Problem Chosen
C

2022 MCM/ICM Summary Sheet

Team Control Number
2227194

RC Algorithms Have Made Us Rich

Abstract

Bitcoin is a digital virtual currency. In recent years, the Bitcoin market has developed rapidly and has been recognized as a new type of gold by many investors. It may replace gold as a hedge against inflation and become a new type of investment asset for financial management. The investment relationship with gold has more and more important research value and practical significance. So, we were asked by a trader to develop a model that uses daily price flow data from September 11, 2016 - September 10, 2021, to determine whether a trader needs to buy, hold or sell investments on a daily basis Assets in a portfolio consisting of USD, Gold and Bitcoin. Therefore, the purpose of our report is to establish a predictive model of investment trading strategies to help traders maximize investment profits. At the same time, we will also provide traders with some trading strategies and suggestions to help him better understand the financial market and reduce the financial risk of his investment. To this end, we established two models: Model 1: a linear regression prediction model based on machine learning; Model 2: KNN algorithm and QDA analysis combined model.

For Model I, first we got the pricing data of gold and bitcoin within 5 years, and then we preprocessed the data, and used machine learning to train and validate the data to predict the price fluctuations and their price fluctuations in the next n days. The average value, according to the results displayed, determine the **linear regression** method to fit the data. Finally, on this basis, to obtain the best investment strategy, we use the future price to change the ratio of gold and Bitcoin and strive to get the highest return. Proportion, that is, using the formula derived from the price change weight according to the **logistics** function to determine the final optimal daily trading strategy, thereby calculating the investment value of the initial amount.

For Model II, we used a hybrid combined model of the KNN algorithm and QDA analysis to try to assess its goodness of fit. Since the most critical data for determining the final ratio is the estimated future price fluctuations of gold and bitcoin, we use feature similarity to judge the degree of fit between the estimated data of gold and bitcoin and the real data, and then use the **KNN algorithm** to determine the degree of fit between the estimated data and the real data. **QDA analysis** makes multiple judgments to verify whether our forecast is accurate enough to provide the greatest possible return. Finally, we calculate the goodness of fit between gold and Bitcoin to help verify our estimated price. Whether the fluctuations are accurate enough.

Eventually, we also conducted a sensitivity analysis on the investment strategy we established and observed the change in the final benefit by changing the size of the initial cost of the linear change. This scheme is what we use to try Verify that the cost and benefit are highly linearly related. The results also fully show that our initial amount has a positive correlation with the final return, and it also reflects that our model is relatively stable and reliable, which can increase investors' great investment confidence, and has room for improvement.

Key words: Virtual currency; Gold; Trading strategy; Machine learning; Best yield; KNN algorithm; QDA analysis; Goodness of fit

Contests

1 Introduction

1.1 Problem Background

Market traders often buy and sell volatile assets, which is a risky investment behavior, and its volatility is usually measured by the standard deviation of fluctuations over a period. Among these tradable and volatile assets, gold, and Bitcoin play an important role.

With the rapid development of social economy, gold investment has become an important part of investors' investment portfolio. Therefore, the gold market has opened a wealth of investment channels for investors in the financial market and improved the investment function of the financial market. For Bitcoin, it is a digital currency with a constant total of 21 million, which has the same characteristics of decentralization, globalization, and anonymity as the Internet. The liquidity and limitation of Bitcoin determine that Bitcoin can serve as a currency function equivalent to a general equivalent, but this measurement is not a physical measurement like gold, but a digital product.

(a) Gold

(b) Bitcoin

Figure 1: Target portfolio

However, the properties and investment attractiveness of gold and Bitcoin are not the same, so how to make the correct investment strategy for the two has become a matter of public concern. Based on this, the trader asked our team for help, and the trader asked us to develop a model that uses only the daily price stream from September 11, 2016 - September 10, 2021, to determine whether a trader should buy, Hold, or sell assets in their portfolio, which includes cash, gold, and bitcoin.

1.2 Restatement of the Problem

In the above request, we learn that the initial state of the transaction is \$1000， and assume α_{gold} = 1% and $\alpha_{bitcoin}$ = 2%. There is no cost to hold an asset. Considering the background information and restricted conditions identified in the problem statement, we need to solve the following problems:

- Develop a model that provides the best daily trading strategy based on price data and how much the initial \$1,000 will return on investment as of September 10, 2021.
- Provide relevant evidence to clearly and accurately demonstrate that the established model provides the best investment strategy.
- Analyze how well the model built is related to transaction costs, and how traders' assumed transaction costs affect strategies and outcomes. Considering your predictive analysis, should these small fishing companies make changes to their operations?
- In the form of a memorandum, we communicate the model, investment strategy we have established to the trader, and fully demonstrate the results we have obtained.

1.3 Our work

In the research of this report, traders asked our team to establish a combination model that can determine the investment strategy of gold and bitcoin according to the price changes of gold and bitcoin in 5 years, to help traders to make better investment planning in the future. Our work mainly includes the following:

- ✧ Based on the historical pricing data of gold and Bitcoin, establish a price fluctuation prediction model for both.
- ✧ Establish a model for effective evaluation of the portfolio strategies we study.
- ✧ Based on the investment portfolio model of the financial industry, this paper studies the relationship between Bitcoin and gold and puts forward investment suggestions for maximizing benefits and conducts a sensitivity analysis of the scheme to put forward reasonable suggestions for improvement.

In summary, the whole modeling process can be shown as follows:

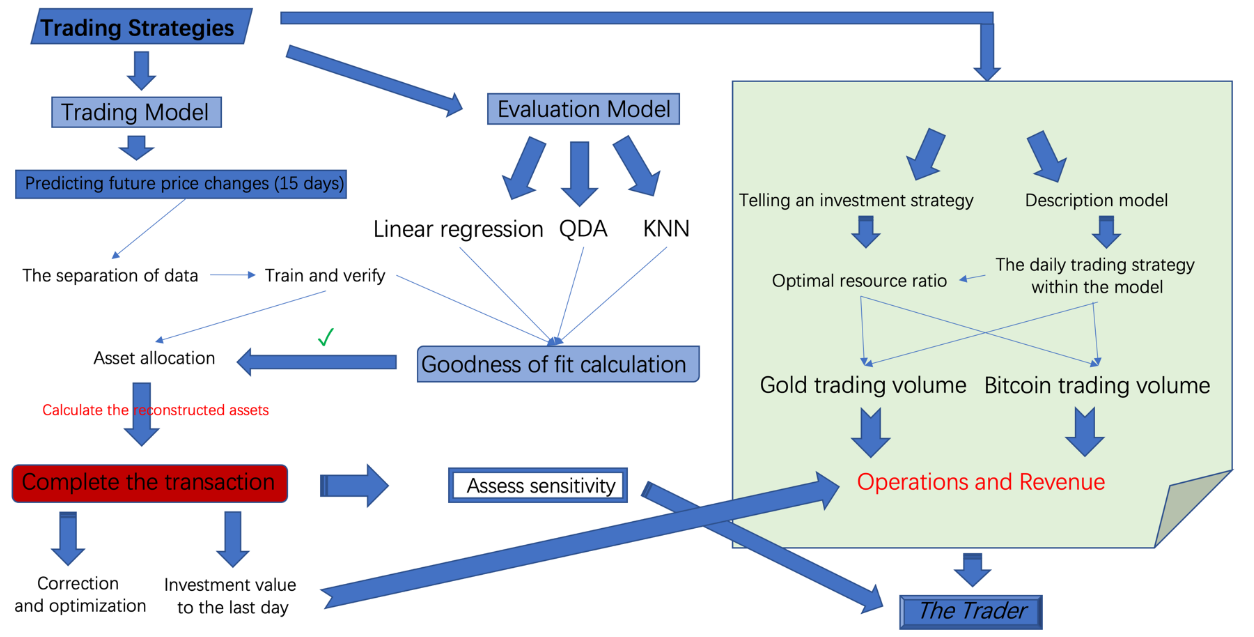


Figure 2: Model Overview

2 Assumptions and Justifications

To simplify the problem, we make the following basic assumptions, each of which is properly justified.

➢ **Assumption 1: Assume a trader only trades on gold trading days.**

Justification: Because gold is only traded on the opening day, which means that the trading status of gold during weekends or holidays will remain in the holding state, so we assume that the trading occurs on the opening day of the gold market.

➢ **Assumption 2: Suppose the trader only makes one trade per day.**

Justification: There is no limit to the number of trades per day in the trader's requirements. To enable the model to achieve higher prediction rates and accuracy, we assume that traders can only make one trade per day to minimize the associated error.

➢ **Assumption 3: It is assumed that each transaction is carried out with the minimum transaction cost method.**

Justification: In the request made by the trader, we know that the commission cost of each transaction is α % of the transaction amount, and α_{gold} = 1% and $\alpha_{bitcoin}$ = 2%, so we need to assume that the transaction cost of each transaction is the minimum state, to maximize the value of the final $1000.

➢ **Assumption 4: It is assumed that there is a decimal point in the transaction amount during the transaction of USD, gold, and Bitcoin.**

Justification: Both gold and bitcoin are easy to divide as currencies in circulation. Bitcoin can be divided to eight decimal places, which is 1 Santoshi. Although gold can also be divided, it is more troublesome than bitcoin. At one time we assumed that the transaction amount of the two can be accurate to the same decimal point during the transaction.

3 Notations

The key mathematical notations used in this paper are listed in Table 1.

Table 1: Notations used in this paper

Symbol	Description	Unit
PCT_change	Past price fluctuation	%
HL_PCT	Maximum price difference in the past	%
$present\_crash$	Cash held after the transaction	$
$present\_gold$	Post-trade gold holdings	oz.t
$present\_bitcoin$	Hold bitcoin after the transaction	BTC
δ_1	Change in the estimated price of gold (15 days later)	$/oz.t
δ_2	Change in the estimated price of Bit (after 15 days)	$/BTC
$thePriceOfGold$	The current price of gold	$

4 Model Preparation

In this section, we give a brief overview of the data we used to build the models with subsequent descriptions.

4.1 Given Data

Our model is built around how to invest in gold and bitcoin to gain maximum value, Based on this, we obtained the daily pricing data for gold and Bitcoin from 10 September 2016 to 10 September 2021, The data are from the London Gold and Silver Market Association and the NASDAQ, From the two pricing data files, LBMA-GOLD.csv and BCHAIN-MKPRU.csv, Not only have we learned about the daily price fluctuation range and rising trend of gold and Bitcoin, It can also be known that Bitcoin can be traded daily, Gold trades only on the opening day, This is a key message for us. Due to the large amount of data provided and not intuitive, we can more visualize the data in the subsequent modeling process.

5 Model I: Linear regression prediction model based on machine learning

5.1 Overview of method

Machine learning[1] is a general term for a class of algorithms designed to exploit hidden laws from large amounts of historical data to be used for prediction or classification? Specifically, machine learning can be seen as finding a function, where the input is the sample data, and the output is the desired result, but the function is too complex to formalize the expression. Machine learning models can be divided into regression models, classification models, and structural learning models. To complete the requirements made by the traders, we will build a regression model for a predictive analysis based on the gold and bitcoin data samples provided by the traders to derive our investment strategy.

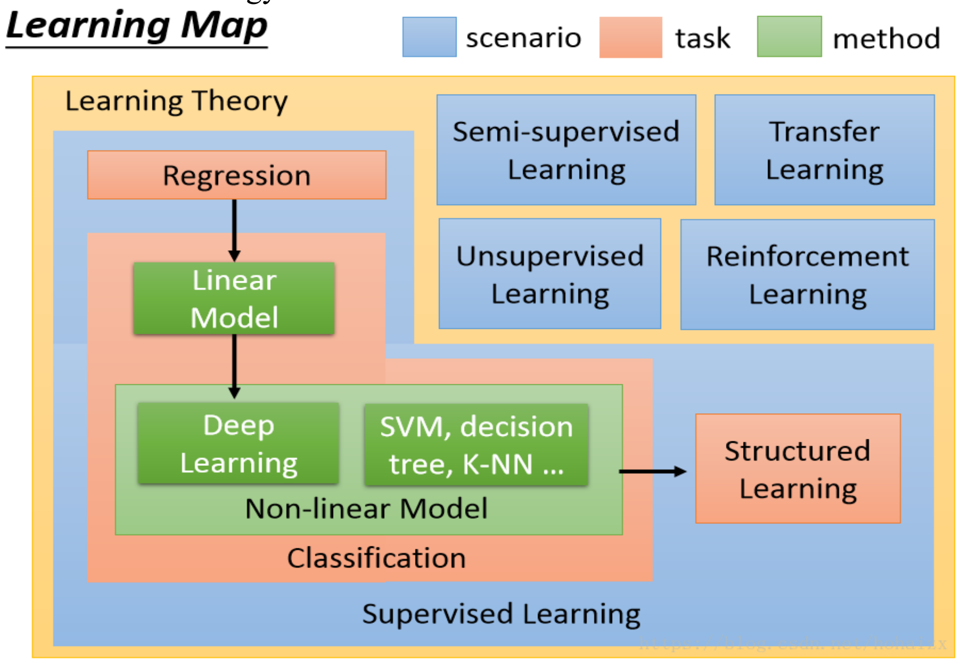


Figure3: Machine learning flow chart

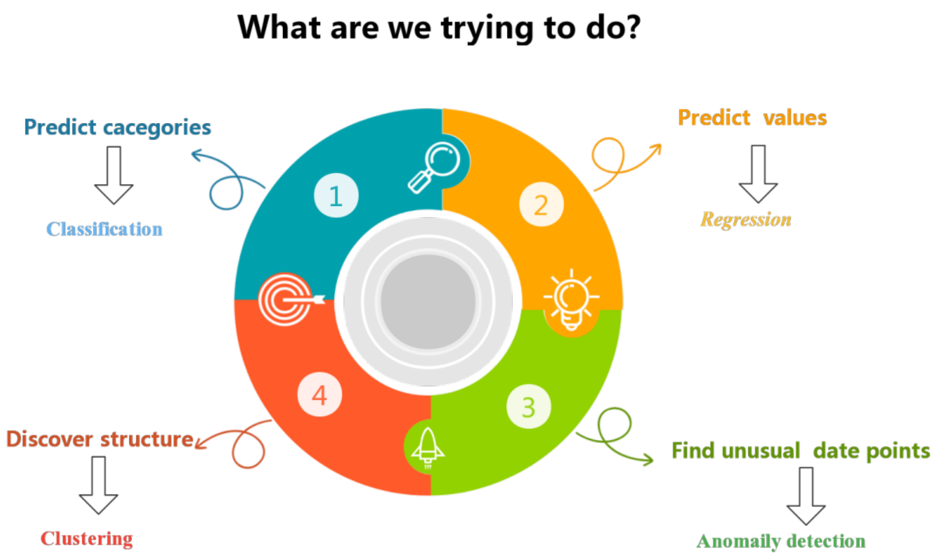


Figure4: Machine learning classification graph

5.2 data preprocessing

Based on the machine learning regression prediction model, we must ensure the accuracy and availability of the data before performing the data analysis. If machine learning is performed based on unreliable data, then accurate predictions cannot be made. According to the data files provided by the traders, gold is only traded on the opening day, so we deleted the blank lines in the LBMA GOLD.csv file. In addition, to ensure the consistency of the data and reduce the prediction error, we also deleted the pricing data corresponding to bitcoin during the gold shutdown day, making the prediction of the model can be more accurate.

5.3 Data segmentation

During the development of the machine learning linear regression prediction model that we have built, it is hoped that the trained model can perform well on new, unseen data. To simulate new, unseen data, we split the pricing data on gold and Bitcoins into two parts. Of these, the first part is a larger subset of the data, used as a training set, representing 90% of the original data, while the second part is a smaller subset, where the remaining 10% is used as a test set.

Next, we used the training set to build a predictive model, and then applied this trained model to the test set for prediction, to select the best model based on the model performance on the test set.

5.4 Linear regression model (LRM)

5.4.1 Regression to the problem

Regression is used to predict the relationship between the input and output variables, especially when the output variable changes when the value of the given input variable changes. The regression model is exactly a function representing the mapping between the input variable and the output variable. The learning of such problems is equivalent to function fitting, where a function curve is chosen to well fit the pricing data for gold and bitcoin for the period from 10 September 2016 to 10 September 2021 to predict future data prices.

5.4.2 Linear regression model description

In the $linear\ regression^2$ analysis that we established, two independent variables were included, and linear relationships were found between the dependent and independent variables. We chose the price fluctuations before the day of gold and the difference between the largest and smallest as the influence factors, where the formula in the multivariate linear regression is as follows:

$$h_\theta(X^i) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n \tag{1}$$

Here, we default that 0 is always equal to 1. For facilitate representation, we consider the hypothesis function can be written in the form of vectors:

$$h_\theta(X^i) = \theta^{\mathrm{T}} X \tag{2}$$

What's more,

$$\theta = [\theta_0, \theta_1, \dots, \theta_n]^T \tag{3}$$

$$X = [1, x_1, x_2, \dots, x_n]^{\mathrm{T}} \tag{4}$$

The influence factor solution formula is as follows:

$$\begin{aligned} Y &= \theta_1 * X_1 + \theta_2 * X_2 + CGold(ETF)price \\ &= \theta_1 * PCT\_change + \theta_2 * HL\_PCT + c \end{aligned} \tag{5}$$

To choose the most suitable linear regression model, we still used the loss function. I wanted to find out the vector that minimized the loss function. We find a line in the given gold and bitcoin data file to fit it, then I assume the equation of the line, then substitute the data point into the assumed equation to get the observation, find the parameters minimizing the sum of square of the actual value from the observation, the following formula:

$$J(\theta) = \frac{1}{2m}\sum_{i=0}^{m}(h_\theta(X^i) - y^i)^2 = \frac{1}{2m}(X_\theta - y)^{\mathrm{T}}(X_\theta - y) \tag{6}$$

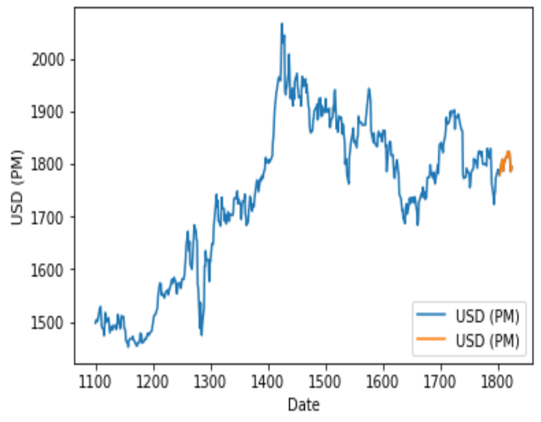
In addition, to obtain the local optimal solution, the gradient descent algorithm was also adopted in the machine learning process, and the whole process uses the following formula:

$$\frac{\partial}{\partial\theta}J(\theta) = \frac{\partial}{\partial\theta}\frac{1}{2}\sum_{i=1}^{m}(h_\theta(x) - y)^2 = (h_\theta(x) - y)x^{(i)} \tag{7}$$

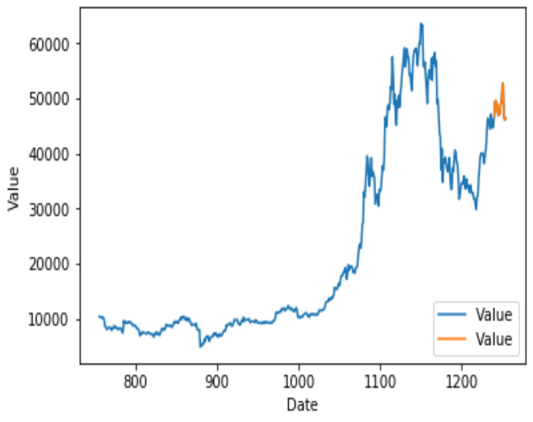
5.5 Results

5.5.1 Forecast results

With the help of python, the above modeling process can be implemented, and through the variables we set, the pricing data of gold and bitcoin can be predicted and the fluctuation range. From our fitting, the prediction accuracy is high, and the results are shown in the following figure:

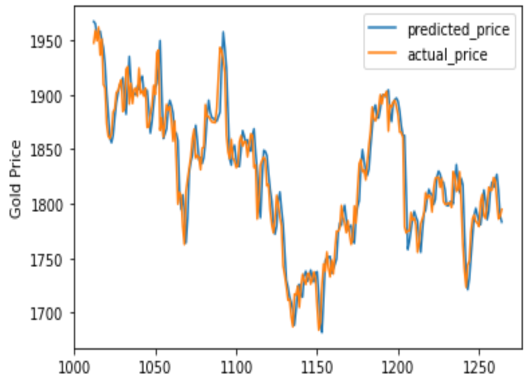


(a) Gold forecast data

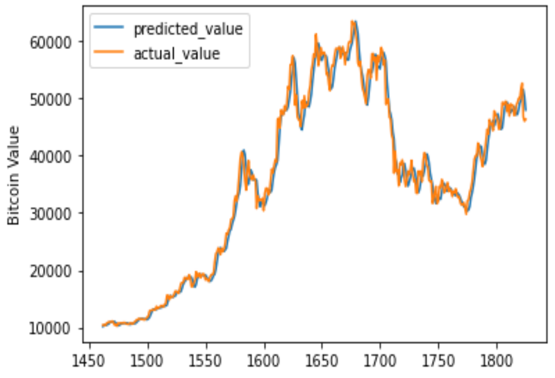


(b) Bitcoin forecast data

Figure 5: Predictions for gold and bitcoin



(a) Gold data fit



(b) Bitcoin data fit

Figure 6: Fit data for gold and Bitcoin

5.5.2 Test of goodness of fit

Mapping process is looking for very mild gold and bitcoin data sequence mathematical model process and after $fitting\ good\ data^3$, in order to better evaluate our prediction accuracy, we adopted the goodness of fit test scheme, with the concept of test probability to evaluate the quality of gold and bitcoin data fit. We used the Chi-square statistics for statistical significance tests with the following formula:

$$X_{df}^2 = \sum_{i=1}^{k} \frac{(O_i - T_i)^2}{T_i} \tag{8}$$

The calculated goodness of fit of gold is 89.44% and bitcoin is 98.43%, and the fitting accuracy is relatively accurate.

5.5.3 Price change weights determine optimal strategy

We predict the price changes of gold and Bitcoin in the fifteen days after that and take the average value as the estimated amount. According to the following formula:

$$present\_crash : present\_gold : present\_bitcoin = logistic(\delta_1 + \delta_2) : logistic(\delta_1) : logistic(\delta_2) \tag{9}$$

Where δ is the estimated average price minus the current price, $\delta_1$is the price expressed as gold, expressed $\delta_2$as the price of bitcoin;α_{gold} = 1% and $\alpha_{bitcoin}$ = 2%. In the above formula, we determine the investment strategy by controlling the daily ratio of gold and bitcoin.

$$\begin{aligned} & present\_crash + present\_gold * thePriceOfGold + present\_bitcoin \\ & * thePriceOfBitcoin \\ & = previous\_crash + previous\_gold * thePriceOfGold + previous\_bitcoin \\ & * thePriceOfBitcoin - \alpha_{gold} * \frac{previous\_gold - present\_gold}{thePriceOfGold} - \alpha_{bitcoin} \\ & * \frac{previous\_bitcoin - present\_bitcoin}{thePriceOfBitcoin} \end{aligned} \tag{10}$$

According to the following formula, we supplement the judgment:

$$\begin{aligned} & crash\_all \\ & < present\_gold + avarage\_gold * thePriceOfGold + avarage\_bit \\ & * thePriceOfBitcoin \end{aligned} \tag{11}$$

If the above formula is true, traders can trade on the opening day of the gold market, otherwise they will not trade. With this algorithm, we succeeded in arriving at the result.

6 Model II: KNN algorithm and QDA analysis combined model

To ensure that the new data predicted by gold and bitcoin are the same points in the original dataset, we decided to use the KNN algorithm to predict feature similarity. The data algorithm uses feature similarity to evaluate the accuracy of both prediction data. In addition, we used a QDA analysis to determine whether our proposed portfolio method maximizes the benefits through our portfolio model, and to provide relevant evidence for traders.

6.1 KNN algorithm application

First, we can use two simple pictures to understand how the $KNN\ algorithm^4$ can help us achieve our goal, including the distribution of red circles (RC) and green squares (GS).

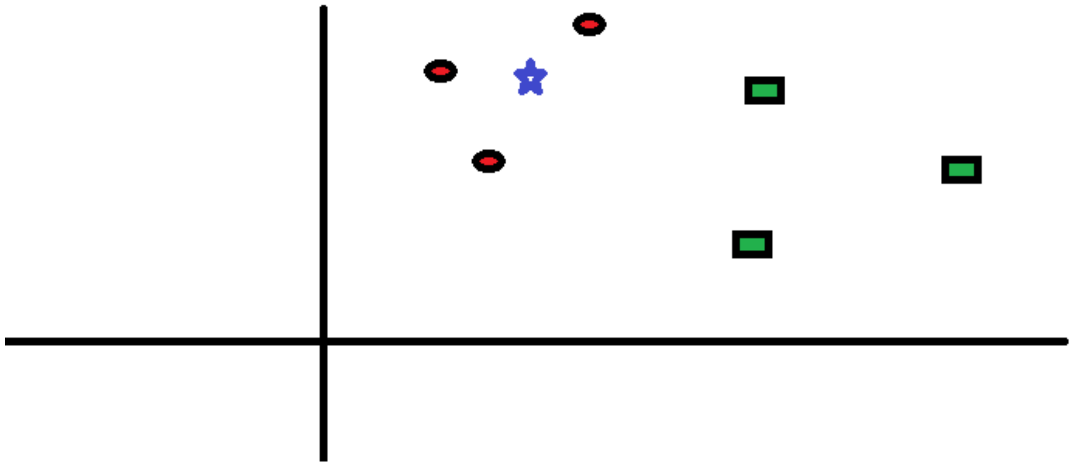


Figure 7: Distribution map 1

We intend to find out the Blue Star (BS) level, the BS can be RC or GS."K" is the nearest neighbor of the KNN algorithm that we want to select from it.

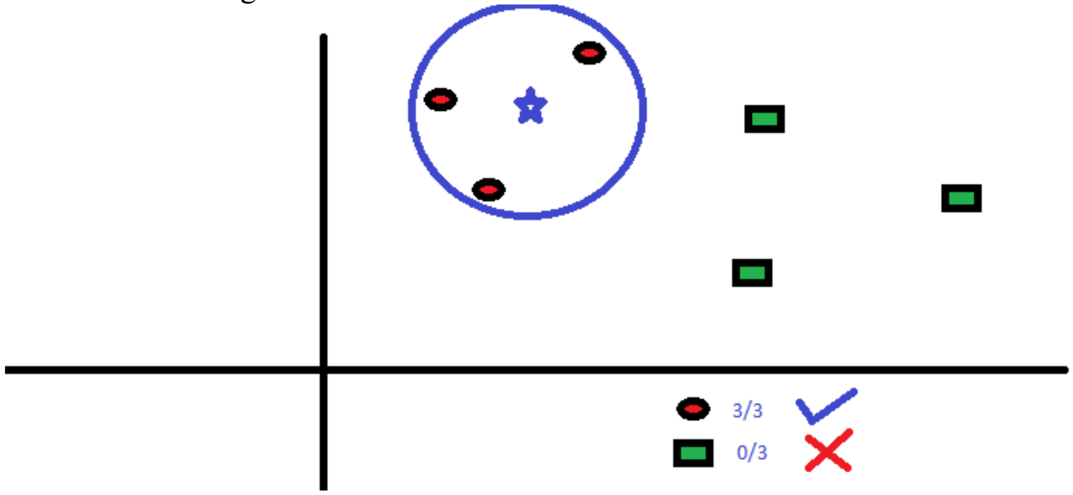


Figure 8: Distribution map 2

6.1.1 Select the appropriate K-value

In this process, we should choose the appropriate K value to predict gold and bitcoin. First, we want to train the error rate and the validation error rate, both of which require us to access two parameters of different K values. The following figure is the training error rate curve with variable K values, and we can see that the boundary becomes smoother with the increasing K value.

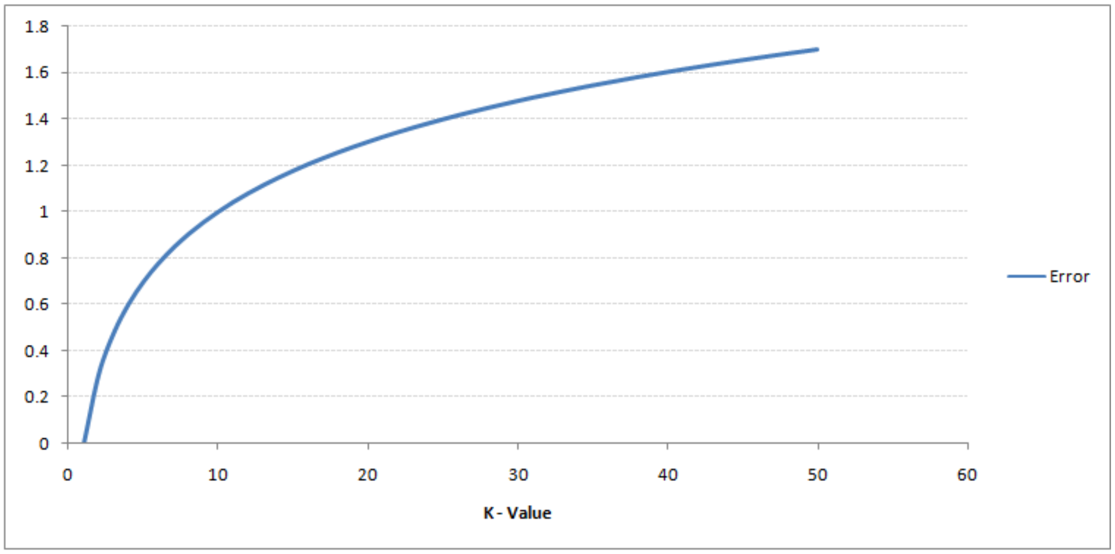


Figure 9: The training error rate curve for the K value

Figure shows that the error rate at K=1 is always zero for the training sample. This is because the point closest to any training data point is itself. Thus, the predictions are always accurate when K=1.If the validation error curve was similar, our chosen K would be 1. The following are the validation error curves with different K-values:

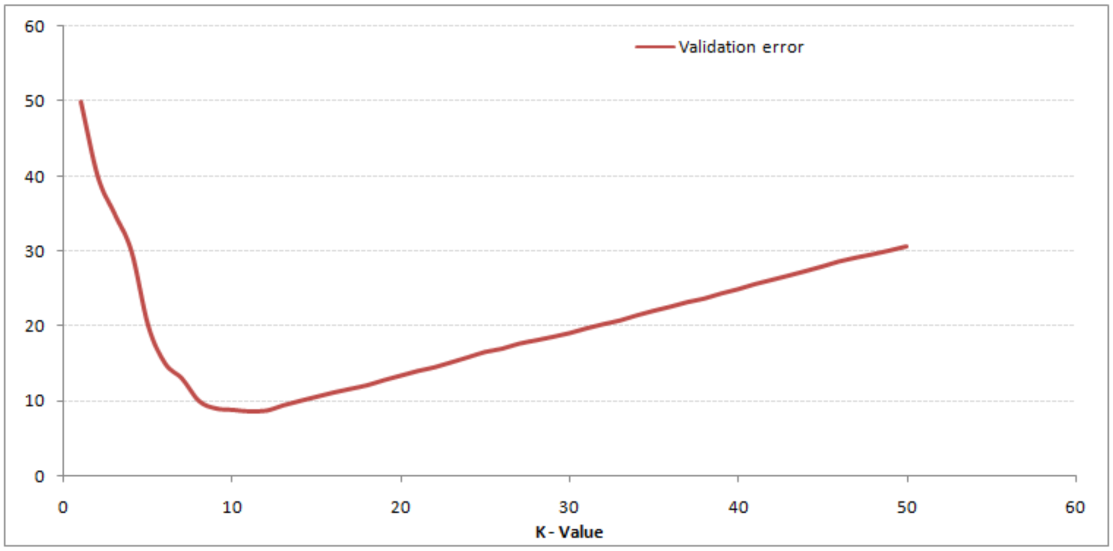


Figure 10: Validation error curves for different K values

6.1.2 Get the results

The above process makes our goals even clearer. At K=1, we overfit the boundary. Thus, the error rate was initially reduced and reached a minimum. After the minimum point, it increases with increasing K. To obtain the optimal value of K, you can isolate the training and validation from the initial dataset. The validation error curve is now plotted to obtain the optimal value of K. This K value can be applied to the prediction of gold and Bitcoin.

6.2 QDA analyze

The QDA[5] we used in the combined model can consider the non-parametric learning method KNN and the LR and LDA methods with a linear classification boundary. The QDA assumes that the classification boundary is quadratic, so it can model the problems we want to solve more accurately than the linear model. Meanwhile, due to the additional assumption of its quadratic boundary, it can outperform the KNN method when the sample size is small.

6.2.1 Establishment of a Bayesian model

According to the Bayesian model, the formula is as follows:

$$P(C_1|x) = \frac{p(C_1,x)}{p(x)} = \frac{p(C_1)p(x|C_1)}{p(C_1)p(x|C_1) + p(C_2)p(x|C_2)}$$

We only need to calculate the $p(C_1)$、$p(x|C_1)$、$p(C_2)$ and$p(x|C_2)$modeling and calculate the search$P(C_1|x)$and$P(C_1)$ belongs to a prior probability, intuitive look should be equal to$\frac{N_{c1}}{N}$。

6.2.2 Compliance to a normal distribution is assumed

For $p(x|C_1)$， we assume to obey a normal distribution and assume that x is a D-dimensional vector, then the normal distribution is as follows:

$$\frac{1}{2\pi^{D/2}|\Sigma|^{1/2}} e^{\left\{-\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)\right\}}$$

Where the parameter $\mu_k$and Σ is what can be solved by the maximum release of the training set.

Where the parameter and is what can be solved by the maximum release of the training set.

6.2.3 Maximum solution

From the above steps, we can conclude that: $p(x|C_1)$ is an exponential function, and $P(C_1)$ is a constant, so the multiplication set of both is also an exponential function, then:

$$\ln(p(C_1|x)) = (x - \mu_1)^T\Sigma^{-1}(x - \mu_1) + C$$

Similarly $p(x|C_2)$is in a similar form, so the:

$$\ln(\frac{p(C_1|x)}{p(C_2|x)}) = (x - \mu_1)^T\Sigma^{-1}(x - \mu_1) - (x - \mu_2)^T\Sigma^{-1}(x - \mu_2) + C$$

Since the two categories are the same， The quadratic term of x is eliminated, so only a one-term form is included， $\ln(\frac{p(C_1|x)}{p(C_2|x)})$ is a linear function of an x， It can be written in the form of $-(\omega^T x + \omega_0)$.

owing to:

It is thus found that$P(C_1|x)$can be expressed in the form of a sigmod function. The function image of the sigmoid is:

$$P(C_1|x) = \frac{p(C_1,x)}{p(x)} = \frac{p(C_1)p(x|C_1)}{p(C_1)p(x|C_1) + p(C_2)p(x|C_2)}$$

$$= \frac{1}{1 + e^{\ln(\frac{p(C_1|x)}{p(C_2|x)})}} = \frac{1}{1 + e^{-(\omega^T x+\omega_0)}}(\omega^T x + \omega_0)(\omega^T x + \omega_0)$$

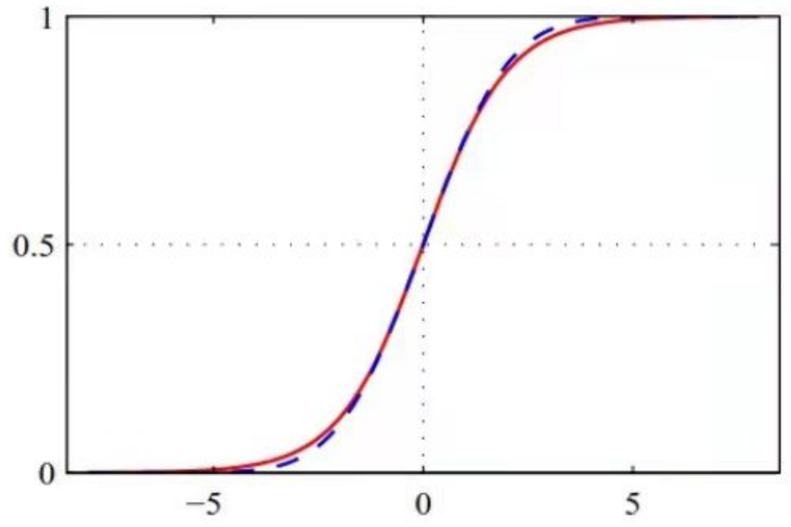


Figure 11: Sigmoid function image

The abscissa is $(\omega^T x + \omega_0)$, and the ordinate is $P(C_1|x)$.

Above, it can be intuitively believed that the parameters w and 0 can be obtained through the training set, so that most of the data x in category 1 have $\omega^T x + \omega_0 > 0$ and most of the data in category 2 have $\omega^T x + \omega_0 < 0$, which is the same model as the discriminant method of fisher (long distance between classes and small variance between classes), but the solution process is different. Which method is all right? Also test the tests.

In all the above discussions, covariance matrix for each class is the same, so the discrimination boundary of the two categories is a straight line (the point in the discrimination boundary of the two classes is $P(C_1|x) = P(C_2|x)$.If the covariance matrices are different, the discrimination boundary is not straight, this case is called quadratic discrimination.

6.3 Score function to determine portfolio

In the end, we adopt the Score function to determine our portfolio scheme, where the Score function is the built-in function, we used in solving using Python. In the results we obtained, we also use the decision coefficient to reflect the extent of the interpretation of the independent variables that we set in gold and Bitcoin. The determination coefficient reflects how much percentage of y fluctuations can be described by fluctuations of x, the percentage of variants characterizing variable Y that can be explained by the controlled independent variable X. The higher the interpretation, the higher the independent variable-induced changes as a percentage of the total change, and the more successful our portfolio is.

7 Test the Model

7.1 Sensitivity Analysis

We used the established model to obtain a series of data by changing the size of the initial amount, the range of which was from \$1,000 to \$10,000. Through Python, we finally drew a straight line. The image is as follows:

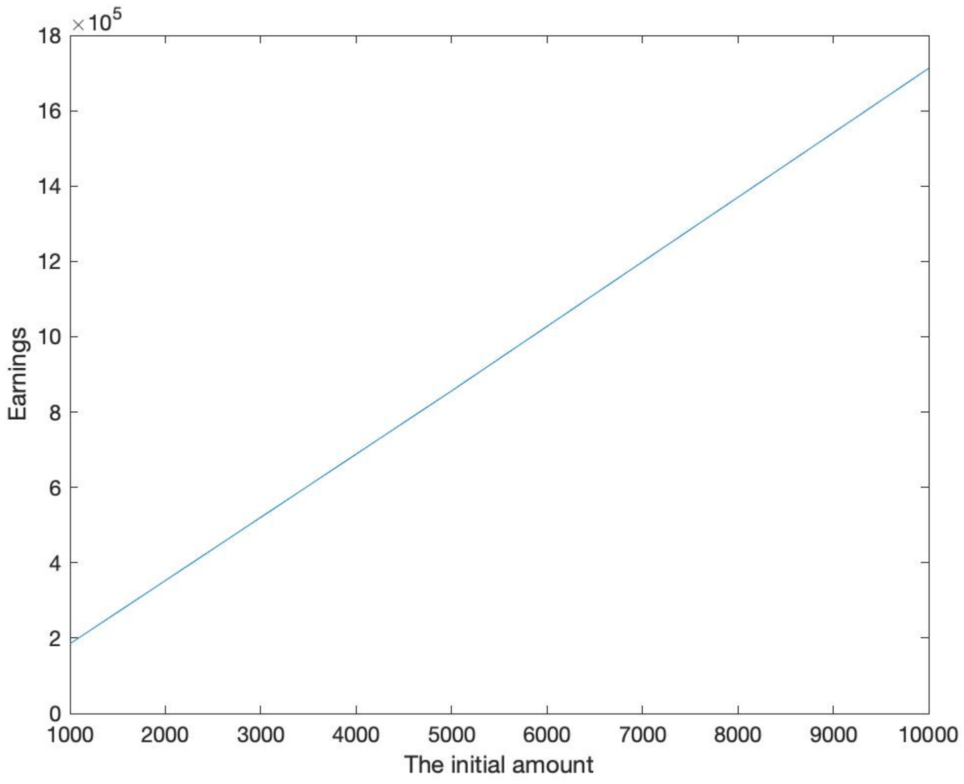


Figure 12: Sensitivity Analysis Chart

This fully shows that our initial amount has a positive relationship with the final return, which indicates that our model is relatively robust and reliable.

7.2 Strengths and Weaknesses

7.2.1 Strengths

- **High accuracy:** After our test and inspection, when we divide the pricing data of gold and Bitcoin into different periods, and then put it into the model we established for prediction, the degree of fit is still very high, which means that the model has high accuracy.
- **High generalizability:** Our model is not only limited to predictive analysis of investment strategies for gold and Bitcoin, when we use machine learning models to analyze data on other trading currencies, we can also obtain high-return investment portfolios Strategy.
- **Clear and simple:** The model contains complex concepts such as machine learning and optimization algorithms, but what we describe can be understood by everyone to fully understand the basic concepts and properties.
- **Efficient and fast:** Our model can start predictive analysis after processing the data, which is beyond the reach of many humans.
- **Objective results:** The model we have established displays all the results visually and displays them mathematically, which reduces traders' understanding of the algorithm.

7.2.2 Weaknesses

- **Data problem:** Since gold is only traded on the market opening day, we have deleted the data of Bitcoin on the opening day of the gold market. Although this is to maintain the consistency of the data, it will still lead to certain errors in the data analysis. , so that the model prediction is not accurate enough to reflect the most real pricing data.
- **Model building is simple:** Although our use of models and algorithms is concise and easy to understand, there are still some shortcomings in the use of simpler models. For this reason, we adopt the form of combined models to consolidate them.
- **Stability is not high:** In the determination of the investment portfolio, although we have carried out a sensitivity analysis on it, and there is a positive correlation trend, the obtained stability still needs to be strengthened.

Memorandum

Dear Mr./Madam Trader,

Hello!

We are honored to know that you would like us to solve your financial investment problems for you. We know that the current financial industry is developing rapidly, and investment and wealth management also have great opportunities and risks. For the relevant model you requested, our team conducted detailed research and came up with our own thinking. To help you build your model quickly, we derive the best trading strategies based on daily price data for gold and bitcoin and provide relevant explanations and analysis.

Briefly put, the model works in three steps:

1. Carry out machine learning and derive the best predictive model by processing the historical data you provide.

2. Use a linear regression model to get the prediction results we expect.

3. Calculation of portfolio benefit maximization by using price change weighting algorithm.

4. Use the KNN algorithm and the mixed model of QDA analysis to evaluate the scheme to provide proof.

We name our model the boom curve model， Our model predicts future price change trends based on previous price fluctuations of gold and Bitcoin, and based on the average of future changes, we regulate the corresponding proportional weights as follows:

$$present\_crash : present\_gold : present\_bitcoin = logistic(\delta_1 + \delta_2) : logistic(\delta_1) : logistic(\delta_2)$$

present_crash In this formula, you present_goldhold the DOLLAR after the regulation, and correspondingly, present_bitcoin it δ_1 is the controlled gold and Bitcoin holdings, and our regulation is based on $\delta_2$the estimated fluctuations of the dollar and gold.

Subsequently, because we need a certain trading commission, gold for$\alpha_1 = 0.01$, bitcoin for$\alpha_2 = 0.02$, to maximize your earnings, so we choose the minimum trading commission method, come up with the following formula:

$$\begin{aligned} present\_crash + present\_gold * thePriceOfGold + present\_bitcoin * thePriceOfBitcoin \\ = previous\_crash + previous\_gold * thePriceOfGold + previous\_bitcoin \\ * thePriceOfBitcoin - \alpha_1 * \frac{previous\_gold - present\_gold}{thePriceOfGold} - \alpha_2 \\ * \frac{previous\_bitcoin - present\_bitcoin}{thePriceOfBitcoin} \end{aligned}$$

Based on the above equation, you can calculate the change in gold dollars after your pre-trade transaction.

Of course, gold is not like Bitcoin, which can open at any time, so we recommend trading on the trading day when gold opens, so that it can be more reliable.

All our trading strategies are automatically derived by algorithms, and we will tell you how to buy and sell at that time, or we can automate them for you, all done by the computer so that they can be accurate.

Finally, to verify the accuracy of our algorithm, we show you our algorithm's final estimate of your total assets, as shown below:

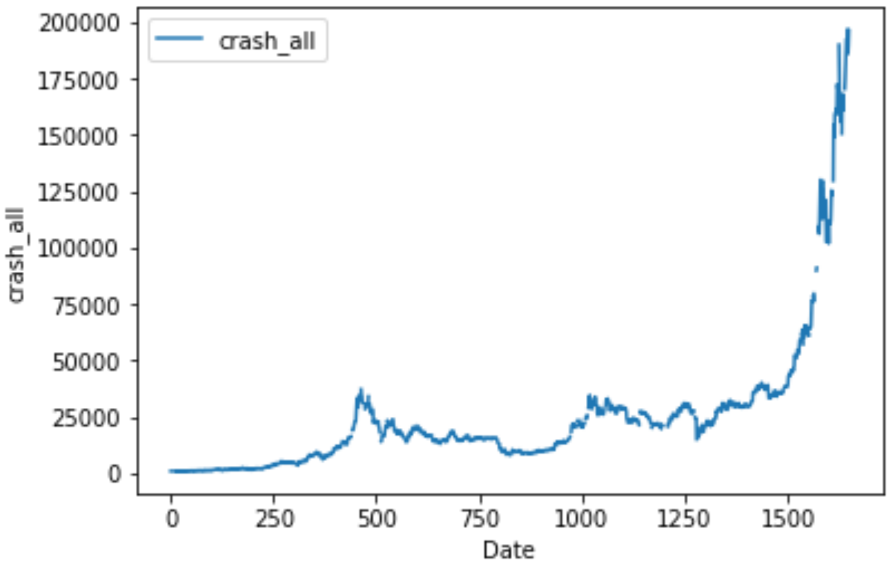
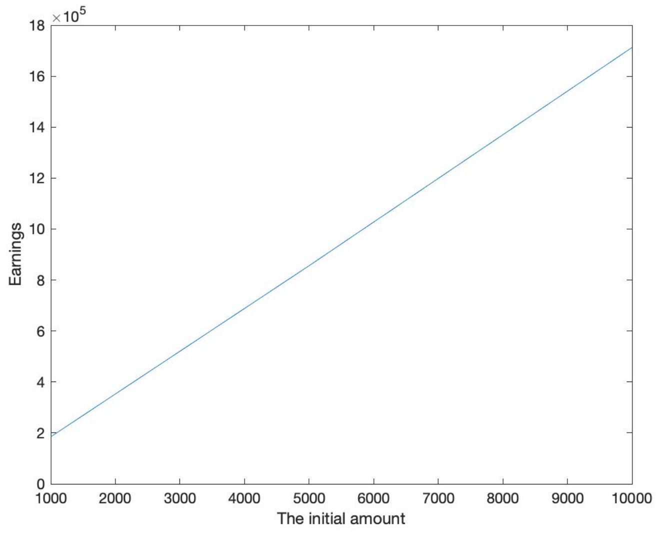


Figure 13: boom curve model

Your asset may have a slow gain in the early stages, but it doesn't matter, the algorithm automatically adjusts the strategy according to the changes in the market, although it will fluctuate at some moments, because after all, the market is not smooth sailing. However, if you stick to it, your assets will grow exponentially, reaching a terrifying value in the long run.

Moreover, after our calculations, we found that your final income is completely proportional to your initial investment, in short, the greater your investment, the greater the return, as shown in the following figure:



:

Figure 14: Sensitivity Analysis Chart

From this, our investment strategy will surely keep you safe from losing money.

We are looking forward to your reply.

Yours sincerely,
Team # 2227194

References

[1] 新智元，机器学习模型训练全流程, **https://zhuanlan.zhihu.com/p/184673895**, **2022.2.20**

[2] sxjcfrd，机器学习-线性回归总结，**https://blog.csdn.net/fengxinlinux/article/details/86556584，2022.2.22**

[3] Tavish Srivastava ，Introduction to k-Nearest Neighbors: A powerful Machine Learning Algorithm (with implementation in Python & R)，**https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/，2022.2.22**

[4] 罗罗 ，线性和二次判别分析，**https://zhuanlan.zhihu.com/p/38641216， 2022.2.22**

[5] 春卷，scikit-learn 线性回归模型的 score 函数，**https://blog.csdn.net/Rex_WUST/article/details/82621952，2022.2.22**

Appendices

Appendix 1
Introduce: Tools and software
Paper written and generated via Office 2019. Graph generated and calculation using Python 3.7.0.

Appendix 2
Introduce: Investment Strategy Model Code

```
import sklearn
from sklearn.linear_model import LinearRegression
from sklearn.neighbors import KNeighborsRegressor
from sklearn.linear_model import Ridge
from sklearn.preprocessing import PolynomialFeatures
from sklearn.pipeline import make_pipeline

import numpy as np
import pandas as pd
import math

import matplotlib.pyplot as plt
from matplotlib import style
import matplotlib as mpl

import datetime


gold_df = pd.read_csv('Python/LBMA-GOLD.csv')
bitcoin_df = pd.read_csv('Python/BCHAIN-MKPRU.csv')


combination_df = pd.merge(bitcoin_df, gold_df, on=['Date'],
how='outer')


# Suppose on day 2 the trader allocates assets as follows

crash = 500.0
gold = 250.0 / 1324.6
bitcoin = 250.0 / 609.67
holdings = [crash, gold, bitcoin]

crash_all = 1000
```

```
# i is days of transaction.

for i in range(10, 1810):

    print('The ', i, 'th has begun !')
    print()

    # combination_df_temp = combination_df.fillna(value=0,
inplace=True)

    if combination_df.iloc[i].at['USD (PM)'] == 0:
        continue

    gold_df =
pd.concat([pd.DataFrame(combination_df[:i]['Date']),

pd.DataFrame(combination_df[:i]['USD (PM)'])], axis=1,
join='outer')
    bitcoin_df =
pd.concat([pd.DataFrame(combination_df[:i]['Date']),

pd.DataFrame(combination_df[:i]['Value'])], axis=1,
join='outer')
    gold_df = gold_df.dropna()
    bitcoin_df = bitcoin_df.dropna()

    # Get moving average

    gold_mavg = gold_df['USD
(PM)'].shift(1).rolling(window=i+1).mean()
    bitcoin_mavg =
bitcoin_df['Value'].shift(1).rolling(window=i+1).mean()
    gold_df = gold_df.dropna()
    bitcoin_df = bitcoin_df.dropna()

    # earnings

    earnings_gold = gold_df['USD (PM)'] / gold_df['USD
(PM)'].shift(1) - 1
    earnings_bit = bitcoin_df['Value'] /
bitcoin_df['Value'].shift(1) - 1

    dfreg_gold = gold_df.loc[:, ['USD (PM)']]
    dfreg_gold['PCT_change'] = gold_df['USD
(PM)'].pct_change() * 10000
    for j in list(gold_df.index):
        dfreg_gold.loc[j, 'HL_PCT_gold'] =
float(dfreg_gold[['USD (PM)']].max(
        )) - float(dfreg_gold[['USD (PM)']].min()) /
dfreg_gold.loc[j, 'USD (PM)'] * 1000
    dfreg_bit = bitcoin_df.loc[:, ['Value']]
    dfreg_bit['PCT_change'] =
bitcoin_df['Value'].pct_change() * 100000
    for j in list(bitcoin_df.index):
        dfreg_bit.loc[j, 'HL_PCT_bit'] =
float(dfreg_bit[['Value']].max(
        )) - float(dfreg_bit[['Value']].min()) /
dfreg_bit.loc[j, 'Value'] * 100000
    dfreg_gold['Date'] = combination_df[['Date']]
    dfreg_bit['Date'] = combination_df[['Date']]
    # dfreg_gold.plot()
    # Drop missing value
    dfreg_gold.fillna(value=-99999, inplace=True)
    dfreg_bit.fillna(value=-99999, inplace=True)
    # The separation of data (training: testing = 9:1)
    # We want to separate 10 percent of the data to
forecast
    forecast_out_gold = int(math.ceil(0.1 *
len(dfreg_gold)))
    forecast_out_bit = int(math.ceil(0.1 * len(dfreg_bit)))
    forecast_col_gold = 'USD (PM)'
    forecast_col_bit = 'Value'
    dfreg_gold['label'] =
dfreg_gold[forecast_col_gold].shift(
        -forecast_out_gold)
    dfreg_bit['label'] =
dfreg_bit[forecast_col_bit].shift(-forecast_out_bit)
    X_gold = np.array(dfreg_gold.drop(['label', 'Date'],
1))
    X_bit = np.array(dfreg_bit.drop(['label', 'Date'], 1))
    # Scale the X so that everyone can have the same
distribution for linear regression
    X_gold = sklearn.preprocessing.scale(X_gold)
    X_bit = sklearn.preprocessing.scale(X_bit)
    # Finally We want to find Data Series of late X and
early X (train) for model generation and evaluation
    X_lately_gold = X_gold[-forecast_out_gold:]
    X_lately_bit = X_bit[-forecast_out_bit:]
    X_gold = X_gold[:-forecast_out_gold]
    X_bit = X_bit[:-forecast_out_bit]
    # Separate label and identify it as y
    y_gold = np.array(dfreg_gold['label'])
    y_gold = y_gold[:-forecast_out_gold]

    y_bit = np.array(dfreg_bit['label'])
    y_bit = y_bit[:-forecast_out_bit]
    X_gold_test = X_gold[-forecast_out_gold:]
    X_bit_test = X_bit[-forecast_out_bit:]
    y_gold_test = y_gold[-forecast_out_gold:]
    y_bit_test = y_bit[-forecast_out_bit:]
    # Linear regression
    clfreg_gold = LinearRegression(n_jobs=-1)
    clfreg_gold.fit(X_gold, y_gold)
    clfreg_bit = LinearRegression(n_jobs=-1)
    clfreg_bit.fit(X_bit, y_bit)
    # Quadratic Regression 2
    clfpoly2_gold = make_pipeline(PolynomialFeatures(2),
Ridge())
    clfpoly2_gold.fit(X_gold, y_gold)
    clfpoly2_bit = make_pipeline(PolynomialFeatures(2),
Ridge())
    clfpoly2_bit.fit(X_bit, y_bit)
    # Quadratic Regression 3
    clfpoly3_gold = make_pipeline(PolynomialFeatures(3),
Ridge())
    clfpoly3_gold.fit(X_gold, y_gold)
    clfpoly3_bit = make_pipeline(PolynomialFeatures(3),
Ridge())
    clfpoly3_bit.fit(X_bit, y_bit)
    clfknn_gold = KNeighborsRegressor(n_neighbors=2)
    clfknn_gold.fit(X_gold, y_gold)
    clfknn_bit = KNeighborsRegressor(n_neighbors=2)
    clfknn_bit.fit(X_bit, y_bit)
    # Forecast the price of the next days
    days = 15
    last_date_gold = pd.to_datetime(gold_df['Date'])
    last_unix_gold = last_date_gold
    next_unix_gold = last_unix_gold +
datetime.timedelta(days=1)
    for j in range(1, days):
        next_date_gold = next_unix_gold
        next_unix_gold += datetime.timedelta(days=1)
        dfreg_gold.loc['Date'] = [
            np.nan for _ in range(len(dfreg_gold.columns)-
1)]+[j]
    dfreg_gold = dfreg_gold.drop('Date')
    bitcoin_df = combination_df.dropna()
    bitcoin_df = bitcoin_df.drop('USD (PM)', 1)
    bitcoin_df = bitcoin_df.reset_index()
    bitcoin_df = bitcoin_df.drop('index', 1)
    dfreg_bit = bitcoin_df
    last_date_bit = pd.to_datetime(bitcoin_df['Date'])
    last_unix_bit = last_date_bit
    next_unix_bit = last_unix_bit +
datetime.timedelta(days=1)
    for j in range(1,days):
        next_date_bit = next_unix_bit
        next_unix_bit += datetime.timedelta(days=1)
        dfreg_bit.loc['Date'] = [
            np.nan for _ in range(len(dfreg_bit.columns)-
1)]+[j]
    dfreg_bit = dfreg_bit.drop('Date')
    combination_df_temp = combination_df.fillna(value=0.0)
    if combination_df_temp.iloc[i].at['USD (PM)'] == 0.0:
        continue
    # Calculate daily changes
    avarage_gold = dfreg_gold[['USD
(PM)']][len(dfreg_gold)-days:].mean()
    avarage_bit = dfreg_bit[['Value']][len(dfreg_bit)-
days:].mean()
    avarage_gold = avarage_gold['USD (PM)']
    avarage_bit = avarage_bit['Value']
    # Define the logistic function
    def logistic(z):
        return 1 / (1 + np.exp(-z))
    # Increase the weight of price changes (Logistics)
    # delta = avarage_gold + avarage_bit
    # avarage_gold = logistic(
    #     avarage_gold - combination_df.iloc[i].at['USD
(PM)'])
    # avarage_bit = logistic(avarage_bit -
combination_df.iloc[i].at['Value'])
    priceOfGold = combination_df.iloc[i].at['USD (PM)']
    priceOfBit = combination_df.iloc[i].at['Value']
    valueOfDallor = crash
    valueOfGold = gold * priceOfGold
    valueOfBit = bitcoin * priceOfBit
    priceOfGold = combination_df.iloc[i].at['USD (PM)']
    priceOfBit = combination_df.iloc[i].at['Value']
    # y2,y2 is quantity
    # present_valueOfDallor = ((priceOfGold * priceOfBit *
(valueOfDallor + valueOfGold + valueOfBit)) - (0.01 *
valueOfGold * priceOfBit + 0.02 * valueOfBit *
priceOfGold)) / (
    #     ((avarage_bit * priceOfBit + avarage_gold *
priceOfGold + 1) * priceOfGold * priceOfBit) - (avarage_bit
* priceOfBit * priceOfGold + avarage_gold * priceOfGold *
priceOfBit))
    # y2,y3 is value
    present_valueOfDallor = ((priceOfGold * priceOfBit *
(valueOfDallor + valueOfGold + valueOfBit)) - (0.01 *
valueOfGold * priceOfBit + 0.02 * valueOfBit *
priceOfGold)) / (
        ((avarage_bit + avarage_gold + 1) * priceOfGold *
priceOfBit) - (avarage_bit * priceOfBit + avarage_gold *
priceOfGold))
    # y2,y3 is quantity
    # presnet_valueOfGold = avarage_gold *
present_valueOfDallor * priceOfGold
    # present_valueOfBit = avarage_bit *
present_valueOfDallor * priceOfBit

    # y2,y3 is value
    presnet_valueOfGold = avarage_gold *
present_valueOfDallor
    present_valueOfBit = avarage_bit *
present_valueOfDallor
    avarage_gold = dfreg_gold[['USD
(PM)']][len(dfreg_gold)-days:].mean()
    avarage_bit = dfreg_bit[['Value']][len(dfreg_bit)-
days:].mean()
    avarage_gold = avarage_gold['USD (PM)']
    avarage_bit = avarage_bit['Value']
    # if crash_all > present_valueOfDallor + avarage_gold *
gold + avarage_bit * bitcoin:
    #     continue
    # Total daily value held ($)
    crash_all = present_valueOfDallor + presnet_valueOfGold
+ present_valueOfBit
    crash = present_valueOfDallor
    gold = presnet_valueOfGold /
combination_df.iloc[i].at['USD (PM)']
    bitcoin = present_valueOfBit /
combination_df.iloc[i].at['Value']
    print('You have ', crash_all, ' !')
    print()
holdings = [crash, gold, bitcoin]
print(holdings)
```