Master Thesis Summary: *title of the thesis*

Jacopo Nasi [s255320]

February 16, 2020

1 Introduction

The development of a software product is not different from any kind of hardware 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 perform hundreds of commit, each one contains a lot of informations 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 in artificial intelligence analysis. Predicting defectiveness during the making of the software can drastically improve the process of development, allocating the correct number of developer could reduce the time required and avoid delay and problems before releases. With a correct preprocessing and evaluated aggregation, machine learnings 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 artificial intelligence algorithms is based on data, the previous experience is used to corretly train the models to allow better prediction with the new data.

Is not enough to have the data to develop good models, it must be cleaned, sorted and maybe even aggregated, this phase is called pre-processing and is used to improve the data before passing it through the mathematical model.
This project is based on a open source database called SEOSS33, is a collection of issue, comments and changes from sourcing platform of 33 open source projects. The data is organized around the issue, the central unit of the platform, from there a lot of other informations like comments, time stamp, version and other stuffs can be retrived, the granularity of the data is really high and a preprocessing is required to prepare the data for the models. The first part remove useless data, the second enrich the collection with data extracted by aggregating data: developer seniority, 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 the opened and closed issue by assing to them a value related to the priority class (critical, major, trivial, minor, blocker) and the mean value of the duration distribution.
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 artificial intelligence can be applied. Four different models are choosen to perform the forecasting: Random Forest, Gradient Boosting and Neural Networks, also some predefined framework will be compared. the training phase can be structured in three different ways: the first apply training and prediction over the same dataset, the second, cross version, use development data of previous version to forecast the new one and the last one, cross project, predict target of a specific version after have been trainined using 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.