Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 6.017

IJCSMC, Vol. 6, Issue. 3, March 2017, pg.148 – 153

A Review of Improving Software Quality using Machine Learning Algorithms

Jyoti Devi
CSE Department, Baddi University, HP, India
Jyotibandhral108@gmail.com

Nancy Seghal
Assistant Professor, CSE Department, Baddi University, HP, India
Nancy.sehgal@baddiuniv.ac.in

***Abstract:* Software is a process and maintains continuous change to improve the functionality and effectiveness of the software quality. During the life cycle of software various problems arises like advanced planning, well documentation and proper process control. This problem may result in not achieving the software quality as desired. With respect to competition in the market it is necessary to remove this problem with the help of software engineering. In this paper software prediction model used to control the classes of software which are often to change. Machine learning algorithms are used for predicting software.**

Keywords: software quality, receiver operating characteristics (roc), software defect prediction.

INTRODUCTION

Software quality:

Quality software is reasonably bugs or defects free, delivered on time and within budget, meets requirements and/or expectations, and is maintainable.

Software quality means a degree to which software or a process meets customer and user needs or expectations. The summarized software to deal with properties such as complexity is cyclic in nature, cohesion i.e. the properties of function which meets customers of the needs and provide product satisfaction. The software attributes are categorized two type's internal and external quality in fig 1.

The software attributes are categorized two type's internal and external quality in fig 1 .The internal quality like efficiency, maintainability, testability, flexibility, reusability etc. The external quality like integrity, usability, reliability and accuracy etc.

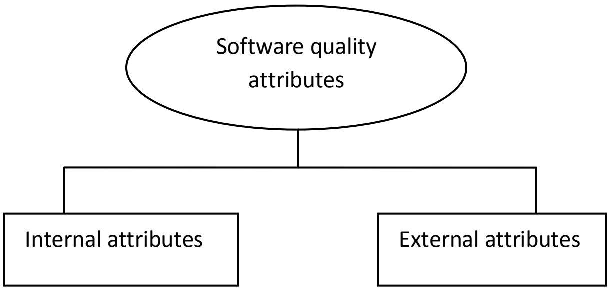

Fig:1

Software quality management activities:

Software quality management is a management process that aims to develop and manage the quality of software to make sure the product satisfies the user. The goal of software quality management is to make sure the product follows all rules and meets the quality standards by the customer.

Software quality management split into three main activities:

- Quality assurance
- Quality planning
- quality control

Software quality management split into three parts like a quality assurance, quality planning and quality control. There are managing the software quality.

Software quality assurance:

Software quality assurance is outlined as a planning and systematic method to analysis of quality, software product necessities, procedures and processes. Software quality assurance contains system of assuring that description and techniques are based on the software development life cycle.

Quality factor is divided into three types:

- Reusability
- Portability
- Usability

Quality planning:

In the case of a software development quality plans develop for each software or system engineering project. A quality plan describes how an organization will achieve its quality objectives. It describes the quality objectives and specifies the quality assurance and control activities to be performed in day-to-day company operations. In the case of a software development organization individual quality plans may be prepared for each software or systems engineering project.

Quality control: Software Quality Control is the set of procedures used by organizations to ensure that a software product will meet its quality goals at the best value to the customer, and to continually improve the organization's ability to produce software products in the future.

Software metrics: Software metrics is measure of the degree to which software quality. The main objective, quantifiable and function able measurements. The multiple applications in estimation software debugging, planning, analysis and control performance, quality assurance testing. The prediction related to the change prone classes is very necessary in testing and maintenance of the software.

Machine learning:

In machine learning , data plays an important role, and the machine learning is used catch on and learn properties from the data. The learning and prediction performance will affect on the quantity and quality of data set.

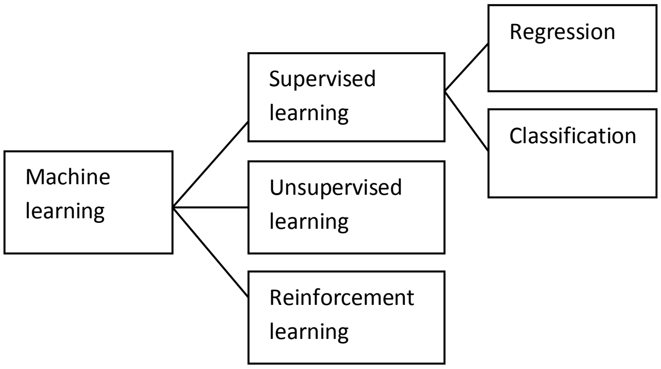

Fig: 2

Types of machine learning:

Supervised learning: Supervised learning is trained a labeled data. Supervised learning generally used in applications where historical data predict like future use. This technique further divided into two categories as regression and classification. In regression the label is continuous quantity. On the other hand, in classification the label is discrete

Unsupervised learning: Unsupervised learning used unlabeled data that has no historical labels.

Reinforcement learning: It is used for gaming, navigation and robotics. This type of learning has three types of primary components: the learner, the environment and actions.

Machine learning techniques:

A machine learning algorithms are developed to build machine learning models and important machine learning process. In this paper we discuss three classifier like decision tree, naïve bayes and support vector machine.

(1) **Decision tree**: decision tree based on supervised learning algorithm. It is one of the predictive model approach's used in machine learning, data mining and statistics.

In decision analysis, a decision tree can used to usually represent decision making and decisions. In data mining, a decision tree characterize data but not decision s, to some extent the resulting allocation tree can be an input for decision making.

There are many decision tree algorithm:

C4.5 (successor of ID3)

CART (classification and regression tree)

MARS: extends decision tree to better handle numerical data.

(2) **Naïve bayes:** Naïve bayes classifier is widely studied probabilistic learning method. Naïve Bayesian classifier conclude that there are no addiction among attributes. This presumption is called conditional independence.

Advantages of naïve bayes:

- It is used a very perceptive techniques.
- It widely studied expectation learning method.
- Naïve bayes classifier are computational fast when executing decision.

(3) Support vector machine (SVM): "Support Vector Machine" (SVM) is a supervised machine learning algorithm which can be used for both classification and regression challenges. However it is mostly used in classification problems. SVM used for classification of both linear and non-linear data.

In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate.

Support vector machine some steps:

- Set up the training data
- Set up of SVM parameters
- Types of SVM kernel
- Train the SVM
- Regions classified by SVMP
- Support vectors

Performance measures: Machine learning checks the prediction performance with the help of various performance measures:

- Precision
- Recall
- Accuracy
- F measure
- Roc(receiver operating characteristics)

There were calculated using the prediction classification confusion matrix table:

		PREDICTED	
		P	N
ACTUAL	P	TRUE POSITIVE (TP)	FALSE NEGTIVE (FN)
	N	FALSE POSITIVE (FP)	TRUE NEGATIVE (TN)

Table 3: confusion matrix

TP (true positive): Number of correct predictions that an instance is positive.

TN (true negative): Number of correct predictions that an instance is negative.

FN (false negative): Number of incorrect predictions that an instance is positive.

FP (false positive): Number of incorrect predictions that instance is negative.

Accuracy: The total number of predictions that were correct:

$$Accuracy(\%) = (TP + TN) \,/\, (TP + FP + FN + TN)$$

Precision: The predicted true pages those were correct:

$$Precision(\%) = TP \,/\, (TP + FP)$$

Recall: The predicted true pages that were correctly identify.

$$Recall(\%) = TP \,/\, (FN + TP)$$

F-Measure: Derives from precision and recall values:

$$FMeasure(\%) = (2 \times recall \times precision) \,/\, (recall + precision)$$

Literature Survey:

Kanika Chandra et.al (2016) explained time and cost spent on the testing and maintenance phase of the SDLC. Efforts have been put to rectify and judge the changes occur in the classes of the concerned software. Therefore software predictions models are used for the prediction of changes required in future. Analysis process is done to learn and stimulate the association between the object orientation specification and concept of changes proneness. Furthermore, testing is done on the data test to find out its available possibilities. Results of testing addressed two points. First is the quantification of parameters to improve the productivity, functionality and quality of the software and second is the predication machine learning technologies are used. The main concern of this paper is to evaluate and compare all the available learning techniques to enhance the result of the performance parameters. The proposed technique helps to reduce the testing time so that developer can predict the changing behavior of the software on the set of the classes and performed efficiently.[1]

Martin Shepperd et al. (2014) have discussed about the factors having largest effect on the predictive performance of the software by conducting a meta analysis of all relevant and high quality primary studies of defects prediction on software module. The experimental results showed that the major factor is researcher group instead of choice of classifier on the performance.[2]

C. SenthilMurugan and S. Prakasamthe(2013) explained the perceptions of Software Quality Assurance make the error-free Software are used and think on complex activities so that it can be completed in time and in cost estimation is prevented. As discussed within the earlier sections of this paper the application exceptional Assurance events, concepts, factors and its methods are applied within the early levels of program engineering progress phases, since of this endeavor the program developer get the expertise concerning the software what he is going to increase, it may just shrink the rework and failures of the software's. These days all of the program progress industries are enforcing the SQA aspect to get excellent software's and to satisfy all the necessities of the client and make a finite application.[3]

Seceleanu et al. (2013) has discussed the method in which the process design can be simplified. Component- established design method is used for setting up application. It's a method where an application process is developed through making use of commercial off the shelf (COTS) accessories and consequently raises the predictability and reuse of the program approach. A software design can also be simplified in this paper if we encapsulate both realistic and further practical attributes. This technique has a couple of challenges of complexity.[4]

Ming li et al. (2012) states that software quality can controlled by software defects prediction. The defect prediction techniques used currently are based on large amount of historical data but in case of new project and for many organizations historical data is often not available. In that case, sample based methods for defects can be used by selecting and testing a small percentage of module and after that build a defect prediction model to predict defect proneness of other modules.[5]

David gray et al: (2012) have explained the reason of significant preprocessing of data set for suitability of defect prediction. Researcher need to analyze the data that how it will be used by removal of constant attributes, repeated attributes, missing values and inconsistent instances. The experiments that have been used are based upon NASA metrics data program that result in errors findings and conclude that errors are mainly because of repeated data points.[6]

Nikolik et al. (2012) has mentioned concerning the financial metrics. The fundamental points taken into consideration for assuring pleasant are artifact defect and artifact test cases. The concept of scan circumstances is used which is a method applied to artifact in order to obtain specific outcome. Defects are treated by way of making use of three procedures prevention, defect removal, defect avoidance. The experiment case rate price is used to calculate ROI which is an efficiency measure used for evaluating effectively. In this paper, case study is performed to determine the change in test case with respect to changes in artifact. It is concluded that for maximizing the test case value and ROI and minimizing the test case cost. In order to get more practical knowledge about the metrics experimentation is needed on large industrial projects. In order to make defect avoidance possible more experimentation is needed. Experimentation on economic release criteria is also needed.[7]

Qinbao song et al.(2011) Describe the framework that comprises schema evaluation and defect prediction components. Analyzing the prediction performance for the given historical data set is done by schemes evaluation and defect predictor constructs models according to the evaluated learning schemes and predicts defects of the software with new data according to the defined constructed model.[8]

Kenettet.al (2010) explores the quality concepts. This paper describes an extended quality conceptual framework which represents an extension of software quality framework. Two fundamental concepts discussed in this paper are assuring the quality and testing the product. The main aim is to place quality in proper prospective in relation to acquisition and development of computer software. Various activities are performed to assure quality which includes establishing requirements and controlling the changes, establishing method of implementation and achieving specified product quality and finally evaluating process and product quality. These three concepts are explored in detail. In order to characterize the extended quality framework a set of definitions and related concepts are first specified and explained in detail. The product quality is specified by using product attributes definitions forms the basis for establishing quality requirements, methods to help satisfy these requirements, and quality evaluation.[9]

Conclusion: In software development life cycle, maximum effort and cost is consumed on the testing and maintenance. In this paper we studied the machine learning techniques like naïve bayes, decision tree and support vector machine. This algorithms used for improving the software quality. Further different machine learning algorithm used to improve software quality and improve the performance in terms of precision, recall and Roc (receiver operating characteristics).

REFERENCES:

[1]. K. Chandra, G. Kapoor, R. Kohli and A. Gupta, "Improving software quality using machine learning," 2016 International Conference on Innovation and Challenges in Cyber Security (ICICCS-INBUSH), Noida, 2016, pp. 115-118.

[2]. Shepperd, Martin, David Bowes, and Tracy Hall. “Researcher bias: The use of machine learning in software defect prediction.” Software Engineering, IEEE Transaction on 40.6(2014):6303-616.

[3]. C. SenthilMurugan and S. Prakasamthe, “A Literal Review of Software Quality Assurance”, International Journal of Computer Applications (0975 – 8887) Volume 78 – No.8, September 2013

[4]. Seceleanu, C, Crnkovic, , “ Component Models for Reasoning” Computer , vol.46, no.11, pp.40,47, November, 2013

[5]. Nikolik, B, “Software quality assurance economics”, Information and Software Technology, 54(11), 1229-1238, 2012

[6]. Li, M., Zhang, H., Wu, R., & Zhou, Z. H. “ Sampled based software defects prediction with active and semi-supervised learning.” Automated Software Engineering 19.2 (2012): 201-230.

[7]. Grey, David, et al. “The misuse of the nasa metrics data program data sets for automated software defect prediction.” Evaluation & Assessment in Software Engineering (2011): 603-616.

[8]. Song, QinbaoJia, Z., Sheppered M., Ying, S., & Liu, J. “A general software defect-proneness prediction framework.” Software Engineering, IEEE Transaction on 37.3(2011): 356-370.

[9]. Kenett, R. S., & Baker, E., “Process Improvement and CMMI® for Systems and Software”, CRC Press, 2010

[10]. Côté, Marc-Alexis, WitoldSuryn, and Elli Georgiadou. "In search for a widely applicable and accepted software quality model for software quality engineering." *Software Quality Journal* 15.4 (2007): 401-416.

[11]. Xu Lai &SjaakBrinkkemper, “Concepts of Product Software: Paving the Road for Urgently Needed Research,” Technical report, Institute of Information and Computing Sciences, Utrecht University, The Netherlands. European Journal of Information Systems 16, 531–541, 2007.

[12]. Jin Huang Jingjing Lu & Charles X. Ling “Comparing naïve bayes ,decision tree and SVM with AUC and Accuracy” The third IEEE International Conference on data mining,2003.

[13]. Lie Zhang, Fuzong Lin, Bo Zhang, “Support vector machine learning for image retrieval” State key Laboratory of intelligent Technology and systems pp.7803-6725, 2001.