International Journal of Statistics and Probability; Vol. 3, No. 3; 2014
ISSN 1927-7032 E-ISSN 1927-7040
Published by Canadian Center of Science and Education

Comparing Decision Tree Method Over Three Data Mining Software

Ida Moghimipour[1] & Malihe Ebrahimpour[1]

[1] Faculty of Science, Department of Statistics, Islamic Azad University, Mashhad branch, Mashhad, Iran

Correspondence: Ida Moghimipour, Faculty of Science, Department of Statistics, Islamic Azad University, Mashhad branch, Mashhad, Iran. Tel: 98-51-3844-1324. E-mail: idamoghimipour@yahoo.com

Received: June 16, 2014 Accepted: July 19, 2014 Online Published: July 28, 2014

doi:10.5539/ijsp.v3n3p147 URL: http://dx.doi.org/10.5539/ijsp.v3n3p147

Abstract

As a result of the growing IT and producing methods and collecting data, it is admitted that today the data can be warehoused faster in comparison with the past. Therefore, knowledge discovery tools are required in order to make use of data mining. Data mining is typically employed as an advanced tool for analyzing the data and knowledge discovery. Indeed, the purpose of data mining is to establish models for decision. These models have the ability to predict the future treatments according to the past analysis and are of the exciting areas of machine learning and adaptive computation. Statistical analysis of the data uses a combination of techniques and artificial intelligence algorithms and data quality information. To utilize the data mining applications, including the commercial and open source applications, numerous programs are currently available.

In this research, we introduce data mining and principal concepts of the decision tree method which are the most effective and widely used classification methods. In addition, a succinct description of the three data mining software, namely *SPSS-Clementine*, *RapidMiner* and *Weka* is also provided. Afterwards, a comparison was performed on 3515 real datasets in terms of classification accuracy between the three different decision tree algorithms in order to illustrate the procedure of this research. The most accurate decision tree algorithm is *Decision Tree* by 92.49% in *Rapidminer*.

Keywords: Data Mining, Decision Tree, *SPSS-Clementine*, *RapidMiner*, *Weka*

1. Introduction

Recently, databases store and receive data from many sources or institutions. The process that converts the raw data into the information requires large amounts of storage and this is called data mining. The purpose of data mining methods in line with developments in the computer technology and programs effectively is to make huge amounts of data. Consequently, development in the computer technology and the increased amount of data held in databases have given rise to new data collection methods, automatic data collection tools, database systems, and augmented use of computers. Data mining was developed in order to combine the knowledge and experience in the software. Data mining has become necessary by rapidly growing the data records, automatic stations, satellite and remote sensing systems, space telescope scans, developments in the gene technology, scientific calculations, as well as simulations and models. Data mining is one component of the exciting area of adaptive computation and machine learning. Some of the data mining software includes *SPSS-Clementine*, *RapidMiner*, and *Weka*.

This article presents an overview of this three data mining software and to leave the reader with a sense of different capabilities, the ease or difficulty of use, and the user's interface of the surveyed software. The contents of this paper are organized as follows: Section 2 deals with some basic definitions of data mining and the decision tree method which are required in the later sections Because the Classification is important problem in data mining. In this research, we apply the decision tree method such as (Satyanarayana, 2013; Patel & Upadhyay, 2012, pp. 20-25). Section 3 provides a brief overview of the decision tree method by using the data mining software. In this paper, an overview of some tools is also presented. We present one of the most popular commercial tools and two open source tools. In Section 4, a brief summary of the key characteristics of the studied software is detailed with reference to using the decision tree method, while comparing the accuracy of three decision tree algorithms which

is consistent with the previously conducted studies such as (Cohen & Lim, 2000). Finally, the last section concludes the research.

2. Basic Relationship

2.1 Theoretical Presentation of Data Mining Process

Data mining is one component of the exciting area of adaptive computation and machine learning. That is a powerful new technology to extract the hidden predictive information from enormous databases, with huge possibilities to help to focus on the most essential information in data warehouses. The tools of data mining are capable of predicting the future behaviors and trends, allowing the businesses to create knowledge-driven and practical decisions, and can address the questions of business that were traditionally too time-consuming to be resolved. The automated prospective analyses presented by data mining move beyond the analyses of the past events which were frequently provided by the retrospective tools typical of decision support systems. Data mining is a procedure that makes use of a range of data analysis tools to find out the relationships and patterns in the data that may be used to create applicable predictions. The primary and the easiest analytical steps in data mining include describing the data, summarizing the statistical attributes such as standard deviations and means, as well as visually reviewing it using the related graphs and charts, and finally to look for potentially meaningful links among the variables including the values that often take place together (Han & Kamber, 2001; King & Satyanarayana, 2013). The final purpose of data mining is to generate a model which can improve the way you read and interpret your existing and future data. As emphasized in the section on the data mining process, collecting, selecting, exploring and modeling of large quantities of the correct data with the aim of discovering the relations or regularities that are seriously important. From the viewpoint of the scientific research, data mining is a comparatively new discipline that has developed mostly from studies approved in other disciplines such as computing, statistics, and marketing. The intention here is to obtain useful and clear results for the owner of the database (Laros, 2005).

2.2 Presentation of the Decision Tree Method

A decision tree is a prediction model and an algorithm that classifies the data inputs by using the branches and leaves of a tree. This tree tries to predict the outcomes based on preexisting database information comprised of exact inputs and their outputs. A tree is learned recursively in a process named recursive partitioning. Such a process encompasses dividing a basis sample into two separate samples based on an attribute value test. Then the process continues recursively until a stopping point is reached. Once a tree is learned, the Gini Impurity test can be used to a set of data to decide the suitable stopping point of the recursive partitioning function and the probability of a piece of data being mislabeled. In comparison with the neural networks, the decision trees engender the results which are simply to interpret: the rules linked with classification explain by decision points and branches (Kabra & Cichkar, 2011, p. 11).

In general, the decision trees are nonparametric and specially appropriate for the purpose of exploratory knowledge discovery. A decision tree is a graphical representation where each internal node shows a test on one of the input variables and the terminal nodes are the decision or prediction. The prediction at the terminal node is the mean of all the response values within that cell. A decision tree is a constant piecewise model and therefore can better cope with non-linearities and interactions in the data. It is one of the fundamental techniques employed in data mining whose major benefit is providing a visual representation of the decision rules at the nodes used for making predictions (Singh, Sharma, & Kaur, 2013). A decision tree is grown as a binary tree, i.e. each node in a tree has two child nodes. Basically, all the trees start with a root node and then at each node we determine the split using the explanatory variable which causes the maximum reduction in the deviance. While traversing down the tree at each given node, a condition is being tested on the basis of which we decide whether to move to the left or to the right sub-branch. If the condition is satisfied, we traverse down through the left sub-branch else down the right sub-branch. The decision is made at the leaf node (Jiang & Li, 2010; Maimon & Rokach, 2010).

Some of the split point selection methods include *Gini index*, *Entropy*, *CART* and *Classification*. A completely grown decision tree will have an obvious error rate of zero or close to zero on the training data from which the tree was built. On the other hand, its accurate error rate, measured by estimating the misclassifications might be greatly higher, when the tree is applied to a test data set. The pruning phase of the construction of classification tree determines on the tree of the correct size in order to prevent to reduce the misclassification error and overfitting. The purpose of the pruning process is to find the sub-tree that yields the smallest right error, taking into account the size complexity of the tree. That contain two methods namely past pruning and prepruning. Some of the evaluation methods are error rate, classification accuracy, the kappa coefficient, the gain, the cost, and the confusion matrix.

The decision trees are employed appling *SPSS-Clementine*, *RapidMiner* and *Weka* which are further explained in details in the Materials and Methods section (Breiman, Friedman, Stone, & Olshen, 1984; Hastie, Tibshirani, & Friedman, 2003; Witten, Frank, & Hall, 2011).

3. Introducing Various Data Mining Software

Nowadays, the growth of a huge amount of data and the complicated analyzing processes have urged manufacturing appropriate software for predicting the situation and finding the regularity in the data. The suitable data mining software should unite the datasets of different software and accordingly enabling us to use and compare them with the various methods. Over the last 10 years, a number of open source and commercial tools have been developed to examine and transform the data. Such software is listed below:

SAS Enterprise Miner, SPSS Clementine, Weka, RapidMiner, R, STATISTICA, Data Miner, Orange Canvas, DataEngine, DBMiner, WebMiner, MARS, Datamite, GainSmart, XLMiner, IntelligentMiner, Darwin, AI Trilogy, Alice, AnswerTree, BrainMaker, JDBCMiner, Braincel, DecisionTime (Satyanarayana, 2013).

Data mining applications are indeed need a computer program to be performed. The major step is to choose the right tools in order to create a good model. From there, the tools are then used to refine the model to make it more useful. As we know, there are many different evaluation methods with different algorithms in the mentioned softwares; as a result, we use the accuracy of the model as an evaluation method that shows the percentages of the records in the model being classified as correct. In this context, the decision trees are employed using *SPSS-Clementine* and the open source data mining programs namely *RapidMiner* and *Weka* (Klosgen & Zytkow, 2002, p. 249)

3.1 The Real World Datatset

In this study, the decision tree method was performed on a real childbirth dataset. However, It is important to notice that this data was collected from Arya hospital of Mashhad, Iran from May, 2010 to August, 2013. This dataset contains 3520 records with 10 attributes which include a single missing attribute value and are removed from the analysis. Our results are thus based on 3513 records and illustrate the information about the childbirth situation and baby characteristics (see Figure 1).

Data field	Description
Mother Age	(Numeric)
Baby Sex	(Girl, Boy)
Born Year	(2010-2013)
ChildBirth Method	(Caesarian, Natural, Forceps, Vacuum)
ChildBirth Situation	(Breech, Cephalic, Foot, Sholder)
Birth Term	(Post Term, Premature, Term)
Weight	(Numeric)
Height	(Numeric)
Head Around	(Numeric)
Body Temperature	(Numeric)

Figure 1. Childbirth information

4. Materials and Methods

4.1 Data Mining With SPSS-Clementine

SPSS-Clementine was the first program that was employed. It is in fact a vigorous system that is reasonably intuitive in it's design and simple to apply. In addition, outstanding documentation exists on how to apply *SPSS-Clementine* when running the decision trees, precious in finishing this study. *SPSS-Clementine* provides an explicit support for the Cross-Industry Standard Process for data mining *CRISP-DM*, the industry standard methodology that guarantees well-timed and reliable results with data mining. It is also provides a helpful set of descriptive tools improved by outstanding graphics. Users habituated to data streaming based upon icons will find this software extremely simple to work with provided that they are aware of the need to connect the icons by right-clicking the mouse. Moreover, every data mining process contains data entrance as well as performing the set of processing on the data and finally sending the data to the special destination steps. The *Clementine manager* window has three sections: *Streams*, *Outputs* and *Models*.

Furthermore, the main menu bar at the bottom of the screen consists of common operations under the *Favorites* tab; these operations are duplicated under subsequent the logical menu tabs entitled *Sources, Record Ops, Field Ops, Modeling, Graphs*, and *Output*. The *help* menu is enormously helpful in exploring the options using logical keywords such as *export, import*, and *graph*. Clementine presents a number of algorithms for *classification, clustering, association*, and *prediction*, as well as algorithms for *time-series forecasting, automated multiple modeling* and *interactive rule building*. These algorithms exist in a *SPSS-Clementine* module and the optional additional modules. Also, the modeling node has four sections: *Automated, Classification, Association* and *Segmentation*. There are two ways of observing the created project: one is the *class* node and the second is the *CRISP-DM* node (SPSS Inc., 2007).

4.1.1 The Performance of Decision Tree Method

The decision tree algorithms are in the *classification* that can be found by clicking on the *modeling* tab. Some of them are *C&R, QUEST, CHAID, C5.0* and the proposed algorithm in this paper is *C&R*. The procedure begins with loading the dataset by clicking on the *Sources* tab in the EXCEL format here. Afterwards, we need to choose *Type* from the *Field Ops* tab; then at first we read the values and afterward we continue by selecting the target value (child-birth methods) direction to *Out* and setting the predictor's to *In*. This data stream that can be found in Figure 2.

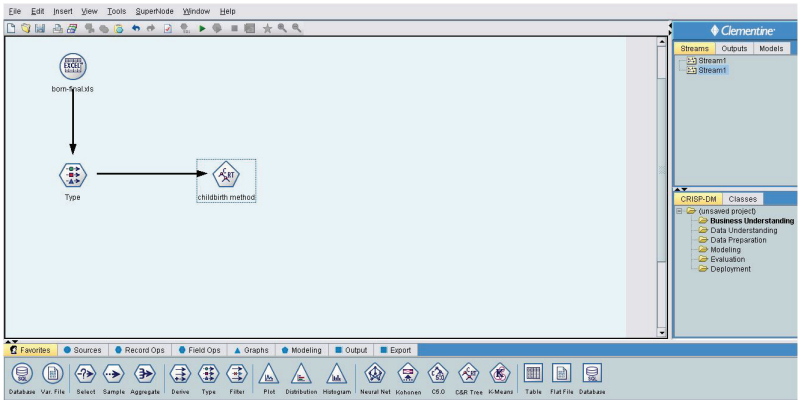

Figure 2. Clementine main page

• C&R Tree Node

The Classification and Regression Tree *C&R* node is a tree-based classification and prediction method which uses recursive partitioning to split the training records into segments with similar output field values. The *CART* implementation is very similar to *C4.5*; the one notable difference is that the *CART* constructs the tree based on a numerical splitting criterion recursively applied to the data. The *C&R* Tree node starts by examining the input fields to find the best split, measured by the reduction in an impurity index that results from the split. The split defines two subgroups, each of which is subsequently split into two more subgroups and so on until one of the stopping criteria is triggered. All the splits are binary, only two subgroups. *C&R* node has 4 sections that can be selected and changed for special purposes which are mentioned below:

The *Model* that contains the *name*, the *partitioned data*, the *launch interactive session* and *tree depth*. The *Expert* that contains the *mode*, the *maximum surrogate*, the *minimum change in impurity*, and the *impurity measure* like *gini* and also the *prune* tree. The *Cost* that contains the *misclassification cost*. The *Analyze* that includes the *model evaluation and propensity scores* (Questier, Put, & Coomans, 2005, pp. 45-54; Venkatadri, 2011).

4.1.2 The Output Interpretation

In this processed, the *training set* uses the currently selected learning algorithm, *C&R* in this case. Then it classifies all the instances in the training data and outputs performance statistics. For illustrative purposes, we evaluate the performance by choosing the *launch interactive* session to find the ideal tree graph and for more results, using once the full *training* sets by selecting *pruning or unpruning* methods and changing the *levels below the root*. At the end, were repeat all of these methods for the *(training or testing/validation)* set again. The output contains 4 sections, namely the *Viewer*: That shows the tree graph, As it can be discerned, the roots start from the childbirth methods as the parent node and the split according to some characteristics like the childbirth situations, the head

around, the weight and height of the baby, the mother's age and the baby's sex. The *Gain*: That illustrates the information about the *gain index*, the *lift index* and their *charts*. The *Risks*: That illustrates the *misclassification matrix*, the *standard error* and the *risk estimate*.

These are important factors for measuring the validity of a decision tree. The classification table is also known as a false positive table and shows the percent of the values classified correctly by the decision tree. Furthermore, we can calculate the *classification accuracy* of the model which is measured by the *mean rank of the error rate* and the *mean error rate*. In the table below, the differences between the *standard errors* are illustrated. Accordingly, when we use a *full training set*, the *standard error and risk estimate* are achieved the lowest amount. Moreover, by changing the *levels below the root* and choosing *pruning*, the *standard error* still remains unchanged and the model with *(testing/validation)* has the highest number of *accuracy* with 76.57%. Summarized results for the *standard error and the risk estimate* can be seen in Figure 3 (Fraiman, Ghattas, & Svarc, 2011, p. 25).

	Spss-Clementine CRT		
	Full Training set	Training / Validation	Testing / Validation
Risk Estimate	0.235	0.236	0.235
Standard Error	0.007	0.01	0.013

Figure 3. Compare Standard errors

4.2 Data Mining With RapidMiner

The second program we used was *RapidMiner*. That formerly known as *Yale*, is an environment for machine learning and data mining processes. It was started under the supervision of artificial intelligent Dortmund university of Germany. This software is an open source data mining tool made in 2001 with Java language but nowadays the usage version *Rapid-I* is being used. The modular operator allows the design of the complex nested operator chains for a huge number of learning problems. Hence, a large number of data processed and the main format that supports is *ARFF*. There are support vector machine learning algorithms as the learning models with a large number of *classification and regression, decision trees, bayesian, logical clusters, association rules*, many algorithms for *clustering, the separation of data preprocessing, normalization* and many features such as *filtering, genetic algorithms, neural networks, 3D and data analysis* (Hofmann & Klinkenberg, 2013).

4.2.1 The Performance of Decision Tree Method

The decision tree algorithms are in the *modeling* section that can be found at first by clicking on the *classification and regression* and next the *tree induction*. Some of these are include *CHAID*, *ID3*, *Random Forest*, *Decision Tree* and the proposed algorithm in this paper is *Decision Tree*. Also there are three necessary operators that should be used in this process which are, the *Set Role*: used to change the role of one or more attributes, the *Apply Model*: applies an already learnt or trained model on an ExampleSet and *Performance*: it delivers a list of performance criteria values. The procedure has started with Loading a dataset by clicking on the *Operators*, *Import*, *Data* and *Read Excel* respectively. By double clicking on the *Read Excel* we can see it in the main process window; then we can connect it to the *set role* operator and set target role as label. Afterward, we continue by connecting it to the *Decision Tree*, the *Apply model* and the *Performance* respectively that is depicted in Figure 4 (Pechenizkiy, 2006).

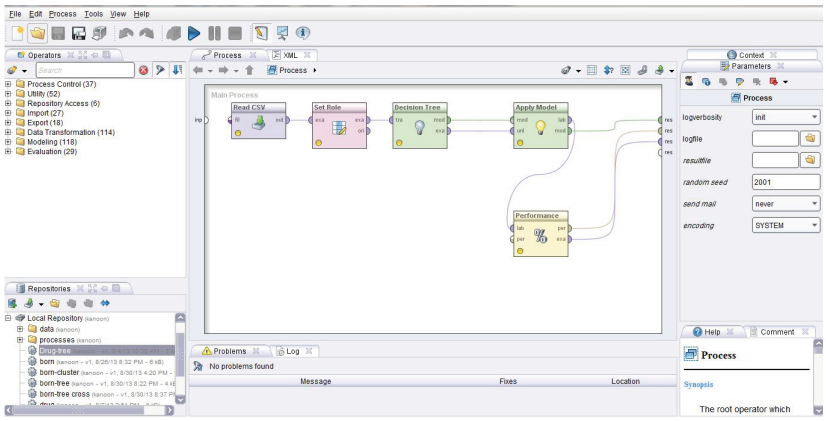

Figure 4. *RapidMiner* main page

• Decision Tree Algorithm

Briefly, generates a *decision tree* for classification of both nominal and numerical data. This *decision tree* learner acts like Quinlan's *C4.5 or CART*. A decision tree is a tree-like graph or model which is more like an inverted tree because it has it's root at the top and it grows downwards. This representation of the data has the advantage compared with other approaches of being meaningful and easy to interpret. The goal is to create a classification model that predicts the value of a target attribute often called *class or label* based on several input attributes of the *ExampleSet*. In *RapidMiner* an attribute with label role is predicted by the *Decision Tree* operator. Each interior node of the tree corresponds to one of the input attributes. The *Decision Tree* has several parts that can be selected and changed for special purpose such as, *criterion* that contains *gain ratio, gini index and accuracy rate, minimal size for split, minimal leaf size, minimal gain, maximal depth, confidence and number of prepruning*.

4.2.2 The Output Interpretation

In this process the *training* set uses the currently selected learning algorithm, *Decision Tree* in this case. Then it classifies all the instances in the training data and outputs performance statistics. For illustrative purposes, we evaluate the performance using once the full *training* sets by selecting *pruning or unpruning* methods and changing the *levels below the root* and again, repeating all of these methods for the *(training or testing/validation)* sets. The output gives information about the *accuracy rate*, the *recall*, the *precision*, the *AUC area* and the *kappa coefficient*. Moreover, by all of the full *training* set and *pruning* methods, the accuracy is equal to 92.49% and the *kappa coefficient* has the highest number of 0.773 versus, the *pruning dataset* and *cross-validation*. However, the *misclassification matrix* shows that the Caesarian childbirth method achieved the highest amount among the childbirth method. Summerized results for the *accuracy rate* and the *kappa coefficient* can be seen in Figure 5.

RapidMiner	Decision Tree			
	Pruning		Unpruning	
	Full Training set	Cross- Validation	Full Training set	Cross- Validation
Accuracy	92.49%	67.08%	76.52%	76.09%
Kappa	0.773	0.047	0.002	0

Figure 5. Compare criteria

4.3 Data Mining With Weka

The third program we used was *Weka* that is a Java software program, free and open source under the GNU General Public License. The Waikato Environment for the Knowledge Analysis *Weka* emerged following the perceived need for a unified workbench that would allow the researchers an easy access to the state of the art techniques in machine learning. This software came from the Waikato university of New Zealand. The *Weka* project aimed to provide the researchers and practitioners alike with a comprehensive collection of machine learning algorithms and data preprocessing tools. The workbench includes the algorithms for *regression, classification, clustering, association rule mining and attribute selection*. The preliminary exploration of data is well catered by data visualization facilities and many preprocessing tools. *Weka* has several graphical user interfaces that enable an easy access to the underlying functionality. The main graphical user interface is the *Explorer*. It has a panel-based interface, where different panels correspond to different data mining tasks. In the first panel, called the *Preprocess* panel, the data can be loaded and transformed by using the *Weka's* data preprocessing tools called the *filters*. The data can be loaded from various sources, including the files, URLs, and databases. The supported file formats include the *Weka's* own *ARFF* format and the others such as *CSV, LibSVMs, C4.5s*. It is also possible to generate the data by using an artificial data source and editing the data manually by using a dataset editor (Hall, Frank, & Holmes, 2007).

4.3.1 The Performance of Decision Tree Method

The decision tree algorithms are in the *classifier* tab that can be found by clicking on the *choose* tab. Some of these algorithms include *DecisionStump*, *ADTree*, *Simplecart*, and *j48* while the proposed algorithm in this paper is *j48*. The procedure started by Loading a dataset through clicking on the *Open file* button in the top left corner of the *Preprocess* panel in EXCEL format here. The attribute, the childbirth method, is the *class* attribute. All of them can be found in Figure 6 (Patel et al., 2012, pp. 20-25).

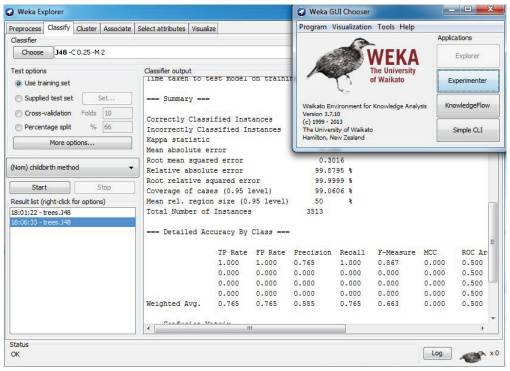

Figure 6. *Weka* main page

• **The J48 Algorithm**

J48 is an open source and a key benefit of *Weka* is it's Java implementation of the *C4.5* algorithm for creating decision trees in the weka data mining tool, especially regarding tree pruning. The *C4.5* algorithm is exceptionally beneficial in generating accurate decision trees. The second panel in the Explorer gives access to *Weka's* classification and regression algorithms. That is called *Classify* because the regression techniques are viewed as the predictors *continuous classes*. In the *classify* panel we can find various types of classifier. We can click the trees entry to reveal its subentries and click *J48* to choose the classifier. By double clicking on the algorithm's name, we can change the tree options like *pruning* and *numfolds*. Then, in the *Test options*, we can choose the *Use training set*, the *Supplied test set*, *Cross-validation*, and the *Percentage split*. Besides, in the *More options* tab, we can choose the classifier evaluation options such as the *output entropy evaluation measures* and the *output confusion matrix*.

By default, the panel runs a *cross-validation* for a selected learning algorithm on the dataset that has been prepared in the *Preprocess* panel to estimate the predictive performance. It also exhibits a textual representation of the model built from the full dataset. However, the other models of evaluation based on a separate *test set* are also supported. If applicable, the panel can provide an access to graphical representations of the models, e.g. the decision trees. Moreover, it can visualize the prediction errors in the *scatter plots* and also allows the evaluation via *ROC curves* and other *threshold curves*. The models can be also saved and loaded in this panel (Patel et al., 2012, pp. 20-25; Pechenizkiy, 2006).

4.3.2 The Output Interpretation

Once the test strategy has been set, the classifier is built and evaluated by pressing the Start button. In this process the full *training* set uses the currently selected learning algorithm, *J48* in this case. Then it classifies all the instances in the training data and outputs the performance statistics. For illustrative purposes, we evaluated the performance, using once the *training* data and again with *Cross-validation* by changing the *levels below the root* and *pruning or unpruning* methods. The results appear in the *Result List* panel located in the lower left corner.

To see the tree, we can right-click on the entry trees. *J48* was added to the result list and the *Visualize tree* was chosen. Now, it is possible to look at the rest of the information in the Classifier Output area. This text states how many and what proportion of the test instances have been correctly classified and also shows the *Kappa statistic*, the *Mean absolute error* and detailed accuracy by the class like the *TP Rate*, the *FP Rate*, the *Precision*, the *Recall*, the *F-Measure*, the *MCC*, the *ROC Area*, the *PRC Area* and the *confusion matrix* (Gholap, 2013). By right-clicking on the trees *J48* entry in the result list, we can select *Visualize margin curves, Visualize classifier errors*, and the *Cost-Benefit analysis*. Most notably, *Wekas* cost is less while the *Correctly Classified Instances, gain rate*, and finally the *accuracy rate* are higher than the ones in the *Cross-Validation* status with 85.19%. Summerized results for the criteria mentioned above can be accessed via Figure 7 (Patel et al., 2012, pp. 20-25).

Weka J48 Number of Leaves: 2024 Size of the Tree: 2360		
	Full Training set	Stratified Cross- Validation
Time taken to build model	0.01 seconds	0.41 seconds
Correctly Classified Instances	2986 84.9986%	2454 69.8548%
Incorrectly Classified Instances	527 15.0014%	1059 30.1452%
Kappa Statistic	0.5154	0.0274
Classifier Accuracy	85.19%	76.48%
Cost	520	826
Gain	565.58	0

Figure 7. Compare criteria

5. Results

In order to compare our procedure with the decision tree method in the three selected data mining software, the subsequent results were obtained:

• *SPSS-Clementine* software is expensive and commercial software which support many different data format such as Excel, SAS, etc. It has an available help menu and supports various kinds of data mining methods with huge amount of dataset without occupying the whole memory. It also provides the best place by correcting and giving online solutions during the procedure. In addition, creating decision trees is done easily and the programs are effortlessly readable and provide us with the interpretable output. It provides a graphic decision tree in addition to outputting a large amount of other important data (SPSS Inc., 2007).

• *RapidMiner* software has the most important connectors: databases, Excel, CSV, etc. In *RapidMiner*, you can use wizards for setting up your data sources and a graphical environment for processing the data flows. It also has a parallel execution plugin, executing the R scripts for the data input, transformation, and graphing. Therefore, you can easily connect the two. Also, it is more intuitive and you can find ready-to-use examples on *my experiment* and have a smaller but still huge range of data mining methods, and also can use the *Weka* library more. Furthermore, it provides a direct access to most relational databases. It is written in Java, so it uses JDBC, and has a graphical modeling interface for ease of use graphical modeling interface for ease of use. *RapidMiner* is categorized among the memory-based systems (Hofmann et al., 2013).

• *Weka* software is an open source statistical and data mining library. Similar to *RapidMiner*, *Weka* is written in Java. It has many machine learning packages, good graphics, specialized for data mining, cheap and easy to work, good for text mining and you can train different learning algorithms at the same time and compare their result. *Weka* is it's intuitive interface and extensive output. Parameters and testing options are easily modifiable (Hall et al., 2007).

We focus on comparing the ones common to all three applications, while *SPSS-Clementine*, *RapidMiner* and *Weka* provide much analytic data. The analysis centers on varying three factors: the percentage split of the testing and training data, the number of cross-validation folds, and the minimum number of instances required for the node splits. Beyond examining the common factors in three different algorithms *C&R*, *Decision Tree, J48*, we also chose one unique feature for all of the applications. For instance, we focus on the program's capability to prune trees, varying the confidence factor used in pruning, and the confidence level necessary for splitting nodes and finally the classification accuracy (Witten et al., 2011).

To evaluate the results and gain insight, we analyzed false positives table and correctly classified instances. Finally, we compared the *classification accuracy* of the model that was obtained from three different decision tree algorithms. The criteria place a decision tree algorithm called the *Decision Tree* at the top, although it is not statistically different from the other two algorithms. Another decision tree algorithm, *J48*, is the second with respect to the two accuracy criteria. The most accurate decision tree algorithm is the *Decision Tree*, which ranks 92.49%, respectively. Although the *Decision Tree* algorithm tends to have good accuracy, they also require relatively long training times. *C&R*, for example, is the third in terms of *classification accuracy* and that employing the exhaustive search usually takes much longer to train than the other algorithms. By results, the *Decision Tree* algorithm in *RapidMiner* easily obtained the higher amount of the model *classification accuracy* in this paper by 92.49% with the lowest complexity and the highest performance in terms of data handling or speed of the implementation (Ramesh, Parkavi, & Ramar, 2013).

6. Conclusions

In conclusion, the data mining programs, *SPSS-Clementine*, *RapidMiner* and *Weka* were explained and the differences due to different programming languages were emphasized. To identify which one is better depends on your background and your needs. According to a survey by KD-nuggets, the most popular data mining tool used by real projects is *RapidMiner* such as (Satyanarayana, 2013). Thus, by the results mentioned above, *RapidMiner* is the free available software that could be the best place for learning data mining and performing the decision tree method in this research. Therefore, our choice for the performing decision tree method is *RapidMiner*. On the other hand, we do not attempt to perform a controlled comparison of the algorithms in each software to decide which one is the strongest, but instead hope to give an idea of the approach to the decision tree method used in each of them. Also, we suggest the dataminer to examine different decision tree methods using the mentioned software for comparing and making their own decisions about such software.

References

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and Regression Trees*. Chapman & Hall/CRC.

Cohen, W., Lim, T. S., Loh, W. Y., & Shih, Y. S. (2000). A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-three old and new classification Algorithms. *Machine Learning, 40*, 203-229. Boston: Kluwer Academic Publishers. http://dx.doi.org/10.1023/A:1007608224229

Fraiman, R., Ghattas, B., & Svarc, M. (2011). Interpretable Clustering using Unsupervised Binary Trees. *Advances in Data Analysis and Classification, 25*.

Gholap, J. (2013). *Performance Tuning of J48 Algorithm for Prediction of Soil Fertility*. 5. Web.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., & Reutemann, P. (2007). *The Weka Data Mining Software: An Update*. Department of Computer Science , University of Waikato, New Zealand.

Han, J., & Kamber, M. (2001). *Data mining concepts and techniques*. Academic Press. Department of Computer Science, University of Illinois at Urbana-Champaign.

Hand, D., Manilla, H., & Smyth, P. (2001). *Principle of data mining* (p. 322). Cambridge: The MIT Press.

Hastie, T., Tibshirani, R., & Friedman, J. H. (2003). *Elements of Statistical Learning*. Springer.

Hofmann, M., & Klinkenberg, R. (2013). *RapidMiner: Use Cases and Business Analytics Applications*. Taylor & Francis Publisher.

Jiang, L., & Li, C. (2010). An Empirical Study on Attribute Selection Measures in Decision Tree Learning. *Journal of Computational Information Systems, 6*(1), 105-112.

Kabra, R., & Cichkar, S. (2011). Performance Prediction of Engineering Students using Decision Tree. *International Journal of Computer Applications, 36*, 11.

King, B., & Satyanarayana, A. (2013). *Teaching Data Mining in the Era of Big Data*. ASEE Annual Conference.

Klosgen, W., & Zytkow, J. (2002). *Hand book of data mining and knowledge discovery* (p. 249). Oxford university press.

Laros, D. (2005). *Discovery Knowledge in Data, An Intruduction to Data Mining*. John Wiley & sons, Canada.

Maimon, O., & Rokach, L. (2010). *Data Mining and Knowledge Discovery Handbook* (2nd ed.). Springer.

Patel, N., & Upadhyay, S. (2012). Study of Various Decision Tree Pruning Methods with their Empirical Comparison in WEKA. *International Journal of Computer Applications, 60*(12), 20-25. Published by Foundation of Computer Science, New York, USA.

Pechenizkiy, M. (2006). *Prototyping DM Techniques with Weka and Yale Open-Source Softwares*. Department of Mathematical Information Technology, University of Jyvskyl.

Questier, F., Put, R., Coomans, D., Walczak, B., & Vander Heyden, Y. (2005). The use of CART and multivariate regression trees for supervised and unsupervised feature selection. *Chemometrics and Intelligent Laboratory Systems, 76*, 45-54.

Ramesh, V., Parkavi, P., & Ramar, K. (2013). Predicting Student Performance: A Statistical and Data Mining

Approach. *International Journal of Computer Applications, 63*.

Satyanarayana, A. (2013). *Software tools for teaching undergraduate data mining course*. New York City College of Technology.

Singh, M., Sharma, S., & Kaur, A. (2013). Performance Analysis of Decision Trees. *International Journal ofComputer Applications, 71*(19), 10-14. Published by Foundation of Computer Science, New York, USA.

SPSS Inc. (2007). *Clementine 11.1 User's Guide*. SPSS Incorporated.

Sumthi, S., & Sivandam, N. (2006). *Introduction to Data Mining Its Application*. New York: Springer. http://dx.doi.org/10.1007/978-3-540-34351-6

Venkatadri, & Lokanatha. (2011). A Comparative Study of Decision Tree Classification Algorithms in Data Mining. *International Journal of Computer Applications in Engineering, Technology and Sciences, 3*(3), 230-240.

Witten, H., Frank, E., & Hall, M. (2011). *Data mining practical machine learning Tools and Techniques* (3rd ed.). Morgan Kaufmann publishers.

Copyrights

Copyright for this article is retained by the author(s), with first publication rights granted to the journal.

This is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (http://creativecommons.org/licenses/by/3.0/).