# Using Market and News to Detect Stock with Positive Market-Residualized Return

## Abstract

Detecting the assets with extra return compared to the broad market has been an area of interest for both researchers, investors and speculators due to the complexity and dynamic nature of stock market. To address the question as to how to model financial information from two under big data background, we focus on improving the accuracy and decreasing the time-space complexity of existing model. The input set is a mix of market and news data. Using a fusion LightGBM with parameter optimization, experiments were conducted, and results have shown that the model proposed can process huge dataset in very small time window and successfully detect assets with remarkably outstanding extra returns.

**Keywords**: Stock market prediction; Data mining; Value discovery; LightGBM

## 1 Introduction

Stock market plays increasingly significant and active role in nowadays financial market. Either investors or speculators in the market are looking forward to gaining better profit by analyzing the information. Efficient Market Hypothesis, proposed by Fama [1], elaborates that stock prices have already included and revealed all the information in the market, and that random walk seems to be the most natural and possible way the stock market should behave. However, Grossman-Stiglitz Paradox proved that market efficiency and competition equilibrium are incompatible due to the existence of information costs, and prices cannot be fully displayed [2]. If the price is valid, no one would spend the cost to collect the information and bear the upfront risks; and if no one is going to get the information and decide his demand accordingly, the new information cannot be aggregated or reflected as quickly as possible. In other words, in order to gain an advantage in competition, it is necessary to develop more useful models and improve information mining techniques.

Approaches have long been studied to analyze vast amount of financial information, such as past stock prices. On the other hand, news contains potential information. If we can use this information reasonably, it may provide a new perspective for us to understand the market. Can we use the content of news analytics to predict stock price performance? Some research shows that stock prices appear to drift or reversal after important news [3]. The ubiquity of data today enables investors at any scale to make better investment decisions. For investors such as venture capitals (VC), information content of text can reflect a subject’s intrinsic value, potential of future growth and so on. However, modeling and analyzing market information has becoming an interesting but challenging problem with such a big volume of data nowadays.

To assist investors’ decisions, more and more institutions rely on advanced technology for information analysis, e.g. Two Sigma has been applying technology and data science to financial forecasts for over 17 years. Their pioneering advances in big data, artificial intelligence (AI), and machine learning have pushed the investment industry forward. In literature several models with computational intelligent techniques are available for prediction of stock price movements[9].

In order to aggregate more than one information sources, including market data such as volume, opening and closing price, raw and adjusted return on 1 or 10 days as well as news data such as audiences, providers, content and so on, into stock prediction system, through exploratory data analysis to decide the features to use. After learning the weights for features, the combined model gives prediction that is supposed to be more accurate than the existing models that based on only fundamental or technical indicators.

The rest of this paper is organized as follows. Section 2 gives a review on major existing researches related. Section 3 introduces experimental design, including dataset, method and data cleaning. Experimental results are reported in Section 4. The conclusion and future work are given in Section 5.

## 2 Related Work

There have been many empirical researches on the predicative power of machine learning methods concerning stock market.

Kim et al. (2000) proposed a new hybrid model of genetic algorithm with artificial neural network (ANN) to measure the technical indexes of the stock market with feature discretization [16]. After then many researchers further developed the model with fuzzy inference rules [17], Hidden Markov Model [18]. Guresen et al. (2011) compared several neural network (NN) models and hybrid models with respect to Mean Square Error and Mean Absolute Deviate and found Multilayer Perceptron gives best results [19].

Ensemble algorithms such as Random Forest (RF) perform more optimistic against single classifiers models including Support Vector Machines (SVM), NN in a number of recent works. Ticknor and Jonathan (2013) found that RF is the best then SVM, Kernel factory, AdaBoost, NN, K-NN and logistic Regression [21]. Similarly, Patel et al. (2015) compared the use of ANN, SVM, RF and Naïve Bayes on prediction stock movement and drew good results with RF [22]. Similarly, Ballings et al. (2015) also concluded that RF ensemble method should be used for stock price direction prediction by comparing SVM, NN and Logistic Regression [23]. Kolanovic and Krishnamachari (2017) from J.P Morgan discussed and compared SVM, ANN, Convolutional Neural Networks (CNN), Long Short-Term Memory networks (LSTM) and XGboost, their results indicated that XGboost gives the best performance [24]. Kohli et al. (2018) found the AdaBoost algorithm performed best as compared to other machine learning techniques [25].

Previous ensemble algorithms work better in respect of accuracy. However, these implementations are very time consuming when handling big data [26]. Nowadays, the ever-increasing volume, velocity and variety (3V’s) of financial data push capital firms to invest in ways to make big data more manageable and turn large amounts of information into actionable insights to maintain a competitive edge [27][26]. Real-time financial information counts [28]. Guolin et al. (2017) from Microsoft proposed LightGBM to deal with large number of data instances and large number of features respectively. The experimental results showed that LightGBM can signiﬁcantly outperform other ensemble algorithms such as XGBoost in terms of computational speed and memory consumption [26].

Given that LightGBM has both advantage of ensemble algorithms and low computational space-time complexity, this paper decides to adopt LightGBM in test.

## 3 Experimental Design

### 3.1 Dataset

This paper will predict future stock price returns based on two sources of data provided by Kaggle [29] from 2007-01-01 to 2016-12-31, including 4072956 samples with16 features in the training market dataset and 9328750 samples with 35 features in the training news dataset.

Market data provided by Intrinio includes financial market information as Table 3-1 shows. Returns are always calculated either open-to-open (from the opening time of one trading day to the open of another) or close-to-close (from the closing time of one trading day to the open of another); are either raw, meaning that the data is not adjusted against any benchmark, or market-residualized (Mktres), meaning that the movement of the market as a whole has been accounted for, leaving only movements inherent to the instrument; can be calculated over any arbitrary interval. Provided here are 1-day and 10-day horizons; are tagged with 'Prev' if they are backwards looking in time, or 'Next' if forwards looking.

The news data contains information at both the news article level and asset level, including source timestamp, first created time, headline, urgency, take sequence, provider, subjects, audiences, body size, company count, headline tag, market commentary afterwards, sentence count, word count, first mentioned sentence, relevance, sentiment class, The 12-hours, 24-hours, 3-days, 5-days, 7-days novelty and maintained of the content within a news item on a particular asset.

### 3.2 Data Cleaning

In order to train LightGBM with two different sources of data, the raw datasets need to be reconstructed into what the algorithm needs.

#### 3.3.1 Exploratory Data Analysis

To better processing the data, having an overall knowledge of the datasets through exploratory data analysis is needed.

First, We have a glance at the number of market and news data everyday respectively as Fig.3-1 and Fig.3-2 shows.



Fig.3-1 Number of Market Data Everyday

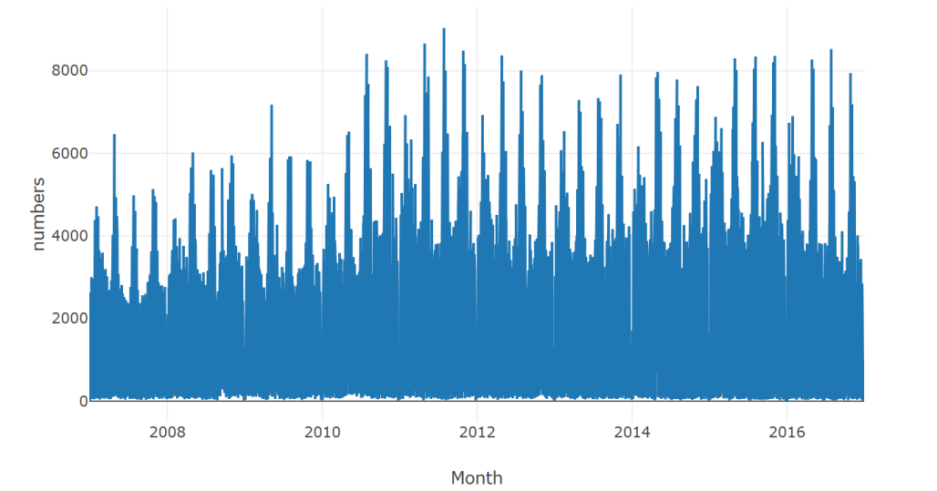


Fig.3-2 Number of Market Data Everyday

From Fig.3-1, something seems to be missing in market data around May 2014. We check the data by analyzing the trading volume by date as Fig.3-3 shows.

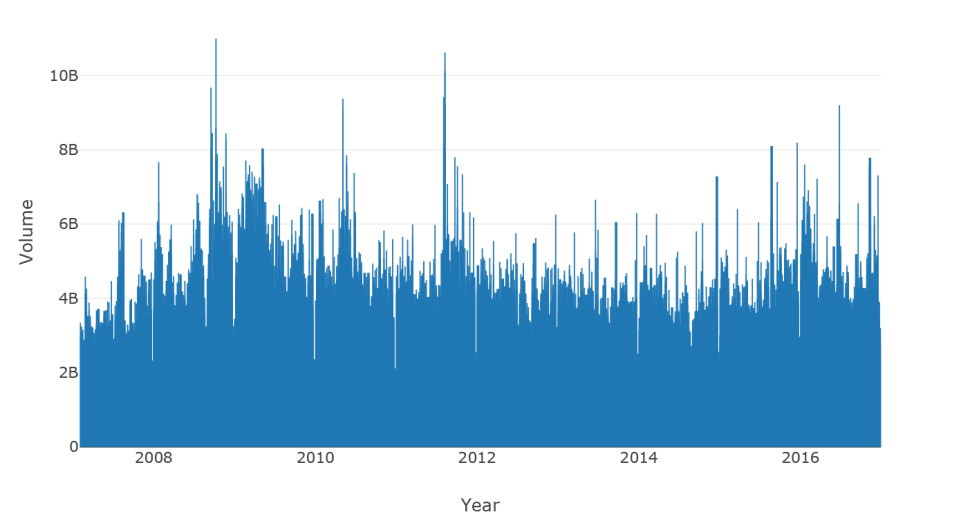


Fig.3-3 Trading Volume Everyday

The volume varies remarkably with a mean of 2.665312e+06, a standard variance of 7.687606e+06, the minimum of 0.000000e+00, the 25% of 4.657968e+05, the 50% of 9.821000e+05, the 75% of 2.403165e+06 and the maximum of 1.226791e+09. There is no exception around May 2014 concerning trading.

We also find that there are 110 unique assets without assetName; average standard deviation of price change within a day in 1.0335; 15980 returnsClosePrevMktres1, 15980 returnsOpenPrevMktres1, 93010 returnsClosePrevMktres10 and 93010 returnsOpenPrevMktres10 are missed.

The correlation between noveltyCount and volumeCounts concerning 12H, 24H, 3D, 5D,7D, 24H-12H, 3D-24H, 5D-3D, 7D-5H is 0.77, 0.67, 0.47, 0.41, 0.38, 0.27, 0.29, 0.31, 0.33 respectively. And the frequently mentioned words are as Fig.3-4 shows.



Fig.3-4 Word Cloud of News Data

#### 3.3.2 Data Processing

Specially, the target is returnsOpenNextMktres10. From the correlation matrix, We aggregate the news data’s quantitative scores by mean and combine them with market data by time and assetCode, then check the correlation matrix of the numerical scores as Fig.3-5 shows.

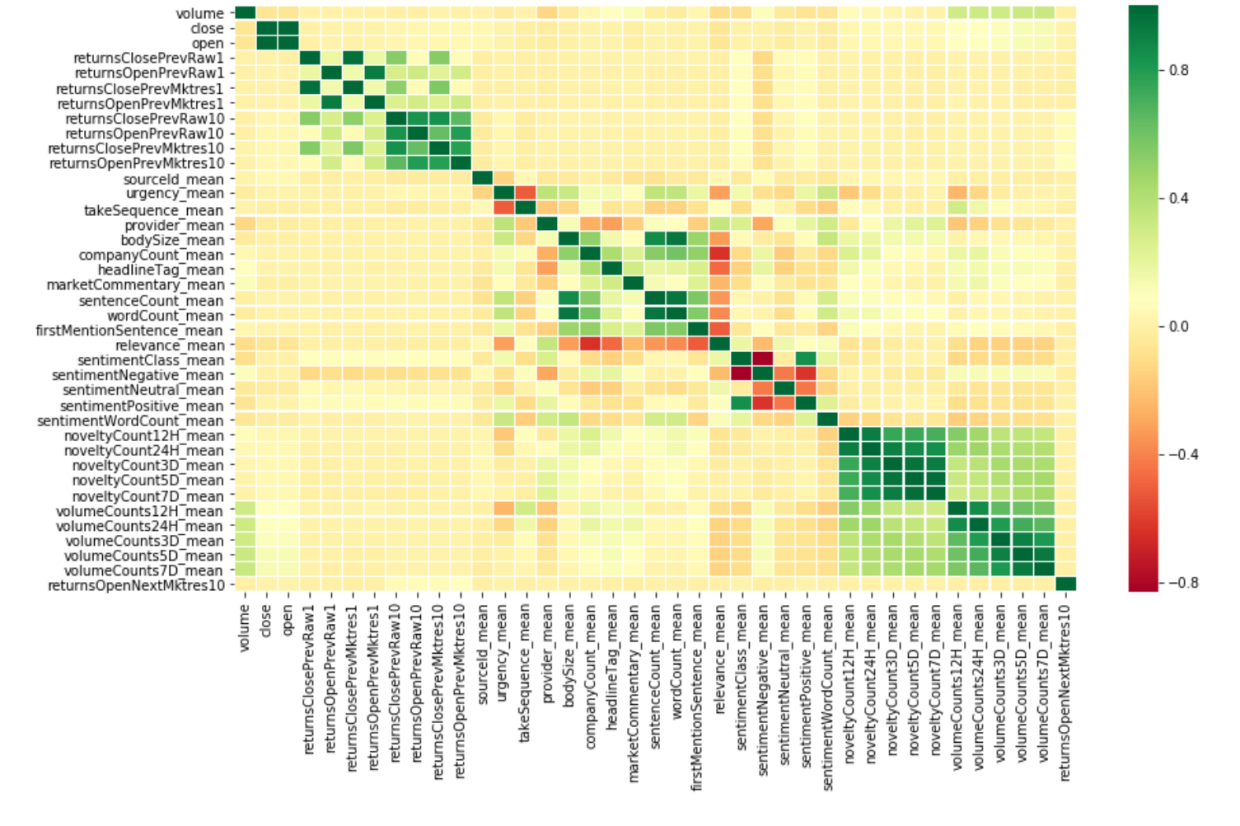


Fig.3-5 Correlation Matrix of Variables

We can find that the correlations are relatively low. EFB will help us to reduce the low contribution features. I fill the missing value with the mean of the feature it belongs. In order to predict the adjusted returns in 10-day horizon, I set a time window of 10 days, and create six new features: highest, lowest, mean close and open price of each assetCode during the time window. Finally, I aggregate the market and news data by assetCode every day and Z-score standardization [30], the processed data conforms to the standard normal distribution, that is, the mean value is 0 and the standard deviation is 1. The conversion function is:

|  |  |  |
| --- | --- | --- |
|  |  | (3-3) |

where and is the mean value and standard variation of the feature subject to.

### 3.3 Variable Definition and Description

After data cleaning and exploratory data analysis, the explained variable, explanatory variables and control variable are designed as Table 3-1 shows.

 Table 3-1 Research Variables Description

|  |  |  |  |
| --- | --- | --- | --- |
| Research Variable | | Symbol | Definition and Description |
| Explained Variable | Market - Residualized Return | Returns  OpenNextMktres10 | 10 day open price return, are market-residualized (Mktres), meaning that the movement of the market as a whole has been accounted for, leaving only movements inherent to the instrument. |
| Explanatory Variables | Historical Performance | open | The open price for the day (not adjusted for splits or dividends), along with highest, lowest and mean open price of 3,7,14 days. |
| Close | Same as above, but close price. |
| returnOpen | Returns are calculated open-to-open over 1 day and 10 day horizons, either raw, meaning that the data is not adjusted against any benchmark, or market-residualized (Mktres) |
| returnClose | Same as above, but close price. |
| Explanatory Variables | News  Impact | Relevance | a decimal number indicating the relevance of the news item to the asset. It ranges from 0 to 1. If the asset is mentioned in the headline, the relevance is set to 1. |
| Sentiment  Class | indicates the predominant sentiment class for this news item with respect to the asset. The indicated class is the one with the highest probability. |
| Control Variables | Trading Scale | volume | The amount of an asset every day |
| News  Scale | bodySize | The size of the current version of the story body in characters |
| Sentence  Count | The total number of sentences in the news item. |
| Story  Types | urgency | differentiates story types  (1: alert, 3: article) |
| Update of News | Novelty  Count | The 12-hours, 24-hours, 3-days, 5-days, 7-days novelty of the content within a news item on a particular asset. It is calculated by comparing it with the asset-specific text over a cache of previous news items that contain the asset. |
| Volume  Counts | Same above, but remain unchanged |

### 3.4 LightGBM

Guolin et al. (2017) proposed a novel GBDT algorithm called LightGBM with two novel techniques: Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) [26].

#### 3.2.1 Gradient-based One-Side Sampling (GOSS)

To address the previous limitation of Gradient boosting decision tree (GBDT) which is an ensemble model of decision tree with the histogram-based algorithm not having efficient sparse optimization solutions. Guolin et al. perfected the previous pre-sorted algorithm with GOSS.

|  |
| --- |
| **Algorithm 1:** Gradient-based One-Side Sampling (GOSS) |
| **Input:** I: training data, d: iterations  **Input:** a: sampling ratio of large gradient data  **Input:** b: sampling ratio of small gradient data  **Input:** loss: loss function L: weak learner  models ← {}, fact ←  topN ← a×len(I), randN← b×len(I)  **for** i = 1 **to** d **do**  preds ← models.predict(I)  g ← loss(I, preds), w ← {1,1,…}  sorted ← GetSortedIndices(abs(g))  topSet ← sorted[1:topN]  randSet ← RandomPick(sorted[topN:len(I)]), randN)  usedSet ← topSet + randSet  w[randSet] ×= fact ▷ Assign weight fact to the small gradient data  Newmodel ← L(I[usedSet], － g[usedSet]), w[usedSet])  models.append(newModel) |

Let be a set of vectors with dimension of input space. In each iteration of gradient boosting, the negative of the loss function with respect to the output of the model are donated as . First, rank the training instances according to their absolute values of their gradients in the descending order; second, keep the top-*a*×100% instances with the larger gradients and get an instance subset A; then, for the remaining set consisting (1 - *a*)×100% instances with smaller gradients, further randomly sample a subset *B* with the size *b*×; finally split the instances in accordance with the estimated variance gain over the subset .

|  |  |
| --- | --- |
|  | (3-1) |

where , , , and the coefficient is used to normalize the sum of the gradients over *B* back to the size of . Donate the approximation error in GOSS as and , . With probability over 1-δ:

|  |  |
| --- | --- |
|  | (3-2) |

where , and .

#### 3.2.2 Exclusive Feature Bundling (EFB)

To effectively reduce the number of features, EFB is introduced.

|  |
| --- |
| **Algorithm 2:** Greedy Bundling |
| **Input:** F: features, *K*:max conflict count  Construct graph *G*  searchOrder ← *G*.sortByDegree()  bundles ← {}, bundlesConflict← {}  **for** *i* ***in*** *searchOrder* **do**  needNew ← True  **for** *j* ***in*** *len(bundles)* **do**  cnt ← ConflictCnt(bundles[j], *F*[i])  **if** *cut+bundlesConflict[i]*≤*K* **then**  bundles[j].add(*F*[i)], needNwe← False  break  **for** *needNew* **then**  Add *F*[i] as a new bundle to *bundles*  **Output:** *bundles* |

High-dimensional data are usually very sparse. The sparsity of the feature space provides us a possibility of designing a nearly lossless approach to reduce the number of features. Speciﬁcally, in a sparse feature space, many features are mutually exclusive, i.e., they never take nonzero values simultaneously. We can safely bundle exclusive features into a single feature (which we call an exclusive feature bundle). By a carefully designed feature scanning algorithm, we can build the same feature histograms from the feature bundles as those from individual features. In this way, the complexity of histogram building changes from O(#data×#feature) to O(#data×#bundle), while #bundle << #feature. Then we can signiﬁcantly speed up the training of GBDT without hurting the accuracy. In the following, I will show how to achieve this in detail.

|  |
| --- |
| **Algorithm 3:** Merge Exclusive Features |
| **Input:** *numData*:number of data  **Input:** *F*: One bundle of exclusive features  binRanges← {0}, totalBin ← 0  **for** *f* ***in*** *F* **do**  totalBin ← f.numBin  binRanges.append(totalBin)  newBin← new Bin(numData)  **for** *i*=1 ***to*** *numData* **do**  newBin[0] ← 0  **for** *j*=1***to*** *len(F)***do**  **if** *F[j].bin[i]*≠0**then**  newBin[i] ←F[j].bin[i] + binRanges[j]  **Output:** *newBin, binRanges* |

## 4 Experimental Results

In Section 3, the datasets are processed to feed the LightGBM, in this section, I will optimize the parameter of the model by Grid Search with Cross Validation (GridSearchCV) [31]. Then, improve the performance of model by model fusion. Finally, test the model proposed by a score similar to Jensen index [32],

### 4.1 Parameter Optimization

After the original data set is divided into a training set and a test set, the test set is used to measure the quality of the model in addition to the adjustment parameters; this results in a final scoring result that is better than the actual result. By further divided into training sets and verification sets. One of the most efficient way to deal with this problem is further divide the training sets and verification sets [31].

The training set is used for model training, the verification set is used to adjust parameters, and the test set is used to measure the performance of the model. Fig.4-1 gives an overview of the process of parameter selection and model evaluation with GridSeachCV.

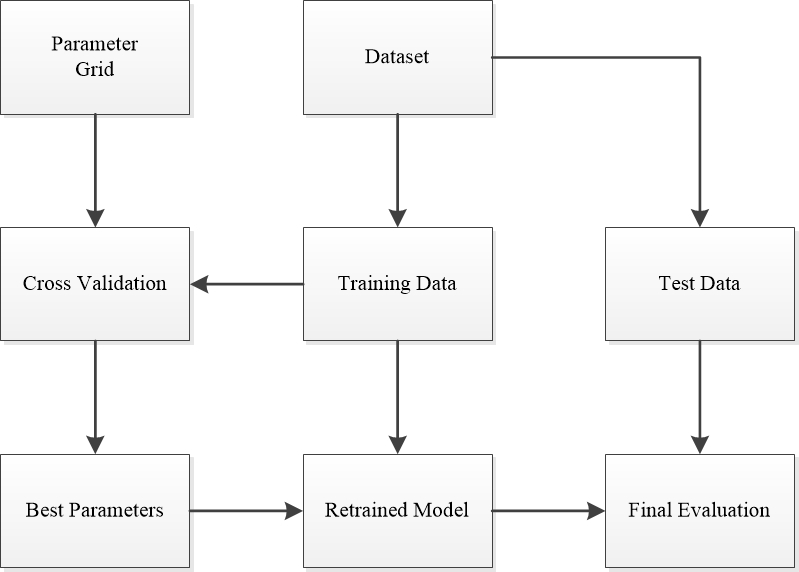


Fig.4-1 Overview of GridSearchCV

First, estimate the learning rate and estimator and its number based on the data scale. Since the considerable amount of data, learning at a small learning rate is extremely time consuming. Therefore, we test the log loss function under different learning rates and estimators as Table 4-1 shows.

Table 4-1 Tentative Result of Parameter Optimization

|  |  |  |
| --- | --- | --- |
| Learning\_reate | N\_estimators | cv\_score |
| 0.2 | 237 | 0.652111 |
| 0.3 | 46 | 0.671707 |
| 0.25 | 102 | 0.6648 |

We can find by setting the learning rate at 0.2 and estimators at 237, the cv score hit the lowest at 0.6521.

Then further improve the accuracy of the model. Considering GridSearchCV is relatively slow and easy to cause the dimension exploration when dealing with big dataset if we use it dierectly, therefore, we first use the coordinate reduction method to find the interval of possible optimal parameters and then apply GridSearchCV. Unlike XGBoost, LightGBM uses the leaf-wise algorithm, so when adjusting the complexity of the tree, num\_leaves is used instead of max\_depth. So first optimize the lightGBM parameter num\_leaves.After finding the optimal num\_leaves, we get a model with strong fitting ability, which will make the model score well on the training set, but the score on the test set will be worse, which is due to the over-fitting problem of the model.

Therefore, in the next step, we need to reduce the overfitting of the model. Min\_data\_in\_leaf The minimum number of data on a leaf. By setting the minimum amount of data on the leaf nodes, the method prevents the model from dividing a leaf node for a small amount of data, which reduces the over-fitting of the model and improves the generalization ability of the model. Max\_bin is the maximum number of bins that feature values will be bucketed in, the maximum number of histograms loaded by the eigenvalues. Generally, the smaller number of histograms will reduce the accuracy of the training but will improve the overall performance.

Finally, future improve the generalization ability through model bagging. By using the coordinate descent, the num\_leaves, min\_data\_in\_leaf, and max\_bin are coarse layered trained. Based on the GridSearchCV of the adjustment parameters, and finally the two models are retained for model fusion, which can further improve the generalization ability of the model and reduce over-fitting problem with a single model. The optimal parameters of two models are as Table 4-2 shows.

Table 4-2 Optimial Parameters of Two LightGBM

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Learing\_rate | Num\_leaves | Min\_data\_in\_leaf | Num\_iteration | Max\_bin |
| 0.190004 | 2452 | 212 | 239 | 202 |
| 0.190168 | 2583 | 213 | 172 | 220 |

Since the result of the output is a confidence level between -1 and 1, the result x1 of the model 1 and the result x2 of the model 2 are used together for the final prediction, and the average is taken, and finally normalized.

|  |  |  |
| --- | --- | --- |
|  |  | (4-1) |

### 4.2 Training Model

We feed the processed dataset into the model with the optimal parameters. The model converges after 239 epochs as Fig.4-2 shows.

Fig.4-2 Taring Loss

### 4.3 Model Evaluation

In previous part, We trained a fusion LightGBM with the market and news data from 2007-01-01 to 2016-12-31 to predict a signed confidence value, , which is multiplied by the market-adjusted return of a given assetCode over a ten-day window. In this part, We evaluate the model on a historical period from 2017-01-01 to 2018-07-31. 10 prediction samples are shown in Table 4-1.

Table 4-1 Part of Output

|  |  |  |
| --- | --- | --- |
| time | assetCode | confidenceValue |
| 2017/1/3 | A.N | 0.254909 |
| 2017/1/3 | AA.N | 0.249345 |
| 2017/1/3 | AAL.O | 0.588423 |
| 2017/1/3 | AAN.N | 0.708047 |
| 2017/1/3 | AAP.N | 0.449481 |
| 2017/1/3 | AAPL.O | -0.09762 |
| 2017/1/3 | ABB.N | 0.484631 |
| 2017/1/3 | ABBV.N | 0.420262 |
| 2017/1/3 | ABC.N | -0.54645 |
| 2017/1/3 | ABCO.O | 0.199280072 |

If you expect a stock to have a large positive return, compared to the broad market, over the next ten days, you might assign it a large, positive confidenceValue (near 1.0). If you expect a stock to have a negative return, you might assign it a large, negative confidenceValue (near -1.0). If unsure, you might assign it a value near zero.

For each day in the evaluation time period, We calculate:

|  |  |  |
| --- | --- | --- |
|  |  | (4-1) |

where is the 10-day market-adjusted leading return for day *t* for instrument *i*, and is a 0/1 universe variable (see the data description for details) that controls whether a particular asset is included in scoring on a particular day.

Final score is then calculated as the mean divided by the standard deviation of daily values:

|  |  |  |
| --- | --- | --- |
|  |  | (4-2) |

where if the standard deviation of predictions is 0, the score is defined as 0.

Though fusion LightGBM proposed, I find a model to detect firms with 0.70737 extra return on average every day and process the huge data in only 10 seconds.

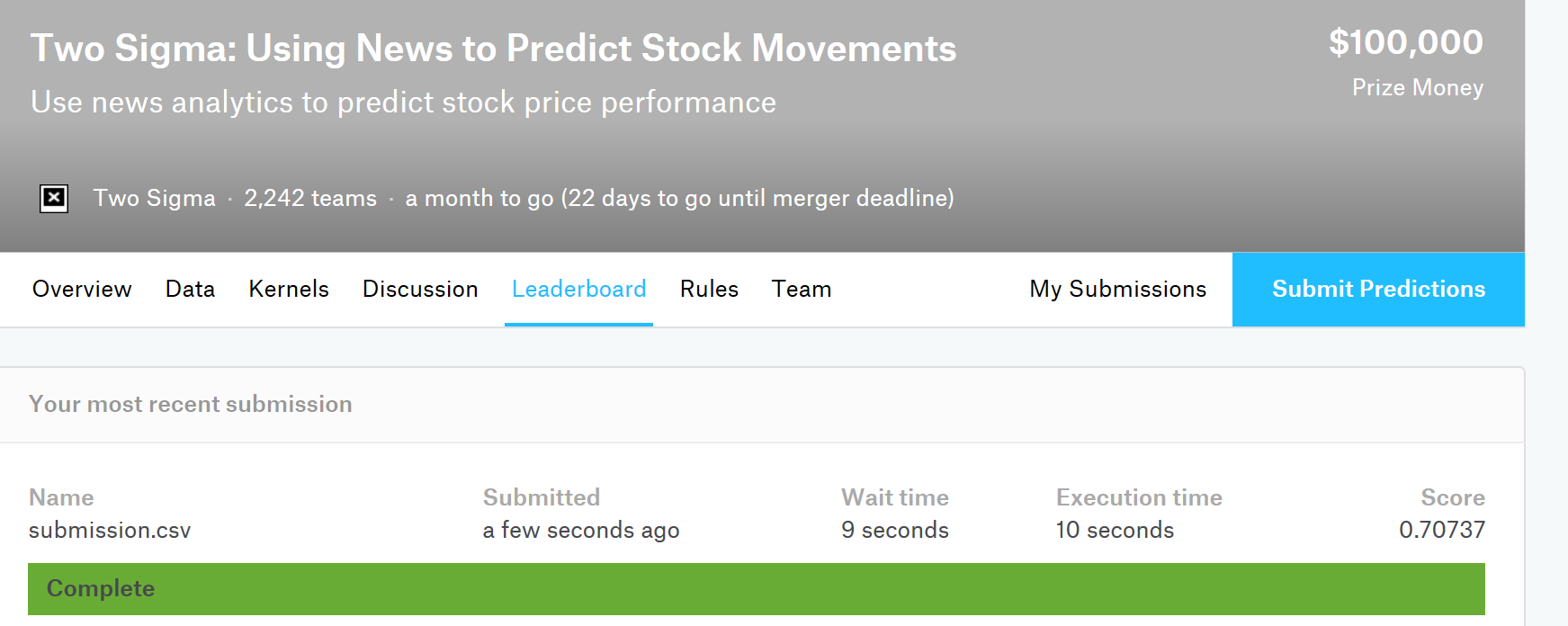


Fig.4-3 Evaluation Results

## 5 Conclusion and Further Work

This paper proposed an approach with high accuracy and low time-space complexity to predicting the price of stock and detect the positive return, compared to the broad market, over the next ten days using market and new data. To consider such events, I employed a fusion LightGBM with GridSearchCV optimization, specifically, a combination of two LightGBM with parameters optimized by Grid Search Cross Validation. The input data is a mix of market and news data from 2007-01-01 to 2016-12-31, including returns, volume, sentiment class and so on. To evaluate the effectiveness of the approach, I evaluated the trained model on a historical period from 2017-01-01 to 2018-07-31. After let the mean extra returns everyday divided by the standard deviation. I find a model to detect firms with 0.70737 extra return on average every day and process the huge data in only 10 seconds.

The model proposed can be used for devising new strategies for trading or to perform stock portfolio management, providing a new perspective for securities analysis, enabling analysts to find a profitable asset in a very small time window with enormous information. For future work, I will check the proposed model’s performance with other methods, such as SVM, NN, RF. The data used have already processed the news data into a structed state, next step is to extract the information in raw news myself as well as check the model with long-term duration.

To conclude, this paper verifies the effectiveness of ensemble model in general in the financial domain and suggest the potential of the LightGBM.

## Reference

1. Malkiel, Burton G., and Eugene F. Fama. "Efficient capital markets: A review of theory and empirical work." *The Journal of Finance* 25.2 (1970): 383-417.
2. Grossman, Sanford J., and J. E. Stiglitz. "On the Impossibility of Informationally Efficient Markets." *American Economic Review* 70.3(1980):393-408.
3. Mcqueen, Grant, and V. V. Roley. "Stock Prices, News, and Business Conditions." *Review of Financial Studies* 6.3(1993):683-707.
4. Ederington, Louis H., and Jae Ha Lee. "How markets process information: News releases and volatility." *The Journal of Finance* 48.4 (1993): 1161-1191.
5. Pearce, Douglas K., and V. V. Roley. "Stock Prices and Economic News." *Journal of Business* 58.1(1985):49-67.
6. Chan, Wesley S. "Stock price reaction to news and no-news: drift and reversal after headlines." *Journal of Financial Economics* 70.2 (2003): 223-260.
7. Boyd, John H., J. Hu, and R. Jagannathan. "The Stock Market's Reaction to Unemployment News: Why Bad News Is Usually Good for Stocks." *Journal of Finance* 60.2(2005):649-672.
8. Frazzini, Andrea. "The Disposition Effect and Underreaction to News." *Journal of Finance* 61.4(2010):2017-2046.
9. Kim, Kyoung Jae. "Financial time series forecasting using support vector machines." *Neurocomputing* 55.1-2(2003):307-319.
10. Schumaker, Robert P., and Hsinchun Chen. "Textual analysis of stock market prediction using breaking financial news: The AZFin text system." *ACM Transactions on Information Systems (TOIS)* 27.2 (2009): 12.
11. Bollen, Johan , H. Mao , and X. Zeng . "Twitter mood predicts the stock market." *Journal of Computational Science* 2.1(2011):1-8.
12. Liu, Fajiang, and Jun Wang. "Fluctuation prediction of stock market index by Legendre neural network with random time strength function." *Neurocomputing* 83 (2012): 12-21.
13. Zhang, Wenping, et al. "Dynamic business network analysis for correlated stock price movement prediction." *IEEE Intelligent Systems* 30.2 (2015): 26-33.
14. Kia, Arash Negahdari, Saman Haratizadeh, and Saeed Bagheri Shouraki. "A hybrid supervised semi-supervised graph-based model to predict one-day ahead movement of global stock markets and commodity prices." *Expert Systems with Applications* 105 (2018): 159-173.
15. McLean, R. David, and Jeffrey Pontiff. "Does academic research destroy stock return predictability?." *The Journal of Finance* 71.1 (2016): 5-32.
16. Kim, Kyoung-jae, and Ingoo Han. "Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index." *Expert systems with Applications* 19.2 (2000): 125-132.
17. Kuo, Ren Jie, C. H. Chen, and Y. C. Hwang. "An intelligent stock trading decision support system through integration of genetic algorithm based fuzzy neural network and artificial neural network." *Fuzzy sets and systems* 118.1 (2001): 21-45.
18. Hassan, Md Rafiul, Baikunth Nath, and Michael Kirley. "A fusion model of HMM, ANN and GA for stock market forecasting." *Expert systems with Applications* 33.1 (2007): 171-180.
19. Guresen, Erkam, Gulgun Kayakutlu, and Tugrul U. Daim. "Using artificial neural network models in stock market index prediction." *Expert Systems with Applications* 38.8 (2011): 10389-10397.
20. Ticknor, Jonathan L. "A Bayesian regularized artificial neural network for stock market forecasting." *Expert Systems with Applications* 40.14 (2013): 5501-5506.
21. Das, Shom Prasad, and Sudarsan Padhy. "A novel hybrid model using teaching–learning-based optimization and a support vector machine for commodity futures index forecasting." *International Journal of Machine Learning and Cybernetics* 9.1 (2018): 97-111.
22. Patel, Jigar, et al. "Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques." *Expert Systems with Applications* 42.1 (2015): 259-268.
23. Ballings, Michel, et al. "Evaluating multiple classifiers for stock price direction prediction." *Expert Systems with Applications* 42.20 (2015): 7046-7056.
24. Kolanovic, M., and R. T. Krishnamachari. "Big Data and AI Strategies: Machine Learning and Alternative Data Approach to Investing." *JP Morgan Global Quantitative & Derivatives Strategy Report* (2017).
25. Kohli, Pahul Preet Singh, et al. "Stock Prediction Using Machine Learning Algorithms." *Applications of Artificial Intelligence Techniques in Engineering* 1 (2018): 405.
26. Ke, Guolin, et al. "Lightgbm: A highly efficient gradient boosting decision tree." *Advances in Neural Information Processing Systems*. 2017.
27. Fang, Bin, and P. Zhang. *Big Data in Finance*. *Big Data Concepts, Theories, and Applications*. 2016.
28. Myers, S. , and N. Majluf . "Corporate Financing and Investment when Firms Have Information Shareholders Do Not Have." *Social Science Electronic Publishing* 13.2(2001):187-221.
29. Market and News datasets. Intrinio, Thomson Reuters (2018). <https://www.kaggle.com/c/two-sigma-financial-news/data>
30. Jain, A . "Score normalization in multimodal biometric systems." *Pattern Recognition* 38.12(2013):2270-2285.
31. Kohavi, Ron. "A study of cross-validation and bootstrap for accuracy estimation and model selection." *International Joint Conference on Artificial Intelligence* 1995.
32. Jensen, Michael C. "The performance of mutual funds in the period 1945–1964." *The Journal of finance* 23.2 (1968): 389-416.