UDC 519.2 **DOI: 10.20998/2079-0023.2021.01.16**

***K. S. YAMKOVYI***

## DEVELOPMENT AND COMPARATIVE ANALYSIS OF SEMI-SUPERVISED LEARNING ALGORITHMS ON A SMALL AMOUNT OF LABELED DATA

The paper is dedicated to the development and comparative experimental analysis of semi-supervised learning approaches based on a mix of unsupervised and supervised approaches for the classification of datasets with a small amount of labeled data, namely, identifying to which of a set of categories a new observation belongs using a training set of data containing observations whose category membership is known. Semi-supervised learning is an approach to machine learning that combines a small amount of labeled data with a large amount of unlabeled data during training. Unlabeled data, when used in combination with a small quantity of labeled data, can produce significant improvement in learning accuracy.
The goal is semi-supervised methods development and analysis along with comparing their accuracy and robustness on different synthetics datasets. The proposed approach is based on the unsupervised *K*-medoids methods, also known as the Partitioning Around Medoid algorithm, however, unlike *K*-medoids the proposed algorithm first calculates medoids using only labeled data and next process unlabeled classes – assign labels of nearest medoid. Another proposed approach is the mix of the supervised method of *K*-nearest neighbor and unsupervised *K*-Means. Thus, the proposed learning algorithm uses information about both the nearest points and classes centers of mass.
The methods have been implemented using Python programming language and experimentally investigated for solving classification problems using datasets with different distribution and spatial characteristics. Datasets were generated using the scikit-learn library. Was compared the developed approaches to find average accuracy on all these datasets. It was shown, that even small amounts of labeled data allow us to use semi-supervised learning, and proposed modifications ensure to improve accuracy and algorithm performance, which was demonstrated during experiments. And with the increase of available label information accuracy of the algorithms grows up. Thus, the developed algorithms are using a distance metric that considers available label information.

**Keywords:** Unsupervised learning, supervised learning. semi-supervised learning, clustering, distance, distance function, nearest neighbor, medoid, center of mass.

***К. С. ЯМКОВИЙ***

## РОЗРОБКА ТА ПОРІВНЯЛЬНИЙ АНАЛІЗ АЛГОРИТМІВ НАВЧАННЯ З ЧАСТКОВИМ ЗАЛУЧЕННЯМ ВЧИТЕЛЯ НА МАЛІЙ КІЛЬКОСТІ РОЗМІЧЕНИХ ДАНИХ

Дана робота присвячена розробці та порівняльному аналізу алгоритмів навчання з частковим залученням вчителя, заснованих на поєднанні неконтрольованих та контрольованих підходів до класифікації наборів даних з невеликою кількістю маркованих даних, а саме виявленню, до якої з набору категорій нове спостереження належить за допомогою навчального набору даних, що містить спостереження, приналежність до категорії яких відома. Навчання з частковим залученням вчителя – це підхід до машинного навчання, який поєднує невелику кількість маркованих даних з великою кількістю немаркованих даних під час навчання. Немарковані дані, якщо їх використовувати в поєднанні з невеликою кількістю маркованих даних, можуть значно покращити точність навчання.
Метою роботи є розробка та аналіз методів навчання з частковим залученням вчителя, а також порівняння їх точності та надійності на різних наборах штучних даних. Запропонований підхід заснований на методі неконтрольованого навчання *K*-медоїдів, також відомий як алгоритм Розбиття навколо медоїдів, однак, на відміну від *K*-медоїдів, запропонований алгоритм спочатку обчислює медоїди, використовуючи лише марковані дані, а далі обробляє не марковані елементи - призначає мітки найближчих медоїд. Іншим запропонованим підходом є поєднання контрольованого методу *K*-найближчіх сусідів та неконтрольованого *K*-середніх. При цьому запропонований алгоритм навчання використовує інформацію як про найближчі точки, так і про класи центрів маси.
Методи були реалізовані з використанням мови програмування Python та експериментально досліджені для вирішення проблем класифікації з використанням наборів даних з різними розподілом та просторовими характеристиками. Набори даних були сформовані за допомогою бібліотеки scikit-learn. Було порівняно розроблені підходи за їх середню точність за всіма датасетами. Було показано, що навіть невеликі кількості маркованих даних дозволяють використовувати навчання з частковим залученням вчителя, а запропоновані модифікації забезпечують підвищення точності та роботи алгоритму, що було продемонстровано під час експериментів. І зі збільшенням доступної інформації про ярлики, точність алгоритмів зростає. Таким чином розроблені алгоритми використовують метрику відстані, яка враховує доступну інформацію про ярлик.

**Ключові слова:** навчання без учителя, навчання з учителем, навчання з частковим залученням вчителя, кластеризація, відстань, функція відстані, найближчий сусід, медоід, центр мас.

***К. С. ЯМКОВОЙ***

## РАЗРАБОТКА И СРАВНИТЕЛЬНЫЙ АНАЛИЗ АЛГОРИТМОВ ОБУЧЕНИЯ С ЧАСТИЧНЫМ ПРИВЛЕЧЕНИЕМ УЧИТЕЛЯ НА МАЛОМ КОЛИЧЕСТВЕ РАЗМЕЧЕННЫХ ДАННЫХ

Данная работа посвящена разработке и сравнительному анализу алгоритмов обучения с частичным привлечением учителя, основанных на сочетании подходов с привлечение учителя и без к классификации наборов данных с небольшим количеством маркированных данных, а именно выявлению, к какой из набора категорий новое наблюдение предстоит с помощью учебного набора данных, содержащего наблюдения, принадлежность к категории которых известна. Обучение с частичным привлечением учителя – это подход к машинному обучению, который сочетает небольшое количество размеченных данных с большим количеством не размеченных данных во время обучения. Не размеченные данные, если их использовать в сочетании с небольшим количеством размеченных данных, могут значительно улучшить точность обучения.
Целью работы является разработка и анализ методов обучения с частичным привлечением учителя, а также сравнение их точности и надежности на различных наборах искусственных данных. Предложенный подход основан на методе обучения без учителя *K*-Медоид, также известного как алгоритм Разбивка вокруг медоид, однако, в отличие от стандартного *K*-Медоид, предложенный алгоритм сначала вычисляет медоиды, используя только размеченные данные, а дальше обрабатывает не размеченные – назначает метки ближайших медоидов. Другим предложенным подходом является сочетание метода обучения с учителем *K*-ближайших соседей и обучения без учителя *K*-средних. При этом предложенный алгоритм обучения использует информацию как о ближайших точки, так и о классах центров масс.
Методы были реализованы с использованием языка программирования Python и экспериментально исследованы для решения проблемы классификации с использованием наборов данных с различными распределением и пространственными характеристиками. Наборы данных были сформированы с помощью библиотеки scikit-learn. Были проведено сравнение разработанных подходов на основе их средней точности

по всем датасета. Было показано, что даже небольшие количества размеченных данных позволяют использовать обучение с частичным привлечением учителя, а предложенные модификации обеспечивают повышение точности и устойчивости алгоритма, что было продемонстрировано во время экспериментов. И с увеличением доступной информации о ярлыки, точность алгоритмов растет. Таким образом, разработанные алгоритмы используют метрику расстояния, учитывающую доступную информацию о метках классов.

**Ключевые слова:** обучение без учителя, обучение с учителем, обучение с частичным привлечением учителя, кластеризация, расстояние, функция расстояния, ближайший сосед, медоид, центр масс.

**Introduction.** A large amount of data was produced recently, and nowadays humanity had the opportunity to store and process all this data. In all spheres of life people try to use these data for optimizing business and life-improving using AI and data mining.

There are several approaches to data processing and analysis problem within the framework of machine learning paradigms. One of them is unsupervised learning [1] when we try to detect inner structure or patterns without human supervision. The most efficient approach in machine learning is supervised learning, when we have some data with labels and try to learn a function on data points as label pairs. In many cases, there is no opportunity to label all data from different cases, causes are too complex and expensive experiments, data streaming with large frequency [2], or just high cost of data labeling. Therefore, in this case a satisfactory compromise is semi-supervised learning when we use datasets with a small amount of labeled that allows learning better inner structure (fig 1).

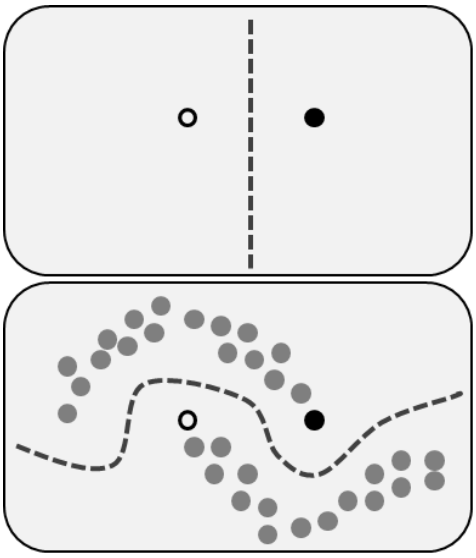


Fig. 1. Example of unlabeled data in semi-supervised learning

Semi-supervised learning includes different approaches, and can be used for any popular data analysis problems, such as clustering [3], anomaly detection, latent variables models.

The object of the study is the process of the data points classifications, namely, identifying to which of a set of categories a new observation belongs using a training set of data containing observations whose category membership is known.

The subject of the study is development of semi-supervised methods for data classification.

The purpose of the work is to develop an improved semi-supervised method using already exist supervised and unsupervised approaches and compare their accuracy and robustness.

**Problem statement.** Given a set of $l$ labeled examples $\{< x_1, y_1 >, \dots, < x_l, y_l >\}$, where $x_i$ – feature vector of i-th example and $y_i$ – its label (class), and a set of $u$ unlabeled data $\{x_{l+1}, \dots, x_{l+u}\}$ $x_1, x_2, \dots, x_{l+u} \in X$ and $y_1, y_2, \dots, x_l \in Y$. The goal is to determine some function using given sets, that will correct map points from X to Y: $f(x_j) = y_j$ for any point from $X$.

**Related work.** The semi-supervised learning described in literature not so widely as unsupervised or supervised, especially algorithms implementation.

In [4] Jesper E. van Engelen and Holger H. Hoose gives an overview of semi-supervised approaches describes assumptions of semi-supervised learning especially: smoothness, low-density and manifold.

Semi-supervised approach demonstrates high efficiency in solving clustering problems, the idea of using of clustering algorithm was described in the review [5]. The majority of these methods are modifications of the popular *k*-means clustering method.

One of the simplest unsupervised approach is *K*-Medoids also known as Partitioning Around Medoid algorithm was proposed in 1987 by Kaufman and Rousseeuw in [6]. A medoid is a point in the cluster, whose average dissimilarities with all the other points in the cluster is minimum.

*K*-medoid is a partitioning technique of clustering, which clusters the data set of n objects into *k* clusters, with the number *k* of clusters assumed known a priori.

Both the *k*-means and *k*-medoids algorithms are partitional, which breaking the dataset up into groups, and both attempt to minimize the distance between points labeled to be in a cluster, and a point designated as the center of that cluster. In contrast to the *k*-means algorithm, *k*-medoids choose data points as centers and can be used with arbitrary distances, while in *k*-means the center of a cluster is the average between the points in the cluster (fig. 2). Consequently, *K*-medoids is more robust to noise and outliers as compared to *K*-means.

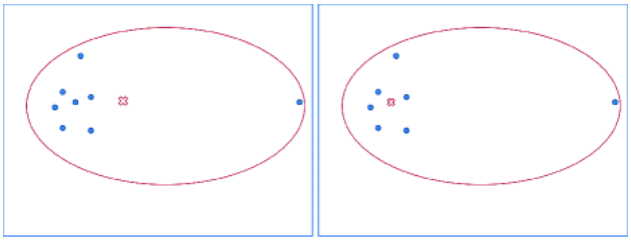



Fig. 2. Mean and medoid difference

The supervised approach described in [7]. The nearest neighbor decision rule assigns to an unclassified sample point the classification of the nearest of a set of previously classified points. Thus, for any number of categories, the probability of error of the nearest neighbor rule is bounded above by twice the Bayes probability of error. In this sense, it may be said that half the classification information in an

infinite sample set is contained in the nearest neighbor.

One of the popular semi-supervised methods is kernels based methods [8], especially Transductive support vector machines [9, 10]. This method has the same pros and cons as classic Support Vector Machine, but the main cons are that the algorithm works only with binary classification and has exponential computation time while a data set to increase.

**Semi-supervised methods.** As a baseline was chosen clustering algorithms implemented in the scikit-learn library [11, 12]. Algorithms use different approaches and library has interfaces for using custom metrics.

The proposed algorithm uses the *K*-medoid approach as a base idea. However, unlike *K*-medoids the proposed algorithm first calculates medoids using only labeled data and next process unlabeled classes – assign labels of nearest medoid.

This algorithm has the following pros:

– reduced processing time, because required only multiple iteration throw points unlike standard *K*-medoid;

– more robustness to wrong assigned labels, because the algorithm gives higher weights to labeled data in the medoids calculation step.

Another proposed approach uses the idea of *K*-nearest neighbors and *K*-Mean algorithm, because for classifying we use both information about the nearest points and classes centers of mass (algorithm 1).

As a distance metric was used Euclidean distance but any metric could be used.

Classes' centers do not recalculate after each assignment, because experiments show that it does not bring results but takes more computation time.

So, the described above method allows:

– consider information about the nearest point, because in most cases point has the same label as its neighbors;

– combine a different kind of information;

– tune weight of different sources using input parameters.

**Experiments.** For experiments purpose was generated multiple datasets using sklearn library. Each dataset contains 250 points in 2D space. Available only 10% of labels as default. In addition, datasets have multiple clusters with different distributions and shapes (fig. 3).

We will compare different approaches to find average accuracy on all these. In Tab. 1 we can see that the best-unsupervised method is *K*-nearest neighbors based algorithm has higher average accuracy. The fig. 4 shown the same result. Especially the *K*-nearest neighbors based approach has better accuracy in case of closely located clusters with the same distribution.

Another required feature of a semi-supervised algorithm is quality versus a number of labels dependency: more labels – higher quality and vice versa. However, fig. 5 shows that the proposed methods perform more accuracy with increasing number of available labels.

Table 1 – Accuracy comparison

| Method name | Dataset name | | | Average accuracy |
|---|---|---|---|---|
| | Moons | Aniso | Varied | |
| *K*-medoids based | 0.860 | 0.864 | 0.904 | 0.876 |
| K-nearest neighbors based ($N$ = 5, $C$ = 2) | 0.904 | 0.900 | 0.912 | 0.905 |

Algorithm 1. Object classification using K-NN based approach

Input:

*X* – feature matrix *n*m*, *n* – number of objects, *m* – number of features

*y* – labels vector of length *n*, *y[i]* = -1 if no label data for *i*-th object

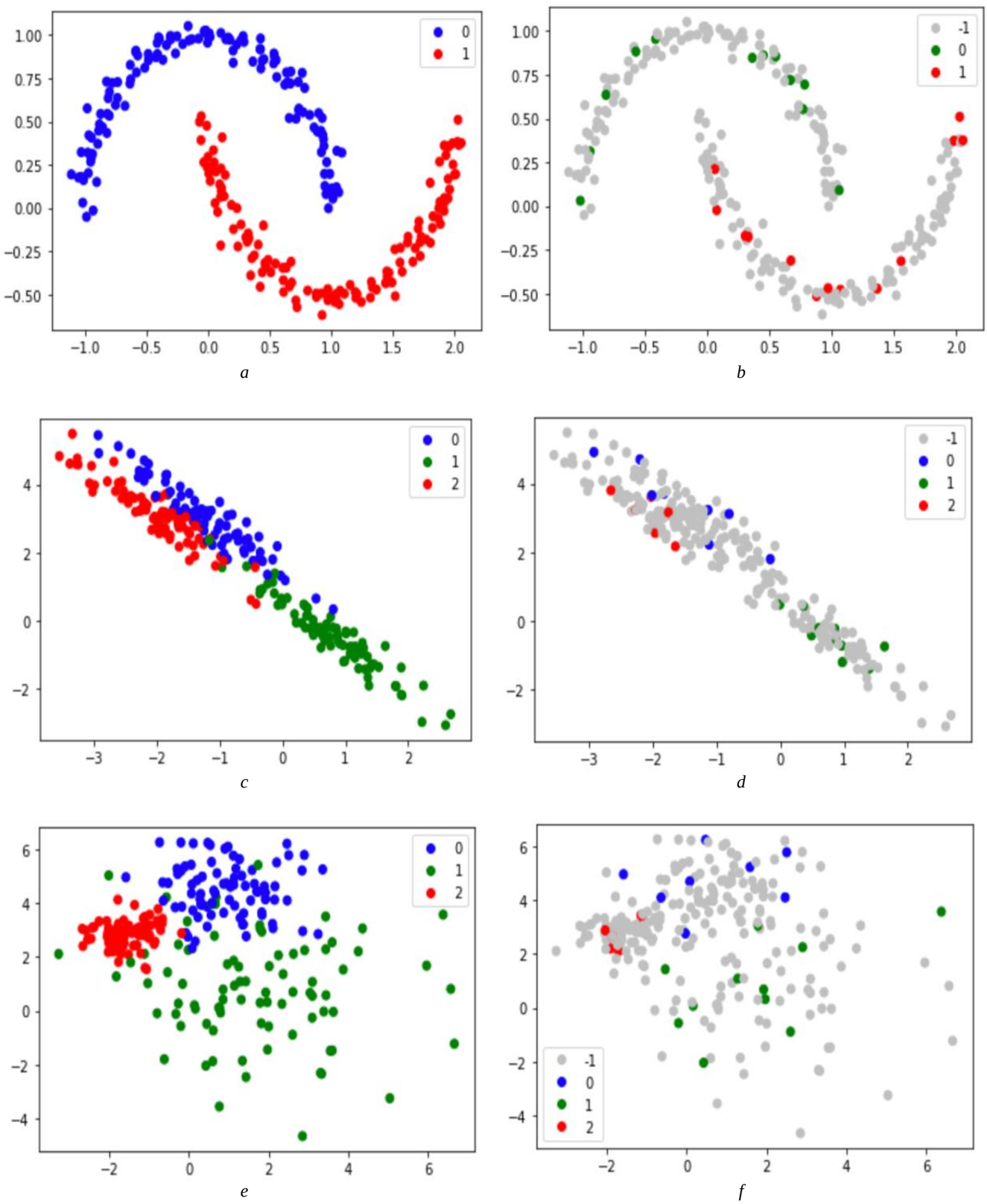
*K* – number of nearest points

*C* – weight of nearest class center

Output:

*y_predicted* – vector of length *n* with object labels

*y_predicted* ← empty list of length *n*
*unlabeled_idxs* ← list of indexes where *y* = -1
*labeled_idxs* ←list of indexes where *y* > -1
*center_coordinates* ← list of center coordinates for each class, calculated using available labels
random shuffle *unlabeld_idxs*
for *i* in *unlabeld_idxs* do
    *distances_i* ← distances from *i*-th object to each object with indexes in *labeled_idxs*
    argsort *distances_i*
    *nearest_idxs* ← indexes of first *K* elements from *distances_i*
    *classes_dist_i* ← distance from *i*-th object to each classes' center
    *neares_class_idx* ← index of nearest class to *i*-th object
    *cls_counts* ← list, where *j*-th element denote numbers of points belong to *j*-th class among *nearest_idxs*
    *cls_counts*[*neares_class_idx*] ← *cls_counts*[*neares_class_idx*] + *C* // add additional value for class with nearest center
    *label* ← argmax(*cls_counts*)
    *y_predicted*[*i*] ← *label*
end for
for *i* in *labeled_idxs* do
    *y_predicted*[*i*] ← *y*[*i*]
end for
return *y_predicted*




*a* *b*

*c* *d*

*e* *f*

Fig. 3. Datasets visualization; the legend shows classes' label, -1 – unlabeled point;
*a*, *b* – moons dataset, 2 classes, with non-convex and separable shapes;
*c*, *d* – aniso dataset, 3 classes, convex shape with same class variation, not separable;
*e*, *f* – varied dataset, 3 classes with a convex shape and different class variation, also not separable

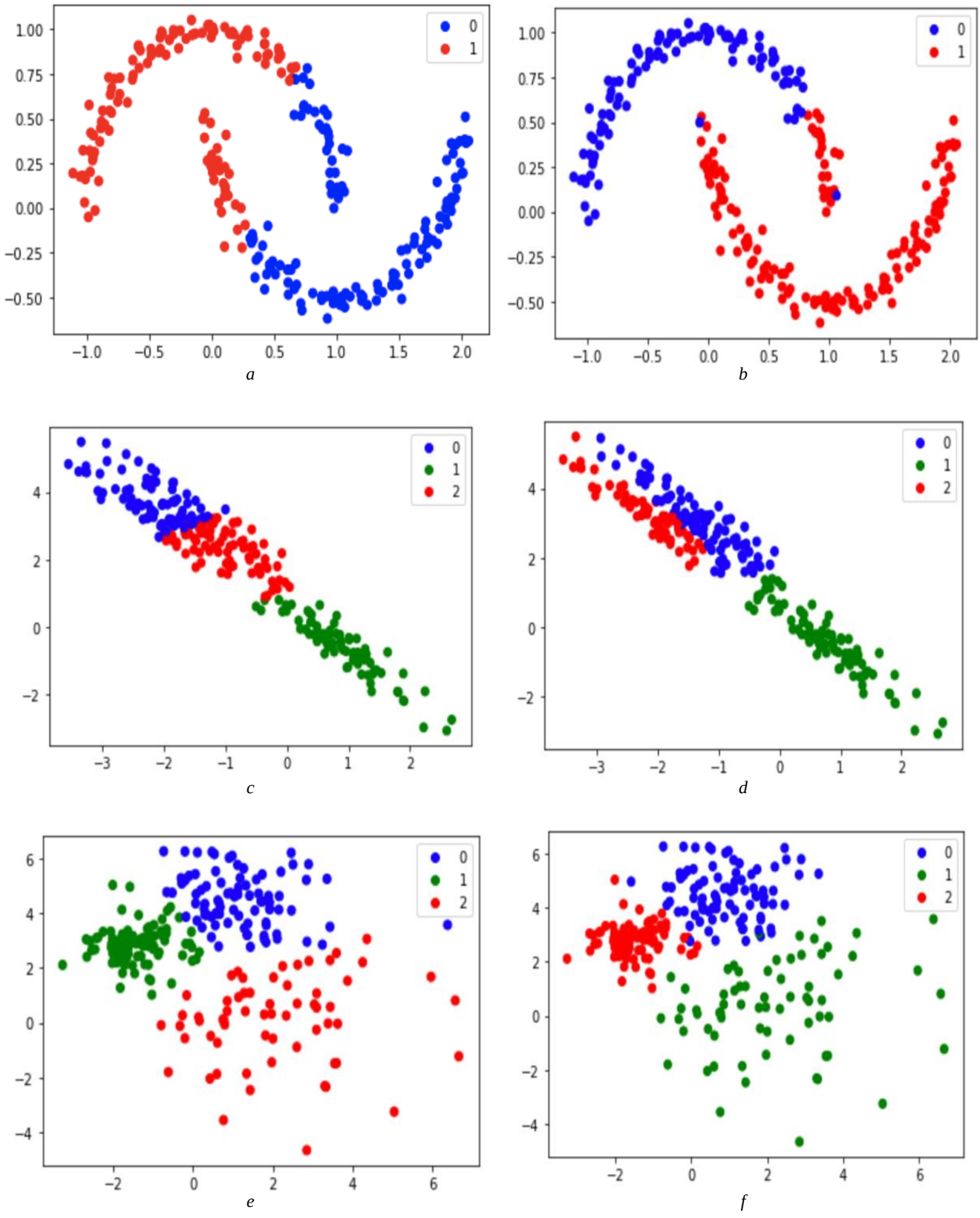



Fig. 4. Predicted labels visualization; *a*, *c*, *f* – unsupervised *K*-Medoids;
*b*, *d*, *f* – semi-supervised *K*-nearest neighbors based method

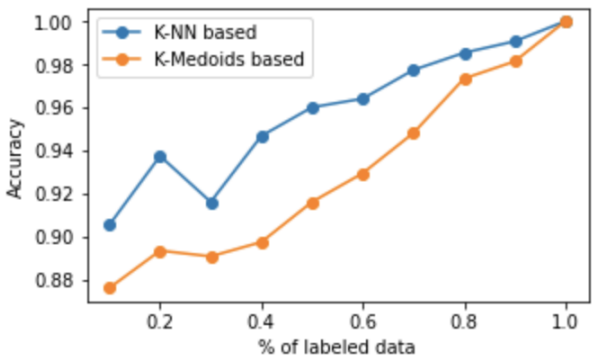



Fig. 5. Accuracy versus the quantity of labeled data comparison plot

**Conclusions.** In this study, we had shown that even small amounts of labeled data allow using of semi-supervised learning and improving accuracy. In addition, semi-supervised learning can improve algorithm performance too. Multiple approaches to semi-supervised learning were proposed, they are using a distance metric that considers available label information.

**References**


1. Hinton G., Sejnowski T. Unsupervised Learning: Foundations of Neural Computation. MIT Press, 1999. 391 p.
2. Lyubchyk L. M., Galuza A. A., Grinberg G. M. Semi-supervised Learning to Rank with Nonlinear Preference. *Recent Developments in Fuzzy Logic and Fuzzy Sets. Studies in Fuzziness and Soft Computing*. Lviv: Springer, 2019. Vol. 391. P. 73–102.
3. Basu S., Bilenko M., Banerjee A., Mooney. R. J. *Probabilistic semi-supervised clustering with constraints.* MIT Press, 2006. P. 73–102.
4. Jesper E., Holger H. *A survey on semi-supervised learning*. Available at: https://doi.org/10.1007/s10994-019-05855-6.
5. Bair E. *Semi-supervised clustering*. Available at: https://arxiv.org/pdf/1307.0252.pdf.
6. Kaufman, L., Rousseeuw P. J. *Finding groups in data: an introduction to cluster analysis.* New York: Wiley, 1990. 342 p.
7. Cover T., Hart P. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*. 1967. Vol. 13, no. 1. P. 21–27.
8. Huang T., Kecman V., Kopriva I. Kernel *Based Algorithms for Mining Huge Data.* New York: Springer, 2006. 208 p.
9. Vapnik V. N. *Statistical Learning Theory*. New York: Wiley, 1998. 768 p.
10. Wang J., Shen X., Pan W. Transductive Support Vector Machines, *Contemporary Mathematics.* 2007. Vol. 443. P. 7–19.
11. Rossum G. *Python programming language.* Available at: http://www.python.org.
12. Cournapeau D. *Scikit-learn. machine learning library for the Python programming language*. Available at: https://scikit-learn.org/stable/.


**References (transliterated)**


1. Hinton G., Sejnowski T. Unsupervised Learning: Foundations of Neural Computation. MIT Press, 1999. 391 p.
2. Lyubchyk L. M., Galuza O. A., Grinberg G. M. Semi-supervised Learning to Rank with Nonlinear Preference. *Recent Developments in Fuzzy Logic and Fuzzy Sets. Studies in Fuzziness and Soft Computing*. Lviv, Springer, 2019, vol. 391, pp. 81–103.
3. Basu S., Bilenko M., Banerjee A., Mooney. R. J. *Probabilistic semi-supervised clustering with constraints*. MIT Press, 2006, pp. 73–102.
4. Jesper E., Holger H. *A survey on semi-supervised learning*. Available at: https://doi.org/10.1007/s10994-019-05855-6.
5. Bair E. *Semi-supervised clustering*. Available at: https://arxiv.org/pdf/1307.0252.pdf.
6. Kaufman, L., Rousseeuw P. J. *Finding groups in data: an introduction to cluster analysis*. New York, Wiley, 1990. 342 p.
7. Cover T., Hart P. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*. 1967, vol. 13, no. 1, pp. 21–27.
8. Huang T., Kecman V., Kopriva I. Kernel *Based Algorithms for Mining Huge Data.* New York, Springer, 2006. 208 p.
9. Vapnik V. N. *Statistical Learning Theory*. New York, Wiley, 1998. 768 p.
10. Wang J., Shen X., Pan W. Transductive Support Vector Machines, *Contemporary Mathematics.* 2007, vol. 443, pp. 7–19.
11. Rossum G. *Python programming language*. Available at: http://www.python.org.
12. Cournapeau D. *Scikit-learn. machine learning library for the Python programming language*. Available at: https://scikit-learn.org/stable/.


*Поступила (received) 10.03.2021*

*Відомості про автора / Сведения об авторе / About the Author*


***Ямковий Клим Сергійович*** – магістр, Національний технічний університет «Харківський політехнічний інститут», аспірант кафедри комп'ютерної математики і аналізу даних; м. Харків, Україна; ORCID: https://orcid.org/0000-0001-9512-4150; e-mail: yamkovou@gmail.com

***Ямковой Клим*** – магистр, Национальный технический университет «Харьковский политехнический институт», аспирант кафедры компьютерной математики і анализа данных; г. Харьков, Украина; ORCID: https://orcid.org/0000-0001-9512-4150; e-mail: yamkovou@gmail.com

***Yamkovyi Klym*** – master, National Technical University "Kharkiv Polytechnic Institute", PhD student of the Department of Computer Mathematics and Data Analysis; Kharkiv, Ukraine; ORCID: https://orcid.org/0000-0001-9512-4150; e-mail: yamkovou@gmail.com