AN EXPERIMENTAL STUDY FOR SOFTWARE QUALITY PREDICTION WITH MACHINE LEARNING METHODS

[1]THORRIKONDA AMULYA,[2]D.SAI KRISHNA
[1]MCA Student, [2] Assistant Professor
Department Of MCA
Sree Chaitanya College of Engineering, Karimnagar

ABSTRACT:

Software quality estimation is an activity needed at various stages of software development. It may be used for planning the project`s quality assurance practices and for benchmarking. In earlier previous studies, two methods (Multiple Criteria Linear Programming and Multiple Criteria Quadratic Programming) for estimating the quality of software had been used. Also, C5.0, SVM and Neutral network were experimented with for quality estimation. These studies have relatively low accuracies. In this study, we aimed to improve estimation accuracy by using relevant features of a large dataset. We used a feature selection method and correlation matrix for reaching higher accuracies. In addition, we have experimented with recent methods shown to be successful for other prediction tasks. Machine learning algorithms such as Xgboost, Random Forest and Decision Tree are applied to the data to predict the software quality and reveal the relation between the quality and development attributes. The experimental results show that the quality level of software can be well estimated by machine learning algorithms.

1. INTRODUCTION:

Software applications may contain defects, originating from requirements analysis, specification and other activities conducted in the software development. Therefore, software quality estimation is an activity needed at various stages. It may be used for planning the project based quality assurance practices and for benchmarking. In addition, the number of defects per unit is considered one of the most important factors that indicate the quality of the software. There are two directly comparable studies on software quality prediction using defect quantities in ISBGS dataset. In the first study, the two methods (MCLP and MCQP) were experimented with the dataset and the results were compared. The quality level was classified according to: number of minor defect + 2*number of major defect + 4*number of extreme defect. The quality of level was to be either high or low. They used k-fold cross-validation technique to measure MCLP and MCQP's performance on the ISBSG database. Release 10 Dataset (released in January 2007) which contained 4,017 records and 106 attributes was used. After preprocessing, 374 records and 11 attributes remained in the dataset. In another study, the same data set was used again . The software belonged to high quality class if it fulfills the following requirements: the extreme defects exist or the number of major defects is more than 1 or the number of minor defects is more than 10. The rest are assumed to belong to low quality class.

After preprocessing, 746 projects and 53 attributes remained in the dataset. They used C5.0, SVM and Neutral network for classification. As an example to a more application oriented study Rashid used case based reasoning (CBR) for software quality estimation. CBR is a machine learning model which performs the learning process using the results of the previous experiments. Line of code, number of function, difficulty level, and development type and programmers experience are entered and these attributes are used for estimation. The deviation is calculated by using Euclidian distance (ED) or The Manhattan distance (MD). If the error in estimation is less than 10% then the record is saved to the database. Number of inputs that can be obtained from the user is limited. Also, it is necessary to have close values in the database in order to estimating precise values. In these studies, quality estimation was done by binary classification. We tried to improve these prediction models, taking into account the size in terms of function points and using 4-level classification. We have experimented with recent classification methods shown to be successful for other prediction tasks.

2. LITERATURE SURVEY:

"Software quality metrics in quality assurance to study the impact of external factors related to time."

This is a survey of software quality metrics. Definitions of the terms quality and software quality are presented. Difference between the use of software metrics and software quality assurance is shown. Importance of software quality is also discussed. Comparison of software metrics strengths and weaknesses are also described. View on software quality is also shown. Software metrics provide a quantitative basis for planning and predicting software development processes. Therefore the quality of software can be controlled and improved easily. Quality in fact aids higher productivity, which has brought software metrics to the forefront. A classification of software quality metrics is presented. The classification is: process, product and project metrics. This paper examines the realm of software engineering to see why software metrics are needed and also reviews their contribution to software quality and reliability. A number of different metrics relating to maintenance are described. The factor on which software quality depends are also presented. Quality aids higher productivity, which has brought software metrics to the forefront. This research paper deals with different views on software quality. Case study on software quality is also explained with an example. Results can be improved further as we acquire additional experience with variety of software metrics. These experiences can yield tremendous benefits in quality and reliability. Also discussed about automatic collection of software metrics data and costs of software metrics.

"Software defect prediction: do different classifiers find the same defects."

During the last 10 years, hundreds of different defect prediction models have been published. The performance of the classifiers used in these models is reported to be similar with models rarely performing

above the predictive performance ceiling of about 80% recall. We investigate the individual defects that four classifiers predict and analyze the level of prediction uncertainty produced by these classifiers. We perform a sensitivity analysis to compare the performance of Random Forest, Naïve Bayes, RPart and SVM classifiers when predicting defects in NASA, open source and commercial datasets. The defect predictions that each classifier makes is captured in a confusion matrix and the prediction uncertainty of each classifier is compared. Despite similar predictive performance values for these four classifiers, each detects different sets of defects. Some classifiers are more consistent in predicting defects than others. Our results confirm that a unique subset of defects can be detected by specific classifiers. However, while some classifiers are consistent in the predictions they make, other classifiers vary in their predictions. Given our results, we conclude that classifier ensembles with decision-making strategies not based on majority voting are likely to perform best in defect prediction.

"A Knowledge Discovery Case Study of Software Quality Prediction: ISBSG Database,"

Software becomes more and more important in modern society. However, the quality of software is influenced by many un-trustworthy factors. This paper applies MCLP model on ISBSG database to predict the quality of software and reveal the relation between the quality and development attributes. The experimental result shows that the quality level of software can be well predicted by MCLP Model. Besides, several useful conclusions have been drawn from the experimental result.

3. EXISTING SYSTEM:

Define the goal of your software. Determine how to measure the success of your software. identify what software quality metrics are important. Choose a test metric that will be easy to implement and analyze. Set up a system for collecting data on this test metric over time.

Disadvantage:

1. Tame taking process

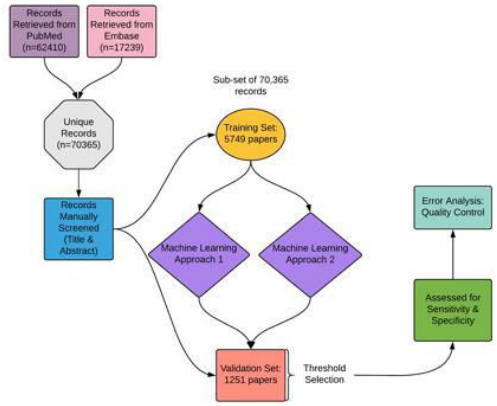
4. Proposed system:

At various stages of software development, software quality estimate is a necessary job. It could be utilized for organizing the project's quality control procedures and for benchmarking. In past investigations, two approaches (many Quadratic Multiple Criteria and Criteria Linear Programming Programming) for determining the level of software quality had been used. Also tested were C5.0, SVM, and neutral networks. for estimating quality. These studies have somewhat small accuracies. We sought to enhance estimation in this work. Accuracy by applying pertinent features from a sizable dataset. Utilizing a method for choosing features and correlation matrix for determining more accuracy. Furthermore, we have tried with current techniques that have been successfully used in other prediction projects. boost and Random Forest are examples of machine learning algorithms.

Advantages:

1. Time consumption is less

5. SYSTEM ARCHITECTURE



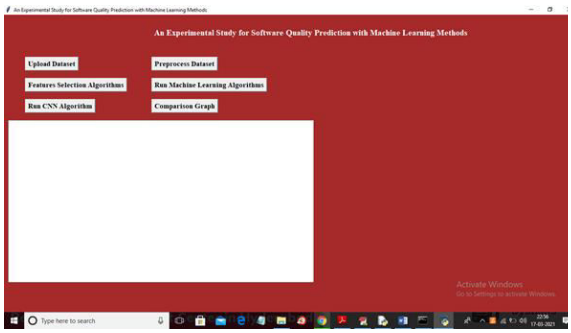
6. IMPLEMENTATION

Modules Information:

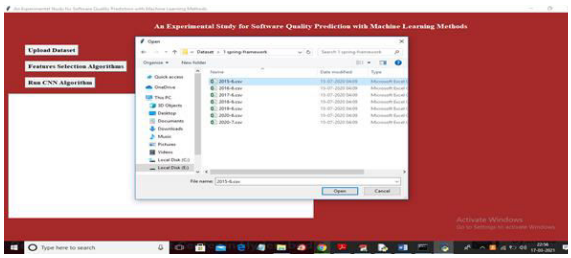
To implement this project we have designed following modules

1. **Upload Dataset :** If needed, select your dataset from list on the Datasets page to open its Import tab
2. **Preprocess Dataset:** Data preprocessing, a component of data preparation, describes any type of processing performed on raw data to prepare it for another data processing procedure. It has traditionally been an important preliminary step for the data mining process.
3. **Features Selection Algorithms**: Feature selection models are of two types: Supervised Models: Supervised feature selection refers to the method which uses the output label class for feature selection. They use the target variables to identify the variables which can increase the efficiency of the model.
4. **Run Machine Learning Algorithms:** A machine learning algorithm is the method by which the AI system conducts its task, generally predicting output values from given input data. The two main processes of machine learning algorithms are classification and regression
5. **Run CNN Algorithm:** CNN utilizes spatial correlations which exist with the input data. Each concurrent layer of the neural network connects some input neurons. This region is called a local receptive field. The local receptive field focuses on hidden neurons.
6. **Comparison Graph:** Comparison diagram or comparative diagram is a general type of diagram, in which a comparison is made between two or more objects, phenomena or groups of data. A comparison diagram or can offer qualitative and/or quantitative information. This type of diagram can also be called comparison chart or comparison chart.
7. **SCREEN SHOTS**

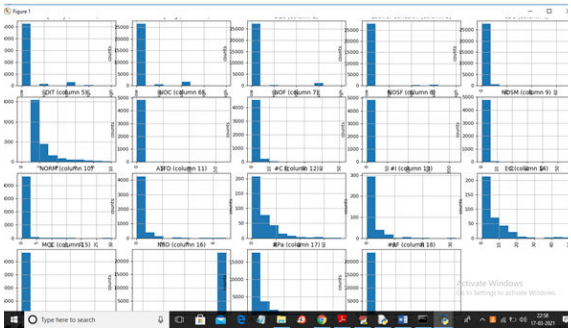
To run project double click on ‘run.bat’ file to get below screen



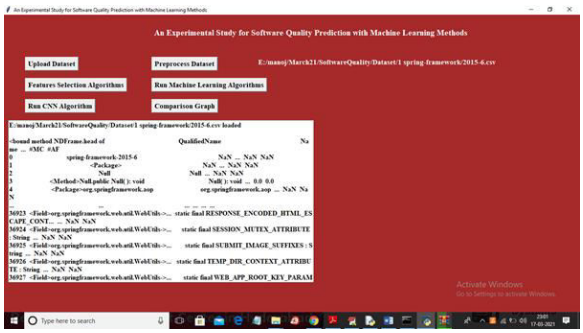
In above screen click on 'Upload Dataset' button to and upload dataset



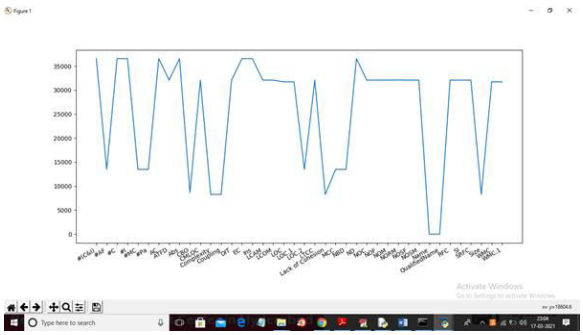
In above screen selecting and uploading '2015-6.csv' dataset file and then click on 'Open' button to load dataset and to get below screen



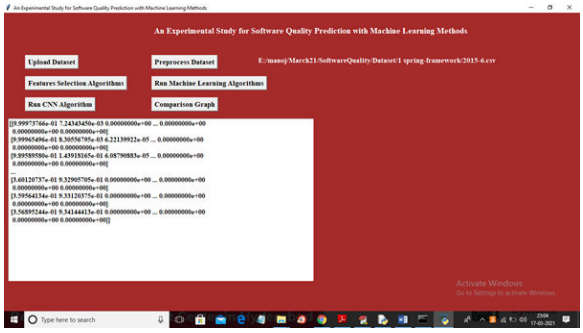
In above graph we can see each graph represents one column from dataset and from that columns its counting each distinct value from and plot in that graph for example in second graph NOC columns 3 different values and its plotting 3 different bars with count and no close above graph to get below screen



In above screen displaying values from dataset and we can see dataset contains NAN (missing values) and string non numeric values and we need to replace all missing and non-numeric values with their count so click on 'Preprocess Dataset' button



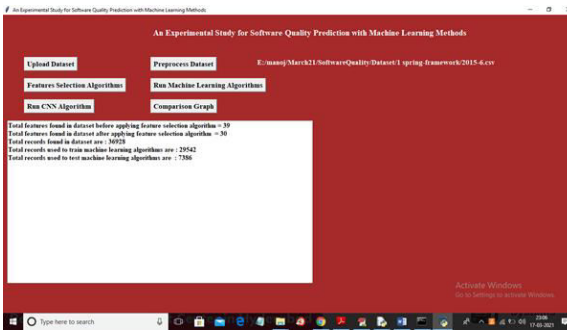
In above graph x-axis represents column names and y-axis represents total missing values counts in that column and now close above graph to get below screen



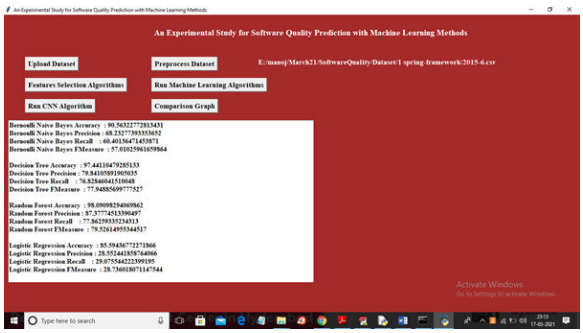
In above screen all missing an string values are replace with numeric values

and now click on 'Features Selection Algorithms' button to select important features from dataset and then split dataset into train and test part

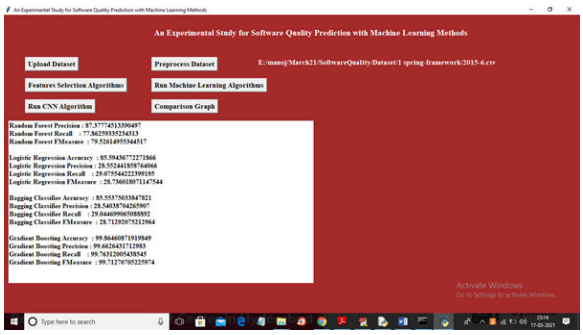
In above graph the box which contains value >0.5 will be consider as important attributes and now close above graph to get below screen



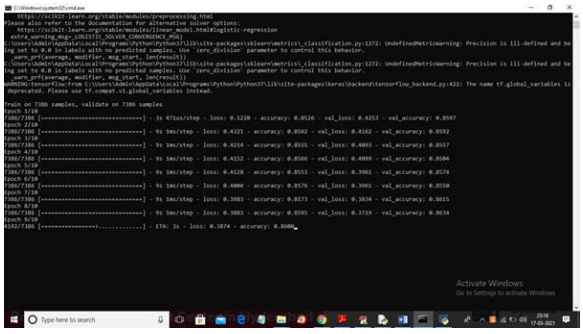
In above screen before applying feature selection algorithm dataset contains 39 features/columns and after applying PCA feature selection we got 30 important features and dataset contains 36928 records and application using 7386 records for testing and 29542 records for training and now both train and test dataset is ready and now click on 'Run Machine Learning Algorithms' button to run all machine learning algorithms



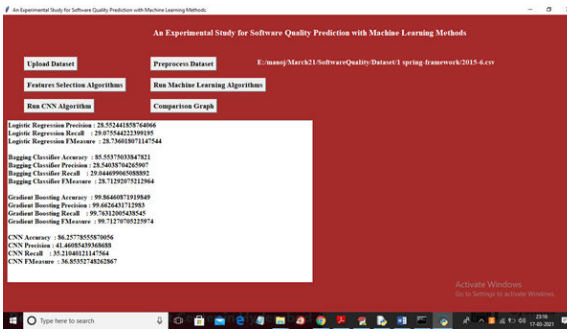
In above screen we can precision, recall, accuracy and fscore for all algorithms



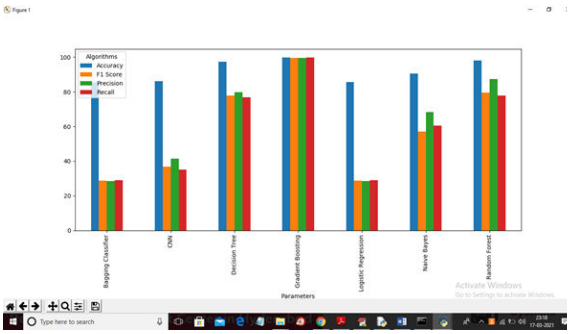
Now click on 'Run CNN Algorithm' button to run CNN algorithm and to get below screen



In above screen to train CNN we took 10 iterations or epoch and at each epoch accuracy get better and loss get reduce and after 10 iterations will get below screen



In above screen we got output values for CNN also and now click on 'Comparison Graph' button to get below screen



In above graph we are plotting accuracy, precision, recall and accuracy for each algorithm

8. CONCLUSION:

In this paper we have experimented classification algorithms using Scikit-learn library on two dataset. We have experimented with recent algorithms that support multi-class classification. The accuracies achieved by using these algorithms are 92.28% on EBSPM Dataset and 92.22% on ISBSG Dataset. In comparison to previous directly comparable studies, acceptable level multiclass quality prediction could be achieved.

9. REFERENCES:

[1] Vijay, T. John, D. M. G. Chand, and D. H. Done. "Software quality metrics in quality assurance to study the impact of external factors related to time." International Journal of Advanced Research in Computer Science and Software Engineering, 2017.

[2] D. Bowes, T. Hall, and J. Petrić, "Software defect prediction: do different classifiers find the same defects?." Software Quality Journal, 26(2), 2018, pp. 525-552.

[3] X. Wang, Y. Zhang, L. Zhang and Y. Shi, "A Knowledge Discovery Case Study of Software Quality Prediction: ISBSG Database," 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, Toronto, ON, 2010, pp. 219-222.

[4] X. Wang, Y. Zhang, L. Zhang and Y. Shi, "A Knowledge Discovery Case Study of Software Quality Prediction Based on Classification Models: ISBSG Database," The 11th International Symposium on Knowledge Systems Sciences (KSS 2010), 2010

[5] E. Rashid, S. Patnaik, and V. Bhattacherjee, "Software quality estimation using machine learning: Case-Based reasoning technique, " International Journal of Computer Applications, 2012

[6] www.isbsg.org

[7] https://goverdson.nl/

[8] H. Huijgens,"Evidence-based software portfolio management: a tool description and evaluation", 20th International Conference on Evaluation and Assessment in Software Engineering (EASE '16).