Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 169 (2020) 780–786

Procedia Computer Science

www.elsevier.com/locate/procedia

Postproceedings of the 10th Annual International Conference on Biologically Inspired Cognitive Architectures, BICA 2019 (Tenth Annual Meeting of the BICA Society)

Constructor of compositions of machine learning models for solving classification problems

Igor Lavrov[a], Jenny Domashova[b]*

[a] *student of the Institute of Cyber Intelligence Systems, National Research Nuclear University MEPhI (Moscow Engineering Physics Institute), Kashirskoe highway 31, Moscow, 115409, Russian Federation, igor@lavrow.ru*

[b] *Candidate of Sciences (PhD) in Economics, Associate Professor of the Institute of Cyber Intelligence Systems, National Research Nuclear University MEPhI (Moscow Engineering Physics Institute), Kashirskoe highway 31, Moscow, 115409, Russian Federation, janedom@mail.ru*

Abstract

The article concerns the description of program software that gives an opportunity to solve classification problem by machine learning methods on input data in context of different subject areas. The list of such tasks can include: identification of suspicious state contracts for collusion of suppliers, forecasting the execution of a government contract, forecasting license revocation from credit organizations and insurance companies, etc. The application allows a user to build a composition of several base machine learning models to solve classification problems. As part of the study, the functional requirements for the product being created are presented, the architecture has been developed, the design and testing of the proposed process for creating machine learning model compositions has been carried out. The use of the application will allow to build compositions of machine learning models in a user-friendly mode in order to increase the accuracy of the classification problem. The use of the proposed tools improves the accuracy of solving the classification problem on input data in the context of various subject areas through the utilization of a composition of machine learning methods.

Peer-review under responsibility of the scientific committee of the 10th Annual International Conference on Biologically Inspired Cognitive Architectures.

* Corresponding author: Igor Lavrov.
E-mail address: igor@lavrow.ru

1877-0509
Peer-review under responsibility of the scientific committee of the 10th Annual International Conference on Biologically Inspired Cognitive Architectures.
10.1016/j.procs.2020.02.165

Keywords: machine learning; neural networks; decision trees; model constructor; model composition; classification problem

1. INTRODUCTION

This article presents a description of a software product that provides the ability to conduct machine learning on input data from different subject areas in order to solve the problem of binary classification. The obtained results of the product, in particular, can be used to analyze and assess the risks of public procurement and business related to the procurement of goods, works and services. The system can also be used, for example, to predict license withdrawal from credit institutions and insurance companies; automated monitoring and detection of suspicious banking operations, etc.

The product being developed should allow the end user to independently choose or design models and set their learning parameters with the possibility of further re-use of both the models themselves and the compositions made from them.

The advantage of such a system will be maximum ease of use for the user, the flexibility of creating compositions of machine learning models, as well as low labor costs for adding new system components, provided by the modularity of the application.

2. MATERIALS AND METHODS

Nowadays there is a variety of tools that provide the user with the ability to use machine learning methods to solve applied problems. Known applications with built-in machine learning tools include STATISTICA Data Miner and IBM SPSS Modeler [1,2]. In addition, there is a special developed language ASDIEL for programming your own machine learning models, as well as the software product Weka [3]. The advantage of ASDIEL is its flexibility and wide functionality, an obvious drawback – the complexity for the user and the need for programming skills to work with the product. The Weka program also contains a wide functionality, but it is user-friendly, however this product is also complex in the part of setting parameters for building models.

In addition, there are neural network constructors allowing to choose the most appropriate neural network architecture for solving a specific applied problem. In this case, the procedure for selecting the architecture in such constructors is visualized [4,5].

The main disadvantage of these systems is the inability to create a composition of machine learning models of different nature. The information system being developed as part of this work takes this disadvantage into account.

When solving complex problems of classification, regression, and forecasting, it often turns out that none of the algorithms provides the desired quality of dependency reconstruction. To improve the quality of models, it is proposed to use compositions or ensembles of machine learning algorithms (methods), in which the errors of individual algorithms are mutually compensated. All compositions are based on the idea of training several (basic) classifiers on the same train-sample and a combination of their predictions according to some rule for new test objects. The composition of the models makes it possible to collectively obtain a more complex model than each of them separately; to avoid over- or under-training; to work with features of different nature and to use different machine learning algorithms.

This paper proposes a toolkit that allows you to construct the composition of arbitrary machine learning models to solve a specific applied problem. The list of used machine learning methods can be extended in the life cycle of the product being developed.

The software product is focused on solving the classification problem, the essence of which is to assign each given set of objects to one of the previously known classes.

When applying machine learning methods, decisions are usually required on the following issues:

- selection of the method of dividing the training sample into Train- and Test-samples – to assess the generalizing ability of the model by checking the learning results of the selected model on test data [6];
- selection of the method of selecting features – for selecting the most informative features;
- selection of parameters for machine learning methods – to fine tune the model to solve the problem [7].

A specific method of machine learning, taken separately, gives adequate results in rare cases, however, the improvement of the quality of classification can be achieved by the composition of separate base models, that is implemented in the proposed product.

In addition, the input data in machine learning tasks can be of arbitrary nature and be heterogeneous, and therefore it is necessary to carry out their preprocessing before working with them. Data preprocessing is performed by the software before running the learning algorithms.

3. RESULTS

The software product allows the construction of several machine learning models from the basic blocks by specifying the type of composition that unites them. The general ideology of using the product is shown on the Fig. 1.

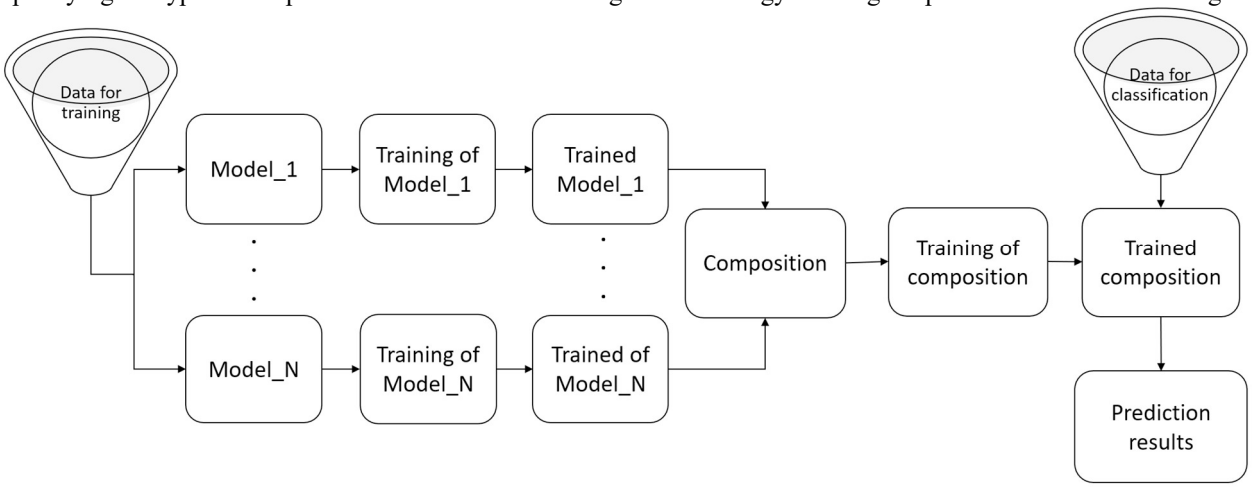

Fig. 1 – The general ideology of product use

The input data are pre-processed. Then, base models are built. Each base model (as shown on the Fig. 2) is determined by the method of splitting into train- and test-samples, the method of features selection, the method of training (e.g. neural networks or decision trees).

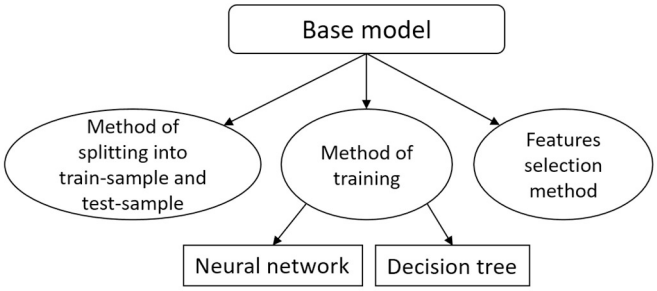

Fig. 2 – The definition of a base model

Base models will be trained only as a part of compositions on data samples loaded by users.

The created model is trained on the input data. Trained models are combined into a composition. Then, the composition of machine learning models is trained. Finally, data for classification are submitted to the trained composition.

Strategies for user interaction with the product may be different. It is possible to build at once the whole set of base models for its training with the subsequent composition. It is also possible to gradually adding base models to a

composition according to the principle of boosting: in order to improve the quality of an already constructed composition, the following added base model should be learnt mainly on those objects of the train-sample on which the already constructed composition gives an error. For supporting this mode, the user has the opportunity to select the desired subsample (objects on which the base models already included in the composition are often wrong), to observe the magnitude of the composition error at each step of its construction.

The software product allows a user to build composition models that can be divided into two groups: trainable compositions and compositions that do not require training (untrainable compositions). After the training of basic methods trainable compositions are trained on the results from the training of basic algorithms. Untrainable compositions aggregate the results of training of basic methods according to a certain rule. Untrainable compositions include voting, probabilistic compositions and bagging. Trainable compositions include weighted voting, boosting (including gradient boosting on decision trees and AdaBoost on decision trees), random forest and stacking.

The initial data are: a training sample, which is a table of initial data with values of attributes; and the data for which the classification will be carried out.

The product being developed has the following functional requirements: it should be able to read the files with the data of the train- or test-samples in CSV format and handle outliers and missing values. The product should provide an opportunity to build and train machine learning models on the selected train-sample, as well as composition of machine learning models, in which machine learning algorithms of different nature can be used. It should be possible to use previously created compositions and base machine learning models for a given sample of data for forecasting.

In order to prevent the user from creating the same models and compositions each time, it is necessary to ensure that the composition parameters are saved and unloaded from the database along with the base machine learning models used. The functionality for saving and unloading from the database of the training results of base models and compositions should also be implemented. Each base model and composition of models, as well as their outputs, refer to a specific user, therefore, it is necessary to take into account users of the product in the database.

The developed product (being an information system) involves the use by many users, each of them has their own set of input data, base models and compositions. In order to avoid the use of other people's base models, compositions and data, as well as to prevent unwanted user actions, authorization and authentication support is required. Access to data and functionality of the information system should be provided only to authorized users, and the available data and functionality should be determined by the role of the user in the system.

The results of work and training of models and compositions of machine learning models should be saved as a report in PDF format for viewing by the user and as a JSON object for transmission to other information systems, in particular to the graphical user interface.

To interact with the user interface, as well as with other external services, the product must provide access to its functionality through a convenient interface, the REST API was chosen as such.

The proposed general architecture of the product is shown in Fig. 3.

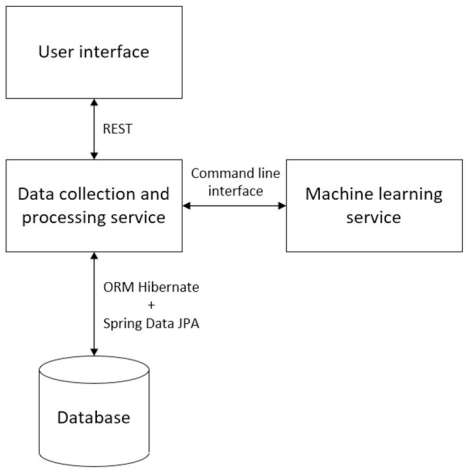

Fig. 3 – General architecture of the product

The data collection and processing service provides the interaction of the entire server part of the product with external systems, including the user interface. The interaction of the server part of the product with external systems is carried out through the REST API. The data collection and processing service receives and processes data for training or prediction, provides the correct call of the necessary machine learning methods on the input data, as well as the correct provision of reports on the results of training of machine learning models and their compositions and on the results of predictions for test-sample. The data collection and processing service provides interaction with the database to obtain the necessary information for the user, as well as to save a new, for example, trained machine learning model, composition of models, train-samples and data for class prediction, results of training and prediction. In addition to the listed functions, this service carries out authorization and authentication of users of the information system.

The machine learning service launches machine learning methods called by the data collection and processing service and returns the results for presentation to the user.

The interaction between the data collection and processing service (Fig. 4) and the machine learning service is carried out using the command line of the operating system. The command line use to invoke machine learning methods is due to the fact that both services are implemented in different programming languages, and the interaction of services through the HTTP protocol is ineffective, since it implies the transfer of large amounts of information.

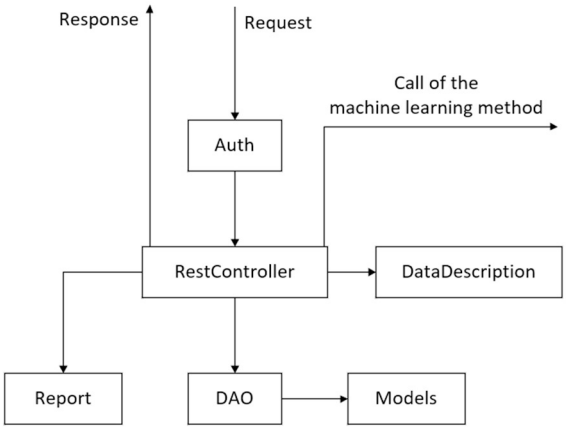

Fig. 4 – Diagram of modules for the service of data collection and processing

The RestController module processes incoming requests and sends the corresponding response. Auth module implements authentication and user rights differentiation.

The DataDescription module provides information on the characteristic types, minimum, maximum and average values, the variance of the characteristic, as well as the correlation coefficient, the name of the attribute with which it has the strongest correlation, according to the sample loaded by the user.

The Report module generates a report file in PDF format with the results of training and prediction. The report contains:

- data selected by the user for classification, with the predicted class label for each object;
- data necessary for plotting error functions for train- and test-samples;
- data necessary for plotting Precision-Recall (accuracy-completeness) graphs for train- and test-samples;
- data necessary for building ROC-curves for train- and test-samples;
- characteristics of base models and composition.

The DAO module implements a database interaction interface.

Machine learning service (Fig. 5) consists of several modules that implement the creation of models based on neural networks and decision trees, as well as their compositions, followed by training and application of input data for predictions.

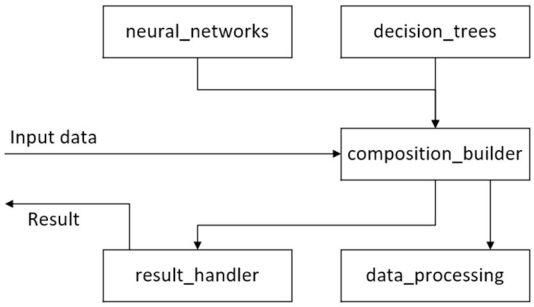

Fig. 5 – Scheme of machine learning service modules

The modules neural_networks and decision_trees respectively contain classes of neural networks and classes of decision trees implemented independently.

The data_processing module preprocesses the input sample: processing outliers and missing values.

The composition_builder module builds a composition of a given type based on the transmitted base machine learning models. The base models can be neural networks and decision trees. A user can create the following compositions: modified weighted, simple voting, probabilistic, bagging.

The result_handler module processes the results of training and composition prediction, compiles data for the report, and sends it to the data collection and processing service for presentation to the user.

The result of the work of the software product is the report files with information about the train-samples, the models used, the training results, as well as the results of the classification of objects from the test-sample.

4. CONSIDERATIONS

The initial data for building the composition originated from the information on the activities of commercial banks until 2017, about 738 financial institutions in total. Financial institutions were represented by features characterizing their financial condition, client activity, shares of individuals and legal entities, as well as features of suspicion regarding suspicious transactions and transactions with fictitious organizations, such as shares of certain types of transactions, volumes of reports within the control framework. The problem of forecasting the revocation of a license from a financial institution was being solved.

To create the composition, the neural network model and the decision tree were used as the base models. The optimal parameters were chosen for both models, in particular, four hidden layers of neurons containing a total of 180 neurons were used in the neural network model. Training and assessment of the accuracy of the classification of base models in the train- and test-samples were conducted. A probabilistic composition was compiled from the constructed models. The accuracy of the classification of basic algorithms in the train- and test-samples, as well as in their composition is presented in Table 1.

Table 1. Accuracy of classification of basic algorithms and their composition.

Model	Train-sample accuracy	Test-sample accuracy
Neural networks	82,57%	79,88%
Decision trees	82,37%	81,11%
Composition	84,38%	83,73%

Thus, we can conclude that the composition of models improves the quality of classification of individual machine learning algorithms.

5. CONCLUSION

As part of the study, the basic requirements for a software product – the constructor of compositions of machine learning models for solving classification problems – have been analyzed and determined.

The features of the architecture of the developed application, the service for data collection and processing, the machine learning service, as well as the principles of composition model formation and the parameters of its base models have been considered.

It should be noted that even a small increase in the predictive ability of the algorithm in solving an applied problem can bring significant economic benefits. Therefore, the use of more complex algorithms for constructing models is always justified, and the presence of a tool endowed with the ease of their design, respectively, is advisable.

References

[1] "Data Mining: STATISTICA Data Miner" *StatSoft.* URL: http://statsoft.ru/products/STATISTICA_Data_Miner/.

[2] "IBM SPSS Modeler – Data mining, text mining, predictive analysis" *SPSS.* URL: http://www.spss.com.hk/software/modeler/.

[3] Kazovsky Ilya Grigoryevich (2011) "The platform for building machine learning algorithms based on rules" *The Messenger of the Russian State University for the Humanities. Series "Record management and archive management. Informatics. Protection of information and information security", no. 13 (75), 2011.*

[4] "To feel neural networks or neural networks constructor" *Habr.* URL: https://habr.com/post/351922/

[5] "A Neural Network Playground" *TensorFlow.* URL: https://playground.tensorflow.org/

[6] Vorontsov Konstantin Vyacheslavovich (2012) "Mathematical methods of training on precedents" URL: http://machinelearning.ru/wiki/images/6/6d/Voron-ML-1.pdf

[7] Kaftannikov Igor Leopoldovich, and Parasich Andrey Viktorovich (2015) "Features of the use of decision trees in classification problems" *The Messenger of the South Ural State University. Series: Computer Technologies, Management, Radionics, vol. 15, no. 3, 2015, pp. 26-32.*