Using News to Predict Stock Movements

Kaggle Challenge Competition

Ivo Tadic
CS 522 – Data Science
Hood College Graduate School
Frederick Maryland USA
it3@hood.edu

ABSTRACT

Do news released have some effect on stock market movement?

Two Sigma has released a Kaggle competition for Data Scientists to try to find out the answer to that question. Based on news released by Thomson Reuters, containing news statistics including sentiment analysis, and market data provided by Intrinio Kagglers included myself are trying to find a good way to use the data to predict stock market movement.

The data provided includes just over four million records of market data and over nine million records of news data with dates from 2007 to 2017. The Kaggle contest has memory and processing time restrictions. Large amounts of data to process along with memory and processing time restrictions call for a fast and memory efficient algorithm. As we'll find later the LGBM classifier fits great in this category, but we also want to look at the problem from a good way to solve it without the memory and processing restrictions and that's where neural networks come into play, specifically the Long short-term memory network (LSTM). LSTMs are an improved version of RNNs. RNNs are good for predicting trends but lack the notion of context, that is where LSTMs improve RNNs and introduce a mechanism to include trends and context in the prediction.

The results show that even though news data features are not the top features in the solution they play a role in the result. That helps the model be more accurate.

CCS CONCEPTS

• Computing methodologies~Boosting

• Computing methodologies~Neural networks

KEYWORDS

Decision Tree Boosting. LGBM. LSTM.

1 Introduction

This project was inspired by the Kaggle competition "Two Sigma: Using News to Predict Stock Movements" [1]. The competition has some rules and restrictions that will be described and followed for the project. One of the advantages of Kaggle contests is that the Datasets have been defined and documented very well.

The competition provides two datasets one with News data and one with Market data. We want to find the connection between the datasets and find if news data can really help predicting market movement.

2 Literature review

News content is mainly plain text, that information in large volumes is very difficult to use and quantify. I'll first review options to translate that content into a quantity that is easier to understand for analysis and easier to understand by a Computer system. We chose sentiment analysis to quantify the meaning of news content and to help us understanding the link between news and market movement.

Onan and Korukoglu state that one of the problems about text sentiment classification is efficiency of the solution. They propose the use of feature selection based on generic rank aggregation. [4]

Yand Li, Jie Wang, Suge Wang, Jiye Liang, and Juanzi Li mention the problem of data imbalance in text sentiment classification. They state that it is a widely existent phenomenon and that by using the LDMRC algorithm is necessary to rebalance the data for text sentiment classification. [5]

Yijing Li, Haixiang Guo, Qingpeng Zhang, Mingyun Gu, and Jianying Yand explore two issues in sentiment classification: domain-sensitivity and data imbalance. To deal with domain-sensitivity they explore a model that incorporates universal and domain-specific knowledge. For the imbalanced data problem, they propose the use of an over-sampling technique. [6]

Song, Wang, Liu, and Zhao describe the use of the emotion adjustment method based on semantic similarity and skip-gram models. They explain both methods and how their weight adjustment improves the accuracy of text sentiment analysis compared to traditional methods. [7]

Songtao Shang, Yong Gan, and Huaiguang Wu describe the feature weighting problem when using Naïve Bayes for text classification. They propose the use of an algorithm entitled TF-Gini to tackle the weighting problem and improve the performance of Naïve Bayes for text classification. [8]

Hema Krishnan, M. Sudheep Elayidom, and T. Santhanakrishnan review a lexicon based approach for analyzing text data and measure text sentiment. [9]

Xie Tie, Zheng Xiao, Zang Lei, and Wang Xiujun use parallelized recursive neural networks to improve efficiency and accuracy on text sentiment analysis. [10]

3 Data Sets

We are going to work with two datasets: News data and Market data. We'll first detail them independently and then define a way to merge them and after merging them we'll analyze the merged dataset and use that as our dataset of study.

3.1 News dataset

The News dataset is provided by Thomson Reuters [3]. The dataset contains the following fields as defined by Thomson Reuters [3]:

"

time(datetime64[ns, UTC]) - UTC timestamp showing when the data was available on the feed (second precision)

sourceTimestamp(datetime64[ns, UTC]) - UTC timestamp of this news item when it was created

firstCreated(datetime64[ns, UTC]) - UTC timestamp for the first version of the item

sourceId(object) - an Id for each news item

headline(object) - the item's headline

urgency(int8) - differentiates story types (1: alert, 3: article)

takeSequence(int16) - the take sequence number of the news item, starting at 1. For a given story, alerts and articles have separate sequences.

provider(category) - identifier for the organization which provided the news item (e.g. RTRS for Reuters News, BSW for Business Wire)

subjects(category) - topic codes and company identifiers that relate to this news item. Topic codes describe the news item's subject matter. These can cover asset classes, geographies, events, industries/sectors, and other types.

audiences(category) - identifies which desktop news product(s) the news item belongs to. They are typically tailored to specific audiences. (e.g. "M" for Money International News Service and "FB" for French General News Service)

bodySize(int32) - the size of the current version of the story body in characters

companyCount(int8) - the number of companies explicitly listed in the news item in the subjects field

headlineTag(object) - the Thomson Reuters headline tag for the news item

marketCommentary(bool) - boolean indicator that the item is discussing general market conditions, such as "After the Bell" summaries

sentenceCount(int16) - the total number of sentences in the news item. Can be used in conjunction with firstMentionSentence to determine the relative position of the first mention in the item.

wordCount(int32) - the total number of lexical tokens (words and punctuation) in the news item

assetCodes(category) - list of assets mentioned in the item

assetName(category) - name of the asset

firstMentionSentence(int16) - the first sentence, starting with the headline, in which the scored asset is mentioned.

1: headline

2: first sentence of the story body

3: second sentence of the body, etc

0: the asset being scored was not found in the news item's headline or body text. As a result, the entire news item's text (headline + body) will be used to determine the sentiment score.

relevance(float32) - 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. When the item is an alert (urgency == 1), relevance should be gauged by firstMentionSentence instead.

sentimentClass(int8) - indicates the predominant sentiment class for this news item with respect to the asset. The indicated class is the one with the highest probability.

sentimentNegative(float32) - probability that the sentiment of the news item was negative for the asset

sentimentNeutral(float32) - probability that the sentiment of the news item was neutral for the asset

sentimentPositive(float32) - probability that the sentiment of the news item was positive for the asset

sentimentWordCount(int32) - the number of lexical tokens in the sections of the item text that are deemed relevant to the asset. This can be used in conjunction with wordCount to determine the proportion of the news item discussing the asset.

noveltyCount12H(int16) - The 12 hour 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.

noveltyCount24H(int16) - same as above, but for 24 hours

noveltyCount3D(int16) - same as above, but for 3 days

noveltyCount5D(int16) - same as above, but for 5 days

noveltyCount7D(int16) - same as above, but for 7 days

volumeCounts12H(int16) - the 12 hour volume of news for each asset. A cache of previous news items is maintained and the number of news items that mention the asset within each of five historical periods is calculated.

volumeCounts24H(int16) - same as above, but for 24 hours

volumeCounts3D(int16) - same as above, but for 3 days

volumeCounts5D(int16) - same as above, but for 5 days

volumeCounts7D(int16) - same as above, but for 7 days

" [3]

3.2 Market dataset

The Market dataset is provided by Intrinio [2]. The following fields are provided and detailed in the contest as follows:

"

time(datetime64[ns, UTC]) - the current time (in marketdata, all rows are taken at 22:00 UTC)

assetCode(object) - a unique id of an asset

assetName(category) - the name that corresponds to a group of assetCodes. These may be "Unknown" if the corresponding assetCode does not have any rows in the news data.

universe(float64) - a boolean indicating whether or not the instrument on that day will be included in scoring. This value is not provided outside of the training data time period. The trading universe on a given date is the set of instruments that are avilable for trading (the scoring function will not consider instruments that are not in the trading universe). The trading universe changes daily.

volume(float64) - trading volume in shares for the day

close(float64) - the close price for the day (not adjusted for splits or dividends)

open(float64) - the open price for the day (not adjusted for splits or dividends)

returnsClosePrevRaw1(float64) - see returns explanation above

returnsOpenPrevRaw1(float64) - see returns explanation above

returnsClosePrevMktres1(float64) - see returns explanation above

returnsOpenPrevMktres1(float64) - see returns explanation above

returnsClosePrevRaw10(float64) - see returns explanation above

returnsOpenPrevRaw10(float64) - see returns explanation above

returnsClosePrevMktres10(float64) - see returns explanation above

returnsOpenPrevMktres10(float64) - see returns explanation above

returnsOpenNextMktres10(float64) - 10 day, market-residualized return. This is the target variable used in competition scoring. The market data has been filtered such that returnsOpenNextMktres10 is always not null.

" [2]

4 Data analysis

From the metadata we find that we can used the time and assetCode fields to combine the data before we can analyze it both datasets contain that information and that's how we merged the datasets into a single dataset.

4.1 Data available

Figure 1 shows the amount of information provided in the market dataset.

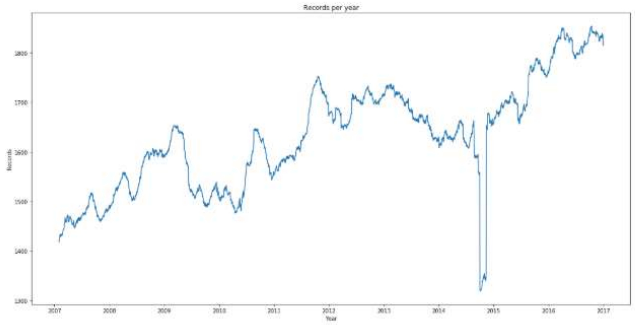


Figure 1: Number of records in the market dataset per year

The initial total number of records, before any cleanup or filtering is:

News dataset: 9328750 total records.

Market dataset: 4072956 total records.

4.2 Data cleanup

The datasets were pre-processed independently and merged after pre-processing. The reason to do filtering before was to use the least memory possible during the merging of the data that is the operation that would take the most amount of memory.

For the News dataset I first filtered out all records older than 2013, the specific date was defined by trying to include as much as possible without going out of the processing time and memory usage limitations. After filtering out the news by date the marketCommentary column is separated to two different columns one to denote there was a commentary and one where there was not. Then the features with very small correlation coefficient calculated during data analysis were dropped, those features are the following: urgency, bodySize, sentimentClass, sentimentWordCount, sentenceCount, wordcount, volumeCounts7D, volumeCounts5D, volumeCounts3D, volumeCounts24H, volumeCounts12H, noveltyCount7D, noveltyCount5D, noveltyCount3D, noveltyCount24H, noveltyCount12H, and fisrstMentionSentence. After dropping out the fields, the dataset was expanded from containing one news for multiple assetCodes to contain a single line of news per assetCode in preparation for merging the data with the Market dataset, that means that we are expanding the number of records. After expanding the news dataset the following features are dropped from the dataset: newsIndex, sourceTimestamp, firstCreated, subjects, audiences, headline, assetCodes, assetName, sourceId.

For the Market dataset we also filtered out records older than 2013 then the following columns are dropped because of their low correlation with the target feature: universe, returnsClosePRevMktres1, returnsOpenPrevMktres1, returnsClosePrevMktres10, returnsOopenPrevMktres10, time, and asssetName.

After filtering and cleaning up the news and market datasets the record count is updated to:

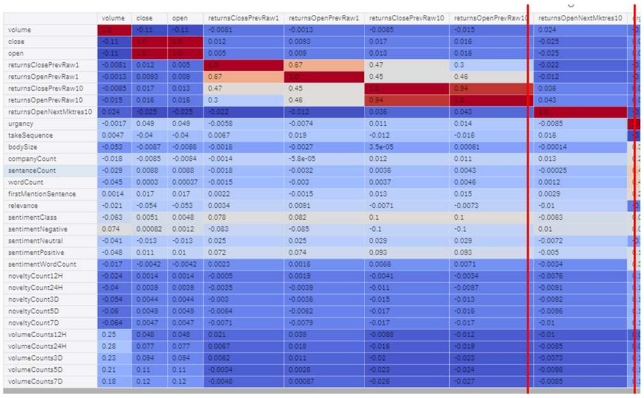
News dataset: 7584679 records.

Market dataset: 2678461 records.

Now that both News and Market datasets are ready, we merge them by assetCode and date and we end up with a combined dataset with the following fields: assetCode, volume, close, open, returnsClosePrevRaw1, returnsOpenPrevRaw1, returnsClosePrevRaw10, returnsOpenPrevRaw10, returnsOpenNextMktres10, date, takeSequence, provider, companyCount, relevance, sentimentNegative, sentimentNeutral, sentimentPositive, comentaryyes, commentaryno.

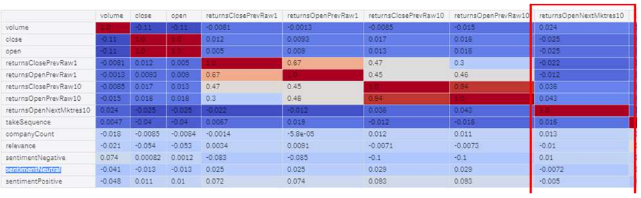
Combined dataset: 2047519 records.

4.3 Correlation



	volume	close	open	returnsClosePrevRaw1	returnsOpenPrevRaw1	returnsClosePrevRaw10	returnsOpenPrevRaw10	returnsOpenNextMktres10
volume	1.0	-0.11	-0.11	-0.0081	-0.0013	-0.0085	-0.015	0.024
close	-0.11	1.0	1.0	0.012	0.0093	0.017	0.016	-0.025
open	-0.11	1.0	1.0	0.005	0.009	0.013	0.016	-0.025
returnsClosePrevRaw1	-0.0081	0.012	0.005	1.0	0.67	0.47	0.3	-0.022
returnsOpenPrevRaw1	-0.0013	0.0093	0.009	0.67	1.0	0.45	0.46	-0.012
returnsClosePrevRaw10	-0.0085	0.017	0.013	0.47	0.45	1.0	0.94	0.036
returnsOpenPrevRaw10	-0.015	0.016	0.016	0.3	0.46	0.94	1.0	0.043
returnsOpenNextMktres10	0.024	-0.025	-0.025	-0.022	-0.012	0.036	0.043	1.0
urgency	-0.0017	0.049	0.049	-0.0058	-0.0074	0.011	0.014	-0.0085
takeSequence	0.0047	-0.04	-0.04	0.0067	0.019	-0.012	-0.016	0.016
bodySize	-0.053	-0.0087	-0.0086	-0.0016	-0.0027	3.5e-05	0.00061	-0.00014
companyCount	-0.018	-0.0085	-0.0084	-0.0014	-5.8e-05	0.012	0.011	0.013
sentenceCount	-0.029	0.0088	0.0088	-0.0018	-0.0032	0.0036	0.0043	-0.00025
wordCount	-0.045	0.0003	0.00037	-0.0015	-0.003	0.0037	0.0046	0.0012
firstMentionSentence	0.0014	0.017	0.017	0.0022	-0.0015	0.013	0.015	0.0029
relevance	-0.021	-0.054	-0.053	0.0034	0.0091	-0.0071	-0.0073	-0.01
sentimentClass	-0.063	0.0051	0.0048	0.078	0.082	0.1	0.1	-0.0063
sentimentNegative	0.074	0.00082	0.0012	-0.083	-0.085	-0.1	-0.1	0.01
sentimentNeutral	-0.041	-0.013	-0.013	0.025	0.025	0.029	0.029	-0.0072
sentimentPositive	-0.048	0.011	0.01	0.072	0.074	0.093	0.093	-0.005
sentimentWordCount	-0.017	-0.0042	-0.0042	0.0023	0.0016	0.0066	0.0071	-0.0034
noveltyCount12H	-0.024	0.0014	0.0014	-0.0005	0.0019	-0.0041	-0.0034	-0.0076
noveltyCount24H	-0.04	0.0039	0.0039	-0.0035	-0.0039	-0.011	-0.0097	-0.0081
noveltyCount3D	-0.054	0.0044	0.0044	-0.003	-0.0036	-0.015	-0.013	-0.0082
noveltyCount5D	-0.06	0.0049	0.0049	-0.0064	-0.0062	-0.017	-0.016	-0.0096
noveltyCount7D	-0.064	0.0047	0.0047	-0.0071	-0.0079	-0.017	-0.017	-0.01
volumeCounts12H	0.25	0.048	0.048	0.021	0.039	-0.0088	-0.012	-0.01
volumeCounts24H	0.28	0.077	0.077	0.0067	0.018	-0.016	-0.019	-0.0085
volumeCounts3D	0.23	0.094	0.094	0.0062	0.011	-0.02	-0.023	-0.0073
volumeCounts5D	0.21	0.11	0.11	-0.0034	0.0028	-0.023	-0.024	-0.0098
volumeCounts7D	0.18	0.12	0.12	-0.0048	0.00087	-0.026	-0.027	-0.0085

Figure 2: Correlation heatmap (before feature selection)



	volume	close	open	returnsClosePrevRaw1	returnsOpenPrevRaw1	returnsClosePrevRaw10	returnsOpenPrevRaw10	returnsOpenNextMktres10
volume	1.0	-0.11	-0.11	-0.0081	-0.0013	-0.0085	-0.015	0.024
close	-0.11	1.0	1.0	0.012	0.0093	0.017	0.016	-0.025
open	-0.11	1.0	1.0	0.005	0.009	0.013	0.016	-0.025
returnsClosePrevRaw1	-0.0081	0.012	0.005	1.0	0.67	0.47	0.3	-0.022
returnsOpenPrevRaw1	-0.0013	0.0093	0.009	0.67	1.0	0.45	0.46	-0.012
returnsClosePrevRaw10	-0.0085	0.017	0.013	0.47	0.45	1.0	0.94	0.036
returnsOpenPrevRaw10	-0.015	0.016	0.016	0.3	0.46	0.94	1.0	0.043
returnsOpenNextMktres10	0.024	-0.025	-0.025	-0.022	-0.012	0.036	0.043	1.0
takeSequence	0.0047	-0.04	-0.04	0.0067	0.019	-0.012	-0.016	0.016
companyCount	-0.018	-0.0085	-0.0084	-0.0014	-5.8e-05	0.012	0.011	0.013
relevance	-0.021	-0.054	-0.053	0.0034	0.0091	-0.0071	-0.0073	-0.01
sentimentNegative	0.074	0.00082	0.0012	-0.083	-0.085	-0.1	-0.1	0.01
sentimentNeutral	-0.041	-0.013	-0.013	0.025	0.025	0.029	0.029	-0.0072
sentimentPositive	-0.048	0.011	0.01	0.072	0.074	0.083	0.083	-0.005

Figure 3: Correlation heatmap (after feature selection)

Both figure 2 and figure 3 show the correlation between the features and the target column (marked with a red box in both figures).

In the heatmap the deeper the blue the better correlation between the two features compared. We can see that our dataset does not have any strong correlation that can help us with the prediction.

4.4 Feature selection

Based on the correlation we have picked the following features for training: assetCode, volume, close, open, returnsClosePrevRaw1, returnsOpenPrevRaw1, provider, returnsClosePrevRaw10, returnsOpenPrevRaw10, returnsOpenNextMktres10, date, takeSequence, companyCount, relevance, sentimentNegative, sentimentNeutral, sentimentPositive, comentaryyes, commentaryno.

We look at the sentiment mean in figure 4.

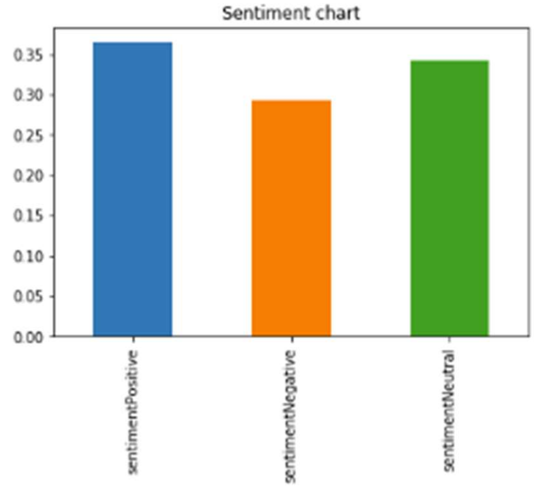


Figure 4: Sentiment mean

To understand how much of the news data was directly related and flagged to be part of the market summary the market commentary is graphed in figure 5.

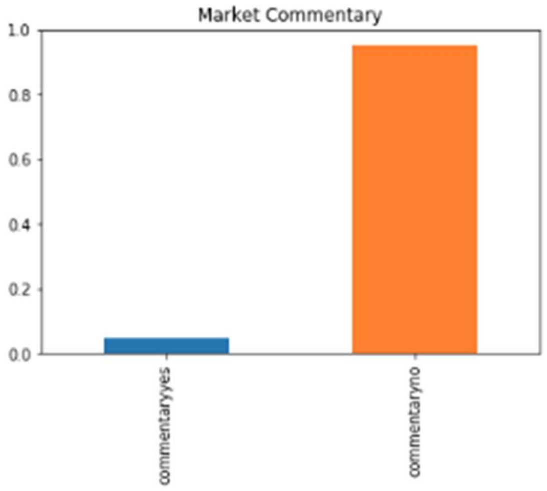


Figure 5: Market commentary

5 Algorithms

Two algorithms were reviewed for the project:

- Long short-term memory
- LGBM Classifier

5.1 Long short-term memory (LSTM)

Long short-term memory networks are an improved type of Recurrent Neural Network. Recurrent Neural Networks are very good for predicting sequences but their inclusion of context based information is very limited. Recurrent Neural Networks only account for the context behind the sequence itself.

Lon short-term memory networks are an improvement of Recurrent Neural Networks because the gating system used in their implementation allows the network to remember and/or forget things selectively allowing the network to account for the trend, previous data, and current data.

Unfortunately, the benefit of LSTMs to store information to account for context, previous and current data means the use of larger amounts of memory and the project is restricted on memory. In the timeframe of the project I tried implementing LSTMs but the processing time and memory exceeded the limitations so the results were inconclusive.

5.2 LGBM Classifier

Since LSTMs are not a feasible solution in this particular case then the search for a small memory usage and fast execution algorithm to start with was the next step. By looking what other classification algorithms I found that the LGBM Classifier would be able to stay within the execution time and memory constraints. LGBM classifiers are a decision tree boosting algorithm, the algorithm is optimized for speed and memory utilization and was a good fit for the project.

6 Competition

6.1 Restrictions

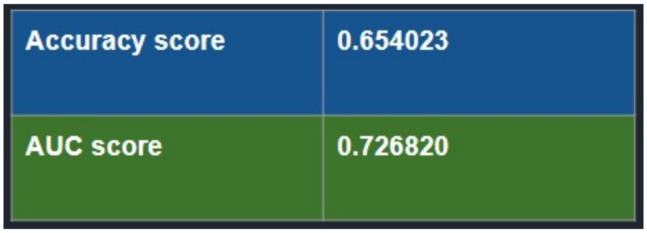
Kaggle uses a Kernel environment, the competition requires that the submissions are in the form of a Kernel. [1]
There is a computational constraint for the submission of Kaggle Kernels [1]:

- 16 GB of memory.
- 6 hours of runtime.
- 2 CPU cores.

7 Results

In the following sections the results of predicting the target feature against the training set was analyzed. For this analysis a 80% train/ 20% test was used.

7.1 Accuracy and AUC scores

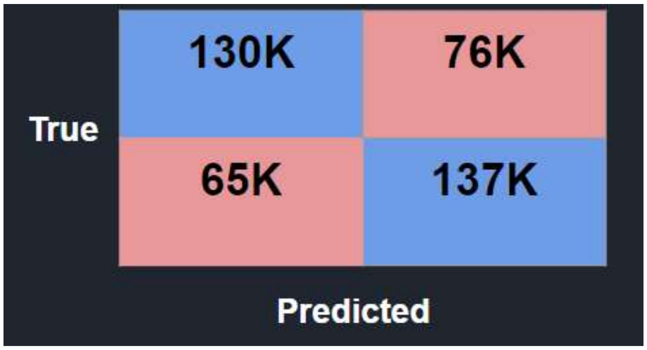


Accuracy score	0.654023
AUC score	0.726820

Figure 6: Accuracy and AUC scores

Figure 6 shows that the accuracy is not excellent, but it is fairly good considering the low correlation found in the heatmap.

7.2 Confusion Matrix



True	130K	76K
	65K	137K
	Predicted	

Figure 7: Confusion Matrix

Figure 7 shows a good amount of the predictions falled in the right place but it also shows that a large number of false positives and true negatives were predicted. That's the number 1 thing that needs to be looked at further in the future.

7.3 Predicted confidence

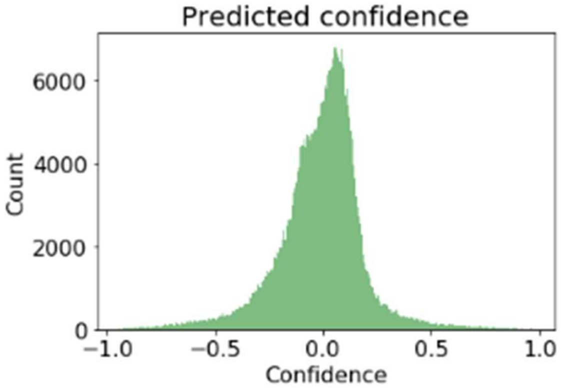


Figure 8: Predicted confidence

7.4 Feature importance

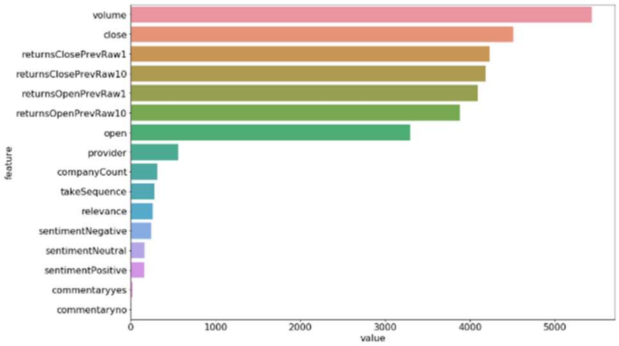


Figure 9: Feature importance

Figure 9 shows the list of features in order on how much it provides to the final model's prediction. Volume is the most important feature followed by close, returns open and close for the past day and ten days and we can also see that sentiment provides just a little value to the model.

ACKNOWLEDGMENTS

I analyzed many Kaggle kernels that people opened for the contest. By reading many kernels it helped me on how to approach the analysis of the data, what type of graph to use. The Kaggle platform itself is very useful and I have to acknowledge that it saved me quite a bit of time of setting up a proper environment to analyze the data and to run my tests.

8 Future work

Unfortunately, the predictions of the final model are prone to a good percentage of false positives and true negatives which the contest penalizes. To improve on that If I had more time, I would analyze the data that contains false positives and true negatives only to understand what's causing it and to minimize those cases. I would try using a neural network algorithm to train a model but train on a very small sample of recent data and compare that with my current final model to see if I would get an improvement in the accuracy.

REFERENCES

[1] Kaggle (2018), Two Sigma: Using News to Predict Stock Movements, https://www.kaggle.com/c/two-sigma-financial-news.

[2] Intrino (2018), Intrino's Main Site, https://intrinio.com/.

[3] Thomson Reuters (2018), Main Site , https://www.thomsonreuters.com/en/products-services/financial.html/.

[4] Aytug Onan and Serdar Korokoglu. 2017. *Journal of Information Science 2017. Vol. 43(I) 25-38*. DOI: https://doi.dox.org/10.1177/0165551515613226.

[5] Li, Y., Wang, J., Wang, S. et al. Int. J. Mach. Learn. & Cyber. (2018). https://doi.org/10.1007/s13042-018-0858-x.

[6] Yijing Li, Haixiang Guo, Qingpeng Zhang, Mingyun Gu, Jianying Yang, Imbalanced text sentiment classification using universal and domain-specific knowledge, Knowledge-Based Systems, Volume 160, 2018, Pages 1-15, ISSN 0950-7051,https://doi.org/10.1016/j.knosys.2018.06.019.

[7] Song M., Wang Y., Liu Y., Zhao Z. (2018) Text Sentiment Analysis Based on Emotion Adjustment. In: Zhou Q., Miao Q., Wang H., Xie W., Wang Y., Lu Z. (eds) Data Science. ICPCSEE 2018. Communications in Computer and Information Science, vol 902. Springer, Singapore.

[8] Songtao Shang, Yong Gan, and Huaiguang Wu. An Improved Text Sentiment Analysis Algorithm based on TF-Gini. International Journal of Performability Engineering Volume 14, Number 9, September 2018, pp. 2008-2014. DOI: 10.23940/ijpe.18.09.p8.20082014

[9] Hema Krishnan, M. Sudheep Elayidom, T. Santhanakrishnan Sentiment Analysis of Tweets for Inferring Popularity of Mobile Phones. International Journal of Computer Applications (0975-887) Volume 157 – No 2, January 2017. https://pdfs.semanticscholar.org/f700/9270213ece551e14e52760598a0c75429c29.pdf

[10] Xie Tie;Zheng Xiao;Zhang Lei;Wang Xiujun. Sentiment Classification of Chinese Short Text Based on Parallelized Recursive Neural Network. 2017-03. NKI.com.cn. http://en.cnki.com.cn/Article_en/CJFDTotal-JYRJ201703037.htm.