| Problem Chosen | 2024 | Team Control Number |
|---|---|---|
| C | MCM/ICM<br>Summary Sheet | 2425258 |

# A prediction system based on momentum formulation and XGBoost for tennis matches

## Summary

In a tennis match, changes in the match are often sudden, which makes it difficult to detect upcoming changes in the match, but through our modeling and analysis, our momentum formula and XGBoost-based prediction system have achieved better results in predicting match fluctuations, which allows players to detect the direction of the match ahead of time and make timely adjustments during the match.

**For Problem 1**, we first define "momentum", which is numerically the weighted sum of Winning Percentage, Serve Advantage Factor, and Set Advantage over a recent period of time. Since it is not possible to get the exact winning percentage in the actual match, we use ARIMA model to predict the score of the last five games to get an approximation of "momentum". "momentum" is a good indicator of a player's condition and can be used to directly assess the performance of a player.

**For Problem 2**, we first simulate a random game and calculate the "momentum" of the random game by the method in Problem 1, then we compare the score and "momentum" of the real game with those of the random game by the Kolmogorov-Smirnov test, if they have the same distribution. After that, we compare whether the score and momentum of the real game and the score and momentum of the randomized game are the same distribution by the Kolmogorov-Smirnov test. Through detailed calculation, we can get that there is a big difference between the random match and the real match, so the fluctuation of the tennis match is not random.

**For Problem 3**, in order to obtain the movement of the player's advantage in the stream of matches, we first define the turning point, and fit XGBoost with the turning point as a label, and momentum, number of consecutive games, etc. As a result, we have achieved good results in predicting the matches. We also classify various features of the current match, and the most helpful feature for classification is the one that is most relevant to the movement of the player's advantage. Based on this, we give suggestions that are favorable to the players.

**For problem 4**, we first used the turning point prediction system for all the games in the dataset, and the resulting metrics show that our system has some generalization ability. However, the momentum and prediction system still needs to be further improved and supplemented for other ball games.

**Keywords: ARIMA, XGBoost, momentum, turningpoint**

# Contents

# 1. Introduce

## 1.1 Background of the problem

In the men's singles final of the 2023 Wimbledon Open, 20-year-old Spanish star Carlos.

Alcaraz defeated 36-year-old Novak Djokovic to end the great player's Grand Slam winning streak. Djokic, ending the great player's Grand Slam winning streak. The exciting twists and turns in the course of a match are often attributed to the shifts in "momentum" between the two players. Therefore, the accurate prediction of "momentum" plays an indispensable role in enabling athletes to make reasonable adjustments and grasp the direction of the match.

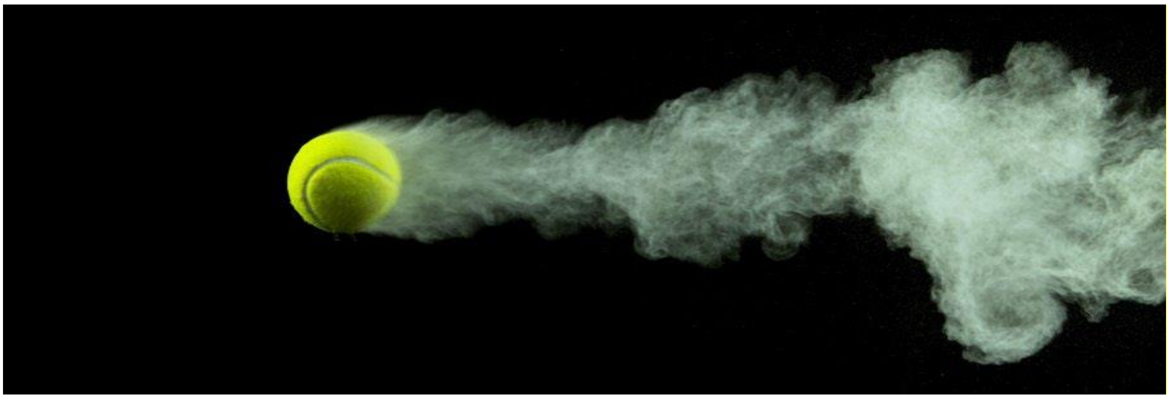

**figure1**

## 1.2 Restatement of the problem

After through in-depth analysis and research on the background of the problem, we can specify that

Our article should cover the following aspects.

- Develop a match flow model to identify player performance and visualize the match progression.
- Apply the model/metric to evaluate the tennis coach's assertion about random swings and success runs.
- Explore indicators predicting changes in match momentum; offer advice based on past differentials for new matchups.
- Test the developed model on additional matches, assess effectiveness, and identify factors for potential model enhancement.

## 1.3 We work

In this problem, we were asked to develop a model to capture the flow of score changes in a game, as well as twists and turns in the course of a game, and apply it to other games, including those of different types, venues, and even other sports, in order to give actionable advice.

The model focuses on data collection using Wimbledon_featured_matches for both sides of the 2023 Wimbledon Open singles final.

To solve the problem shown in 1.2, we did the following:

- As preparation, we mined the data and checked the plausibility of the dataset, and selected, cleaned and normalized the test data.
- We developed an ARIMA time series model to evaluate match points. On this basis, we quantified the real-time "momentum" of both players using the match score as an indicator, taking into account the advantage of the serve.

- In order to verify the inevitable influence of "momentum" on the game process, we use a stochastic process simulation to compare the data of the actual game with the randomly generated game results, so as to verify the stochasticity of the momentum transformation.
- We built a turning point prediction model based on XGBoost, by inputting some features, the model can output which side the game will be in favor of, this model can effectively help the players to find out the change of their situation in time and make timely adjustments. Better suggestions can be made for tournament players accordingly.

## 2. Assumptions and justifications

In order to simplify the problem, we have made the following reasonable assumptions based on the actual situation of the game and the relevant data, each of which has been well verified for its correctness.

- All match data is real
- Both players are not affected by non-playing factors that could lead to major mood swings prior to taking the court, and both players are technically stable, and the process of the game can proceed normally without interruptions
- Assuming that the current game score has a strong correlation with previous game scores, we can use the previous game scores to predict the next scores
- External factors such as venue, weather conditions, clothing and hair accessories have very little influence on the course of the competition and are not taken into account.

## 3. Notations

For the sake of what follows, we have listed some of the main notations here

Momentum Definition:

| | |
|---|---|
| $W_p$ | Winning this round |
| $W_{server}$ | Serving team wins the round |
| $\delta_{server}$ | Winning Gains from Serving |
| $P_{i,t}$ | the look of things |
| $l_{window}$ | Sliding window for counting wins |
| $E_{window}(W_p)$ | Mathematical expectation of the number of winning rounds |
| $D_{set}$ | Difference between current player and opponent's handicap |
| $\lambda$ | $D_{set}$Difference coefficients for |

Time series modeling:

| | |
|---|---|
| $y_t$ | Observations of the event sequence at moment t |
| $\phi$ | autoregressive coefficient |
| $\varepsilon_t$ | white noise error term |

| | |
|---|---|
| $\theta_q$ | moving average coefficient |

Randomness test:

| | |
|---|---|
| $F_{1,n}(X)$ | Distribution function of actual match experience |
| $F_{2,m}(X)$ | Predictive match experience distribution function |
| $P_{value}$ | Predictive indicators |
| $D_N$ | statistic |

Match flow factor analysis:

| | |
|---|---|
| $\delta_{dif}$ | Points difference between the two real-time games |
| $grad(\delta_{dif})$ | derivative of a difference (math.) |
| $p1_{pw}$ | Player 1's real-time score |
| $p2_{pw}$ | Player 2's real-time score |

# 4. Problem 1

In this section, based on the available data, we chose to use an ARIMA model to predict the change in scoring in the succeeding match, evaluating each scoring point of the match and taking into account the advantage of the serve, which in turn utilizes the results of the ARIMA model to give a measure for determining which player is performing better at any given moment of the match - the - "momentum".

## 4.1 Definition of "momentum"

As we all know, the level of a tennis player's performance on the court is not only related to the player's own strength, but also affected by a number of factors, including but not limited to whether the player has made mistakes, the current score, and the opponent's status. The level of performance is reflected in the player's score in the following rounds and even in the victory or defeat of the whole game. If a player's score goes up, we can say that the player has a lot of "momentum". In other words, "momentum" is a proxy for the player's current level of play. In the next two sections, we will define "momentum" in more detail.

### 4.1.1 Serve dominance factor

In tennis, the probability of winning the score is much higher for the serving team, and we would like to incorporate this factor into the model in some way, which requires us to quantify the gain from serving. We therefore introduce the concept of **a serve advantage factor**.

Let's assume that who serves only has a positive effect on who wins the set in the current round, and has no effect on who wins the next round or on who receives the serve. Let the event win this round be $W_p$ , the event serve win this round be $W_{server}$ , and the gain of serving for the server be $\delta_{server}$ . The gain to the server from serving to win the round is.

$$\delta_{server} = P(W_{server}|W_p) = \frac{\sum W_{server}}{\sum W_p}(4.1.1)$$

Statistically, the probability that the server will win the serve round is about$67.341\%$ , and the server does have a certain advantage, which is in line with our expectation. We set the serve advantage factor as a constant$0.67$ , i.e.:$\delta_{server} = 0.67$ .

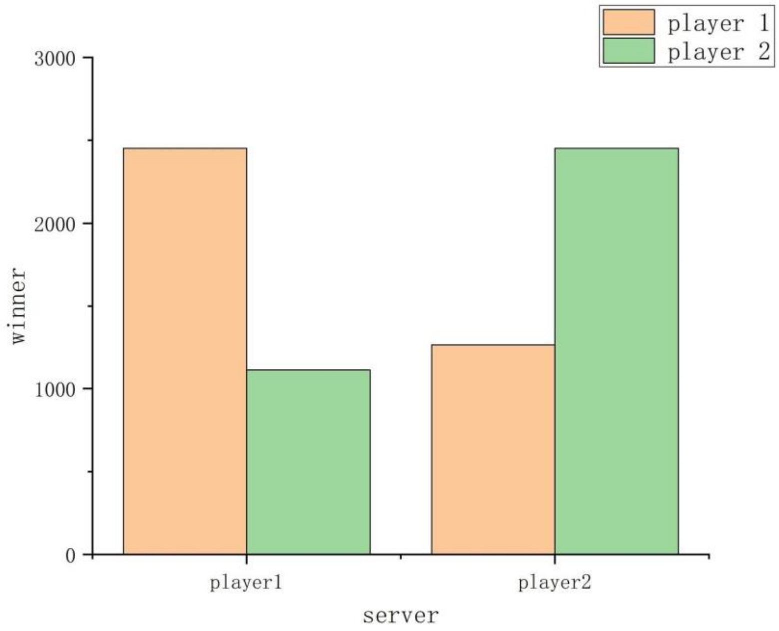

**figure2**

### 4.1.2 Standard definition of "momentum"

In a game, we define "momentum" as the weighted sum of the expectation of the number of rounds won, the serve advantage factor, and the number of sets won in a window centered on the current point in time. A more rigorous description. The "momentum" of a player$i$ in round $t$ is$P_{i,t}$ . Take a window of size$l_{window}$ centered on the current round, in this case we specify $l_{window} = 5$ . Define the winning round of the event player$i$ within the window as$W_p$ , the expectation of the number of winning rounds as$E_{window}(W_p)$ , and the serve advantage factor as$\delta_{server}$ . The expected number of winning rounds is$D_{set}\lambda$ , and the serve advantage factor is . is the difference between the current player and his opponent's handicap, and is the difference factor. The standard definition of "momentum" is given by.

$$P_{i,t} = E_{window}(W_p) + \frac{\delta_{server}}{l_{window}} + \lambda D_{set}(4.2.1)$$
$$i \in \{1, 2\}, t \in N$$

Equation 4.2.1 can be applied and calculated to get a line graph of the "momentum" of the two sides, which helps us to easily see which player has the advantage in the flow of the game, and by how much.

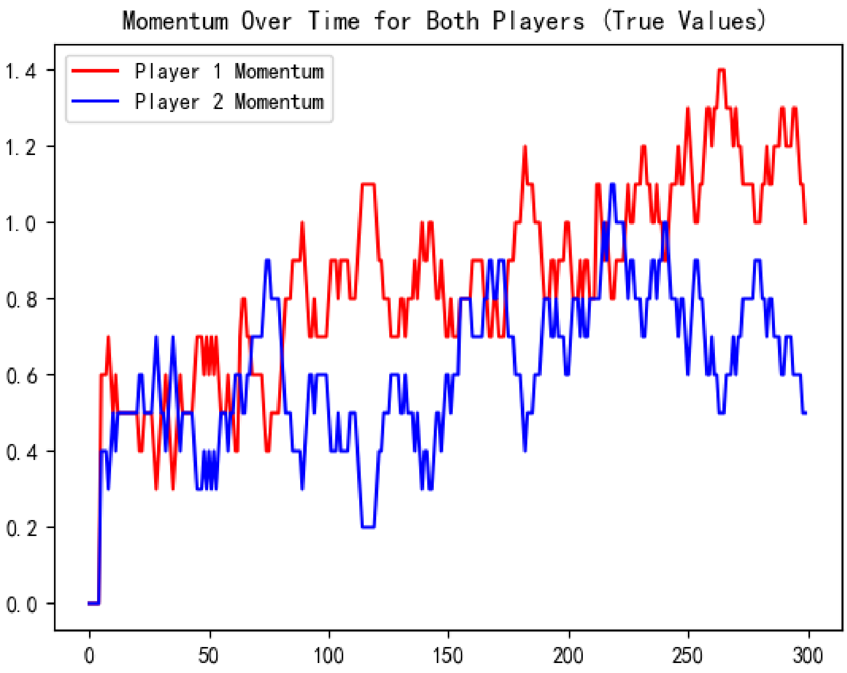

**figure3**

In the remainder of this chapter, we will improve this formula to make it more realistic and usable.

## 4.2 Approximation of "momentum"

In the previous section we defined the concept of "momentum" and its quantification. However, the calculation of "momentum" is done with the help of the results of the current time step t followed by t, t+1.... ...t+5 games after the current time step t. In practice this is not possible. Therefore, we will use an ARIMA model to predict the results of these matches, and then calculate the Momentum based on the predicted results. These are the topics of the next two subsections.

### 4.2.1 ARIMA model

The ARIMA model, or Autoregressive Integrated Moving Average with Difference, is a common forecasting model used in data analysis. It uses time series to predict future trends. AR (autoregressive term), I (difference order) and MA (moving average term) are its three important parameters, which are formulated as follows:

- Autoregressive (AR): an AR (autoregressive) model describes the relationship between current and historical values, utilizing historical time data on a variable to predict itself. the order of the AR model is usually recorded as the p-value.

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \ldots + \phi_p y_{t-p} + \varepsilon_t \quad (1)$$

- Integrated(I): when the time series becomes smooth, it needs to be differenced, and the order of the difference is usually recorded as a d value. Generally, first order differencing is sufficient.

$$\Delta^d y_t = \varepsilon_t \quad (2)$$

- Moving average (MA): the moving average model is concerned with the accumulation of the error term in an autoregressive model. the order of the MA model is usually recorded as the q value.

$$y_t = \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \ldots + \theta_q \varepsilon_{t-q} + \varepsilon_t \quad (3)$$

It has the following conditions of use.

- Data series are smooth, which means that the mean and variance should not change over time. The series can be made smooth by logarithmic transformation or differencing.
- The input data must be a univariate series because ARIMA utilizes past values to predict future values.

Next, we will test the series and determine the parameters of the model, and then use the ARIMA model to predict the total number of points (p1_points_won and p2_points_won) for each player, and provide the results to Equation 4.2.1 to calculate the predicted value of the "momentum"[3].

### 4.2.1.1 ADF test

The ADF test is to determine whether a unit root exists in the series: if the series is smooth, there is no unit root; otherwise, there is a unit root. So, the H0 hypothesis of the ADF test is that there is a unit root, and if the significance test statistic obtained is less than three confidence levels (10%, 5%, 1%), it corresponds to having (90%, 95%, 99%) certainty to reject the original hypothesis.[3]

The total number of points (p1_points_won and p2_points_won) of each player in this problem is basically consistent with the above condition at difference order d=1, and we can verify it by **ADF test**.

**Table:arguments of ADF**

| | coef | std_err | t_value | p_value |
|---|---|---|---|---|
| x1 | -1.17 | 0.083 | -14.101 | 0.0 |
| x2 | 0.1439 | 0.058 | 2.488 | 0.013 |
| const | -0.6088 | 0.097 | -6.288 | 0.0 |
| x3 | 0.0004 | 0.001 | 0.28 | 0.78 |
| x4 | -2.216e-06 | 4.39e-06 | -0.505 | 0.614 |

**Table:issue results of ADF**

| p_value | Critical Value (1%) | Critical Value (5%) | Critical Value (10%) |
|---|---|---|---|
| 2.689637e+02 | -3.452561e+00 | -2.871321e+00 | -2.571982e+00 |

**(1) Critical value test**

Critical values of 1%, 5%, and 10% reject the statistical value of the original hypothesis at different levels, comparing the hypothesis test value t with the critical value, t is less than 1%, 5%, and 10% at the same time means that the hypothesis is very well rejected. In this data, the ADF hypothesis test value t of the original series is 0.763835, which is greater than the statistical value of three LEVELS, so it is non-stationary. The ADF result of the first order difference series is -14.101. which is less than three levels of statistical value of 1%, 5% and 10% at the same time, indicating that the first order difference series of the original data is smooth.

**(2) Significance test p<0.05**

Turning to the significance p-value of the first order difference series. The p-value is $2.69 \times 10^{-26}$ , which is quite close to 0,indicating smoothness.[4]

After second-order differencing, compared with first-order differencing, the degree of significance is only expanded and the accuracy is not improved, so for this sequence, it is more appropriate to use first-order differencing. In general, the sequence can be made smooth by using first and second order differencing.[4]

**So the difference order in the parameters of the ARIMA model I = 1**

### 4.2.1.2 Determination of p- and q-values

Next, we will plot ACF and PACF on the data to better determine the other two important parameters in the ARIMA model.

**Drag tail and cut off tail**

If the sample autocorrelation coefficients and the sample partial autocorrelation coefficients are significantly larger than 2 times the standard deviation at the initial order (dashed line below), and then almost 95% of the coefficients fall within 2 times the standard deviation, and the non-zero coefficients decay into small value fluctuations very abruptly, this is usually considered to be a kth-order truncation of the tail.

If more than 5% of the samples have correlation coefficients greater than two times the standard deviation, or if the non-zero coefficients decay into small value fluctuations more slowly or continuously, they are usually considered to be trailing.

**Autocorrelation coefficient (ACF)**

The autocorrelation coefficient measures the degree of correlation between the same event over two different time periods, or figuratively speaking, the effect of one's past behavior on one's present. The maximum lag of the autocorrelation coefficient (ACF) plot can be used to approximate the q-value.

**Partial Autocorrelation Coefficient (PACF)**

When calculating the influence or correlation of one element on another, the influence of the other elements is regarded as a constant, i.e., the influence of the other elements is not taken into account for the time being, and the closeness of the interrelationship between those two elements is examined in isolation, it is known as partial autocorrelation. The p-value can be approximated here by the maximum lag point of the partial autocorrelation coefficient (PACF) plot.

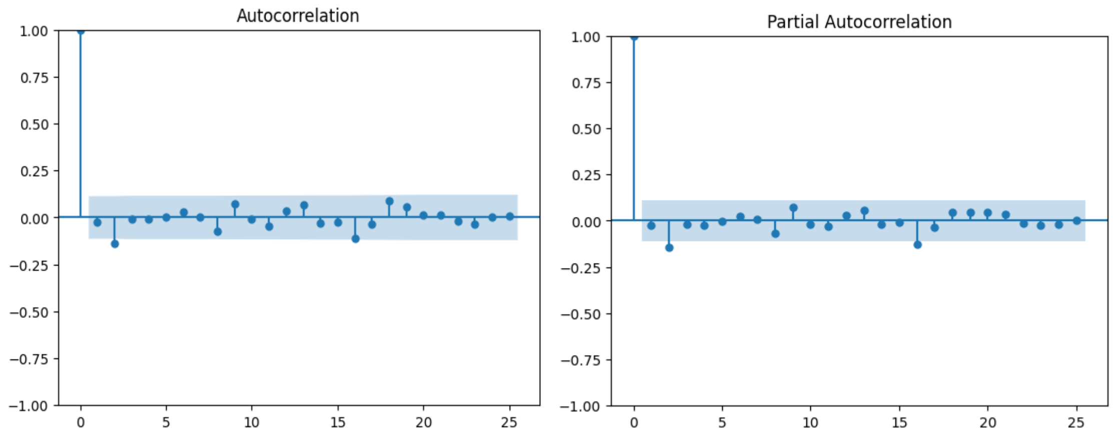

**figure4**

Based on the images, we can determine two other model parameters: AR=3, MA=3. Thus, we have completed the test of the series and determined the important parameters of the model.

### 4.2.1.3 Predicting the total number of points for players

After training, we can get the prediction result of the player's total points.

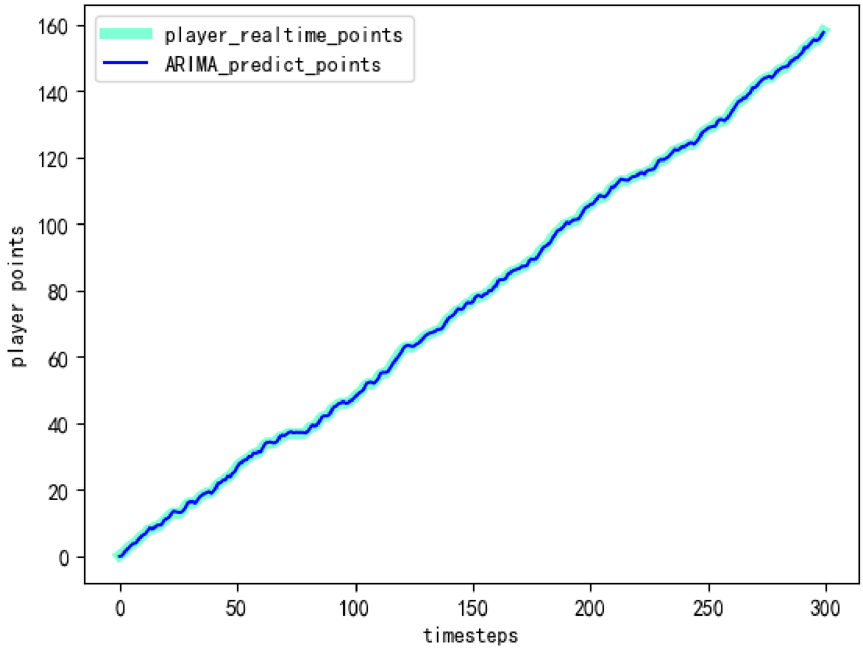

**figure5**

We briefly evaluate the prediction results: the ARIMA model is evaluated using three separate metrics:

- MSE (Mean Squared Error) measures the accuracy of data using the square of the difference between the actual value and the predicted value.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- AIC (Akaike Information Criterion) A balance between the number of parameters in the predictive model and the likelihood of the model, with lower AIC values implying a better fit of the model

$$\mathrm{AIC} = 2k - 2\ln(\hat{L})$$

- BIC (Bayesian Information Criterion) is based on the parameter tuning of the AIC, which provides a more rigorous evaluation of additional parameters, and a lower BIC also means that the model is more effective

$$\mathrm{BIC} = k\ln(n) - 2\ln(\hat{L})$$

| Symble | Value |
|---|---|
| MSE (Mean Squared Error) | 0.76 |
| AIC (Akaike Information Criterion) | 202.014319 |
| BIC (Bayesian Information Criterion) | 216.829449 |

The MSE is close to 0, and the AIC and BIC metrics are minimized at p=3, d=3, so the ARIMA model predicts the total number of player points well. The prediction results are credible.

## 4.2.2 Approximations and errors in "momentum" (optimization)

### 4.2.2.1 Approximation of "momentum"

Now we define the approximation of the "momentum" of the player$i$ in the round$t$ as$\hat{P}_{i,t}$ . $E_{prev}(W_p)$This is the expectation of the number of rounds won in$[t-5, t-4...t-1]$ , and $E_{pred}(W_p)$ is the expectation of the number of rounds won in$[t, t+1...t+4]$ . The "momentum" formula changes to.

$$\hat{P}_{i,t} = E_{prev}(W_p) + E_{pred}(W_p) + \frac{\delta_{server}}{l_{window}} + \lambda D_{set}$$

$$i \in \{1,2\}, t \in N$$

Where$E_{pred}(W_p)$ is defined as the expectation of the ARIMA model to predict the number of winning rounds in the next 5 rounds. By substituting the formula we get a line graph of the approximation of Momentum.

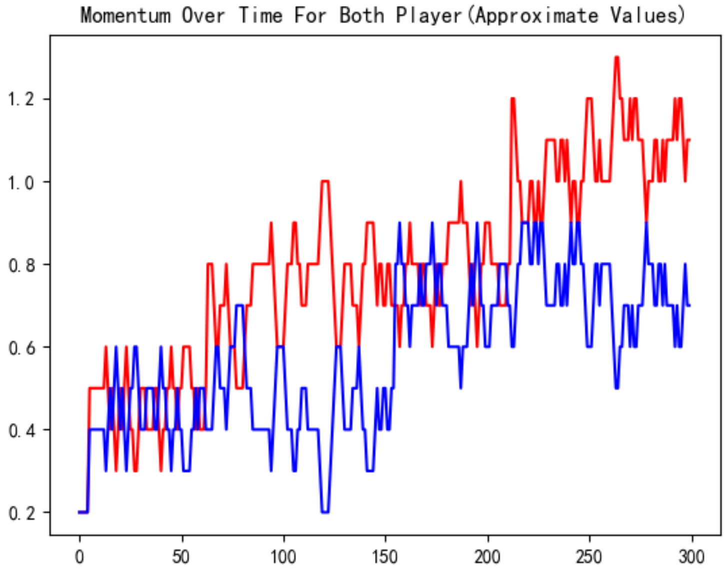

**figure6**

From the graph, the approximation curve of Momentum roughly agrees with the exact value curve of Momentum. We evaluate this result in the next section.

### 4.2.2.2 Errors in approximations

To assess how good our predictive model is, we calculated MAPE (Mean Absolute Percentage Error)

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\%$$

After calculating MAPE = 1.254%, we can get good results from our model. Therefore, this result can be analyzed and used in practice instead of the exact value of "momentum".

### 4.3 Summary

In this chapter, we define a formula for calculating Momentum and an approximation of Momentum, and derive a feasible algorithm to quantify a player's Momentum in a match, which consists of Current Winning Percentage, Serve Advantage Factor, and Plate Advantage, and allows us to compare the status of the competitors to determine the direction of the flow of the match.

## 5. Problem 2

A tennis coach argued that the changes in the match were random, in order to disprove this claim. We next need to verify that the course of the match is random using our model as well as the follow-through test. Due to the simplicity and accuracy of the K-S test, we first generate random match scores using random numbers, and then we use the K-S test to determine whether the actual course of the match and the random course of the match are identically distributed.

### 5.1 Stochastic process modeling: hypothesis analysis

We begin by randomly generating the score of a match and factoring the serve into the win rate. Quoting here from the analysis in Section 4.4.1, the probability of the serve side scoring is close to 0.67, so we perform random generation of match scores with a win rate of 0.67 for the serve side.The momentum of a random match is calculated after randomly generating match scores. The following figure shows the image of randomly generated match score and momentum.

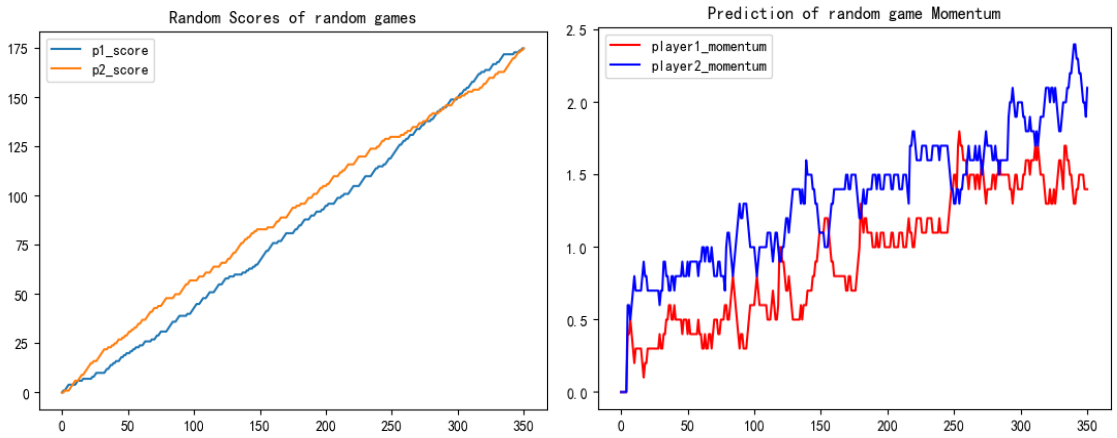

**figure7**

### 5.2 Kolmogorov-Smirnov test

Kolmogorov-Smirnov test is often used to test whether 2 data distributions are consistent, Kolmogorov-Smirnov test is a nonparametric test, widely used in data analysis and machine learning.

The problems in the K-S test given two hypothesis tests are as follows:

- $H_0$: the population from which the sample comes follows a certain distribution
- $H_1$: the sample comes from a population that does not obey a certain distribution

Using the Kolmogorov-Smirnov test it is possible to test which of the two hypotheses, H0 and H1, is correct.

We first need to find the empirical distribution function of the momentum of the actual race and the random race$F_{1,n}(X)$ 和 $F_{2,m}(X)$ , where n and m are the data size, $\min = \min\{x_1, x_2, \ldots\}$ ,$\max = \max\{x_1, x_2, \ldots\}$ The empirical distribution function is calculated as follows:

$$F(X) = \begin{cases} 0 & X & < \min \\ \frac{num\ of\ below\ X}{num\ of\ observations} \quad \min < & X & < \max \\ 1 & X & > \max \end{cases}$$

Next, we compare the empirical distribution functions of the two data and calculate the test statistic Dn.

$$D_N = \max|F_{1,n}(X) - F_{2,m}(X)|$$

The line graphs of the two empirical functions are given below, and it can be observed that the largest differences occur between X = 1.1 and X = 1.2.

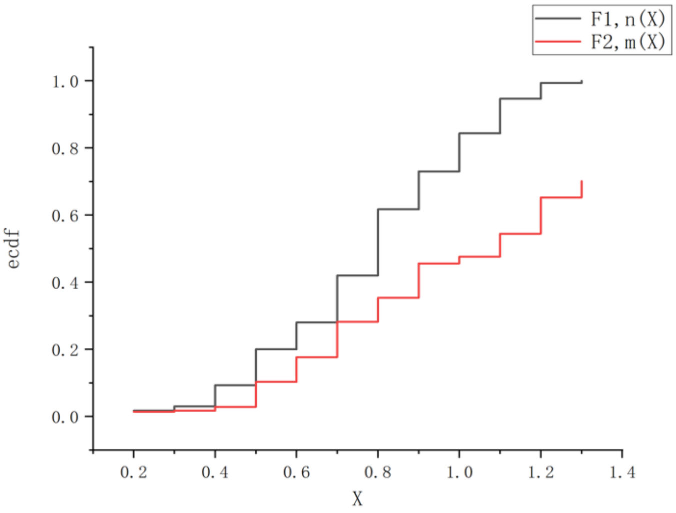

**figure8**

In the Kolmogorov-Smirnov test, we usually utilize the p_value to assess the difference between two models, and at the end we need to calculate the p_value.

$$P\_value = 2e^{-2\left(D_n^2\right)^2 \frac{n*m}{n+m}}$$

**Table: The value of symbol**

| Symbol | Value ( )$10^{-4}$ |
|---|---|
| Dn | 15.1 |
| P_value | 1.01 |

We can get P_value<0.05, so we do not reject the H1 hypothesis: the two data are not identically distributed, i.e., the fluctuation of the game is not random, but has some regularity.

## 5.3 Analysis of results

By randomly simulating the matches and performing the KS test, after analyzing the results, it can be concluded that the fluctuations of the matches do not occur randomly but with a certain regularity, and this important result shows that the results of the tennis matches can be predicted, which is of some significance for the tennis matches.

# 6. Problem 3

In this section, in order to identify the changes that affect the flow of the game, we design a new mathematical model that quantitatively describes the "turning point", which occurs at the moment when the flow of the game changes, by using the difference in points between the two sides of the game and the derivative of the difference in points between the two sides.

## 6.1 Definition of Turning Point

Analyzing the game score data, we find that the turning point occurs when the flow of the game changes. From the game score diagram, we can see that the turning point occurs when the score of the two teams is close to and exceeds the score of the game. In order to quantitatively describe the "turning point", we introduce two new concepts for modeling:$\delta_{dif}$ and$grad(\delta_{dif})$ , which are the derivative of the score difference.

We give the following definition.

$$\delta_{dif} = p1_{pw} - p2_{pw}(6.1.1)$$

The original obtained by direct differencing will have obvious burr phenomenon, we use Savitzky-Golay filter to smooth the original differencing data, so as to remove the burr phenomenon in the data, where the window size is$l_{window}$ and the polynomial fitting order is poly_order. so as to get the smoothed differencing$\delta_{dif}$ . $grad(\delta_{dif})$The first order derivative of$\delta_{dif}$ is used to describe the trend of the variance.

From the line graph of the scores of both sides of the game, it can be seen that the appearance of turning points is often accompanied by two phenomena:

- Rapidly approaching scoring by both sides of the game.
- There was a back-and-forth game score.

Further, we plotted histograms of the variance and the derivative of the variance, comparing each of the statistical measures of the derivative of the variance, including the mean, the median, the square root of the standard deviation of the standard deviation, and the maximum value (the order corresponds to the following table)

**Table: The Statistical indicator of** $grad(\delta_{dif})$

| $grad(\delta_{difmean})$ | $grad(\delta_{difmedian})$ | $grad(\delta_{difsd})$<br>stand deviation | $grad(\delta_{difsr})$<br>square root | $grad(\delta_{difmax})$ |
|---|---|---|---|---|
| 0.0600789798 | 0.0787712451 | 0.6868794577 | 0.8287819120 | 1.2050628662 |

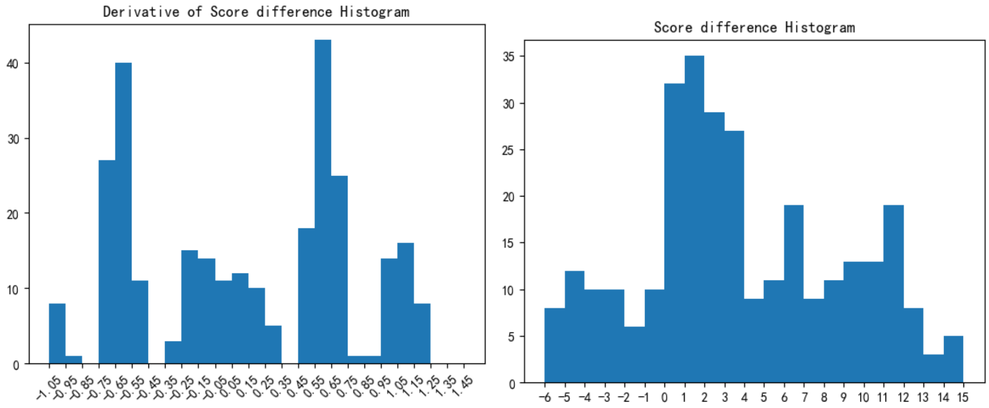

**figure9**

For the quantification of the above two indicators, we find that the

- $grad(\delta_{dif}) >= 0.75$
- $grad(\delta_{dif}) >= 0.55, |\delta_{dif}| < 1$

Satisfying any of the quantitative indicators in 1, 2, which are consistent with the occurrence of turning points during the actual match, and plotting the scores of the two sides of the match containing turning points, we find that the mathematical modeling of turning points can indeed satisfy the description of the change of the flow of the match very well.

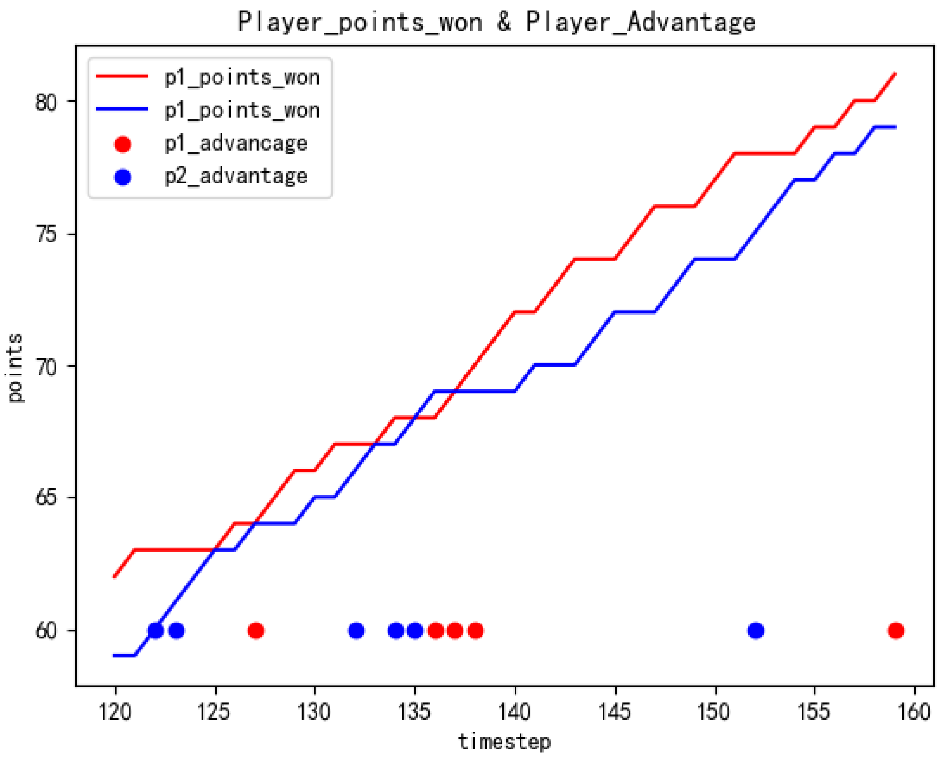

**figure10**

## 6.2 Analysis of relevant factors

With the help of the model we want to analyze the factors that are most relevant to the turns in the race stream. So we use the XGBoost (eXtreme Gradient Boosting) model to predict the turns and derive the features that are most helpful in categorizing the turns, i.e. the most relevant factors.

### 6.2.1 Introduction to XGBoost

XGBoost can be seen as an optimization of GradientBoost. Its principle is still based on GradientBoost, and its innovations are as follows:

- Regularization: XGboost supports L1 and L2 regularization, which helps to prevent overfitting and the "Longe" phenomenon to some extent[1]
- Automatic handling of missing values and pruning: able to automatically handle missing values to simplify model training; using regularization terms to effectively control the complexity of the tree[1]
- Feature Importance and Flexible Objective Functions: Provide feature importance assessment to help understand the decision-making process of the model; and allow user-defined objective functions, with strong generalization capabilities[2]

### 6.2.2 Data processing

We used three randomly selected games (2023-wimbledon-1301, 2023-wimbledon-1401, 2023-wimbledon-1408) as the dataset, respectively, and we briefly processed the data before proceeding with the XGBoost model. This includes data cleaning, feature selection and normalization.

- Smoothing and Derivation

By subtracting p1_points_won and p2_points_won bit by bit from the supplied data, we obtain the set of differentials $\delta_{dif}$ . Next we try to get the derivatives of the differences $grad(\delta_{dif})$ array.

In trying to use Central Finite Difference Approximations, we ran into difficulties. The scores of the two players in a game are discrete, unincreasing arrays. Jumps or stagnation in scores between two neighboring time steps can have a large effect on the derivative at that point. To reduce noise in the data and extract trends and turning points in the scores, we used a Savitzky-Golay filter to smooth the difference $\delta_{dif}$ array. This filter performs a local polynomial fit within a certain window size and replaces the original values with the fitted values. This removes high frequency noise from the data and makes the score curve smoother. After the smoothing of the score array is complete, we derive a derivative for it to obtain the rate of change of the score. This helps us to identify increases and decreases in scores, and thus the location and trend of turning points.

An array of turning points is obtained by bringing back each item in the array of divergences and the array of derivatives of divergences to Eq. (6.1.1). We use this array as a label for what the model needs to predict.

- standardization

In addition to scores, we note that features such as player bat speed (speed_mph) and mileage (e.g., p1_distance_run) have large variance. In order to eliminate the scale differences between different features, we normalized these features. A Z-score normalization method was used, where the value of each feature was subtracted from its mean and then divided by the standard deviation. This allows different features to have similar scales, which facilitates the model to better understand the relationship between them.

- feature mining

In addition to the original features, we also perform new feature mining to enrich the inputs to the model. One of the new features we extracted from the score array is the consecutive score (combo), which indicates the number of consecutive scores. Noting that the importance of each round is different, we introduce the 01 array to indicate whether this is a set point or not. Finally, we introduce "momentum" as one of the new features mined.

Through the above data processing steps, we obtain the dataset after smoothing, derivation, normalization and feature mining, which lays the foundation for subsequent model training and analysis.

### 6.2.3 Training results

The optimal parameters are searched by GridSearch. The optimal parameter combination is n_estimators=90, max_depth=6, please refer to appendix a for the search range and optimal parameters.

We obtained the following results under 50% discount cross validation.

**Table:Optimal parameters**

| Accuracy | Precision | Recall | F1 |
|---|---|---|---|
| 76.33% | 72.88% | 76.33% | 72.56% |

### 6.2.4 Most relevant factors

We visualized the contribution of each factor to the model, with the main factors being.

1. Current score difference (score_dif_p1)
2. Both sides "momentum" (p1_momentum,p2_momentum)
3. Number of consecutive wins for both sides (p1_comble,p2_comble')
4. Main serve (server)

These features contribute more to the classification results and can be considered as correlates that are more related to the turning points.

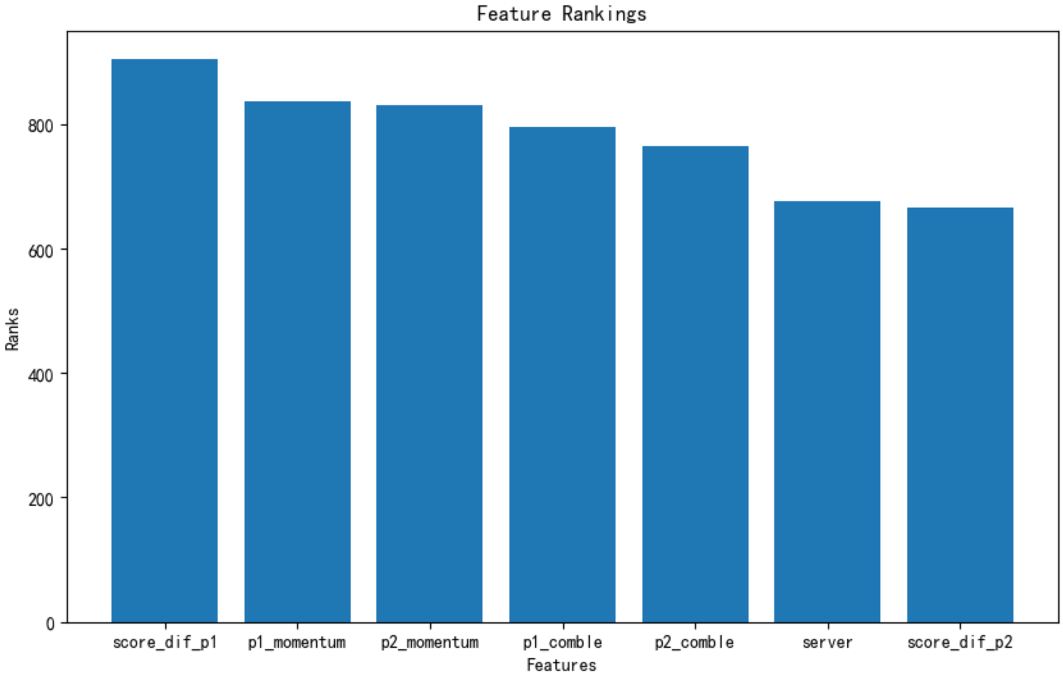

**figure11**

## 6.3 Advice to players

Based on the results of the first and third questions, we give the following recommendations.

1. Discuss with your coach to get as much data as possible about the enemy player's previous matches, compare your strength with the enemy player's strength, if there is a big gap between the enemy's strength and your own, it is better to play the match immediately when your side is in an advantageous situation, and vice versa.
2. Under the premise that you and your opponent are of equal strength, or the strength of the

enemy player is not yet clear, if the momentum value of your side is greater than that of the enemy at the beginning of the match, it is best to choose to continue the match. If there is a situation where the momentum value is converted in the course of the match, i.e., there is a turning point of the match, you can choose to apply for a halftime break to recover your own state and then continue the match.

3. If more data of the opponent's previous matches are obtained in advance, the momentum change of the opponent can be analyzed in advance in the same game as the current match (a match of equal strength), so as to be familiar with the general trend of the opponent's serve, receive, and momentum change, and then in the course of the actual match, to do the targeting, i.e., to adjust the real-time status of their own matches better.

# 7. Problem 4

In this chapter we will evaluate the classification model obtained in the previous chapter to test its generalization. We will also analyze the model in other ball categories (e.g., table tennis, basketball, etc.) to show that our method can be generalized to other game domains.

## 7.1 Model evaluation

In the previous chapter, we trained an XGBoost classification model for predicting turning points, and we used only three games as a dataset, which is a smaller volume of data and allows us to better evaluate the generalization performance of the turning point prediction system. We do the same for the other games, use the prediction system to predict them, and evaluate the results.

We arrived at the following results.

avg_metricx

| | |
|---|---|
| Accuracy | 73.17% |
| Precision | 71.16% |
| Recall | 73.17% |
| F1 | 70.74% |

## 7.2 Generalization to other competitions

Based on the prediction results of the momentum model and the turning point model on other matches, we found that the results of the model's effect on different ball games showed some differences, as shown in the following:

- Momentum modeling strategies show very good accuracy and prediction for other matches in the same tennis category
- For other types of games, it may be necessary to modify and add some new indicators to make a more accurate prediction of the game.

Considering the fact that the mechanism of tennis matches is based on two-player pairs and score accumulation, which is different from that of most ball games, such as basketball, soccer and other group-competitive ball games, we believe that the applicability of the momentum model should be restricted to ball games with similar rules, including other competitive tennis matches, ping-pong, billiards, and so on. After relevant analysis, we find that for competitive tennis matches, the momentum model shows quite accurate characteristics. For other types of matches, we might add some metrics to train our model and thus make small corrections to the momentum model. For example, in table tennis, we will add the important metric of the front and back of the paddle used by the player to catch the ball. All in all, I also very much expect that we will improve our model by discovering the shortcomings of our model in our tests on other matches.

## 8. Conclusion and Memorandum

Conclusion.

1. The direction of the game scores predicted by the time series model is indeed valid and gives a reasonable picture of how the situation changes during the course of the game.

2. Momentum modeling can indeed reflect the direction of the game in real time, so that appropriate adjustments can be made according to the momentum changes, which can increase the chances of winning the game and effectively avoid the emergence of cold results.

3. Based on the analysis of the turning point model and the XGBOOST model training, after the (reasonable assessment) method, the player's stability after scoring and the accuracy when hitting the ball are targeted, so as to effectively improve and stabilize the player's level and power

4. The quantitative description of the momentum model and the turning point model has the generalization of the ball game, and we have measured it with different game data

   For different players and different times of various ball games, including badminton, table tennis and other two-player round robin matches, you can use these two data models with appropriate adjustment of the parameters to provide reasonable training for players.

With the prevalence of the concept of healthy life, tennis as a challenging sport is popular among the public. In recent years, more and more tennis stars join in, making the situation of a match often complex and rich in changes, which also tests the improvisation ability and psychological quality of tennis players. In order to make more reasonable arrangements in real-time matches and make more scientific pre-match training. Based on the data from the Wimbledon Men's Singles Final, our team has built a momentum model and a turning point model to help you coach your players better.

Advice.

When to rest: According to our momentum model, if your player's momentum is at a disadvantage, it means that your player's real-time status is not very good, and playing the game immediately may lead to a loss of score, so it's a good choice to apply for a 20-second reasonable rest time for your player to adjust his status. On the contrary, if your player's momentum is higher than the enemy's, then continue the game and take advantage of the victory.

How to cope with the impact of the flow of play.

Prior to analyze the relevant data of the opponent's previous matches, mark the key time points where the opponent's momentum is at an advantage and disadvantage, and in the process of pre-match simulation training, select and the opponent's playing rhythm similar to the accompaniment and thus carry out targeted training to enhance the stability of the strength of their own players to play, and to reduce the uncertainty and psychological state of the errors brought about by the factors.

## References

[1] Wu Yiming. Research on Large-scale Gene Regulatory Network Inference Method Based on Model Fusion [D]. Dalian Maritime University, 2023. DOI: 10.26989/d.cnki.gdlhu.2023.000380.

[2] Nie Shuyuan. Comparison of GDP Forecasting Effects Based on Four Types of Time Series Models [J]. Statistics and Decision, 2024, 40(02): 63-67. DOI: 10.13546/j.cnki.tjyjc.2024.02.011.

[3] Zhou Jiashu. Research on Integration Estimation of Integration Based on Quasi-Monte Carlo Method and XGBoost Algorithm [D]. Shandong University, 2023. DOI: 10.27272/d.cnki.gshdu.2023.007033.

[4] Zhang Anni, Zhou Xiao, Lou Lidu, et al. Research on Multivariate Time Series Prediction Based on VAR Model [J]. Modern Computer, 2023, 29(21): 36-40.

## Appendices

### a. GridSearch parameters

XGBoost parameter search range

| Symble | Value |
|---|---|
| max_depth | [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] |
| min_child_weight | [1, 2, 3, 4, 5, 6] |
| gamma | [0, 0.03, 0.05, 0.08, 0.1, 0.3, 0.5, 0.7, 0.9, 1] |
| colsample_bytree | [0.3, 0.5, 0.6, 0.7, 0.8, 0.9, 1] |
| n_estimators | [50,60,70,80,90,100,120,150,200] |
| eta | [0.05, 0.1, 0,2, 0.3] |

Optimal parameter combinations (including default parameters)

| Symble | Value |
|---|---|
| max_depth | 6 |
| min_child_weight | 1 |
| gamma | 0.1 |
| colsample_bytree | 1 |
| n_estimators | 90 |
| subsample | 1 |
| reg_alpha | 0.01 |
| reg_lambda | 3 |
| eta | 0.3 |