Multi Factor Stock Selection Model Based on AdaBoost in China A-share market

Prepared by
Beibei Feng, Ye Hua, Haokai Ma
Columbia University

Advising
Professor Alexander Greyserman

For
Professor Alexander Greyserman
MATH 5220: Quantitative Methods in Investment Management
December 20, 2020

Table of Contents

Background

Since first coming to 2012, some of the reform measures have been aimed at deepening China's financial market and giving stock markets a greater role in financing corporate investment in China. However, unlike the U.S. economy plays an important role in raising investment funding for its corporations, China's stock market has often been likened to a casino, dominated by unsophisticated retail investors gambling their wealth rather than looking for long-term sound investments.

As China is looking to expand the depth and role of its stock markets and wishes to open its capital account to attract more foreign investors, our project would give a general introduction of the China A-share market and introduce our multi factor stock selection model based on AdaBoost in the China A-share market.

Our Objective

Recent years, the combination of multi-factor stock selection model and machine learning algorithm has attracted wide attention from academia and industry in China. The factors do not only have the ability to explain the structure of stock returns, and they also provide portfolio and risk managers with a framework to categorize stocks.

Our idea originated from the paper "Multi-factor Stock Selection Model Based on Adaboost" by Ru Zhang and Tong Cao in 2018. This paper suggests that Adaboost multi factor stock selection is effective, and excess return can be gained than market index.

In this project, we aim to establish a multi-factor stock selection model based on the AdaBoost algorithm, by which we select stocks through the analysis of various indicators of a stock.

Introduction

An Introduction to our Data

All data for this project are daily China A-Share stock price data from 1997-04-30 to 2020-09-30, and we also include companies who have left the A-share market.

We use the data from BaoStock, which is the biggest open source equity-related data platform in China. It provides trustworthy data for Chinese quantitative finance experts.

Introduction to China A-share Market

China A-shares are primarily traded amongst domestic investors on the Shanghai and Shenzhen

Exchanges. There are a total of 4215 stocks in the A-shares market, and our trading strategy includes all 4215 stocks.

We set SSE Composite Index as the benchmark for backtesting as **CSI 300 Index** has not been established in 1997. The SSE Index stands for Shanghai Stock Exchange Index and is a stock market index of all stocks that are traded at the Shanghai Stock Exchange.

***CSI 300 Index**: a capitalization-weighted stock market index designed to replicate the performance of the top 300 stocks traded on the Shanghai Stock Exchange and the Shenzhen Stock Exchange.

Differences between US Stock Market and China A-share Market

US companies are heavily dependent on equity financing, but in China, only a small percentage (around 5%) of total corporate financing is funded by equity. Chinese corporations rely much more heavily on bank loans and retained earnings.

There are a series of complex regulations and rules for short selling in China under the pressure of Chinese government and stock market regulators, so it would be irrational to use a combination of long and short position in our strategy. Thus, we only pick the assets to long in the stock selection part.

An Introduction to Our Factors

- Initially, we select a total of 83 factors.
- 68 factors are from A-share company financial statements.

	Factor	Abbreviation
0	Proportion of three expenses	Pote
1	Main business profit	Mbp
2	Main business profit rate	Mbpr
3	Main business cost rate	Mbcr
4	Main business income	Mbi
5	Main business income growth rate	Mbigr
6	Main business profit proportion	Mbpp
7	Equity ratio	Er
8	Net profit	Np
9	Parent company net profit	Pcnp
10	Net profit growth rate	Npgr
11	Net assets growth rate	Nagr
12	return on equity	roe
13	weighted return on equity	wroe

14	interest payment multiples	ipm
15	total profit	tp
16	net fixed assets ratio	nfar
17	Fixed asset turnover rate	Fatr
18	Fixed assets proportion	Fap
19	Basic earnings per share	Beps
20	Inventory turnover days	Itd
21	Inventory turnover rate	Itr
22	Accounts receivable turnover days	Artd
23	Accounts receivable turnover rate	Artr
24	Total liabilities	Tl
25	Total assets	Ta
26	Total assets net profit rate	Tanpr
27	Total assets profit rate	Tapr
28	Total assets turnover days	Tatd
29	Total assets turnover rate	Tatr
30	Total assets growth rate	Tagr
31	Cost expense margin	Cem
32	Investment income	Ii
33	Net assets per share	Naps
34	Net cash flow from operating activities per share	Ncffoaps
35	Current ratio	Cr
36	Current liabilities	Cl
37	Current assets	Ca
38	Current asset turnover days	Catd
39	Current asset turnover rate	Catr
40	Liquidation value ratio	Lvr
41	Net increase in cash and cash equivalents	Niicace
42	Cash ratio	Cashr
43	Cash flow ratio	Cfr
44	Net cash flow from operating activities	Ncffoa
45	ratio of net operating cash flow to net profit	ronocftnp
46	ratio of net operating cash flow to debt	ronocftd
47	ratio of net operating cash flow to sales revenue	ronocftsr
48	Shareholders_equity does not include minority ...	Sdnims
49	Shareholders_equity to fixed assets ratio	Stfar
50	Shareholders_equity ratio	Sr
51	Return on equity	Roe
52	Operating profit	Op
53	Operating profit rate	Opr

54	Net non-operating income and expenditure	Nniae
55	Debt-to-owner equity ratio	Der
56	Return on assets	Roa
57	Return on operating cash flow of assets	Roocfoa
58	Asset-liability ratio	Ar
59	Capitalization ratio	Capr
60	Capital immobilization ratio	Cir
61	quick ratio	qr
62	sales net profit margin	snpm
63	sales gross profit margin	sgpm
64	long-term debt to working capital ratio	ldtwcr
65	long-term debt ratio	ldr
66	Long-term assets to long-term capital ratio	Latlcr
67	Non-main business proportion	Nbp

- 10 factors are technical indicators.
  - MACD: moving average convergence divergence
  - DIF: MACD line
  - DEA: signal line
  - K, D and J indicators
  - RSV: raw stochastic value
  - MA: 30-day moving average
  - vol: standard deviation
  - turn_mean: average monthly turnover rate
- 5 factors are valuation indicators.
  - P/E ratio, P/B ratio, P/S ratio, P/CF ratio and Market cap

*In this paper, when talking about factor value, we are using the monthly value in T^{th} month, and when we talk about return, we mean the stock return in $(T+1)^{th}$ month.

Data Processing

Data Cleaning

We remove four factors, "MACD", "DEA", "DIFF", and "Beps" from our factor universe as they have too many missing values.

Factor Processing

Firstly, we convert the 79 factors to monthly values. The second step is standardization, or in other words feature scaling. This step makes our future algorithm less sensitive to the outliers.

And then, we sort the value of factors from smallest to largest. The last step of our factor processing is stratification. We divide each factor into 10 groups from smallest to largest and set the group with the largest value of factors G9, and the group with the smallest value of factors G0.

Before talking about our sequential factor screening strategy, we need to introduce two important features at first. They are Information Coefficient (IC) and Information Gain (IG).

Information Coefficient (IC)

Information Coefficient gives an overview of the factor forecasting ability. More precisely, this is a measure of how well the factor ranks the stocks on a forward return basis. IC value is defined as the rank correlation between the factor and the forward return. In statistical terms, the rank correlation is a nonparametric measure of dependances between these two measures. And Obviously, ICs must be as high as possible in absolute terms.

Information Gain (IG)

Information Gain measures the reduction in entropy, or in other words surprise, by transforming a dataset and is often used in training decision trees. A larger information gain suggests a lower entropy group and hence is less surprising. In General, we desire a smaller entropy as values sampled from it are more predictable.

Sequential Factor Screening

In order to evaluate each factor and searching for the most important factors among the many factors, the following steps are taken:

1. Calculate the IC value of each factor.

2. Calculate the information gain(IG) of each factor.

3. Calculate the monthly average return of each factor during our backtesting period from 1997-4-30 to 2009-12-31. When the correlation coefficient between the factor value and return is positive, G9 group is the advantage group and G0 group is the disadvantage group. On the contrary, when the correlation is negative, G0 is the advantage group and G9 is the disadvantage group. Then, we record the monthly average returns of the advantage group and the disadvantage group of each factor.

4. Define the advantage group and disadvantage group according to the method we mentioned in (3) and then calculate the probability of the monthly return of the advantage group and the disadvantage group outperforming the relative month SSE Index during the backtesting period.

*In this paper, the time interval of factor evaluation is from April 30, 1997 to December 31, 2009 (only in-sample data).

And the result is shown in the form below.

	Factor	IC	IG	Advantage_Return	Disadvantage_Return	Advantage_Probility	Disadvantage_Probility
0	pe	0.075409	3.582215	0.007854	0.009522	0.522876	0.555556
1	pb	0.044445	3.581631	0.000683	0.017317	0.457516	0.581699
2	ps	0.062896	3.580316	0.001734	0.013361	0.464052	0.555556
3	pcf	0.105548	3.582991	0.006615	0.006642	0.464052	0.483660
4	Market_cap	0.067590	3.580823	0.001480	0.017265	0.503268	0.594771
5	turn_mean	0.033349	3.583614	-0.008739	0.011366	0.379085	0.549020
6	vol	0.054690	3.583049	-0.001500	0.010269	0.450980	0.549020
7	MA	0.030138	3.582053	-0.000263	0.014421	0.483660	0.549020
8	RSV	0.064922	3.581825	0.006393	0.011315	0.509804	0.496732
9	K	0.082756	3.583124	0.010704	0.011370	0.549020	0.490196
10	D	0.096759	3.582848	0.012032	0.010377	0.575163	0.483660
11	J	0.072999	3.583285	0.007732	0.009499	0.522876	0.477124
12	Pote	0.104039	3.579364	0.015378	0.008548	0.562092	0.542484
13	Mbp	0.114155	3.583570	0.007507	0.011940	0.568627	0.575163
14	Mbpr	0.112236	3.582560	0.010636	0.009984	0.509804	0.516340
15	Mbcr	0.095715	3.581339	0.010861	0.012871	0.509804	0.529412
16	Mbi	0.108657	3.582953	0.008770	0.012736	0.575163	0.581699
17	Mbigr	0.120549	3.584177	0.010911	0.008130	0.542484	0.555556
18	Mbpp	0.110926	3.583067	0.008063	0.011812	0.490196	0.588235
19	Er	0.096839	3.583634	0.009162	0.009232	0.529412	0.522876
20	Np	0.114199	3.586410	0.005066	0.013605	0.575163	0.581699
21	Pcnp	0.126510	3.582148	0.013192	0.011248	0.542484	0.535948
22	Npgr	0.123554	3.580002	0.010779	0.012964	0.496732	0.555556
23	Nagr	0.108118	3.581665	0.008261	0.009546	0.477124	0.542484
24	roe	0.119843	3.585666	0.009152	0.013090	0.509804	0.588235
25	wroe	0.124725	3.583076	0.011242	0.010240	0.503268	0.535948
26	ipm	0.111564	3.580974	0.009398	0.011177	0.503268	0.562092
27	tp	0.114871	3.581663	0.006333	0.011962	0.581699	0.575163
28	nfar	0.097667	3.581150	0.008315	0.010301	0.470588	0.509804
29	Fatr	0.110918	3.579100	0.008941	0.010435	0.509804	0.542484
30	Fap	0.111728	3.584481	0.012315	0.008415	0.542484	0.529412
31	Itd	0.096670	3.580333	0.008714	0.010948	0.549020	0.522876
32	Itr	0.110745	3.585293	0.010739	0.010138	0.516340	0.542484
33	Artd	0.091855	3.581039	0.007696	0.009200	0.516340	0.568627
34	Artr	0.115564	3.583174	0.009043	0.007693	0.575163	0.516340
35	Tl	0.096983	3.582053	0.006704	0.014485	0.496732	0.568627
36	Ta	0.098967	3.583364	0.006043	0.013264	0.542484	0.562092

37	Tanpr	0.121277	3.583763	0.010030	0.012487	0.509804	0.575163
38	Tapr	0.121243	3.583353	0.009923	0.012825	0.522876	0.575163
39	Tatd	0.085152	3.583123	0.007718	0.009076	0.509804	0.496732
40	Tatr	0.122056	3.582266	0.009104	0.010486	0.490196	0.549020
41	Tagr	0.107268	3.582538	0.010094	0.007027	0.490196	0.555556
42	Cem	0.112144	3.585019	0.010022	0.011425	0.562092	0.562092
43	Ii	0.098911	3.579826	0.005353	0.012779	0.444444	0.588235
44	Naps	0.116879	3.584298	0.006931	0.009323	0.496732	0.522876
45	Ncffoaps	0.100874	3.582459	0.012800	0.008152	0.549020	0.516340
46	Cr	0.108121	3.579415	0.008531	0.009535	0.509804	0.529412
47	Cl	0.096120	3.581186	0.007419	0.012849	0.503268	0.588235
48	Ca	0.095427	3.581049	0.007421	0.012755	0.522876	0.549020
49	Catd	0.086852	3.582203	0.008221	0.010530	0.529412	0.535948
50	Catr	0.120477	3.581311	0.010678	0.008138	0.549020	0.522876
51	Lvr	0.112123	3.583373	0.008926	0.010416	0.542484	0.562092
52	Niicace	0.104799	3.581058	0.012905	0.007880	0.555556	0.516340
53	Cashr	0.111653	3.583390	0.009493	0.007773	0.529412	0.529412
54	Cfr	0.117186	3.583280	0.012184	0.005536	0.588235	0.503268
55	Ncffoa	0.113744	3.581963	0.010810	0.007384	0.555556	0.535948
56	ronocftnp	0.112264	3.582122	0.009468	0.004945	0.522876	0.529412
57	ronocftd	0.119016	3.580942	0.012787	0.007926	0.588235	0.509804
58	ronocftsr	0.112817	3.582293	0.010096	0.005744	0.555556	0.516340
59	Sdnims	0.101419	3.585174	0.005602	0.014015	0.490196	0.568627
60	Stfar	0.101617	3.581865	0.008347	0.010729	0.503268	0.529412
61	Sr	0.111584	3.582183	0.009493	0.008549	0.529412	0.503268
62	Roe	0.113511	3.582250	0.012972	0.008974	0.575163	0.581699
63	Op	0.116343	3.585684	0.006178	0.011862	0.542484	0.542484
64	Opr	0.112899	3.582463	0.011140	0.012957	0.562092	0.575163
65	Nniae	0.100209	3.580110	0.009895	0.008741	0.535948	0.503268
66	Der	0.096207	3.581246	0.008437	0.009417	0.522876	0.509804
67	Roa	0.110156	3.578872	0.012563	0.010107	0.568627	0.542484
68	Roocfoa	0.116075	3.581232	0.012756	0.007172	0.627451	0.522876
69	Ar	0.095765	3.581658	0.008301	0.009555	0.509804	0.516340
70	Capr	0.096743	3.583087	0.010418	0.009575	0.529412	0.549020
71	Cir	0.101634	3.582431	0.008611	0.008029	0.535948	0.490196
72	qr	0.106782	3.580642	0.008648	0.011078	0.509804	0.535948
73	snpm	0.110944	3.585960	0.010012	0.012986	0.555556	0.575163
74	sgpm	0.111977	3.583438	0.011306	0.010842	0.542484	0.516340
75	ldtwcr	0.101451	3.580422	0.010522	0.011082	0.477124	0.549020
76	ldr	0.095966	3.579693	0.011502	0.014006	0.555556	0.555556
77	Latlcr	0.102316	3.581401	0.009136	0.007586	0.529412	0.483660
78	Nbp	0.097971	3.582529	0.009204	0.013178	0.529412	0.555556

We can see that the differences between information gain are very small. In addition, IC is used for linear relationship measurement. Considering that our subsequent model needs to use nonlinear machine learning, we decide to screen the factors through the return and outperformance probability at first.

Also, we assume the difference in return between the advantage and disadvantage group reflects a factor's profit potential, and the difference in outperformance probability between the advantage and disadvantage group reflects the stability of a factor's abnormal returns. Therefore, we only keep the factors with advantage group returns greater than disadvantage group returns and advantage group outperformance probability greater than the disadvantage group outperformance probability.

And the result is shown in the form below.

	Factor	IC	IG	Advantage_Return	Disadvantage_Return	Advantage_Probility	Disadvantage_Probility
0	D	0.096759	3.582848	0.012032	0.010377	0.575163	0.483660
1	Pote	0.104039	3.579364	0.015378	0.008548	0.562092	0.542484
2	Pcnp	0.126510	3.582148	0.013192	0.011248	0.542484	0.535948
3	Fap	0.111728	3.584481	0.012315	0.008415	0.542484	0.529412
4	Artr	0.115564	3.583174	0.009043	0.007693	0.575163	0.516340
5	Ncffoaps	0.100874	3.582459	0.012800	0.008152	0.549020	0.516340
6	Catr	0.120477	3.581311	0.010678	0.008138	0.549020	0.522876
7	Niicace	0.104799	3.581058	0.012905	0.007880	0.555556	0.516340
8	Cfr	0.117186	3.583280	0.012184	0.005536	0.588235	0.503268
9	Ncffoa	0.113744	3.581963	0.010810	0.007384	0.555556	0.535948
10	ronocftd	0.119016	3.580942	0.012787	0.007926	0.588235	0.509804
11	ronocftsr	0.112817	3.582293	0.010096	0.005744	0.555556	0.516340
12	Sr	0.111584	3.582183	0.009493	0.008549	0.529412	0.503268
13	Nniae	0.100209	3.580110	0.009895	0.008741	0.535948	0.503268
14	Roa	0.110156	3.578872	0.012563	0.010107	0.568627	0.542484
15	Roocfoa	0.116075	3.581232	0.012756	0.007172	0.627451	0.522876
16	Cir	0.101634	3.582431	0.008611	0.008029	0.535948	0.490196
17	sgpm	0.111977	3.583438	0.011306	0.010842	0.542484	0.516340
18	Latlcr	0.102316	3.581401	0.009136	0.007586	0.529412	0.483660

Among the remaining 19 factors, we hope to select the factors with higher IC and Information Gain. Therefore, we score the sum of the orders for IC and Information Gain(IG).

	Factor	IC	IG	Advantage_Return	Disadvantage_Return	Advantage_Probility	Disadvantage_Probility	Score
0	Cfr	15	16	0.012184	0.005536	0.588235	0.503268	31

1	Artr	13	15	0.009043	0.007693	0.575163	0.516340	28
2	Pcnp	18	9	0.013192	0.011248	0.542484	0.535948	27
3	sgpm	10	17	0.011306	0.010842	0.542484	0.516340	27
4	Fap	9	18	0.012315	0.008415	0.542484	0.529412	27
5	Catr	17	6	0.010678	0.008138	0.549020	0.522876	23
6	ronocftsr	11	11	0.010096	0.005744	0.555556	0.516340	22
7	Ncffoa	12	8	0.010810	0.007384	0.555556	0.535948	20
8	Roocfoa	14	5	0.012756	0.007172	0.627451	0.522876	19
9	ronocftd	16	3	0.012787	0.007926	0.588235	0.509804	19
10	Sr	8	10	0.009493	0.008549	0.529412	0.503268	18
11	Cir	3	12	0.008611	0.008029	0.535948	0.490196	15
12	Ncffoaps	2	13	0.012800	0.008152	0.549020	0.516340	15
13	D	0	14	0.012032	0.010377	0.575163	0.483660	14
14	Latlcr	4	7	0.009136	0.007586	0.529412	0.483660	11
15	Niicace	6	4	0.012905	0.007880	0.555556	0.516340	10
16	Roa	7	0	0.012563	0.010107	0.568627	0.542484	7
17	Pote	5	1	0.015378	0.008548	0.562092	0.542484	6
18	Nniae	1	2	0.009895	0.008741	0.535948	0.503268	3

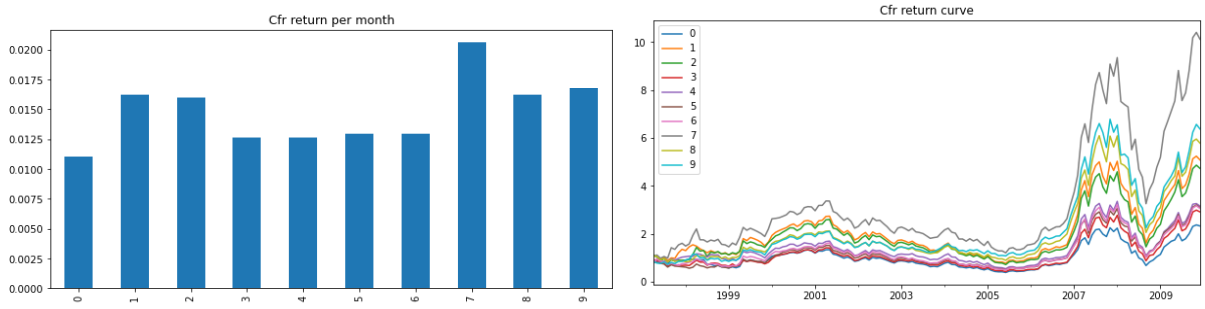
Then, we delete the factors whose score is below the average(which is 18). And the remaining 10 factors are shown as following:

	Feature	**IC**	**Gain**	**Advantaged_Return**	**Disadvantaged_Return**	**Advantaged_Probility**	**Disadvantaged_Probility**	**Score**
0	Cfr	15	16	0.012184	0.005536	0.588235	0.503268	31
1	Artr	13	15	0.009043	0.007693	0.575163	0.516340	28
2	Pcnp	18	9	0.013192	0.011248	0.542484	0.535948	27
3	sgpm	10	17	0.011306	0.010842	0.542484	0.516340	27
4	Fap	9	18	0.012315	0.008415	0.542484	0.529412	27
5	Catr	17	6	0.010678	0.008138	0.549020	0.522876	23
6	ronocftsr	11	11	0.010096	0.005744	0.555556	0.516340	22
7	Ncffoa	12	8	0.010810	0.007384	0.555556	0.535948	20
8	Roocfoa	14	5	0.012756	0.007172	0.627451	0.522876	19
9	ronocftd	16	3	0.012787	0.007926	0.588235	0.509804	19

We will use the above 10 factors in our Adaboost model.

Factors Analysis

Cash Flow Ratio



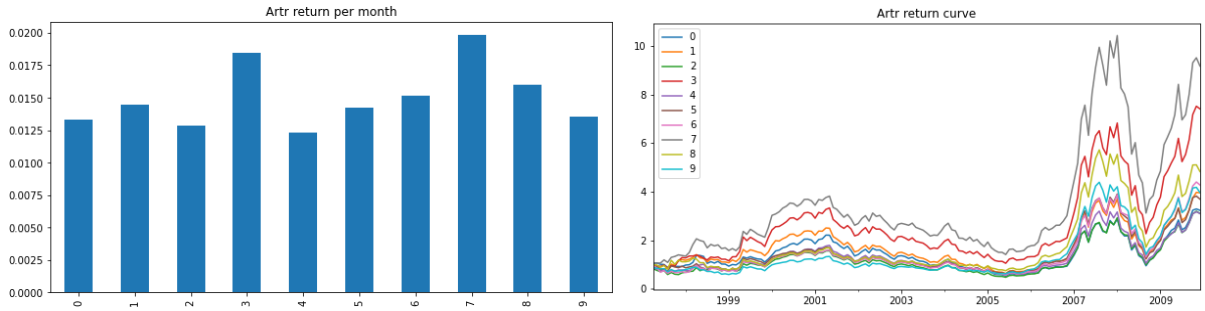
*From the average monthly return plot(left) and stratified return plot(right) above, 0 is the disadvantage group and 9 is the advantage group of the factor return.

It can be seen that the relationship between the factor and the return is not purely linear, and there is a potential non-linear correlation. For example, for the CFR factor above, the highest return is group 7, but the overall returns of 7, 8 and 9 are higher than the others. This is more or less the case for the other 9 factors.

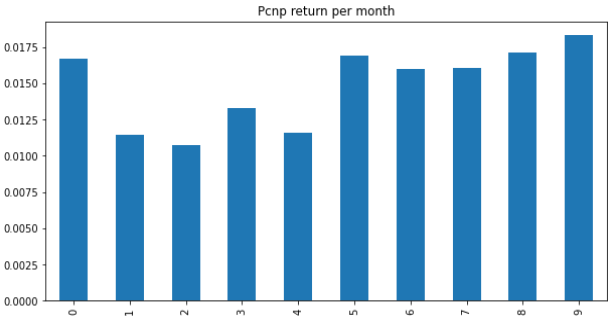
Although the advantage group(9) is not the one with the highest return, if the groups adjacent to the advantage group are considered as a whole, this whole group has comparative advantage compared with the other groups. Thus, we plan to design a classification model to identify high-return stocks. Specifically, groups 0, 1 and 2 are designated as the low return category, groups 3, 4, 5 and 6 are designated as the medium return category, and groups 7, 8 and 9 are designated as the high return category. Thus, we decide to use an Adaboost algorithm model in our strategy.

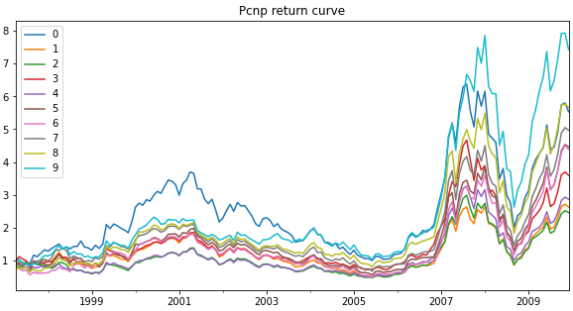
The plots of the remaining 9 factors during the in-sample period are shown below.

Accounts Receivable Turnover Rate

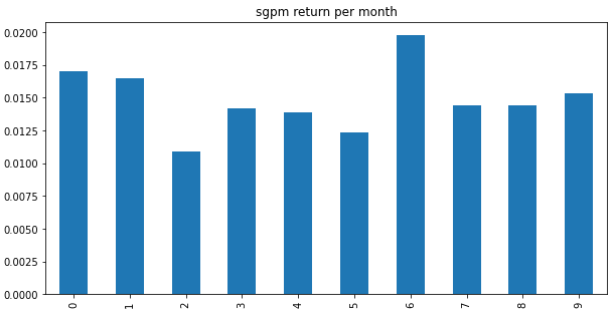


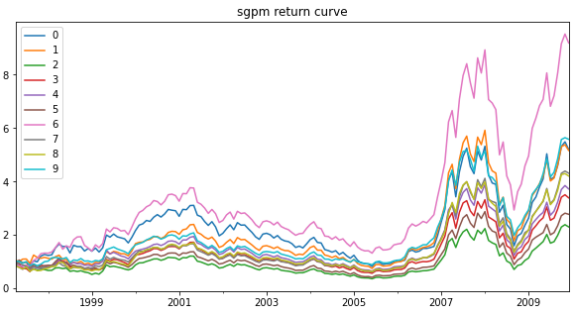
Parent Company Net Profit



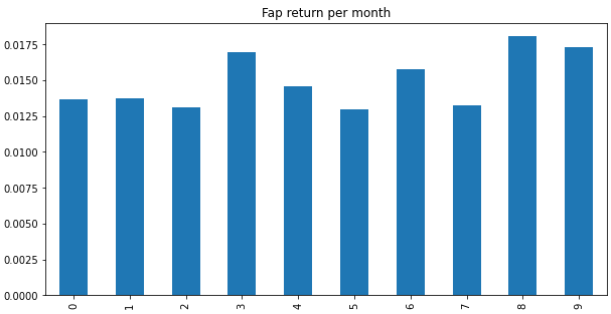


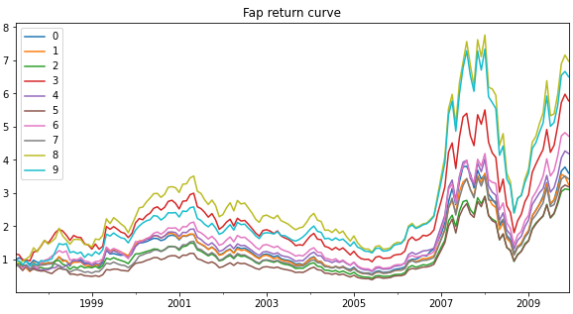
Sales Gross Profit Margin



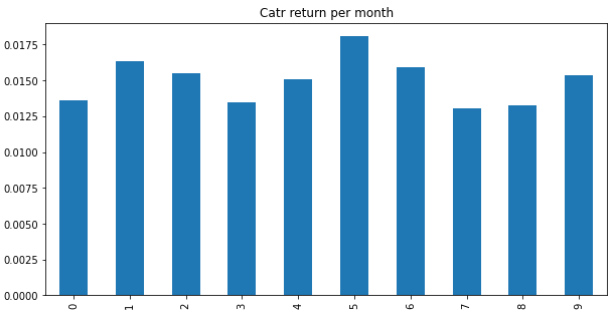


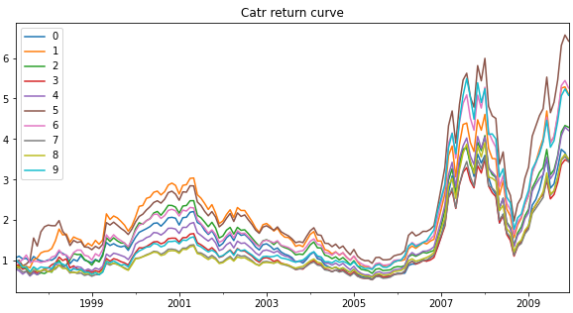
Fixed Assets Proportion



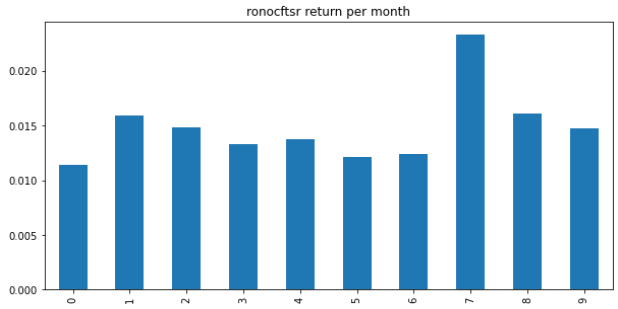


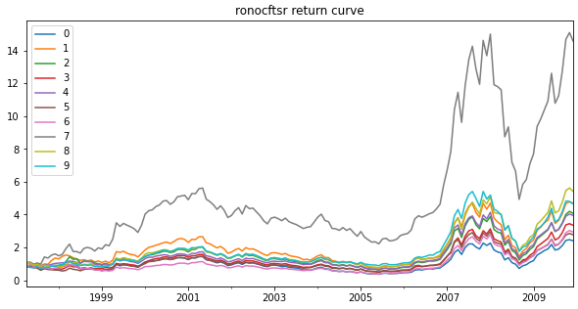
Current Asset Turnover Rate



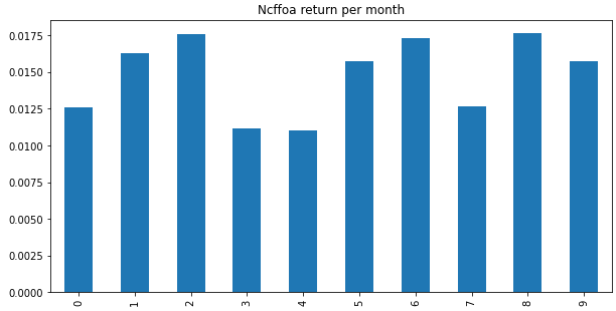


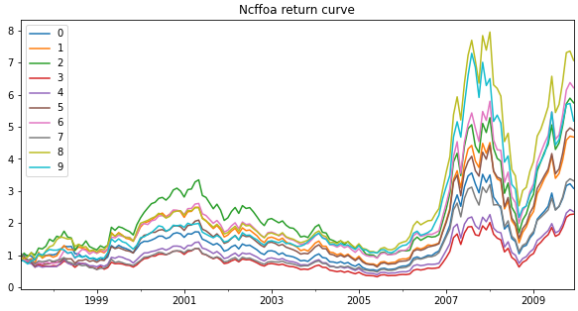
Ratio of Net Operating Cash Flow to Sales Revenue



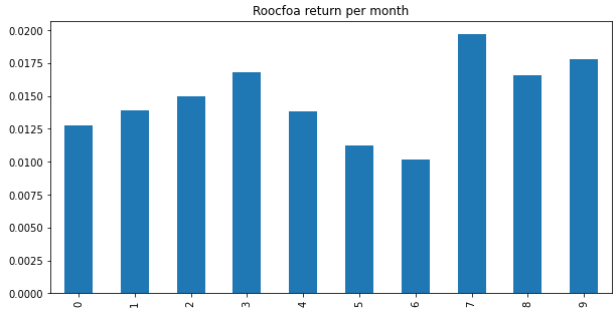


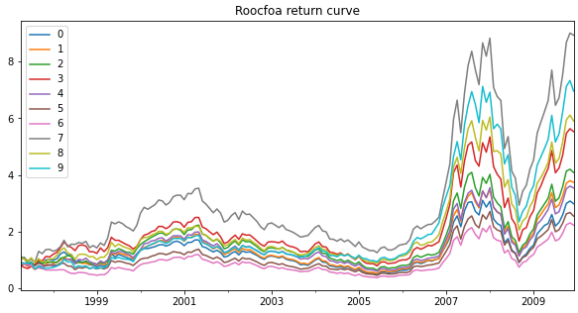
Net Cash Flow from Operating Activities



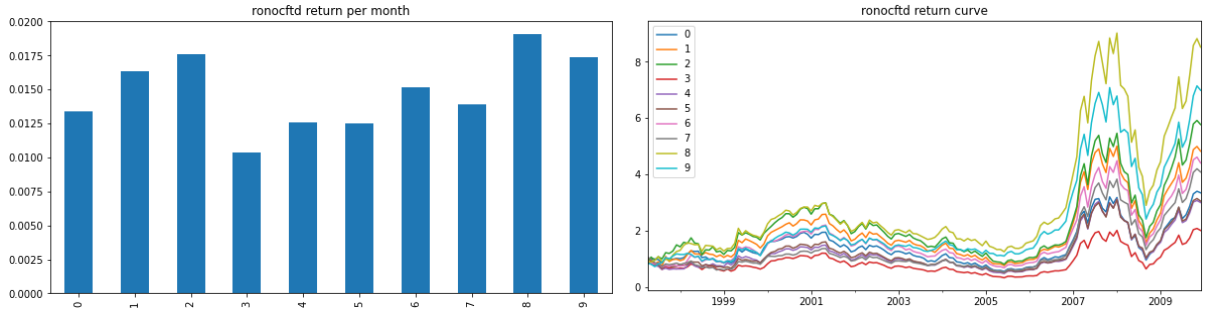


Return on Operating Cash Flow of Assets





Ratio of Net Operating Cash Flow to Debt



Methodology

Adaptive Boosting(Adaboost)

Boosting method refers to a group of algorithms which converts weak learners to a powerful learner. It is a class of ensemble machine learning algorithms that combine a large number of classifiers to produce more accurate and robust predictions.

Adaboost can be used to enhance the classification function of the weak classifier(Dhagat A & Hellerstein L, 1994), and has more efficient performance in the problem of classification. The concept of AdaBoost revolves around correcting previous classifier mistakes. Each classifier gets trained on the sample set and learns to predict. The misclassification errors are then fed into the next classifier in the chain and are used to correct the mistakes until the final model predicts accurate results.

AdaBoost combines the predictions from short one-level decision stumps. It allows us to capture many of these non-linear relationships, which translates into better prediction accuracy on finding the best stocks to purchase.

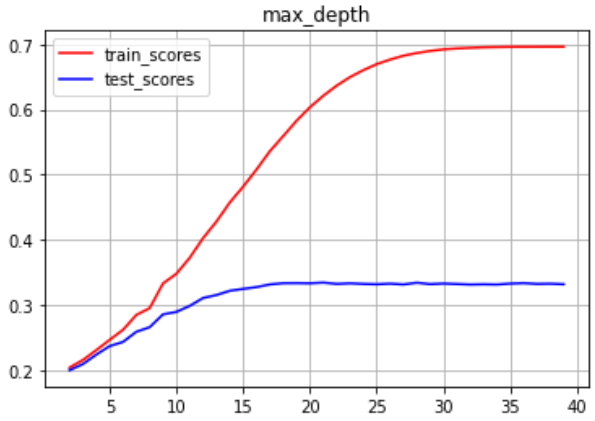
Decision Tree Classifier

As Adaboost is not a classifier itself, so, in order to use Adaboost to enhance classification algorithms, a weak classifier algorithm is required. And we chose decision trees as our classifier. The goal of using decision trees is to create a training model that can be used to predict the class of the target variable by learning simple decision rules inferred from training dataset.

Hyperparameter Optimization

Decision Tree Learning Curve

Maximum depth of a decision tree is described as the length of the longest path from the tree root to a leaf. As maximum depth is a very important hyperparameter in decision trees, we tune the depth of the decision tree at first. We then set scoring = f1_macro in order to maximize the Macro-F1 score and receive the learning curves as shown.



As the depth of the decision tree increases, the training score and the test score are increasing. It can be seen from the plot that the learning curve approaches flatness after depth reaching 15. Therefore, we determine the maximum depth of the decision tree is approximately between 10 and 30.

Decision Tree Grid Search

We tune the hyperparameters of the decision trees using the grid search infrastructure in sklearn. Our goal is to maximize the number of points that are correctly classified in the training set. We evaluate the minimum number of samples in a leaf(min_samples_leaf) ranging from 5 to 50 with a step size of 5, and the minimum number of samples required to split an internal node(min_samples_split) ranging from 10 to 100 with a step size of 10.

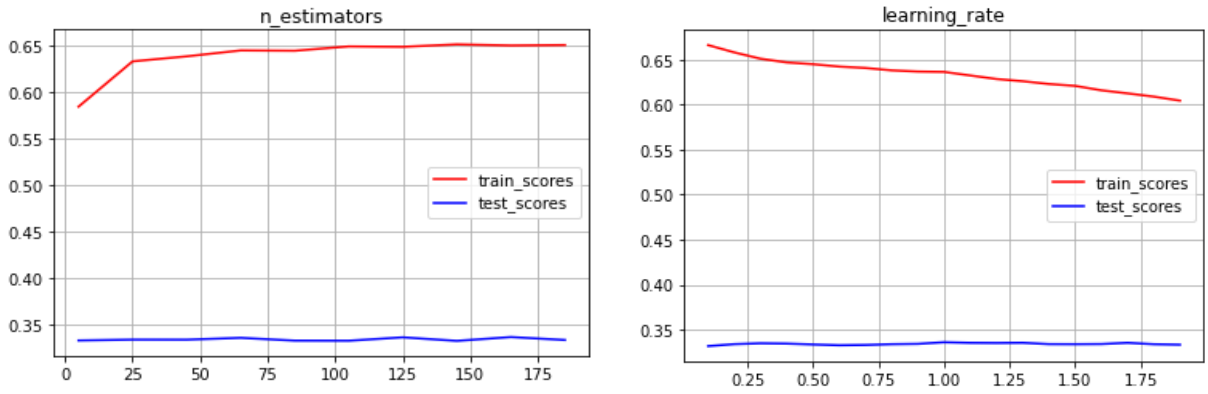
```
GridSearchCV(cv=3, estimator=DecisionTreeClassifier(criterion='entropy'),
             n_jobs=-1,
             param_grid={'max_depth': range(10, 30),
                         'min_samples_leaf': range(5, 50, 5),
                         'min_samples_split': range(10, 100, 10)},
             scoring='f1_macro', verbose=1)
```

```
DecisionTreeClassifier(criterion='entropy', max_depth=26, min_samples_leaf=5,
                       min_samples_split=20)
```

The best result was achieved with a max_depth = 26, min_samples_leaf value = 5 and min_samples_split = 20.

AdaBoost Hyperparameters Learning Curve

After finishing tuning the hyperparameters of decision trees, we continue discussing the main hyperparameters of AdaBoost algorithm. The learning curves for the hyperparameters n_estimators and learning_rate are shown as follows.



We can see that the performance does not improve as the number of training points increases for both hyperparameters. So we continue using GridsearchCV to search for optimal values.

Hyperparameter Grid Search in AdaBoost

We tune the size of the hyperparameters in AdaBoost using the grid search infrastructure like before. We evaluate n_estimators value ranging from 10 to 50 with a step size of 10 and learning_rate value ranging from 0.1 to 1.5 with a step size of 0.2.

```
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 4 concurrent workers.
[Parallel(n_jobs=-1)]: Done  42 tasks      | elapsed:  2.1min
[Parallel(n_jobs=-1)]: Done  84 out of  84 | elapsed:  4.3min finished
```

```
GridSearchCV(cv=3,
             estimator=AdaBoostClassifier(base_estimator=DecisionTreeClassifier(criterion='entropy',
                                                                                max_depth=26,
                                                                                min_samples_leaf=5,
                                                                                min_samples_split=20)),
             n_jobs=-1,
             param_grid={'learning_rate': array([0.1, 0.3, 0.5, 0.7, 0.9, 1.1, 1.3]),
                         'n_estimators': range(10, 50, 10)},
             scoring='f1_macro', verbose=1)
```

```
{'learning_rate': 0.1, 'n_estimators': 10}
```

The search result shows that n_estimators is 10 and learning_rate is 0.1, both of which are taken at the lower boundaries. So we reset the value of n_estimators ranging from 5 to 15 with a step size of 1 and learning_rate from 0.01 to 0.2 with a step size of 0.02 for a more precise search.

```
Fitting 3 folds for each of 100 candidates, totalling 300 fits


[Parallel(n_jobs=-1)]: Using backend LokyBackend with 4 concurrent workers.
[Parallel(n_jobs=-1)]: Done  42 tasks      | elapsed:   43.5s
[Parallel(n_jobs=-1)]: Done 192 tasks      | elapsed:  3.4min
[Parallel(n_jobs=-1)]: Done 300 out of 300 | elapsed:  5.4min finished


GridSearchCV(cv=3,
             estimator=AdaBoostClassifier(base_estimator=DecisionTreeClassifier(criterion='entropy',
                                                                                max_depth=26,
                                                                                min_samples_leaf=5,
                                                                                min_samples_split=20)),
             n_jobs=-1,
             param_grid={'learning_rate': array([0.01, 0.03, 0.05, 0.07, 0.09, 0.11, 0.13, 0.15, 0.17, 0.19]),
                         'n_estimators': range(5, 15)},
             scoring='f1_macro', verbose=1)



{'learning_rate': 0.16999999999999998, 'n_estimators': 7}
```

Finally, the best result was achieved with n_estimators = 7 and learning_rate = 0.17.

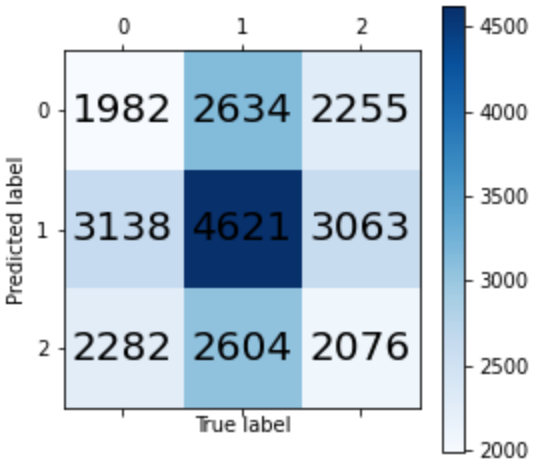
Classification Report

In our Adaboost model, we split 80% of the in-sample data into training data, and the rest 20% becomes test data for validation. The classification report is shown in the table below.

	precision	recall	f1-score	support
0	0.29	0.27	0.28	7402
1	0.43	0.47	0.45	9859
2	0.30	0.28	0.29	7394
accuracy			0.35	24655
macro avg	0.34	0.34	0.34	24655
weighted avg	0.35	0.35	0.35	24655

Adaboost outputs a classification(Class 0, Class1 and Class 2) for each of our 4215 stocks. Class 0 is the low return stocks(last 30%), Class 1 is the medium return stocks(middle 40%) and Class 2 is the high return stocks(first 30%). We only trade the stocks in Class 2, which are the highest ranked stocks classified from AdaBoost algorithm.

Confusion Matrix



From the classification report and confusion matrix, we can see that Class 1 gives the best performance in fitting our model followed by Class 2. And the classification accuracy of our model is 35%. Improvement in classification accuracy will lead to better performance of our model, and we will talk about methods of further improvement in the last part of our paper.

Factor Importance

	Feature	Importance
0	Artr	0.127337
1	sgpm	0.124481
2	Fap	0.115233
3	Catr	0.114673
4	Pcnp	0.099227
5	Roocfoa	0.098229
6	Ncffoa	0.090334
7	ronocftsr	0.085468
8	ronocftd	0.079500
9	Cfr	0.065518

We can have a look at the importance of each factor in the table above. AdaBoost feature importance is determined by the average feature importance provided by each Decision Tree. Accounts receivable turnover rate(Artr), sales gross profit margin(sgpm) and fixed assets proportion(Fap) give the highest feature importance.

Portfolio Allocation

We have chosen to rebalance the portfolio at the end of every month. When rebalancing our portfolio, the implemented model is calibrated to only consider business days. In the case of a holiday, the nearest business day is chosen for the calculations. And we simply set equal weighting between stocks.

To determine the model's effectiveness, it will be back-tested over a 12 years period from 1997-4-30 to 2009-12-31. And then we evaluate the performance of our strategy using out of sample data from 2010-1-31 to 2020-8-31.

Risk Management and Transaction Costs

Our trading strategy used a standard 10% stop-loss rule to mitigate loss. When more than 10% of the initial asset value is lost, we would sell the stock.

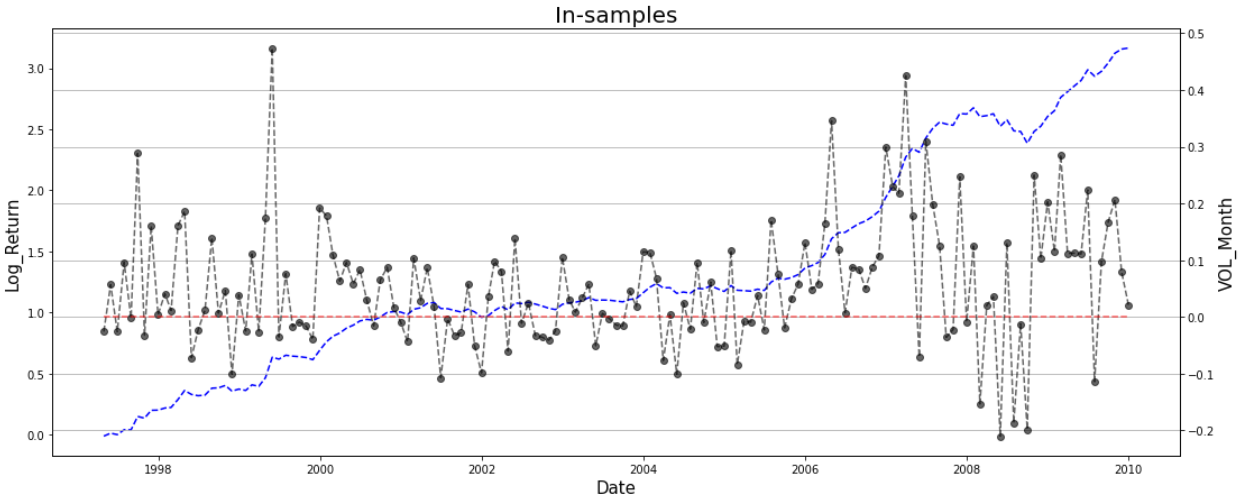
The current major costs of A-share transactions include 0.1% stamp duty, brokerage trading commissions ranging from 0.01% to 0.02%, securities management fee and transfer fees etc. And we applied a constant 0.125% fee to all transactions.

Portfolio Performance Evaluation

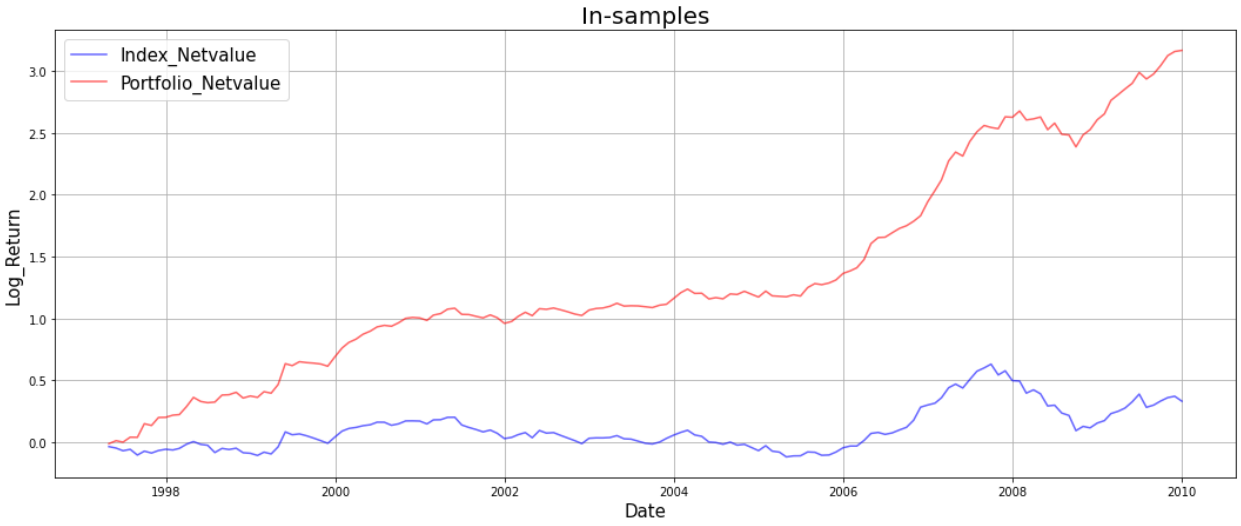
Backtesting with In-Sample Data

- In-sample data is from 1997-4-30 to 2009-12-31. As our project has a long backtesting period. During this period, the risk-free rate has been declined significantly in China. Therefore, we did not set risk-free rate in calculating the Sharpe ratio.
- Alpha measures the amount that the investment has returned in comparison to our benchmark SSE index.
- Beta measures the volatility of our strategy, and it is an indication of relative risk.
- Percent Winners: ratio of the number of months with positive returns to the total number of months.
- Average Winner: average returns for the number of months with positive returns.
- Best Winner: maximum return for a single month.
- Average Loser: average losses for the number of months with negative returns.
- Worst Loser: maximum drawdown for a single month.

Sharpe	Sortino	Annualized Return	Cumulative Return	Beta	Alpha	Percent Winners	Average Winner	Best Winner	Average Loser	Worst Loser
1.71	3.80	77.74%	146227.13%	1.09	0.04	64.71%	0.110304	0.474109	-0.048464	-0.21057



- Blue Line: Log Return
- Black Line: Monthly Volatility
- Red Line: Zero Volatility

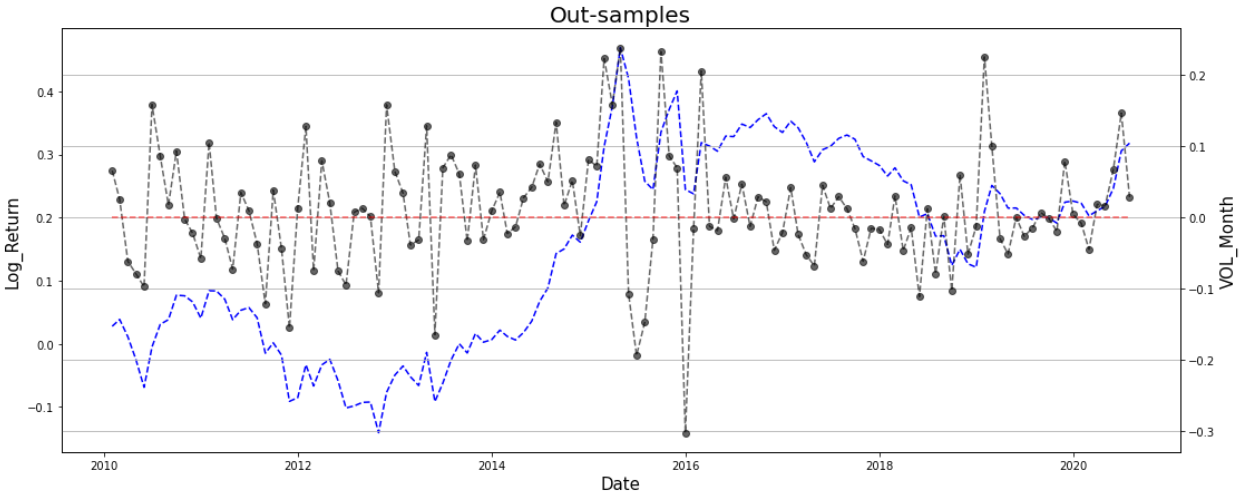


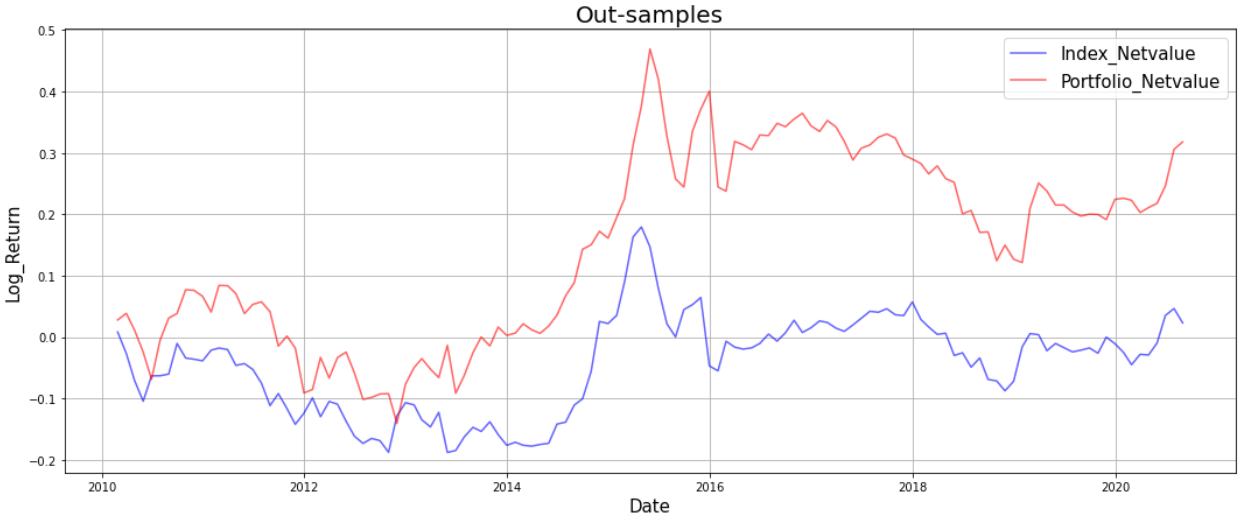
This graph mainly shows the comparison between the log return of our strategy and SSE index from 1997-4-30 to 2009-12-31. Our portfolio generates a Sharpe ratio of 1.71 and an annualized return of 77.74%. The Sharpe ratio of the benchmark SSE index during the same period is 0.35. Our strategy gives a good performance during the in-sample period.

It can be seen from the second plot that when the index is volatile, our strategy will basically rise and fall in line with the index. In addition, the strategy portfolio is likely to show an advantage from late 2008 to early 2010. Market index dropped significantly after late 2007 might be caused by a combination of 2008 global financial crisis, collapsing in oil price and Great Sichuan Earthquake in China.

Out of Sample Performance Analysis

Sharpe	Sortino	Annualized Return	Cumulative Return	Beta	Alpha	Percent Winners	Average Winner	Best Winner	Average Loser	Worst Loser
0.38	0.59	7.157%	107.88%	0.12	0.009	52.76%	0.066885	0.238387	-0.05499	-0.301926





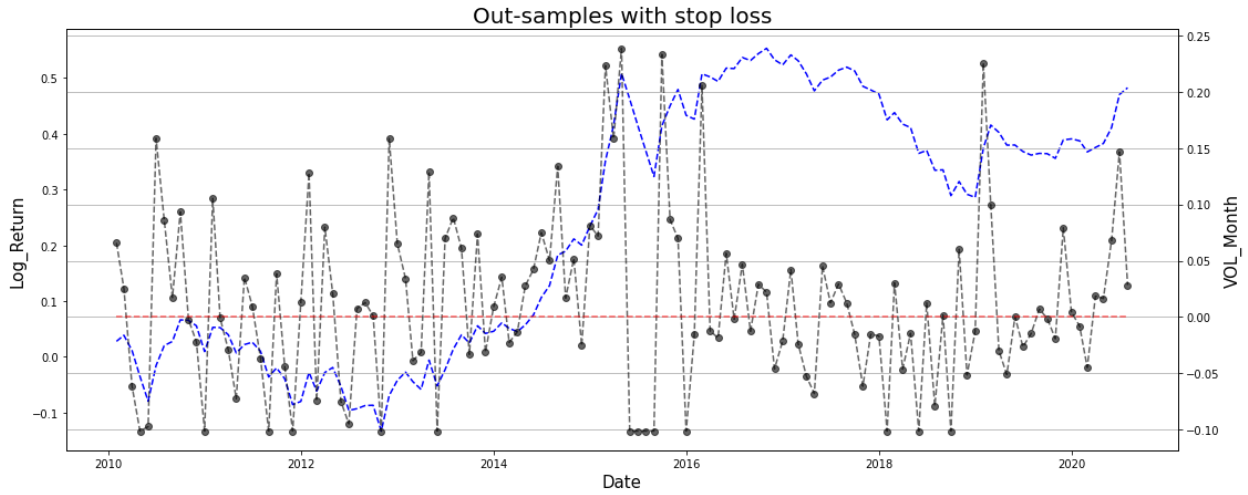
These two graphs show the comparison between the log return of our strategy and SSE index from 2010-1-31 to 2020-8-31. Our portfolio generates a Sharpe ratio of 0.38 and an annualized return of 7.16%. SSE index generated a Sharpe ratio of 0.20 over the same period. Our strategy could easily beat the market index during the out-of-sample period.

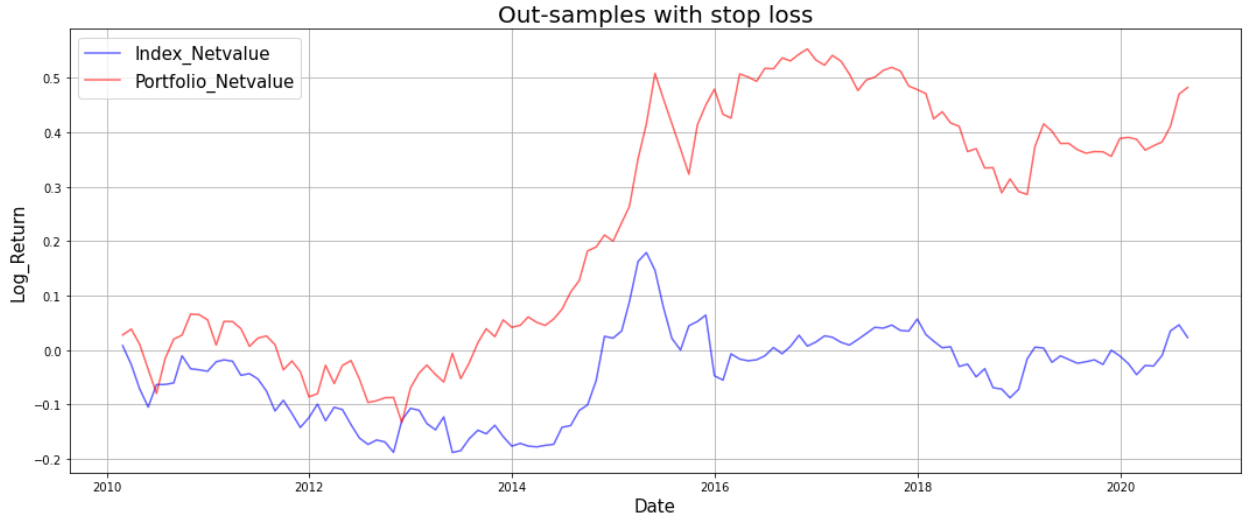
We can see that when the index is volatile, the strategy will basically follow the trend of the index. And the red line has some degree of hysteresis compared with the index. From June 2015, Chinese stock market turbulence began due to the popping of the stock market bubble, and a

third of the value of A-shares on the Shanghai Stock Exchange was lost within one month. Values of the A-share stock market continued to drop despite efforts made by Chinese government, and the index reached the largest fall since 2007.

Out of Sample Performance Analysis with 10% Stop-Loss

Sharpe	Sortino	Annualized Return	Cumulative Return	Beta	Alpha	Percent Winners	Average Winner	Best Winner	Average Loser	Worst Loser
0.52	1.12	11.068%	203.83%	0.077	0.011	52.76%	0.066885	0.238387	-0.050025	-0.101125





Adding risk management by 10% stop-loss gives better performance. As we can see from the tables above, the maximum drawdown decreases from -0.30 to -0.10. The Sharpe ratio increases from 0.38 to 0.52 and annualized return rises from 7.16% to 11.07%, both of which give a better value than without stop-loss. The result indicates that the overall effect of our strategy is good, and especially in late 2016 and early 2017.

Conclusion

Multi factor stock selection model heavily relies on the choice of factors and to carefully select and evaluate these are a key activity in our project. Our final model consists of 10 factors: cash flow ratio, accounts receivable turnover rate, parent company net profit, sales gross profit margin, fixed assets proportion, current asset turnover rate, ratio of net operating cash flow to sales revenue, net cash flow from operating activities, return on operating cash flow of assets and ratio of net operating cash flow to debt.

Our model is tested against our benchmark SSE Index, and the result shows that our strategy yields an abnormal return relative to the market. We can summarize that multi factor model for stock selection provides an understandable way to communicate investment themes and valuable information.

Evaluation and Future Improvements

10 factors information are incorporated into the Adaboost algorithm, and the result shows that AdaBoost algorithm is effective in selecting stocks. However, we found that our AdaBoost algorithm is not good enough because the accuracy rate of the test data is not satisfying.

For the improvement of the model, the factors can be selected according to larger classes, which can further improve the performance of our model. Moreover, our model did not take into account general neutral problems. Market neutral is a direction that can be further improved.

In addition, targeting the shortcomings of this stock selection model, we could distinguish and classify stocks into different industries, and then choose proper factors and conduct model training regarding a specific industry in order to obtain better classification accuracy and results.

The classification method used in our model is decision tree, and in fact, the classification result is highly related to the classifier. Therefore, a better classifier can expect to get better backtesting results. The enhancement of Adaboost for other classifiers may further improve the effectiveness of stock selection. For example, nearest neighbor classifier, neural network, support vector machine and others. The improvement of classifier algorithms can provide new ideas for our strategy.

References

Dhagat, A., & Hellerstein, L. (1994). PAC learning with irrelevant attributes. Symposium on Foundations of Computer Science. IEEE Computer Society, 64-74. Retrieved December 18, 2020, from https://doi.org/10.1109/SFCS.1994.365704

Yan, Y., & Ding, X. Q. (2008). Improved AdaBoost algorithm based on multi-step correction. Journal of Tsinghua University: Natural Science Edition, 1613-1616. Retrieved December 18, 2020.

Zhang, R., & Cao, T. (2018). Multi-factor Stock Selection Model Based on Adaboost. Business and Economic Research. Macrothink Institute. Retrieved December 18, 2020, from http://www.macrothink.org/journal/index.php/ber/article/view/13942

S. Yutong and H. Zhao, "Stock selection model based on advanced AdaBoost algorithm," 2015 7th International Conference on Modelling, Identification and Control (ICMIC), Sousse, 2015, pp. 1-7, doi: 10.1109/ICMIC.2015.7409386.