Stocks And Tweets: What Is Naive And What Is Not

Andrey Teterin **Dmitry Malishev**

Viktor Iudin **Kirill Gerasimov**

Abstract

This work is devoted to an old popular subject - the study of the relationship between publications in social networks and the corresponding stock prices. After analyzing the sources, the article [Jaggi M. at al., 2021] with the published code and dataset of 6.4 million tweets for 25 stocks was chosen as the basis.

At the first stage of the project, we managed to improve the results from this article, using a simple Naive Bayes model on TF-IDF features. Improvements were achieved mainly due to the joint use of word and character level n-grams.

Several significant shortcomings noted in the approaches of the article led us to organizing the "second" project stage - dataset regeneration and retraining of several models, followed by their ensembling. As a result, we got an accuracy of 0.582 with a trivial random model baseline of 0.500, which seems to be a very good result for this type of task.

Introduction

Understanding stock market movements is a problem of great interest among finance researchers. Modern machine learning approaches together with publicly available social media datasets give the opportunity to learn the primary cause of these movements – people. Previous studies in this domain give diverse and contradictory results. Also, the very idea of stock market algorithmic prediction conflicts with some financial theories.

The thesis of this work is to apply different machine learning methods to investigate the relationship of social media and the stock market. To decrease ambiguity all intermediate entities like sentiments are skipped to establish direct connection of text sentences and subsequent stock numeric values changes. Twitter messages are chosen as a dataset since it is a source of instant human reactions given in a short unprocessed form.

After a brief review of the sources, we chose the work [Jaggi M. at al., 2021] **(hereinafter referred to as "the article"**) as the reference, the main advantage of it is the publicly posted dataset and source code. However, a detailed study revealed shortcomings both in terms of machine learning approaches and in terms of the subject area. This is discussed in more detail in the section "Criticizing the related article".

As a result, the work on the project was organized in the following stages:

- searching for existing articles on the topic and selecting the most suitable of them for the role of the baseline
- obtaining and analyzing a dataset from the selected article ("Dataset V1")
- reproduction of the results from the article, and an attempt to improve them
- the realization that there are fundamental flaws in the dataset and approaches
- formation of a corrected dataset ("Dataset V2")
- writing code and checking several types of ML models on this new dataset
- writing this report

Below we have tried to cover all these stages.

Team

4 people took part in the work on the project. The roles were distributed as follows:

- Kirill Gerasimov - search for existing articles on the topic, analysis of potential problems in approaches
- Viktor Iudin - reproducing and improving the results from the article
- Dmitry Malishev - writing and running CatBoost, Transformers models for Dataset V2, writing the report
- Andrey Teterin - writing and launching Naive Bayes, RNN models for Dataset V2, general coordination of work, writing the report

Related Work

We have reviewed many articles on the topic of the connection of texts with the value of assets. First of all, we took practical publications on smart-lab.ru. Despite the relative superficiality of the approaches, this type of publications usually contains practical results, plus valuable comments from professionals in the subject area.

Many publications were related to analysis of financial statements, provided by companies themselves (10-k, etc., [Adusumilli, R. et al., 2020]). However, despite the good reviews of this approach, the disadvantage is that these reports are issued only 1 time per quarter or 1 time per year. Therefore, it was decided to take another popular type of text data - messages from Twitter, which mention the names of companies from the top list of NYSE / NASDAQ.

There is quite a lot of work on this topic (an extensive recent literature review is given in [Yilmaz et al. 2022]). In most publications, a "two-stage" scheme is used, when source texts are first processed using classical sentiment analysis models, and then this output (positive/negative) is used for detecting influence on the asset price.

This "two-stage" approach has many disadvantages:

- Usually sentiment evaluation is performed entirely for the text, and not for a specific entity (asset), which could become a severe problem for large texts.
- Manual labeling by people is required, and these labelers should be highly skilled in understanding special terms and slang in the posts
- Sometimes it is difficult even for a trained person to tell whether the text is positive or negative in the context of the future value of assets. For example, there are historical cases when the message about

the upcoming change of CEO did not eventually lead to a fall in the stock price, as many expected, but to an increase.

Due to the shortcomings outlined above, it was decided to consider only those works where "direct" approaches are used, when the sentiment analysis stage is skipped, and the models directly work with input texts and output financial results. As a result, we chose the work [Jaggi M. at al., 2021] to be the base article for the project ("the article").

Criticizing the article

As already mentioned in the Introduction section, a detailed analysis of the article and reproduction of the results revealed a large number of controversial points. Below are the most significant ones.

Issues in the subject area:

- For some reason, for target variable the authors took stock price change for the date of the corresponding tweet publication, not the next day, etc. (for more details, see the Assembling Dataset V2 section)
- When preparing the data, equivalent tickers were not merged together (for example, GOOG, GOOGL)
- The gaps in financial data are filled in a strange way (for details, see Assembling Dataset V2)

Issues in ML implementation:

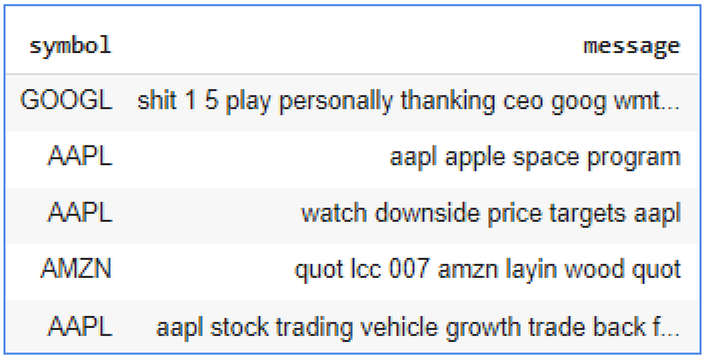
- When processing texts of tweets, all stop words were removed, even those that are vital for message sense (for more details, see the Datasets section)
- in the data preparation and model training code, the random seed was not fixed, which led to different results on each run
- Datasets were not fully balanced, which for some types of metrics can cause problems with training and evaluation of models, especially given the complexity of the tasks of this type
- The authors did not check trivial models (all-zeroes, all-ones, random) for getting a rough baseline
- For models on n-grams, only word level n-grams were checked (not character-level), and only of a fixed size (2)

Nevertheless, despite the shortcomings, the article authors received quite interesting results for some types of labeling schemes. Therefore, it was decided to divide our work into two large parts - firstly, checking and improving the results on the original dataset (V1) from the article, and, secondly, generating a new corrected dataset (V2) and testing different approaches on it - both from the article and our own.

Part 1: Analysis and Enhancing Results from the Article

Understanding Dataset V1

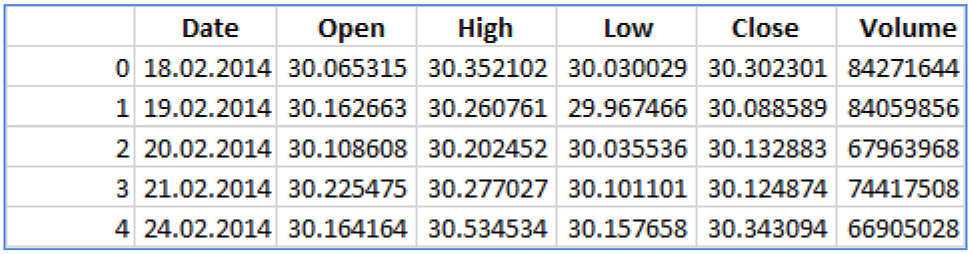
As already mentioned above, at the first stage of the project, the original dataset from the article was taken for work. The data was downloaded from the link from this github page:
https://github.com/mjag7682/NLP-of-StockTwits-data-for-predicting-stocks
The core of the dataset are Twitter messages (from StockTwits) that mention the most popular stocks (total 6.4 million messages for 25 stock tickers, with 10 years depth).



symbol	message
GOOGL	shit 1 5 play personally thanking ceo goog wmt...
AAPL	aapl apple space program
AAPL	watch downside price targets aapl
AMZN	quot lcc 007 amzn layin wood quot
AAPL	aapl stock trading vehicle growth trade back f...

Example of texts in the Dataset V1

For label generation the article authors took financial data from yfinance.



	Date	Open	High	Low	Close	Volume
0	18.02.2014	30.065315	30.352102	30.030029	30.302301	84271644
1	19.02.2014	30.162663	30.260761	29.967466	30.088589	84059856
2	20.02.2014	30.108608	30.202452	30.035536	30.132883	67963968
3	21.02.2014	30.225475	30.277027	30.101101	30.124874	74417508
4	24.02.2014	30.164164	30.534534	30.157658	30.343094	66905028

Example of yfinance data for GOOGL

Here are the formulas for converting financial data to labels:

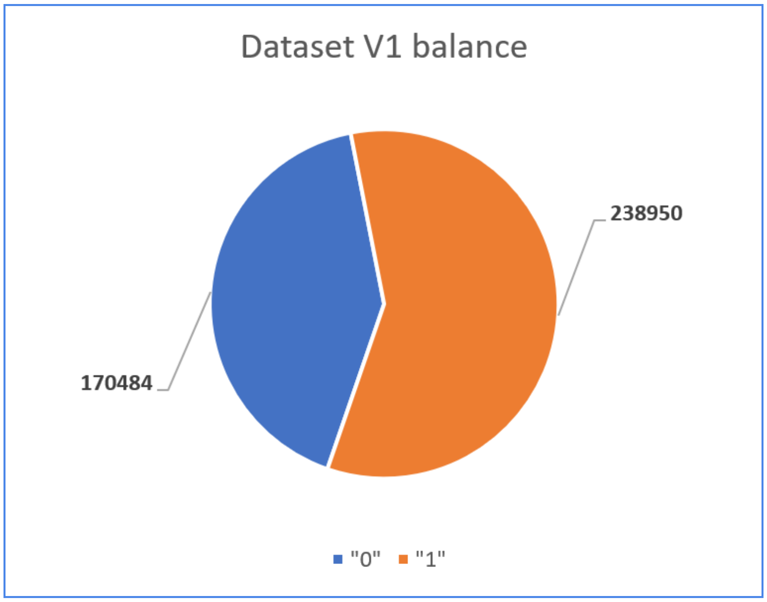
$$percentage\ change = \frac{Close_{\ same\ day} - Open_{\ same\ day}}{Open_{\ same\ day}}$$

$$label = \begin{cases} 1, if\ percentage\ change > 0.5 \\ 0, if\ percentage\ change < -0.5 \end{cases}$$

The "same day" issue was already mentioned above, and will be discussed in more detail in Part 2.

Another tricky thing is that samples, for which percentage change is in the range [-0.5 .. 0.5], are just discarded, so that there are only 2 classes instead of 3. At first, this approach seemed doubtful to us, but then it turned out that it has an interesting advantage - that in the case of a small error in model prediction (for example, 0.49 instead of 0.51 or vice versa), the model is not penalized, as it would be in the case of a three-class classification.

The balance of dataset classes turned out to be as follows: 58% to 42%. The division into train and test set in the article was made in the ratio of 0.9 to 0.1.



Class balance for the dataset from the article.

Models

From numerous models (> 10) described in the article, we focused on the Naive Bayes classifier over TF-IDF features. The reason for this choice was that this model is quite simple in terms of calculations, and at the same time, the authors of the article obtained very good metrics on it.

First, to reproduce the results, we took exactly the same model parameters as in the article (TF-IDF on 2-word n-grams). The accuracy metric we received coincided with the figure given in the article with an accuracy of 1%, that is quite good.

Further, to improve the results, the n-gram parameters of this model were changed, plus another Naive Bayes model was trained on *symbolic* n-grams. After that, an ensemble of these two models was created to obtain the final accuracy value. The detailed parameters are described in the Results section of Part 1.

Experiments

Metrics

To assess the quality of the models, the Accuracy metric was used - it is simple, intuitive, symmetrical with respect to classes, works well on decently balanced datasets, plus the authors of the article used it in all experiments (along with other metrics).

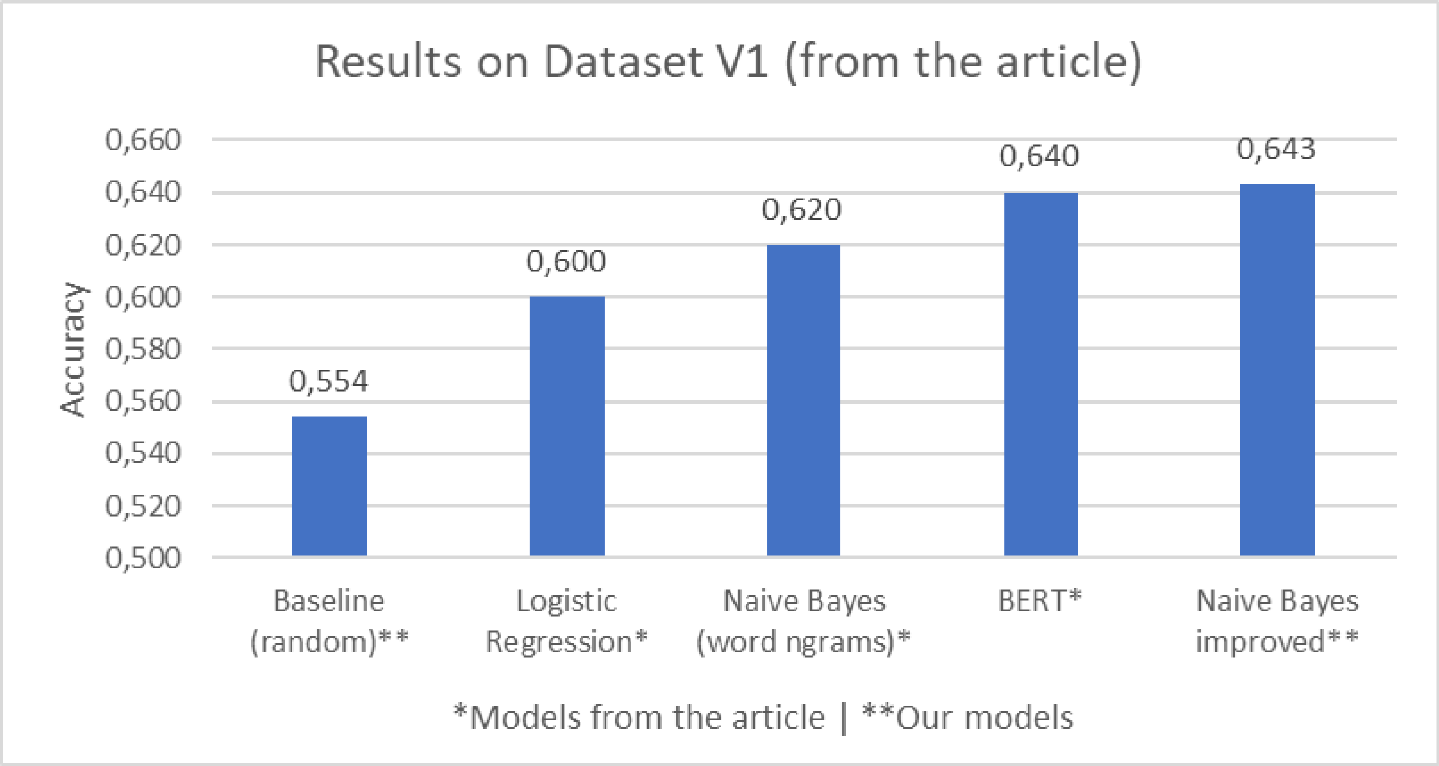
Experiment Setup

In the first part of our work, we preserved the setup from the article: after reading the dataset, it is divided into train and test sets in a ratio of 0.9 to 0.1. Next, the standard procedure for training and testing the model is carried out.

Baseline

As already mentioned, there were no measurements on trivial models in the article. Therefore, as an additional baseline, we launched a simple DummyClassifier model with strategy 'uniform'.

Results



Results achieved on the article dataset
(- accuracy figures taken from the article, ** - our own launches)*

Below are details for our own model launches (with two asterisks):

Baseline (random)** - trivial DummyClassifier with strategy 'uniform'. Accuracy ~55% is slightly below the theoretical 58% (based on the class balance).

Naive Bayes improved** - an ensemble of two Naive Bayes models with TF-IDF features:
1) model on *word* n-grams with length from 1 to 5
2) model on *character* n-grams of length 9
The weight of the first model in the ensemble is 0.6, the second is 0.4.

Thus, even with relatively simple modifications to the Naive Bayes model (adding word-level n-grams of different lengths and adding character-level n-grams), we managed to get better results than the complex BERT model from the article.

Part 2: Introducing New Dataset and Models

Assembling Dataset V2

In the Related Work section, it was mentioned that several issues were found in the process of working with the dataset from the article. Here is a more detailed description of the most significant of them:

1. The authors of the article removed stop words from the message texts - these are about 200 tokens, including such vital words for the meaning as: *not, don't*, etc.

For example, this original message (fragment):
"..in front of you not the one you think it should be"
turned into
"...front one think"

2. When filling gaps in the financial data, the article used price averaging between the previous and next (to gap) date, which could potentially lead to "seeing into the future" issues, when, for example, the price on Saturday and Sunday is interpolated from the price of Friday and Monday. The standard approach in such a situation is filling in with the closing price of the previous day (Friday in our example).

3. When calculating the target variable, the article authors used a doubtful approach, when the change in the share price was taken right on the day the message was published, which doesn't allow establishing a causal relationship and reduces the possible practical usefulness of the idea.

As a result, a script was written to generate a new dataset (V2), in which all the above problems were corrected: removal of stop words was switched off, the standard algorithm for filling in missing data was used, data labels were calculated in a different way (with isolation in time of the message release date and the price change date - in our work, we took the next day after the message):

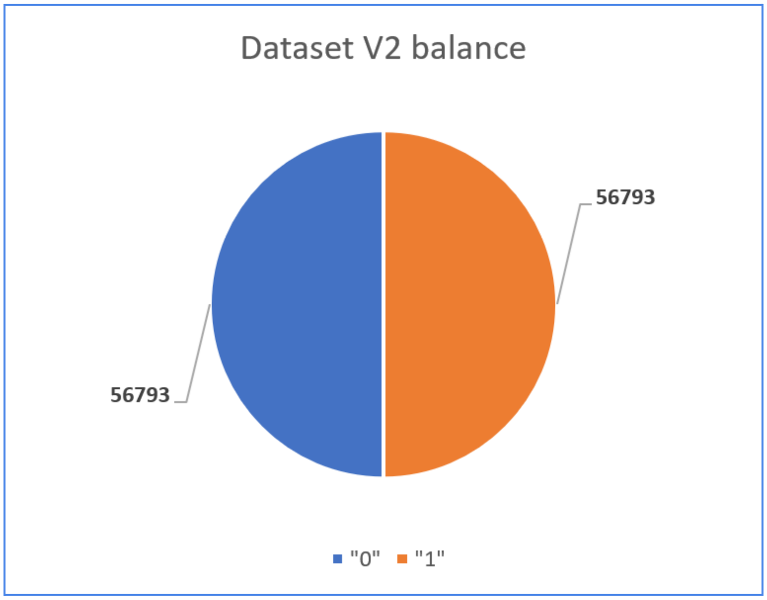
$$percentage\ change = \frac{Close_{next\ day} - Open_{next\ day}}{Open_{next\ day}}$$

The label generation formula is left the same (though the 0.5% threshold could be a subject for further experiments):

$$label = \begin{cases} 1, if\ percentage\ change > 0.5 \\ 0, if\ percentage\ change < -0.5 \end{cases}$$

In addition to the above corrections, it was decided to make the dataset absolutely class-balanced by discarding a certain number of samples with label 1. This made it possible to simplify model evaluation (accuracy 0.5 corresponds to a random choice of labels, while accuracy 0.55 can be considered as a rather good result for this type of prediction task). The total number of rows in the new dataset is approximately 3 times less than in the old one, because to speed up the launch of complex models, only 2 out of 5 stocks were taken.

The division into train and test set was made in the ratio of 0.85 to 0.15. Interestingly, an attempt to use stratification at splitting code led to the fact that for different random seed values, the resulting train and test sets turned out to consist exactly of the same set of rows, but in a rearranged order. Therefore, it was decided to disable stratification (it was not used in the related article either).



Class balance of the re-assembled dataset.

Models

To archive project’s target a number of models were evaluated based on the following criteria:

1. Utilize the most diverse machine learning techniques which are yet able to handle text data.

2. Models should come with the frameworks which let launching models with small effort.

3. Models should be stable and reproduce results with slight changes in an environment and dataset (random state for splitting, etc.)

Three models were finally selected for in-depth study and test:

1. **Transformer** model with its advantage of the most modern and powerful technique to process text data.

2. **GBDT** (gradient boosted decision tree), as the most untypical method of handling texts. Since the core algorithm is not able to have a text as the input, CatBoost framework was chosen for its in-built text preprocessing routine.

3. **Naïve Bayes,** as one of the simplest algorithms; it is used in the baseline paper and demonstrated surprisingly good results

Each algorithm was evaluated independently, including optimizing parameters for best accuracy. Then the results were combined using a blending approach that proved its efficiency in situations when combining diverse models results in accuracy improvement.

Experiments

Metrics

Similar to Part 1, we used the Accuracy metric. As the new dataset is perfectly balanced, the metric is even more intuitive, as the value 0.5 corresponds to random noise.

Experiment Setup

Of all the dataset fields, only one is used - the preprocessed message text, the rest of the features (date, time, author, etc.) are intentionally not used in the pipeline of this work to avoid distortion in the search for the correlation of the entities under study.

After loading the dataset, it is divided into train and test sets in a ratio of 0.85 to 0.15. To check the stability and variability of each model, the split is carried out several times with a different seed value. Each partition is tested independently of the others.

Next, the standard procedure for training and testing (inference) of the model is carried out. The results (and model files) are saved.

After training and saving all the models, an ensemble of models is created to evaluate the cross-correlation of the models and obtain the final maximum achievable accuracy value.

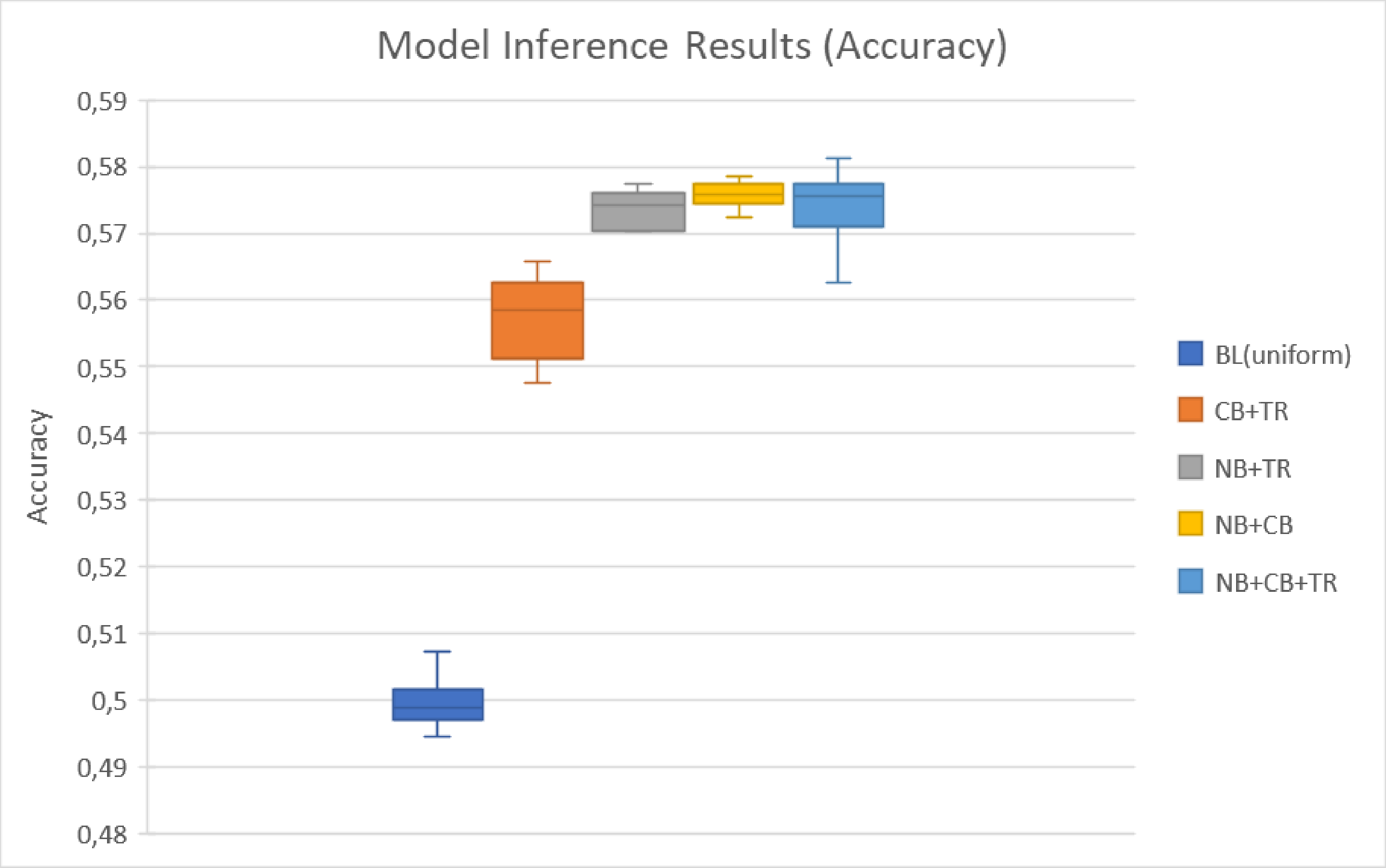
Baseline

Like in Part 1, we used random prediction as a baseline (DummyClassifier with strategy 'uniform'). The result of this trivial model is used to check the dataset and then compare it with other models.

Results

Model Type	Accuracy	Model File Size	Comments
Baseline (Random Guess)	0.500 +- 0.003	-	It’s not an actual model (DummyClassifier with strategy="uniform")
Transformer (Roberta)	0.545 +- 0.002	350 MB	Roberta and DistilRoberta had the same accuracy in our experiments
GBDT (Catboost)	0.560 +- 0.003	5 MB	Model details: CatBoostClassifier n_estimators=300, max_depth = 8
Naive Bayes	0.568 +- 0.002	58 MB + 49 MB	Model details: MultinomialNB on TF-IDF features for word n-grams with length from 1 to 3, alpha = 0.1

Table with individual accuracy of the models



Boxplot graphs for pairwise and triple ensembles.
(BL - baseline model (uniform), CB - CatBoost model
NB - Naive Bayes model, TR - Transformer model)

Let's try to interpret the results.
The accuracy of 0.500 +- 0.003 for the baseline model (random guess) is fully consistent with the theoretical estimate for a fully class-balanced data set.
The Transformer model gives unexpectedly low accuracy (as in the related article). Even the pre-trained version of Roberta, which is the most effective in third-party tests, does not help.
The GBDT (CatBoost) model performs unexpectedly well given the small model size and high learning and inference rates.
The Naive Bayes model shows the highest accuracy, thus giving the most unexpected and surprising result of the entire project. In a related article, this model also showed results at the transformers level.
All models have high stability and reproducibility of results.

The ensemble of models, as expected, makes it possible to slightly increase the efficiency of predictions, especially for models that are initially close in accuracy.

At the moment, the only logical justification for such a distribution of model efficiency is the hypothesis that the main informative elements in messages are keywords and short phrases, and not more complex syntactic constructions. Further research and experiments are required to clarify this.

Source code

The project files are posted in this public repository: https://github.com/fin-algo-lake-ai/stocks-tweets-project
Description of the main files is given in README.md

Conclusion

The work on this project brought some interesting results and observations!
The choice of the reference article for analysis turned out to be quite successful: the results were reproduced and even slightly improved using simple methods.

At the same time, in the process of examining the article and the code, we bumped into so many nuances that it became necessary to organize the second part of the project: re-create the dataset with corrections and train the models again. The results of this second stage also turned out to be promising - despite the complexity of the subject area (predicting future price behavior), we managed to get statistically significant accuracy values (0.582) with a random baseline of 0.500 + -0.003.
A separate interesting conclusion was that the size and algorithmic complexity of machine learning models do not always determine the result of their application in each specific case, requiring researchers to conduct more versatile experiments at the modeling stage.

Possible next steps:

- Calculate business-oriented metrics (closer to the real potential usage of the idea)
- Labels: Take not just the next calendar day, but the next N days (up to a week). Also it's possible to experiment with the 0.5% threshold that is used for class label calculation.
- Models: Finish results with RNN approaches (LSTM / GRU) - the main runs have already been carried out, they need to be added to the final ensemble.

References

Yilmaz, E. S., Ozpolat, A., & Destek, M. A. (2022). Do Twitter sentiments really effective on energy stocks? Evidence from the intercompany dependency. Environmental Science and Pollution Research, 29(52), 78757-78767.
https://www.academia.edu/download/89681397/s11356-022-21269-9.pdf

Jaggi, M., Mandal, P., Narang, S., Naseem, U., & Khushi, M. (2021). Text Mining of Stocktwits Data for Predicting Stock Prices. arXiv preprint arXiv:2103.16388.
https://arxiv.org/abs/2103.16388

Rahul Pandey R. (2021) Common Pitfalls to Avoid in Forecasting Models for Stock Price Prediction
https://medium.com/geekculture/common-pitfalls-to-avoid-in-forecasting-models-for-stock-price-prediction-3a7c3ff8b80

Adusumilli, R. (2020). NLP in the Stock Market. Leveraging sentiment analysis on 10-k fillings as an edge
https://towardsdatascience.com/nlp-in-the-stock-market-8760d062eb92

Sun, Y., Liu, X., Chen, G., Hao, Y., & Zhang, Z. J. (2020). How mood affects the stock market: Empirical evidence from microblogs. Information & Management, 57(5), 103181.
https://www.sciencedirect.com/science/article/pii/S0378720618307183

Yuz, T. (2018). A Sentiment Analysis Approach to Predicting Stock Returns
https://medium.com/@tomyuz/a-sentiment-analysis-approach-to-predicting-stock-returns-d5ca8b75a42