# Vulnerability prediction modeling

## Summary

In this class exercise, we will introduce you to vulnerability prediction modeling. Part 1 should be performed without the aid of electronic devices.

## Prerequisites

* MATLAB
* VPM project files (see <https://gitlab.com/reubenajohnston/vpm-case-study/-/archive/master/vpm-case-study-master.zip>)

## Details

### Part 1

* List ten characteristics that might influence the creation of vulnerabilities
* Data mining and machine learning
* Study the vulnerability prediction diagram below and discuss its flow amongst your group.



Figure -Machine learning based prediction model (Nguyen, Tran, 2010)

* Discuss how this machine learning technique could be used with the characteristics discussed previously to assist the location of vulnerabilities. In a few sentences, explain the approach.

### Part 2

VPM case study from Theisen et al, 2015



Figure -VPM data flow

#### Data collection

* Christopher Theisen provided a partial dataset, a snippet is below and shows filenames, number of crashes, code-churn, number of unique authors, and a classification indicating the file contained a security flaw

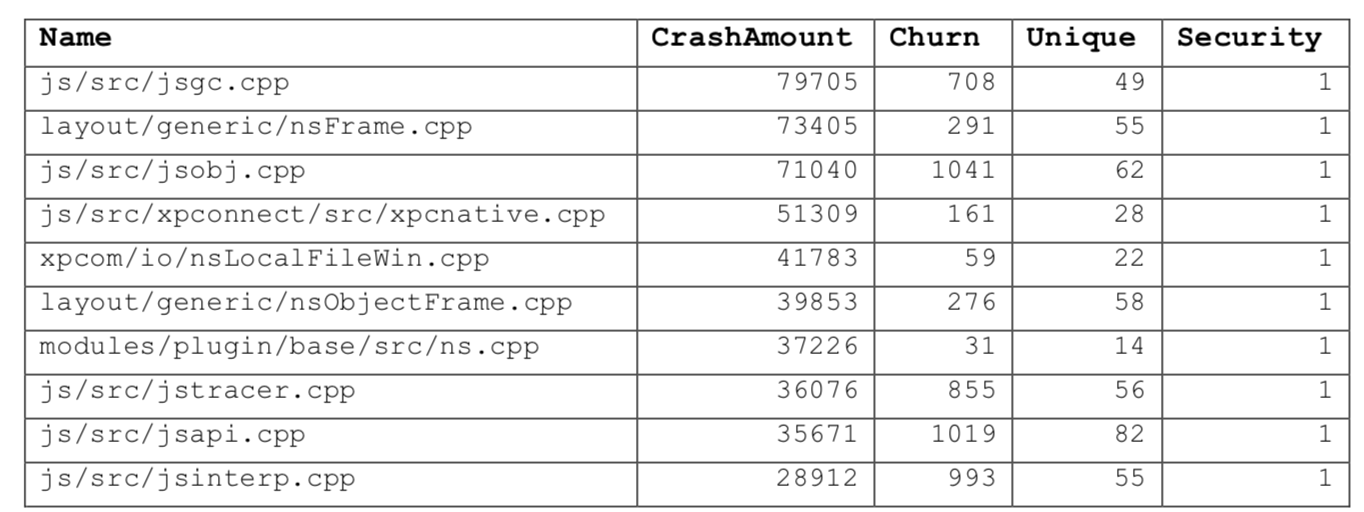


Figure -Partial dataset snippet

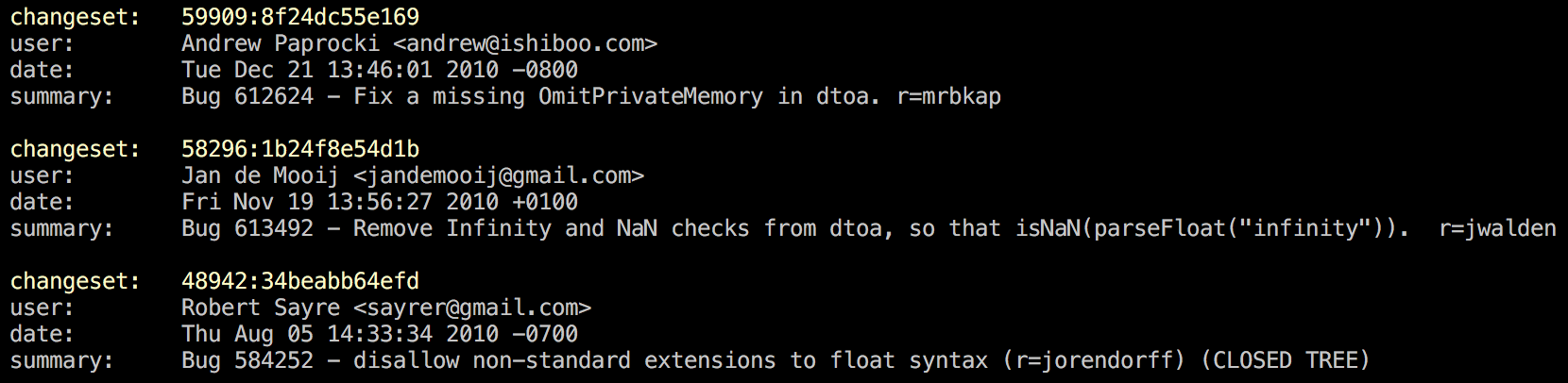


Figure -Data mining the full-dataset

* Mercurial repository for Firefox was mined for the full-dataset (all the files)
  + Collected samples that did not have security flaws and had not crashed from May 2010 to March 2012

$ hg log –d "may 2010 to mar 2012" <path\_to\_firefox> + <file\_path>

#### Vulnerability prediction modeling results

#### Run the model in MATLAB by running the main.m script from the prompt

#### Evaluation of results

Acronyms:

* True positives-tp
* True negatives-tn
* False positives-fp
* False negative-fn

Statistical measures used for evaluation

* Precision:
* Recall:
* True negative rate:
* Accuracy:

Recall is a more meaningful metric, since, we are looking for vulnerabilities in this application

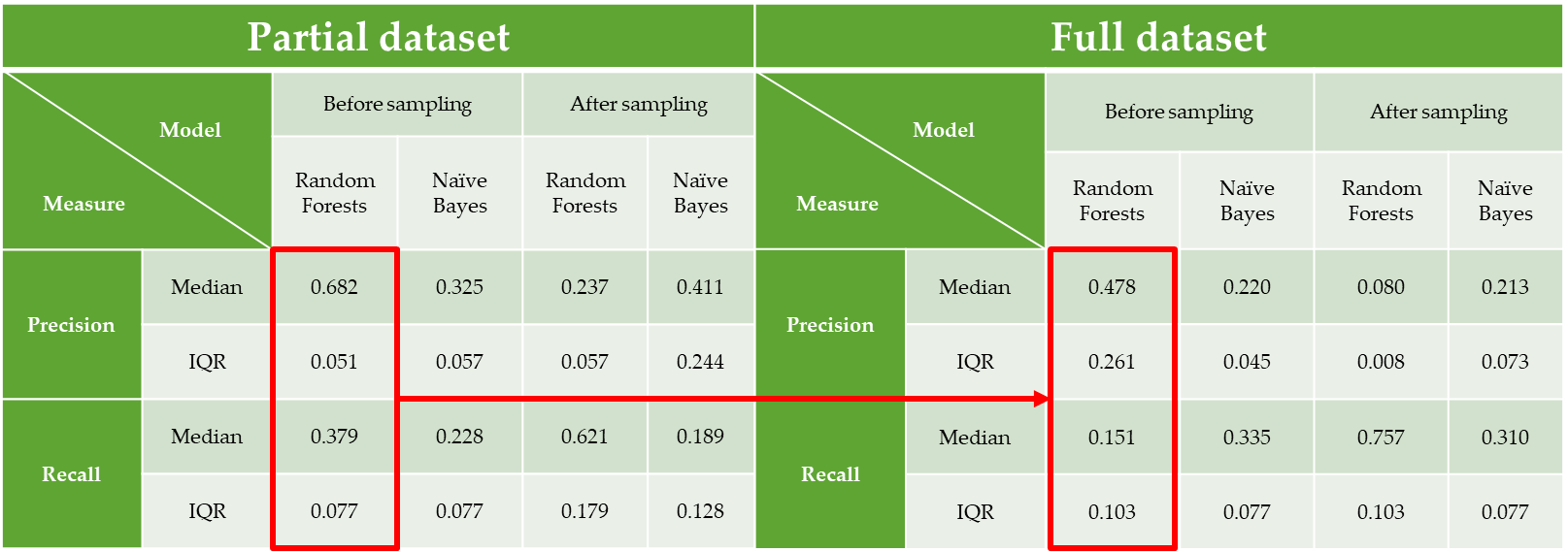


Figure -Random forest model results vs Naïve Bayes

Interestingly, the partial dataset provided better results

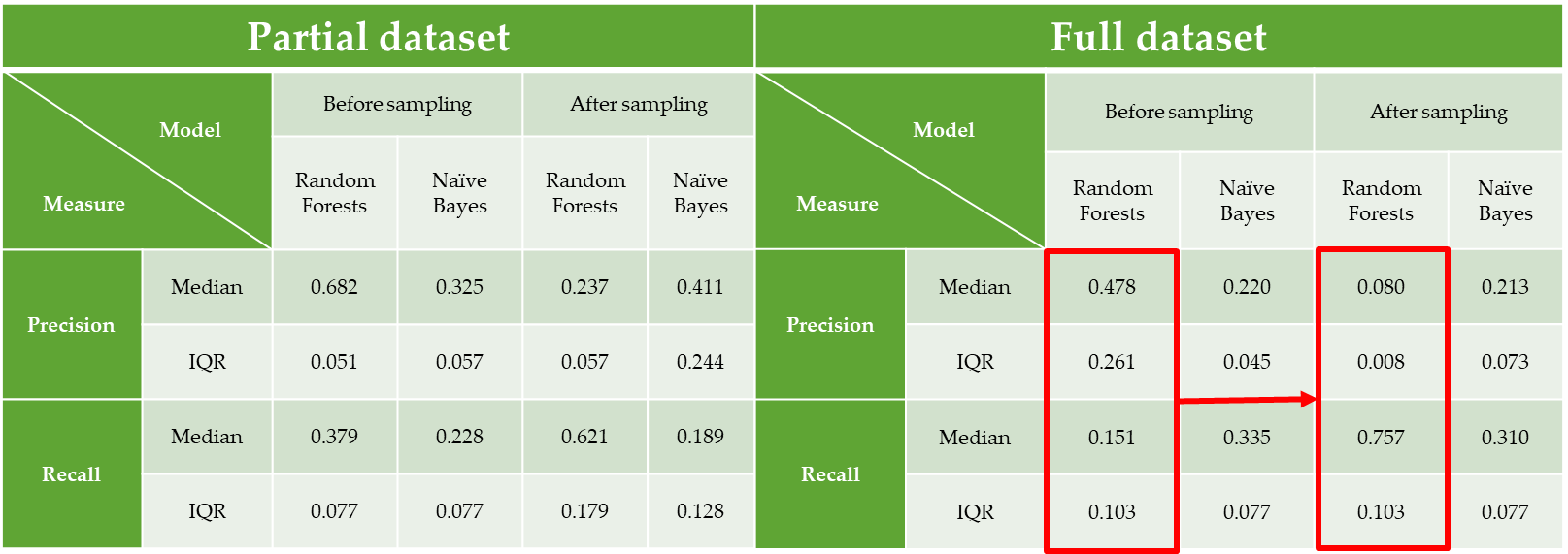


Figure - Random forest model results vs Naïve Bayes

Random forests performs better than Naïve Bayes

#### Using the results



Figure -Resulting decision tree

* Left number (red) is the number of files without security vulnerabilities
* Right number (blue) is the files with security vulnerabilities

Discuss the meaning of these results amongst your team?

Discuss how might you apply these concepts to a new application?

Spend the remainder of the class period analyzing the code

### Definitions

* Data mining-process of discovering useful patterns, automatically or semi-automatically, in large quantities of data
* Learning-process that seeks optimal changes to behavior towards performing better in the future (Witten, Frank, 2006)
* Machine learning-algorithms that analyze mined data and automatically identify patterns supporting improved decisions
* Code churn-a measure for the rate of change (edits) to a file by developers

### References

* [Christopher Theisen, Rahul Krishna, Lauri Williams, "Strengthening the Evidence that Attack Surfaces Can Be Approximated with Stack Traces," TR-2015-10, NC State University, Computer Science, November 3, 2015](ftp://ftp.ncsu.edu/pub/unity/lockers/ftp/csc_anon/tech/2015/TR-2015-10.pdf)
* [Christopher Theisen, Kim Herzig, Patrick Morrison, Brendan Murphy, Laurie Williams, "Approximating Attack Surfaces with Stack Traces," Proceedings of the 2015 IEEE International Conference on Software Engineering, Florence, Italy](https://dx.doi.org/10.1109/ICSE.2015.148)
* [Nguyen, Tran, Predicting vulnerable software components with dependency graphs, Proceedings of the 6th International Workshop on Security Measurements and Metrics, Bolzano, Italy, September, 2010](https://doi.org/10.1145/1853919.1853923)
* Witten, Frank, Data Mining, Practical Machine Learning Tools and Techniques, Elsevier, 2006