COMPUTER RESEARCH AND MODELING
2023 VOL. 15 NO. 1 P. 157–170
DOI: 10.20537/2076-7633-2023-15-1-157-170

Ки&М

ENGINEERING AND TELECOMMUNICATIONS

UDC: 519.8

Development of and research into a rigid algorithm for analyzing Twitter publications and its influence on the movements of the cryptocurrency market

I. S. Makarov[a], E. R. Bagantsova[b], P. A. Iashin[c], M. D. Kovaleva[d], E. M. Zakharova[e]

Moscow Institute of Physics and Technology,
9 Institutskiy per., Dolgoprudny, Moscow Region, 141701, Russia

E-mail: [a] i.s.m.mipt@yandex.ru, [b] bagantsova.er@phystech.edu, [c] iashin.pa@phystech.edu,
[d] kovaleva.md@phystech.edu, [e] zakharova.em@mipt.ru

Received 01.11.2022.
Accepted for publication 23.12.2022.

Social media is a crucial indicator of the position of assets in the financial market. The paper describes the rigid solution for the classification problem to determine the influence of social media activity on financial market movements. Reputable crypto traders influencers are selected. Twitter posts packages are used as data. The methods of text, which are characterized by the numerous use of slang words and abbreviations, and preprocessing consist in lemmatization of Stanza and the use of regular expressions. A word is considered as an element of a vector of a data unit in the course of solving the problem of binary classification. The best markup parameters for processing Binance candles are searched for. Methods of feature selection, which is necessary for a precise description of text data and the subsequent process of establishing dependence, are represented by machine learning and statistical analysis. First, the feature selection is used based on the information criterion. This approach is implemented in a random forest model and is relevant for the task of feature selection for splitting nodes in a decision tree. The second one is based on the rigid compilation of a binary vector during a rough check of the presence or absence of a word in the package and counting the sum of the elements of this vector. Then a decision is made depending on the superiority of this sum over the threshold value that is predetermined previously by analyzing the frequency distribution of mentions of the word. The algorithm used to solve the problem was named benchmark and analyzed as a tool. Similar algorithms are often used in automated trading strategies. In the course of the study, observations of the influence of frequently occurring words, which are used as a basis of dimension 2 and 3 in vectorization, are described as well.

Keywords: text analysis, natural language processing, Twitter activity, frequency analysis, feature selection, classification problem, financial markets, decision tree, random forest, benchmark

Citation: *Computer Research and Modeling*, 2023, vol. 15, no. 1, pp. 157–170.

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2023 Т. 15 № 1 С. 157–170
DOI: 10.20537/2076-7633-2023-15-1-157-170

Ки&М

ИНЖИНИРИНГ И ТЕЛЕКОММУНИКАЦИИ

УДК: 519.8

Разработка и исследование жесткого алгоритма анализа публикаций в Twitter и их влияния на движение рынка криптовалют

И. С. Макаров[a], Е. Р. Баганцова[b], П. А. Яшин[c], М. Д. Ковалёва[d], Е. М. Захарова[e]

Московский физико-технический институт,
Россия, 141701, Московская область, г. Долгопрудный, Институтский пер., 9

E-mail: [a] i.s.m.mipt@yandex.ru, [b] bagantsova.er@phystech.edu, [c] iashin.pa@phystech.edu,
[d] kovaleva.md@phystech.edu, [e] zakharova.em@mipt.ru

Получено 01.11.2022.
Принято к публикации 23.12.2022.

Посты в социальных сетях являются важным индикатором, отображающим положение активов на финансовом рынке. В статье описывается жесткое решение задачи классификации для определения влияния активности в социальных сетях на движение финансового рынка. Отбираются аккаунты авторитетных в сообществе крипто-трейдеров-инфлюенсеров. В качестве данных используются специальные пакеты сообщений, которые состоят из текстовых постов, взятых из Twitter. Приведены способы предобработки текста, заключающиеся в лемматизации Stanza и применении регулярных выражений, для очищения зашумленных текстов, особенностью которых является многочисленное употребление сленговых слов и сокращений. Решается задача бинарной классификации, где слово рассматривается как элемент вектора единицы данных. Для более точного описания криптовалютной активности ищутся наилучшие параметры разметки для обработки свечей Binance. Методы выявления признаков, необходимых для точного описания текстовых данных и последующего процесса установления зависимости, представлены в виде машинного обучения и статистического анализа. В качестве первого используется отбор признаков на основе критерия информативности, который применяется при разбиении решающего дерева на поддеревья. Такой подход реализован в модели случайного леса и актуален для задачи выбора значимых для «стрижки деревьев» признаков. Второй же основан на жестком составлении бинарного вектора в ходе грубой проверки наличия либо отсутствия слова в пакете и подсчете суммы элементов этого вектора. Затем принимается решение в зависимости от преодоления этой суммой порогового значения, базирующегося на уровне, предварительно подобранном с помощью анализа частотного распределения упоминаний слова. Алгоритм, используемый для решения проблемы, был назван бенчмарком и проанализирован в качестве инструмента. Подобные алгоритмы часто используются в автоматизированных торговых стратегиях. В процессе исследования также описаны наблюдения влияния часто встречающихся в тексте слов, которые используются в качестве базиса размерностью 2 и 3 при векторизации.

Ключевые слова: анализ текста, обработка естественного языка, активность в Twitter, частотный анализ, отбор признаков, задача классификации, финансовые рынки, бенчмарк, случайный лес, решающие деревья

Introduction

Global community usually expresses their emotions and feelings about world news on the Internet. The most influential people are able to form opinions and coerce to action. It has been suggested that the sentiments expressed may help in predicting price changes of financial assets. Even cryptocurrencies experience the effects of social media engagements. Moreover, coins become more flexible depending on some people's opinion. The most vivid example of it is Elon Musk's (Tesla CEO) tweets affecting the Bitcoin price [itZone, 2022].

Related research demonstrates the potential of the approach of analyzing Twitter activity [Otabek, Choi, 2022]. The number of followers of the poster, the number of comments on a tweet, the number of likes, and the number of retweets are utilized there. Our study doesn't allow one to gather such data in a correct way because of the problem of data download mentioned below.

The step of preprocessing plays one of the crucial roles in our paper. In the study [Nosrati et al., 2022] the approach of removing commonly used words is suggested. There is a demonstration of the method to solve the spam classification problem using Naive Bayesian classifier as well. Another research [Kumar, Harish, 2018] supports the idea of removing syntax from the text, however, we consider that it can influence the sentiment of the sentences, so we keep it there for finer score of Stanford CoreNLP [Stanford CoreNLP, 2022; Stanza, 2022; Qi et al., 2020].

There are different methods of representing financial assets' prices and trends. One of the most valuable approaches is to use indicators, a special value counted using candlesticks' data [Neeson, 2020]. Usually an algorithmic approach is applied to analyze several indicators to make predictions and trades. The article of K. Senthamarai Kannan, P. Sailapathi Sekar, M. Mohamed Sathik and P. Arumugam [Senthamarai et al., 2010] describes multiple indicators such as Typical Price, Bollinger Bands, Relative Strength Index, Chaikin Money Flow Indicator and Moving Average. Prices and indicators are also used as features to create a model using machine learning [Zhao, Rinaldo, Brookins, 2019].

The feature selection stage could be divided by several types of methods: exploiting machine learning models or building hard algorithms. In the related research forward feature selection and recursive feature elimination are considered [Ansari, Ahmad, Fatima, 2019]. Due to the immense number of words used as features the operation of learning requires excessive computational cost and has no advantage in comparison with classic random forests.

The article presents the rigid method for solving the classification problem to predict financial market movements. The benchmark algorithm is used to categorize natural language texts according to content. The paper describes stages of preparing, analyzing, and featuring text data. Procedures of processing casual social media speech are contemplated as well. Statistics and machine learning provide methods to describe a set of words based on their frequency distribution and information criterion. The results of solving the classification problem are considered.

Training sample data preparation

Data selection

As a text data source Twitter API v2 is used [Twitter API]. The initial data contain the information about posting time in the format of ISO 8601 (UTC), author id, tweet id, and, of course, text of a tweet itself. The list of users has been taken from LunarCrush's top of influencers [LunarCrush, 2022]. Overall data consist of more than 250,000 tweets dated from 24 February 2021 posted by 138 users. Because of Twitter limits it is not allowed to download data distributed evenly in time using the academic research access. The data about candles for markup is provided by Binance [Binance, 2022].

Data preparation

Twitter provides the very raw test data that is needed to be processed for further research. Regular expressions are utilized to exclude emoji, hashtags, links, non-English language words, and slang abbreviations of words, while Stanza is used to lemmatize text. Uppercase letters in a string also are converted to lowercase.

A unit of data is formatted by the set of tweets, whose post time is included in the interval of time $t_1 - t_0$ with the period of $dt = t_2 - t_1 \leqslant t_1 - t_0$. In further references to the set of tweets we use a name of packages. If dt is less than $t_1 - t_0$ there is a problem of overlay: this circumstance may lead us to the risk of having equal data in different packages, and to overfitting, consequently. Also due to the download problem packages are not equal in size, their magnitude is distributed in the following way: earlier packages contain less data that the later ones.

Data markup

The price data is taken from Binance. The label is stated by the following rules.

- Look for a label for each coin we are interested in.
  - Consider the moment of time — the upper bound of the package.
  - Find the candlestick corresponding to the upper bound.
  - Find the candlestick that is t_3 away from the upper bound.
  - Find the difference between the close price of both candlesticks.
  - If the difference found is less than zero, then the binary label is -1, otherwise $+1$.
- Summarize all labels by each coin.
  - If the sum of all binary labels is less than zero, the binary label of the package is -1, otherwise $+1$.

The process of data markup is shown in Figure 1.

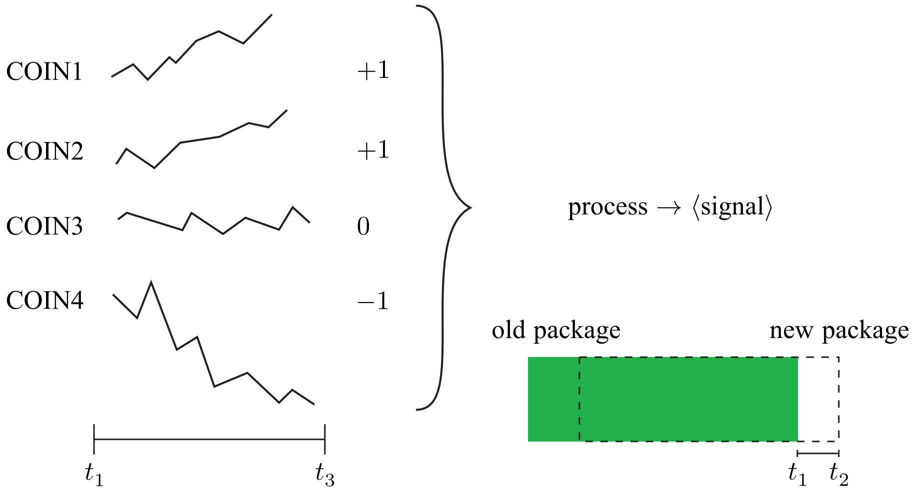

Figure 1. Package markup visualization

Feature selection

The main problem in building an algorithm for solving the classification problem is selecting features. Features in our case are defined as words that could be encountered in a package. First of all, the frequency that is defined as

$$F_1 = \frac{N}{M}, \tag{1}$$

where N is the number of packages where the word exists and M is the number of packages overall, is counted for each word. Also, we explored each word frequency of existence in a package F_2 and how distribution is changed depending on the market movements. In other words, the frequency of mentions of a word in a package with different markup (+1 and −1) is checked.

To decrease the influence of the download problem, we use relative values instead of absolute ones. That is why the more relative value of F_2 is defined as the ratio of the number of times this word has appeared in the package to the number of words in the package. Passing through the packages and counting F_2 lead us to observing histograms with the distribution of this value.

The structure of a feature is defined with several parameters: a word itself, the existence in a package of which should be checked, and the set of zones in histograms for the word. Each zone contains the left and the right boards and power — the amount that is added to the result of the package when it enters the zone.

The parameters of a feature are as follows.

1. **Power**: the value is defined with the help of intersection of histograms with different markups. The smaller the area of intersection, the more powerful the zone. The histogram intersection value might be counted using algorithm 1.

2. **Zones**: the areas are chosen in the histogram, where there is a clear superiority of one type of packages over another one. Meanwhile emissions in the distribution of a minor type are ignored. Zone borders might be found using algorithm 2.

Algorithm 1. Histogram intersection

```
$square_{max} \leftarrow 0$
$square_{min} \leftarrow 0$
for $i$ in $1 : hist_1$ do
    $square_{max}$ += $\max(hist_1[i], hist_2[i])$
    $square_{min}$ += $\min(hist_1[i], hist_2[i])$
end for
return $\frac{square_{min}}{square_{max}}$
```

Another option to build features is usage of machine learning methods. The most relevant method is to use decision trees. They are built from the root to the leaves, and at each stage there is an attempt to split the vertex into two. We need to select the attribute to split a vertex, and the threshold with which the value of this attribute will be compared. If the attribute value is smaller than this threshold, then the object is sent to the left subtree, if it is larger, to the right subtree. The selection of the attribute and threshold is carried out according to the following criteria:

$$Q(X_m, j, t) = \frac{|X_l|}{|X_m|}H(X_l) + \frac{|X_r|}{|X_m|}H(X_r) \to \min_{j,t}, \tag{2}$$

where m is the vertex to be split, t is the threshold value, j is a feature used for splitting, X_m is a set of objects from training sample, $X_l = \{x \in X_m \mid x^j \leqslant t\}$, $X_r = \{x \in X_m \mid x^j > t\}|$. $H(X)$ is an information

Algorithm 2. Searching of histogram areas

```
define $areas$ as an empty array
$l\_border \leftarrow 0$
$r\_border \leftarrow 0$
for $i$ in $1 : hist_1$ do
  if $l\_border == 0$ and $hist_1[i] \geqslant hist_2[i]$ then
    $l\_border \leftarrow i$
  end if
  if $l\_border \mathrel{!}= 0$ and $hist_1[i] \leqslant hist_2[i]$ then
    $r\_border \leftarrow i$
    if $r\_border - l\_border \geqslant min\_zone\_length$ and histogram intersection of $hist_1[l\_border : r\_border]$ and $hist_2[l\_border : r\_border] \leqslant max\_intersection$ then
      push area with defined $l\_border$ and $r\_border$ to $areas$
    end if
    $l\_border \leftarrow 0$
    $r\_border \leftarrow 0$
  end if
end for
```

criterion, which value should be the smaller the smaller the spread of responses in X. In our research the Gini index is used. Let p_k be the proportion of objects of class k in the sample X:

$$p_k = \frac{1}{X}\sum_{i\in X}[y_i = k]. \tag{3}$$

Finally, the Gini index is defined as follows:

$$H(X) = \sum_{k=1}^{K} p_k(1 - p_k). \tag{4}$$

The less the given weighted sum Q, the more features and thresholds are suitable for vertex division. The more the sum of reduction of the information criterion in vertex where feature j is used for splitting:

$$R_j = \sum_m H(X_m) - \frac{|X_l|}{|X_m|}H(X_l) - \frac{|X_r|}{|X_m|}H(X_r), \tag{5}$$

the more informative feature was in building the tree.

Benchmark algorithm

Structure

Benchmark is a rigid algorithm for solving the classification problem. Its structure looks like the following.

1. A package is sent to the input of the algorithm.

2. For this package, the presence of one or another feature is checked. If it is available, a certain number is added or subtracted to the total bill of the package.

3. If, according to the results of the check, the threshold value has been overcome by all signs, then we get a certain signal at the output (+1 or −1, depending on the settings), otherwise the opposite.

Thus, benchmark is a sequence of if-else blocks built for checking the existence of one of the features in a package. With a fixed set of features, a one-parameter optimization problem is solved by threshold value. In this case, a full search algorithm was used to solve this problem.

Testing results

In the course of the research, different types of benchmark models were studied. The threshold value search was carried out on a sample containing 2500 more packages of tweets. The data was split into testing and training samples in a 80 to 20 ratio. The testing sample consists of the newest packages, i. e., the sample was sorted by time in ascending order before separation. As the result of testing the model, the value of the accuracy of predicting labels on the marked-up test data is given in Table 1.

Table 1. Benchmark results with different approaches to data splitting and tokens search

Model name	Training		Testing	
	Accuracy	Threshold	Accuracy	Threshold
Testing	0.729	−0.6	0.665	−0.6
Validation	0.732	−1.1	0.609	−1.7
Random-Forest	0.684	0.1	0.625	0.7

The model described first is the Testing model. It consists of twenty tokens including words such as “break”, “high”, “no”, “top”. The histograms of these words are represented in Figure 2.

The horizontal axis represents the value of F_2 for each package in the sample and the vertical axis represents the number of packages overall. A single bin as high as many packages contains a current token with parameter F_2 between bin borders.

Orange bins represent values for negative packages (i. e., leading to a decrease in price) and light-blue bins represent values for positive packages (i. e., leading to an increase in price). Several positive and negative zones might be extracted using the histograms and then used to create the model. For example, very strong negative zones are in the “high” token’s histogram, as well as in the “break” token’s histogram. In the “no” and “top” tokens’ histograms several less strong positive and negative zones might be pulled out.

An interesting trend is that other words in the benchmark algorithm are not so valuable for independent human analysis. For example, there is no possibility for one person to recognize the mood and feelings of people analyzing the word “when”. Otherwise, for the algorithm the word seems to be valuable. The histogram of this token is represented in Figure 3.

So at least one strong negative zone might be extracted similarly to the described tokens “high” and “break” by utilizing their distribution. Moreover, the algorithm of zones extracting pulls out more weaker zones.

The parameters presented in Table 1 are the best according to the full-search algorithm we used to find the threshold value. The result of algorithm work is represented in Figure 4.

The horizontal axis represents the threshold value and the vertical one represents the accuracy of the algorithm using a current threshold value. The blue line demonstrates the result of work of the model for the training sample and the orange line shows the same for the testing sample. It might be found out that accuracy becomes constant if the absolute value of threshold is too large whether it is negative or positive. This means that the amount of tokens’ strength is not enough to break the threshold.

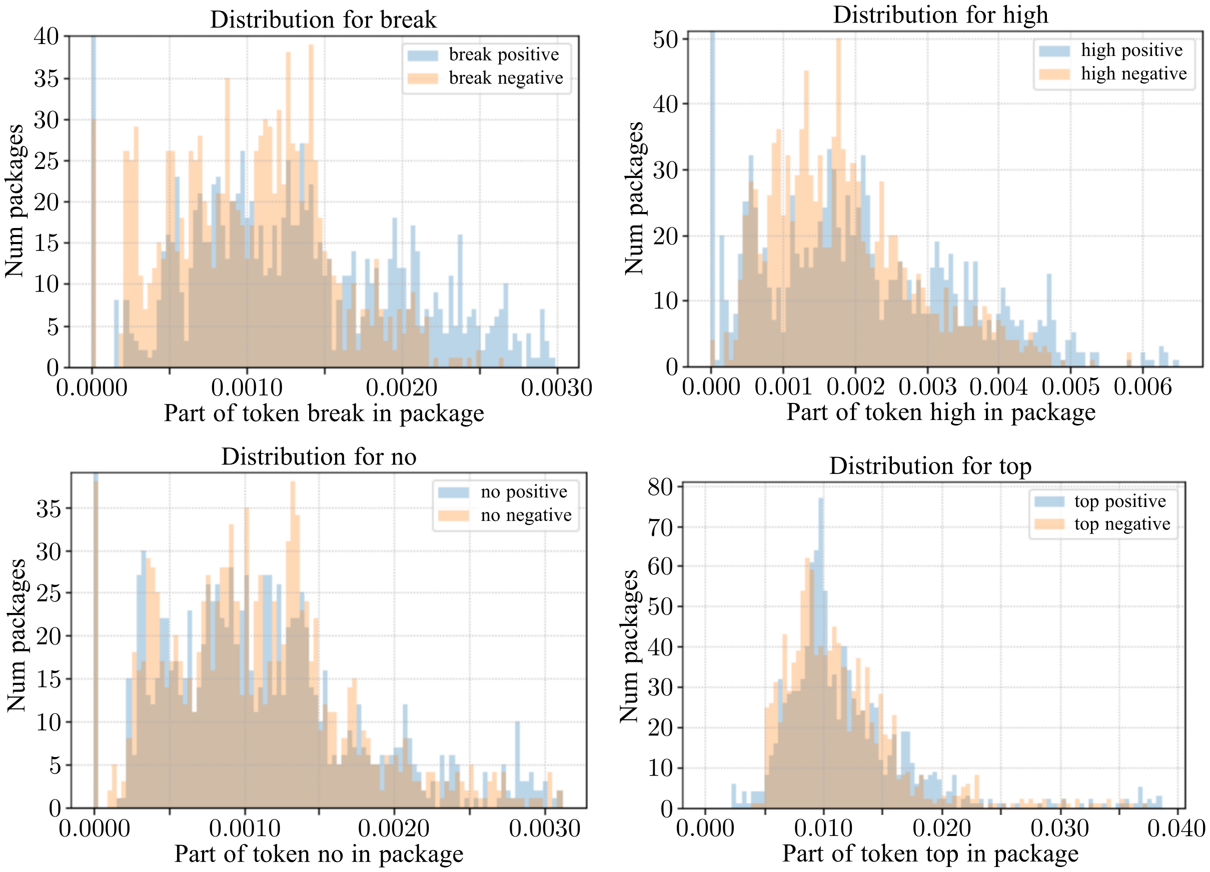

Figure 2. Distributions for tokens “break” (left upper corner), “high” (right upper corner), “no” (left lower corner), “top” (right lower corner)

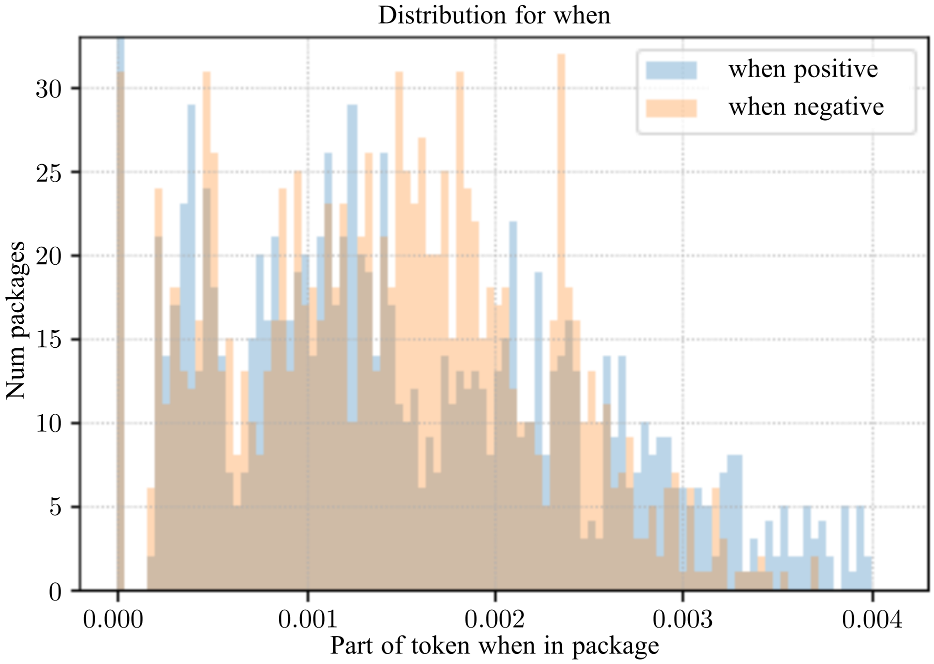

Figure 3. Histogram for token “when”

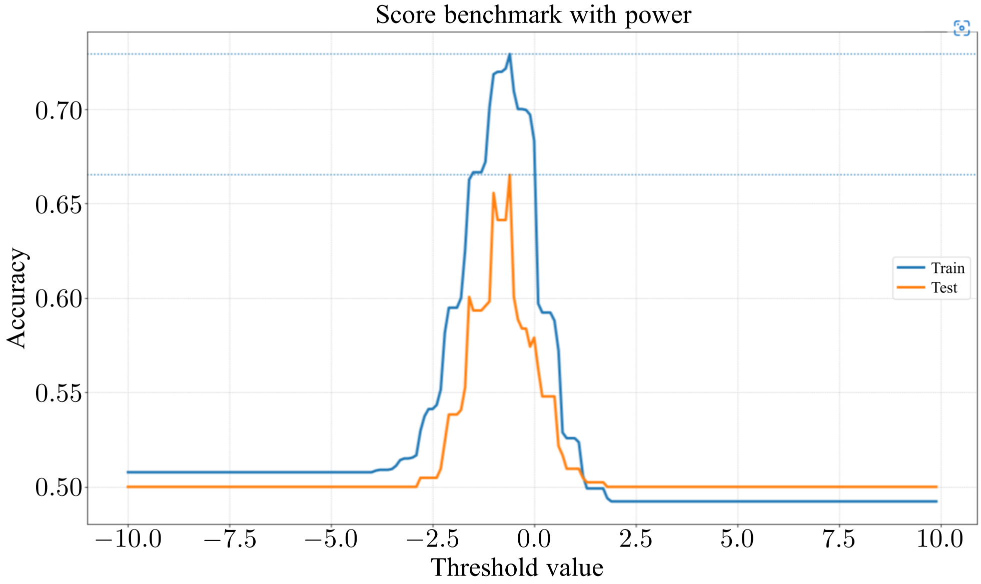

Figure 4. Full-search algorithm report for the benchmark with sorted by time testing and training packages

The model produces best predictions with threshold value equal to −0.6, which corresponds to 0.665 of accuracy for the last 20 % of the analyzed time interval.

Block shuffle results

Since we have been working with market data samples and trying to predict the market movements, the data sample needs to be shuffled before searching for the algorithm parameters. The article [Fabrice, 2019] describes the problem of splitting time series which is similar to the problem of package overlaying. But the problem does not allow us to shuffle the data randomly.

It was decided to do the block shuffle to verify the results obtained. The block shuffle is the method of shuffling the sample by dividing it into a few blocks. In the case of cryptocurrency market data it was decided to use August 2021 and December 2021 periods as testing sample and the other periods as training sample because of the variety of price movement types during the testing periods we choose.

Table 1 represents results for the Validation model. The best benchmark contains tokens such as “strong”, “green”, “keep”, “high”, whose histograms are represented in Figure 5.

Note that, there are strong negative zones in the histograms for the tokens “strong”, “keep”, and “high”. Several medium strength zones of “keep”, and “high” tokens might be extracted as well.

The full-search algorithm is involved in obtaining the best threshold value. The result of the algorithm work is represented in Figure 6.

The model produces best predictions with threshold value equal to −1.7, which corresponds to 0.609 of accuracy for the August 2021 and December 2021 time periods.

Moreover, the value of testing result maximum accuracy threshold is not equal to the training result maximum accuracy threshold. Otherwise, the accuracy of training is high for threshold value and equal to −1.7.

Benchmark with random forest tokens results

The random forest classifier was also used in searching for tokens for benchmark structure. Using different methods, tokens were obtained and the algorithm was examined. The best result of the

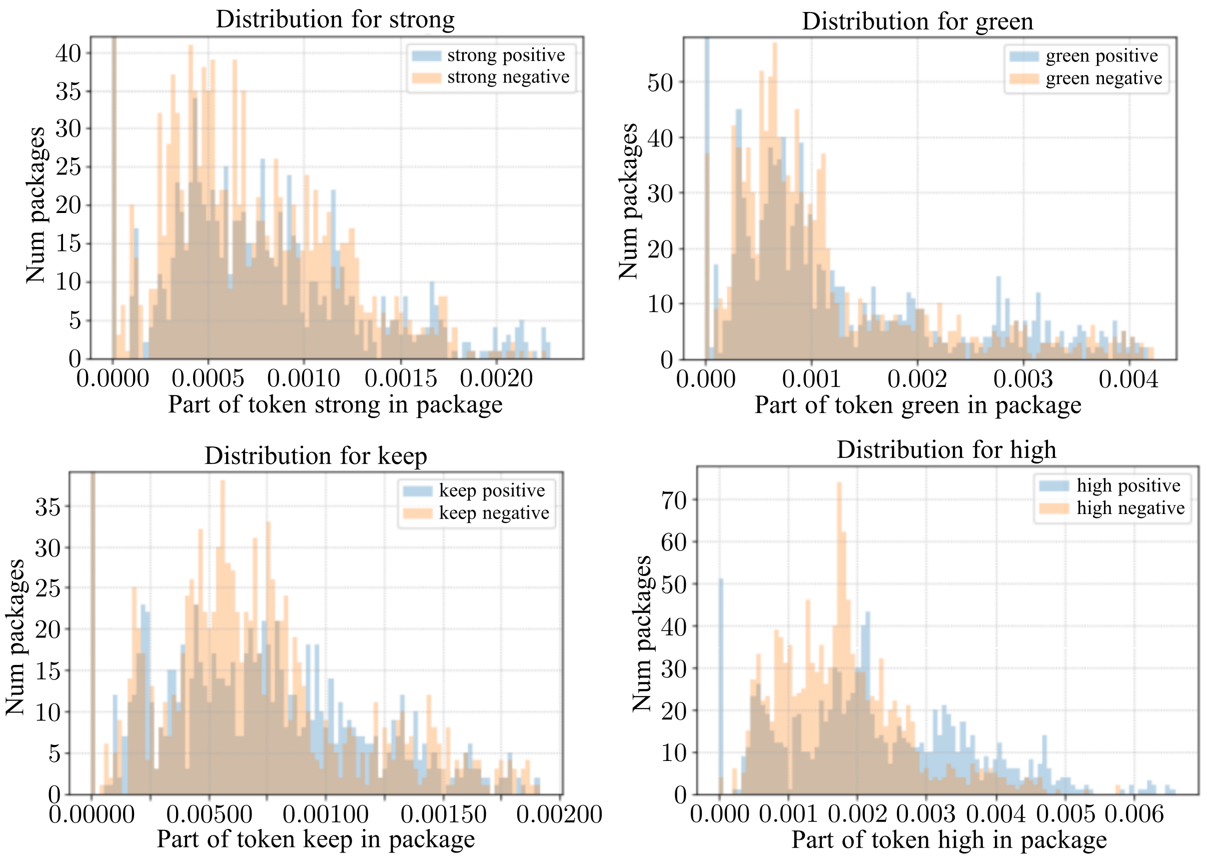

Figure 5. Distributions for tokens “strong” (left upper corner), “green” (right upper corner), “keep” (left lower corner), “high” (right lower corner)

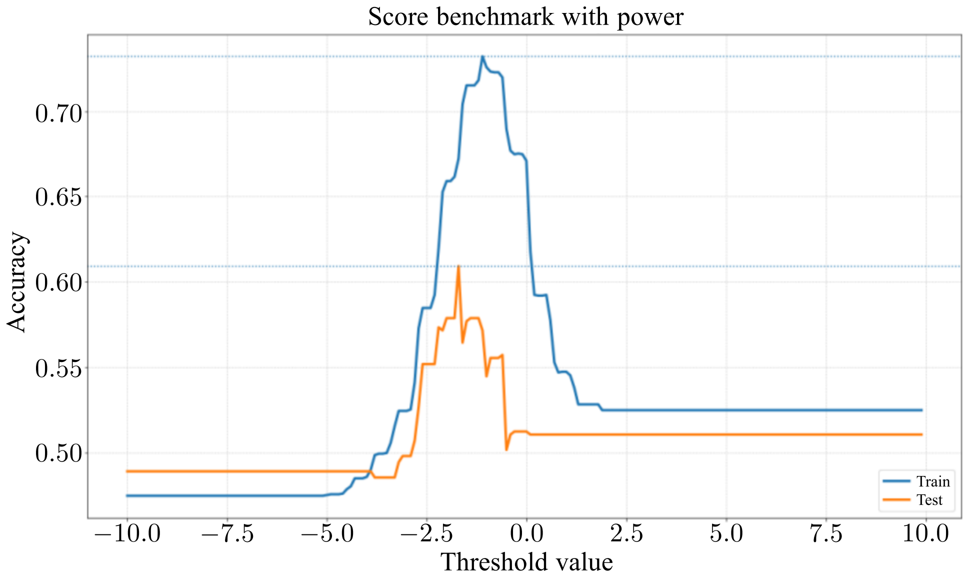

Figure 6. Full-search algorithm report for the benchmark with blocked shuffled testing and training packages

approach includes tokens chosen by the information criterion described above. The algorithm matches such words as “big”, “will”, “not”, “trade” and others up to 20 overall.

The random forest classifier provides tokens for the structure of benchmark and the threshold value was obtained by the full-search algorithm. The result for the Random-Forest model are in Table 1. Figure 7 represents the result of the algorithm testing and training. Samples were separated using the block shuffle approach. The testing sample consists of August 2021, December 2021 and May 2022 data.

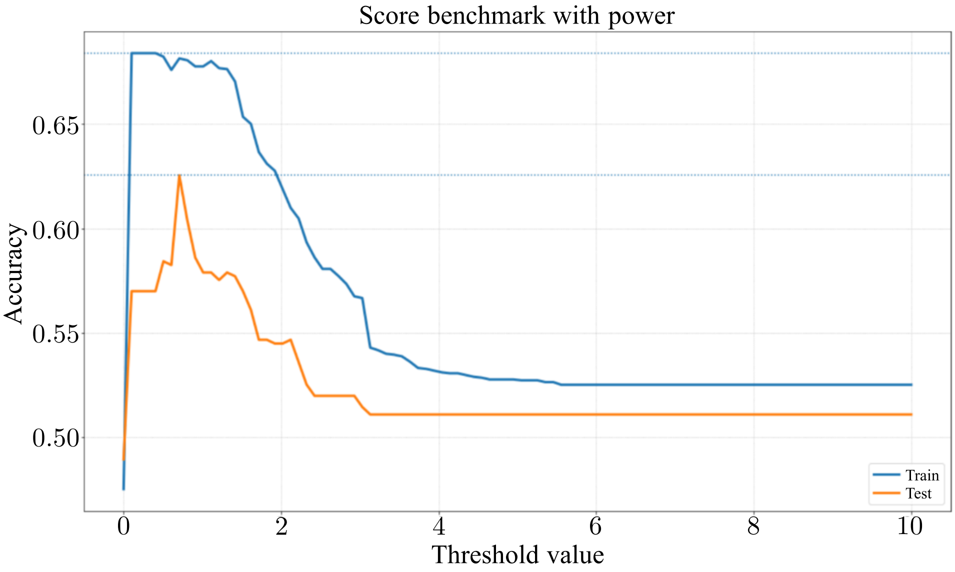

Figure 7. Full-search algorithm report for the benchmark with blocked shuffled testing and training packages

The model provides best results of prediction which is 0.625 of accuracy for the testing periods. The threshold value is 0.7 in this case.

The phenomenon of minor amount of tokens

During the investigations an interesting phenomenon was discovered. As a benchmark algorithm consists of a different number of tokens, it might be interesting to create such an algorithm using one, two, or three, which are minor numbers, tokens only. The most interesting results are represented in Figure 8. It contains a standard full-search algorithm report for benchmarks with a single token. Tokens mentioned are “but”, “daily”, “few”, “that”.

The tokens represented in Figure 8 provide better result than many other benchmark algorithms with multiple tokens.

A difference from the previous figures is that in the case of a minor number of tokens a full-search algorithm report is stepped. It is connected with zones of tokens structure and their powers.

For example, if a benchmark worked without any token, a single kink would be with threshold value equal to zero because without any tokens for every package benchmark predicts the same: negative or positive related to threshold value. If it is negative, then the prediction is positive and, otherwise, if it is positive the prediction is negative.

If a single token of the benchmark consisted of single negative zone, a right border of a step would be zero and the left border of the step would be equal to a power of zone.

In the case of multiple tokens and even multiple zones of single token the picture is more complicated. Otherwise, it still obeys the described principles.

As the result for single tokens we obtained that it is possible to conduct another experiment on the double token algorithms. Using the previous result, tokens “daily” and “few” were concatenated. Figure 9 represents the full-search algorithm report in the case of created double token benchmark.

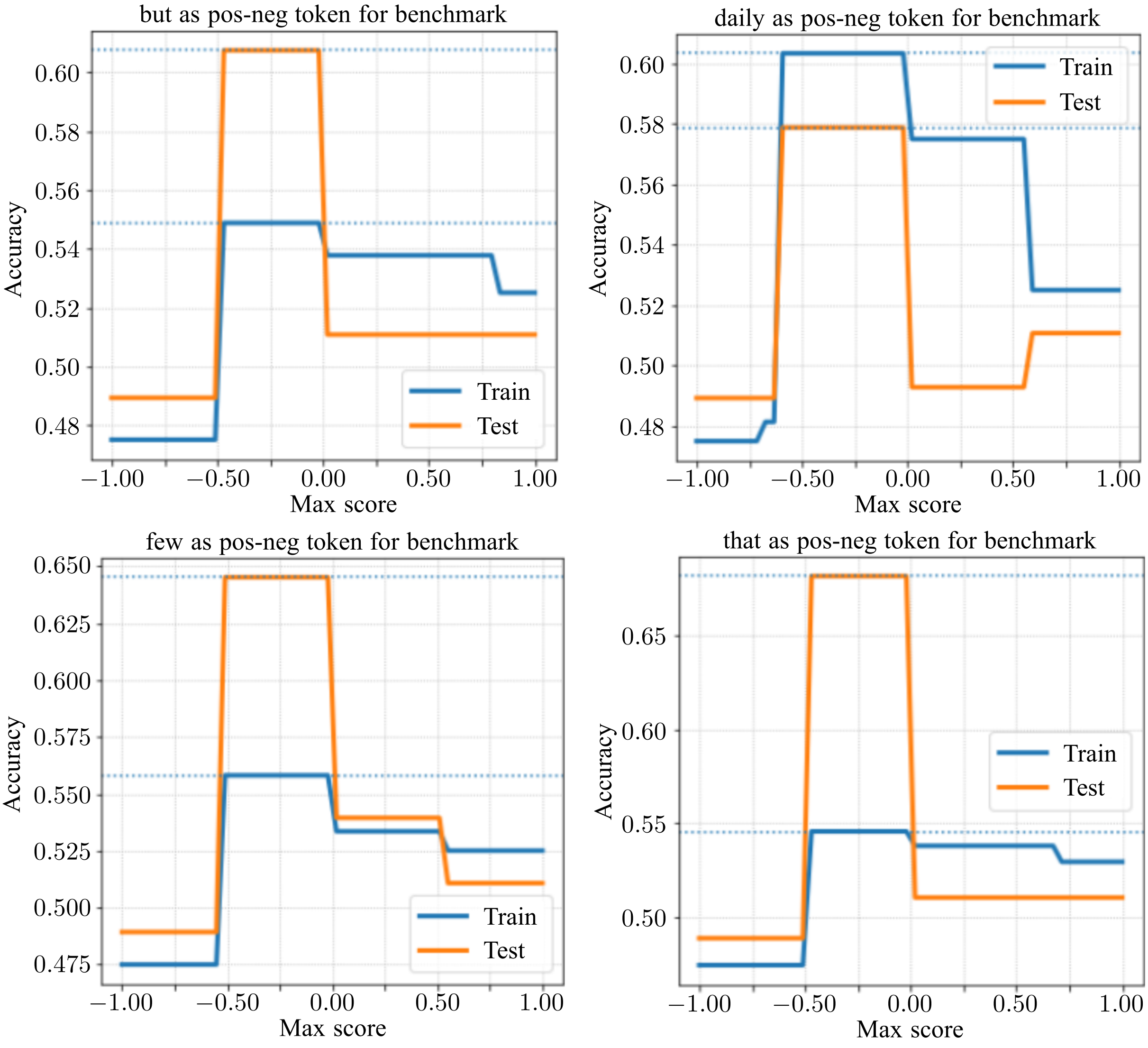

Figure 8. Full-search algorithm for threshold value reports for tokens "but" (left upper corner), "daily" (right upper corner), "few" (left lower corner), "that" (right lower corner)

The combined tokens produce even better result than single token algorithms including these words. This means that the investigation of the best set of tokens for the benchmark is in the appropriate direction. Moreover, the stepped structure is more complicated than the one in the case of a single token benchmark.

The next step was to join another token and to test the triple token benchmark. A full-search algorithm among the words mentioned has been used to select the third token. One of the best obtained result includes the word "that". A report of the full-search threshold value algorithm is in Figure 10.

To conclude, there are several tokens among the words in packages which cover the others optimally. This means that there is no necessity to create the algorithm using lots of tokens. Instead, it might be possible to find the best tokens according to the phenomenon described and create the model.

Conclusion

This article describes the results of data preparation and markup, as well as studies of various parameters of a rigid algorithm designed to solve the problem of binary text classification based on

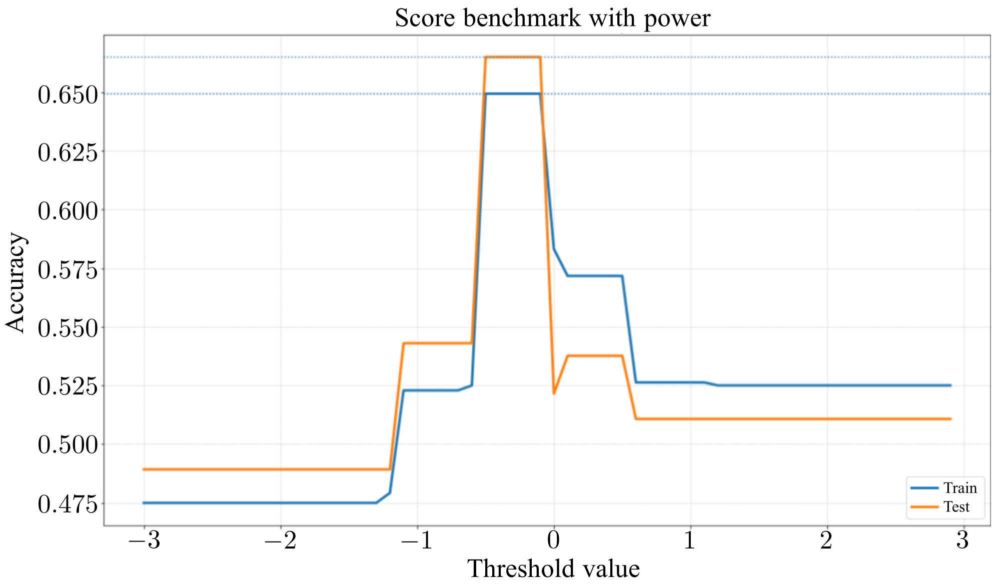

Figure 9. Full-search algorithm report for the benchmark with blocked shuffled testing and training packages

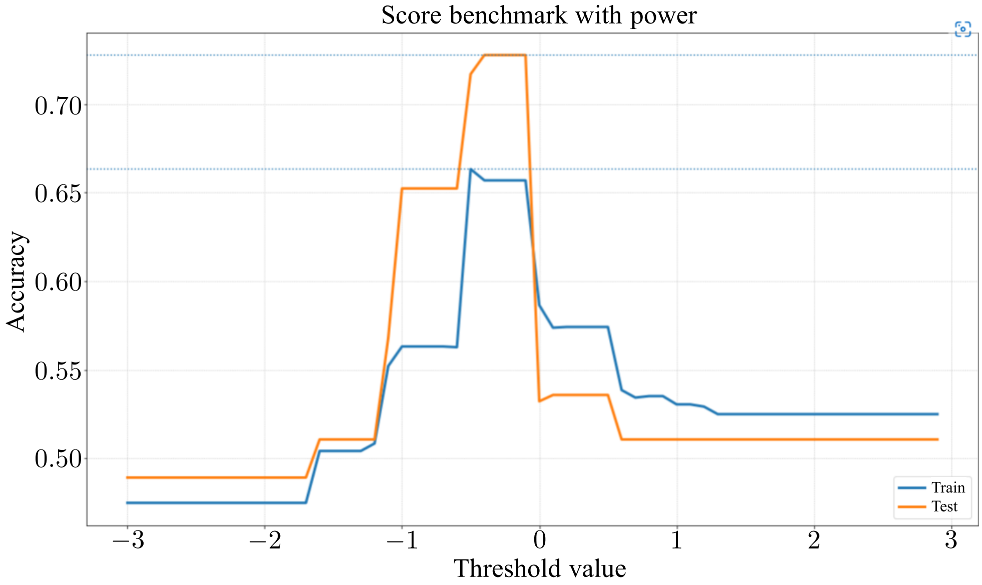

Figure 10. Full-search algorithm report for the benchmark with blocked shuffled testing and training packages

labeled data, including the initial data of candlestick charts taken from Binance. The text data represents tweets of most influential people, whose activity is devoted to highly market capitalized tokens, from the top formed by LunarCrush. The best results were achieved using features selected based on the information criterion, and on a small number of features that enable one to cover large parts of the dataset. In the course of research, the phenomenon of data shuffling was illustrated as well. Hence, the most credible result for the model built on the random forest tokens is 0.625. The resulting model can be applied in automated trading systems.

References

Ansari M. Z., Ahmad T., Fatima A. Feature Selection on Noisy Twitter Short Text Messages for Language Identification // International Journal of Recent Technology and Engineering. — 2019. — Vol. 8, No. 4. — P. 10505–10510.

Binance — a cryptocurrency exchange. — [Electronic resource]. — URL: https://www.binance.com/ (accessed: October 20, 2022).

Fabrice D. Financial Time Series Data Processing for Machine Learning // arXiv:1907.03010. — 2019.

itZone. Elon Musk tweets alluding to “break up” with Bitcoin. — [Electronic resource]. — URL: https://itzone.com.vn/en/article/elon-musk-tweets-alluding-to-break-up-with-bitcoin/ (accessed: October 20, 2022).

Kumar H. M. Keerthi, Harish B. S. A New Feature Selection Method for Sentiment Analysis in Short Text // Journal of Intelligent Systems. — 2018. — December 4.

LunarCrush — Social Intelligence for Crypto, NFTs and Stocks. — [Electronic resource]. — URL: https://lunarcrush.com/ (accessed: October 20, 2022).

Neeson S. Japanese candles. Graphical Analysis of Financial Markets. — Intellectual Literature, 2020. — 290 p. (in Russian).

Nosrati V., Rahmani M., Jolfaei A., Seifollahi S. A Weak-Region Enhanced Bayesian Classification for Spam Content-Based Filtering // ACM Transactions on Asian and Low-Resource Language Information Processing. — 2022. — DOI: https://dl.acm.org/doi/10.1145/3510420

Otabek S., Choi J. Twitter Attribute Classification with Q-Learning on Bitcoin Price Prediction // arXiv:2208.02610. — 2022.

Qi P., Yuhao Zhang Y., Yuhui Zhang Y., Jason Bolton J., Manning C. D. Stanza: A Python Natural Language Processing Toolkit for Many Human Languages // Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. — 2020. — P. 101–108.

Senthamarai Kannan K. et al. Financial stock market forecast using data mining techniques // Proceedings of the International Multiconference of Engineers and computer scientists. — 2010. — Vol. I.

Stanford CoreNLP. — [Electronic resource]. — URL: https://stanfordnlp.github.io/CoreNLP (accessed: October 20, 2022).

Stanza — A Python NLP Package for Many Human Languages. — [Electronic resource]. — URL: https://stanfordnlp.github.io/stanza/ (accessed: October 20, 2022).

Twitter API. — [Electronic resource]. — URL: https://developer.twitter.com/en/docs/twitter-api

Zhao D., Rinaldo A., Brookins C. Cryptocurrency Price Prediction and Trading Strategies Using Support Vector Machines // Preprint. Under review. — 2019.