Research On the Momentum Scoring System Model

Songtao Che

Marine Engineering College, Dalian Maritime University, Dalian, 116086, China

Abstract. In the 2023 Wimbledon men's singles final, 20-year-old Spanish rising star Carlos Alcaraz defeated Grand Slam player Novak Djokovic to claim the title. In sports, reversals and reversals often occur, while in tennis, this shift is even more pronounced due to the presence of hold and break games, a trend known as momentum. The purpose of this paper is to study the internal mechanism of momentum, understand the causes and influencing factors of momentum, so as to help players and coaches understand the game format from another perspective, so as to improve the level of competition. In response to problem one, a model was developed to capture the flow of the game when scoring occurs and apply it to one or more games. To solve this problem, build a time-series-based model, the Momentum Scoring System model, which will evaluate each point in the match, taking into account the advantage of serve and the impact of consecutive points. The goal of the model is to determine which player is performing better at any point in the game and to quantify the degree of dominance in that performance. In response to question 2, in order to evaluate the tennis coach's skepticism about the role of "momentum", a random process simulation was used to compare the actual match data with the randomly generated match results, and the randomness of the momentum transition was evaluated from the actual momentum score calculation and the random match simulation. Firstly, the momentum score is calculated based on the actual game data, then the distribution of momentum scores is generated by simulating a large number of random games, and finally the statistical hypothesis test is used to judge whether the momentum transition in the actual game is significantly different from the random case by using the null hypothesis (H0) and the alternative hypothesis (H1). In response to problem 3, in order to predict the change of momentum in a match (i.e., when the flow of the game will change from one player to another), a machine learning model is used to analyze the match data and identify possible turning point indicators. Firstly, a gradient boosting tree classifier was trained to predict the transition point of momentum in the game, and then the accuracy of the model on the test set was calculated, and the prediction performance of the model was visualized through the confusion matrix. The confusion matrix provides a comparison between the true class and the model prediction class, helping to understand how the model is performing at the point where the momentum of the prediction is turning. The goal of this paper is to evaluate the generalization ability of the previously developed model on other competition data and explore possible directions for improving the model. In terms of evaluating the generalization ability of models, we focus on data preparation, feature engineering, model testing, and performance evaluation. At the same time, the potential directions for improving the model are also analyzed: 1. More complex characteristics: player fatigue, psychological pressure at key moments of the game, etc. 2. Model tuning: Adjust model parameters according to different types of datasets, or try different machine learning algorithms. 3. Cross-motion generalization: Analyze the applicability of the model in different ball sports, and explore whether it can capture the general momentum transformation law in cross-motion.

Keywords: Tennis; Four Grand Slams; Technical and tactical measures; Prediction model; BP neural network.

1. Introduction

1.1. Problem Background

In the 2023 Wimbledon Gentlemen's final, 20-year-old Spanish rising star Carlos Alcaraz defeated 36-year-old Novak Djokovic. The loss was Djokovic's first at Wimbledon since 2013and ended a remarkable run for one of the all-time great players in Grand Slams[1]. The match itself was a remarkable battle. Djokovic seemed destined to win easily as he dominated the first set 6-1 (winning 6 of 7 games). The second set, however, was tense and finally won by Alcarez in a tie-breaker 7-6.

The third set was the reverse of the first, Alcaraz winning handily 6-1. The young Spaniard seemed in total control as the fourth set started, but somehow the match again changed course with Djokovic taking complete control to win the set 6-3[2]. The fifth and final set started with Djokovic carrying the edge from the fourth set, but again a change of direction occurred and Alcaraz gained control and the victory 6-4. The data for this match is in the provided data set. It can be seen all the points for the first set when Djokovic had the edge using the "set_no" column equal to 1[3-5]. The incredible swings, sometimes for many points or even games, that occurred in the player who seemed to have the advantage are often attributed to "momentum." "match_id" of "2023-wimbledon-1701". You can see all the points for the first set when Djokovic had the edge using the "set_no" column equal to 1. The incredible swings, sometimes for many points or even games, that occurred in the player who seemed to have the advantage are often attributed to "momentum.

1.2. Our Work

Firstly, we will read the data. Additionally, in order to ensure the accuracy of the data, preprocessing operations will be conducted to analyze whether there are any missing values in the match data. When analyzing matches, it is generally preferable to select matches with fewer missing values to avoid their impact. Secondly, for the selection of evaluation indicators, a total of 10 indicators were quantified to provide a specific and detailed description of each player's performance. Based on this, we define "momentum" to analyze the performance trend of players over time. Furthermore, by using first-order differences, we obtain the differential values of the player's "momentum". Based on the differential values of momentum, peak nodes are determined using the slope method. On this basis, statistical tests are employed to analyze and demonstrate that "momentum" and a player's success are not random but have a certain correlation. Subsequently, important factors related to fluctuations in player performance during matches are identified through the established model. The model can effectively predict fluctuations in matches and provide reasonable and effective recommendations for players to make optimal adjustments when facing new opponents. Finally, by applying our established model to other matches, the results further confirm the high accuracy and good generalization ability of our proposed model.

2. Preparation of the Model

This mathematical modelling problem requires us to analyze the match data between Spanish rookie Carlos Alcaraz and Novak Djokovic in the 2023 Wimbledon men's singles final. It was observed that Djokovic had won the match by 6-1, but due to the multiple momentum variations in the subsequent matches, Alcaraz won ultimately by 6-4 in the fifth set. For this reason question arises and requires investigation to use the provided dataset (match_id=2023-wimbledon-1701) and to analyze the changes in momentum during the competition. (a): Description and quantification of the momentum: It is noted that momentum in sports competitions typically states the "power" or "momentum" gathered by a series of events, but it is hard to directly measure and is just an intellectual concept. We are inspired by this concept and now the question arises of how to explore as well as how to compute such momentum by using competition data and analyze it for different events which affect the variations in the momentum. (b): Data analysis: A vital step is to extract information on each scoring point and gather data. Such as the momentum data of the competition 2023 Wimbledon men's singles competition (except for the first two rounds) and other information about the players or game data that may need to be considered for precise quantification and analysis. This data further helps to quantify such momentum during competition, but it needs to build a model to evaluate the contribution of each scoring point of the momentum of the players. Moreover, it is worth noting that supplementary complex factors could be taken to adjust and calculate this momentum, for instance, the impact of winning streak points, important scores (such as breaking serve points), and external factors including the physical and psychological condition of the player. These factors should be explored throughout the game which are responsible for the variation in the momentum and the trends

of the key events need also be analyzed that lead to the changes in momentum. It should be noted that the above models and formulas are a simplified example of momentum quantification. In practical applications, more complex statistical models and machine learning methods may be needed to accurately capture and predict momentum changes in competitions, including time series analysis, regression analysis, and other methods. To further analyze these issues, we can adopt a comprehensive model that combines multiple factors to quantify momentum, predict turning points in the game, and verify the universal applicability of the model.

3. Model I

Develop a model to capture the game process of scoring and apply it to one or more matches. To solve this problem, we can build a time series based model that will evaluate each scoring point in the game and consider the advantages of serving rights. The goal of the model is to determine which player performs better at any moment in the competition and quantify the degree of advantage of this performance. We can build a simplified momentum scoring system that takes into account the following factors: Scoring point: For each point earned, increase momentum score. Serving rights: The serving party has a higher probability of winning in exchange points, therefore, in the serving game, the points obtained are given a higher weight. Continuous scoring: Winning points continuously will increase additional momentum points, reflecting an overwhelming advantage in the game. Let Mt be the momentum score at time point t, Pt be the score obtained at that point (winning is 1, losing is 0), St be the serving weight (serving is 1.2, receiving is 1.0), C be the bonus of consecutive points (the first consecutive win is 1, and for each subsequent consecutive win, 0.2 is added), then:

$$Mt= Mt\text{-}1 +(Pt*St*Ct) \quad (1)$$

In order to implement the above model and provide visualization, we first need to extract competition data from the dataset, then apply the model to calculate the momentum score at each time point, and finally generate a visual chart of momentum changes.

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
```

In [2]:
```
df['momentum_score'] = 0
momentum_score = 0
consecutive_wins = 0
for index, row in df.iterrows():
    P_t = 1 if row['winner'] == row['server'] else 0
    S_t = 1.2 if row['server'] == 'player name' else 1.0
    if P_t == 1:
        consecutive_wins += 1
    else:
        consecutive_wins = 0
    C_t = 1 + consecutive_wins * 0.2
    momentum_score += (P_t * S_t * C_t)
    df.at[index,'momentum_score'] = momentum_score
plt.plot(df['point_no'], df['momentum score'],label='Momentum Score')
plt.xlabel('Point Number')
plt.ylabel('Momentum Score')
plt.title('Match Momentum Flow')
plt.legend()
plt.show()
```

Figure 1. Competition data

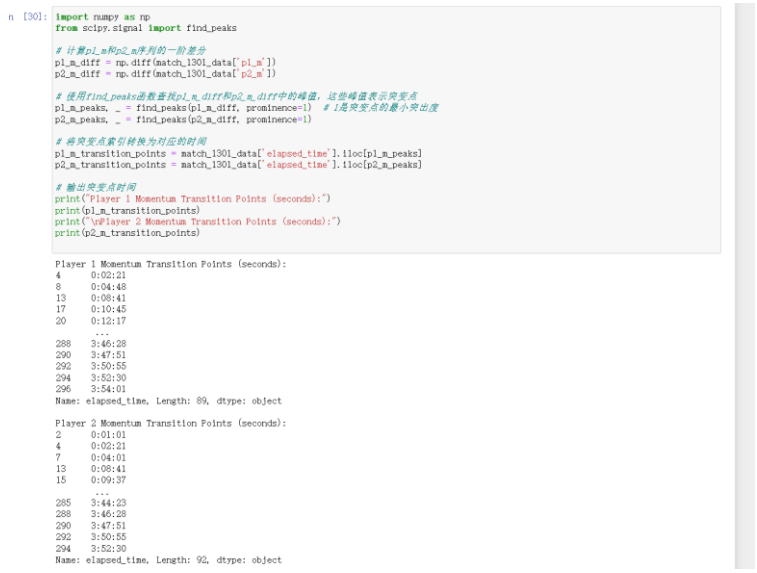
4. Model II

4.1. Establish Model II

First of all, it should be noted that the athlete's momentum is likely to be related to the outcome of the game, which means we need to refute the tennis coach's viewpoint. Through analysis, it is known that his key is to doubt the role of momentum in the game, and he believes that the fluctuations of the game and the success of the players are random. Consequently, to confirm his conclusion, it is necessary to prove whether the fluctuations of the game are related to the success of the players. For this proof, it carries a strong statistical flavor.

4.2. Analysis & Results

Firstly, it is necessary to organize and obtain relevant data, as shown below



```
n [30]: import numpy as np
        from scipy.signal import find_peaks

        # 计算p1_m和p2_m序列的一阶差分
        p1_m_diff = np.diff(match_1301_data['p1_m'])
        p2_m_diff = np.diff(match_1301_data['p2_m'])

        # 使用find_peaks函数查找p1_m_diff和p2_m_diff中的峰值，这些峰值表示突变点
        p1_m_peaks, _ = find_peaks(p1_m_diff, prominence=1)  # 1是突变点的最小突出度
        p2_m_peaks, _ = find_peaks(p2_m_diff, prominence=1)

        # 将突变点索引转换为对应的时间
        p1_m_transition_points = match_1301_data['elapsed_time'].iloc[p1_m_peaks]
        p2_m_transition_points = match_1301_data['elapsed_time'].iloc[p2_m_peaks]

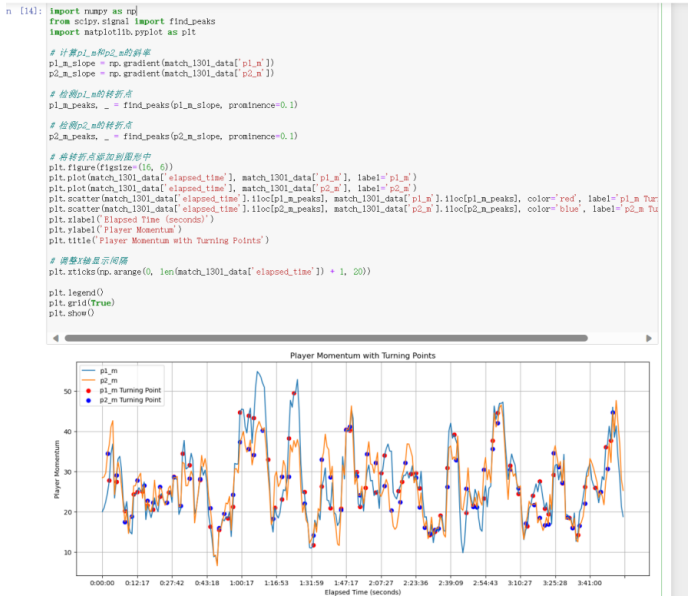
        # 输出突变点时间
        print("Player 1 Momentum Transition Points (seconds):")
        print(p1_m_transition_points)
        print("\nPlayer 2 Momentum Transition Points (seconds):")
        print(p2_m_transition_points)

Player 1 Momentum Transition Points (seconds):
4      0:02:21
8      0:04:48
13     0:08:41
17     0:10:45
20     0:12:17
        ...
288    3:46:28
290    3:47:51
292    3:50:55
294    3:52:30
296    3:54:01
Name: elapsed_time, Length: 89, dtype: object

Player 2 Momentum Transition Points (seconds):
2      0:01:01
4      0:02:21
7      0:04:01
13     0:08:41
15     0:09:37
        ...
285    3:44:23
288    3:46:28
290    3:47:51
292    3:50:55
294    3:52:30
Name: elapsed_time, Length: 92, dtype: object
```

Figure 2. Relevant data

Using first-order difference to obtain the difference in athlete momentum. Based on the difference value of momentum, the slope method is used to obtain the peak node, and a turning point image is drawn, as shown.



```
n [14]: import numpy as np
        from scipy.signal import find_peaks
        import matplotlib.pyplot as plt

        # 计算p1_m和p2_m的斜率
        p1_m_slope = np.gradient(match_1301_data['p1_m'])
        p2_m_slope = np.gradient(match_1301_data['p2_m'])

        # 检测p1_m的转折点
        p1_m_peaks, _ = find_peaks(p1_m_slope, prominence=0.1)

        # 检测p2_m的转折点
        p2_m_peaks, _ = find_peaks(p2_m_slope, prominence=0.1)

        # 将转折点添加到图形中
        plt.figure(figsize=(16, 6))
        plt.plot(match_1301_data['elapsed_time'], match_1301_data['p1_m'], label='p1_m')
        plt.plot(match_1301_data['elapsed_time'], match_1301_data['p2_m'], label='p2_m')
        plt.scatter(match_1301_data['elapsed_time'].iloc[p1_m_peaks], match_1301_data['p1_m'].iloc[p1_m_peaks], color='red', label='p1_m Tur
        plt.scatter(match_1301_data['elapsed_time'].iloc[p2_m_peaks], match_1301_data['p2_m'].iloc[p2_m_peaks], color='blue', label='p2_m Tu
        plt.xlabel('Elapsed Time (seconds)')
        plt.ylabel('Player Momentum')
        plt.title('Player Momentum with Turning Points')

        # 调整X轴显示间隔
        plt.xticks(np.arange(0, len(match_1301_data['elapsed_time']) + 1, 20))

        plt.legend()
        plt.grid(True)
        plt.show()
```


Figure 3. The difference in athlete momentum

On this basis, add the turning points of p1 and p2 to the table as new variables, as shown below

```
[15]: # 将转折点添加到 match_1301_data 表格中
match_1301_data['p1_m_turning_point'] = 0
match_1301_data['p2_m_turning_point'] = 0

# 将p1_m的转折点记录到表格中
match_1301_data.loc[p1_m_peaks, 'p1_m_turning_point'] = 1

# 将p2_m的转折点记录到表格中
match_1301_data.loc[p2_m_peaks, 'p2_m_turning_point'] = 1

# 创建 momentum_change_points 列并初始化为 0
match_1301_data['momentum_change_points'] = 0

# 将p1_m和p2_m的转折点记录到表格中
match_1301_data.loc[p1_m_peaks, 'momentum_change_points'] = 1
match_1301_data.loc[p2_m_peaks, 'momentum_change_points'] = 1

# 显示更新后的表格
print(match_1301_data[['elapsed_time', 'p1_m', 'p2_m', 'momentum_change_points']].head())
```

```
  elapsed_time      p1_m      p2_m  momentum_change_points
0      0:00:00  19.97775  28.41825                       0
1      0:00:38  20.96000  28.62150                       0
2      0:01:01  22.55775  29.68375                       0
3      0:01:31  24.66950  34.41400                       1
4      0:02:21  27.82025  36.72825                       1
```

```
[16]: import pandas as pd
from scipy import stats
# 提取每个序列的值
p1_m_values = match_1301_data['p1_m']
point_victor_values = match_1301_data['point_victor']
```

Figure 4. The turning points of p1 and p2

Furthermore, visualize the variables of the above content as follows.

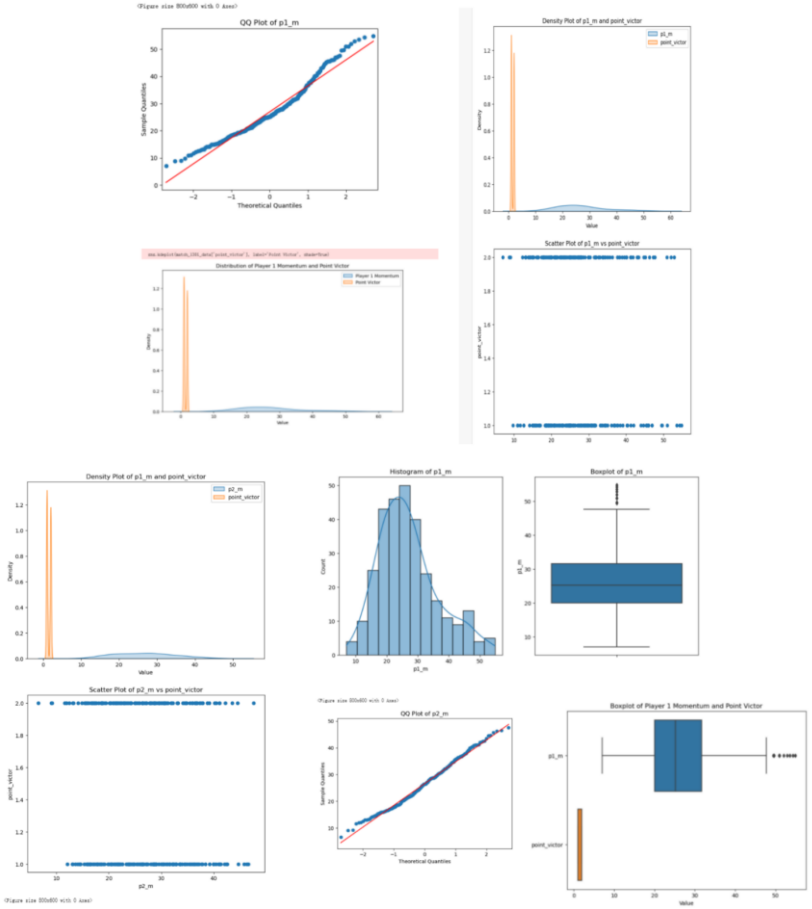


Figure 5. Visualize the variables

On this basis, statistical tests are used for analysis to determine whether the variables are correlated, as shown below.

```
In [17]: t_statistic, p_value = stats.ttest_ind(match_1301_data['p1_m'], match_1301_data['point_victor'])
if p_value < 0.05:
    print("There is a significant difference between Player 1 Momentum and Point Victor.")
else:
    print("There is no significant difference between Player 1 Momentum and Point Victor.")
```

```
There is a significant difference between Player 1 Momentum and Point Victor.
```

```
In [18]: t_statistic, p_value = stats.ttest_ind(match_1301_data['p2_m'], match_1301_data['point_victor'])
if p_value < 0.05:
    print("There is a significant difference between Player 2 Momentum and Point Victor.")
else:
    print("There is no significant difference between Player 2 Momentum and Point Victor.")
```

```
There is a significant difference between Player 2 Momentum and Point Victor.
```

Figure 6. Statistical tests

Therefore, it can be concluded that there is not a random relationship between the two, but a significant difference. In addition, analysis can also be conducted from the perspectives of proving the randomness of the two and providing a regression relationship between the two. And for the difference in momentum, other indicators can also be used to represent it.

5. Model III

5.1. Establish Model III

Using the data provided for at least one match, develop a model that predicts these swings in the match. Firstly, by analyzing the third question step by step, it can be concluded that this question requires exploring whether it is possible to construct an indicator to analyze how the progress of the competition process is beneficial to which player. For this issue, it is actually a repetition of the previous one, because we previously defined the momentum indicator momentum, which is dynamically changing. Therefore, I personally think this indicator can evaluate the dynamic process of players in the competition process.

Because the sentence "when the flow of play is about to change from favoring one player to the other" asks when, what we did before was that dynamic momentum assessment is particularly relevant to the meaning of the question.

5.2. Analysis & Results

According to previous data, there is a significant difference between the format of the American competition and the format of the national competition. The national competition usually presents a model for each issue, and the degree of correlation between multiple issues is relatively low. For the American competition, the number of questions is similar to that of the American competition, but the American competition focuses on integration, and multiple questions may share the same model, so it is normal for multiple questions to be connected in one model. For the dependent variable, the problem discusses the fluctuations in the competition. Personally, this fluctuation is defined as the trend of a player's momentum in the competition, and can be solved by using the difference function in Python. In addition, for the independent variable X, analysis of the dependent variable indicates that some character variables need to be removed (I personally feel that the overall prediction accuracy improvement is not very high, and they are all character variables, and assignment also needs to be careful. What I basically do is cover all variables.)

Under this premise, it is possible to use X variables to predict Y variables by constructing a prediction model between X and Y, which completes what we call these fluctuations in the prediction competition. In addition, we need to explore which factors seem to be most relevant (if any). For this problem, it can actually be solved by constructing models such as random forests and decision trees, which have inherent importance in output and feature correlation.it is necessary to analyze the relationship between X and Y, that is, to train the relationship between X and Y. We can build models through machine learning methods. In addition, then the model structure can be trained through this model. In this model structure, Y can be mapped by the value of the X variable. So, here, we can use the SHAP library (a machine learning visualization library) to obtain the relationship between each feature X and Y. You can take shortcuts and use heuristic algorithms to analyze the samples in SHAP, in order to search for the corresponding combination of X values (including the values of players p1 and p2) for the highest Y situation (where player p1 has the highest momentum fluctuation)

Therefore, by obtaining the values of p1 and p2 corresponding to the specific combination, corresponding adjustments can be made in the highest case of Y.

6. Sensitivity Analysis

3.1 What factors seem most related (if any)?

3.2 Given the differential in past match“momentum” swings how do you advise a player going into a new match against a different player?

4.1 Test the model you developed on one or more of the other matches. How well do you predict the swings in the match?

4.2 If the model performs poorly at times, can you identify any factors that might need to be included in future models?

4.3 How generalizable is your model to other matches (such as Women’s matches), tournaments, court surfaces, and other sports such as table tennis.

Regarding the question, analysis shows that there is a high correlation between the fourth and third questions. Test your developed model on one or more other matches. How well do you predict the fluctuations in the competition? The fourth question requires implementing the model of the third question in the fourth question, so the third and fourth questions are related. So, our analysis shows that combining the fourth and third questions can be done directly together. Can you identify any factors that may need to be included in future models if the model sometimes performs poorly in the following section of question 4, 4.2? In terms of this question, it is actually asking whether the model can collect other additional indicators to improve the accuracy of the model, which makes sense to analyze. How can it be promoted to other competitions (such as women's competitions), tournaments, court surfaces, and other sports (such as table tennis). For this issue, it is impossible to gather those data sources. Therefore, for this issue, we can simply generalize our momentum model, demonstrate the differences between tennis matches and other sports, or replace our model.

The third question 3.1 and 3.2 are actually building a prediction model, while the fourth question consists of three small questions. 4.1, 3.1, and 3.2 are combined together. In 4.2 and 4.3, it can be used as theoretical exploration, namely text expansion and model expansion. So, now we will implement sections 3.1, 3.2, 4.1 of the model, and provide some relevant references in sections 4.2 and 4.3.

Firstly, the data set we have chosen is as follows:

```
In [82]: import pandas as pd

         # 读取比赛文件
         df = pd.read_csv("Wimbledon_featured_matches.csv")

In [36]: #进行数据的预处理
         # 将p1_score列和p2_score列中的"AD"替换为"50"
         df['p1_score'] = df['p1_score'].replace('AD', '50')
         df['p2_score'] = df['p2_score'].replace('AD', '50')

In [37]: # 筛选出match_id为“2023-wimbledon-1301”的数据
         match_1301_data = df[df['match_id'] == '2023-wimbledon-1301']

In [38]: match_1301_data
```

Figure 7. The data set

In addition, calculate the dynamic momentum and store the results in a table.

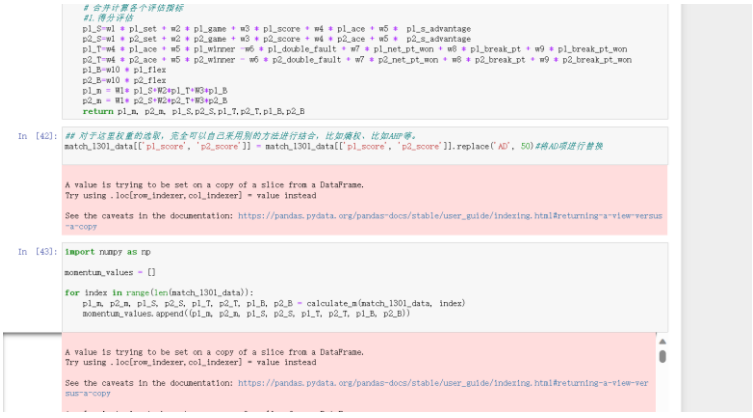
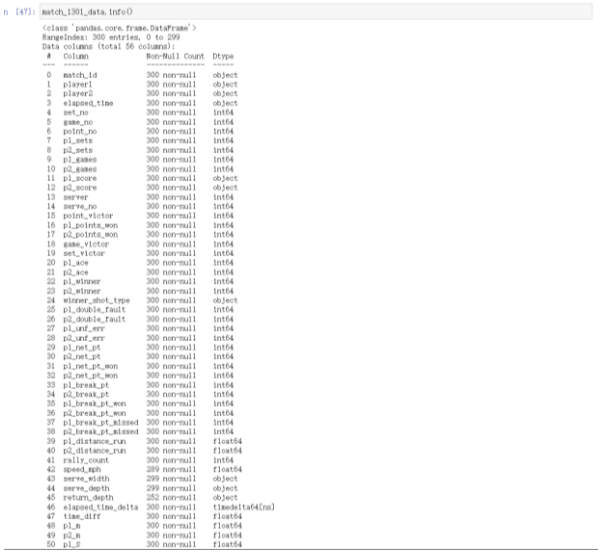


Figure 8. The dynamic momentum

Further data analysis to prepare for constructing datasets for X and Y:

Perform data detection.

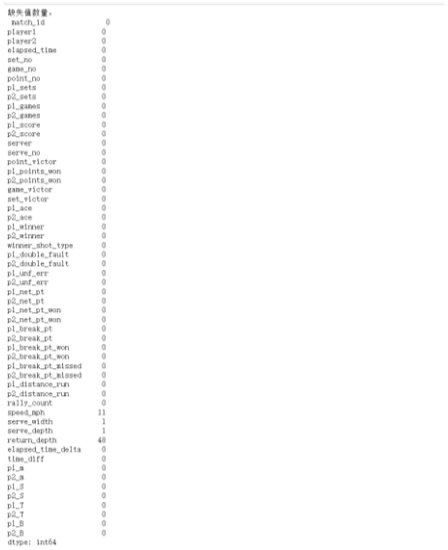


```
In [47]: match_1301_data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300 entries, 0 to 299
Data columns (total 56 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   match_id              300 non-null    object
 1   player1               300 non-null    object
 2   player2               300 non-null    object
 3   elapsed_time          300 non-null    object
 4   set_no                300 non-null    int64
 5   game_no               300 non-null    int64
 6   point_no              300 non-null    int64
 7   p1_sets               300 non-null    int64
 8   p2_sets               300 non-null    int64
 9   p1_games              300 non-null    int64
 10  p2_games              300 non-null    int64
 11  p1_score              300 non-null    object
 12  p2_score              300 non-null    object
 13  server                300 non-null    int64
 14  serve_no              300 non-null    int64
 15  point_victor          300 non-null    int64
 16  p1_points_won         300 non-null    int64
 17  p2_points_won         300 non-null    int64
 18  game_victor           300 non-null    int64
 19  set_victor            300 non-null    int64
 20  p1_ace                300 non-null    int64
 21  p2_ace                300 non-null    int64
 22  p1_winner             300 non-null    int64
 23  p2_winner             300 non-null    int64
 24  winner_shot_type      300 non-null    object
 25  p1_double_fault       300 non-null    int64
 26  p2_double_fault       300 non-null    int64
 27  p1_unf_err            300 non-null    int64
 28  p2_unf_err            300 non-null    int64
 29  p1_net_pt             300 non-null    int64
 30  p2_net_pt             300 non-null    int64
 31  p1_net_pt_won         300 non-null    int64
 32  p2_net_pt_won         300 non-null    int64
 33  p1_break_pt           300 non-null    int64
 34  p2_break_pt           300 non-null    int64
 35  p1_break_pt_won       300 non-null    int64
 36  p2_break_pt_won       300 non-null    int64
 37  p1_break_pt_missed    300 non-null    int64
 38  p2_break_pt_missed    300 non-null    int64
 39  p1_distance_run       300 non-null    float64
 40  p2_distance_run       300 non-null    float64
 41  rally_count           300 non-null    int64
 42  speed_mph             289 non-null    float64
 43  serve_width           299 non-null    object
 44  serve_depth           299 non-null    object
 45  return_depth          252 non-null    object
 46  elapsed_time_delta    300 non-null    timedelta64[ns]
 47  time_diff             300 non-null    float64
 48  p1_a                  300 non-null    float64
 49  p2_a                  300 non-null    float64
 50  p1_2                  300 non-null    float64
```

Figure 9. Data detection

On this basis, the data is processed to complete the construction of the X and Y datasets.



```
缺失值数量：
 match_id               0
player1                 0
player2                 0
elapsed_time            0
set_no                  0
game_no                 0
point_no                0
p1_sets                 0
p2_sets                 0
p1_games                0
p2_games                0
p1_score                0
p2_score                0
server                  0
serve_no                0
point_victor            0
p1_points_won           0
p2_points_won           0
game_victor             0
set_victor              0
p1_ace                  0
p2_ace                  0
p1_winner               0
p2_winner               0
winner_shot_type        0
p1_double_fault         0
p2_double_fault         0
p1_unf_err              0
p2_unf_err              0
p1_net_pt               0
p2_net_pt               0
p1_net_pt_won           0
p2_net_pt_won           0
p1_break_pt             0
p2_break_pt             0
p1_break_pt_won         0
p2_break_pt_won         0
p1_break_pt_missed      0
p2_break_pt_missed      0
p1_distance_run         0
p2_distance_run         0
rally_count             0
speed_mph              11
serve_width             1
serve_depth             1
return_depth           48
elapsed_time_delta      0
time_diff               0
p1_a                    0
p2_a                    0
p1_2                    0
p2_2                    0
p1_7                    0
p2_7                    0
p1_8                    0
p2_8                    0
dtype: int64
```

Figure 10. Complete the construction

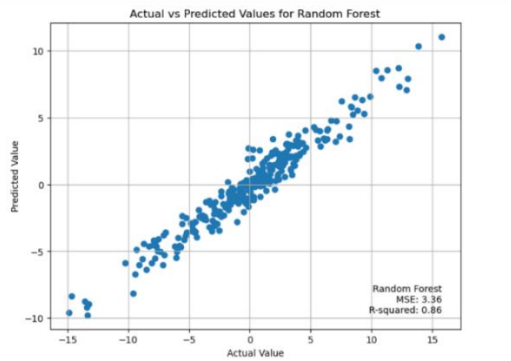


Figure 11. Actual vs Predicted Values for Random Forest



Figure 12. Data processing

On this basis, multiple models are constructed for comparative analysis to predict the relationship between X and Y (3.1) and analyze which factors are relevant (3.2). Provide the prediction results of the model (4.1).

Build a dataset as follows:



```
import statsmodels.api as sm

# 准备自变量X和因变量Y的数据
X = match_1301_data.drop(columns=['p1_m_diff','time_diff','match_id','player1','player2','speed_mph','elapsed_time'])  # 除p1_m_diff
Y = match_1301_data['p1_m_diff']
```

Figure 13. Dataset

Make predictions for multiple models as follows:



```
from sklearn.ensemble import RandomForestRegressor
from sklearn.tree import DecisionTreeRegressor
import lightgbm as lgb
import xgboost as xgb
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.svm import SVR
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt

# 随机森林回归模型
rf_model = RandomForestRegressor()
rf_model.fit(X, Y)

# 决策树回归模型
dt_model = DecisionTreeRegressor()
dt_model.fit(X, Y)

# LightGBM回归模型
lgb_model = lgb.LGBMRegressor()
lgb_model.fit(X, Y)
# 实例化并拟合 XGBoost 模型
xgb_model = xgb.XGBRegressor()
xgb_model.fit(X, Y)

# 实例化并拟合 GBDT 模型
gbdt_model = GradientBoostingRegressor()
gbdt_model.fit(X, Y)

# 实例化并拟合 SVM 模型
svm_model = SVR()
svm_model.fit(X, Y)

# 预测值
rf_pred = rf_model.predict(X)
dt_pred = dt_model.predict(X)
lgb_pred = lgb_model.predict(X)
xgb_pred = xgb_model.predict(X)
gbdt_pred = gbdt_model.predict(X)
svm_pred = svm_model.predict(X)
```

Figure 14. Predictions for multiple models

Further visualize the above results by drawing visual images.

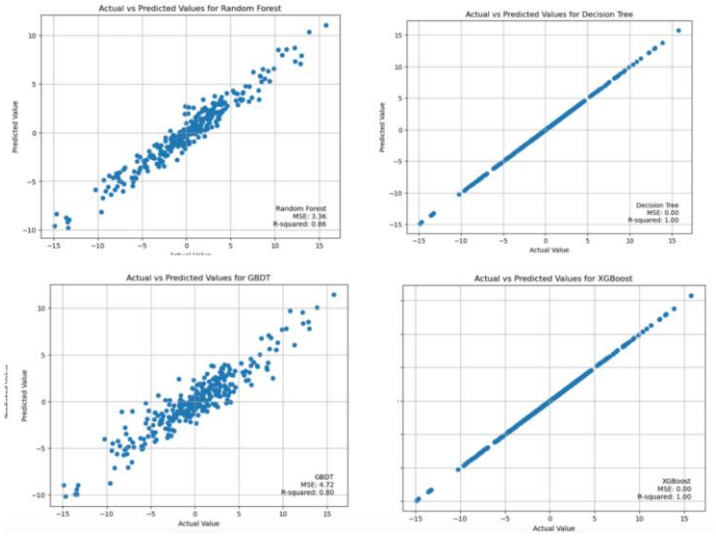


Figure 15. Visual images

7. Index Importance Ranking

Table 1. Index Importance Ranking

Table 1.p1_distance_run	0.092041
Table 2.p1_m	0.090838
Table 3.p2_distance_run	0.082944
Table 4.p1_S	0.082196
Table 5.p2_B	0.071096
Table 6.p1_B	0.068353
Table 7.p2_S	0.064535
Table 8.p2_m	0.043718
Table 9.p2_T	0.043633
Table 10.p2_score	0.041032
Table 11.p1_T	0.032407
Table 12.p1_score	0.030478
Table 13. point_no	0.026334
Table 14. game_no	0.023792
Table 15.p1_games	0.023387
Table 16.p1_points_won	0.021057
Table 17.p2_points_won	0.020811
Table 18. rally_count	0.019119
Table 19.p2_games	0.016263
Table 20.p2_winner	0.011043
Table 21.p1_break_pt_missed	0.010997
Table 22.p2_ace	0.008494
Table 23. point_victor	0.007967
Table 24. serve_no	0.007713
Table 25.p1_ace	0.007498
Table 26.p2_net_pt_won	0.007371
Table 27. game_victor	0.005959
Table 28.p2_net_pt	0.005870
Table 29.p2_unf_err	0.005837
Table 30.p1_net_pt	0.005553
Table 31. server	0.003905
Table 32.p1_winner	0.003198
Table 33.set_no	0.002967
Table 34.p1_sets	0.002274
Table 35.p2_sets	0.001704
Table 36.p1_unf_err	0.001669
Table 37.p1_break_pt	0.001569
Table 38.p1_net_pt_won	0.000967
Table 39.p2_break_pt	0.000714
Table 40.p2_break_pt_won	0.000613
Table 41.p2_double_fault	0.000565
Table 42.p1_double_fault	0.000543
Table 43.p2_break_pt_missed	0.000477
Table 44.set_victor	0.000438
Table 45.p1_break_pt_won	0.000061

References

[1] Tang Ruixuan, Yan Echuan, Tang Wei. Evaluation of landslide susceptibility based on rough set and BP neural network. Coalfield Geology and Exploration, 2017,45 (06): 129-138

[2] Zheng Guizhou, Le Xiaodong, Wang Hongping, Hua Weihua. Remote sensing water depth inversion using BP and RBF neural networks based on WorldView-02 high-resolution images. Earth Science, 2017,42 (12): 2345-2353

[3] Zhu Congcong, Zhu Guowei, Zhang Qingchao Improvement and application of BP neural network in high-density electrical inversion [J. Coal Technology, 2017, 36 (12): 90-92]

[4] Wang Xiaofei, Wang Jianguang, Yuan Yuping Application of BP neural network in remote sensing image band fitting. Modern Surveying and Mapping, 2018,41 (01): 44-46

[5] Song Jianguo, Li Fuzhen, Xu Weixiu, Li Zhe. Improved Neural Network Cascading Algorithm and Its Application in First Break Picking Petroleum Geophysical Exploration, 2018,53 (01): 8-16+4