On the Evaluation of Defect Prediction Models

Thilo Mende

Software Engineering Group, University of Bremen, Germany (Alumni)

15th COW
24.10.2011

Many different eperimental setups are used in the literature

A review of 107 papers shows:

- A large range of data set sizes
- More than 11 different **evaluation measures**
- 7 different **resampling schemes**
- Only few comparisons against simple **baseline models**

Many different eperimental setups are used in the literature

A review of 107 papers shows:

- A large range of data set sizes
- More than 11 different **evaluation measures**
- 7 different **resampling schemes**
- Only few comparisons against simple **baseline models**

Question: What influence does this have on the stability of results and the practical predictive benefits?

Experimental Setup

Data Sets

- NASA MDP
- AR (Tosun et al., 2009)
- Eclipse (Zimmermann et al., 2007)

# of instances 36–17000

Algorithms

- RPart
- RandomForest
- GLM/Logistic Regression
- Naive Bayes

One important aspect is conclusion stability

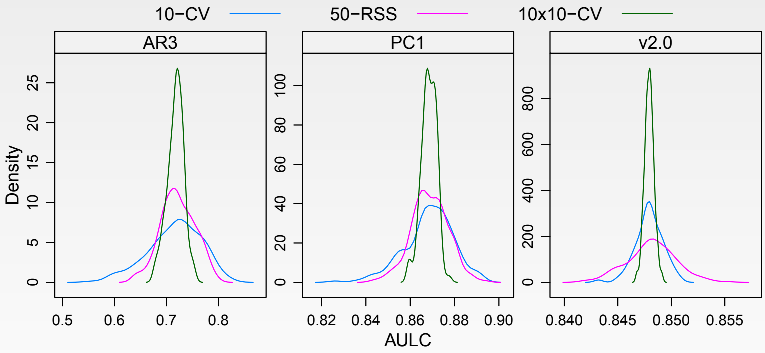
- **Stability:** Consistent results for repeated executions
  - Ensure reproducability
  - Protect against cherry picking
- **Randomization** (due to resampling or the learning algorithm) may lead to **unstable results**

In the following, we use 200 runs for each algorithm on each data set

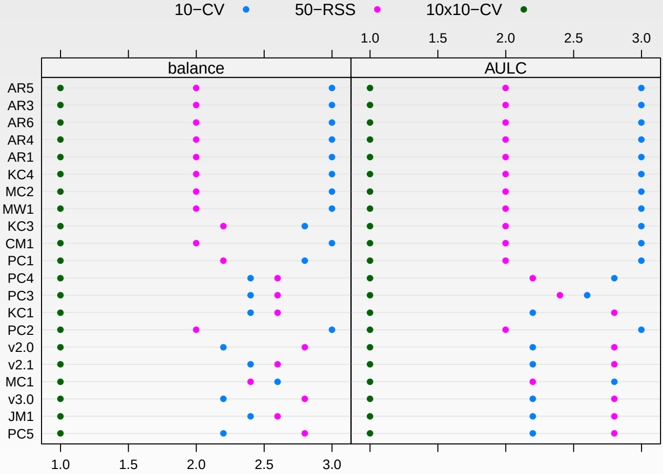
Three resampling schemes are often used

- 10-fold Cross Validation (10-CV)
- 50-times repeated random split (50-RSS)
- 10-times 10-fold Cross Validation (10×10-CV)

Resampling schemes differ in terms of variance

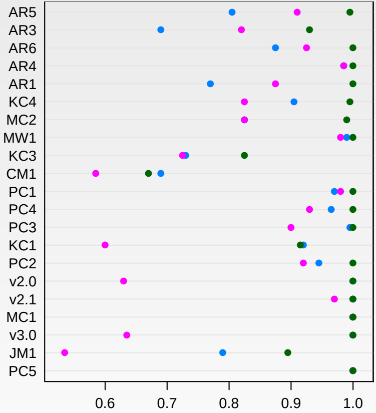


Resampling schemes differ in terms of variance (contd.)



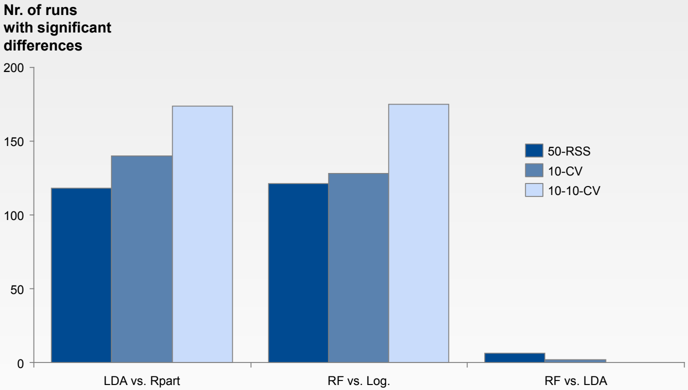
Ranking according to variance

Does higher variance matter?



Consistency when comparing Logistic Regression and LDA

The variance has an influence, e.g. when Demsar’s test is used



There are more sources of variance

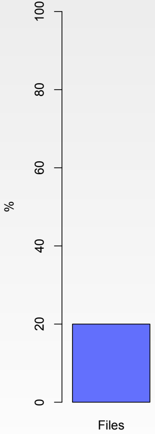
- Evaluation measures
- Class Imbalance
- …

There are more sources of variance

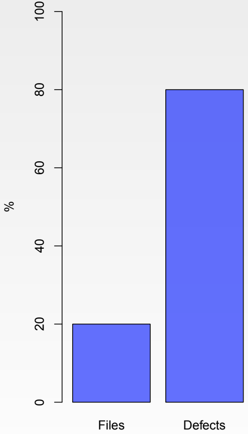
- Evaluation measures
- Class Imbalance
- …

… so one has to be careful to get **reproducable** results

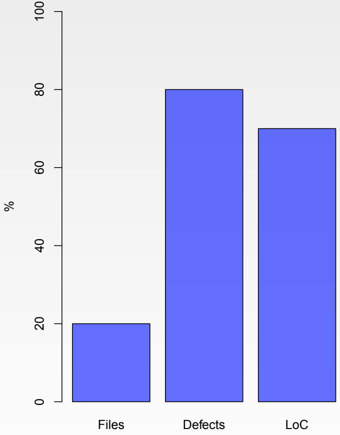
Traditional Performance Measures are often optimistic



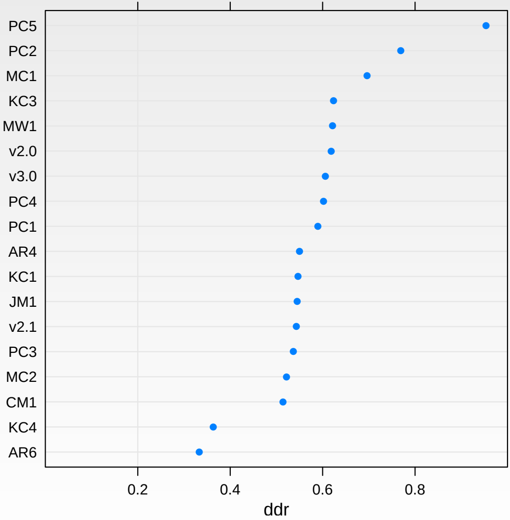
Traditional Performance Measures are often optimistic



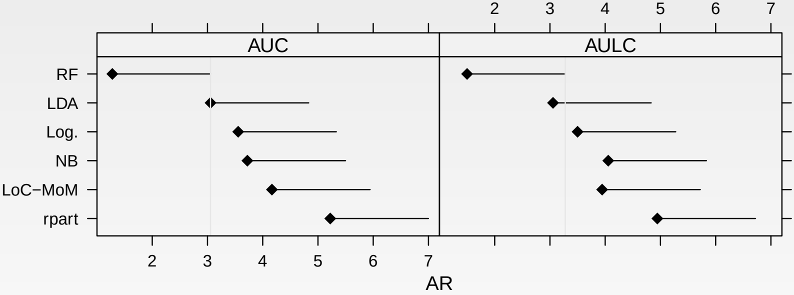
Traditional Performance Measures are often optimistic



The largest 20% of the files contain most of the defects

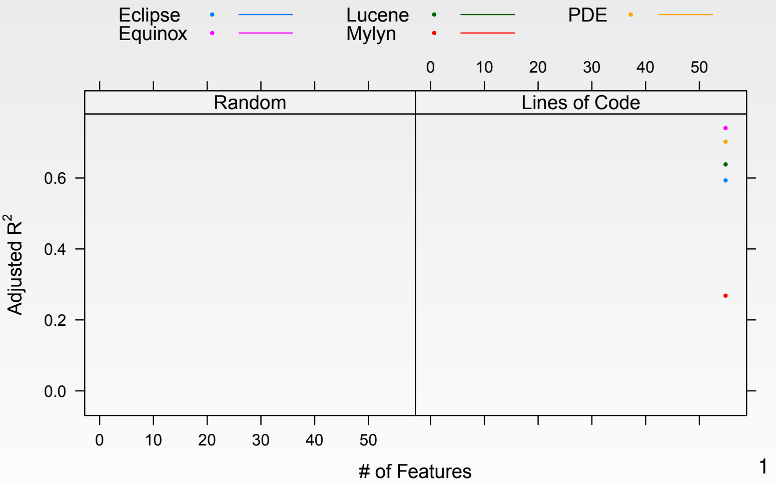


. . . only RandomForst performs significantly better



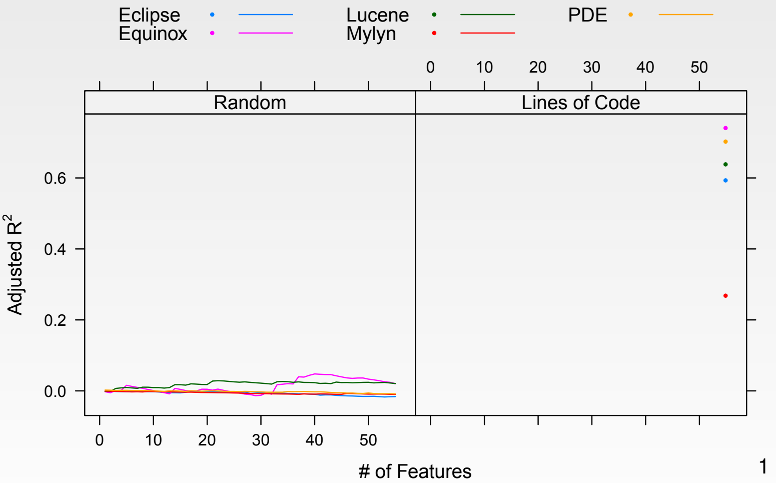
General pattern is consistent across data sets (Mende et al., 2009; Mende, 2010; Mende et al., 2011)

Random features can perform well for regression models



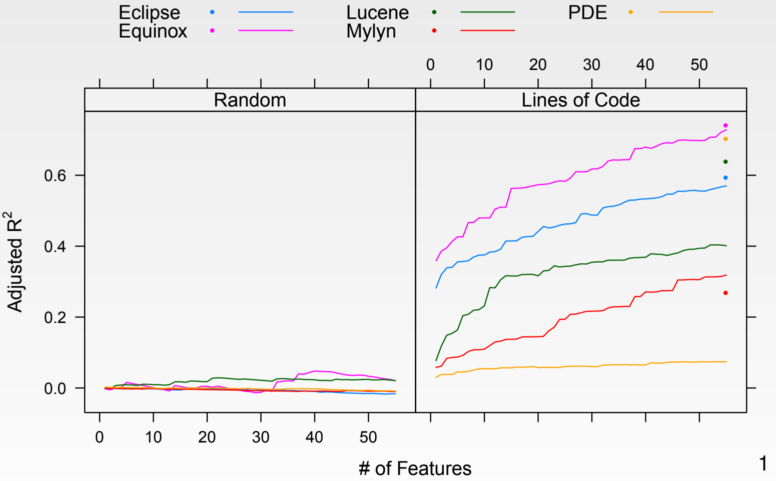
[1] These results are based on data sets provided by D'Ambros et al. (2010)

Random features can perform well for regression models



[1] These results are based on data sets provided by D'Ambros et al. (2010)

Random features can perform well for regression models



[1]

[1] These results are based on data sets provided by D'Ambros et al. (2010)

Recommendations

In general

- Use **10×10-CV** (with stratification)
- Use simple models as **benchmarks**
- Consider the treatment effort

Recommendations

In general

- Use **10×10-CV** (with stratification)
- Use simple models as **benchmarks**
- Consider the treatment effort

For SBSE

- Use simple models as **benchmarks**
- Consider the **variance**

  → to avoid cherry picking

Recommendations

In general

- Use **10×10-CV** (with stratification)
- Use simple models as **benchmarks**
- Consider the treatment effort

For SBSE

- Use simple models as **benchmarks**
- Consider the **variance**

  → to avoid cherry picking

Opportunity for SBSE: Identified defects vs. treatment effort?

On the Evaluation of Defect Prediction Models

Thilo Mende
Software Engineering Group, University of Bremen, Germany (Alumni)
tmende@informatik.uni-bremen.de

Universität Bremen

Poll: How do you calculate F1?

…when there are **invalid** partitions

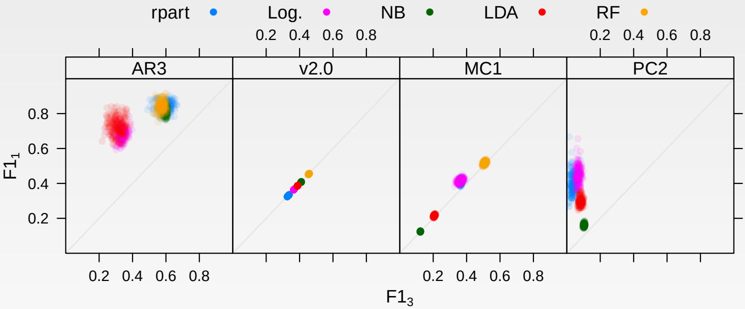
1. Average over all partitions, ignoring invalid ones
2. Average over all partitions, using 0 for invalid ones

Poll: How do you calculate F1?

. . . when there are **invalid** partitions

1. Average over all partitions, ignoring invalid ones
2. Average over all partitions, using 0 for invalid ones
3. Calculate TP/FP/FN per partition, and calculate F1 across all partitions? (Forman and Scholz, 2010)

Ooops. . .



D'Ambros, M., M. Lanza, and R. Robbes (2010). An extensive comparison of bug prediction approaches. In *MSR*. IEEE Computer Society.

Forman, G. and M. Scholz (2010, November). Apples-to-apples in cross-validation studies: pitfalls in classifier performance measurement. *ACM SIGKDD Explorations Newsletter 12*, 49–57.

Mende, T. (2010). Replication of defect prediction studies: Problems, pitfalls and recommendations. New York, NY, USA, pp. 1–10.

Mende, T., R. Koschke, and M. Leszak (2009, March). Evaluating defect prediction models for a large, evolving software system. pp. 247–250.

Mende, T., R. Koschke, and J. Peleska (2011). On the utility of a defect prediction model during HW/SW integration testing: A retrospective case study. pp. 259–268.

Tosun, A., B. Turhan, and A. Bener (2009). Validation of network measures as indicators of defective modules in software systems. New York, NY, USA, pp. 1–9. ACM.

Zimmermann, T., R. Premraj, and A. Zeller (2007). Predicting defects for Eclipse. IEEE Computer Society.