Methods of Classification and Clustering in Data Analysis

Chumachenko Aksinia
Product Analytics Team Lead, Simpals
Chisinau, Moldova
Email address: aksinia.chumachenko@999.md

Abstract— *This article provides a systematic overview of key classification and clustering methods that form the foundation of modern data analysis. It begins with a general introduction to supervised and unsupervised learning, illustrating the fundamental differences between classification tasks and those aimed at uncovering hidden structures. The following sections examine major classification approaches, including linear and nonlinear models, ensemble methods, and probabilistic algorithms. In the clustering section, the discussion focuses on how various algorithms (K-means, hierarchical clustering, and DBSCAN) detect complex data shapes differing in density and form. The author's contribution addresses the interpretation of results, comparative analysis, and practical visualization examples, offering a more meaningful application of these models. Such a comprehensive methodology and examples of use highlight the flexibility of machine learning and its suitability for solving a wide range of tasks related to large and heterogeneous datasets.*

Keywords— *Machine Learning, Classification, Clustering, Result Interpretation, Quality Assessment, Ensemble Methods, Data Science.*

I. Introduction

In today's scientific and industrial landscape, classification and clustering have become fundamental tools of data analysis, serving as a crucial link in tasks of intelligent information processing. The continuous growth in both the volume and variety of data, generated in research, business processes, and technological services, drives the urgent need for automated methods capable not only of extracting patterns but also of providing quantitatively and qualitatively sound conclusions for decision-making. For instance, banks employ advanced classifiers to assess credit risk and detect fraudulent transactions, the pharmaceutical industry actively uses clustering techniques to group biomarkers and personalize treatment approaches, and major internet companies apply similar algorithms in recommendation systems and advertising [1].

Despite the widespread use and theoretical foundation of core ideas, researchers and practitioners face numerous unresolved questions when choosing a specific classification or clustering method. Which parameters determine the performance of ensemble models? How robust are kernel-based approaches to outliers? How can one accurately select the number of clusters or assess clustering quality when the true structure of the data remains unknown? Attempts to answer these questions often encounter the lack of a universal solution, as well as existing differences regarding interpretability, computational complexity, and algorithmic stability in dealing with real-world, often incomplete or "noisy" datasets.

Moreover, there is a growing demand for research that aims to develop more coherent guidelines: when and why should logistic regression or Random Forest be employed, in which situations do ensemble methods outperform single models, and how can one correctly evaluate clustering outcomes given the diversity of available metrics?

While the standard strategy for classification and clustering covers virtually all stages of data analysis—from the initial detection of patterns to complex predictive analytics tasks in large-scale datasets [2]—there remains a noticeable lack of in-depth systematic overviews and comparisons of methods, especially in terms of their practical application to real-world data. The issue of interpretability holds a prominent place in contemporary studies, as creating an effective algorithm that yields accurate results is insufficient without a transparent decision-making mechanism. In medicine, this directly affects diagnostic responsibility, and in the banking sector, regulatory compliance [3].

This work aims to form a comprehensive understanding of classification and clustering methods, including their theoretical underpinnings, existing subtypes, underlying principles, and practical applicability, as well as to provide author's recommendations for their rational selection and interpretation.

1. General Theoretical Background

Data analysis in machine learning is traditionally divided into two domains: supervised learning and unsupervised learning [4,5]. The former assumes the presence of a target variable (label), while the latter focuses on discovering the internal structure of a dataset without any additional information about correct answers.

Supervised learning addresses tasks related to predicting class labels (classification) or numerical values (regression). The learning process uses training pairs (x,y), where x represents input features and y represents the target output. The objective is to construct a function f(x) that minimizes prediction error on new, unseen data.

Unsupervised learning, on the other hand, analyzes unlabeled data to identify underlying patterns and structures. Common applications include clustering (grouping similar objects), dimensionality reduction (such as PCA and t-SNE for data visualization), and association rule mining [4,5]. These methods reveal natural groupings and relationships within data without prior knowledge of correct outcomes.

From a mathematical standpoint, supervised learning is formally described by optimizing a loss function L(f(x),y). For support vector machines (SVMs), this may involve maximizing the margin between classes, while for linear models such as logistic regression, it often entails minimizing cross-entropy. In contrast, unsupervised learning lacks a clear notion of a “correct” answer, typically focusing on maximizing intra-cluster similarity (or minimizing inter-cluster distances) or reducing dimensionality with minimal information loss.

Practical applications help clarify the difference:

- *Finance:* Banks employ supervised learning (classifiers) to identify fraudulent transactions and unsupervised methods to detect unusual patterns in financial flows.
- *Medicine:* Classification helps diagnose diseases based on labeled symptom data, while clustering reveals previously unknown disease subtypes by grouping patients with similar pathological patterns.
- *Marketing:* Supervised models optimize advertisements by predicting conversion rates from past campaign data, while unsupervised methods segment customers into natural groups based on behavior patterns.

Below is Table 1, providing a brief summary of the main distinctions.

TABLE 1. Differences in Supervised and Unsupervised Learning [4,5]

Characteristic	Supervised Learning	Unsupervised Learning
Presence of Target Label	Yes (classes, regression)	No
Main Objective	Predict ŷ	Discover groups/patterns
Typical Tasks	Classification, Regression	Clustering, Dimensionality Reduction
Example Metrics	Accuracy, Precision, Recall, ROC AUC	Silhouette Score, Sum of Squared Errors (SSE), Adjusted Rand Index (ARI)

Thus, the fundamental distinction between supervised and unsupervised learning determines the choice of methods and analysis strategies. The following sections will examine in detail the two key types of learning tasks: classification (the logic of building supervised models) and clustering (a primary example of unsupervised methods).

2. Classification

Classification, as a key representative of the supervised learning paradigm, aims to determine the membership of objects in predefined categories. Formally, classification involves finding a mapping f: $R^n \rightarrow Y$, where $x \in R^n$ represents a feature vector and Y is a finite set of classes. The quality of this mapping is evaluated through a loss function L(f(x), y), such as cross-entropy or classification error. The objective is to minimize this function on the training data while maintaining the model's ability to generalize to unseen instances. Closely related to this concept, the model’s generalization capacity defines how reliably the algorithm will handle instances not encountered during training [6].

To illustrate different approaches in classification, it is useful to consider a hypothetical two-dimensional problem with two overlapping classes. Suppose objects of the first class occupy the upper-right region of a plot, while those of the second class are in the lower-left (Fig. 1). Various algorithms build the separating boundary differently. The simplest linear boundary (e.g., logistic regression) may fail to capture complex structures, but it is easily interpretable. More flexible methods, such as decision trees and their ensembles, can form nonlinear separating surfaces but may increase the risk of overfitting. This trade-off between model complexity and generalization is widely documented in classification literature [7].

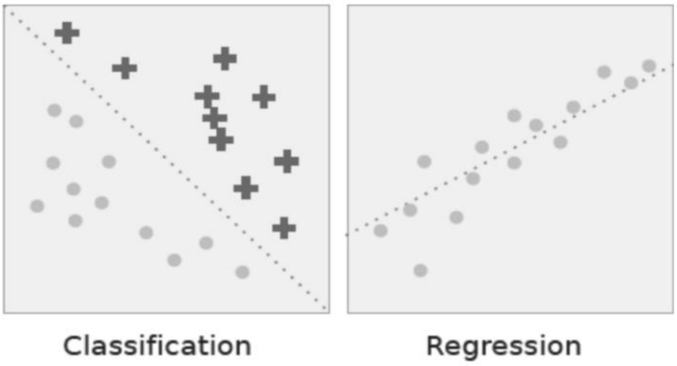

Figure 1. A schematic illustration [7]

Classification vs. regression. In classification the dotted line represents a linear boundary that separates the two classes; in regression, the dotted line models the linear relationship between the two variables.

Below is a brief overview of the main classification algorithms, considering their characteristic features and areas of application.

The earliest classification approaches assumed linear separability, giving rise to simple but still relevant models. Logistic regression remains a fundamental approach, creating decision boundaries using the logistic function to estimate class membership probabilities. Its primary advantages include interpretable coefficients and effective regularization options, though it cannot capture nonlinear relationships in data.

Nonlinear algorithms form a broad category. Decision trees recursively partition the feature space into simpler regions, with the tree’s leaves representing final classes. However, individual trees are prone to overfitting, and tree depth directly influences the balance between accuracy and generalization. Tree ensembles (Random Forest, Gradient Boosting) combine multiple base trees into a single model. Random Forest uses bootstrapping and aggregation (bagging) of multiple trees to reduce variance and improve noise resistance. Gradient Boosting builds trees sequentially, with each tree focusing on correcting previous errors, typically achieving high accuracy but requiring careful parameter optimization

Among the classic “non-parametric” methods is the k-Nearest Neighbors (k-NN) algorithm. It assigns a new object’s class based on the majority vote among its k closest training instances, according to a chosen metric. The advantage is that no explicit “training” is needed—everything relies on the distances between objects. The drawback is the substantial computational cost when the dataset grows and sensitivity to noisy features.

Another important algorithm is the Support Vector Machine (SVM). It constructs a hyperplane or set of hyperplanes that optimally separate the classes and maximize the margin between them. By employing nonlinear kernels (radial, polynomial, etc.), the method can handle complex decision boundaries. SVMs are fairly stable but require careful kernel selection and parameter tuning.

Finally, the Naive Bayes method, based on Bayes' theorem and the assumption of feature independence, deserves mention. Despite the roughness of this assumption, Naive Bayes often performs competitively, especially in text classification tasks (e.g., spam detection), and it trains and predicts very quickly [6,9].

For a concise comparison of key characteristics, see Table 2.

TABLE 2. Key Characteristics of Classification Algorithms [6,7,9]

Algorithm	**Type of Decision Boundary**	**Key Advantages**	**Main Disadvantages**
Logistic Regression	Linear	High interpretability; Efficient training	Limited to linear relationships
Decision Trees	Nonlinear (rules)	Transparent decision rules; Handles mixed data types	Prone to overfitting
Random Forest	Ensemble of trees	Robust performance; Handles missing data	Limited interpretability; Memory intensive
Gradient Boosting	Sequential tree ensemble	High accuracy; Feature importance ranking	Complex parameter tuning; Training time
k-NN	Distance-based	No training phase; Nonparametric	Computationally expensive online
SVM	Optimal hyperplane	Effective in high dimensions; Robust separation	Kernel selection complexity; Scaling issues
Naive Bayes	Linear (assuming independence)	Fast training and inference; Memory efficient	Strong independence assumption

These classification methods, while differing in their underlying approaches and characteristics, share the fundamental goal of achieving optimal class separation while maintaining generalization capability. The selection of an appropriate method depends on several key factors: data characteristics (dimensionality, noise level, feature types), computational constraints (training time, memory requirements, inference speed), and application requirements (interpretability, accuracy thresholds, deployment environment).

3. Clustering

Clustering falls under unsupervised learning methods, as it does not require predefined labels for the data points. Its aim is to partition a dataset into groups (clusters) so that objects within each cluster are as similar as possible, while objects from different clusters differ significantly. Unlike classification, there is no "correct answer" here; the focus lies primarily on the data's internal structure. This approach is particularly important in tasks such as market segmentation in marketing, gene grouping in bioinformatics, or searching for similar patterns in large image databases.

The foundation of clustering rests on distance or similarity metrics between data points. For numerical features, Euclidean distance (d = ||xi - xj||) serves as the standard metric. However, in high-dimensional tasks or those involving specialized data formats (such as text), alternative metrics like cosine similarity may be preferable. Determining the optimal cluster structure (how many clusters and their arrangement) is not always straightforward. Selecting the number of clusters in methods like K-means can be done via heuristic approaches such as the "elbow method" or silhouette analysis, though the choice remains largely subjective. Further complicating the matter is the fact that clusters may have irregular shapes or varying densities, making it unwise to rely solely on "spherical" assumptions [9].

Classical overviews highlight several fundamental approaches that differ in how they group objects [9]:

1. *K-means.* This is the most popular algorithm due to its simplicity and ease of implementation. The idea involves randomly initializing k centroids, then iteratively updating their positions by assigning objects to the nearest centers. The process continues until stabilization when the centroids stop changing.

Advantages:

- Computational efficiency (O(nkd) per iteration);
- Simple implementation and interpretation;
- Guaranteed convergence to local optimum.

Limitations:

- Requires pre-specified number of clusters (k);
- Assumes spherical cluster shapes;
- Sensitive to initial centroid positions;
- May converge to suboptimal solutions.

2. Hierarchical Clustering. This method constructs a tree-like structure called a dendrogram. In the agglomerative (bottom-up) variant, the algorithm starts with isolated points and gradually merges the closest clusters until only one remains. In the divisive (top-down) variant, it begins with the entire dataset as one cluster and progressively splits it.

Advantages:

- No predefined cluster number is required;
- Provides hierarchical data structure visualization;
- Supports multiple distance metrics and linkage criteria.

Limitations:

- Time complexity $O(n^3)$ for naive implementation;
- Memory requirements $O(n^2)$;
- Sensitive to linkage criterion selection (single, complete, average).

3. DBSCAN (Density-Based Spatial Clustering of Applications with Noise). DBSCAN is density-based: objects belong to the same cluster if they are close enough to a "cluster core." The parameters ε (eps) and min_samples define the neighborhood radius and the minimum number of points required to form a cluster.

Advantages:

- Discovers arbitrary-shaped clusters;

- Automatically handles noise/outliers;
- No preset cluster number needed;
- Time complexity O(n log n) with spatial indexing.

Limitations:

- Parameter selection challenges;
- Struggles with varying density clusters;
- Memory requirements for distance calculations [9].

Below is a conceptual two-dimensional diagram that can depict several groups of points with varying shapes and densities. By coloring the regions differently, one can illustrate how each algorithm interprets the data structure (Fig. 2) [8].

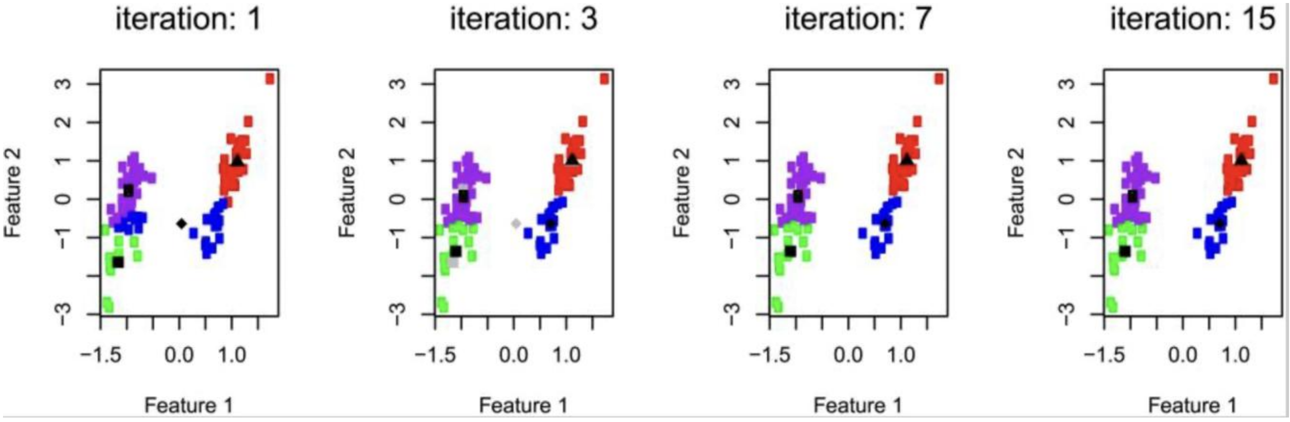

Figure 2. Example of Clustering [8]

K-means will attempt to partition these points into roughly spherical clusters of similar shape and size. If k=2 is chosen, it will likely group the compact set into one cluster and the elongated region into another, ignoring outliers or distributing them to the nearest centers.

Hierarchical clustering builds a dendrogram, and a vertical “cut” at the desired level determines the number of resulting groups. Cutting at two clusters might recreate a division close to that of K-means. Cutting at three or four clusters could isolate noisy points into separate small groups or merge them with similar segments.

DBSCAN identifies clusters based on density patterns, naturally grouping dense regions while marking isolated points as noise. Its ability to discover clusters of arbitrary shapes and automatic noise detection offer advantages over K-means, despite requiring careful parameter selection for ε and min_samples.

While such a diagram does not aim for strict accuracy, it effectively illustrates the differences among these algorithms. The choice of a particular method depends on requirements related to cluster shape, dataset scale, and noise levels.

In the end, clustering results in partitioning the dataset into clusters that can be interpreted as potentially “related” groups. The ultimate success of this partitioning depends on chosen evaluation metrics, feature characteristics, and hyperparameter settings. Compared to classification, the outcome does not rely on external labels, placing greater responsibility on the researcher to interpret the results and draw meaningful conclusions.

4. Quality Assessment and Practical Examples

Models used in classification and clustering tasks produce specific outputs—either a predicted class or an assigned cluster. However, without proper interpretation and objective quality assessment, these outputs remain just numbers or labels, lacking substantial analytical value. The analyst must be able to read the results, understand their limitations, and select suitable metrics for comparison and diagnostics. Moreover, practical implementation examples are essential to verify a method's reproducibility and effectiveness.

Interpreting classification models often involves three aspects: understanding the model’s logic (which features are most important and how decisions are formed), evaluating key metrics, and analyzing behavioral patterns (where and why the model might fail). Logistic regression provides coefficients for each feature, simplifying the explanation of decisions. Decision trees can be examined by tracing the path from the root to the leaf—this is especially valuable in medical diagnostics, where it is crucial for a physician to understand the rationale behind a given diagnosis. More complex ensemble algorithms (Random Forest, Boosting) or SVMs are often less interpretable directly. For these, interpretation techniques such as SHAP (SHapley Additive exPlanations) or LIME (Local Interpretable Model-agnostic Explanations) are increasingly employed to quantitatively estimate each feature's contribution to the final prediction.

Clustering requires a dual interpretation strategy: on one hand, the quality of the discovered groups should be examined using internal metrics (e.g., silhouette index, Sum of Squared Errors [SSE], Davies–Bouldin index); on the other hand, one must consider the domain-specific logic of grouping. Since clusters lack external validation labels, the researcher must meaningfully describe the resulting groups. For K-means, analyzing the mean feature values within each cluster (its center) can reveal the defining characteristics of each segment. Hierarchical clustering allows step-by-step examination of merging stages—a dendrogram shows the nested structure of clusters and may suggest which hierarchical level yields the most interpretable segmentation. In addition to identifying clusters, DBSCAN labels outliers as anomalies, so if the data truly contain unusual patterns, the interpretation can focus on understanding the nature of these isolated objects.

Classification model quality assessment often involves a standard set of metrics (Accuracy, Precision, Recall, F1-score, ROC AUC), whose selection depends on the task's specifics. In financial fraud detection, Recall is critical to avoid missing fraudulent activities. In spam filtering, high precision is desirable to minimize blocking legitimate emails. Clustering, given the lack of a reference solution, more frequently relies on internal metrics (silhouette index, Davies–Bouldin) that measure cluster density and separation. When partial external information is available, metrics like Adjusted Rand Index (ARI) or Normalized Mutual Information (NMI) can be used if some objects have at least partial labeling.

Even after choosing a metric, it is important to recognize that no single indicator provides a complete picture. In real-world scenarios (product analytics, medicine, social studies), practitioners often establish a comprehensive measurement framework. For example, they may consider not only the F1-score but also the economic or practical impact of implementing the model, response times in operational systems, or the ease of explaining results to decision-makers.

A visual example could involve comparing several classifiers trained on a hypothetical dataset for spam detection. Suppose we have Logistic Regression, Random Forest, and SVM. For simplicity, assume we have already split the data into training and test sets, tuned hyperparameters, and obtained the following test metrics (Table 3).

TABLE 3. Test Set Metrics

Model	Accuracy	Precision (spam)	Recall (spam)	F1-score (spam)	ROC AUC
Logistic Regression	0.92	0.88	0.76	0.82	0.93
Random Forest	0.95	0.90	0.85	0.87	0.96
SVM (RBF)	0.93	0.87	0.81	0.84	0.94

The classifiers are evaluated using standard performance metrics: Accuracy (overall correct predictions), Precision (proportion of correct spam identifications), Recall (proportion of actual spam detected), F1-score (harmonic mean of Precision and Recall), and ROC AUC (Area Under the Receiver Operating Characteristic Curve). We see that Random Forest performs best on most metrics, although Logistic Regression is slightly easier to interpret. Depending on the objective, an analyst might prefer the model offering higher accuracy and F1-score, or opt for greater interpretability.

To enhance clarity, ROC curves are often plotted to compare models. Below is a conceptual diagram (Fig. 3) showing three ROC curves and their associated AUC values. A higher AUC indicates that the model better distinguishes spam from non-spam at various threshold settings.

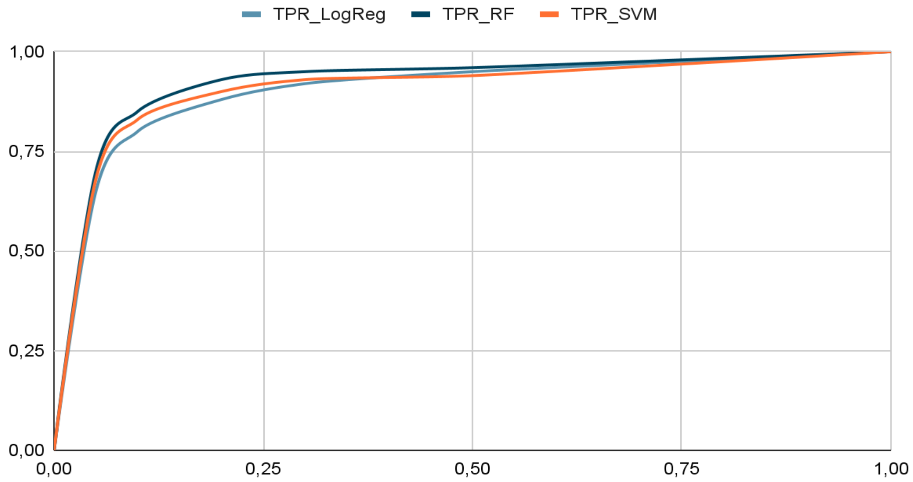

Figure 3. Example of ROC Curves for Model Comparison

Logistic Regression (AUC=0.93); Random Forest (AUC=0.96); SVM (RBF) (AUC=0.94).

Each point on the ROC curves corresponds to different classification thresholds. The best curves approach the top-left corner of the plot, and their AUC values are higher. In this case, Random Forest achieves the highest AUC = 0.96.

Such visualizations help compare the quality of multiple methods and facilitate the selection of a specific model for deployment. Furthermore, if we were addressing a segmentation task (clustering), we could similarly create summary tables or scatter plots colored by cluster membership (e.g., K-means vs. DBSCAN results) and evaluate indices like the silhouette score.

Thus, combining summary tables of metrics with visual aids (ROC curves, scatter plots, dendrograms for hierarchical clustering, etc.) allows not only for metric calculation but also for a visual assessment of which model best meets the task's requirements and a better understanding of each approach's strengths and weaknesses.

II. CONCLUSION

The findings confirm that the choice of a particular classification or clustering method is largely determined by the data's structural characteristics, available resources, and the analyst's objectives. While algorithmic performance is crucial, practical considerations often take precedence—interpretability may be more valuable than maximizing accuracy, particularly in domains like medical diagnostics. Different data challenges, such as noise and high dimensionality, require specific methodological approaches. The result is a holistic understanding of the applicability boundaries of various algorithms, helping to develop more reliable and meaningful solutions when analyzing large data sets. A judicious combination of metrics, result visualization, and awareness of potential pitfalls forms the foundation for effectively deploying such solutions in real-world scenarios, whether for fraud detection, medical diagnostics, or marketing segmentation. Through such a comprehensive evaluation, machine learning methodology serves not just as a tool, but also as a conceptual framework for interdisciplinary developments.

REFERENCES

1. Patil P. H. et al. Analysis of different data mining tools using classification, clustering and association rule mining //International Journal of Computer Applications. - 2014. - Vol. 93. - No. 8.
2. Lapina M. A. et al. Application of machine learning technologies for web attack detection // Cybersecurity Issues. - 2024. - Vol. 8822. - No. 4. - P. 5332.
3. Mohammed M. A. et al. The effectiveness of big data classification control based on principal component analysis //Bulletin of Electrical Engineering and Informatics. - 2023. - Vol. 12. - No. 1. - P. 427-434.
4. Alloghani M. et al. A systematic review on supervised and unsupervised machine learning algorithms for data science //Supervised and unsupervised learning for data science. – 2020. – P. 3-21.
5. Kondratenok E. V., Makarenya S. N. Machine learning as a tool for data analysis. – 2022.
6. Castelli M., Vanneschi L., Largo Á. R. Supervised learning: classification //por Ranganathan, S., M. Grisbskov, K. Nakai y C. Schönbach. – 2018. – Vol. 1. – P. 342-349.
7. Sarker I. H. Machine learning: Algorithms, real-world applications and research directions //SN computer science. – 2021. – T. 2. – №. 3. – C. 160.
8. Rodriguez M. Z. et al. Clustering algorithms: A comparative approach //PloS one. – 2019. – T. 14. – №. 1. – C. e0210236.
9. Ahuja R. et al. Classification and clustering algorithms of machine learning with their applications //Nature-inspired computation in data mining and machine learning. – 2020. – P. 225-248.