Master Thesis Summary: *title of the thesis*

Jacopo Nasi [s255320]

February 20, 2020

1 Introduction

The development of a software product is not different from any other product development, after the first phase of design the production starts, during it problems emerge systematically and must be managed before the release in production.

Every day each a software house performs hundreds of commit, each one contains a lot of information linked to an issue, with structured commit, driven issue report and other software engineering stuff, the quality of the sourcing improve, allowing data to be used for statistical analysis. Predicting defectiveness during the making of the software can drastically improve the process of development, allocating the correct number of developers could reduce the time required and avoid delays and problems before releases. With a correct preprocessing and evaluated aggregation, machine learning models can be used to predict the defectiveness trend. These forecast can be further used to improve the software engineering background of the related project.

2 Data

The application of statistics and machine learning methods is based on data, the previous experience is used to drive the model train to improve the quality of future prediction.

Having the data is not enough to train good models, it must be cleaned, sorted and maybe even aggregated, this phase is called pre-processing and is used to enhance the data before passing it through the mathematical model.

This project is based on an open-source database called SEOSS33 that is a collection of issue, comments and changes from sourcing platforms of 33 open source projects. The data is organized around the issue, the central entity of the data model, from there a lot of other information like comments, timestamps, version and other features can be retrieved, the granularity of the data is high and a preprocessing phase is required to prepare the data for the training. The first part removes useless data, the second enrich the collection with data extracted by aggregating data: developer seniority, a bag of word of components, release version and other mathematical related information. Then the data is weekly aggregated computing, like target value, the difference between opened and closed issue by translating labels (critical, major, trivial, minor, blocker) to the mean value of the duration distribution of each.

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

3 Forecasting

Once the data is available the algorithms can be applied. Four 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 new one and the last one, 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.