The Journal of Systems and Software 100 (2015) 80–90

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Comprehensible software fault and effort prediction: A data mining approach

Julie Moeyersoms[a,*], Enric Junqué de Fortuny[a], Karel Dejaeger[b], Bart Baesens[b], David Martens[a]

[a] Faculty of Applied Economics, University of Antwerp, Prinsstraat 13, Antwerp B-2000, Belgium
[b] Department of Decision Sciences and Information Management, Katholieke Universiteit Leuven, Naamsestraat 69, Leuven B-3000, Belgium

ARTICLE INFO

Article history:
Received 26 September 2013
Received in revised form 6 October 2014
Accepted 19 October 2014
Available online 27 October 2014

Keywords:
Rule extraction
Software fault and effort prediction
Comprehensibility

ABSTRACT

Software fault and effort prediction are important tasks to minimize costs of a software project. In software effort prediction the aim is to forecast the effort needed to complete a software project, whereas software fault prediction tries to identify fault-prone modules. In this research both tasks are considered, thereby using different data mining techniques. The predictive models not only need to be accurate but also comprehensible, demanding that the user can understand the motivation behind the model's prediction. Unfortunately, to obtain predictive performance, comprehensibility is often sacrificed and vice versa. To overcome this problem, we extract trees from well performing Random Forests (RFs) and Support Vector Machines for regression (SVRs) making use of a rule extraction algorithm ALPA. This method builds trees (using C4.5 and REPTree) that mimic the black-box model (RF, SVR) as closely as possible. The proposed methodology is applied to publicly available datasets, complemented with new datasets that we have put together based on the Android repository. Surprisingly, the trees extracted from the black-box models by ALPA are not only comprehensible and explain how the black-box model makes (most of) its predictions, but are also more accurate than the trees obtained by working directly on the data.

1. Introduction

The worldwide enterprise software development market was valued at $244 billion in 2010 according to information technology research and advisory firm Gartner, which proves the importance of this sector globally (Dejaeger, 2012). Yet, the software industry suffers from frequent cost overruns (Jørgensen and Moløkken-Østvold, 2006; Uwano et al., 2011; Grimstad et al., 2006). As a consequence this can lead to serious problems for software companies and sometimes even jeopardize their existence (Bloch et al., 2012). It is therefore important as a software development company to minimize costs as much as possible. In order to do so, activities such as software effort estimation and software fault prediction can be crucial, and constitute the topic of this paper. Software effort estimation is the basis for project bidding, budgeting and planning (Jørgensen and Moløkken-Østvold, 2006). Software fault prediction on the other hand aims to identify error prone software modules in a timely manner (Lessmann et al., 2008). It is crucial to identify faults in the early stages of development since the cost of fixing or reworking software can be surprisingly high if they are detected in the later phases of the software development life cycle (Dejaeger et al., 2013; Fagan, 1999; Boehm and Papaccio, 1988).

In this paper we predict software faults and effort, making use of different data mining techniques. Data mining entails the process of extracting knowledge from large amounts of data (Vandecruys et al., 2008). In the literature (see e.g. Witten et al., 2011) different types of data mining are discussed such as regression, classification and association rule mining. Regression and classification are predictive data mining tasks, where the target variable is continuous and discrete respectively. Association rule mining is a descriptive data mining task and aims at learning frequently occurring patterns (Vandecruys et al., 2008). The focus in this research lies on regression for software effort prediction and classification for software fault prediction. In both cases, statistical predictive models are built in order to generate predictions of new observations (Shmueli and Koppius, 2011). A simplified example for both prediction tasks is presented in Figs. 1 and 2, where it is shown that a classification or regression model is built based on historical data in order

* Corresponding author. Tel.: +32 3 265 42 05.
E-mail addresses: julie.moeyersoms@uantwerpen.be (J. Moeyersoms), Karel.Dejaeger@kuleuven.be (K. Dejaeger), Bart.Baesens@kuleuven.be (B. Baesens), david.martens@uantwerpen.be (D. Martens).

http://dx.doi.org/10.1016/j.jss.2014.10.032
0164-1212/

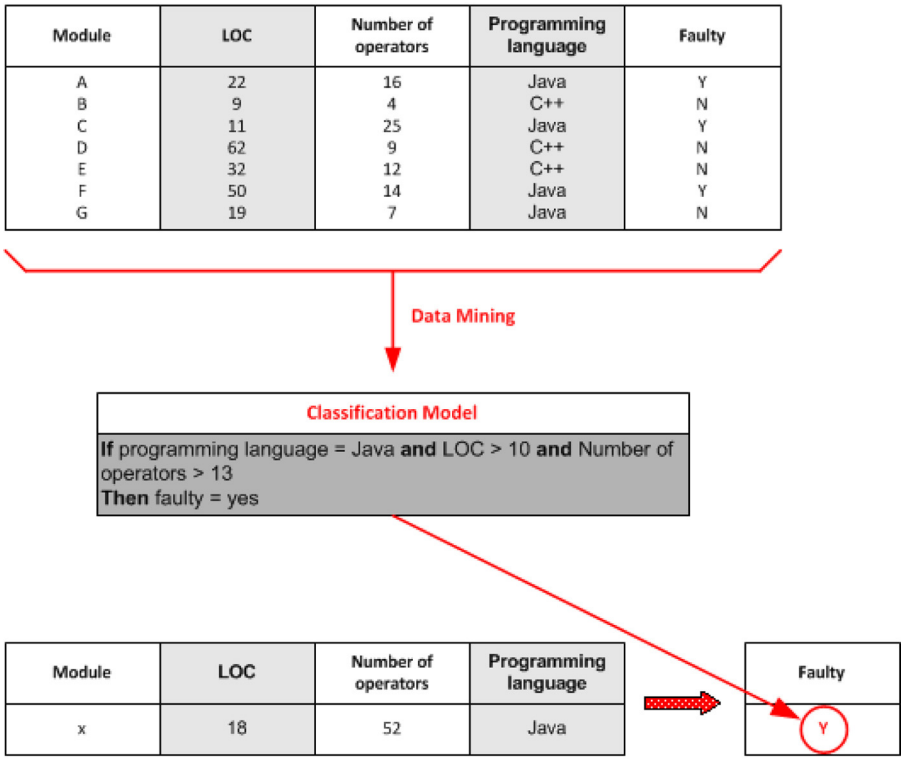

Fig. 1. Building a classification model with data mining.

to generate accurate predictions of new observations. Data mining techniques are applied in many domains. Some well-known examples include credit scoring (Baesens et al., 2003b), churn prediction (Verbeke et al., 2012) and applications in the medical sector such as the selection of the best in-vitro fertilized embryo (Passmore et al., 2003).

Although research on fault and effort prediction often emphasizes the predictive performance of a model, comprehensibility is

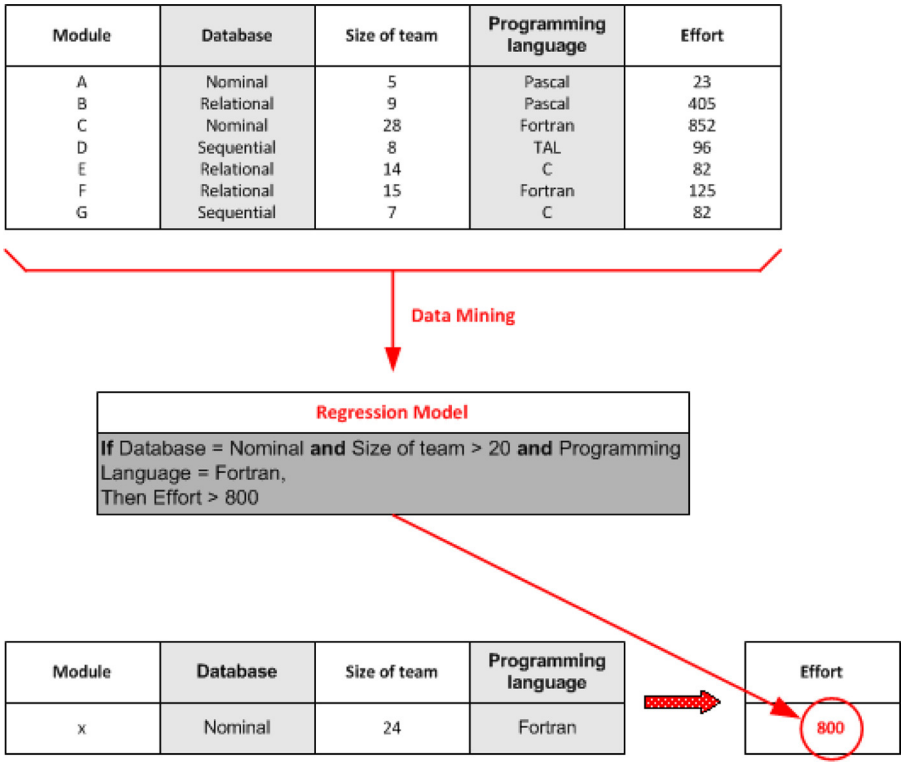

Fig. 2. Building a regression model with data mining.