Signal mining based on machine learning

Team Member

Student Name	Student ID
Li panyu	2001212358
Li Linxiong	2001212357
Hu Xiaoyin	2001212340

Project Introduction

In the past 10 years, deep learning models based on neural networks have led the development of artificial intelligence. Different from traditional machine learning, deep learning models directly extract features from raw data and make predictions for targets in an `end-to-end` manner, thereby avoiding manual intervention and information loss in multi-step learning. However, when deep learning is applied to `multi-factor stock selection`, the effect of applying existing models may not meet expectations, and a suitable network structure needs to be tailored.
In order to integrate the factor generation and multi-factor synthesis steps in multi-factor stock selection, this project designs two types of network structure to `predict the rise or fall` of each stock over the next 10 days inspired by Huatai Research report (see here): the first is `AlphaNet` using `raw price-volume data` as input, the other is using `price-volume data after feature extraction`, including: CNN, LogisticRegression and Random Forest.

Data Analysis

1. Data Source

- Stock pool: CSI 300 component stocks from 08/10/2010 to 05/11/2021
- Data: volume and price data of individual stocks without feature engineering, transfer the volume and price data into 9*30 `data pictures`, and 30 is the number of historical days
- Labels: rise or fall of each stock over the next 10 days

Raw data sample

	t-30	t-29	t-28	t-27	t-26	t-25	t-24	t-23
open	0.000	17.320	18.060	17.990	18.650	18.200	19.400	19.120
high	17.340	18.360	18.150	18.550	19.290	19.380	20.060	19.630
low	16.500	17.320	17.800	17.970	18.310	18.180	19.100	18.830
close	17.220	18.040	17.970	18.540	18.340	19.150	19.170	19.550
volume	53671585.000	87403016.000	47759448.000	82596178.000	100374925.000	116893821.000	107408195.000	58071926.000
pct_chg	6.165	4.762	-0.388	3.172	-1.079	4.417	0.104	1.982
vwap	17.045	17.899	17.942	18.335	18.820	18.996	19.578	19.078
turn	1.728	2.815	1.538	2.660	3.232	3.764	3.459	1.870
free_turn_n	2.077	3.383	1.848	3.197	3.885	4.524	4.157	2.247

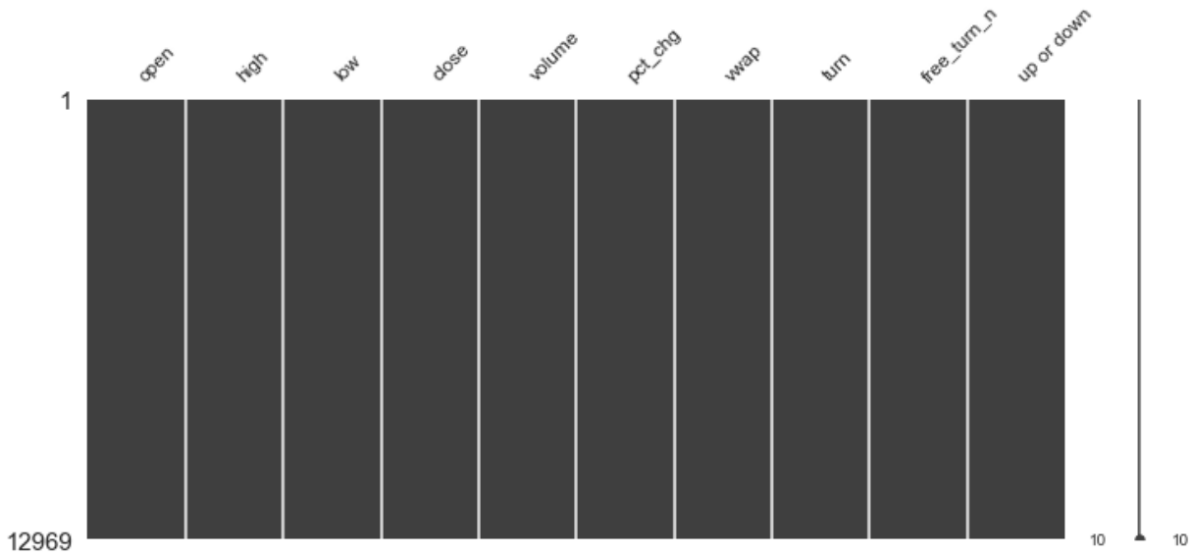
Data sample after feature extraction

	feature1	feature2	feature3	feature4	feature5	feature6	feature7	feature8	feature9	feature10	...	feature126
t-21	3.876	3.895	3.193	48637882.616	-10.803	3.616	1.566	1.882	0.714	0.613	...	3.033
t-20	0.399	0.397	0.281	-2259042.035	-1.263	0.353	-0.073	-0.087	0.364	0.331	...	3.033
t-19	0.277	0.235	0.166	-286609.252	-0.668	0.219	-0.009	-0.011	0.265	0.257	...	2.876
t-18	0.187	0.181	0.089	-3178984.853	-0.780	0.143	-0.102	-0.123	0.175	0.142	...	2.855
t-17	0.118	0.131	0.064	-1259481.073	-0.481	0.090	-0.041	-0.049	0.128	0.120	...	2.688

2. Data processing

- Deal with missing data: open, high, low, close fill with the previous data; volume, pct_chg, vwap, turn, free_turn_n use 0 to fill NAN. After processing, the data has no NAN values.
- One-hot encoding: One-hot encoder is used to convert categorical data into integer data.
- Feature Standardization： to avoid unit effect, we normalized the data along the feature direction

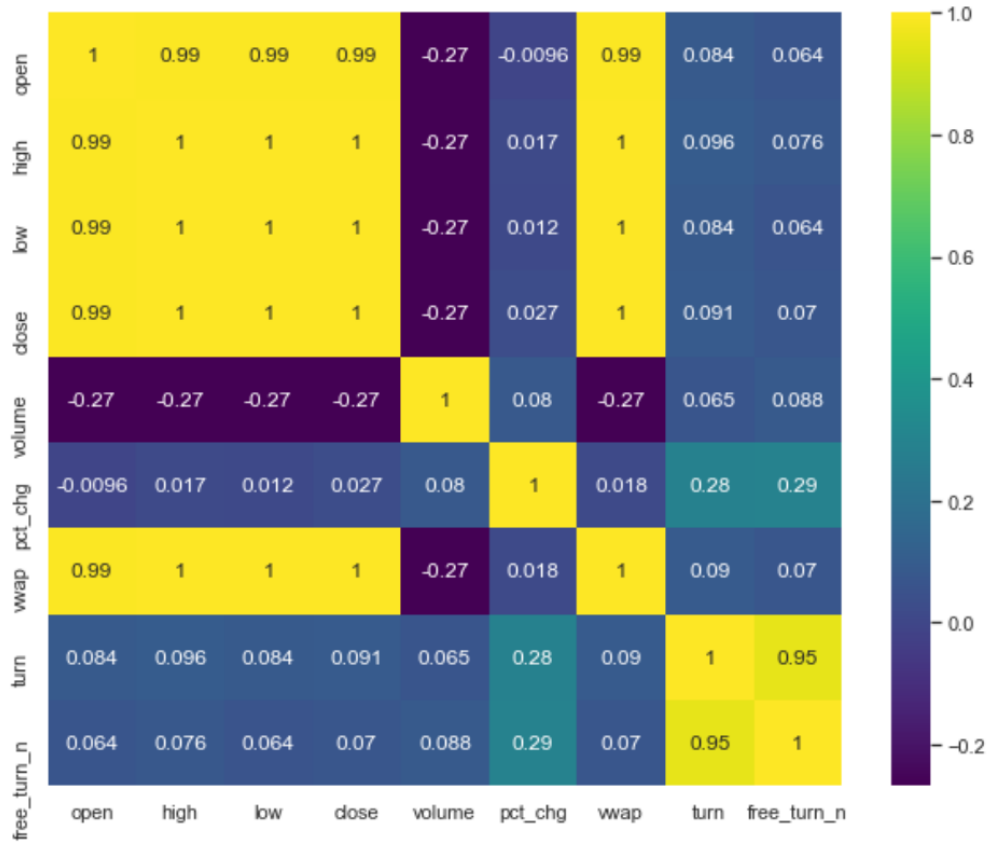
Raw data after dropna



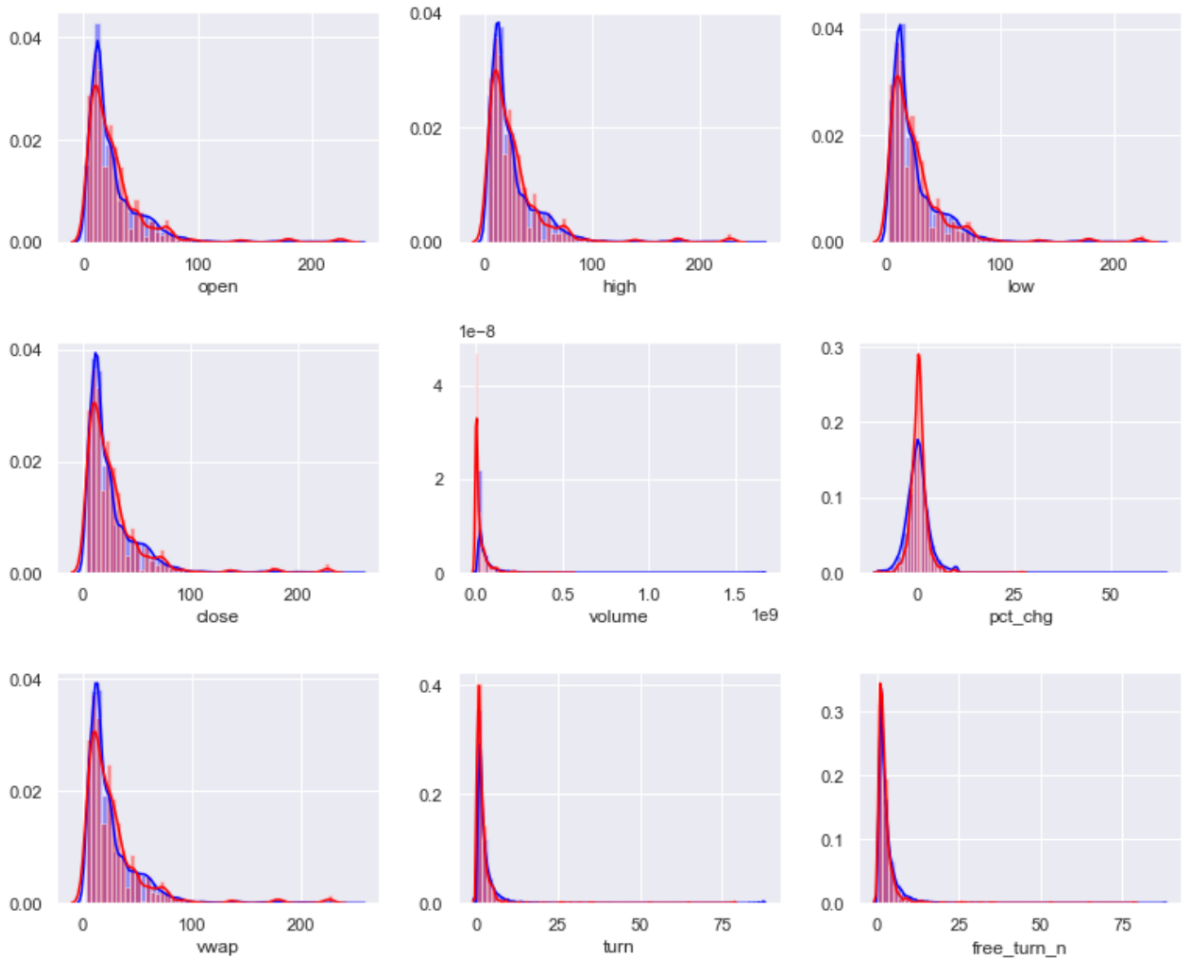
3. Feature Correlation and Distributions

To get a better understanding of the properties of different features, we plot the features `correlations heatmap` and the `distributions` of the features. Through the corrletion graph, we can see that features from the same type has high correlation. Through the distributions graph, we can see that most of the features do not have a normal distribution.

Feature correlation



Feature distribution



Model

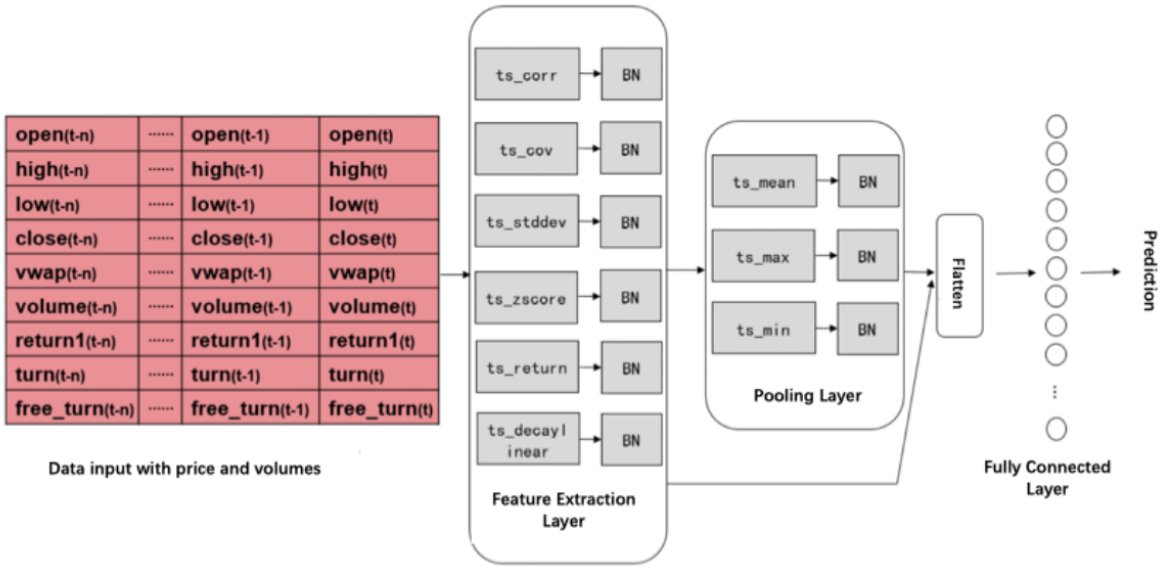
We devide our data set into training set and validate set. According to datetime, the former 80% belongs to the training set and the rest belongs to validate set. We devide data according to datetime in order to avoid the influence of so called future information.

1. Alphanet: use raw price-volume data as input

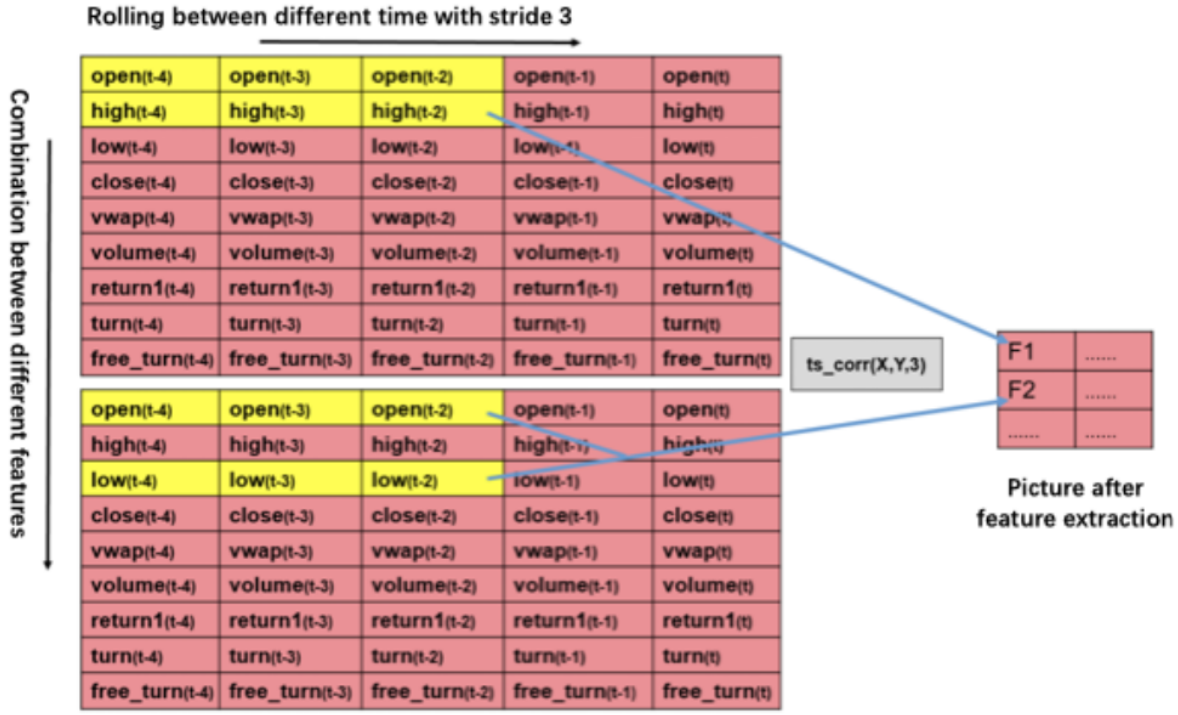
In order to effectively extract features from the original stock volume and price data, AlphaNet uses the idea of feature construction in genetic programming and uses a variety of operator functions as a custom network layer for feature extraction. AlphaNet consists of four parts:

- `Data input`: adopting CNN data format, the original volume and price data of individual stocks are sorted into two-dimensional "data pictures"
- `Feature extraction layer`: the most critical part of AlphaNet, it implemented a variety of custom computing network layers to extract features, and use the Batch Normalization layer for feature standardization.
- `Pooling layer`: consistent with pooling layer in CNN, the characteristics of the upper layer are "blurred".
- `Fully connected layer`: weighted synthesis of extracted features and uses `sigmoid` activation function to make classification

Network structure



Example show the process of feature extraction



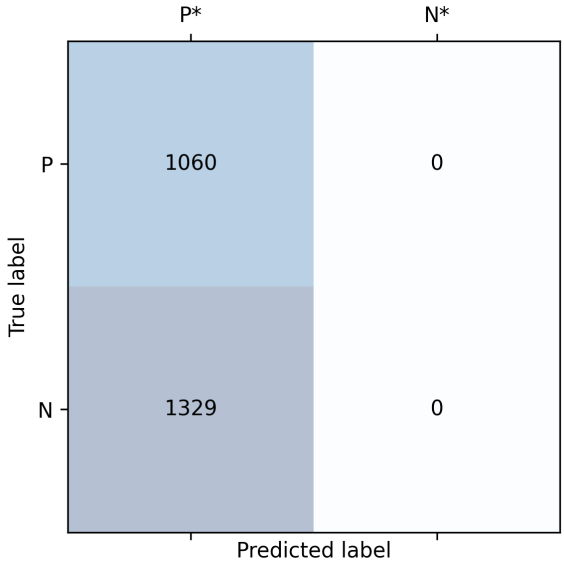
Network structure details

Layer name	Components	Parameters
Feature extraction	ts_corr(X, Y, stride), ts_cov(X, Y, stride), ts_stddev(X, stride), ts_zscore(X, stride), ts_return(X, stride), ts_decaylinear(X, stride), ts_mean(X, stride), BN	stride=10
Pooling layer	ts_mean(X, stride), ts_max(X, stride), ts_min(X, stride), BN	stride=3
Fully Connected Layer	30 neurons	dropout rate: 0.5, activation function: relu
Output Layer	1 neurons	activation function: sigmoid
Other params	`Loss function`: Binary_cross_entropy, `Optimizer`: Adam, `Learning rate`: 0.001, `batch_size`: 256	

Toolkit: PyTorch

PyTorch is an open source machine learning library and is widely used in deep learning scenarios for its flexibility. PyTorch uses dynamic computational graphs rather than static graphs, which can be regarded as the mean difference between it and other deep learning frameworks.

Confusion matrix



2. Other models: use the price-volume data after feature extraction as input

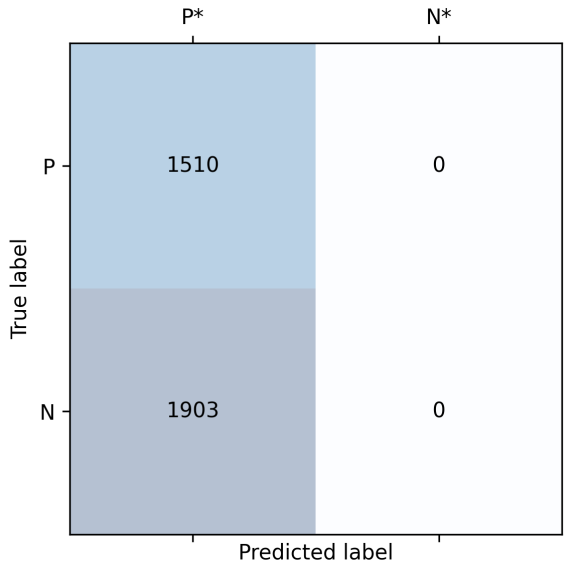
2.1 Convolutional Neural Network

Convolutional Neural Network (CNN) is the most influential model in the field of computer vision research and application. The structure of Convolutional Neural Network mimics the working principle of the visual nerve of the eye. The optic nerve is the bridge of the it and the brain, the visual nerve of a lot of collaboration, each responsible for a small parts of the visual image, then the image of different combination of the local feature abstraction to the top of visual concept, make the human has the visual recognition ability Convolution neural network is also similar, includes input layer, hidden layer and output layer. The hidden layer is composed of `convolutional layer, pooling layer and full connection layer`. Feature extraction is carried out by multiple convolution kernels in the local area of the image, and dimension reduction is carried out by pooling layer, and finally feature synthesis is carried out by full connection layer.
In this project, we build a `CNN with one convolutional layer, one max pooling layer with kernal (3, 3) and two fully connected layer with neurons 32, 1`, respectively.

Network structure

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_9 (Conv2D)            (None, 133, 19, 32)       320
_________________________________________________________________
batch_normalization_33 (Batc (None, 133, 19, 32)       532
_________________________________________________________________
max_pooling2d_19 (MaxPooling (None, 44, 6, 32)         0
_________________________________________________________________
batch_normalization_34 (Batc (None, 44, 6, 32)         176
_________________________________________________________________
flatten_14 (Flatten)         (None, 8448)              0
_________________________________________________________________
dense_28 (Dense)             (None, 32)                270368
_________________________________________________________________
dense_29 (Dense)             (None, 1)                 33
=================================================================
Total params: 271,429
Trainable params: 271,075
Non-trainable params: 354
_________________________________________________________________
```

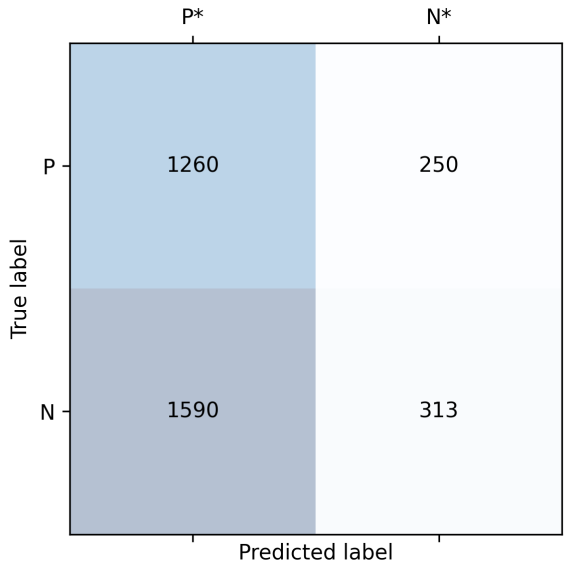
Confusion matrix



2.2 LogisticRegression

In order to reduce the time of computing, we use pca method to deduct dimensions. We observed that the 3 most import features can explain almost 90% of all the variance, so we used pca method and set components equal to 3.
To improve the performance of models, we use GridSearch to find the best C value from [0.01, 0.1, 1, 10, 100].

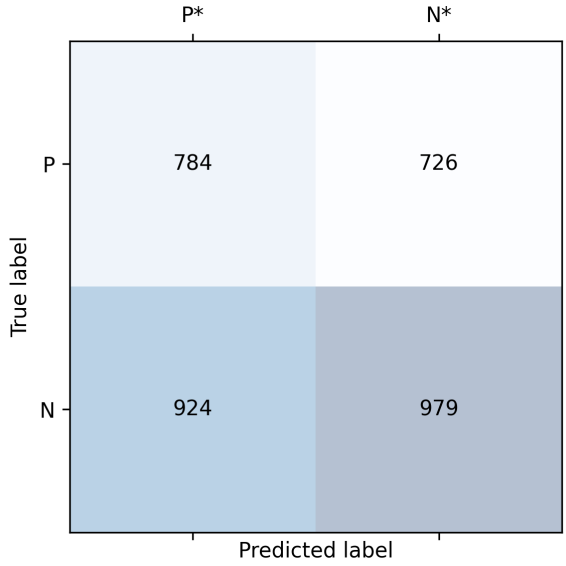
Confusion matrix



2.3 RandomForest

To improve the performance of models, we use GridSearch to find the best max depth from [10, 20, 30, 40] and best max features from [5, 15, 25, 35, 45, 55].

Confusion matrix



Factor Validity Test

1.Rank_IC

Firstly, we calculate Rank IC of factors that we get from four different machine learning methods: Alphanet, CNN, Linear Logistic and Random Forest). Rank IC shows the relation between the rank of factors and the rank of asset return. **A high positive or a low negative correlation mean our factor is well explained the asset return and have a good prediction.**

$$Rank\_IC = Corr(Rank(factor), Rank(asset\_return))$$

In addition, we check the standarlized IC which is called information ratio (IR):

$$Rank\_IC\_IR = Mean(Rank\_IC)/Std(Rank\_IC)$$

In order to analyze the incremental information of the synthetic factor, we also show the test results after the market value is neutralized.

$$factor = \beta_0 + \beta_1 \times ln\_mv$$

$$factor\_mv\_neutral = factor - \hat{factor} = factor - \hat{\beta}_0 - \hat{\beta}_1 \times ln\_mv$$

Rank IC and IR results

	Rank_IC_mean	Rank_IC_std	Rank_IC_IR	Rank_IC>0
alphanet_factor	nan%	nan%	nan	0.00%
cnn_factor	1.64%	10.98%	0.15	36.27%
cnn_mv_neutral_factor	0.68%	12.23%	0.06	35.78%
LR_factor	0.37%	13.72%	0.03	51.96%
LR_mv_neutral_factor	0.11%	12.63%	0.01	49.51%
RF_factor	0.22%	11.79%	0.02	52.45%
RF_mv_neutral_factor	0.23%	10.69%	0.02	49.51%

There must something wrong in alphanet since it has no correlation with return. After checking the factor data, we find CNN find nothing after learning. All factors of each stock are totally same.

We guess that there are several reasons for this failure：

- We do not use enough sample data, only 500-trading-day data, to train the module.
- Our computer does not have enough computing power to support the operation of this algorithm.

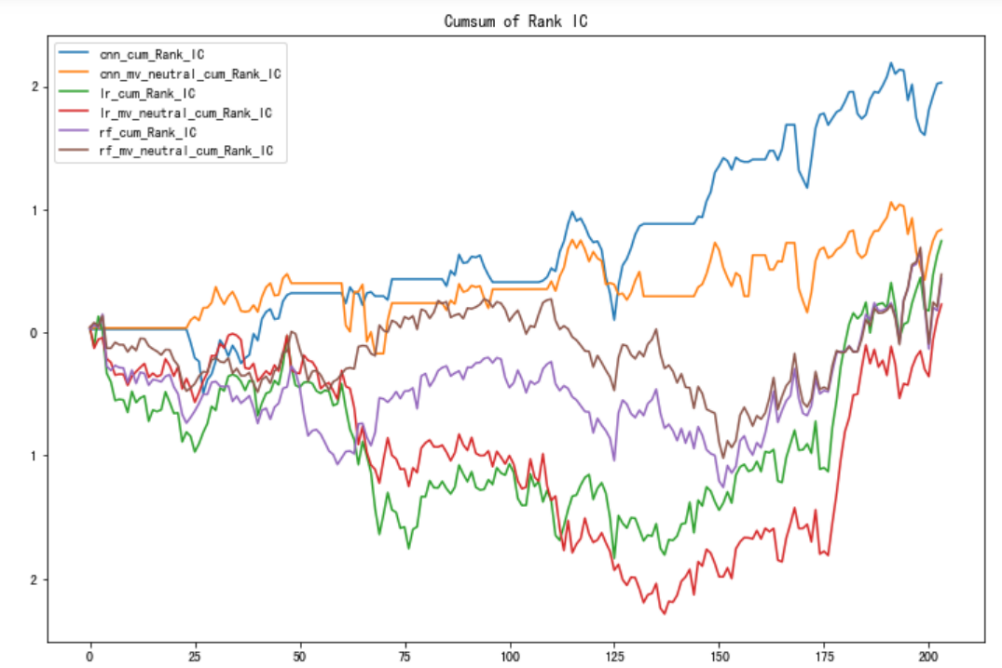
prediction result of Alphanet

	0	1	2	3	4	5	6	7	8	9	...	290	291
0	0.396033	0.396033	0.396033	0.396033	0.396033	0.396033	0.396033	NaN	NaN	0.396033	...	0.396033	NaN
1	0.396033	0.396033	0.396033	0.396033	0.396033	0.396033	0.396033	NaN	NaN	0.396033	...	0.396033	NaN
2	0.396033	NaN	0.396033	0.396033	0.396033	0.396033	NaN	NaN	NaN	0.396033	...	0.396033	NaN
3	0.396033	NaN	0.396033	0.396033	0.396033	0.396033	NaN	NaN	NaN	0.396033	...	0.396033	NaN
4	0.396033	NaN	0.396033	0.396033	0.396033	0.396033	NaN	NaN	NaN	0.396033	...	0.396033	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...
199	0.464561	0.464561	0.464561	0.464561	0.464561	0.464561	0.464561	0.464561	0.464561	0.464561	...	0.464561	0.464561
200	0.464561	0.464561	0.464561	0.464561	0.464561	0.464561	0.464561	0.464561	0.464561	0.464561	...	0.464561	0.464561
201	0.464561	0.464561	0.464561	0.464561	0.464561	0.464561	0.464561	0.464561	0.464561	0.464561	...	0.464561	0.464561
202	0.464561	0.464561	NaN	0.464561	0.464561	0.464561	0.464561	0.464561	0.464561	0.464561	...	0.464561	0.464561
203	0.464561	0.464561	NaN	0.464561	0.464561	0.464561	0.464561	0.464561	0.464561	0.464561	...	0.464561	0.464561

204 rows × 300 columns

Therefore, in subsequent inspections and backtests, we will no longer test the Alphanet method

Cumsum of Rank_IC of each method



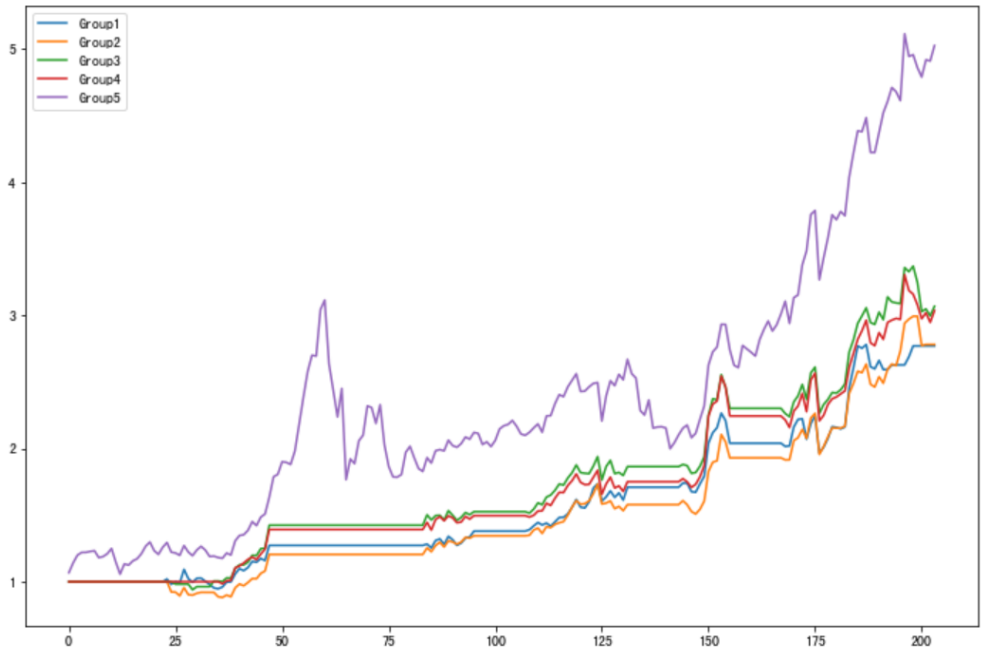
The synthetic factors do not have a pretty well explanation of asset returns. Compared with logistice regression and random forest, CNN has better correlation with asset returns with a highest mean of Rank_IC.

2.Hierarchical backtest (market_value weighted)

We also conducted a hierarchical back-testing test. **The stocks are divided into five groups according to the order of factors from small to large**, and the market value of each group is weighted to analyze the net value trend during the test period. If the five groups of trends are stratified obviously, it means that the factor explains the asset return well.

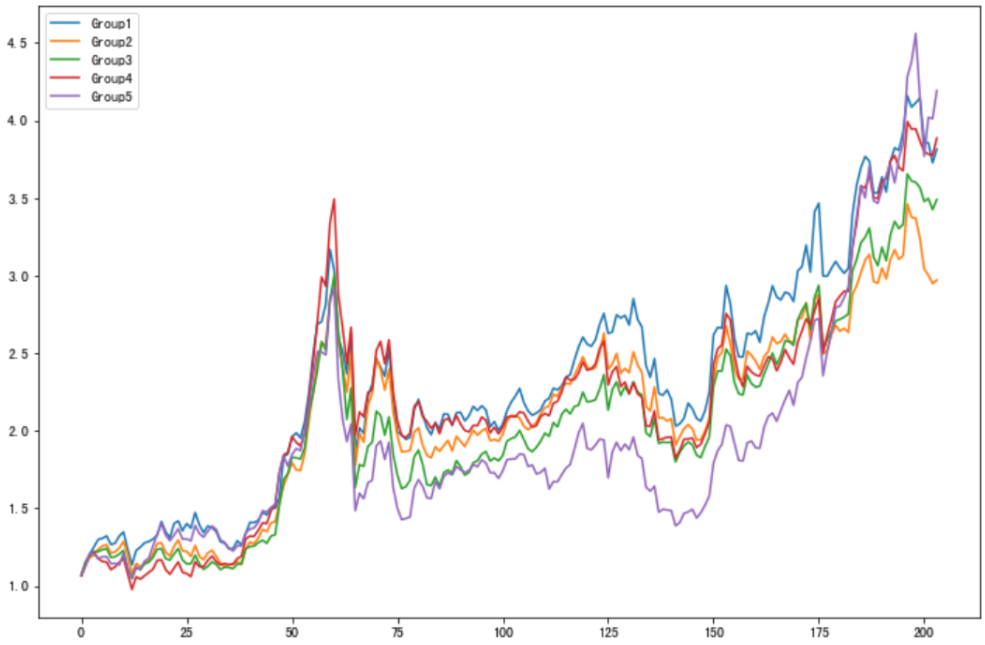
2.1 CNN Hierarchical backtest

The fifth group which the factors value are highest makes a much better performance than other four groups. However, The other four groups are not distinguished obviously.



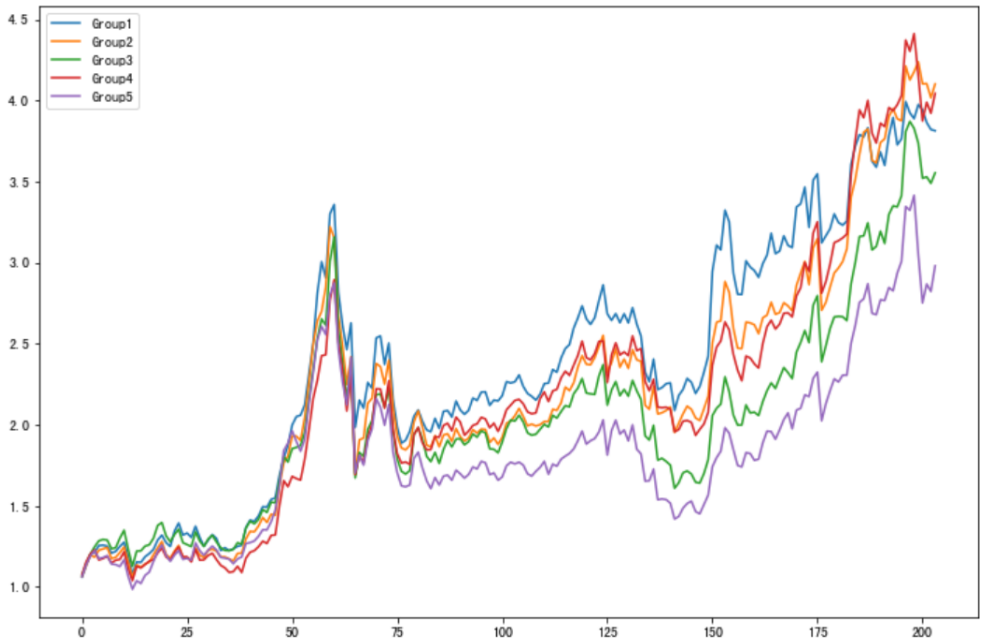
2.2 Logistic Regression Hierarchical backtest

The five groups devided by LR do not perform differently.



2.3 Random Forest Hierarchical backtest

The five groups devided by RF do not perform differently.

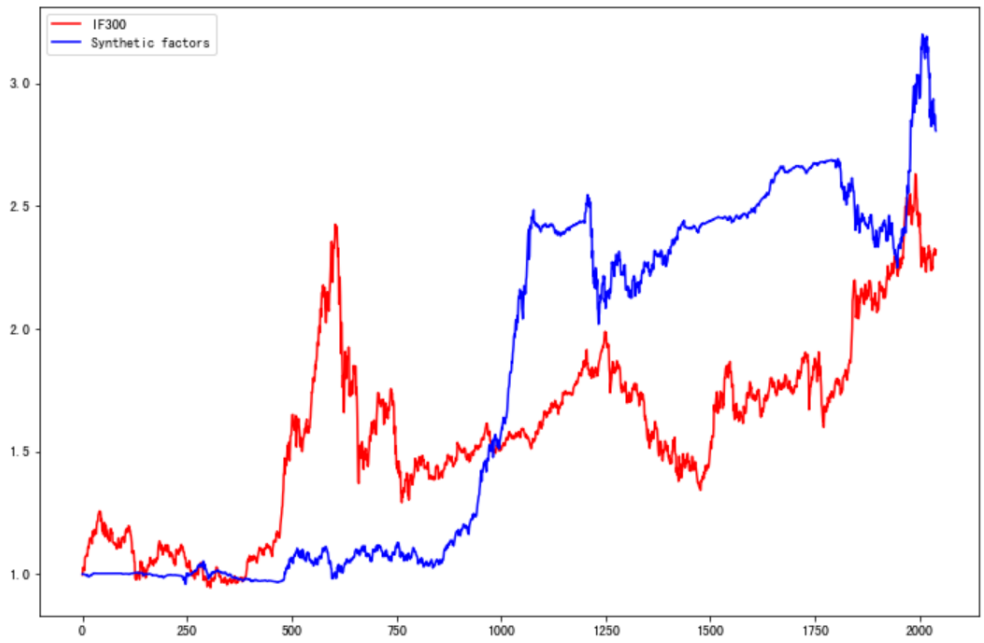


Net value backtest

In this part, we make backtest of each maching leaning method and draw the net value curve to compare it with CSI 300 Index(IF300). The blue curve is the backetest result of our machine learning mehtod and the red curve is the cumulative net value of CSI 300 Index.

1. CNN net value backtest

2. Logistic regression net value backtest



3. Random forest net value backtest

Ramdom forset cannnot beat IF300 well.

We analyze four important indexes which are annualized rate of return, annualized volatility, sharp ratio and maximum drawdown.

According to those indexed, we can conclude that CNN ,logistic regression and random forest do not perform so well since their sharp ratio are all around or less than 1.

	annualized rate of return	annualized volatility	sharp ratio	maximum drawdown
CNN	13.15%	12.46%	1.06	12.28%
Logistic regression	13.48%	13.19%	1.02	12.28%
Random forest	10.50%	12.48%	0.84	8.83%