Suetming Ma
November 21st, 2016

China's Stock Movement Predicting

Project Overview

To predict the stock price is a problem in stock market all over the world. Investment firms, hedge funds and even individuals have been building their own model to predict the stock price and make profitable investments and trades.

This project will focus on algorithmically method, exactly machine learning algorithms which are widely used in information science and computer science to build stock price movement predictor.

There are a lot of stock exchanges around the world and there are many public stock dataset online, I chose China's stock daily trading data because of the stock price volatility and the history of stock market is very short, that means is more profitable.

Problem Statement

The China's stock price forecasting is a supervised classification problem. There are China's stock data from 2015-01-01 to now, The goal is to extract proper features of 10 days and build an effective model to predict the stock correct movement of the day 10.

I will fetch the data set and preprocess the data set to the training data set and the test data set, then I will verify the hypothesis and use several classification methods to predict the stock price movement. I will find the optimal parameters of the model and combination of these models. Finally I will validate the performance of the classifiers on different dataset and parameters.

In this project, if I take all years of historical stock data, it will be a huge computing and memory demand for my computer, so as to speed computing, I will only select the China's HUSHEN 300 stocks data since 2015, and validate the model on the rest topics. From the datasets, that lists each stock ticker, along with the associated company name and industry area. There is 9 different industries areas as follows:

1. Media entertainment
2. Power industry

3. Electronic and information technology
4. Real estate industry
5. Building materials
6. Transportation
7. Financials
8. Automobile manufacturing
9. Biopharmaceutical

I will fit all of these industries that the different models to look at the effect of different models and also will fit each industry that the different models to look at the effect of different models in different industries.

Metrics

As it is a multi-class classification problem with thousands of samples, I will use AUC[3] (area under curve) as my cross-validation metric since it has a neat probabilistic interoperation in classifier without being sensitive to the relative sizes of the classes and it is a common evaluation metric for binary classification problems.

Analysis

Data Exploration

The initial dataset can be acquired from the website of Sina finance (finance.sina.com.cn) and economics. The China's stock market was established in 1989. As of June 2016, there is a total of 2964 listed companies in the a-share market consists of the motherboard, small and medium-sized plate and the gem.

A total of 240 stocks in this above 9 different industry areas I could fetch from the internet. The remaining stock dataset which is not complete is removed, A stock of the initial datasets contains the date of the stock exchange, the opening price, the closing price adjusted for stock splits and dividends, the highest price, the lowest price, the volume that how many stocks were traded and the stock code.

The opening price is the price of a security at the beginning of a day of trading on the stock market. When a stock exchange opens, each security has an initial trading price at which it is bought (and sold) on the first trade of the day. The calculation of open price differs depending on the stock exchange.

The closing price is is the final price at which a security is traded on a given trading day. The closing price represents the most up-to-date valuation of a security until trading commences again on the next trading day and the adjusted closing price is a stock's closing price on any given day of trading that has been amended to include any distributions and corporate actions that occurred at any time prior to the next day's open. The adjusted closing price is often used when examining historical returns or performing a detailed analysis on historical returns.

The highest price is the highest price at which a stock traded during the course of the day and the lowest price is the lowest price at which a stock traded during the course of the day.

The volume is the number of shares or contracts traded in a security or an entire market during a given period of time. For every buyer, there is a seller, and each transaction contributes to the count of total volume. That is, when buyers and sellers agree to make a transaction at a certain price, it is considered one transaction. If only five transactions occur in a day, the volume for the day is five.

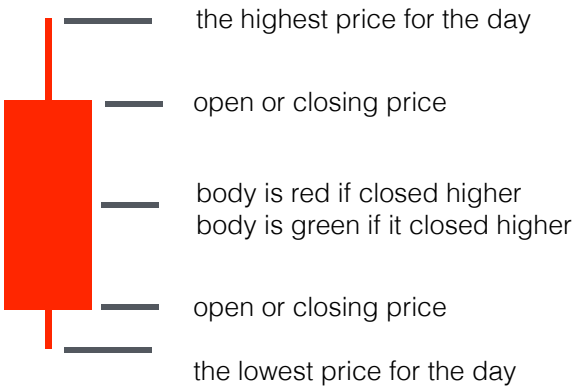
The sample stock data as follows:

DATE	OPEN	CLOSE	HIGH	LOW	VOLUME	CODE
2015-01-05	6.201	6.486	6.496	6.161	1146261	601872
2015-01-06	6.791	7.136	7.136	6.732	1004619	601872
2015-01-07	7.372	6.87	7.48	6.771	1933811	601872
...	...	...	...	...	...	...

The datasets for each of our industry areas using the generated stock tickers in from January 01, 2015 to December 02, 2016.

Exploratory Visualization

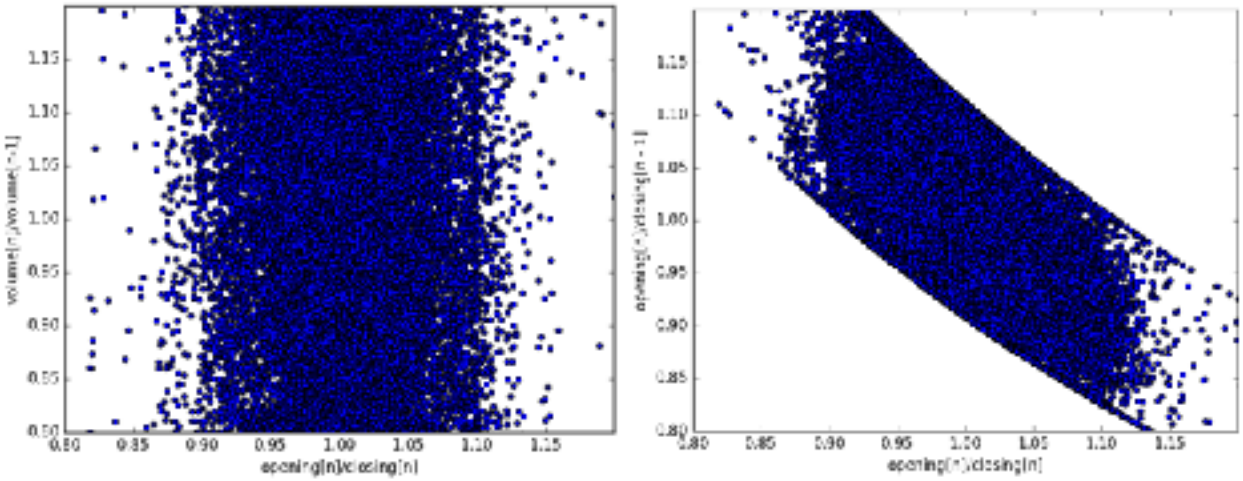
First, I want find a way to visualize stock data, and the most common way is candlestick chart. The candlestick techniques originated in the style of technical charting used by the Japanese for over 100 years before the West developed the bar and point-and-figure analysis systems.



As shown in the above, just like a bar chart, the daily candlestick line contains the market's open, high, low and close of a specific day. The rise and fall as much as 10% in China's Stock Exchange. I checked the HUSHEN 300 index data from January 01, 2015 to December 02, 2016 through candlestick chart as follows:

As is known to above, the stock price curve that has a rise and fall are generally decline. For the purpose of exploratory analysis, I focus first on understanding the background behind what could drive an increase in stock price. Although I have the previous many days of stock data for any stock that I am learning about, I think that not all of this information is necessarily for making a predictive model, because of our understanding of efficient markets [1] and momentum effects[2].

In financial economics, the efficient-market hypothesis (EMH) states that asset prices fully reflect all available information. A direct implication is that it is impossible to "beat the market" consistently on a risk-adjusted basis since market prices should only react to new information or changes in discount rates (the latter may be predictable or unpredictable). And the momentum effect says that what was strongly going up in the past will probably continue to go up in the near future. Stocks which outperform peers on 3-12 month period tend to perform well also in the future.



Base on the above, I believe that not all of this information is necessarily useful for making a predictive model, so I should find a way to filter features of the dataset. The inter-day change of volume and the intraday change (close on day n divided by open on day n) figure that means the increase of stock or decrease of stock as above is covered all the range is not useful as you can see. And the highest price and the lowest price is also not necessarily useful for predicting the stock movement. Finally I seek to correlate intraday change (open on day n divided by close on day n-1) and inter-day change (close on day n divided by open on day n). As shown in the above there is a negative correlation between interlay change (ratio between closing and opening price today), and inter-day change (ratio between opening price today and closing price yesterday). So, I will only utilize the opening price and the adjusted closing price.

As a result of the above findings, one can certainly make an arguments that there are strong market correction effects at play. The short-term market reaction may overreact to events that happen during closing hours, and then correct themselves.

Algorithms and Techniques

With the datasets from the above, I wish to explore several common statistical classifiers in order to investigate the predictive power they may yield on this data set. The logistic regression classifier such as ridge regression and lasso regression methods which are the model of regression analysis to predict and analysis, then the random forest and gradient boosting decision tree, the the characteristics are as following:

- ridge (logistic regression classifier)
  - Pros: a linear classifier, low variance and so is less prone to over-fitting
  - Cons: model interpretability is low
- lasso (logistic regression classifier)
  - Pros: a linear classifier, the sparse coefficient would be better
  - Cons: can not do group selection
- random forest
  - Pros: scalable, help average out the bias and reduce the variance, the training speed is fast.
  - Cons: In some noise classification scene would be prone to overfitting.
- gradient boosting decision tree
  - Pros: scalable, help average out the bias and reduce the variance, the training speed is fast.
  - Cons: requires careful tuning and can not extrapolate[6]

The four models has different advantages in these stock dataset on our 9 different industry datasets. The AUC as our cross validation metrics is in order to evaluate the performance of the models and tuning parameters.

After the best parameters of four different model is fixed, I could use platt scaling[5] which is a way of transforming outputs of a classification model into a probability distribution over classes to blend the different model together. The blending model has all the advantage of the different models I hope.

The Python scikit-learn package is used for training and testing the model, as well as the blending model.

Benchmark

The AUC score ranges from 0.5 to 1, with 1 being perfect classification at 0.5 being no better than luck. If the AUC score increase from 0.5 to 1, I hope my algorithm gets better and better at correctly classifying outcomes.

I don't know which threshold value I should set. I search the similar problem sets on the internet and found the article "https://goo.gl/tIM8a7" and the article "https://goo.gl/uSSARz" and the leader AUC scores roughly 0.55, so if I set the AUC score of my model greater than or equal to 0.55, I think that would be ok.

Methodology

Data Preprocessing

Now, to take in a text file with stock tickers (one per row), the desired data set will be produce. I assume that I will be using 10 days of data to predict whether the closing price on the 10th day will be greater than or less than the opening price on the 10th day. The dataset will be composed of approximately 2 years of stock price data, which consists of HUSHEN 300 stocks.

Since I can use a rolling time window to test our data, for any stock ticker, stock price data for n days will produce n−9 rows of data. In the above of section, the datasets has been preprocessed.

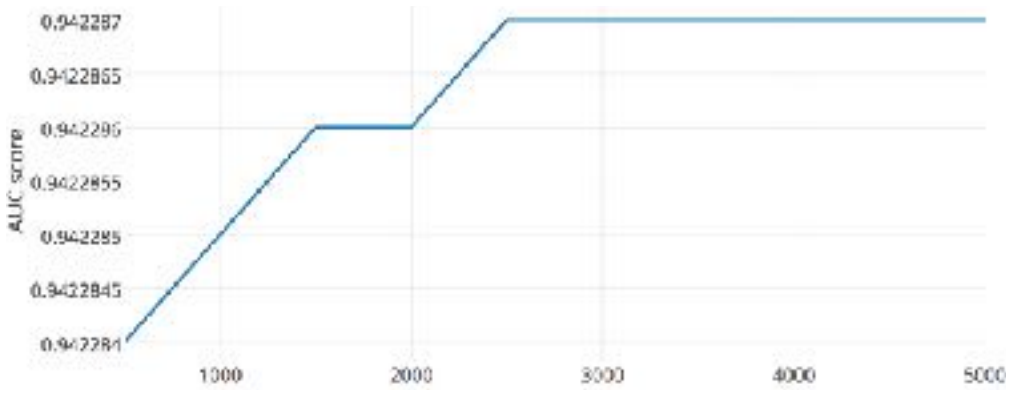
The window of 470 days of data in a stack of rolling 10 days worth of data, which is broken up. I simply take in the columns that I want, that the 10 opening days and 9 closing days worth of data. In both cases, I normalize each row of the dataset so that all stock prices are expressed as a ratio with the opening on day one. The x matrix contains rows of stock open and close prices and the y matrix contains indicators of whether or not the stock went up that

day. The total size of x matrix is 128920. 80% of the processed dataset is used for training and the rest of the dataset is used for testing. After normalize the original dataset, the sample was looked like:

				X				Y
1	1	0.99479167	1.01052632	…	0.98315789	0.99473684	0.99473684	1
2	1	0.99479167	0.996875	…	0.9625	0.975	0.95625	1
3	1	1.01220339	1.00644068	…	0.95924765	0.96551724	0.96238245	0
…	…	…	…	…	…	…	…	…
128918	1	1.01220339	1.00644068	…	1.12677966	1.1088135	1.10881356	1
128919	1	0.98450657	1.03873358	…	1.08858201	1.13270461	1.11384304	0
128920	1	1.02140078	0.96692607	…	1.07230869	1.0541504	0.93093385	1

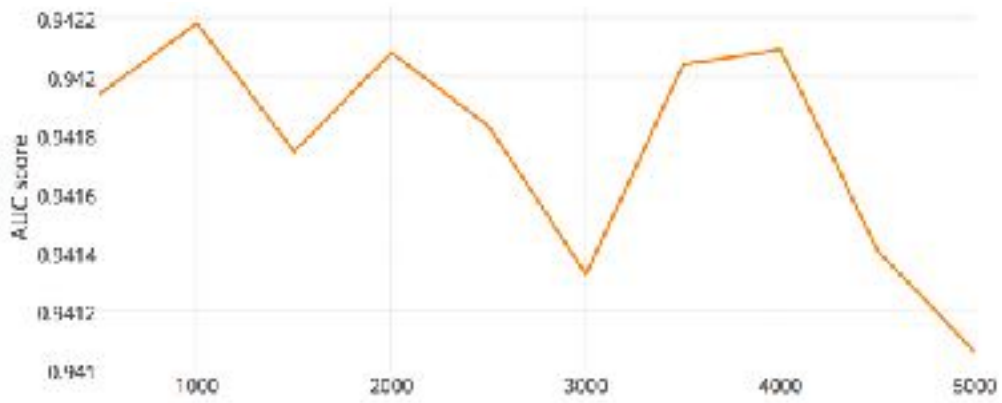
Implementation

After processing the data and gathering an X and a y, to find the cross validation scores of the best model from ridge, lasso, random forest and gradient boosting decision tree. By calculating the cross-validation scores for a number of tuning parameters, and then selecting the model and the tuning parameter that leads to the best cross-validation score. The tuning parameters of all the models as follows:



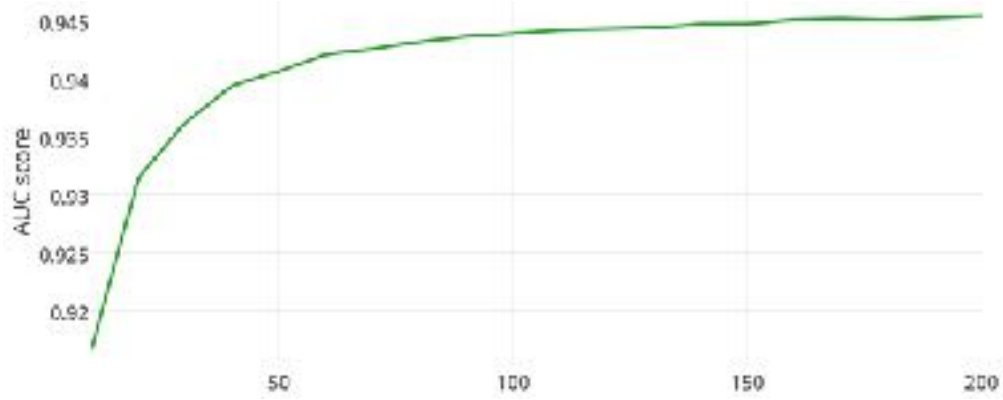
a. ridge

From the above, the x here is the list of tuning parameters that I want to test, I use the function "sklearn.linear_model.LogisticRegression" of scikit-learn package, the penalty is

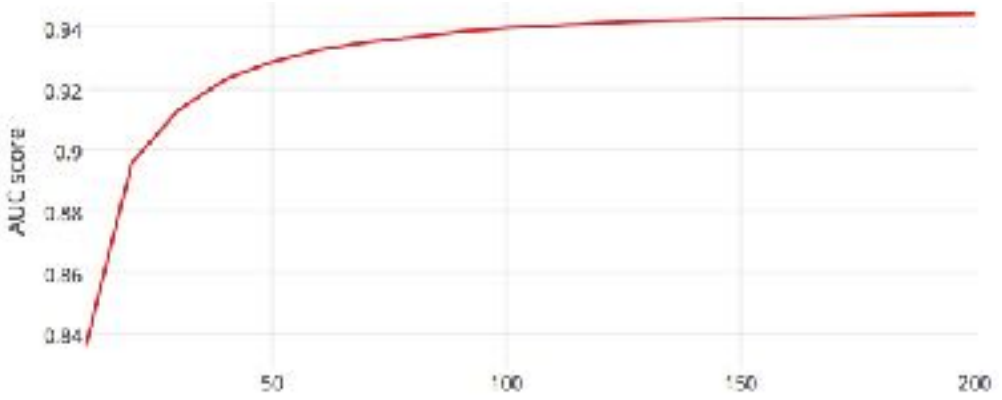


"l2", and the C that inverse of regularization strength, which I set the value from 500, 1000, 1500, 2000, 2500, 3000, 3500, 4000, 4500, 5000.

b. lasso



And figure b. lasso is as same as the ridge except the penalty is "l1". The figure c. random



forest, I use the function "sklearn.ensemble.RandomForestClassifier" of scikit-learn package. the parameter of n_estimators (the number of trees in the forest) , which I set the value from 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130, 140, 150, 160, 170, 180, 180, 190, 200. And the figure d. gradient boosting decision tree is as same as the random forest.

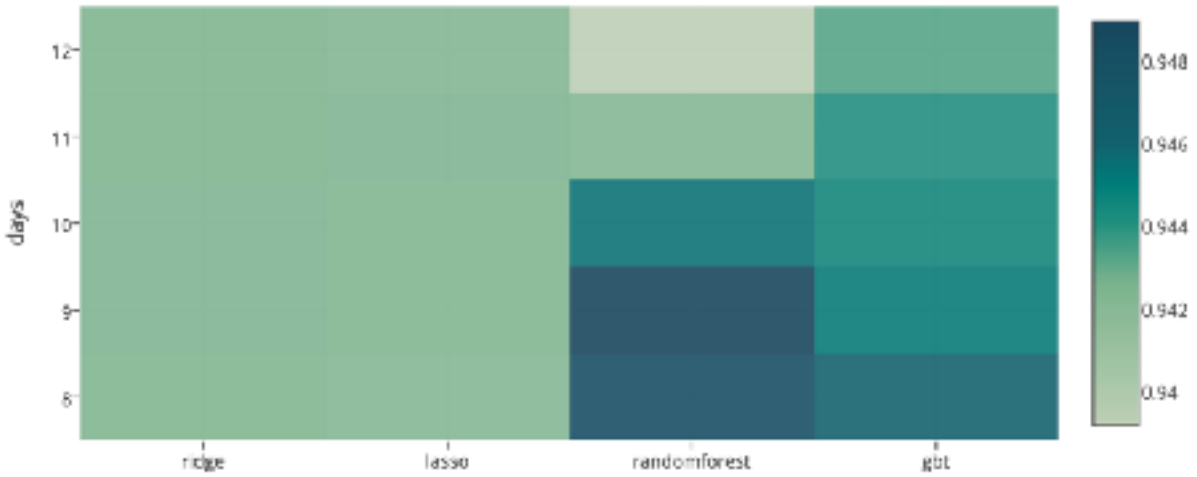
c. random forest

d. gradient boosting decision tree

Testing out the four different individual model and found the best parameters.

Refinement

At first, I has used 10 days of data to predict whether the closing price on the 10th day will be greater than or less than the opening price on the 10th day. And I want to know how many days of data to predict will be get higher AUC score, thus I select 8 days, 9 days, 11 days and 12 days to predict whether the closing price on the 8th, 9th, 11th and 12th days will be greater than or less than the opening price on the 8th, 9th, 11th and 12th day.



From the above figure, the score gap in different days of the model of ridge and lasso is very small. The best AUC score of the random forest is 9 days and the best AUC score of the random forest is 8 days. The highest score is 0.948922 which is the score of random forest.

Can I get higher scores? And I had the hypothesis that combining the models into some sort of ensemble would let us have better predictive performance in terms of AUC. The Platt Scaling[1] will be used. I place the probability estimates of each of our 4 models into a logistic regression model[4], to form a blended model that takes in contributions from all of the input models. I want use platt scaling which is a way of transforming outputs of a classification model into a probability distribution over classes to blend the different model together to test the AUC score whether could be better.

To create out-of-sample estimates for each of the four models (e.g. the probability estimates for a stock cannot have been trained on the actual increase indicator for that stock), else the input would be subject to overfitting, and the model would reach inflated CV scores due to the actual answer being included in the training of the predictors. Then with the matrix of predicted probabilities from all of the models (new_X), and a reordered responses (new_Y),

The blended model can be made, access it via cross validation, fit a model for prediction, and examine the coefficients that blended model gives to each of the input models.

Finally the weights in ridge, lasso, random forest and GBDT for the HUSHEN 300 industrials stocks all in one datasets were 5.95108936, -2.48526746, 5.32113583, 0.00796277 respectively. To rescale these to become proportions instead of slopes, that each model gets assigned a ≈67.67%, ≈-28.26%, ≈60.50%, ≈0.09% weight respectively. Because the bias of the LASSO model was positively correlated with the biases from at least one of the other models. Thus a blended model that assigned negative weight to LASSO and positive weight to another model whose biases were positively correlated with those of LASSO helped us correct the biases and lead to a better model. Also ran the blended model for a blend of Ridge and RF. the weights were 3.71557897,5.3371647 respectively. The proportions is assigned to 41.04% and 58.96%. Now with our 9 datasets, 4 individual models, and 2 blended models, the modern data all still has around 89% AUC.

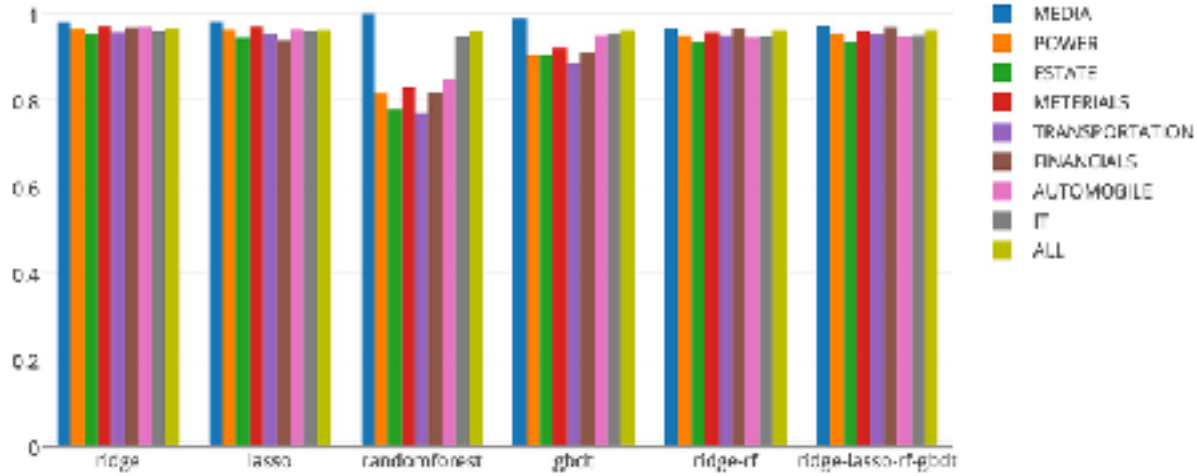
Results

Model Evaluation and Validation

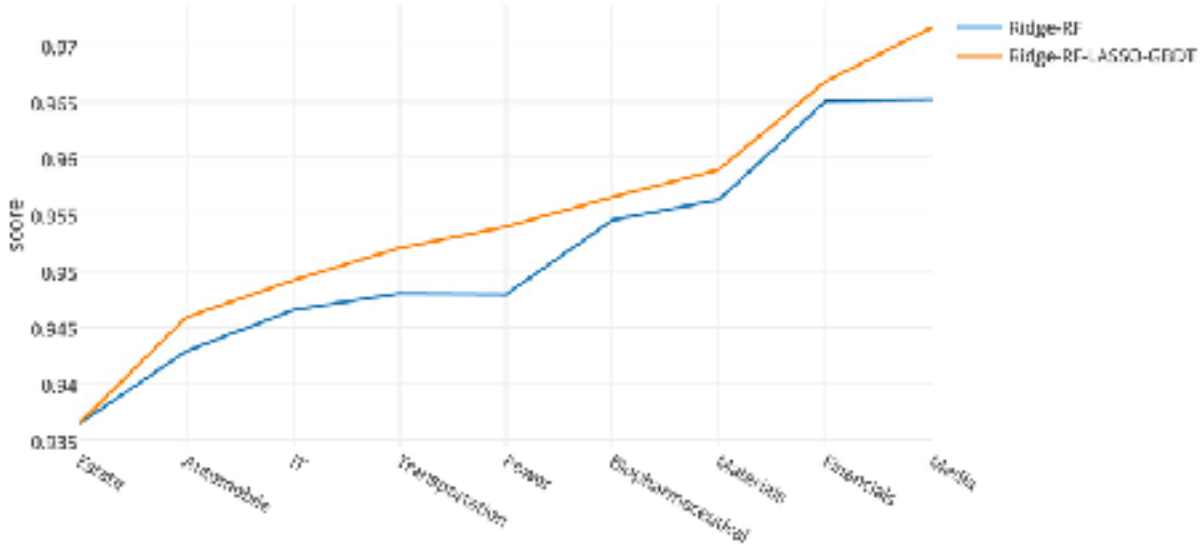
In this part, I use the function AUC to evaluation. The data set between 2015-01-01 and 2016-12-05 is used to train the model, and the data set between 2016-12-06 and 2017-03-08 is used to test the model:

- Use all of the HUSHEN 300 stocks to train the model;
- Use the different industries data to test the model.

Take a look at several models' testing AUC score on various subsets of data as below.



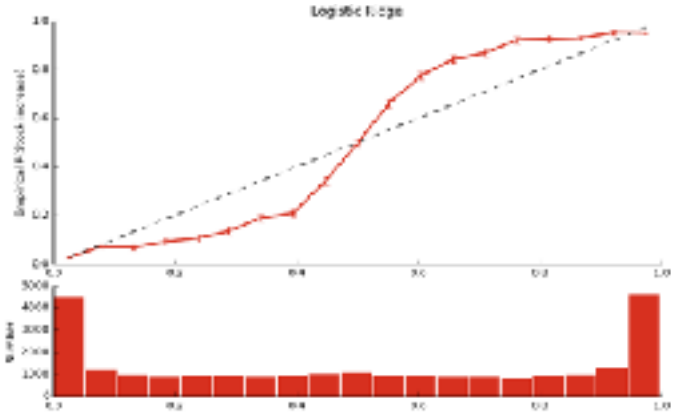
As shown in the above, the random forest performed the worst in all of the individual model. Ridge regression performed better than all the other model because the features were fairly linear with respect to the logit probability, and lasso did do as well as lasso. The blended models did do as well as ridge, and the blend of the ridge and random forest models perform just a little less than the blend between all four individual models. I use curve plot to distinguish the performance difference between Ridge-RF and Ridge-RF-LASSO-GBDT.

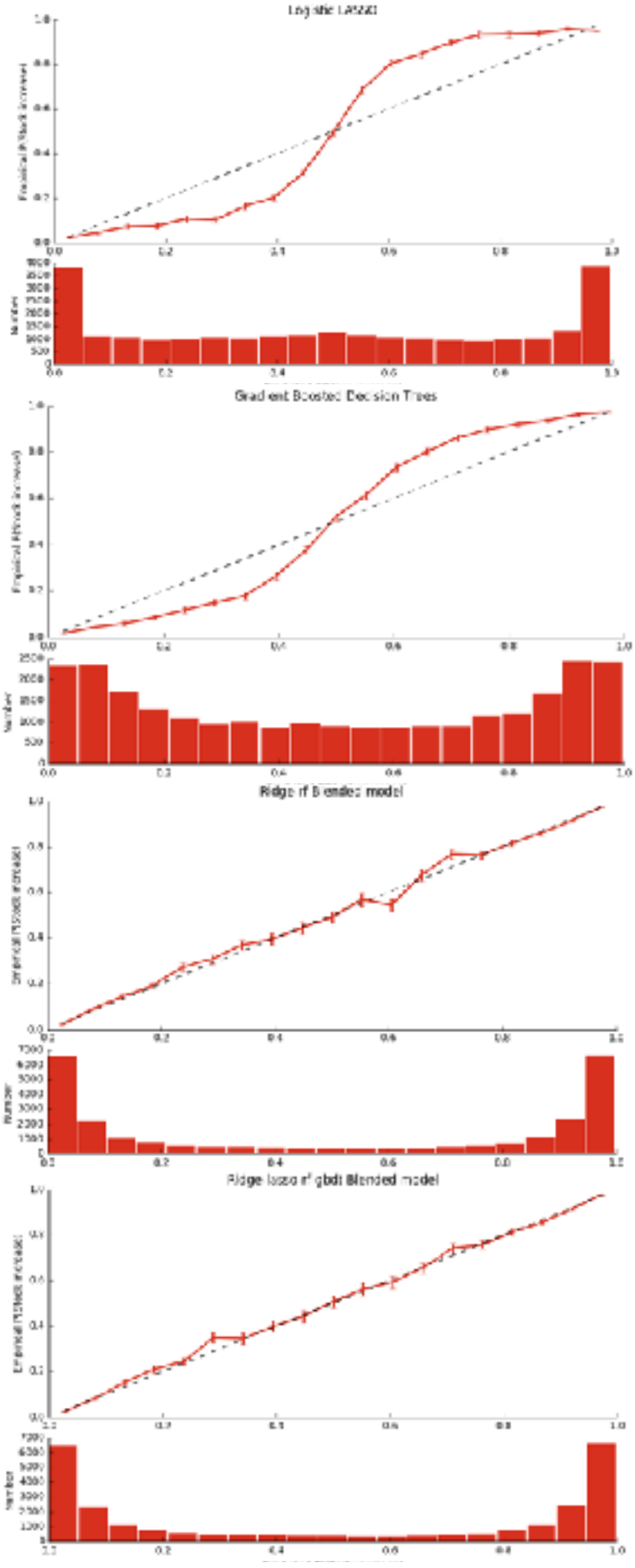


Justification

In order to understand why the blended model performed performed better, the calibration plots for each of models will be produced. To understand out of sample prediction, the models have be trained on the training set and the calibration model has been produced for the test set.

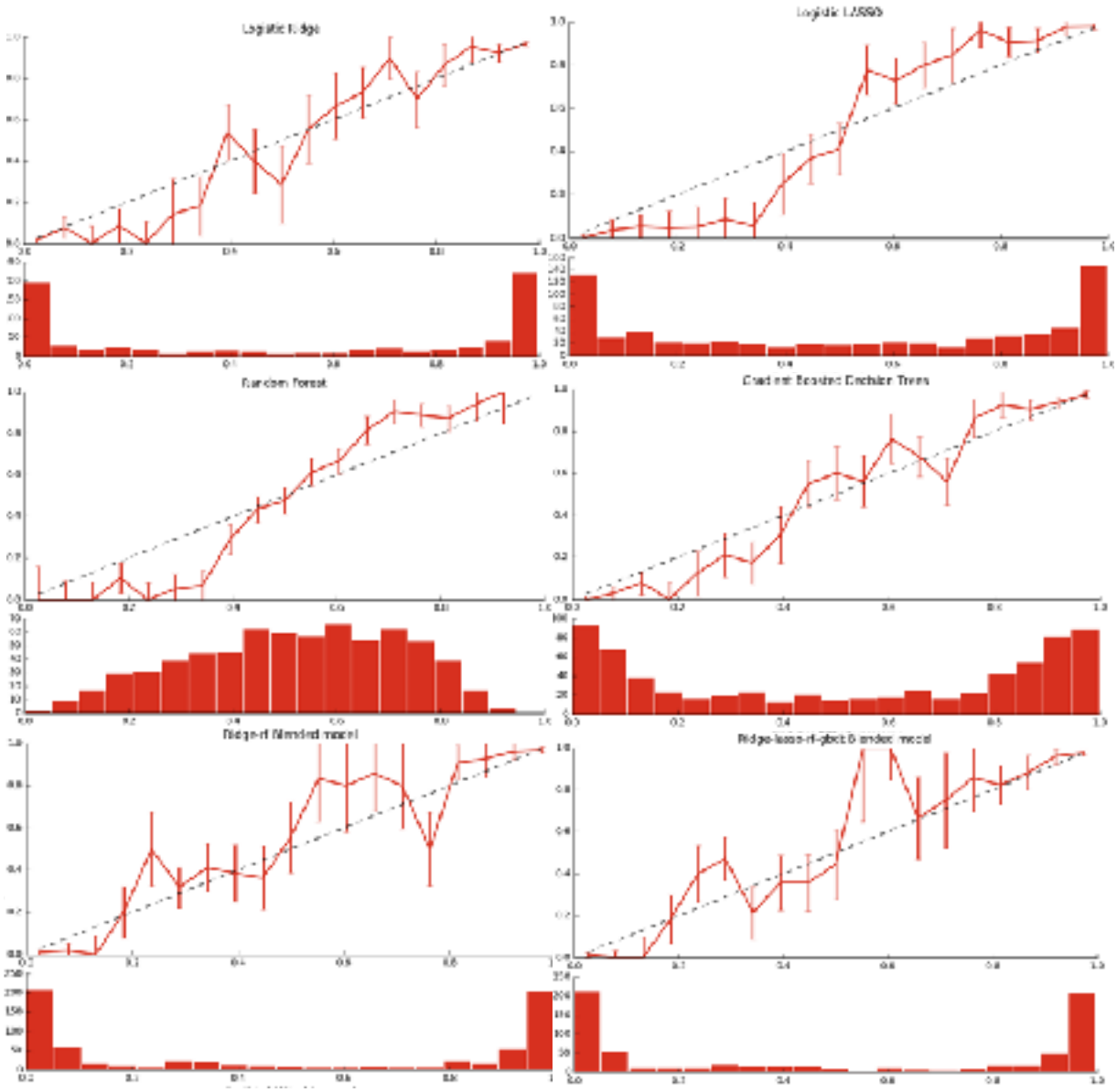
As shown below, we see that regularization models (lasso and ridge) under the elastic net regularization spectrum appear to yield predictions that mostly include extreme values. The calibration plot tends to indicate that predictions tend to be confident, meaning that many of the predictions are of probabilities near 0 and 1. The calibration plot also indicates that for the predictions that are not in the extreme ranges, our model is biased towards predicting




Gradient Boosted Decision Trees
Ridge if Blended model
Ridge lasso rf gbdt Blended model

probabilities closer to 50%, as the predicted probabilities are closer to 50% than the empirical probabilities. And the blended models have better calibration than the individual models. The individual models present bias as in general they are overconfident or under confident about their predictions. Therefore, is that the blended models help average out those biases and make more well-calibrated models. the blended models perform better than our individual models with respect to AUC.

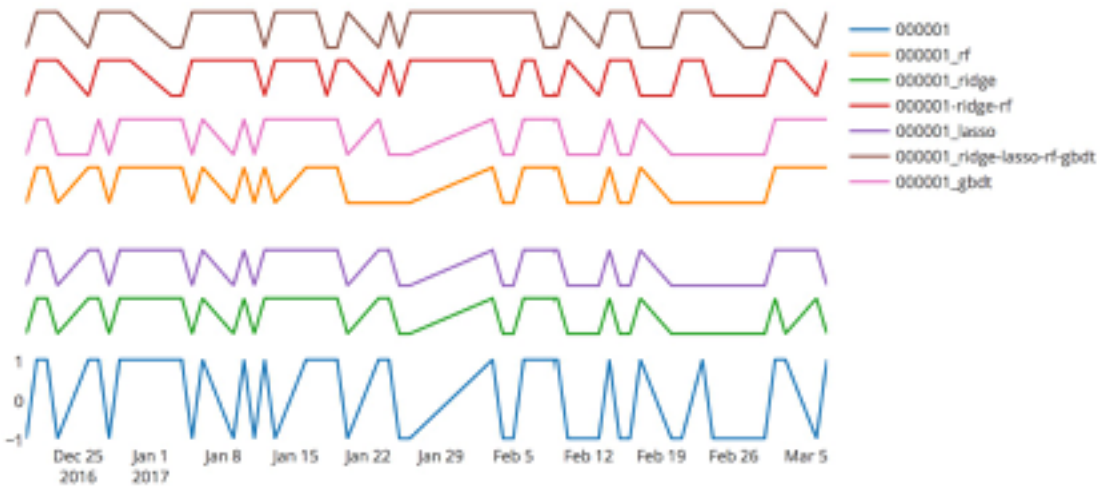
For a specific about 330 stocks in market, I test my model between 2014-01-01 and 2015-01-01, and the predict result is not very well. Most of these result of AUC score is below 0.5, but the industries result is better than the specific stock. I analysis the sample of the The industry of transportation, the prediction accuracy as shown in the figure below.

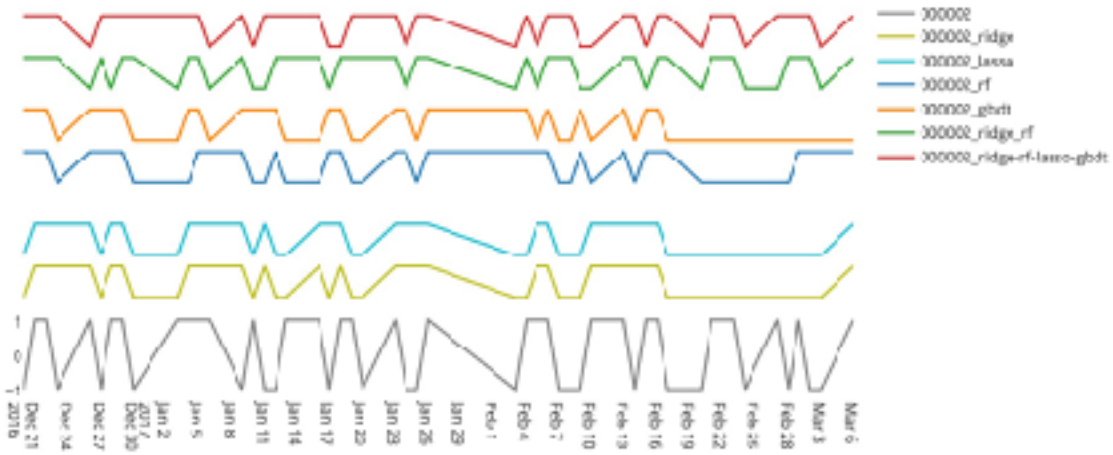


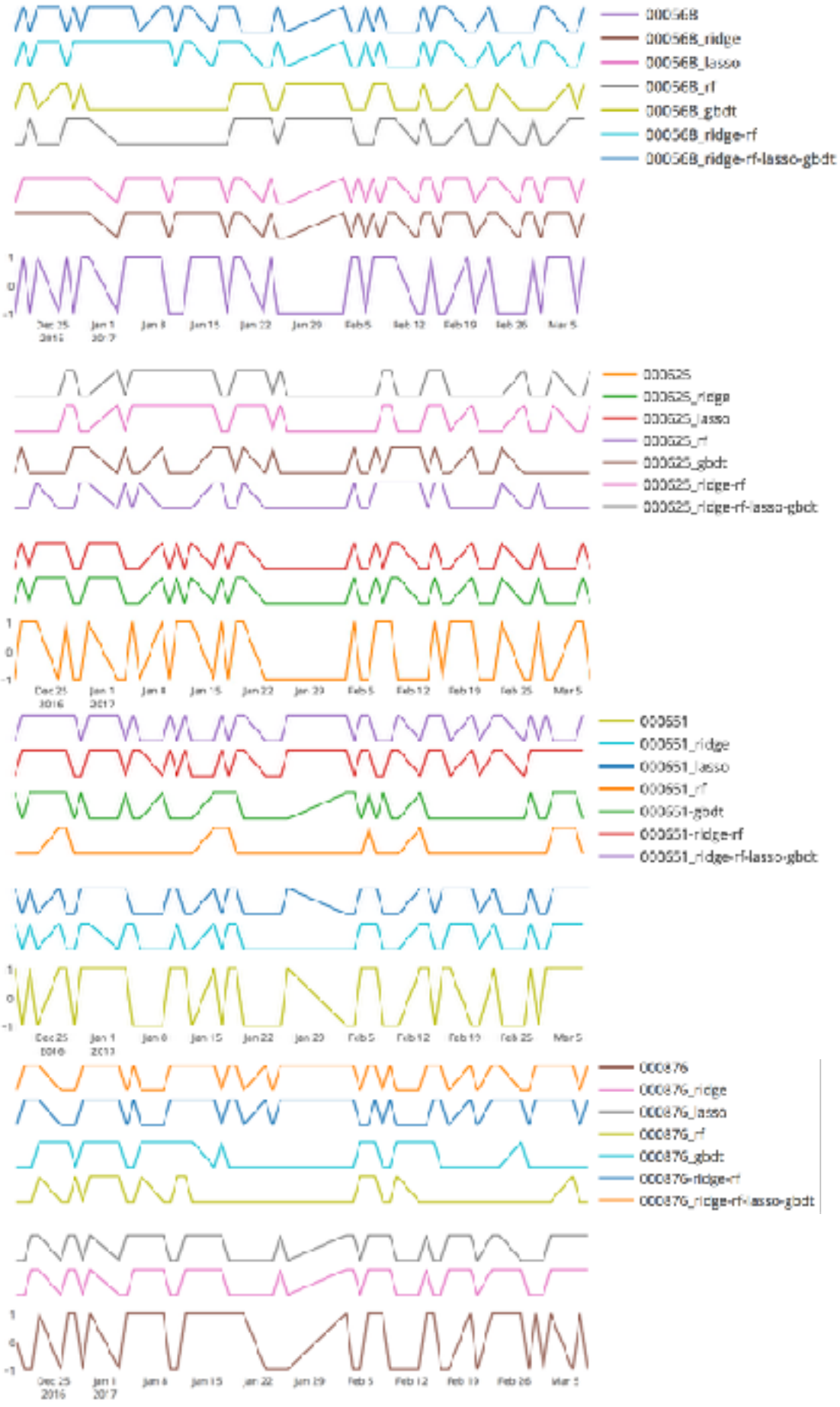
Conclusion

Free-Form Visualization

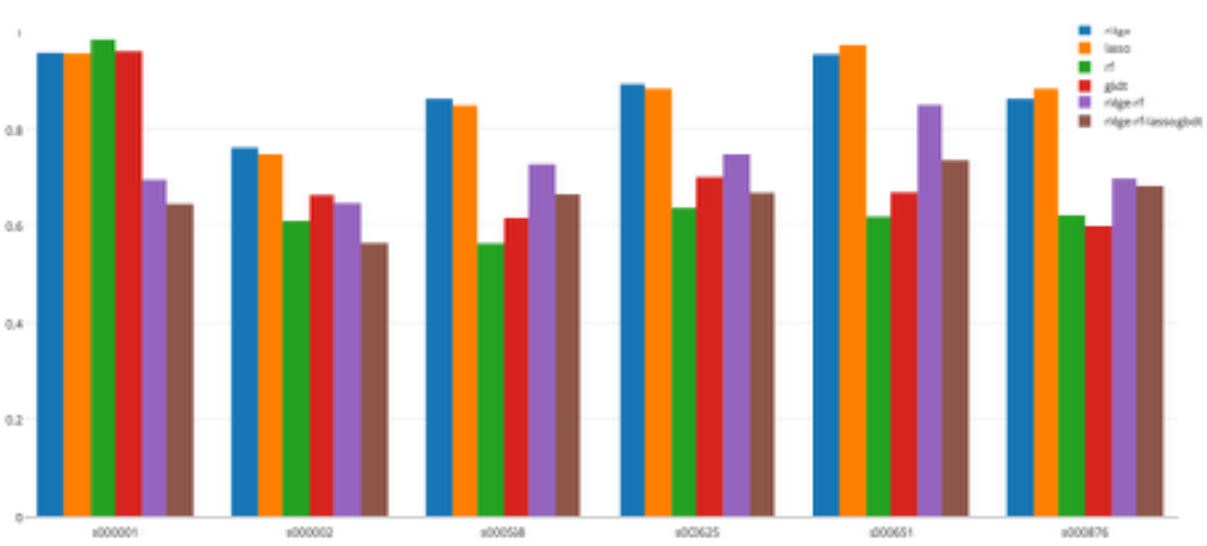
I randomly choose 6 China's stocks from different industries to test the six different model. All models are classification model which can only predict movements. The shares of stock code is 000001, 000002, 000568, 000625, 000651 and 000876. As shown in the figure below, look from the y axis, the value "1" means that the higher, on the contrary the value "-1" means that fall, each figure from down to up is the real movement (rise and fall) of stock, ridge , lasso, random forest, GBDT, the blended model between ridge and random forest and the blended model for for different models, which is between 2016-12-06 and 2017-03-08.






000568
000568_ridge
000568_lasso
000568_rf
000568_gbdt
000568_ridge-rf
000568_ridge-rf-lasso-gbdt
000525
000525_ridge
000525_lasso
000525_rf
000525_gbdt
000525_ridge-rf
000525_ridge-rf-lasso-gbdt
000551
000551_ridge
000551_lasso
000551_rf
000551-gbdt
000551-ridge-rf
000551_ridge-rf-lasso-gbdt
000876
000876_ridge
000876_lasso
000876_rf
000876_gbdt
000876-ridge-rf
000876_ridge-rf-lasso-gbdt

As shown in the figure below, the AUC score of ridge and lasso is better than the other models in most cases, and the random forest and GBDT is the worst. The lowest score all of the models is greater than 0.5967. The result of predict the individual stocks is inferior to predict the result of the industries, however the individual stocks can be predictable.



Reflection

From the heat map as shown in the above, that it is possible to achieve 88%+ AUC when trying to predict whether a closing price of a stock is greater than or less than the opening price of a stock, so direction movements of stocks are predictable. However that this does not directly correspond to accuracy, and we do not test the direct accuracy of predictions. The simple layman interpretation of AUC, however, is that if I were given a random data point where the closing price is higher than the opening price and another where the opposite is true, that would be able to correctly differentiate the two with over 88% accuracy. This may come at a surprise to many familiar with the stock market. However, I analyze the prediction result for all stocks of a specific day after 2016-12-02, and the score of AUC is less than 88%, even up to less than 55%. The usual response to hearing this result is to inquire as to the viability of a trading strategy focused on these patterns. Unfortunately, the predictability of directional movement does not translate to the certainty of returns, for the following several reasons:

a. things can go wrong fast, it may be possible that the trading strategy may yield high accuracy, but when it's wrong, then we lose big.
b. trading strategies focus on the actual magnitude of the movement of a stock, not just the direction.

c. the stock transaction fees that basically may wipe away the daily returns.

So in the actual operation, many factors must be considered. And the markets may overreact to events that happen during closing hours, and then correct themselves, thus the market corrections and momentum effects drive the predictability of directional movement, that would explain any negative correlation.

Improvement

In this project, First, I use 10 days of data to predict whether the closing price on the 10th day. If I use less or more days data to predict closing price on that day, may I have different results.

Second, I blended just only these models, and these models is more or less from the calibration plots. If I use more models, the result might be better. Finally, I did not consider all of stocks in different individual industry and I did not use some of the stock metrics or theory, for example, MA, MACD, BOLL RSI or Gann theory. If I combine them, may be we could predict the price of stock.

Reference

[1] Market correction: https://goo.gl/5WQl6G

[2] Momentum effect: https://goo.gl/SOxU8H

[3] AUC & ROC: https://goo.gl/8nqoPW

[4] Logistic regression: http://dataunion.org/20514.html

[5] Platt scaling: https://en.wikipedia.org/wiki/Platt_scaling

[6] GBDT: https://goo.gl/ZjzoTJ