Problem Chosen	2022 MCM/ICM Summary Sheet	Team Control Number
C		2226886

Midas Touch: Maximize Your Retrun with PBRT Model

Summary

The rise and drop of volatile assets' prices provide traders with the opportunity to maximize their total assets. However, due to the instability of volatile assets' prices, it is extremely difficult to determine how much to trade the volatile assets based only on the price data up to that day. As a result, it is quite difficult to maximize the total assets with such a small amount of information.

To overcome the above-mentioned challenge, we propose a novel model called **PBRT** which consists of a novel **hybrid time series prediction model** to predict the future values of gold and bitcoins, a bull market assessment model to assess whether it is a bull market or not, a **risk assessment model** to evaluate the trading risk and a **trading model** to decide what and how much to trade. Firstly, we hybrid the linear regression, ARIMA, and extreme gradient boosting with gradient boosting method to develop a prediction model. This model shows much more accuracy on validation set with a maximum mean-square error (hereinafter referred to as MSE) of 3.156 than simple ARIMA (MSE 25.09), linear regression (MSE 65.78) and extreme gradient boosting (MSE 890.87). Secondly, based on the two assets' average bias ratio (hereinafter referred to as BIAS), rise of certain previous days and voting method, we can calculate the bull market assessment indicator to determine whether it is now a bull market or not. Thirdly, we will calculate the risk of trading according to the bull market assessment and the BIAS. In the end, we decide how much to trade a certain asset based on the risk assessment, BIAS and bull market indicator. According to the PBRT model, we can increase our total assets to \$76211.40.

To prove that the PBRT model is optimal, we adopt dynamic programming and greedy algorithm to calculate how many assets we can get **under the circumstance that we know the price of all times**. By dynamic programming, we can get at most \$358,499. By greedy algorithm, we can only get \$50,782 at most. Considering that the dynamic programming and greedy algorithms are adopted when we assume we know the price of all times, it is apparent our PBRT model is the best.

To investigate how sensitive our strategy is to the change of transaction cost, we conducted 100,000 simulated experiments to check its sensitivity to the transaction cost and use linear regression to find out the pattern. The results show that if 1% transaction cost of gold is increased, our final asset will decrease by 2.1%. For bitcoin, if 1% transaction cost is increased, our final asset will decrease by 2.2%. To conclude, **the PBRT model is insensitive to transaction cost change**.

Based on the PBRT model, several suggestions are proposed in the form of memorandum to the trader. We propose suggestions on how to evaluate the risk of trading, when you should sell or buy a certain asset, how much you should sell or buy certain assets and.

Our PBRT model shows a strong accuracy and robustness by verifying our model with different hyperparameters. So, it may be possible to be implemented to other kinds of data.

Keywords: PBRT; ARIMA; XGBoost; Risk prediction; Bias Ratio; Data Analysis; Price Prediction

Contents

Memorandum

To: Mr.Trader

From: MCM Team # 2226886

Subject: Trading Strategy

Date: February 22, 2022

According to your requirements, we analyze the gold and bitcoins daily prices. We develop one hybrid model dubbed as PBRT model which consists of a novel hybrid time series prediction model to predict the future values of gold and bitcoins, a **bull market assessment model** to assess whether it is a bull market or not, a risk model to evaluate the trading risk and a trading model to decide what and how much to trade. And we get some meaningful results which will be conducive to increasing your assets.

Firstly, we construct a novel hybrid time series prediction model which excels at accurate prediction in a short period of time. The model can not only grasp the trend component in the time series, but also have the ability to learn the complex patterns in the detrended data series. As a result, this model can help you predict the rise and the price of assets' prices. We believe this model would be significant to your work when you are confronted with instability of the volatile assets' prices.

Secondly, we develop a **bull market assessment model** to help you determine whether it is now a bull market or not. The model is based on the two assets' average bias ratio (hereinafter referred to as BIAS), rise of certain previous days and voting method.

Thirdly, we will calculate the risk of trading according to the bull market assessment and the BIAS.

In the end, we decide how much to trade a certain asset based on the risk assessment, BIAS and bull market indicator.

According to the PBRT model, we can increase our total assets to $76211.40 dollars. We adopt **dynamic programming** and **greedy algorithm** to calculate how much assets we can get under the circumstance that we know the price of all times. By dynamic programming, we can get at most $358,499. By greedy algorithm, we can only get $50,782 at most. Considering that the dynamic programming and greedy algorithms are adopted when we assume we know price of all times, it is apparent our PBRT model is the best.

Then, we conducted 100000 simulated experiments to check its sensitivity to the transaction cost and we find out that if 1% transaction cost of gold is increased, our final asset will decrease by 4%. For bitcoin, if 1% transaction cost is increased, our final asset will decrease by 2.5%. Also, indicators are easy to measure and the PBRT is quite simple. Consequently, this model has great robustness.

According to our analysis and PBRT model's results, we formulate reasonable trading strategies for you:

- When the BIAS rises, even when the price rises, the risk of losing is increasing.
- When the price decreases in a stable manner for a relatively long period of time, you can buy this asset because it might rise dramatically later.
- When the average BIAS increases dramatically and the price rises dramatically as well for a relatively long period, it is time for you to sell the asset.
- When you want to sell an asset, you should take the risk, bull market assessment, predicted rise and BIAS into consideration.
- If you want to trade an unstable asset to earn more money, you should always keep in mind: you should buy some other assets which are stable and safe.

Thanks for taking the time out of your busy schedule to read my letter. Hope our advice can help.

1 Introduction

1.1 Background

Bitcoin is rapidly gaining recognition as an alternative and disruptive asset class, for its decentralization and low transaction rate, etc. While gold is stable and high liquid, making it an effective means to deal with inflation.

In this problem, market traders buy and sell these two assets using some strategy to maximize their total return. However there is usually a commission for each transaction, which limites transaction frequency.

1.2 Clarification and Restatement

A trader wants a model that uses only the past stream of daily prices to date to determine each day if the trader should buy, hold, or sell their assets in their portfolio. We are given two spreadsheets of historical daily prices of bitcoin and gold to develop the model.

We will start investment with $1000 on 9/11/2016 and end on 9/10/2021. On each trading day, the trader will have a portfolio consisting of cash, gold, and bitcoin [C, G, B].The initial state is [1000, 0, 0]. The commission for each transaction (purchase or sale) costs $\alpha\%$ of the amount traded(Assume α_{gold} = 1% and $\alpha_{bitcoin}$ = 2%). There is no cost to hold an asset. Bitcoin can be traded every day, but gold is only traded on days the market is open(usually Mon. to Fri.). We should only use the two spreadsheets to solve the following problems:

- Develop a model that gives the best daily trading strategy based only on price data up to that day. Calculate the initial $1000 investment worth on 9/10/2021 using our model and strategy.
- Present evidence that our model provides the best strategy.
- Determine how sensitive the strategy is to transaction costs, and analysis how do transaction costs affect the strategy and results.
- Communicate our strategy, model, and results to the trader in a memorandum.

2 Data Processing and Analysis

2.1 Data Processing

Step1. Datasets merging:

Because we want to investigate whether our prediction model has a good generalization of our datasets, we divide our datasets into training sets and validation sets and the validation sets are used to find out the generalization capabilities of the prediction model. Instead of randomly dividing the datasets, we separate the datasets by time series order to avoid losing the series dependency in the time series. In the end, we give the training sets 1000 samples and the validation set 827 samples. The gold datasets and bitcoin datasets are merged on date by means of outer joints. Consequently, Missing values emerge.

Step2. Missing Value Processing:
To better save the information in the datasets, we use the B spline interpolation to fill the missing values in the dataset and 571 missing values are found and filled with suitable values generated by B spline interpolation algorithm.

Step3. Dividing the validation and training sets:
Because we want to investigate whether our prediction model has a good generalization of our datasets, we divide our datasets into training sets and validation sets and the validation sets are used to find out the generalization capabilities of the prediction model. Instead of randomly dividing the datasets, we separate the datasets by time series order to avoid losing the series dependency in the time series. In the end, we give the training sets 1000 samples and the validation sets 827 samples.

Step4. Adding Features to the Datasets and Normalization of the processed Data:
To have a better understanding of the datasets and train a better prediction model, we perform feature engineering on the datasets. We add the rise of gold and bitcoin, the bias ratio of gold and bitcoin, the trend of time series, seasonality of time series, series dependency of time series, judgement of bull or bear market to the datasets. And these features do not have the same dimensionality. As a result, in order to eliminate the influence of different dimensionality between indicators, data normalization is needed to address the incomparability between data indicators. These features' and normalization's details will be explained later.

2.2 Time Serial Data Analysis

2.2.1 Trend and seasonality

As it is clearly displayed in the Figure 2.1 and Figure 2.2, we can detect a persistent and long-term change in the mean of the time series which is called the trend. The trend there represents the largest time scale of importance. As we can also see in the figures, the trends are increasing in both figures which means that the markets of bitcoin and gold are expanding.

In an attempt to find out the seasonal differences in the time series data, we depict the seasonal plots(Figure 2.3 and Figure 2.5) which show segments of time series plotted against some common

Figure 2.1: The Increasing Trend Component Detected in Bitcoin Time Series Data

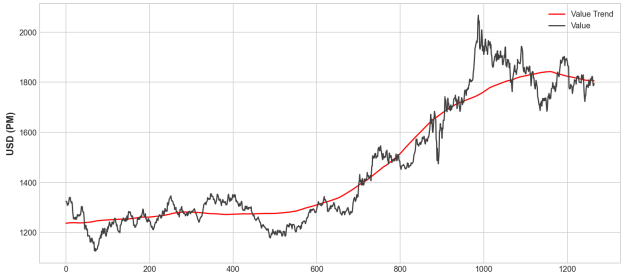

Figure 2.2: The Increasing Trend Component Detected in Gold Time Series Data

period and the seasonality you want to detect and Periodograms(Figure 2.4 and Figure 2.6) which utilizes the frequencies of different time series are adopted to discover the seasonal pattern.

As we can clearly see in the Periodograms and Seasonal Plots, seasonal patterns can be barely found out in these two time series datasets. As a result, we can safely come to the conclusion that seasonal patterns are not that salient.

2.2.2 Serial Dependence Properties

To investigate the Serial Dependence Properties in the two time serial datasets, we draw plots of autocorrelations and partial autocorrelations.

As we can see these two time series datasets have strong series dependencies, especially autocorrelations, we can make use of them to predict accurate outcomes. By investigating the four figures Figure 2.7 Figure 2.8 Figure 2.9 Figure 2.10 , we can make use of autocorrelations by adding lag features to our datasets, because hundreds of lag features are above the credential interval.

2.2.3 Feature Engineering

1. To utilize the increasing trend features, we add the square of the time dummy to the feature set:

$$Target = A \times Time^2 + B \times Time + C \tag{1}$$

   The linear regression algorithm we introduce later will learn the coefficient A, B and C in the data.

2. To make sure we do not ignore the seasonal and cycle patterns in the time series data, we add two pairs of fourier features which are pairs of sine and cosine curves(See Figure 2.11) to the

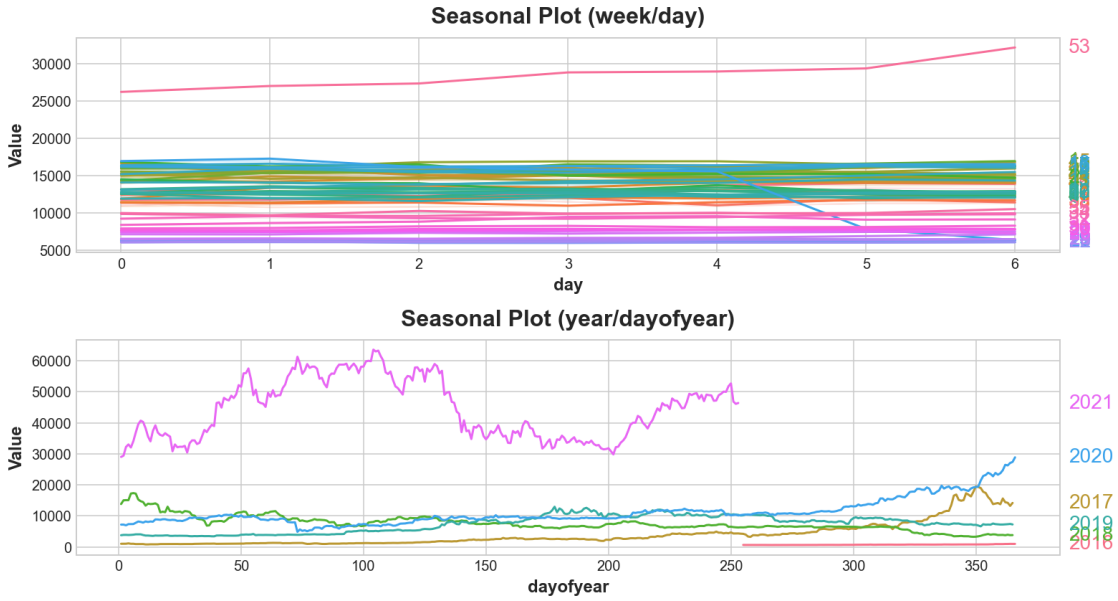

Figure 2.3: Bitcoin's Seasonal Plots of different seasonal indicators

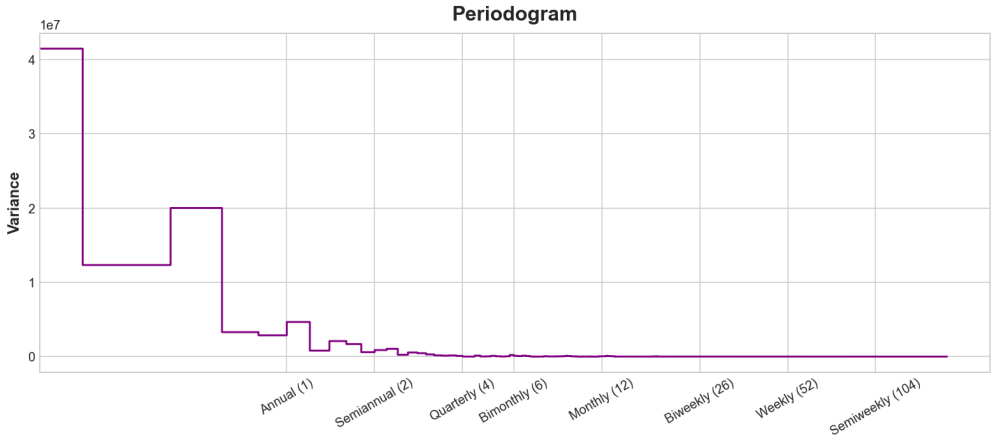

Figure 2.4: Periodogram for Bitcoin Series

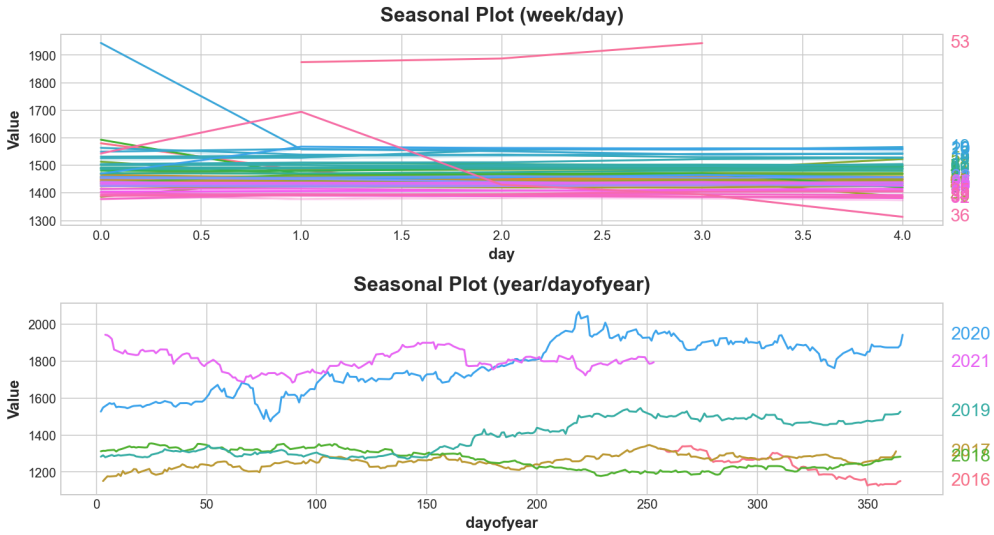

Figure 2.5: Gold's Seasonal Plots of different seasonal indicators

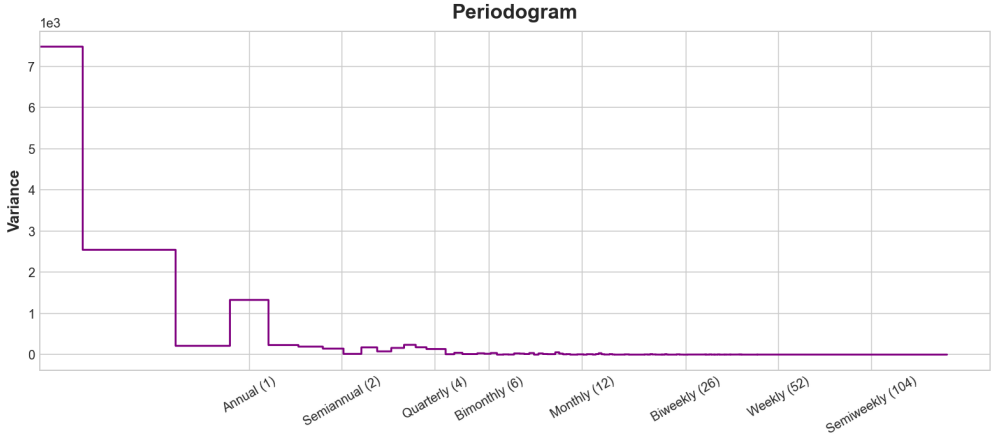

Figure 2.6: Periodogram for Gold Series

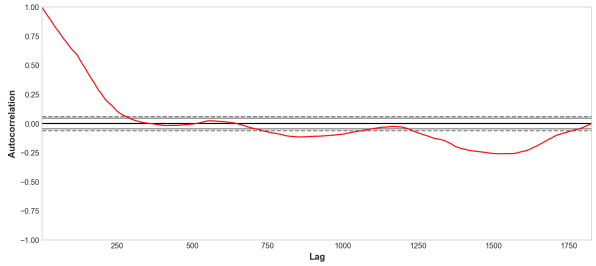

Figure 2.7: Autocorrelations of Bitcoin

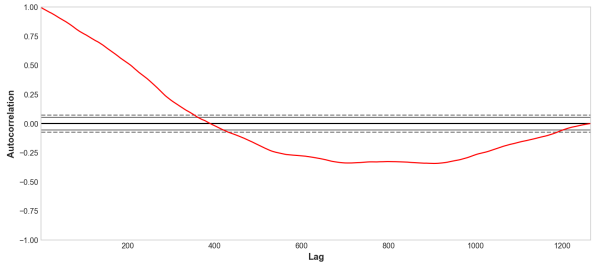

Figure 2.8: Autocorrelation of Gold

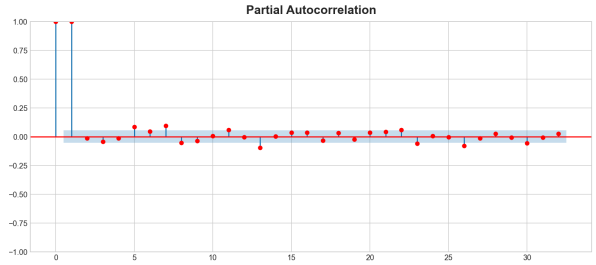

Figure 2.9: Partial Autocorrelation of Gold

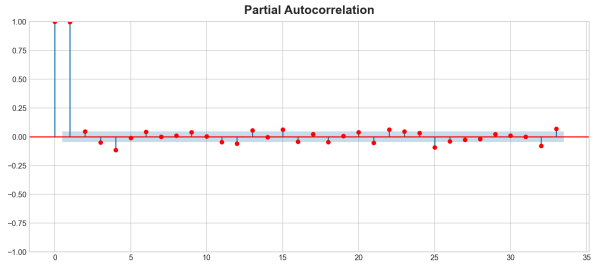

Figure 2.10: Partial Autocorrelation of Bitcoin

dataset:

$$A \times \sin t + B \times \cos t \tag{2}$$

The linear regression algorithm we introduce later will learn the coefficient A, B and C in the data.

3. Lag Features: As the Figure 2.8, Figure 2.7, Figure 2.9, Figure 2.10 indicated, lag_1 and lag_2 not only have strong autocorrelations but also have strong partial autocorrelations. Consequently, we add lag_1 and lag_2 to our models.

2.3 Finantial Analysis

2.3.1 Rise (without distinguishing whether it is a trading day)

To better evaluate the BIAS and average value of gold and bitcoin, we plotted the rise (Figure 2.12 and Figure 2.13) of gold and bitcoin based on the increase of different days. The average value of too many days will miss the dramatic rise and the average value of too few days will be too sensitive to change and can not give us appropriate reference on the risk of trading, namely we will be too conservative to trade. So it is important for us to investigate the best number of days to calculate the average price.

As we can see in the Figure 2.13, plots based on the five-day increase in the calculation of the average value, the maximum increase is 2000, but based on the first ten days of the increase in the calculation of the average value, the maximum increase will drop to 1000, which is easy to miss the dramatic rise, so we will calculate our bias rate based on the five-day increase .

Based on Figure 2.13, Gold rise is small, and the average value can be calculated according to the 15 days of the rise.

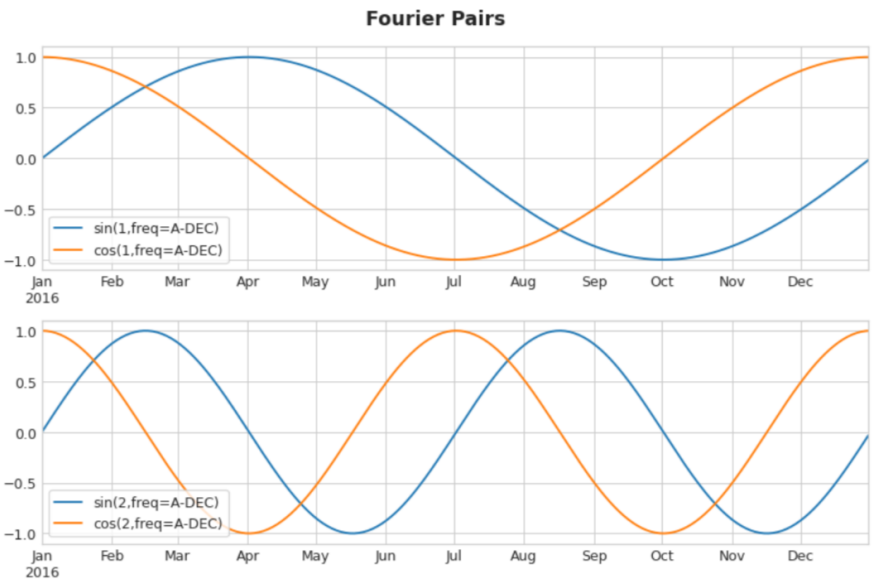

Figure 2.11: Fourier Pairs

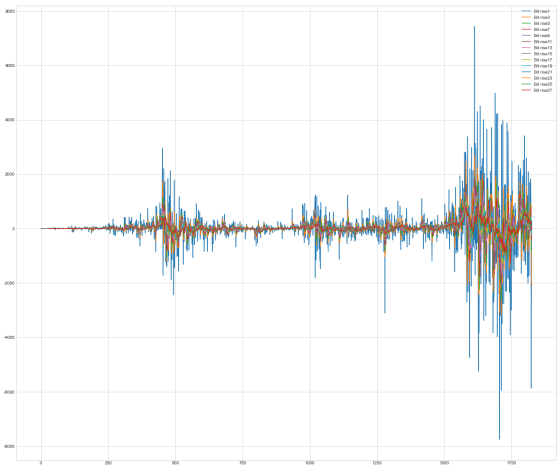

Figure 2.12: The rise of bitcoin based on the rise of different days

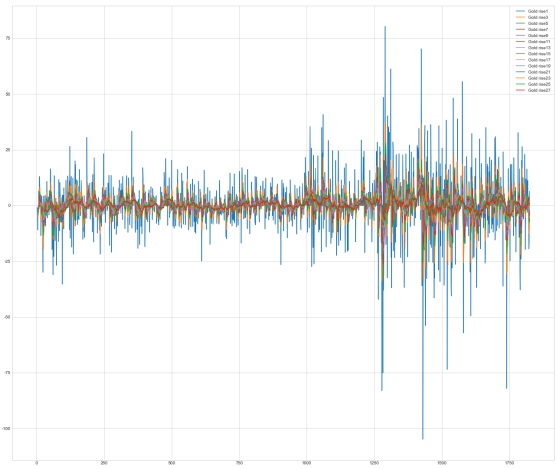

Figure 2.13: The average value of gold based on the rise of different days

2.3.2 Average Value (without distinguishing whether it is a trading day)

Based on the days of rise we select, we can calculate the average value accordingly on Figure 2.14.

2.3.3 BIAS(without distinguishing whether it is a trading day)

The BIAS is calculated according to the equation below:

$$BIAS = (CV - Ave_D)/Ave_D \tag{3}$$

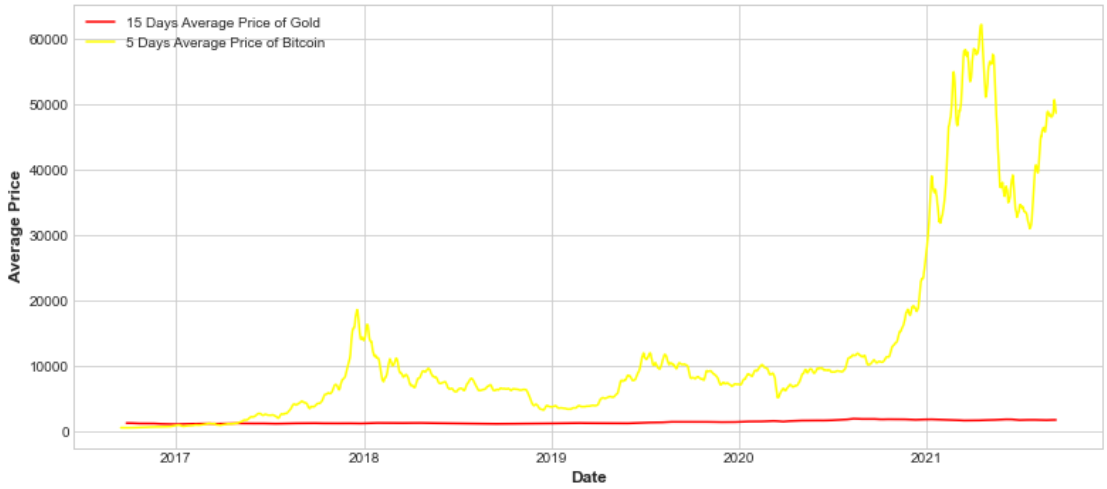

Figure 2.14: The final choice of average value

where CV means current price and the means D days average price.

And we can plot BIAS figure (Figure 2.15) according to the equation above.

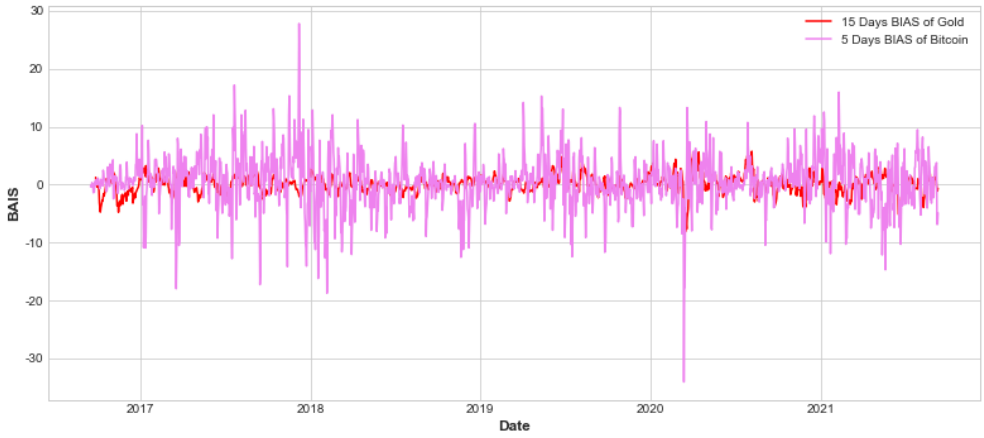

Figure 2.15: The BIAS of Gold and Bitcoin

As we can clearly see from the Figure 2.15, bitcoin has much higher BIAS than gold which means you may earn much more or lose much more if you trade bitcoin than trade gold.

2.3.4 Bull Market Assessmemt

As we can clearly see in the Figure1,2,4, the data is relatively concentrated, as a result we will use Min-Max Normalization to Normalize the data we get. The Min-Max Normalization can be described as:

$$Normalization = \frac{CV - MIN}{MAX - MIN} \tag{4}$$

where MIN means the minimal value of the data and the MAX means the maximal of the data.

We can calculate the Bull market assessment indicators according to the empirical equation:

$$BullMarketIndicator = 0.66 \times Ave_{D1} + 0.333 \times BIAS_{D2} \tag{5}$$

where $BIAS_{D2}$ means the average BIAS through previous days. And the gold's D1 is 90 and D2 is 15 while the bitcoin's D1 is 30 and D2 is 15. And We obtain the weight 0.666 ,0.333 and the D1, D2 of gold and bitcoin by means of 100000 iterations of Monte Carlo Simulation.

According to the equation, we can plot the indicator (Figure 2.16, Figure 2.17).

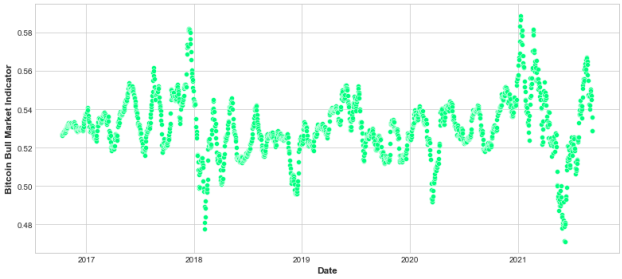

Figure 2.16: Bitcoin Bull Market Indicator

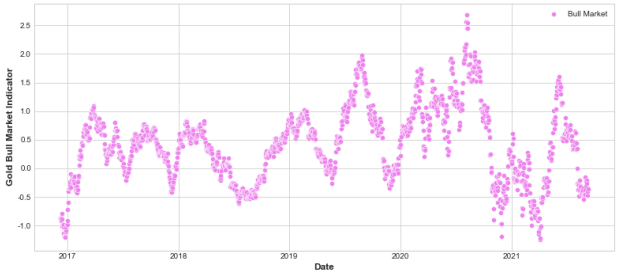

Figure 2.17: Gold Bull Market Indicator

Through 100000 iterations of Monte Carlo Simulation, we find out that when the indicator is above 0.57 is appropriate to define whether it is now a bull market for gold or bitcoin. According to it, we can plot the initial distribution of bull market and bear market of gold and bitcoin (Figure 2.18 and Figure 2.19)

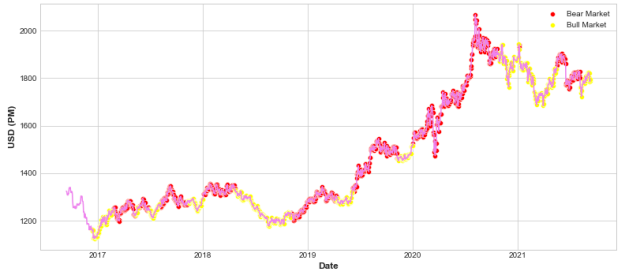

Figure 2.18: The Initial Distribution of Bull and Market for Gold

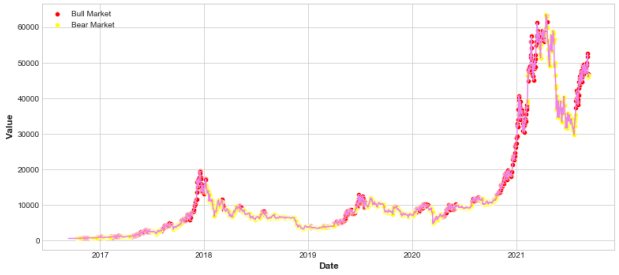

Figure 2.19: The Initial Distribution of Bull and Market for Bitcoin

However, problems will arise if we determine the distribution of bull and bear market. For example, according to the indicator concluded that today is a gold bull market, then from a quarter ago to today is a bull market, but yesterday was calculated as a bear market, and tomorrow is also calculated as a bear market, then it is possible that the error of the results of today's calculation is large.

To solve this error, the initial value is 0 for all times, and if the current calculation is a bull market, the value of the previous quarter is added by 1, and if it is a bear market, it is subtracted by one. The final result is greater than 0 for a bull market and less than 0 for a bear market.

According to this optimization, we can plot the final distribution of bull market and bear market of gold and bitcoin (Figure 2.20 and Figure 2.21)

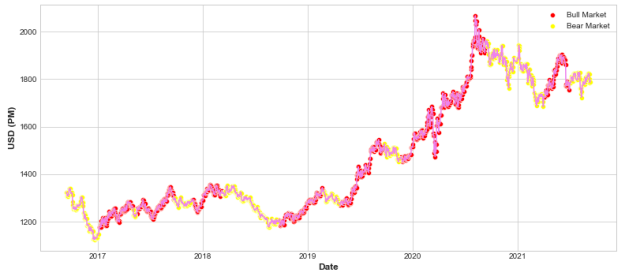

Figure 2.20: The Final Distribution of Bull and Market for Gold

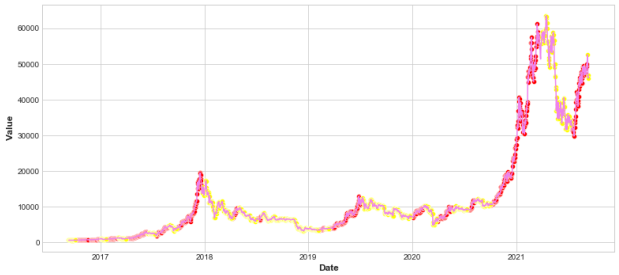

Figure 2.21: The Final Distribution of Bull and Market for Bitcoin

2.3.5 Rise Analysis

Based on the BIAS and the bull market assessment indicator, we can define the risk of gold and bitcoin as:

$$Risk = 0.666 \times BMI + 0.333 \times BIAS_D \tag{6}$$

where BMI means the bull market indicator. And D for the gold is 15 and for the bitcoin is 5. And We obtain the weight 0.666 ,0.333 and the D1 of gold and bitcoin by means of 100000 iterations of Monte Carlo Simulation.

According to the equation, we can plot the risk of trading gold and bitcoin (Figure 2.22)

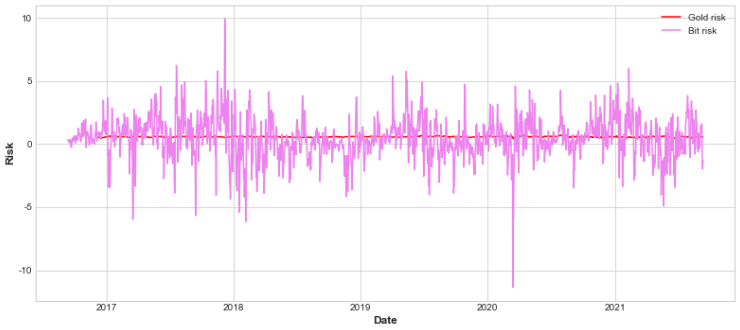

Figure 2.22: The Risk of Trading Gold and Bitcoin

3 Model Construction

3.1 Machine Learning Model

3.1.1 Linear Regression and Extreme Gradient Boosting

We develop multivariable linear regression[3] model based on the features we engineer above. The model can be described as:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_m x_m + \epsilon \tag{7}$$

$$\epsilon \sim \mathcal{N}(0, \sigma^2) \tag{8}$$

In the equation $\beta_0, \beta_1, \beta_2, \cdots, \beta_m, \sigma^2$ are unknown variables which have no correlation with $x_1, x_2, \cdots, x_m$; $\beta_0, \beta_1, \beta_2, \ldots, \beta_m$ are regression coefficient, y is the dependent variable and $x_1, x_2, \ldots, x_m$ are independent variables.

By virtual of least square method, we calculate the outcome(Figure 3.1):

Figure 3.1: Fitting Figure of Bitcoin by means of Linear Regression

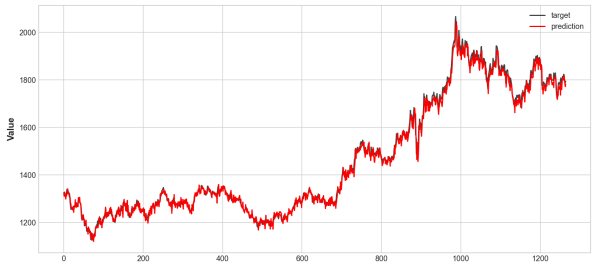

Figure 3.2: Fitting Figure of Gold by means of Linear Regression

The R2 of the bitcoin linear regression model is 0.92 and the R2 of the gold linear regression is 0.93, so the models are relatively appropriate for predictions.

From the figures(Figure 3.1 and Figure 3.2) and the model equation, we can learn that the linear regression is able to predict target values beyond the training set. However, without appropriate features, the linear regression is not able to learn patterns with complexity. The figure 3.3 and figure 3.4 show that complex patterns exist in the residue between the target values and linear regression prediction. Consequently, we introduce extreme gradient boosting algorithm to learn the complex pattern in the residue. This ensemble technique is called gradient boosting.

Extreme Gradient Boosting[2] is an efficient implementation of the Gradient Boosted Trees algorithm. It is an supervised learning method that is based on function approximation by optimizing specific loss functions as well as applying several regularization techniques. Considering we have a dataset which has m samples and n features:

$$\mathcal{D} = \{(X_i, Y_i)\}(|\mathcal{D}| = n, X_i \in \mathbb{R}^m, y_i \in \mathbb{R}) \tag{9}$$

According to the ensemble idea, we need to use k additive functions to get the final prediction $\hat{y}_i$.

$$\hat{y}_i = \phi(X_i) = \sum_{k=1}^{K} f_k(X_i), f_k \in \mathcal{F} \tag{10}$$

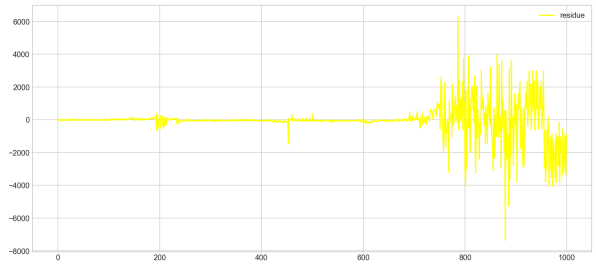

Figure 3.3: The Residue between the Linear Regression Prediction and Target Values of Bitcoin

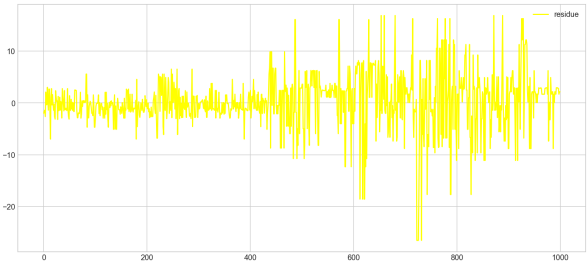

Figure 3.4: The Residue between the Linear Regression Prediction and Target Values of Gold

The f here represents the structure of each regression tree which can be described as

$$\mathcal{F} = \{f(x) = w_{q(x)}\}(q\colon \mathbb{R}^m \to T, w \in \mathbb{R}^T) \tag{11}$$

The q here represents the concrete structure of a tree, w represents the score of the tree here and the T here represents the number of the nodes there.

As a result, we can come to the scoring function:

$$\begin{gathered}\tilde{\mathcal{L}}^{(t)}(q) = -\frac{1}{2}\sum_{j=1}^{T}\frac{(\Sigma_{i\in I_j} g_i)^2}{\Sigma_{i\in I_j} h_i + \lambda} + \gamma T \\ \text{where } h_i = \partial_{y_i{}^{(t-1)}} l(y_i, \hat{y}^{(t-1)}) \text{ and } h_i = \partial^2_{y_i{}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})\end{gathered} \tag{12}$$

are the first and second order gradient statistic on the loss function, l is a differentiable convex loss function that measures the difference between the prediction $\hat{y}_i$ and the target y_i. And the γT is the regularized Term.

Normally it is impossible to enumerate all the possible tree structures q. A greedy algorithm that starts from a single leaf and iteratively adds branches to the tree is used instead. Assume that IL and IR are the instance sets of left and right nodes after the split. Lettting I = IL $\cup$ IR, then the loss reduction after the split is given by

$$\mathcal{L}_{split} = \frac{1}{2}\left[\frac{(\Sigma_{i\in I_L} g_i)^2}{\Sigma_{i\in I_L} h_i + \lambda} + \frac{(\Sigma_{i\in I_R} g_i)^2}{\Sigma_{i\in I_R} h_i + \lambda} - \frac{(\Sigma_{i\in I_I} g_i)^2}{\Sigma_{i\in I_I} h_i + \lambda}\right] \tag{13}$$

This formula is usually used in practice for evaluating the split candidates[2].

Next, we apply the exact greedy algorithm to find the best split of trees.

By training through the dataset, we finally generate an extreme gradient boosting tree which can be utilized to learn the complex pattern of the residue. As we can see in the algorithm and the two figures (figure 3.5 and Figure 3.6), the extreme gradient boosting can only make good predictions within the training size. Once the target values are beyond the bound of training values, the algorithm will make poor predictions. As a result, the extreme gradient boosting can not learn the trend component in the time series datasets.

Figure 3.5: Fitting Figure of Bitcoin
by means of Extreme Gradient Boosting

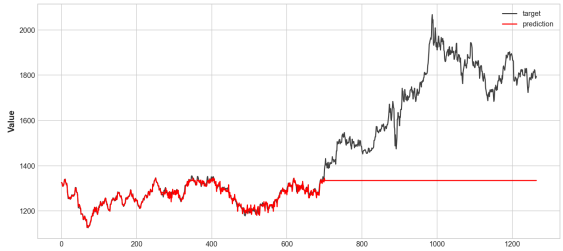

Figure 3.6: Fitting Figure of Gold
by means of Extreme Gradient Boosting

3.1.2 Ensemble linear regression, extreme gradient boosting

By means of the gradient boosting, we ensemble linear regression and extreme gradient boosting and came to the final hybrid prediction model of machine learning. The ensemble algorithm uses linear regression to learn the trend values in the time series datasets and applies extreme gradient boosting algorithm to detrended residues. The result (Figure 3.7 Figure 3.8) is more than perfect with both MSE of prediction on the validation dataset are below 15.

Figure 3.7: Fitting Figure of Bitcoin
by means of the Ensemble model

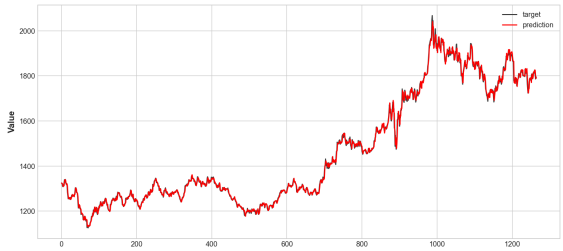

Figure 3.8: Fitting Figure of Gold
by means of the Ensemble model

3.2 ARIMA Time Series Analysis

Plotting the gold price trend(Figure 3.9), we can see that the stock price trend for gold is broadly up, but there is some volatility. After the outbreak in 2020, there was a more significant decline in the stock price because of the impact of the epidemic on the global economy.

ACF (autocorrelation coefficient) describes whether the present stock price is correlated with all prices between a certain period in the past. PACF (partial autocorrelation coefficient) describes how the present price is simply correlated with a certain price in the past. Both values range from -1 to 1. The closer the absolute value is to 1, the more obvious the correlation is; the closer the absolute value is to 0, the weaker the linear correlation is between the two. The blue shading in Figure 2.9 and the black dashed part in Figure 3.10 indicate the error range, and when the value does not exceed the range, it can be basically considered that they are not correlated.

Based on the fact that the ACF mostly exceeds the critical value, we can assume that this data set is not white noise data. If it is white noise, then there is no need to continue modeling. Because white noise data is a randomly wandering data series, such data cannot be used to build a model to predict the trend.

Figure 3.9: Gold price trend

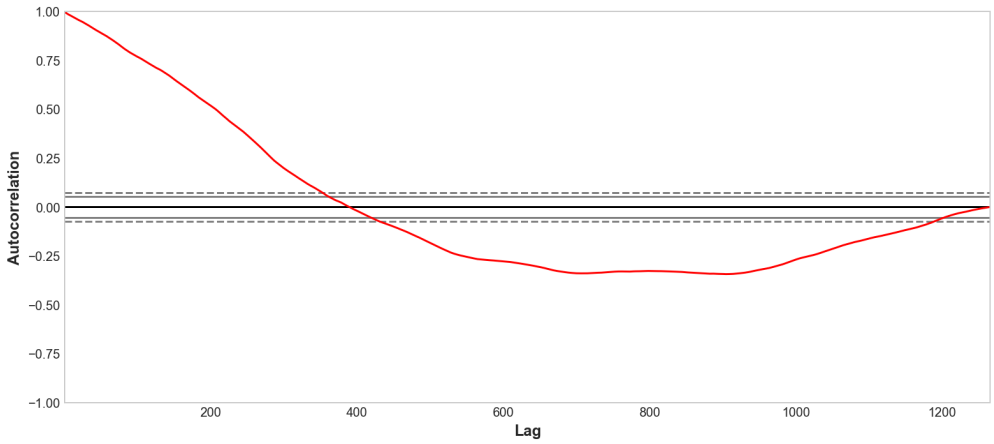

Figure 3.10: ACF chart for gold

Based on the condition that ACF slowly approaches zero and PACF plummets to zero since lag_2, we can assume that this data is a non-stationary data series.

In the above process, we can conclude that this data series has a pattern worthy of our exploration, but it is a non-stationary data. And the basis of using ARIMA[1] model is that the data analyzed must be a smooth time series. So we now need to use the difference method to transform the unsteady data into steady data. After the transformation, ADF test is performed and if the p-value is less than 0.01, the original hypothesis is rejected and then the data is smooth.

The plotted gold stock price differential comparison graph is shown in Figure 3.11. By calculation, it can be obtained that the p-value of the original data of gold stock price is greater than 0.05, which does not meet the requirement of smoothness, and the p-value of the first-order difference data is less than 0.05, and the t-value is less than 1% under the statistical value, which can be highly significant to reject the original hypothesis, indicating that the data is smooth. Because the first-order difference data are already smooth, there is no need to calculate the second-order difference. We use the exhaustive method for ARIMA model auto-fitting to find the more suitable

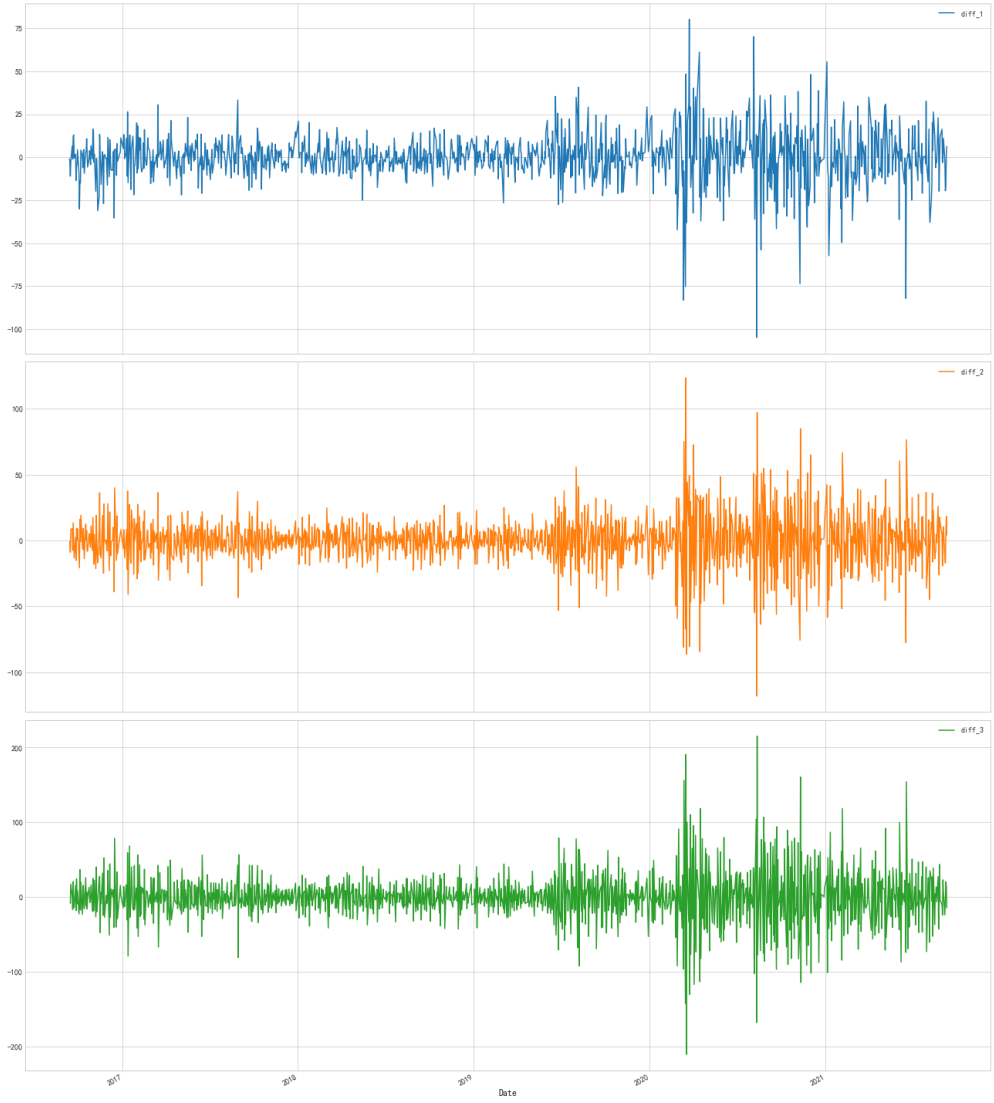

Figure 3.11: Gold differential comparison chart

p-value and q-value. If the model passes the residual test, it means that there is no autocorrelation between the residuals, and the information in the time series is basically mined, so there is no need for further correction and adjustment of the model. Then the results are roughly in line with white noise, so we plotted the gold stock price ARIMA residual QQ chart(See Figure 3.12).

Next, we use the ARIMA model to forecast, the forecast results are also first-order difference, that is, the increase in the price of gold shares, as shown in Figure 3.13.

The prediction for bitcoin's rise is similar to the gold stock price prediction method described above and will not be repeated.

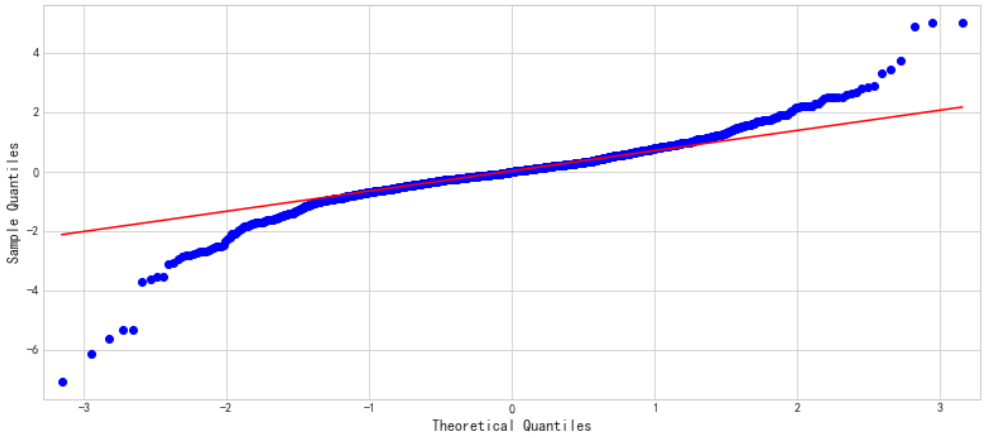

Figure 3.12: Gold stock price ARIMA residual QQ plot

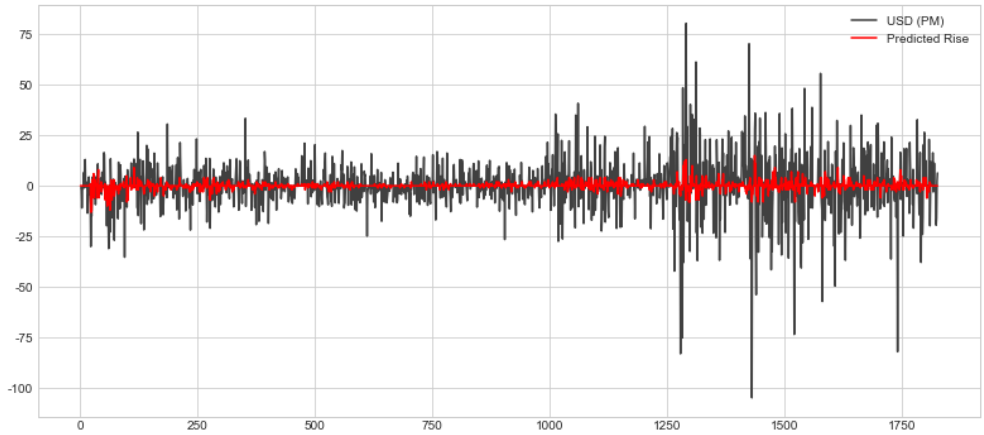

Figure 3.13: Gold stock price prediction increase chart

3.3 Final Hybrid Prediction Model

To achieve better accuracy of prediction, we modify the machine learning model by ensembling linear regression and ARIMA together. We regard the average of their output as the first step output of the hybrid machine learning model. This measure further diminishes the error the linear regression makes. Consequently, we achieve better accuracy with maximum MSE of 3.156 which is much better than any model we have proposed. The fitting figures are Figure 3.14 and Figure 3.15.

3.4 Buying Strategy

We intend to make buying decisions by quantifying the respective buy scores of gold, bitcoin for each day, whether to buy or sell, and how much it is.

$$S = 10R + 5N - C + \frac{1}{Risk} \tag{14}$$

Figure 3.14: Bitcoin Fitting Figure

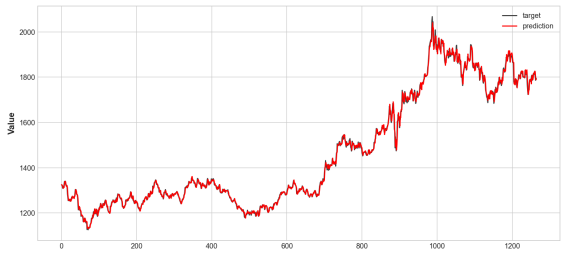

Figure 3.15: Gold Fitting Figure

The above equation is our proposed formula for calculating the buy score, where the meanings of the letters are listed in table 1. The coefficients in front of each indicator in the formula are reasonable

Symbol	Definition
S	Buy score from appraisal
R	Expected increase
N	Indicators to determine if it is a bull market
C	Residual value of purchased items
$Risk$	Evaluation of the resulting purchase risk

Table 1: Notations of equation 14

values obtained by the team through exhaustive enumeration and several active adjustments. To verify their reasonableness, we plotted the daily buy scores, as shown in Figure 3.16 Figure 3.17

Figure 3.16: Gold Buy Score Chart

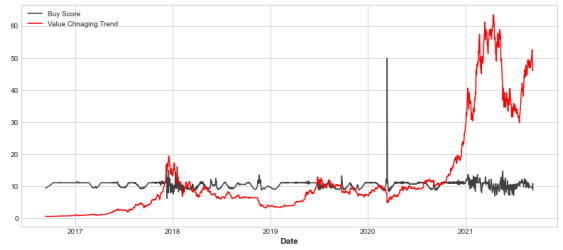

Figure 3.17: Bitcoin Buy Score Chart

As can be seen from the graph, there is a very large value in Figure 3.17 that will affect the rest of the results when normalizing, so we remove the maximum value and reset the value after normalization. The normalized bitcoin buy score graph is shown in Figure 3.18.

The price trends of gold and bitcoin are plotted separately and compared to the buying score of both. The coefficient is considered reasonable if it is observed that over the same time period, the rating decreases when the price increases and increases when the price decreases.

3.4.1 Simulated Purchase

We set thresholds for sell and buy scores to make purchase decisions. The normalized buy score ranges from 0 to 1, and we obtain the threshold value that maximizes the benefit by exhaustive

Figure 3.18: Bitcoin scoring graph (normalized)

enumeration in the range 0 to 1. The final thresholds we choose are 0.3 for gold and 0.58 for buy, and 0.56 for bitcoin and 0.62 for bitcoin. Bitcoin's thresholds are larger than gold's because gold is more stable than bitcoin, and the scores calculated from the formula are more stable and more sensitive to large values after normalisation. The threshold range for bitcoin is smaller, and we speculate that the model is willing to take a greater risk in buying bitcoin because of the greater likelihood of profitability due to the larger rise in bitcoin. Our plot of the buy scores is shown in Figure 3.19 and Figure 3.20.

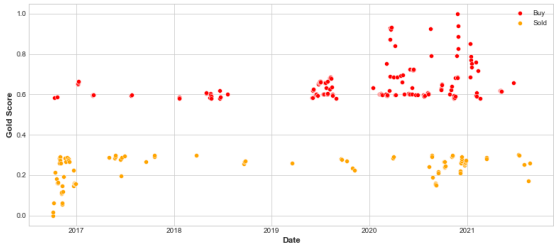

Figure 3.19: Gold rating comparison chart

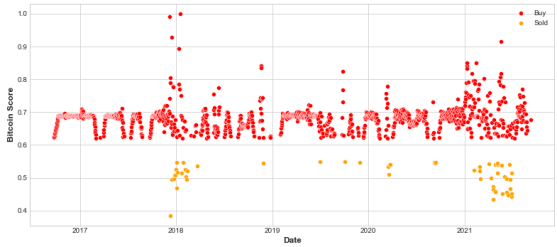

Figure 3.20: Bitcoin scoring comparison chart

The rules for buying and selling are as follows.

- Buy if gold score is greater than 0.58, sell if less than 0.3, buy if bitcoin score is greater than 0.62, sell if less than 0.56.
- Determine if it is a gold trading day, if yes, consider gold; if no, do not consider gold. When bitcoin and gold can be bought at the same time, if $(\text{Gold Buy Score} - 0.58) > 2 \times (\text{Bitcoin Buy Score} - 0.62)$ (gold score buy criteria is lower than bitcoin, more upside).

$$\text{Buy Amount} = \text{Current Cash Amount} \times \text{Buy Score} \times \frac{(1 - \text{Commission})}{\text{Current Price}} \tag{15}$$

$$\text{Sell Amount} = \text{Share Held} \times (1 - \text{Score} + \text{Sell Criteria}) \tag{16}$$

Starting a simulated trade according to the rules, we can plot a total asset trend roughly as follows, with a final total asset of approximately $76211.40

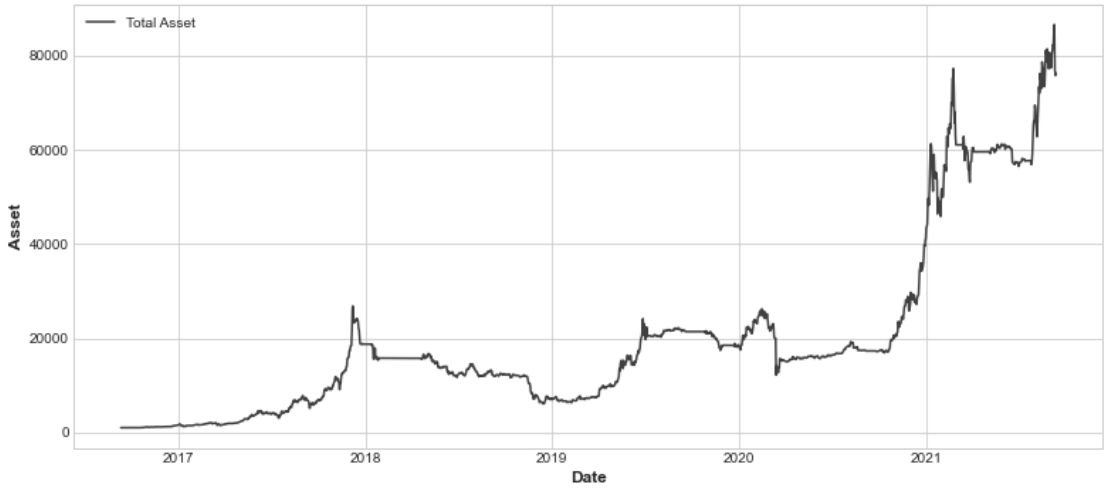

Figure 3.21: Total assets trend

3.4.2 Explore the sensitivity of the model to transaction costs

We simulate the model with all trades at 1% to 10% gold transaction costs and 1% to 20% bitcoin transaction costs and plot the maximum total assets at different transaction costs as follows. After applying linear regression and least square method to the simulation data, we can come to the equation below:

$$\begin{aligned} &\text{Total Assets} = \alpha_1 q_1 + \alpha_2 q_2 \\ &\text{where } \alpha_1 = -158839.52\ ,\ \alpha_2 = -148597.59\ , \\ &q_1 \text{ is transactioncost of bitcoin and } q_2 \text{ is transaction cost of gold.} \end{aligned} \tag{17}$$

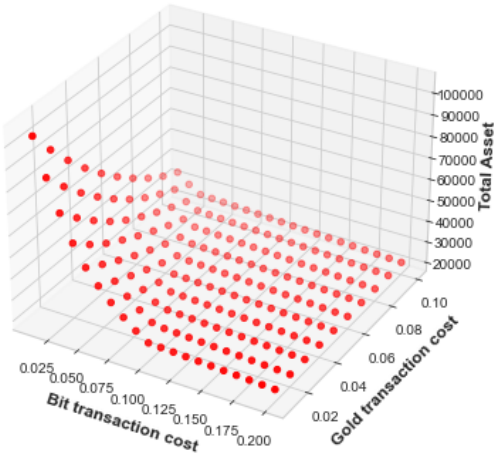

Figure 3.22: Trend of maximum total assets with different fees

4 Model Validation

To validate the model, we use global dynamic programming in "omniscient perspective" to find the theoretical maximum profit of $7357.45 and $358,499.29 for the gold-only and bitcoin-only scenarios respectively. In the gold-only case, the theoretical maximum profit is much smaller than the result of our model simulation because the price of gold is more stable. In the case of trading only bitcoin, the theoretical maximum profit is larger than the result predicted by our model. This is because in "omniscient perspective", the price fluctuations of bitcoin are completely predicted in advance, while we tend not to invest all our money in hoarding bitcoin when the price is about to rise sharply in real trading, given the trading risks. However, in this case, the theoretical maximum profit does not exceed the predicted value of our model by much. We also adopt greedy algorithm under the circumstance that we know the prices of all time. And the result is $50,782. Therefore, we believe that our model is relatively reasonable and valid.

5 Conclusion

In order to solve the problem of using only the past stream of daily prices to date to determine each day if the trader should buy, hold, or sell their assets. We propose a series of novel models to find the patterns of gold and bitcoin price fluctuations as much as possible, and use the patterns to achieve the prediction of future price fluctuations with the aim of obtaining the highest possible profit.

1. We hybrid linear regression, ARIMA and extreme gradient boosting together to develop our prediction model. Considering the characteristics of small and slow change of gold price and large and fast change of bitcoin price, we adopt to predict the closing price of both using the past 15 days and 5 days respectively. We analyze the past 90 days and 30 days of data, respectively, to make judgments about the market, risk, and other factors to derive the various elements we need in our trading model.

2. We propose an PBRT trading model based on buy score. The buy-sell thresholds are set through Monte Carlo methods with human adjustments, and by considering elements such as cash and purchase scores, we ensure that the trade does not lead to an unrealistic situation where one item becomes negative.

3. To verify the model robustness and effectiveness, we derived the maximum profit obtained under different scenarios by changing the weights of each element in the PBRT model. We also compare the maximum profit value with the results obtained from our model using the "omniscient perspective" case of trading only bitcoin or gold. Conclusions are drawn and it is found that the results obtained by our model are much better than the other cases mentioned above. Also, we vary the transaction cost, which is the alpha, plot the theoretical maximum for the case of different alpha values, and perform a two-dimensional fit. From the fitting results, it is clear that our model is more stable and has good continuity.

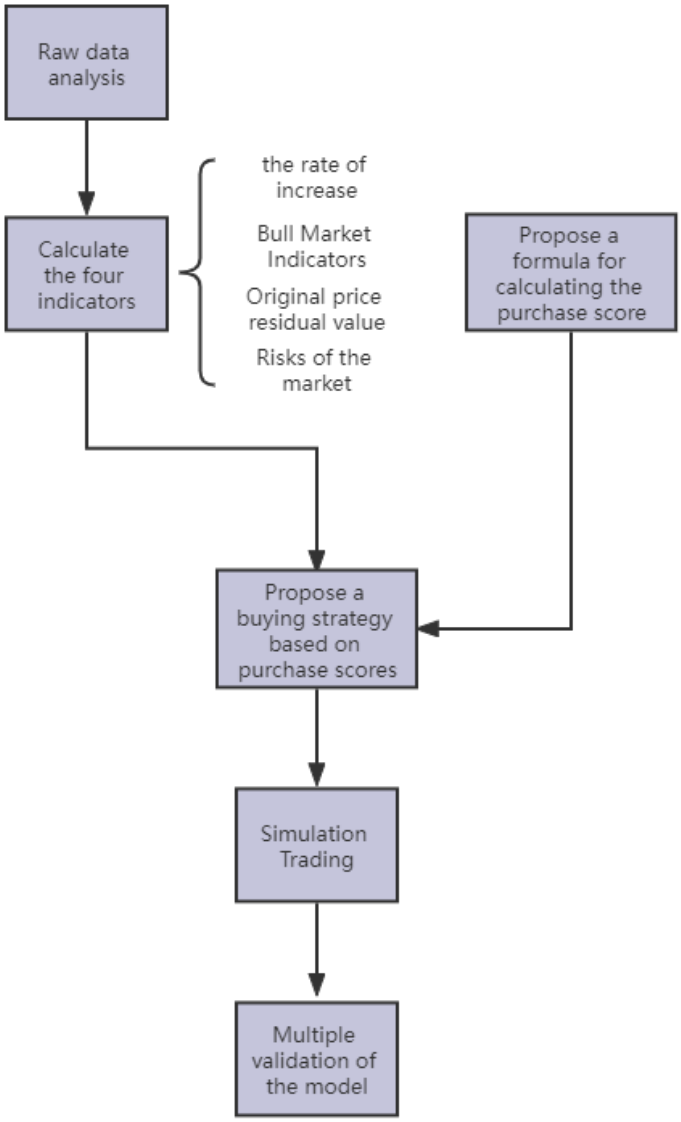

Figure 5.1:

References

[1] Matyjaszek, M., Fernández, P., Krzemień, A., Wodarski, K., & Valverde, G. (2019). Forecasting coking coal prices by means of ARIMA models and neural networks, considering the transgenic time series theory. Resources Policy, Volume 61, June 2019, Pages 283-292

[2] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16). Association for Computing Machinery, New York, NY, USA, 785–794. DOI:https://doi.org/10.1145/2939672.2939785

[3] https://en.wikipedia.org/wiki/Linear_regression

Core source code

A. Self-design machine learning estimator

```
class xgb_lin():
    def __init__(self):
        self.xgb=CatBoostRegressor(gpu_cat_features_storage=True,devices='gpu',
\
    verbose=False)
        self.lin=LinearRegression()
    def fit(self,x,y):
        lin_fit=self.lin.fit(x,y).predict(x)
        self.xgb.fit(x,y-lin_fit)
        print(self.lin.score(x,y))
    def predict(self,x):
        sns.lineplot(x=np.arange(len(x)),y=self.xgb.predict(x),color='yellow',label= \
    'residue')
        return self.lin.predict(x)+self.xgb.predict(x)
```

B. Find out the best ARIMA Model

```
best_aic = np.inf
best_order = None
best_mdl = smt.ARIMA(data['Value'],order=(8,1,8)).fit(method='mle',trend='nc')
#8 1 8
ads=data['Value']
pq_rng = range(10) # [0,1,2,3,4]
d_rng = range(3) # [0,1] ARMA
for i in pq_rng:
    for d in d_rng:
        for j in pq_rng:
            try:
                tmp_mdl = smt.ARIMA(ads, order=(i,d,j)).fit(method='mle', trend='nc')
                tmp_aic = tmp_mdl.aic
                if tmp_aic < best_aic:
                    best_aic = tmp_aic
                    best_order = (i, d, j)
                    best_mdl = tmp_mdl
            except: continue

print('aic: {:6.5f} | order: {}'.format(best_aic, best_order))
_ = plt.plot(best_mdl.resid, lags=8)
print(best_mdl.summary())
## Durbin-Watson
print(sm.stats.durbin_watson(best_mdl.resid.values))
```