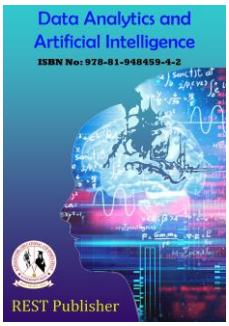
Data Analytics and Artificial Intelligence

Vol: 3(2), 2023

REST Publisher; ISBN: 978-81-948459-4-2

Website: http://restpublisher.com/book-series/daai/

DOI: https://doi.org/10.46632/daai/3/2/7



Software Defect Prediction Using Machine Learning Techniques

*** G.Cauvery, Dhina Suresh**

St. Joseph college of arts and science for women, Hosur, Tamil Nādu, India.

*Corresponding Author Email: cauverysrini308@gmail.com

Abstract. *Software defect prediction provides development groups with observable outcomes while contributing to industrial results and development faults predicting defective code areas can help developers identify bugs and organize their test activities. The percentage of classification providing the proper prediction is essential for early identification. Moreover, software- defected data sets are supported and at least partially recognized due to their enormous dimension.*

Keywords: *defect prediction, machine learning methods, quality software.*

1. INTRODUCTION

Nowadays developing a software system is a difficult process that involves planning, analyzing, designing, implementing, testing, in targeting, and maintenance. A software engineer's work is developing a system in time with a limited budget which is done in the planning phase. While doing the development process we can have a few defects like not proper design, where the logic is poor, data handling is improper, etc. and these defects cause errors which lead to re-do the work, increasing in development and cost of maintenance. These all are responsible for the decrease in customer satisfaction. In this point of view, faults are grouped based on sternness, corrective and advance actions are taken as per the sternness defined. Problem Statement In the past decade, humans have progressively focused on software-based systems in which software quality is regarded as the most critical element in user functionality. Because of the vast production of application software, software quality remains an unresolved problem that gives inadequate output for industrial and private applications. Designs of defect prediction are commonly utilized by industries and Such models help in predicting faults, estimating effort testing software reliability, hazard analysis, etc. during the growth stage. Proposed Work: We propose an efficient system, which belongs to the traditional machine learning concept. A dataset is used to train the model hence less execution time and also yield efficient results. One of typically using traditional machine algorithm. Advantages: More Efficient, High accuracy.

2. LITERATURE SURVEY

Integrated Approach to Software Defect Prediction: Software defect prediction provides action able outputs to software teams while contributing to industrial success. Empirical studies have been conducted on software defect prediction for both cross project and within-project defect prediction. However, existing studies have yet to demonstrate method of predicting the number of defects in an upcoming product release. This paper presents such a method using predictor variables derived from the defect acceleration, namely, the defect density, defect velocity, and defect introduction time, and determines the correlation of each predictor variable with the number of defects. We report the application of an integrated machine learning approach based on regression models constructed from These predict or variables. An experiment was conducted on ten different data sets collected from the PROMISE repository, containing 22838 instances. There gression model constructed as a function of the average defect velocity achieved an adjusted R-square of 98.6%, with a *p*-value of < 0.001. The average defect velocity is strongly positively correlated with the number of defects, with a correlation coefficient of 0.98. Thus, it is demonstrated that this technique can provide a blue print for program testing to enhance the effectiveness of software development activities.

A Review on Software Defect Prediction Techniques Using Product Metrics: Presently, complexity and volume of software systems are increasing with a rapid rate. In some cases it improves performance and brings efficient outcome, but unfortunately in several situation sit leads to elevated cost for testing, meaning less outcome and inferior quality, even there is no trust worthiness of the products. Fault prediction in software plays a vital role in enhancing the software excellence as well asit helps in software testing to decrease the price and time. Conventionally, to describe the difficulty

Copyright@ REST Publisher

and calculate the duration of the programming, software metrics can be utilized. To forecast the amount of faults in module and utilizing software metrics, an extensive investigation is performed. With the purpose of recognizing the cause's whichimportantlyenhancesthefaultpredictionmodelsrelatedtoproductmetrics, thisempirical research is made. This paper visitsvarioussoftwaremetricsandsuggestedproceduresthroughwhichsoftwaredefectprediction is enhanced and also summarizes those techniques.

3. SYSTEM DESIGN AND MODELING

System Architecture design-identifies the overall hyper media structure for the Web App. Architecture design is tied to the goals establish for a Web App, the content to be presented, the users who will visit, and then aviation philosophy that has been established. Content architecture, focuses on the manner in which content objects and Structured for presentation and navigation. Web App architecture, addresses the manner in which the application is structure to manage user interaction, handle internal processing tasks, effect navigation, and present content. Web App architecture is defined within the context of the development environment in which the application is to be implemented.

Flowchart: It is important to complete all tasks and meet deadlines. There are many project management tools that are available to help project managers manage their tasks and schedule and one of the mis the flowchart. A flow chart is one of the seven basic quality tools used in project management and it displays the actions that are necessary to meet the goals of a particular task in the most practical sequence. Also called as process maps, this type of tool displays a series of steps with branching possibilities that depict one or more inputs and transforms them to outputs. The advantage of flowcharts is that they show the activities involved in a project including the decision points, parallel paths, branching loops as well as the overall sequence of processing through mapping the operational details within the horizontal value chain. Moreover, this particular tool is very used in estimating and understanding the cost of quality for a particular process.

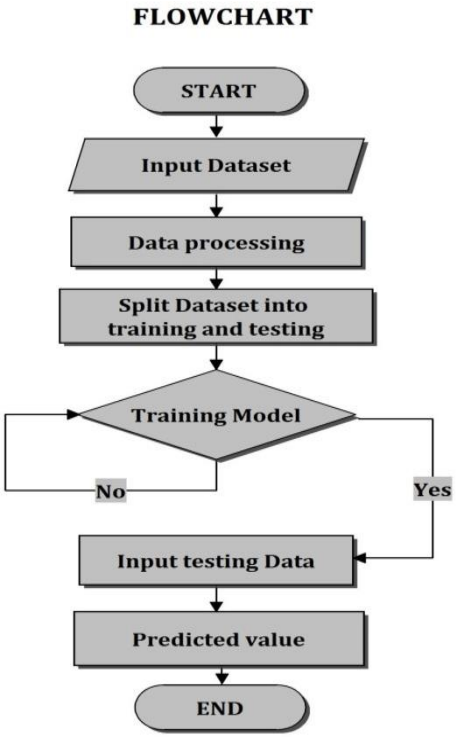


FIGURE 1. Flowchart

4. MODULES

Data collection and preprocessing: In this module data are uploading in top and as data frame and it will enter into pre-processing to correct the missing data.

Training data and testing data split: Once data are preprocessed this system has to split data into division training data and testing data. Usually training should be large for accurate result.

Copyright@ REST Publisher

Training process methodology: In this method the training data set with label has give to any one of the machine learning technique like random forest, this module will extract the feature from the label data keep it ready prediction process.

Prediction methodology: In this method the test data without label has to give prediction model which generate using training method this prediction module accept the test data and process. Finally it will give the prediction for Features.

Data visualization: This method uses matplotlib python tool for producing graph from training data as well as testing data set.

5. ANALYSIS

In this final phase, we will test our classification model on our prepared data set and also measure the performance on our dataset. To evaluate the performance of our created classification and make it comparable to current approaches, we use accuracy to measure the effectiveness of classifiers. After model building, knowing the power of model prediction on a new instance, is very important issue. Once a predictive model is developed using the historical data, one would be curious as to how the model will perform on the data that it has not seen during the model building process. One might even try multiple model types for the same prediction problem, and then, would like to know which model is the one to use for the real-world decision making situation, simply by comparing them on their prediction performance (e.g., accuracy). To measure the performance of a predictor, there are commonly used performance metrics, such as accuracy, recall etc. First, the most commonly used performance metrics will be described, and then some famous estimation methodologies are explained and compared to each other. "Performance Metrics for Predictive Modeling In classification problems, the primary source of performance measurements is a coincidence matrix (classification matrix or a contingency table)". Above figure shows a coincidence matrix for a two-class classification problem. The equations of the most commonly used metrics that can be calculated from the coincidence matrix are also given in Figure 2.

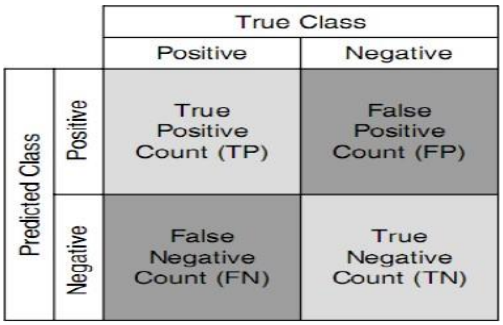


FIGURE 2. coincidence matrix

$$\text{True Positive Rate} = \frac{TP}{TP + FN}$$

$$\text{True Negative Rate} = \frac{TN}{TN + FP}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

FIGURE 3. Confusion matrix and formulae

As being seen in above figure, the numbers along the diagonal from upper-left to lower-right represent the correct decisions made, and the numbers outside this diagonal represent the errors. "The true positive rate (also called hit rate or recall) of a classifier is estimated by dividing the correctly classified positives (the true positive count) by the total positive count. The false positive rate (also called a false alarm rate) of the classifier is estimated by dividing the incorrectly classified negatives (the false negative count)by the total negatives. The overall accuracy of a classifier is estimated by dividing the total correctly classified positives and negatives by the total number of samples. The architecture of a convnet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in are stricted region of the visual field known as the Receptive Field. A collection of such fields over lap to cover the entire visual area.

Copyright@ REST Publisher

Flexibility: Sometimes we just don't want to use what is already there but we want to define something of our own (for example a cost function, a metric, a layer, etc.). Although Keras2 has been designed in such a way that you can implement almost everything you want but we all know that low-level libraries provide more flexibility. Same is the case with TF. You can tweak TF much more as compared to Keras.

Functionality: Although Keras provides all the general purpose functionalities for building Deep learning models, it doesn't provide as much as TF. Tensor Flow offers more advanced operations as compared to Keras. This comes very handy if you are doing a research or developing some special kind of deep learning models.

Threading and Queues: Queues area powerful mechanism for computing tensors asynchronously in a graph. Similarly, you can execute multiple threads for the same Session for parallel computations and hence speed up your operations.

Debugger: Another extra power of TF. With Tensor Flow, you get a specialized debugger. It provides visibility into the internal structure and states of running Tensor Flow graphs. Insights from debugger can be used to facilitate de bugging of various types of bugs during both training and inference.

Control: The more control you have over your network, more better understanding you have of what's going on with your network. With TF, you get such a control over your network. You can control what every ouwant in your network. Operations on weights or gradients can be done like a charm in TF.

Numpy: Numpy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed. This tutorial explains the basics of NumPy such as its architecture and environment. It also discusses the various array functions, types of indexing, etc. An introduction to Matplotlib is also provided. All this is explained with the help of examples for better understanding. Numpy is a Python package. It stands for 'Numerical Python'. It is a library consisting of multidimensional array objects and a collection of routines for processing of array.

Numeric: the ancest or of NumPy, was developed by Jim Hugunin. Another package Numarray was also developed, having some additional functionality. In 2005, Travis Oliphant created NumPy package by incorporating the features of Num array into Numeric package. There are many contributors to this open source project.

Operations using Num Py Using Num Py, a developer can perform the following operations.

- Mathematical and logical operations on arrays.
- Fourier transforms and routines for shape manipulation.
- Operations related to linear algebra. Num Py has in-built functions for linear algebra and random number generation.

Num Py A Replacement for Mat Lab. Num Py is often used along with packages like SciPy and Matplotlib (plotting library).This combination is widely used as a replacement for Mat Lab, a popular plat form for technical computing.

6. CONCLUSION

The main goal of this research is to forecast software problems using information-mining techniques. Additionally, this area has become an important area of research where a variety of techniques have been investigated to improve the effectiveness of identifying software flaws or fore seeing vulnerabilities. The problem of classification accuracy forlargedatasetswashandledwithusingfeaturereductionsandclassification.Forthepurpose of identifying program defects, the neural network classification method is employed. The results of the research show that the suggested strategy is effective. If there is no approach for selecting features and anytime the strategies for selecting features are used, there is no discernible variance in classifier accuracy. While there is no set of features and when the methods for choosing the function are being employed, there is a gap in the classifiers accuracy. As a result, it is seen that applying feature selection strategies reduces the time and space difficulties for defect prediction without lowering prediction accuracy.

REFERENCES

[1]. Jayanthi, R. and Florence,L.,2019.Software defect prediction techniques using metrics based on neural network classifiers. Cluster Computing, 22(1), pp.77-88.

[2]. Felix, E.A. and Lee,S.P.,2017.Integratedapproachtosoftwaredefectprediction.IEEEAccess,5, pp.21524-21547.

[3]. Wang, T., Zhang, Z., Jing, X., Zhang, L.: Multiple kernel ensembles learning for software defect prediction. Autom. Softw. Eng.23, 569–590 (2015).

[4]. Xu,Z., Xuan,J., Liu,J., Cui,X.: MICHAC: defect prediction via feature selection based on maximal information coefficient with hierarchical agglomerative clustering. In: 2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Re engineering (SANER), Suita, pp. 370–381 (2016).

Copyright@ REST Publisher