International Journal of Engineering Education Vol. 29, No. 3, pp. 674–687, 2013
Printed in Great Britain

0949-149X/91 $3.00+0.00

White-Box Decision Tree Algorithms: A Pilot Study on Perceived Usefulness, Perceived Ease of Use, and Perceived Understanding*

BORIS DELIBAŠIĆ, MILAN VUKIĆEVIĆ and MILOŠ JOVANOVIĆ
Faculty of Organizational Sciences, University of Belgrade, Jove Ilica 154, Belgrade, Serbia.
E-mail: {boris.delibasic, milan.vukicevic, milos.jovanovic}@fon.bg.ac.rs

The mainstream in undergraduate data mining algorithm education is using algorithms as black-boxes with known inputs and outputs, while students have the possibility to adjust parameters. Newly proposed white-box algorithms provide students a deeper insight into the structure of an algorithm, and allow them to assemble algorithms from algorithm design components. In this paper a recently proposed data mining framework for white-box decision tree algorithms design will be evaluated. As the white-box approach has been experimentally proven very useful for producing algorithms that perform better on data, in this paper it is reported how students perceive the white-box approach. An open source data mining platform for white-box algorithm design will be evaluated as technologically enhanced learning tool for teaching decision tree algorithms. An experiment on 51 students was conducted. A repeated measures experiment was done: the students first worked with the black-box approach, and then with the white box approach on the same data mining platform. Student's accuracy and time efficiency were measured. Constructs from the technology acceptance model (TAM) were used to measure the acceptance of the proposed platform. It was concluded that, in comparison to the black-box algorithm approach, there is no difference in perceived usefulness, as well as in the accuracy of produced decision tree models. On the other hand, the black-box approach is easier for users than the white-box approach. However, perceived understanding of white-box algorithms is significantly higher. Evidence is given that the proposed platform could be very useful for student's education in learning data mining algorithms.

Keywords: white-box algorithms; decision trees; perceived usefulness; perceived ease of use; perceived understanding

1. Introduction

Data mining for undergraduate students is being offered at a large number of universities worldwide [1]. Data mining is prevalently thought with the black-box (BB) approach paradigm. The most famous textbooks for data mining courses are [2, 3]. These course books utilize the BB approach.

The BB approach enables users to use predefined algorithms, to set parameters and to retrieve models which help them detect regularities (patterns) in data. In this way, ease of use is achieved, since users do not have to understand the underlying process of the algorithms they use. BB approach users have to search for the best suitable algorithm (and optimal parameters) for the data at hand on a trial-and-error basis, which is a time consuming process.

This paper evaluates a newly proposed paradigm in data mining, i.e. the white-box approach (WB). It is based on the idea of assembling algorithms through algorithm parts which solve certain algorithm sub-problems (e.g. how to evaluate a decision tree split). These algorithm building-blocks enable the student a deeper insight into the algorithm structure. Students are taught that algorithm building blocks can be found not only in existing BB algorithms, but also in partial improvements of solutions for algorithm sub-problems. Algorithm parts exchange can improve algorithm performance, as reported in [4].

In [5] several WB decision tree algorithms were analysed and a generic decision tree structure which can reproduce a large family of known black-box decision trees has been proposed. This structure allows creating even more new algorithms by algorithm parts exchange of black-box algorithms and their improvements. The same authors propose a generic structure that consists of sub-problems which are parts of algorithms that must be solved in order to make an algorithm work (e.g. decide what evaluation measure should be used for splits evaluation). There are many algorithmic components also known as reusable components for solving the sub-problems (e.g. for the evaluate split sub-problem there is: gain ration, gini index, etc). By combining reusable components (RCs) through sub-problems, a plethora of algorithms can be designed. [4] proved that it is beneficial to use these algorithms as interchange of RCs with white-box algorithm design which can produce more accurate decision tree models than the original algorithms.

Besides, finding the near optimal algorithms for

* Accepted 15 January 2013.

the data at hand [4] showed that designing white-box algorithm enables more detailed characterization of algorithms and potentially better evaluation of algorithm parts and algorithms as units. On the other hand, there is no mathematical proof that the algorithmic parts in black-box algorithms are optimally assembled, i.e. there can be datasets where these algorithmic parts or their synergy prevent algorithms to perform well. White-box design enables overcoming hard bindings of components in black-box algorithms that can, depending on data, prevent an algorithm from performing well. On the other hand, in white-box design it is possible to combine RCs freely so this problem can occur rarely.

A major problem that emerged with white-box algorithm design is an abundant pool of available algorithms, making it even harder to choose an appropriate algorithm for the data at hand. Therefore, [6] proposed an evolutionary procedure for searching for the best RC-based decision tree algorithm and its parameterization.

In this paper, we will evaluate how the white-box algorithm design approach is perceived by students, and whether this approach is suitable for educating students in data mining algorithms.

We made a pilot experiment, testing students' acceptance of the white-box approach with two constructs from the famous technology acceptance model [7], i.e. perceived usefulness and perceived ease of use. In addition to testing these constructs, we also tested perceived understanding as, due to the white-box approach, this could be one of the benefits of using white-box algorithm design for students.

In this paper, we evaluate the open-source data mining platform for white-box decision tree design WhiBo [8] as a potential technologically enhanced learning tool for teaching data mining algorithms. As there is no formal proof why algorithm components should be assembled like in black-box algorithms, we believe that engineering education can benefit from using the WhiBo platform, as it can stimulate their understanding of algorithms and additionally enable them to achieve better results in data mining.

2. Related work

The paramount approach in data mining algorithm design and application is black-box approach identified top 10 most popular data mining algorithms. All of these are black-box algorithms. Although black-box algorithms are the prevailing way of using algorithms, it has recently been shown that white-box algorithms can achieve better results than black-box algorithms [4].

With the development of reusable design in architecture [10] and software engineering [11] white-box design was also proposed in data mining [12, 5] as it could supposedly produce better algorithms while combining advances of existing algorithms.

This is also supported by the series of NFL theories where it is shown that for every problem (dataset) one can theoretically assemble the best algorithm, while this algorithm can perform poorly even on datasets that differ only slightly from original datasets [13]. Therefore, in [4] it is shown that, for a problem, it is reasonable to search for the best RC interplay, instead of choosing the best among the predefined black-box algorithms. The search for the most accurate algorithm, however, is an optimization problem and it is difficult due to many available algorithms and parameters settings.

The white-box approach is distinctive for its greater openness to system design, than the black-box approach, which allows users and developers to have deeper insight of algorithm characteristics [4]. Openness of systems has always been an interesting research topic. [14] has done a research in the area of marketing DSS openness. She analyzed the influence that openness has on mental model quality, experience, decision confidence [15], and intensity of use. The author shows that openness decreases the reliance effect [16] but does not have influence on the decision makers' evaluation of their decision. [17] show that transparency (openness) has a positive impact on user's trust and user's acceptance of a content-based art recommender. However, showing how certain the system was of a recommendation had no influence on the trust and acceptance.

To evaluate the proposed approach and open-source framework for white-box algorithm design, we utilized constructs from the technology acceptance model (TAM) [7], i.e. perceived usefulness, and perceived ease of use. These constructs are the key determinants for actual user acceptance of a model. TAM has been widely accepted and applied in many areas similar to the research in this paper. E.g. [18] proposed a user acceptance model for open-source software which analyzed software quality, system capability, social influence, and software flexibility, and their influence on perceived usefulness and perceived ease of use, which further influenced intention to use and usage behavior. [1] analyzed user acceptance of enterprise resource planning systems. They showed that user's perception or perceived usefulness, easy usage, and level of intrinsic involvement affected user's intention to use an ERP. [20] showed that user guidance influenced both perceived usefulness and ability to learn, which further influenced user satisfaction. [21] evaluated an intelligent DSS and showed that perceived understanding had positive impact on perceived usefulness. Although the proposed system had low ease of use, it ranked well in all other analyzed criteria due to its openness. Although open systems are generally more complex, [22] provides evidence that although users tend to use less complex models, they are willing to use more complex models if the benefits shown to them are ''made more salient''.

Perceived understanding of users has been found to contribute to successful adoption of a system [16, 23, 24]. [25] identified perceived understanding as one of the key factors influencing analyst's decision to continue to use conceptual modeling.

The interaction users have with decision trees was previously researched as decision trees produce, per se, comprehensive models. [24] analyzed accuracy, response time and answering confidence of users working with comprehensible models (decision tables, decision trees and rule-based predictive models). [26] proposed an interactive decision tree classifier in Weka [27] and reported how experts could interactively be involved in building decision tree models. [28] made a pilot study how users interacted with machine learning system. They evaluated the email spam filter, how users understood it and how they could better interact with it. [29] evaluated effectiveness of game based learning and influence of cognitive styles. The same authors evaluated knowledge and game based learning with a model driven approach [30].

Teaching students algorithms is also an interesting research topic. E.g. [31] made a pilot study about a computer environment for teaching beginners to sort algorithms. [32] further evaluated this environment taking into account gender and learning styles as well.

3. White-box decision tree algorithms

In [5], authors proposed RCs for white-box decision tree design. In Table 1 we show RCs used in the experiment in this paper, as well as their parameterization. RCs are grouped according to sub-problems they resolve. Every sub-problem has standardized I/O structure, and can be solved with one or more RCs.

Every sub-problem used in the experiment will be shortly described.

Create split: In order to make decision tree grow, splits have to be created. Splits are dependent on the type of attributes. Therefore, in Table 1 there is a sub-problem for numerical, as well as for categorical attributes. For CSN, there is only one RC, i.e. ''binary'' so there is only the possibility either to choose or not to choose this RC (2 options).

For the CSC sub-problem there are three RCs:

1. Binary: that produces two branches, and has the effect to produce deep and thin trees,
2. Multiway: that produces as many branches as there are categories in categorical attributes, thus producing shallow and wide trees, and
3. Significant: that groups statistically similar categories in one branch, and can thus produce between binary and multiway branches (inclusive binary and multiway) dependent on

Table 1. Decision tree sub-problems and RCs with corresponding parameters

Sub-problem	Reusable component	Parameters
Create split (Numerical) abbreviation: **CSN**		
	BINARY ''bn''	
Create split (Categorical) abbreviation: **CSC**		
	BINARY ''bc''	
	MULTIWAY ''mc''	
	SIGNIFICANT ''sc''	merge alpha (def. 0.05, min 0, max 1), split alpha (def. 0.05,min 0, max 1)
	ALL	
Evaluate split abbreviation: **ES**		
	CHI SQUARE ''cs''	
	INFORMATION GAIN ''ig''	
	GAIN RATIO ''gr''	
	GINI ''gi''	
	DISTANCE MEASURE ''dm''	
Stop criteria abbreviation: **SC**		
	MAXIMAL TREE DEPTH ''mtd''	tree depth (def. 10000, min 1, max 10000)
	MINIMAL NODE SIZE ''mns''	node size (def. 1, min 1, max 1000)
Prune tree abbreviation: **PT**		
	PESSIMISTIC ERROR PRUNING ''pep''	pruning severity (def. 0.0001, min 0.0001, max 0.5)
	MINIMAL LEAF SIZE ''mls''	leaf size (def. 1, min 1 max 1000)

whether there are significant attributes between categories or no.

Additionally, one can choose to select none of these RCs (this is only possible if CSN RC was previously chosen), to choose only one RC, to choose all RCs at the same time, or to choose any combination of RCs together (e.g. binary and multi-way). Therefore, there are totally 7 options of combining the three RCs in the CSC sub-problem.

Evaluate split: Candidate splits have to be evaluated with an evaluation measure. This is an obligatory step and splits can only be evaluated with one RC in a decision tree model. Therefore, the users have 5 options available.

Stop criteria: Trees have natural stopping criteria, i.e. when all cases from a dataset are assigned to leaves (terminal nodes), or when all leaves are pure. Choosing RCs from this sub-problem is, therefore, optional. As there are two RCs available, and because it is possible to choose either one or both of these together, there are totally 4 options users can choose when selecting RCs for this sub-problem.

Prune tree: After a tree is grown, it can be pruned optionally so its complexity is reduced while classification accuracy improves (a solution for handling the "over fitting" problem). A user, according to Table 1, has the option to choose "pep", "mls", no RC, or both together, thereby making 4 options for a user.

Therefore, with this setup of white-box design (Table 1), a user has the option to build 1279 (2*7*5*4*4 –1) algorithms by combining the available RCs for solving sub-problems. On the other hand, we used the white-box algorithm design environment to reconstruct popular black-box decision tree algorithms (C4.5, CART, and CHAID). This was done to achieve fairness between algorithm comparisons [33]. They were reconstructed as:

1. C4.5: CSN = "bn"; CSC = "mc"; ES = "gr"; SC = "mtd", "mns"; PT = "pep", "mls"
2. CART: CSN = "bn"; CSC = "bc"; ES = "gi"; SC = "mtd", "mns"; PT = "pep", "mls"
3. CHAID: CSN = "bn"; CSC = "sc"; ES = "cs"; SC = "mtd", "mns"; PT = "pep", "mls"

The default values for parameters "mtd", "mns"; "pep", and "mls" were set to such values that don't influence a decision tree model. This was done with purpose, as users could, by changing these parameters, produce variations of algorithm, making 48 possible variations of algorithms in total.

Although there is a huge misbalance between the number of white-box and black-box algorithms, it is necessary, because it reflects the nature of things. Note that there are only two more RCs in white-box approach for evaluation split that haven't been used in black-box algorithms (information gain, and distance measure). Hence, by just combining RCs that are held within black-box algorithms, 671 white-box algorithms could be produced. By adding even more RCs for each sub-problem, the number of possible combinations grows rapidly.

4. Experimental evaluation

We conducted the experiment on 51 senior year management students doing a course in business intelligence who had already been acquainted with popular black-box decision tree algorithms. The experiment was conducted on the Rapid Miner data mining platform version 4.4. The experiment consisted of two parts:

1. Searching in ten trials for the most accurate decision tree with the black-box (BB) approach.
2. Searching in ten trials for the most accurate decision tree with the white-box (WB) approach.

The task of the users was to choose algorithms and setup parameters in the black-box approach, and then to design and setup parameters with the white-box approach. As students had already been familiar with the black-box approach through standard courses at the Faculty, we opted for the students to work first with the black-box approach, and then with the white-box approach.

Before the experiment started, the students had been told what the goal of the experiment had been, and they had been shown, as an example, how to use C4.5 and how to design a white-box algorithm of their own. The participants received user manuals (Appendix B) which should help them to understand quickly the algorithm parameters as well as RCs for white-box users.

Data mining streams were set like in Fig. 1. The dataset was divided into train and test dataset (2:1), but the students were unaware of that proportion. After each trial students received reports of the achieved accuracy of a decision tree model. The students wrote down the achieved accuracy of each model.

When working with the black-box algorithms, students were able to choose among three algorithms and setup parameters. Fig. 2 shows default parameters for C4.5 and CART, that take the same values as in white-box algorithms RCs (default values don't have an influence on decision tree growth). The students were told that parameters default values had been chosen in a way that doesn't influence decision tree growth. The only parameters that could have influence with the default values

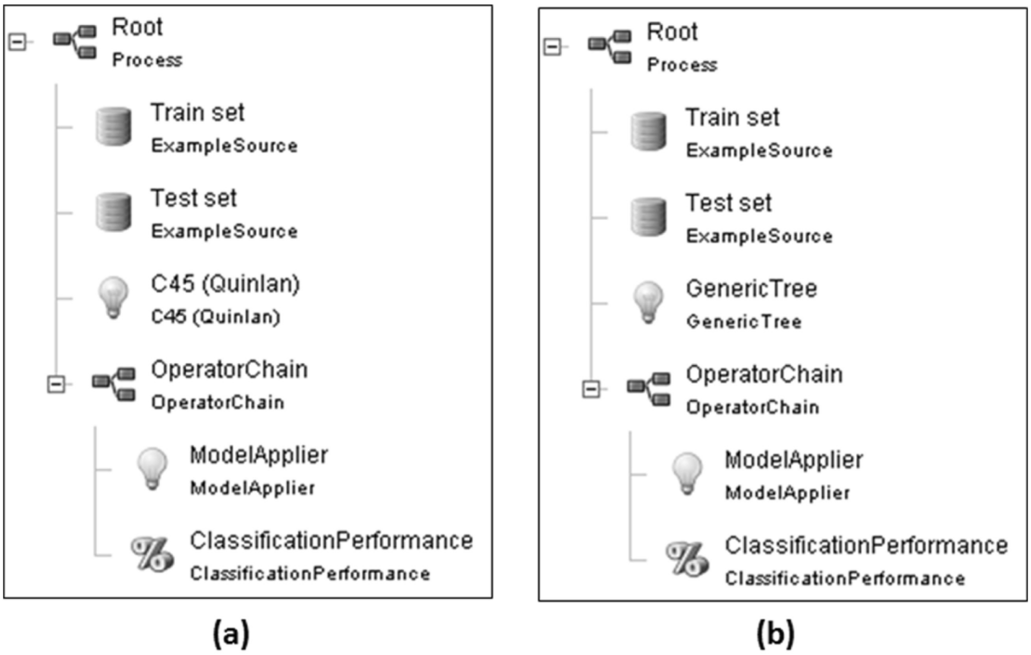


Fig. 1. Rapid Miner Streams: (a) black-box (b) white-box.

max_tree_depth	10000
min_node_size	1
min_leaf_size	1
prunning_severity	1.0E-4

Fig. 2. Parameter adjustment for C4.5 and CART.

were the merge and split parameters in CHAID, and the students were aware of that.

In the second part of the experiment, participants worked with white-box algorithms. They then designed algorithms and set the parameters of RCs as shown in Fig. 3.

We used the WhiBo plug-in [8] for Rapid Miner [34] as the experimental environment in which we evaluated the white-box approach. Within the WhiBo environment, we used CART, CHAID and C4.5 as the black-box algorithms, and the WhiBo decision tree designer (Fig. 3) for building white-box decision-trees.

WhiBo decision tree designer contains four adaptive panels (Fig. 3):

- The left panel contains several buttons. Every button represents a sub-problem. When a sub-problem is selected, the upper central panel shows available RCs for solving it.
- The upper central panel allows users to choose an RC and save it for solving a sub-problem. The lower central panel shows parameters for a selected RC. Users can also choose multiple RCs for some sub-problems (e.g. multiple stopping criteria).
- The right panel documents the designed generic tree algorithm (selected RCs and their parameters).
- The top panel contains options for creating new, saving current or opening existing white-box algorithms.

All participants were randomly assigned to a dataset. All students received a dataset description (attributes and their values, label attribute and its value) and the goal of the classification. From these, students could find out what data types they should work with, how many categories there are in categorical attributes, how many attributes there are in

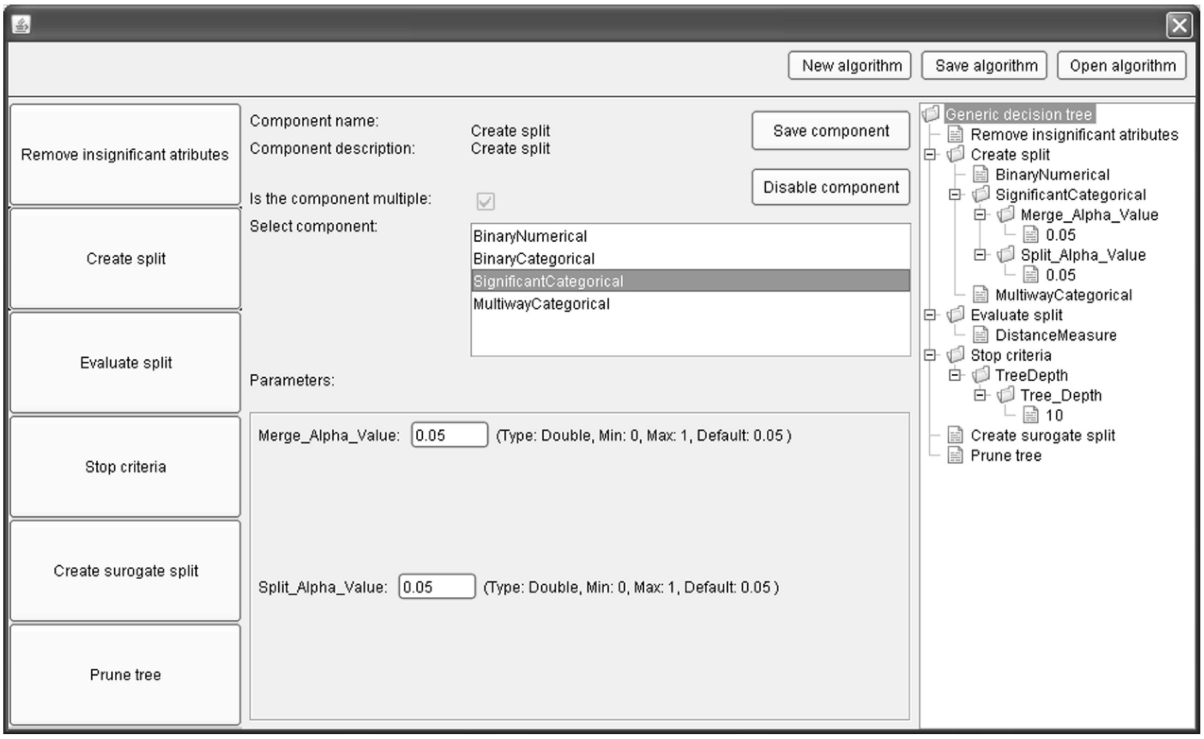


Fig. 3. White-box algorithms design and parameter setup

the dataset etc. The students were randomly assigned to one of the following datasets (Table 2):

1. Car evaluation (car), available through UCI [35],
2. Nursery (nur), available through UCI [35], and
3. Telco (tel), a churn dataset available through SPSS.

Students were assigned to "car" (18), "nur" (16), and "tel" (17) datasets. We tested users on three datasets because we didn't want the results to be dataset-dependent, as it is common in data mining, for algorithms performance to depend on dataset characteristics. We showed later that datasets had no impact on students' answers to the questionnaire they were given after they had evaluated the algorithms.

Datasets were chosen to be "people-oriented", i.e. the classification problem was to be understandable to students (they introduced the meaning and characteristics of each dataset). They were also chosen according to the number of significant differences that could be found on a dataset when comparing algorithms in pairs. From [4] we knew that on "car" and "nursery" datasets 58% and 43% of significant differences were found in 80 component-based algorithms paired comparisons, which means that students could find more accurate algorithms more easily. On the other hand, using the same methodology, we found only 4% of significant differences on "telco" which means that, for students, it would be more difficult to find significantly more accurate algorithms. In this way, we had two datasets where there were a lot of significant differences and one dataset where there were only a small number of these differences. Additionally, on the "car" dataset black-box algorithms CART and CHAID were in the group of most dominant algorithms. Finally, we set the following hypothesis:

H1: Users will find at least as good algorithms with the white-box approach as with the black-box approach.

[4] showed that white-box algorithms can outperform black-box algorithms. In the aforementioned research, authors tested 80 component-based algorithms and reported the most-accurate algorithms. All were white-box algorithms, but on some data-

Table 2. Datasets used in the experiment and their basic characteristics

Dataset	No. cat. attrib.	No. num. attrib.	No. records	No. classes
car	6	0	1728	4
nur	8	0	12960	4
tel	19	22	1000	2

sets, in the group of best performing algorithms, there were also well-known algorithms (e.g. on the "car" dataset). In the research in this paper, students could theoretically evaluate 1279 white-box algorithms, while they were given the chance to analyze only 10. On the other hand, students working with the black-box algorithms used 3 algorithms (C4.5, CART, and CHAID) which theoretically could produce, in total, 48 algorithm variations of the original algorithms (by using pruning and stop criteria as parameters).

White-box users had clearly a harder job to perform. Furthermore, white-box algorithms could also produce worse algorithms than black-box algorithms, as they cover a larger algorithmic space. Although the students have a larger algorithmic space to explore using the white-box approach, we expect them to find at least as good algorithms or better ones than with the black-box approach, as we believe the understanding of the white-box approach will enable students to design good decision tree models.

H2: Users will experience greater perceived understanding while working with white-box algorithms than while working with black-box algorithms.

The main benefit of white-box algorithms for users, besides the possibility to gain more accurate algorithms as shown in [4], is that they should experience a better understanding of the algorithms they use, which can lead, besides better education effects, to increased acceptance of white-box algorithms in the education process as shown in e.g. [17].

H3. White-box users will have no less perceived usefulness than black-box users

[22] showed that users are more willing to use more complex models if they are explicitly aware of the benefits they can achieve. Although white-box models are generally more complex, we believe that students will perceive their usefulness, which shouldn't be smaller than the one of black-box algorithms.

H4. White-box users will experience less ease of use than black box users

The white-box approach offers more possibilities for users. On the other hand it is more complicated, as users will have more options to perform. This will supposedly make black-box algorithms easier to use.

5. Results

We used a questionnaire to test perceived usefulness and perceived ease of use, proposed by [7]. In addition, we added six questions of our own to test perceived understanding. This questionnaire contained in total 18 items measured on a 1–5 Likert scale (see Appendix A for details). Internal consistency of these questions was measured with Cronbach alpha. All groups of questions received a fair Cronbach alpha value. Perceived understanding had a 0.73, perceived usefulness 0.76, and perceived ease of use 0.75. Due to a small number of participants, we didn't do any more sophisticated analysis which could give a clearer picture of the righteousness of the perceived understanding questions. Therefore, the reported results in this paper are reported as a pilot experiment, and would need more thorough analysis in the future.

To show that datasets had no impact on questionnaire results, we performed clustering on the questionnaire results on 18 questions about perceived understanding, perceived usefulness, and perceived ease of use. The dataset label was regarded as real clustering. If the results were different across datasets, all answers would be clustered in datasets, which would mean that the respondent results are dataset-dependent.

The number of clusters was set to be 3 as there were three datasets, and for the initial centroids of these clusters the mean of respondents that worked on a dataset were calculated. We performed standard K-means algorithms and measured two evaluation measures, i.e. adjusted rand index [36] and adjusted mutual information [37]. We obtained the following results 0.0078 and 0.0057 which indicates a non-existing clustering structure, which means that the choice of a dataset had no influence on respondent results.

We also performed ANOVA F-tests on the questionnaire results to measure whether perceived understanding, perceived ease of use, and perceived usefulness with white-box and black-box design were significantly different dependent on a dataset (Table 3). The results indicate that the selection of datasets had no influence on perceived ease of use, usefulness and understanding on the sample of 51 analyzed students.

Since white-approach enables design of large space of algorithms, we wanted to investigate stu-

Table 3. Results of ANOVA F-test of perceived understanding (WB and BB), perceived usefulness (WB and BB), and perceived ease of use (WB and BB) on three datasets

	F	Sig.
Perceived understanding WB	3.15	0.0518
Perceived understanding BB	0.54	0.5868
Perceived usefulness WB	0.08	0.9199
Perceived usefulness BB	1.08	0.3475
Perceived ease of use WB	0.15	0.8613
Perceived ease of use BB	0.93	0.4031

Table 4. Accuracy of 80 white-box algorithms from [4] on three datasets learned on $\frac{2}{3}$ of each dataset and evaluated on $\frac{1}{3}$ of each dataset

Dataset	**avg**	**std**	**max**	**min**	**max-min**
car	90.65%	2.92%	94.13%	84.28%	9.85%
nur	88.13%	1.50%	91.71%	85.11%	6.60%
tel	70.35%	4.12%	76.72%	63.13%	13.59%

Table 5. Maximal and average accuracies with standard deviations with black-box approach users achieved in 10 trials on 3 datasets and average on all datasets. Bold value is significantly better (p<0.05) compared to the white-box approach. Bold-italic value is significantly better (p<0.01) compared to the black-box approach.

Dataset	**avg**	**std**	**max**	**min**	**max-min**
car	89.10%	5.03%	**94.33%**	**80.77%**	13.56%
nur	87.97%	2.18%	91.08%	**84.26%**	6.82%
tel	72.15%	2.74%	75.82%	67.69%	8.13%
ALL	82.89%	3.36%	86.92%	***77.32%***	9.60%

Table 6. Maximal and average accuracies with standard deviations with white-box approach users achieved in 10 trials on 3 datasets and average on all datasets. Bold-italic value is significantly better (p<0.01) compared to the black-box approach

Dataset	**avg**	**std**	**max**	**min**	**max-min**
car	87.88%	6.19%	93.79%	76.13%	17.66%
nur	86.00%	10.08%	***93.14%***	66.94%	26.20%
tel	71.77%	3.01%	75.62%	66.69%	8.93%
ALL	81.72%	6.29%	87.30%	70.03%	17.27%

dents' objective understanding of WB algorithms and datasets. So we compared the results of students, to the best results students were able to achieve theoretically. In Table 4, we show 80 algorithms proposed in [4] and their achieved accuracies in the experimental setup done in this paper.

As there is, theoretically, an infinite number of algorithms that could be evaluated (due to parameter setting) we additionally searched the white-box algorithm space with an evolutionary algorithms from [6] for the most accurate algorithm. The evolutionary algorithm found:

1. on "car" dataset 41 algorithms with maximal accuracy of 98,6%;
2. on "nur" dataset one algorithm with maximal accuracy 93.6% (CSN = "bn"; CSC = "sc" (0,312; 0,036); ES = "ig"; PT= "pep" (0,031)); and
3. on "tel" dataset 32 algorithms with maximal accuracy of 78,2%.

The evolutionary search found the most accurate algorithms, as it had an efficient way to search through the RC and parameter space.

We will show how the students performed on the given task. Tables 5 and 6 show average (with standard deviation), maximal, and minimal accuracies of algorithms students tested on 3 datasets and average on all datasets with the black-box (Table 5) and white-box (Table 6) approach. Regarding the "max-min" column, we notice that WB users had a larger value, which indicates that, for them, it was more difficult to find the most appropriate algorithm.

Table 7. P-values from the independent samples from t-test. Bold and bold-italic values show significant differences on the (p<0.05) and (p<0.01) level respectively

Dataset	**avg**	**max**	**min**
car	0.2256	**0.0421**	**0.0242**
nur	0.1934	***0.0019***	**0.0234**
tel	0.4528	0.6880	0.3417
ALL	0.4708	0.8217	***0.0059***

From Table 7 we conclude that there is no difference in average achieved accuracies in all datasets. The results are, however, dependent on the choice from a dataset. On the "nur" dataset, participants managed to find more accurate algorithms with the white-box algorithms, while on "car", users found significantly better results with the black-box approach. This is, however, not surprising knowing that the black-box algorithms CART, and CHAID are in the group of the best algorithms on "car" dataset [4]. On the "tel" dataset, there were no significant differences between the white-box and black-box algorithm approach.

It is interesting, though, that those participants' average minimal accuracies were generally larger with the white-box approach, than with the black-box approach. That is also concluded on the "car" and "nur" dataset, but not on "tel". The general conclusion would be that, with the WB approach, participants can achieve both better and worse results due to many possibilities of algorithm design and parameter settings. Still, students managed to find at least as good results with the white-box approach as with the black-box approach. In this way, students also showed that they understand the white-box design as it helped them to achieve competitive results. Therefore, we accept Hypothesis 1.

We further report results on perceived understanding, perceived usefulness, and perceived ease of use.

Users' experiences with white-box algorithms show more understanding than with black-box algorithms. White-box usage had average perceived understanding of 3.62 (std. dev. 0.63) while black-box had average of 3.3 (0.76). The difference was significant on a 0.05 level ($t = 2.159$, $Sig = 0.036$), so we accept Hypothesis 2.

As perceived understanding of users contributes to successful adoption of a system [3,14,8,9], we believe that this result shows that it is expected that students will accept the proposed system, which will help them to better explore data mining algorithms and enable them to have a better understanding of the proposed algorithms.

White-box design can be applied in educating students in data mining; as except objective benefits students can achieve (better results, better testing of algorithms and their parts) users have better perceived understanding. Because of that, we believe that users will better accept the proposed system than a black-box system for educational purposes.

Regarding perceived usefulness, white-box usage scored a perceived usefulness of 3.3 (std. dev. 0.66) while black-box had average of 3.5 (0.76). The difference was insignificant ($t = -1.587$, $Sig = 0.119$) so, in this sample, there is no evidence that H3 is not true.

As for ease of use, white-box usage scored a perceived ease of use of 2.6 (std. dev. 0.7) while black-box scored 3.42 (0.9). The difference was significant on a 0.0001 level ($t = 6.188$, $Sig < 0.0001$). Black-box users perceived more ease of use than white-box users, and therefore we also accept H4.

We also report the average time needed for students to work with white box algorithms and black-box algorithms, as an objective ease of use criteria. Students needed average of 30.07 (9.27) minutes with the white-box approach and average of 19.53 (6.23) minutes for running 10 times the black-boxes algorithms. The time needed for white-box design was significantly higher ($t = 50.78$, $Sig < 0.0001$).

6. Conclusions

In this paper, we evaluated the white-box algorithm platform for decision trees—WhiBo. It is a newly proposed platform for design of algorithms and better exploration of performance of algorithms and their parts. The mainstream in engineering data mining education is using algorithms as black-boxes which hide algorithm details from the user. We were interested how the newly proposed white-box would be useful for engineering education compared to the state-of the art. Therefore, we conducted an experiment testing users' perceiving of white-box and black-box algorithms. Our findings are the following:

1. With white-box algorithms, students can find as good algorithms as with black-box algorithms. Although it was more complicated for students to search through the white-box algorithms space, they showed enough understanding to find competitive results with the black-box approach. Additionally, on two datasets, some black-box algorithms were part of the most accurate algorithms ("car", and "tel") so no difference between black-box and white-box algorithms could be found in any way.
2. Students had better understanding with the white-box approach, so it is expected, based on the findings from previous research that they would continue to use it.
3. The proposed approach is found as useful as the prevalent black-box approach.
4. The white-box approach is more difficult to use than the black-box approach. However, users are willing to use more complex models if they understand the benefits such a system could provide to them.

We conclude that students perceived WhiBo well, so we conclude that it can be beneficial to use WhiBo in the process of teaching student's data mining algorithms and data mining education, because it can help students understand the algorithms better, and achieve better results in data mining.

Acknowledgements—This research is partially funded by a grant from the Serbian Ministry of Science and Technological Development, project ID III 41008 and project ID III 47003.

References

1. F. Wu, Discussion on experimental teaching of data warehouse & data mining course for undergraduate education, *In proc. of the 7th International Conference on Computer Science & Education (ICCSE)*, Australia, 2012, pp. 1425–1429.
2. J. Han, M. Kamber and P. Horn, Data Mining: Concept and Techniques, Third Edition, *Morgan Kaufmann*, 2011.
3. P-N. Tan, M. Steinbach and V. Kumar, Introduction to Data Mining, First Edition, *Addison Wesley*, 2005.
4. B. Delibašić, M. Jovanović, M. Vukićević, M. Suknović, Obradović, Z., Component-based decision trees for classification, *Intelligent Data Analysis*, **15**(5), 2011, pp. 671–693.
5. M. Suknovic, B. Delibasic, M. Jovanovic, M. Vukicevic, D. Becajski-Vujaklija, Z. Obradovic, Reusable components in

decision trees induction algorithms, *Computational Statistics*, **27**(1), 2012, pp. 127–148.
6. M. Jovanović, B. Delibašić, M. Vukićević, M. Suknović, Optimizing performance of decision tree component-based algorithms using evolutionary algorithms in Rapid Miner, *In proc. of the 2nd RapidMiner Community Meeting and Conference*, Dublin, Ireland, 2011.
7. F. D. Davis, Information Technology Introduction, *Management Information Systems*, **13**(3), 1989, pp. 319–340.
8. M. Vukićević, M. Jovanović, B. Delibašić, M. Suknović, WhiBo—RapidMiner plug-in for component based data mining algorithm design, *In proc. of the 1st RapidMiner Community Meeting and Conference*, Dortmund, Germany, 2010, pp. 33–38.
9. X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. F. M. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand and D. Steinberg, Top 10 algorithms in data mining, *Knowledge and Information Systems*, **14**(1), 2007, pp. 1–37.
10. C. Alexander, *The Timeless Way of Building*, Oxford University Press, New York, 1979.
11. J. O. Coplien and D. C. Schmidt, *Pattern Languages of Program Design*, Addison-Wesley, 1995.
12. B. Delibašić, K. Kirchner, J. Ruhland, M. Jovanović and M. Vukićević, Reusable components for partitioning clustering algorithms, *Artificial Intelligence Review,* **32**(1–4), 2009, pp. 59–75.
13. D. H. Wolpert, The lack of a priori distinctions between learning algorithms, *Neural Computation*, **8**(7), 1996, pp. 1341–1390.
14. N. Demoulin, Marketing decision support system openness: A means of improving managers' understanding of marketing phenomena, *Decision Support Systems*, **44**(1), 2007, pp. 79–92.
15. G. Vanbruggen, A. Smidts and B. Wierenga, The impact of the quality of a marketing decision support system: An experimental study, *International Journal of Research in Marketing*, **13**(4), 1996, pp. 331–343.
16. S. H. Barr and R. Sharda, Effectiveness of decision support systems: development or reliance effect? *Decision Support Systems*, **21**(2), 1997, pp. 133–146.
17. H. Cramer, V. Evers, S. Ramlal, M. van Someren, L. Rutledge, N. Stash, L. Aroyo and B. Wielinga, The effects of transparency on trust in and acceptance of a content-based art recommender, *User Modeling and UserAdapted Interaction*, **18**(5), 2008, pp. 455–496.
18. M. D. Gallego, P. Luna and S. Bueno, User acceptance model of open source software, *Computers in Human Behavior*, **24**(5), 2008, pp. 2199–2216.
19. K. Amoakogyampah, Perceived usefulness, user involvement and behavioral intention: an empirical study of ERP implementation, *Computers in Human Behavior*, **23**(3), 2007, pp. 1232–1248.
20. F. Calisir, The relation of interface usability characteristics, perceived usefulness, and perceived ease of use to end-user satisfaction with enterprise resource planning (ERP) systems, *Computers in Human Behavior*, **20**(4), 2004, pp. 505–515.
21. K. N. Papamichail and S. French, Design and evaluation of an intelligent decision support system for nuclear emergencies, *Decision Support Systems*, **41**(1), 2005, pp. 84–111.
22. T. Chenoweth, K. Dowling and R. St. Louis, Convincing DSS users that complex models are worth the effort, *Decision Support Systems*, **37**(1), 2004, pp. 71–82.
23. J. E. Bailey and S. W. Pearson, Development of a tool for measuring and analyzing computer user satisfaction, *Management Science,* **29**(5), 1983, pp. 530–545.
24. J. Huysmans, K. Dejaeger, C. Mues, J. Vanthienen and B. Baesens, An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models, *Decision Support Systems*, **51**(1), 2011, pp. 141–154.
25. I. Davies, P. Green, M. Rosemann, M. Indulska and S. Gallo, How do practitioners use conceptual modeling in practice? *Data & Knowledge Engineering*, **58**(3), 2006, pp. 358–380.
26. M. Ware, Interactive machine learning: letting users build classifiers, *International Journal of Human-Computer Studies*, **55**(3), 2001, pp. 281–292.
27. M. Hall, E. Frank, G Holmes, B. Pfahringer, P. Reutemann and I. H. Witten, The WEKA Data Mining Software: An Update, *SIGKDD Explorations*, **11**(1), 2009, pp. 10–18.
28. S. Stumpf, V. Rajaram, L. Li, W.-K. Wong, M. Burnett, T. Dietterich, E. Sullivan, et al., Interacting meaningfully with machine learning systems: Three experiments, *International Journal of Human-Computer Studies*, **67**(8), 2009, pp. 639–662.
29. M. Milovanović, M. Minović, I. Kovačević, J. Minović and D. Starčević, Effectiveness of Game-Based Learning: Influence of Cognitive Style, *Communications in Computer and Information Science*, **49**(1), 2009, pp. 87–96.
30. M. Minović, M. Milovanović and D. Starčević, Modelling, Knowledge and Game Based Learning: Model Driven Approach, *Journal of Universal Computer Science*, **17**(9), 2011, pp. 1241–1260.
31. M. Kordaki, M. Miatidis and G. Kapsampelis, A computer environment for beginners' learning of sorting algorithms: Design and pilot evaluation, *Computers & Education*, **51**(2), 2008, pp. 708–723.
32. W. W. F. Lau, and A. H. K. Yuen, Promoting conceptual change of learning sorting algorithm through the diagnosis of mental models: The effects of gender and learning styles, *Computers & Education*, **54**(1), 2010, pp. 275–288.
33. S. Sonnenburg, M. L. Braun, C. Soon Ong, S. Bengio, L. Bottou, G. Holmes, Y. LeCun, K-R. Muller, F. Pereira, C. E. Rasmussen, G. Ratsch, B. Scholkopf, A. Smola, P. Vincent, J. Weston and R. Williamson, The Need for Open Source Software in Machine Learning, *Journal of Machine Learning Research*, **8**, 2007, pp. 2443–2466.
34. I. Mierswa, M. Wurst, R. Klinkenberg, M. Scholz and T. Euler, YALE: Rapid Prototyping for Complex Data Mining Tasks. In L. Ungar, M. Craven, D. Gunopulos and T. Eliassi-Rad (eds.), *KDD 06 Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining,* Philadelphia, USA, 2006, pp. 935–940.
35. A. Frank and A. Asuncion, UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science, 2010.
36. G. W. Milligan and M. C. Cooper, Methodology Review: Clustering Methods, *Applied Psychological Measurement*, **11**(4), 1987, pp. 329–354.
37. N. X. Vinh, Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance, *Journal of Machine Learning Research*, **11**, 2010, pp. 2837–2854.

Appendix A—The questionnaire

Perceived understanding

1. I think I could easily adapt algorithms to data.
2. Better understanding of data would help me choose better algorithms.
3. If I wasn't satisfied with the results, I had many options to try to improve my algorithms.
4. I feel that the algorithms I've used are very applicable for solving real problems.
5. I feel that I understand the algorithm design.
6. Overall, I feel I have an understanding how the algorithms work.

Perceived usefulness
1. Using the platform enabled me to accomplish tasks more quickly.
2. Using the data mining platform improved the quality of my decision tree models.
3. Using the data mining platform increased my productivity.
4. Using the data mining platform enhanced my effectiveness.
5. Using the data mining platform made it easier to build decision tree models.
6. I found the data mining platform useful for building decision tree models.

Perceived ease of use
1. Learning to operate the data mining platform would be easy for me.
2. I find it easy to get the data mining platform to do what I want it to do.
3. My interaction with data mining platform would be clear and understandable.
4. I find the data mining platform to be flexible to interact with.
5. It would be easy for me to become skillful at using the data mining platform.
6. I find the data mining platform easy to use.

Appendix B—Black-box algorithms user manual

CART
Splitting (numerical attributes): Binary „bn“
Splitting (categorical attributes): Binary „bc“
Split evaluation: Gini index „gi“
Parameters (default; min; max):
- Maximum tree depth „mtd“ (10,000; 1; 10,000)
- Minimal node size „mns“ (1; 1; 1000)
- Minimal leaf size „mls“ (1; 1; 1000)
- Pruning severity „ps“ (0.0001; 0.0001; 0.5). 0.5 means more intense pruning

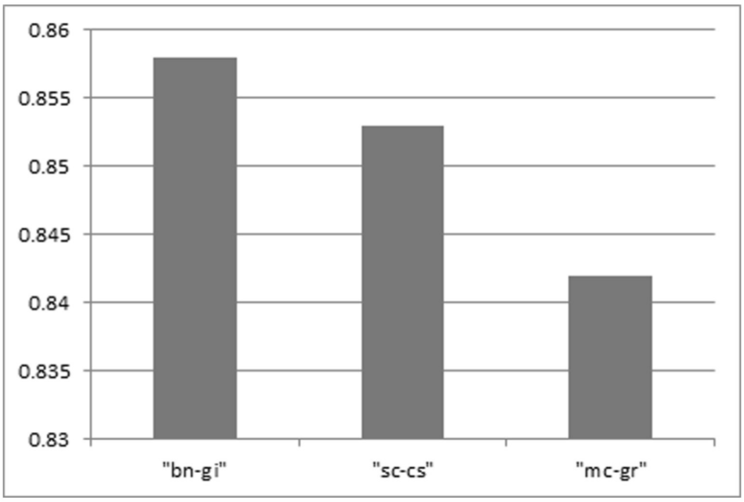
Strength: Accuracy

C4.5
Splitting (numerical attributes): Binary „bn“
Splitting (categorical attributes): Multiway „mc“
Split evaluation: Gain ratio „gr“
Parameters (default; min; max):
- Maximum tree depth „mtd“ (10,000; 1; 10,000)
- Minimal node size „mns“ (1; 1; 1000)
- Minimal leaf size „mls“ (1; 1; 1000)
- Pruning severity „ps“ (0.0001; 0.0001; 0.5). 0.5 means more intense pruning

Strength: Speed

CHAID
Splitting (numerical attributes): Binary „bn“
Splitting (categorical attributes): Significant „sc“
Split evaluation: Chi-square „cs“
Parameters (default; min; max):
- Merge parameter (0.05; 0; 1), 1 prevents merging
- Split parameter (should be *leq* Merge parameter) split threshold of previously merged categories (0.05; 0; 1), 1 – splits all merged categories
- Maximum tree depth „mtd“ (10,000; 1; 10,000)
- Minimal node size „mns“ (1; 1; 1000)
- Minimal leaf size „mls“ (1; 1; 1000)
- Pruning severity „ps“ (0.0001; 0.0001; 0.5). 0.5 means more intense pruning

Strength: Interpretability, grouping similar categories in same branches



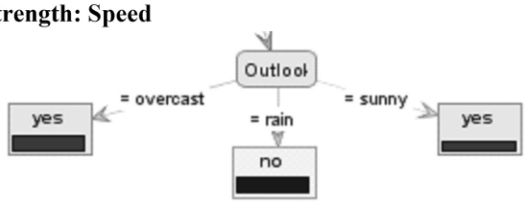
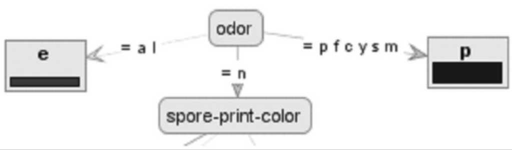
Average algorithm performance of CART „bn-gi", CHAID „sc-cs", and C4.5 „mc-gr"
(Delibasic et al, 2011)

This is generally valid, and not necessarily on every dataset

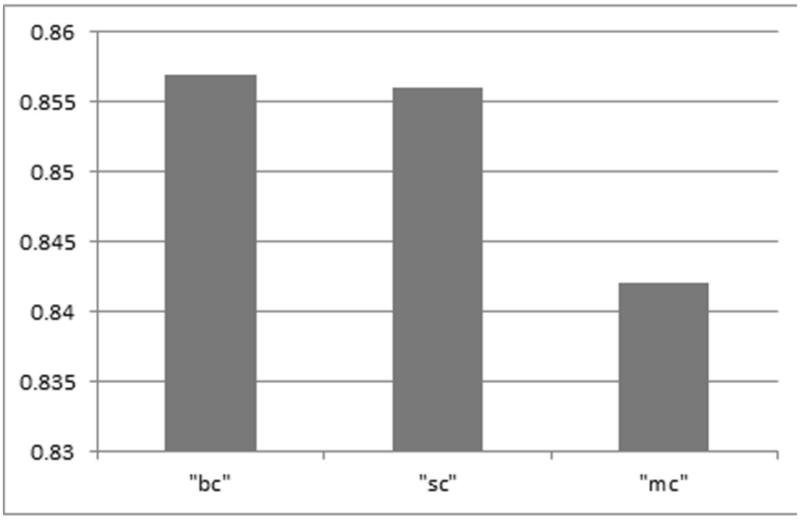
Appendix C—White-box algorithms user manual

<table>
<tr><th colspan="2">Splitting types RCs</th></tr>
<tr><th>Binary (numerical attributes)</th><th>Binary (categorical attributes)</th></tr>
<tr><td>Produces deep and thin trees
Strength: Accuracy

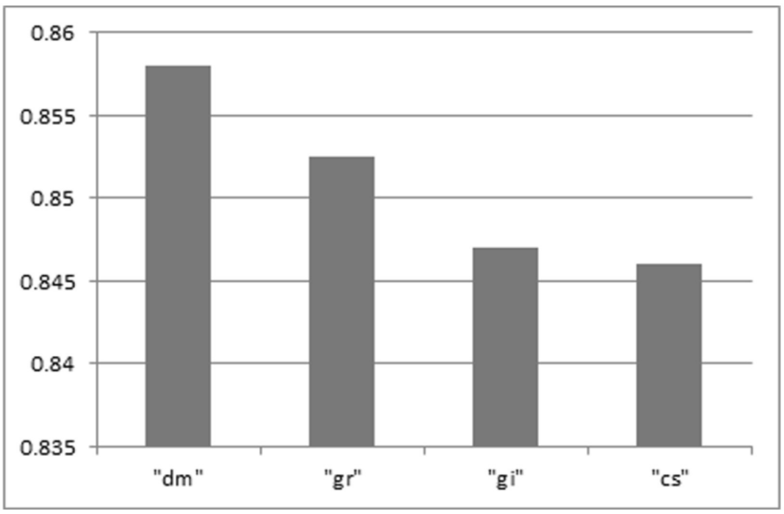
</td>
<td>Produces deep and thin trees
Strength: Accuracy

</td></tr>
<tr><th>Multiway</th><th>Significant</th></tr>
<tr><td>Produces wide and shallow trees
Strength: Speed

</td>
<td>Parameters (default; min; max):
Merge parameter (0.05; 0; 1), 1 prevents merging
Split parameter (0.05; 0; 1). Should be ≤ „Merge parameter". Split threshold of previously merged categories. 1 splits all merged categories
Produces trees with significantly grouped categories in branches.
Strength: Interpretability, Accuracy

</td></tr>
</table>

Sub-problem	Reusable component	Parameters
Evaluate split		
	CHI SQUARE “cs”	
	INFORMATION GAIN “ig”	
	GAIN RATIO “gr”	
	GINI “gi”	
	DISTANCE MEASURE “dm”	
Stop criteria		
	MAXIMAL TREE DEPTH “mtd”	tree depth (def. 10000, min 1, max 10000)
	MINIMAL NODE SIZE “mns”	node size (def. 1, min 1, max 1000)
Prune tree		
	PESSIMISTIC ERROR PRUNING “pep”	confidence (def. 0.0001, min 0.0001, max 0.5)
	MINIMAL LEAF SIZE “mls”	leaf size (def. 1, min 1 max 1000)



Average accuracies of algorithm containing split categorical RCs: binary „bc“ , significant „sc“ and multiway „mc“ (Delibasic et al., 2011)
This is generally valid, and not necessarily on every dataset



Average accuracies of algorithms containing split evaluation RCs: distance measure „dm“, gain ratio „gr“, gini index „gi“ and chi square „cs“ (Delibasic et al., 2011)
This is generally valid, and not necessarily on every dataset

Boris Delibašić is assistant professor at the University of Belgrade, Serbia, at the Faculty of Organizational Sciences, at the Department for Business Decision Making since 2007. He holds a Ph.D. for "Formalization of the Business Decision Making identification of reusable components to support the decision-making process.

Milan Vukićević is teaching and research assistant at the Faculty of Organizational Sciences, University of Belgrade, within the Center for Business Decision Making. He is currently a Ph.D. student. His main research field interests are clustering and classification algorithm design, data mining, decision support, and meta learning.

Miloš Jovanović is teaching and research assistant at the Faculty of Organizational Sciences, University of Belgrade, within the Center for Business Decision Making. He is currently a Ph.D. student. His main research field interests are data mining, decision support systems, data warehousing, optimization and artificial intelligence.