DOI: https://doi.org/10.15276/aait.05.2022.11
UDC 004.832.2

Methods of analysis of multimodal data to increase the accuracy of classification

Nataliya I. Boyko[1)]
ORCID: https://orcid.org/0000-0002-6962-9363; nataliya.i.boyko@lpnu.ua. Scopus Author ID: 57191967462
Mykhaylo V. Muzyka[1)]
ORCID: https://orcid.org/0000-0001-8285-1631; muzyka.m.00@gmail.com
[1)] Lviv Polytechnic National University, 1, Profesorska Str. Lviv, 79013, Ukraine

ABSTRACT

This paper proposes methods for analyzing multimodal data that will help improve the overall accuracy of the results and plans for classifying K-Nearest Neighbor (KNN) to minimize their risk. The mechanism of increasing the accuracy of KNN classification is considered. The research methods used in this work are comparison, analysis, induction, and experiment. This work aimed to improve the accuracy of KNN classification by comparing existing algorithms and applying new methods. Many literary and media sources on the classification according to the algorithm k of the nearest neighbors were analyzed, and the most exciting variations of the given algorithm were selected. Emphasis will be placed on achieving maximum classification accuracy by comparing existing and improving methods for choosing the number k and finding the nearest class. Algorithms with and without data analysis and pre-processing are also compared. All the strategies discussed in this article will be achieved purely practically. An experimental classification by k nearest neighbors with different variations was performed. Data for the experiment used two different data sets of various sizes. Different classifications k and the test sample size were taken as classification arguments. The paper studies three variants of the algorithm k nearest neighbors: the classical KNN, KNN with the lowest average and hybrid KNN. These algorithms are compared for different test sample sizes for other numbers k. The article analyzes the data before classification. As for selecting the number k, no simple method would give the maximum result with great accuracy. The essence of the algorithm is to find k closest to the sample of objects already classified by predefined and numbered classes. Then, among these k objects, you need to count how often the class occurs and assign the most common class to the selected object. If two classes' occurrences are the largest and the same, the class with the smaller number is assigned.

Keywords: Method; algorithm; analysis; machine learning; multimodal data; classification; K-Nearest Neighbor.

***For citation*: Boyko N. I., Muzyka M. V. "Methods of analysis of multimodal data to increase the accuracy of classification." *Applied Aspects of Information Technology.* 2022; Vol. 5 No. 2: 147–160. DOI: https://doi.org/10.15276/aait.05.2021.11.**

INTRODUCTION

I. The classification issue is rising not only in Data Science but also unconsciously in real life. When we meet a foreigner on the street, we can classify him as specific nationality considering his clothes, language, habits etc. We will take a particular stereotypical representative of a nation as a basis. As in machine learning, we classify objects according to their features into predefined classes. One classification method is K-Nearest Neighbor (KNN) [1, 10], [23].

The essence of the algorithm is to find k closest objects from a sample already classified with pre-specified and numbered classes. Then, among these k objects, we need to count how often this class occurs and assign the type that occurs most often to our entity. If two categories' occurrences are the largest and the same, then the class with the smaller number is assigned [7, 19], [25].

The actuality of the theme: classification technology is widely used in various fields, not only in information technology. It seems that almost everywhere, you can find applications of this technology: medicine (diagnoses based on patient tests), banking (approval-rejection of the credit), and utilities (subsidies) [12, 24]. This list can go on for a very long time. In some cases, maximum accuracy with the least risk of misclassification is required.

The goal of the work: compare different KNN classification methods, offer new KNN classification methods and compare them with existing ones, show the importance of data analysis before KNN classification, to increase the accuracy of KNN classification using the techniques I have proposed [2, 18], [28].

The study aims to improve the accuracy of classification using KNN by comparing existing algorithms proposed by the author.

ANALYSIS OF LITERARY DATA

The classification problem was studied in L. Demidov and Y. Sokolov's "Classification of data based on SVM-algorithm and algorithm of k-nearest neighbours" [22]. This work provided a general idea of KNN but did not consider the data analysis before classification. Comparative characterization of the KNN classification with the SVM algorithm was performed here. However, no relative characteristics were given directly in the studied method.

If we talk about a more detailed analysis of the selected algorithm, we need to refer to the works of Matsugi, O. M., Arkhangelskaya, Y. M., Yushchenko, N. M. describes the features of this classification in his textbook "Information technologies of pattern recognition" [20, 26], [32]. For example, it indicates how to determine the class correctly. The K numbers of neighbours are two types are the most common. They suggest choosing the class with the most significant number of representatives in the training sample. But will this method always be effective because the training sample can consist of the same number of representatives of different classes? In this case, such an analysis would be futile.

In his research, Vaskiv, I. noted the exciting application of the KNN classification "Machine-learning methods in tasks of detection the atypical behaviour of complex system" [21, 27]. The author suggests looking for emissions using this algorithm. With KNN, you can classify the presented data, then you need to find the average distance between all classes, and if one of the classes is more distant from the others, then this group can be considered emissions.

The analysis of literature sources on this topic highlighted several insufficiently covered issues in the research. Therefore, this issue is essential, especially the issue of data analysis before classification. As for selecting the number k, no straightforward method would give the maximum result with great accuracy. Therefore, this method is quite interesting for research.

FORMULATION OF THE PROBLEM. ACTUALITY OF THE CLASSIFICATION PROBLEM USING THE ALGORITHM K NEAREST NEIGHBOURS

II. Many literary and media sources on the subject of classification using the K-Nearest Neighbor(KNN) Algorithm for Machine Learning were analyzed, and the most interesting, in our opinion, variations of this algorithm were selected.

The most common in the presented studies is the classical KNN algorithm, where a certain number of *k* is selected, and with the help of the Euclidean distance, *k* of the nearest neighbours is sought. The number of occurrences of each class is calculated. The class that has the most events is assigned to the classified object.

There is another algorithm – with the lowest average. Here everything happens the same as in the previous study, except for the final class choice. It does not count the number but calculates the average distance to each class in *k* nearest neighbours.

The latest algorithm combines the previous two. In most cases, this is a standard classical algorithm. Still, when the *k* nearest neighbours have the exact most significant occurrence of representatives of several classes, then the average distance for these classes will be calculated.

Another important detail from this work is using a hybrid variation of KNN; when there are only two classes, and k is an odd number, it makes no sense. In this case, the hybrid algorithm will work similarly to the classical algorithm. Data sets will always have two classes in the research topic: confirmed case and not verified. Therefore, this fact should be taken into account. If you use the classical variation and k is odd, then sorting the classes in the initial sample will not make sense. And the use of a hybrid method will not make sense at all.

However, our study will focus on the classical variation of KNN. In addition to the situation when several classes will have the same most significant number of representatives in the selected *k* nearest neighbours, we will avoid otherwise, adapting the algorithm to the task.

Emphasis will be placed on achieving maximum classification accuracy by comparing existing and proposed methods of selecting the *k* number and finding the nearest class. This requires a comparative algorithm characterisation with and without data analysis and preprocessing. All forms considered in this research will be compared purely practically [6, 16], [31].

The most common version of the *k* nearest neighbour algorithm is the classic KNN. It should be noted that each variation of this classification method requires a set of objects with predefined classes. For a classic KNN, you must choose a certain number *k* - the number of things closest to the given. Moreover, the variant with $k = 1$ is called the method of the nearest neighbor, and for $k = n$, where n is the number of objects in the initial sample, the classification loses its meaning. Therefore, selecting this number is an essential procedure for this algorithm. After selecting *k*, you need to look for the distances from this object to each classified object of the sample [8, 9], [29].

The most commonly used Euclidean space is determined by the following Equation 1:

$$d\,(p,\,q)=\sqrt{\sum_{i=1}^{n}(p_i-q_i)^2}\,. \qquad (1)$$

In the next step, from the all-founded distances, *k*, the smallest ones are selected, and among these *k* objects, the number of occurrences of each class is counted. The class with the most events is assigned to the selected object. If you cannot choose *k* nearest objects (the distance to (*k* + 1) thing is equal to the distance to *k* object in the sorted by distances sample), then the minimum possible larger than *k* number is taken. If *k* is larger than the size of the selection of the classified objects, then all are taken into account. But in this case, the classification loses its meaning because the selected object will be assigned to the most common in the initial sample class. If several types have the same number of occurrences in *k* nearest neighbours, the class with the smaller sequence number is assigned. Then the classified object is added to the general sample and considered in classifying the following ones [17, 19], [28].

As shown in Fig.1 and Fig.3 classes and 3 objects must be classified, *k*=5 is taken. First, we take the first object and look for the distances to each sample element with classified objects; then, we select the nearest 5. Lines have been drawn to these 5 chosen objects. As can be seen from Fig. 1, from the first object, 2 lines are drawn to the things of the second class, 2 – to the objects of the third class and 1 to the object of the first class. A line is also drawn to the second unclassified object, which should be returned when classifying it. Thus, the second class is assigned among the 5 nearest neighbors, second- and third-class representatives, because the second class has a smaller serial number [11, 13], [30].

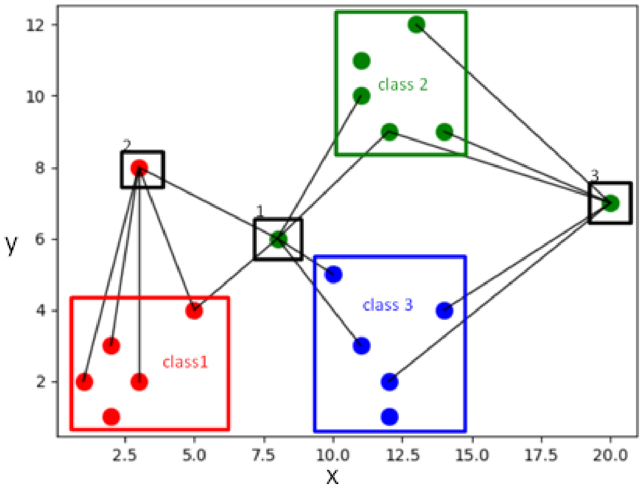


Fig. 1. **Visualization of the classification procedure using the classic KNN**
***Source:* compiled by the authors**

Let's move on to the classification of the second object. As shown in Fig. 1, among the five nearest neighbours, there are 4 first class representatives, so it is assigned to the first class. There is also one line drawn to the representative of the second class, which was not classified at the previous stage but after classification was added to the general sample and taken into account when ranking the second object.

In the next step, we move on to the third unclassified object. Everything is simple here – among the 5 nearest neighbors, there are 3 representatives of the second class and 2 representatives of the third class, so we assign it the second class [15, 28].

This research will take into account some variations of the classic algorithm. For example, the lowest mean KNN. In this case, the same steps as the previous method will be performed until the final class definition. The number of occurrences of each class in k nearest neighbors will not be counted, but the average distance for each class among them will be calculated. Then you need to assign the classified object to the class with the smallest average distance (Fig. 2).

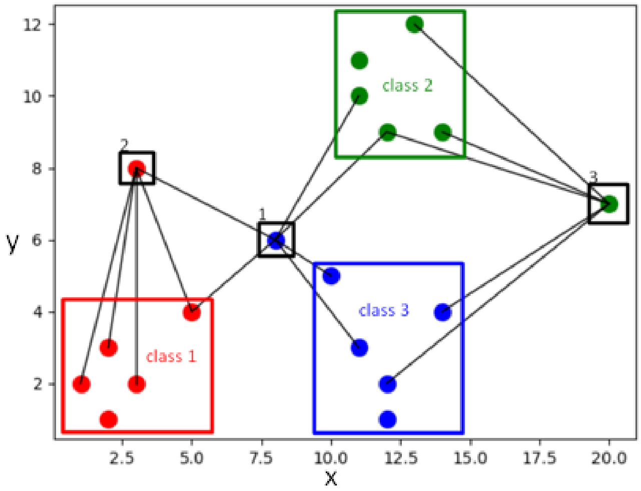


Fig. 2. **Visualization of the classification procedure using the lowest mean KNN**
Source: compiled by the authors

In Fig. 2, only the classification for the first object has changed, and visually it seems that this object is closer to the third class than to the second. But there is a problem – what if, for example an emission of a particular class is very close to the object being classified. Then, accordingly, this object will classify incorrectly (Fig. 3).

As shown in Fig. 3, a fourth class has been added, which is remote to the first object, but with a single emission close to it. And this object is assigned to the fourth class, which is wrong. Therefore, it would be more appropriate to use a hybrid of classical KNN and KNN with the lowest average [14, 17], [26]. The algorithm is executed sequentially, as in the classical KNN, until there is the exact most significant number of representatives of different classes among the k nearest neighbors. Then it would help if you took the smallest average distance among these classes (Fig. 4).

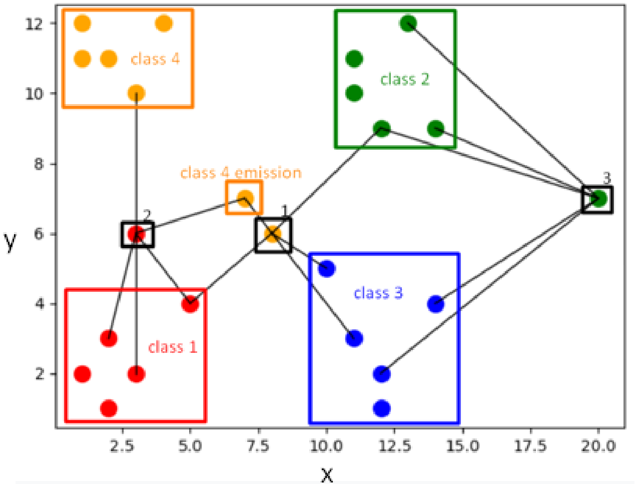


Fig. 3. **Visualization of the classification procedure using KNN with the lowest average with close emission of the remote class**
Source: **compiled by the authors**

As we see from Fig. 4, the first object is classified as the third class, although the nearest is the object of the fourth class, and the smallest average distance will be to the fourth class. But among the 5 nearest neighbors, there are 2 representatives of the second and third class and only 1 representative of the fourth. Further, if the classic KNN were used, the object would be classified as the second class because its serial number is smaller. But the average distance to the third class is the smallest, so the third class is assigned to the first object.

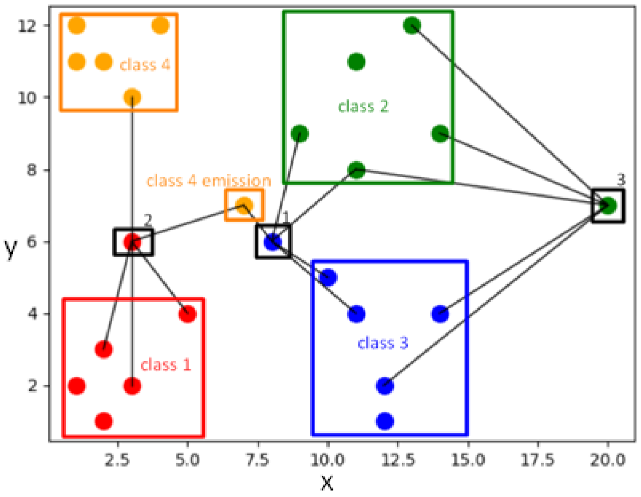


Fig. 4. **Visualization of the classification procedure using hybrid KNN with close emission of the remote class**
Source: **compiled by the authors**

The next problem of the k nearest neighbor's algorithm is the selection of the number k. Again, if you use $k = 1$ – this is the algorithm of the nearest neighbor, and the situation will be similar to the situation in Fig. 3. The research found that this option is not the best. If we take $k \geq n$, where n is the number of elements in the initial sample, then the object will be assigned a class that is most common in the initial model in the case of the classical KNN (Fig. 5).

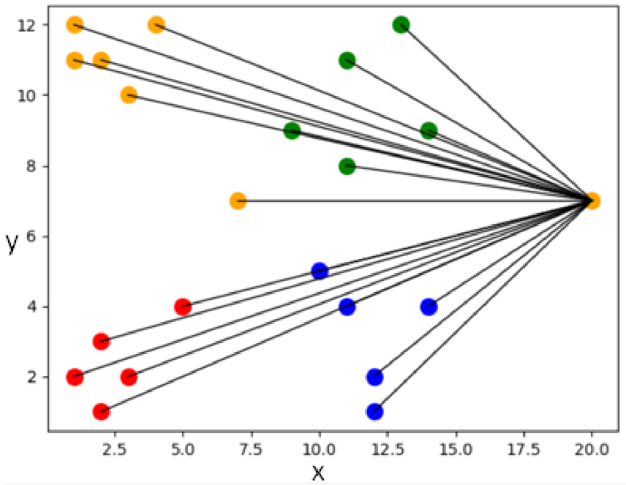


Fig. 5. **Visualization of the classification procedure using the classic KNN at *k* = *n***
Source: **compiled by the authors**

This problem is solved at the lowest mean KNN (Fig. 6).

In this case, a hybrid KNN can only be used if it is known that the initial sample has the same number of representatives of each class. Otherwise, you will get the same situation as with the classic KNN.

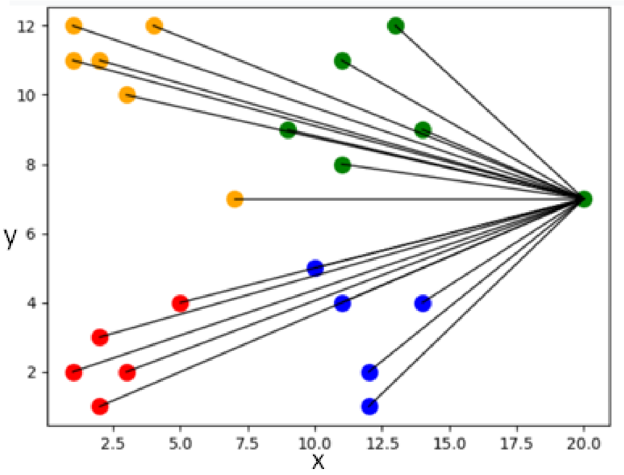
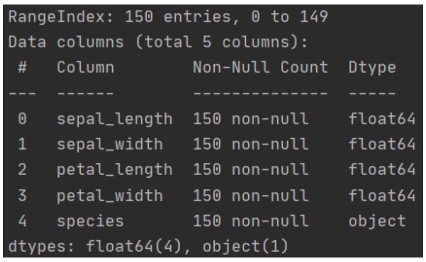


Fig. 6. **Visualization of the classification procedure using the lowest mean KNN at *k* = *n***
Source: **compiled by the authors**

In the next step, we will try to perform the classification on the example of accurate data. To quickly check the adequacy of the classification, the “Iris Flower Dataset” from the web resource kaggle.com was chosen [3, 28].

As can be seen from Fig. 7, this dataset contains information about 150 flowers, 4 features (length and width of the petal and sepals) and the class to which the flower with such features belongs.



```
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   sepal_length  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150 non-null    float64
 3   petal_width   150 non-null    float64
 4   species       150 non-null    object
dtypes: float64(4), object(1)
```

Fig. 7. **“Iris Flower Dataset” data**
***Source:* compiled by the authors**

Let's divide the dataset into test and training samples. The test sample will be formed randomly.

Let's classify flowers by three different variations of classification by the method of k nearest neighbours with the following parameters:

– training sample size – 140;
– test sample size – 10;
– k = 10.

Table 1. **Analysis of classified flowers by different variations**

Number	Classic KNN	The lowest mean KNN	Hybrid KNN	Real classes
1	Iris-versicolor	Iris-versicolor	Iris-versicolor	Iris-versicolor
2	Iris-virginica	Iris-virginica	Iris-virginica	Iris-virginica
3	Iris-setosa	Iris-setosa	Iris-setosa	Iris-setosa
4	Iris-versicolor	Iris-versicolor	Iris-versicolor	Iris-versicolor
5	Iris-virginica	Iris-virginica	Iris-virginica	Iris-virginica
6	Iris-setosa	Iris-setosa	Iris-setosa	Iris-setosa
7	Iris-versicolor	Iris-versicolor	Iris-versicolor	Iris-versicolor
8	Iris-setosa	Iris-setosa	Iris-setosa	Iris-setosa
9	Iris-virginica	Iris-versicolor	Iris-virginica	Iris-virginica
10	Iris-versicolor	Iris-virginica	Iris-virginica	Iris-virginica
Accuracy, %	90	90	100	

***Source:* compiled by the authors**

As we can see from Table 1, the classification is entirely accurate, and, as expected, the hybrid KNN algorithm gives the best results for the classification. But, since the test data is selected each time randomly, each new run of the program will provide a different result.

MATERIALS AND METHODS OF RESEARCH. COMPARE K ALGORITHMS K NEAREST NEIGHBOURS: CLASSIC KNN, KNN WITH THE LOWEST AVERAGE AND HYBRID KNN

We will study three variations of the algorithm k nearest neighbours: the classical KNN, KNN with the lowest mean and hybrid KNN. In this research, we will compare these algorithms for different sizes of the test sample and other numbers k.

A “FIFA 19 complete player dataset” from the kaggle.com web resource was selected for this experiment [4]. It contains 89 features of the 18207 players from the video game FIFA 19. Based on their football skills, let's try to classify them by positions.

A) Dataset preprocessing

To begin with, it is necessary to leave only the essential features for the classification: all the football skills of each player and their position. After that, there are only 35 columns.

The next step will be to analyze for missing data. Missed data is present in 48 cases for each football skill and 60 points for the position. After deleting the gaps, the size of the dataset decreased by 60 places. Therefore, this means that 48 players have no information about their skills. It is impossible to predict or calculate them, and 12 players have no position information, which is a crucial column for us. Therefore, you need to delete omissions in the dataset.

In Fig. 8, the remaining columns were analyzed, and a thermal map of the correlation between them was compiled.

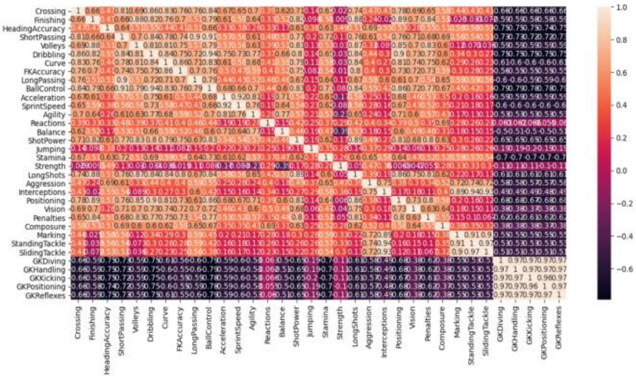


Fig. 8. **Correlation matrix between the features of the dataset**
***Source:* compiled by the authors**

From Fig. 8, you can see that some features have a very high correlation. If the correlation between the two parts is more significant than 0.9, one of them should be removed. After that, we have 27 columns left; 8 columns have been removed. You need to check whether any columns have a high correlation (Fig. 9).

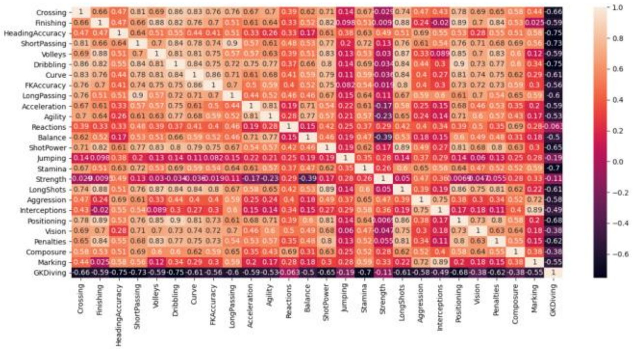


Fig. 9. **Correlation matrix between dataset columns after removing highly correlated columns**
Source: compiled by the authors

Next, we show a thermal map of the correlation between the positions. To do this, for each feature, you need to calculate the average by position (Fig. 10).

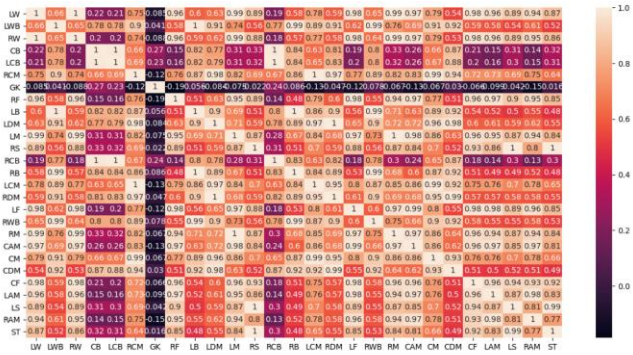


Fig. 10. **Correlation matrix between positions**
***Source:* compiled by the authors**

From Fig.10, it is seen that there are positions with a very high correlation; in some cases, it is even very close to 1. Let's add another feature to the selected dataset: the player's working leg. When the working leg is right – we will add 100 to the dataset and in the case when the active portion is left – 0. After that, the heat map should be shown again
(Fig. 11).

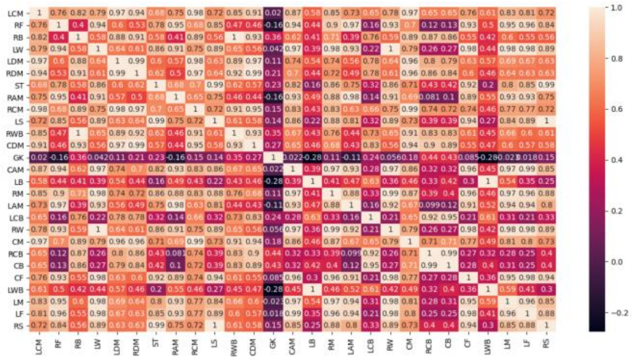


Fig. 11. **Correlation matrix between positions after adding a column with a working leg**
***Source:* compiled by the authors**

Fig. 11 shows that the situation has improved somewhat but remains a position with a very high correlation. In this case, it is not possible to adequately classify the objects. Therefore, it is necessary to group classes, according to Fig. 11.

Table 2. **Grouping of positions to classes**

No.	Class	Positions
1	Striker	LS, RS, ST
2	Attacking Player	LAM, CAM, RAM, LM, RM, LW, RF, RW, LF, CF
3	Defending Midfielder	CM, LCM, CDM, RCM, RDM, LDM
4	Right Defender	RB, RWB
5	Left Defender	LB, LWB
6	Central Defender	RCB, CD, LCB
7	Goalkeeper	GK

***Source:* compiled by the authors**

Table 2 shows that it is 7 classes. Again, we offer a heat map and check whether there are classes with a high correlation (Fig. 12).

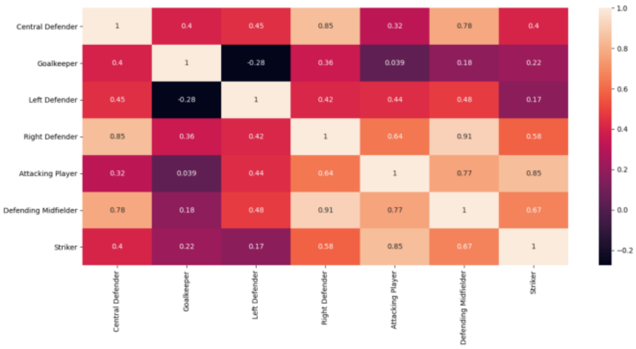


Fig. 12. **Correlation matrix between classes**
***Source:* compiled by the authors**

The heat map in Fig. 12 shows that some classes have a high correlation, but you can work with them and try to classify players.

After preprocessing, the dataset has 18147 objects with 28 features each.

B) Experimental Researches

Having received a ready-to-use dataset, you can check how the variations of the classification algorithm by the method of k nearest neighbours will work. We will conduct experiments with different k numbers and the test sample size. To begin with, it is necessary to make the table for the algorithm of classic KNN. To do this, take the training sample sizes 0.05, 0.1 and 0.2 of the total sample size, and take the k numbers 1, 2, 10, 50, 500, 5000 and 15000.

Also, it is worth noting that you need to change the distance search function. Instead of normalizing the data throughout the dataset, you should normalize it directly in the process to preserve its original view. In each skills list of the football player, it is necessary to look for the most outstanding value, divide 100 by this value and multiply each skill by the received coefficient. Thus, each player will have at least one parameter with an index of 100 (Table 3).

Table 3. **Accuracy of classic KNN algorithm classification, %**

k	Test sample size, % from whole sample		
	0.05	0.1	0.2
1	73.9	72.9	76.0
2	73.8	72.8	74.3
10	80.4	81.0	82.6
50	80.0	81.7	82.3
500	75.9	77.4	74.2
5000	59.8	61.4	60.8
15000	22.1	22.3	22.6

Source: **compiled by the authors**

The following should be Table 4, but for the KNN algorithm with the lowest mean. The same sample size and number *k* should be selected as parameters.

Table 4. **Accuracy of average mean KNN algorithm classification, %**

k	Test sample size, % from whole sample		
	0.05	0.1	0.2
1	73.9	72.9	76.0
2	73.9	72.8	76.0
10	62.3	61.8	62.9
50	49.4	46.2	46.1
500	50.4	43.3	41.3
5000	74.0	71.3	70.8
15000	59.8	56.7	57.2

Source: **compiled by the authors**

By the same principle, we build Table 5 for the hybrid KNN algorithm.

Table 5. **Accuracy of hybrid KNN algorithm classification, %**

k	Test sample size, % from whole sample		
	0.05	0.1	0.2
1	73.9	72.9	76.0
2	73.9	73.2	76.6
10	80.0	81.0	82.9
50	80.4	81.7	82.3
500	75.9	77.4	74.2
5000	59.8	61.4	60.9
15000	22.1	22.4	22.6

Source: **compiled by the authors**

According to the data obtained from Table 3, Table 4 and Table 5 we should build graphs (Fig. 13).

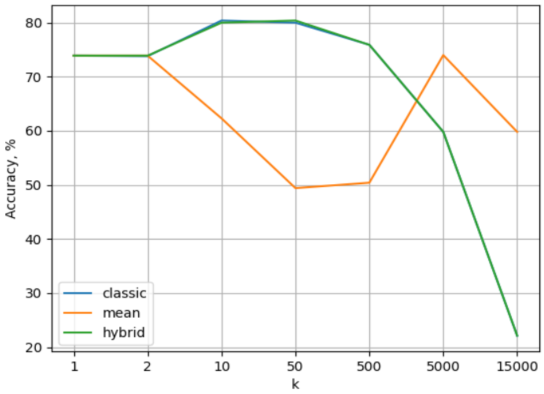


Fig. 13. **Graph of the accuracy of classification on the number k in different variations of the KNN algorithm for the dataset "FIFA 19 complete player dataset" and the size of the test sample 0.05**
Source: **compiled by the authors**

From Fig. 13, you can see that very similar results are given by the algorithms of classic and hybrid KNN, and the KNN algorithm with the lowest mean is distinguished. Moreover, the result is much worse, and the dynamics of accuracy are entirely different compared to the other two methods.

As we see in Fig. 14 and Fig.15, the overall accuracy dynamics did not change with increasing test sample size.

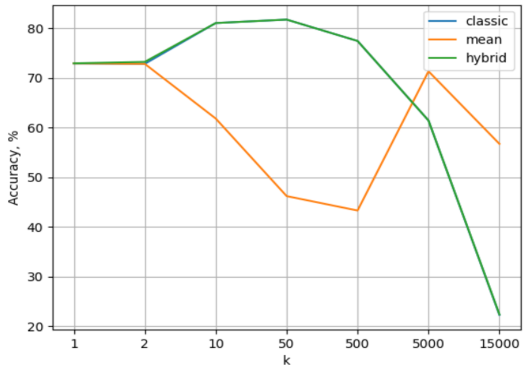


Fig. 14. **Graph of the accuracy of classification on the number k in different variations of the KNN algorithm for the dataset "FIFA 19 complete player dataset" and the size of the test sample 0.1**
Source: **compiled by the authors**

Fig. 13, Fig.14 and Fig. 15 does not clearly show the difference between the algorithms of the classic and hybrid k nearest neighbors. Therefore, we take the dataset which we worked on within the previous section ("Iris Flower Dataset" [4]). It has much less data than the previous one and will clearly show the difference between the two algorithms. Take the sizes of the training sample 0.1 %, 0.2 %, 0.3 %, 0.4 % and 0.5 % of the total sample size, and take the k number 1, 2, 5, 10, 20, 30, 40 and 50 (Fig. 16, Fig.17, Fig.18, Fig.19 and Fig.20).

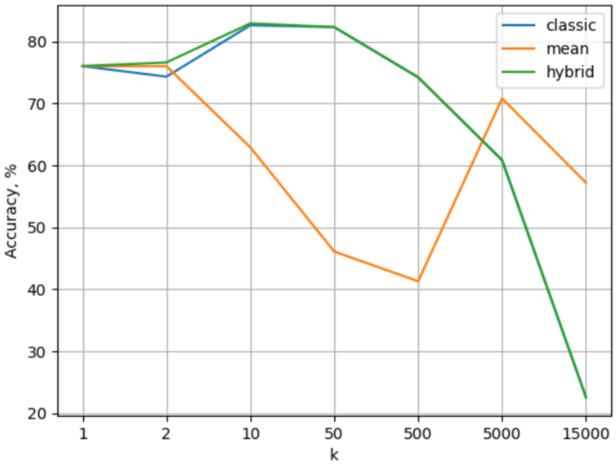


Fig. 15. **Graph of the accuracy of classification on the number k in different variations of the KNN algorithm for the dataset “FIFA 19 complete player dataset” and the size of the test sample 0.2**

Source: **compiled by the authors**

Table 6. **Accuracy of classic KNN algorithm classification, %**

k	Test sample size, % from whole sample				
	0.1	0.2	0,3	0.4	0.5
1	93.3	93.3	95.6	95.0	90.7
2	93.3	96.7	93.3	90.0	90.7
5	100.0	93.3	95.6	93.3	92.0
10	93.3	96.7	95.6	95.0	92.0
20	93.3	93.3	95.6	91.7	92.0
30	100.0	93.3	97.8	93.3	93.3
40	93.3	90.0	97.8	93.3	92.0
50	93.3	90.0	97.8	91.7	29.3

Source: **compiled by the authors**

According to the data obtained from Table 6, Table 7 and Table 8, we should build graphs.

In Fig. 16, with fewer data and classes, the difference between hybrid and classic KNN is visible. First of all, because with more periodic data and styles, it is much more likely that *k* nearest neighbours will have the same number of representatives of several classes, and in just a few such cases, it is much better seen in the graph.

Table 7. **Accuracy of lowest mean KNN algorithm classification, %**

k	Test sample size, % from whole sample				
	0.1	0.2	0.3	0.4	0.5
1	93.3	93.3	95.6	95.0	90.7
2	93.3	93.3	95.6	95.0	90.7
5	93.3	93.3	91.1	91.7	88.0
10	93.3	93.3	93.3	88.3	72.0
20	93.3	96.7	86.7	85.0	90.7
30	86.7	93.3	88.9	83.3	90.7
40	100	93.3	93.3	95.0	92.0
50	93.3	93.3	93.3	93.3	90.7

Source: **compiled by the authors**

Table 8. ***Accuracy of hybrid KNN algorithm classification, %***

k	Test sample size, % from whole sample				
	0.1	0.2	0.3	0.4	0.5
1	93.3	93.3	95.6	95.0	90.7
2	93.3	93.3	95.6	95.0	92.0
5	100.0	93.3	95.6	93.3	93.3
10	100.0	93.3	95.6	95.0	92.0
20	100.0	93.3	95.6	93.3	92.0
30	100.0	93.3	97.8	93.3	93.3
40	93.3	93.3	97.8	93.3	92.0
50	93.3	93.3	100.0	91.7	29.3

Source: **compiled by the authors**

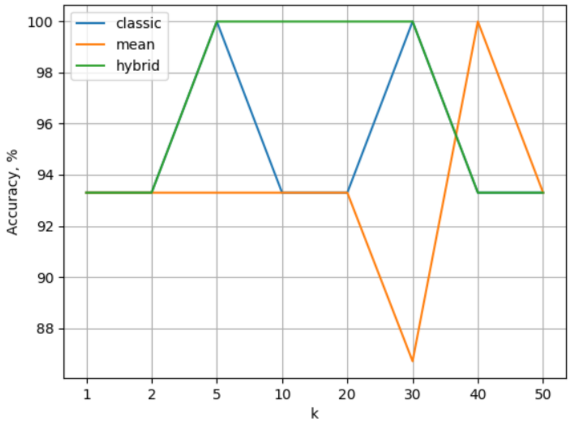


Fig. 16. **Graph of the accuracy of classification on the number k in different variations of the KNN algorithm for the dataset “Iris Flower Dataset” and the size of the test sample 0.1**

Source: **compiled by the authors**

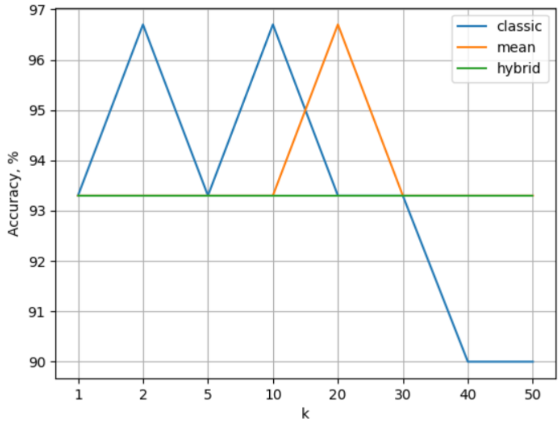


Fig. 17. **Graph of the accuracy of classification on the number k in different variations of the KNN algorithm for the dataset “Iris Flower Dataset” and the size of the test sample 0.2**

Source: **compiled by the authors**

In Fig.17, you can see that not always a hybrid variation will be better than the classic. But in this case, it is instead an exception because, with a small amount of data, the hybrid variation works at least not worse.

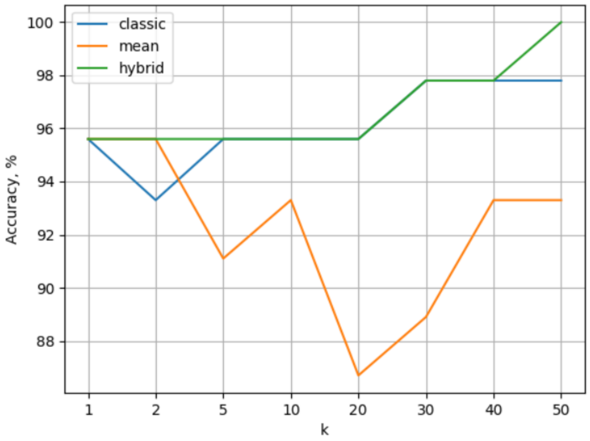


Fig. 18. **Graph of the accuracy of classification on the number k in different variations of the KNN algorithm for the dataset "Iris Flower Dataset" and the size of the test sample 0.3**
***Source:* compiled by the authors**

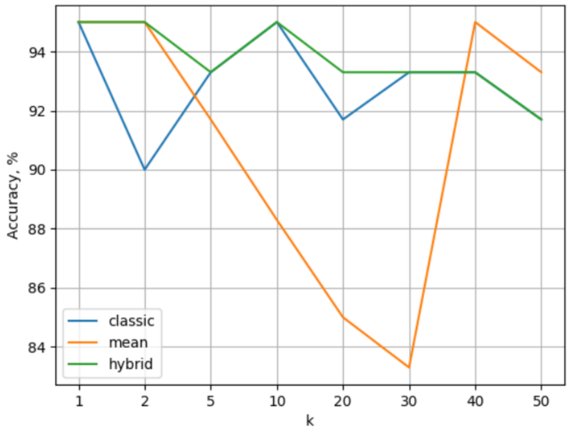


Fig. 19. **Graph of the accuracy of classification on the number k in different variations of the KNN algorithm for the dataset "Iris Flower Dataset" and the size of the test sample 0.4**
***Source:* compiled by the authors**

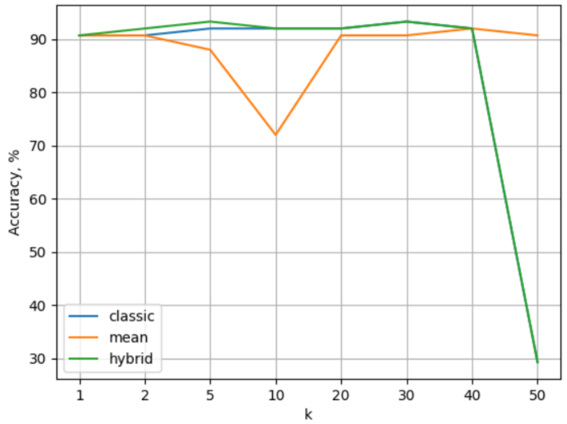


Fig. 20. **Graph of the accuracy of classification on the number k in different variations of the KNN algorithm for the dataset "Iris Flower Dataset" and the size of the test sample 0.5**
***Source:* compiled by the authors**

Fig. 16, Fig.17, Fig.18, Fig.19 and Fig.20 confirm that the hybrid variation of KNN works at least not worse than the classic one in most cases.

We will conduct independent research for the case when there are only two classes. To do this, use the dataset "Diabetes Data Set" [5]. We conduct experiments only for different k numbers. Take the following variants of the number *k*: 1, 2, 5, 10, 15, 30, 50. For the size of the test sample, we take 0.1. This research will be conducted only for classical and hybrid KNN (Table 9, Fig. 21).

Table 9. **Accuracy of KNN algorithm classification, %**

k	Classic KNN	Hybrid KNN
1	72.7	72.7
2	72.7	72.7
5	67.5	67.5
10	75.3	76.6
15	74.0	74.0
30	79.2	81.4
50	75.3	77.9

***Source:* compiled by the authors**

From Fig. 21, we can see that the accuracy is similar for classic and hybrid variations for each case where *k* is an odd number. It's because mixed interpretation differs from traditional only in the case with the two most extensive classes in *k* nearest neighbours. But *k* can't be an odd number because every odd number is not divisible for 2.

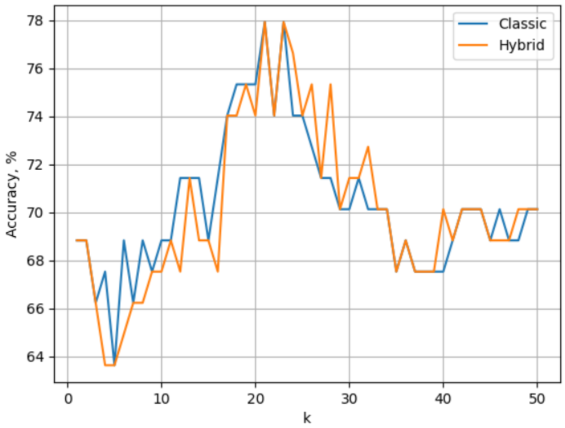


Fig. 21. **Graph of classification accuracy dependence on the number k in different variations of the KNN algorithm for the "Diabetes Data Set"**
***Source:* compiled by the authors**

It is also necessary to compare the operating time of these three algorithm variations. To do this, use the dataset "Diabetes Data Set, which has only two classes; we will take only even numbers *k* better to compare the speed of classical and hybrid variations [5].

As can be seen in Fig. 22, which consistently lasts the longest KNN with the lowest mean, and in most cases, the hybrid works even better than the classic. Quite exciting results, why it turned out will be discussed in the next section.

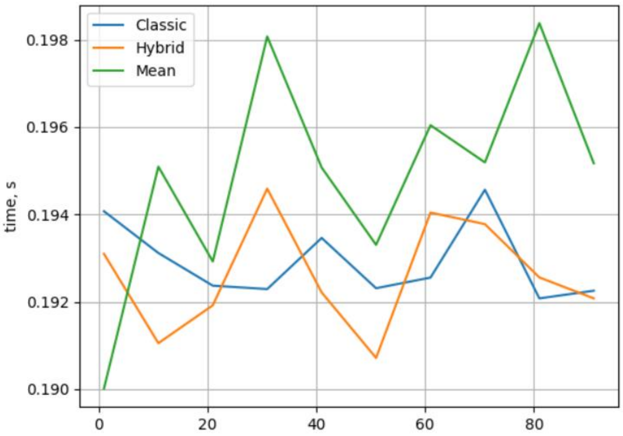


Fig. 22. **Graph of the dependence of the algorithm operation time on the number k in different variations of the KNN algorithm**
Source: **compiled by the authors**

RESULTS OF THE RESEARCH. ADVANTAGES OF ALGORITHMS FOR DIFFERENT TEST SAMPLE SIZES AND DIFFERENT K NUMBERS

In the previous section, research was performed for different classification variations by the algorithm of the *k* nearest neighbors for different test sample sizes and *k* numbers. The study was conducted on three datasets of various sizes. Parameters were selected separately for each dataset.

Let's analyze the results of the research for the first dataset. We can see that for $k = 1$, the accuracy is always the same (Table 3, Table 4 and Table 5). This will be the same for $k = 1$ in all cases because only one nearest neighbor is taken, which will be the same for all variations and, therefore, belong to the same class. To increase the speed of the nearest neighbor algorithm, you can search for the nearest object and get its class.

As shown in Fig. 13, Fig. 14 and Fig. 15, for the first dataset, the classification accuracy does not depend on the test sample size. Why so? There were 18147 objects in this dataset. The sizes of the test sample 0.05, 0.1 and 0.2 were chosen. So in the first case, the size of the training sample was 17239 (18147-0.05·18147), in the second case – 16332 (18147-0.1·18147), and in the third – 14517 (18147-0.2·18147). The difference between the sizes was insignificant for significant changes in classification accuracy. But you can make general graphs of classification accuracy on the *k* number (Fig. 23).

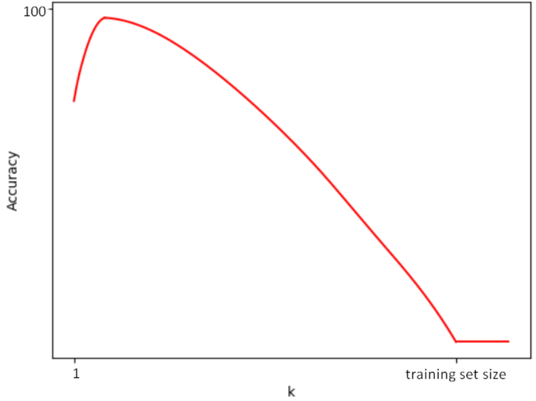


Fig. 23. **General view of the dependence of the classification accuracy on the number k for the classical KNN**
Source: **compiled by the authors**

As can be seen from Fig. 23, at first, the accuracy increases sharply and then gradually decreases. This can be explained by the fact that most of the *k* nearest neighbors first includes objects of the required class, but over time, things in this class end up in the training sample, and elements of other types begin adding. When k reaches the size of the training sample, then for each subsequent number *k*, the accuracy will not change because if *k* is greater than or equal to the size of the training sample, then this number will be similar to the size of the training sample. This graph is designed when only one object must be classified for a group of things. The chart will be different. In this case, classified elements will always be added to the training sample, the training sample size will be dynamic, and *k* is a constant. In practice, this algorithm will more often be used to classify one object. So, the Oy axis is not accurate in the classification. It’s the percentage of the object being classified correctly.

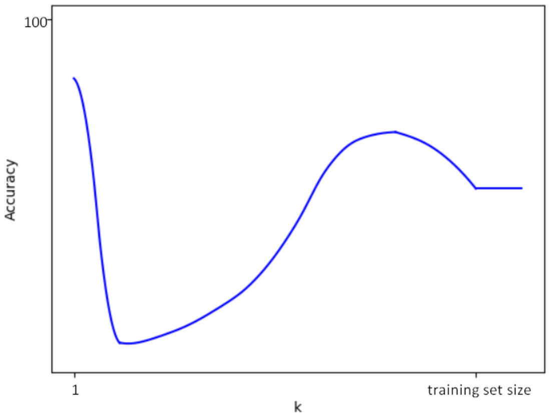


Fig. 24. **General view of the dependence of the classification accuracy on the number k for the lowest mean KNN**
Source: **compiled by the authors**

In Fig. 24, the algorithm *k* of the nearest neighbors is presented. At first, the accuracy decreases sharply and then gradually increases until a particular moment, after which it begins to fall smoothly. Again, when *k* is greater than or equal to the size of the training sample, the percentage will always be the same on the same data.

From Fig. 23 and Fig. 24, it can be seen that the maximum possible accuracy is higher in the classical KNN than in the KNN with the lowest mean because, as noted earlier, when *k* = 1 on the same data, the accuracy of these variations is the same, and in the second case it is the maximum accuracy. This may be because, at a small *k*, the sample of the nearest elements collects some emissions from other classes. As shown in Fig. 3, these are class emissions. The classical KNN rejects them because this class loses to the true one numerically. The KNN with the lowest mean also shows that the accuracy decreases in the end. The problem is reversed: a sample of the nearest elements begins to collect emissions of the required class, which are located far away and negatively affect the average class distance.

From Fig. 13, Fig. 14 and Fig. 15, the difference between the classic KNN and the hybrid KNN is challenging to see, so a dataset with fewer objects was taken (Table VII-IX) and plotted graphs based on his experimental data (Fig. 16, Fig. 17, Fig. 18, Fig. 19 and Fig. 20). These graphs show that the hybrid variation of KNN is usually not worse than the classic one, but there are exceptions (Fig. 17). The comparison of the general views of graphs of the dependence of accuracy on k for these variations will look as follows (Fig. 25).

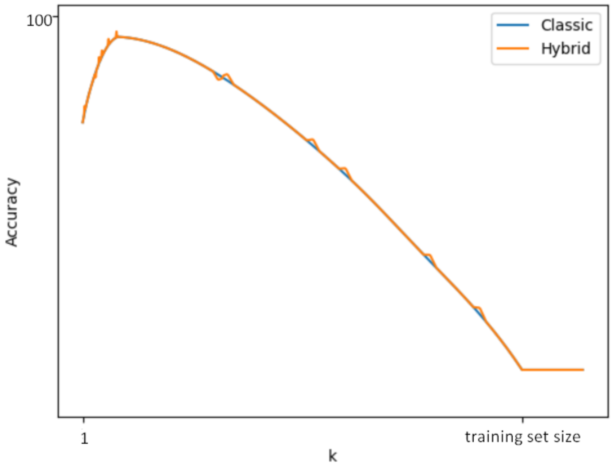


Fig. 25. **General view of the dependence of classification accuracy on the number k for classic and hybrid KNN**

***Source:* compiled by the authors**

It should also be noted that possible changes in the accuracy between the classical and hybrid KNN will be only if there are several classes among the k nearest neighbors with the exact most significant representation. So, it is necessary to distinguish cases in which the use of this variation is meaningless. For example, in cases where there are only two classes, *k* is an odd number. In this case, a situation with the exact most significant representation of several classes is impossible.

This can be seen in Fig. 21 that the classification accuracy of the hybrid and classical variation is the same for all odd numbers *k* in this case. Why so? The answer is simple. In this case, we have only two classes. Therefore, the distribution of classes in *k* nearest neighbours cannot be the same; consequently, the hybrid KNN algorithm will not work. Applying a hybrid variation of the *k* nearest neighbours algorithm does not make sense if we have only two classes, and *k* is an odd number. And does it make sense to use this algorithm if there are only two classes? From the identical Fig. 23, we see that at first, mainly the classical KNN gives more accurate results, but over time the algorithm of the hybrid KNN begins to give a better result.

Now, let's discuss the experiment's results with the time of operation of these three variations of the KNN algorithm because it is pretty interesting what we obtained in Fig. 24. First, it so happened that in most cases, the hybrid variant is faster than the classic because the same thing is done, except when the algorithm begins to calculate the average distances. At each iteration, there is a check whether there is not the same maximum number of representatives of one class, and then the hybrid KNN could not be faster. Here we can say that if the latter condition is met, the hybrid variation works longer than the classic one. For the classic KNN, it is necessary to put the classes in descending order of the number of their representatives in the training sample to increase accuracy when this operation is meaningless for a hybrid. The hybrid variation will always work faster in cases where there is only one most common class among *k* nearest neighbors.

So, why then does the variation with the lowest mean work much longer? Here the answer is much more straightforward. Simply finding the average for each class involves counting the representatives of each class and, in addition, finding the sum of all the distances for each class, which will always be longer.

Suppose the hybrid algorithm states that there are several classes, the representatives of which are most common. In that case, it is still faster than the variation with the lowest mean. By fulfilling this condition, the averages will be calculated only for those equally the most common classes, while the algorithm with the lowest average is for everyone. This is also an advantage because classes whose emissions are close to our object will not be considered. This will also increase accuracy.

Table 10 shows a comparison of the described methods.

Table 10. **Comparison of different variations of knn algorithm**

Variation KNN	Accuracy	Speed	Overall
Classic	1.5	1.5	3
Hybrid	1	1	2
With the lowest average	3	3	6

***Source:* compiled by the authors**

Table 10 results of the comparison once again prove the proposed hypothesis that the hybrid method in the analysis of the proposed datasets is the best.

CONCLUSION

In this work, the problem of increasing classification accuracy, and more specifically - improving the accuracy of type using the algorithm *k* nearest neighbours and its variations, was considered. This algorithm's three variations were considered: classic, with the lowest mean and hybrid.

A hybrid variation of the *k* nearest neighbour algorithm was developed in this work. In experiments, it proved to be quite good; giving the accuracy of classification is usually not less than the algorithm of the classic KNN.

But still, the hybrid KNN algorithm will not always be better than the classic KNN algorithm. Studies have shown the inexpediency of using the KNN algorithm with the lowest mean. At the peak of its accuracy, it surpassed the previous two algorithms. Moreover, it contains more mathematical operations, even slower than two other algorithms. This can be used only as part of a hybrid algorithm because it still has higher accuracy than random class choice. The hybrid KNN will only be relevant if the classification accuracy provided by the lowest mean KNN algorithm is significantly greater than the relative number of the most common class in the training sample.

Also, the hybrid algorithm is the fastest among the other two variations.

So, here are some theses for the correct use of classification algorithms:

1) If you use the algorithm of the classical KNN, it is necessary to number the classes in descending order of the number of occurrences of each class in the training sample.

2) The use of hybrid variation will only make sense if the accuracy of the KNN classification with the lowest mean gives accuracy much greater than the relative number of occurrences of the most popular class in the training sample.

3) Use a hybrid variation of KNN when only two classes and *k* are odd numbers. It has no sense. In this case, the hybrid algorithm will work the same way as the classic algorithm.

4) Using variation with the lowest mean makes sense only in the hybrid algorithm. Otherwise, you can always choose a number *k* for which the accuracy of another algorithm will be higher than that of KNN with the lowest mean with the current number *k*.

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. p. 359–367. DOI: https://doi.org/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. p. 27–34. DOI: https://doi.org/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: 188–200. – Available from: https://doi.org/10.1007/978-3-030-82014-5_13. – [Accessed June 2020].

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: 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. p. 418–427.

6. Boyko, N. “Application of mathematical models for improvement of “cloud” data processes organization”. *Mathematical Modeling and Computing*. 2016; Vol.3(2): 111–119. DOI: https://doi.org/10.23939/mmc2016.02.111.

7. Bengio, Y., Simard, P. & Frasconi, P. “Learning long-term dependencies with gradient descent is difficult”. *IEEE Trans. Neural Networks*. 1994; 5 (2): 157–166. DOI: https://doi.org/10.1109/72.279181.

8. 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.* 0853. 2017. p. 1246–1259. DOI: https://doi.org/10.1101/114892.

9. Chaudhary, K., Poirion, O. B., Lu, L. & Garmire, L. X. "Deep learning–based multi-omics integration robustly predicts survival in liver cance". *Clin. Cancer Res.* 2018; 24 (6): 1248–1259. DOI: https://doi.org/10.1158/1078-0432.CCR-17-0853.

10. Cheng, B., Liu, M., Zhang, D., Musell, B. C. & Shen, D. Domain Transfer Learning for MCI Conversion Prediction. *IEEE Trans. Biomed. Eng.* 2015; 62 (7): 1805–1817. DOI: https://doi.org/10.1109/TBME.2015.2404809.

11. Choi, E., Schuetz, A., Stewart, W. F. & Sun, J. Using recurrent neural network models for early detection of heart failure onset. *J. Am. Med. Inf. Assoc.* 2017; 24 (2): 361–370.

12. Deng, L., Hinton, G. & Kingsbury, B. "New types of deep neural network learning for speech recognition and related applications: An overview. *In: Acoustics, Speech and Signal Processing (ICASSP)". IEEE International Conference on: 2013*. p. 8599–8603. DOI: https://doi.org/10.1109/ICASSP.2013.6639344.

13. Huang, M., Yang, W., Feng, Q., Chen, W., Weiner, M. W., Aisen, P., et al. "Longitudinal measurement and hierarchical classification framework for the prediction of Alzheimer's disease". *Sci Rep* 7. 2017. DOI: https://doi.org/10.1038/srep39880.

14. Lama, R. K., Gwak, J., Park, J.-S. & Lee, S.-W. "Diagnosis of Alzheimer's disease based on structural MRI images using a regularized extreme learning machine and PCA features". *J. Healthcare Eng.* 2017. DOI: https://doi.org/10.1155/2017/5485080.

15. Larranaga, P., Calvo, B., Santana, R., Bielza, C., Galdiano, J., Inza, I., et al. "Machine learning in bioinformatics". *Briefings Bioinf.* 2006; 7 (1): 86–112. DOI: https://doi.org/10.1093/bib/bbk007.

16. LeCun, Y., Bengio, Y. & Hinton, G. "Deep learning". *Nature.* 2015; 521 (7553): 436–444. DOI: https://doi.org/10.1038/nature14539.

17. Lee, G., Nho, K., Kang, B., Sohn, K. A. & Kim, D. "Predicting Alzheimer's disease progression using multi-modal deep learning approach". *Sci. Rep.* 2019; 9(1): p. 1952. DOI: https://doi.org/10.1038/s41598-018-37769-z.

18. Lu, D., Popuri, K., Ding, G. W., Balachandar, R. & Beg, M. F. "Multimodal and multiscale deep neural networks for the early diagnosis of Alzheimer's disease using structural mr and fdg-pet images". *Sci Rep 9.* 2018. DOI: https://doi.org/10.1038/s41598-018-22871-z.

19. Sandeep, C., Kumar, A., Mahadevan, K. & Manoj, P. "Feature extraction of MRI brain images for the early detection of Alzheimer's disease". *Bioprocess Eng.* 2017; 1 (2): 35–42. DOI: https://doi.org/10.1109/I2C2.2017.8321780.

20. Young, J., Modat, M., Cardoso, M. J., Mendelson, A., Cash, D., Ourselin, S. Initiative AsDN: accurate multimodal probabilistic prediction of conversion to Alzheimer's disease in patients with mild cognitive impairment. *NeuroImage Clin.* 2013; 2: 735–745. DOI: https://doi.org/10.1016/j.nicl.2013.05.004.

21. Zhang, D. & Shen, D. "Initiative AsDN: multi-modal multi-task learning for joint prediction of multiple regression and classification variables in Alzheimer's disease". *NeuroImage.* 2012; .59 (2): 895–907. DOI: https://doi.org/10.1016/j.neuroimage.2011.09.069.

22. Zhang, D. & Shen, D. "Initiative AsDN: predicting future clinical changes of MCI patients using longitudinal and multimodal biomarkers". *PloS One.* 2012; 7 (3). DOI: https://doi.org/10.1371/journal.pone.0033182.

23. Diks, C., Hommes, C. & Wang, J. "Critical slowing down as an early warning signal for financial crises?" *Empirical Economics.* 2019; 57 (4): 1201–1228. DOI: https://doi.org/10.1007/s00181-018-1527-3.

24. Kölbel, J. F., Busch, T. &Jancso, L. M. "How media coverage of corporate social irresponsibility increases financial risk". *Strategic Management Journal.* 2017; 38(11): 2266–2284. DOI: https://doi.org/10.1002/smj.2647.

25. Bouslah, K., Kryzanowski, L. & M'Zali, B. "Social performance and firm risk: impact of the financial crisis". *Journal of Business Ethics.* 2018; 149 (3): 643–669. DOI: https://doi.org/10.1007/s10551-016-3017-x.

26. Srinivasan, S. & Kamalakannan, T. Multi criteria decision making in financial risk management with a multi-objective genetic algorithm. *Computational Economics.* 2018; 52 (2): 443–457. DOI: https://doi.org/10.1007/s10614-017-9683-7.

27. 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.* 2019; 13 (2): 521–526. DOI: http://doi.org/10.11591/ijeecs.v13.i2.pp521-526.

28. Jothi, R., Mohanty, S. K. & Ojha, A. DK-means: a deterministic k-means clustering algorithm for gene expression analysis. *Pattern Analysis and Applications.* 2019; 22(2): 649–667. DOI: https://doi.org/10.1007/s10044-017-0673-0.

29. Shakeel, P. M., Baskar, S., Dhulipala, V. S. & Jaber, M. M. “Cloud based framework for diagnosis of diabetes mellitus using K-means clustering”. *Health Information Science and Systems*. 2018; 6 (1): 1–7. DOI: https://doi.org/10.22937/IJCSNS.2021.21.6.31.

30. Slamet, C., Rahman, A., Ramdhani, M. A. & Darmalaksana, W. “Clustering the verses of the Holy Qur'an using K-means algorithm”. *Asian Journal of Information Technology*. 2016; Vol.15 No.24: 5159–5162.

31. Bekiros, S., Nguyen, D. K., Sandoval Junior, L. & Uddin, G. S. “Information diffusion, cluster formation and entropy-based network dynamics in equity and commodity markets”. *European Journal of Operational Research*. 2017; 256(3): 945–961. DOI: https://doi.org/10.1016/j.ejor.2016.06.052.

32. Polyakova, M. V. & Krylov, V. N. “Data normalization methods to improve the quality of classification in the breast cancer diagnostic system”. *Applied Aspects of Information Technology*. 2022; 5(1): 55–63. DOI: https://doi.org/10.15276/aait.05.2022.5.

Conflicts of Interest: the authors declare no conflict of interest

Received 20.01.2021
Received after revision 27.02.2021
Accepted 14.03.2021

DOI: https://doi.org/10.15276/aait.05.2022.11
УДК 004.832.2

Методи машинного навчання для класифікації мультимодальних даних

Наталія Іванівна Бойко[1)]
ORCID: https://orcid.org/0000-0002-6962-9363; nataliya.i.boyko@lpnu.ua. Scopus Author ID: 57191967462
Михайло Васильович Музика[1)]
ORCID: https://orcid.org/0000-0001-8285-1631; muzyka.m.00@gmail.com
[1)] Національний університет “Львівська політехніка”, вул. Професорська, 1. Львів, 79013, Україна

АНОТАЦІЯ

У цій роботі запропоновані методи аналізу мультимодальних методів даних, які сприябть підвищенню загальної точності результатів, а також методи класифікації К-найближчого сусіда (KNN) для мінімізації їх ризику. Розглядається механізм підвищення точності класифікації KNN. Методами дослідження, які використовуються в даній роботі, є порівняння, аналіз, індукція, експеримент. Ця робота була спрямована на підвищення точності класифікації KNN шляхом порівняння вже існуючих алгоритмів та застосування нових методів. Було проаналізовано багато літературних та медійних джерел на тему класифікації за алгоритмом k найближчих сусідів та обрано найцікавіші, варіації поданого алгоритму. Акцент буде зроблено на досягненні максимальної точності класифікації шляхом порівняння існуючих і їх удосконалення існуючих методів вибору числа k і знаходження найближчого класу. Також порівнюються алгоритми з аналізом і попередньою обробкою даних і без них. Усі стратегії, які розглядаються в цій статті, будуть досягнуті суто практичним шляхом. Проведено експериментальну класифікацію за k найближчими сусідами з різними варіаціями. Даними для експерименту використовувались два різних набори даних різного розміру. В якості аргументів класифікації були взяті різні класифікації k і розмір тестової вибірки. В роботі вивчаються три варіанти алгоритму k найближчих сусідів: класичний KNN, KNN з найменшим середнім і гібридний KNN. Здійснюється порівняння цих алгоритмів для різних розмірів тестової вибірки для інших чисел k. У статті аналізуються дані перед класифікацією. Що стосується підбору числа k, то не існує простого методу, який би дав максимальний результат з великою точністю. Суть алгоритму полягає в тому, щоб знайти k найближчих до вибірки об'єктів, які вже класифіковані за попередньо заданими та пронумерованими класами. Потім серед цих k об’єктів потрібно порахувати, скільки разів зустрічається клас, і призначити обраному об’єкту найпоширеніший клас.

Ключові слова: метод; алгоритм; аналіз; машинне навчання; мультимодальні дані; класифікація; К-найближчий сусід

ABOUT THE AUTHORS

Nataliya I. Boyko - Candidate of Economic Sciences, Associate Professor of the Artificial Intelligent Systems Department of Lviv Polytechnic National University, 1, Profesorska Street. Lviv, 79013, Ukraine
ORCID: https://orcid.org/0000-0002-6962-9363; nataliya.i.boyko@lpnu.ua. Scopus Author ID: 57191967462
Research field: Machine learning; data visualization; intellectual data analysis; system analysis

Наталія Іванівна Бойко - кандидат економічних наук, доцент кафедри Системи штучного інтелекту. Національний університет “Львівська політехніка”, вул. Професорська, 1. Львів, 79013, Україна

Mykhaylo V. Muzyka - Student of the Artificial Intelligent Systems Department of Lviv Polytechnic National University, 1, Profesorska Street. Lviv, 79013, Ukraine
ORCID: https://orcid.org/0000-0001-8285-1631; muzyka.m.00@gmail.com.
Research field: machine learning; data visualization

Михайло Васильович Музика - студент 4-го курсу кафедри Системи штучного інтелекту. Національний університет “Львівська політехніка”, вул. Професорська, 1. Львів, 79013, Україна