# Software Defect Prediction Analysis Using Machine Learning Algorithms

Sonu Kumar
Delhi Technological University
New Delhi, India

sonukumar_2k19it124@dtu.ac.in

Soham Bhardwaj
Delhi Technological University
New Delhi, India

sohambhardwaj_2k19it122@dtu.ac.in

*Abstract*— **Software Quality is the most important aspect of a software. Software Defect Prediction can directly affect quality and has achieved significant popularity in last few years. Defective software modules have a massive impact over software's quality leading to cost overruns, delayed timelines and much higher maintenance costs. In this paper we have analyzed the most popular and widely used Machine Learning algorithms - DT (Decision Trees), NB (Naïve Bayes) and LC (Linear classifier) Random Forest Classifier (CLF). The four algorithms were analyzed using python language and validated using k-fold cross validation technique. Datasets used in this research were obtained from open source NASA Promise dataset repository. Three datasets were selected for defect prediction analysis. Classification was performed on these 3 datasets and validated using 10 fold cross validation. The results demonstrated the dominance of Linear Classifier and Random Forest Classifier over other algorithms in terms of defect prediction accuracy.**

***Keywords—Software Quality, Defect Prediction, Machine Learning, NASA Promise dataset Introduction.***

## I. INTRODUCTION

Software Quality is the most important aspect of a software. It is the degree of accordance to both explicitly defined and implied requirements or expectations of a customer in a software program. It has a huge impact over business as it can glorify or ruin the brand image of a company. If not handled properly, it can lead to cost overruns, delayed timelines and much higher maintenance costs [1]. If defect occurs after the software goes "live" i.e., after the product is released publicly, entire defective module has to be re-examined for bug(s) and the code is altered to fix them as part of maintenance. Then the ripple effect is taken care in the form of Regression Testing. As per International Software Testing Qualifications Board (ISTQB), As per International Software Testing Qualifications Board (ISTQB), Quality is "The degree to which a component, system or process meets specified requirements and/or user/customer needs and expectations". It defines Software Quality as "The totality of functionality and features of a software product that bear on its ability to satisfy stated or implied needs". [Standard glossary of terms used in Software Testing, ISTQB V 2.2]. Software Quality is inversely proportional to the density of defects. As per ISTQB, Defect is "A flaw in a component or system that can cause the component or system to fail to perform its required function". If it is encountered during execution of the program, it may cause failure of the component or even the whole system. Defects are nightmares for deemed organizations. It impacts the reputation of the organization leading to customer dissatisfaction which further effects the market hold of the company. Neither any software can be 100% defect-free nor 100% testing can be achieved but a good quality software has much less number of defects and is more reliable [4] Many real-world problems, often highly complex in nature, need to be handled with customized" algorithms according to their need. Inventing specialized algorithms to handle such problems every time is unrealistic, if not impossible. Machine Learning Algorithms make it possible to handle such problems in a customized manner. As per the definition of Machine Learning given by Tom Mitchell of Carnegie Mellon University, "A computer program is said to learn from experience E with respect t o some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E". There are two types of machine learning algorithms, Supervised machine learning algorithms and Unsupervised machine learning algorithms. Supervised machine learning algorithms are fed with predefined set of training data. The algorithms then gains "experience" from training dataset and generates rules to predict the class label for new set of data."Learning" phase consists of using mathematical algorithms to produce and improve the predictor function. The training data used in this phase have an input value of an attribute and its known output value.

The predicted value of ML algorithm is compared with already known output. It is then "modified" to make it accurate with respect to the prediction output. This is repeated in several "generations" of training data until a threshold prediction accuracy is obtained or maximum number of cycles are performed In case of unsupervised machine learning algorithms, the output value of the class label in data is not known. Instead, the program is loaded with a cluster of data and the algorithm finds patterns and relationships within it. The main focus is on relationships amongst the attributes within data. An example would be identifying a circle of friends in a social networking website. The identification of defect prone modules are prioritized higher in testing phase of SDLC and the non-defect prone modules are tested as the time and cost allows. The classification function, known as the classifier, analyzes the relationship between the attributes and the class label of the training dataset and forms classification rules. These rules are then used in identifying the class labels of future datasets. Thus, we can classify the unknown datasets with the help of classification rules and a classifier.

Research questions which we aim to answer in this paper:

RQ1 **:** Which data sets are good to use in SDP studies?

RQ2 : Which method gives hig hest accuracy amongst the selected methods?

## II. RESEARCH METHODOLOGY

Various algorithms were studied for doing the comparative analysis research. The selection was made such that it covers most popular and various type of Machine Learning algorithms. Then statistical test are performed for the significance of results, the test which are used are Friedman which is followed by wilcoxon signed rank test. Then various plots are made for illustration of results. Following are the selected algorithms along with their brief description:

### *A. Random Forest Classifier*

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees.

Operates by constructing a multitude of decision trees at training time . For classification tasks, the output of the random forest is the class selected by most trees

### *B. Decision Trees*

Decision Trees [5], in comparison to neural networks, represent only rules. The rules formed in this approach are easy to comprehend for the human brain. They can also be directly utilized in database languages like SQL to retrieve records falling into a specific class. Decision tree permits calculations to be d one in forward and reverse manner which enhance decision correctness. Decision tree has a tree structure where every node is either a Leaf node or a Decision node. Leaf nodes display the value of the attribute assigned to them whereas decision nodes indicate branching, where a specific test has to be done on attribute with one branch and sub-tree as a result of the test.

This learning technique uses decision tree as a predictive model in which item's target value conclusion is mapped with observation about the item. This is a predictive modeling technique which is used in data mining, statistics and machine learning. A decision tree can be utilized to visually classify the decision making and the decisions. The goal of this algorithm is to build a structure table that is able to predict value of target variable relying on various input variables. Every interior node represents one of the input variables. The goal of this algorithm is to build a structure that is able to predict value of target variable relying on various input variables. Every interior node represents one of the input variables. For each possible value of these variables, there are edges to their children. Leaf in the tree represents target variable value in which the given values of the input variables can be traversed by path from root to the leaf.

### *C. Naïve Bayes classifier*

Naïve Bayes classifier [3] is an administered learning algorithm which is utilized for information grouping utilizing statistical strategy. This is a probabilistic classifier that characterizes particular input over an arrangement of classes utilizing mathematical probability distribution. As the name suggest Naïve this method innocently assumes that attributes of a given class are independent. Grouping is then finished by applying Bayes theorem to calculate the probability of the right class given the specific attributes of a situation.

These come from a simple probabilistic classifier family which is based on the theorem of Bayes. Features have an assumption of strong or naïve independence. It is a simple technique to build classifiers. It acts like a model which is assigned to problem objects as a class labels, assigned as a vector which is used to draw the class labels from finite sets. It is not just an algorithm but algorithm family is worked on principles: naive classifiers assume that feature value is not dependent on any value which has class variables. For instance, apple is a fruit which has red color, round in shape and approximate 10 cm diameter. A naive Bayes considers probability of each feature .For example an apple, regardless of its features of color, correlation, roundness or diameter.

For other probability models, it is trained in a supervised learning background in an efficient manner. In other applications, maximum likelihood method is used as a parameter estimation of the naive Bayes, in other words, anybody can work with it without the acceptance of Bayesian Probability and methods. Naive Bayes are classified very well in the complex situation of real-world problems in spite of their quite simplified design and assumptions.

### *D. Linear classifier*

In the field of machine learning, the objective of measurable order is to utilize an item's qualities to distinguish which class (or group) it belongs to. A Linear classifier accomplishes this by settling on an order choice in view of the estimation of a direct combination of the attributes. An item's attributes are otherwise called highlight values and are normally introduced to the machine in a vector called a feature vector. These classifiers are used efficiently in the problems like document classification and the problems of variables; need to reach the level of accuracy compare to non-linear classifier during the time of training which is less.

Linear classifiers belong to a particular class of support vector machines, which are supervised learning models. They contain learning algorithms which help in analyzing the data

used in classification and regression analysis. An SVM model is represented as sample points in space which are mapped such that they are separated by a gap as far as possible according to the categories they belong to. The category of new samples is predicted according to the side of the gap they fall on, after being mapped into that same space.

Technically a support vector machine builds a set of hyper-plane in a multi-dimensional space, which is used in regression, classification etc. A hyper-plane having the greatest distance to the data point of a specific class is called functional margin. In general, functional margin is inversely proportional to the generalization error. Higher the functional margin, lower is the generalization error.

## III. DATASET

There are various open source datasets available online. The datasets were obtained from NASA Promise dataset repository. Three datasets namely CM1, JM1, PC5 were used.

Below table provides detailed information about each dataset such as its attributes, instances, faulty and non-faulty instances, missing attributes etc.

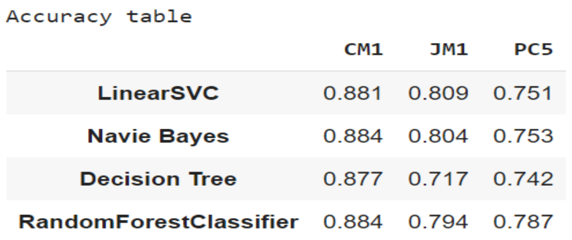
TABLE I. CHARACTERSTICS OF DATA USED

| Dataset | no of attributes | no of instance | Non-faulty instance | Faulty instance | defective instance | Missing attribute |
|---|---|---|---|---|---|---|
| CM1 | 38 | 498 | 449 | 49 | 9.83 % | None |
| JM1 | 22 | 10885 | 8779 | 2106 | 19.35 % | 5 |
| PC5 | 39 | 1711 | 1240 | 471 | 27.52% | None |

## IV. RESULTS AND ANALYSIS

After research and analysis, The datasets were obtained from NASA Promise dataset repository. Three datasets namely CM1, JM1, PC5, were used. The algorithms selected for the analysis are Naïve Bayesian (NB-C), Decision Tree (TARGET-C), Linear Classifier and Random Forest classifier.

Summary of Test results is shown in the following table. It depicts the accuracy rate of each algorithm (percentage-wise). The highest algorithm in a dataset is marked bold to indicate it amongst others.



Accuracy table

| | CM1 | JM1 | PC5 |
|---|---|---|---|
| LinearSVC | 0.881 | 0.809 | 0.751 |
| Navie Bayes | 0.884 | 0.804 | 0.753 |
| Decision Tree | 0.877 | 0.717 | 0.742 |
| RandomForestClassifier | 0.884 | 0.794 | 0.787 |

Fig. 1. Accuracy of different algorithms on datasets

As it is clearly visible from the results that Random Forest and Linear Classifier algorithm has highest defect prediction accuracy in two of the three selected datasets, it is the most reliable technique due to its greater accuracy. The rest three algorithms having highest accuracy in one dataset each were Naïve Bayesian, Decision Tree. Next plots represents the accuracy of algorithms over different datasets. This table in figure 3 represents the ROC curve value of results from predicted defect. The lowest roc curve value of Linear SVC algorithm. In case of a tie in two algorithms in terms of defect prediction accuracy, the roc curve value will help for breaking the tie between two algorithms .

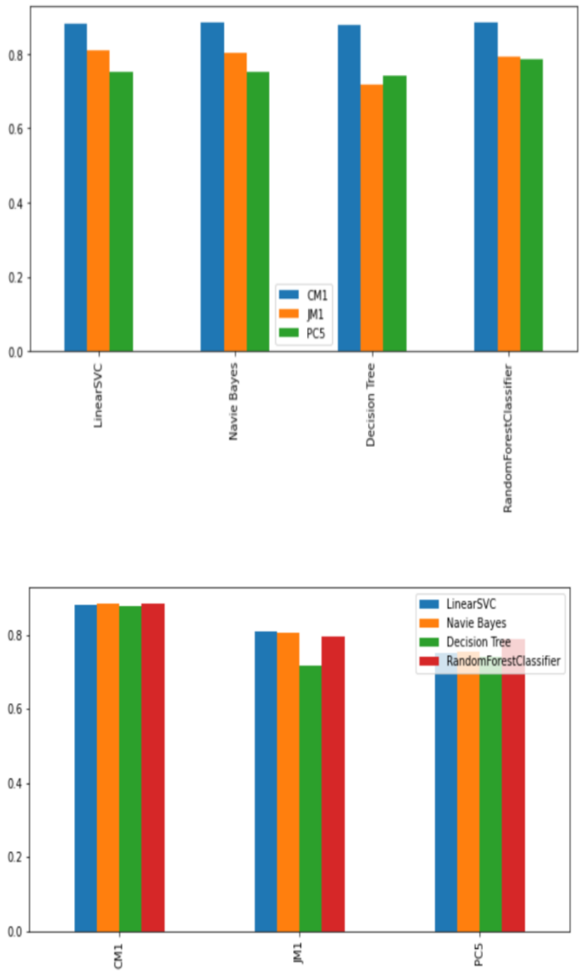
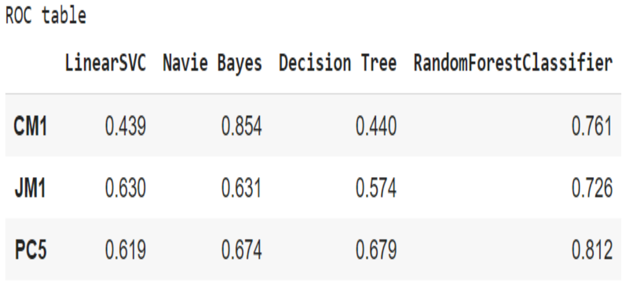


Fig. 2. Accuracy of different algorithms on datasets

Receiver operating characteristic (ROC) curves compare sensitivity versus specificity across a range of values for the ability to predict a dichotomous outcome. Area under the ROC curve is another measure of test performance .The higher the roc curve value, higher the performance.



ROC table

| | LinearSVC | Navie Bayes | Decision Tree | RandomForestClassifier |
|---|---|---|---|---|
| CM1 | 0.439 | 0.854 | 0.440 | 0.761 |
| JM1 | 0.630 | 0.631 | 0.574 | 0.726 |
| PC5 | 0.619 | 0.674 | 0.679 | 0.812 |

Fig. 3. ROC curve value of different algorithms on datasets

It is clear from the results that Naive Bayes has highest roc value followed by Random Forest Classifier. However the highest accuracy rate lies with Linear Classifiers.

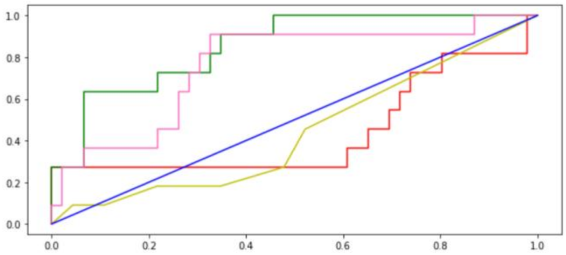


Fig. 4. ROC curve for CM1 Datasets

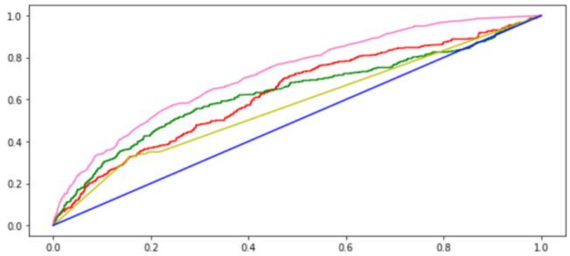


Fig. 5. ROC curve for JM1 Datasets

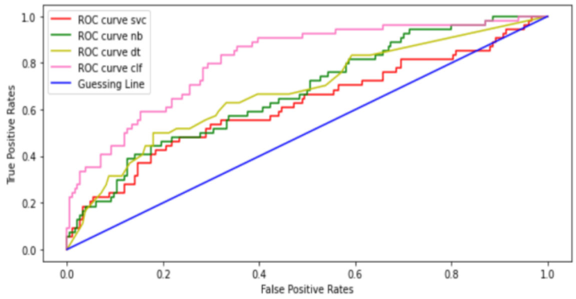


Fig. 6. ROC curve for PC5 Datasets

So for a case of tie in accuracy prediction, we can use roc curve value metric to determine which algorithm gives best results.

## V. CONCLUSION AND FUTURE SCOPE

The research questions laid in the beginning of this paper are answered as follows

Most widely used metrics in Software Defect Prediction [2] are Traditional static code metrics defined by Halstead and McCabe, Size metrics such as LOC (lines of code) metric, Object-oriented metrics like cohesion, coupling and inheritance, Hybrid metrics used both Object Oriented as well as procedural metrics for defect proneness prediction.

Dataset CM1, JM1 and PC5 are good for study purpose as they are less biased comparison to other datasets provided by NASA datasets and have small, medium and large size respectively.

Most widely used metrics in Software Defect Prediction are: NASA datasets which are publicly available in the NASA repository by NASA Metrics Data Programme and are the most commonly used datasets for SDP, PROMISE repository datasets and NASA datasets along with other datasets donated by individuals from research back-ground.

As per the results obtained of accuracy, it is evident that the Random Forest Classifier algorithm has highest defect prediction accuracy amongst Two of the Three selected datasets, hence proving to be the most reliable technique amongst supervised learning algorithms in Data Mining. The datasets in which it had maximum prediction accuracy were CM1,PC5.As per the analysis performed in this paper, Random Forest Classifier is the most accurate and reliable amongst defect prediction algorithms. Three datasets were selected for defect prediction analysis. Classification was performed on these s three datasets and were validated using 10 fold cross validation using python program. The results demonstrated the supremacy of Random Forest over other algorithms in terms of defect prediction accuracy.

It is clear from the results that Linear classifier have the lowest roc curve value in the experiment followed by Decision Trees. However the good accuracy rate lies with Linear Classifiers. In case of a tie in accuracy prediction, we can use roc curve value metric to determine which algorithm gives best results.

These results can be further refined by using more number of datasets. Increased number of datasets will strengthen the results. Also, comparison can be done amongst more number of algorithms. Most popular and widespread used algorithms were taken into account in this research , hopefully new techniques will arise in future and could be included in the comparative analysis, providing new and improved results.

## REFERENCES

[1] **A**. Chug and S. Dhall, "Software defect prediction using supervised learning algorithm and unsupervised learning algorithm", Confluence 2013: The Next Generation Information Technology Summit (4th International Conference), pp. 173 — 179, 2013.

[2] **R**. Malhotra, "A systematic review of machine learning techniques for software fault prediction", Applied Soft Computing, Elsevier Science Publishers B. V. Amsterdam, pp. 504-518,2015.

[3] **J.**Ratzinger, T.Sigmund and H. C. Gall, "On the relation of refactoring and software defect prediction", In Proceedings of the 2008 international working conference on Mining software repositories ACM, pp. 35-38, 2008.

[4] **R**. Malhotra and Y. Singh, "On the Applicability of Machine Learning Techniques for Object Oriented Software Fault Prediction", Software Engineering: An International Journal (SEIJ), Vol. 1, pp. 24-37, 2011.

[5] **A**. Balasundaram, P. T. V. Bhuvaneswari,"Comparative study on decision tree based data mining algorithm to assess risk of epidemic", IET Chennai Fourth International Conference (SEISCON), pp. 390-396, 2013.