# TECHNICAL DEBT IDENTIFICATION USING TEXT CLASSIFICATION

**PROJECT REPORT**

***Submitted by***

**Kaliappan**

**David Sundaraj**

**Prashanth Lidwin Jessuva**

*submitted to the Faculty of*

**INFORMATION SCIENCE AND TECHNOLOGY**

*in partial fulfillment for the award of the degree*
***of***

**BACHELOR OF TECHNOLOGY**
***in***

**INFORMATION TECHNOLOGY**

**DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY**
**COLLEGE OF ENGINEERING, GUINDY**
**ANNA UNIVERSITY**
**CHENNAI 600 025**
**MONTH YEAR**

# ANNA UNIVERSITY

# CHENNAI - 600 025

# BONA FIDE CERTIFICATE

Certified that this project report titled Technical Debt Identification Using Text Classification is the bona fide work of Kaliappan, David, Prashanth who carried out project work under my supervision. Certified further that to the best of my knowledge and belief, the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or an award was conferred on an earlier occasion on this or any other candidate.

**PLACE: CHENNAI**
**DATE: 31/11/2018**

**Dr. SASWATI MUKHERJEE**
**PROFESSOR**
**PROJECT GUIDE**
**DEPARTMENT OF IST, CEG**
**ANNA UNIVERSITY**
**CHENNAI 600025**

**COUNTERSIGNED**

**Dr. SASWATI MUKHERJEE**
**HEAD OF THE DEPARTMENT**
**DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY**
**COLLEGE OF ENGINEERING, GUINDY**
**ANNA UNIVERSITY**
**CHENNAI 600025**

# ABSTRACT

The issue of Technical debt has been a subject of controversy for sometime now, Technical debt as such is not harmful for a software project but it is essential to identify such technical debts in the project as soon as possible because the later you find the more the debt increases and at last it becomes unbearable to pay back similar to the financial debts in our real life.

There have been many methods suggested to identify technical debts in the project like Code Smells, Source Code Analysis, Comment analysis etc., In this project we have come with a new and novel way to identify these technical debts and it is by going through the bug reports of that particular software. On applying text processing to the bug reports we have classified a bug report as either a technical report or a non-technical report. Thus the developers and testers can concentrate in detail on the bugs that were marked as technical.

For this project we have concentrated on the Chromium Project, it is a open source project from Google. The bug can be filed by anyone using a software called Monorail. We have scraped 700 of these issues for now as our dataset and given this to software engineering experts who classified these bugs as either technical or non-technical. This is the dataset using which we have experimented with various Text Classification methods. In essential we are using NLP methods to analyse and understand the issues and then classify it as either Technical or Non-Technical.

*Keywords*: Technical Debt, Chromium Project, Code Smells, NLP.

# ACKNOWLEDGEMENT

We would like to thank our HOD for having faith in us and giving us this project to work on and we would like to thank her for her guidance and support. We also thank Dr.Prabhavathy who helped us to understand things when HOD is busy with a work. We also thank our project lab professor Dr.Sridhar for giving us some valuable advice and guiding us in this project.

# TABLE OF CONTENTS

# LIST OF SYMBOLS AND ABBREVIATIONS

| | |
|---|---|
| $-, \neg, \sim$ | Negation operator |
| $+, \vee, \cup$ | Disjunction operator |
| $X, \wedge$ | Conjunction operator |
| $\rightarrow$ | Conditional operator |
| $\leftrightarrow$ | Biconditional operator |
| $\diamond$ | Future tense modal operator |
| $\alpha$ | Action |

# CHAPTER 1

# INTRODUCTION

## 1.1 MOTIVATION:

Technical Debt is a shortcut taken in a project development which is taken in order to speed up the delivery or completion of the project. Technical Debt as such is not harmful for example not commenting important information in some parts of the code and releasing it in a hurry is also a form of technical debt but the very crucial thing is after it is delivered the development team must make sure that they go back to the released skeleton and make sure that they comment it as soon as they get time after releasing the first version.

The real problem kicks in when after releasing the project if the team still does not add the left out comments, then future development of this code is hampered. So technical debt causes a problem when we let the debt to accumulate very similar to Financial Debt, in finance we make sure that the Debt we borrowed does not reach a humongous proportions, if it reaches such levels then we will not be able to pay off our Debts. Similarly if we do not periodically monitor and pay back our Technical Debts then it causes a huge disaster for companies.

But Technical Debt is not always identifiable, because sometimes by accident a Technical Debt is formed, so it is very hard to identify a Technical Debt in a software project.So it is becoming very important and pertinent for many companies around the world in the present to identify a very efficient and accurate method to identify a Technical Debt in a software project. Finding a accurate method which detects Technical Debt is the motivation of our project.

## 1.2 PROJECT BACKGROUND:

For our project we are working on the Chromium Dataset that we got from Penn State University. Chromium is the open source counterpart of Google Chrome and the dataset are the bug issues of the browser Chromium, and there is a need for Chromium project to find out the bugs that have Technical Debt inherent with them, this is necessary because a bug due to Technical Debt causes a big software disaster and hence those bugs must be identified as soon as possible. At present Chromium project makes use of Software engineering experts to identify the Technical Debt bugs.

Chromium project has a open source issue tracker from where this data is extracted. Our Dataset consists of 700 issues out of which 200 issues are classified by software engineering experts and remaining 500 issues are classified by CMU. All 700 issues can be seen in the Issue-Tracker. We also have scraped out some issues from the issue tracker for our project. The 700 issue dataset has a binary classification label- either label T which indicates the description is Technical or NT which indicates that the issue is Non-Technical.

The dataset that we got has the following fields: Id, Author, Comments, Comments-Date, Comments-Label, Date, Description, Role, Status, Title, Type, Closed, Priority, Rating, Date, Label and Keywords. So we are applying our NLP and ML logic on this dataset.

## 1.3 NECESSITY:

- At present the issues are being classified by Software Engineering experts. It takes a lot of time and money as it involves human Labour.
- Labelling each issue involves looking into a great amount of data

about that issue which can be overwhelming.

- So it is necessary that inorder to fasten this process there is need to automate this labelling process, for this we need to come up with a algorithm that reads a issue and then gives it a label which is as accurate as given by a human expert.

## 1.4 CHALLENGES:

- No one has tried this method of automating this process
- We have to find a suitable feature selection method that will help in classification of these documents.

# CHAPTER 2

# LITERATURE SURVEY/RELATED WORK

In order to understand about technical debt and the current methods of identifying this in a project there was an extensive Literature Review done, as part of the Literature Review many IEEE and ACM papers were read so that there is a complete understanding about the existing methods and the places where the method can be improved.

## 2.1 THE WYCASH PORTFOLIO MANAGEMENT SYSTEM.

In the development of a software named WyCash+ for Wyatt Technology, there was a problem of a new feature fitting poorly in existing architecture, by using a makeshift method the new feature was accommodated and released. Later this makeshift method was replaced with the fully functional feature. Cunningham had first used the analogy of technical debt in this research paper.

## 2.2 TRACKING TECHNICAL DEBT.

In this paper the authors had monitored a real MNC application. The application they monitored made use of MS-Exchange Server. At the time when they started work on this application the latest version of MS Exchange Server they had was 2003 version and there was the news that MS Exchange Server 2007 was going to release sooner or later, but they started developing the project with dependency on MS Exchange 2003. But unfortunately within 6 months of release the project MS Exchange 2007 version got released and hence they had to scrape out the released software thus causing a huge loss for the MNC.

## 2.3 TECHNICAL DEBT AND AGILE SOFTWARE DEVELOPMENT PRACTICES AND PROCESSES.

Agile software development process and practices have an effect on technical debt. Agile practices safeguarding software implementation have the most positive effect. Technical debt knowledge is implicit and hence the concept is under utilised.

## 2.4 IDENTIFYING SELF-ADMITTED TECHNICAL DEBT IN OPEN SOURCE PROJECTS USING TEXT MINING.

In this paper the authors made use of text mining to identify self admitted technical debt. Self admitted technical debt is the technique of adding information about technical debts in source codes by adding comments etc., Like for example someone may comment saying that they have added a temporary variable as a shortcut etc., Thus on mining these comments we must identify whether it happens to be a Technical Debt or not. They have made use of natural processing techniques in order to identify the Technical Debts in the source file.

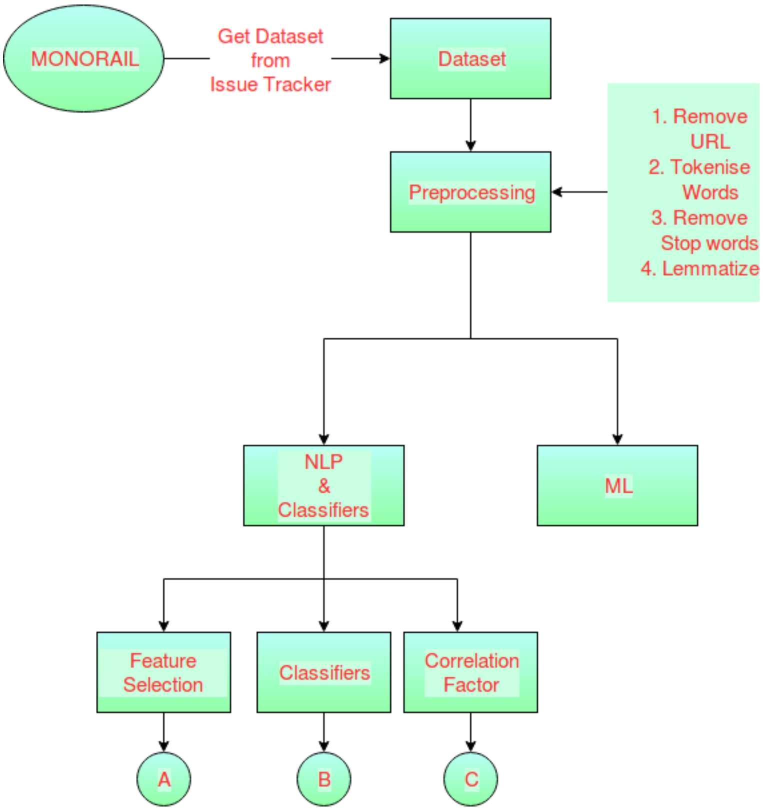
## 2.5 DETECTING TECHNICAL DEBT FROM ISSUE TRACKERS.

In this paper the authors made use of text mining on the issues that are found in the Issue trackers of the software. For this paper the authors had used the issues of a particular software whose issue can be got from the issue tracker. And then using a key word approach they extracted features and then gave it to Naive Bayes Classifier and thus got the required results.

# CHAPTER 3

# DESIGN OF YOUR WORK

## 3.1 ARCHITECTURE DIAGRAM:



# CHAPTER 4

# IMPLEMENTATION OF YOUR WORK

The steps done in the implementation are as follows:

## 4.1 PREPROCESSING:

The given dataset from the issue tracker has many unimportant things like punctuation, hyperlinks, stop words . These things are removed in the Preprocessing stage.

## 4.2 FEATURE SELECTION:

The right features must be selected from this dataset and for this we need to find the right feature selecetion technique and some of the feature selection techniques that was used in this project are: Pos-tags, Tfidf, Word embeddings.

## 4.3 CLASSIFICATION:

In this step we made ue of staple classifiers to find out the accuracy of our method, We had to experiment on different classifiers to find out which shows good accuracy.

## 4.4 ENSEMBLE METHODS:

After trying the normal classifiers we made use of Ensemble methods to find out if they improve the accuracy in classification.

## 4.5 RESULTS:

The results were genereated in the form of confusion matrix and analysed, We found that the Ensemble method XGBoost to give us high accuracy.

# CHAPTER 5

# MODULES:

## 5.1 MODULE 1:

Initially we are going to do a trial and error process where we will try out as many NLP methods and at the same time try out as many Machine learning techniques like classification, prediction etc., Then we are trying to mix Machine learning and NLP feature selection technique. Initially we are working by taking only the Description and Label fields of the Dataset. Then we are going to try the Ensemble methods to see their classification acuracy. So in the second major part we will be making use of the other fields that are available in the dataset.

## 5.2 MODULE 2:

In this phase we will try and finish the Semi-Supervised Learning Method. Then we are planning to do a social network analysis of the creators and moderators of the particular issue, and we are going to include their social networking behaviours to the classification process, So that essentially we give importance to the comments of the people who are really technical people and not some non technical people who may have some less insight into the real nature of the bug.

# CHAPTER 6

# CONCLUSION AND FUTURE WORK

We found that out of the current features and classification methods the ensemble method proves to give the highest accuracy. Now for future work we must and try and improve the feature selection method that we are using in this project. And then test and see how the feature selection method improves the accuracy.

# REFERENCES

[1] Anna University PhD-Regulations 2015. `https://cfr.annauniv.edu/research/regulation/PhD-Regualtion-2015.pdf`. Accessed: 20 March 2015.

[2] K Alishahi, F Marvasti, V A Aref, and P Pad. Bounds on the sum capacity of synchronous binary cdma channels. *Journal of Chemical Education*, 55:3577–3593, 2009.

[3] T G Conley and D W Galeson. Nativity and wealth in mid-nineteenth century. *Journal of Economic History*, 58:468–493, 1998.

[4] S Waldron. Generalized welch bound equality sequences are tight frames. *IEEE Transactions on Information Theory*, 49:2017–2309, 2008.

[5] Richard E Fikes and Nils J Nilsson. STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. In *Proceedings of the 2Nd International Joint Conference on Artificial Intelligence*, IJCAI'71, pages 608–620, 1971.

[6] Weiguo Fan, Michael D Gordon, and Praveen Pathak. Personalization of Search Engine Services for Effective Retrieval and Knowledge Management. In *Proceedings of the Twenty First International Conference on Information Systems*, ICIS '00, pages 20–34, 2000.

[7] D H Holt. *Management Principles and Practices*. Prentice-Hall, Sydney, 1997.

[8] Philippe Aghion and Steven Durlauf, editors. *Handbook of Economic Growth*, volume 1. Elsevier, 1 edition, 2005.

[9] Dan Riley. *Industrial relations in Australian education / edited by Dan Riley*. Social Science Press [Wentworth Falls, N.S.W.], 1992.

[10] J P Hos. *Mechanochemically synthesized nanomaterials for intermediate temperature solid oxide fuel cell membranes*. PhD thesis, University of Western Australia, 2005.

[11] A H Cookson. Particle trap for compressed gas insulated transmission systems, 1985. US Patent 4554399.

[12] J Ionesco. Federal Election: New Chip in Politics. *The Advertiser*, page 10, 2010.