The Scalar Metric of Classification Algorithm Choice in Machine Learning Problems Based on the Scheme of Nonlinear Compromises

Igor Puleko[1], Oleksandra Svintsytska[1], Victor Chumakevych[2], Vadym Ptashnyk[3] and Yuliia Polishchuk[4]

[1] *Zhytomyr Polytechnic State University, 103, Chydnivska srt., Zhytomyr, 10005, Ukraine*
[2] *Lviv Polytechnic National University, 12, S. Bandery str., Lviv, 79013, Ukraine*
[3] *Lviv National Agrarian University, 1, V.Velykoho str., Dubliany-Lviv, 80381, Ukraine*
[4] *National Aviation University, 1, Liubomyra Huzara ave., Kyiv, 03058, Ukraine*

Abstract

A classic example of machine learning methods is machine classification algorithms. At present, a large number of machine classification methods have been developed, which are offered in the form of ready-made software. The presence of a large number of machine classification algorithms raises the problem of choosing the best algorithm for solving a particular task. This problem is also complicated by the ambiguity with the choice of quality indicators since for the analysis of the quality of the classification there exist a number of indicators (metrics) of the classification. It is difficult for an inexperienced user to understand and choose his priorities. However, the problem of evaluating the classification algorithm's quality can be considered a problem of multicriteria decision-making. It is proposed to evaluate the algorithm's quality by means of one scalar indicator obtained by convolution of other indicators by a nonlinear scheme of compromises.

Keywords

Communications, software, machine learning, classification evaluation metrics, accuracy, recall, precision, F1, AUC_ROC, nonlinear scheme of compromises.

1. Introduction

In the sphere of information technology (IT), machine learning (ML) methods, which are used to solve a number of applied problems, have become widely used. In essence, ML is a class of methods of artificial intelligence, the characteristic feature of which is not a direct solving the problem but learning through the use of solutions to many similar problems. Typically, such methods use mathematical statistics, probability theory, mathematical analysis, optimization methods, numerical methods, graph theory, and various techniques for working with data in digital form [1].

Among the existing methods of machine learning, the most researched and developed are the methods of machine classification, which belong to the controlled type of learning or learning with a teacher (supervised learning) [2]. A classification task is a task in which there are many objects (situations), divided into classes in some way. Also, a finite set of objects is defined, for which it is known to which classes they belong. This set is called a sample (training sample). The class belonging of other objects is unknown. We need to construct an algorithm that can classify an arbitrary object (specify the number or name of the class to which the object belongs) from the initial set.

COLINS-2022: 6th International Conference on Computational Linguistics and Intelligent Systems, May 12–13, 2022, Gliwice, Poland.
EMAIL: pulekoigor@gmail.com (I. Puleco); sasha_1904@ukr.net (O. Svintsytska); chumakevich@ukr.net (V. Chumakevych); ptashnykproject@gmail.com (V. Ptashnyk); polishchuk.yu.ya@gmail.com (Yu. Polishchuk)
ORCID: 0000-0001-8875-017X (I. Puleco); 0000-0002-2613-2437 (O. Svintsytska); 0000-0002-5773-393X (V. Chumakevych); 0000-0002-1018-1138 (V. Ptashnyk); 0000-0002-0686-2328 (Yu. Polishchuk)

Algorithms that solve the classification problem have been known for a long time, and in mathematical statistics, they are also called problems of discriminant analysis [3]. In ML, the problem of classification is solved, in particular, by means of a large number of algorithms, including those with the application of methods of artificial neural networks.

Today, nearly all leading IT companies, to some extent, develop, use, or provide as a service various methods and algorithms of ML. For example [4, 5], Microsoft's Azure Machine Learning Studio has more than a dozen classification algorithms, each of which can perform the set task (see Figure 1). The variety of supply raises the problem of choice. Which of the algorithms is better to choose to perform a task? The answer to this question is far from unambiguous since the existing metrics for evaluating the classification quality do not provide an unambiguous result. It is especially challenging for beginners to make such a choice.

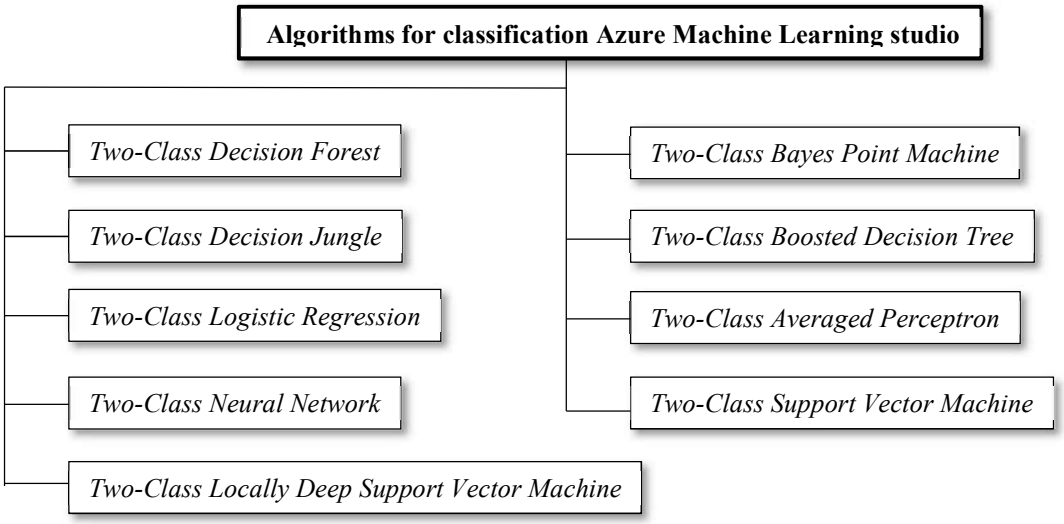

Figure 1: Classification Algorithms Azure Machine Learning Studio

2. Related Works

The most common and frequently used classification algorithms today are [6]: Naive Bayes, Decision Trees, Logistic Regression, K-Nearest Neighbors, Support Vector Machines and others.

According to classical theory, several indicators (classification metrics) are used to evaluate the quality of such classification algorithms. Consider them in more detail.

The confusion matrix is a table used to describe the effectiveness of a classifier. It is usually extracted from a test data set for which basic true values are known [7].

Here the results of assignment to each class are analyzed and the share of incorrectly assigned classes is determined. In the process of constructing the above table, we are dealing with several key metrics that play a very important role in machine learning.

For the classification problem, given the actual label and the predicted label, the first thing we can do is divide our samples into 4 segments [7]:

- True Positive (TP): actual = 1, predicted = 1;
- False Positive (FP): actual = 0, predicted = 1;
- False Negative (FN): actual = 1, predicted = 0;
- True Negative (TN): actual = 0, predicted = 0.

Table 1
Classification error matrix

	y = 1	y = 0
a(x) = 1	TP	FP
a(x) = 0	FN	TN

A number of other characteristics are built on the basis of this matrix (Table 1). Consider each of them in more detail.

2.1. Accuracy

Accuracy [8] is the proportion of accurate predictions relative to the total number of predictions, i.e., it is the probability that the class will be predicted correctly (1).

$$\frac{TP + TN}{TP + TN + FP + FN}. \quad (1)$$

Thus, accuracy is the fraction of the correct answers of the algorithm.

Although accuracy is a quick and informative indicator of model performance, we cannot rely on it alone. This is due to the fact that it hides the presence of a shift in the model, which is common if the data set is unbalanced, i.e., the negative aspects are much more than the positive ones, or vice versa. That is, this metric is useless in problems with unequal classes, which, as an option, can be corrected using sampling algorithms. Sampling (data sampling) is a method of adjusting the training sample in order to balance the distribution of classes in the original data set [9].

2.2. Precision

Precision is the proportion of the correct answers of the model within the class, i.e., the proportion of objects that really belong to this class relative to the total number of objects that the system has assigned to this class [8].

$$Precision = \frac{TP}{TP + FP}. \quad (2)$$

The introduction of precision does not allow us to assign all objects to one class since in this case we have an increase in the level of FP.

2.3. Recall

Recall is the share of true positive rate (TPR) [8]. Recall shows what share of objects that actually belong to the positive class we have predicted correctly. In other words, this is the proportion of options classified as positive that actually proved to be positive.

$$Recall = \frac{TP}{TP + FN}. \quad (3)$$

Recall demonstrates the ability of the algorithm to detect this class in general.

2.4. F-score

Precision and recall do not depend on the relationship of classes (as opposed to accuracy) and, therefore, can be used in unbalanced samples. Often in real practice, the task is to find the optimal (for the customer) balance between these two metrics. It is obvious that the higher the precision and recall, the better. However, in real life, maximum precision and recall are unattainable simultaneously, and, therefore, we have to search for some balance. Thus, it would be convenient to have some metrics that would combine information about the precision and recall of our algorithm. In this case, it will be easier for us to decide which implementation to launch into production. The F-score serves precisely these needs.

The F-score is the harmonic mean between precision and recall. It tends to zero if accuracy or completeness tends to zero.

$$F1 = \frac{2 \times precision \times recall}{precision + recall}. \quad (4)$$

This formula gives the same weight to precision and recall, so the F-score will fall equally with decreasing precision and recall. It is possible to calculate the F-score by giving different weights to precision and recall if you consciously prioritize one of these metrics when developing an algorithm.

The F-score is a good candidate for a formal classifier quality evaluation metric. It reduces to one value these two basic metrics: precision and recall. Having the F-score is much easier to answer the question: "Has the algorithm changed for the better or not?"

2.5. ROC-curve

Receiver operating characteristics (ROC) curve is used to analyze the behavior of classifiers at different thresholds [10]. ROC-curve allows us to consider all threshold values for this classifier. It demonstrates the proportion of false positive rate (FPR) compared to the proportion of true positive rate (TPR) (see Figure 2).

$$TPR = \frac{TP}{TP + FN} = Recall\,, \tag{5}$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}\,. \tag{6}$$

The share of FPR is the proportion of negative samples that were incorrectly classified as positive.

$$FPR = 1 - TNR\,, \tag{7}$$

where TNR is the share of true negative rate (TNR), which is the proportion of negative samples that were correctly classified as negative.

The TNR fraction is also called specificity. Thus, the ROC-curve depicts sensitivity, i.e., recall, compared to the difference: 1 - specificity.

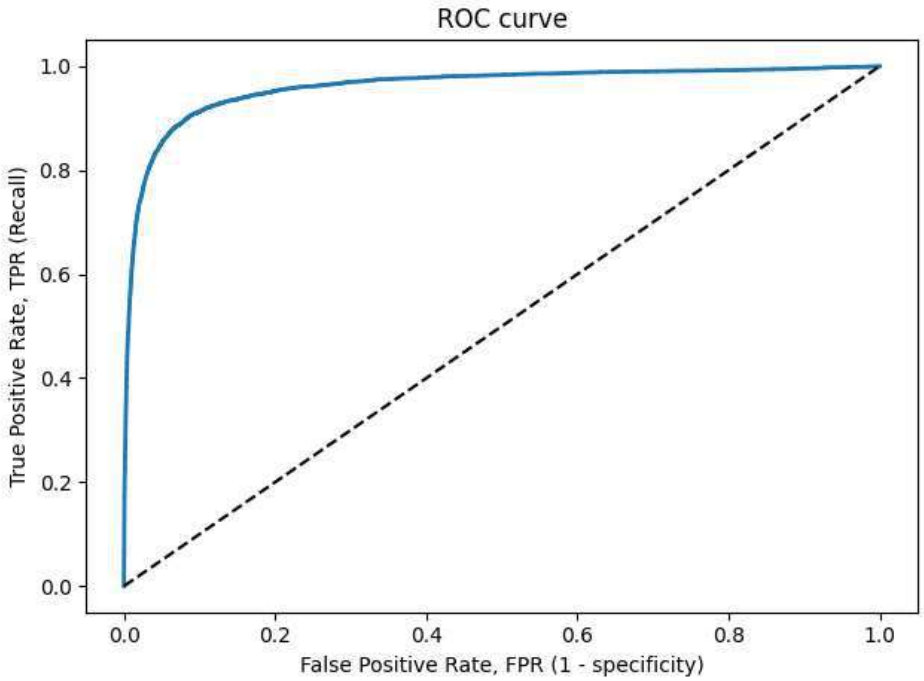

Figure 2: ROC-curve

The scalar indicator that follows from this indicator and allows us to compare classifiers is the value of the area under the curve (AUC). A perfect classifier will have an area under the ROC-curve (AUC-ROC) equal to 1, while a random classifier will have an area of 0.5.

The ROC chart helps us decide where to impose a classification threshold for maximizing a true positive rate or minimizing a pseudo-positive rate, which is ultimately a business decision.

The scalar indicator that follows from this indicator and allows us to compare classifiers is the value of the area under the curve (AUC). A perfect classifier will have an area under the ROC-curve (AUC-ROC) equal to 1, while a random classifier will have an area of 0.5.

The ROC chart helps us decide where to impose a classification threshold for maximizing a true positive rate or minimizing a pseudo-positive rate, which is ultimately a business decision.

2.6. PR-curve

The precision-recall curve determines the sensitivity to the ratio of classes. If the positive class is significantly smaller, the AUC-ROC may provide an inadequate estimate of the algorithm's quality, as it measures the proportion of incorrectly accepted objects in relation to the total number of negative ones.

We can eliminate this problem with unbalanced classes by passing from the ROC-curve to the precision-recall (PR) curve. The PR-curve is determined similarly to the ROC-curve; the only difference is that on the axes, we lay not FPR and TPR but recall (abscissa) and precision (ordinate). The criterion for the quality of the family of algorithms is the area under the PR-curve (AUC-PR).

The quality indicators of the classification considered here are far from all. Categorical Crossentropy or Log Loss / Binary Crossentropy and others may also be used in specific cases [7]. Even from the above list, it can be seen that comparing classification algorithms and choosing the best one is a difficult task, especially for beginners.

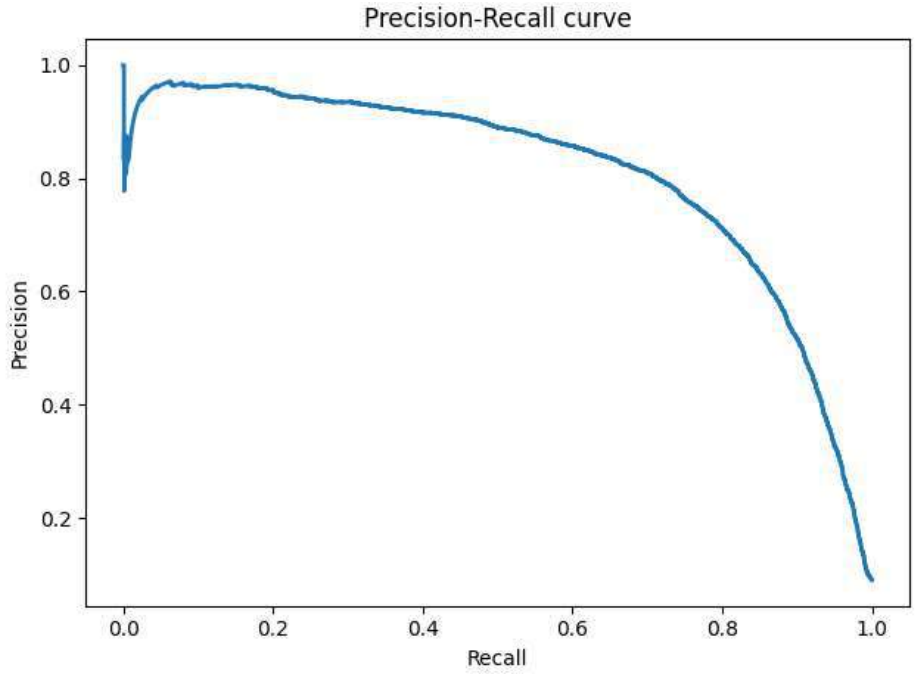

Figure 3: PR-curve

3. Methods

Quite often, when using different classification algorithms, similar quality indicators are obtained and it is difficult for a user to choose one of them. This is especially true for the criterion of "highest efficiency", which is calculated for each system separately and depends on business objectives. This problem can be considered as a multicriteria optimization problem [11].

In practice, such multicriteria problems are solved quite successfully, but a strict mathematical solution of multicriteria optimization problems still does not exist. In practice, several approaches are used, each of which has advantages and disadvantages. Here the authors propose to apply the method for solving multicriteria problems based on a nonlinear scheme of compromises presented in work by Voronin A. M. [12] having the form:

$$x^* = \arg\min \sum_{i=1}^{n} I_{mi}[I_{mi} - I_i(x)]^{-1}. \quad (8)$$

where I_{mi} is the upper limit for the partial criterion I_i.

If necessary, we can introduce the weight coefficients C_i into the nonlinear convolution (8) [13].

$$x^* = \arg\min \sum_{i=1}^{n} C_i I_{mi}[I_{mi} - I_i(x)]^{-1}. \quad (9)$$

The introduction of coefficients allows us to give preference to one or another criterion being better adapted to the specific business task.

Since the best quality of the classification algorithm is needed, the efficiency criterion must be maximized, and then the calculation formula takes the following form:

$$NSC = \arg\max \sum_{i=1}^{n} I_{mi}[I_{mi} - I_i(x)]^{-1}, \quad (10)$$

where NSC is the scalar indicator on a nonlinear scheme of compromises.

As partial criteria of the classification quality it is suggested to use known indicators of quality: accuracy, recall, precision and F1. Then the calculation formula has the final form:

$$NSC = \arg\max\left(\frac{1}{1-Acc} + \frac{1}{1-Pr} + \frac{1}{1-Rec} + \frac{1}{1-F1}\right) \quad (11)$$

where Acc is accuracy; Pr is precision; Rec is recall; $F1$ – F1-score.

The obtained scalar number will not have any physical meaning. Its values can also vary from dozens to dozens of thousands. The highest scalar value of the indicator NSC will determine the best algorithm for implementing a particular classification task.

The advantages of the method of nonlinear compromise scheme [14] are, first of all, that this method is quite simple in terms of computational costs and allows us to obtain solutions from the Pareto set taking into account constraints on the principle of "as far from constraints as possible". Second, the scalar convolution (10), under convexity of partial criteria, has the property of unimodality (i.e., the problem becomes one-extreme one). Moreover, the nonlinear scheme of compromises has the property of continuous adaptation to different situations in which it is necessary to accept a multi-criteria solution. In the tense situations (when one or more partial criteria are in dangerous proximity to constraints) it acts equivalent to the minimax model; in the fairly calm situations, the convolution (10) or (11) acts equivalent to the model of integrated optimality (i.e., economic scheme of compromises). In the interval between the two poles, the nonlinear convolution gives different degrees of alignment of the partial criteria. Thus, the application of the nonlinear scheme of compromises allows us to increase the accuracy of the decision due to continuity of adaptation [15].

4. Experiment

To verify the efficiency and evaluate the efficiency, experiments have been conducted on the application of the proposed metric together with the calculation of known indicators. The experiments were conducted using the Python programming language and a number of its libraries, such as scikit-learn, pandas and others [16].

The scikit-learn library has many classification algorithms that can be used to build a machine learning model. All scikit-learn machine learning models are implemented in their own classes, which are called Estimators .

- The following learning models have been created:
- Logistic regression or logit model (LR);
- Linear discriminant analysis (LDA);
- K-nearest neighbors method (KNN);
- Classification and regression with using trees (CART);
- Naive Bayes classifier (NB);
- Support vector machines (SVM);

We used a mixture of simple linear (LR and LDA) and nonlinear (KNN, CART, NB and SVM) algorithms.

As data for research, the "Iris" [17], well-known classical data set in machine learning and statistics, was used. It is included in the datasets module of the scikit-learn library and is run by the command: *url = https://raw.githubusercontent.com/jbrownlee/Datasets/master/iris.csv*

Based on the dataset, a machine learning model has been built, which predicts iris varieties for a new set of measurements. Before we apply our model to the new set, we need to make sure that the model actually works and its predictions can be trusted.

Unfortunately, we cannot use the data we took to build the model in order to evaluate the quality of the model. This is because our model memorizes the entire training set, and therefore it will always predict the correct label for any data point in the training set. This "memorization" tells us nothing about the quality of the model (in other words, we do not know if this model works properly on a new dataset).

To evaluate the effectiveness of the model, we present it with new labeled data. This is usually done by splitting the collected data (in this case, 150 flowers) into two parts. One piece of data is used to build our machine learning model and is called training data or training set. Other data will be used to evaluate the quality of the model, and they are called test data.

We will use stratified 10-fold cross-validation to improve the accuracy of the model [18]. This is an additional procedure, and, in general, we do not need to use it when the amount of input data is great. The dataset of our case is 150 lines (50 of each type), which is relatively small. Therefore, it is needed to increase the accuracy of the model.

5. Results

Scikit-learn Python contains many built-in features for analyzing the performance of models. In this task, we use some of these metrics and have written our own quality assessment functions from scratch to compare them with known ones [19 - 23].

The following indicators from sklearn.metrics are programmed:

- confusion_matrix (matrix of errors or matrix of inaccuracies or confusion);
- accuracy_score (accuracy);
- recall_score (recall);
- precision_score (precision);
- F1_score (F-score);
- roc_curve (ROC-curve);
- roc_auc_score (AUC-ROC).

Additionally, we propose our own indicator of the quality of classification algorithms - nonlinear scheme of compromises (*NSC*).

The libraries presented in Figure 4 were used for software development.

The obtained research results are presented in Tables 2-4.

Table 2

Comparative table for Iris-setosa class

Classification algorithm	Accuracy	Recall	Precision	F1	NSC
LR	0,942	1,00	1,00	1,00	Non
LDA	0,975	1,00	1,00	1,00	Non
KNN	0,958	1,00	1,00	1,00	Non
CART	0,95	1,00	1,00	1,00	Non
NB	0,95	1,00	1,00	1,00	Non
SVM	0,983	1,00	1,00	1,00	Non

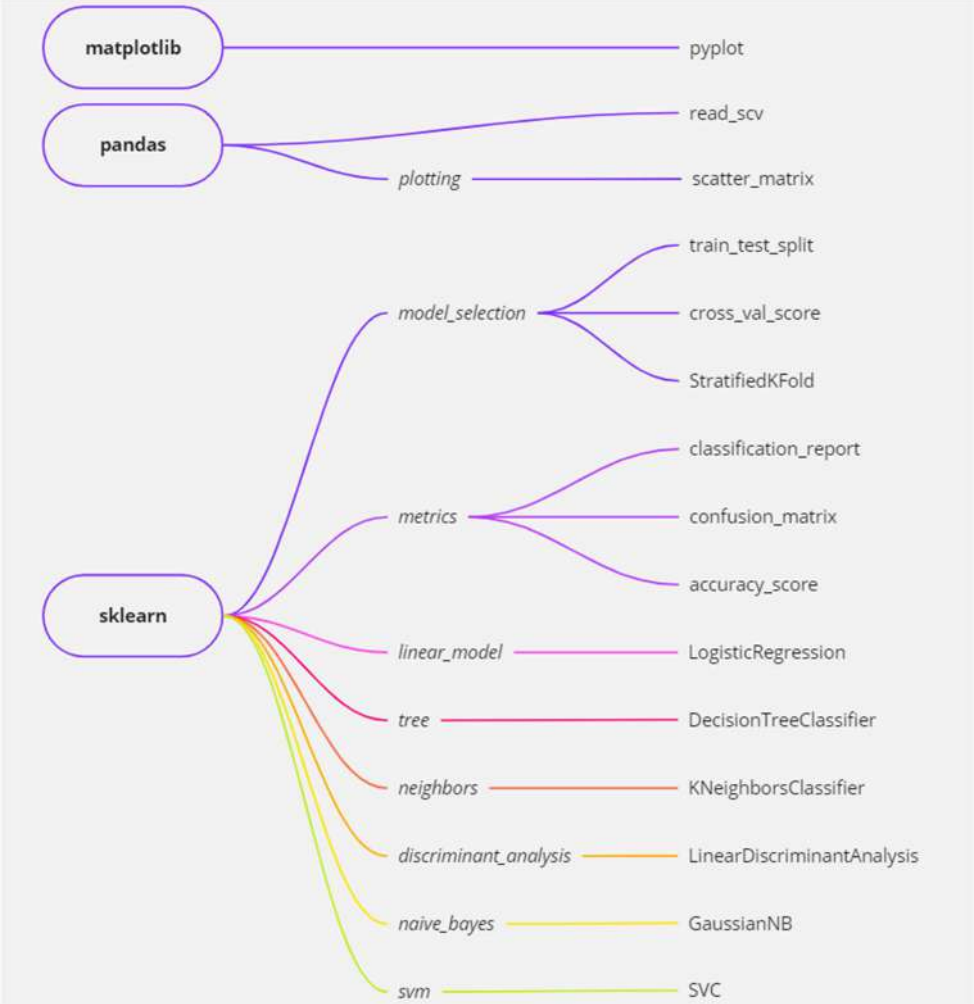

Figure 4: Used software libraries

Table 3

Comparative table for Iris-versicolor class

Classification algorithm	Accuracy	Recall	Precision	F1	NSC
LR	0,942	0,918	1,00	0,958	10053
LDA	0,975	0,92	1,00	0,96	10078
KNN	0,958	0,92	1,00	0,96	10069
CART	0,95	0,92	1,00	0,96	10058
NB	0,95	0,92	1,00	0,958	10056
SVM	0,983	0,92	1,00	0,96	10096

Table 4

Comparative table for Iris-virginica class

Classification algorithm	Accuracy	Recall	Precision	F1	NSC
LR	0,942	0,86	1,00	0,92	10037
LDA	0,975	0,83	1,00	0,91	10058
KNN	0,958	0,86	1,00	0,92	10043
CART	0,95	0,88	1,00	0,93	10040
NB	0,95	0,86	1,00	0,92	10039
SVM	0,983	0,9	1,00	0,94	10083

Since the experiments were performed for a well-known data set, which was well balanced and tested, the additional experiments were performed for a two-class classification of a real data set with 15,758 copies. The results obtained are summarized in Table 5.

Table 5
Comparative table of classification algorithms for real data set

Classification algorithm	Accuracy	Recall	Precision	F1	NSC
LR	0,57	0,61	0,68	0,64	10,79
LDA	0,61	0,62	0,71	0,66	11,58
KNN	0,67	0,63	0,69	0,66	11,9
CART	0,67	0,65	0,71	0,68	12,46
NB	0,65	0,63	0,68	0,65	11,54
SVM	0,67	0,66	0,70	0,68	12,43

6. Discussions

Analysis of the results of experiment 1 (Tables 2-4) shows that in this case, we have a balanced date set for which all algorithms show high-quality values, close to the standard.

The comparison table for the class Iris-setosa (Table 2) shows that if all other indicators are equal, it is possible to determine based on a single indicator (accuracy) and it makes no sense to calculate the NSC . With respect to a single indicator of accuracy, the highest accuracy was found for the SVM algorithm.

We should note that in tense moments, when the partial indicator attains the maximum '1', the calculation of NSC is not possible. In such cases, it is necessary to increase the accuracy of calculations, not to use rounding of digital values, or not to take into account such an indicator in the convolution. Another option may be to use instead of '1' the nearest number of required accuracy, for example, 0.99999.

Analysis of Table 3 shows that in the classification of Iris-versicolor, the best NSC indicator is shown by the SVM algorithm.

Analysis of Table 4 shows that in the classification of Iris-virginica, the best NSC indicator is also shown by the SVM algorithm.

For the second data set, approximately the same quality assessment results are observed for the algorithms of CART and SVM. Due to the calculation of the NSC, it is possible to reasonably determine the best algorithm, namely for this case of the CART classification.

7. Conclusions

Thus, in the paper, we propose for evaluating the classification quality to use the developed scalar quality indicator, which is a scalar nonlinear convolution of the known quality indicators, such as accuracy, recall, precision, F1, AUC_ROC.

This indicator allows us to give preference to one or another classification algorithm when the values of typical indicators are almost the same or have contradictions.

Studies have confirmed the usefulness of the proposed indicator NSC.

In the future, it is advisable to study the scalar indicator of quality in more detail, determine its limits and develop recommendations for the use of the NSC.

8. References

[1] M. Mitchell, Artificial Intelligence. A Guide for Thinking Humans, Penguin, London, 2020.
[2] E. Raj, Engineering MLOps, Packt Publishing, Birmingham, 2021.

[3] J. Dj. Novakovi´c, A. Veljovic´, S. S. Ilic´, Z. Papic´, M. Tomovic´, Evaluation of Classification Models in Machine Learning, Theory and Applications of Mathematics & Computer Science 7(1) (2017) 39–46.
[4] C. Körner, K. Waaijer, Mastering Azure Machine Learning. Perform large-scale end-to-end advanced machine learning in the cloud with Microsoft Azure Machine Learning, Packt Publishing, Birmingham, 2020.
[5] I. Puleko, S. Kravchenko, V. Chumakevych, V. Ptashnyk, Method of Machine Learning Based on Discrete Orthogonal Polynomials of Chebyshev, in: Proceedings of the 4th International Conference on Computational Linguistics and Intelligent Systems, COLINS 2020, CEUR, Lviv, 2020, pp. 67–76.
[6] Machine Learning Crash Course, Classification: True vs. False and Positive vs. Negative, 2021. URL: https://developers.google.com/machine-learning/crash-course/classification/true-false-positive-negative.
[7] S. Minaee, 20 Popular Machine Learning Metrics. Part 1: Classification & Regression Evaluation Metrics, 2019. URL: https://towardsdatascience.com/20-popular-machine-learning-metrics-part-1-classification-regression-evaluation-metrics-1ca3e282a2ce.
[8] W. Koehrsen, Beyond Accuracy: Precision and Recall, 2018. URL: https://towardsdatascience.com/beyond-accuracy-precision-and-recall-3da06bea9f6c.
[9] J. Brownlee, Assessing and Comparing Classifier Performance with ROC Curves, 2020. URL: https://machinelearningmastery.com/assessing-comparing-classifier-performance-roc-curves-2/.
[10] Ž. Đ. Vujović, Classification Model Evaluation Metrics, International Journal of Advanced Computer Science and Applications 12(6) (2021) 599–606. doi: 10.14569/IJACSA.2021.0120670
[11] A. Voronin, Yu. Ziatdinov, O. Kozlov, V. Chabaniuk, Vector Optimization of Dynamical Systems, Tehnika, Kyiv, 1999.(in russian).
[12] A. Voronin, Nonlinear Tradeoff Scheme in Multicriteria Estimation and Optimization Problems, Cybernetics and Systems Analysis 45(4) 2009 106 – 115.
[13] A. Zasjadko, Methods of Simplifying the Problem of Nonlinear Programming on the Basis of Classification of Limitations, Information Processing Systems 161 (2020) 59–70. doi: 10.30748/soi.2020.161.07.
[14] A. Voronin, Yu. Ziatdinov, Theory and Practice of Multicriteria Decisions: Models, Methods, Implementation, Lambert Academic Publishing, 2013. (in russian).
[15] A. Voronin, Yu. Ziatdinov, I. Varlamov, Non-Linear Trade-off Scheme in Multicriteria Decision-Making Problems, International Journal “Information Technologies & Knowledge” 11(1) (2017) 3–22
[16] Aurelien Geron. Hands-On Machine Learning with Scikit-Learn and TensorFlow. Concepts, Tools, and Techniques to Build Intelligent Systems. Boston. O’REILLY. 2018.
[17] A. Subasi, Practical Machine Learning For Data Analysis Using Python, Academic Press is an imprint of Elsevier, London, 2020.
[18] D. Kopec. Classic Computer Science Problems in Python, Manning Publications Co, New York, 2019.
[19] D. Paper, Data Science Fundamentals for Python and MongoDB, Apress, New York, 2018. doi:10.1007/978-1-4842-3597-3.
[20] M. Swamynathan, Mastering Machine Learning with Python in Six Steps, second edition, Apress, New York, 2019. doi: 10.1007/978-1-4842-4947-5.
[21] R. Lakshmanamoorthy, Python Code for Evaluation Metrics in ML/AI for Classification Problems, 2021. URL: https://analyticsindiamag.com/evaluation-metrics-in-ml-ai-for-classification-problems-wpython-code/.
[22] J. Brownlee, Metrics to Evaluate Machine Learning Algorithms in Python, 2020. URL: https://machinelearningmastery.com/metrics-evaluate-machine-learning-algorithms-python/.
[23] Scikit-learn, Metrics and Scoring: Quantifying the Quality of Predictions, 2020. URL: https://scikit-learn.org/stable/modules/model_evaluation.html.