RESEARCH PAPERS OF WROCŁAW UNIVERSITY OF ECONOMICS
No. 85 2009
Advanced Information Technologies for Management – AITM 2009

Dorota Dżega[1], Wiesław Pietruszkiewicz[2]

[1] West Pomeranian Business School, Szczecin, Poland,
[2] West Pomeranian University of Technology, Szczecin, Poland
e-mails: ddzega@zpsb.szczecin.pl; wpietruszkiewicz@wi.zut.edu.pl

AN EMPIRICAL STUDY OF ARTIFICIAL INTELLIGENCE SUPPORT IN SOFTWARE PROJECT MANAGEMENT

Abstract: In this article we present an application of Artificial Intelligence in software project management by estimation of software metrics. Due to increasing significance of Open Source, we have selected project being hosted on the leading hosting platform of Open Source projects – Sourceforge.net. In the first part of the article, we describe steps of data extraction which was a large scale task because of the data source size and complexity. Moreover, we used data that were originally gathered to be used by project collaboration web-based system not to predict any features. Therefore extraction of meaningful data required an analysis of databases structure and transformation of sets of records into the meaningful datasets. These datasets were constructed to predict four factors important to project management, i.e. skills, time, costs and effectiveness. In the later part of the article, we present the results of prediction experiments, that were performed using C4.5, RandomTree, CART, Neural Networks and Bayesian Belief Networks algorithms. Finally, we analyse influence of several methods' parameters on estimation accuracy and size of developed knowledge base.

1. Introduction

Nowadays many software applications are being developed as Free Libre/Open Source Software (OS later in this article). The results of these projects, done by team members cooperating via web-systems, sometimes over-perform proprietary software. Therefore OS projects become strong competitors to the classic closed-source software products. As a result, methods of software management must be extended by solutions specially tailored to OS characteristic.

The basic assumption of project management is that the knowledge and experience, acquired from the past projects, help to effectively manage risk during software development. It is coherent with basic purpose of Artificial Intelligence which focuses on knowledge extraction from the past observations and its conversion to forms easily understandable and usable in the future.

Herein, we present a large scale AI application, the aim of which was to create a set of models supporting decision making by providing prognosis of the most important features of software project. All data used in presented experiments were extracted from SourceForge.net being the leading OS hosting platform. The number of existing tables and the number of stored records were a serious test of capabilities of prediction methods which applied to this complicated real life problem must have proven their speed, precision and low memory requirements. Moreover, the methods of machine learning used herein were carefully selected after analysing their ability to work with datasets containing large number of unordered, non-numeric attributes.

2. Data source and AI in project management

The source of data about OS projects was "A Repository of Free/Libre/Open Source Software Research Data" being a copy of internal databases gathered by SourceForge.net platform [Madey 2008]. This platform hosts many popular software projects, for example phpMyAdmin, OpenBravo ERP, Notepad ++, WinSCP or even very well known Artificial Intelligence tools like RapidMiner (formerly YALE) and Weka.

The potential source of data contained a high number of features and its form of storage was not designed to be later used in knowledge extraction process. It was built to store all data necessary to run a web-based project collaboration service, e.g. source code exchange or forum. This caused that presented problem was a modelling example of a real-life AI application. It is important, because sometimes methods which work fine during purely scientific experiments are useless in a real-life, large-scale problems. Technically, the data source contained almost 100 tables and a monthly increase of data was estimated at 25 GB [Madey 2008].

Prediction of Open Source Software was previously a subject of other researches, e.g. Raja and Tretter [2006] examined logistic regression, decision trees and neural networks in analysis of success factors for Open Source Software. Another approach was presented in [Weiss 2005] that proposed measurement of OS Projects success using web search engines. The other research [Gao et al. 2004] examined similarity measures for OSS projects and performed clustering, while the research of English and Schweik [2007] described how abandonment of an OS project could be predicted. A study on connection between Open Source updates and number of contributors was presented in [Eilhard, Meniere 2009].

Unlike the previous researches, presenting selected aspects of AI applied to OSS, we present herein a complete analysis of four project dimensions. The described research was based on an assumption that it is more important to offer a prediction of a few factors, which will aid a successful project management, than to predict if project will become successful or not.

The research presented herein was planned to be a potential AI-based support for project management (Figure 1). Currently existing Open Source platforms use web systems to make users' cooperation effective, but these management tools do not provide any decision supporting mechanism for decision makers. By using AI methods it is possible to build a knowledge base supporting project leaders by estimating the most important factors of software projects.

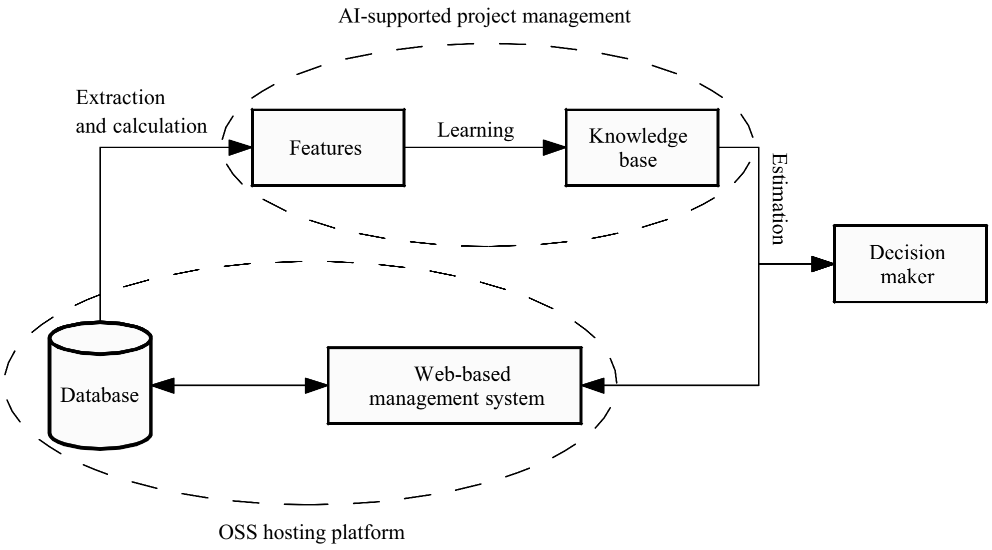

AI-supported project management
Extraction and calculation
Features
Learning
Knowledge base
Estimation
Decision maker
Database
Web-based management system
OSS hosting platform

Figure 1. AI-based support of OSS project management

Source: own elaboration.

The extracted dataset contained information about the most important aspects of OS projects, creating these features we selected groups of attributes that could be useful in measuring software, a general guideline of software metrics may be found e.g. in [Munson 2003]. These four datasets were:

– project scope Z_t – the number of months of duration of particular project category, from the moment of project initialization (project registration) decision till the last published presentation of the project effects; containing 39 attributes, including 8 attributes pertaining to the project field (d_p), 28 attributes pertaining to the project resources (z_p) and 3 attributes pertaining to project communication (k_p),
– project time C_t – the time of task completion expressed in working hours spent on completing a particular task; containing 12 attributes, including 7 attributes pertaining to general conditions of task completion (w_t), and 5 attributes pertaining to the resources of persons completing the task (z_t),
– project cost K_t – the average number of working hours spent by a particular project contractor on task completion; containing 18 attributes, including 8 attributes pertaining to the participant competence (z_u) and 10 attributes pertaining to the participant activity (a_u),

– project effects E_t – the number of completed tasks as of the date of diagnosis; containing 21 attributes, including 16 attributes pertaining to activity of project execution (r_a) and 5 attributes pertaining to communication activity related to project execution (k_a).

The numeric characteristic of created dataset was presented in Table 1. The column "Reduced records" denotes how many records passed through filters, e.g. we have excluded empty projects or projects with an empty development team.

Table 1. Details for scope, time, costs and effects datasets

Dataset	Unique records	Reduced records	Objects	Attributes
Scope	167 698	167 698	2 881	39
Time	233 139	104 912	77 592	12
Costs	127 208	20 353	10 889	18
Effects	96 830	15 492	64 960	21

Source: own calculations.

To select the most important attributes we have used Information Gain Ratio which in its core calculates change of entropy from state X to $X|A$ (information gain caused by feature A). Assuming that $H(..)$ is an entropy function, the information gain may be calculated as $IG(X, A) = H(X) – H(X|A)$. Selecting a subset of attributes helped to limit space decisions and increase classification accuracy by rejecting unnecessary features being an informational noise. For each dataset we have examined various models, where number of input was changing from 1 to D_i, where D_i was a number of attributes for i dataset and accuracy was calculated for training set, 5- and 10-fold Cross-Validation. Figure 2 presents the results of that part of experiments done for all four datasets. It must be mentioned that this evaluation was made by Naive Bayes [Mitchell 1997]. As this algorithm was not used in the later stages of experiments, it was possible to conduct a fair selection of features and omit a chance of selecting features in favour of any algorithm.

Analysing the results of this stage, we prepared an explanation of individual information attributes for project scope Z_t: the average number of working hours spent on completion of the project tasks z_1; number of selected project tasks d_8; subject scope of the software d_1; number of project contractors z_4; number of selected subprojects d_7; number of project contractors at the position of a developer z_8; number of unique time zones z_2; software language d_5; user interface d_3.

For project time dimension selected attributes were C_t: task completion time, expressed as a number of days from the task initialization till its completion w_1; number of tasks within a subproject w_7; number of preceding tasks which a particular task depends on w_5; percentage of task completion w_2; skills of a project contractor completing the task z_5; task status w_4; role/position of a project contractor complet-

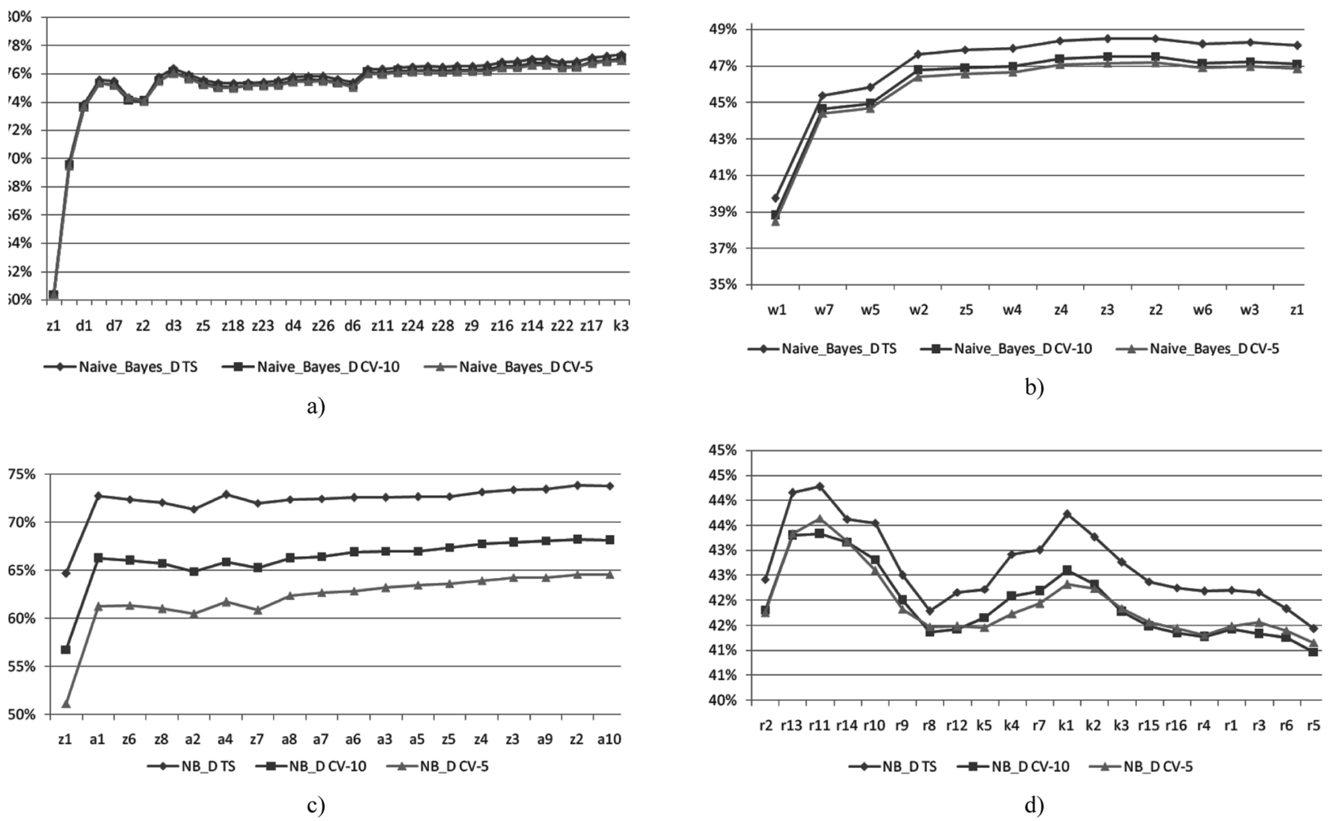


Figure 2. Estimation by Naive Bayes models with different number of attributes for Scope (a), Time (b), Costs (c) and Effects (d)

Source: own calculations.

ing the task z_4; role/position of a project contractor assigning the task z_3; number of contractors assigned to complete the task z_2.

The project cost could be predicted by K_t: registered time of the project contractor (mths) z_1; number of assigned tasks a_1; number of unique skills of project contractor z_6; the most frequently held role/assigned position z_8; number of projects executed by a particular contractor a_2; number of posts sent by a particular contractors to different project-related Web forums a_4.

The last dimension project effects contained attributes E_t: the average task completion status in percentage r_2; number of tests in CVS version control system r_{13}; number of open artefacts (additional artefacts in the project) r_{11}; number of comments in CVS version control system r_{14}; number of closed error reports r_{10}; number of open error reports r_9; number of requests for help r_8; number of closed artefacts (additional artefacts in the project) r_{12}; number of project mailing lists k_5; number of threads on forums assigned to particular project k_4; number of closed binary code modifications r_7; number of project documentation groups k_1.

These attributes were used in the next stage of experiments. It must be noticed that some of them, to be created, were calculated from other database fields or were results of frequency analysis. We also found out that number of features may be reduced even more, as there existed smaller subsets for which prediction accuracy was not significantly worse than maximum. The selected smaller subsets of information attributes, required for the prediction of project features, were: $Z_t = \{z_1, d_8, d_1, z_4\}$, $C_t = \{w_1, w_7, w_5, w_2\}$, $K_t = \{z_1, a_1\}$ and $E_t = \{r_2, r_{13}, r_{11}\}$.

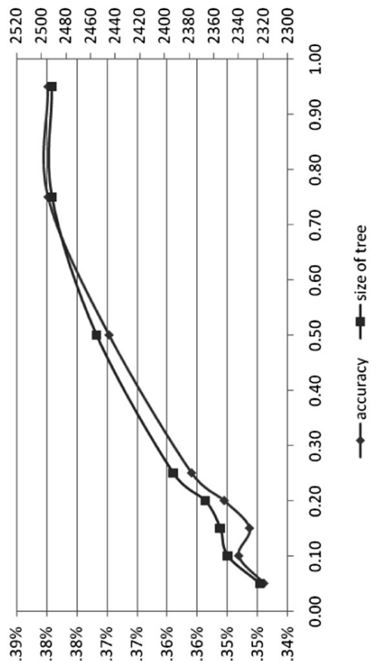
3. Experiments with classifiers

The practical applications of software estimation often use description of software risk or complexity in the form of label, e.g., high, mid or low. Thus, we decided to use classification as a method of prediction instead of regression that often gives large errors for software. To ensure unbiased environment, each dataset was filtered to form uniform distribution of its classes. It must be kept in mind that for 5 uniform classes, the accuracy of blind choice equals 20%.

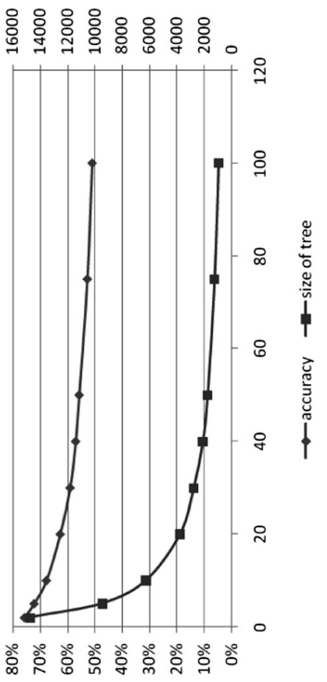
Table 3. Comparison of prediction models for 4 datasets: Scope, Time, Costs and Effects

Classifier	Scope	Time	Cost	Effects
C4.5	97.36%	67.65%	78.27%	55.31%
RT	99.28%	71.87%	91.98%	76.83%
CART	96.47%	64.92%	74.56%	70.33%
NN	44.44%	34.57%	38.61%	38.51%
BBN	93.61%	60.24%	69.80%	46.19%

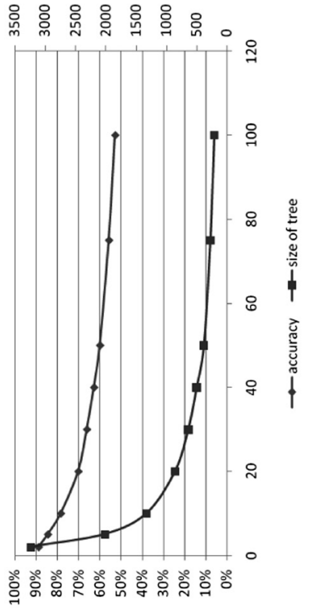
Source: own calculations.



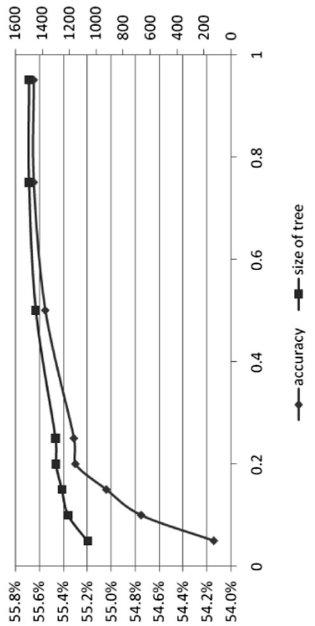
a) confidence factor for Scope dataset



b) minimum number of instances per leaf for Time dataset



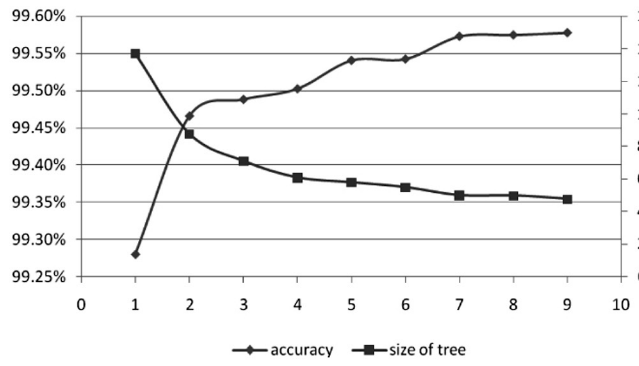
c) minimum number of instances per leaf for Costs dataset



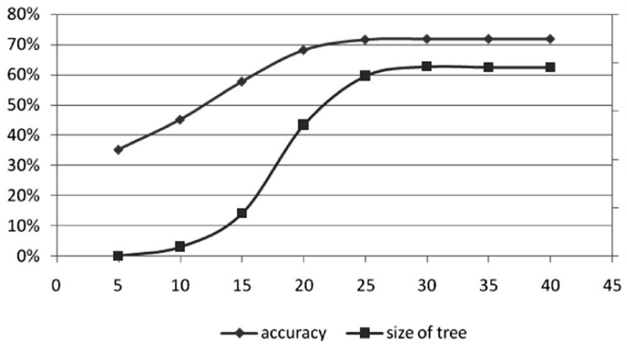
d) confidence factor for Effects dataset

Figure 3. Influence of confidence factor and number of supporting instances on C4.5 estimation accuracy and size of tree for Scope (a), Time (b), Costs (c) and Effects (d) datasets

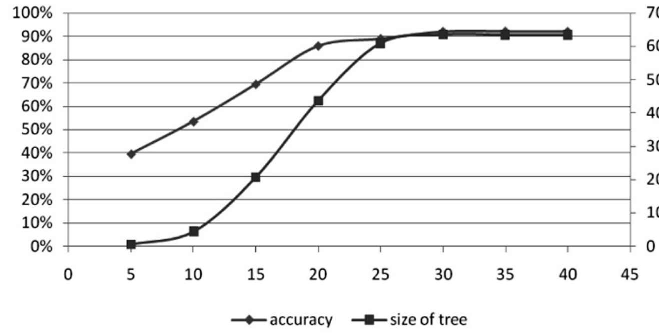
Source: own elaboration.



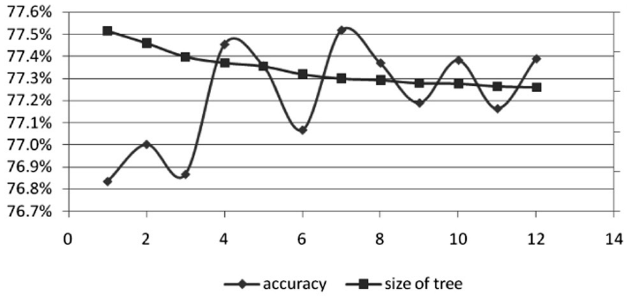
a) number of random attributes for Scope



b) maximum depth of the tree Time



c) maximum depth of the tree for Costs



d) number of random attributes for Effects

Figure 4. Influence of number of random attributes and maximum depth on RandomTree estimation accuracy for Scope (a), Time (b), Costs (c) and Effects (d) datasets

Source: own elaboration.

Table 3 contains comparison of accuracy ratios for C4.5, RandomTree (RT), Classification and Regression Tree (CART), Neural Networks (NN) and Bayesian Belief Networks (BBN) classifiers – the detailed information about these methods may be found in [Quinlan 1993; Breiman et al. 1984; Breiman 2001; McColm 2003; Mitchell 1997]. Values presented in Table 3 are the best accuracies achieved by each classifier during learning process (to ensure fair selection Naive Bayes was used only during reduction of attributes' space). To understand why Neural Networks achieved the lowest accuracy for all datasets, it must be remembered that all datasets contained mostly unordered nominal attributes, while Neural Networks require numeric ones. The process of conversion of nominal attributes into numeric attributes caused large errors for NN.

As it can be noticed, the most accurate method for each dataset was RandomTree, due to effective internal validation during tree construction. This method also did not require careful and sophisticated adjustments of parameters like other methods. It must be mentioned that the most popular decision tree classifier, i.e. C4.5, had the largest number of adjusting parameters. That increases searching space in which researchers must be looking for an optimal configuration. Moreover, comparing the speed of learning showed that RandomTree was the fastest algorithm, which is important factor in practical AI applications.

Size of tree is important factor in practical application of AI, creation of rules ought to comply Ockham's razor [Russell, Norvig 1995]. On Figure 3 we present influence of confidence factor and minimum number of supporting instances on estimation accuracy (left axis) and size of tree (right axis) for each of four datasets. The charts on Figure 4 present influence of number of random attributes and maximum depth of tree on RandomTree estimation accuracy and size of tree.

When analysing Figures 3 and 4 we observed that there were scopes of parameters' changes that significantly reduced size of decision tree, simplifying an application of inducted knowledge, with a slight and acceptable decrease of accuracy.

4. Conclusions

The research presented herein proves that Open Source projects management may be supported by AI methods. As all four presented datasets were extracted or calculated from database fields, designed to store data required by web-based project management platform, it is possible to predict important factors for project without necessity to use other-external data sources.

The examined classifiers showed large differences in performance, e.g. Neural Networks and Support Vector Machines (rejected at early stage) had low accuracy for each dataset. It was caused by large number of unordered nominal attributes. This resulted in decrease of NN accuracy. However, several methods from decision trees family showed high accuracy for examined data and we claim that it designates them to problems with a similar profile of datasets. In the future research, we plan to analyse the dynamics of modelled process, that could lead to better understanding and more effective decision support.

References

Breiman L. (2001), Random forests, *Machine Learning*, Vol. 45, No. 1, pp. 5-32.

Breiman L., Friedman J.H., Olshen R.A., Stone C.J. (1984), *Classification and Regression Trees*, Wadsworth International Group, Belmont.

Eilhard J., Meniere Y. (2009), *A look inside the forge: Developer productivity and spillovers in open source project*, SSRN Working Paper Series, http://ssrn.com/abstract=1316772.

English R., Schweik C.M. (2007), Identifying success and abandonment of floss commons: A classification of sourceforge.net projects, *Upgrade: The European Journal for the Informatics Professional*, Vol. 8, No. 6, pp. 54-59.

Gao Y., Huang Y., Madey G. (2004), Data mining project history in open source software communities, [in:] *North American Association for Computational Social and Organization Sciences 2004*, Pittsburgh.

Madey G. (2008), *The Sourceforge research data archive (srda)*, http://zerlot.cse.nd.edu.

McColm G.L. (2003), An introduction to random trees, *Research on Language & Computation*, Vol. 1, No. 3-4, pp. 203-227.

Mitchell T.M. (1997), *Machine Learning*, McGraw Hill, New York.

Munson J.C. (2003), *Software Engineering Measurement*, CRC Press, Boca Raton.

Quinlan R. (1993), *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, San Mateo.

Raja U., Tretter M.J. (2006), Investigating Open Source Project success – A data mining approach to model formulation, testing and validation, [in:] *Proceedings of the Thirty first Annual SAS Users Group International Conference 2006*, San Francisco, pp. 71-78.

Russell S.J., Norvig P. (1995), *Artificial Intelligence: A Modern Approach*, Prentice Hall, Englewood Cliffs.

Weiss D. (2005), Measuring success of open source projects using web search engines, [in:] *The First International Conference on Open Source Systems* (OSS 2005), Genova, pp. 93-99.