International Journal of Scientific Research in Science, Engineering and Technology
Print ISSN: 2395-1990 | Online ISSN : 2394-4099 (www.ijsrset.com)
doi : https://doi.org/10.32628/IJSRSET

# Software Failure Prediction System

**Chande. Y[1], Borude. S[2], Kate. N[3]**

[1,2]Associate professor , HSBPVT College of Engineering, Kashti, Maharashtra, India

[3]HSBPVT College of Engineering, Kashti, Maharashtra, India

**Article Info**
Volume 8, Issue 4
Page Number: 35-39

**Publication Issue :**
July-August-2021

**Article History**
Accepted : 02 July 2021
Published: 05 July 2021

## ABSTRACT

Software Defect Prediction [SDP] plays an important role in the active research areas of software engineering. A software defect is an error, bug, flaw, fault, mistake in software that causes it to create wrong or unexpected outcome. The major risk factors related with a software defect which is not detected during the early phase of software development are time, quality, cost, effort and wastage of resources. Defects may occur in any phase of software development. Booming software companies focus concentration on software quality, particularly during the early phase of the software development .Thus the key objective of any organization is to determine and correct the defects in an early phase of Software Development Life Cycle [SDLC]. To improve the quality of software, datamining techniques have been applied to build predictions regarding the failure of software components by exploiting past data of software components and their defects. Finally, in this study we discovered the application of machine learning on software defect management and prediction.

**Keywords :-** Softwear defect prediction, Softwear defect management , Softwear quality , Machine learning.

## I. INTRODUCTION

Software failure in system through time it automatically leads to software defect. Software defect are an error that are introduced by software developer and stakeholders. The main objective of software defect prediction is to improve the quality, minimized cost and time of software products. Software defect is also referred to as bug can be defined as shortage in the software product that causes the software not to perform its task as the programmer and customer needed. Machine Learning is one of the most vital and motivating area of research with the objective of finding meaningful information from huge data sets. The basic purpose of machine learning is to extract useful pattern from the data, mining data may be structured format (example. multiple data base) or text mining: unstructured data. In this study, we deeply observed the major factor of software failures that lead the software company to software defect, consume cost and times to test and maintenance after delivered to the stakeholders. In addition, we examine the recommended solutions to software failures, machine learning concepts and application of machine learning on software

engineering. Machine Learning is one of the most vital and motivating areas of research with the objective of finding meaningful information from huge data sets. The basic purpose of machine learning is to extract a useful pattern from the data, mining data may be structured format (example. multiple databases) or text mining: unstructured data (example, natural language document). In this study, we deeply observed the major factor of software failures that lead the software company to a software defect, consumer cost and times to test and maintenance after delivered to the stakeholders. In addition, we examine the recommended solutions to software failures, machine learning concepts.

## II. METHODOLOGY

Input data information

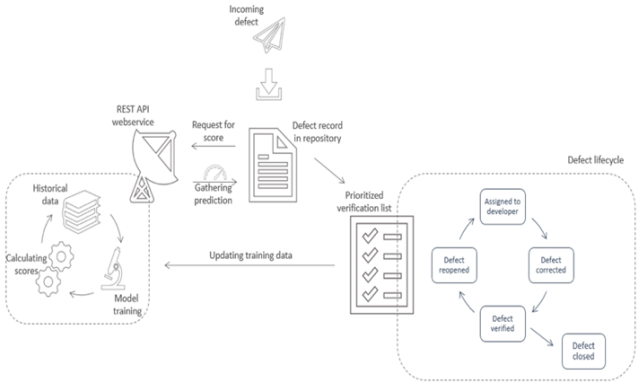


Figure 1: Overall system methodology

Each reported bug contains set of fields that must be filled out by bug creator, bug resolver (software developer), project area, state, Resolution, creation date, type, severity, Title, bug validator (test engineer), priority, etc. Those fields are used to create training data (used to train a model) as well as scoring data (used to get prediction). Figure 1: Overall system methodology.

Creation of training data set Each bug verified by test engineer has information that tells us if a bug has been fixed (the "Status" field). "Status" field is our target column we want to predict this field value [false, true] for not verified bugs. Verified bugs are used as training data since the target is known here. This is called supervised learning since we provide (supervise) learning process by providing target values in the training data set. Based on data gathered from verified bugs we create training data set consisting of feature columns and target column.

## III. MODELING AND ANALYSIS

INPUT DATA SET MODELING


Figure 2 : Input data set modeling.

CERATION OF TRANING DATA SET MODELING

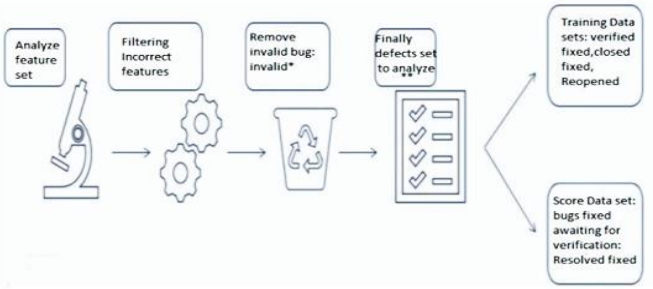


Figure 3 : Training data set modeling

## IV. RESULTS AND DISCUSSION

In this section, we validate our Bug reopen predictor. Experimental environment performance of Reopen-Predictor is pycharm IDE and Python 3.7 .

I would like to investigate, whether different numbers of selected features affect the I deal, since users don't know how to choose the best number of

selected features for a new dataset, the performance of Bug Reopen Predictor should be relatively stable for different numbers of selected features, as long as they are within a reasonable range.

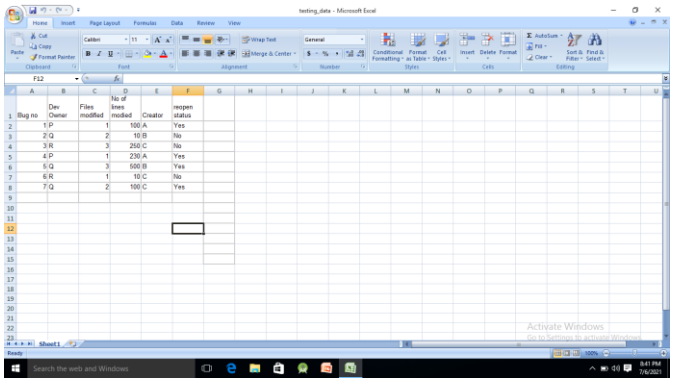

Figure : 5 Sample Bug data in Exel Sheet

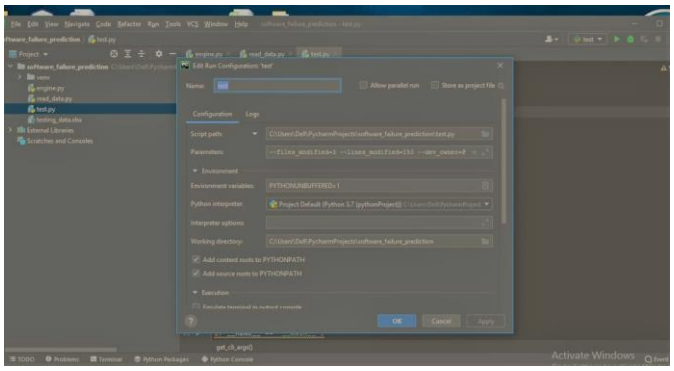

Figure : 6 Parameter Arguments

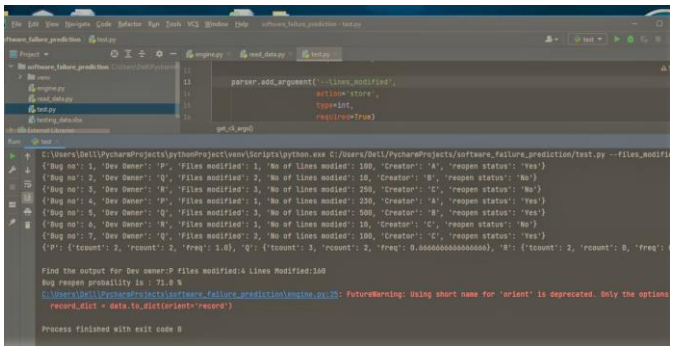

Figure : 7 Bug reopen probability result 1

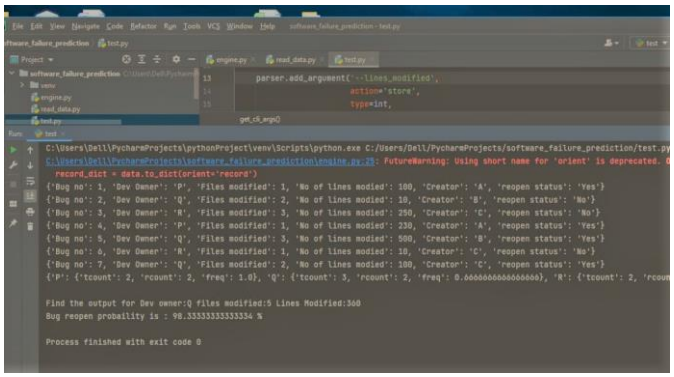

Figure : 8 Bug reopen probability result 2

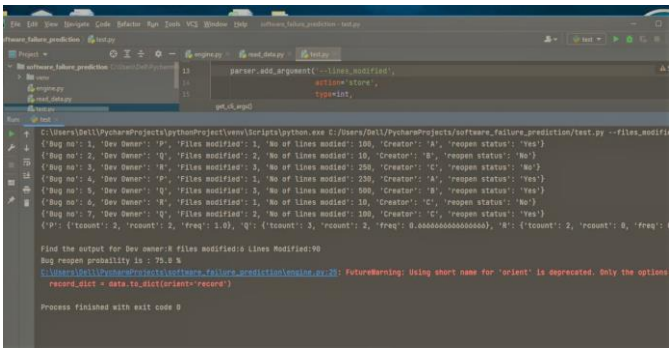

Figure : 9 Bug reopen probability result 3

Save prediction results in bug record:

Predicion result as well as probability are being saved in bug records as two additional fields. That allows to easily sort and validate the list of bugs awaiting verification.

Figure 10. Scrrenshot of evaluation events chart.

## V. CONCLUSION

Now a day the development of software based system are increasing from the previous years due to its benefit. However, the quality of the system is required before it is delivered to end users. Software defects prediction can effectively improve the efficiency of software testing and guide the allocation of resources and also improve the quality of the software development by predicting software fault at early stage of life cycle itself. In this project to predict the software faults a various machine learning technique and data mining technique can be used.

## VI. ACKNOWLEDGEMENTS

It gives us great pleasure in presenting the preliminary project report on 'Software Failure Prediction' I would like to take this opportunity to thank my internal guide Prof.P.M.Surywanshi. for giving me all the help and guidance I needed. I am really grateful to them for their kind support. Their valuable suggestions were very helpful. I am also grateful to Prof. Hiranwale S.B., Head of Computer Engineering Department,Parikrama College Of Engineering for his indispensable support, suggestions.At the end our special thanks to Supporting Staff and Lab Assistant for providing various resources such as laboratory with all needed software platforms, continuous Internet connection for Our Project.

**Complete Solution and integration with Bug tool: -**

1. Based on historical data (i.e. previously verified bug) create training data set.
2. Train predictive model using Enhanced classification (i.e. Naïve's Bayes, SVM etc.) algorithm that meets defined quality criteria.
3. Model should be exposed as an online REST service, so we can easily call it and make prediction requests for new incoming defects.
4. Bug records that are stored in code repository, are automatically updated with scoring result: prediction score/accuracy score.
5. Bugs should be sorted by prediction probability/accuracy score, so QA engineer could start testing bugs with highest probability of incorrect fix.
6. If new bugs are successfully verified and label value is known (correctly fixed / incorrectly fixed), such records are marked as new training data and stored in feedback data store.
7. Feedback data store is used to evaluate served model quality, automatically retrain deployed model, and finally to re-deploy new model version. The goal of such continuous learning system, is to ensure the highest possible quality of exposed model.

## VII. CONCLUSION

Now a day the development of software based system are increasing from the previous years due to its benefit. However, the quality of the system is required before it is delivered to end users. Software defects prediction can effectively improve the efficiency of software testing and guide the allocation of resources and also improve the quality of the software development by predicting software fault at early stage of life cycle itself. In this project to predict the software faults a various machine learning technique and data mining technique can be used.

## VIII. ACKNOWLEDGEMENTS

It gives us great pleasure in presenting the preliminary project report on 'Software Failure Prediction' I would like to take this opportunity to thank my internal guide Prof. P. M. Surywanshi. for giving me all the help and guidance, I needed. I am really grateful to them for their kind support. Their valuable suggestions were very helpful. I am also grateful to Prof. Hiranwale S.B., Head of Computer Engineering Department, Parikrama College Of Engineering for his indispensable support, suggestions. At the end our special thanks to Supporting Staff and Lab Assistant for providing various resources such as laboratory with all needed software platforms, continuous Internet connection for Our Project.

## IX. REFERENCES

[1]. Grishma, B. R., and C. Anjali. "Software root cause prediction using clustering techniques: A review." Communication Technologies (GCCT), 2015 Global Conference on. IEEE, 2015.

[2]. He, Peng, et al. "An empirical study on software defect prediction with a simplified metric set." Information and Software Technology 59 (2015): 170-190.

[3]. Chug, Anuradha, and Shafali Dhall. "Software defect prediction using supervised learning algorithm and unsupervised learning algorithm." (2013): 5-01.

[4]. Nam, Jaechang, et al. "Heterogeneous defect prediction." IEEE Transactions on Software Engineering (2017).

[5]. Perreault, Logan, et al. "Using Classifiers for Software Defect Detection." 26th International Conference on Software Engineering and Data Engineering, SEDE. 2017.

[6]. Felix, Ebubeogu Amarachukwu, and Sai Peck Lee. "Integrated Approach to Software Defect Prediction." IEEE Access 5 (2017): 21524-21547.

[7]. Li, Yong, et al. "Evaluating Data Filter on Cross-Project Defect Prediction: Comparison and Improvements." IEEE Access 5 (2017): 25646-25656.

[8]. Yu, Xiao, et al. "Using Class Imbalance Learning for Cross-Company Defect Prediction." 29th International Conference on Software Engineering and Knowledge Engineering (SEKE 2017). KSI Research Inc. and Knowledge Systems Institute, 2017.

[9]. Huda, Shamsul, et al. "A Framework for Software Defect Prediction and Metric Selection." IEEE Access (2017).

[10]. Ni, Chao, et al. "A Cluster Based Feature Selection Method for Cross-Project Software Defect Prediction." Journal of Computer Science and Technology 32.6 (2017): 1090- 1107.

[11]. Zimmermann, Thomas, et al. "Cross-project defect prediction: a large scale experiment on data vs. domain vs. process." Proceedings of the the 7th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering. ACM, 2009.

[12]. Laradji, Issam H., Mohammad Alshayeb, and Lahouari Ghouti. "Software defect prediction using ensemble learning on selected features." Information and Software Technology 58 (2015): 388-402.

[13]. Rajbahadur, Gopi Krishnan, et al. "The impact of using regression models to build defect classifiers." Proceedings of the 14th International Conference on Mining Software Repositories. IEEE Press, 2017.

[14]. Yang, Xinli, et al. "TLEL: A two-layer ensemble learning approach for just-in-time defect prediction." Information and Software Technology 87 (2017): 206-220.

[15]. Turhan, Burak, et al. "On the relative value of cross-company and within-company data for defect prediction." Empirical Software Engineering 14.5 (2009): 540-578. International Journal of Pure and Applied Mathematics Special

**Cite this article as :**

Chande. Y, Borude. S, Kate. N, "Software Failure Prediction System", International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), Online ISSN : 2394-4099, Print ISSN : 2395-1990, Volume 8 Issue 4, pp. 35-39, July-August 2021.
Journal URL : https://ijsrset.com/IJSRSET2183302