Data Mining 2020

Assignment 1

Classification Trees, Bagging and Random Forests

Bentis Nikolaos (6662889) Polos Markos (6943721) Tsogias Ioannis (6966985)

1. A short description of the data.

The dataset contains defects from the bug database of Eclipse to source code locations. It lists the number of pre- and post-release defects for every package and file in the Eclipse releases 2.0, 2.1, and 3.0 **[1]**. For this task, only a subset of the data is used as all metrics except the ones listed in Table 1 of the accompanying article and the number of pre-release bugs, were removed. The remaining features contain information about several complexity metrics that were computed for classes or methods by using average (avg), maximum (max), and accumulation (sum) to file and package level.

Furthermore, "post" was preprocessed since it will be used as the target variable and needs to become categorical (all values that were not equal to 0 were transformed into 1). In this assignment, the package level data will be analyzed, using release 2.0 as the training set, and release 3.0 as the test set. The same preprocessing was followed for both sets. The final database consists of 41 predictor variables and the target variable "post".

2. A picture of the first two splits of the single tree

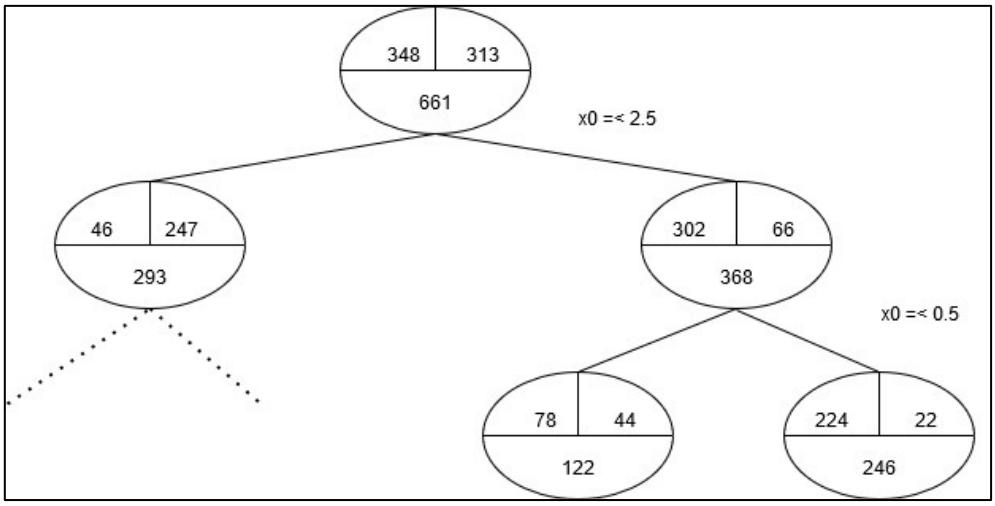

Figure 1. First two splits of the classification tree

As can be seen from the simplified Classification Tree presented on **Figure 1**, the first two splits occur on the "x0" variable, which is unusual considering that there were 41 different variables and both times splitting on variable "x0" led to splits with lower impurity. On the first split, if "x0" is higher than 2.5 the instance will be classified as "1", agreeing with 0.843 of the cases on that leaf node. When "x0" is lower or equal than 2.5 another splitting occurs on "x0" at the 0.5 value. When an instance has "x0" lower or equal than 0.5 the "0" class will be assigned to it, as 0.91 of the instances of the training set. When an instance has "x0" between 0.5 and 2.5 it belongs to an impure node where 44 of the instances belong to class "1" and 78

on “0”. These cases will be classified as “0”, but they will agree with only 0.63 of the cases from the training set.

The fact that the Classification Tree performs the first two splits on the same variable shall now be further investigated. “x0” corresponds to the variable “pre” which counts the pre-release defects for every package and file in the Eclipse releases. Intuitively, “pre” must be the most crucial predictive variable in this dataset. It is logical that if one wants to predict whether there exist post-release defects or not, he/she must know the number of the pre-release defects. After seeing the first two splits, a correlation(pearson) test was conducted between “pre” and “post” in order to confirm or reject our intuitive hypothesis. The correlation test indeed showed that there is high correlation between the two variables that equals to 0.807 which means that when “pre” is low then “post” will most likely be equal to 0 and when the first’s value is high, “post” should be equal to 1. Therefore, these classification rules make sense, given the meaning of the attributes.

3. Single Classification Tree, Bagging and Random Forests Analyses.

The analysis of three models will be presented in this section. The first model is a **Single Classification Tree** with *nmin = 15, minleaf = 5* and *nfeat = 41*. For the second model **Bagging** was used with *m = 100* and the same parameters as the first model. The last model used **Random Forests** with the same parameters as Bagging except that *nfeat = 6*, that is $\sqrt{41}$ rounded to the nearest integer. **Table 1** showcases the required quality measures (***Accuracy, Precision, Recall***) and **Tables 2-4** are the ***Confusion Matrices*** for each model.

	Single Classification Tree	Bagging	Random Forests
Accuracy	0.906	0.937	0.931
Precision	0.919	0.956	0.955
Recall	0.878	0.910	0.897

Table 1 Quality measures for the three requested models

As Table 1 indicates, the best result came from the Bagging model that achieved 0.937 accuracy, which was the highest among the three models. However, the fact that this model required about 15 minutes for the training process is something that one should take under consideration before naming it “the best model” out of the three, as Bagging needed 8x(RandomForests_Time) and the comparison with the Single Classification Tree seems unnecessary since this model’s training was a matter of a few seconds.

	Predicted 1	Predicted 0
Actual 1	324	24
Actual 0	38	275

Table 2 Confusion Matrix for the Single Classification Tree

	Predicted 1	Predicted 0
Actual 1	335	13
Actual 0	28	285

Table 3 Confusion Matrix for the Bagging model

	Predicted 1	Predicted 0
Actual 1	335	13
Actual 0	32	281

Table 4 Confusion Matrix for the Random Forests model

4. McNemar's Test

In this section a discussion of whether the differences in accuracy found on the test set are statistically significant, will take place. For this purpose, we've found a statistical test in order to compare all models and observe whether the null hypothesis H_0 is being rejected or not and that is the McNemar's test. For this analysis, a "corrected" formula was used since approximately 1 year after Quinn McNemar published the McNemar Test **[2]**, Edwards **[3]** proposed a continuity corrected version, which is the more commonly used variant today.

In McNemar's Test, we formulate the null hypothesis that the probabilities p(b) and p(c) are the same, or in simplified terms: None of the two models performs better than the other. Thus, the alternative hypothesis is that the performances of the two models are not equal. In order to perform such a test, it is required to form 3 different contingency matrices that contain information about the instances that the classifiers both predicted or not predicted correctly and the times that one of them predicted correctly while the other one did not. The McNemar test statistic ("chi-squared") can be computed as follows (**Figure 2**):

$$\chi^2 = \frac{(|\, b - c \,| - 1)^2}{(b + c)}.$$

Figure 2. Mc Nemar's Chi squared formula

Tables 5-7 showcase the contingency matrices for all three comparisons. Indexes (a) – (d) in **Table 5** are used to help the reader figure out which values are actually used in the aforementioned formula.

	BAG correct	BAG incorrect
SCT correct	(a) 584	(b) 15
SCT incorrect	(c) 36	(d) 26

Table 5. Contingency matrix for Single Classification Tree and Bagging model

	BAG correct	BAG incorrect
RF correct	606	10
RF incorrect	14	31

Table 6. Contingency matrix for Random Forests and Bagging model

	SCT correct	SCT incorrect
RF correct	577	39
RF incorrect	22	23

Table 7. Contingency matrix for Random Forests and Bagging model

The chi-squared formula was applied on the contingency matrices and the results are presented in **Table 8**.

	SCT vs BAG	RF vs BAG	RF VS SCT
X^2	7.843	0.375	4.196
pvalue	0.0005	0.54	0.04

Table 8. Mc Nemar's Test Results

In order to test the null hypothesis that the predictive performance of two models are equal a significance level of $a = 0.05$ was used. As one may see from the table above, $pavlue < a$ in the comparisons of Single Classification Tree against Bagging and Random Forests so in these

cases the null hypothesis that both models perform equally well on this dataset and the differences in accuracy found on the test set are statistically significant, since the p-value is, in both cases, smaller than α.

In the third case, where Random Forests and Bagging are being compared, there was a slight difference in the way that pvalue was computed since an exact binomial test is recommended for small sample sizes ($b+c<25$ **[4]**) as the chi-squared value is may not be well-approximated by the chi-squared distribution. The exact p-value can be computed as follows **(Figure 3)**:

$$p = 2\sum_{i=b}^{n} \binom{n}{i} 0.5^i (1-0.5)^{n-i}$$

Figure 3. Formula for the exact binomial test

So, after computing the exact pvalue for this particular case which is showcased in Table 8, it was observed that $pvalue = 0.54 >> a$ which indicates that we cannot reject our null hypothesis and assume that there is no significant difference between the two predictive models.

5. References

[1] Predicting Defects for Eclipse [Revised for Dataset Version 2.0a], Thomas Zimmermann , Rahul Premraj , Andreas Zeller.

[2] McNemar, Quinn, 1947. "Note on the sampling error of the difference between correlated proportions or percentages". Psychometrika. 12 (2): 153–157.

[3] Edwards AL: Note on the “correction for continuity” in testing the significance of the difference between correlated proportions. Psychometrika. 1948, 13 (3): 185-187.

[4] https://en.wikipedia.org/wiki/McNemar%27s_test