Decision Support System for the Machine Learning Methods Selection in Big Data Mining

Nicolay Rudnichenko[1] [0000-0002-7343-8076], Vladimir Vychuzhanin[1] [0000-0002-5244-5808], Igor Petrov[2] [0000-0002-8740-6198], Denis Shibaev[3] [0000-0002-3260-5843]

[1]Odessa National Polytechnic University, Odessa, Ukraine
nickolay.rud@gmail.com, vint532@yandex.ua
[2]National University "Odessa Maritime Academy", Odessa, Ukraine
firmness@list.ru
[3]Odessa National Maritime University, Odessa, Ukraine
denscreamer@gmail.com

Abstract. This article focuses on the aspects of the decision support system for the machine learning methods selection in big data mining development. The paper includes the results of the analysis of the problem of intellectual processing and analysis of big data. It describes proposal ways of using metadata as a basis for the formation of an analytical rating for evaluating machine learning methods. The paper presents the results of designing and using a decision support system for evaluating machine learning methods for solving data mining problems. The developed decision support system allows us to reduce the analysis time of suitable methods for solving machine learning problems by a data science analyst, taking into account the specifics of the input data arrays, their volumes, structure and other metadata.

Keywords: decision support system, big data, data mining, data science, machine learning, data analysis

1 Introduction

Currently, there is a steady trend of regular growth in the volume of data collected in the various business organizations f production, operational and research activities processes [1]. The sources of the such big data volumes (Big Data) appearance are often customers various behavioral factors, the frequency and size of payments for the services or goods, parameters and characteristics of installed technical equipment, medical indicators for diagnosing human health and others [2-4].

Due to the statistical visibility and representativeness, the value of such data lies in the possibility of using it to search for hidden and unobvious relationships between individual factors (attributes) and target actions of clients to adjust and formulate business development strategies [5].

In fact, the implementation of such tasks becomes possible based on the use of data mining methods in order to extract new knowledge by forming and proving hypothe-

ses about significant relationships between the individual attributes of data samples [6-8].

Thus, the constant companies need to ensure a sufficient level of quality in the provision of goods and services, due to the high level of competition in the organization's business targets requires usage of the data science, data mining methods and technologies in key business processes [9]. For this purpose, machine learning (ML) methods can be used to build various regression and predictive models for the business objectives.

2 Description of Problem

The concepts which are laid down in this approach allow data mining specialists to conduct a comprehensive phased Big Data analysis and processing, sequentially implementing the necessary processes, including distributed structuring of heterogeneous data, their consolidation, aggregation, cleaning and pre-processing, eliminating anomalies, omissions, errors side values [10-12].

However, all these processes are time-consuming for analysts on the experimental selection and mathematical model selection with the corresponding hyperparameters and quality assessment metrics. One of the key factors to provide a successful and prompt solution of a target problem by an analyst is his experience in building ML models, business problems vision depth and the software tools, technologies and libraries knowledge [13,14].

This can lead to subjectivity of the analysis results and affect the accuracy and generalizing ability of the formed ML models. At the same time, the computational costs of the computer equipment used in the data analysis process are also significant, which affects the total cost of developing ML models [15]

An additional complication is the correlation, evaluation, and required ML method or their combination selection processes for the effective solution of the data mining posed problem (with the achievement of the model created by a sufficient accuracy, adequacy level and generalizing ability) without lengthy computational experiments.

This problem becomes especially relevant in cases where the Big Data sample size exceeds the permissible amount of disk space in the used data warehouses [16-19]. If there are limitations in the throughput capacity Big Data transmission in serial or parallel mode in a local or global network, then effective analysis of such data becomes difficult [20-24]. A possible solution is to compress, transform or structure the data with the extraction of quantitative and qualitative meta-information from them [25-28].

In this regard, the urgent and relevant task is to automate the processes of selecting suitable ML methods based on the input Big Data volumes analysis, their various statistical and probabilistic characteristics, taking into account the subject area specifics and the ML problem type. This can be done by developing a decision support system (DSS) with a number of intellectual functions for the formation and accounting of meta-information about data attributes, their structure, level of generalization and significance.

Currently, there are various analytical systems on the market for comparing and comparing various ML methods for solving classification and regression problems on Big Data, however, their functionality is limited and does not allow to fully take into account the nature of the input data, their volume and subject area of analysis. DSS that can be adapted for such ML tasks are Wolfram Mathematica, EIDOS, Expert Choice and ViEA. But, there is no possibility of a flexible system configuration for the user's needs, filling with new functionality and updating dependencies is not performed regularly. This makes it impossible to use such solutions to obtain reliable and reliable data mining results [29]. The purpose of this article is to develop a DSS project for choosing ML methods to solve data mining tasks on user-specified data sets based on their structure and volume to reduce time spent on detailed experimental calculations.

3 Decision support system development

3.1 System concept

The functioning of the proposed system is carried out in several stages:

- data import;
- formation of meta-information from the downloaded data;
- Data Mining tasks type and methods selection;
- specification of criteria and metrics for ML model quality assessing;
- obtained ML models creation and evaluation;
- ML models obtained results visualization and the issuance of a methods ranked list which provides the highest quality solution to the problem.

The process of DSS in general is shown in fig. 1.

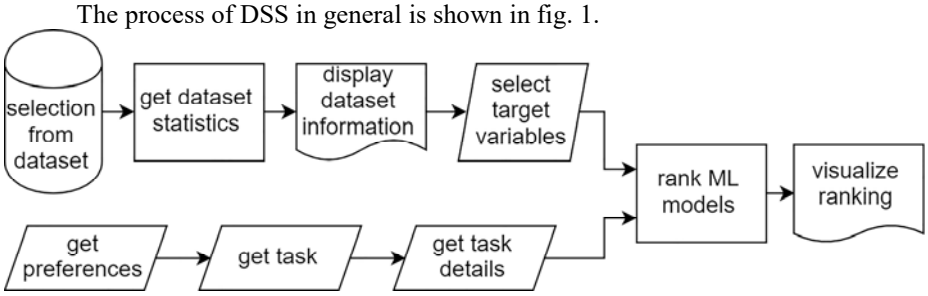

Fig. 1. General system operation

DSS supports the ability to import various training datasets and test ML models in *.csv, *.xls and *.json formats. If necessary, the processing of structured relational data supports the integration of SQL database management systems MySQL and Postgresql, as well as NoSQL MongoDB. Data loading is performed in multi-threaded

mode with blocks from 2 to 128 MB in each, which allows us to distribute computing processes and scale the system in the future.

After importing data, the system gives a message about the success of the operations performed and provides a brief meta-information, which includes: the total number of records in the loaded data set, signs number, signs and records number ratio, data amount. The user can set the block size for further analysis depending on the displayed data.

Since the size of the analyzed data volume in data mining tasks can reach large values, this can reduce the efficiency of computing operations in the analysis process. Therefore, the user is invited to choose one of the options: make a selection of a given size from the imported set from 0 to 100% or select a value randomly with the possibility of stratified sampling by a specific column to maintain proportions. This allows us to reduce the amount of RAM used to store data and speed up the computing analysis operations process. To ensure the analysis process flexibility the user has the opportunity to select the necessary features by disabling or removing unnecessary from the system, as well as specify one or more target output variables.

For each column from the dataset table, the selection indicates whether it is numeric or categorical. For columns with numerical values, the following statistical information is displayed: range of values, standard distribution, average value, median, asymmetry coefficient, kurtosis coefficient, chi-square test of the normal distribution test, Pearson correlation with the target variable (if one was specified) and its confidence. For columns with categorical values, the following is displayed: number of categories, relative mode frequency and Gini coefficient.

The system allows us to take into account the specifics of the data mining task being solved and set the priority of assessing the ML model quality based on the operation speed or the accuracy obtained. This is one of the target criteria for forming a ranked methods list. DSS implements support for 2 types of data mining tasks: classification and regression. Depending on the type of task, the user must specify the necessary metric for evaluating ML models.

For classification tasks, the following metrics are supported: accuracy (share of correct answers), recall (share of found objects of a positive class), precision (share of truly positive from classified positive objects), F1 (harmonious average), AUC. For regression tasks, metrics are supported: mean squared error (MSE), and mean absolute error (MAE). After the data import is completed, the analysis process is started in the background. The analysis consists in the phased creation of model instances with hyperparameter values in the ranges specified by the user (or in the default range specified for each ML method) and their assessment by the selected metrics. It is possible to view detailed logs of obtained metric estimates at individual iterations and epochs of training and testing ML models in *.txt format.

After the system performs data analysis procedures, the results are displayed in the form of a summary report on the most suitable ML methods for the criteria selected by users, which automatically saves to the *.xls file and is shown at the system's log screen. The structure of the general report has a tabular form and contains: the name of the method, the level of its adequacy for the selected data set and task (in relative units from 0 to 100), the approximate predicted RAM amount needed for training and

testing ML method on the sample, and the predicted time spent on carrying out computing processes.

User can select the criteria for ordering methods in a window for displaying analysis results. In particular, it is possible to sort the methods by individual metrics, speed, accuracy or ease of results interpretation (which depends on the data amount).

For a simpler and more understandable the analysis results interpretation DSS supports visualization using a bar chart, where each column reflects the quality of an individual model in the rating form for a given metric. It is possible to save the constructed graphic visualizations in *.png format.

Based on the developed concept of the system's functioning, its design and software implementation can be carried out next.

3.2 DSS project implementation

To display the relationship between users and the system, a diagram of use cases was compiled (fig. 2). The main unary scenarios of user interaction with the system are: selecting a data set, viewing data statistics, updating the task, viewing models rating. The server side provides support of optional data set selected part retrieval, imported data set statistics generation, computational processes for compiling the rating and its graphical visualization, saving results to a file.

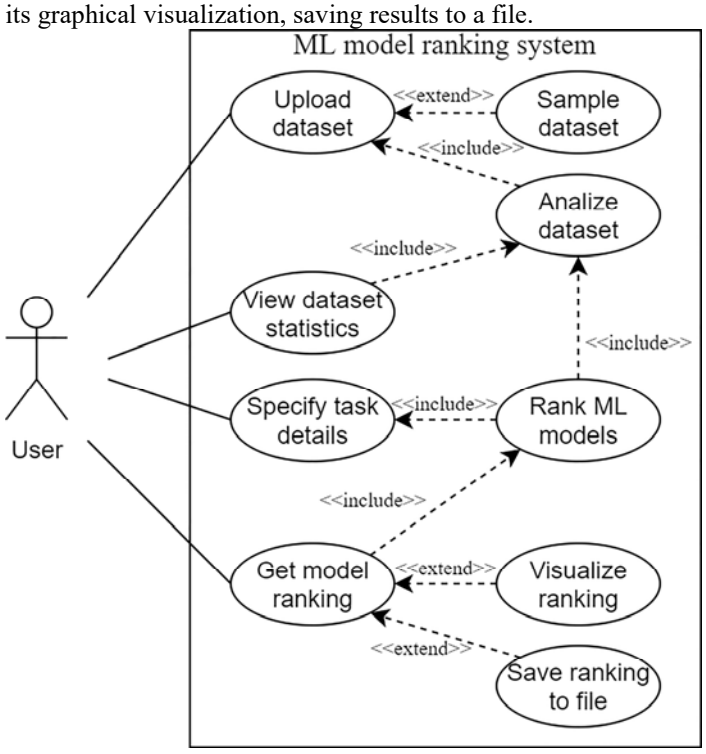

Fig. 2. DSS use cases diagram

The system is divided into some components that perform various parts of the task. These components, as well as the data exchange between them, are shown in fig.3. In this diagram, the UserInterface component is responsible for user interaction with the system, providing controls, as well as reflecting the selected table, its statistics and the ranking list generated. The TableManager component is responsible for loading and storing tabular data, it receives the path and data loading mode from the interface, after which it provides the other components with a dataset table.

The TableAnalyzer component is responsible for removing statistics from a table. After extracting statistical data and obtaining additional information from the user, the ModelRanker component determines which of the ML models are most likely to be suitable for this task. After that, the generated rating of models is displayed to the user, and is also saved in the file system by the RankingSaver component.

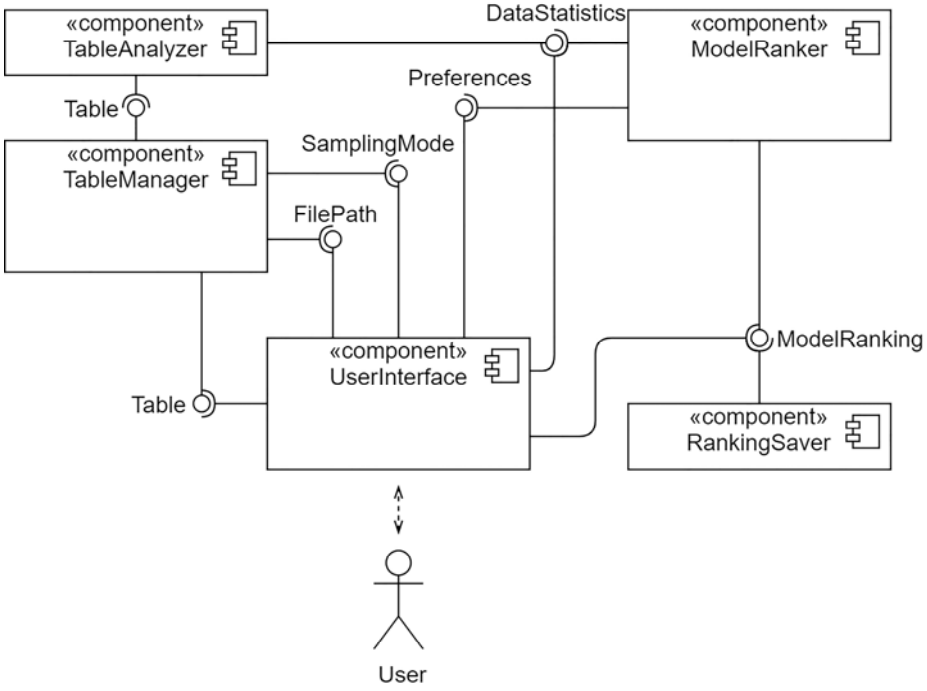

Fig. 3. DSS component diagram

To formalize the key functional processes that are carried out in the system and their relationship, a sequence diagram of actions has been drawn up, which reflects the basic call operations (fig. 4). The user through the interface sets the path to the data in the TableManager, which returns meta-information for the selected dataset table. After specifying the sampling mode and confirming the selected set, TableAnalyzer extracts statistical information, showing the result to the user. Next, user specifies the necessary ML methods, hyperparameters, model quality assessment metrics and selects the target and input columns (stored in the Preferences object), based on

which the required metadata is determined and computational operations are performed to evaluate the method.

ModelRanker object sequentially downloads the resulting ML models to generate a consolidated rating by storing a detailed calculation log in a file and displaying the output data in short form to the user.

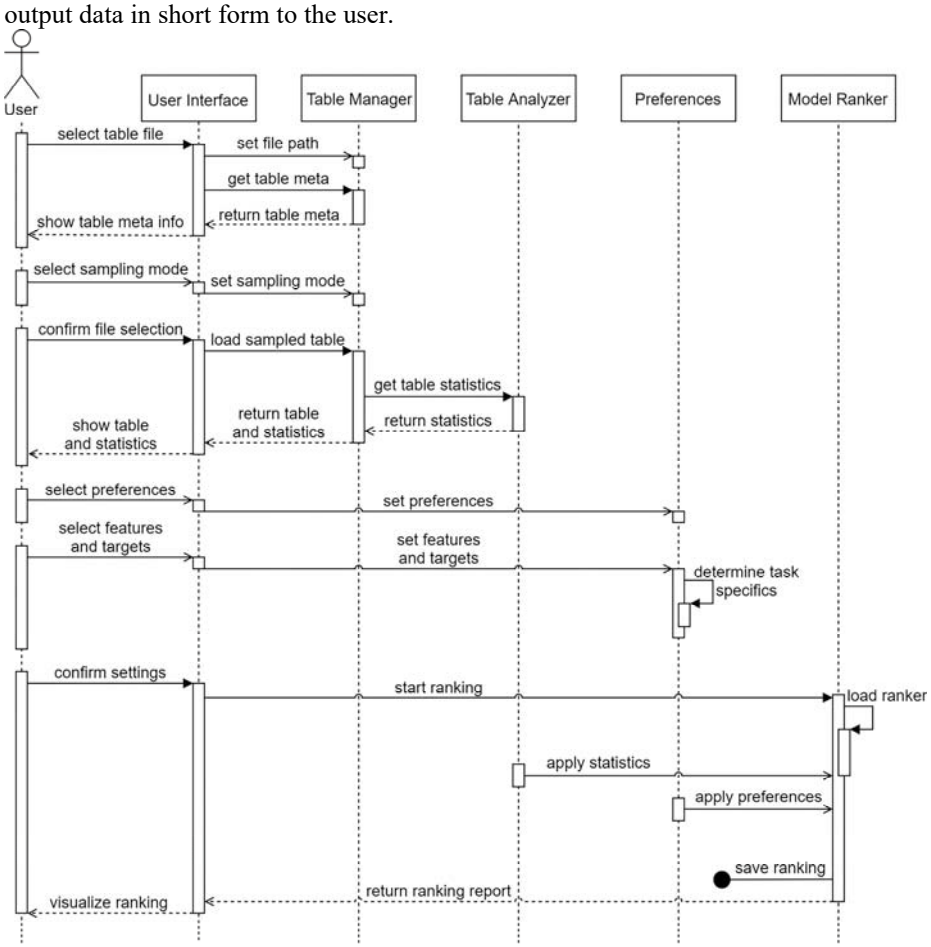

Fig. 4. DSS user sequence diagram

To visualize the system software implementation structure, a class diagram was created, a fragment of which is shown in fig. 5. This diagram shows the structural relationships between the various classes in the DSS. The user interface object has one-way associations with the table manager object (which in turn is associated with the table analysis object), the report saving object and the rating creating object. An analyzing table object creates an object containing metadata and statistical data for each column, which is a necessary process for rating models. After the user enters all the necessary options for data analysis, UserInterface creates an object of the preferences class.

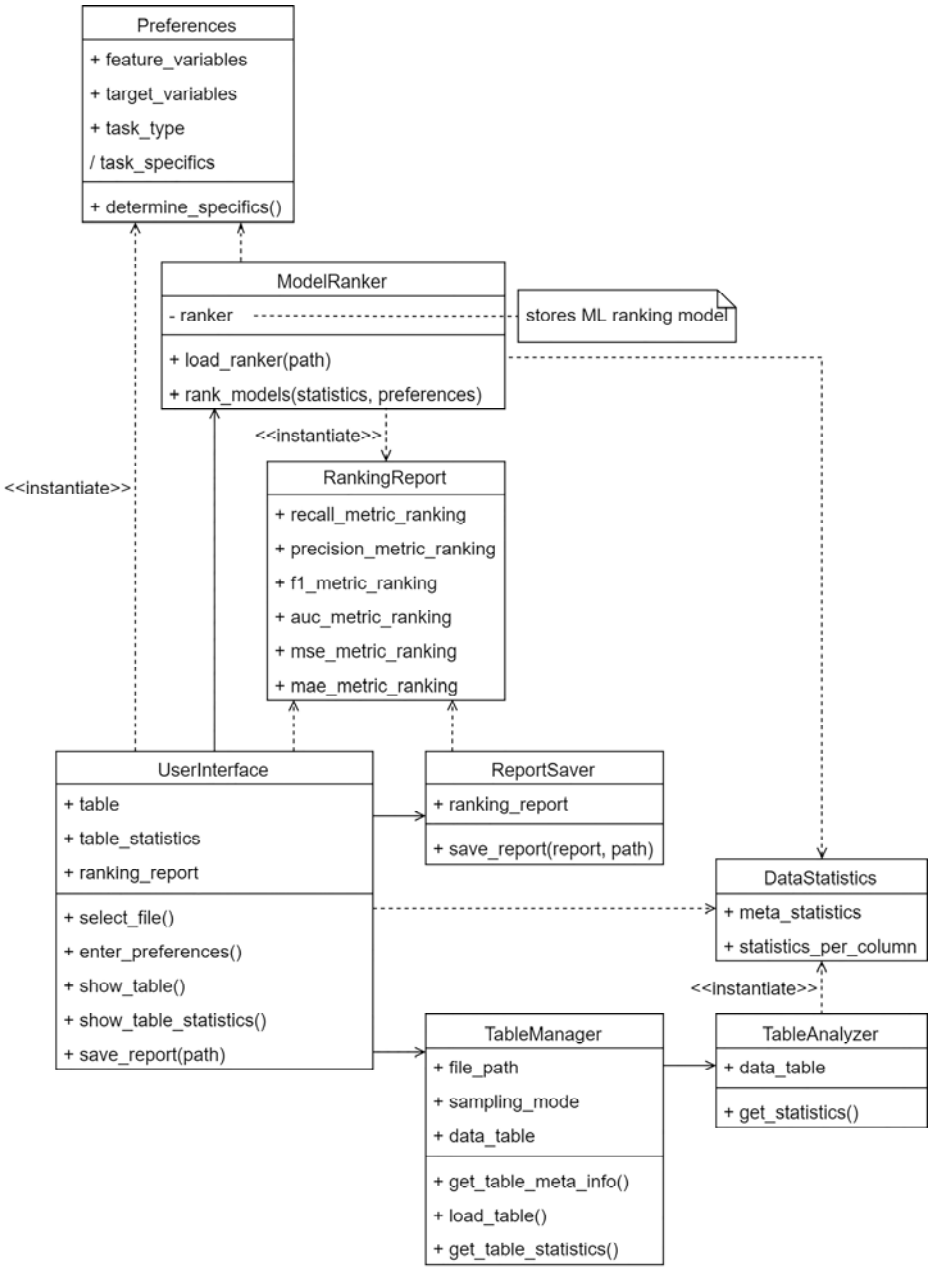

Fig. 5. DSS main classes diagram fragment

The functionality of this class is used as the basis for creating a rating. After creating a rating of models according to user-specified metrics, this object is used by the user interface to display and the ReportSaver class to save it to the file system. A

special method has been created in the ModelRanker class to load the ML model used to compile the rating along a given path. This allows us to simplify the process of choosing a suitable model for ranking in the ranked list form by quickly changing the model estimation algorithm used, which will allow analysts to train and use the created models in the future.

After user receiving the rating it's possible to estimate the approximate time of the ML model training process for each of the methods involved and the RAM amount. The approximate time of creating a model for non-iterative learning algorithms is estimated based on extrapolating the obtained model training duration on a given records number and attributes on an active workstation by the records number ratio and attributes with the data set used. The approximate model training time is calculated by multiplying the measured training time by the computational complexity with the substituted ratios. The training speed can largely depend on the user's workstation hardware therefore, it is not calculated accurately, the error can be up to 15-20%. The approximate spatial complexity is calculated in a similar way using the known memory usage and the table size and features ratio.

In the creating DSS process were used several technologies such as: Python 3.7 programming language, Pandas library for analyzing and manipulating data, library for working with ML sklearn models, XGBoost library, libraries for performing mathematical operations and scientific calculations NumPy and SciPy. PyCharm is used as an IDE. The matplotlib library was used to create the graph visualization interface.

The system user interface is implemented using block layout in the web application form. It includes 4 tabs with graphical components for managing data import processes (clicking on the corresponding button and selecting the desired data set in the dialog box), displaying data and statistical information in a tabular form, selecting and setting model parameters in text fields and drop-down lists and viewing the results of operations in the graphs and charts forms using Matplotlib and Chart.js libraries.

4 Experiments and results analysis

During the DSS creation several algorithm variants were considered for ranking the models, taking into account the possibilities of their training and the implementation complexity, due to the need in each sign of the dataset and metadata. Analysis of literary sources [15-19] let us to perform the following algorithm which based on the use of an artificial neural network (NN) multilayer perceptron:

1. For each attribute, the values of its statistics, metadata, and task type are used as input for NN. The NN model provides values that reflect the relative ranking of ML models by metrics for each attribute.

2. The generated ratings for each characteristic are taken into account by calculating the arithmetic average of all issued ratings.

3. The obtained models relative ratings values by metrics are reduced to a range from 0 to 1 so that 0 are not suitable models (with a low metrics rating), and 1 is the

most suitable. If a certain metric isn't compatible with the task type, it will not be displayed to the user.

This algorithm provides the possibility of combined accounting for the different task type, meta-information about data and statistical data for each attribute in the rating. If a single feature doesn't affect the quality of the compared models, NN is able to produce values close to 0 for the rating of such a feature that will not introduce significant errors and will not affect the final analysis result. Since all operations in this algorithm can be differentiated, it is possible to train the used ML model by the stochastic gradient descent method.

The model formation for ranking is based on the learning algorithm with a teacher. Metadata, task type, and attribute statistics taken from imported data sets are used as input for training.

As the target variables, model ratings obtained by experimental models comparison using various metrics on data sets are used. Ratings are converted to a range from 0 to 1 so that the value higher, the algorithm is better in evaluating relative to others. The loss function ensures the correspondence of the generated rating for each ML experimental rating model.

The developed DSS has been tested on several experimental model comparisons. The methods of linear regression, decision tree, random forest, support vector method (for classification and regression), xgboost, logistic regression, and naive Bayes classifier were evaluated.

The comparable metrics used to assess the models quality were AUC, F1, precision and recall for classification and MSE with MAE for regression. When creating models, the values of their hyperparameters are set by default in accordance with the predefined values by the library, except for random forest and xgboost algorithms, in which the number of trees was set to 64.

For experimental comparison, a cross-validation method was used on the basis of dividing the sample into 10 equal parts, with a stratified breakdown for the classification problem.

In the study of the system operation the Diabetes dataset was used for the regression task. The data set has 442 records and 10 attributes with real values. Signs indicate age, gender, weight to height, average blood pressure, and 6 blood counts. The target variable is the diabetes degree.

At the stage of data cleaning, objects in which at least one of the attributes departed by more than 3 standard deviations for all values of this attribute were deleted from the dataset.

After cleaning, 97.3% of the original objects remained. Due to the fact that during the experimental comparison some of the models used require input data normalization, all features were normalized to the range from -1 to 1, and the target variable to the range from 0 to 1.

After testing the models using the cross-validation method, regression model metrics (MSE, MAE) were obtained (Table 1). Results are sorted by MSE metric (best to worst).

Table 1. Experimental model metrics

Model Name	*MSE*	*MAE*
Linear Regression	0.030	0.141
Random Forest	0.033	0.149
Support Vector Regressor	0.034	0.144
XGBoost	0.034	0.149
Decision Tree	0.062	0.193

The experimentally obtained metrics were converted to the models ranking. A value 1 is the best result, 0 is the worst (Table 2). The results for experimental part were obtained using the developed software script in Anaconda and Jupiter notebook environment in usual "step by step" mode. The results for DSS part were obtained by importing dataset with default model's hyperparameter setting.

Further, using the collected statistics on the characteristics, metadata and a data set, the system generated a ML models rating for this task. As can be seen from the obtained results, the order of the best models in the rating issued by the system approximately coincides with the experimental rating.

Table 2. Experimental and analytical model ratings

Model Name	*Experimental*		*DSS*	
	MSE	MAE	MSE	MAE
Linear Regression	1.000	1.000	0.731	0.728
Random Forest	0.897	0.856	0.653	0.731
Support Vector Regressor	0.892	0.938	0.682	0.383
XGBoost	0.880	0.842	0.657	0.580
Decision Tree	0.000	0.000	0.368	0.368

For ease of comparison, a diagram was presented showing the comparative rating of models obtained experimentally and generated by the developed DSS (fig. 6).

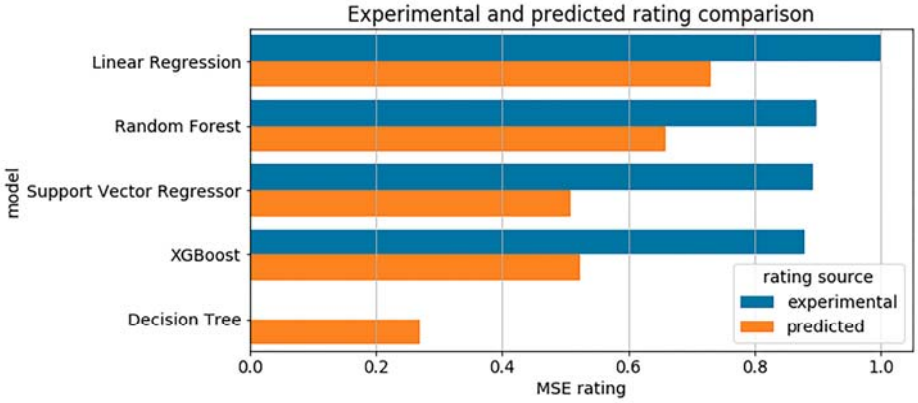

Fig. 6. Comparison of the experimental and rated by the DSS

Based on this rating, the system was able to determine the best model (according to experimental comparison), but with less confidence. It is also noticeable that the evaluation of the support vector mechanism and XGBoost was put in the wrong order. Evaluation of a bad model, decision trees, was put below others, but far from experimental values. The reason for the decision tree low rating could be that in regression tasks the tree outputs only discrete values, which will increase the error of MSE and MAE.

5 Conclusion

Summarizing the comparisons of the estimated and experimental model ratings, we can say that DSS is capable of fairly accurate ML models estimation depending on the type of task, metadata, and data set statistical information, its level of accuracy reaches 70-75%. Errors sources in the ML model ratings are: a relatively small training set and the test data set similarity with the training one, which negatively affects the model's generalizing ability to analyze data dependencies.

Since the recommended ML methods ranked list issued by the DSS is not always accurate, in tasks with critical requirements for the model's reliability, completeness and accuracy it is advisable to conduct additional exploratory analysis on the selected fragment of the data set according to the first 3 methods issued by the system.

Due to the fact that the data set table's size may be larger than free memory, it was important to make possible records selection without loading the file completely into memory. The implementation of this functionality depends on which sampling mode is selected and in what format the table is saved. With stratified random sampling, memory usage is higher than in other modes, since it is necessary to calculate the different values number in the selected column. Therefore, there may be cases when some data are missing in the table; therefore, this problem must be taken into account when collecting statistics from the table.

When compiling statistics for some data sets, the missing fragments in the samples were ignored, which also introduced some errors in the data analysis process. The ability to fill in the missing data using various algorithms is one of the options for the future system development and upgrading.

Reference

1. Rudnichenko, N., Vychuzhanin, V., Shybaieva, N., Shybaiev, D., Otradskaya, T., Petrov, I.: The use of machine learning methods to automate the classification of text data arrays large amounts. Information management systems and technologies. Problems and solutions. Ecology, Odessa, pp.31-46 (2019).
2. Sandryhaila, A., Moura, J.M.: Big data analysis with signal processing on graphs:representation and processing of massive data sets with irregular structure. IEEE-Signal Process. vol. 31(5), pp. 80–90 (2014).
3. Dietrich, D., Heller, B., Yang, B.: Data Science & Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data. Wiley, Hoboken (2015).

4. Rudnichenko, N., Vychuzhanin, V., Shybaieva, N., Shybaiev, D.: Big data intellectual analysis in the diagnosis of the transportation systems technical condition. Systems and means of transport. Problems of operation and diagnostics. KSMA, Kherson, pp.57-69 (2019).
5. Jordan, M.I., Mitchell, T.M.: Machine learning: trends, perspectives, and prospects. Science, vol. 349(6245), pp. 255–260 (2015).
6. Chen, H., Chiang, R.H., Storey, V.C.: Business intelligence and analytics: From big data to big impact. MIS Q. vol 4, pp. 1165–1188 (2012).
7. Vychuzhanin, V.V., Shibaev, D.S., Boyko, V.D., Shibaeva, N.O., Rudnichenko N.D.: Big data mapping in the geopositioning systems for fishing industry. International Scientific and Technical Conference on Computer Sciences and Information Technologies. pp.28-31 (2017).
8. Phillips-Wren, G.: Ai Tools in Decision Making Support Systems: A Review. International Journal on Artificial Intelligence Tools. vol.21 (2012).
9. Rudnichenko N.D., Vychuzhanin, V.V., Shybaiev, D.S.: The use of cluster data analysis to highlight measures of factors affecting the performance similarity of complex technical systems. Informatics and mathematical methods in simulation. vol. 3, pp.214-219 (2017)
10. Zeng, X., Lu, J.: Decision Support Systems with Uncertainties in Big Data Environments. Knowledge-Based Systems. vol.143 (2018).
11. Rudnichenko, M.D., Gezha, N.I., Belyaev, K.O., Kuzmin, A.D.: Performance analysis of machine learning model ensembles. In III All-Ukrainian scientific-practical conference of young scientists, students and cadets “Information protection in information and communication systems”. Lviv. pp.259-260 (2019).
12. Gomez, D., Rojas, A.: An empirical overview of the no free lunch theorem and its effect on real-world machine learning classification (2015).
13. Han, J., Kamber, M., Pei, J.: Data mining: concepts and techniques, Morgan Kaufmann (2011).
14. Padhy, N., Mishra P., Panigrahi, R.: The survey of data mining applications (2012).
15. Sumiran, K.: An overview of data mining techniques and their application in industrial engineering (2018).
16. Ramageri, B. M.: Data mining techniques and applications (2010).
17. Engels, C., Bratsas, C., Koupidis, K., Musyaffa, F.: Requirements for statistical analytics and data mining (2016).
18. Shalev-Shwartz, S., Ben-David, S.: Understanding machine learning: from theory to algorithms. Cambridge University Press (2014).
19. Jordan, M. I., Mitchell, T. M.: Machine learning: trends, perspectives, and prospects (2015).
20. Ayon, D.: Machine learning algorithms: A Review (2016).
21. Chugreev, V.L.: Decision support systems using machine learning methods and predictive analytics. Problems of economic growth and sustainable development of the Vologda territories, pp.79-83 (2016).
22. Sinitsyn, E.V., Tolmachev, A.V.: Model of the system of decision support in the financial markets for enterprises on the basis of probabilistic analysis and machine learning. Herald UFU. Economics and Management Series. vol.18. pp.378-393 (2019).
23. Savenkov, P.A.: Using machine learning methods and algorithms in management decision support systems. Journal of Science and Education, vol. 55, pp.23-25 (2019).
24. Korneev, S.: Decision support systems in business. Networks and business. vol.25, pp.102-110 (2005).

25. Chichirin, E.N.: Intelligent methods in simulation of decision making processes. Computer tools, networks and systems, vol. 17, pp.86-94 (2018).
26. Kogalovsky, M.P.: Metadata, their properties, functions, classification and presentation tools. In the 14th All-Russian Scientific Conference "Electronic libraries: perspective methods and technologies, electronic collections". Yaroslavl (2012).
27. Kogalovsky M.P.: Metadata in computer systems. Programming, MAIK Science "Interperiodica", vol.39, pp.28-46 (2013).
28. Skvortsov, N. A., Bryukhov, D. O., Kalinichenko, L. A., Kovalev, D., Stupnikov. S. A.: Metadata on scientific methods to ensure their reuse and reproducibility of results.. RCDL. Yaroslavl (2014).
29. Kalinichenko, L. A., Stupnikov, S. A., Vovchenko, A. E., Kovalev, D. A.: Conceptual declarative problem specification and solving in data intensive domains. Informatics and Applications, vol. 7. (2013).