Substantiation of multinomial classification using ensemble learning approach

Lipika Goel¹, D Damodaran², Mayank Sharma¹ and Sunil Kumar Khatri¹

¹Amity University Uttar Pradesh, Noida, India
²Centre for Reliability, Thiruvanmiyur, Chennai India

lipika.bose@gmail.com, damodaran2@gmail.com, msharma22@amity.edu, sunilkkhatri@gmail.com

Abstract. The software defect prediction is the most operative research domain in software engineering as it enhances its reliability. The availability of defect data related to different projects leads to cross project defect prediction an open issue. This paper instanced on multiclass/multinomial classification of defect prediction on different categories of cross projects. The ensemble learning statistical models – Random forest and Gradient Boosting are used for classification. An empirical study is carried out to determine the predictive performance of the within project and cross project prediction models. Depending on the number of defects, class level information is classified into one of three defined multiclass. The homogeneous set of object oriented metrics is used for training the model. Furthermore, k-fold cross validation is done to evaluate the training accuracy of the statistical models. Major outcome of the paper concludes that multinomial/multiclass classification is applicable on cross project data and has comparable results to within project defect data with statistical significance.

1. Introduction

Identification of the defects prone classes before actual testing reduces the testing cost and efforts. It also leads to a more focused testing thereby enhancing the probability of fault free software. Much research work has been carried out in within projects data. The availability of many different projects data leads to the motivation of cross project defect prediction. In the past decade much research has focused on binary classification of CPDP. The multinomial classification in CPDP is still an open area to be focused.

The multinomial classification provides detailed information of severity of defect prone classes. It classifies the class level defect data into different groups depending on the total number of defects in each class. The Table1. summarizes the different categories of defect prediction.

The main research question addressed in this paper is Whether multinomial classification of CPDP is comparable to WPDP? This paper is divided into the following sections: Section 2 gives an overview of the literature survey. Section 3 summarizes the datasets, metrics and the models used. Section 4 states the performance measures used followed by Section 5 stating the methodology used in prediction. Section 6 tabulates the results. The conclusion is stated in the Section 7 of this paper.
| Category         | Description                                                                 |
|------------------|------------------------------------------------------------------------------|
| Strict CPDP      | No information of the testing data remains in the training dataset.          |
| Mixed CPDP       | Older version of the target data is added to the training dataset.           |
| Mixed CPDP+      | Few percentage of testing data is also included in training dataset.         |
| 10% training data|                                                                               |
| Pairwise CPDP    | Pairwise training and testing is carried out.                                |
| WPDP             | Certain percentage of the same project is used as training and the remaining percentage as testing. |

2. Background

Turhan et al [1] in 2009 used K-NN to minimize the data distribution difference followed by Naïve Bayes classifier for defect prediction. T. Zimmermann et al [2] in 2009 gave a new dimension to defect prediction, by proposing CPDP using Logistic Regression model. Ma et al. [3] in 2012 presented a novel approach Transfer Naïve Bayes for CPDP. Jaechang Nam et al. [4] in 2013 applied a method of Transfer Component Analysis for CPDP. Jaechang Nam et al. [4] in 2013 applied a method of Transfer Component Analysis for CPDP. Herbold [5] in 2013 proposed distance-based strategies (EM Clustering & NN Filtering) for selecting the training dataset. Nam et al. [9] and X. Y. Jing et al. [10] in 2015 provided solutions for HCPDP (heterogeneous). Also, projected a solution to heterogeneous CPDP. Duksan Ryuet al. [28] in 2016 gave a solution to the class imbalance problem by creating a novel model of SVM with cognitive boosting. The combined approach of Genetic and Ensemble learning gave good results. Zhang et al. [29] in 2016 used unsupervised classifier for heterogeneous CPDP. Xin Xia et al. [30] in 2016 proposed a Hybrid Model Reconstruction Approach. Steffen Herbold et al. [31] (2017) did a comparative study to benchmark cross project defect prediction approaches. He concluded that CPDP has still not reached to a level where it can be used for the application in practice.

3. Prerequisite knowledge

3.1. Dataset & Metrics

We have taken the well-defined datasets from the publicly available PROMISE repository (www.Promisedata.org) for NASA related projects and from http://opensecience.us. We have taken class level information of each dataset.

The description of the dataset is given in Table 2.

| Project  | Language | Total no. of classes | Defect prone classes |
|----------|----------|----------------------|----------------------|
| Ant 1.7  | Java     | 745                  | 168                  |
| Camel 1.0| Java     | 339                  | 13                   |
| Camel 1.2| Java     | 608                  | 216                  |
| Camel 1.4| Java     | 872                  | 145                  |
| Camel 1.6| Java     | 965                  | 189                  |
| Ivy1.1   | Java     | 111                  | 63                   |
| Ivy1.4   | Java     | 241                  | 17                   |
| Ivy2.0   | Java     | 352                  | 40                   |
| Jedit 3.2| Java     | 272                  | 47                   |
| Jedit 4.0| Java     | 306                  | 75                   |
| Jedit 4.1| Java     | 312                  | 79                   |
| Jedit 4.2| Java     | 367                  | 47                   |
| Jedit 4.3| Java     | 492                  | 11                   |
| Prop1    | Java     | 18471                | 2738                 |
| Prop2    | Java     | 23041                | 2431                 |
In this work we have focused on homogeneous CPDP for multinomial classification. The CK metrics [15] are selected and are used for software defect prediction.

The CK Metrics used in the analysis are as follows[22]:

- DIT (Depth Of Inheritance)
- WMC (Weighted Methods Per Class)
- CBO (Coupling Between Objects)
- NOC (Number of children)
- LCOM (Lack Of Cohesion Of Methods)
- RFC (Response For Class)
- LOC (Lines of Code)

3.2. Ensemble Learning Models

An ensemble contains a combination of base/weak learners to improve the performance of the model. The prediction accuracy of an ensemble is much higher than the base learners. Base learning algorithms can be neural network, decision tree or any other machine learning algorithm. The majority voting for classification and weighted averaging for the regression are the common strategies of combining base learners. The popular and effective ensemble methods are Bagging, Boosting and Stacking. The figure 1. gives a diagrammatic representation of ensemble approach.

![Figure 1. Ensemble Learning Approach.](image)

The ensemble learning models used in this experiment are Random Forest and Gradient Boosting. Random forest model is a bagging technique whereas Gradient Boosting is a boosting approach[21].

Random Forest: Random Forest is a solution to most of the data science problems. It is a flexible machine learning model which is competent to perform both classification and regression. It handles missing values, dimensionality reduction methods, outliers and thereby having better results. It is an ensemble of weak models (decision trees). To classify an object each tree votes, the forest chooses the classification which has the highest number of votes. For regression it computes the average of the output produced by each tree. The major advantage of Random forest is avoiding the problem of overfitting.

Given a training set: -

\[ X = x_1, x_2, \ldots, x_n \] with responses \( y_1, y_2, \ldots, y_n \). bagging it B times- selecting random samples.
For \( b = 1, 2, 3, \ldots, N \) we will train a regression or classification tree on \( X_b, Y_b \). After training the prediction of the sample \( x \) can be done either by averaging the prediction of all the regression trees or taking the majority votes in the case of classification trees.

\[
F = \frac{1}{N} \sum_{b=1}^{N} f_b(x')
\]

The standard deviation of the predictions from different regression trees gives the estimate of uncertainty of the prediction.

\[
\sigma = \sum (f_b(x') - F)^2 / (N - 1)
\]

Gradient Boosting: Gradient Boosting is an ensemble learning technique which has sequential predictors. In this technique the subsequent predictors enhance its learning by the faults of the previous predictors. It involves three elements:
- A loss function to be optimized.
- A weak learner to make predictions.
- Model to add weak learners.
Gradient boosting uses decision trees as weak learners. A tree is parameterized and its parameters are modified and then it is added to the model to minimize the loss.

For \( i = 1, \ldots, n \)

\[
F_0(x) = \arg \min \sum L(y_i, \gamma)
\]

\[
\gamma_m = \arg \min \sum L(y_i, F_{m-1}(x_i) + \gamma h_m(x_i))
\]

\[
F_m(x) = F_{m-1}(x) + \gamma m h_m(x).
\]

Where \((x_i, y_i)\) is the training set \(L(y,F(x))\) is a differential loss function, \(M\) is the no. of iterations, \(h_m(x)\) is the base learner, \(\gamma_m\) is the multiplier.

### 4. Performance measures

The Table 3 summarizes the performance measures considered for multinomial classification.

| Performance Measure | Description |
|---------------------|-------------|
| Precision           | It is the proportion of cases that are correctly identified to belong to class A among all the cases which classifier claims to belong to A. |
| Recall              | It is the proportion of cases that are correctly identified to belong to class A among all the cases that truly belong to A. |
| F-measure           | \((2 \times \text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})\) |
| Accuracy            | It is the ratio of all correctly identified and classified cases. |

### 5. Proposed methodology

We have conducted the experiment to classify our data of cross and within project into multi-classes. The three classes defined are as follows:
- Class 0: no bugs.
- Class 1: bugs are greater than 0 and less than 5.
- Class 2: bugs are greater than and equal to 5.

The technique used for learning and classification is Ensemble Learning approach. The models used are:
- Gradient Boosting
- Random Forest

The training datasets belong to different categories of CPDP. They are:
- Strict CPDP
- Mixed CPDP.
- Mixed with target Data.
- Pairwise CPDP.

Training and testing is also performed on within project dataset. Furthermore, we have carried out k-fold cross validation to evaluate the training accuracy of the data.

6. Results & Discussion

In this section we present results of the experiment conducted. The results are summarized in the below table. The performance evaluation measures used are accuracy, recall, precision, recall and f1-score. The k-fold cross validation is performed to evaluate the training accuracy of the datasets.

RQ: Whether multinomial classification of CPDP is comparable to WPDP?

From the Table 4 following observations are noted:

- The testing accuracy of CPDP was as high as 0.91 in the case of Strict CPDP. In the case of Mixed CPDP average accuracy was 0.88, 0.85 in the case of Mixed CPDP+10% WP data and 0.82 in pairwise CPDP. These values are comparable to the highest accuracy value of 0.93 achieved in the case of WPDP.
- Similarly, the value of precision was as high as 0.91 in the case of Strict CPDP. Values of 0.88 0.85, 0.76 was achieved in the case of Mixed CPDP, Mixed CPDP+10% WP data and Pairwise CPDP respectively. These values are comparable to the highest value of 0.91 achieved in the case of WPDP.
- On analyzing the value of Recall, the highest value was 0.93 for WPDP. The recall values of Strict, Mixed CPDP, Mixed CPDP+10% WP data and Pairwise CPDP was as high as 0.91, 0.88, 0.85 and 0.76 respectively. These values are thereby comparable to the recall value of WPDP.
- The F1 score in the case of Strict CPDP was 0.9 whereas for Mixed CPDP and Mixed CPDP+10% WP data the values were 0.83, 0.86 respectively. These values are also comparable to the value of 0.92 achieved in the case of WPDP.

The above stated observations infer the comparable performance of multinomial classification of CPDP to WPDP.

| Category | Training | Testing | Classifier | Class | Accuracy Training | Accuracy Testing | Precision Training | Precision Testing | Recall Training | Recall Testing | F1 score |
|----------|----------|---------|------------|-------|------------------|-----------------|-------------------|------------------|----------------|---------------|---------|
| Strict CPDP | jedit-3.2, jedit-4.0, jedit-4.1, jedit-4.2, jedit-4.3, prop1 | camel-1.4 | Random Forest | 0 | 0.89 | 0.77 | 0.85 | 0.9 | 0.87 |
| | | | 1 | 0.77 | 0.5 | 0.21 | 0.17 | 0.19 |
| | | | 2 | 0.75 | 0.75 | 0.78 | 0.77 | 0.76 |
| | | | AVG/TOTAL | 0.89 | 0.77 | 0.86 | 0.86 | 0.86 |
| | | | Gradient Boosting | 0 | 0.88 | 0.75 | 0.23 | 0.26 | 0.24 |
| | | | 1 | 0.75 | 0.25 | 0.06 | 0.01 |
| | | | AVG/TOTAL | 0.88 | 0.75 | 0.76 | 0.76 | 0.76 |
| | | | Random Forest | 0 | 0.93 | 0.81 | 0.96 | 0.85 | 0.9 |
| | | | 1 | 0.81 | 0.14 | 0.31 | 0.2 |
| | | | 2 | 0.79 | 0.21 | 0.05 | 0.12 |
| | | | AVG/TOTAL | 0.93 | 0.81 | 0.91 | 0.83 | 0.86 |
| | | | Gradient Boosting | 0 | 0.84 | 0.91 | 0.94 | 0.97 | 0.95 |
| | | | 1 | 0.91 | 0.17 | 0.06 | 0.09 |
| | | | 2 | 0.84 | 0.23 | 0.07 | 0.12 |
| | | | AVG/TOTAL | 0.84 | 0.91 | 0.89 | 0.91 | 0.9 |
| | | | Random Forest | 0 | 0.92 | 0.74 | 0.87 | 0.82 | 0.85 |
| | | | 1 | 0.78 | 0.44 | 0.47 | 0.46 |
| | | | 2 | 0.73 | 0.38 | 0.73 | 0.5 |
| | | | AVG/TOTAL | 0.92 | 0.74 | 0.76 | 0.74 | 0.75 |
| Mixed CPDP | ivy-1.1, camel-1.4, jedit-3.2, jedit-4.0, prop2 | jedit-4.1 | Random Forest | 0 | 0.91 | 0.77 | 0.79 | 0.99 | 0.88 |
| | | | 1 | 0.77 | 0.6 | 0.09 | 0.15 |
| | | | 2 | 0.74 | 0.6 | 0.55 | 0.57 |
| | | | AVG/TOTAL | 0.91 | 0.77 | 0.74 | 0.78 | 0.71 |
| | | | Gradient Boosting | 0 | 0.92 | 0.88 | 0.69 | 1 | 0.94 |
| | | | 1 | 0.88 | 0.27 | 0.01 | 0.02 |
| | | | 2 | 0.81 | 0.22 | 0.51 | 0.6 |
| | | | AVG/TOTAL | 0.92 | 0.88 | 0.81 | 0.88 | 0.83 |
The software reliability is directly proportional to probability of error free software. Software defect prediction leads to its more focused and rigorous testing. In this paper we analyzed the performance in multinominal classification of cross and within project defect prediction. The datasets were collected from the NASA projects. CK based object oriented metrics was effective for CPDP. We first labeled our class information in three different levels depending on the number of defects in each class. The experiment was conducted by training the model on different categories of cross project defect prediction. K-fold cross validation was done to estimate the predictive performance of the models. Ensemble learning approach validated the classification results. The results indicate that multinominal classification on CPDP is feasible and comparable to WPDP. There are many verticals on which the work can be extended. Multinominal classification of heterogeneous CPDP can be focused. Proper training dataset selection and class imbalance to optimize the performance of defect prediction are still open issues in CPDP.

### 7. Conclusion & Future scope

The software reliability is directly proportional to probability of error free software. Software defect prediction leads to its more focused and rigorous testing. In this paper we analyzed the performance in multinominal classification of cross and within project defect prediction. The datasets were collected from the NASA projects. CK based object oriented metrics was effective for CPDP. We first labeled our class information in three different levels depending on the number of defects in each class. The experiment was conducted by training the model on different categories of cross project defect prediction. K-fold cross validation was done to estimate the predictive performance of the models. Ensemble learning approach validated the classification results. The results indicate that multinominal classification on CPDP is feasible and comparable to WPDP. There are many verticals on which the work can be extended. Multinominal classification of heterogeneous CPDP can be focused. Proper training dataset selection and class imbalance to optimize the performance of defect prediction are still open issues in CPDP.
References

[1] B. Turhan, T. Menzies, A. B. Bener, and J. Di Stefano, “On the relative value of cross-company and within-company data for defect prediction,” Empirical Software Engineering, 14(5):540-578, 2009.

[2] T. Zimmermann, N. Nagappan, H. Gall, E. Giger, and B. Murphy, “Cross-project defect prediction,” In Proceedings of the 7th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE), pages 91-100, 2009. O

[3] Y. Ma, G. Luo, X. Zeng, and A. Chen, “Transfer learning for cross-company software defect prediction. Information and Software Technology”, 54(3):248-256, 2012. O

[4] J. Nam, S. J. Pan, and S. Kim, “Transfer defect learning,” In Proceedings of the 35th International Conference on Software Engineering (ICSE), pages 382-391, 2013. O

[5] S. Herbold, “Training data selection for cross-project defect prediction” In Proceedings of the 9th International Conference on Predictive Models in Software Engineering (PROMISE), pages 6-16, 2013. O

[6] F. Peters, T. Menzies, and A. Marcus, “Better cross company defect prediction” In 10th IEEE Working Conference on Mining Software Repositories (MSR), pages 409-418, 2013. O

[7] Panichella, R. Oliveto, and A. De Lucia, “Cross-project defect prediction models: L’Union fait la force,” In Software Evolution Week-IEEE Conference on Software Maintenance, Reengineering and Reverse Engineering (CSMR-WCRE), pages 164-173, 2014.

[8] D. Ryu, O. Choi, & J. Baik, “Value-cognitive boosting with a support vector machine for crossproject defect prediction,” 2016 Empirical Software Engineering. doi:10.1007/s10664-014-9346-4.

[9] J. Nam, and S. Kim, “Heterogeneous Defect Prediction” Proceeding ESEC/FSE 2015 Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering (pp. 508-519). New York, NY, USA: ACM. doi:10.1145/2786805.2786814 D

[10] S.J. Pan, Q. Yang, “A survey on transfer learning, Technical Report” HKUST-CS 08-08, Department of Computer Science and Engineering, Hong Kong University of Science and Technology, 2008.

[11] S. J. Pan and Q. Yang, “A survey on transfer learning”. IEEE Trans. on Knowl. and Data Eng., 22:1345–1359, October 2010.

[12] S. Kim, H. Zhang , R. Wu, & L. Gong, “Dealing with noise in defect prediction”. 2011 33rd International Conference on Software Engineering (ICSE) (pp. 481 - 490). Honolulu, HI: IEEE. doi:10.1145/1985793.1985859

[13] M. D’Ambros, M. Lanza, and R. Robbes. “Evaluating defect prediction approaches: A benchmark and an extensive comparison”. Empirical Software Engg., 17(4-5):531–577, Aug. 2012.

[14] M. H. Halstead. “Elements of Software Science (Operating and Programming Systems Series)”. Elsevier Science Inc., New York, NY, USA, 1977.

[15] S. R. Chidamber and C. F. Kemerer. “A metrics suite for object oriented design”. IEEE Trans. Softw. Eng., 20:476–493, June 1994. D

[16] T. McCabe. “A complexity measure. Software Engineering”, IEEE Transactions on, SE-2(4):308–320, Dec 1976.

[17] J. Pan and Q. Yang. “A survey on transfer learning”. IEEE Trans. on Knowl. and Data Eng., 22:1345–1359, October 2010. D

[18] X. Y. Jing, F. Wu, X. Dong, F. Qi, and B. Xu, — “Heterogeneous cross company defect prediction by unified metric representation and cca- based transfer learning,” In Proceedings of the 10th Joint Meeting on Foundations of Software Engineering, 2015, pp. 496–507 D

[19] G. Canfora, A. De Lucia, M. Di Penta, R. Oliveto, A. Panichella, and S. Panichella. “Multi-objective cross-project defect prediction”. In IEEE 6th International Conference on Software Testing, Verification and Validation (ICST), pages 252-261, 2013.
[20] Z. He, F. Shu, Y. Yang, M. Li, and Q. Wang. “An investigation on the feasibility of cross-project defect prediction.” Automated Software Engineering, 19(2):167-199, 2012.

[21] W. Dai, Q. Yang, G. Xue, & Y. Yu, “Boosting for transfer learning. In Proceedings of the 24th international conference on Machine learning” 2007 (pp. 193–200). http://dl.acm.org/citation.cfm?id= 1273521. Accessed February 25, 2014.

[22] Singh, P., Verma, S., & Vyas, O.”Cross Company and within Company Fault Prediction using Object Oriented Metrics”. 2013, International Journal of Computer Applications, 5-11.

[23] K. Dejaeger, “Toward Comprehensible Software Fault Prediction Models Using Bayesian Network Classifiers,”. IEEE Transactions on Software Engineering, 39(2), 2013, PP. 237–257.

[24] K. O. Elish, and M. O. Elish,“Predicting defect-prone software modules using support vector machines,” Journal of Systems and Software, 81(5), 2008, pp. 649–660. doi:10.1016/j.jss.2007.07.040.

[25] O. Mizuno, Y. Hirata, “A Cross-Project Evaluation of Text-Based Fault-Prone Module Prediction,”. Proceeding IWESEP '14 Proceedings of the 2014 6th International Workshop on Empirical Software Engineering in Practice (pp. 43-48). Washington, DC, USA: IEEE Computer Society.

[26] B. Turhan, A. Misirl, & A. Bener, “Empirical evaluation of the effects of mixed project data on learning defect predictors. Information and Software Technology.”, 2013, 55(6), 1101–1118. doi:10.1016/j.infsof.2012.10.003

[27] P. He, B. Li, & Y. Ma, “Towards Cross-Project Defect Prediction with Imbalanced Feature Sets,”. Cornell University Library, 2014.

[28] D. Ryu, J-I.Jang, & J. Baik, “A transfer cost-sensitive boosting approach for cross-project defect prediction,”. Software Quality Journal, 2015, PP.1-38. doi:10.1007/s11219-015-9287-1

[29] Zhang, Feng, et al. "Cross-project defect prediction using a connectivity-based unsupervised classifier." Proceedings of the 38th International Conference on Software Engineering. ACM, 2016.

[30] Xia, Xin, et al. "Hydra: Massively compositional model for cross-project defect prediction." IEEE Transactions on software Engineering 42.10 (2016): 977-998.

[31] Steffen Herbold et al. “A Comparative Study to Benchmark Cross-project Defect Prediction Approaches”. Ieee Transactions On Software Engineering, 2017.