In this report, we will describe a classification tree which predicts whether post release bugs in eclipse have been reported or not. The classification tree uses chosen metrics as well as the number of pre-released bugs, and is trained on the Eclipse bug data set. We found that the classification rule that is used eventually is \_\_\_ and the differences in accuracy are/are not statistically significant.

**Data description**The classification tree will trained and tested with selected subsets from the Eclipse bug data set. This set was computed and made publicly available by Zimmermann, Premraj and Zeller in 2007 (Zimmermann, 2007). We have used the data on the package level, with the release 2.0 data for training, and the release 3.0 data for testing.

In figure 1 an overview of the number of cases in the data is given.

|  |  |
| --- | --- |
| Release version | Number of cases in the package |
| 2.0 | 377 |
| 3.0 | 661 |

Per case, we look at the number of reported pre-release defects and at their selected complexity metrics. These consist of 14 variables, listed in figure 2

The complexity metrics are listed in table 1, and in total this means 41 different predictor variables are used.

Insert table with numbers/metrics table?

@inproceedings{zimmermann-promise-2007,

title = "Predicting Defects for Eclipse",

author = "Thomas Zimmermann and Rahul Premraj and Andreas Zeller",

year = "2007",

month = "May",

booktitle = "Proceedings of the Third International Workshop on Predictor Models in Software Engineering",

location = "Minneapolis, MN, USA",

}