Master Thesis Summary: *title of the thesis*

Jacopo Nasi [s255320]

February 21, 2020

1 Introduction

The development of a software product is not different from any other product development, after the first phase of design the coding phase starts, during it problems emerge and must be fixed before the release of the product.
A software project performs several commits per day, each commit contains a lot of information that can be used for statistical analysis. Predicting defectivity during the development of the software can drastically improve the process, allocating the correct number of developers to fix defects, therefore reducing the time required for fixes. Also machine learning models can be used to predict the defectivity in a software project.

2 Data

The application of machine learning methods needs data from the past to train models then used for prediction.
Having the data is not enough to train good models, data must be cleaned, sorted and maybe even aggregated. This phase is called pre-processing and is used to enhance the data before the model training phase.
The work of this thesis is based on an open-source database called SEOSS33 that is a collection of issues, comments, and changes from 33 open source projects. The data is organized around the issue, the central concept of the development process. Attached to an issue there is other information like comments,

timestamps, version number, and other features. Preprocessing to clean the data before training involves removing useless data, adding developer seniority, a bag of word of components, release version and other mathematical related information. Then the data is aggregated per week, computing the difference between opened and closed issues, coding the severity of each issue (critical, major, trivial, minor, blocker).

The data is then generated to perform prediction among different time horizons, from 1 week to 52.

3 Forecasting

Once the data is pre processed the ML algorithms can be applied. The goal is to predict defects of a project for a future period, that ranges from 1 to 52 weeks. Three different models are chosen to perform the forecasting: Random Forest, Gradient Boosting, Neural Networks and also some predefined framework will be taken into account for the result evaluation. The training phase is applied using three different methodologies: the first apply training and prediction over the same dataset, the second, cross-version, use development data of the previous version to forecast the following and last, cross-project, predict the trend of a specific version after have been trained on data of different software project.

All the different ways can achieve good results, the best is the last one, cross-project with a precision greater than 90% up to 4 weeks and still greater than 70% up to 20 weeks.