plot elements
plt.xlabel('X-axis Label', fontsize=14)  # Increase X-axis label font size
    ↪ to 14
plt.ylabel('Y-axis Label', fontsize=14)  # Increase Y-axis label font size
    ↪ to 14
plt.title('Plot Title', fontsize=16)      # Increase plot title font size
    ↪ to 16

# Increase font size for tick labels on both axes
plt.xticks(fontsize=12)  # Increase X-axis tick label font size to 12
plt.yticks(fontsize=12)  # Increase Y-axis tick label font size to 12

plt.show()
```

In this example:

‘fontsize‘ is set to the desired font size (e.g., 14, 16, 12) for each element. ‘xlabel()‘, ‘ylabel()‘, and ‘title()‘ functions are used to set the labels and title, with the specified font sizes. ‘xticks()‘ and ‘yticks()‘ functions are used to set the font size for tick labels on the X-axis and Y-axis, respectively.

You can adjust the font size values (e.g., 14, 16, 12) to your preferred size in order to increase or decrease the font size for different plot elements according to your needs.

5. ChatGPT-4
   Query: What should I do if I want to display the square in the legend of the figure in python?
   Output: If you want to display a square in the legend of a figure in Python using Matplotlib, you can achieve this by specifying a custom legend marker. You can create a custom legend handler that uses a square marker, and then add it to your legend. Here’s an example of how to do it:

```
import matplotlib.pyplot as plt
from matplotlib.lines import Line2D

# Create a simple plot
plt.plot([1, 2, 3, 4], [1, 4, 9, 16], label='Data')

# Create a custom legend handler with a square marker
legend_square = Line2D([0], [0], marker='s', color='w', markersize=10,
    ↪ markerfacecolor='b', label='Square')

# Add the custom legend handler to the legend
plt.legend(handles=[legend_square])

plt.show()
```

6. ChatGPT-4
   Query: Format and scoring rules of the Wingelden Tennis Championships
   Output: I’m sorry, but I don’t have information on the "Wingelden Tennis Championships" as of my last knowledge update in January 2022. It’s possible that this may be a fictional or lesser-known tennis tournament or event that has come into existence after my last update.

   Tennis tournaments typically follow established formats and scoring rules, but the specific details can vary depending on the tournament’s level, such as Grand Slam events (e.g., Wimbledon, the US Open), ATP Tour tournaments, WTA Tour tournaments, or local competitions. The rules and formats are usually outlined in the tournament’s official regulations and can change over time.

   If "Wingelden Tennis Championships" is a real tournament or event that has been introduced after my last update, I recommend checking the official website or relevant sources for the most up-to-date information on its format, scoring rules, and other details. Additionally, you may find the tournament’s rules and regulations in the official rulebook provided by the organizing body.

7. ChatGPT-4
   Query: How to use charts to represent large-scale discrete data processing results
   Output: Representing large-scale discrete data processing results using charts can help convey

information effectively and make it easier for others to understand and interpret the data. Here are some common types of charts and how to use them for this purpose:

1. **Bar Charts**: - **Vertical Bar Chart**: Use vertical bar charts to display discrete data categories on the x-axis and the corresponding counts, frequencies, or metrics on the y-axis. This is suitable when you have a large number of discrete categories to compare. - **Horizontal Bar Chart**: Similar to vertical bar charts, but with categories on the y-axis and values on the x-axis. This can be useful when category labels are long or when you want to conserve horizontal space.

2. **Pie Charts**: - Pie charts are useful for displaying the composition or distribution of categories within a whole. Each slice represents a category, and the size of the slice corresponds to its proportion of the whole. Be cautious when using pie charts for large numbers of categories, as they can become cluttered and hard to interpret.

3. **Stacked Bar Charts**: - Stacked bar charts can be used when you want to show how individual categories contribute to a whole. Each bar represents the total for a specific category, and segments within the bar represent subcategories or components of that total.

4. **Treemaps**: - Treemaps are useful when you have hierarchical data with multiple levels. They divide a rectangular area into nested rectangles, with each level representing a category and subcategories. The size of each rectangle can be proportional to a metric or value.

5. **Heatmaps**: - Heatmaps are effective for visualizing large-scale data matrices, such as correlation matrices or frequency distributions. Color intensity represents the magnitude of the values, making it easy to spot patterns and trends.

6. **Scatterplots**: - Scatterplots can be used when you want to compare two continuous variables within large-scale discrete data. Each point on the plot represents an observation, and the x and y axes represent the two variables. You can use color or shape to represent different categories.

7. **Line Charts**: - Line charts are useful when you want to visualize trends or changes over time or other continuous dimensions. Each line represents a category or variable, and data points connected by lines show how values change over the continuous dimension.

8. **Box Plots**: - Box plots are suitable for displaying the distribution of a dataset, including median, quartiles, and potential outliers. They are especially useful when you want to compare distributions across multiple categories.

9. **Violin Plots**: - Violin plots combine elements of box plots and kernel density plots to display the distribution of data, making it easier to see both central tendencies and variability.

When representing large-scale discrete data processing results, it's important to choose the chart type that best suits the nature of your data and the message you want to convey. Additionally, consider labeling axes, providing a legend or color key, and using appropriate titles and annotations to make your charts more informative and understandable.

8. ChatGPT-4
Query: I want to randomly select data with 5 game_id values. How should I write the code?
Output: Here's a breakdown of the code:

```python
import pandas as pd
import numpy as np  # for generating random numbers

# Assuming you already have a DataFrame named 'df'
# Generate a list of 5 random unique 'game_id' values
unique_game_ids = df['game_id'].unique()
random_game_ids = np.random.choice(unique_game_ids, size=5, replace=False)

# Filter the DataFrame to select rows with these 5 'game_id' values
selected_data = df[df['game_id'].isin(random_game_ids)]

# Now, 'selected_data' contains the rows with 5 randomly selected 'game_id'
    ↪ values
```

9. ChatGPT-4
   Query: plt.title('Momentum Trend in a Match') plt.xlabel('Point in Match') plt.ylabel('Match Momentum Score') plt.legend() plt.show()

   Modify this code to remove the grid
   Output: To remove the grid from your Matplotlib plot, you can use the 'plt.grid(False)' function. Here's the modified code with the grid removed:

```python
import matplotlib.pyplot as plt

# Assuming you have already set up your plot
plt.title('Momentum Trend in a Match')
plt.xlabel('Point in Match')
plt.ylabel('Match Momentum Score')
plt.legend()

# Disable the grid
plt.grid(False)

plt.show()
```