Stock Prediction and Quantitative Analysis Based on Machine Learning Method

By

zhenghan3

September 2020

Abstract

The stock market is affected by many factors which leads to drastic internal changes that are hard to predict. With the rapid development of the stock market, large quantities of data have been generated concerning stocks, which are suitable for implementing machine learning methods to analyze and learn from the data to find the internal patterns.

This paper studies the intrinsic relationship between the stocks' multiple factors and the investment value of the stocks listed in China Securities Index (CSI) 800 Index through the machine method. The investment system pipeline has been implemented including data acquirement, data preprocessing, model tuning and selection based on the XGBoost boosted tree model.

The multi-factor data from the year 2010 to the year 2020 have been selected to predict the monthly stocks' return yield. The system takes a year's data prior to the current time as the training set, separate the training set in chronological order as the development set for 5 folders cross-validation, take the stocks' multi-factor data of the current time as the test set, to predict the stocks' next month's return yield, after which a list of the stocks that worth buying each month is generated.

The system achieved total strategy returns of 200.08% with the excess returns of 122.25%, sharp ratio of 0.287, information ratio of 0.689, whereas the total returns of the benchmark CSI 300 Index were 35.38% in the backtesting from 2010 to 2020. The strategy returns far exceed the benchmark, which indicated that the system could be used as the method to auxiliary the quantitative stock selection, which can provide helpful suggestions on investment for retail investors.

Table of Contents

List of Figures

List of Tables

List of Codes

List of Abbreviations

ACC - Accuracy

API - Application Programming Interface

APT - Arbitrage Pricing Theory

AR - Autoregressive

AUC - Area Under Curve

AZFinText - Arizona Financial Text

CAPM - Capital Asset Pricing Model

CSI - China Securities Index

DNN - Deep Neural Networks

EMA - Exponential Moving Average

GBDT - Gradient Boosting Decision Tree

IC - Information Coefficient

KOSPI - Korea Composite Stock Price Index

LSTM - Long Short-Term Memory

NLP - Nature Language Processing

ROC - Receiver Operating Curve

S&P - Standard & Poor's

ST - Special Treatment

SVM - Supported Vector Machines

SWS - Shenwan Hongyuan Securities

XGBoost - eXtreme Gradient Boosting

1 Introduction

With the continuous growth of people's wealth and the improvement of financial management awareness in recent years, stock investment is getting more and more attractions. People wish to get as much profit as possible and have minimum risk at the same time in the investment progress.

Traditional manual trading is the usual investment method which relies heavily on the judgement depth analysis and can be easily affected by human emotion. With the evolution of computer technology, quantitative trading has become a newly sprouted stuff that has developed rapidly in recent years.

Quantitative trading is composed of various trading strategies which build the automated trading system relying on mathematical models(Ta et al. 2020). It implements an advanced mathematical model to replace the subjective judgement which using a large quantity of the historical data to analyze the portfolio with the higher probability of obtaining excess returns(Ge and Zhou 2020).

In the year of 1952, Markowitz took the lead in introducing mathematical tools into financial research which contributed to the birth of the modern financial economics(Rubinstein 2002). The multi-factor model based on the arbitrage theory of capital asset pricing(Ross 2013) is one of the most famous mathematical models at the moment. The idea is using the exposure of factors in stocks to predict the next time price change(Ge and Zhou 2020).

From the perspective of machine learning, the multi-factor model is corresponding to the multiple linear regression(Fang 2018). Therefore, the machine learning model can be used to predict stock returns. The model takes the historical financial indicators as the features to do the regression on the stock returns, then uses the factors of the current time to predict the return yield in the future.

Currently, there are few pieces of research on quantitative analysis in the A-shares market. The project has set up an investment system pipeline based on the multi-factor and machine learning model. The system uses the data of the whole year before the current time as the training set and implements the optimized XGBoost model to predict the stock return yield of the next month at the

current time, after which selects the top 50 stocks that most likely to rise in the next month for equal weight investment. The automated trading system was also preliminarily explored in the later stage of the project after the recommended stocks have been obtained.

The project provides a reference for the research gap of the quantitative investment in the A-shares market to a certain extent.

1.1 Aims and Objectives

The project aims to implement the index enhancement strategy based on multi-factor and machine learning model to obtain higher excess returns than the benchmark.

The objectives of the project are as follows,

i. Retrieve multiple factors from different aspects of the stocks on the A-shares market from different quantitative financial platforms.
ii. Apply different data preprocessing methods such as winsorization, missing data imputation, neutralization and standardization to clean the data.
iii. Implement the model parameter tuning process on the Extreme Gradient Boosting (XGBoost) method to find the optimal model on the specified date with the highest Area Under Curve (AUC) on the development set.
iv. Use the optimal model found each month to select a basket of stocks with higher predicted return yield.
v. Compare the selected stocks' returns with the CSI 300 Index benchmark returns based on historical data to analyze the strategy benefit.
vi. Preliminary exploration of the automated trading system.

1.2 Program Activity Diagram

The program is executed in the following order,

Data acquirement → Data preprocessing → Model tuning → Model selection → Model analysis → Automated trading

The program activity diagram is shown in Figure 1.1.

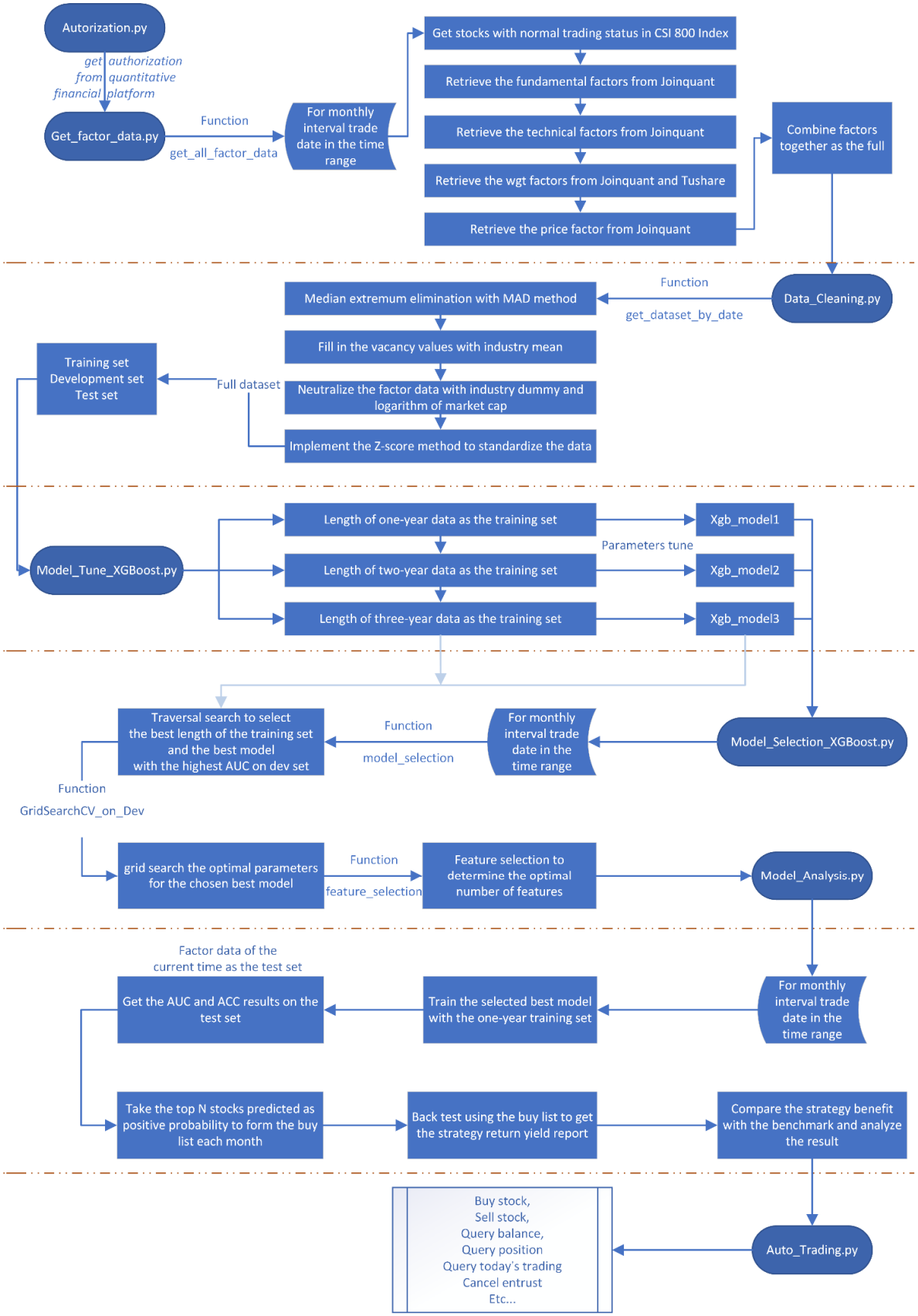

Figure 1.1 Program activity diagram

2 Literature Review

Concerning the research of the stock quantitative analysis, studies from different views can be found. Generally speaking, it can be divided into two main aspects, analysis regarding selection and analysis regarding timing.

Selection, means choosing the stocks may rise better in the future within the basket of the stocks.

Sorensen et al. implemented the decision tree model based on CART methodology to construct several screening strategies and evaluated their performance. The screens constrained on the evaluation criteria such as liquidity, investment style, fundamental earning performance and so on to predict the performance of the stock returns with the conclusion that the decision trees based on single-factor or multi-factor were all produced significantly better sharp ratios than the simple ranking model(Sorensen et al. 2000). This paper has been referenced by many financial securities. The decision-tree-based method has been widely used by the industry until now given its explanation on the factor selection is very comprehensive.

Lin and Chen have applied the random forest model with different dataset partitioning methods to train the model and predict the stock returns in CSI300, CSI500 and all the stocks within the A-shares market. The model achieved an average accuracy of 57% on the test set, the excess returns of the stocks selected from CSI 300 were 2.5% ~ 7.5%. They found that the momentum reverse factors have been regarded as the most important features among almost all the time by the classifier during the training process(Lin and Chen 2017a).

Ge and Zhou selected top-100-stock in the A-shares market and implemented factor calculation, feature preprocessing, factor analysis through the machine learning method based on XGBoost model which is trained through a daily-frequency window. The result showed that the selection based on XGBoost model has a significant improvement over the traditional equally weighted multi-factor selection method(Ge and Zhou 2020). Liu and Zhang selected eight categories of factors from 2010 to 2019 to constitute factor library and implemented the XGBoost model to select the important

factors to construct the portfolio, the returns of the portfolio turned to 24.81% higher than the CSI 300 benchmark(Liu and Zhang 2020).

Fan and Palaniswami used the Supported Vector Machines (SVM) with the classification approach to “beat the market”. They have taken the financial accounting and price data of the stocks on the Australia Stock Exchange to identify the stocks that were likely to have exceptional returns outperformed the market. They applied the equally weighted portfolio selection strategy through the SVM model. The portfolio achieved 208% of the profit over the five years, 71% more than the benchmark. They have also set an interception as the probability measurement to take the top 25% stocks in the market(Fan and Palaniswami 2001).

In the 1990s, the neural network was applied to improve the stock selection. Ghosen and Bengio have designed a specially structured neural network to let the different networks of the different stocks to share part of the middle layers to carry out the multi-task learning in the Canadian stock market. The network achieved more than 14% over the benchmark during the yearly returns(Ghosn and Bengio 1997).

Timing, refers to infer the optimal buying or selling moment through the time series modelling. Deep learning and neural network have been widely explored in this area.

Based on the Ghosn and Bengio’s work, Saad et al. compared and analysed the recurrent, time delay and the probabilistic neural network model separately which validated that the neural network can be used for predicting the term trend for the stocks(Saad et al. 1998).

Chong et al. have applied three unsupervised feature extraction methods - the restricted Boltzmann machine, autoencoder, and the principal component analysis and used the high-frequency intraday stocks’ returns information (five minutes data) as the input data on the Deep Neural Networks (DNN) model to predict the future market behaviour on the Korean Korea Composite Stock Price Index (KOSPI) stock market. An attempt to provide an objective and comprehensive assessment of the deep learning model for predicting the stock market trend has been tried. They concluded that the

networks could extract extras information from the residuals of the autoregressive (AR) model. The estimation of the covariance was also improved while the predictive network is applied to the covariance-based analysis(Chong et al. 2017).

Fischer and Krauss deployed the Long Short-Term Memory (LSTM) networks for predicting the out-of-sample price movements for the constituent stocks within the Standard & Poor's (S&P) 500 Index from 1992 to 2015. Through the comparison with the memory-free classification methods, i.e., logistic regression classifier, decision tree classifier, deep neural network, they found that the LTSM model has a better performance over the three benchmark models(Fischer and Krauss 2018). Kai et al. have also implemented the LSTM in China stock market. The historical data has been transformed into 30-day-long sequences with 10 factors included, the 3-day earning rate has been set as the label. The model improved the predicted accuracy from 14.3% to 27.2% compared to the random prediction method(Chen et al. 2015).

Zhang et al. proposed a state frequency memory recurrent network that can be used to figure the patterns of the stock price in multi-frequency trading to make long and short term predictions over time. They found the new proposed model can enable more accurate predictions over the different period that outperformed the AR and LSTM model(Zhang et al. 2017).

Other than the literature regarding selection and timing, some other explorations have also been tried.

Schumaker and Chen have implemented the synthesis of the statistical, financial and linguistic to create the Arizona Financial Text system (AZFinText System). The system conducted the textual representation and statistical machine learning analysis on the financial news that grouped by similar industry. The research demonstrated that the stocks predicted by the system with the simulated trading returns of 8.50%, whereas the returns for the S&P 500 were 5.62% at the same time. The paper has explored the new possibility of the combination of the quantitative analysis and the Nature Language Processing (NLP)(Schumaker and Chen 2009a).

Nguyen et al. built the model to predict the stock movement using the sentiment from social media(Nguyen et al. 2015). Schumaker and Chen examined the machine learning approach using different textual representations based on breaking financial news(Schumaker and Chen 2009b). These papers show that predicting the stock movement through sentiment analysis is the popular research direction.

2.1 Development of the Multi-factor Model

In the year 1966, Sharpe et al. proposed the theory of the Capital Asset Pricing Model (CAPM) which explained the systematic risks β of financial assets, that became the theoretical basis for measuring the investment performance(Fama and French 2004).

In the year 1976, Ross proposed the Arbitrage Pricing Theory (APT) based on the CAPM theory(Ross 2013). The APT assumes that the returns on assets are linearly related to a group of common factors, that is

$$R_i = E(R_i) + \sum_{j=1}^{N} b_{ij} F_j + \varepsilon \qquad \text{(Equation 1)}$$

Where,

$R_i = returns\ of\ the\ asset\ i$;

$E(R_i) = expected\ returns\ of\ the\ asset\ i$;

$N = number\ of\ common\ factors$;

$F_j = deviation\ from\ its\ expected\ value\ for\ common\ factor\ j\ on\ asset\ i$;

$b_{ij} = the\ sensitivity\ of\ the\ asset\ i's\ returns\ to\ common\ factor\ j$;

$\varepsilon = random\ perturbation\ term\ of\ the\ asset$;

In the year 1992, Fama and French found there were stock excess returns that can not be explained by the β in CAPM, thus proposed the theory of Fama–French 3–factor model. The model found that the expected returns of the stock are not only related to the systematic risk of the market but also related to the factors of firm size and book-to-market equity(Fama and French 1993). In the following year of 2015, Fama and French found that profitability and investment patterns should also be

included except the three factors mentioned in the 3–factor model(Fama and French 2015).

The multi-factor model was developed based on the Fama–French factor model which considered that the stock returns could be decomposed into the linear combination of some factor returns. The model tried to merge more factors to form a more complex model. The factors can be fundamental factors composed of the financial information, technical factors composed of price information and so on except the factors from the Fama–French factor model. The multi-factor model can be written as,

$$R_s = \sum_{i=0}^{N} \omega_i F_i + \sum_{i=0}^{N} \sum_{j=0}^{N} \omega_{ij} F_i F_j + \varepsilon \qquad \text{(Equation 2)}$$

Where,

$R_s = returns\ of\ the\ stock$;

$N = number\ of\ factors$;

$F_i = the\ i-th\ factor$;

$\omega_i = the\ weight\ of\ the\ i-th\ factor$;

$\omega_{ij} = the\ cross\ weight\ of\ the\ i-th\ factor\ and\ j-th\ factor$;

$\varepsilon = bias$;

The returns of the stock are affected by multiple factors, which are the linear combination of factors and intersections between factors.

The multi-factor model considers many external factors of stocks. Even if the stock market environment is always changing, there are always some factors that can describe the current market situation no matter how the market fluctuates, which makes the multi-factor model relatively more stable in describing the trend of stocks.

2.2 Principle of the XGBoost Model

From the previous literature review, the decision-tree-based machine learning models not only have a good performance in terms of the stock prediction but also are able to output the features that are important during the selection process. It shows the excellent interpretability that can guide the

following analysis process.

In the year 2016, Chen and Guestrin proposed the XGBoost boosted tree model(Chen and Guestrin 2016). It is based on the improvement of the Gradient Boosting Decision Tree (GBDT) algorithm which has already shown great power in the industry compared to the traditional boosted tree algorithms.

The idea of the XGBoost is to iterate through the weak classifiers to generate the new tree. The objective function is gradually improved by finding the optimized structure and leaf fraction of the CART tree.

The XGBoost constantly splits the features to grow a tree. A new function is learned by adding one tree at a time to fit the residual of the last prediction. The objective function can be written as,

$$L^{t} = \sum_{i=1}^{n} l(y_i, \widehat{y}_i^{\,t-1} + f_t(x_i)) + \Omega(f_t) + \text{constant} \qquad \text{(Equation 3)}$$

Where,

$L^{t} = the\ objective\ function;$

$n = number\ of\ the\ examples;$

$\widehat{y}_i^{\,t} = the\ prediction\ of\ the\ i-th\ instance\ at\ the\ t-th\ iteration;$

$f_t = the\ independent\ trees\ in\ the\ space\ of\ the\ regression\ trees\ at\ the\ t-th\ iteration;$

$x_i = the\ i-th\ input\ of\ the\ examples;$

$l = the\ loss\ function\ measures\ the\ different\ between\ prediciton\ \widehat{y}_i\ and\ the\ target\ y_i;$

$\Omega = regularization\ term;$

Equation 3 can be optimized in the general setting by applying the Second-order approximation.

$$L^{t} \simeq \sum_{i}^{n} [l(y_i, \widehat{y}_i^{\,t-1}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) + \text{constant} \qquad \text{(Equation 4)}$$

Where,

$g_i = \partial_{\hat{y}_i^{t-1}} l(y_i, \hat{y}_i^{t-1});$

$h_i = \partial^2_{\hat{y}_i^{t-1}} l(y_i, \hat{y}_i^{t-1});$

Represent the first and the second gradient statistics on the loss function.

Rewrite the Equation 4 by expanding the regularization term and merging the formula. Equation 5 can be obtained,

$$L^t \simeq \sum_{j=1}^{T} [(\sum_{i \in I_j} g_i)\omega_j + \frac{1}{2}(\sum_{i \in I_j} h_i + \lambda)\omega^2{}_j] + \gamma T \qquad \text{(Equation 5)}$$

The optimal weight ω_j^* of leaf j can be calculated by deriving Equation 5,

$$\omega_j^* = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda} \qquad \text{(Equation 6)}$$

Substitute the ω_j^* into Equation 5, the corresponding optimal value can be calculated,

$$L^t(q) = -\frac{1}{2}\sum_{j=1}^{T} \frac{(\sum_{i \in I_j} g_i)^2}{\sum_{i \in I_j} h_i + \lambda} + \gamma T \qquad \text{(Equation 7)}$$

The quality of the tree structure q can be measured by the scoring function shown in Equation 7. The smaller the score, the better the structure, which can be used to select the best segmentation point by comparing the value changed of Equation 7 after the splitting. The greedy algorithm that iteratively adds the branches to the tree is applied during the splitting process.

$$L_{split} = \frac{1}{2}[\frac{(\sum_{i \in I_L} g_i)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{(\sum_{i \in I_R} g_i)^2}{\sum_{i \in I_R} h_i + \lambda} - \frac{(\sum_{i \in I_i} g_i)^2}{\sum_{i \in I_i} h_i + \lambda}] - \Upsilon \qquad \text{(Equation 8)}$$

Equation 8 is used for evaluating the candidates for splitting.

XGBoost algorithm has the advantages of fast, parallel implementation. It introduces the regularization term L_1 and L_2 to control the complexity of the model while supporting the column sampling at the same time(Liu and Zhang 2020) which is suitable to be selected as the project model.

3 Data Preparation

This chapter describes the process for data acquisition, including factor selection, preprocessing and acquiring.

3.1 Factor Selection

The multi-factor model theory holds that the factors affecting the stock returns mainly come from the following three aspects(darcylike 2018).

i. Company Level

The company factors reflect the company microstructure to some extent while showing the production and the operational activities at the same time. Most factors can be obtained through the company's financial reports and indicators, which reflect the company's profit, operation, liability, cash flow and growth, etc. They not only depict the scale and operational capability of a company but also show the financial changes of a company under the market. Common factors such as market capitalization, price-to-book ratio, the return of equity, etc.

ii. Market Performance Level

Market performance factors mainly reflect the stock trading process. Price change, trading volume change, transaction frequency change are the sources. They describe the financial technology indicators such as stock’s risk, momentum, capital flow, etc. Market performance factors also describe the short-term fluctuations and trends of stocks. Common market performance factors such as turnover ratio, MACD, RSI, etc.

iii. External Environment Level

External environmental factors are mainly influenced by policies, laws, macroeconomic environments, changes in social customs and technological development. These factors not only

depict the current economic environment and situation but also depict the trend of future economic development. External environmental factors are essential in long-term prediction. However, they are rarely used in the current quantitative analysis because of the difficulty of acquisition and analysis. Common external environmental factors such as macroeconomic variables, patents, etc.

Furthermore, the factors can be subdivided into the following categories (Table 3.1).

Table 3.1 Factor subdivided category

Category	Description
Scale	factors related to enterprise-scale, such as market capitalisation, circulating market capitalisation, etc.
Value	factors related to common valuation indicators, such as price-to-earnings ratio, price-to-book ratio, price-to-cash flow ratio, etc.
Growth	factors related to enterprise performance growth, such as year-on-year growth rate of total revenue, year-on-year growth rate of operation profit, quarter-on-quarter growth rate of net profit to shareholders, etc.
Quality	reflect the features of the company's capital structure and profit quality, such as the return of equity, equity ratio, net profit margin and so on.
Technical	Mainly some classical technical analysis indicators such as moving average convergence divergence, relative strength index, psychological line, etc.
Style	factors related to the market-style change such as momentum, residual volatility, liquidity, etc.
Others	Factors that have not been classified into previous categories.

The factor selection should cover the above categories as much as possible to make sure comprehensively describe the different aspects of the stock. The project selected dozens of factors through the combination of the previous work from Fang(Fang 2018), Lin and Chen(Lin and Chen 2017a), while also including some part of the existing factors from JoinQuant factor library(JoinQuant 2020b). The selected factors validity has been verified to make sure the factors' rank Information Coefficient (IC) with the stock return yield greater than the correlation threshold.

46 factors have been included to form the factor library (Table 3.2).

Table 3.2 The list of factors been selected into the factor library

Category	Abbreviation	Name	Description
Scale	MC	market capitalisation	closing price * total share capital
	CMC	circulating market capitalisation	closing price * total share capital in circulation
Emotion	TR	turnover ratio	total sales / average shareholders' equity
Value	PE	price-to-earnings ratio	market value per share / earnings per share
	PB	price-to-book ratio	market price per share / book value per share
	PCF	price-to-cash flow ratio	market price per share / cash flow per share
Quality	ROE	return of equity	net income / average shareholders' equity
	DER	equity ratio	total shareholder equity / total assets
	NPTTR	net profit margin	net profit / total revenue
	GPM	gross profit margin	gross profit / total revenue
	ETTR	expense to total revenue	expense / total revenue
	OETTR	operating expense to total revenue	operating expense / total revenue
	GETTR	ga exense to total revenue	management expense / total revenue
	OPTP	operating profit to profit	operating profit / profit
	APTP	adjust profit to profit	adjusted profit / net profit
	GSASTR	good sale and service to revenue	cash received from goods sell and services provided / revenue
Growth	ITRYOY	inc total revenu year on year	year-on-year growth rate of total revenue
	IRYOY	inc revenue year on year	year-on-year growth rate of revenue
	IOPYOY	inc operation profit year on year	year-on-year growth rate of operation profit
	INPYOY	inc net profit year on year	year-on-year growth rate of net profit
	INPA	inc net profit annual	quarter-on-quarter growth rate of net profit

Category	Abbreviation	Name	Description
	INPTSA	inc net profit to shareholders annual	quarter-on-quarter growth rate of net profit to shareholders
Emotion	VOL20	average turnover ratio on 20 days	
Risk	sharpe_ratio_20	average sharp ratio on 20 days	(Rp - Rf) / Sigma, where, Rp is the annualized rate of return, Rf is the risk-free interest rate and Sigma is the return volatility (standard deviation)
Momentum	BIAS20	average deviation rate on 20 days	
	ROC20	price rate of change on 20 days	
Technical	MFI14	money flow index	
Style	momentum	momentum	the difference between relatively strong stocks and weak stocks in the past two years
	residual_volatility	residual volatility	the difference in yield caused by the volatility after stripping off the market risk
	liquidity	liquidity	the difference in return rate caused by the different relative trading activity of stocks
	earnings_yield	profitability	the income difference caused by profit income
	leverage	leverage	the income difference between high-leverage stocks and low-leverage stocks
Technical	ATR	average true range	n=14
	MTM	momentum line	n=20
	MACD	moving average convergence divergence	short=12, long=26, mid=9
	RSI	relative strength index	n=6
	PSY	psychological line	n=12
	CYR	relative market strength index	n=13, m=5
Momentum reversal	wgt_return_1m	wgt_return_1m	refer to the explanation of section 3.1.1.4
	wgt_return_3m	wgt_return_3m	
	wgt_return_6m	wgt_return_6m	

Category	Abbreviation	Name	Description
	wgt_return_12m	wgt_return_12m	
	exp_wgt_return_1m	exp_wgt_return_1m	
	exp_wgt_return_3m	exp_wgt_return_3m	
	exp_wgt_return_6m	exp_wgt_return_6m	
	exp_wgt_return_12m	exp_wgt_return_12m	

3.1.1 Examples of Some Special Factors

3.1.1.1 Financial Factors

Most of the factors belonging to the category of scale, value, quality and growth are financial factors.

Financial data are divided into the time scope of the quarterly and reporting period. In the single quarter financial data, the income statement and cash flow statement are counted according to the single quarter. Therefore, special attention should be paid that the valuation table is updated daily, and the balance table is updated according to the reporting period. The market data adopts the pre-reinstatement data by default(JoinQuant 2020c).

3.1.1.2 MACD

MACD (Moving Average Convergence Divergence) was proposed by Appel in 1985(Appel 1985). It is a trend-following momentum indicator that shows the relationship between two moving averages of a security's price(HAYES 2020b). The indicator has been widely used by the investors given it comprehensively displays aspects such as cycle and trend, etc.

The calculation of the MACD subtracts the 26-period Exponential Moving Average (EMA) from the 12-period EMA(HAYES 2020a). The formula is written as,

$$DIF = EMA(close,12) - EMA(close,26) \quad \text{(Equation 9)}$$

$$DEA = EMA(DIF,9) \quad \text{(Equation 10)}$$

$$MACD = DIF - DEA \quad \text{(Equation 11)}$$

The EMA represents the exponential moving average which can be calculated by,

$$EMA_{Today}(close, N) = \frac{2}{N+1} \times close + (1 - \frac{2}{N+1}) \times EMA_{Yesterday}(close, N) \quad \text{(Equation 12)}$$

An initial value will be set at the beginning which will gradually converge after multiple recursions.

3.1.1.3 Residual Volatility

Style factors were all obtained from JoinQuant factor library showing the style of the market(JoinQuant 2020b).

Take the residual volatility factor as an example. Residual volatility describes the difference in yield caused by the volatility after stripping off the market risk, it is calculated by,

$$residual\_volatility = 0.74 \times daily\_standard\_deviation + 0.16 \times cumulative\_range + 0.10 \times historical\_sigma$$

(Equation 13)

Where,

$daily\ standard\ deviation$

$= exponentially\ weighted\ standard\ deviation\ of\ excess\ returns\ in\ the\ past\ 252\ days;$

$cumulative\ range$

$= the\ difference\ between\ the\ maximum\ and\ minimum\ monthly\ return\ yield\ in\ the\ past\ 12\ months;$

historical_sigma

$= standard\ deviation\ of\ the\ past\ 252\ trading\ days\ of\ the\ return\ residual\ term\ in\ beta;$

3.1.1.4 Momentum Reversal

The momentum reversal factor was the idea proposed by Huatai securities. Lin and Chen found that the priority of the momentum reversal factors has always been very high during the output of the feature importance list through the random forest algorithm(Lin and Chen 2017a), thus been selected into the factor library.

The momentum reversal factor can be calculated by,

$$wgt\_return\_Nm = \frac{\sum_{i=1}^{N_{day}} turnover\_ratio_{daily\_i} \times return_{daily\_i}}{N_{day}} \quad \text{(Equation 14)}$$

Where,

$wgt\_return\_Nm = N\ months\ momentum\ reversal\ factor;$

$i = The\ i-th\ trading\ day\ before\ the\ current\ date;$

$N_{day} = total\ days\ within\ N\ months;$

$turnover\_ratio_{daily\_i} = daily\ turnover\ ratio\ of\ i;$

$return_{daily\_i} = daily\ return\ yield\ of\ i;$

The exponential momentum reversal factor is calculated by,

$$\exp\_wgt\_return\_Nm = \frac{\sum_{i=1}^{N_{day}} turnover\_ratio_{daily\_i} \times e^{-\frac{i}{N\times 4}} \times return_{daily\_i}}{N_{day}} \quad \text{(Equation 15)}$$

N was taken for 1, 3, 6 and 12 months during the calculation respectively.

3.1.2 Factor Correlation Test

The situation exists where there is a strong correlation between factors. Some of the correlation factors need to be removed if the situation happens; otherwise, factor redundancy will be caused.

The factor correlation heatmap can be obtained through the “DataFrame.corr()” function provided by “pandas” library(Pandas Documentation 2020). The heatmap (Figure 3.1) was generated based on the data of 31 December 2015.

From the heatmap, the pairwise correlations between the factors were basically below 0.2 except the momentum reversal factors, which represented that the factor correlation test basically passed. Time inclusion relation was the reason for the high correlations between the momentum reversal factors.

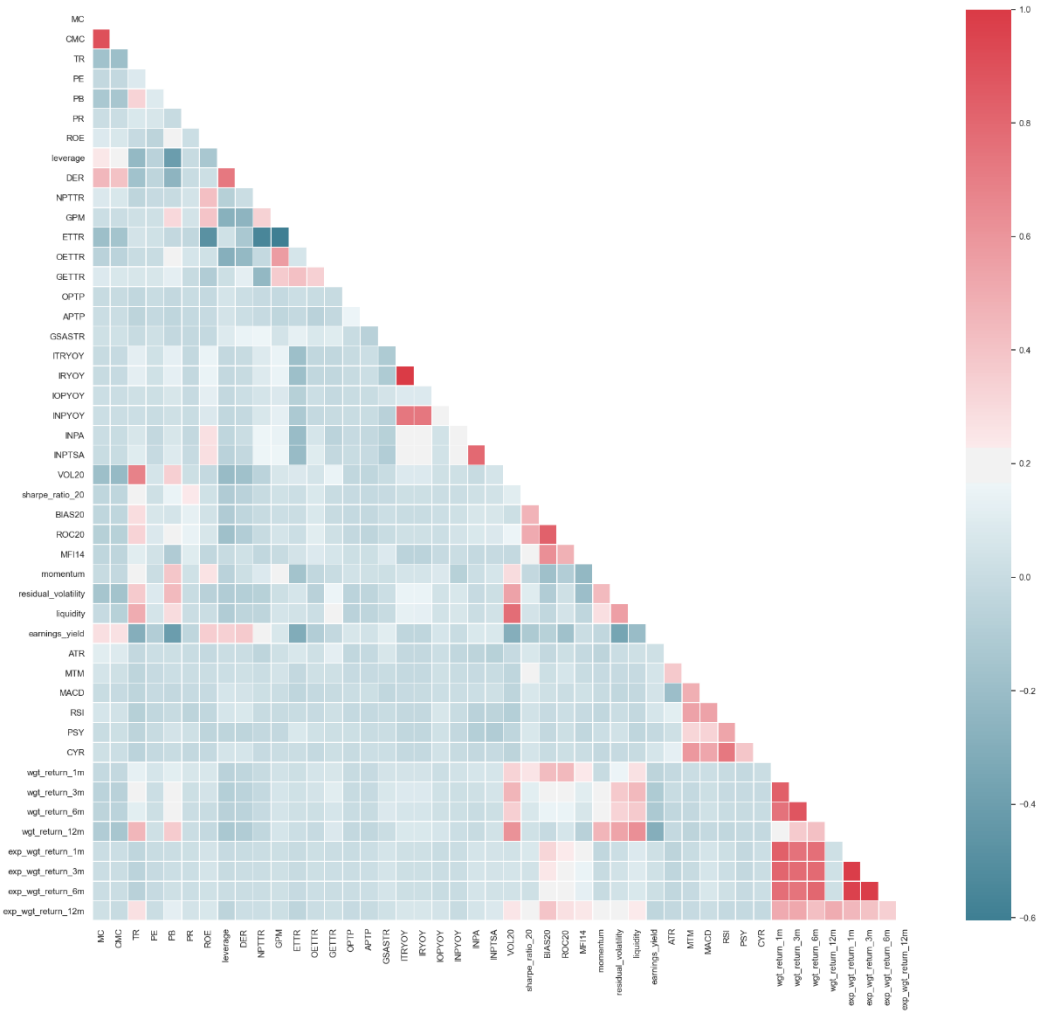

Figure 3.1 Factor correlation heatmap

3.2 Factor Preprocessing

The initial data obtained from different data sources will have the problems of noisy, missing values, inconsistent dimensions, etc. The existing of such “dirty data” will directly affect the effectiveness of the model. Therefore, data preprocessing before the modelling is the essential step to ensure the data regularization. Preprocessing usually consists of winsorization, missing data imputation, neutralization and standardization. The data comparison below is based on the date of 31 December 2015.

3.2.1 Winsorization

Winsorization is the transformation method by replacing the extreme values outside the limit with the upper and lower limits to reduce the effect of possibly spurious outliers(Wilcox 2005).

The extreme values may affect the analysis result, especially when doing the regression. Thus action needs to be taken to reduce the influence. The usual winsorization methods including the method based on the mean and variance, the MAD-Median method and the boxplot method(Wilcox 2011). Compared with the former, the latter two are less affected by the outliers and their processing effect is more robust. Generally speaking, the MAD-Median method is more widely used in engineering practice which has been selected as the project method to deal with extreme values.

The formula of the MAD-median method is,

$$X_i^{'} = \begin{cases} D_M + nD_{MAD}, & if\ X_i > D_M + nD_{MAD} \\ D_M - nD_{MAD}, & if\ X_i < D_M + nD_{MAD} \\ X_i, & if\ D_M + nD_{MAD} < X_i < D_M + nD_{MAD} \end{cases} \quad \text{(Equation 16)}$$

Where,

$X_i' = i - th\ element\ of\ the\ processed\ new\ series\ X'$;

$X_i = i - th\ element\ of\ the\ original\ series\ X$;

$D_M = the\ median\ of\ the\ factor\ series\ X\ to\ be\ processed$;

$D_{MAD} = the\ median\ of\ the\ absolute\ deviation\ of\ each\ factor\ from\ the\ D_M, that\ is\ |X_i - D_M|$;

$n = deviation\ range, the\ value\ is\ 5\ in\ this\ project$;

The stocks “601299” and “601766” have been winsorized as in table 3.3.

Table 3.3 Comparison of the factors (part of) before and after the winsorization

code	MC	CMC	TR	PE	PB	PR
Before						
601299.XSHG	1.68E+11	1391.324	1.7524	29.6082	3.5536	25.492
000559.XSHE	3.43E+10	341.8822	2.5912	46.8005	9.4839	16.7259
002233.XSHE	1.02E+10	86.6767	2.1755	18.469	2.4971	-46.7385
600482.XSHG	1.31E+10	128.2541	4.9911	91.0714	6.3584	104.868
601766.XSHG	1.77E+11	1336.488	1.8514	31.6047	4.501	78.4605
After						
601299.XSHG	6.71E+10	538.6507	1.7524	29.6082	3.5536	25.492
000559.XSHE	3.43E+10	341.8822	2.5912	46.8005	9.4839	16.7259
002233.XSHE	1.02E+10	86.6767	2.1755	18.469	2.4971	-46.7385
600482.XSHG	1.31E+10	128.2541	4.9911	91.0714	6.3584	104.868
601766.XSHG	6.71E+10	538.6507	1.8514	31.6047	4.501	78.4605

3.2.2 Missing Data Imputation

It is important to use all the available data and not discard the data with missing values, which puts forward the concept of missing data imputation(Jerez et al. 2010). The missing data in the dataset can reduce the fitting of the model due to the relationship or the correlation between variables not identified correctly, which may lead to the wrong prediction. Missing data of the financial factors are the common problem due to the untimely or incomplete releasement of the financial report of the stocks.

The ways to deal with the missing values including elimination directly or numerical interpolation. The data can be eliminated directly if there is less missing data given the less information loss; otherwise, it is essential to manipulate appropriate values for missing value interpolation. Usual methods consist of filling with mean, industry mean and industry median, etc. It can also be filled with the estimated values by K-nearest neighbour or other algorithms.

The project adopts the commonly used industry mean filling method. The idea is each stock has been classified into the corresponding industry by Shenwan First-class Industry Standard which was published by Shenwan Hongyuan Securities (SWS) research. The vacancy factors of one stock will be filled in with the mean value of all the stocks within the same industry. If the stocks are not classified into any industry, the rest of them will be considered as the same industry.

The missing data imputation, in which the vacancy values have been filled with the industry mean, is shown in table 3.4.

Table 3.4 Comparison of the factors (part of) before and after the missing data imputation

code	GPM	ETTR	OETTR	GETTR	OPTP	APTP	GSASTR
Before							
000166.XSHE	Nan	60.445	Nan	26.21	55.78	0.980835	Nan
000728.XSHE	Nan	60.445	Nan	26.21	31.83	1.000172	Nan
000686.XSHE	Nan	60.445	Nan	26.21	33.47	0.992935	Nan
000063.XSHE	35.32	100.08	11.83	2.75	20.78	0.538114	114.45
601099.XSHG	Nan	60.445	Nan	26.21	27.87	0.95901	Nan
After							
000166.XSHE	28.8777	60.445	4.744262	26.21	55.78	0.980835	121.3402

code	GPM	ETTR	OETTR	GETTR	OPTP	APTP	GSASTR
000728.XSHE	28.8777	60.445	4.744262	26.21	31.83	1.000172	121.3402
000686.XSHE	28.8777	60.445	4.744262	26.21	33.47	0.992935	121.3402
000063.XSHE	35.32	100.08	11.83	2.75	20.78	0.538114	114.45
601099.XSHG	28.8777	60.445	4.744262	26.21	27.87	0.95901	121.3402

3.2.3 Neutralization

The selected stocks may have a bias due to the influence of some specified factors. For example, the price-to-book ratio has a high correlation with the market capitalization, which leads to the selection result concentrated at specified market capitalization. That is, due to the influence of some specified factors, the selection result will have a redundant preference. The neutralization method has been proposed to solve this kind of problem by eliminating the influence of the unwanted factors to make the selection more dispersed.

Market capitalization and industry are the usual influential factors needed to be neutralized. The main idea is to use the influential factors as the independent variables to do the regression on the original factor and extract the residual as the new factor after subtraction. The neutralized factor is strictly linearly independent with influential factors after the process. The neutralization can be calculated by,

$$Factor_i = \beta_M \times \ln(MC) + \sum_{j=1}^{n} \beta_j \times Industry_{j,i} + \varepsilon_i \qquad \text{(Equation 17)}$$

Where,

$Factor_i = the\ i-th\ original\ factor;$

$MC = the\ value\ of\ the\ market\ capitalization;$

$Industry_{j,i} = industry\ j\ dummy\ matrix\ of\ \ factor\ i;$

$\beta = the\ regression\ coefficient\ of\ responding\ influencing\ factors\ to\ factor\ i;$

$\varepsilon_i = the\ neutralized\ new\ factor;$

The regression was implemented by the python library “statsmodels”(Statsmodels Documentation 2020) in the project. The factors are compared (table 3.5) before and after the neutralization.

Table 3.5 Comparison of the factors (part of) before and after the neutralization

code	MC	CMC	TR	PE	PB	PR
Before						
601010.XSHG	1.26E+10	126.1814	2.3534	28.4388	3.1963	-56.4023
601608.XSHG	2.35E+10	64.6555	0.0471	64.576	2.9633	-33.8191
600380.XSHG	1.31E+10	130.6231	1.4687	40.9534	3.0301	34.2362
002025.XSHE	6.74E+09	67.405	2.2727	32.2797	3.8102	275.3115
002179.XSHE	1.36E+10	130.0193	2.6711	36.0138	4.6754	-41.263
After						
601010.XSHG	-1.2E+09	-10.7308	-0.92798	-15.2489	-0.2744	-46.1975
601608.XSHG	-6.2E+09	-163.313	-3.26499	4.242331	-1.44752	-33.0932
600380.XSHG	-2.8E+08	13.48837	-1.28575	-15.456	-2.82633	27.4907
002025.XSHE	8.17E+09	62.14912	-1.24319	-28.1334	-0.66142	271.8472
002179.XSHE	-2.6E+09	-4.77866	-0.79271	-23.0145	-0.20046	-50.3853

3.2.4 Standardization

Standardization has a series of meanings in statistics. It has the effect of eliminating the influence of the dimension. The processed data is transformed in a fixed range, which makes the data more concentrated, or enables data on the different scale be able to compare(Noy-Meir et al. 1975).

Z-score is the commonly used method which measured in terms of standard deviations from the mean. It is calculated by,

$$X_i' = \frac{X_i - X_{mean}}{\sigma} \quad \text{(Equation 18)}$$

Where,

$X_i' = i - th\ element\ of\ the\ processed\ new\ series\ X'$;

$X_i = i - th\ element\ of\ the\ original\ series\ X$;

$X_{mean} = the\ mean\ of\ the\ factor\ series\ X\ to\ be\ processed$;

$\sigma = the\ standard\ deviation\ of\ the\ factor\ series\ X\ to\ be\ processed$;

The factors are compared (table 3.6) before and after the standardization.

Table 3.6 Comparison of the factors (part of) before and after the standardization

code	MC	CMC	TR	PE	PB	PR
Before						
601010.XSHG	-1.2E+09	-10.7308	-0.92798	-15.2489	-0.2744	-46.1975
601608.XSHG	-6.2E+09	-163.313	-3.26499	4.242331	-1.44752	-33.0932

600380.XSHG	-2.8E+08	13.48837	-1.28575	-15.456	-2.82633	27.4907
002025.XSHE	8.17E+09	62.14912	-1.24319	-28.1334	-0.66142	271.8472
002179.XSHE	-2.6E+09	-4.77866	-0.79271	-23.0145	-0.20046	-50.3853
After						
601010.XSHG	-0.20434	-0.15159	-0.4476	-0.3258	-0.11999	-0.36514
601608.XSHG	-1.06958	-2.30699	-1.57482	0.09064	-0.63295	-0.26157
600380.XSHG	-0.04883	0.190539	-0.62016	-0.33023	-1.23587	0.217283
002025.XSHE	1.405808	0.87793	-0.59964	-0.60109	-0.28922	2.148649
002179.XSHE	-0.45513	-0.0675	-0.38235	-0.49172	-0.08765	-0.39824

3.2.5 The Sequence of the Preprocessing Methods

The different preprocessing sequence may lead to a huge difference in the preprocessed data.

Winsorization and missing data imputation usually be implemented first. Which one should be done first, however, depends on the missing amount of the data. Missing data imputation is recommended to implement first if the missing amount is large; otherwise, there is no difference. Standardization usually carries out at last.

Given the explanation above, the sequence should be winsorization, missing data imputation, neutralization and standardization.

3.3 Data and Time Scope

The CSI 800 Index is a capitalization-weighted stock market index which consists of all the constituents of CSI 300 Index and CSI 500 Index. It is designed to reflect the overall performance of the large, mid and small cap stocks traded on the Shanghai Stock Exchange and the Shenzhen Stock Exchange of the A-shares market. The index is compiled by the China Securities Index Company, Ltd(China Securities Index Company 2020).

The constituents of the CSI 800 Index have been selected as the selection pool. The stocks listed less than 90 days, the stocks have been delisted, the stocks suspended and the Special Treatment (ST) stocks, have been removed on the specified trading date to ensure that the selected stocks can be traded normally on that day.

The factor data of the constituents within the CSI 800 Index of each trading date has been sourced from JoinQuant(JoinQuant 2020b) and Tushare(Tushare 2020) quantitative financial platform. JoinQuant and Tushare are available to all kinds of financial investors and researchers. The platforms contain the data of stocks, funds, futures, bonds, foreign exchange, and industry big data, providing all kinds of financial investment and researchers with applicable data and tools.

The data from 10 January 2010 to 10 June 2020 with the time interval of 20 trading days (approximately 1 month in real life) have been acquired as the dataset. The date list containing trading days with the time interval of 20 trading days (approximately 12 months*10 years=120 trading days) has also been generated.

The starting time is 2010 because the CSI 800 Index data before 2007 can not be obtained while the model wishes to take the data of the previous three years of the current time as the training set. The ending time is 10, June 2020, because the project starts in June, and the last next month's return yield can be obtained is July.

3.4 Handling of the Label

Concerning the selection of the Y label (prediction result), the next month’s return yield of the current time is the usual choice. With reference to relevant works of literature (Lin and Chen 2017a; Fang 2018), the processing of the Y label was, in the cross-section period of each trading date, took the next month’s yield of top 30% stocks as positive (Y=1), the last 30% stocks as negative (Y=0) and discarded the data in the middle.

3.5 Dataset Display

Table 3.7 shows partial test set based on the date of 31 December 2015. “pchg” is the next month return yield. “label” column is the classification label calculated based on the pchg column in descending order. There are only 446 stocks in the test set while the selection pool has nearly 800 stocks; the reason for that is 40% of the stocks in the middle have been discarded.

Table 3.7 Test set base on the date of 31 December 2015

Index	code	MC	CMC		exp_wgt_return_3m	exp_wgt_return_6m	exp_wgt_return_12m	pchg	label
1	000983.XSHE	-1.6877	-0.2224		0.0862	-0.0685	-0.2283	0.1755	1
2	600383.XSHG	1.0160	1.5785		-0.5283	-0.4507	-0.0126	0.1733	1
3	000933.XSHE	0.4853	0.5119		2.2293	2.2442	2.3617	0.1313	1
…				…	…				
444	002308.XSHE	-0.3375	0.4255		1.7487	1.5482	2.3431	-0.4696	0
445	601777.XSHG	-0.9428	-0.8104		1.8100	1.7735	0.6744	-0.4769	0
446	300257.XSHE	-1.2202	0.0724		1.5052	0.9697	0.3836	-0.4910	0

4 Methodology

This chapter mainly describes the model parameter tuning regarding the different lengths of the training sets, and model tuning process. The selection of the feature number has also been elaborated at the end of the chapter.

The aim of the chapter is, for each monthly interval trading date in date list, select the optimal model to predict the constituent stocks' next month's return yield.

4.1 Time Length of the Training Set

The system takes the factors of the current time as the test set, takes the length of certain-year data before the current time as the training set.

Concerning the selection of the time length, different choices have been adopted, and year is usually selected as the time span. Fang selected the data of the previous two years as the training set(Fang 2018). Lin and Chen chose the length of one-year data(Lin and Chen 2017a).

Training sets of different lengths did have great differences in prediction accuracy through the setting up of the simple XGBoost model.

It was decided to select the length of 1-year, 2-year and 3-year data separately as the training set

through the referring to the previous literature. In the next section, three corresponding optimal models have been tuned separately concerning the three different training sets.

Take the 1-year data training set as an example, the current date of the test set is 31 December 2015. The training set has 12 sample trading dates to form the training set (Table 4.1). The final training date has 8631 pieces of the data which is the sum of all the trading dates' stocks.

Table 4.1 Sampling trading dates of the 31 December 2015

Trading date	Number of stocks	Trading date	Number of stocks
2015-12-03	736	2015-06-08	709
2015-11-05	722	2015-05-11	740
2015-10-08	709	2015-04-10	732
2015-09-01	694	2015-03-12	750
2015-08-04	690	2015-02-05	754
2015-07-07	634	2015-01-08	761

The 2-year and 3-year data follow the same principle, the 2-year training set has 24 sample trading dates while the 3-year training set has 36 sample trading dates.

4.2 Parameter Tuning

The project selected the XGBoost as the training model given the statement from section 2.2.

There are many parameters in XGBoost algorithm. It is an essential step to optimize the parameters of the model in order to improve the generalization ability of the model.

4.2.1 Background

i. GridSearchCV and RandomizedSearchCV

The GridSearchCV is the method provided by Scikit-learn library, which can implement tedious grid search tasks. At the end of the searching process, the grid_search.best_params_ shows the best combination of the hyperparameters. Similarly, the grid_search.best_estimator_ and the grid_search.best_score display the configuration of the model estimator and the optimal model's score(Scikit-learn Documentation 2020a). Table 4.2 shows some important hyperparameters of

GridSearchCV.

Table 4.2 Important hyperparameters of GridSearchCV

Parameters	Description
estimator	model to be tuned, wrapped as the scikit-learn estimator interface.
param_grid	lists of parameter settings to try as values.
scoring	scoring function to evaluate the predictions.
cv	determines the cross-validation splitting strategy.

If the search space is large, Randomized Search is preferable. RandomizedSearchCV is a method similar to GridSearchCV. It assesses a certain number of random combinations at each iteration in place of searching all combinations of hyperparameters. RandomizedSearchCV will also return the three parameters after the searching process. Compared with GridSearchCV, RandomizedSearchCV have the advantages of controlling the computing resource and elapsed time by defining the number of iterations(Scikit-learn Documentation 2020b). Table 4.3 shows some important hyperparameters of RandomizedSearchCV which are similar to GridSearchCV.

Table 4.3 Important hyperparameters of RandomizedSearchCV

Parameters	Description
estimator	model to be tuned, wrapped as the scikit-learn estimator interface.
param_distributions	lists of parameter settings to try as values.
n_iter	the number of parameter settings that are sampled.
scoring	scoring function to evaluate the predictions.
cv	determines the cross-validation splitting strategy.

ii. The Development Set and the Cross-validation Method

Development set is the dataset used to tune the hyperparameters of the model. It is usually split from the training set. The evaluation and the selection of the optimal model and hyperparameters can be processed based on execution.

Cross-validation, as one of the validation techniques for saving the training data(Marsland 2015), divides the training data into subsets that complement one another. Each model is trained using a different combination of subsets, and the leftover subsets are used to evaluate the model performance. After the iteration of the models, the fruit of the cross-validation is the average of the scores computed in the loop.

iii. AUC Score

ROC (Receiver Operating Curve) is the curve displaying the relationship of the confusion matrix at all possible thresholds for 2-classification model(Rosset 2004). The Area Under the Curve (AUC) is the area under the ROC curve. It shows the possibility to choose an element correctly classified into the specified type. It also demonstrated the trade-off between false positive and true positive rates. On this case, the area of the AUC is used as a calibration to measure the classifier.

AUC is a better measurement than ACC (Accuracy) given that the ACC ignores probability estimations of classification class labels' preference (Ling et al. 2003).

4.2.2 Parameter Tuning Process

XGBoost model has many hyperparameters that influencing the effect of the model, the common practice is to keep other parameters unchanged while adjusting the certain parameters(Vecmanis 2019).

For the time of the different lengths training sets, the corresponding optimal models are generated by the tuning process, respectively. To simplify the problem, the project used the date of 24 January 2017's data as the training set. The following shows the one-year training set parameter tuning process and directly given the optimal parameters of the two-year and three-year directly at the end of the section.

In the parameter tuning process, 5-folder cross-validation has been used as the development set strategy and AUC score as been chosen as the scoring function.

The parameter tuning process of the XGBoost can generally be divided into the following three steps,

i. Choose the learning rate. In general, the learning rate is set to 0.1. For the different problems, the ideal learning rate usually fluctuates between 0.05 and 0.3. The learning rate is corresponding to the ideal number of trees.

ii. For the given learning rate, optimize the specific parameters separately (max_depth, min_child_weight, gamma, subsample, colsample_bytree, etc.).

iii. Tune the regularization terms (alpha, lambda). The parameters can reduce the complexity of the model and improve the performance of the model.

The tuning process according to the three steps, will be demonstrated in the following of the section.

First, determine the optimal learning rate, grid-search with the learning rate between 0.05 and 0.3 has been conducted due to the uncertainty. It can be seen that the optimal learning rate at around 0.05 (Figure 4.1).

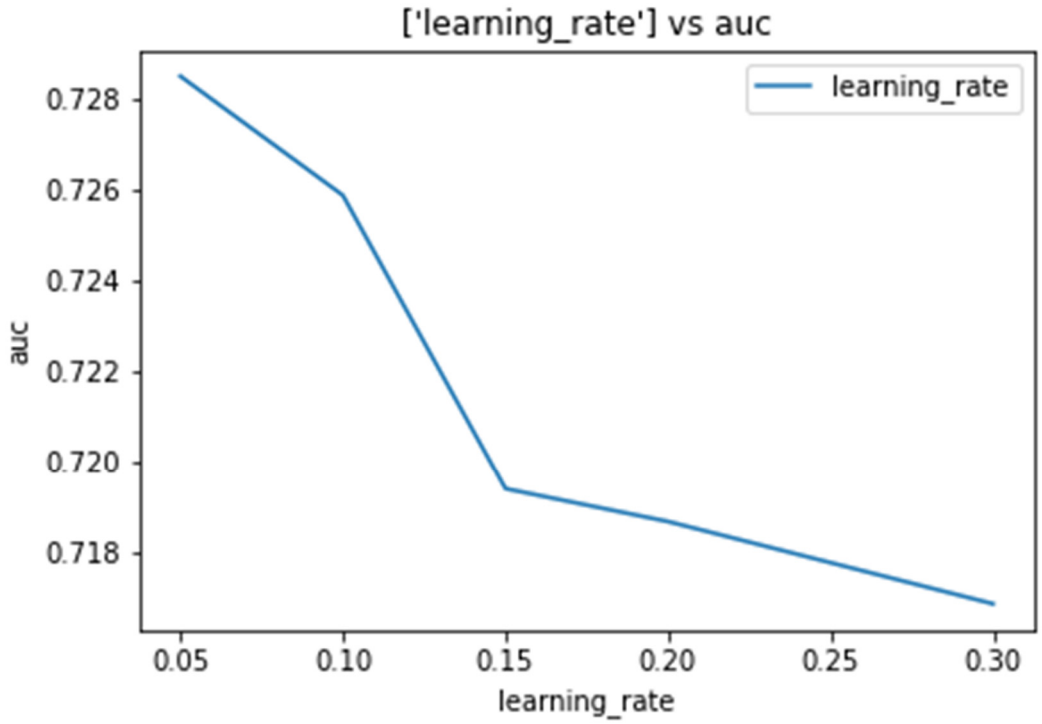

Figure 4.1 Rough search for the learning rate

Next, narrow down the search space. The optimal learning rate was stable at 0.04 at last (Figure 4.2).

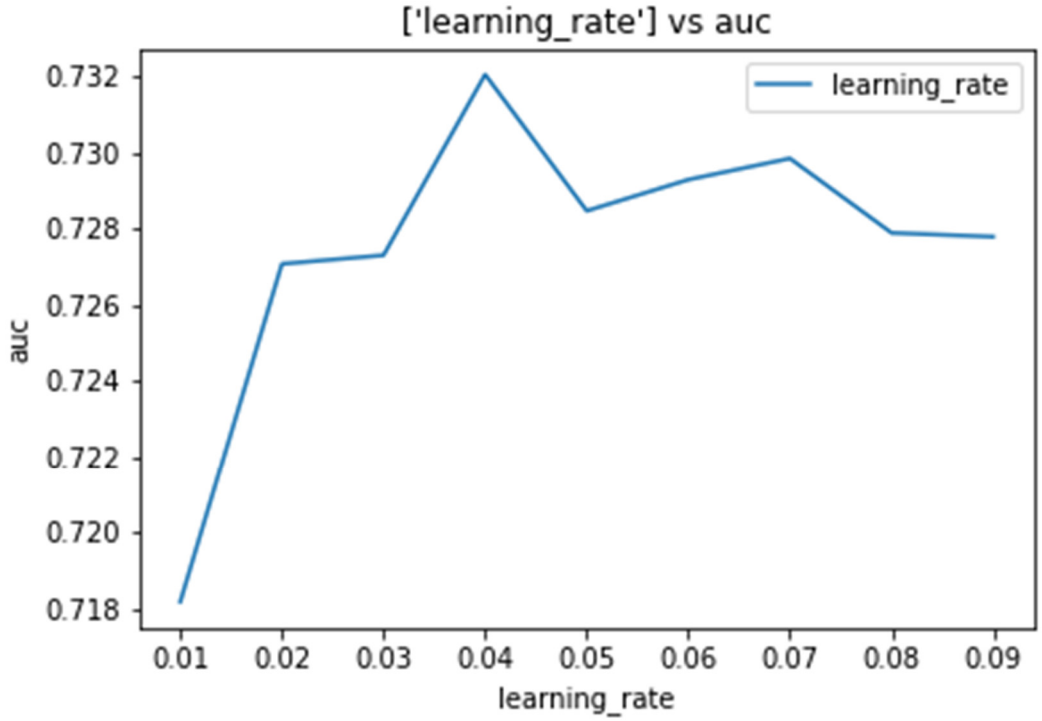

Figure 4.2 Fine search for the learning rate

Second, determine the number of estimators (n_estimators) for the given learning rate. The n_estimators parameter specifies the number of sequential trees that are made to correct for prior trees(XGBoost Documentation 2020b). Through the cross-validation method and the early-stopping strategy, the optimal number of estimators can be obtained. Figure 4.3 shows the AUC curve about the training set (orange) and the validation set (blue). After the number of the estimators reached 200, the training set AUC was greater than 95% and the validation set AUC almost remained the same. Therefore, the optimal n_estimators was set to 200.

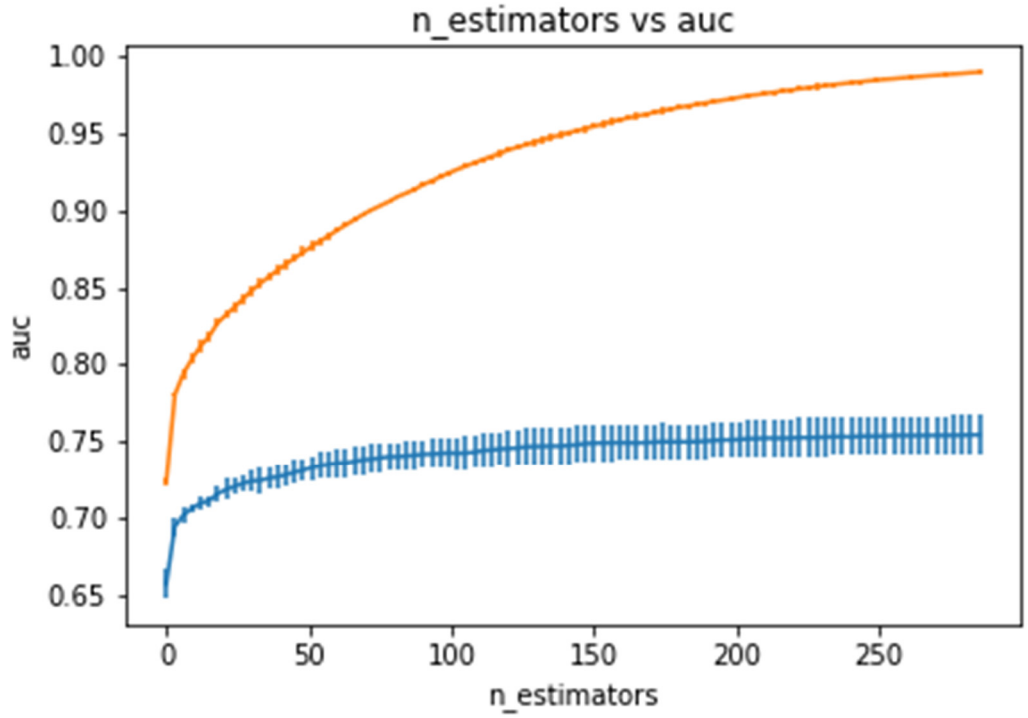

Figure 4.3 AUC curve of the n_estimators

After determining the optimal learning rate and estimator number, it was the time to tune the important parameters in turn.

The max_depth parameter determines the depth that each estimator is permitted to build a tree and the min_child_weight represents the minimum sum of instance weight needed in a child(XGBoost Documentation 2020b). These two parameters usually are tuned together.

Implement the rough search for the max_depth and the min_child_weight (Figure 4.4).

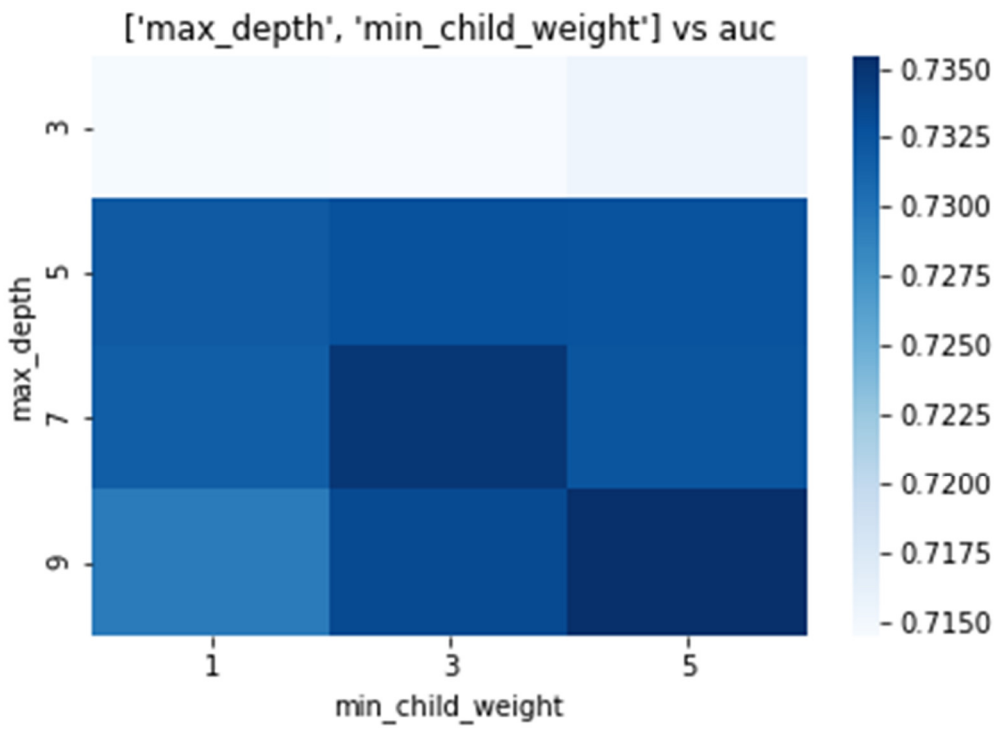

Figure 4.4 Rough search for the max_depth and the min_child_weight

Make the fine search near (7, 3). The final optimal pair was (8, 4) (Figure 4.5).

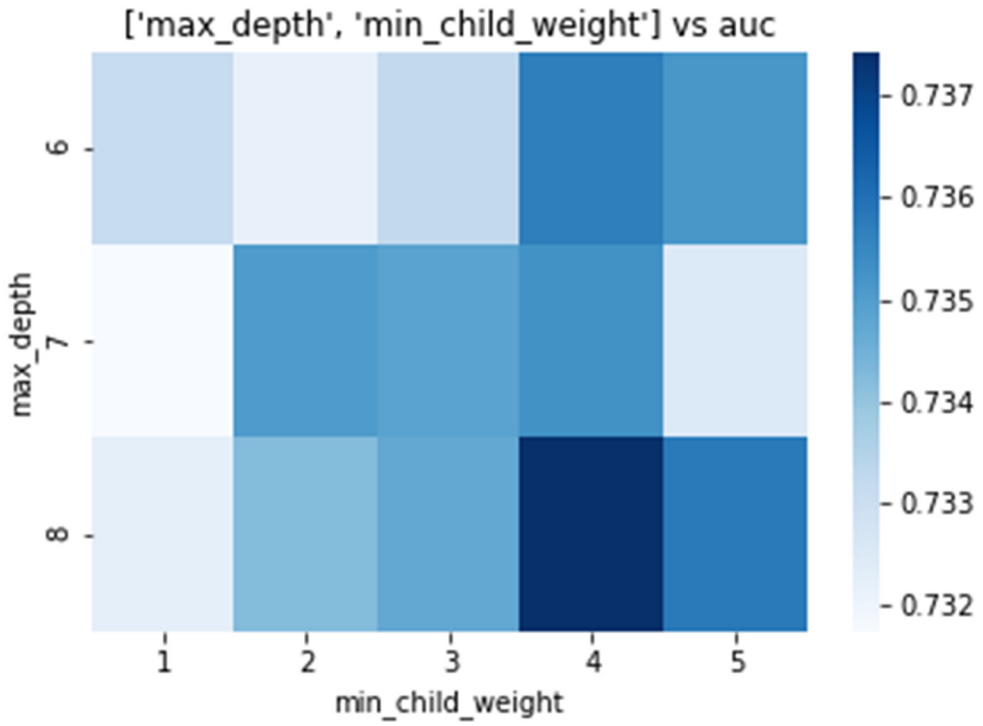

Figure 4.5 Fine search for the max_depth and the min_child_weight

Continue to determine the optimal value of gamma. “gamma” represents the minimum loss reduction required to make a further partition on a leaf node of the tree. Gamma determines the conservative degree of the algorithm (XGBoost Documentation 2020b).

Conduct the rough search for the gamma (Figure 4.6). The gamma did not influence the performance given that the AUC curve has always remained the same. Therefore, set the gamma to default value 0.

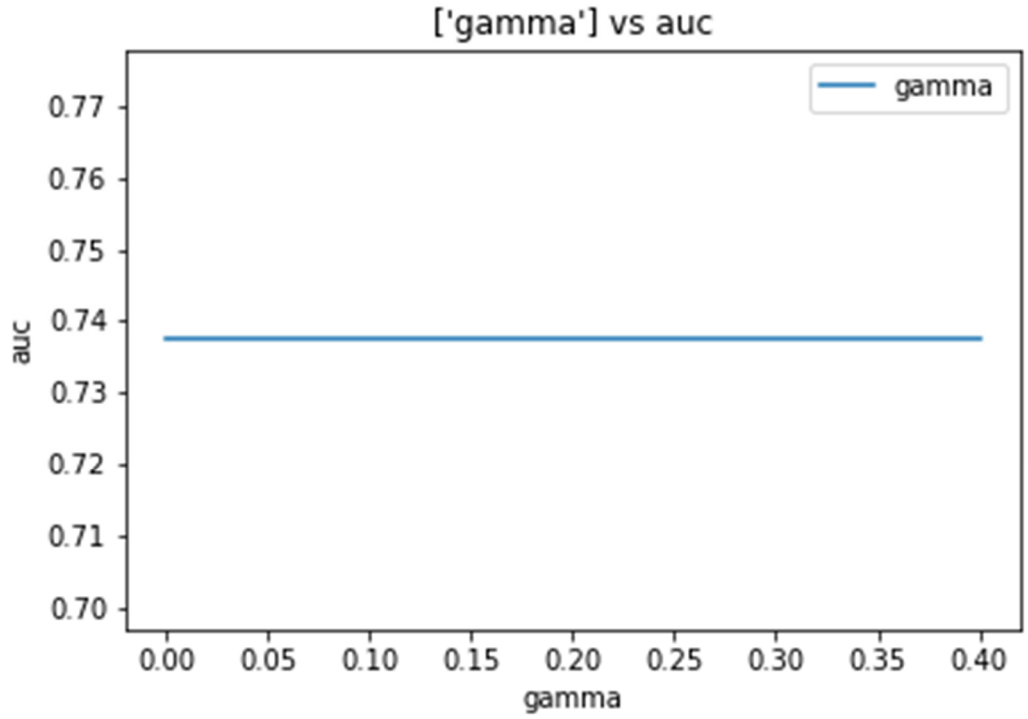

Figure 4.6 AUC curve of the gamma

Tune the parameter subsample and colsample_bytree. “subsample” is the subsample ratio of the training instances. “colsample_bytree” is the parameter for subsampling of columns. Rough search for the two (Figure 4.7).

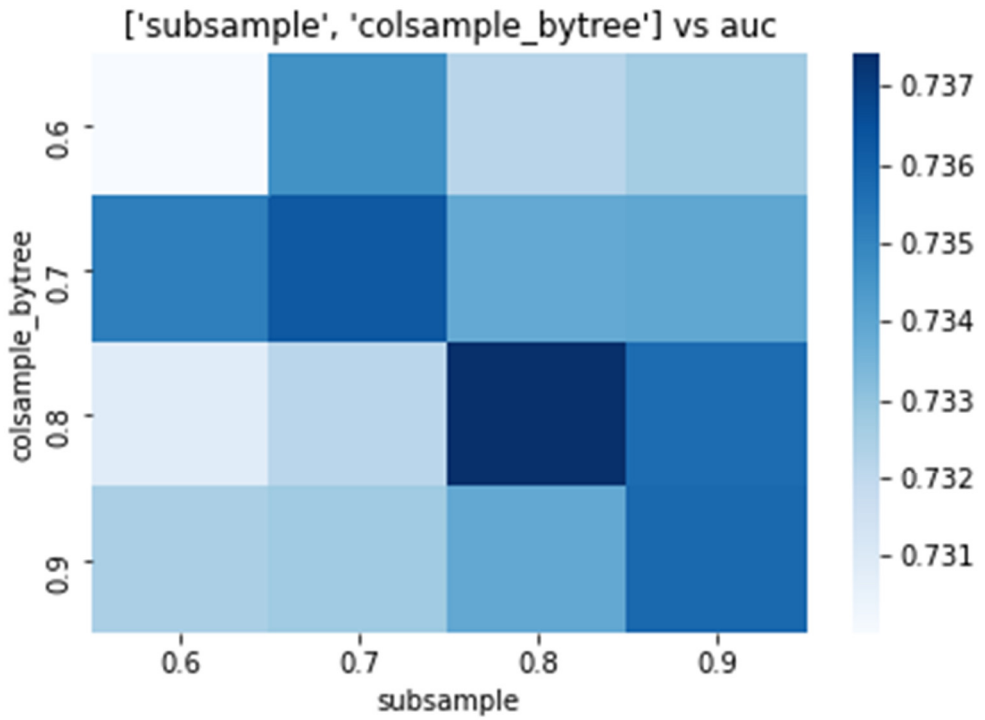

Figure 4.7 Rough search for the subsample and colsample_bytree

Implement the fine search near (0.8, 0.8). The final optimal pair was (0.8, 0.8) (Figure 4.5).

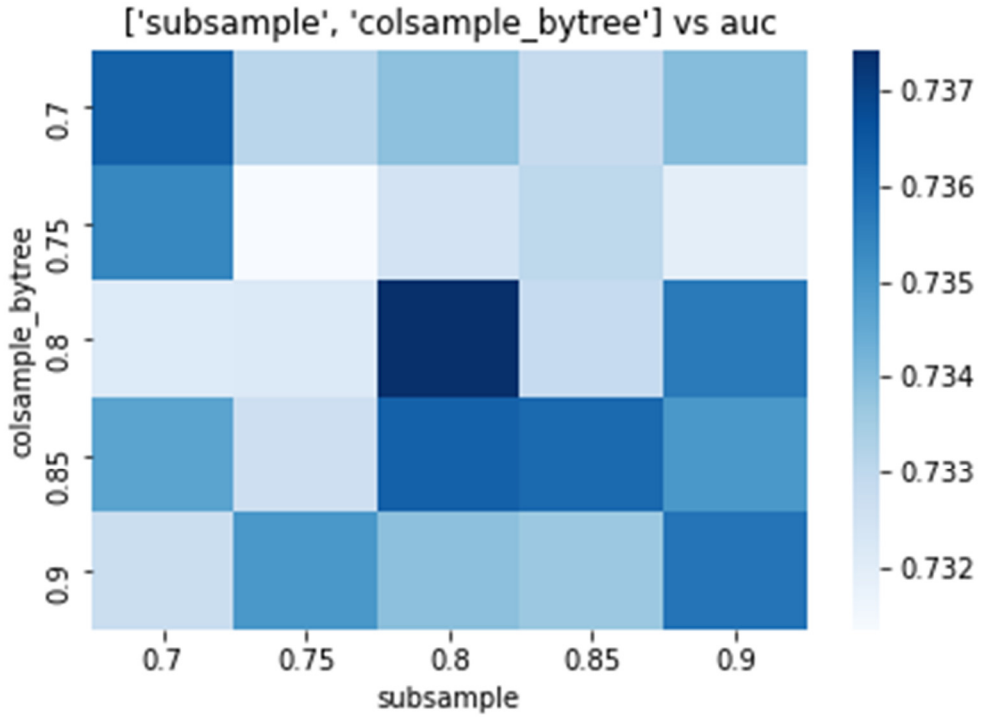

Figure 4.8 Fine search for the subsample and colsample_bytree

Finally, adjust the regularization terms reg_alpha and reg_lamda. “reg_alpha” is the L1 regularization term on weights while reg_lamda is the L2 regularization term(XGBoost Documentation 2020b).

First, take the rough search for alpha (Figure 4.9).

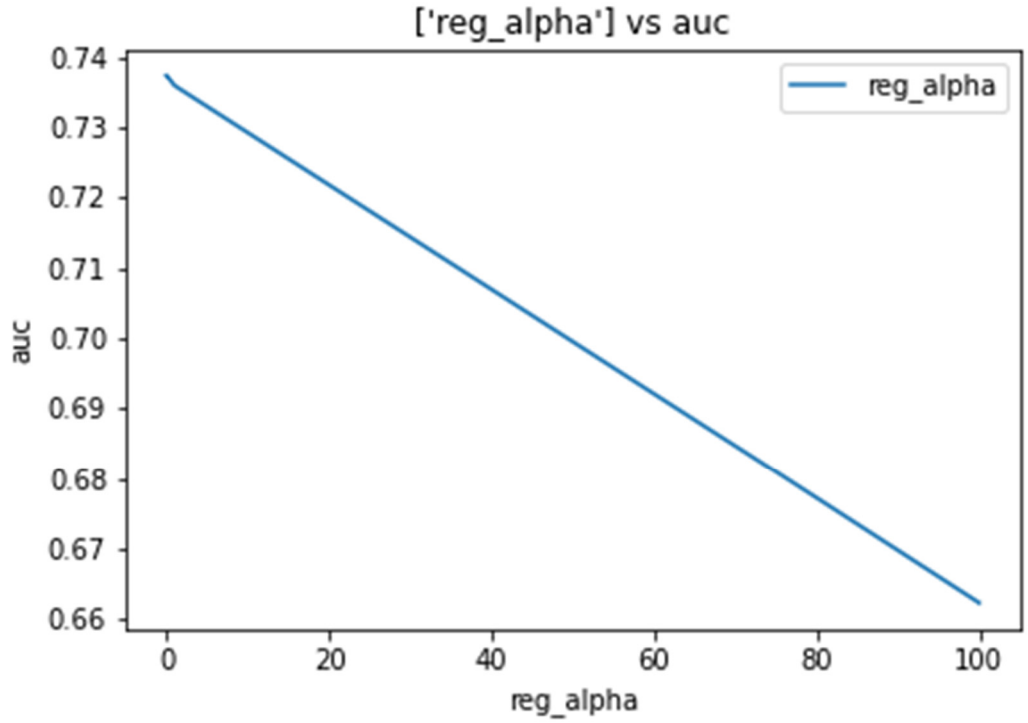

Figure 4.9 Rough search for the reg_alpha

The AUC reached the highest when reg_alpha was set to 0.01. Conduct the fine search around 0.01. The optimal value is 0.005 (Figure 4.10).

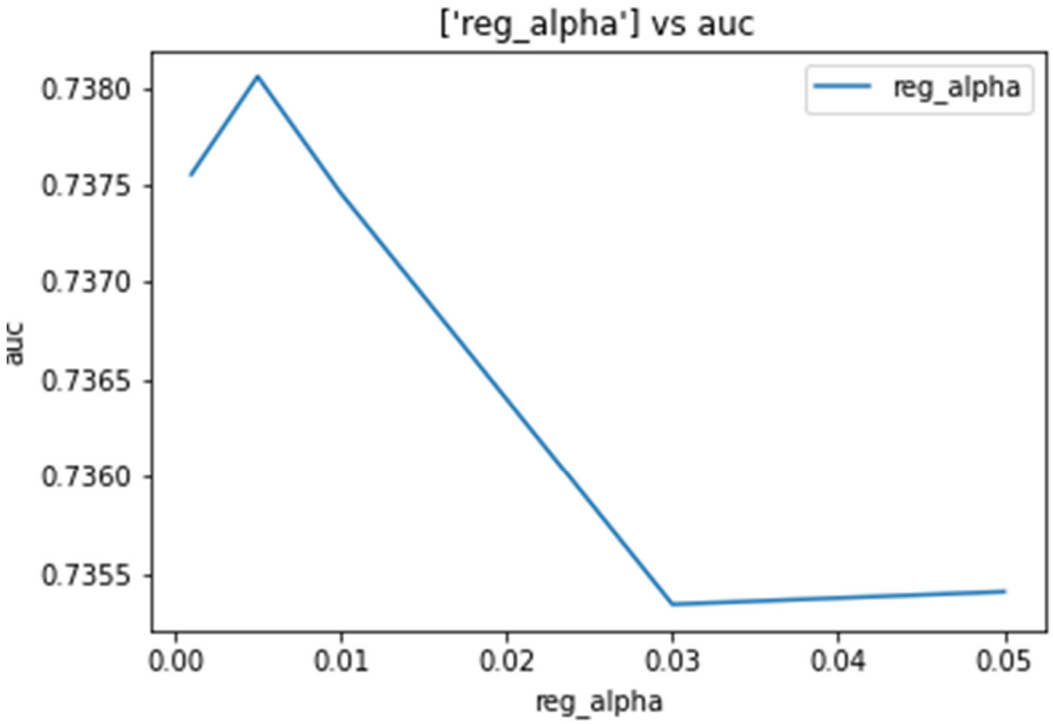

Figure 4.10 Fine search for the reg_alpha

After finishing tuning reg_alpha, determine the optimal value of reg_lamda. Follow the same principle of rough search. The value of 0.01 hit the best (Figure 4.11).

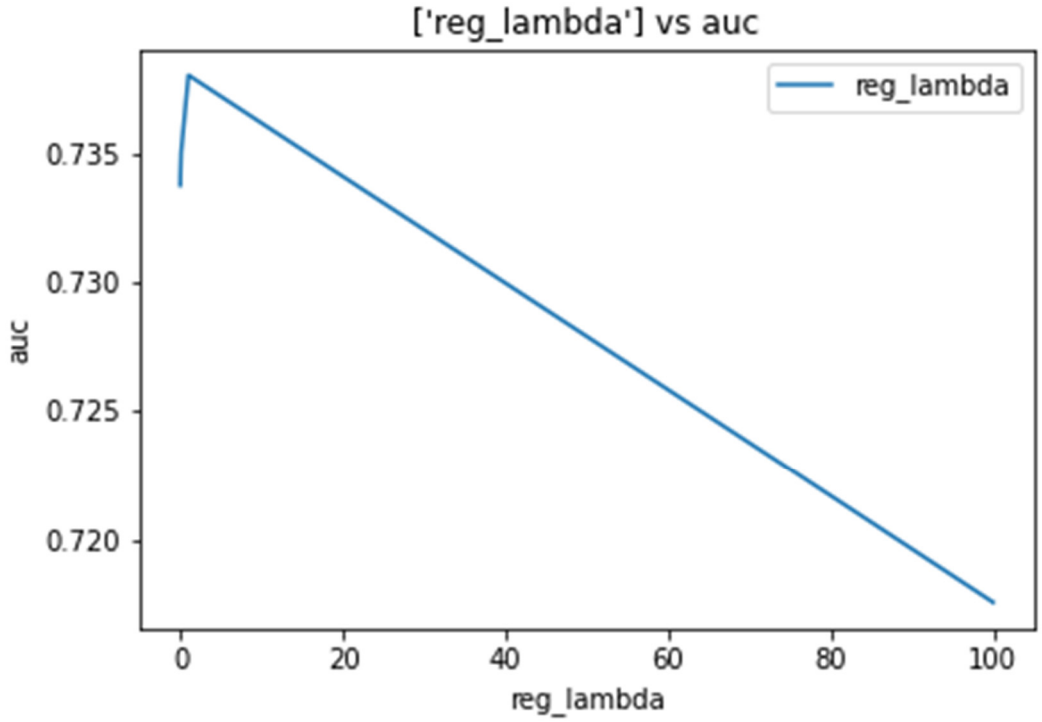

Figure 4.11 Rough search for the reg_lamda

Narrow the search space around 0.01, the optimal value was 1 (Figure 4.12)

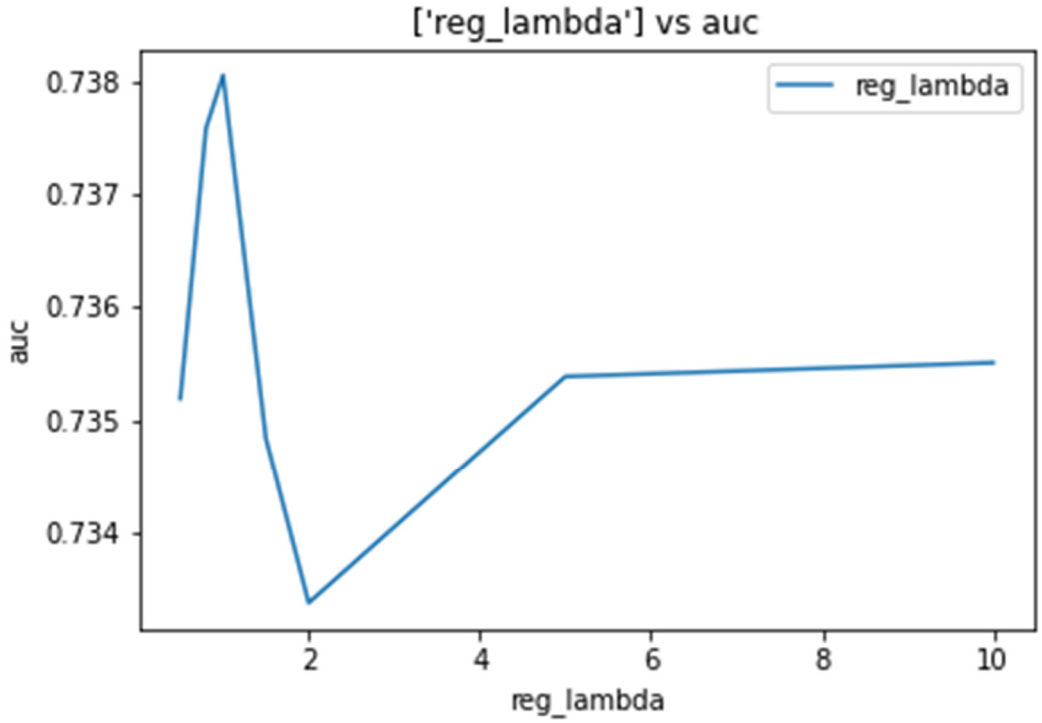

Figure 4.12 Fine search for the reg_lamda

After the previous tuning process, all the optimal values of the parameters have been determined. The final XGBoost model concerning the 1-year training set generated (Code 4.1).

Code 4.1 Optimal parameters of the 1-year training set

```
# Final model of 1-year training set
xgb_model = XGBClassifier(
    learning_rate =0.04,
    n_estimators=200,
    max_depth=8,
    min_child_weight=4,
    gamma=0,
    subsample=0.8,
    colsample_bytree=0.8,
    reg_alpha=0.005,
    reg_lamda=1,
    objective= 'binary:logistic',
    nthread=-1,
    scale_pos_weight=1,
```

```
    seed=27,
    tree_method="gpu_hist" # Gpu boost enable
    )
```

The tuning process of the 2-year and 3-year model keeps the same principle.

The final XGBoost model concerning the 2-year data generated (Code 4.2).

Code 4.2 Optimal parameters of the 2-year training set

```
# Final model of 2-year training set
xgb_model2 = XGBClassifier(
    learning_rate=0.24,
    n_estimators=200,
    max_depth=8,
    min_child_weight=2,
    gamma=0,
    subsample=0.8,
    colsample_bytree=0.8,
    reg_alpha=1e-6,
    reg_lamda=1,
    objective='binary:logistic',
    nthread=-1,
    scale_pos_weight=1,
    seed=27,
)
```

The final XGBoost model concerning the 3-year training set generated (Code 4.3).

Code 4.3 Optimal parameters of the 3-year training set

```
# Final model of 3-year training set
xgb_model3 = XGBClassifier(
    learning_rate=0.1,
    n_estimators=200,
    max_depth=7,
    min_child_weight=4,
    gamma=0,
    subsample=0.75,
    colsample_bytree=0.85,
```

```
    reg_alpha=1e-5,
    reg_lamda=0.8,
    objective='binary:logistic',
    nthread=-1,
    scale_pos_weight=1,
    seed=27,
)
```

4.3 Model Selection

The style of the market is constantly changing. Sometimes the stocks are rising in the current month and sometimes falling, and more often in a state of sideways. Therefore, the optimal model for the trading date in the date list could be totally different, and a single model can not suitable for all the trading date. Therefore, it is crucial to select the optimal model with the highest AUC for the specified trading date.

4.3.1 Traversal Search the Optimal Training Set and the Model

The three different lengths of the training sets and the corresponding three optimal models have been generated. In view of the continuous changes in market style, traversal search has been conducted to find the optimal pair. 3*3=9 permutation combinations of the training sets and models were tried respectively (Code 4.4).

Code 4.4 Pseudocode of the traversal search

```
training_set= [1_year_training_set, 2_year_training_set, 3_year_training_set]
xgb_model = [1_year_xgb_model, 2_year_xgb_model, 3_year_xgb_model]

for set in training_set:
    for model in xgb_model:
        cross-validation to select the optimal pair with the highest AUC
```

After the completion of the traversal search, the 1-year training set has always been selected as the optimal training set by the classifiers in all the trading dates within 10 years. Therefore, the conclusion can be drawn that the sequential time series do have a significant influence on the results.

4.3.2 Randomized Search the Optimal Model

From the last section, sequential time series matters. In the previous searches of the cross-validation, the influence of the timing sequence has not been taken into consideration. The cross-validation method randomly splits the training set into certain subsets.

The optimization of the cross-validation splitting strategy has been explored. To guarantee the time continuity, the training set was divided into five subsets in chronological order (Table 4.4).

Table 4.4 Belonging validation subset index of the training set months

Months before the current time	1	2	3	4	5	6
Belonging validation subset index	4	4	4	3	3	3
Months before the current time	7	8	9	10	11	12
Belonging validation subset index	2	2	1	1	0	0

Each piece of the data in the training set has been assigned the corresponding validation subset index. The cv parameter of the GridSearchCV or RandomizedSearchCV can accept the number of folders in a (Stratified)KFold by PredefinedSplit provided by Scikit-learn (Code 4.5).

Code 4.5 Grid search on the specified development set index

```
1. # Implement the PredefinedSplit to grid search on the dev set
2. ps = PredefinedSplit(test_fold=dev_set_index)
3.
4. grid_search_model = RandomizedSearchCV(estimator=model,param_distributions=para_list, n_it
   er=24, scoring='roc_auc', n_jobs=-1, iid=False, cv=ps, verbose=1)
```

Due to the introduction of the new cross-validation method, a slight parameter tuning of the current time optimal model was required. In the previous parameter tuning process, it has found that the learning rate, subsample and the colsample_bytree had a significant impact on the model performance. Hence, small-scale search for three parameters has been implemented (Code 4.6).

Code 4.6 The slight tuning parameter list

```
para_list = {
    'learning_rate': [0.04, 0.1, 0.24],
    'subsample': [i / 100.0 for i in range(75, 91, 5)],
    'colsample_bytree': [i / 100.0 for i in range(75, 91, 5)],
}
```

On the method selection of the grid search, a total number of 3*4*4*5=240 pieces of training needed for exhausting search where 5 is the number of the subsets for the cross-validation. To save the training time and system resource, the RandomizedSearchCV has replaced the GridSearchCV to become the grid search method.

For each trading date, grid search the optimal model using cross-validation method in chronological order with the 1-year training set and RandomizedSearchCV.

4.4 Feature Number Selection

The benefit of the gradient boosting method is that it is relatively straightforward to retrieve importance scores for each attribute after the boosted trees are constructed.

Generally, importance provides a score that demonstrates the importance degree of each feature was in the construction of the trees in a model. The relative importance is higher if more times of the corresponding feature are used (Brownlee 216).

XGBoost has the feature_importances_ method that is able to export the type of “gain” importance. The gain implies the relative contribution of the corresponding feature calculated by taking each feature’s contribution for each tree in the model(XGBoost Documentation 2020a). The gain is the most relevant attribute to interpret the relative importance of each feature, which represents the improvement in accuracy brought by a feature to the branches it is on.

Feature importance scores can be used for feature number selection in combination with the SelectFromModel method provided by Scikit-learn. It uses a threshold to select the most relative

features to the model above the threshold and discard the features that below. The optimal number of the most relative features can be obtained by feature number selection.

Through the plotting of the AUC figure, most of the months' patterns were similar to Figure 4.13 and Figure 4.14. As the number of features decreased, AUC also decreased gradually. At the same time, there was a specified optimal feature number (the number was 38 in Figure 4.13) not 46 that can reach the highest AUC. This demonstrated the significance of the feature number selection.

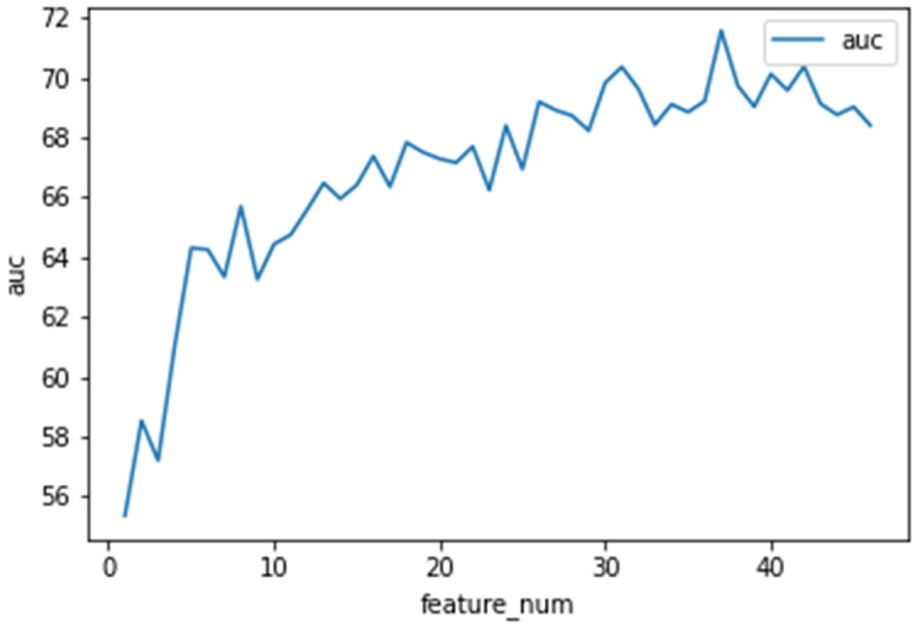

Figure 4.13 feature_num vs AUC based on the date of 25 January 2010

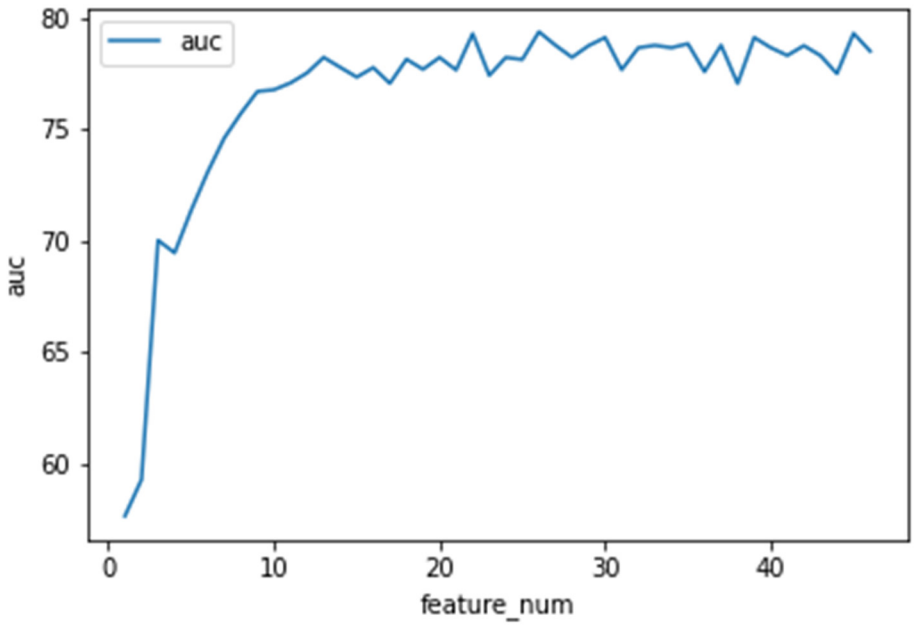

Figure 4.14 feature_num vs AUC based on the date of 18 July 2012

The prediction on the test set with and without feature selection has also been conducted. The result (Table 4.5) showed that although for most of the time, AUC on development set will be slightly improved through feature selection. However, in some cases, ACC on the test set will decline obviously because of the feature selection, and there was no certain rule to follow.

Table 4.5 Partial ACC on the test set with and without feature selection

Date	2012-04-20	2012-06-19	2012-10-17	2013-03-18
optimal feature number	18	24	34	28
ACC on test with threshold	0.4787472	0.4978632	0.470339	0.506383
ACC on test without threshold	0.5033557	0.5619658	0.502119	0.546809

The cause for the above phenomenon might be that the factors can not sufficiently explain the excess returns of stocks after screening. Some factors that are important to the market of the current time have been deleted.

For the above reason, feature selection has not been adopted in the model selection process due to the uncontrollability. However, the exploration demonstrated the significance of the feature number selection.

5 Result

The chapter demonstrates the prediction on the test set based on the optimal model, after which a list of stocks worth buying generated for each trading date through a certain strategy. The strategy report has been generated after the strategy return yield has been gained. The discussion on the important features selected by the classifiers has also been carried out.

5.1 ACC According to Date

The prediction on the test set based on the optimal model with 1-year training set for all the trading days has been conducted, after which the accuracy scores have been obtained (Figure 5.1).

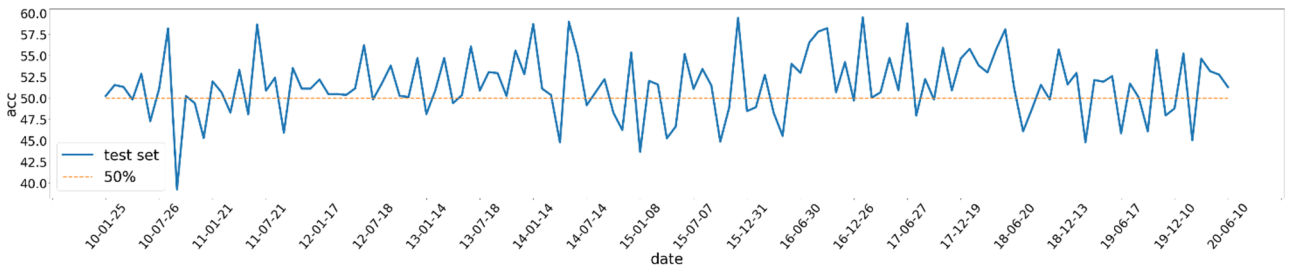

Figure 5.1 ACC according to date

The orange line represents 50% accuracy. The accuracies were above 50% most of the time, which proved that the prediction based on the XGBoost model better than the stochastic prediction. The idea was somewhat similar to that of boosting algorithm, the accuracies were slightly higher than 50% most of the time which were corresponding to the weak classifiers. The stock returns will gradually accumulate due to the integration of better predictions, which was similar to the boosting of the weak classifiers.

There were also sometimes that the accuracies were below 50% which were due to the style changing of the market. In that case, the training set did not help a lot to predict the return yield of the future, which led to the failure of the classifiers. This situation did not account for a large proportion among all the trading dates, however.

The dataset based on weekly frequency has also been obtained to inspect the possibility of the ACC improvement. That was, the training set has been expanded to 4 times more extensive compared to the origin. However, after a long time of the data acquisition and model selection, the prediction performance basically no improvement. The possible reason was that most of the selected factors were calculated based on the company financial report. The report usually issued quarterly. The data based on the monthly level were enough to reflect the changing of the report while the weekly frequency data contained redundant information.

5.2 Feature Importance According to Date

One of the most advantages of the decision-tree-based algorithms is the ability to output the important features selected by the classifiers.

Similar to the methods were used in feature selection, the feature importance scores were obtained on each trading days, the feature importance heatmap according to date was generated at the end of the prediction (Figure 5.2).

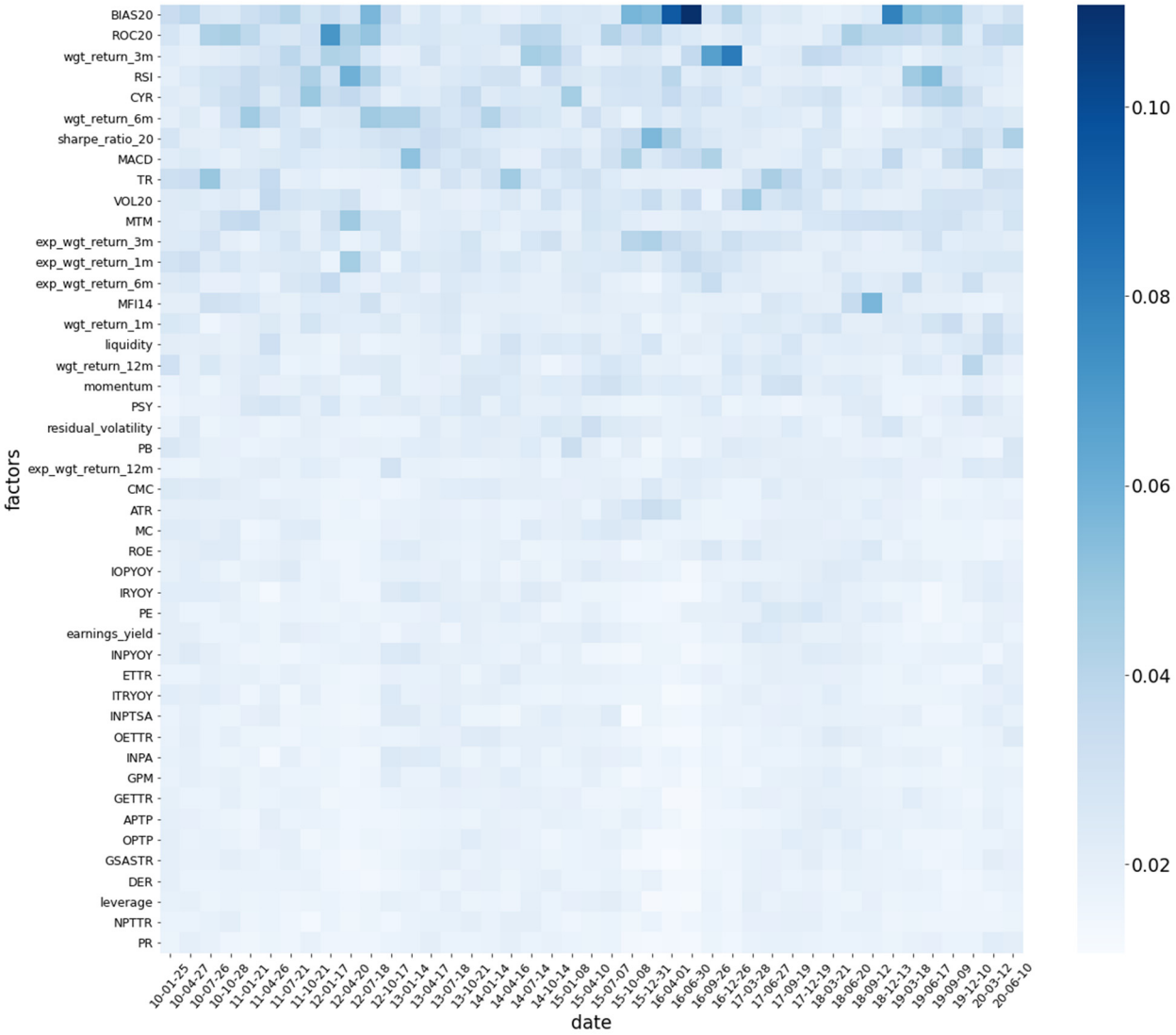

Figure 5.2 Feature importance according to date

As time goes by, the importance of each feature was continually changing. However, the technical factors such as BIAS20, ROC20, RSI and MACD, etc. were always occupied the top of the feature importance list. The momentum reversal factors calculated separately in section 3.1.1 did play an important role in prediction. This is also the reason that technical factors are always chosen in the stock timing strategy. In contrast, the weight of the fundamental factors was not that important.

5.3 Generate the Recommended Stock Portfolio

XGBoost has the predict_proba method which can display the probability of the possibility that predicted as the specific class.

Fan and Palaniswami measured and ranked the probability to select the top stocks and the strategy to maintain the equally weighted portfolio when they implemented the SVM to predict the stock returns on Australian stock market(Fan and Palaniswami 2001). The idea can be used for reference by the project.

After the prediction on the test set, the probability list was generated. Through sorting the probability of being classified as positive and selecting the top stocks in the list, the portfolio composed of the top N stocks can be generated. The top N stocks were considered by the classifier to be the most recommended for purchase in the next month. The N was selected as 20, 50, 100, respectively and validate the corresponding portfolio returns within 10 years. The maintained portfolio was regenerated and updated on each trading date.

The equally weighted portfolio strategy is the commonly used simple strategy, means for each stock within the portfolio, equal allocation of the funds to buy the same amount. The equally weighted portfolios contained the top N stocks for each trading date have been generated after the whole process.

The transaction process is, on each trading day, sell the stocks that are not in that day's portfolio and buy the stocks in the portfolio with equal weight.

5.4 Backtesting Report Analysis

In addition to providing the factor data, JoinQuant also has the backtesting platform to evaluate the historical returns of the trading strategy and generate the backtesting report(JoinQuant 2020a).

An investment system, or namely investment strategy, has been designed for backtesting in the project. The backtesting period was from 1 January 2010 to 9 July 2020. During this time, different market styles have experienced completely such as bull market, bear market and shock market, so that the effectiveness of the strategy can be tested comprehensively. The transaction was executed in strict consistency with the strategy during the backtesting period.

The initial fund was set to ¥1000,000. The trading commission were 0.3‰ of the buying and selling turnover amount. Various taxes and fees were set according to the real situation of A-shares. The CSI 300 Index has been set as the compared benchmark. The benchmark returns are the income that can be obtained by tracking the constituent stocks in the CSI 300 Index.

Before the analysis of the backtesting report, introduce the relevant technical indicators of the backtesting report (Table 5.1).

Table 5.1 Interpretations of relevant technical indicators

Name	Description
Total Returns	$$Total\ Returns = \frac{P_{end} - P_{start}}{P_{start}} \times 100\%$$ Where, $P_{end} = the\ total\ fund\ of\ the\ strategy's\ stock\ and\ cash\ at\ the\ end;$ $P_{start} = the\ total\ fund\ of\ the\ strategy's\ stock\ and\ cash\ at\ the\ beginning;$
Total Annualized Returns	$$Total\ Annualized\ Returns = R_p = ((1+P)^{\frac{250}{n}} - 1) \times 100\%$$ Where, $P = Total\ Returns;$ $n = strategy\ running\ days;$
Benchmark Returns	returns of the benchmark.
Excess Returns	$$Excess\ \mathrm{Returns} = \frac{Strategy\ \mathrm{Returns} + 100\%}{Benchmark\ \mathrm{Returns} + 100\%} - 100\%$$ returns exceeding the benchmark.
Alpha	investment is faced with systematic risk (Beta) and non-systematic risk (Alpha), and Alpha means that investors get returns unrelated to market fluctuations.
Beta	the systematic risk of investment and reflects the sensitivity of the strategy to changes in the market.
Sharpe	it indicates how much excess returns will be generated for each unit of total risk. The benefits and risks of the strategy can be comprehensively considered at the same time.
Information Ratio	$$\text{Information Ratio} = \frac{R_p - R_m}{\sigma_t}$$ Where, $R_P = Total\ Annualized\ Returns\ of\ the\ strategy;$ $R_m = Total\ Annualized\ Returns\ of\ the\ benchmark;$

Name	Description
	$\sigma_t = The\ annualized\ standard\ deviation\ of\ the\ difference\ between\ strategy\ and\ benchmark\ daily\ earnings;$ measure the excess returns brought by unit excess risk. The larger the information ratio, the higher the excess returns. The reasonable investment goal is to pursue high information ratio under the moderate risk.
Algorithm Volatility	measure the risk of strategy. The greater the fluctuation, the higher the risk of strategy.
Max Drawdown	describe the worst possible situation and the most extreme possible loss of the strategy.

Input the portfolio generated before into the backtesting platform through programming, the backtesting report will be produced. Through the comparison of the strategy composed of the top 20, 50,100 stocks, the top 50 one achieved the best result (Figure 5.3).

Figure 5.3 Backtesting report

The relevant technical indicators of the strategy (Table 5.2) showed that the system achieved total strategy returns of 200.08% with the excess returns of 122.25%, sharp ratio of 0.287, information ratio of 0.689, whereas the returns of the benchmark CSI 300 Index were 35.38% in the backtesting from 2010 to 2020. The strategy returns far exceeded the benchmark, which proved the maintained portfolio totally “beat the market”. The strategy performed very well and obtained a large α which proved that the classifier worked well on prediction. At the same time, the strategy achieved a far higher increase than the benchmark in the incremental market (the bull market), which proved that the application in the bull market is relatively more valuable.

Table 5.2 Technical indicators of the strategy

Name	Description
Total Returns	200.88%
Total Annualized Returns	11.38%
Benchmark Returns	35.38%
Excess Returns	122.25%
Alpha	0.084
Beta	0.978
Sharpe	0.287
Information Ratio	0.689
Algorithm Volatility	0.257
Max Drawdown	51.24%

The strategy still had the problem of massive retracement of the bear market, which is common to the equally weighted strategy without risk management. This problem should not be considered in the aim of the stock selection, however.

Generate the industry configuration through the recommended stock portfolio (Table 5.3). The Financial, Information Technology, Industry, Optional Consumption and Material stocks accounted for most of the weight which was consistent with common sense. Financial and Information Technology are widely regarded as high-value industries. Industry, Optional Consumption and Material are the traditional advantageous industries.

Table 5.3 Industry configuration comparison

Industry name	Strategy configuration	Benchmark configuration	Weight allocation difference
Financial Index	9.80%	36.88%	-27.08%
Material Index	16.72%	9.01%	7.71%
Information Technology Index	13.18%	5.99%	7.19%
Industry Index	20.51%	14.85%	5.66%
Optional consumption Index	15.03%	9.90%	5.13%
Healthcare Index	7.61%	5.44%	2.17%
Cash	1.87%	0.00%	1.87%
Energy Index	3.54%	4.72%	-1.18%
Daily consumption Index	6.44%	7.59%	-1.15%
Real estate Index	1.81%	1.53%	0.28%
Telecommunications Service Index	0.81%	1.07%	-0.26%
Utilities Index	2.67%	2.81%	-0.14%
Other	0.00%	0.05%	-0.05%

5.5 Label Improvement

In the previous Y label treatment, 40% of the stocks in the middle have been discarded to make the boundary between positive and negative stocks more obvious. In the actual application of the strategy, it is not operable, however. Assume to make the prediction on the stock return yield of the next month based on the current real-time, since next month is the future time which leads to the next month return yield can not be obtained. This makes the test set data cannot be discarded due to the unknown of the future data. If no process applied to the test set, it will also lead to the problem that the test set not consistent with the training set. On the other hand, discarding the original data is also artificial screening, which causes the classifier unable to foresee the complete data.

Therefore, a new way of handling the label has been conducted. The data in the middle no longer discarded, took the top 30% as positive and took the last 70% as negative, after which, the improved strategy was validated (Table 5.4).

Table 5.4 Technical indicators of the improved strategy

Name	Description
Total Returns	197.11%
Total Annualized Returns	11.24%
Benchmark Returns	35.38%

Name	Description
Excess Returns	119.46%
Alpha	0.082
Beta	0.999
Sharpe	0.276
Information Ratio	0.666
Algorithm Volatility	0.231
Max Drawdown	54.17%

The total returns of the new strategy were slightly lower than the older one. However, the new strategy did not require to screen the data artificially, the prediction on the future time of the present time can be gained directly. Moreover, the algorithm volatility of the new strategy was lower which turned to safer. All these proved the validity of the new strategy.

6 Preliminary Exploration of the Automated Trading

Automated trading refers to through establishing the rules for entering and exiting, the computer is able to execute by following the rules automatically(Kemp et al. 2017). In section 5.3, the stock list consists of the top 50 stocks each trading date has been generated. It would be a great burden If the stocks are bought and sold manually on the trading day once a month. Moreover, If the strategy is improved and the trading frequency is further raised to the daily level, it is even more impossible to trade manually. This is the significance of the existence of automated trading.

Usually, automated trading sends entrusted orders for execution through the Application Programming Interface (API). Currently, automated trading API has not been opened to the A-shares market due to the consideration of transaction security by government regulatory authorities. Therefore, there is no particularly mature solution to this at present.

“easytrader” is the python library used for automated trading by simulating the keyboard and mouse actions through the reading of the window handle(easytrader Documentation 2020). Its bottom layer is based on the support of PyWinAuto library which can automate the Microsoft Windows GUI(PyWinAuto Documentation 2020).

After reading and modifying the corresponding window handle, through the implementation of the easytrader, program connected with the Tonghuashun stock trading client. The source code of the easytrader also has been modified a lot to adapt to the current version of the client.

The actions such as buy, sell, query balance (Figure 6.1), query position (Figure 6.2), query today's trades (Figure 6.3), query today's entrusts (Figure 6.4), etc. have been called and been deployed successfully.

Figure 6.1 Related screenshot of the query balance

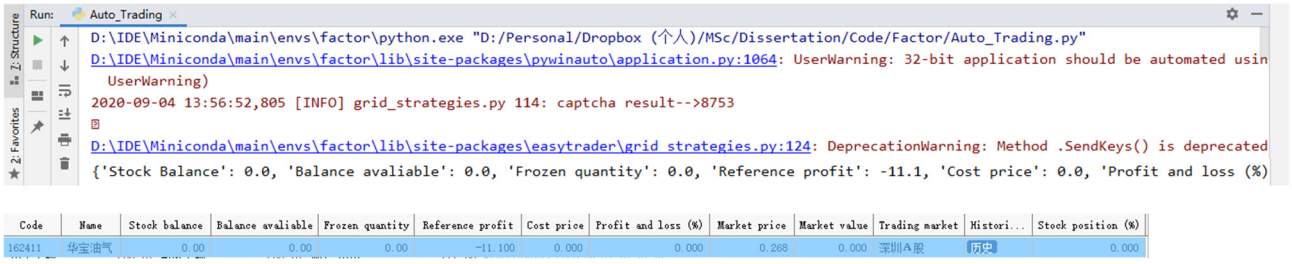

Figure 6.2 Related screenshot of the query position

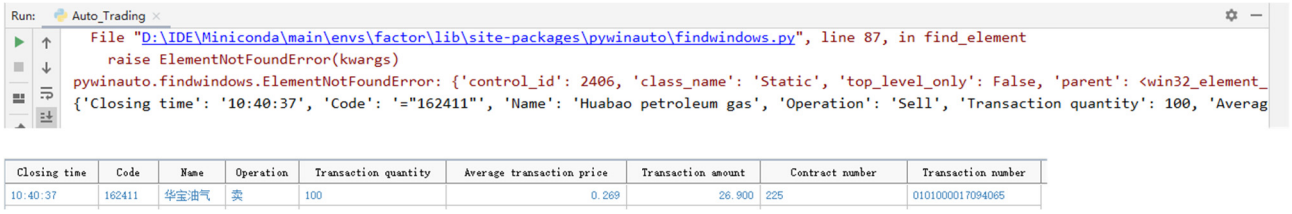

Figure 6.3 Related screenshot of the query today's trades

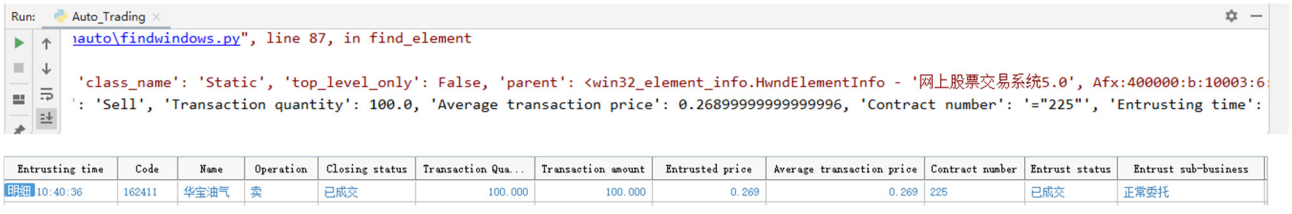

Figure 6.4 Related screenshot of the query today's entrusts

In the process of trading, encountered the problem of captcha verification. After configuring the relevant parameters of the tesseract library, the verification code can be recognized automatically. “tesseract” can also improve the recognition accuracy by self-training through intercepting a large number of relevant verification codes. After obtaining the verification code, call PyWinAuto library for automatic input (Figure 6.5).

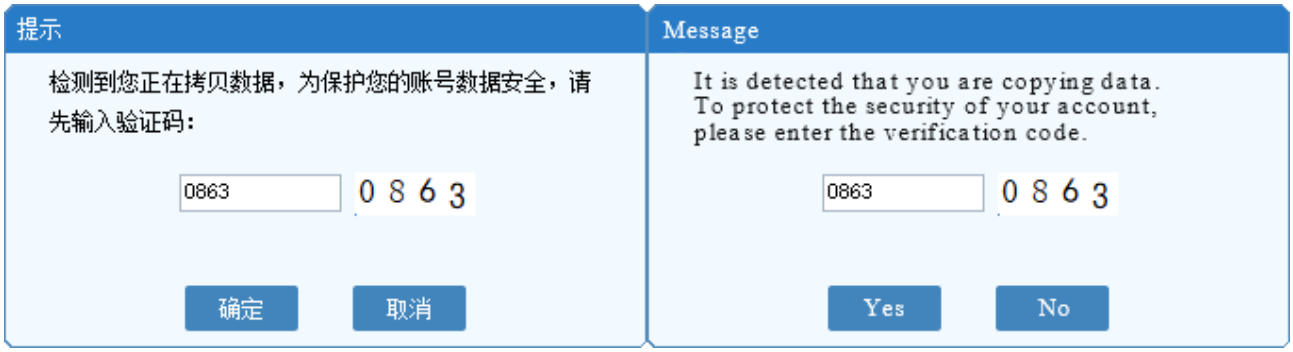

Figure 6.5 Automatically enter the verification code

During the implementation, only transactions of a few stocks were verified. No attempt was made to a more enormous amount. However, many problems also have been raised during the process. The first one is that the software was often hot updated, which causes the window handle changed and needed to get the new handle again. Otherwise, the program unable to continue execution normally. Secondly, the program was based on the simulation operation of the mouse and keyboard. This might cause a certain step in the middle can not be carried out successfully due to the computer stuck at the moment, thereby causing the subsequent failure. Thirdly, the connection between the program and the trading client through the pywinauto was also unstable and often interrupted.

Reliability and security should be the primary factors to be considered in automated trading. The existing solution implemented by the easytrader does not satisfy the requirement. The reliable way would be implemented through API. However, this usually requires a large number of funds to negotiate directly with the stock exchange. Despite the existence of such problems, the project made a preliminary exploration of automated trading and put forward a possible landing scheme. If time permits, more mature solutions can be explored.

7 Conclusion and Future Works

The project has set up a complete investment system pipeline based on the multi-factor and machine learning model. The system mainly includes the process of data preparation, model selection and strategy generation.

The workflow is to use the data of the whole year before the current time as the training set and implement the optimized XGBoost model to predict the stock return yield of the next month at the current time, after which the system selects the top 50 stocks that most likely to rise in the next month for equal weight investment. And then employ the automated trading system to maintain the portfolio on the designated trading day automatically.

The system achieved total strategy returns of 200.08% with the excess returns of 122.25%, sharp ratio of 0.287, information ratio of 0.689, whereas the total returns of the benchmark CSI 300 Index were 35.38% in the backtesting from 2010 to 2020.

The strategy returns far exceeded the benchmark, which indicated that the system could be used as the method to auxiliary the quantitative stock selection, which can provide helpful suggestions on investment for retail investors. The automated trading system was also preliminarily explored in the project. The project provides a reference for the research gap of the quantitative investment in the A-shares market.

Concerning future works, there are many possible research directions because of the large scale of the project,

i. If a factor has been widely known, according to the assumption of market efficiency, the excess returns of the factor that can be obtained are limited. The factors selected in the project were common to the public. For the next step, some new and rare factors can be explored to obtain higher excess returns. At the same time, factor validation is also a grand subject that can be further into.
ii. As for the division of training set and test set, it is also a popular way to use the data of previous

several years to predict the stock return yield of the next year, which can be compared.

iii. Due to the tight schedule of the project, XGBoost which belongs to the category of the relatively mature solution decision-tree-based model has been chosen. The next step could be comparing more models such as SVM, DNN, LSTM, etc. models to see the strategy returns. However, Lin and Chen have also mentioned that in the monthly multi-factor stock selection, the comprehensive performance of neural network model is not as good as XGBoost which might due to the data of monthly frequency is not large enough for deep learning to exert its strength(Lin and Chen 2017b).

iv. About risk management, equal weight allocation is one of the simpler ways without the need for risk management. A simple deployment of risk management, for example, to stop trading for a period after a continuous decline in recent trades to avoid a bear market. More professional ways, such as implementing the Markowitz mean-variance portfolio model or other portfolio management methods are all the possible ways(Steinbach 2001). These methods enable to get higher returns with the least risk.

v. Automated trading based on easytrader has certain reliability problems. Exploration of more mature solutions could be made.

9 References

Appel, G. 1985. *The moving average convergence-divergence trading method: advanced version*. Scientific Investment Systems

Brownlee, J. 216. *Feature Importance and Feature Selection With XGBoost in Python*. Available at: https://machinelearningmastery.com/feature-importance-and-feature-selection-with-xgboost-in-python/ [Accessed: 30 August 2020].

Chen, K. et al. eds. 2015. *A LSTM-based method for stock returns prediction: A case study of China stock market. 2015 IEEE international conference on big data (big data).* IEEE.

Chen, T. and Guestrin, C. eds. 2016. *Xgboost: A scalable tree boosting system. Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining.*

China Securities Index Company. 2020. *CSI 800 Index*. Available at: http://www.csindex.com.cn/en/indices/index-detail/000906 [Accessed: 30 August 2020].

Chong, E. et al. 2017. Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies. *Expert Systems with Applications* 83, pp. 187-205.

darcylike. 2018. *[Factor stock selection and backtesting]*. BigQuant. Available at: https://bigquant.com/community/t/topic/128196 [Accessed: 27 August 2020]. (in Chinese)

easytrader Documentation. 2020. *[easytrader Documentation]*. easytrader. Available at: https://easytrader.readthedocs.io/zh/master/ [Accessed: 30 August 2020]. (in Chinese)

Fama, E. F. and French, K. R. 1993. Common risk factors in the returns on stocks and bonds. *Journal of financial economics* 33(1), pp. 3-56.

Fama, E. F. and French, K. R. 2004. The capital asset pricing model: Theory and evidence. *Journal of economic perspectives* 18(3), pp. 25-46.

Fama, E. F. and French, K. R. 2015. A five-factor asset pricing model. *Journal of financial economics* 116(1), pp. 1-22.

Fan, A. and Palaniswami, M. eds. 2001. *Stock selection using support vector machines*. *IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No. 01CH37222).* IEEE.

Fang, P. 2018. *[A stock prediction and quantitative investment system based on machine learning].* MSc Dissertation, ZHE JIANG UNIVERSITY. (in Chinese)

Fischer, T. and Krauss, C. 2018. Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research* 270(2), pp. 654-669.

Ge, L. and Zhou, X. 2020. [Multi-Factor Stock Selection Model Based on XGBoost Algorithm]. *Information Technology and Standardization* 2020(5), pp. 36-41. (in Chinese)

Ghosn, J. and Bengio, Y. eds. 1997. *Multi-task learning for stock selection*. *Advances in neural information processing systems.*

HAYES, A. 2020a. *Exponential Moving Average (EMA)*. Investopedia. Available at: https://www.investopedia.com/terms/e/ema.asp [Accessed: 28 August 2020].

HAYES, A. 2020b. *Moving Average Convergence Divergence – MACD*. Investopedia. Available at: https://www.investopedia.com/terms/m/macd.asp [Accessed: 28 August 2020].

Jerez, J. M. et al. 2010. Missing data imputation using statistical and machine learning methods in a real breast cancer problem. *Artificial intelligence in medicine* 50(2), pp. 105-115.

JoinQuant. 2020a. *[JoinQuant Backtesting Platform]*. Available at: https://www.joinquant.com/algorithm/index/list [Accessed: 27 August 2020]. (in Chinese)

JoinQuant. 2020b. *[JoinQuant factor library]*. Available at: https://www.joinquant.com/help/api/help?name=factor_values [Accessed: 27 August 2020]. (in Chinese)

JoinQuant. 2020c. *[Summary of frequently asked questions about data]*. Available at: https://www.joinquant.com/view/community/detail/1226a48b1f9b7bd90dc3516feea8b5cc?type=2 [Accessed: 27 August 2020]. (in Chinese)

Kemp, I. G. A. et al. 2017. System and method for automated trading. Google Patents

Lin, X. and Chen, Y. 2017a. *[Huatai securities artificial intelligence series 5 - Random forest model for artificial intelligence stock selection]*. Huatai Securities Financial Engineering Research Department. in Chinese)

Lin, X. and Chen, Y. 2017b. *[Huatai securities artificial intelligence series 9 - LSTM model for artificial intelligence stock selection]*. Huatai Securities Financial Engineering Research Department. in Chinese)

Ling, C. X. et al. eds. 2003. *AUC: a better measure than accuracy in comparing learning algorithms*. *Conference of the canadian society for computational studies of intelligence.* Springer.

Liu, J. and Zhang, J. 2020. [Multi-factor stock selection model based on machine learning]. *Times Finance* (17), pp. 99-103. (in Chinese)

Marsland, S. 2015. *Machine learning: an algorithmic perspective*. CRC press

Nguyen, T. H. et al. 2015. Sentiment analysis on social media for stock movement prediction. *Expert Systems with Applications* 42(24), pp. 9603-9611.

Noy-Meir, I. et al. 1975. Data transformations in ecological ordination: II. On the meaning of data standardization. *The Journal of Ecology*, pp. 779-800.

Pandas Documentation. 2020. *pandas.DataFrame.corr*. pandas. Available at: https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.corr.html [Accessed: 30 August 2020].

PyWinAuto Documentation. 2020. *PyWinAuto Documentation*. PyWinAuto. Available at: https://pywinauto.readthedocs.io/en/latest/ [Accessed: 30 August 2020].

Ross, S. A. 2013. The arbitrage theory of capital asset pricing.*Handbook of the fundamentals of financial decision making: Part I*. World Scientific, pp. 11-30

Rosset, S. ed. 2004. *Model selection via the AUC*. *Proceedings of the twenty-first international conference on Machine learning*.

Rubinstein, M. 2002. Markowitz's" portfolio selection": A fifty-year retrospective. *The Journal of finance* 57(3), pp. 1041-1045.

Saad, E. W. et al. 1998. Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks. *IEEE Transactions on neural networks* 9(6), pp. 1456-1470.

Schumaker, R. P. and Chen, H. 2009a. A quantitative stock prediction system based on financial news. *Information Processing & Management* 45(5), pp. 571-583.

Schumaker, R. P. and Chen, H. 2009b. Textual analysis of stock market prediction using breaking financial news: The AZFin text system. *ACM Transactions on Information Systems (TOIS)* 27(2), pp. 1-19.

Scikit-learn Documentation. 2020a. *sklearn.model_selection.GridSearchCV*. Scikit-learn. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html [Accessed: 30 August 2020].

Scikit-learn Documentation. 2020b. *sklearn.model_selection.RandomizedSearchCV*. Scikit-learn. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html [Accessed: 30 August 2020].

Sorensen, E. H. et al. 2000. The decision tree approach to stock selection. *The Journal of Portfolio Management* 27(1), pp. 42-52.

Statsmodels Documentation. 2020. *Statsmodels Documentation*. statsmodels. Available at: https://www.statsmodels.org/ [Accessed: 30 August 2020].

Steinbach, M. C. 2001. Markowitz revisited: Mean-variance models in financial portfolio analysis. *SIAM review* 43(1), pp. 31-85.

Ta, V.-D. et al. 2020. Portfolio Optimization-Based Stock Prediction Using Long-Short Term Memory Network in Quantitative Trading. *Applied Sciences* 10(2), doi: 10.3390/app10020437

Tushare. 2020. *[API documentation]*. Available at: https://tushare.pro/document/2 [Accessed: 30 August 2020]. (in Chinese)

Vecmanis, K. 2019. *XGBoost Hyperparameter Tuning - A Visual Guide*. Available at: https://kevinvecmanis.io/machine%20learning/hyperparameter%20tuning/dataviz/python/2019/05/11/XGBoost-Tuning-Visual-Guide.html [Accessed: 30 August 2020].

Wilcox, R. 2005. Trimming and winsorization. *Encyclopedia of biostatistics* 8,

Wilcox, R. 2011. *Modern statistics for the social and behavioral sciences: A practical introduction*. CRC press

XGBoost Documentation. 2020a. *Python API Reference*. XGBoost. Available at: https://xgboost.readthedocs.io/en/latest/python/python_api.html# [Accessed: 30 August 2020].

XGBoost Documentation. 2020b. *XGBoost Parameters*. XGBoost. Available at: https://xgboost.readthedocs.io/en/latest/parameter.html [Accessed: 30 August 2020].

Zhang, L. et al. eds. 2017. *Stock price prediction via discovering multi-frequency trading patterns*. *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining.*