International Journal of Open Source Software and Processes
Volume 10 • Issue 1 • January-March 2019

Predicting the Severity of Open Source Bug Reports Using Unsupervised and Supervised Techniques

Pushpalatha M N, Ramaiah Institute of Technology, Bengaluru, India
Mrunalini M, Ramaiah Institute of Technology, Bengaluru, India

ABSTRACT

The severity of the bug report helps for the bug triagers to prioritize the handling of bug reports for giving more importance to high critical bugs than less critical bugs, since the inexperienced developers and new users can make mistakes while assigning the severity. The manual labeling of severity is labor-intensive and time-consuming. In this article, both unsupervised and supervised learning algorithms are used to automate the prediction of bug report severity. Because the data was unlabeled, the Gaussian Mixture Model is used to group similar kinds of bug reports. The result is labeled data with the severity level given for each bug reports. Then, the training of classifiers is performed to predict the severity of new bug reports submitted by the user using Multinomial Naïve Bayes Classifier, Logistic Regression Classifier and Stochastic Gradient Descent Classifier. Using these methods, around 85% accuracy is obtained. More accurate predictions can be done using the authors approach.

KEYWORDS

Bug Reports, Machine Learning, Severity

1. INTRODUCTION

Open source projects such as Firefox, Mozilla, Android, Bugzilla, etc., receives lot of bug reports from all over the world by user and developers, which are stored in the Bugzilla bug Repositories; on average of 29 bug reports per day (Anvik et al., 2006). If assume that the triager takes 5 minutes to examine and handle the bug report, then the traiger has to spend around two hour per day on bug reports. If the large number of bug reports are present in the repositories only fractions of bugs will get chance for fixing. Before fixing any bugs, it needs to be prioritized according to its severity. Severity field helps to identify how urgent the bug needs to be fixed. Prioritizing the bug reports based on the severity helps for the bug triager to assigns the high severe bug report to appropriate developer in order to speed up the fixing process.

DOI: 10.4018/IJOSSP.2019010101

New users and inexperienced developers make a mistake in identifying the correct severity label while reporting the bug reports. (Lamkanfi et al., 2010; Pushpalatha et al., 2016) authors not considered the normal severity for severity prediction, because it is default option. In (Lamkanfi et al., 2010) authors suspected that submitters did not assess the severity of bug report consciously. They confirmed this with manual sampling. (Yuan et al., 2016) done research on checking the reliability of the bug report severity labels. Duplicate bug reports refer to the same problem, each duplicate bug report contains different severity label, even though they refer the same problem. There is inconsistency in assigning the severity label. Manual checking and assigning the correct severity will take lot of time and resources, because of lots of bug reports in the bug repository.

In (Lamkanfi et al., 2010; Lamkanfi et al., 2011; Pushpalatha et al., 2016) used Naïve Bayes, Naïve Bayes multinomial, Support vector machine, K-nearest neighbor, J48 and bagging classifiers for predicting the severity of new bug reports using already labeled historical bug reports. (Nachai et al., 2014) authors used the Expectation Maximization (EM) and X-means clustering algorithm for grouping the similar kind of bug reports based on the similarity and done experimentation on Jira bug reports. (Guo, Chen & Li, 2017) Android bug reports severity is predicted using Naïve bayes. Model is built using the labeled data of Eclipse and Mozilla bug reports and built model is tested on the unlabeled bug reports of Android software.

In this work, Gaussian mixture model is used grouping similar kind of bug reports and experimentation is done on the Firefox bug reports taken from the Bugzilla Bug repository. Using this model, a bug triager can use the bugs reported by their users and developer, group them into clusters. After assigning the label to each cluster, labeled data is used for predicting the label of unlabelled bug reports for that used different supervised algorithms such as Naïve Bayes, logistic regression and Stochastic Gradient Descent on the labeled data. This will help to save time and resources.

Contribution of this work:

- Grouped the similar kind bug reports and predicted the labels of bug reports using unsupervised learning;
- Using labeled data Predicted the severity of bug reports using supervised learning such as Logistic Regression, Multinomial Naïve Bayes and Stochastic Gradient descent.

Organization of the paper is as follow, section 2 discusses about the Framework of proposed approach, section 3 gives background information, result and discussion in section 4, section 5 explains about the related work. Finally, in section 6, we conclude and discuss future enhancements of our work.

2. FRAMEWORK FOR PROPOSED WORK

Figure 1 shows the Framework for the proposed model, each step of the Framework is explained in the following sections.

2.1. Collect Bug Reports in CSV Format From the Bug Repository

Collected bug reports for Firefox for desktop and android bug reports in Comma Separated Value (CSV) format from Bugzilla open source bug repository which contains bug reports of different open source software such as Bugzilla, Firefox, etc.

2.2. Apply Preprocessing Techniques

Different preprocessing steps are applied on the collected bug reports such as Tokenization, Stop word removal, Stemming, weighted tf-idf calculation and dimensionality reduction using Singular value decomposition.