

The MCM Thesis of Team 2423546

Summary

Keywords: Logistic Regression Model, Lazypredict, ExtraTreesRegressor Model

This paper focuses on the issue of momentum in tennis, developing a model that can reflect changes in momentum throughout the course of a tennis match, thereby providing a basis for predicting match outcomes. The model is also expected to evaluate and examine certain metrics, such as determining when the direction of the game shifts from favoring one player to the advantage of another. Variables including the players' abilities, court conditions, and match strategies, among others, are considered to enable a more accurate simulation of actual match conditions.

For question 1, designing a problem that captures the scoring process during the game, that is, dynamically evaluating the player's performance, we used a logistic regression model to calculate each factor and its weight that may affect the game result, and finally concluded that 10 factors have a certain impact on the result. Based on these weights, we built a model in section 4 and specifically gave the dynamic performance of the players in the two games match-1315 and match-1701.

For question 2, we explore the definition of momentum, appropriating the weights derived in question one. We propose a method for the quantification of momentum, which, in conjunction with the logistic regression model, calculates the influence of momentum on the match. The accuracy value obtained is 0.839396019, indicating that momentum is a crucial factor within a tennis match. We also calculated the feature weights [2.40360592, 2.50528734], signifying the pivotal role that momentum plays throughout the match and the general influences it has on the players.

For question 3, find the optimal algorithm model through lazypredict, use this optimal algorithm model ExtraTreesRegressor for training, extract the data of one of the games, use the existing model to predict the change in winning rate, and get the derivative curve, which can be approximated from the figure. Before the game, use historical training data to complete the radar chart of athlete ability evaluation indicators and make tactical arrangements based on the opponent's weaknesses. During the game, use the prediction model to predict the momentum fluctuation curve, derive the momentum change rate curve, and determine whether we need to adjust tactics based on our momentum response.

For question 4, based on question three, we incorporate new features through fine-tuning, achieving generality. By comparing performance and analyzing features of the data set, we establish the factors that may need to be included.

Finally, we analyzed the future upgrade direction of the model based on our results to make our results more scientific and practical.

Contents

1	Introduction	3
1.1	Background	3
1.2	Our Work	3
2	Analysis of the Problem	4
3	Logistic Regression Model and Weight Calculation	5
3.1	Model Description	5
3.2	Data Preprocessing	5
3.3	Conclusion	6
4	Weighted Dynamic Scoring Model	7
4.1	Model Description	7
4.2	Model application	7
4.3	Conclusion	8
5	Model Analysis of Momentum in Tennis Matches	9
5.1	Model Description	9
5.2	Analysis of Logistic Regression Model Results	10
6	Turning Prediction Model with ExtraTreesRegressor	10
6.1	Analysis of Model Performance with LazyRegressor	10
6.2	ExtraTreesRegressor Model Analysis	11
6.2.1	Performance Metrics	11
6.2.2	Learning Curves	12
6.2.3	Residual Plots	12
6.2.4	conclusion	13
6.3	Data Prediction and Analysis with ExtraTreesRegressor	13
6.3.1	Predicting and Visualizing the Data	13
6.3.2	Curve Fitting	13

6.3.3	Comparing Predicted and Actual Values	13
6.3.4	Analyzing the Rate of Change of Momentum	14
7	Model generalization and related suggestions	15
7.1	Model generalization	15
7.2	suggestions	15
8	Future Promotion	17
	Appendices	18
	Appendix A First appendix	19
	Appendix B Second appendix	19
	Appendix C Third appendix	20

1 Introduction

1.1 Background

In any sport, we will find that strength is not the only factor that determines victory or defeat. There are often many variables that lead to the outcome of the game, such as hot hand in basketball and continuous scoring in table tennis. The 2023 Wimbledon competition continued to trigger our thinking on this issue. With this topic, we focused on using logistic regression models, ExtraTreesRegressor and other models to specifically and systematically launch a comprehensive and scientific discussion on the impact of momentum.

1.2 Our Work

Addressing question one, which aims to capture the scoring process of the match, i.e., the dynamic assessment of a player's performance, we employ a logistic regression model to calculate each factor that could potentially influence the match outcome, along with their respective weights. We concluded that factors such as game advantage, set advantage, scoring advantage, serving first, errors, net-front performance, running distance, etc., all contribute to some extent to the end results. Based on these weights, we establish a model in Section 4, specifying the dynamic performance of players in matches 1315 and 1701.

In response to question two, we explore the definition of momentum, appropriating the weights derived in question one. We propose a method for the quantification of momentum, which, in conjunction with the logistic regression model, calculates the influence of momentum on the match. The accuracy value obtained is 0.839396019, indicating that momentum is a crucial factor within a tennis match. We also calculated the feature weights [2.40360592, 2.50528734], signifying the pivotal role that momentum plays throughout the match and the general influences it has on the players.

Regarding question three, we initially perform data cleaning, then utilize Lazypredict to find the optimal algorithm model. Using this optimal algorithm model, ExtraTreesRegressor, for training, we apply Principal Component Analysis to identify the most related factors. After extracting data from a specific match, we use our trained model to predict changes in the winning probability, derive a derivative curve, and approximate the start of the turning point. Before the match, using the results from the previous principal component analysis, we complete a radar chart of the player's ability assessment indicators using historic training data. This allows us to specifically identify and rectify any weaknesses before the game, and plan tactics based on the opponent's weaknesses during the match. We use the predictive model from question three, predicting the momentum fluctuation curve, deriving the change rate curve of momentum and determining whether tactical or psychological adjustments are needed based on our momentum.

Concerning question four, based on question three, we incorporate new features through fine-tuning, achieving generality. By comparing performance and analyzing features of the data set, we establish the factors that may need to be included.

2 Analysis of the Problem

This question provides a detailed description of the 2023 Wimbledon Tennis Open men's singles final, along with the match data provided. It explains the definition of "momentum" in sports and the difficulty of measuring it. The issue poses questions about tennis matches and asks us to provide solutions. Below is our introduction to the problem and the ideas behind each model.

Problem One: Modeling the Match Process When Match Points Occur This is a time-series prediction regression problem. Model the occurrence of match points as a state transition process, predicting match points based on historical match data and technical statistics. Since the scoring model is based on games and sets, Grand Slam matches require winning 5 sets to win, with each set won by winning 6 games, or at 6:6, the tiebreaker decides the winner. In this game, the first player to reach 7 points wins, and if at 6:6, one must win by a clear margin of 2 points to win the set. In data preprocessing, previous sets won can be used as a feature; if 2 out of the first 4 sets have been won, then directly predict whether match points will occur in the third set. Since scoring is converted into a time-series data, the scoring data can be transformed into event data, such as recording the occurrence of each match point, the score when the match point occurs, and the winner of the match point. Then, use a time-series regression model for prediction.

Problem Two: Model to Evaluate the Impact of "Momentum" Assess whether "momentum" really plays a role in tennis matches or whether changes during the match are merely random fluctuations. Predicting match progress: Use models to predict when momentum will shift from one player to another during the match, and develop a model to predict these changes in the match process. Identify potentially relevant factors and provide players with advice on how to deal with different opponents in new matches based on past changes in momentum. This is a factor analysis problem, using statistical and machine learning methods to assess whether players' performance is affected by "momentum," exploring the existence and extent of the momentum effect.

Problem Three: Indicators to Judge When the Match Process Shifts Use models to predict when momentum will shift from one player to another during the match, and develop a model to predict these changes in the match process. Determine potentially relevant factors, and provide players with advice on how to deal with different opponents in new matches based on past changes in momentum. This is a classification problem, whether Player A has a higher probability of winning or Player B. Establish a state transition model to judge when the match process will shift to another player, considering technical statistics and score changes.

Problem Four: Establishing a Fluctuation Prediction Model Test the model's predictive ability on other matches and identify other factors that may need to be included in the model to assess its generalizability and applicability to other matches, events, court surfaces, and other sports like table tennis. This is also a time-series prediction problem, using match data and technical statistics to establish a fluctuation prediction model to determine the volatility of match outcomes and identify the most relevant factors.

3 Logistic Regression Model and Weight Calculation

3.1 Model Description

This model is based on **Logistic Regression**, a statistical method used for binary classification problems. In this particular model, our objective is to predict the winner of each point in a tennis match. To achieve this, we consider the following features:

3.2 Data Preprocessing

- Number of sets won by players (p1_sets, p2_sets)
- Number of games won by players (p1_games, p2_games)
- Scores of players (p1_score, p2_score)
- Aces by players (p1_ace, p2_ace)
- Winners by players (p1_winner, p2_winner)
- Double faults by players (p1_double_fault, p2_double_fault)
- Unforced errors by players (p1_unf_err, p2_unf_err)
- Net points won by players (p1_net_pt_won, p2_net_pt_won)
- Break points won by players (p1_break_pt_won, p2_break_pt_won)
- Distance run by players (p1_distance_run, p2_distance_run)
- Rally count (rally_count)
- Ball speed (speed_mph)

The target variable is *point_victor*, representing the winner of each point. The model employs **standardization** to scale the data to the same magnitude. The data is split into training and test sets using **train_test_split**, and the **LogisticRegression** model is applied for training. After training, we obtain the **weights** of each feature relative to the prediction outcome, reflecting their importance in the prediction process.

Based on the trained logistic regression model, we have identified key features and their importances in predicting tennis match outcomes. As illustrated in Figure , each feature contributes differently to the prediction of match results. The most influential features are:

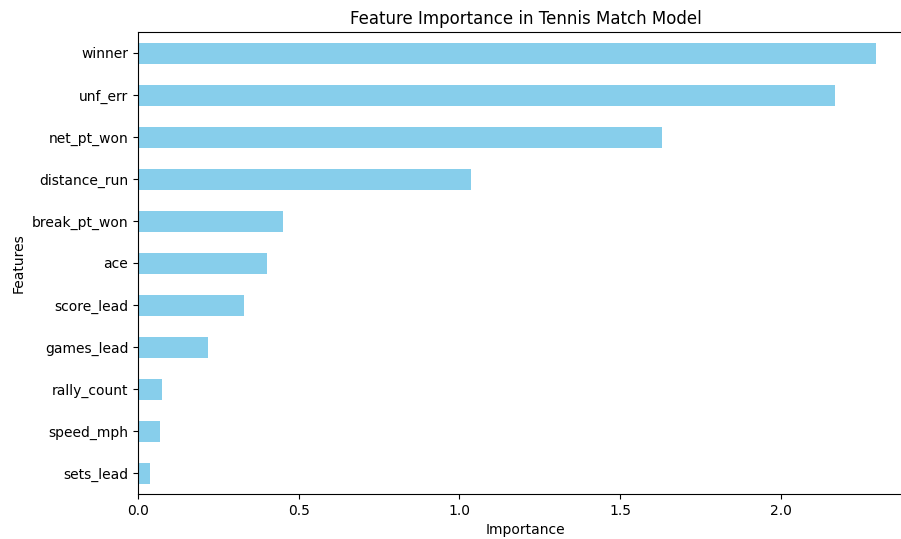


Figure 1: Feature Importance in Tennis Match Model

3.3 Conclusion

- **Winners** - The most significant predictive factor in the model with an importance score of 2.296553, indicating that the number of points won by a player is the most critical indicator of their match-winning ability.
- **Unforced Errors** - The second most significant feature strongly correlated with match victory, with an importance score of 2.169042, suggesting that fewer unforced errors correlate positively with scoring points in the match.
- **Net Points Won** - With a score of 1.630424, it shows the importance of skill at the net in winning points.
- **Distance Run** - With a score of 1.037050, indicating the mobility and effort of a player during the match.
- **Break Points Won** - The ability to win points on an opponent's serve, with an importance score of 0.451047.

Other statistics such as Aces, Score Lead, Games Lead, Rally Count, Speed MPH, and Sets Lead, while contributing to the predictive model, are of relatively lower importance. These findings provide valuable insights for strategic planning and training in tennis matches and can guide players' decision-making during the game.

In summary, these findings underscore the importance of accurate play, minimizing errors, and sharp reactions at the net in competitive tennis matches.

4 Weighted Dynamic Scoring Model

4.1 Model Description

The model presented in this paper is designed to dynamically evaluate the performance of tennis players during a match. Performance scores are calculated using a weighted sum of normalized features, where weights are derived from the features' perceived importance in the match. The features include:

- Distance run (d)
- Aces (a)
- Unforced errors (u)
- Break points (b)
- Points won (p)

Given a set of features $F = \{f_1, f_2, \dots, f_n\}$ and corresponding weights $W = \{w_1, w_2, \dots, w_n\}$, the performance score (S) for a player is calculated as:

$$S = \sum_{i=1}^n w_i \cdot f_i$$

where f_i is the normalized value of feature i and w_i is the weight assigned to feature i .

The model's visualization can be represented by a plot showing the progression of performance scores over the course of the match.

4.2 Model application

The Python script processes match data from a CSV file, where each match is uniquely identified. For a selected match, features are extracted and normalized. Additional features such as player fatigue, serve advantage, and score advantage are calculated to enrich the model's input. The model assigns weights to the normalized features to compute a performance score for each player. The performance scores are then plotted against the point number to illustrate the players' dynamic scoring throughout the match.

We selected two specific matches, namely, 2023-wimbledon-1315 and 2023-wimbledon-1701, for in-depth analysis. The visualization of the dynamic scoring model is presented as follows.

0.5 represents the player's regular performance. A value above 0.5 is a highlight performance, and a value below 0.5 is a sluggish performance.

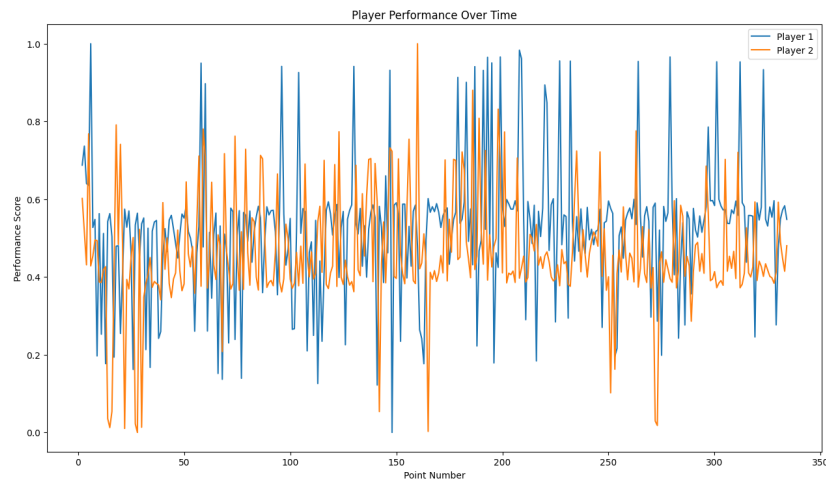


Figure 2: 2023-wimbledon-1315

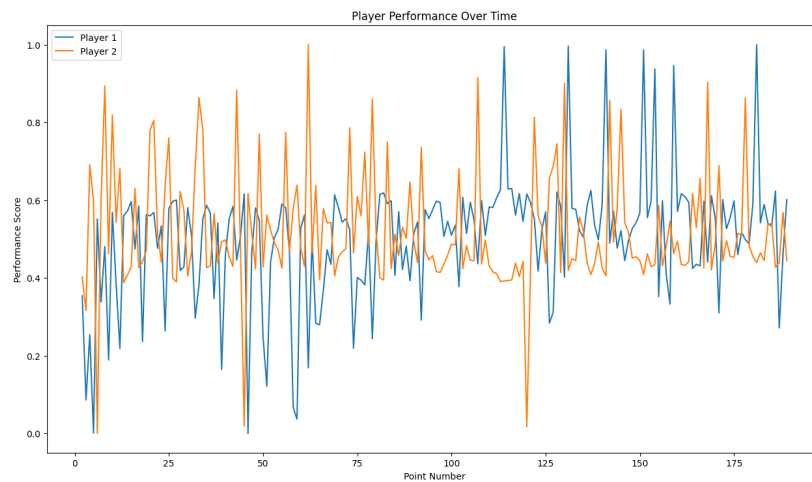


Figure 3: 2023-wimbledon-1701

4.3 Conclusion

This model provides a valuable tool for a quantitative and dynamic evaluation of tennis players' performance, offering insights into their in-match capabilities and potential strategies for improvement.

5 Model Analysis of Momentum in Tennis Matches

5.1 Model Description

In this section, we present an analysis of a logistic regression model used to predict the point winner in tennis matches based on players' momentum. The momentum for each player is calculated using various match statistics, providing a quantitative assessment of their performance at any given point in the match.

The momentum calculation for each player is given by:

$$\begin{aligned}\text{momentum}_1 &= \sum_{i=1}^n w_{1i} \cdot \text{stat}_{1i} \\ \text{momentum}_2 &= \sum_{i=1}^n w_{2i} \cdot \text{stat}_{2i}\end{aligned}$$

where w_{1i} and w_{2i} are the weights assigned to the i^{th} statistic for player 1 and player 2, respectively, and stat_{1i} and stat_{2i} are the values of the i^{th} statistic for each player.

For logistic regression analysis, the probability of player 1 winning the point is modeled as:

$$P(\text{point victor} = 1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 \cdot \text{momentum}_1 + \beta_2 \cdot \text{momentum}_2)}} \quad (1)$$

where β_0 is the intercept, and β_1 and β_2 are the coefficients for the momentum of player 1 and player 2, respectively.

The model is trained on a scaled feature set to ensure that all variables contribute equally to the prediction. The scaling is performed using the standard score:

$$z = \frac{(x - \mu)}{\sigma} \quad (2)$$

where x is the original feature value, μ is the mean of the feature values, and σ is the standard deviation of the feature values.

After training the model on a subset of the data, predictions are made on the test set, and the accuracy is calculated as the ratio of the number of correctly predicted points to the total number of points in the test set:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \quad (3)$$

Finally, the feature weights obtained from the logistic regression model are interpreted

as the influence of each player's momentum on the probability of winning a point. The weights are given by the model's coefficients:

$$\text{Feature Weights} = \beta_1, \beta_2 \quad (4)$$

These weights help us understand the relative importance of each player's momentum in determining the match outcome.

5.2 Analysis of Logistic Regression Model Results

The logistic regression model's accuracy and feature weights provide insights into its predictive performance and the importance of each feature. The model's accuracy is a measure of its overall performance and is given by the percentage of correct predictions made out of all predictions. The accuracy of the model is calculated as:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} = 83.94\% \quad (5)$$

This high accuracy indicates that the model is able to predict the point winner with a high level of certainty.

The feature weights in logistic regression represent the contribution of each feature towards the prediction. In our model, the feature weights are given by:

$$\text{Feature Weights} = [2.4036 \quad 2.5053] \quad (6)$$

These weights suggest that both features, which represent the momentum of player 1 and player 2, have a significant and almost equal positive influence on the model's predictions. The slightly higher weight for the second feature implies that player 2's momentum is marginally more influential in determining the point winner compared to player 1's momentum.

6 Turning Prediction Model with ExtraTreesRegressor

6.1 Analysis of Model Performance with LazyRegressor

We use the LazyRegressor to automatically compare the performance of multiple regression models, aiming to identify the best-performing model on a specific dataset. LazyRegressor is a convenient tool that quickly compares and evaluates various models, providing a basis for further model selection.

The results of LazyRegressor provide a performance assessment of each model, offering guidance on how to select the appropriate model. This method allows us to effectively screen for machine learning models that perform well on specific problems.

	Adjusted R-Squared	R-Squared	RMSE	Time Taken
Model				
RandomForestRegressor	0.99	0.99	2.71	2.73
BaggingRegressor	0.99	0.99	2.84	0.29
DecisionTreeRegressor	0.99	0.99	3.58	0.11
XGBRegressor	0.98	0.98	4.56	0.22
LGBMRegressor	0.97	0.97	5.52	0.16
HistGradientBoostingRegressor	0.96	0.97	5.89	0.49
ExtraTreesRegressor	0.90	0.90	10.13	2.09
GradientBoostingRegressor	0.78	0.78	14.78	0.80
AdaBoostRegressor	0.48	0.49	22.76	0.19
ExtraTreeRegressor	0.46	0.47	23.21	0.07
MLPRegressor	0.12	0.13	29.61	7.08
NuSVR	0.04	0.06	30.78	1.28
SVR	0.04	0.06	30.82	1.89
BayesianRidge	0.00	0.02	31.47	0.02
ElasticNetCV	0.00	0.02	31.48	0.20

Figure 4: Analysis of Model Performance

6.2 ExtraTreesRegressor Model Analysis

The ExtraTreesRegressor model is an ensemble learning approach that utilizes a multitude of decision trees to output a more accurate prediction by averaging the results. This model is particularly effective in capturing complex structures within the data set. Below we present a detailed analysis of the model's performance:

6.2.1 Performance Metrics

- **Adjusted R-Squared (R^2_{adj}):** This metric adjusts the R-Squared (R^2) value for the number of predictors in the model. It is computed as:

$$R^2_{\text{adj}} = 1 - \frac{(1 - R^2)(n - 1)}{n - p - 1} \quad (7)$$

where n is the sample size and p is the number of predictors. A value close to 1 indicates a model that closely fits the data.

- **Root Mean Squared Error (RMSE):** RMSE is the square root of the average squared differences between the predicted and actual values, calculated as:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}} \quad (8)$$

where y_i are the actual observed values, $\hat{y}_i$ are the predicted values, and n is the number of observations. Lower RMSE values indicate better fitting models.

6.2.2 Learning Curves

Learning curves are crucial for understanding a model's performance as more training data is used. The model's learning curve is depicted below, which illustrates the training and cross-validation score as a function of the training set size.

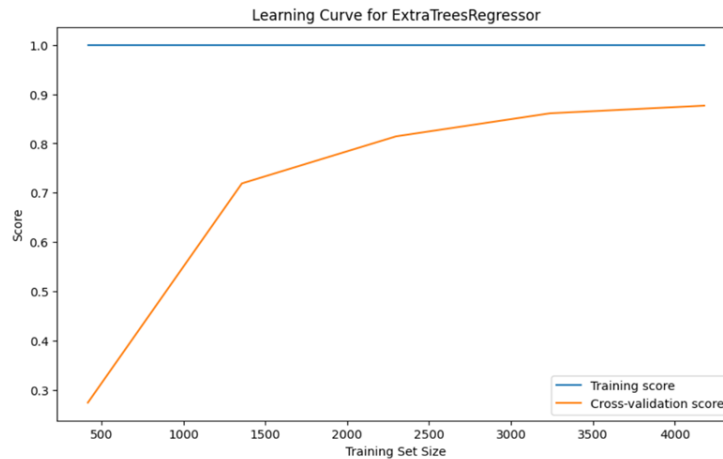


Figure 5: Learning curve for the ExtraTreesRegressor model

6.2.3 Residual Plots

Residual plots are used to visualize the differences between the actual and predicted values. These plots can reveal patterns in the residuals, suggesting issues like non-linearity or outliers.

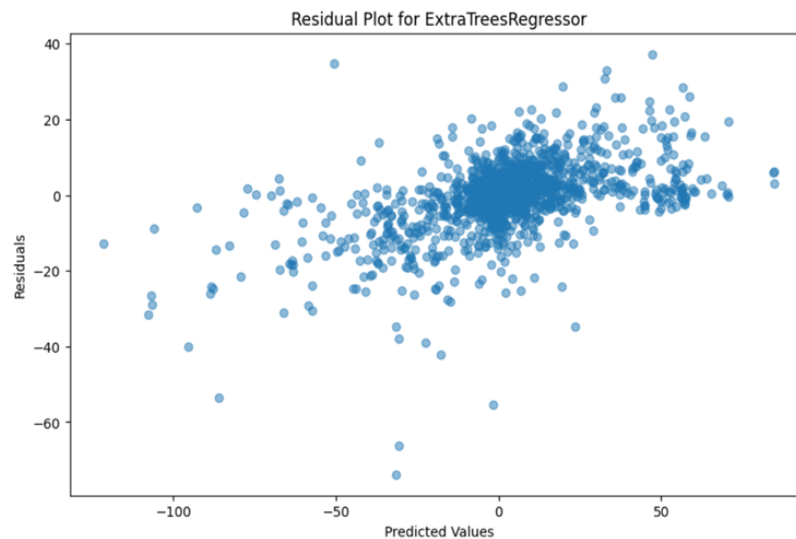


Figure 6: Residual plot for the ExtraTreesRegressor model

6.2.4 conclusion

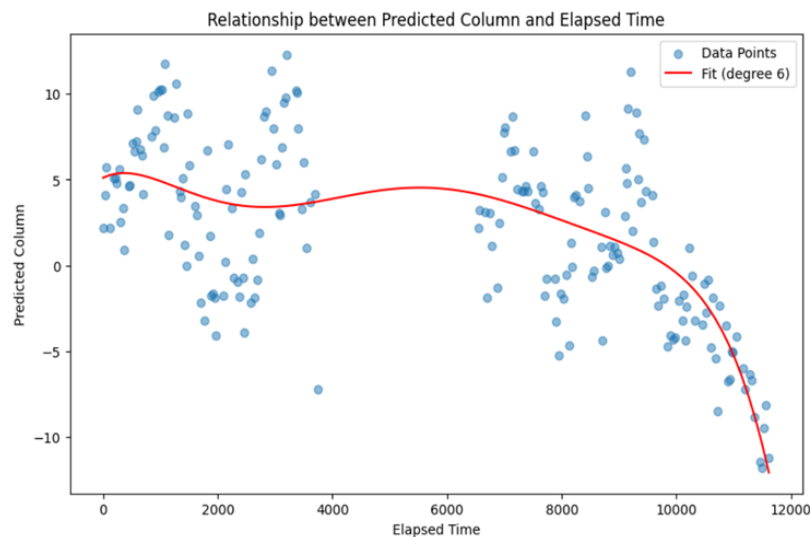
In conclusion, the ExtraTreesRegressor model demonstrates a robust predictive performance as indicated by the high Adjusted R-Squared and relatively low RMSE values. The learning curves suggest that the model benefits from additional training data, and the residual plots do not reveal any concerning patterns, indicating a good fit to the data.

6.3 Data Prediction and Analysis with ExtraTreesRegressor

6.3.1 Predicting and Visualizing the Data

Using the ExtraTreesRegressor model, we predict a specific column from our dataset and visualize the relationship between the predicted values and the elapsed time.

```
new_df1 = X
new_df1['predicted_column'] = model.predict(X)
new_df1 = new_df1.loc[new_df1.match_id == 1]
```



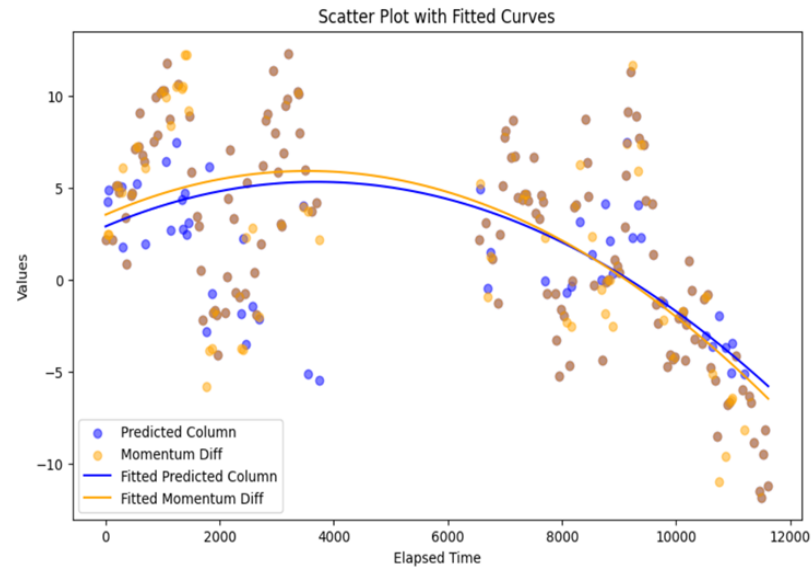
6.3.2 Curve Fitting

We fit a polynomial curve of degree 6 to our predicted data, which can be expressed mathematically as a polynomial function:

$$P(x) = a_0 + a_1x + a_2x^2 + \cdots + a_6x^6 \quad (9)$$

6.3.3 Comparing Predicted and Actual Values

The scatter plot below compares the predicted values and the actual momentum differences over time. A polynomial fitting of degree 2 is applied to both sets of data.

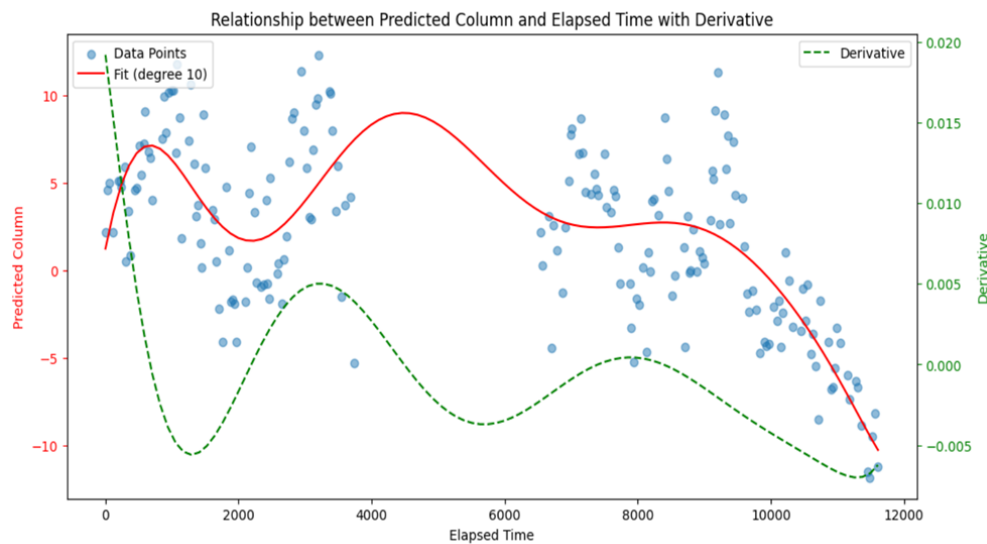


6.3.4 Analyzing the Rate of Change of Momentum

The derivative of the fitted polynomial represents the rate of change of momentum. The inflection points of the derivative can be indicative of significant moments in the dataset.

The derivative is used to obtain the rate of change of momentum. The turning point of the derivative represents the turning point of the game.

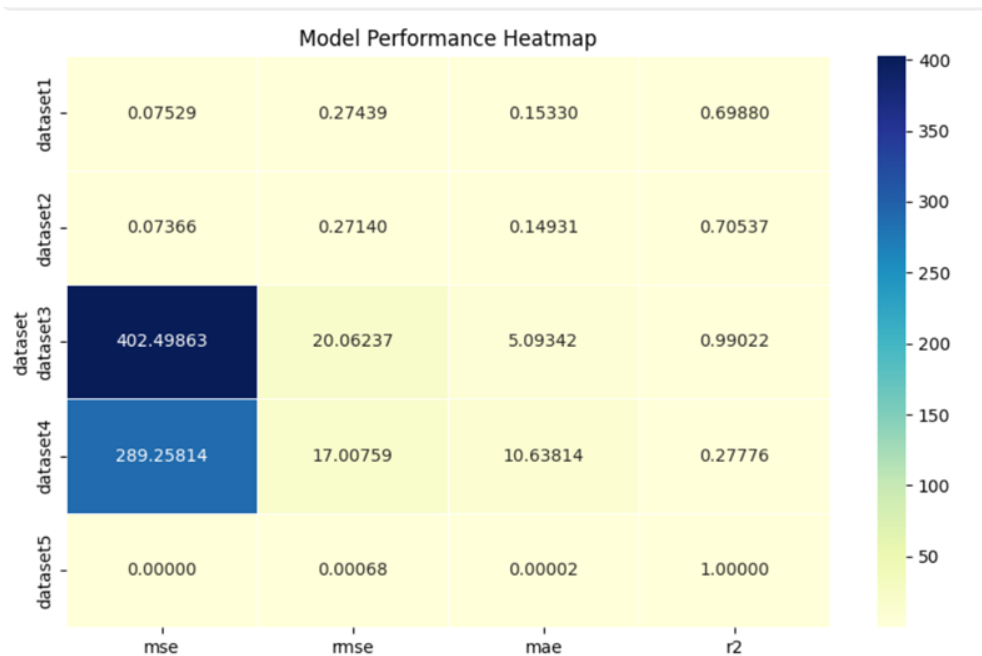
$$P'(x) = \frac{d}{dx}(a_0 + a_1x + a_2x^2 + \cdots + a_6x^6) \quad (10)$$



The dual-axis plot below shows the fitted polynomial and its derivative, offering insights into how the predicted values evolve over time and the rate at which changes occur.

7 Model generalization and related suggestions

7.1 Model generalization



We have selected four datasets from distinct matches and applied the fine-tuning method to obtain the results. The first two datasets are similar to the ones provided by the competition organizers, hence the training results are relatively reasonable. These datasets include detailed parameters such as time axis, the number of aces, and forced errors, which are crucial for a comprehensive analysis.

The latter two datasets also pertain to tennis matches but lack detailed parameters. They only include scores as a reference point, indicating the importance of technical indicators. Additionally, these datasets encompass variations in parameters such as match location and surface type. Due to the absence of more reference points, these are merely considered potential related factors.

The final dataset concerns table tennis and clearly did not meet expectations, highlighting the significant impact of the type of sport on the model, especially when the feature parameters differ substantially.

7.2 suggestions

In this section, we analyze the performance of two players in various aspects of match preparation and visually represent this analysis through a radar chart. We have selected several key indicators, such as differences in distance run, net points, and unforced errors, to reflect the overall performance of the players.

Firstly, we calculated statistical data for the positive frequency of each indicator, which is used to characterize the frequency at which a player outperforms their opponent in a

particular aspect. The calculation is as follows:

$$\text{Positive Frequency} = \frac{\text{Number of positive values in the data}}{\text{Total number of data points}} \quad (11)$$

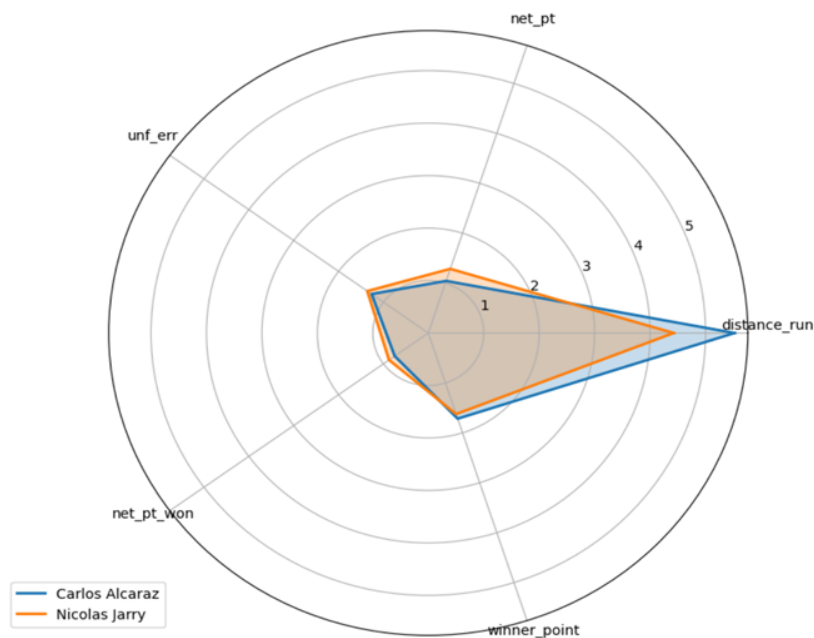
Similarly, we calculated the negative frequency to characterize the frequency at which a player underperforms compared to their opponent:

$$\text{Negative Frequency} = \frac{\text{Number of negative values in the data}}{\text{Total number of data points}} \quad (12)$$

These frequencies were then translated into scores on the radar chart, ranging from 0 to 10. The formula for score conversion is:

$$\text{Score} = \text{Frequency} \times 10 \quad (13)$$

By this method, we can quantify and compare the match preparation performance of two players. Below is an example of a radar chart that displays the scores of two players on key indicators:



This radar chart reveals the relative strengths of the players in different aspects of match preparation. For example, Player 1 shows a significant advantage in net points (net_pt), while Player 2 is more dominant in scoring points (winner_point). This method of analysis is helpful in tactical planning before matches, targeting the opponent's weaknesses, and also assists in real-time strategic adjustments during the game.

8 Future Promotion

Consideration of More Complex Features: In addition to statistical data, factors such as players' fatigue levels, psychological states, match rhythm, and intensity of confrontations could also impact match outcomes. For instance, fatigue level can be estimated based on the rest time before the match and the density of matches in a season; psychological pressure might be inferred from key points won or lost and the significance of the match (such as whether it is a final). By including these more nuanced features, our model can capture a more comprehensive range of factors affecting the match.

Model Adjustments: Different types of datasets may require different model approaches. For example, in cases where there is a great diversity of feature variables, more complex models might be needed to capture the nonlinear relationships between these features. Simultaneously, the choice of machine learning algorithms should also be adjusted according to the characteristics of the data.

Generalization Across Sports: There might be some common momentum shift patterns across different ball sports, such as patterns in athletes' energy consumption and psychological pressure management during matches. By analyzing data from different sports, we might discover these general patterns and construct a more universal model for predicting momentum shifts.

References

- [1] Burns, B. D. (2001). The hot hand in basketball: Fallacy or adaptive thinking, Proceedings of the Twenty-third Annual Meeting of the Cognitive Science Society, pp. 152–157.
- [2] Del Corral, J. (2009). Competitive balance and match uncertainty in grand-slam tennis: effects of seeding system, gender, and court surface, Journal of Sports Economics
- [3] Dietl, Helmut Nessler, Cornel. (2017). Momentum in tennis: Controlling the match. International Journal of Sport Psychology. 48. 10.7352/IJSP2017.48.459.
- [4] Hughes, M. Franks, I. (2015). Essentials of performance analysis in sport, Routledge.
- [5] Dietl, H. M. N. C. . (2017). Momentum in tennis: controlling the match. International journal of sport psychology, 48(4).
- [6] Ruizhi, L. , Zhihao, S. , Jie, R. . (2017). Competition momentum analysis model of table tennis based on double moving average. Journal of Shanghai University of Sport.
- [7] Richardson, P.A., Adler, W.P., Hanks, D.M. (1988). Game, Set, Match: Psychological Momentum in Tennis. Sport Psychologist, 2, 69-76.

Appendices

MEMORANDUM

To: MCM office

From: MCM Team 2423546

Subject: Something About Momentum

Date: February 5, 2024

Dear Sir or Madam:

In sports, we often find that a player's skill is not the only factor determining the outcome of a game. There are often different variables in play, such as the hot streaks in basketball, consecutive scoring in table tennis, and the unexpected outcomes in the 2023 Wimbledon tournament. This topic gives us an opportunity to explore the impact of momentum in games. Over four days, we discussed and attempted various models, including Markov chains, fuzzy comprehensive evaluation, Bayesian networks, etc. Finally, we chose to use logistic regression models and ExtraTreesRegressor for our research and achieved satisfying conclusions:

Momentum indeed has an impact on games, and this effect is valid for every athlete in our sample. We can indeed quantify momentum to identify turning points in a match, which can be used as a basis for timely adjustments during the game.

In the future, we plan to explore the impact of momentum in other sports such as football, basketball, badminton, and table tennis, beyond tennis. Moreover, through our research, we are more convinced that skill is not the only factor determining victory. Factors like momentum and mindset can also sway the results of a game. In some ways, this might also be why we love competitive sports.

If momentum can influence a game, I firmly believe it can also influence life. I wish everyone good momentum in 2024!

Yours Sincerely,

Team 2423546

Appendix A First appendix

Python source code for section 3 and 4:

```

#Import related dependencies, omitted here
data =pd.read_csv('Wimbledon_featured_matches.csv')
match_data = data.copy()
print("Initial data volume:", match_data.shape)
features = ['p1_sets', 'p2_sets', 'p1_games', 'p2_games', 'p1_score', 'p2_score',
            'p1_ace', 'p2_ace', 'p1_winner', 'p2_winner', 'p1_double_fault',
            'p2_double_fault', 'p1_unf_err', 'p2_unf_err',
            'p1_net_pt_won', 'p2_net_pt_won',
            'p1_break_pt_won', 'p2_break_pt_won',
            'p1_distance_run', 'p2_distance_run',
            'rally_count', 'speed_mph']
target = 'point_victor'
for feature in features:
    match_data[feature] = pd.to_numeric
    (match_data[feature], errors='coerce')
print("Data type conversion:", match_data.shape)
match_data.dropna(subset=features + [target], inplace=True)
print("After dropping missing values:", match_data.shape)
if match_data.shape[0] == 0:
    raise ValueError("No data available after preprocessing")
X = match_data[features]
y = match_data[target]
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
X_train, X_test, y_train,
y_test = train_test_split(X_scaled, y, test_size=0.3, random_state=42)
model = LogisticRegression()
model.fit(X_train, y_train)
weights = pd.Series(model.coef_[0], index=features)
print(weights.sort_values(ascending=False))

```

Appendix B Second appendix

Python source code for section5:

```

#Import related dependencies, omitted here
file_path = 'Wimbledon_featured_matches.csv'
data = pd.read_csv(file_path)
data['momentum_1'] = 0.0
data['momentum_2'] = 0.0
for index, row in data.iterrows():
    # Select features and target variables, omitted here
    features = ['momentum_1', 'momentum_2']
    target = 'point_victor'
    X = data[features]
    y = data[target]
    scaler = StandardScaler()
    X_scaled = scaler.fit_transform(X)
    X_train, X_test, y_train, y_test = train_test_split
    (X_scaled, y, test_size=0.2, random_state=42)

```

```

model = LogisticRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f": {accuracy}")
print(f": {model.coef_}")

```

Appendix C Third appendix

Python source code for section 6:

```

model=ExtraTreesRegressor()
start_time = time.time()
model.fit(X_train, y_train)
training_time = time.time() - start_time
y_pred = model.predict(X_test)
r2 = r2_score(y_test, y_pred)
rmse = np.sqrt(mean_squared_error(y_test, y_pred))
joblib.dump(model, 'model_2024C1.pkl')
print(f"Adjusted R-Squared: {r2:.2f}")
print(f"R-Squared: {r2:.2f}")
print(f"RMSE: {rmse:.2f}")
print(f"Time Taken: {training_time:.2f} seconds")
new_df1=X
new_df1['predicted_column'] = model.predict(X)
new_df1=new_df1.loc[new_df1.match_id==1]
plt.figure(figsize=(10, 6))
plt.scatter(new_df1['elapsed_time'],
new_df1['predicted_column'], alpha=0.5)
plt.title('Relationship between Predicted Column and Elapsed Time')
plt.xlabel('Elapsed Time')
plt.ylabel('Predicted Column')
plt.show()
degree = 6
coefficients = np.polyfit(new_df1['elapsed_time'],
new_df1['predicted_column'], degree)
x_values = np.linspace(min(new_df1['elapsed_time']),
max(new_df1['elapsed_time']), 100)
y_values = np.polyval(coefficients, x_values)
plt.figure(figsize=(10, 6))
plt.scatter(new_df1['elapsed_time'], new_df1['predicted_column'],
alpha=0.5, label='Data Points')
plt.plot(x_values, y_values, color='red',
label=f'Fit (degree {degree})')
plt.title('Relationship between Predicted Column and Elapsed Time')
plt.xlabel('Elapsed Time')
plt.ylabel('Predicted Column')
plt.legend()
plt.show()
plt.figure(figsize=(10, 6))
plt.scatter(new_df1['elapsed_time'], new_df1['predicted_column'],
label='Predicted Column', color='blue', alpha=0.5)
plt.scatter(new_df1['elapsed_time'], new_df1['momentum_1_momentum_2_diff'],

```

```
label='Momentum Diff', color='orange', alpha=0.5)
degree = 2
coeff_pred = np.polyfit(new_df1['elapsed_time'],
new_df1['predicted_column'], degree)
coeff_momentum = np.polyfit(new_df1['elapsed_time'],
new_df1['momentum_1_momentum_2_diff'], degree)
x_pred = np.linspace(min(new_df1['elapsed_time']),
max(new_df1['elapsed_time']), 100)
y_pred_fit = np.polyval(coeff_pred, x_pred)
x_momentum = np.linspace(min(new_df1['elapsed_time']),
max(new_df1['elapsed_time']), 100)
y_momentum_fit = np.polyval(coeff_momentum, x_momentum)
plt.plot(x_pred, y_pred_fit, label=
'Fitted Predicted Column', color='blue')
plt.plot(x_momentum, y_momentum_fit, label='Fitted Momentum Diff',
color='orange')
plt.xlabel('Elapsed Time')
plt.ylabel('Values')
plt.title('Scatter Plot with Fitted Curves')
plt.legend()
plt.show()
import numpy as np
import matplotlib.pyplot as plt
degree = 10
coefficients = np.polyfit(new_df1['elapsed_time'], new_df1['predicted_column'],
degree)
fit_polynomial = np.poly1d(coefficients)
derivative = np.polyder(fit_polynomial)
x_values = np.linspace(min(new_df1['elapsed_time']),
max(new_df1['elapsed_time']), 100)
y_values = fit_polynomial(x_values)
derivative_values = derivative(x_values)
fig, ax1 = plt.subplots(figsize=(12, 6))
ax1.scatter(new_df1['elapsed_time'], new_df1['predicted_column'],
alpha=0.5, label='Data Points')
ax1.plot(x_values, y_values, color='red', label=f'Fit (degree {degree})')
ax1.set_xlabel('Elapsed Time')
ax1.set_ylabel('Predicted Column', color='red')
ax1.tick_params('y', colors='red')
ax1.legend(loc='upper left')
ax2 = ax1.twinx()
ax2.plot(x_values, derivative_values, color='green', linestyle='--',
label='Derivative')
ax2.set_ylabel('Derivative', color='green')
ax2.tick_params('y', colors='green')
ax2.legend(loc='upper right')
plt.title('Relationship between Predicted Column and Elapsed Time with
Derivative')
plt.show()
```
