KDD Tools for Do-It-Yourself Analyses

PETR BERKA
University of Finance and Administration, Prague, Czech Republic

Abstract: People of three different professions must usually participate in a real-world data mining task: domain experts who understand the problem area and are able to specify the task and to assess the results, data experts who are familiar with the way how data are collected, stored and retrieved, and data mining experts who are familiar with the data mining methods and algorithms and are able to use, often very complex, data mining tools and systems. But recently, we have seen some effort towards automatization of the data mining tasks within various data mining systems. Such automation will allow the domain experts (or knowledge workers) to perform data mining tasks without cooperation with data mining experts, in a "do-it-yourself" way. The paper discusses different approaches to creating automated tools for data mining and gives examples of systems that offer different ways of data mining automatization.

Keywords: data mining systems, data mining automation, machine learning

JEL Classification: C55

1 Introduction

People of three different professions must usually participate in a real-world data mining task: domain experts who understand the problem area and are able to specify the task and to assess the results, data experts who are familiar with the way how data are collected, stored and retrieved, and data mining experts who are familiar with the data mining methods and algorithms and are able to use, often very complex, data mining tools and systems. But recently, we have seen some effort towards automatization of the data mining tasks within various data mining systems. Such automation will allow the domain experts (or knowledge workers) to perform data mining tasks without cooperation with data mining experts, in a "do-it-yourself" way. Not all parts of the knowledge discovery in databases (KDD) can be automated to the same extent. We will show which steps of the KDD process are supported by some automatization included in existing (standard) or newly created data mining tools.

The rest of the paper is organized as follows: Section 2 describes the KDD process and discusses the possibilities of its automatization, Section 3 presents different approaches to including automatization in data mining tools and gives some examples of such tools. Section 4 concludes the paper.

2 KDD Process and Possibilities for its Automatization

CRISP-DM (CRoss-Industry Standard Process for Data Mining) is a European Commission-funded project for defining a standard process model for carrying out data mining projects [1]. According to CRISP-DM, the life cycle of a data mining project consists of six phases shown in Figure 1.

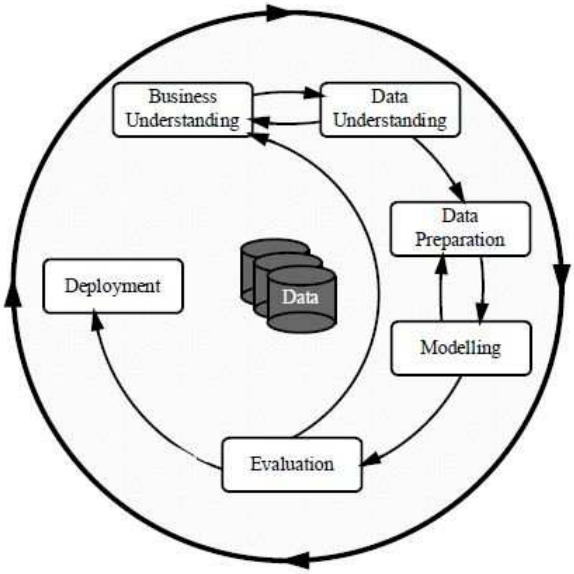

Fig. 1 *CRISP-DM Methodology*

Business understanding is the initial phase that focuses on understanding the project objectives and requirements from a business perspective, and then converting this knowledge into a data mining problem definition, and a preliminary plan designed to achieve the objectives.

The *data understanding* phase starts with initial data collection and proceeds with activities in order to get familiar with the data, to identify data quality problems, to discover first insights into the data, or to detect interesting subsets to form hypotheses for hidden information.

The *data preparation* phase covers all activities to construct the final dataset (data that will be fed into the modeling tool(s)) from the initial raw data. Data preparation tasks are likely to be performed multiple times, and not in any prescribed order. The tasks include table, record, and attribute selection as well as transformation and cleaning of data for modeling tools.

In the *modeling* phase, various modeling techniques are selected and applied, and their parameters are calibrated to optimal values. Typically, there are several techniques for the same data mining problem type. Some techniques have specific requirements for the form of data. Therefore, stepping back to the data preparation phase is often needed. (This phase corresponds with the data mining step in the narrow sense.)

At the *evaluation* stage in the project, the built model (or models) appears to have high quality from a data analysis perspective. Before proceeding to the final deployment of the model, it is important to evaluate the model more thoroughly and review the steps executed to construct the model, to be certain it properly achieves the business objectives. A key objective is to determine if there is an important business issue that has not been considered sufficiently. At the end of this phase, a decision on the use of the data mining results should be reached.

The creation of the model is generally not the end of the project. Depending on the requirements, the *deployment* phase can be as simple as generating a report or as complex as

implementing a repeatable data mining process. In many cases, it will be the customer, not the data analyst, who will carry out the deployment.

Which steps of the KDD process can be automated? Business understanding and deployment steps are closely related to the application domain, so its automatization in a general way would be very difficult or even impossible. Data understanding is already supported by computing the basic characteristics of the data. So, the added value of data mining automatization is oriented towards supporting the data preparation (pre-processing) and modeling (learning) steps.

Data pre-processing is the most time-consuming and most difficult step in the whole KDD process. The aim of this step is to select (or create) from the available data characteristics that are relevant to the given data mining task. Proper data preparation is crucial for successful modeling as we can easily observe the effect known as „garbage in, garbage out“. A lot of pre-processing is closely related to the understanding of the domain, but still, there is a number of data transformations that can be done automatically.

The modeling (learning) step can be understood as a search of the space of concept descriptions (in this case, we learn both structure and parameters of the model), or as approximation within a class of models (in this case, we learn “only” parameters of the model). In the latter case, we need some insight into the problem and intuition about the expected results to properly choose the class of the models. This choice, recently known as „hyperparameter tuning“ (an example is configuring of an artificial neural network), is another space for applying some automation.

3 Approaches to Data Mining Automatization

We will discuss three different approaches to data mining automatization: extending existing data mining systems, creating a lite, easy-to-use version of a standard tool, and creating a completely new system.

3.1 Extending Standard Data Mining Systems

Some already existing systems have been extended by components enabling automatization of particular steps. The advantage of this approach is that there is a (usually large) community of users who are already used to working with the standard version. Let us consider Weka and RapidMiner as examples of this approach.

Weka (Waikato Environment for Knowledge Analysis) is a data mining system that has been under development at the Waikato University at New Zealand since 1997. Weka was the first widely used data mining tool. It gained its popularity because it was described (and thus recommended) in a well-known textbook on data mining by Eibe Frank and Ian Witten (the first edition in 1999, the latest, third edition in 2011 [8]). Weka integrates a large number of different machine learning algorithms (oriented mainly on classification and prediction tasks) and data pre-processing methods. The authors of Weka also encourage the users to include their implemented algorithms in the system. Weka offers four modes of operation: a simple command-line interface, Explorer (for a single analysis in an easy-to-use standard Windows interface), Experimenter (to create a batch of runs in a standard Windows interface) and KnowledgeFlow (for a single analysis using a graphical interface that allows composing a task as a graph where nodes correspond to particular operations and arrows indicate the data flow). The system is available at http://www.cs.waikato.ac.nz/ml/weka/ .

To create a classification model in Weka, we have to proceed in two steps: choose a machine learning algorithm and then set its parameters. Auto-Weka, an extension that can easily be installed within Weka (https://www.cs.ubc.ca/labs/beta/Projects/autoweka/), helps in both of these steps [4]. Auto-WEKA considers the problem of choosing a suitable machine learning algorithm by simultaneously selecting a learning algorithm and setting its hyperparameters. Auto-WEKA does this using a fully automated approach based on Bayesian optimization. The difference between a "standard" Weka and Auto-WEKA is illustrated in Fig. 2 and Fig. 3. While in "standard" Weka, when applying logistic regression, one has to fill in inputs shown in Fig. 2, Auto-WEKA (in our example shown in Fig. 3) selects logistic regression as the best algorithm and finds optimal parameters for this method. So, Auto-WEKA will help non-expert users to more effectively identify machine learning algorithms and hyperparameter settings appropriate to their applications, and hence to achieve improved performance.

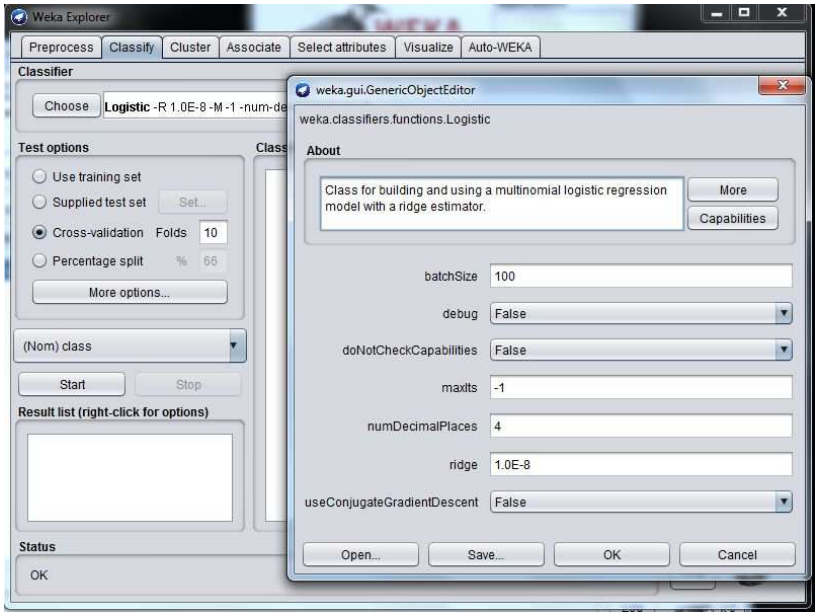

Fig. 2 *Standard input for logistic regression in Weka*

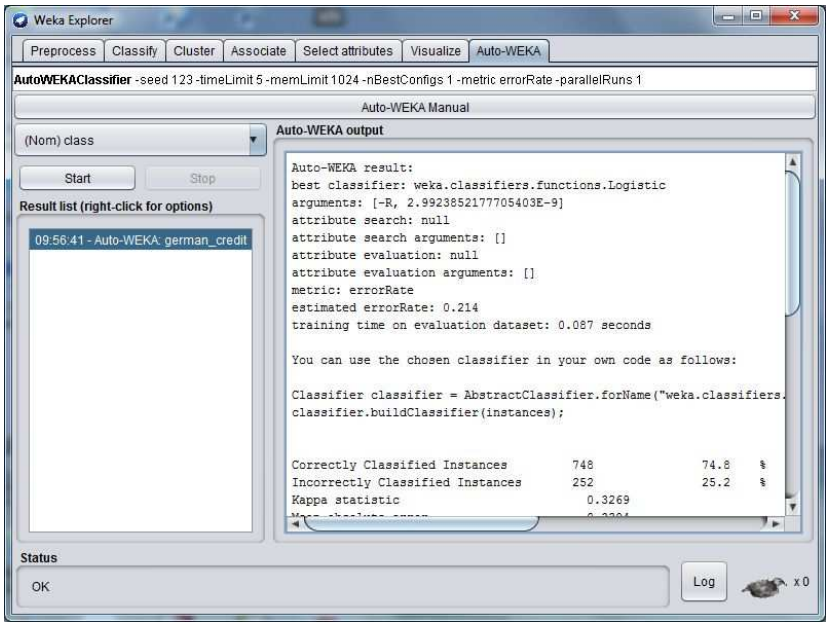

Fig. 3 *Auto-WEKA results showing tuned logistic regression*

RapidMiner, Inc. (https://rapidminer.com) is developing an open-source data science platform. The company focuses on the creation, delivery, and maintenance of predictive analytics. It offers RapidMiner Studio, a visual programming environment for predictive analytic workflows; RapidMiner Server that enables users to share, reuse, and operationalize the predictive models and results created in RapidMiner Studio; and RapidMiner Radoop that provides a graphical environment for big data analytics using Hadoop and Spark.

The main data mining product is RapidMiner Studio, an integrated visual environment for data preparation, machine learning, deep learning, text mining, and predictive analytics (see Fig. 4. for a snapshot). RapidMiner Studio functionality can be further extended with additional plugins from the RapidMiner Marketplace in an easy way, similar to downloading applications into a mobile phone. The development of the system started in 2001, at the University of Dortmund, under the name YALE (Yet Another Learning System), so the company emerged from a spin-off at that university (the company is now based in Boston, US). RapidMiner Inc. has a mixture business model. RapidMiner Studio is available both as a free system (the so-called community edition with the limit of 10 000 rows in the data table) and as a paid enterprise edition [3]. The standard way of using Rapid Miner is to create a process as shown in Fig. 4.

Rapid Miner offers (in the paid version) two extensions towards automatization of the KDD process: Turbo Prep and Auto Model. Turbo Prep aims at intuitive data preparation. This extension allows users to explore and visualize the data interactively, simplifies data cleansing (automatically removes low quality and correlated data columns), and merges multiple datasets together by automatically identifying matching columns to merge. Fig. 5 shows the auto-cleansing option, which is a part of Turbo Prep extension. Auto Model finds the best model using multiple machine learning algorithms and hyperparameter optimization. Besides this, Auto Model performs automated feature engineering to improve the model accuracy. Auto Model can be used for classification (prediction), segmentation and outlier detection. Fig. 6

shows the Model types selection step of Auto Prep extension; Fig. 7 shows the results of the creating of models. It should be mentioned that there is also a web-based version of Auto Model accessible at https://automodel.rapidminer.com .

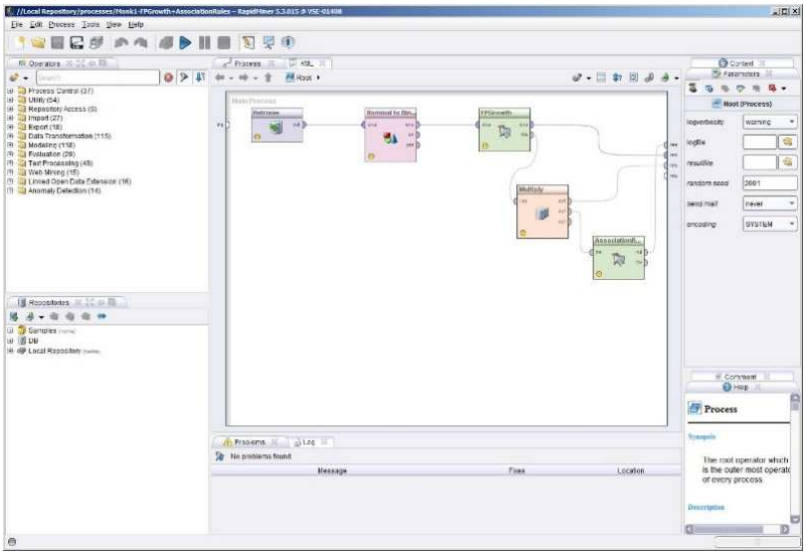

Fig. 4 *Rapid Miner Studio*

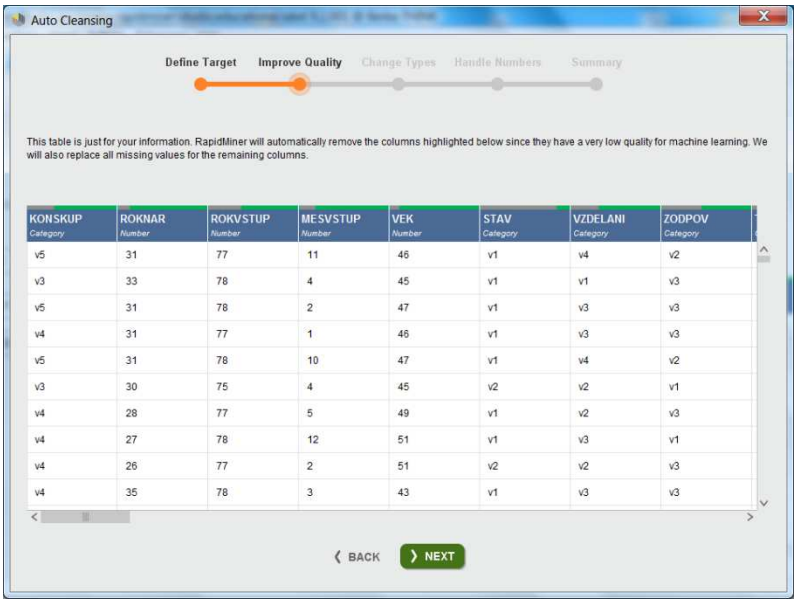

Fig. 5 *Auto Cleansing option within Rapid Miner Turbo Prep*

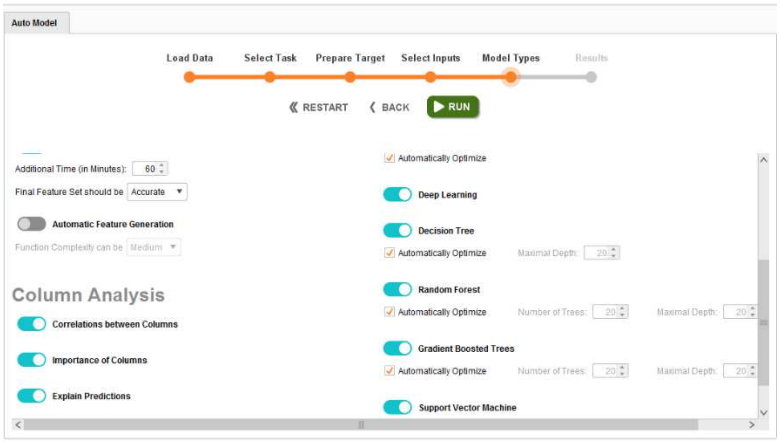

Fig. 6 *Setting parameters for Auto Model in Rapid Miner*

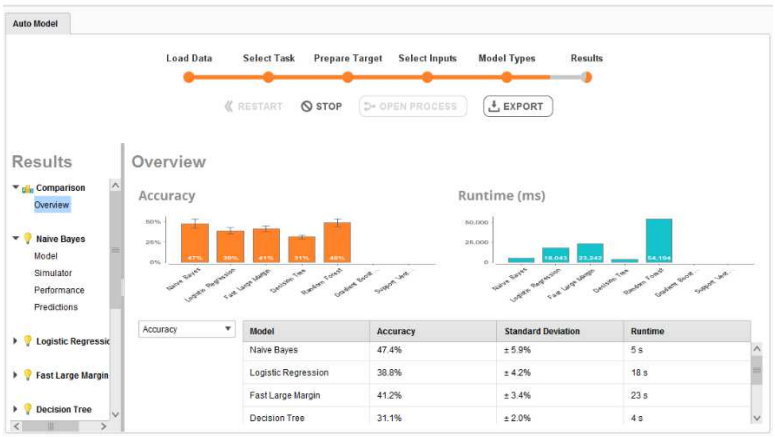

Fig. 7 *Auto Model results*

3.2 Creating Lite, Easy-to-use Version of a Standard Tool

The next possibility of how to create a tool for an unskilled user is to create a lite version of an existing KDD system by hiding unnecessary details. This approach is illustrated on the LISp-Miner system.

LISp-Miner, a freely available system that has been developed at the University of Economics, Prague, since 1996, implements various procedures that mine for different types of (mostly rule-like) knowledge patterns (http://lispminer.vse.cz). The typical knowledge patterns have the form of relations between two conjunctions derived from values of categorical attributes (columns) of an analyzed data table. Various types of relations between these conjunctions are used, including relations corresponding to statistical hypothesis tests. An interested reader should refer to [5] for more information.

The LISp-Miner data mining procedures require the setting of many parameters that allow fine-tuning of the analysis. Consider, for instance, the 4ft-Miner procedure that looks for association rules in the form

$$\phi \approx \varphi / \gamma \tag{1}$$

where ϕ (called antecedent), φ (called succedent) and γ (called condition) are cedents and ≈ (called quantifier) defines a type of relation which is evaluated on the subset of examples that satisfy the condition. When working with this procedure, we have to specify:

- the definition of partial cedents for antecedent, succedent and condition respectively,
- maximum length of antecedent, succedent and condition,
- the quantifiers ≈ and threshold values for their values,
- other task parameters, e.g. how to handle missing values.

Fig. 8 shows the necessary input to run the 4ft-Miner procedure within LISp-Miner.

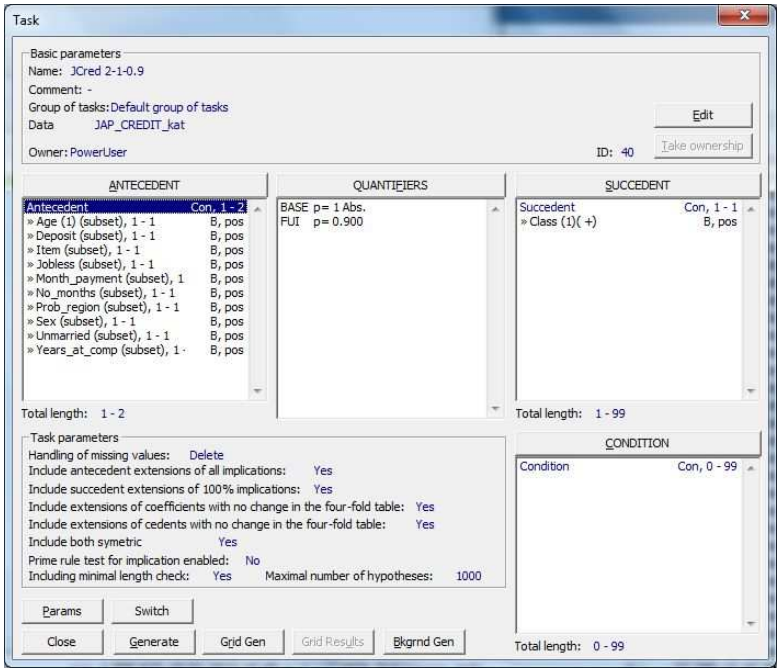

Fig. 8 *Parameter setting for 4ft-Miner procedure*

To simplify the parameter setting, reasonable default values are assigned to these parameters. But this might not be sufficient for an unskilled user. Therefore, a lite version of LISp-Miner, called EasyMiner (https://www.easyminer.eu), has been created. EasyMiner is a web-based system for interpretable machine learning based on frequent itemsets. It currently offers association rule learning (apriori, FP-Growth) and classification (CBA) [7]. Fig. 9 shows how to input the same information as shown in Fig. 8 into EasyMiner - for the „full“ system.

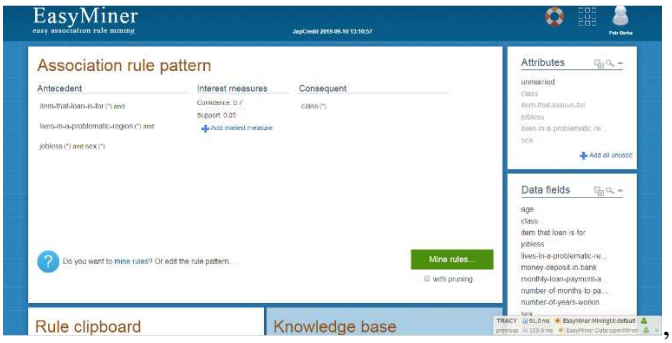

Fig. 9 *Parameter setting in EasyMiner*

3.3 Creating a Completely New System

The last option discussed in the paper is to create a completely new system. H2O driverless AI and DataRobot are examples of this approach.

H2O driverless AI is an automatic machine learning platform developed by an open-source software company (https://www.h2o.ai/products/h2o-driverless-ai/). H2O driverless AI employs the techniques of expert data scientists in an easy to use application that helps unskilled users to develop trusted machine learning models. The platform automatizes a number of KDD steps or their parts. It offers automated feature engineering for the data preparation step and automatically generates scoring pipelines to deploy the models. The platform also provides robust interpretability of machine learning models to explain modeling results and automatically generates visualizations and creates data plots that are most relevant from a statistical perspective and thus very useful for the data understanding step. Not only tabelar data but also time series and textual data can be automatically pre-processed in the system [2].

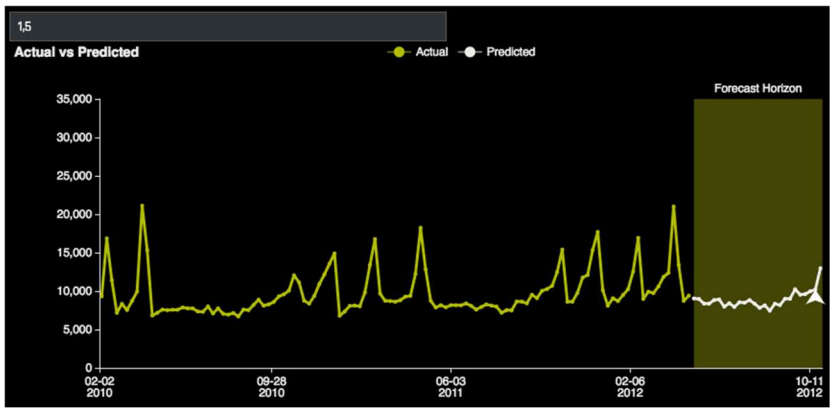

Fig. 10 *Time series prediction by H2O driverless AI*

DataRobot automated machine learning platform empowers users to build highly accurate predictive models quickly, easily and with full transparency (https://www.datarobot.com/). Incorporating the best practices of the world's top data scientists, the system automatically selects the best machine learning algorithms, and it automatically optimizes data pre-processing, feature engineering and tuning parameters for each algorithm to create and rank highly accurate models. DataRobot also offers high interpretability and explainability, so you

can easily understand how models were built, as well as explain why a model made the prediction it did [6].

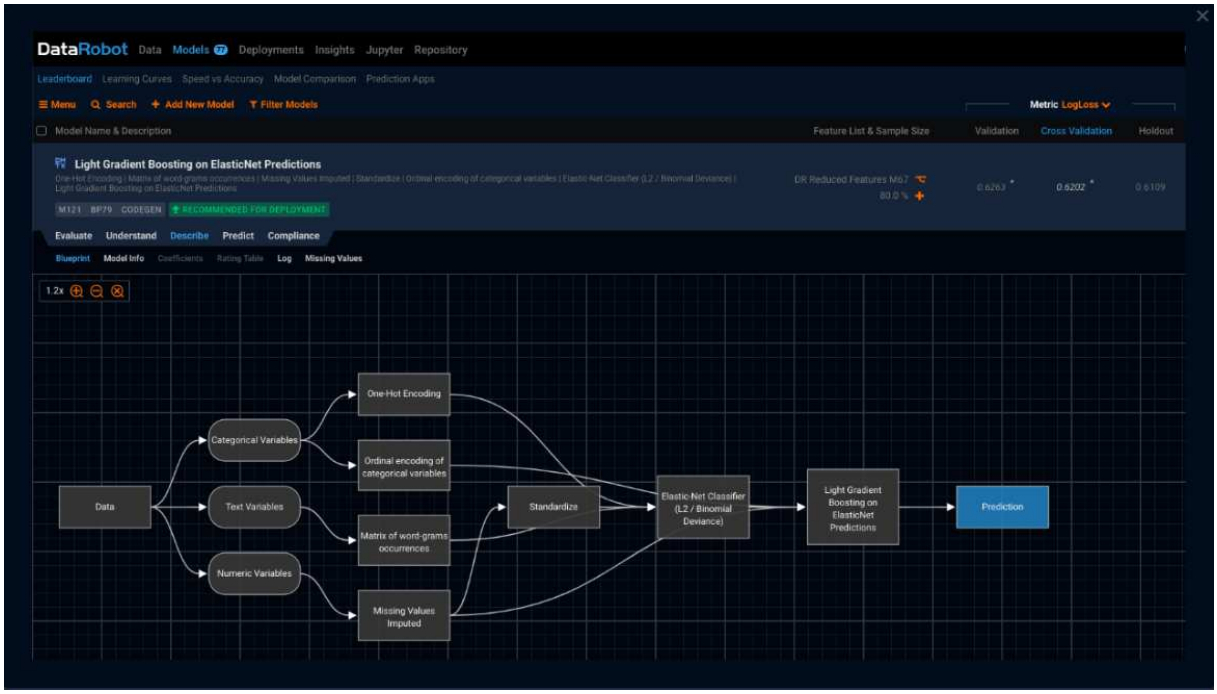

Fig. 11 *Feature engineering in DataRobot*

4 Conclusions

The idea of automated data analyses rapidly changes the way how real-world data mining tasks can be performed. The existing tools are extended by automatization capabilities, or new systems for automated machine learning appear on the market. The KDD nuggets website (http://www.kdnuggets.org) lists about 25 systems in the category "Automated Data Science and Machine Learning", some of which have been presented in the paper.

As the authors of the DataRobot platform believe: "Automated machine learning creates a new class of citizen data scientists with the power to create advanced machine learning models, all without having to learn to code or understand when and how to apply certain algorithms" [6]. So, automated data mining systems can help knowledge workers to solve data mining tasks in an intuitive, easy-to-use, do-it-yourself way.

References

1. CHAPMAN, P., CLINTON, J., KERBER, R., KHABAZA, T., REINARTZ, T., SHEARER, C. and WIRTH,R., 2000. *CRISP-DM 1.0 Step-by-step data mining guide*. SPSS Inc.
2. COOK, D. 2016. *Practical machine learning with H2O: powerful, scalable techniques for deep learning and AI*. O'Reilly Media Inc.
3. HOFMANN, M. and KLINKENBERG, R., 2013. RapidMiner: Data Mining Use Cases and Business Analytics Applications. Chapman and Hall/CRC.
4. KOTTHOFF, L., THORTNTON, C., HOOS, H.H., HUTTER, F. and LEYTON-BROWN, K., 2016. Auto-WEKA 2.0: Automatic model selection and hyperparameter optimization in WEKA. Journal of Machine Learning Research 17, pp. 1-5.
5. RAUCH, J., 2013. Observational Calculi and Association Rules. Studies in Computational Intelligece, Vol. 469, Springer.
6. UNDERWOOD, J., 2017. Data Preparation for Automated Machine Learning. Impact Analytix, LLC.
7. VOJÍŘ, S., ZEMAN, V., KUCHAŘ, J. and KLIEGR, T., 2018. EasyMiner.eu: Web framework for interpretable machine learning based on rules and frequent itemsets. Knowledge-Based Systems, Volume 150, 15, pp. 111-115.
8. WITTEN, I. H., FRANK, E. and HALL, M.A., 2011. Data Mining: Practical Machine Learning Tools and Techniques, Third Edition. Morgan Kaufmann.

Contact data:

Prof. Ing. Petr Berka, CSc.
University of Finance and Administration,
Estonská 500, 10100 Praha 10, Czech Republic
berka@vse.cz