Supervised Learning in Short-Term Price Trends Forecasting

Keywords: Supervised learning, Price trends, Factors, Majority vote

By Xiangwu Li.

November 12th, 2018

I. Overview

1. Introduction

According to Modern Portfolio Theory, the expected return of stocks is the reward for the risk undertaken. The multi-factor model is a quantitative expression of the risk-return relationship, and different factors represent the explanatory variables for different risk types.

The research aims to draw on factors to predict stock price trends using machine learning algorithms.

Specifically, taking the constituent stocks of CSI300 index as research objects(300 stocks), ranked the next day yields and categorized into three classes(each class contained the same number of stocks, which meant the original accuracy by random guesses on it was 0.3333), labeled as "0, -1, 1" respectively indicating "decline, steadiness, rise" of price trends; drew on the correlation analysis to determine final factor dataset including RSI, MACD, CCI, ATR, etc.; used accuracy and ROC curve to evaluate classifier output quality, and it found that among selected models, Logistic Regression, MLP, SVM, AdaBoost gave a great performance; besides, combined models by introducing into Majority Vote and finally increased the accuracy to **0.5285 on three-class prediction for short-term price trends** in out-sample test. (code and data of this project can be found at https://github.com/Luffy-wu/Predicting-the-Stock-Price-Trends-using-AI .)

2. Classifiers

The following classifiers were selected:

Classifiers	Description
Logistic Regression	Logistic Regression is a common classification algorithm based on probability
MLP	The existence of hidden layers helps interpret complex relationships within the data
K-Nearest Neighbors	Use K surrounding labels to classify
Support Vector Machines(SVM)	By constructing hyperplanes in a high-dimensional space, it can capture complex relationships and create non-linear decision boundaries
Decision Tree	Create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features
Random Forest(Ensemble)	An ensemble learning method by constructing a multitude of decision trees (Parallel combination)

AdaBoost(Ensemble)	A meta-estimator that begins by fitting a classifier and then fits additional copies of the classifier but where the weights of incorrectly classified instances are adjusted (Serial combination)
Naïve Bayes(Gaussian)	A probabilistic classifier based on applying Bayes' theorem
Discriminant Analysis(Quadratic)	Separate measurements of classes of objects or events by a quadric surface

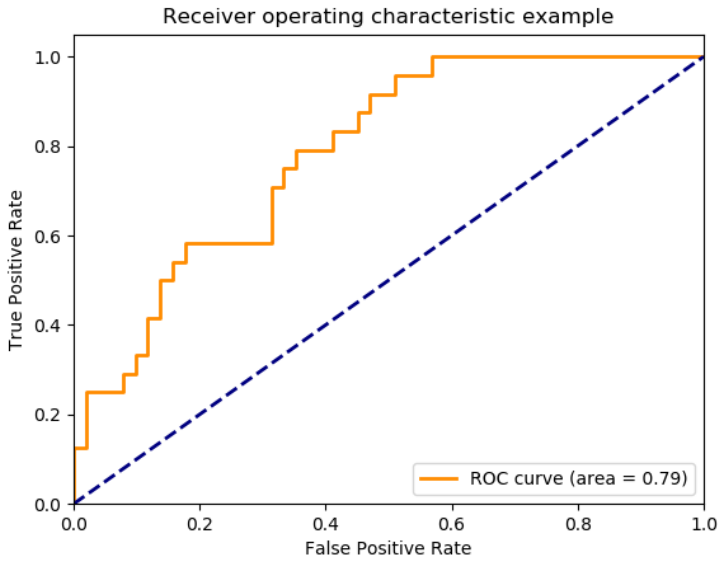
3. Metrics

To evaluate the quality of predictions, the most popular approach is to measure a model's accuracy, precision and F-score. As for classification problems, considering simplicity and practicability, I chose accuracy and ROC curves to evaluate classifier output quality.

Accuracy measures how often the classifier makes the correct prediction.

$$accuracy = \frac{\sum Right}{\sum Right + \sum Wrong}$$

Roc curves typically feature true positive rate on the Y axis, and false positive rate on the X axis. This means that a larger area under the curve (AUC) is usually better. Besides, the "steepness" of ROC curves is also important.



II. Data Exploration

1. Data Collection

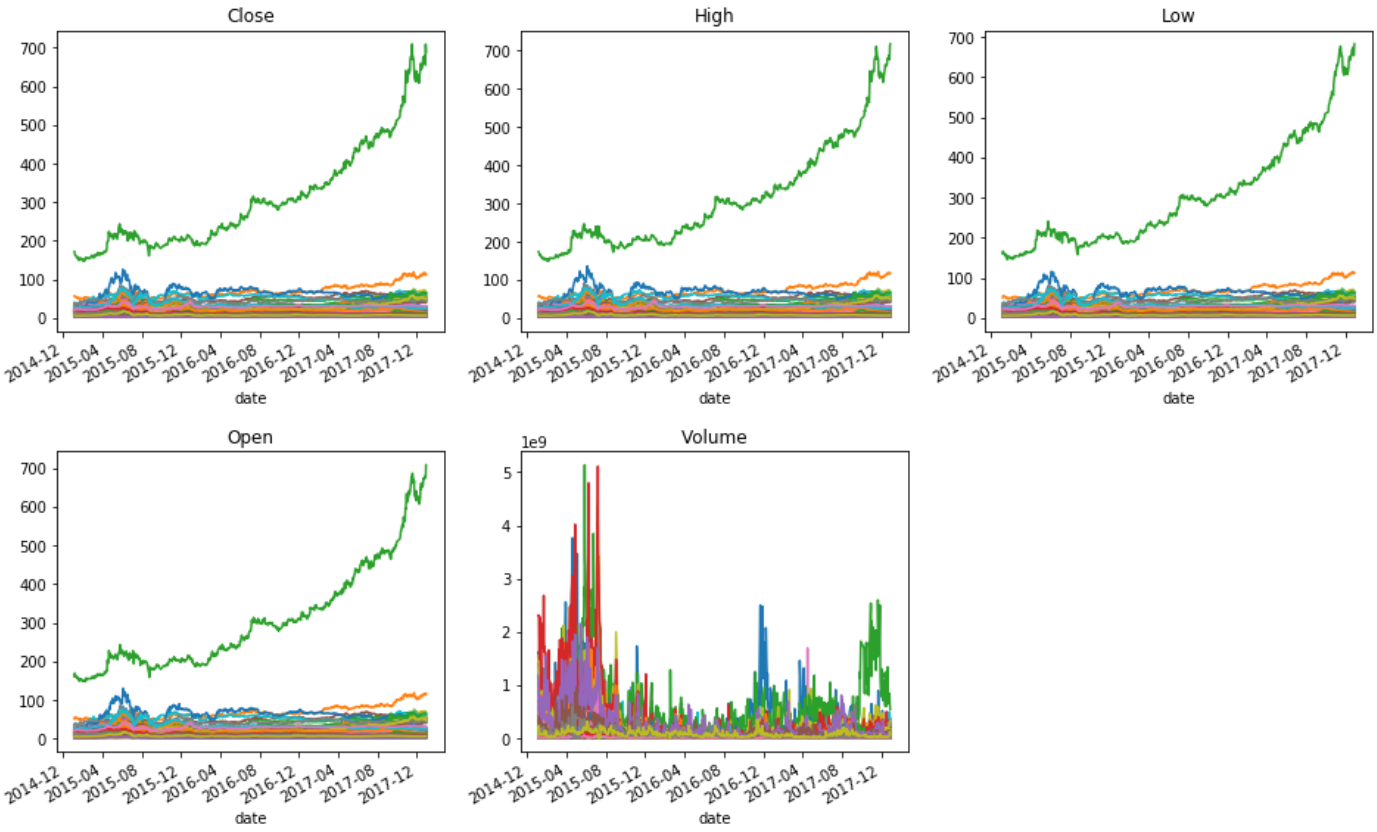
The original datasets included the open, high, low, close, volume of all the constituent stocks of CSI300 index from Jan.1st 2015 to Dec.31st 2017(300 stocks, daily). Especially, removed stocks that had been suspended for more than 30 days.

2. Data Visualization

a. Original Market Data

Below was a plot of original market data (open, high, low, close, volume)

Market Data



b. Generated and Filtered Factors

Generated the factors based on original market data. First, I chose the below as candidate factors

Candidate Factors

Factors	Objects	Groups	Description
MA	close	Overlap Studies	Moving average
EMA	close	Overlap Studies	Exponential Moving Average
AROON	high, low	Momentum Indicators	Aroon
BOP	open, high, low, close	Momentum Indicators	Balance of Power
MFI	high, low, close, volume	Momentum Indicators	Money Flow Index
CCI	high, low, close	Momentum Indicators	Commodity Channel Index
CMO	close	Momentum Indicators	Chande Momentum Oscillator

MACD	close	Momentum Indicators	Moving Average Convergence/Divergence
RSI	close	Momentum Indicators	Relative Strength Index
STOCH	high, low, close	Momentum Indicators	Stochastic
AD	high, low, close, volume	Volume Indicators	Chaikin A/D Line
ADOSC	high, low, close, volume	Volume Indicators	Chaikin A/D Oscillator
OBV	close, volume	Volume Indicators	On Balance Volume
ATR	high, low, close	Volatility Indicators	Average True Range
NATR	high, low, close	Volatility Indicators	Normalized Average True Range
TRANGE	high, low, close	Volatility Indicators	True Range
BETA	high, low	Statistic Functions	Beta
CORREL	high, low	Statistic Functions	Pearson's Correlation Coefficient (r)
TSF	close	Statistic Functions	Time Series Forecast

Then, I calculated future yield and performed correlation analysis to measure the predictive capability of factors above; removed those with low correlation, and determined final factors as model input, including RSI, MACD, CCI, CMO, ATR, BOP, MFI, ADOSC, BETA (below is the correlation matrix of selected factors and future yields, “returns” in the below chart presenting future yields)

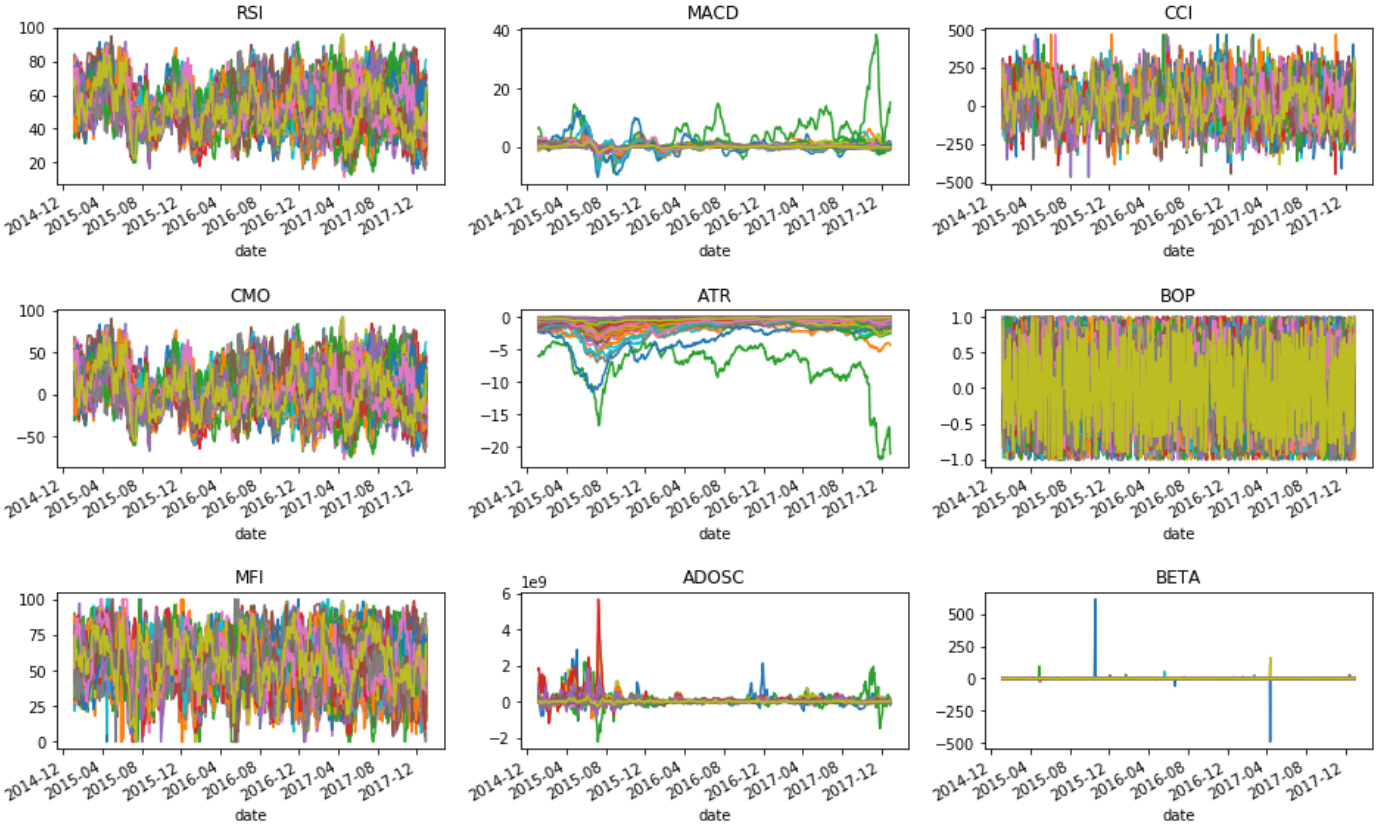
Correlation Matrix for Selected Factors and Future Yields

	RSI	MACD	CCI	CMO	ATR	BOP	MFI	ADOSC	BETA	returns
RSI	1.000000	0.842055	0.745477	0.823416	0.084434	0.235900	0.649300	0.291901	0.263322	0.343681
MACD	0.842055	1.000000	0.406823	0.842055	0.145696	0.006534	0.538486	-0.045322	0.199254	0.072925
CCI	0.745477	0.406823	1.000000	0.745477	0.224976	0.275383	0.678978	0.273050	0.257596	0.400500
CMO	0.823416	0.823416	0.823416	0.823416	0.823416	0.823416	0.823416	0.823416	0.823416	0.823416
ATR	0.084434	0.145696	0.224976	0.084434	1.000000	0.021427	0.231019	-0.474157	0.155256	0.046776
BOP	0.235900	0.006534	0.275383	0.235900	0.021427	1.000000	0.107827	0.144707	0.147933	0.744529
MFI	0.649300	0.538486	0.678978	0.649300	0.231019	0.107827	1.000000	0.148251	0.296287	0.166755
ADOSC	0.291901	-0.045322	0.273050	0.291901	-0.474157	0.144707	0.148251	1.000000	0.057233	0.166747
BETA	0.263322	0.199254	0.257596	0.263322	0.155256	0.147933	0.296287	0.057233	1.000000	0.159104
returns	0.343681	0.072925	0.400500	0.343681	0.046776	0.744529	0.166755	0.166747	0.159104	1.000000

c. Factors Visualization

Below was a plot of selected factors (RSI, MACD, CCI, CMO, ATR, BOP, MFI, ADOSC, BETA), which showed that there was a need to standardize data.

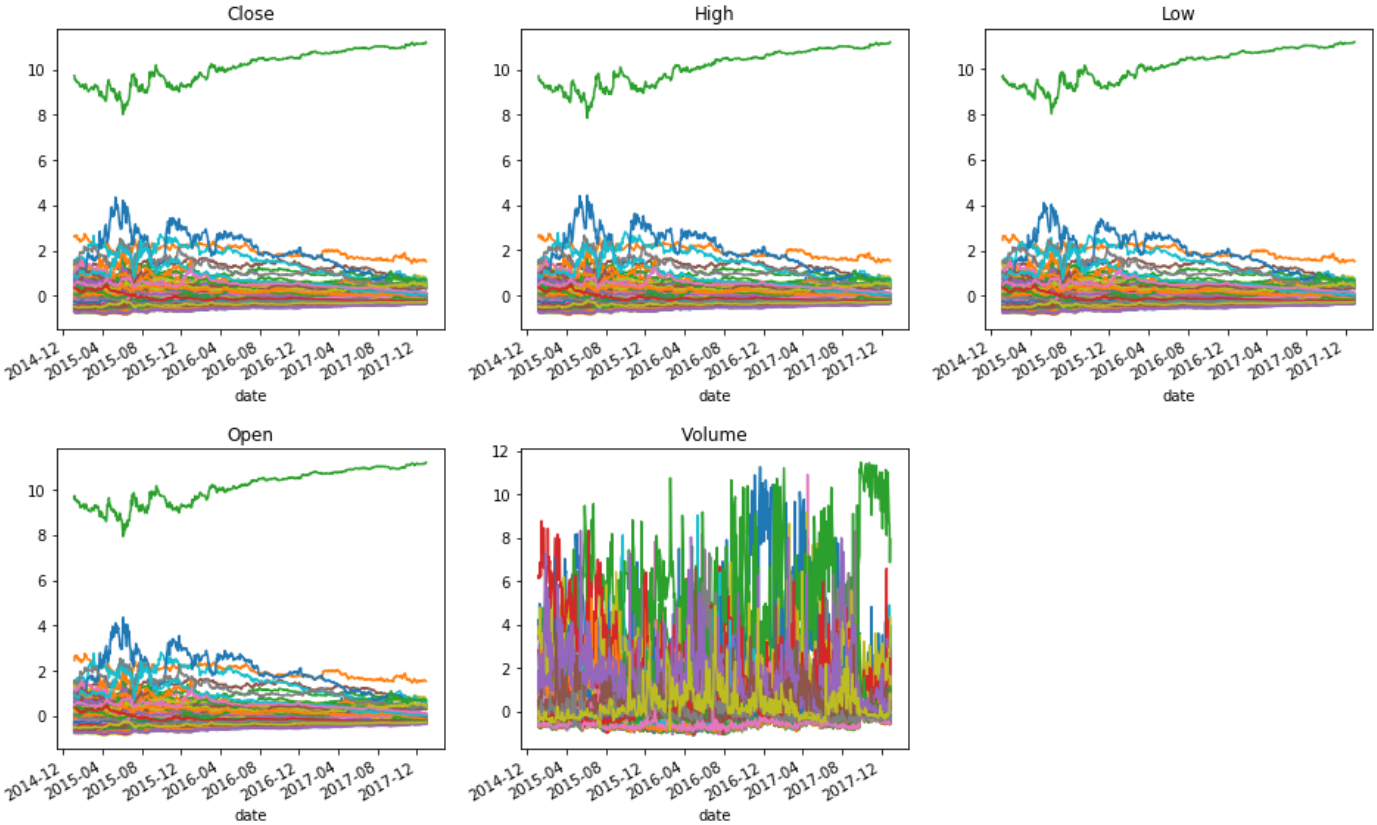
Selected Factors



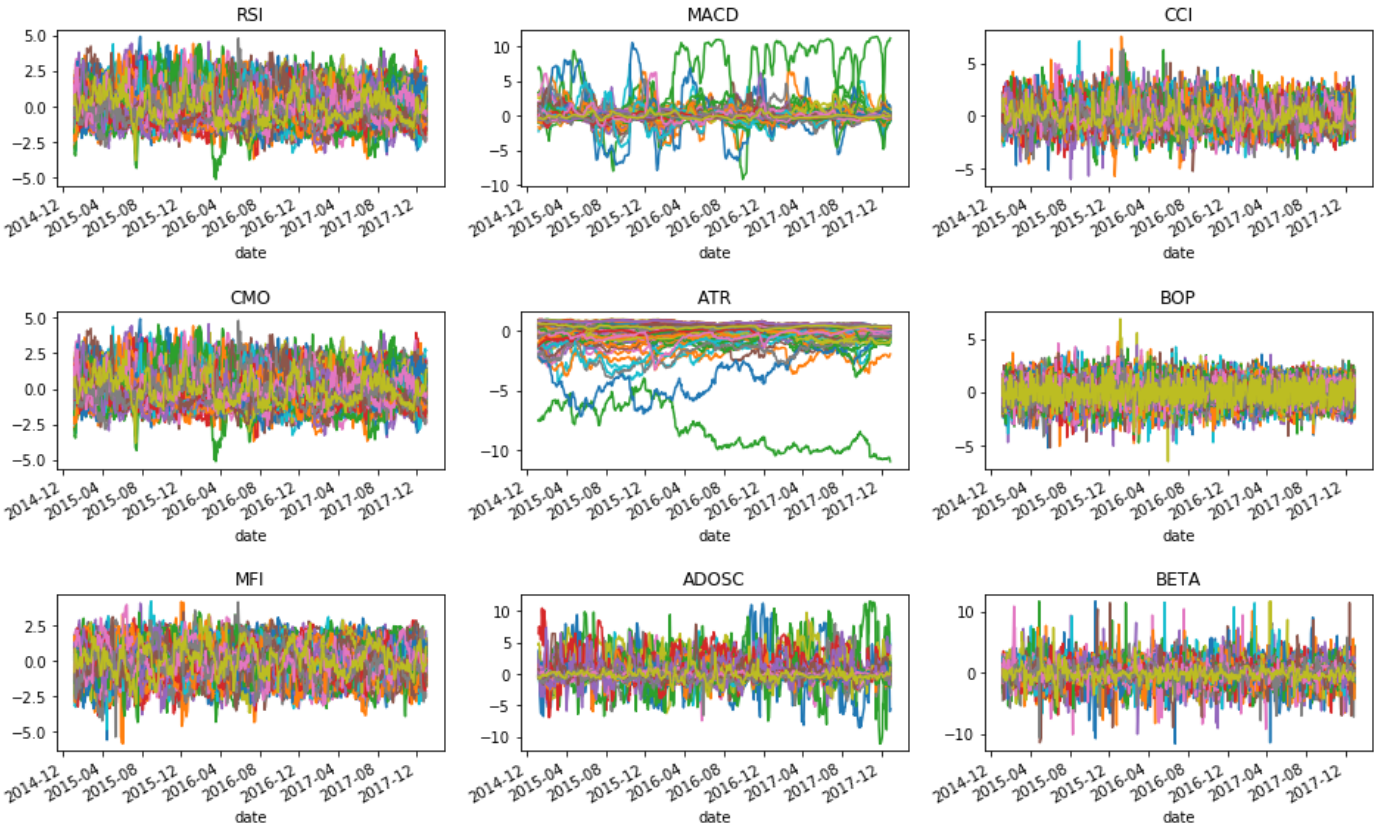
d. Standardization

Standardized original data and the selected factors by line, respectively.

Standardized Market Data



Standardized Selected Factors



The above showed that standardized data presented better distribution characteristics for model input.

III. Experiment and Result Analysis

1. Preparation

- Used market data and the selected factor respectively as X
- Ranked the next day yields using quantiles and categorized into three classes, “0, -1, 1” respectively indicating “decline, steadiness, rise” of the stock price, as Y
- Divided it into training and test set

2. Trained Models Using Market Data

Used the market data as X and categorized next day yields as Y to train a set of machine learning models.

Out-of-sample Accuracy of Models Using Market Data

Classifiers	Accuracy
Logistic Regression	0.3634
MLP	0.3519
K-Nearest Neighbors	0.3492
Support Vector Machines(SVM)	0.3420
Decision Tree	0.3465
Random Forest(Ensemble)	0.3482

AdaBoost(Ensemble)	0.3536
Naïve Bayes(Gaussian)	0.3355
Discriminant Analysis(Quadratic)	0.3508

The accuracy of models was just slightly higher than 0.3333 (the original accuracy by random guesses on three-class classification problem), which showed original market data was not an effective input for models.

3. Trained Models Using Factors

Used the selected factors as X and categorized yields as Y to train the models.

Out-of-sample Accuracy of Models Using Factors

Classifiers	Accuracy
Logistic Regression	0.4545
MLP	0.4443
K-Nearest Neighbors	0.3831
Support Vector Machines(SVM)	0.4325
Decision Tree	0.3800
Random Forest(Ensemble)	0.4009
AdaBoost(Ensemble)	0.4321
Naïve Bayes(Gaussian)	0.3904
Discriminant Analysis(Quadratic)	0.4293

Trained with factors, the accuracy of models was much higher than before; the models like Logistic Regression, MLP, SVM, AdaBoost performed well, achieving an accuracy rate higher than 0.43, which suggested that factors offered more valuable information for models.

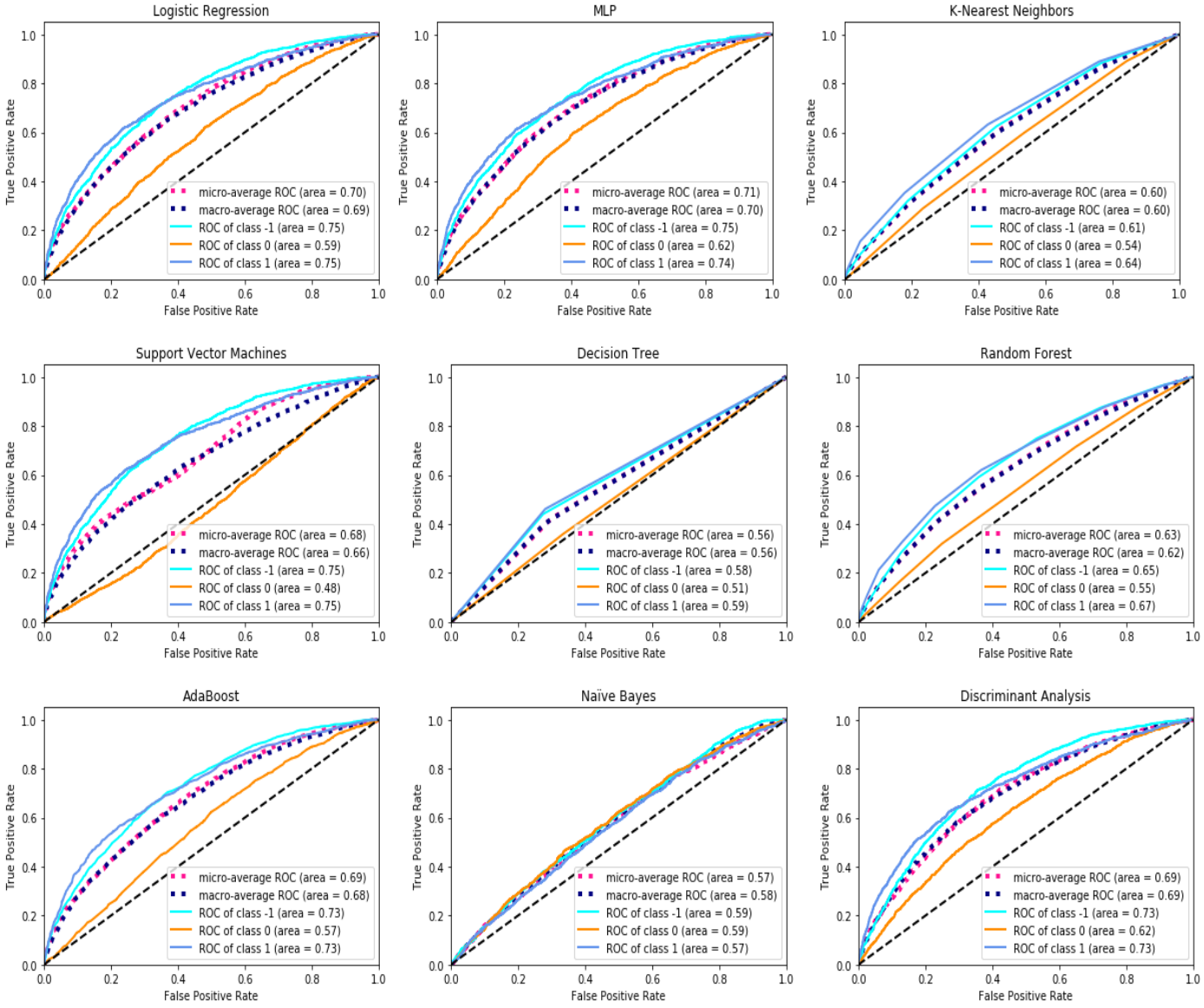
Out-of-sample Accuracy on Different Factors

Factors	Accuracy
RSI	0.4780
MACD	0.3933
CCI	0.4467
CMO	0.4764
ATR	0.3662
BOP	0.4770
MFI	0.3868
ADOSC	0.3777
BETA	0.3451

The average accuracy of models on different factors showed that performance among factors was quite different, factors like RSI, CMO, BOP offering the most valuable information for classification.

4. ROC Analysis

To better evaluate the prediction quality of different models trained by factors, I used ROC curves to evaluate classifier output quality.



As above, we can conclude that the area of ROC of all models for both micro and macro average was larger than 0.5, suggesting that all models achieved an effective classification. Besides, almost all models performed better in class “-1” and class “1” than class “0” (the area is smallest).

Comparing the area size under the curve of different models, we can easily find that Logistic Regression, MLP, Support Vector Machines, AdaBoost, Discriminant Analysis were more outstanding, indicating a better performance in the multi-class classification.

Considering steepness of ROC curves, it suggested that Logistic Regression and MLP performed quite well in all three classes.

5. Model combination

At last, I introduced “Majority Vote” into the classification to combine all models above to give a more precise prediction; that is, combined all selected factors (RSI, MACD, CCI, …, etc.) and respectively trained

the models, then took into consideration prediction results of all models and chose the class predicted by the majority as final prediction; it finally achieved an optimal accuracy of 0.5285 on three-class classification problems in out-sample test.

Optimal Accuracy by Combining Models

```
0.40756799369333857
0.3642096964919196
0.4205754828537643
----------------AdaBoost----------------
0.42845880961765864
----------------------------------------
0.42451714623571146
0.40717382735514385
0.3756405202995664
0.35829720141899885
0.33307055577453687
0.3579030350808041
0.3713046905794245
0.3445013795821837
0.3512022073314939
----------------Naïve Bayes----------------
0.38904217579818684
----------------------------------------
0.334253054789121
```

```
print('----------------'+'final'+'----------------')
print(np.mean(y_pred_all == y_test))
```

```
----------------final----------------
0.528577059519117
```

IV. Conclusion

In this project, I applied a set of machine learning models to predicting price trends for constituent stocks of CSI300 index. My finds can be summarized into three aspects:

1. Compared with original market data, factors based on technical indicators like RSI, CCI, ATR, etc. offered more valuable information as input for models.

2. Accuracy and ROC curve analysis suggested that among selected models, Logistic Regression, MLP, SVM, AdaBoost gave a better performance in the multi-class classification; Especially, Logistic Regression and MLP performed excellently in all three classes.

3. By introducing Majority Vote principle to combine all selected models, it achieved an optimal out-of-sample accuracy of 0.5285 on this three-class prediction for short-term price trends.