Tampere University

Mikel Robredo Manero

MEASURING THE IMPACT OF SONARQUBE ON THE DEVELOPMENT VELOCITY USING REGRESSION ANALYSIS

Master of Science Thesis
Faculty of Information Technology and Communication Sciences (ITC)
Thesis tutor: Professor Davide Taibi
May 2023

ABSTRACT

Mikel Robredo Manero: Measuring the impact of SonarQube on the development velocity using regression analysis
Master of Science Thesis
Tampere University
May 2023

The study of development velocity has gained importance in software engineering research within the last decades. Not only software development projects but many fields are interested in analyzing the impact specific factors have on the development velocity, since this one stands as a useful metric to measure the productivity with which teams perform when working on different types of tasks. One of these factors is SonarQube, a widely used software considered to be one of the most used code analysis tools by developers in software development.

This thesis aims to analyse the impact of SonarQube as a factor affecting the variance of the development velocity in software development projects. Furthermore, based on expert knowledge from the field, a set of different confounder variables that are believed to have an impact on the development velocity are included in the analysis. Thus, an additional goal of this thesis is to analyse which is the relationship of the considered variables with the development velocity that better describes its variance. Regression analysis was selected to conduct the analysis of this thesis, and the statistical software R was the computational tool. The collected data included information about 337 mature software development projects in the Apache Software Foundation obtained through a cohort study design.

The conducted analysis considers a complete regression analysis process, first understanding the shape of the data and drawing initial distributional assumptions. Consequently, the analysis considers using Linear Models as well as Generalized Linear Models under the drawn assumptions. By performing a backward selection process variables in models under different distributional assumptions, results showed a low statistical significance in the exposure of projects to SonarQube. Moreover, all the observed models denoted a low predictive power towards the development velocity, hence showing a low ability to describe its variance. Additionally, ensemble learning was used to discover that results behaved in the say way under an agglomerating approach.

In the same way, the model selection showed a better fit with models assuming distributions depicting high skewness. These results suggested that potential work could be done inspecting further non-parametric methods that assume the observed skewness in the distribution of the development velocity. Furthermore, the obtained results do not show the significance of the use of SonarQube to describe the development velocity, a fact that differs from the software development field. This suggests the possibility of finding alternative data collection designs that may understand capture the connection between SonarQube and the development velocity in a more accurate way. These could consider periodic measurements of the velocity level in measurements, as well as a different variable structure when performing regression analysis, among many other options.

Keywords: Regression analysis, Development velocity, Empirical Software Engineering, SonarQube, Cohort Study

PREFACE

Here I present the master thesis "Measuring the impact of SonarQube on the development velocity using regression analysis." This work stands as the final requirement to fulfill the graduation process of the Master's Programme in Computing Sciences - Statistical Data Analytics at Tampere University. The duration of the thesis work started on January 2023 and finished on May 2023.

During my studies in Tampere, I noticed that most of the data analysis related assignments assumed ideal cases were the data existed suitably for the analysis of interest. Once I was done with the compulsory subjects in my programme I was sure that if I wanted to learn from new challenges, my thesis would need to act as a real world case were the data would not be given, but collected by myself. In this way, I have performed complete data collection processes, were I have got used to working with web APIs, be familiar with scripting, as well as make suitable data sets for the specific needs of the moment. Also, I have learnt how the academia world works, as well as in which moments you can struggle in that discipline. Thence, this experience has opened my eyes to new opportunities, same as taught me to be a better professional and team worker.

I would like to thank my two supervisors, Dr. Davide Taibi for being flexible and allowing me to apply my knowledge in a field that was unknown for me before starting this thesis, and specially M.Sc. Nyyti Saarimäki for being there all the time I needed it, helping me to learn skills which were completely new for me and making me see that all of us should always trust in the work we do. I would also like to thank them the space they gave me in their research office as well as the trust they put in me, It made me feel I was part of a team.

Finally, I would like to thank my family for being there during these two years of studies even if 3,381km were in between us. Thank as well my friends from Tampere who have been a crucial part for me in this journey, for what they have thought me and for what we have lived together. And last but most important, I want to thank you Nuria, because you have been the support that has made this ship not to sink during all this time, I love you.

Tampere, 23rd May 2023

Mikel Robredo Manero

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Software repositories are a vital component for many fields like Software Engineering (SE) nowadays, indeed, projects store their data in repositories in an organized way so that data can be selected easily. Within SE, Mining Software Repository (MSR) is one of the research fields that has evolved rapidly during the last years. MSR studies utilize mainly the data available in software repositories, analysing it and understanding features and phenomena within the software development process [1].

Among the different aspects of software development, there is the so-called *development velocity*. In a nutshell, the development velocity can be explained as the productivity of software development teams with their tasks, that is, how much time they need to accomplish a task from the moment in which the same task was opened.

It is a widely known fact that often developers spend time locating and fixing bugs that make the program work incorrectly, rather than creating new features for their clients [2]. Similarly, the same software development field has introduced different tools to help developers fix errors in their code, such as *SonarQube* (SQ), a *Static Analysis Tool* (SAT) introduced in Section 2.2.

As will be shown in Section 2.1, different MSR and not only MSR studies have been performed to measure the impact of SATs on software development team's velocity. One of the main goals of research in SE is to discover which is the most efficient way to develop new software in terms of productivity, and what tools and factors have an impact on this phenomenon. However, no exact approach has been yet identified to present enough robustness to measure this impact.

Consequently, the goal of this thesis is to analyze the impact SQ has on the software development velocity of mature projects. For that, this work seeks to define the best regression analysis that quantitatively describes the statistical significance of SQ, and how this one explains the variance of the development velocity. In parallel, another object of analysis in this thesis is the impact of potentially significant project characteristics on the variance of the development velocity, as well as analyzing whether their impact makes the one generated by SQ to be lost or deprecated.

To conduct this analysis, given the novelty of the chosen type of analysis in this specific topic, different distributional assumptions are considered based on a preliminary

exploratory analysis. Based on the mentioned step, Linear Models (LMs), as well as Generalized Linear Models (GLMs) are utilized for the regression analysis. Within the regression process, a *Backward variable selection* process is followed where considered information criteria and methods are used for model and variable comparison.

The data collection considered for this thesis is part of a cohort study conducted by Saarimäki et. al [3]. The data consists of information about mature software development projects from the *Apache Software Foundation* (ASF) between January 2020 and December 2022.

In the next chapter, a literature review is displayed for the reader covering the specific concepts that are an object of study in this thesis. Next in Chapter 3, a detailed description of the theoretical background behind the conducted regression analysis is given. Then, Chapter 4 describes the observational design followed in this thesis, as well as the different data collection steps performed to obtain the final data set. Chapter 5 presents the results obtained from the performed regression analysis and, finally, Chapter 6 covers the conclusions derived from the work performed in this thesis.

2. LITERATURE REVIEW

This chapter describes the important concepts to understand the background of the analysis performed in this thesis. First, concepts coming from the SE field which are the subject of study in this thesis are described in Section 2.1. Then, Section 2.3 covers the description of cohort studies, which is the theoretical background of the observational design adopted in this thesis.

2.1 Software Engineering concepts

This Section describes the main concepts covered in this thesis, which will be the subject of study. First, the notion of *development issue* will be defined in Section 2.1.1, and similarly, *development velocity* will be explained in Section 2.1.2.

2.1.1 Development issue

The concept *issue* in software development can refer to different types of procedures or actions. In fact, according to [4] issues in software development can depict a bug, a task within a project as well as a leave request form, and more. In this sense, issues resemble the different types of blocks or actions that construct a project.

As an analogy, given a software development project, the same developers of the product service are the subjects that report an issue that has to be fixed or accomplished, as mentioned before this could be a bug in the code or a new feature task. The newly created issue is reported to one or multiple assignees that must fix or accomplish the task.

In this way, in software development projects as well as in another type of organizations, tasks are configured as issue blocks that can vary based on their issue type. Still, the purpose of creating different types of issues remains the same, creating an organized system of blocks that build a project in a controlled manner.

To follow the aforementioned working system, projects adopt different software to control their work with issues so that in the case of software development projects *issue-tracking* helps teams manage their code, estimate the workload and keep track of the existing reported issue [4].

For instance, projects in *Apache Software Foundation* (ASF) have adopted software such as GitHub and Jira as their issue-tracking systems.

2.1.2 Development velocity

In SE, development velocity denotes the speed in which teams accomplish tasks such as feature development, testing, software release, or issue fixing [5], i.e., how productive teams are in accomplishing different types of tasks. Hence, the faster and more efficient projects are in accomplishing tasks, the higher is their velocity level [6]. Similarly, development velocity can be influenced by different factors such as developers' skills, the complexity of the tasks as well as the development tools used in the team to improve their development velocity [3].

Software development projects aim to keep their development velocity high in order to maintain a fluid production of new features to clients. To achieve this, among the development tools used by teams, SATs, such as the already mentioned SQ, have an important role in keeping the quality of the code as high as possible and, therefore, help developers fix issues at a higher speed. Therefore, the development velocity stands as a key response variable describing the impact of development tools like SQ.

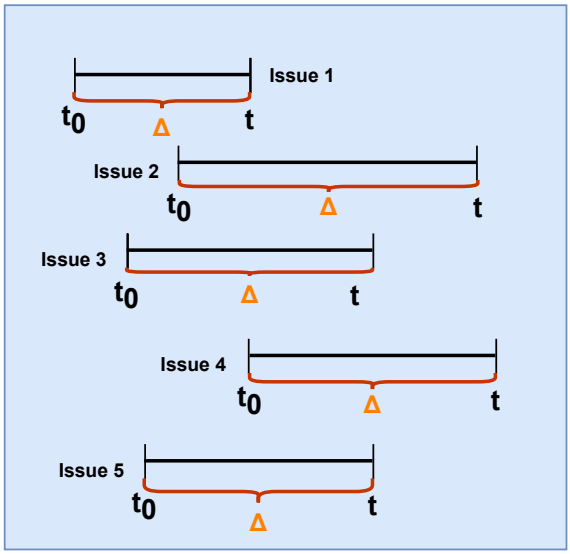


Figure 2.1. *Issue velocity calculation within a fixed time period.*

The approach adopted for calculating velocity in this thesis is based on the calculation of the average velocity of a set of reported issues within a limited amount of period of time. Figure 2.1 graphically explains the mentioned approach for a set of issues. Similarly, Equation 2.1 explains the calculation as follows:

$$\textit{Velocity average} = \frac{\sum_{i=1}^{n}(\textit{End time} - \textit{Start time})}{n} \tag{2.1}$$

where *n* is the number of issues reported during the mentioned calculation windows.

2.2 SonarQube

SonarQube (SQ) is one of the most used open-source Static Analysis Tools (SATs) in SE nowadays. Provided by `sonarcloud.io` [7], SQ can be used on a private server by downloading it or as a *software-as-a-service* (SaaS) too.

SQ is widely known for the set of metrics it offers to software development teams, such as the number of lines or the complexity in the code among others [2]. Furthermore, it defines a set of rules that determine coding standards, thus, they help developers orientating them on the proper coding practices. In this sense, software development teams who desire to understand the quality of their code, analyze it with SQ which generates different types of issues when the mentioned rules are broken in the submitted code.

In relation to the definition of the development issue in Section 2.1.1, issues reported by SQ might denote current *bugs* in the code that can already impact the functionality of the program, same as warning about *code smells* that denote possible future functionality problems due to the analyzed code [8]. This warning induces developers into a process called *issue-fixing*, and often requires more time than the one spent developing new features in the program [8]. It is here where the importance of issue-trackers mentioned in Section 2.1.1 is highlighted since software development teams might start tracking some of the issues reported by SQ in issue-trackers.

Given the current massive use of SonarQube in the SE field, different studies have been conducted lately to analyze the effect of SQ on the efficiency of development teams [2, 8, 9, 10]. However, this thesis does not concentrate on the impact of different levels of issues, but rather on the global impact of SQ on the development velocity of a software project, this latter one being calculated based on development issues reported in the already mentioned issue-trackers.

2.2.1 Apache Software Foundation

As mentioned initially in Chapter 1, in order to study the effect of SonarQube on the development velocity of software development projects, this thesis considers existing mature software development projects from the *Apache Software Foundation* (ASF).

ASF is a well-known organization that provides services and support for software development communities that choose to follow the *Apache way* or standards [11]. The foundation is formed by active and independent open-source projects, but it classifies projects into three different statutes. Projects that fulfil the criteria of the ASF policy are called *Mature* and are the subject of study in this thesis. Meanwhile, projects that are be-

ing helped by the ASF in order to get promoted to be Mature are considered to be in the *Incubator*. And finally, projects that no longer fulfil the requirements of ASF are relegated to the *Attic*, and archived by the ASF.

In addition, ASF has a predefined policy that every project in the foundation must strictly follow in order to be a mature project [11]. Among the criteria specified in the mentioned policy, projects must store their data in GitHub repositories under the organization name of Apache since this software is the official repository of the foundation. Similarly, ASF officially uses Jira, GitHub and Bugzilla as their official issue trackers, which only the first two are considered for the study in this thesis.

As a matter of fact, majority of all ASF projects contain multiple repositories so that each of them covers a different functionality of the software service the project is offering. Thence, each repository has a linked issue-tracker to track the flow of issues for that specific functionality, as well as SQ or other software tools likewise.

In fact, issue-trackers and software tools are deployed in a *1vs1* relationship with each repository of a project. Hence, each project containing multiple repositories can contemplate the use of different software tools by teams depending on the given repository. Due to this fact, this thesis considers repositories containing enough data from ASF mature projects as eligible singular projects, since not all repositories have the same or enough activity to be considered.

In relation to the aforementioned importance gained by SATs like SQ in recent years, multiple projects in ASF have adopted SQ and, thus, the comparison among ASF projects using and not using SQ stands as a potential case of analysis.

2.3 Cohort studies

This section covers the theoretical background of *cohort studies*, a type of study which is widely applied in research fields such as epidemiology. Cohort studies study the effect of a specific exposure on a given population to explain the outcome of interest [12].

To accomplish that, first an observational time period is proposed in which some of the subjects in a population are naturally exposed to the effect variable that is being studied, i.e., the *independent variable*. Then, two observational windows are defined within the defined observational period time, one for the initial collection and calculation of the independent variable and possible *confounders*, and a second one after a predefined follow-up period where the outcome or *dependent variable* is measured. Figure 2.2 graphically explains the comparison between exposed population and non-exposed population in a cohort study.

Cohort studies aim at choosing sample populations where subjects are similar in almost

all the considered variables but on the independent variable. Therefore, this type of study becomes useful when comparing two populations in order to explain the impact of a considered factor, SQ in the case of this thesis.

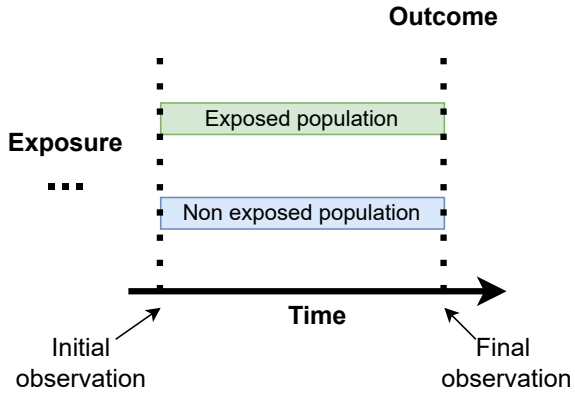


Figure 2.2. *Schema of a cohort study.*

For this thesis, the retrospective cohort study is chosen in order to construct the data. Retrospective types of studies are performed at the present time but look into the past to analyse possible effects explaining the outcome of interest. A graphical representation of the logic behind retrospective studies is given in Figure 2.3 below.

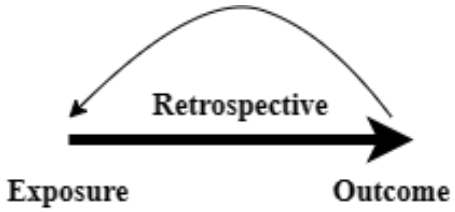


Figure 2.3. *Retrospective studies return in time to the moment where the exposure was made in order to study its effect on the current outcome.*

As mentioned above, the robustness of cohort studies stands on the principle of the relative similarity among the group in the population that is under the effect of exposure and the group which is not. Thus, the comparison offers strong validity and gives a margin to analyze in depth the effect of the exposure on the outcome of the different groups, as well as the effect of possible potential confounders [12].

It is important to mention that even though the exposure is received by a part of the considered population, it is the entire population that obtains the outcome, the development velocity at the end of the analysis period. It is in this way that the cohort study stands as a suitable study method to analyze the effect of a given exposure.

3. REGRESSION ANALYSIS

This chapter covers the considered regression models and assumed distributions within this thesis, as well as their theoretical background and selection criteria. The chapter begins on Section 3.1 with the initial basic assumptions to be considered when implementing regression analysis on a study, as such, *Multiple Linear Models* (LM) and how the model estimation is performed through *Ordinary Least Squared* (OLS). Based on the assumption of non-normality within the response variable, Section 3.2 covers the *Generalized Linear Models* (GLM) and how the model estimation is achieved through *Maximum Likelihood* (ML). The following Section 3.3 describes the main theoretical details of the distributions considered for the regression analysis of the response variable in this thesis. The second Section 3.4 considers the application of *ensemble learning* through Random Forest, where regression trees are combined to observe the effect of agglomerating regression. Section 3.5 introduces the so-called *Backward selection* process to manage regression models in order to perform variable selection. And finally, the last section of this chapter describes the information criteria and model selection methods considered in this thesis.

3.1 Linear Models

Multivariable Linear Models (LM) present a regression analysis that through linearity describe the existing relationship between a response variable and a set of explanatory variables [13]. For a given number of dependent variables $\boldsymbol{y} = (y_1, ..., y_n)^T$ where *n* is the size of the population, and similarly $\boldsymbol{\mu} = (\mu_1, ..., \mu_n)^T$ where $\mu = E(y_i)$, an LM can be described in the following way. Given a covariance matrix:

$$\boldsymbol{V} = var(\boldsymbol{y}) = E[(\boldsymbol{y} - \boldsymbol{\mu})(\boldsymbol{y} - \boldsymbol{\mu})^T] \tag{3.1}$$

And defining the model matrix as $\boldsymbol{X} = (x_{ij})$ with $n \times p$ dimensions where x_{ij} is the explanatory variable j at observation i, model fitting can be performed based on

$$\boldsymbol{\mu} = \boldsymbol{X}\boldsymbol{\beta} \qquad \text{and} \qquad \boldsymbol{V} = \sigma^2 \boldsymbol{I} \tag{3.2}$$

We consider $\boldsymbol{\beta}$ as the $p \times 1$ parameter vector where $p \leq n$ and $\boldsymbol{I}$ as $n \times n$ identity matrix, thus making the variance-covariance matrix a diagonal matrix of σ^2. In this way the *multiple linear model* would be described as follows:

$$\boldsymbol{X} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{p1} \\ \vdots & \vdots & & \vdots \\ 1 & x_{1n} & \cdots & x_{pn} \end{pmatrix} \qquad \boldsymbol{y} = \begin{pmatrix} y_1 \\ . \\ . \\ . \\ y_n \end{pmatrix} \qquad \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix} \qquad \boldsymbol{\epsilon} = \begin{pmatrix} \epsilon_1 \\ \vdots \\ \epsilon_n \end{pmatrix}.$$

According to [13], by assuming normal distribution in $\boldsymbol{y}$ the model becomes the *normal linear model* (LM), which similarly can be described as a generalized linear model (GLM) with identity link function. This and more options offered by GLMs are discussed in later Section 3.2. A more widely used expression of the **normal** LM in matrix notation can be

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{3.3}$$

where the expected value of error $\boldsymbol{\epsilon}$ is $E(\boldsymbol{\epsilon}) = 0$ and the variance-covariance matrix is defined as in Equation (3.2) [14].

Classically, the most commonly used method to obtain the best linear *estimates* is the *least squares* method, through which parameter estimates $\hat{\boldsymbol{\beta}}$, and thus *fitted values* $\hat{\boldsymbol{\mu}} = \boldsymbol{X}\hat{\boldsymbol{\beta}}$ are obtained.

3.1.1 Least Squares Estimation

As mentioned, the best estimate of $\hat{\boldsymbol{\mu}}$ performs the minimization

$$||\boldsymbol{y} - \hat{\boldsymbol{\mu}}||^2 = \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} \hat{\beta}_j x_{ij} \right)^2 \tag{3.4}$$

Furthermore, when the normality assumption is added to the model (in this thesis through normalization of the data as presented later) least squares yield maximum likelihood [13] by $L(\boldsymbol{\beta}) = \sum_i (y_i - \mu_i)^2 = \sum_i (y_i - \sum_j \beta_j x_{ij})^2$. Agresti has proven the solution for these better called *Likelihood equations* in [13] as follows:

$$\begin{aligned} \boldsymbol{L(\beta) = ||y - X\beta||^2 = (y - X\beta)^T(y - X\beta) =} \\ \boldsymbol{y^T y - 2y^T X\beta + \beta^T X^T X\beta} \end{aligned} \tag{3.5}$$

By deriving the matrix minimization on 3.5

$$\boldsymbol{\frac{\partial L(\beta}{\partial \beta} = -2X^T(y - X\beta)} \tag{3.6}$$

Hence, by becoming $\boldsymbol{\beta}$ into $\boldsymbol{\hat{\beta}}$

$$\boldsymbol{X^T y = X^T X\hat{\beta}} \tag{3.7}$$

We obtain the least squares estimator

$$\boldsymbol{\hat{\beta} = (X^T X)^{-1} X^T y} \tag{3.8}$$

which yields the fittest predictions of the regression's random variable given the values from the explanatory variables. The minimization problem seen before presents a mathematically correct procedure to build a prediction model, still, normality cannot be yielded so easily which leads this thesis into the next chapter.

3.2 Generalized Linear Models

Linear Models explain the relationship of the random component and a set of explanatory variables through linearity, and assuming that the distribution of the former one follows normality. In this thesis, since the analysis approach for the concrete research field has not been standardized yet, we do not consider only such assumption but pursue to perform linear regression under non-normality by using *Generalized Linear Models* (GLMs). McCullagh and Nelder define in [15] the three components that describe GLMs, these are the *random component*, the *linear predictor* and the *link function*.

The former one represents the independent variable of the linear regression, for instance, it could be defined as $\boldsymbol{y} = (y_1, ..., y_n)^T$, this one being distributed in the *exponential family of distributions* (see [15] for the complete mathematical definition of the exponential family). Similarly, the *linear predictor* could be described as the connection between the explanatory variables and the *expected value* of y_i, in fact, as A. Agresti defines in [13] "The *linear predictor* of a GLM relates parameters η_i pertaining to $E(y_i)$ to the explanatory variables $x_1, \ldots, x_p$ using a linear combination of them"

$$\eta_i = \sum_{j=1}^{p} \beta_j x_{ij}, i = 1, \ldots, n. \tag{3.9}$$

And finally, the *link function* performs the connection between the response and the linear predictor $\eta_i = g(\mu_i)$ through a differentiable function

$$g(\mu_i) = \sum_{j=1}^{p} \beta_j x_{ij}, i = 1, \ldots, n. \tag{3.10}$$

Function $g(\cdot)$ can perform different transformations based on the chosen distribution family and link function. It is in this stage when the accomplishment of the considered assumptions must be tested through the observation of the explanatory variables to be included in the model.

3.2.1 Maximum Likelihood Estimation

By far one of the most well-known model fitting methods in statistics, *Maximum Likelihood Estimation* (MLE) is based on the notion that, given the sample data, the most accurate model parameters should be found so that the sample's response parameter values are most likely generated from them [16].

For instance, if we consider that the response variable $\boldsymbol{y}$ belongs to the exponential family distribution, [13] describes the fitting procedure as follows, starting from the definition of the log-likelihood in GLMs

$$L(\boldsymbol{\beta}) = \sum_{i=1}^{n} L_i = \sum_{i=1}^{n} logf(y_i; \theta_i.\phi) = \sum_{i=1}^{n} \frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + \sum_{i=1}^{n} c(y_i, \phi) \tag{3.11}$$

The likelihood equations thus are obtained through the chain rule of differentiation with respect to the model parameter β_j,

$$\frac{\partial L_i}{\partial \beta_j} = \frac{\partial L_i}{\partial \theta_i}\frac{\partial \theta_i}{\partial \mu_i}\frac{\partial \mu_i}{\partial \eta_i}\frac{\partial \eta_i}{\partial \beta_j} \tag{3.12}$$

Since the mean and variance of the exponential family of a random component are $\mu_i = b'(\theta_i)$ and $var(y_i) = b''(\theta_i)a(\phi)$ by definition, $\partial L_i/\partial\theta_i = [y_i - b'(\theta_i)]/a(\theta_i)$, and since $\eta_i = \sum_{j=1}^{p} \beta_j x_{ij}$ hence $\partial\eta_i/\partial\beta_j = x_{ij}$ the differentiation is summarized into

$$\boldsymbol{u} = \frac{\partial L(\boldsymbol{\beta})}{\partial \beta_j} = \sum_{i=1}^{n} \frac{(y_i - \mu_i)x_{ij}}{var(y_i)}\frac{\partial \mu_i}{\partial \eta_i} = 0, j = 1, 2, \ldots, p. \tag{3.13}$$

The MLE $\hat{\theta}$ must satisfy the likelihood equations for the GLM

$$\boldsymbol{u = X^T DV^{-1}(y - \mu) = 0.} \tag{3.14}$$

where β is implicitly inside μ and

$$\boldsymbol{X} = \begin{pmatrix} \frac{\partial \mu_1}{\partial \eta_1} & 0 & \cdots & 0 \\ 0 & \frac{\partial \mu_2}{\partial \eta_2} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \frac{\partial \mu_n}{\partial \eta_n} \end{pmatrix}, \quad \boldsymbol{V} = \begin{pmatrix} Var(Y_1) & 0 & \cdots & 0 \\ 0 & Var(Y_2) & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & Var(Y_n) \end{pmatrix}.$$

By nature, the likelihood equations are nonlinear functions of β that are used to obtain the maximum of the log-likelihood through the Taylor series expansion

$$L(\boldsymbol{\beta}) \approx L(\boldsymbol{\beta})^{(t)} \boldsymbol{u}^{(t)T}(\boldsymbol{\beta} - \boldsymbol{\beta}^{(t)}) + \left(\frac{1}{2}\right)(\boldsymbol{\beta} - \boldsymbol{\beta}^{(t)})^T \boldsymbol{H}^{(t)}(\boldsymbol{\beta} - \boldsymbol{\beta}^{(t)}) \tag{3.15}$$

where $\boldsymbol{H}$ resembles the *Hessian matrix*.

One well-known method to solve the system of nonlinear equations is the *Newton-Raphson method* which iteratively seeks for the maximum point.

$$\frac{\partial L(\beta)}{\beta} \approx u_t + H_t(\beta - \beta_t) = 0,$$
$$\boldsymbol{\beta_{t+1} = \beta_t - H_t^{-1} u_t} \tag{3.16}$$

where

$$\boldsymbol{H} = \left(\frac{\partial^2 L(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta'}}\right)$$

Alternatively, the *Fisher Scoring method* can be used for the same purpose, since the unique difference in comparison with 3.16 comes from the use of the expected value of the Hessian matrix instead of the matrix itself.

$$\boldsymbol{\beta_{t+1} = \beta_t + F_t^{-1} u_t} \tag{3.17}$$

where

$$\boldsymbol{F} = -E(\boldsymbol{H})$$

Similarly, if we consider approximation on GLMs, under standard regularity conditions for a large sample size the model parameters follow an approximate normal distribution [17]

$$\hat{\boldsymbol{\beta}} \sim N\Big(\boldsymbol{\beta}, (\mathbf{X}'\hat{\mathbf{W}}\mathbf{X})^{-1}\Big), \tag{3.18}$$

where $\hat{(}\boldsymbol{W})$ is the estimated $\boldsymbol{W}$ for $\hat{\boldsymbol{\beta}}$ defined as

$$\boldsymbol{W} = \begin{pmatrix} \frac{(\frac{\partial \mu_1}{\partial \eta_1})^2}{Var(Y_1)} & 0 & \cdots & 0 \\ 0 & \frac{(\frac{\partial \mu_2}{\partial \eta_2})^2}{Var(Y_2)} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \frac{(\frac{\partial \mu_n}{\partial \eta_n})^2}{Var(Y_n)} \end{pmatrix}$$

3.3 Distributional assumptions

This section covers the description of the distributions considered for the regression analysis and their theoretical background given the nature of the response variable. To begin with, Gaussian (or Normal) distribution is explained in Section 3.3.1 as an initial distributional assumption after denoting that the response variable is continuously distributed. This and further assumptions based on the data covered in Chapter 5 transferred the attention into other distributional assumptions fitting the nature of continuous distributions. Thus, Section 3.3.2 and Section 3.3.3 describe further additional distributions for statistical modelling in this thesis, Gamma distribution and Inverse Gaussian distribution.

3.3.1 Gaussian distribution

As an initial alternative to the traditional way to model data, that is, transforming *y* in order to reach normality through approximation, considering the assumption under which *y* is distributed by the *exponential family* is covered in this section [13]. Additionally, given the continuous nature of *y*, *Gaussian distribution* fits as the best initial assumption to start modelling with GLMs. The Gaussian or *Normal* consists of two main parameters μ and σ^2 as its mean and variance and the *probability density distribution* (pdf) representation [18] is presented as follows

$$f(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma}} e^{-(x-\mu)^2/(2\sigma^2)}, \qquad -\infty < x < \infty \tag{3.19}$$

In practice, the possible link functions $g(\mu_i)$ for the Gaussian distribution are

$$\begin{aligned} \mu_i &= \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}, && \textit{identity link} \\ log(\mu_i) &= \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}, && \textit{log link} \\ \frac{1}{\mu_i} &= \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}, && \textit{inverse link} \end{aligned}$$

3.3.2 Gamma distribution

This section of the thesis emerges as a non-predefined step within the analysis since it is due to the characteristics of the residuals from the random component that the study case considers another option than the assumption of normality. Further details are given in Chapter 5.

The gamma distribution or *gamma family of distributions* has been always related to variable distributions denoting skewness, as well as non-negativity $[0, \infty)$ within the values of the random component [18]. In fact, given the pdf of the gamma distribution

$$f(x|\alpha,\beta) = \frac{1}{\Gamma(\alpha)\beta^\alpha} x^{\alpha-1} e^{-x/\beta}, \qquad 0 < x < \infty, \qquad \alpha > 0, \qquad \beta > 0 \tag{3.20}$$

$$E(f(x|\alpha,\beta)) = \frac{\alpha}{\beta} = \mu_i, \qquad Var(f(x|\alpha,\beta)) = \frac{\alpha}{\beta^2} = \phi\mu^2$$

where α is the shape parameter describing the peakedness of the distribution, β is the scale parameter which defines the spread of the distribution and $\phi = \alpha^{-1}$ (see [18] for formal derivations of the gamma family of distributions).

Based on the properties of the distribution for y_i that are covered in Chapter 5, that is, variance depends on the mean due to the fact that both increase proportionally to rate μ_i^2, gamma distribution appears as an alternative GLM approach [13] for the analysis within this thesis.

Under Gamma distribution, possible link functions $g(\mu_i)$ are

$$\mu_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}, \qquad \textit{identity link}$$
$$log(\mu_i) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}, \qquad \textit{log link}$$
$$\frac{1}{\mu_i} = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}, \qquad \textit{inverse link}$$

3.3.3 Inverse Gaussian distribution

In this subsection, a direct competitor distribution for the Gamma distribution is presented. In fact, similarly to the distribution described in the previous subsection, the *Inverse Gaussian* distribution is suitable for modelling situations in which the response variable can only have non-negative values $[0, \infty)$, and the variance increases proportionally to rate μ_i^3 [13]. The pdf can be described by

$$f(y_i|\mu, \gamma) = \sqrt{\frac{\gamma}{2\pi y_i^3}} exp\left(\frac{-\gamma(y_i - \mu_i)^2}{2\mu_i^2 y_i}\right), \qquad y_i > 0 \qquad (3.21)$$

$$E(f(y_i|\alpha, \gamma)) = \mu_i, \qquad Var(f(y_i|\alpha, \gamma)) = \phi\mu_i^3,$$

where $\phi = \gamma^{-1}$.

Under Inverse Gaussian distributions, the possible options as link function $g(\mu_i)$ are

$$\mu_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}, \qquad \textit{identity link}$$
$$log(\mu_i) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}, \qquad \textit{log link}$$
$$\frac{1}{\mu_i} = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}, \qquad \textit{inverse link}$$
$$\frac{1}{\mu_i^2} = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}, \qquad \textit{canonical link}$$

3.4 Using Random Forests

This section introduces the additional approach of applying regression through the concept of *Ensemble Learning* concretely with *Random Forests* (RF) and its theoretical background. This approach is supported by the library `randomForest` which is offered as an algorithm-based random forests of trees for classification and regression [19].

To begin with, the concept of ensemble learning describes the idea of combining several different regressors, that even though they may perform acceptably alone, assessing their combined regression results through majority voting ensures better accuracy [20]. In the case of this thesis, ensemble learning is applied through regression trees, which can be fitted as follows:

```
> model <- randomForest(response ~ . , data = Data, type = 'regression')
```

After fitting the model, one way to assess its power can be through the *means squared error* which will be explained in detail in Section 3.6.3. For that, the library `randomForest` provides the following function to obtain the model estimates, and similarly, the function `summary` can offer details describing the building process of the random forest.

```
> estimates <- predict(model)
```

```
> summary(model)
```

Similarly, the performance of the random forest can be graphically assessed through the `plot` function, which denotes the diminishing prediction error as the number of regression trees is included, thus showing the effect of ensemble learning on the prediction power.

```
> plot(model)
```

In addition, to compare the significance of the considered variables within the model, the importance recorded through the regression can be numerically obtained and graphically plotted as follows:

```
> randomForest::importance(model)
```

```
> varImpPlot(model, main = "Variable importance in 'model'")
```

This subsection describes in a neat way the contributions that `randomForest` can offer to this thesis as an additional resource of comparison. However, this library offers further options on how to manually customize the shape and structure of the random forest, but they are not covered in this thesis. For further reading, the usage of the library in R [19] and the theoretical background behind the computed algorithms [20] are referred.

3.5 Backward variable selection

This section covers a commonly used approach among statisticians, the *backward variable selection* process. In fact, as its name defines, this process begins by considering a complex model with multiple components considered. Gradually, it starts to remove terms from the model based on different criteria although with a common objective, removing terms whose effect is negative towards the model. This process is constructed on the basis of model comparison through considered information criteria and model selection

techniques further explained in Section 3.6. The procedure finishes when additional term exclusions do not improve the model but weaken its fit [13].

A recommended format for model-building through the backward elimination procedure is shown by Hosmer et al. [21], better called *purposeful selection*. This approach considers at the initial stage of the process the main effects of those explanatory variables and potential confounders that are known to be important in the given field. Once the full set is built, the backward elimination process begins until the final set of variables is found. At this moment, researchers can consider possible interactions among model variables and test their significance based on considered significance tests for the study.

3.6 Information Criteria and model selection

In observational studies where regression is implemented, often researchers aim to find the best fitted combination of explanatory variables that better explain the outcome of the study. Hence, the model selection task stands as an important step within the analysis since under-fitting may not identify the complete nature of the variability in the response variable, as well as over-fitting the model may suppose generality loss in the problem [22]. In this thesis, the model selection was performed when the regression modelling had been applied based on the observed patterns from the data, and before performing hypothesis testing to see whether the relationship between the included explanatory variables could have significant changes in the chosen model. The selected information criteria and measurements have been based on the commonly used techniques observed in similar studies.

3.6.1 Akaike Information Criteria

The *Akaike Information Criteria* (AIC) assesses models by means of how close their fit can be from the true model fit. Furthermore, according to [13] given a population of interest, a simpler model may give a better fit than a complex model in which multiple variables are considered. This means that AIC penalizes models with a high number of parameters, thus aiding in variable selection too. The AIC can be determined as follows:

$$AIC = 2K - 2log(L(\hat{\theta}/y)) \tag{3.22}$$

where K means the parameters to be estimated and $log(L(\hat{\theta}/y))$ resembles the maximum of the log-likelihood of the given model [23]. Additionally, based on C.M. Hurvich and C.L. Tsai's work in [24] a refined estimate called $AICc$ can be further used when the sample data is small enough. It develops AIC in the following way:

$$AICc = AIC + \frac{2K(K+1)}{n-K-1} \tag{3.23}$$

where n is treated as the sample size, and the rest of the parameters perform the same role as explained above. This extension may offer no difference compared to AIC if n results into a large value. Thence, given a set of models under comparison, the model showing the lowest AIC or $AICc$ score stands as the best fitting the data.

3.6.2 Bayesian Information Criteria

The *Bayesian Information Criteria* (*BIC*) appears as an alternative to the AIC but remains similar to the formal representation of the latter statistic [25]

$$BIC = 2K - log(n)log(L(\hat{\theta}/y)) \tag{3.24}$$

In fact, compared to Formula 3.22, BIC replaces factor 2 by the logarithm of the sample size, thus penalizing the model for the number of the used model parameters. In this way, meanwhile, AIC fluctuates into more complex models, and BIC moves less rapidly in that direction [13].

3.6.3 Mean Squared Error criteria

The means squared error (MSE) criteria is one of the most widely used predictive evaluation measurements in fields with observed data [26]. In this thesis, we will focus on the definition describing a predictor:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{3.25}$$

Sometimes finding the *best linear unbiased predictor* (BLUP) with small variance can be difficult, because of this rather than seeking a BLUP researchers prefer to choose predictors providing the smallest MSE [27]. Based on the outcome range of the MSE $[0, +\infty)$, the model with the smallest quadratic error is the best one.

4. DATA

This chapter describes the specific procedure followed to obtain the data set used for the conducted regression analysis. This process is formed by different steps that are covered in the existing section within the current chapter. To begin with, Section 4.1 describes the criteria followed to conduct the data collection process based on the principles of cohort studies covered in Section 2.3. Then, the description of the data mining process and its procedure are covered in Section 4.2. Next, Section 4.3 covers the data-cleaning stage of the collection process and defines the considered pruning decisions. Finally, with the data mining process being conducted based on the defined study design, and once it is clean, Section 4.4 describes the steps followed to obtain a data set containing the variables' data needed for the regression analysis.

The data collection process in this thesis is conducted using Jupyter Notebook and Python scripting environments, as well as Python the unique programming language. At the end of the data collection process, the final data set contains information from 337 open-source repositories, which are composed of 52 repositories using SQ (15%), that is the *treatment* cases, and 285 not using SQ (85%), or in other words the *control* cases of the study. In fact, as mentioned in Section 2.2.1 ASF projects contain multiple repositories defining their different functionalities. As mentioned in Section 2.2.1, this thesis considers inspecting every single repository as a unit project since SQ similarly performs analyses of single repositories rather to complete projects, making the service a *1vs1* relationship.

4.1 Observational design

According to the background on cohort studies considered in Section 2.3, the observational design is defined as an application of a cohort study based on the own goals of the thesis. To begin with, an observational period of two years is considered, starting from the beginning of 2020 until the end of 2022, with the aim to give projects that may have started using SonarQube before 2020 [7] enough time to familiarize to the software tool and ensure that it does have an impact on project's velocity.

Yet another time, the logic behind the considered observational design is to measure the velocity of projects at an initial time period in which the impact of the exposure of interest is not affecting the outcome, that is, the development velocity. In the same way, other

variables that may have impact in the outcome will be measured at the initial time period so that there are no omitted factors in the analysis. Likewise, after a follow-up period that is understood to give a software tool enough time to have impact on the development velocity, a second time period is considered to measure the outcome a second time. Thus, possible anomalies in the outcome values can be linked to if not the independent variable of the study, then the additional possible confounder variables.

To understand the time setup of the observational design, a graphical representation is offered in Figure 4.1

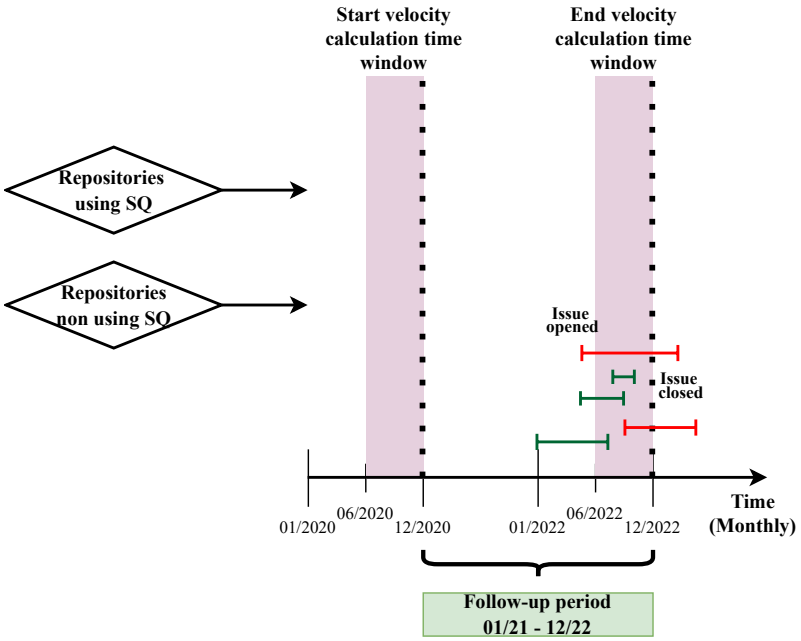


Figure 4.1. *Graphical schema of the observational design for the data collection.*

Similarly, in order to investigate the effect of SQ during the defined observational period, two time windows of exactly 6 months are defined to collect data about the considered independent variable and potential confounders. Furthermore, two years are considered as the follow-up period in order to give time margin to projects to use SQ. A graphical illustration of the mentioned procedure can be seen in Figure 4.1 and Figure 4.2. Additionally, while development velocity is calculated at the first time window and at the last window based on the comparison goal of the thesis, data about confounders are collected at the beginning of the follow-up period depicted in Figure 4.1 with the aim of capturing the shape of the project before the exposure of SQ has any impact on any confounder.

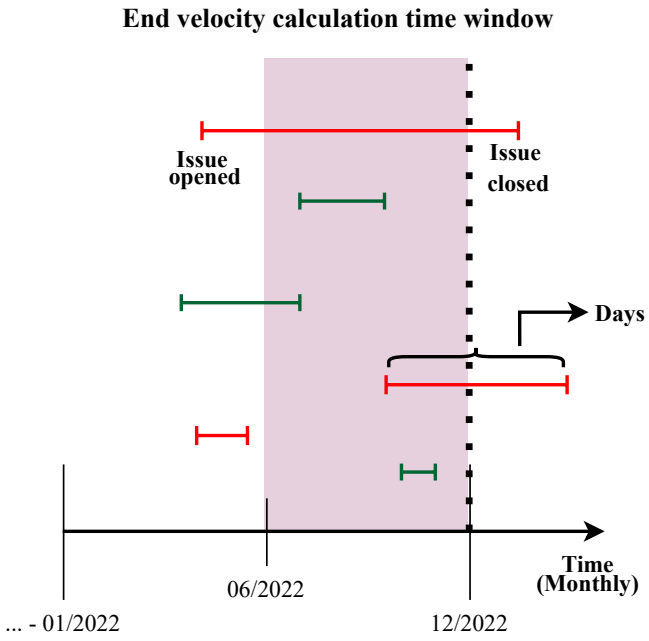


Figure 4.2. *Velocity calculation time window schema.*

Figure 4.2 offers a graphical view of the *modus operandi* for the velocity calculation in this study design. Only issues reported to be closed during the time window are considered in order to calculate the mean velocity measurement per each project.

The goal of the described observational design is to collect the data about the eligible projects in such a way that the considered variables act as potential factors to describe the changes registered in projects at the end of the follow-up period. In order to accomplish this, the following sections describe in detail the procedure to collect and process the data with the goal of obtaining a data set capable of offering the required characteristics in the covered observational design.

A graphical representation of the result of the data collection process is represented in Figure 4.3.

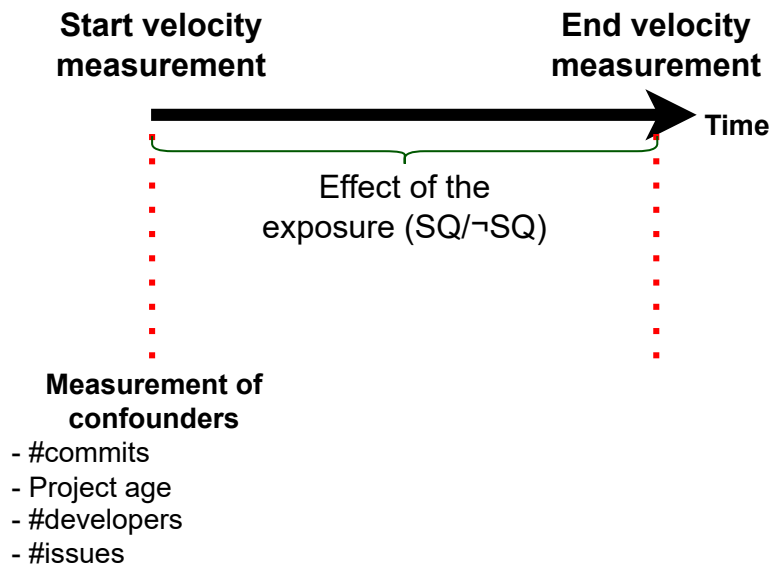


Figure 4.3. *Graphical summary of the observational design setup.*

4.2 Data Mining

The analysis subjects in this thesis are official software projects from Apache Software Foundation (ASF) [11] and only the projects listed on their website when the first collection of project names was performed on 07/02/2023 are considered in the study. In fact, ASF is an active association that is in constant development, so active projects can vary eventually. The first exclusion criterion is the status of the project. In ASF the projects are divided into three groups; Mature, Attic (retired or archived) and Incubator (not yet mature). From this classification 41 projects are in the Incubator, 51 in the Attic and 291 are identified as Mature. Likewise, these categories act as filtering stages in the incoming procedures as depicted in Figure 4.4.

Variable	Source	Description
Project full name	GitHub	Source name used in the project repository
Language	GitHub	Main programming language included in the repository
#commitsBeforeFollowup	GitHub	Number of commits performed before 31/12/2020
#commitsDuringFollowup	GitHub	Number of commits performed after 31/12/2020 and before 31/12/22
Age	GitHub	Age of the project repository measure from the first commit (Days)
#issuesBeforeFollowup	GitHub & Jira	Number of issues performed before 31/12/2020
#issuesDuringFollowup	GitHub & Jira	Number of issues performed after 31/12/2020 and before 31/12/22
SQ/nonSQ	SonarCloud	Boolean variable denoting if the project was identified in SonarCloud (Using SonarQube)

Table 4.1. *Variables collected within the data mining process used in the analysis.*

The collection procedure starts with collecting data from the GitHub [28] repositories under ASF ownership through its API. In this initial step, a total number of 2,375 repositories were identified. The variables aimed to collect are presented in Table 4.2. These variables denote the potential confounders that based on Software Engineering expertise, can affect the development process of a team, hence the development velocity as well.

Thus, these variables are considered in the regression analysis as explanatory variables and later examined to test their explanatory significance.

The next step in the data mining process is to search for information over GitHub's and Jira's ASF-owned repositories, in order to collect data related to the projects' issues along with their registered starting and closing dates so as to calculate the velocity from each reported issue. In this case, while the same total number of GitHub repositories initially identified are similarly identified in GitHub for their respective reported issues, only 483 repositories are identified in Jira through its' API [29]. And thus, before the data cleaning is performed, data from 2,858 repositories regarding issue activity are mined from GitHub and Jira in total.

4.3 Data Cleaning

The next step after performing the data collection process covers the cleaning of cases that do not fit in the analysis, and because of this, as shown in Figure 4.4 the criteria considered is divided into different consecutive pruning decisions (I-V).

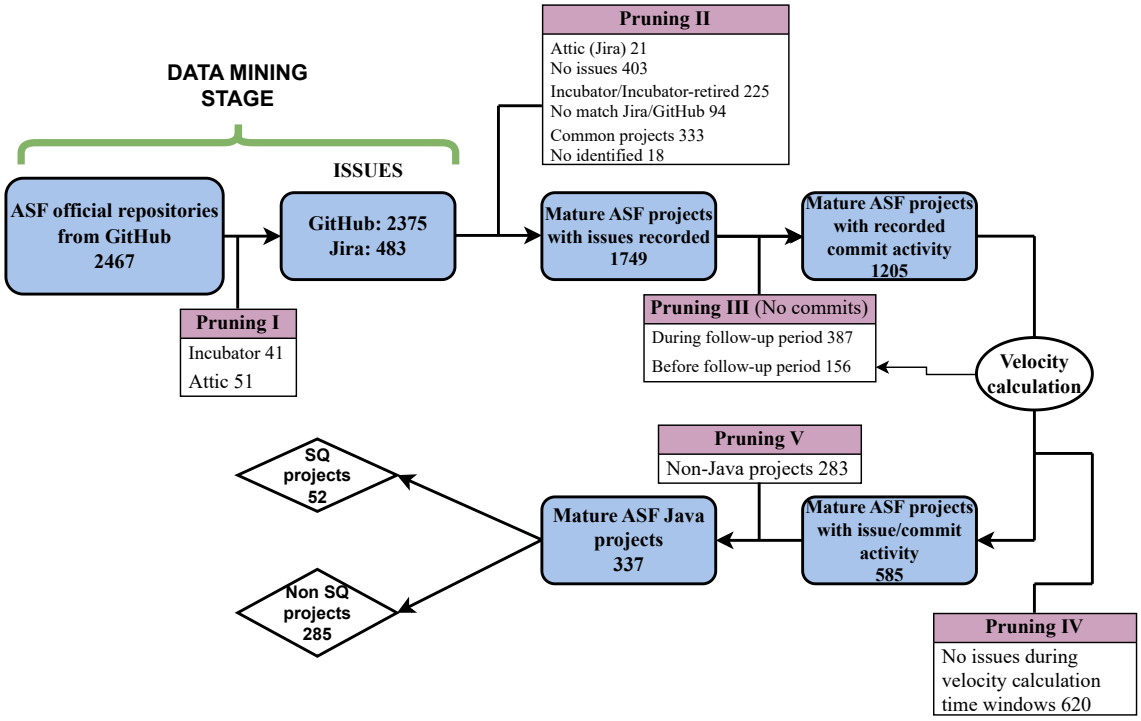


Figure 4.4. *Data collection pipeline.*

Following the graphical description from Figure 4.4 the initial pruning decision (I) is already performed during the data mining stage since it excludes directly any repository that existed in the projects inside the Incubator and the Attic groups.

Consequently, the second (II) pruning decision is defined in order to combine registered issues from the same project that exist in repositories from GitHub and Jira since some

projects are identified to have activity in both issue-tracking systems. Hence, these repositories are merged reducing the total number of existing repositories by the number of merged ones (333). At the same time, projects from Jira that are newly identified as retired (Attic) (21) and projects whose repository name includes the suffix "incubator" are excluded as well (225).

As an additional pruning step within the second pruning decision, project repositories crawled from Jira but that do not match their counterpart in GitHub are excluded too (94), in fact, since the commit information is to be considered as a variable confounder, projects with no commit information cannot be used in the analysis. And, as mentioned before, ASF projects only perform version control in GitHub based on their policy. Similarly, those Jira projects that are not identified with any project in GitHub nor in the ASF list of projects are excluded as considered unidentified (18).

Finally, as the core part of the second pruning decision, project repositories with no registered issues during the follow-up period whereas repositories with no issue activity before the follow-up period are excluded (403). Indeed, as the core part of the analysis is based on the effect of SQ on the issue velocity, projects without information regarding issue activity are ineffective for the purpose of the analysis, making the number of suitable projects at this stage to be 1,748.

However, the pruning process does not end here since there are still two groups of the same number of repositories concerning different data, one of them based on commit data and the other one on issue data. And, since the goal of the data collection process is to generate a single data file, the third pruning decision (III) stands as a merge process in which 156 repositories result not having commit-data before the follow-up period and other 387 present the same diagnostic during the follow-up period. In addition to the third pruning decision, at this point the velocity calculation is executed based on the specifications given in Section 2.1.2. This results in a lack of existing valid issues during the velocity calculation time window on 620 projects. Hence, this robust pruning decision (IV) leaves the size of the analysis population in 585 project repositories.

The final pruning decision (V) of the data cleaning process resembles a major problem for the analysis. After performing the presented pruning stages, the population using SQ shows a common pattern: using Java as the main programming language. In front of this, for the sake of simplicity and since projects' workflow might differ based on the programming language they use, it is decided to perform the analysis only on ASF projects whose main programming language is Java (337). In fact, if this pruning would not be performed, the identified confounder variable *language* would not be comparable among treatment and control cases.

Finally, after the complete cleaning process is performed, a single data file is defined in which 52 ASF Java projects report using SQ (15%) and 285 do not use it (85%).

4.4 Data Preprocessing

The mining process collects unstructured data from the mentioned version control repositories and issue-tracking systems. It is in this section the needed preprocessing of the data is explained to perform the data analysis. The result of the complete data collection process can be observed in Table 4.2.

Variable	Status	Type	Description
Velocity end	Dependent	Continuous	Mean value of the registered velocity at the end calculation window.
SQ/nonSQ	Independent	Categorical	The project has adopted SQ or not.
Velocity start	Confounder	Continuous	Mean value of the registered velocity at the start calculation window.
#commits	Confounder	Continuous	Number of commits at the beginning of the follow-up period.
Project age	Confounder	Continuous	Age in days of the project at the beginning of the follow-up period.
#developers	Confounder	Continuous	Number of developers participating at the beginning of the follow-up period.
#issues	Confounder	Continuous	Number of existing issues at the beginning of the follow-up period.

Table 4.2. *Final list of variables used in the analysis.*

Initially, the data from time variables such as velocity measurements and project age comes in `DateTime` format, which for simplicity in the regression analysis is converted into continuous decimal format.

It is at this point that, due to the difference in time size among different ASF projects, multiple suitable projects present values extremely close to zero while others present high values, making the data set unbalanced. Due to this, it is considered to uniformly scale the data set continuous variables through *min-max scaling* as follows

$$x' = \frac{x - min(x)}{max(x) - min(x)}$$

In addition, since by definition all the data measurements are non-negative and non-zero,and since some observations present 0 values after scaling, it is decided to manipulate all projects' data in the analysis by an agreed 0.001 decimal modification. In fact, even if by definition the considered variables are non-negative, values presenting 0 due to the variable type transformation appear to collapse with the distributional assumptions mentioned in Section 3.3 while performing the regression analysis.

5. RESULTS

In this chapter the results of the analysis performed in this thesis are presented. In contrast with the environment used in the data collection, for the analysis version 4.1.1 of the object-oriented programming language R was used in R Studio, a specific text editor environment dedicated to statistical computations. The first Section 5.1 of this chapter describes the nature of the data through an exploratory analysis. Next, Sections 5.2 and 5.3 describe the backward selection criteria followed in LMs and GLMs. Finally Section 5.4 declares the results of model comparison among the best-identified regression models. As an additional section, Section 5.5 shows the results from the agglomerating regression performed with random forests.

5.1 Exploratory analysis

Before assuming normality, the first step of the analysis was to perform an exploratory analysis. In fact, the goal of this stage was to understand in a better way which would be the best distributional assumption before knowing the results from the different analyzed regression models.

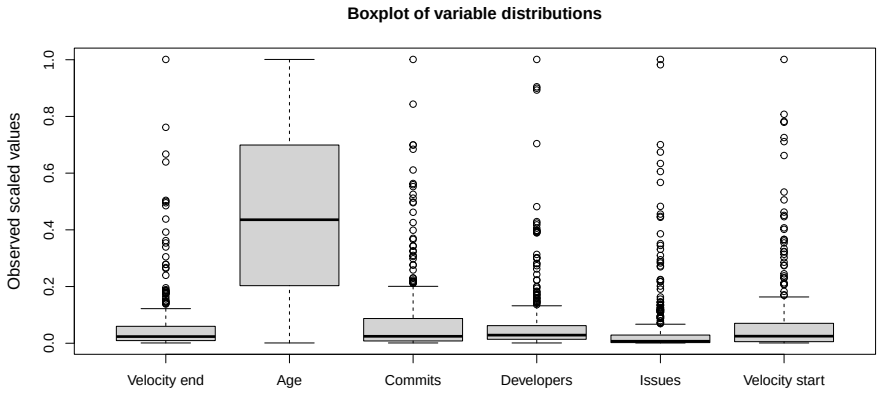


Figure 5.1. *Boxplot for exploratory analysis of scaled data.*

Graphical representation from the continuously distributed variables was displayed in boxplot format as can be seen in Figure 5.1. It was evident that a high number of subjects in the sample population was considered as outliers, which denoted a pattern for assuming

skewness in the distributions of the variables. In fact, this consideration was confirmed by the histograms displayed in Figure 5.2. Variable `Exposure` was not included in the mentioned plots as its binary nature would not show explanatory information for the current purpose.

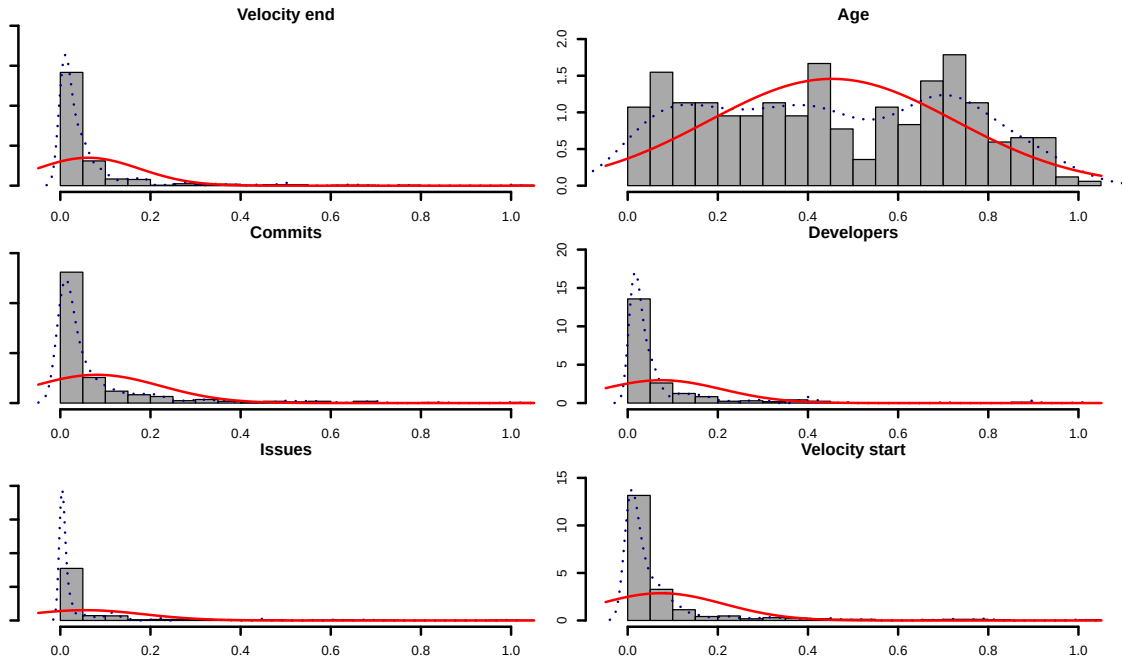


Figure 5.2. *Histograms of the scaled variables (Blue dotted line explains the normal fits and and red line density distributions of each variable.)*

The next considered step in the preliminary analysis was to observe the linear dependence among the considered variables. Figure 5.3 denoted low linear dependence in the sample population.

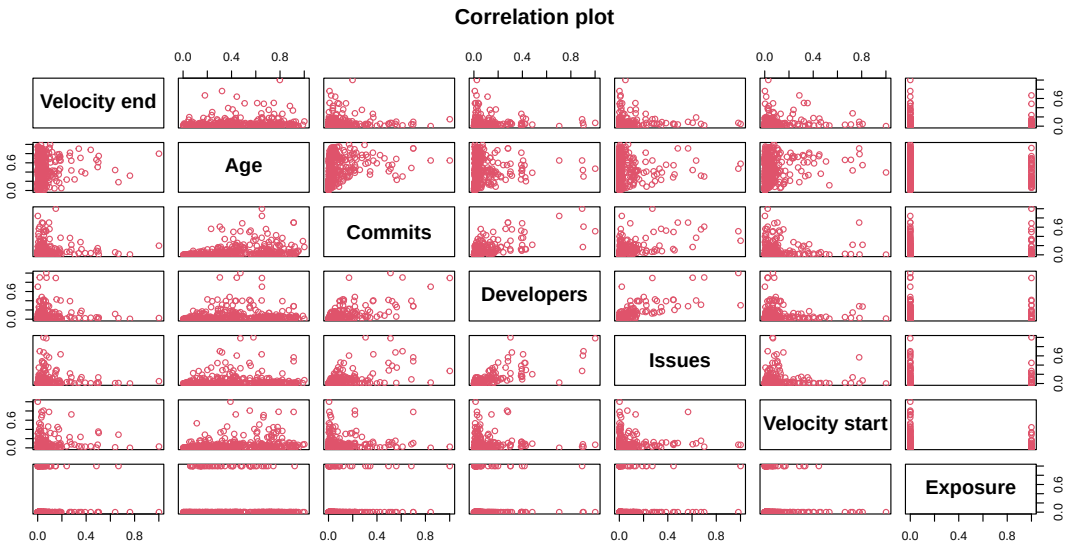


Figure 5.3. *Correlogram of scaled data.*

Furthermore, by taking a closer look at Figure 5.3 it was easy to see that the linear dependence among the dependent variable `Velocity end` and the rest of the explanatory variables were explicitly skewed and distant for being linear.

5.2 Regression with Linear Models

Even though the exploratory analysis in Section 5.1 did not denote high evidence of linearity in the data, linearity was assumed for the initial modelling step in order to check the effect of the performed feature scaling in the preprocessing stage as mentioned in Section 4.4.

As a common approach in the diverse assumptions considered during the analysis process, the initial models considered the main effects of all the considered explanatory variables $M_{M.E}$, and considered two $M_{v|v}$ and three-way interaction $M_{v|v|v}$ models as well. Similarly, as mentioned in Section 3.5, Backward Selection (BS) modelling was implemented through `step()` R function based on AIC criteria in order to discover the model best fitting the data.

Model	AIC	BIC	MSE
$M_{M.E}$	-497.1978	-466.6609	0.0127
$M_{v\|v}$	-476.3616	-388.568	0.0123
$M_{v\|v\|v}$	-467.8699	-303.7341	0.0112
BS model	-504.1395	-488.8711	0.0127

Table 5.1. *Model comparison results for Linear Models.*

As can be observed from Table 5.1, the BS process offered a slightly better model, with the explanatory variable set reduced to `Age` and `Velocity start`. However, as it can be seen in Table 5.2 the survival variables hardly obtained statistical significance from their p-values, a fact that was further analyzed in the stage covered in Section 5.4.

	Estimate	Std. Error	t value	p value
(Intercept)	0.0392	0.0121	3.25	0.0013 **
Age	0.0368	0.0230	1.60	0.1101
Velocity start	0.0770	0.0453	1.70	0.0896 .
	Multiple R^2	0.0195	**p value**	0.0378

Table 5.2. *Summary statistics from the observed best Linear Model.*

In addition, results for the model in Table 5.2 showed a low *coefficient of determination* or R^2 which denotes a weak predictive power for the model in case. Indeed, the p-value itself rejected the hypothesis of the model being able explain the data [13].

5.3 Regression with Generalized Linear Models

Turning now to the regression performed under the assumption of non-linearity, as mentioned in Section 3.2 three distributions are assumed for the analysis in this thesis. The organization of the regression analysis considers the same structure seen in Section 5.2. Section 5.3.1 considers the assumption of the Gaussian distribution for the regression, and similarly, Gamma and Inverse Gaussian distributions are covered in Sections 5.3.2 and 5.3.3 respectively based on the shape of the data seen in the performed explanatory analysis.

5.3.1 Assuming Gaussian distribution

The Gaussian distribution was expected to offer similar results as the ones observed with Linear Models, still, the new regression opportunities offered by the link functions offered possible suitable modelling options either considering only main effects and interactions too.

Link function	**Model**	**AIC**	**BIC**	**MSE**
Identity	$M_{M.E}$	-497.1978	-466.6609	0.0127
Identity	$M_{v\|v}$	-476.3616	-388.5680	0.0123
Identity	$M_{v\|v\|v}$	-467.8699	-303.7341	0.0112
Logarithmic	$M_{M.E}$	-495.6492	-465.1123	8.2473
Logarithmic	$M_{v\|v}$	-500.4273	-412.6338	**58.2010**
Logarithmic	$M_{v\|v\|v}$	-569.9655	-405.8298	**1,152.7540**
Inverse	$M_{M.E}$	-449.6051	-419.0683	**1,073.620**
Inverse	$M_{v\|v}$	1,044.0880	1,131.8810	**12,201,733**
Inverse	$M_{v\|v\|v}$	-610.9207	-446.7849	**14,401.990**
Identity	BS model	-492.0284	-381.3322	0.0113

Table 5.3. *Summary table of model comparison scores for Gaussian distribution models.*

Table 5.3 summarizes the modelling performed with the considered Gaussian distributions. Within this process with each of the link functions interactions were investigated, as well as BS was applied as mentioned before. To begin with, MSE results obtained for logarithmic and inverse link functions directly stood as a sufficient argument to reject them as suitable models. Similarly, the BS process conducted for the logarithmic and inverse link function-based models did not offer better results neither.

However, turning to the identity link, while the main effect model was presenting the best information criteria results among the initially considered identity link models, the latter ones presented slightly better MSE results. Within the BS process models derived from

the main effect model and the two-way interaction model resulted to offer similar results, to which the derived model from the three-way interaction model offered better results, as well as statistically significant interactions among the explanatory variables (see Table 5.4)

	Estimate	**Std. Error**	**t value**	**p value**
(Intercept)	0.0357	0.0173	2.06	0.0398 *
Age	0.0137	0.0361	0.38	0.7043
Commits	0.1998	0.4205	0.48	0.6351
Developers	0.8542	0.4306	1.98	0.0482 *
Issues	-1.0415	0.5302	-1.96	0.0504 .
Velocity start	-0.0970	0.1618	-0.60	0.5495
Exposed	-0.0423	0.0376	-1.13	0.2615
Age:commits	-0.1550	0.6288	-0.25	0.8055
Age:developers	-0.8701	0.8446	-1.03	0.3037
Age:issues	1.8219	1.1672	1.56	0.1196
Age:velocity start	0.3778	0.2748	1.38	0.1701
Age:exposed	0.0826	0.0887	0.93	0.3524
Commits:developers	-6.3591	2.2971	-2.77	0.0060 **
Commits:issues	3.0142	1.6127	1.87	0.0626 .
Commits:velocity start	5.2774	3.6968	1.43	0.1544
Developers:issues	2.4639	1.6304	1.51	0.1318
Developers:velocity start	-5.5899	2.0257	-2.76	0.0061 **
Issues:velocity start	-1.3758	4.5626	-0.30	0.7632
Velocity start:exposed	1.2658	0.3890	3.25	0.0013 **
Age:commits:developers	8.8954	3.4574	2.57	0.0106 *
Age:commits:issues	-4.7867	2.2712	-2.11	0.0359 *
Age:commits:velocity start	-9.4293	5.7965	-1.63	0.1048
Age:developers:issues	-6.2342	3.4094	-1.83	0.0684 .
Age:issues:velocity start	15.6692	8.2549	1.90	0.0586 .
Age:velocity start:exposed	-2.2693	0.7325	-3.10	0.0021 **
Commits:developers:issues	1.6385	1.1277	1.45	0.1473
Commits:developers:velocity start	19.2188	8.5848	2.24	0.0259 *
Commits:issues:velocity start	-18.2837	5.9418	-3.08	0.0023 **
	Multiple R^2	0.1240	**Adjusted** R^2	0.0472

Table 5.4. *Summary statistics from the best observed* $Mv|v|v$ *Gaussian GLM model with identity link.*

From the aforementioned table two main effects were observed to have an impact on the additional interactions, in fact, `velocity start` and `issues`. Indeed, it was in interactions where the mentioned effects would have an impact where the highest statistical significance would be registered. But, the model itself did not denote high predictive power, in fact, the obtained low R^2 and adjusted R^2 results clearly showed that the data was being explained by the model in a low level.

These observations denoted an evident impact of the interaction among the considered variables on the significance of the model to predict the response variable. Still, further analysis had to be done on different distributional assumptions, a subject to be explained in the following sections.

5.3.2 Assuming Gamma distribution

From the exploratory analysis described in Section 5.1 it can be seen evidence of non-normality in the data, in fact, the skewness Figure 5.2 denoted a graphical pattern more related to Gamma distribution.

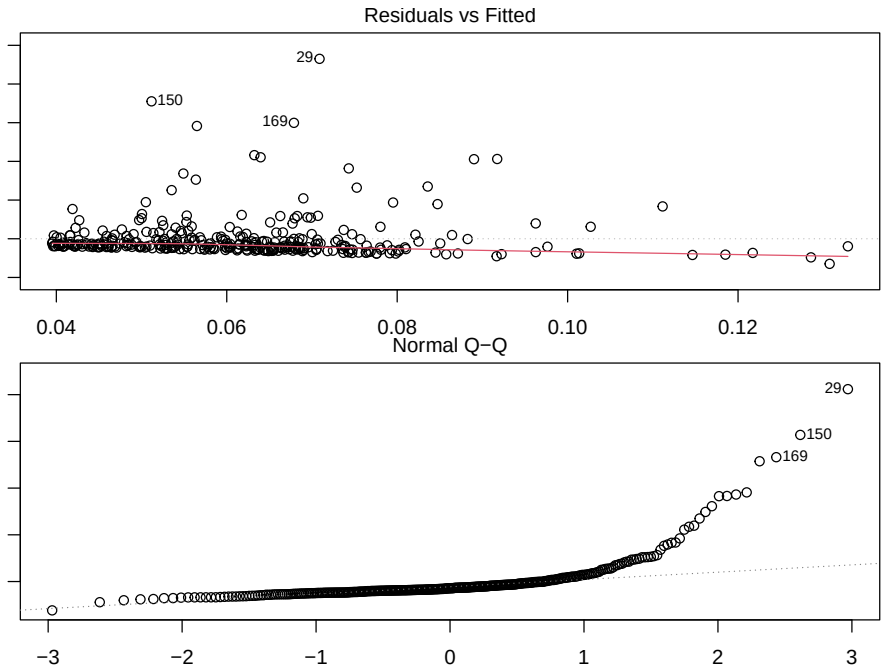


Figure 5.4. *Summary of residuals from the best Linear Regression Model.*

Furthermore, given the fact that by definition the considered variables in the model are non-negative, and residuals from the initial model denoted an increase in the variance proportional to the mean as it can be seen in Figure 5.4, there was enough evidence to conduct a regression analysis based on Gamma distributions.

Directly Table 5.5 there can be seen two notable results. First, the algorithm for interaction models with identity and inverse link functions did not converge with the shape of the data. And then, notable high errors were detected in the MSE results for models with a logarithmic link and the main effect model with an inverse link.

From the resulting modelling options, it was the BS process for the main effect model with identity link the one that presented the best model. Interestingly, the pattern was similar to the result from the BS process when assuming linearity since variables `Age` and `Velocity start` were the only surviving ones in the model.

The resulting model offered a higher statistical significance for the model variables, visible in Table 5.6. However, even with Gamma distribution low results such as Multiple R^2 and Adjusted R^2 denoted small predictive power in the model.

Link function	Model	AIC	BIC	MSE
Identity	$M_{M.E}$	-1262.253	-1231.716	0.0128
Identity	$M_{v\|v}$	–	–	–
Identity	$M_{v\|v\|v}$	–	–	–
Logarithmic	$M_{M.E}$	-1259.181	-1259.181	**8.4478**
Logarithmic	$M_{v\|v}$	-1245.864	-1158.071	**8.7548**
Logarithmic	$M_{v\|v\|v}$	-1242.885	-1078.749	**9.3374**
Inverse	$M_{M.E}$	-1254.619	-1224.082	**303.4082**
Inverse	$M_{v\|v}$	–	–	–
Inverse	$M_{v\|v\|v}$	–	–	–
Identity	BS model	-1271.197	-1255.929	0.0128

Table 5.5. *Summary table of model comparison scores for Gamma distribution models.*

	Estimate	Std. Error	t value	p value
(Intercept)	0.0277	0.0076	3.62	0.0003 ***
Age	0.0547	0.0207	2.64	0.0086 **
Velocity start	0.1475	0.0855	1.73	0.0854 .
	Multiple R^2	0.0499	**Adjusted** R^2	0.0442

Table 5.6. *Summary statistics from the best observed $M_{M.E}$ Gamma GLM model with identity link.*

5.3.3 Assuming Inverse Gaussian distribution

As mentioned in Section 3.3.3, the Inverse Gaussian distribution is presented as a clear competitor against Gamma distribution, in fact, both distributions follow the same assumptions but they slightly differ on the proportionality of the variance towards the mean of the response variable.

Still, the Inverse Gaussian distribution results to be a difficult distribution to fit, and in fact, it can be seen from Table 5.7 the algorithms for most of the link functions did not converge.

Interestingly though, in the case of Inverse Gaussian distribution, the BS process reached the best model with identity link as in previous cases, but this time the best model only discarded `developers` variable. Besides, it is visible in Table 5.8 that all the main effects of the considered variables are considered statistically significant, despite the independent variable of the analysis `Exposed`.

Additionally, looking at the values for the multiple and adjusted R^2 the pattern of all the models observed during the regression analysis denote the same pattern, in fact, the predictive power remains low given any of the assumed distributions. This specific point

Link function	Model	AIC	BIC	MSE
Identity	$M_{M.E}$	-1276.161	-1245.624	0.021
Identity	$M_{v\|v}$	–	–	–
Identity	$M_{v\|v\|v}$	–	–	–
Logarithmic	$M_{M.E}$	-1265.063	-1234.526	**8.3899**
Logarithmic	$M_{v\|v}$	–	–	–
Logarithmic	$M_{v\|v\|v}$	–	–	–
Inverse	$M_{M.E}$	–	–	–
Inverse	$M_{v\|v}$	–	–	–
Inverse	$M_{v\|v\|v}$	–	–	–
Canonical	$M_{M.E}$	–	–	–
Canonical	$M_{v\|v}$	–	–	–
Canonical	$M_{v\|v\|v}$	–	–	–
Identity	BS model	-1278.026	-1251.307	0.0172

Table 5.7. *Summary table of model comparison scores for Inverse Gaussian distribution models.*

	Estimate	**Std. Error**	**t value**	**p value**
(Intercept)	0.0306	0.0062	4.95	$1.21e{-6}$ ***
Age	0.0572	0.0185	3.08	0.0022 **
Commits	-0.0925	0.0125	-7.38	$1.33e^{-12}$ ***
Issues	0.5740	0.2591	2.22	0.0274 *
Velocity start	-0.0524	0.0068	-7.70	$1.55e^{-13}$ ***
Exposed	0.0284	0.0229	1.24	0.2159
	Multiple R^2	0.0463	**Adjusted** R^2	0.0318

Table 5.8. *Summary statistics from the best observed $M_{M.E}$ Inverse Gaussian GLM model with identity link.*

is further discussed in Chapter 6.

5.4 Resulting regression model

This final section aims on comparing the best models observed from the considered assumptions in previous sections. Table 5.9 offers an overview of the model selection performed at this last stage of the regression analysis.

Among the initial considerations derived from the aforementioned table, the clear difference between distributions assuming normality and the ones that do not arise first. In fact,

Model	AIC	AICc	BIC	MSE
$M_{M.E}$ LM	-504.139	-504.019	-488.871	0.0127
Gaussian $M_{v\|v\|v}$ GLM	-492.028	-486.342	-381.332	**0.0114**
Gamma $M_{M.E}$ GLM	-1271.197	-1271.076	**-1255.929**	0.0128
Inv. Gaussian $M_{M.E}$ GLM	**-1278.026**	**-1277.685**	-1251.307	0.0172

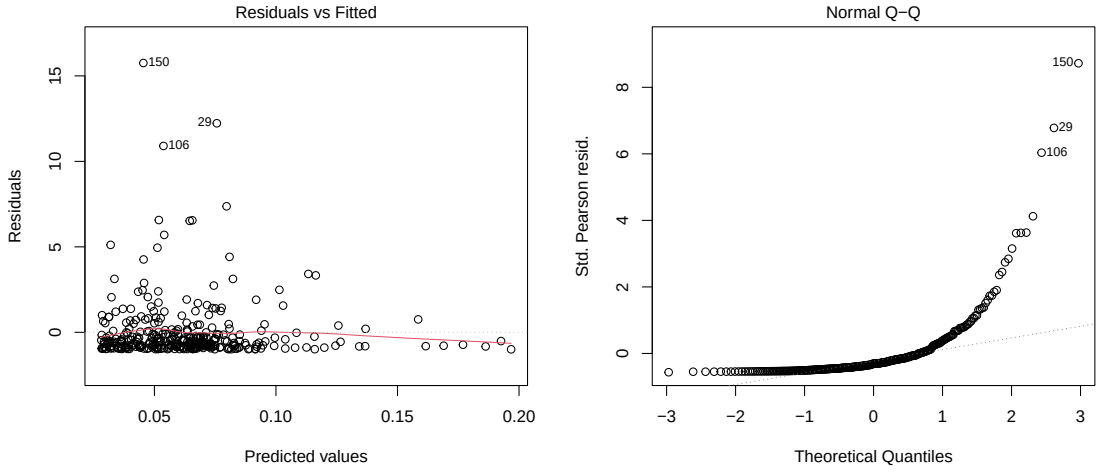
Table 5.9. *Summary table of model comparison comparison between the best observed models.*

despite the best MSE result came from the Gaussian interaction model, the rest of the indicators resembled a clear difference supporting distributions like Gamma and Inverse Gaussian.

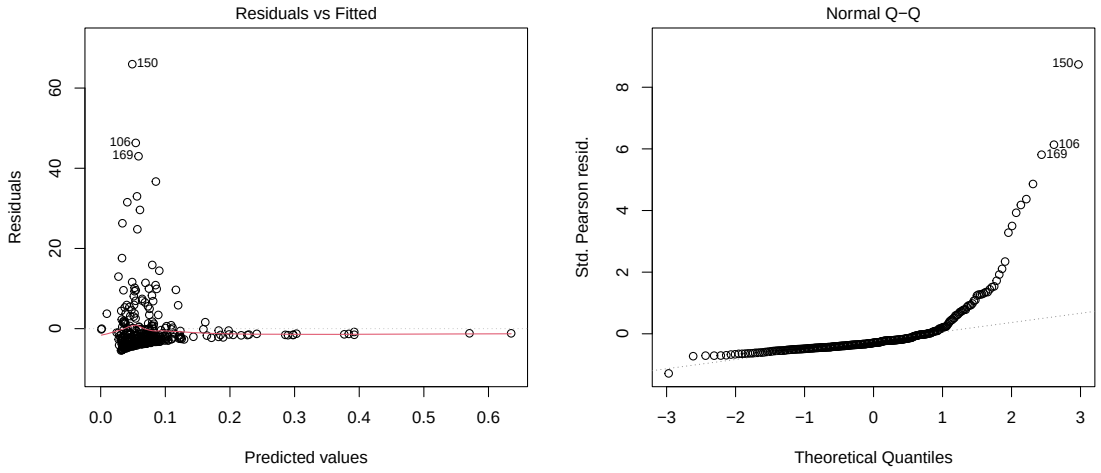
In addition, the low significance of the AICc test became evident due to its close results to the AIC values, so the former one did not provide much information within the analysis.

Turning into the non-normal models, already when analyzing the summary statistics from the observed best models it was evident that despite the explanatory variable had acquired statistical significance in comparison to models assuming normality, this change was not occurring at a model level. In fact, the observed best models did not present high predictive power since the explanatory variables were not able to describe the variance of the response variable.

Taking a closer look at the residuals obtained from the considered Gamma and Inverse Gaussian models, Figure 5.5 shows two evident patterns to understand the results from the regression. First, the *QQ* plot in both models depicts a clear distance from normality, which matches the model comparison results presented in Table 5.9. Secondly, the *residuals vs fitted* plot shows with enough robustness the non-linear nature of the response variable.



(a) *Residuals from Gamma $M_{M.E}$ model.*



(b) *Residuals from Inverse Gaussian $M_{M.E}$ model.*

Figure 5.5. *Residual plots from the observed best non-normally distributed models.*

Interestingly enough, the presented final results provide two main considerations to be further discussed in Chapter 6. In fact, on one hand, all the resulting best-fitted models denoted a lack of ability to describe the variance of the response variable. On the order hand, only one of the models offering the best results, Gaussian $M_{v|v|v}$ GLM, offered a case scenario in which the interaction of the explanatory variables played an important role, the rest of the models only considered main effects models.

And to close this section, it must be noted how the variable `exposed` disappeared from the best models, or did not reflect statistical significance to explain the variance of the development velocity at the end of the follow-up period.

5.5 Regression with Random Forests

This section covers the results from the application of ensemble learning through random forests. The library `randomForest` in R provides the option of performing regression by providing a specific model. In this sense, the models presented in Table 5.9 were computed with different number of trees in order to observe whether the effect of ensemble learning could show changes in the regression results.

Figure 5.6 shows the values obtained from the MSE assessment of the different random forests implemented. In fact, only three model structures were used when implementing ensemble learning, these were the main effect model obtained in the best LM and Gamma distributed GLM, the interaction model obtained in the best Gaussian GLM and the main effect model obtained in the best Inverse Gaussian GLM excluding `Developers`.

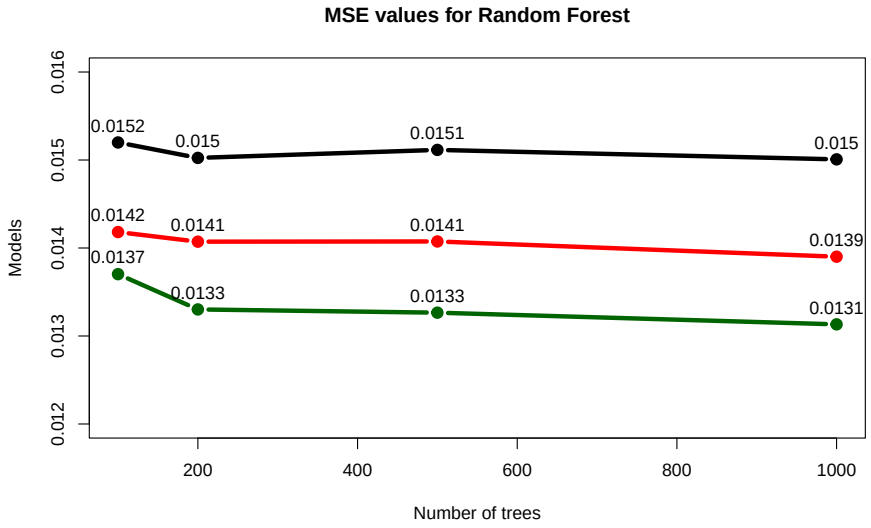


Figure 5.6. *MSE values for model structure with simplified $M_{M.E}$ (black), $M_{v|v|v}$ (red) and $M_{M.E}$ without Developers variable (green).*

The aforementioned results described the same scenario as in Section 5.4 with the model used with Inverse Gaussian distribution being the best model fitting the data. Additionally, Figure 5.7 clearly described the variable importance of the considered explanatory variables or predictors when explaining the response variable.

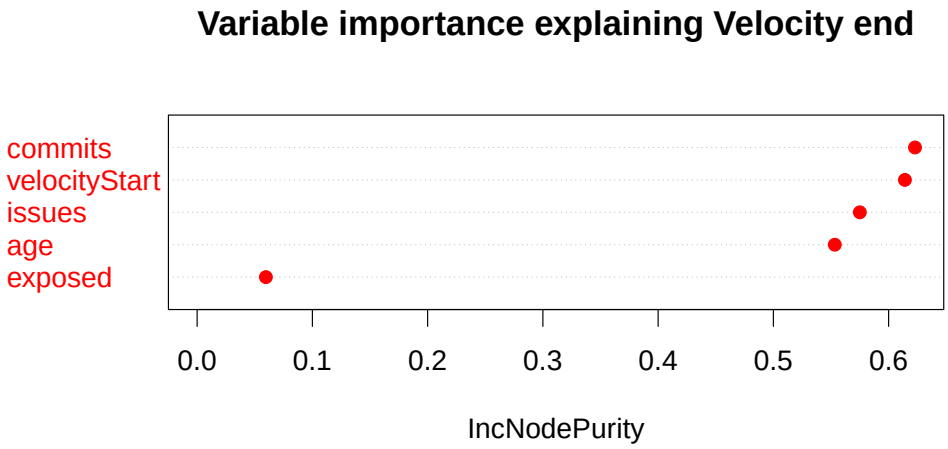


Figure 5.7. *Graphical representation of model predictors based on average increase in node purity.*

This chapter described an additional implementation for the regression analysis performed that, distant from differing from the results obtained with LM and GLM models, it confirmed the same model structure developed in the previous chapter.

6. CONCLUSIONS

The goal of this thesis was to analyze the impact of SQ on the development velocity of software development projects, for that regression analysis was the considered statistical analysis method. Additionally, this thesis aimed to understand the relationship of the considered explanatory variables with the dependent variable, that is, how able the predictors were to describe the variance of the development velocity. This chapter closes this thesis by summarizing and discussing the main results obtained in Chapter 5. Similarly, at the end of this chapter considered further research is addressed.

In order to run the regression analysis, a set of considered explanatory variables was obtained from a data collection process following an observational design based on cohort studies theory. The data collection process returned a total number of 337 eligible projects, 52 using SQ and becoming the exposed subjects, 285 being therefore the non-exposed. The structure of the regression analysis started with the assumption of linearity as a first step and continued with the use of GLMs by not assuming linearity. Pursuing the goal of this thesis, the BS process was performed under the different considered assumptions in order to discover the respective best model.

Moving on to the obtained best models, the regression models obtained through the BS process described considerable results that need a interpretation. Running the BS process under LMs and when assuming Gamma distribution resulted in the same reduced model, including only variables age and velocity at the start of the observational period. BS resulted in a main effects model without considering the number of developers when assuming Inverse Gaussian distribution where all the explanatory variables resulted being statistically significant despite the exposure of the projects to SQ, the key variable in this thesis. And finally, Gaussian distribution modelling resulted in a three-way interaction model offering statistical significance in multiple interaction cases regarding explanatory variables such as the exposure to SQ, the velocity at the start of the observational period and the age of the project.

These resulting models offered a first important interpretation, in three out of four considered best models the exposure of SQ was not significant to explain the variance of the development velocity of projects. Additionally, only under the Gaussian distribution, the interaction between the predictors offered statistical significance, a factor subject to sur-

prise given the mixture of effects that impact simultaneously on the development velocity of a team when working on a task.

Turning now to the model comparison results obtained for the described best models, the initial assumptions of non-linearity were confirmed. In fact, the Inverse Gaussian model was the one fitting the data best, followed closely by the model assuming Gamma distribution. These results clearly denoted the tendency of the variance of the development velocity to increase proportionally to the mean. However, the model comparison did not describe the real predictive power of the considered to be the best models. Generally in all models analyzed within the thesis, observed statistics such as the multiple R^2 described a low ability in models to describe the variance of the development velocity, in this way offering a low predictive power too.

With the aim to conclude this manuscript, the regression analysis performed offered one of the few carried statistical analyses to understand the impact of SATs such as SQ on software development projects. Results demonstrate that there is much further research to be done in order to understand the quantitative relationships that lie behind the variance of development velocity. Moreover, the results from this thesis suggest the need for different possible approaches to obtain the data, i.e., considering measuring the development velocity in longitudinal structured format, so that the nature of the variance of the development velocity is better described by the used data. Therefore, it would be interesting to see that future work considers this thesis as a baseline and, with a possible bigger size of data, can study possible semi-parametric or non-parametric that may describe better the impact of potential predictions on the variance of the development velocity.

REFERENCES

[1] Khleel, N. A. A. and Nehéz, K. Mining Software Repository: an Overview. *Doktoranduszok Fóruma* (), p. 108.

[2] Lenarduzzi, V., Lomio, F., Huttunen, H. and Taibi, D. Are sonarqube rules inducing bugs?: *2020 IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE. 2020, pp. 501–511.

[3] Saarimäki, N., Robredo, M., Lenarduzzi, V., Vegas, S., Oko-Oboh, G., Juristo, N. and Taibi, D. Does SonarQube Impact the Development Velocity? A Preliminary Study. *Submitted to International Symposium on Empirical Software Engineering and Measurement (ESEM)*. Under review. IEEE, 2023.

[4] *What is an issue?* Atlassian. Pty Ltd. URL: https://support.atlassian.com/jira-software-cloud/docs/what-is-an-issue/ (visited on 03/20/2020).

[5] Rubin, K. S. *Essential Scrum: A practical guide to the most popular Agile process*. Addison-Wesley, 2012.

[6] Al-Sabbagh, K. W. and Gren, L. The connections between group maturity, software development velocity, and planning effectiveness. *Journal of Software: Evolution and Process* 30.1 (2018), e1896.

[7] *SonarCloud. SonarCloud organization profile site for the Apache Software Foundation*. SonarSource S.A. URL: https://sonarcloud.io/organizations/apache/projects?sort=size (visited on 04/20/2023).

[8] Lomio, F., Moreschini, S. and Lenarduzzi, V. A machine and deep learning analysis among SonarQube rules, product, and process metrics for fault prediction. *Empirical Software Engineering* 27.7 (2022), p. 189.

[9] Lenarduzzi, V., Lujan, S., Saarimaki, N. and Palomba, F. A critical comparison on six static analysis tools: detection, agreement, and precision. *arXiv preprint arXiv:2101.08832* (2021).

[10] Lenarduzzi, V., Besker, T., Taibi, D., Martini, A. and Fontana, F. A. A systematic literature review on technical debt prioritization: Strategies, processes, factors, and tools. *Journal of Systems and Software* 171 (2021), p. 110827.

[11] *Apache Projects List*. The Apache Software Foundation. URL: https://projects.apache.org/projects.html?name (visited on 04/20/2023).

[12] Song, J. W. and Chung, K. C. Observational studies: cohort and case-control studies. *Plastic and reconstructive surgery* 126.6 (2010), p. 2234.

[13] Agresti, A. *Foundations of linear and generalized linear models*. John Wiley & Sons, 2015.

[14] Chatterjee, S. and Simonoff, J. S. *Handbook of regression analysis*. John Wiley & Sons, 2013.

[15] McCullagh, P. *Generalized linear models*. Routledge, 2019.

[16] Dunteman, G. H., Ho, M.-H. R. and Ho, M.-H. R. *An introduction to generalized linear models*. Vol. 145. Sage, 2006.

[17] Cox, D. R. and Hinkley, D. V. *Theoretical statistics*. CRC Press, 1979.

[18] Casella, G. and Berger, R. L. *Statistical inference*. Cengage Learning, 2021.

[19] Fortran oroginal by Leo Breiman and Adele Cutler, R port by Andy Liaw and Mathew Wiener. *Breiman and Cutler's Random Forests for Classification and Regression.* CRAN. 2022. URL: https://cran.r-project.org/web/packages/randomForest/randomForest.pdf.

[20] Breiman, L. Random forests. *Machine learning* 45 (2001), pp. 5–32.

[21] Hosmer Jr, D. W., Lemeshow, S. and Sturdivant, R. X. *Applied logistic regression*. Vol. 398. John Wiley & Sons, 2013.

[22] Snipes, M. and Taylor, D. C. Model selection and Akaike Information Criteria: An example from wine ratings and prices. *Wine Economics and Policy* 3.1 (2014), pp. 3–9.

[23] deLeeuw, J. Introduction to Akaike (1973) Information Theory and an Extension of the Maximum Likelihood Principle. *Breakthroughs in Statistics: Foundations and Basic Theory*. Ed. by S. Kotz and N. L. Johnson. New York, NY: Springer New York, 1992, pp. 599–609. ISBN: 978-1-4612-0919-5. DOI: 10.1007/978-1-4612-0919-5_37. URL: https://doi.org/10.1007/978-1-4612-0919-5_37.

[24] HURVICH, C. M. and TSAI, C.-L. Regression and time series model selection in small samples. eng. *Biometrika* 76.2 (1989), pp. 297–307. ISSN: 0006-3444.

[25] Vrieze, S. I. Model selection and psychological theory: a discussion of the differences between the Akaike information criterion (AIC) and the Bayesian information criterion (BIC). *Psychological methods* 17.2 (2012), p. 228.

[26] Gupta, H. V., Kling, H., Yilmaz, K. K. and Martinez, G. F. Decomposition of the mean squared error and NSE performance criteria: Implications for improving hydrological modelling. *Journal of Hydrology* 377.1 (2009), pp. 80–91. ISSN: 0022-1694. DOI: https://doi.org/10.1016/j.jhydrol.2009.08.003. URL: https://www.sciencedirect.com/science/article/pii/S0022169409004843.

[27] Gentle, J. E. *Computational statistics*. Springer, 2010.

[28] *The Apache Software Foundation. Official GitHub repository of the Apache Software Foundation*. GitHub Inc. URL: https://github.com/apache (visited on 04/20/2023).

[29] *ASF Jira. System Dashboard - ASF Jira*. Atlassian. Pty Ltd. URL: https://issues.apache.org/jira/secure/Dashboard.jspa (visited on 04/20/2023).