Data-Mining Accuracy Boost Technique for Software Deformity Prophecy Datasets Model

Salahuddin Shaikh[1], Maaz Rasheed Malik[2], Saifullah Adnan[1], Dhani Bux[1]

[1]Center for Computing Research Department of Computer Science & Software Engineering, Jinnah University for Women, Karachi, Pakistan

[2]Dept. Of Information Communication Engineering, Guilin University of Electronic Technology, Guilin, China,

*E-mail: dhanibux@hotmail.com

ABSTRACT

Data mining may be defined as the process of extracting important and useful information from a large database. Data mining research may be used to make business decisions that will save costs, increase revenue, and increase operational efficiency in the human services sector while maintaining high levels of patient attention. When it comes to the machine learning field, the Software Deformity Prophecy model is based on the arrangement of producing information that has the capacity to distinguish between defected and non-defected models based on a large number of rules. The ability of these Software Deformity Prophecy model to categories a model in this way is essentially a component of the nature of creating datasets model. Discretization is the most appropriate technique in each progression of preprocessing in order to improve the accuracy of the classification model for Software Deformity Prophecy model in order to increase the accuracy of the classification model for Software Deformity Prophecy datasets model.

We have utilized a data-preprocessing method known as Discretization in our study for software deformity prophecy datasets, and it has been shown to be effective. This is our method for increasing the accuracy of our classification accuracy boost for our software deformity prophecy datasets model. With the assistance of the discretization preprocess technique, we were able to increase the accuracy of our observation and analysis by using several classifiers. The results of all the tests clearly show that no classifier can be considered ideal in terms of accuracy and efficiency when used to create software deformity prophecy dataset models.

While the efficiency and accuracy of the decision stump, hoeffding tree, and lmt are not particularly high in correctly classified instances, they are significantly higher when compared to when they are used in a non-discretionary manner. In the case of correctly classified instances, we can easily judge the improvement of each classifier. The position of stacking is very poor and should not be used in these tests since their efficiency and accuracy have not improved, and in fact, seem to be worse in all cases, as a result.

Keywords: Data Mining, Defect-prone, Classification, Machine Learning, Software-defect, Bug; Software-model, Data-Pre-process.

INTRODUCTION

The vast quantity of data that is stored in databases includes valuable but obscured information that allows the client to enhance the presentation of fundamental leadership processes to their employees and customers. Database innovation, machine learning, insights, knowledge-based systems, pattern recognition, superior processing, data recovery, artificial intelligence, artificial neural networks, and data perception are all examples of work areas in the field of research known as data mining. Data mining is defined as the process of extracting important and useful information from a large database, or as an effort to do so. It is possible to make business choices based on data mining research that will lower the cost of providing care to patients while simultaneously increasing their revenue and increasing their operational productivity in the human services sector (LeiQiaoa, Xueson). There are a few notable data mining systems that have been developed and are now in use in the field of data mining. Executives in the field of social insurance use data mining techniques for a variety of purposes, including diagnosis and treatment, healthcare endeavor management, customer relationship management, fraud and anomaly detection, and customer relationship management. The use of data mining methods may assist doctors in distinguishing between strong medicines and excellent practices, while patients can profit from more efficient and affordable social insurance administrations (Nana Zhang, Kun Zhu).

Data mining is the process of semi-consequentially examining large datasets in order to find patterns and trends. Achieving this involves examining and making decisions based on the data that has already been in databases, which is called data mining. It uncovers important information that has been buried amid huge quantities of data. Data mining is also defined as a system of operations that identifies patterns in large amounts of data. When considering possible applications of the data mining approach, it is important to note that the characteristics of a data archive that may be utilized may be very different from those that were first seen when the data was collected. As shown in the understanding, machine learning and data mining are used in a broad variety of fields and situations (P. Singh, S. Verma).

Various data mining techniques have been suggested for software deformity analysis in the past, but only a few of them have been successful in addressing all of the problems associated with software deformity. Many data categorization models' assessments are difficult to comprehend, and they also provide the exact number of flaws, which is very hazardous, particularly at the beginning of a job when there is a limited amount of data available. On the other hand, categorization models that predict potential faultiness may be specific while being less helpful in terms of giving insight into the actual number of defects. A large number of scholars employed a wide variety of methods with a wide variety of datasets to predict faultiness. In any event, there are such a large variety of categorization algorithms available that it may be difficult to predict when anything will go wrong. Every one of these problems serves as a source of inspiration for our study in the area of software deformity prediction (Ishani Arora a, Vivek Tetarwala, et.al) (Ahmed H. Yousef).

Our contributions are that we have proposed one technique known as Discretization used for accuracy boost techniques for software deformity prophecy model. We have used NASA MDP datasets and these datasets are analyzed by evaluation measures, where we have used tp-rate, f-measure, area under curve and correctly classified instance. Data is analyzed by WEKA 3.9.3, in which 12 different classifiers we have used and performed classification for knowing the highest accuracy and efficiency in between these classifiers. We did experiments two times and verify it. It took too long time to get results due to the cashed storage full in the system (Wang, Shuo, Xin Yao).

OBJECTIVES

Our main objectives are that to improve the accuracy of Software Deformity Prophecy Datasets model, because our class of interest is True Positive and True Negative class. To improve the accuracy, we have focused on classification, with the help of Discretization classification, we have improved our model.

METHODOLOGY

Classification Accuracy Boost way

For Software Deformity Prophecy datasets, discretization is the most appropriate technique to use in each stage of the preprocessing procedure in order to improve the accuracy of the classification model for such datasets.

Since data preprocessing is an essential advancement in data mining, and preprocessing resolves the many types of data problems that occur in large databases in order to provide high-quality data for the mining job, it is important to understand how data preprocessing works. It is possible to convert an ordinal attribute from a non-stop attribute via the process of discretization. One or two classes are used to organize an almost infinite number of characteristics. It is a process that converts quantitative data into subjective data, and it is often used in data mining applications to transform quantitative data into subjective information.

A common use of discretization is in classification, and many classification algorithms provide excellent results when both the free and bounding variables have just a few distinguishing characteristics. The discretization of constant highlights or characteristics plays an important role in the preparation of machine learning data during the training stage. Many machine learning algorithms even create their own discretization system for classification and arrangement of the qualities of classes and computed data in every possible split, which they then use to classify and arrange the qualities of classes and computed data. The split that restricts data excludes breakpoints between characteristics that have a position in a comparable class since this will result in the creation of new data when this is done. Discretization consistently applies the equivalent to the subsequent interims until some halting foundation is fulfilled.

Datasets & Evaluation Measure

The open-source datasets model, also known as the NASA repository datasets model, was utilized for our study activities, which is widely recognizable to the scientific community. These are essentially public dataset models, and the NASA MDP data sets model is publicly accessible as part of the NASA MDP data sets model. Many academics have utilized these dataset models to investigate software deformity problems, and they have been quite successful. Table **1** lists the 17 dataset models that we selected from among the many available. The primary reason for the usage of these dataset models is because they fall into two types of models: one is the defective dataset model, which falls under class (y), and the other is the non-defective dataset model, which falls under class (x) (Y). Table **1** provides some essential facts, which may be summarized as the software datasets are included inside each dataset model within a particular dataset model. Software dataset models may be classified as buggy or faulty or as non-buggy or non-faulty; they can contain attributes and a total number of models.

Table 1. Nasa Mdp Datasets Model.

S.NO	Datasets	Attributes	Model	Defective	Non-Defective
1	JM1	22	7782	1672	6110
2	AR1	30	121	9	112
3	KC3	40	194	36	158
4	AR6	30	101	15	86
5	CM1	38	327	42	285
6	KC2	22	522	107	415
7	MC1	39	1988	46	1942
8	MC2	40	125	44	81
9	PC5	39	17186	516	16670
10	AR3	30	63	8	55
11	PC1	38	705	61	644
12	AR4	30	107	20	87
13	MW1	38	253	27	226
14	AR5	30	36	8	28
15	PC3	38	1077	134	942
16	PC4	38	1458	158	1289
17	PC2	37	745	16	729

Evaluation Measure: Different classifiers have been utilized for our dataset models in data mining classification, and we have used a variety of them. The measurements for all of these classifiers exist, or they are assessed by the measures for data evaluation. Because data assessment methods are straightforward to apply and may assist you in determining the efficiency and correctness of a dataset model, they are often used [17]. Evaluation measures are used to analyses the results of all of the trials. The tp-rate, the f-measure positive accuracy, the area under the curve, and the number of properly categorized cases are the most helpful assessment metrics we have employed in our work. We have concentrated our efforts on the class (y) model. This section analyses the performance of classification methods employed. The performance is examined and evaluated via a range of confusion matrix measurements. The following parameters include a confusion matrix:

True Positive (TP): Instances that are positive and categorized as positive.

False Positive (FP): Instances that are indeed negative but show up as positive.

False Negative (FN): Positive but negative instances.

True Negative (TN): occurrences that are both negative and negative.

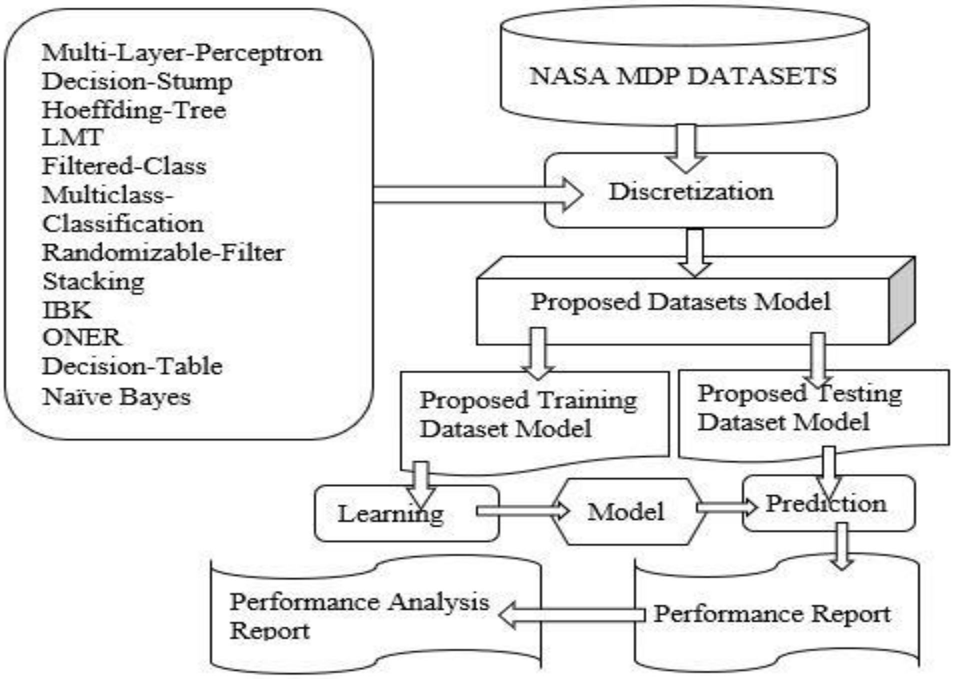
TP-Rate, F-measurement, ROC & Correctly Classified Instances (Accuracy) assess the categorization methods [18].

$$\text{F-measure} = \frac{\text{Precision} * \text{Recall} * 2}{(\text{Precision} + \text{Recall})}$$

$$AUC = \frac{1 + TP_r - FP_r}{2}$$

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

Classification Accuracy Frame-work Model & Experiments



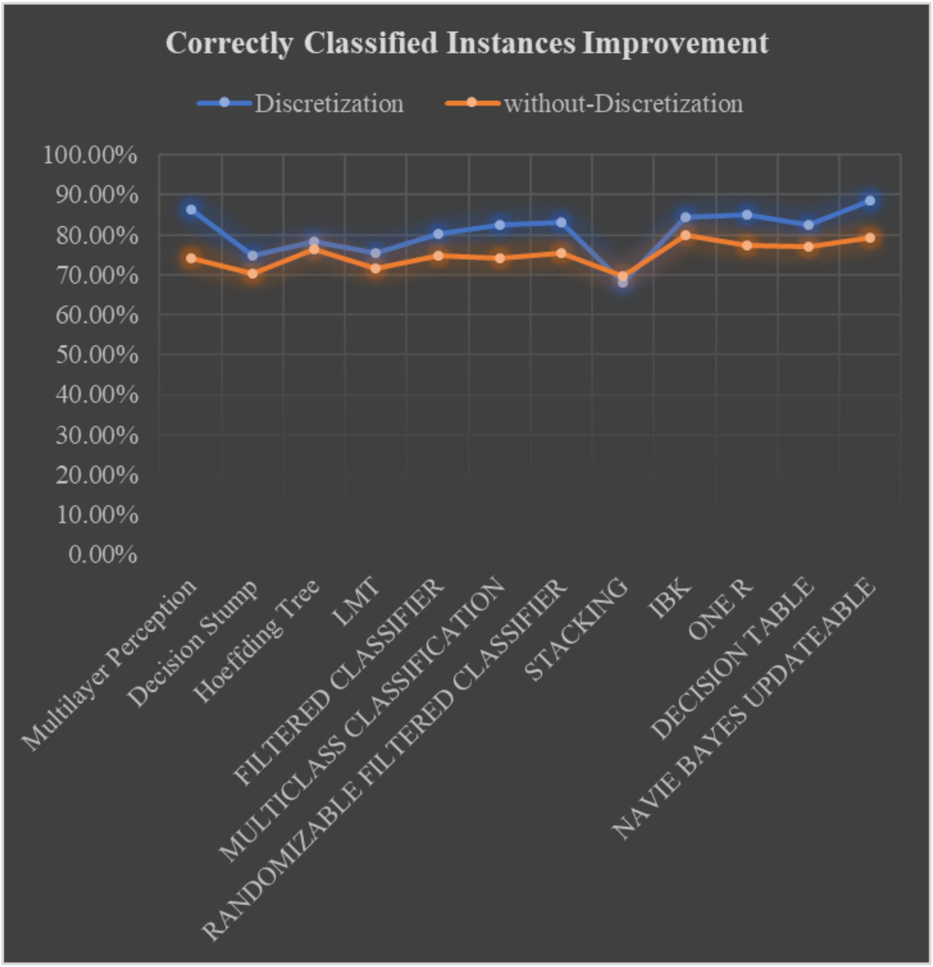


Table 2. Correctly Classified Instances Improvement (Accuracy)

Classifier	Discretization	Without Discretization	Improvement
Multilayer-Perceptron	86.45%	74.25%	12.20%
Decision-Stump	74.66%	70.20%	4.46%
Hoeffding-Tree	78.36%	76.49%	1.87%
LMT	75.55%	71.50%	4.45%
Filtered-Classifier	80.35%	74.86%	5.51%
Multiclass-Classification	82.45%	74.15%	8.30%
Randomizable-Filtered	83.15%	75.50%	7.65%
Stacking	68.15%	69.50%	1.35%
Ibk	84.25%	80.00%	4.25%
One-r	85.10%	77.33%	7.77%
Decision-table	82.35%	77.15%	5.20%
Naïve-Bayes-updateable	88.56%	79.40%	9.16%

Table 3. TP-Rate, F-Measure & AUC Accuracy

Classifiers	TP-Rate		F-Measure		Area Under Curve	
	Discretization	Without-Discretization	Discretization	Without -Discretization	Discretization	Without -Discretization
Multlaer-Percep	0.44	0.25	0.35	0.25	0.65	0.48
Decision Stump	0.21	0.18	0.21	0.18	0.48	0.38
Hoeffding-Tree	0.23	0.2	0.24	0.28	0.55	0.38
LMT	0.23	0.15	0.23	0.25	0.45	0.29
Filtered-Classifier	0.42	0.32	0.42	0.32	0.71	0.32
Mutclass-Classif	0.65	0.35	0.55	0.35	0.55	0.35
Randomizable-Fil	0.54	0.4	0.54	0.40	0.64	0.41
Stacking	0.144	0.18	0.25	0.29	0.73	0.49
Ibk	0.521	0.38	0.59	0.38	0.59	0.38
One-r	0.52	0.45	0.48	0.45	0.59	0.45
Decision-table	0.44	0.35	0.55	0.35	0.65	0.55
Naïve-Bayes	0.62	0.38	0.65	0.38	0.76	0.48

The dataset model that we utilized in our tests is shown in above table 1. All of these datasets are part of the software deformity prophecy, and the models of interest are both defect-prone and non-defect-prone models, respectively. For the categorization of these classes, we have employed a number of different classifiers. The tp-rate, the f-measure, the area under the curve, and the number of properly categorized cases are the assessment metrics we use. We have utilized a pre-processing method known as Discretization to improve the accuracy of these classifiers, and the results have been promising. From Fig1 to Fig4, we demonstrated that the accuracy and efficiency of the vast majority of classifiers may be improved or increased with the assistance of the discretization technique. With regard to TP-RATE analysis, we have found that the TP-RATE of Naive Bayes Updateable, Multiclass, Randomizable, Ibk, and oneR is enhanced. However, for a few classifiers, such as the decision stump, the hoeffding tree, and the lmt, the TP-rate could not be raised by a substantial amount. The use of filtered class, multilayer perceptron, and naive Bayes has shown us that they update their improvements very well and boost them extremely effectively in the event of positive accuracy and area under the curve.

While the efficiency and accuracy of classifiers such as decision stump, hoeffding tree, and lmt are not particularly good in the case of correctly classified instances where we can easily judge the improvement of each classifier, their efficiency and accuracy are improved when compared to when using these classifiers without discretization. In all instances, the progress efficiency and accuracy of Ibk,oneR, randomizable, multiclass, and Naive Bayes models have been significantly improved, with the exception of one. Although the stacking position seems to be very poor and not suitable for use in these studies due to the fact that their efficiency and accuracy have not improved, it appears to be worse in all instances. In every trial, it was apparent that no classifier could be considered ideal in terms of accuracy and efficiency when it came to software deformity predicting dataset models.

CONCLUSION

In our study for software deformity prophesy datasets, we used a data-preprocessing method known as Discretization, which is short for discretization. This is the approach we used to improve the classification accuracy of our software deformity prophesy dataset model. With the assistance of the discretization preprocessing technique, we were able to increase the accuracy of our observation and analysis by using several classifiers. In every trial, it was apparent that no classifier could be considered ideal in terms of accuracy and efficiency when it came to software deformity predicting dataset models. While the efficiency and accuracy of classifiers such as decision stump, hoeffding tree, and lmt are not particularly good in the case of correctly classified instances where we can easily judge the improvement of each classifier, their efficiency and accuracy are improved when compared to when using these classifiers without discretization. Although the stacking position seems to be very poor and not suitable for use in these studies due to the fact that their efficiency and accuracy have not improved, it appears to be worse in all instances.

REFERENCES

1. LeiQiaoa, Xuesong "Deep learning-based software defect prediction" Neurocomputing, Volume 385, 14 April 2020, Pages 100-110.
2. Nana Zhang, Kun Zhu " A Performance Fault Diagnosis Method for SaaS Software Based on GBDT Algorithm" International journal of Computers, Materials and Continua, Published - 20 May 2020
3. P. Singh, S. Verma, "An Investigation of the Effect of Discretization on Defect Prediction Using Static Measures," 2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies, Trivandrum, Kerala, 2009, pp.837-839.
4. Ishani Arora a, Vivek Tetarwala, Anju Sahaa at.al, Open issues in software defect prediction, Elsevier, 2015.
5. Ahmed H. Yousef, “Extracting software static defect models using data mining, Elsevier ,2015.
6. Wang, Shuo, Xin Yao. "Using class imbalance learning for software defect prediction." IEEE Transactions on Reliability 62.2 (2013): 434-443.
7. Ren, J., Qin, K., Ma, Y., & Luo, G. On software defect prediction using machine learning. Journal of Applied Mathematics,2014.