Przemysław Pospieszny[1]

Application of Data Mining Techniques in Project Management – an Overview

Abstract

In recent years data mining has been experiencing growing popularity. It has been applied for various purposes and become commonly used in day-to-day operations for knowledge discovery, especially in areas where uncertainty is substantial. Data mining is replacing traditional error prone and often ineffective techniques or is used in conjunction. Due to a large number of projects either struggling or even failing the researchers recognize its potential application in the project management discipline in order to increase project success rates. It can be used for different estimation problems like effort, duration, quality or maintenance cost. This paper presents a critical review of potential applications of data mining techniques contributing to the project management field.

Keywords: data mining, knowledge discovery in databases, project management, software effort estimation, project monitoring, software quality, maintenance cost, data mining applications

1. Introduction

In the modern fast-paced world, where information plays a significant role, it is crucial for every organization to manage their knowledge in order to gain a competitive advantage. A rapid increase in the amount of data stored, which goes in petabytes or exabytes, business requirements for high quality, low costs and short time-to-market products have generated an urgent need for new techniques and tools that can intelligently and automatically transform the processed vast amount of data into useful information and knowledge.

From the requirements mentioned above a data mining concept has emerged, which is a step in the knowledge discovery process and is responsible for exploration and analysis of large quantities of data, using interdisciplinary techniques

[1] Warsaw School of Economics, Collegium of Economic Analysis.

from mostly statistics and artificial intelligence areas, in order to discover meaningful patterns and rules[2].

In recent years the application of data mining by the public and private sector has become extremely popular, especially in industries like banking, telecommunications, insurance, retailing and medicine for such purposes as fraud detection, credit scoring, customer retention, product placement and drug testing. It is used for cost reduction, sales increase and research enhancement that translates into competitive advantage.

Although project management is a crucial discipline in terms of delivering and enhancing new products or services, data mining is not being widely applied by information technology practitioners to support the project implementation process. Risk, uncertainty and estimation are key terms in a project environment on which success of a project is dependent. According to the Standish Group[3] survey results only 37%[4] of IT projects are implemented within the initially assumed budget, resources, duration and scope. A majority of projects are struggling (42%) with preconceived constraints or even failing (21%). This is mostly caused by uncertainty at the initial stages of a project lifecycle (initiation, planning), where project managers are expected to estimate the budget, schedule and scope with a limited set of tools using mostly estimation by analogy or bottom-up/ top-down techniques.

More advanced and mature project-oriented organizations use parametric estimation techniques based on lines of code or function points. These techniques, however, exhibit many limitations, such as: they tend to be complex to apply; may overlook important factors due to limited input variables; they may lead to inaccurate estimation because of poor sizing inputs and not taking into account enterprise environmental factors; and finally they do not fully experience from the completed projects and lessons learnt.

A majority of organizations store project performance data generated at every project stage. This data consists of information about resources, financials, quality and other project metrics which can be explored using data mining models in order to support ongoing or further projects in activities like initial

---

[2] M. Berry, G. Linoff, *Data Mining Techniques for Marketing Sales and Customer Support*, Wiley, USA 2004, p. 7.

[3] The Standish Group International, Inc., based in Boston, USA, is a market research and advisory firm specializing in mission-critical software and electronic commerce – http://blog.standishgroup.com/

[4] The Standish Group International, *Chaos Summary for 2010*, Boston 2010, p. 3.

estimation and ongoing monitoring (budget, duration), risk identification and evaluation or software quality.

This paper aims to present different potential applications of data mining for software project management which may contribute to decreasing risks and uncertainty in the project decision making process and increasing project success rates.

2. Knowledge and Project Management

Knowledge management is a process of transferring tacit into explicit knowledge and is one of key resources and success factors for effective project management. A proper application of knowledge and additional skills, tools and techniques contributes to meeting project requirements and goals. In principle, project knowledge management can be divided into two dimensions[5]:

- Micro-knowledge- knowledge needed to perform one task (or its part) or to solve a problem (or its part),
- Macro-knowledge- total knowledge possessed by a given subject,
  - Individual macro-knowledge (knowledge possessed by a team member),
  - Project team macro-knowledge (knowledge possessed by a project team),
  - Organization macro-knowledge (knowledge possessed by an organization),
  - Global macro-knowledge (knowledge possessed by the whole global community of project managers).

Knowledge is transformed into explicit one at every project stage and captured in project documentation[6]:

- Initiation – define a new project, obtain authorization to start the project through project charter, initial estimations – budget, time, resources, scope,
- Planning – establish the scope of the project, refine the objectives, and define the course of action required to attain the objectives that the project was undertaken to achieve through project management, scope, schedule, budget, quality, resources, risks plan definition,
- Executing – complete the work defined in the project management plan to satisfy the project specifications,

[5] S. Gasik, *A Model of Project Knowledge Management*, Wiley, USA 2010, pp. 3–4.

[6] Project Management Institute, *A Guide to the Project Management Body of Knowledge (PMBOK®guide)*, Project Management Institute, USA 2013, 5th ed., pp. 27–28, 61, 418.

- Monitoring and controlling – track, review, and regulate the progress and performance of the project, identify any areas in which changes to the plan are required and initiate the corresponding changes through verification of the scope, schedule, budget, quality and risks; time reporting, required expenditure and disbursement reviews,
- Closing – project review and knowledge base update through lessons learned, final project audits, project evaluations, product validations, and acceptance criteria.

According to the PMI[7] all five stages interfere with ten knowledge areas[8] which are a set of concepts, terms, and activities that enable project team members to define outcomes at each project stage.

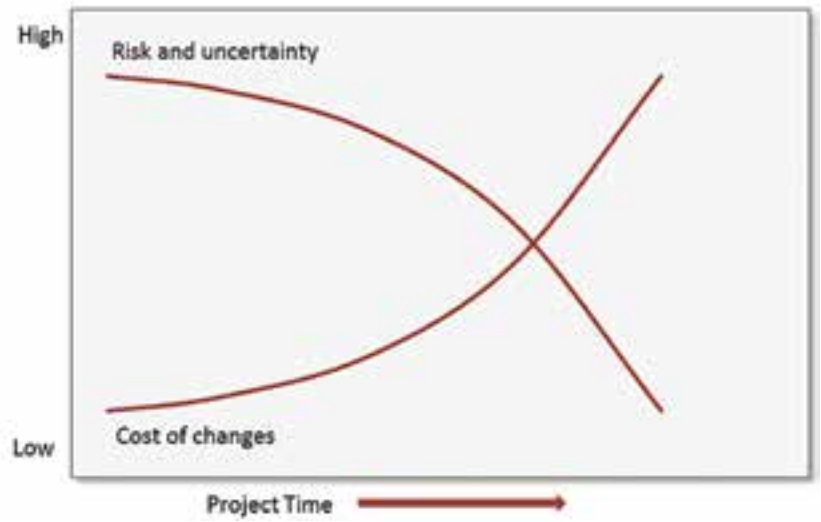

Figure 1. Uncertainty and Cost of Changes over Project Time

Source: the author's own elaboration based on Project Management Institute, *A Guide to the Project Management Body of Knowledge* (PMBOK®guide), Project Management Institute, USA 2013, 5th ed.

Uncertainty is a key aspect of project management which influences project factors like the risk and quality of a product or service delivered. Through

[7] The Project Management Institute (PMI) is a not-for-profit professional organization for the project management profession with the purpose of advancing project management.

[8] Project Integration Management, Project Scope Management, Project Time Management, Project Quality Management, Project Human Resource Management, Project Communications Management, Project Risk Management, Project Procurement Management and Project Stakeholder Management.

the project lifecycle it is the highest at the initiation of a project, where limited information assumptions about the project budget, duration and resources need to be defined. These factors become part of a project business case, i.e. they are communicated to a project sponsor who approves them and grants an estimated budget to conduct project tasks. An inaccurate estimation of these factors may result in a cost, schedule or allocation overrun in the next project phases and may impact the project business case, which in turn may contribute even to the project cancellation because the cost exceeds the benefits. Therefore, often the defined budget, duration and resources are called project constraints and cause proper estimation crucial in project initiation, which influences the whole endeavor. Unfortunately, a lack of modern estimation tools and techniques in place for project managers results in a low rate of projects being accomplished within the initially assumed budget, timeframe and resources.

In the planning phase, project objectives are broken down into detailed activities, which results in a reduction of uncertainty and enables a revision of project estimates. During this time, project constraints may be altered and approved by the project sponsor in order to consume the available knowledge, decrease the probability of risk occurrence and to improve the quality of the product delivered. At the execution stage, project tasks are performed with the use of monitoring and controlling techniques. If project constrains/estimates are incorrectly defined, any modification applied to them have a high impact, especially on the project budget, because the cost of adaptation to the encountered challenges is substantial at that stage.

In the last phase (project closure) the product or service is reviewed to determine whether it meets the assumed requirements. Moreover, the project documentation is updated, lessons learnt are gathered and project metrics, stored within the organizational project knowledge base, are revised.

The organizational knowledge base, usually maintained by the Project Management Office, contains information related to ongoing and completed projects and consists of the following, but not limited to[9]:

- Financials – labor hours, incurred costs, budgets, and any project cost overruns,
- Historical information and lessons learnt e.g., project records and documents, all project closure information and documentation, information regarding both the results of previous project selection decisions and previous project performance information, and information from risk management activities,

[9] Project Management Institute, op.cit., p. 28.

- Issues and defects – issue and defect status, control information, issue and defect resolution, and action item results,
- Process measurement – used to collect and make available measurement data on processes and products, and
- Project files from previous projects – e.g., scope, cost, schedule, and performance measurement baselines, project calendars, project schedule network diagrams, risk registers, planned response actions, and defined risk impact.

The information stored in the knowledge base is mainly retrieved and used among project practitioners for continuous improvement and sharing of best practices. It helps prevent the repetition of mistakes and wheel reinvention. Additionally, the knowledge base is widely used for estimation by analogy of a new project's cost, duration and resources.

The vast amount of data about projects stored in knowledge repositories presents a great opportunity and potential for knowledge discovery. Based on historic information, data mining models can be deployed to support project practitioners in activities such as estimation, project monitoring, quality and risk management. Thus, it may contribute to establishing modern tools that derive in statistics and artificial intelligence, to optimize the allocation of necessary input factors in order to meet the established objectives.

3. Data Exploration and Knowledge Discovery

Extraction of useful knowledge requires a large collection of data that can take a form of numerical values stored in simple tables or more complex spatial databases, text files or even figures. Simple retrieval of data provides only basic information that is usually preceded by manual and time-consuming activities. In order to take a full advantage of stored data, discover useful patterns and interpret them, it is essential to apply automated tools and techniques dedicated to support users in this process.

From the requirements above and rapid growth of data stored by organizations the data mining concept emerged in the 1990 s[10] with the roots in data warehousing and decision support systems. It is a vital step in the knowledge

[10] The term knowledge discovery in databases was originally introduced at the first KDD workshop in 1989 to emphasize that knowledge is the end product of a data-driven discovery.

discovery in databases process (KDD)[11] and essentially is a set of multidisciplinary techniques and algorithms (i.e. statistics, artificial intelligence, machine learning) focused on extracting meaningful knowledge.

In recent years data mining has become very popular among organizations from different disciplines and areas due to its powerful capability of discovering knowledge that is crucial in today's information-based world. An additional and proper knowledge may turn into decisions that can lead to the most important terms for each organization: an increase in revenue and cost cuts, eventually gaining the competitive advantage. Therefore, it is widely applied in many branches of economy, such as:

- Banking– credit scoring, fraud detection, Anti Money Laundering (AML),
- Telecommunications– customer retention (churn), data traffic and resources usage analysis, fraud detection,
- Insurance– customer retention, fraud detection, cross-selling/up-selling, product targeting, direct marketing,
- Retail– sales forecasts, product display, discovering new purchasing trends, customer behavior buying patterns identification,
- Finance– stock and currency exchange forecasts,
- Education– student performance forecasts, student modeling, social network analysis, lectures planning and scheduling,
- Healthcare & medicine– clinical decision support systems, volume of patient visits forecasts, drug testing,
- Transportation– distribution of stock among warehouses, loading patterns and route optimization.

The application of data mining for industry needs is accomplished through different data mining tasks, depending on desirable results[12]:

- Predictive data mining- the most common type, it refers to the prediction of unknown data values based on patterns discovered in historic data,
  - Classification- discovery of a predictive learning function that classifies a new data item into one of several predefined classes; deals with discrete outcomes,
  - Regression- similar to classification, except that the outcome variable is numerical rather than categorical,

[11] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, *From Data Mining to Knowledge Discovery in Databases*, American Association for Artificial Intelligence, USA 1996.

[12] D. Larose, *Discovering Knowledge in Data*, Wiley, USA 2005, pp. 11–18; M. Kantardzic, *Data Mining: Concepts, Models, Methods, and Algorithms*, Wiley, USA 2011.

  - Time series analysis- focuses on predicting future values of time series at different points in time.
- Descriptive data mining- identification of patterns and relationships within the examined data;
  - Clustering- a task of segmenting a data set into a number of subgroups based on similarities,
  - Association rule- seeks to uncover rules for quantifying the relationship between two or more attributes, known as affinity analysis or market basket analysis,
  - Anomaly detection- identification of rare cases which do not conform to an expected pattern,
  - Profiling- description of the examined data without a pre-specified purpose, focused on understanding the data,
  - Retrieval – derivation of useful patterns, similarities from text or images, known as text and image mining,
  - Process mining- relatively new, focused on the analysis of business processes based on event logs.

Both predictive and descriptive tasks derive from artificial intelligence, machine learning, pattern recognition, case base reasoning and statistical techniques in order to build models. Predictive data mining utilizes supervised learning techniques which predict values based on training labeled data such as decision trees, neural network, support vector machines, regression (linear and logistic) or generalized linear models. Descriptive ones aim to find a hidden structure in unlabeled data, therefore use unsupervised techniques like the k-means, k-nearest neighbor or hierarchical clustering.

Generally, it can be stated that project management is a "green field" in terms of application of data mining and there can be found hardly any real life use of those techniques. Nonetheless, there is a potential use of especially supervised data mining techniques, due to challenges with the correct estimation of project cost, time and resources, which seems to be the biggest difficulty. Information stored in the project knowledge base about ongoing and completed projects can be mined in order to support initial project estimations by applying regression techniques to predict exact values of the budget, duration and effort needed. This can be also achieved by using classification techniques that can predict the range of values a particular project attributes and thus they can provide project managers with range estimates that in some cases are more preferable due to accuracy concerns. Additionally, regression and classification techniques can be applied for project monitoring to support earned value analysis and provide

better estimates. Unsupervised techniques can be adopted for the identification of potential risks, issues and quality concerns (so-called anomaly detection).

The next sections present an overview of the mentioned above and other potential applications of data mining for project management recognized by researchers and project practitioners.

4. Data Mining Applications in Project Management

4.1. Initial Estimation

An accurate estimation of resources required to perform a project is recognized as a crucial aspect of project management and on which project success is dependable. It is mostly performed at the beginning of a project, at the initiation phase, in order to put together project boundaries: the business case and project charter. Later, in the project planning phase where activities to complete the project are established and uncertainty decreases, resources are re-estimated to extrapolate a more accurate budget, as well as timeframe and effort[13] needed for the execution, monitoring and closing phase.

In order to support the above-mentioned process, traditional estimation techniques are widely applied by project practitioners such as expert judgment, estimation by analogy or top-down/ button-up, WBS based[14] and Delphi. They all are based on human expert knowledge, past experience and personal traits of a project manager. Different people may estimate differently the same resource problem due to a riskier propensity or other bias factors. Although traditional techniques are error prone and subjective, they are used mostly as the only estimation method because of usage simplicity. They become very powerful tools in hands of experienced project managers and for estimation of small and medium-sized projects.

Many large and mature organisations, from the project management perspective, have incorporated size estimation models based on function points (IFPUG, NESMA, COSMIC) or parametric ones dependent on lines of software code (COCOMO II, SLIM, SEER-SEM) in order to improve the accuracy

[13] Effort is measured mostly in man-days but also in man-hours and man-months.

[14] Work Breakdown Structure (WBS) – hierarchical decomposition of project tasks into smaller components.

of estimates. Software measurement standard ISO/IEC 14143–6:2012 has been developed to provide guidelines for the Functional Size Measurement Method (FSMM) implementation. It is complemented with other measurement norms, for instance[15]:

- ISO/IEC 20926, developed by the International Function Point Users Group (IFPUG);
- ISO/IEC 20968, developed by the United Kingdom Software Metrics Association (UKSMA);
- ISO/IEC 24570, proposed by the Netherlands Software Metrics Association (NESMA);
- ISO/IEC 19761, developed by the Common Software Measurement International Consortium (COSMIC);
- ISO/IEC 29881, developed by the Finnish Software Metrics Association (FiSMA).

Both size estimation and parametric models are mostly used for medium-sized and large projects and depend on design parameters and mathematical algorithms. Techniques based on lines of codes (SLOC) are applied to estimate the amount of effort per person-month required to develop a program as well as to quantify productivity or effort once the software is produced[16] using size measurement in code lines. In contrast, the function points analysis (FPA) is independent from the technology or programming language used for building software and focuses on measuring the delivered functionality from the end user's perspective. This approach is based on the gathered requirements, system's inputs, outputs, internal logic files, interface files and queries in order to calculate function points and ultimately estimate effort needed to conduct a project.

For the last thirty years parametric and size based techniques (SLOC and FPA) have been extensively developed in order to meet expectations of practitioners. They provide an alternative to traditional techniques, incorporate mathematical algorithms, overcome subjective judgment of project managers and mostly result in improvement of estimation accuracy especially for large projects. Nevertheless, these methods have substantial drawbacks that exclude them from a wide usage. Firstly, poor sizing inputs and excessive optimism often result in underestimation of the scope and ultimately effort. Secondly, estimation

[15] B. Czarnacka-Chrobot, *Standardization of Software Size Measurement*, in: *Internet – Technical Development and Application*, eds. E. Tkacz, A. Kapczynski, Springer-Verlag, Berlin 2009, pp. 149–156.

[16] B. Boehm, D. Reifer, *Software Sizing, Estimation and Risk Management*, Auerbach, USA 2006, p. 10.

models are based on simple algorithms that do not include a variety of factors which influence a project. Additionally, they do not cope with the heterogeneity of data, which results in poor adjustment and high deviations. Finally, they focus on software projects, where an outcome is an implemented or enhanced software system. As a result, change management projects, which play an important role in every organisation's project portfolio, are excluded.

Given the deficiency of the approaches stated above, a need for new, more advanced techniques has emerged that would improve the estimation process of effort at the initial project stages and ultimately increase project success rates. Therefore, researchers have turned their attention into data mining and machine learning, which are recently increasingly popular, as mentioned in section 3. For the last ten years, numerous publications have been written on application of data mining techniques to support estimation of resources needed to conduct a project. The next paragraphs present examples of different approaches taken and techniques used by researchers for the mentioned problem.

For effort estimation Dzega and Pietruszkiewicz[17] decided to build predictive models using classification techniques (the categorical target variable) instead of regression ones that are considered to be less accurate. To do so, they decided to apply a database of open source projects: 'A Repository of Free/Libre/Open Source Software Research Data' from SourceForge.net platform. The database is divided into four datasets, each representing a different estimation aim: duration, time of task completion, the number of working hours spent by a particular project contractor on task completion and the number of completed tasks as of the date of diagnosis. For data pre-processing and variables, the selection information gain ratio was used. Each dataset consisted of approximately six input variables such as the number of selected project tasks, the software language or the number of project contractors. Models were built using three decision trees techniques: C4.5, the random tree (RT), and the classification and regression tree (CART). The most accurate predictions were given by the random tree algorithm (approx. 85% for all the datasets) but Dzega and Pietruszkiewicz indicated that a proper configuration of other two may result in a similar performance. Additionally, the authors explored the utilization of two boosting methods, both the adaboost and bagging metaclassifier increased accuracy of

[17] D. Dzega, W. Pietruszkiewicz, *Classification and Metaclassification in Large Scale Data Mining Application for Estimation of Software Projects*, IEEE 9th International Conference 2010.

prediction[18] by a few percent. In the paper's conclusions the authors indicated that in the preliminary analysis two algorithms widely applied and known for their accuracy, namely the neural networks and support vector machines, were excluded due to their poor performance. It needs to be stated that the database used comes from an open source platform, therefore, the credibility of the data may be questioned. Additionally, it can also be concluded that for each dataset there can be obtained different results using specific techniques. Therefore, the designed model prepared for industrial deployment cannot rely on a single algorithm in order to be capable of adapting to different organisational cultures, methodology in place and foremost specific project data.

Bakir, Turhan and Bener[19] proposed an extension to the classification approach. Instead of manually defining categories of output, variable clustering can be used in order to determine effort classes. Based on two public datasets (overall seven datasets) the Promise Data Repository and the Software Engineering Research Laboratory (SoftLab) Repository, they firstly grouped similar projects together by applying the cluster analysis, determined the effort intervals for each cluster and classes within each cluster and at the end classified new projects into one of the effort classes using the k-nearest neighbour (k-NN), linear discrimination (LD), and decision tree (DT). The first one performed the best, LD slightly worse but the underperformer was DT. Bakir, Turhan and Bener achieved the prediction accuracy for all the data sets of around 90%. In contrast, a similar study[20] using the Promise dataset and classification techniques but without the utilization of cluster analysis for interval definition obtained 70% of the positive prediction rate. Therefore, it can be stated that the approach proposed by the researchers may increase the accuracy of effort estimation using classification techniques.

For estimation of resources at the initial project stages data mining predictive algorithms can be applied, which aim to indicate the exact value rather than effort class to which a new project belongs. Barcelos Tronto, Simoes da Silva and Santa'Anna[21] in their paper applied the artificial neural network and

---

[18] Prediction accuracy is validated mostly by the coefficient of determination (R-squared) also accompanied with the magnitude of relative error (MRE), mean magnitude of relative error (MMRE) and median magnitude of relative error (MdMRE).

[19] A. Bakir, B. Turhan, A. Bener, *A Comparative Study for Estimating Software Development Effort Intervals*, "Software Quality Journal" 2011, vol. 19, pp. 537–552.

[20] P. Sentas, L. Angelis, I. Stamelos, *Multinominal Logistic Regression Applied on Software Productivity Prediction*, 9th Panhellenic Cconference in Informatics, Thessaloniki 2003.

[21] I. Barcelos Tronto, J. Simoes da Silva, N. Sant'Anna, *Comparison of Artificial Neural Network and Regression Models in Software Effort Estimation*, International Joint Conference on Neural Networks, Orlando 2007, pp. 771–776.

linear regression on a publicly available COCOMO dataset[22]. Both techniques performed well with the r-square for ANN 85% and LR 83%. They compared their results with the traditional models analysed by Kemerer[23] on the same dataset. For the Function Point Analysis, COCOMO and SLIM he obtained the r-square 58%, 70% and 89%, respectively. Although ANN and LR presented a significant advantage over the traditional method it needs to be stated that the dataset used (53 projects after pre-processing) could cause overfitting. For that reason, it is recommended to train data mining models on large datasets.

Villanueva-Balsera, Ortega-Fernandez, Rodrigez-Montequin and Concepcion-Suarez[24] for their research used a large ISBSG[25] dataset which consisted of approx. 6000 information technology related projects from different institutions across the world. Their aim was to build an effort estimation model using the predictive MARS decision tree technique which handles well numerical input values. To achieve that, they decided to follow the CRISP-DM[26] methodology. After data pre-processing, 2000 records were selected due to missing values, data quality issues and other reasons. Self-organizing maps were used to identify relations among the data. After that, the authors set the output continuous variable as the effort measured in hours and trained the model using the MARS technique. When comparing the real value of effort with the estimated one, they achieved the r-square of 88%, which is considered as a robust result. Additionally, the researchers conducted a sensitivity analysis to discover variables which impact estimated effort discovering the influence of such attributes as the used development platform, type of the programming language and type of the sizing method used. Although the achieved accuracy is quite substantial the deployed model used on a different data set in a real life scenario could behave variously. Therefore, the addition of other techniques like the neural network

---

[22] Database of 63 software projects used by Barry W. Bohem to develop the Constructive Cost Model (COCOMO)

[23] C. Kemerer, *An Empirical Validation of Software Cost Estimation Models*," Communication of ACM" 1987, vol. 30, pp. 416–429.

[24] J. Villanueva-Balsera, F. Ortega-Fernandez, V. Rodrigez-Montequin, R. Concepcion-Suarez, *Effort Estimation in Information Systems Projects Using Data Mining Techniques*, "Proceedings of the WSEAES 13th International Conference on Computers" 2009, pp. 652–657.

[25] International Software Benchmarking Standards Group (ISBSG) project repository, The newest release R12 consists of approx, 6000 IT projects from 1989–2012.

[26] Cross Industry Standard Process for Data Mining (CRISP-DM) – data mining process model that consists of six phases: business understanding, data understanding, data preparation, modeling, evaluation and deployment.

and generalized linear model to support effort prediction could enable the application of the model in practice.

In addition to the above-mentioned publications, a large number of studies have been performed to assess the most accurate effort estimation approach and techniques but no vital conclusion has been reached up to date. Dejaeger, Verbeke, Martens, and Baesens[27] strived to address this issue and performed a comprehensive study using nine available project datasets commonly used by researchers (Coc81, and Desharnais, Cocnasa, Maxwell, and USP05, Experience, ESA, ISBSG, and Euroclear) and thirteen different models[28] including tree/rule-based models (M5 and CART), linear models (various types of linear regression), nonlinear models (MARS, multilayered perceptron neural networks, radial basis function networks, and least squares support vector machines), and estimation techniques (the case-based reasoning approach). In their benchmark study, they excluded the observations with missing values due to inability of some techniques to cope with it. Moreover, they removed variables that are not known at the moment of the effort estimation like duration or cost. For each dataset an individual model was build using different techniques. The results indicated that the ordinary least squares regression in combination with a logarithmic transformation preformed the best. Surprisingly, nonlinear models like the neural network and decision trees, known from their good prediction capabilities based on previous studies, struggled with accuracy. The explanation to this can lay in the data pre-processing and model calibration approach taken by Dejaeger, Verbeke, Martens, and Baesens. The small data sets used (apart from ISBSG), exclusion of missing values, the standard data preparation approach to all the models and unknown model adjustment parameters could result in the findings presented in this publication. The authors conclude their study with a comparison to traditional methods stating that data mining techniques provide a valuable contribution to the estimation process but should not be used as a standalone approach, rather in conjunction with expert judgment and domain knowledge. From their results in can be stated that due to the heterogeneity of datasets the multiple model combination approach could yield an improvement in prediction

[27] K. Dejaeger, W. Verbeke, D. Martens, B. Baesens, *Data Mining Techniques for Software Effort Estimation: A Comparative Study*, "IEEE Transactions on Software Engineering" 2012, vol. 38, pp. 375–397.

[28] Ordinary least squares regression, OLS regression with log transformation, OLS regression with Box Cox (BC) transformation, robust and ridge regression, Least median squares regression, MARS, CART, model tree, MLP neural network, Radial basis function networks, Case-based reasoning, Least squares support vector machines.

accuracy. Additionally, from this study it can be also concluded that the data pre-processing approach like missing values input algorithms plays an important role and if not properly applied, may result in misleading estimates.

4.2. Project Monitoring

After conducting a project estimation, it is crucial to track any deviations at every stage due to occurring risks and uncertainty. Software estimation performed at an early stage is not sufficient to be relied on, therefore, it requires controlling and refinement. Proper monitoring of the resources used up to date and required to complete the project is an essential process that ensures the project completion within the assumed constraints. Any aberration discovered in terms of project effort or duration should be altered in order to prevent a negative impact on the project budget and schedule.

There are numerous project monitoring approaches for both cost and duration and all require manual calculations from the project team of the progress up to date, resources consumed and needed to perform the remaining work. They are mostly based on performance reviews, variance analysis and forecasting. The most significant one is recognized Earned Value Management (EVM) which emerged initially for financial analysis but later was adopted for project monitoring. According to the Project Management Institute EVM is "a methodology that combines scope, schedule, and resource measurements to assess project performance and progress. It integrates the scope baseline with the cost baseline, along with the schedule baseline, to form the performance baseline, which helps the project management team assess and measure project performance and progress"[29]. For assessment EVM uses indicators like planed, earned and actual value (PV, EV, AC) to compute schedule and cost variances. Ultimately, this technique allows for calculation of the total resourced needed to perform the project during its progress though estimation at completion (EAC) or required to finish all tasks – estimate to complete (EAC).

In order to mitigate manual calculations that may be error prone and overlook important factors influencing a project, data mining techniques may be used to support controlling the baseline. Iranmenesh and Mokhtari[30] suggested an application of decision trees, neural network and association rules

[29] Project Management Institute, op.cit, pp. 226.

[30] S. Iranmanesh, Z. Mokhtari, *Application of Data Mining Tools to Predicate Completion Time of a Project*, "World Academy of Science: Engineering and Technology" 2008, vol. 42.

for predicting the total project duration during the execution phase using EVM and EAC indicator as an output. For this purpose, they built their models on a Kulish– Hartmann dataset (Project Scheduling Problem Library – PSPLIB) using one project with 90 activities updated day by day. Firstly, they calculated EVM input indicators like PV, EV, AC, EAC (9 variables in total) for 76 periods focusing on duration. Secondly, they built models using decision trees, neural network and association rules with EAC as an output variable. At the end, they interpreted the results for each model but not indicated the achieved accuracy. Nevertheless, their work presents a great potential of DM application for predicting overall effort and duration needed to complete a project (EAC) during the execution phase due to EVM popularity of use in practice and could be extended in further research, ultimately deployed within organisations.

Another interesting approach to the problem was proposed by Azzeh, Crowling and Neagu[31]. They focused on predicting stage effort utilizing information from previous ones. The proposed approach combines the fuzzy set theory and association rule mining. The first technique was used to define the corresponding linguistic variable for each interval achieved after dividing the dataset. Next, the association rules between the prior stage and target stage were determined using the apriori algorithm and the calculation of predicted output was made by defuzzifing all the expected outputs with relation to the target stage. For building the model they used ISBSG dataset which contains information about the effort for each of six phases (plan, specification, design, building, testing, implementation). The archived results show that for the majority of stages the model presents a good prediction capability (MdMRE<25%). Additionally, when compared to the exponential regression, they present a significant improvement in estimation accuracy. Nevertheless, due to the deficiency of comprehensive research studies conducted in the stage efforts estimation area it would be beneficial to validate the results with other data mining techniques which could be applied for this purpose.

4.3. Software Quality

Poor software quality delivered has been commonly recognized as one of the reasons for project struggling or even failure. In recent years, a significant

[31] M. Azzeh, P. Cowling, D. Neagu, *Software Stage-Effort Estimation Based on Association Rule Mining and Fuzzy Set Theory*, 10th IEEE International Conference on Computer and Information Technology 2010, pp. 249–256.

amount of effort has been put in order to implement proper standards and procedures within organizations that aim to deal with that problem. Most companies have put in place the ISO 9001 quality management standard that strives to support any organization regardless of its field of activity. For software quality assurance the ISO 9126 norm was designed originating in 1996, and revised in 2001. The main characteristic of this standard was the recognition of human biases, both external and internal, that may affect product usability and end-user perception[32]. In 2011 the model was replaced by ISO 25010 Systems and software Quality Requirements and Evaluation (SQuaRE). Nevertheless, it can be observed that from year to year quality of software delivered is decreasing contributing to budget and schedule overruns, and ultimately stakeholders', sponsors' and users' dissatisfaction.

Prior software deployment is mostly verified, but not limited to, in the implementation phase thorough conducting various tests like development, functional or user acceptance. During these activities deviations from the desired functionality (commonly known as failures or bugs) are identified and captured with respective attributes in defect logs or bug tracking systems (i.e. JIRA) that are updated during their lifecycle. The data stored provides a great opportunity for the application of various techniques for defect prevention, detection and correction. The traditional approach is based on line of codes and COCOMO II (i.e. Constructive Quality Model- COQUALMO) or the linear regression technique. These models assume that bugs are directly related to software complexity ignoring human bias, programmers coding preferences or poor software specifications. Therefore, for the last twenty years machine learning approach has been researched in order to predict product quality, the number and complexity of potential bugs or fix duration. The information discovered allows project managers to improve test planning: allocation of required resources and preparation of more accurate test schedules, which is reflected in the examples below.

Villanueva-Balsera, Rodrigez-Montequin, Ortega-Fernandez, Rodrigez-Montequin and González-Fanjul[33] developed a model for prediction of minor, major and extreme defects at the initial stage of a project. For this purpose, they used MARS decision trees to build three separate models, each predicting independently the number of potential bugs in the defect category. They trained them on ISBSG

[32] A. Kobyliński, *ISO/IEC 9126 – Analiza Modelu Jakości Produktów Programowych*, Konferencja „Systemy Wspomagania Organizacji" 2003.

[33] J. Villanueva Balsera, V. Rodriguez Montequin, F. Ortega Fernandez, C. Alba González-Fanjul, *Data Mining Applied to the Improvement of Project Management*, in: *Data Mining Applications in Engineering and Medicine*, ed. A. Karahoca, InTech 2012.

dataset and used twenty input variables such as effort per each phase, duration, methodology, software language, and development type or team size. The obtained results differ slightly for each model but present prediction accuracy above 90%, which may be recognised for models deployment. Therefore, they propose the implementation of the solution in a real life situation, flagging that the models need to be trained on the data specific to a particular organisation. Additionally, the approach may be combined with the project effort estimation model resulting in prediction of effort and product quality at project initiation that may improve the project planning process.

Data stored in software bug repositories from previous projects can be utilized as well for defect fix duration. Nagwani and Bhanasli[34], using open source MySQL bug dataset, proposed a three-step method in order to extract this information. Firstly, they clustered the defect repository based on bug complexity (simple, medium, complex) and known fix duration using the k-means algorithm. For a new bug fix duration is predicted using the mentioned technique then assigned to a proper complexity group. A different approach to the problem was suggested by Wiess, Premraj, Zimmermann and Zeller[35]. In order to predict fix duration, they used effort spent on fixing an issue as an output and the combination of the k-nearest neighbour (k-NN) technique and text similarity measuring engine- Lucene[36]. The model queried J. Boss project dataset of resolved issues for similarities using primarily the k-NN with the support of Lucene approach, which is commercially widely applied for text comparison (i.e. FedEx, New Scientist Magazine, MIT). The resulted model demonstrated a good prediction capability and outperformed the naive bayesian approach executed on the same dataset[37]. It may be concluded that an addition of other methods to DM techniques may boost prediction results.

---

[34] N. Nagwani, A. Bhansali, *A Data Mining Model to Predict Software Bug Complexity Using Bug Estimation and Clustering,* International Conference on Recent Trends in Information, Telecommunication and Computing 2010, pp. 13–17.

[35] C. Weiß, R. Premraj, T. Zimmermann, A. Zeller, *How Long Will it Take to Fix This Bug?*, International Conference on Software Engineering, IEEE Computer Society Washington, USA 2007.

[36] Apache Lucene is a free/open source information retrieval software library.

[37] D. Cubranic, G. Murphy, *Automatic Bug Triage Using Text Categorization*, "International Conference on Software Engineering & Knowledge Engineering (SEKE)" 2004, pp. 92–97.

4.4. Maintenance Effort

At the beginning of a project or prior to deployment, when uncertainty of the delivered product complexity decreases, it is vital to establish the effort involved in software maintenance. The information is crucial especially when the project transfers to the business as a usual mode and a maintenance budget for next year(s) needs to be put in place. Additionally, it is beneficial information during a project set up and business case justification.

Traditional estimation approaches based on line of codes and function points focus mostly on programming effort, overlooking additional factors that influence the maintenance cost. Therefore, they are used as partial input variables for machine learning models that take into consideration additional ones crucial for effort estimation.

Shukla and Misra[38] in their work proposed an approach to the problem with the use of the neural network technique. Their model utilized fourteen input cost driver variables (project attributes) like the number of files, datasets, the nature of service level agreement and number of time zones the users that need support are spread across. They extended their research in another publication[39] with addition of the system dynamics and designed a hybrid model due to ongoing modifications implemented to the maintained system. This is caused mostly by changes to configuration, user requirements or turnover of personnel which impacts input variables and ultimately the model accuracy. Shukla, Misra, Marwala and Clark apart from 14 static attributes applied additional five dynamic factors with the corresponding effort in order to train the NN model (incremental learning). Although a large number of attributes were input, they archived a good prediction capability (the r-square around 85%), which they concluded with an indication that it preforms better than without dynamic factors. However, the model was built based on a survey dataset, data collected from industry experts, which tend to be biased. Therefore, further research should be conducted on a real life dataset like ISBSG.

[38] R. Shukla, A. Misra, *Estimating Software Maintenance Effort – A Neural Network Approach*, "Software Engineering Conference Proceedings 2008, pp. 19–22.

[39] R. Shukla, M. Misra, A. Misra, T. Marwala, W. Clarke, *Dynamic Software Maintenance Effort Estimation Modeling Using Neural Network, Rule Engine and Multi-regression Approach*, "Computational Science and Its Applications – ICCSA" 2012, pp. 157–169.

5. Conclusions

The aim of this paper was to provide an overview of how data mining may be applied to the project management field through presenting different examples of studies conducted by researchers up to date. Due to the imperfection of traditional techniques and high rate of project failure, a need for additional tools to support project team members arose. The researchers came up with various solutions utilizing machine learning and other data mining models whose popularity has been growing in recent years because of their known estimation accuracy.

The paper presented the usefulness of data mining models in all stages of a project's lifecycle- from software estimation, monitoring, quality assurance to maintenance. However, the usage of DM goes beyond that. The most explored area by researchers is initial software estimation where different approaches like classification, prediction or clustering were proposed. This area is the most crucial from project practitioners' perspective due to high uncertainty and a lack of effective tools to support the estimation process of effort, cost and duration. During the execution phase data mining may be used for monitoring that a project is on track, detecting any deviations and proposing new estimates. Additionally, it may improve testing the planning process, schedule and resources required by estimating the number of potential defects and the effort needed to fix them. Finally, when a project business case is created, but prior to product deployment due to budgeting purposes, it is essential to predict an effort involved with product maintenance and, as presented, data mining can support this as well.

Although the models proposed by researchers present good prediction accuracy, it can be noticed that practically there are no real life deployments within organizations and data mining is not being used by project practitioners to support their activities. The reason for this may potentially be that the proposed approaches focus on individual models and techniques and mostly are tailored to a particular dataset instead of presenting complex models robust to noise, change and data heterogeneity. Moreover, very little is shown how to put them in practice. Thanks to common deployment of credit scoring models in the finance sector using for example IBM-SPSS or SAS software it can be achieved with moderate customization required. The models can be trained on the existing organizational project database, tweaked for desired needs and deployed within the project management office to provide an additional supportive tool for project managers, project practitioners in their day-to-day operations.

It can be concluded that certainly there is a need for advanced machine learning models in project management but they should be treated as a complementary decision support tool used in conjunction with traditional ones in order to truly boost estimation accuracy and minimize error factors. This may in overall contribute to an increase in a project's success rate and customer satisfaction.

References

Azzeh M., Cowling P., Neagu D., *Software Stage-Effort Estimation Based on Association Rule Mining and Fuzzy Set Theory*, 10th IEEE International Conference on Computer and Information Technology 2010, pp. 249–256.

Bakir A., Turhan B., Bener A., *A Comparative Study for Estimating Software Development Effort Intervals*, "Software Quality Journal" 2011, vol. 19, pp. 537–552.

Barcelos Tronto I., Simoes da Silva J., Sant'Anna N., *Comparison of Artificial Neural Network and Regression Models in Software Effort Estimation*, International Joint Conference on Neural Networks, Orlando 2007, pp. 771–776.

Berry M., Linoff G., *Data Mining Techniques for Marketing Sales and Customer Support*, Wiley, USA 2004, p. 7.

Boehm B., Reifer D., *Software Sizing, Estimation and Risk Management*, Auerbach, USA 2006, p. 10.

Cubranic D., Murphy G., *Automatic Bug Triage Using Text Categorization*, International Conference on Software Engineering & Knowledge Engineering (SEKE) 2004, pp. 92–97.

Czarnacka-Chrobot B., *Standardization of Software Size Measurement*, in: *Internet – Technical Development and Application*, eds. E. Tkacz and A. Kapczynski, Springer-Verlag, Berlin 2009, pp. 149–156.

Dzega D., Pietruszkiewicz W., *Classification and Metaclassification in Large Scale Data Mining Application for Estimation of Software Projects*, IEEE 9th International Conference 2010.

Fayyad U., Piatetsky-Shapiro G., Smyth P., *From Data Mining to Knowledge Discovery in Databases*, American Association for Artificial Intelligence, USA 1996.

Gasik S., *A Model of Project Knowledge Management*, Wiley Periodicals, USA 2010, pp. 3–4.

Iranmanesh S., Mokhtari Z., *Application of Data Mining Tools to Predicate Completion Time of a Project*, "World Academy of Science: Engineering and Technology" 2008, vol. 42.

Kantardzic M., *Data Mining: Concepts, Models, Methods, and Algorithms*, Wiley, USA 2011.

Kemerer C., *An Empirical Validation of Software Cost Estimation Models*, "Communication of ACM" 1987, vol. 30, pp. 416–429.

Kobyliński A., *ISO/IEC 9126 – Analiza Modelu Jakości Produktów Programowych*, „Konferencja Systemy Wspomagania Organizacji" 2003.

Larose D., *Discovering Knowledge in Data*, Wiley, USA 2005, pp. 11–18.

Nagwani N., Bhansali A., *A Data Mining Model to Predict Software Bug Complexity Using Bug Estimation and Clustering*, "International Conference on Recent Trends in Information, Telecommunication and Computing" 2010, pp. 13–17.

Project Management Institute, *A Guide to the Project Management Body of Knowledge (PMBOK®guide)*, Project Management Institute, USA 2013, 5th ed., pp. 27–28, 61, 226, 418.

Sentas P., Angelis L., Stamelos I., *Multinominal Logistic Regression Applied on Software Productivity Prediction*, 9th Panhellenic Conference in Informatics, Thessaloniki 2003.

Shukla R., Misra A., *Estimating Software Maintenance Effort – A Neural Network Approach*, "India Software Engineering Conference Proceedings" 2008, pp. 19–22.

Shukla R., Misra M., Misra A., Marwala T., Clarke W., *Dynamic Software Maintenance Effort Estimation Modeling Using Neural Network*, Rule Engine and Multiregression Approach, "Computational Science and Its Applications – ICCSA" 2012, pp. 157–169.

The Standish Group International, *Chaos Summary for 2010*, Boston 2010, p. 3.

Villanueva Balsera J., Rodriguez Montequin V., Ortega Fernandez F., Alba González-Fanjul C., *Data Mining Applied to the Improvement of Project Management*, in: *Data Mining Applications in Engineering and Medicine*, ed. A. Karahoca, InTech 2012.

Villanueva-Balsera J., Ortega-Fernandez F., Rodrigez-Montequin V., Concepcion-Suarez R., *Effort Estimation in Information Systems Projects Using Data Mining Techniques*, in: Proceedings of the WSEAES 13th International Conference on Computers 2009, pp. 652–657.

Weiß C., Premraj R., Zimmermann T., Zeller A., *How Long Will It Take to Fix This Bug?*, International Conference on Software Engineering, IEEE Computer Society Washington, USA 2007.