UDC 004.021

K-NN'S NEAREST NEIGHBORS METHOD FOR CLASSIFYING TEXT DOCUMENTS BY THEIR TOPICS

Boyko N. I. – Candidate of Economics, Associate Professor, Associate Professor of the Department of Artificial Intelligence Systems, Lviv Polytechnic National University, Lviv, Ukraine.

Mykhailyshyn V. Yu. – Assistant Professor, Department of Artificial Intelligence Systems, Lviv Polytechnic National University, Lviv, Ukraine.

ABSTRACT

Context. Optimization of the method of nearest neighbors *k*-NN for the classification of text documents by their topics and experimentally solving the problem based on the method.

Objective. The study aims to study the method of nearest neighbors *k*-NN for classifying text documents by their topics. The task of the study is to classify text documents by their topics based on a dataset for the optimal time and with high accuracy.

Method. The *k*-nearest neighbors (*k*-NN) method is a metric algorithm for automatic object classification or regression. The *k*-NN algorithm stores all existing data and categorizes the new point based on the distance between the new point and all points in the training set. For this, a certain distance metric, such as Euclidean distance, is used. In the learning process, *k*-NN stores all the data from the training set, so it belongs to the "lazy" algorithms since learning takes place at the time of classification. The algorithm makes no assumptions about the distribution of data and it is nonparametric. The task of the *k*-NN algorithm is to assign a certain category to the test document *x* based on the categories *k* of the nearest neighbors from the training dataset. The similarity between the test document *x* and each of the closest neighbors is scored by the category to which the neighbor belongs. If several of *k*'s closest neighbors belong to the same category, then the similarity score of that category for the test document *x* is calculated as the sum of the category scores for each of these closest neighbors. After that, the categories are ranked by score, and the test document is assigned to the category with the highest score.

Results. The *k*-NN method for classifying text documents has been successfully implemented. Experiments have been conducted with various methods that affect the efficiency of *k*-NN, such as the choice of algorithm and metrics. The results of the experiments showed that the use of certain methods can improve the accuracy of classification and the efficiency of the model.

Conclusions. Displaying the results on different metrics and algorithms showed that choosing a particular algorithm and metric can have a significant impact on the accuracy of predictions. The application of the ball tree algorithm, as well as the use of different metrics, such as Manhattan or Euclidean distance, can lead to improved results. Using clustering before applying *k*-NN has been shown to have a positive effect on results and allows for better grouping of data and reduces the impact of noise or misclassified points, which leads to improved accuracy and class distribution.

KEYWORDS: method, cluster, classification, text document, subject, ball tree algorithm, metric.

ABBREVIATIONS

k-NN is a *k*-nearest neighbor method;

kd-tree is a *k*-dimensional tree;

*L*1 – distance is a Manhattan metric;

TF-IDF is a *TF* – term frequency, *IDF* – inverse document frequency;

CSV is a comma-separated value.

NOMENCLATURE

di is a text document;

CI is a appropriate classification of the document;

$f(x)$ is a label intended for the test document x;

Score (x,C_j) is a score assigned to a category based on the points of category K of the nearest neighbors to the test document X;

$sim\ (x,d_i)$ is a similarities between X and the training document D;

$y(d_i, C_j) \in \{0,1\}$ is a binary value of the category for the educational document regarding d_iC_j;

$w_{c_i^0}^{j+1}(t)$ is a denotes the new weight of the word t in the cluster C_i^0;

$w_{c_i^0}^{j}$ is a weight of the word t in the cluster C_i^0;

$w(t)_p$ is a weight of the word t in the text p;

$C_i^0$is a number of texts contained in the cluster C_i^0;

ClusterScore(x,C_j) is a score assigned to a category based on category points to test the document C_jX;

$\text{sim}(x,C_i^0)$ is a similarity between X and cluster in model $C_i^0m_0$;

$y(C_i^0, C_j) \in \{0,1\}$ is a cluster relative to $C_i^0C_j$;

N is a total number of signs;

count(c_i, C) is a number of votes for the class c_i in the set C;

argmax is a function that returns the index of the maximum value.

INTRODUCTION

In today's world, a large amount of information is created and accumulated daily in various formats. As the volume of textual information grows in various fields, effective methods of processing and analysis are increasingly needed. Therefore, it is important to be able to analyze and classify this information effectively.

Classification of documents on their topics can be useful for many tasks, for example, selecting documents that meet certain criteria, building recommender systems, analyzing text data in social networks, etc.

DOI 10.15588/1607-3274-2023-3-9

The importance of the task of classifying text documents by the method of nearest neighbors will reduce the dimension of the data, save information about the classification and increase its accuracy. It is also quite easy to use and does not require a lot of computing power, which makes it popular in many areas [1, 3].

This method does not require pre-modeling, which allows it to be used for online classification and for the classification of text documents with a small data set, which is a fairly common situation in natural language processing. In addition, *k*-NN can be applied to the classification of documents without regard to their contents, only based on information about the topics. In addition, *k*-NN is a fairly flexible algorithm, since it is possible to use different distance metrics and distinguish the weight of each sample depending on its significance for classification [5, 6].

The task of the nearest neighbors method is to classify new data based on their similarity with known data (training data set). The problem of the nearest neighbors method is guided by the concept that if points in the data space are close to each other, then the probability that they belong to the same class is high, therefore, it is solved accordingly according to the principle such that in the variant *k*-NN each feature belongs to the predominant class of nearest neighbors, where *k* is the method parameter. The basis of the *k*-NN method is the fact that, according to the compactness hypothesis, it is expected that the test feature *d* will have the same label as the learning features in the local region surrounding the sign *d* [2, 4].

In the case of researching the use of the nearest neighbors method *k*-NN to classify text documents according to their topics, the novelty is that it offers the use of a method that is quite simple and effective to solve the complex problem of classifying text documents by their topics. The study proposes the use of clustering and dimensionality reduction to improve the quality of text classification. In addition, the study compares the efficiency of different types of term oscillation and different *k* values in the *k*-NN method for classifying text documents. Thus, the study expands our understanding of how the *k*-NN nearest neighbors method can be applied to classify text documents by their topics and helps to improve methods for classifying texts.

The aim of the study is to train the method of *k*-NN's nearest neighbors to classify text documents by their topics.

The subject of research is the creation and optimization of the method of nearest neighbors *k*-NN for the classification of text documents by their topics, as well as the solution of the problem based on the method experimentally.

The main objectives of the study are:

– General overview of the *k*-NN nearest neighbors method for creating a software solution for classifying text documents by their topics.

– Development of a system that can automatically classify text documents by their topics.

– Reducing classification errors to improve system accuracy.

– Research on the effectiveness of *k*-NNs in classifying documents with different numbers of categories and developing methods to improve efficiency in such cases.

1 PROBLEM STATEMENT

The purpose of the study is to build a model that can automatically assign a category to a new text document.

Suppose we have a set of text documents $D = \{d_1, d_2, ..., d_n\}$, where each document is represented as a sequence of words or tokens. Each d_i document belongs to one of the predefined classes or categories $C = \{c_1, c_2, ..., c_k\}$.

To build a model, we have a training dataset consisting of pre-classified documents and corresponding classes.

Mathematically, the problem of classification of text documents can be formulated as follows:

Given: Training dataset $D\_train = \{(d_1, c_1), (d_2, c_2), ..., (d_m, c_m)\}$, where d_i is a text document and c_i is its corresponding classification.

Find: *Function* $f:D\_test \rightarrow C$ which can categorize a new text document from test set *D_test* into one of *C* classes.

2 LITERATURE REVIEW

A special role for research is the methods of classification and clustering of text data. In the study [1, 3], the authors provide an introduction to the *k*-NN nearest neighborhood method and consider its application to the classification of text documents. They describe how the *k*-NN method can be used to classify texts and provide examples of how this method can be applied to data from various fields, including biology, medicine, and e-commerce.

This paper [7] is an important source for studying the basics of textual information processing and data retrieval. The book covers a wide range of topicss from the field of information retrieval, including index building, weighted estimation methods, vector models, thematic modeling, ranking, and more.

The paper [8] describes a wide range of methods of machine learning and statistical data analysis. It is useful for researchers working with the *k*-NN closest neighbor method to classify text documents by their topics.

The book [10] contains a lot of material about probabilistic models, multivariate data analysis, teaching methods, and graphical models. These techniques can be useful for improving the accuracy of classification using the *k*-NN closest neighbor method. In addition, the book contains numerous examples that demonstrate how different machine-learning methods can be applied to solve real-world problems.

The study [13] describes in detail the theoretical and practical aspects of working with text, including topicss

DOI 10.15588/1607-3274-2023-3-9

such as statistical models of language, thematic modeling, tone analysis, machine translation, and others.

The book discusses various methods of vector representation of text that can be used to build a model for the *k*-NN method and also discusses other machine-learning methods for classifying text documents, such as the naïve Bayesian classifier and the support vector method.

In particular, the authors [14] describe in detail the use of bag-of-words models and address the construction of a dataset for training and testing the model, as well as the choice of model parameters, such as the number of nearest neighbors to be used to classify documents.

The research [15] is devoted to the analysis of text data and methods of their processing, in particular the problems of classification and clustering of documents. It describes various methods, including the *k*-NN nearest neighbor method.

The book [12] discusses the following problems:

– Training, statistical methods, and association rules in the field of text mining, which describe different approaches to analyzing text data, including machine techniques.

– Building machine learning models used to classify and cluster documents, in particular the *k*-NN closest neighbor method.

– Preparing data for text analysis, including feature selection and data dimensionality reduction. The book describes in detail how to select the most significant features from the text that will improve the results of data analysis. The book also discusses methods for reducing the dimensionality of data, such as the principal component method and clustering method.

– The book contains many examples of applications of text mining techniques, including text tone analysis, document classification, and email spam detection.

Thus, the analysis of text documents is quite an important topics in our time, since the number of their application in various fields is increasing every day. Accordingly, consideration is relevant for this study and will help in improving the accuracy of the model.

3 MATERIALS AND METHODS

The *k*-nearest neighbor method is a metric algorithm for automatically classifying objects or regression. The *k*-NN algorithm stores all existing data and classifies the new point based on the distance between the new point and all points in the training set. To do this, use a specific distance metric, such as Euclidean distance. In the process of learning, *k*-NN stores all the data from the training set, so it belongs to the "lazy" algorithms since learning takes place at the time of classification. The algorithm makes no assumptions about the distribution of data and it is nonparametric [1, 11].

For the classification of text documents on the topics, the *k*-nearest neighbors method was chosen because of several reasons: simplicity and ease of implementation, high accuracy, and the ability to take into account the importance of each feature:

The *k*-NN method also has some disadvantages, in particular, it can be sensitive to noise and a large number of features and may require a significant amount of memory and computing resources when processing large amounts of data. Therefore, before proceeding with the classification of text documents, you can apply actions that can improve the quality and performance of the algorithm and its accuracy, namely [13]:

– Apply noise or unnecessary signs to data before using the *k*-NN method, which can reduce their impact on forecasting and help make the algorithm more efficient.

– Use distributed computing systems to handle large amounts of data, which can reduce the load on memory and computing resources. You can also use the approach of reducing the dimensionality of the data, which allows you to reduce the number of features and simplify the data space.

– Weights can be assigned to each sign depending on its importance for forecasting. This can help reduce the impact of less important features on forecasting and increase accuracy.

The task of the *k*-NN algorithm is to assign a test document x a certain category based on the categories k of closest neighbors from the training dataset. The similarity between the test document x and each of the closest neighbors is scored by the category to which the neighbor belongs. If several of k's closest neighbors belong to the same category, then the similarity score of that category for the test document x is calculated as the sum of the category points for each of these closest neighbors. After that, the categories are ranked by score, and the test document is assigned to the category with the highest score. The decision rule for *k*-NN can be written as follows (Formula 1):

$$f(x) = \arg\max Score(x, C_j) = \\ = \sum_{d_i \in kNN} sim(x, d_i) y(d_i, C_j), \quad (1)$$

This approach is effective, nonparametric and easy to implement. However, the classification time is very long, and accuracy is seriously impaired by the presence of noise training documents [9].

To improve the accuracy of the *k*-NN algorithm for text data classification in the study, a number of actions will be performed, such as:

1. Representation of documents/text as a vector space model, where each document/text is represented as a vector in an *n*-dimensional word space where each word is represented as coordinates. The more often a certain term appears in a document, the greater its significance in this document and the greater its coordinate in the vector representation of the document. Accordingly, this will speed up the work and classification of the *k*-NN method. The weight of each word in a document is calculated by weighing how often that word is used in the document and throughout the document collection. If the word is

DOI 10.15588/1607-3274-2023-3-9

used often in a document, but rarely in other documents, then its weight will be high.

2. Create a classification model based on clustering. For this, one-pass clustering algorithm with constraints should be used. This algorithm provides incremental clustering with time complexity close to linear.

3. During the clustering process, each cluster is represented as a cluster vector according to the centroid vector for each cluster [14, 20]. The change in word weight of each cluster is calculated by the formula 2:

$$w_{c_i^0}^{j+1}(t) = \frac{w_{C_i^0}^{j}(t) \times w(t)_p}{\left|C_i^0\right| + 1}. \quad (2)$$

According to the applied changes, the decision-making formula for *k*-NN will look like this (Formula 3):

$$f(x) = \arg\max ClusterScore(x, C_j) = \\ = \sum_{C_i^0 \in kNN} sim(x, C_i^0) y(C_i^0, C_j). \quad (3)$$

Suppose we have a training set of text documents with known classes that match their topicss. Each text document is represented as a feature vector $X = [x_1, x_2, ..., x_N]$, where x_i is a sign (for example, word, term) for the i-th document, and N is the total number of signs. Using a certain similarity metric, such as cosine similarity, we calculate the similarity between the feature vectors of two documents. Let $sim(x, y)$ denotes similarities between documents x and y. Accordingly, we find k documents from the training kit that have the greatest similarity with the new document. Denote these documents as $S = \{s_1, s_2, ..., s_k\}$, where s_i is the i-th closest neighbor. Hence we determine the class of the new document, by voting or by majority among k nearest neighbors. Let $C = \{c_1, c_2, ..., c_k\}$ – document classes $s_1, s_2, ..., s_k$. The class of the new document will be the class with the most votes among these k closest neighbors.

Thus, the definition of the class of a new document, based on the vote of the nearest neighbors, can be written as follows (Formula 4):

$$argmax_i(count(c_i, C)), i=1 \rightarrow k. \quad (4)$$

In the context of *k*-nearest neighbors (*k*-NN), hyperparameters are used to tune the algorithm itself, not to train a model with data. Hyperparameters determine the behavior of *k*-NN and its characteristics. The main hyperparameters include [17, 19]:

1. *k*: This is the main *k*-NN hyperparameter that determines the number of nearest neighbors to be used for decision-making.

2. Neighbor search algorithm: *k*-NN can use different neighbor search algorithms, such as “ball tree”, “kd tree” and others.

3. Distance or metric: *k*-NN uses distance or metric to determine how close points are to each other. For example, Euclidean distance, Manhattan distance, or cosine similarity.

In our context, we will apply them to analyze their impact on *k*-NN and, through them, try to improve the accuracy of the method.

The following Neighbor search algorithms are considered in the study:

1. The Ball Tree algorithm is one of the methods of constructing a data structure for the efficient execution of operations of the nearest neighbor in classification and clustering problems. It is based on the idea of partitioning a data space into minimally convex balls, known as “balls”. The basic principle of constructing a Ball Tree is to recursively partition data into subsets by calculating the center point (center of the “ball”) and the radius of the ball that best covers the data. The Ball Tree method allows you to quickly find *k*-nearest neighbors, reducing the number of comparisons between points and speeding up searches. It is especially useful for large data sets or when the distances between points have a large difference.

2. A *kd*-tree is a data structure used in the *k*-NN algorithm to quickly find the nearest neighbors. It divides a data space as a binary tree, where each node represents a point in space and divides that space into two subdomains. A key feature of the *kd*-tree is the way data is divided in space using hyperplanes parallel to the coordinate axes. The *kd*-tree significantly reduces the number of comparisons operations required to find the nearest neighbors, which makes it an effective method for *k*-NN. It is especially useful in problems with a large number of points in the data space.

The Minkow metric is a general term for a family of metrics that include Manhattan distance and Euclidean distance as partial cases of them. It is used to calculate the distance between two points in *n*-dimensional space. Formally, the Minkov metric is defined as follows for two points $P(p_1, p_2, ..., p_n)$ and $Q(q_1, q_2, ..., q_n)$ in *n*-dimensional space (Formula 5):

$$d = (|p_1 - q_1|^p + |p_2 - q_2|^p + ... + |p_n - q_n|^p). \quad (5)$$

In this formula, p is a parameter that controls the shape of the metric. Depending on the value of p, the Minkov metric can vary from Manhattan distance ($p = 1$) to Euclidean distance ($p = 2$). Using the p parameter, the Minkow metric can simulate various types of distances, including $L1$ distance (Manhattan), $L2$ distance (Euclidean), and others [16, 18].

The Minkowi metric is a popular choice in the *k*-NN algorithm. It allows you to determine the distance between objects and take into account their location in space. Depending on the type of data and the nature of the task, it is advantageous to use different values of the pa-

DOI 10.15588/1607-3274-2023-3-9

rameter p for optimal results. Minkowian metrics can be computationally efficient, especially for large datasets.

The choice of the Reuters dataset to classify text documents by their subject using *k*-nearest neighbors is the most optimal for several reasons:

– Variety of topicss: The Reuters-21578 dataset contains a wide range of topicss, covering news from various fields such as finance, technology, sports, politics, and others. This ensures representativeness and diversity of data, which is important for the classification of texts.

– Data volume: The Reuters-21578 dataset contains a significant number of documents on various topicss. Most classification algorithms, including *k*-NN, require enough data to achieve reliable results. Therefore, the presence of a large amount of data in the Reuters-21578 dataset makes it attractive for *k*-NN applications.

– The similarity of text documents: The *k*-NN algorithm is based on the hypothesis that similar data have similar classes. The Reuters-21578 dataset contains news articles that may have similar characteristics depending on the topics. This supports the *k*-NN hypothesis and contributes to its effectiveness in classifying these texts.

In total, the Reuters-21578 dataset contains about 21,578 news documents written in English. Each document includes a title, date, text table of contents, and category labels assigned to it (Table 1). The documents in the dataset are classified into 135 different categories, such as “foreign news”, “sports”, “politics”, etc. Each document can have many categories to which it belongs.

Table 1 – Data Set fields

Field name	Type	Description
Title	String	Contains text data represented as a character string.
Date	String	Represents the date of publication of the article.
Topicss	Sequence	Represents categories or topicss related to the article. It is stored as a set or list of lines, where each line represents a topics or category label.
Places	Sequence	Represents the geographical locations mentioned in the article. It is stored as a set or list of rows, where each row represents a place label.
People	Sequence	Contains the names of persons mentioned in the article. It is stored as a set or list of strings.
Orgs	Sequence	Contains the names of organizations mentioned in the article. It is stored as a set or list of strings.
Exchanges	Sequence	Represents mentions of stock exchanges or financial markets in an article. It is usually stored as a set or list of strings.
Text	String	Contains the main text of the news article and is presented as a string of characters.

Since the dataset Reuters-21578 may contain data or their type, which may worsen the results of the study, it must carry out preliminary processing of the data.

First, we need to balance the text and process it further. Some documents may contain symbols, punctuation, or numbers that do not carry essential information for text analysis. Data pre-processing allows you to remove these unnecessary elements and focus on essential aspects of the text. We also need to reduce everything to lowercase to ensure uniformity. Breaking the text into separate tokens or words is also an important step for further analysis. Tokenization helps to understand the structure of a text and divide it into separate units, which facilitates further processing and use. To improve your workout results, you need to remove stop words, which will reduce noise. Stop words are common words that do not carry essential information for text analysis, for example, “the”, “and”, “is”, etc. Another very important step is that it is necessary to reduce words to their basic form (lemmatization). This reduces the number of unique words in the text and makes it easier to recognize the semantic relationship between them. For example, given that we have texts in English, words like “running”, “run” and “ran” will be reduced to the lemma “run”.

Secondly, we need to convert the words’ text to vector format. Vectorization is the process of converting text data into numerical vectors that can be used to further analyze or train machine learning models. This is achieved using methods such as bag-of-words or TF-IDF (Term Frequency-Inverse Document Frequency) in our case, where each word or term is represented by a numeric value. This allows textual data to be treated as numerical features that can be used in machine learning models for classification, clustering, or other analysis.

Therefore, these steps will help prepare data from the Reuters-21578 dataset for further application of machine learning and text analysis models, since data preprocessing helps to improve data quality and representativeness, reduce noise, etc.

4 EXPERIMENTS

The next step is to study the effectiveness of the *k*-NN closest neighbors method for classifying text documents by their topics. Therefore, we will show in more detail the influence of the parameter *k* and the choice of distance metric and other factors.

To begin with, we display the number of articles belonging to a certain category (Fig. 1):

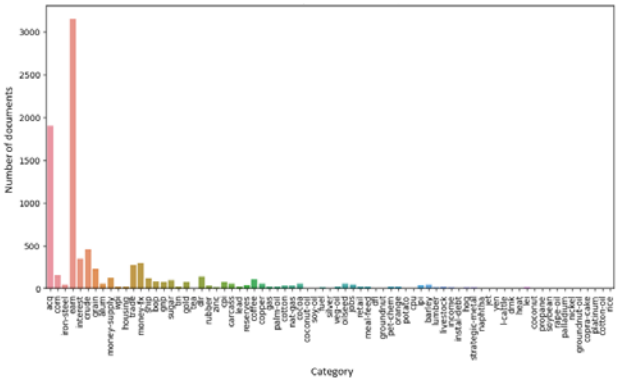

Figure 1 – A bar chart showing the categorization

We divide the experiments, the purpose of which is to improve the accuracy of classification, into:

1. Effect of parameter *k*: Experiments will be performed with different values of the parameter *k* (number

DOI 10.15588/1607-3274-2023-3-9

of neighbors) and the accuracy of the model on the test set will be measured.

2. Clustering Impact: Experiments will be conducted with the generated clustering-based classification model and its impact on the accuracy of the *k*-NN model on the test set.

3. Choosing a distance metric and *k*-NN hyperparameters: Experiments will be conducted with different distance metrics and with different *k*-NN hyperparameters, namely an algorithm such as "ball_tree" (partitioning the data space into minimally convex balls) and a *k*-distance tree and metrics (such as the Manhattan metric and Euclidean distance), and the accuracy will be measured on the test set.

To assess the quality of the classification model, we will use metrics. Here is a brief explanation of each metric:

1. Accuracy: Measures the ratio of the number of correctly classified documents to the total number of documents. The higher the value, the better the model.

2. Precision: Measures the ratio of the number of correctly positively classified documents to the total number of positively classified documents. This indicates how accurately the model identifies positive documents.

3. Recall: Measures the ratio of the number of correctly positively classified documents to the total number of documents belonging to the positive class (correctly classified positive documents plus false negative documents). This indicates how fully the model defines positive documents.

4. F1 score: This is the harmonic average between accuracy and completeness. It is used as a compromise metric that combines information about accuracy and completeness. It takes into account both the accuracy and completeness of the model and uses its harmonic average to calculate the final value.

To estimate the error of our model, we will build a histogram comparing real and predicted data, where count represents the number of cases or frequency of each category in the data set, and category respectively the category itself. We will also build a prediction matrix. The prediction matrix reflects the correspondence between actual and predicted classes and allows quantitative analysis of classification results. It consists of rows and columns, where the rows represent the actual classes, and the columns represent the provided classes. Each cell in the matrix shows the number of samples that belong to a certain actual class and have been mistakenly classified into a specific predicted class. For both options, we will take the top 15 values to see the result better.

These studies will allow us to understand under what parameters and characteristics *k*-NN shows itself best.

The purpose of the experiment №1 is to study the influence of the number of neighbors (parameter *k*) on the results of a particular operation or algorithm.

We will conduct a study on the parameters k = 1, 5, 10, 15, 20 where k is the number of neighbors. We calculate the assessment of the quality of the classification model at k = 1 (Fig. 2).

```
Accuracy:0.79935125115848
Precision:0.8054679150341048
Recall:0.79935125115848
F1 score:0.7980253922562135
```

Figure 2 – Evaluation of the quality of the classification model

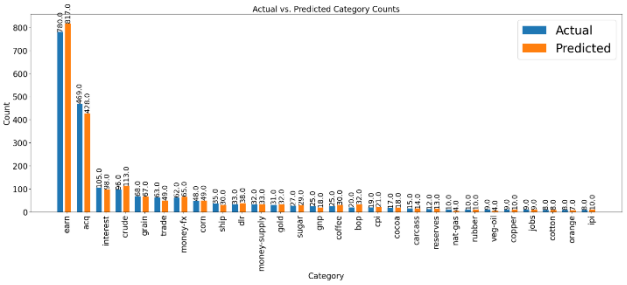

Figure 3 – Histogram comparison number of predicted and real data for *k*=1

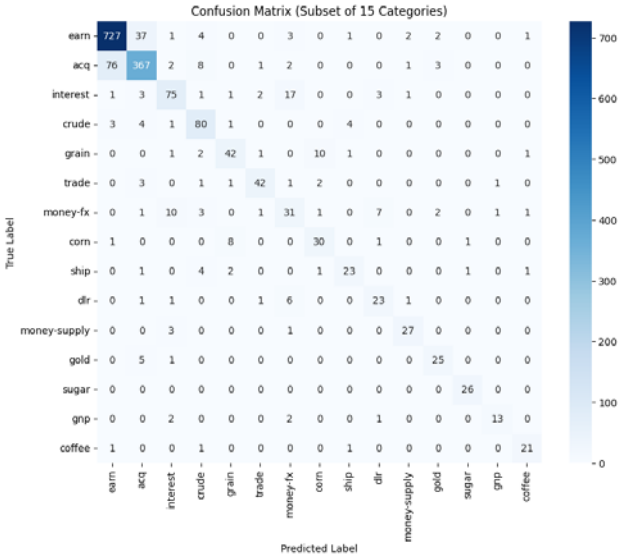

Figure 4 – Prediction matrix for *k*=1

As we can see from the Fig. 2, the results are satisfactory, but they should be improved. In Fig. 3 shows that the largest error belongs to the EARN category. From the matrix of predictions in Fig. 4 we can also see that most often the algorithm was wrong in determining the category of EARN and ACQ.

We calculate the assessment of the quality of the classification model at k = 5 (Fig. 5).

```
Accuracy:0.8178869323447636
Precision:0.8153930325993035
Recall:0.8178869323447636
F1 score:0.8101419178618491
```

Figure 5 – Evaluation of the quality of the classification model

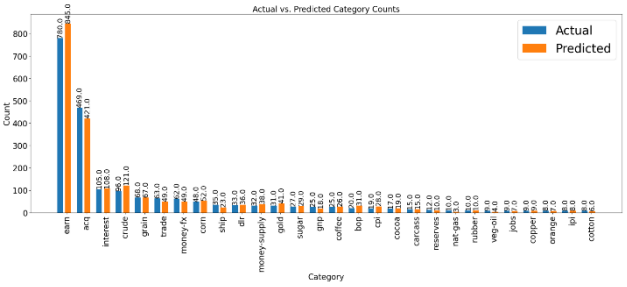

Figure 6 – Histogram comparison number of predicted and real data for *k*=5

DOI 10.15588/1607-3274-2023-3-9

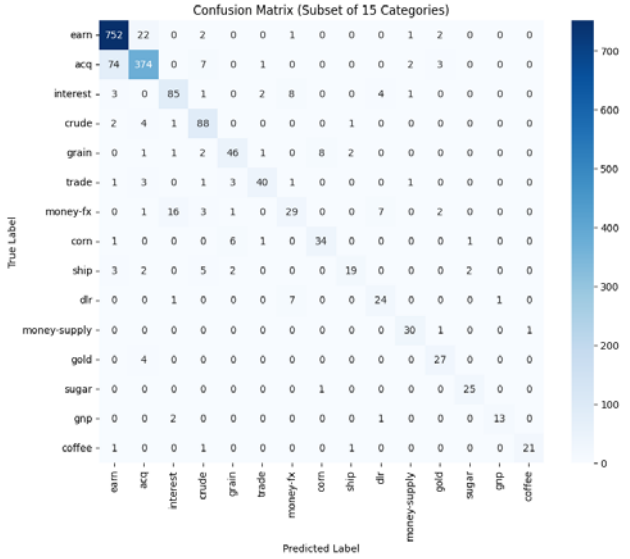

Figure 7 – Prediction matrix for k=5

As we can see from the Fig. 5, the results improved compared to the results in Fig. 2, but they should be improved. In Fig. 6 it can be seen that again the greatest error belongs to the prediction of data in EARN, but compared to the results achieved at k = 1, the error has increased. Following the matrix of predictions in Fig. 7 we can also see that most often the algorithm was wrong in determining the category EARN and ACQ, but for ACQ, it decreased compared to the results at k = 1. However, the number of correct distributions has increased.

We calculate the assessment of the quality of the classification model at k = 10 (Fig. 8).

```
Accuracy:0.8341056533827618
Precision:0.8317522368556206
Recall:0.8341056533827618
F1 score:0.8252637674684895
```

Figure 8 – Evaluation of the quality of the classification model

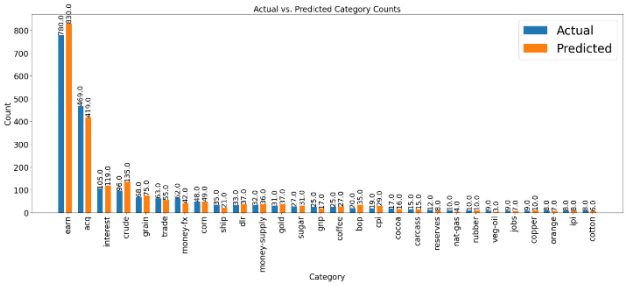

Figure 9 – Histogram comparison number of predicted and real data for k=10

As we can see from Fig. 8, the results improved compared to the results in Fig. 5 which makes them quite accurate. In Fig. 9 it can be seen that again the greatest error belongs to the prediction of data in EARN, but compared to the results achieved at k = 5, the error has decreased. Following the matrix of predictions in Fig. 10 we can also see that the algorithm was most often wrong in determining the categories of EARN and CRUDE, instead of ACQ. However, the number of correct distributions has increased.

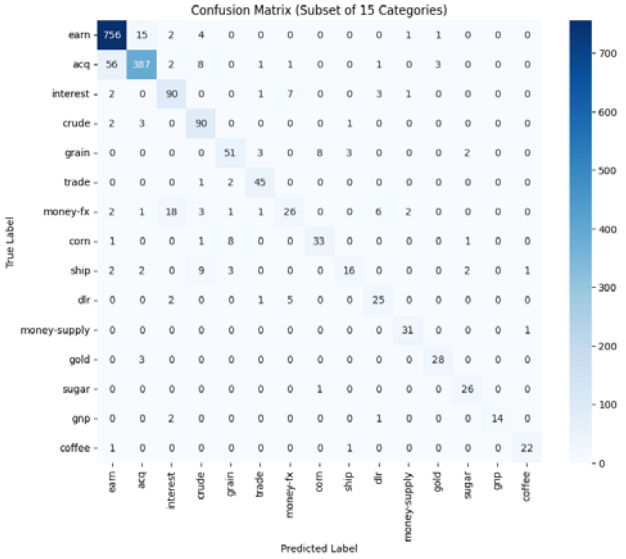

Figure 10 – Prediction matrix for k=10

We calculate the quality assessment of the classification model at k = 15 (Fig. 11).

```
Accuracy:0.8387395736793327
Precision:0.8352195686724123
Recall:0.8387395736793327
F1 score:0.8290827008957881
```

Figure 11 – Evaluation of the quality of the classification model

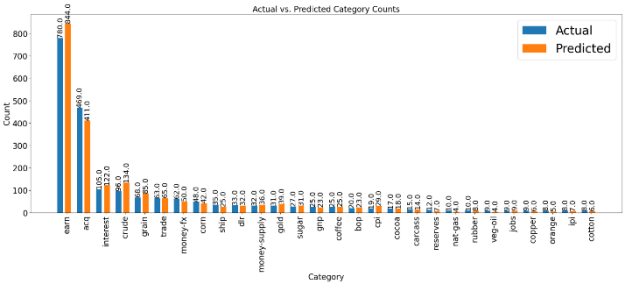

Figure 12 – Histogram comparison number of predicted and real data for k=15

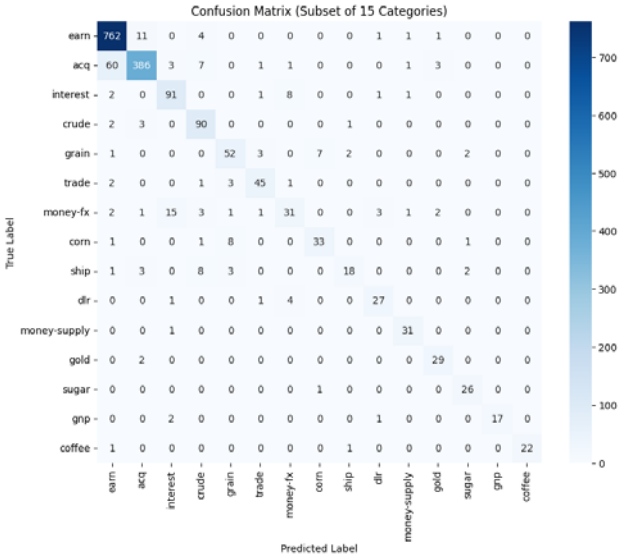

Figure 13 – Prediction matrix for k=15

DOI 10.15588/1607-3274-2023-3-9

As we can see from the Fig. 11, the results improved compared to the results in Fig. 8. In Fig. 12 it can be seen that again the greatest error belongs to the prediction of data in EARN, but compared to the results achieved at k = 10, the error has decreased. In accordance with the matrix of predictions in Fig. 13 we can also see that the algorithm was most often wrong in defining the EARN and CRUDE categories, instead of ACQ. The number of correct distributions has increased, although not significantly.

We calculate the assessment of the quality of the classification model at k = 20 (Fig. 14).

```
Accuracy:0.8387395736793327
Precision:0.8336147608484434
Recall:0.8387395736793327
F1 score:0.8289611340421243
```

Figure 14 – Evaluation of the quality of the classification model

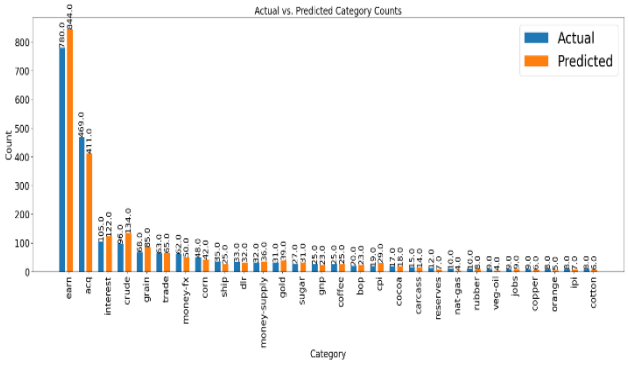

Figure 15 – Histogram comparison number of predicted and real data for k = 20

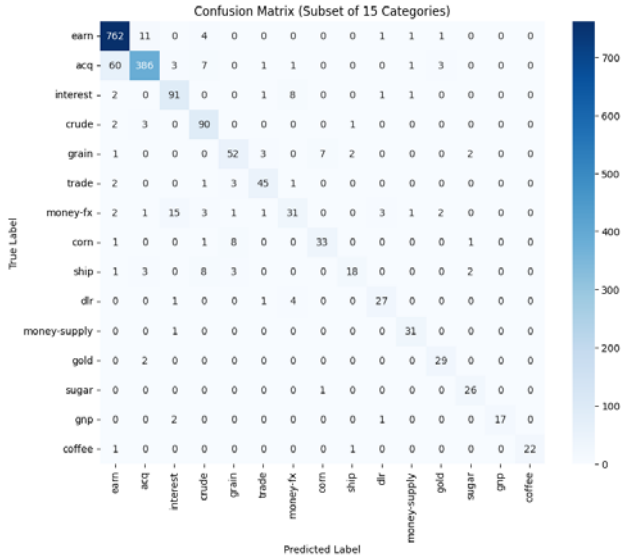

Figure 16 – Prediction matrix for k=20

As we can see from Fig. 14, the results remained the same in Fig. 11. This means that with a selected number of neighbors, the model has reached its maximum level of accuracy and it does not make sense to increase or decrease the number of neighbors. Accordingly, everything coincides with the results at k = 15.

The purpose of the experiment №2 is to study the effect of clustering on the algorithm k-NN. Clustering is a method of grouping similar objects into clusters based on their characteristics or distances to each other. In the context of k-NN, clustering can affect the results of an algorithm by changing the neighborhood of objects and hence determining their classification.

We will conduct a study on the parameters k = 5, 15, 20, c = 5, 15, 20, where k is the number of neighbors, and c is the number of clusters. We calculate the assessment of the quality of the classification model at k = 5 and c = 5 (Fig. 17).

```
Accuracy:0.8276181649675626
Precision:0.8236345856370154
Recall:0.8276181649675626
F1 score:0.8213364658704626
```

Figure 17 – Evaluation of the quality of the classification model

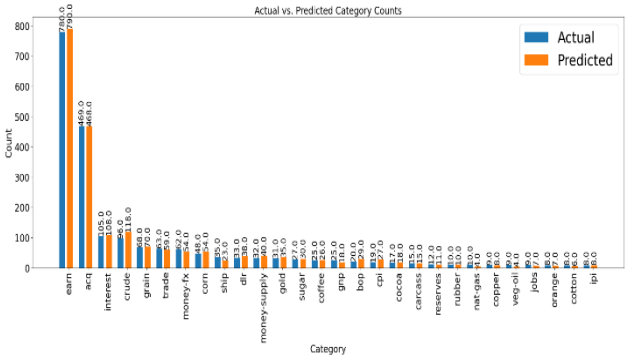

Figure 18 – Histogram comparing the number of predicted and real data for k=5 and c=5

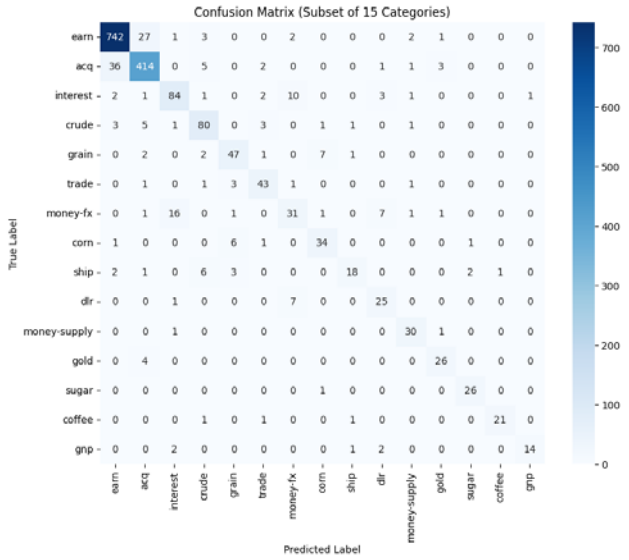

Figure 19 – Matrix of predictions for k = 5 and c = 5

As we can see from Fig. 17, the results are very good. If we compare the results of metrics with Fig. 5 and with Fig. 17, then with clustering there is a good increase in accuracy at k = 5. In Fig. 18 we can see that the attitude to categories has improved significantly with clustering than

DOI 10.15588/1607-3274-2023-3-9

without it, and the error has decreased accordingly. By the matrix of predictions in Fig. 19. We can also see that correct data allocation has improved a lot.

We calculate the assessment of the quality of the classification model at $k = 15$ and $c = 15$ (Fig. 20).

```
Accuracy:0.8419833178869324
Precision:0.836402372291527
Recall:0.8419833178869324
F1 score:0.8345115914215016
```

Figure 20 – Evaluation of the quality of the classification model

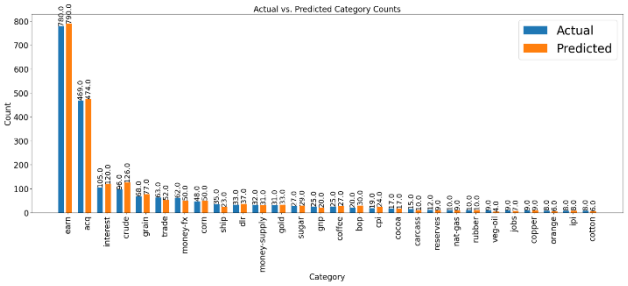

Figure 21 – Bar chart comparing the number of predicted and real data for k=15 and c=15

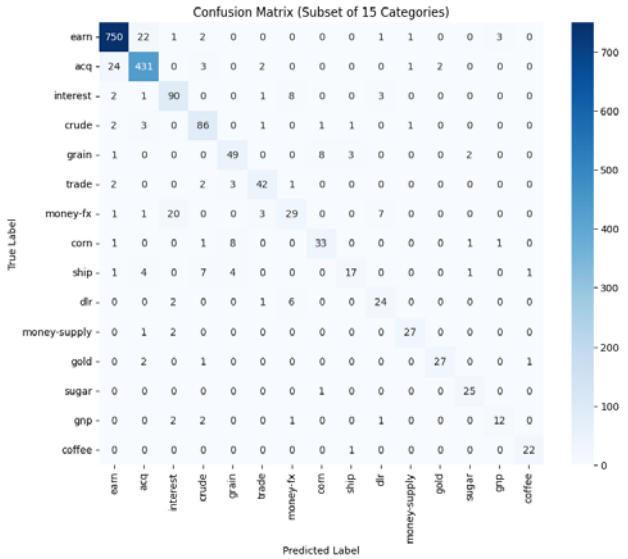

Figure 22 – Matrix of predictions for $k = 15$ and $c = 15$

As we can see from Fig. 20 results are very good. In Fig. 21 we can see that the attitude to categories has improved significantly with clustering than without it, and the predicted data almost coincides with the real data, which indicates a greater amount of properly distributed data. Following the matrix of predictions in Fig. 22 can also see that the correct distribution of data has improved a lot.

We calculate the quality assessment of the classification model at $k = 20$ and $c = 20$.

```
Accuracy:0.8456904541241891
Precision:0.8406501695176786
Recall:0.8456904541241891
F1 score:0.8374692034810864
```

Figure 23 – Assessment of the quality of the classification model

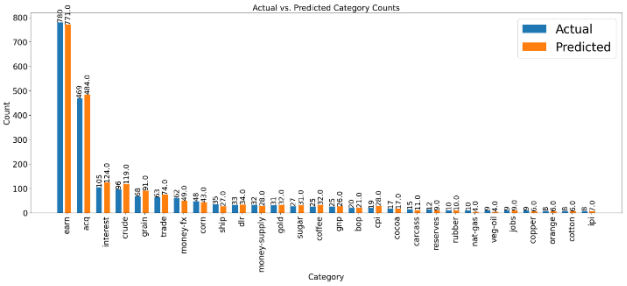

Figure 24 – Bar chart comparing the number of predicted and real data for k=20 and c=20

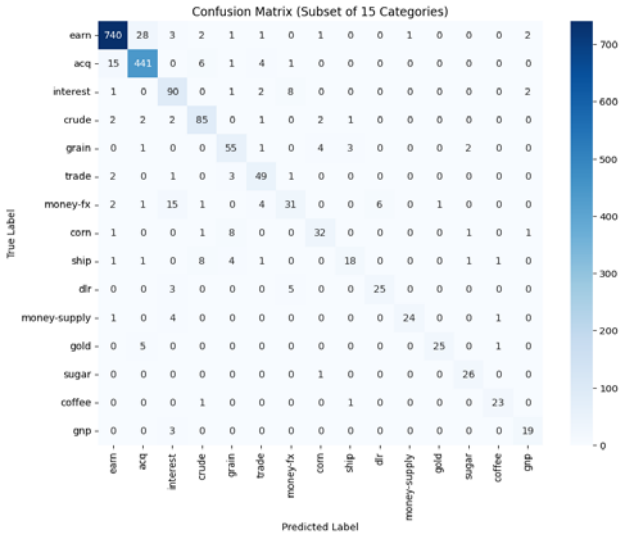

Figure 25 – Matrix of predictions for $k = 20$ and $c = 20$

The results, which are shown in Fig. 23–24 and in the matrix of predictions in Fig. 25, indicate very good results using clustering. From Fig. 23 we see that the accuracy of the model has again increased relative to $k = 15$ and $c = 15$. This means that clustering allows you to better determine the belonging of objects to the corresponding categories. In Fig. 24 shows that attitudes towards categories again improve significantly when clustering is used. Matrix of predictions in Fig. 25 also demonstrates that proper data allocation is greatly improved when clustering is used.

The experiment №3 aims to find hyperparameters that maximize the performance of the k-NN algorithm, as well as to understand the influence of hyperparameters on the classification results. The study can help identify which parameters are critical to achieving high accuracy and efficiency in a specific data context. We will use 15 neighbors for research.

We calculate the quality assessment of the classification model with the algorithm ‘ball_tree’ and the metric ‘manhattan’ at $k = 15$.

```
KNeighborsClassifier(algorithm='ball_tree', n_neighbors=15)
['earn' 'earn' 'earn' ... 'trade' 'cpi' 'crude']
Accuracy:0.8456904541241891
Precision:0.8406501695176786
Recall:0.8456904541241891
F1 score:0.8374692034810864
```

Figure 26 – Quality assessment of classification model with ‘ball_tree’ algorithm and ‘manhattan’ metric

DOI 10.15588/1607-3274-2023-3-9

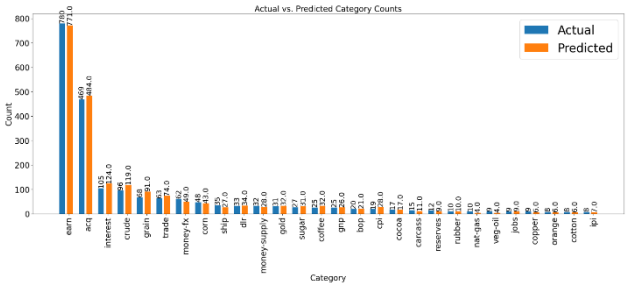

Figure 27 – Histogram comparing the number of predicted and real data with the ‘ball_tree’ algorithm and the ‘manhattan’ metric

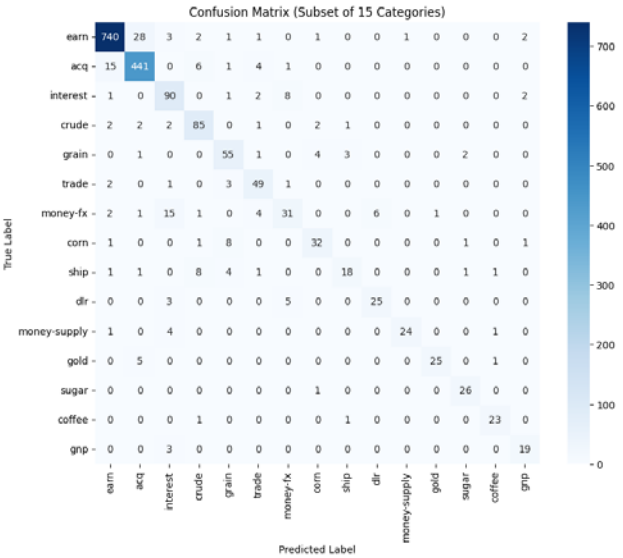

Figure 28 – Prediction Matrix with ‘ball_tree’ algorithm and ‘manhattan’ metric

As we can see from Fig. 26 results improved by comparing the usual k-NN with k = 15 in Fig. 11. In Fig. 27 we can see that, in general, the predicted data almost coincide with the real data, which indicates a more correct distribution of the data. Following the matrix of predictions in Fig. 28 We can also see that the correct distribution of data has improved a lot.

We calculate the quality assessment of the classification model with the algorithm ‘ball_tree’ and the metric ‘euclidean’ at k = 15.

```
Accuracy:0.8456904541241891
Precision:0.8406501695176786
Recall:0.8456904541241891
F1 score:0.8374692034810864
```

Figure 29 – Quality assessment of classification model with ‘ball_tree’ algorithm and ‘euclidean’ metric

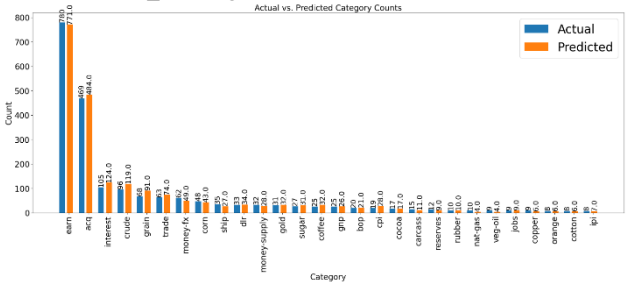

Figure 30 – Histogram comparing the number of predicted and real data with the ‘ball_tree’ algorithm and the ‘euclidean’ metric

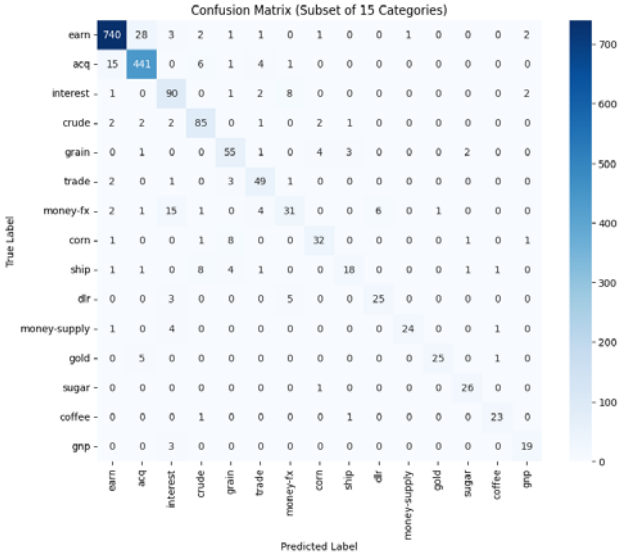

Figure 31 – Prediction matrix with ‘ball_tree’ algorithm and ‘euclidean’ metric

Comparing the results of the ‘ball_tree’ algorithm with different metrics, we can conclude that the introduction of these hyperparameters has increased the accuracy and efficiency of the model, but in two cases of using metrics, the result is the same, which may mean that both metrics measure the distance between two points with the same accuracy, or perhaps that the points are in a space where both metrics are equivalent.

We calculate the quality assessment of the classification model with the algorithm kd_tree’ and the metric ‘manhattan’ at k = 15.

```
KNeighborsClassifier(algorithm='kd_tree', n_neighbors=15)
['earn' 'earn' 'earn' ... 'palm-oil' 'cpi' 'crude']
Accuracy:0.8387395736793327
Precision:0.8352195686724123
Recall:0.8387395736793327
F1 score:0.8290827008957881
```

Figure 32 – Quality assessment of classification model with ‘kd_tree’ algorithm and ‘manhattan’ metric

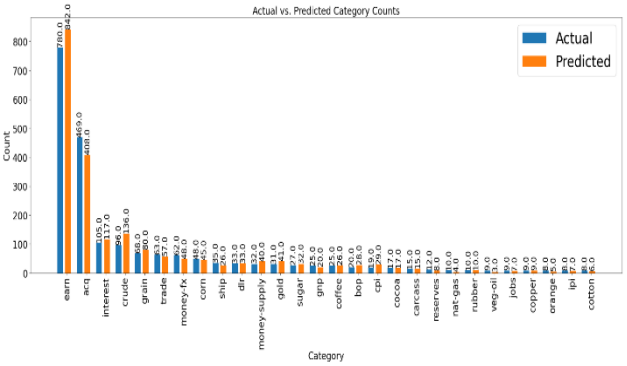

Figure 33 – Histogram comparing the number of predicted and real data with the ‘kd_tree’ algorithm and the ‘manhattan’ metric

DOI 10.15588/1607-3274-2023-3-9

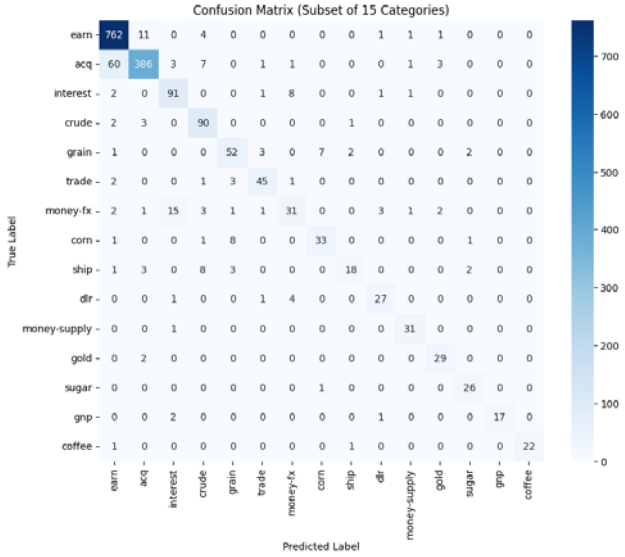

Figure 34 – Prediction matrix with ‘kd_tree’ algorithm and ‘manhattan’ metric

As you can see from Fig. 32–34, the application of ‘kd_tree’ with the ‘manhattan’ metric did not live up to expectations, since the results coincide with the use of the *k*-NN model without parameters at $k = 15$.

We calculate the quality assessment of the classification model with the algorithm kd_tree’ and the metric ‘euclidean’ at $k = 15$.

```
KNeighborsClassifier(algorithm='kd_tree', metric='euclidean', n_neighbors=15)
['earn' 'earn' 'earn' ... 'palm-oil' 'cpi' 'crude']
Accuracy:0.8387395736793327
Precision:0.8352195686724123
Recall:0.8387395736793327
F1 score:0.8290827008957881
```

Figure 35 – Evaluation of the quality of the classification model with the algorithm ‘kd_tree’ and the metric ‘euclidean’

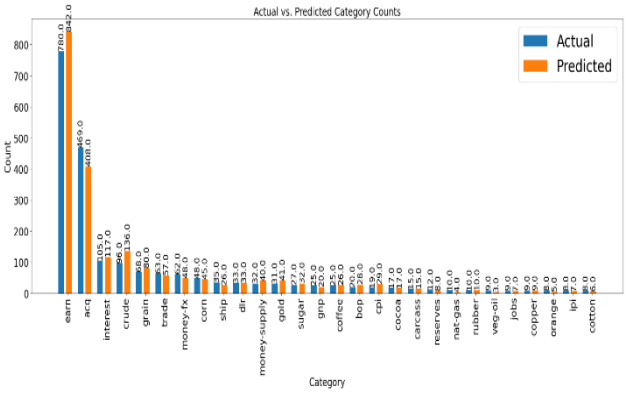

Figure 36 – Histogram comparing the number of predicted and real data with the ‘kd_tree’ algorithm and the ‘euclidean’ metric

As you can see from Fig. 35–37, the application of ‘kd_tree’ with the ‘euclidean’ metric as well as with ‘manhattan’ did not live up to expectations, since the results coincide with the use of the *k*-NN model without parameters at $k = 15$.

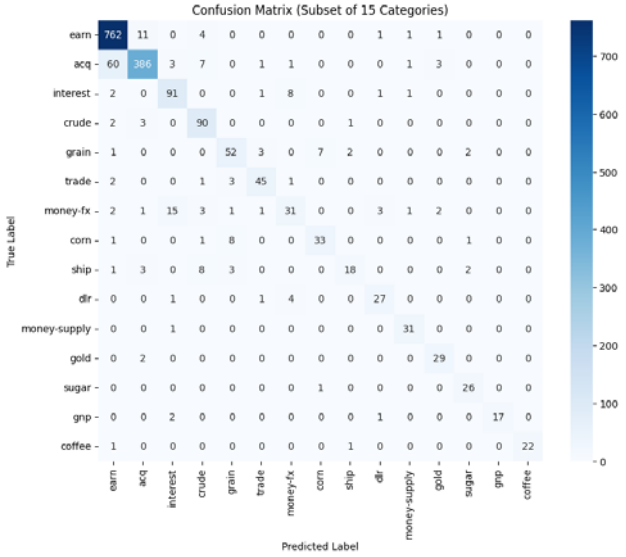

Figure 37 – Histogram comparing the number of predicted and real data with the ‘kd_tree’ algorithm and the ‘euclidean’ metric

5 RESULTS

In the previous section, the *k*-NN method for classifying text documents was implemented. Experiments have been conducted with various methods affecting the efficiency of *k*-NN, such as algorithm selection and metrics. Based on this, you can summarize the results.

In the first experiment, the results showed that the selected number of neighbors (parameter *k*) has a significant impact on the accuracy of classification. The best value of *k* is 15 and it achieves the maximum accuracy of classification of text documents, namely – 0.8387, which is equal to 83.87%. This means that increasing the number of neighbors does not significantly affect the results and does not bring additional improvements. This can be explained by the fact that too small *k* values can lose information, and too large *k* values can lead to overtraining of the model.

Therefore, for the classification of text documents, it is important to choose the optimal value of the parameter *k*, which provides the highest accuracy of classification. In this case, the use of $k = 15$ led to the maximum accuracy of classification of text documents, while at $k = 1$ the worst results are achieved, where the accuracy is – 0.7980 (79.80%).

In the second experiment, the use of clustering showed very good results for the classification of text documents. Maximum accuracy was achieved at $k = 20$ and is equal to 0.8457 (84.57%). Comparing the results when using the clusterless model *k*-NN in Fig. 4.13 and clustered in Fig. 4.22. With a value of $k = 20$, you can see how the accuracy and efficiency of classifying text documents have changed by better-determining similarities between documents and assigning them to appropriate categories.

The use of clustering changes the neighborhood of objects in space, that is, objects belonging to the same cluster become adjacent to each other. This allows the model to better distinguish objects from different categories

DOI 10.15588/1607-3274-2023-3-9

since internal similarities in clusters can be more pronounced than general similarities between all objects. Accordingly, the effectiveness and importance of this can be seen by comparing the results of models with and without clustering.

In the third experiment, which investigated the influence of hyperparameters on the *k*-NN algorithm, it was found that the choice of different algorithms and metrics can have a significant impact on the accuracy and efficiency of the model. This is achieved because different algorithms and metrics use different approaches to calculating distances between objects and determining their neighborhood. The choice of an algorithm, such as the Ball tree or *kd*-tree, affects the structure of the tree used to organize the data. In our case, the Ball tree works better, because when applied with Manhattan and Euclidean distance metrics, the maximum accuracy that has been achieved is 0.8457 (84.57%), which is on par with maximum accuracy when using clustering. Therefore, we can conclude that for datasets with a large number of features, the Ball tree works better. At this time, the *kd*-tree may be more efficient for datasets with fewer features, so in our case it was not very efficient and was able to achieve – 0.8387 (83.87%), which is not a bad result, but not very good either.

6 DISCUSSION

In our case, when using two different algorithms to find neighbors *k*-NN, metrics such as Manhattan and Euclidean distance were used. However, as can be seen from the experiments, there was no difference between them. This is because both metrics measure the distance between two points with the same precision, or the points may be in space where both metrics are equivalent, resulting in the fact that they give the same results.

So, summing up, in this case, the use of the *k*-NN method for the classification of text documents showed good results. *k*-NN takes into account the context of text documents using immediate neighbors and does not require complex data assumptions. These advantages make the *k*-NN method an attractive option for classifying text documents. However, to maximize classification accuracy, certain improvements need to be applied, such as choosing the optimal value of the *k* parameter, applying clustering, and using appropriate algorithms and metrics to improve the accuracy and efficiency of the model in our study. Based on experiments, the maximum results were shown by models *k*-NN at $k = 20$ with clustering and with hyperparameters, as an algorithm for finding neighbors – Ball tree at $k = 15$. At the same time, *k*-NN takes into account the context of text documents using immediate neighbors and does not require complex data assumptions. These advantages make the *k*-NN method an attractive option for classifying text documents.

However, to achieve maximum accuracy of classification, certain improvements must be applied. Using the optimal value of the *k* parameter, clustering, and selecting appropriate algorithms and metrics can significantly improve the quality and efficiency of the *k*-NN model.

In general, these experiments showed good results and confirmed the suitability of the *k*-NN nearest neighboring method for classifying text documents. In further research, it is recommended to pay attention to improving the model by optimizing parameters and using more complex algorithms to improve its efficiency.

So, summarizing all of the above, the nearest neighbors method is a very good method for classifying text documents. Given that the algorithm does not take much time, is flexible and is quite accurate, this makes it one of the best options for the classification task.

CONCLUSIONS

In this study, analysis and experiments were conducted using the *k*-NN nearest neighbor method to classify text documents. The results showed that the use of the *k*-NN method proved to be effective and a very good option for classifying text documents. The use of nearest neighbors allows the *k*-NN method to take into account the context of text documents and does not require complex data guesses. This makes it a flexible and versatile approach to classification.

In the case of researching the use of the nearest neighbors method *k*-NN to classify text documents by their topics, the **scientific novelty lies** in the fact that it offers the use of a method that is quite simple and effective to solve the complex problem of classifying text documents by their topics. The study proposes the use of clustering and dimensionality reduction to improve the quality of text classification. In addition, the study compares the efficiency of different types of term oscillation and different values of *k* in the *k*-NN method for classifying text documents. Thus, the study expands our understanding of how the nearest neighbors *k*-NN method can be applied to classify text documents by their topics and helps to improve methods for classifying texts.

The practical significance of the obtained results lies in a general review of the method of the nearest neighbors *k*-NN and the creation of a software solution for classifying text documents by their topics using this method. After all, the developed system can automatically classify text documents by their topics. It reduces classification errors, thereby improving the accuracy of the system. The study proposes developed software that implements the proposed indicators, and also experiments were conducted to study their properties. The results of the experiment allow for recommending the proposed indicators for use in practice, as well as determining the effective conditions for applying the proposed indicators.

Prospects for further research are to study the proposed algorithms for a wide class of practical problems.

ACKNOWLEDGEMENTS

The study was created within the framework of the project financed by the National Research Fund of Ukraine, registered No 30/0103 from 01.05.2023, “Methods and means of researching markers of aging and their influence on post-aging effects for prolonging the working period”, which is carried out at the Department of

DOI 10.15588/1607-3274-2023-3-9

Artificial Intelligence Systems of the Institute of Computer Sciences and Information of technologies of the National University "Lviv Polytechnic".

REFERENCES

1. Tung A. K., Hou J., Han J. Spatial clustering in the presence of obstacles, *The 17th Intern. conf. on data engineering (ICDE'01)*. Heidelberg, 2001, pp. 359–367. DOI: 10.1109/ICDM.2002.1184042
2. Boehm C., Kailing K., Kriegel H., Kroeger P. Density connected clustering with local subspace preferences, *IEEE Computer Society. Proc. of the 4th IEEE Intern. conf. on data mining. Los Alamitos*, 2004, pp. 27–34. DOI: 10.1007/978-0-387-39940-9_605
3. Boyko N., Kmetyk-Podubinska K., Andrusiak I. Application of Ensemble Methods of Strengthening in Search of Legal Information, *Lecture Notes on Data Engineering and Communications Technologies*, 2021, Vol. 77, pp. 188–200. https://doi.org/10.1007/978-3-030-82014-5_13.
4. Boyko N., Hetman S., Kots I. Comparison of Clustering Algorithms for Revenue and Cost Analysis, *Proceedings of the 5th International Conference on Computational Linguistics and Intelligent Systems (COLINS 2021)*. Lviv, Ukraine, 2021, Vol. 1, pp. 1866–1877.
5. Procopiuc C. M., Jones M., Agarwal P. K., Murali T. M. A Monte Carlo algorithm for fast projective clustering, *ACM SIGMOD Intern. conf. on management of data*. Madison, Wisconsin, USA, 2002, pp. 418–427.
6. Sharma A., J. Nirmal Kumar S, Rana D., Setia S. A Review On Collaborative Filtering Using Knn Algorithm, *OPJU International Technology Conference on Emerging Technologies for Sustainable Development (OTCON)*, 2023, pp. 1–6. DOI: 10.1109/OTCON56053.2023.10113985
7. Faye G. C. Gamboa, Matthew B. Concepcion, Antolin J. Alipio, Dan Michael A. Cortez, Andrew G. Bitancor, Myra S.J. Santos, Francis Arlando L. Atienza, Mark Anthony S. Mercado Further Enhancement of KNN Algorithm Based on Clustering Applied to IT Support Ticket Routing, 3rd International Conference on Computing, Networks and Internet of Things (CNIOT), 2022, pp. 186–190. DOI: 10.1109/CNIOT55862.2022.00040
8. Yang J.-K., Huang K.-Ch., Chung Ch.-Y., Chen Yu-Chi, Wu T.-W. Efficient Privacy Preserving Nearest Neighboring Classification from Tree Structures and Secret Sharing, *IEEE International Conference on Communications,* 2022, pp. 5615–5620. DOI: 10.1109/ICC45855.2022.9838718
9. Zhang Yu., Zhou Y., Xiao M., Shang X. Comment Text Grading for Chinese Graduate Academic Dissertation Using Attention Convolutional Neural Networks, *7th International Conference on Systems and Informatics (ICSAI),* 2021, pp. 1–6. DOI: 10.1109/ICSAI53574.2021.9664159
10. Rohwinasakti S., Irawan B., Setianingsih C. Sentiment Analysis on Online Transportation Service Products Using K-Nearest Neighbor Method, *International Conference on Computer, Information and Telecommunication Systems (CITS),* 2021, pp. 1–6.
11. Javid J., Ali Mughal M., Karim M. Using kNN Algorithm for classification of Distribution transformers Health index, *International Conference on Innovative Computing (ICIC),* 2021, pp. 1–6. DOI: 10.1109/ICIC53490.2021.9693013
12. Bansal A., Jain A. Analysis of Focussed Under-Sampling Techniques with Machine Learning Classifiers, *IEEE/ACIS 19th International Conference on Software Engineering Research, Management and Applications (SERA),* 2021, pp. 91–96. DOI: 10.1109/SERA51205.2021.9509270
13. Bellad Sagar. C., Mahapatra A., Ghule S. Dilip, Shetty S. Sridhar, Sountharrajan S, Karthiga M, Suganya Prostate Cancer Prognosis-a comparative approach using Machine Learning Techniques, *5th International Conference on Intelligent Computing and Control Systems (ICICCS),* 2021, pp. 1722–1728. DOI: 10.1109/ICICCS51141.2021.9432173
14. Pokharkar Swapnil R., Wagh Sanjeev J., Deshmukh Sachin N. Machine Learning Based Predictive Mechanism for Internet Bandwidth, *6th International Conference for Convergence in Technology (I2CT),* 2021, pp. 1–4. DOI: 10.1109/I2CT51068.2021.9418164
15. Chaudhary K., Poirion O. B., Lu L., Garmire L. X. Deep learning based multi-omics integration robustly predicts survival in liver cancer, *Clin. Can. Res.*, 2017, 0853, pp. 1246–1259. doi: 10.1101/114892
16. Cheng B., Liu M., Zhang D., Musell B. C., Shen D. Domain Transfer Learning for MCI Conversion Prediction, *IEEE Trans. Biomed. Eng.*, 2015, Vol. 62 (7), pp. 1805–1817. doi: 10.1109/TBME.2015.2404809
17. Huang M., Yang W., Feng Q., Chen W., Weiner M. W., Aisen P. Longitudinal measurement and hierarchical classification framework for the prediction of Alzheimer's disease, *Sci. Rep.*, 2017, Vol. 7, P. 39880. doi: 10.1038/srep39880
18. Hossain M. Z., Akhtar M. N., Ahmad R. B., Rahman M. A dynamic K-means clustering for data mining, *Indonesian Journal of Electrical Engineering and Computer Science*, 2017, Vol. 13 (2), pp. 521–526. DOI: http://doi.org/10.11591/ijeecs.v13.i2.pp521–526
19. Jothi R., Mohanty S. K., Ojha A. DK-means: a deterministic *k*-means clustering algorithm for gene expression analysis, *Pattern Analysis and Applications*, 2019, Vol. 22(2), pp. 649–667. DOI: 10.1007/s10044-017-0673-0
20. Polyakova M. V., Krylov V. N. Data normalization methods toimprovethe quality ofclassificationin the breast cancerdiagnostic system, *Applied Aspects of Information Technology,* 2022, Vol. 5(1), pp. 55–63. DOI: https://doi.org/10.15276/aait.05.2022.5

Received 15.05.2023.
Accepted 20.08.2023.

УДК 004.021

МЕТОД *k* НАЙБЛИЖЧИХ СУСІДІВ ДЛЯ КЛАСИФІКАЦІЇ ТЕКСТОВИХ ДОКУМЕНТІВ ЗА ЇХ ТЕМАТИКОЮ

Бойко Н. І. – канд. економ. наук, доцент, доцент кафедри Систем штучного інтелекту, Національний університет «Львівська політехніка», Львів, Україна.

Михайлишин В. Ю. – асистент кафедри Системи штучного інтелекту, Національний університет «Львівська політехніка», Львів, Україна.

АНОТАЦІЯ

Актуальність. Оптимізація методу найближчих сусідів *k*-NN для класифікації текстових документів за їх темою, а також розв'язок задачі на основі методу експериментальним шляхом.

Мета роботи є вивчення методу найближчих сусідів *k*-NN для класифікації текстових документів за їх темою. Завданням дослідження є на основі набору даних провести класифікацію текстових документів за їх темою за оптимальний час та з високою точністю.

Метод. Метод *k*-найближчих сусідів – це метричний алгоритм для автоматичної класифікації об'єктів або регресії. Алгоритм *k*-NN зберігає всі наявні дані та класифікує нову точку на основі відстані між новою точкою та всіма точками в нав-

DOI 10.15588/1607-3274-2023-3-9

чальному наборі. Для цього використовується певна метрика відстані, така як Евклідова відстань. У процесі навчання *k*-NN зберігає всі дані з навчального набору, тому він відноситься до «ледачих» алгоритмів, оскільки навчання відбувається в момент класифікації. Алгоритм не робить ніяких припущень про розподіл даних та він є непараметричним. Завдання алгоритму *k*-NN полягає в тому, щоб призначити тестовому документу *x* певну категорію на основі категорій *k* найближчих сусідів з навчального набору даних. Схожість між тестовим документом *x* та кожним з найближчих сусідів оцінюється балом категорії, до якої належить сусід. Якщо декілька з *k* найближчих сусідів належать до однієї категорії, то бал схожості цієї категорії для тестового документа *x* обчислюється як сума балів категорії для кожного з цих найближчих сусідів. Після цього, категорії ранжуються за балами, і тестовий документ призначається категорії з найвищим балом.

Результати. Успішно реалізовано метод *k*-NN для класифікації текстових документів. Було проведено експерименти з різними методами, що впливають на ефективність *k*-NN, такими як вибір алгоритму та метрики. Результати експериментів показали, що використання певних методів може покращити точність класифікації та ефективність моделі.

Висновки. Відображення результатів на різних метриках та алгоритмах показало, що вибір конкретного алгоритму та метрики може мати значний вплив на точність передбачень. Застосування алгоритму ball tree, а також використання різних метрик, таких як манхетівська або євклідова відстань, може призвести до покращення результатів. Використання кластеризації перед застосуванням *k*-NN показало позитивний вплив на результати та дозволяє краще групувати дані і зменшує вплив шуму або неправильно класифікованих точок, що призводить до покращення точності та розподілу класів.

КЛЮЧОВІ СЛОВА: метод, кластер, класифікація, текстовий документ, тема, алгоритм ball tree, метрика.

ЛІТЕРАТУРА

1. Tung A. K. Spatial clustering in the presence of obstacles / A. K. Tung, J. Hou, J. Han // The 17th Intern. conf. on data engineering (ICDE'01), Heidelberg. – 2001. – P. 359–367. DOI: 10.1109/ICDM.2002.1184042
2. Density connected clustering with local subspace preferences / [C. Boehm, K. Kailing, H. Kriegel, P. Kroeger] // IEEE Computer Society. Proc. of the 4th IEEE Intern. conf. on data mining. Los Alamitos. – 2004. – P. 27–34. DOI: 10.1007/978-0-387-39940-9_605
3. Boyko N. Application of Ensemble Methods of Strengthening in Search of Legal Information / N. Boyko, K. Kmetyk-Podubinska, I. Andrusiak // Lecture Notes on Data Engineering and Communications Technologies. – 2021. – Vol. 77. – P. 188–200. https://doi.org/10.1007/978-3-030-82014-5_13.
4. Boyko N. Comparison of Clustering Algorithms for Revenue and Cost Analysis / N. Boyko, S. Hetman, I. Kots // Proceedings of the 5th International Conference on Computational Linguistics and Intelligent Systems (COLINS 2021). Lviv, Ukraine. – 2021. – Vol. 1. – P. 1866–1877.
5. A Monte Carlo algorithm for fast projective clustering / [C. M. Procopiuc, M. Jones, P. K. Agarwal, T. M. Murali] // ACM SIGMOD Intern. conf. on management of data, Madison, Wisconsin, USA. – 2002. – P. 418–427.
6. A Review On Collaborative Filtering Using Knn Algorithm / [A. Sharma, J. Nirmal Kumar S, D. Rana, S. Setia] // OPJU International Technology Conference on Emerging Technologies for Sustainable Development (OTCON). – 2023. – P. 1–6. DOI: 10.1109/OTCON56053.2023.10113985
7. Gamboa Further Enhancement of KNN Algorithm Based on Clustering Applied to IT Support Ticket Routing / [C. Faye G. Gamboa, Matthew B. Concepcion, Antolin J. Alipio et al.] // 3rd International Conference on Computing, Networks and Internet of Things (CNIOT). – 2022. – P.186–190. DOI: 10.1109/CNIOT55862.2022.00040
8. Efficient Privacy Preserving Nearest Neighboring Classification from Tree Structures and Secret Sharing / [J.-K. Yang, K.-Ch. Huang, Ch.-Y. Chung et al.] // IEEE International Conference on Communications. – 2022. – P. 5615–5620. DOI: 10.1109/ICC45855.2022.9838718
9. Zhang Yu. Comment Text Grading for Chinese Graduate Academic Dissertation Using Attention Convolutional Neural Networks / [Y. Zhang, Y. Zhou, M. Xiao, X. Shang] // 7th International Conference on Systems and Informatics (ICSAI). – 2021. – P. 1–6. DOI: 10.1109/ICSAI53574.2021.9664159
10. Rohwinasakti S. Sentiment Analysis on Online Transportation Service Products Using *K*-Nearest Neighbor Method / S. Rohwinasakti, B. Irawan, C. Setianingsih // International Conference on Computer, Information and Telecommunication Systems (CITS). – 2021. – P.1–6.
11. Javid J. Using kNN Algorithm for classification of Distribution transformers Health index / J. Javid, M. Ali Mughal, M. Karim // International Conference on Innovative Computing (ICIC). – 2021. – P. 1–6. DOI: 10.1109/ICIC53490.2021.9693013
12. Bansal A. Analysis of Focussed Under-Sampling Techniques with Machine Learning Classifiers / A. Bansal, A. Jain // IEEE/ACIS 19th International Conference on Software Engineering Research, Management and Applications (SERA). – 2021. – P. 91–96. DOI: 10.1109/SERA51205.2021.9509270
13. Bellad Sagar.C. Prostate Cancer Prognosis-a comparative approach using Machine Learning Techniques / [Sagar.C. Bellad, A. Mahapatra, S. Dilip Ghule et al.] // 5th International Conference on Intelligent Computing and Control Systems (ICICCS). – 2021. – P. 1722–1728. DOI: 10.1109/ICICCS51141.2021.9432173
14. Pokharkar Swapnil R. Machine Learning Based Predictive Mechanism for Internet Bandwidth / Swapnil R. Pokharkar, Sanjeev J. Wagh, Sachin N. Deshmukh // 6th International Conference for Convergence in Technology (I2CT). – 2021. – P.1–4. DOI: 10.1109/I2CT51068.2021.9418164
15. Deep learning based multi-omics integration robustly predicts survival in liver cancer / [K. Chaudhary, O. B. Poirion, L. Lu, L. X. Garmire] // Clin. Can. Res. – 2017. – 0853. – P. 1246–1259. doi: 10.1101/114892
16. Domain Transfer Learning for MCI Conversion Prediction / [B. Cheng, M. Liu, D. Zhang et al.] // IEEE Trans. Biomed. Eng. – 2015. – Vol. 62 (7). – P. 1805–1817. doi: 10.1109/TBME.2015.2404809
17. Longitudinal measurement and hierarchical classification framework for the prediction of Alzheimer's disease / [M. Huang, W. Yang, Q. Feng et al.] // Sci. Rep. – 2017. – Vol. 7. – P. 39880. doi: 10.1038/srep39880
18. A dynamic K-means clustering for data mining / [M. Z. Hossain, M. N. Akhtar, R. B. Ahmad, M. Rahman] // Indonesian Journal of Electrical Engineering and Computer Science. – 2017. – Vol. 13 (2). – P. 521–526. DOI: http://doi.org/10.11591/ijeecs.v13.i2.pp521–526
19. Jothi R. DK-means: a deterministic k-means clustering algorithm for gene expression analysis / R. Jothi, S. K. Mohanty, A. Ojha // Pattern Analysis and Applications. – 2019. – Vol. 22(2). – P. 649–667. DOI: 10.1007/s10044-017-0673-0
20. Polyakova M. V. Data normalization methods toimprovethe quality ofclassificationin the breast cancerdiagnostic system / M. V. Polyakova, V. N. Krylov // Applied Aspects of Information Technology. – 2022. – Vol. 5(1). – P. 55–63. DOI: https://doi.org/10.15276/aait.05.2022.5

DOI 10.15588/1607-3274-2023-3-9