



Cost Sensitive Classification and Feature Selection for Software Defect Prediction

¹Aruna S, ²Dilsha Dominic, ³Radhika R, ⁴Dr. Swathi J N

^{1, 2, 3}M.Tech Computer Science, VIT, Vellore, Tamilnadu, India

⁴Associate Prof., Dept. of Computer Science, VIT Vellore, Tamilnadu, India

Abstract: *Measuring software defect is of prime importance to minimise cost and improve the overall effectiveness in testing process. Several datasets exists which can be mined in order to extract knowledge regarding the possible defects. Classification models are more specific in predicting the faults. Adding feature selection reduces data sets with fewer features and maintains or improves the prediction capability over the original data sets. The resulting datasets used to predict the defect in the software modules. Further, cost sensitivity is also included in both the stages in order to make the analysis more accurate. The performance of the cost sensitive Naive Bayes is compared against other classification algorithms and the benefits are discussed. The motivation to do this project is to conduct a study on the applications and the benefits of using data mining techniques for software defect prediction.*

Keywords: *Naive Bayes, Cost Sensitive Learning, Classification, Feature Selection, Defect Prediction*

I. INTRODUCTION

A software defect can be defined in many ways. Sometimes it may be the case when the resultant software product does not meet the customer requirement specification or user expectations. Since each phase of software life cycle is a human activity, mistakes are inevitable. While designing or coding the software, mistakes can occur, resulting in a deviation from the accepted results. Any kind of such deviation which results in malfunctioning or makes the software to behave in unintended ways can be called as a defect.

To detect and correct the software defects is one of the most expensive activities in terms of cost, quality and time. Eliminating all the defects in a software system is impossible but the impact caused by them can be reduced by minimizing the defects. If the defects could be detected in the early stages then it can lead to reduced development cost and high quality software system. In software defect prediction the defective modules will be identified before the final stage of software development, so the final product will contain only a few defects.

Many hidden factors associated with software defect prediction can be discovered by the data mining approach. The use of data mining techniques will help to identify the modules that have a high probability of being defective. Effective Defect prediction is based on good data mining model [1]. The techniques of data mining that are used for software defect prediction are as follows:-

1. Regression.
2. Association.
3. Clustering.
4. Classification
5. Feature Selection

The quality and the effectiveness of the software development process can be guaranteed by adopting measures to predict the defect prone modules. It also allows the person in charge to allocate the resources effectively. In this paper Naive Bayes classification algorithm and filter based feature selection are used for software defect prediction.

In classification, based on a given input an outcome can be predicted. For this, a training set with a set of attributes is processed by the algorithm and the corresponding output called prediction attribute is generated. In software defect analysis classifying the defects and defects prone modules is an important task [2]. Bayesian Network is a data mining model which graphically represents the casual or influential relationship between a set of attributes of our interest. It favours taking several products metric at same time and to analyse the effect [3].

Feature selection is an important part in determining process, to deal with the situations where excessive number of features cause computational burdens in various feature extraction techniques. Feature selection has been widely used in many pattern recognition and machine learning applications [4]. Filter based feature selection is one of the methods. "Filter-type methods select features according to some criteria, and does not involve any learning algorithm. Hence, filter-type methods are usually adopted in practice due to their simplicity and computational efficiency" [5].

The visualization of these two steps are obtained using WEKA tool [7]. "Weka is a very good tool used for solving various purposes of data mining. Weka has four application interfaces: explorer, experimenter, knowledge flow and simple command line. The task can be processed using any of these interfaces" [8].

II. CLASSIFICATION USING NAIVE BAYES

Data Mining or mining knowledge from data is the process of finding and extracting meaningful information and patterns from large sets of databases. The process of Data Mining employs different techniques such as Pattern Matching, Clustering, Feature Selection, Association and Classification, to identify the relationships from the given data and use them to extract the required information[1]. Mining techniques can be applied on a software defect repository to analyze the data that are defective. Software defect repository contains data obtained from a defect tracking system on a major project. These data can be used to evaluate whether the software module is defect prone or not. Data mining techniques has significant influence on finding and extracting information from this software defect data, as it helps the software developers to improve the quality of software. It is aimed to determine whether software module has a higher failure risk or not.

Classification is the process of assigning an object to a certain class based on its similarity to previous examples of other objects [4]. It can be done with a reference to the original data or using a model of the data. Classification involves prediction of accurate class assignment for the given test data. The process divides the data samples into target classes. With respect to a software defect prediction data set, the target classes can be Defective or Non-Defective. Classification can be Supervised or Unsupervised. In supervised method, we should do the mapping of classes based on prior knowledge or experience. The technique involves the use of probabilistic methods such as Bayesian networks and Naive Bayes. A training set is used where all the inputs are already associated with known class labels. The classification algorithm learns from the training set and builds a model. The model is used to classify new set of inputs. The Bayesian method of classification applies the Bayes theorem of conditional probability which can be stated as follows:

$P(A/B) = [P(B/A).P(A)]/P(B)$, where $P(A)$: Prior probability of hypothesis A

$P(B)$: Prior probability of training data B

$P(A/B)$: Probability of A given B (also known as the Posterior Probability of the class A(target) when B is the given predictor(attribute))

$P(B/A)$: Probability of B given A

The probability that the given record in the input dataset belongs to a particular class can be predicted using this conditional probability. This is the main approach used in the Bayesian Classifiers. Naïve Bayes is also based on the Bayes Theorem, but it has strong independence assumptions. The Naïve Bayes classifier assumes that the effect of the value of a predictor (B) on a given class (A) is independent of the value of the other predictors. Hence, the existence of a particular target class or its feature does not depend on the existence of other features. Due to this approach, it is simple to implement and also more robust than other classification algorithms. It is used when the dimensionality of the input is very high. It trains and constructs predictor by analyzing historical data of the software modules and based on the predictor it will predict whether the new model is defective or not. Therefore, in this work we have chosen Naïve Bayes classifier for the software defect prediction. It is important to consider the cost of every type of error so as to avoid the costliest of errors. This helps to achieve classification accuracy. Naïve Bayes is a good classifier where we can add the option of cost sensitive learning. Here, instead of predicting the attribute with the highest probability, the attribute with the lowest cost is selected. This helps to increase the number of correctly classified instances. The evaluation of the misclassification costs from the different types of errors focuses on minimizing the total costs rather than lower classification error rates.

III. FEATURE SELECTION

Feature selection methods are used to reduce computation time it also improve the performance of the prediction. There are many applications of feature selection like it will understand the data in machine learning or pattern recognition applications [19].The process of variable elimination helps in reducing the effect of troubles of dimensionality and improves the performance of the defect prediction process. The aim of feature selection is to select a subset of variables. From the selected subset we can efficiently describe the input data [6].” It reduces effects from noise or irrelevant variables and still provides good prediction results”. “Directly evaluating all the subsets of features for a given data is an NP-hard problem”. Hence a suboptimal procedure must be used which can remove redundant data with tractable computations. “There are three methods to perform feature selection– Filter method, Wrapper method and the Embedded method”. Wrapper methods are basically a search problem which considers the selection of a set of features. It is computationally expensive. In the case of Embedded methods that will learn which features those are best contribute to the accuracy of the model while the model is being created. Filter feature selection methods make use of statistical information in order to assign scores to the features. The features will be ranked based on their scores. The ranks are used in order to decide whether the particular feature should be sent for classification or removed from the dataset. The feature subset selection process involves the following steps [17]:-

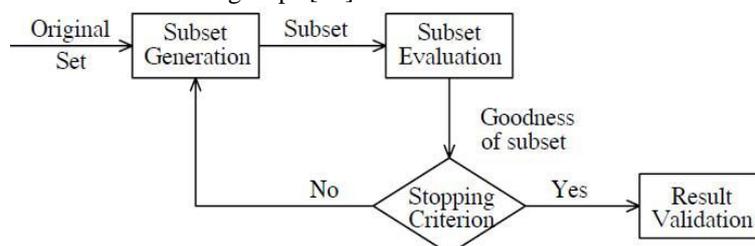


Fig 1. Steps for feature selection [19]

1. Original dataset is given to the subset generation process as input.
2. Generated subset is evaluated by the subset evaluation process.
3. If the quality of the retrieved subset is not sufficient repeat step 1 to 3 else proceed to step 4.
4. The result is given to the result validation process.

There are three types of popular filter-type feature selection methods-” Variance, Laplacian Score, and Constraint Score” these are closely connected to the proposed methods [6]. “Variance score (VS) is a simple unsupervised evaluation criterion of features”. Among all the available samples VS selects features that has maximum variance. “The basic idea of VS is variance among a feature space that reflects the representative power of feature”. Another popular method is the unsupervised feature selection. “Laplacian Score (LS) prefers features with larger variances which have more representative power and prefers features with stronger locality preserving ability”. “Constraint Score (CS) is a semi-supervised feature selection method, which performs feature selection according to the constraint preserving ability of features.” [6] Filter-type methods are usually adopted in practice due their simplicity and computational efficiency.

IV. EXPERIMENTAL EVALUATION

The datasets provided by the PROMISE software repository are extensively used to carry out studies on Software Defect Prediction. The datasets used in this study are the NASA Metrics Data Program (MDP) datasets. The datasets are part of various space exploration related software projects such as NASA space craft instrument(CM1, written in C), storage management system for ground data(KC1 and KC2), real time predictive ground systems(JM1, also written in C) and flight software for an Earth orbiting satellite(PC1, data from C functions.) . CM1 has 498 numbers of instances and 22 metrics which includes both McCabe and Halstead measures. The McCabe metrics reflect pathways within a code module. The Halstead measures count the number of concepts or unique operators in a module. The JM1 dataset has 10885 instances, KC2 with 522 instances and both with 22 metrics. The experimental study is done using WEKA data mining tool.

In filter methods, the subset selection procedure is independent of the learning algorithm and is generally a pre-processing step. Here we are using Genetic Search algorithm to perform a cost sensitive feature selection. GA based attribute selection focuses mainly on classification tasks. Hence it assists in the succeeding classification process of mining the defects. The selected attributes are given as input to the Naïve Bayes classifier in order to predict the defective modules in the software. The cost matrix has been modified in order to select the attributes with the least cost. Hence, we have implemented a two stage cost sensitive learning here, first at the feature selection stage and later at the classification stage.

The confusion matrix is used to evaluate and analyze the performance of the classification algorithm, particularly in supervised technique.

Table1: Confusion Matrix for Defect Prediction

	Predicted	Defect prone	Non-Defect prone
Actual			
Defect prone		True Positive	False Negative
Non-Defect prone		False Positive	True Negative

The classification accuracy calculates the proportion of the correctly classified instances.

Accuracy= $(TP+TN)/TP+TN+FP+FN$

The classifier’s ability to identify the negative results is called True Negative Rate or Specificity. It is computed as $TN/ (TN+FP)$.

The proportion of the actual positives which are correctly identified by the identifier is called True Positive rate or **Sensitivity** or **Recall**. It is calculated as $TP/ (TP+FN)$.

Table 2: Results

Data set	No. of Instances	Correctly classified	Incorrectly classified	Precision	Recall	Accuracy
CM1	498	425	73	0.862	0.853	0.85
KC1	2109	1737	372	0.816	0.824	0.823
KC2	522	436	86	0.82	0.835	0.835
JM1	10885	8754	2131	0.765	0.804	0.804
PC1	1109	989	120	0.899	0.892	0.892

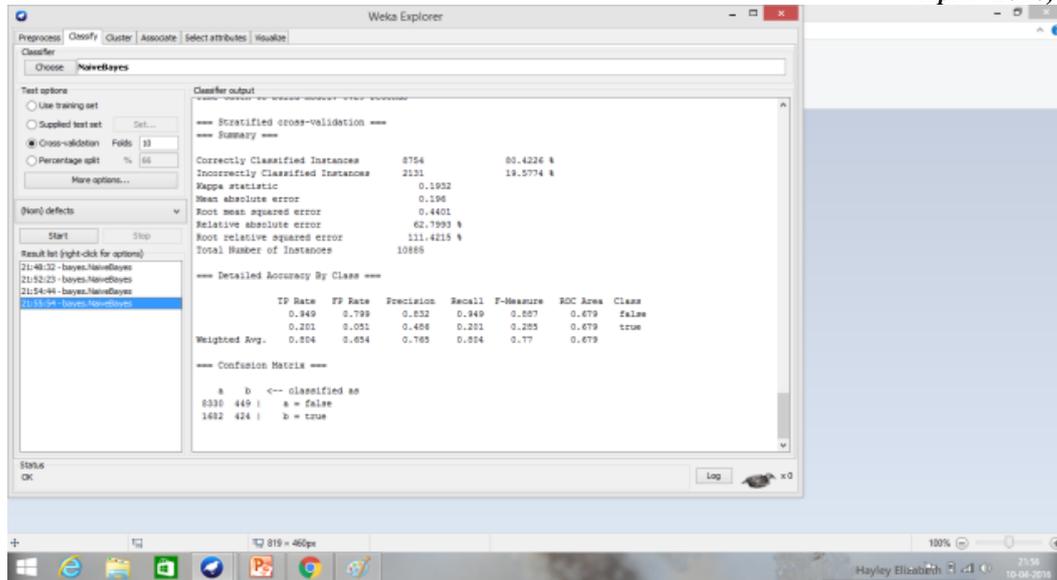


Fig 2. Defect prediction and analysis for JM1 dataset

Table 3: Comparison of Classification Algorithms for Defect Prediction

Classifier	Measure	CM1	KC1	KC2	PC1	JM1
ZeroR	Accuracy	0.901	0.854	0.795	0.931	0.807
	Sensitivity	0	0	0	0	0
	Specificity	1	1	1	1	1
J48	Accuracy	0.879	0.845	0.814	0.933	0.814
	Sensitivity	0.061	0.331	0.495	0.234	0.495
	Specificity	0.969	0.939	0.104	0.985	0.104
Classification via Clustering	Accuracy	0.843	0.815	0.806	0.884	0.806
	Sensitivity	0.286	0.417	0.719	0.312	0.719
	Specificity	0.904	0.887	0.829	0.927	0.829
Conjunctive Rule	Accuracy	0.901	0.844	0.801	0.931	0.801
	Sensitivity	0	0	0.224	0	0.224
	Specificity	1	1	0.949	1	0.949
Naive Bayes	Accuracy	0.85	0.823	0.835	0.892	0.804
	Sensitivity	0.853	0.824	0.835	0.892	0.804
	Specificity	0.862	0.816	0.82	0.899	0.765

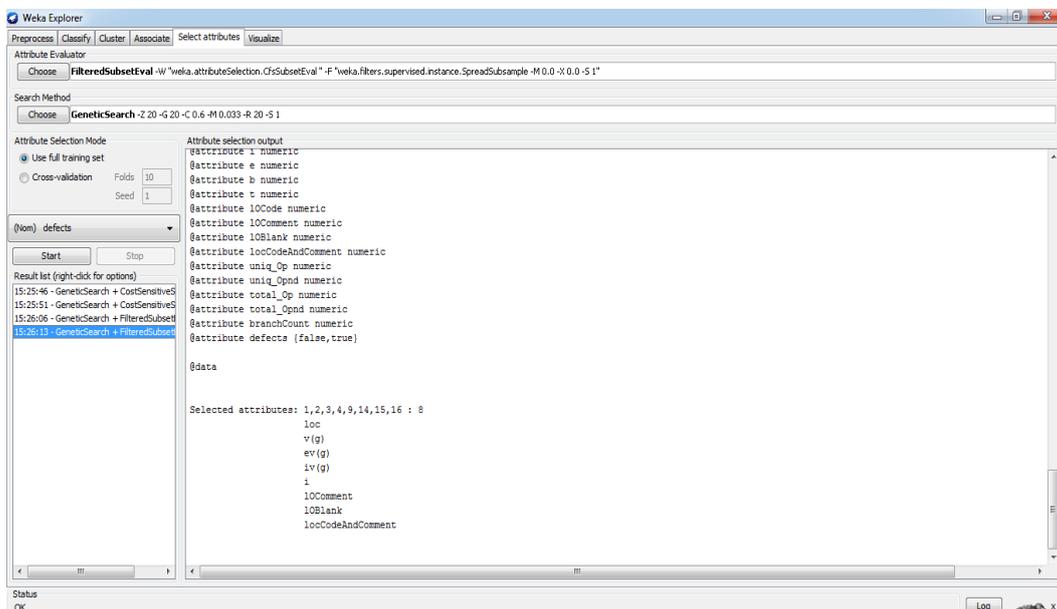


Fig 3: Genetic Search based Filtered Feature Selection for JM1 dataset

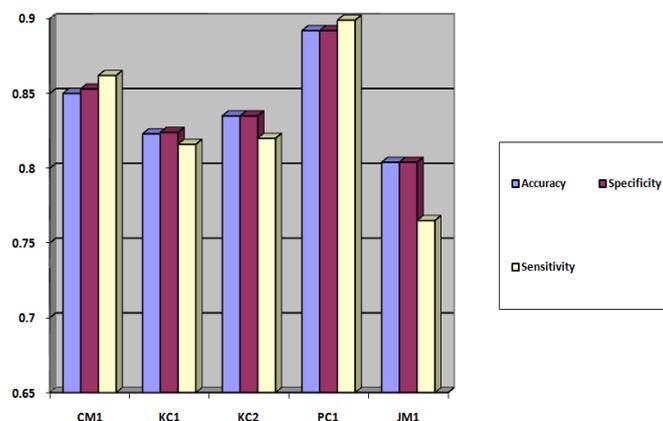


Fig 4: Performance measures for two stage cost sensitive learning using Naive Bayes

From the results it can be seen that the performance measures vary according to the datasets. The ZeroR algorithm gives better accuracy for the CM1 as well as the PC1 datasets compared to the cost sensitive Naive Bayes. However, if the consistency is evaluated, it is observed that cost sensitive Naive Bayes combined with cost sensitive feature selection shows a consistent performance for all the datasets. If neural classifiers alone are compared, it is noted that Conjunctive Rule based classifier performs better than cost sensitive Naive Bayes with feature selection, as the size of the data sets increases. It is observed that Conjunctive Rule and cost sensitive Naive Bayes classifier would be better algorithms for software defect predictions using neural networks. J48 and ZeroR would be suitable for datasets such as CM1, KC1. However, large datasets including those from the space exploration softwares and geo sensing applications require effective data mining techniques in order to predict the defects. Hence, neural computing based defect prediction algorithms along with a proper feature selection method are required in order to achieve accuracy in predicting defects.

V. CONCLUSION AND FUTURE WORK

Defects that arise during the process of software development affect the quality of the software. This can lead to errors and failures in the software. The quality of the software relies on how effectively we predict and analyze the defects. In this paper, we have used data mining techniques for software defect prediction. Due to the presence of large number of data sets, defect prediction can be a very tedious task. For this purpose the principles of data mining is utilized completely in order to guarantee the quality and efficiency of data mining.

This research work has investigated the effectiveness of the Naive Bayes classifier as well as the filter type feature selection for software defect prediction with the application of two stage cost sensitive learning. Experimental results from WEKA show that Naive Bayes is accurate and the addition of cost sensitivity helps to increase the number of correctly classified instances, compared to cost insensitive techniques. Furthermore, the use of filter based feature selection with a cost sensitive Genetic Search algorithm selects the target attributes with the least costs, which are passed to the classifier. This makes the learning more precise and the accuracy is enhanced. We can extend our study to include cost sensitivity to more advanced machine learning algorithms. Cross company defect prediction is an important aspect in the area of defect prediction and analysis and we hope to extend this work towards the same direction.

ACKNOWLEDGMENT

We would like to express our profuse sense of gratitude to our guide Prof. Dr. Swathi J.N for extending her full support in the successful completion of this work. We would also like to thank the staff and technicians of VIT University, Vellore for providing the necessary assistance.

REFERENCES

- [1] Azeem, N., & Usmani, S. (2011). Analysis of data mining based software defect prediction techniques. *Global Journal of Computer Science and Technology*, 11(16).
- [2] Liu, Y., Cheah, W. P., Kim, B. K., & Park, H. (2008). Predict software failure-prone by learning bayesian network. *International Journal of Advanced Science and Technology*, 1(1), 35-42.
- [3] Srivastava, S. (2014). Weka: a tool for data preprocessing, classification, ensemble, clustering and association rule mining. *International Journal of Computer Applications*, 88(10).
- [4] Sathyaraj, R., & Prabu, S. (2015). An Approach for Software Fault Prediction to Measure the Quality of Different Prediction Methodologies using Software Metrics. *Indian Journal of Science and Technology*, 8(35).
- [5] Ma, Y., Luo, G., Li, J., & Chen, A. (2011, October). Software defect prediction using transfer method. In *Computational Problem-Solving (ICCP), 2011 International Conference on* (pp. 610-613). IEEE.
- [6] Liu, M., Miao, L., & Zhang, D. (2014). Two-stage cost-sensitive learning for software defect prediction. *Reliability, IEEE Transactions on*, 63(2), 676-686.
- [7] Lu, J., Behbood, V., Hao, P., Zuo, H., Xue, S., & Zhang, G. (2015). Transfer learning using computational intelligence: a survey. *Knowledge-Based Systems*, 80, 14-23.
- [8] Okutan, A., & Yildiz, O. T. (2014). Software defect prediction using Bayesian networks. *Empirical Software Engineering*, 19(1), 154-181.

- [9] Fenton, N., Neil, M., & Marquez, D. (2008). Using Bayesian networks to predict software defects and reliability. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 222(4), 701-712.
- [10] Lessmann, S., Baesens, B., Mues, C., & Pietsch, S. (2008). Benchmarking classification models for software defect prediction: A proposed framework and novel findings. *Software Engineering, IEEE Transactions on*, 34(4), 485-496.
- [11] Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI magazine*, 17(3), 37.
- [12] Jordan, A. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in neural information processing systems*, 14, 841.
- [13] Zhang, F., Mockus, A., Keivanloo, I., & Zou, Y. (2014, May). Towards building a universal defect prediction model. In *Proceedings of the 11th Working Conference on Mining Software Repositories* (pp. 182-191). ACM.
- [14] Elish, K. O., & Elish, M. O. (2008). Predicting defect-prone software modules using support vector machines. *Journal of Systems and Software*, 81(5), 649-660.
- [15] Bouckaert, R. R. (2004). *Bayesian network classifiers in weka*. Department of Computer Science, University of Waikato.
- [16] Sahana, D. C. (2013). *Software Defect Prediction Based on Classification Rule Mining* (Doctoral dissertation).
- [17] Kaur¹, R., & Bajaj, P. (2014). A Review on Software Defect Prediction Models Based on Different Data Mining Techniques.
- [18] Anbu & G.S. Anandha Mala (2015) Investigation of Software Defect Prediction Using Data Mining Framework .
- [19] Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1), 16-28.
- [20] Tu, L., Fowler, B., & Silver, D. L. (2010, May). CsMTL MLP For WEKA: Neural Network Learning with Inductive Transfer. In *FLAIRS Conference*.
- [21] Thangaraju, P., & Mala, N. Effectiveness of Searching Techniques in Feature Subset Selection: A Review.
- [22] Dash, Manoranjan, and Huan Liu. "Consistency-based search in feature selection." *Artificial intelligence* 151, no. 1 (2003): 155-176.