Categorizing Software Applications for Maintenance

Collin McMillan[1]
Mario Linares-Vásquez[2]
Denys Poshyvanyk[1]
Mark Grechanik[3]

[1]College of William & Mary
[2]Universidad Nacional de Colombia
[3]Accenture Technology Labs

Oceans of Code

- Programmers have created huge amounts of code

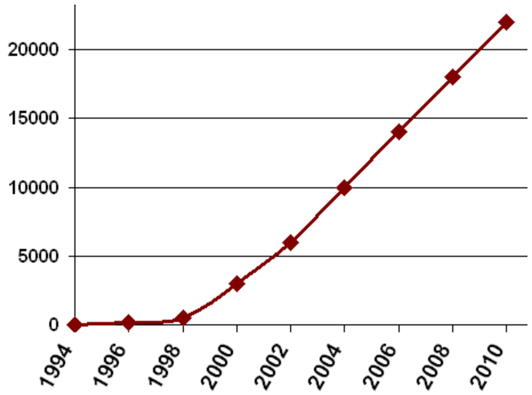
- How much code?
  - *U.S. Bureau of Labor*: **1.3m** programmers in USA
  - *Linux Journal Magazine* poll: **~150 KLOC** per programmer
  - **~195 billion LOC** written in USA alone

  (comparison: ~650 billion sentences ever published)

What happens to all that code?

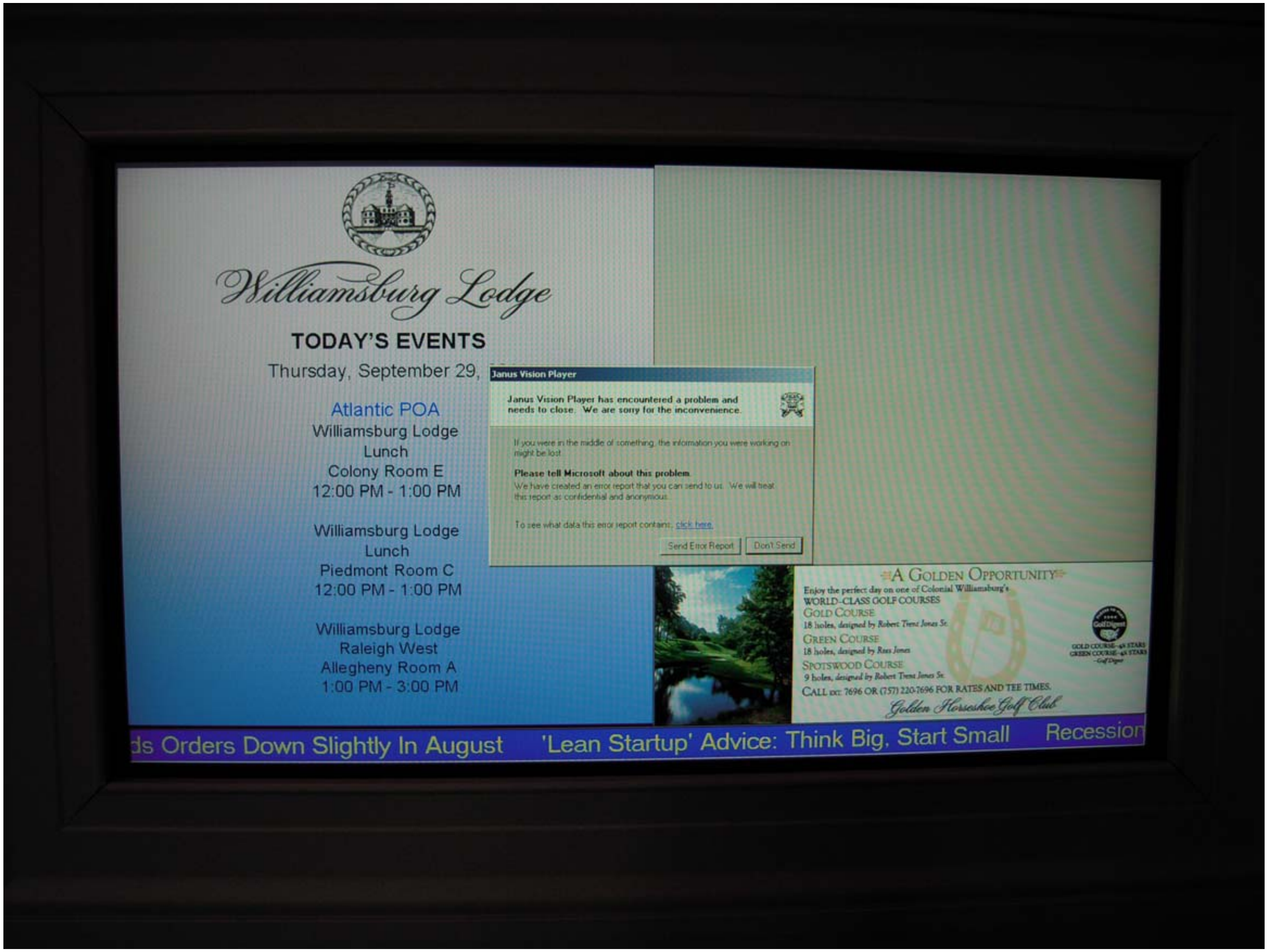
Oceans of Code

- Software Repositories are growing
  - SourceForge, 300k applications
  - FreeBSD Ports, 22k applications, **270 Million LOC**



- Corporate software development is also growing
  - Accenture, founded 1989, 250k employees
  - IBM, founded 1911, 425k employees

Oceans of (BUGGY) Code



Williamsburg Lodge

TODAY'S EVENTS

Thursday, September 29,

Atlantic POA
Williamsburg Lodge
Lunch
Colony Room E
12:00 PM - 1:00 PM

Williamsburg Lodge
Lunch
Piedmont Room C
12:00 PM - 1:00 PM

Williamsburg Lodge
Raleigh West
Allegheny Room A
1:00 PM - 3:00 PM

Janus Vision Player

Janus Vision Player has encountered a problem and needs to close. We are sorry for the inconvenience.

If you were in the middle of something, the information you were working on might be lost.

Please tell Microsoft about this problem.

We have created an error report that you can send to us. We will treat this report as confidential and anonymous.

To see what data this error report contains, click here.

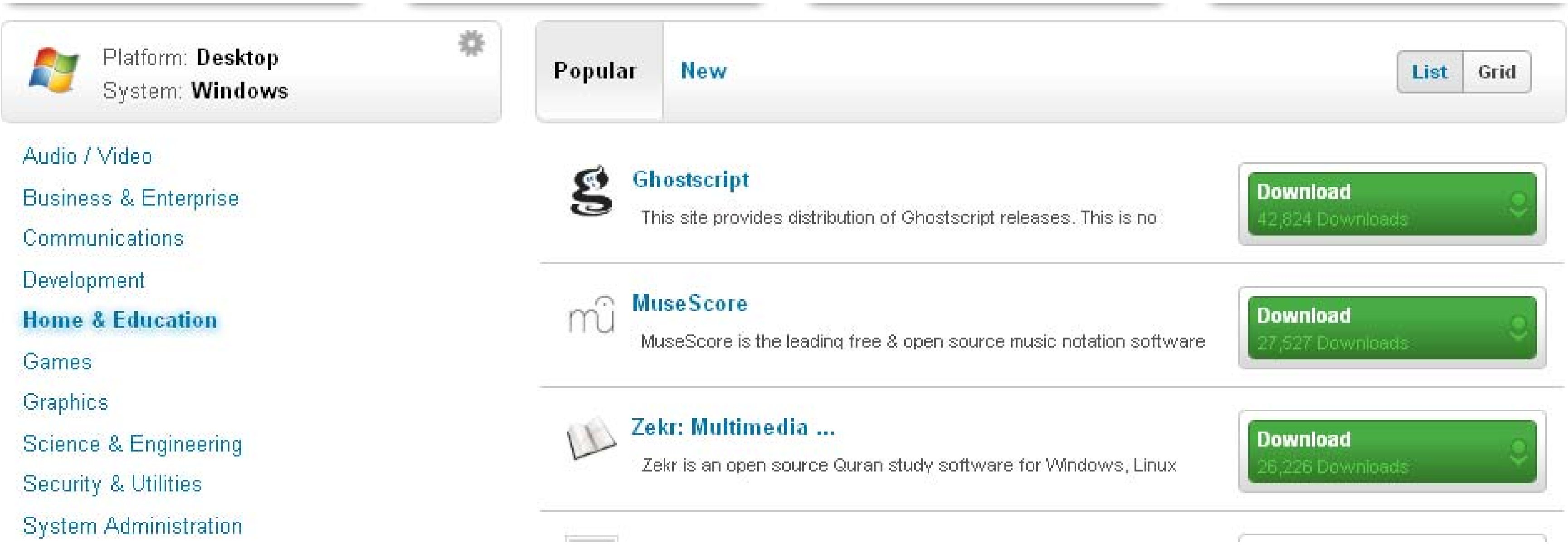
A GOLDEN OPPORTUNITY

Enjoy the perfect day on one of Colonial Williamsburg's
WORLD-CLASS GOLF COURSES
GOLD COURSE
18 holes, designed by Robert Trent Jones Sr.
GREEN COURSE
18 holes, designed by Rees Jones
SPOTSWOOD COURSE
9 holes, designed by Robert Trent Jones Sr.
CALL ext. 7696 OR (757) 220-7696 FOR RATES AND TEE TIMES.
Golden Horseshoe Golf Club

GOLD COURSE–4½ STARS
GREEN COURSE–4½ STARS
–Golf Digest

ds Orders Down Slightly In August 'Lean Startup' Advice: Think Big, Start Small Recession

Categorization is Useful



Platform: **Desktop**
System: **Windows**

Audio / Video

Business & Enterprise

Communications

Development

Home & Education

Games

Graphics

Science & Engineering

Security & Utilities

System Administration

Ghostscript
This site provides distribution of Ghostscript releases. This is no
Download
42,824 Downloads

MuseScore
MuseScore is the leading free & open source music notation software
Download
27,527 Downloads

Zekr: Multimedia ...
Zekr is an open source Quran study software for Windows, Linux
Download
26,226 Downloads

Categorization for Maintenance

- Software is more than Source Code
  - Binaries, Features, Bug Reports, etc.

- Domain analysis and Decision-Making
  - Are we maintaining unpopular features?
  - What differentiates our product from others?
  - Does similar software experience similar bugs?

How to Categorize?

- Manual Solutions
  - Self-reporting
  - Sorting / Cataloging
- Some problems
  - Legacy code
  - New categories
  - Number of applications labeled “other”
- An **automated solution** is desirable

The Categorization Game

- I will show you a fragment of code
- You have 15 seconds to categorize it

Text Editor

Web Browser

Music Player

```
import java.awt.event.*;
import javax.swing.*;
import javax.sound.midi.*;

/**
 * Illustrates general MIDI melody instruments and MIDI controllers.
 *
 * @version @(#)MidiSynth.java    1.15 99/12/03
 * @author Brian Lichtenwalter
 */
public class MidiSynth extends JPanel implements ControlContext {
    public void open() {
        try {
            if (synthesizer == null) {
                if ((synthesizer = MidiSystem.getSynthesizer()) == null) {
                    System.out.println("getSynthesizer() failed!");
                    return;
                }
            }
            synthesizer.open();
            sequencer = MidiSystem.getSequencer();
            sequence = new Sequence(Sequence.PPQ, 10);
        } catch (Exception ex) { ex.printStackTrace(); return; }

        Soundbank sb = synthesizer.getDefaultSoundbank();
   if (sb != null) {
            instruments =
   synthesizer.getDefaultSoundbank().getInstruments();
            synthesizer.loadInstrument(instruments[0]);
        }
        MidiChannel midiChannels[] = synthesizer.getChannels();
```

Done!

- Who thinks the code was from a ~~**text editor**~~?
  MIDI music player

- We did not read the code
- We guessed based on the **keyword clues**

```
import java.awt.event.*;
import javax.swing.*;
import javax.sound.midi.*;

/**
 * Illustrates general MIDI melody instruments and MIDI controllers.
 *
 * @version @(#)MidiSynth.java    1.15 99/12/03
 * @author Brian Lichtenwalter
 */
public class MidiSynth extends JPanel implements ControlContext {
    public void open() {
        try {
            if (synthesizer == null) {
                if ((synthesizer = MidiSystem.getSynthesizer()) == null) {
                    System.out.println("getSynthesizer() failed!");
                    return;
                }
            }
            synthesizer.open();
            sequencer = MidiSystem.getSequencer();
            sequence = new Sequence(Sequence.PPQ, 10);
        } catch (Exception ex) { ex.printStackTrace(); return; }

        Soundbank sb = synthesizer.getDefaultSoundbank();
   if (sb != null) {
            instruments =
   synthesizer.getDefaultSoundbank().getInstruments();
            synthesizer.loadInstrument(instruments[0]);
        }
        MidiChannel midiChannels[] = synthesizer.getChannels();
```

State-of-the-Art

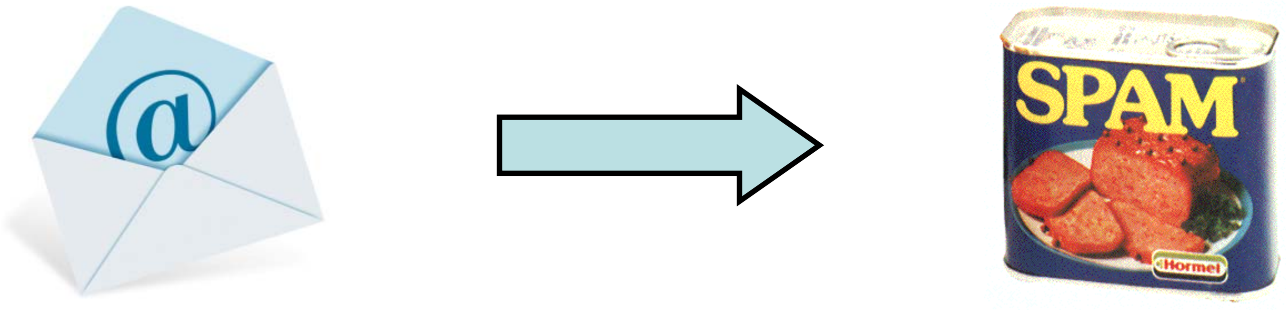
- Categorize based purely on the keywords from source code

- Keywords as attributes for machine learning and classification

Relies on Source Code as Text

Machine Learning Approaches

Binary



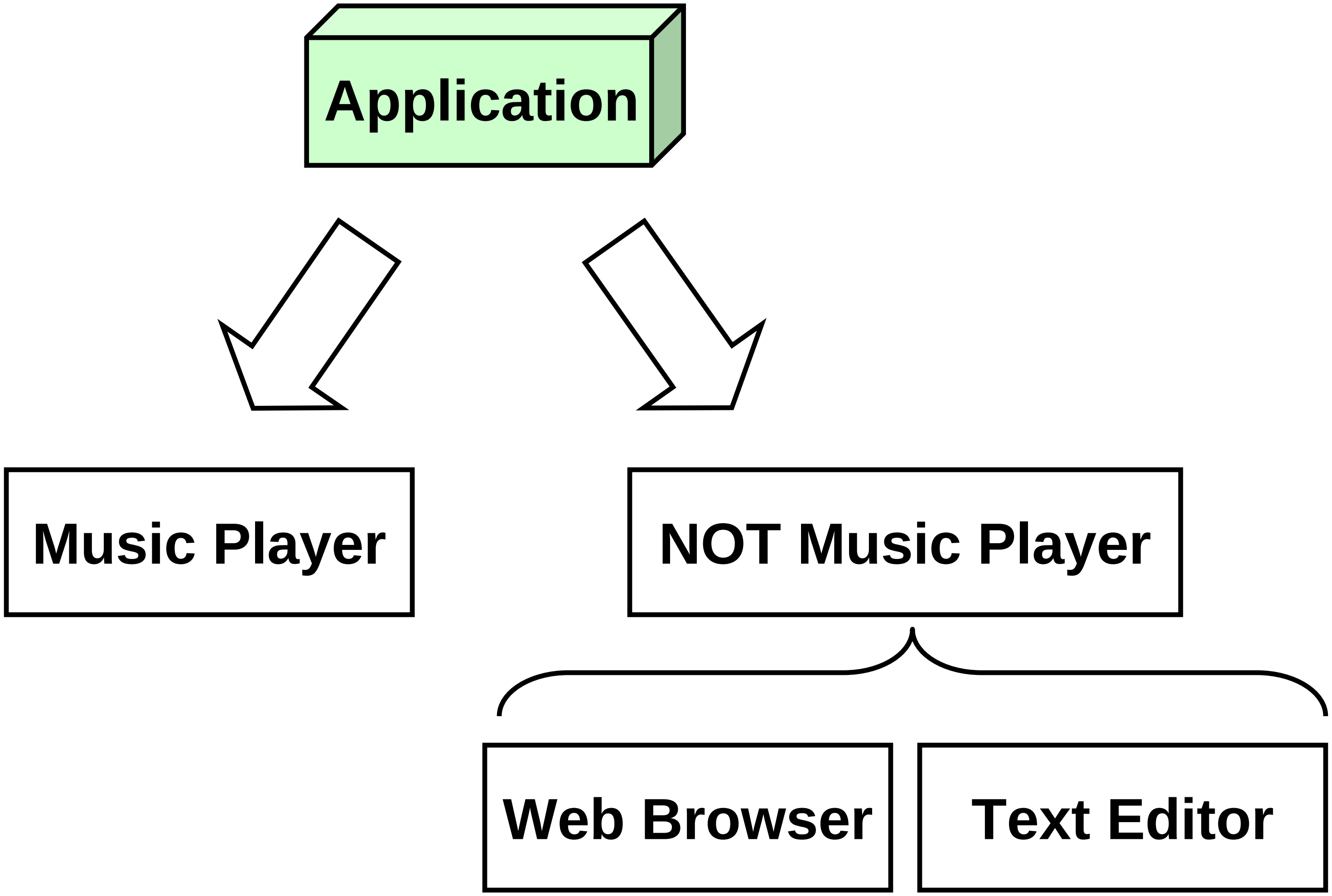
Multiclass

Winter is here.

"Winter is here."

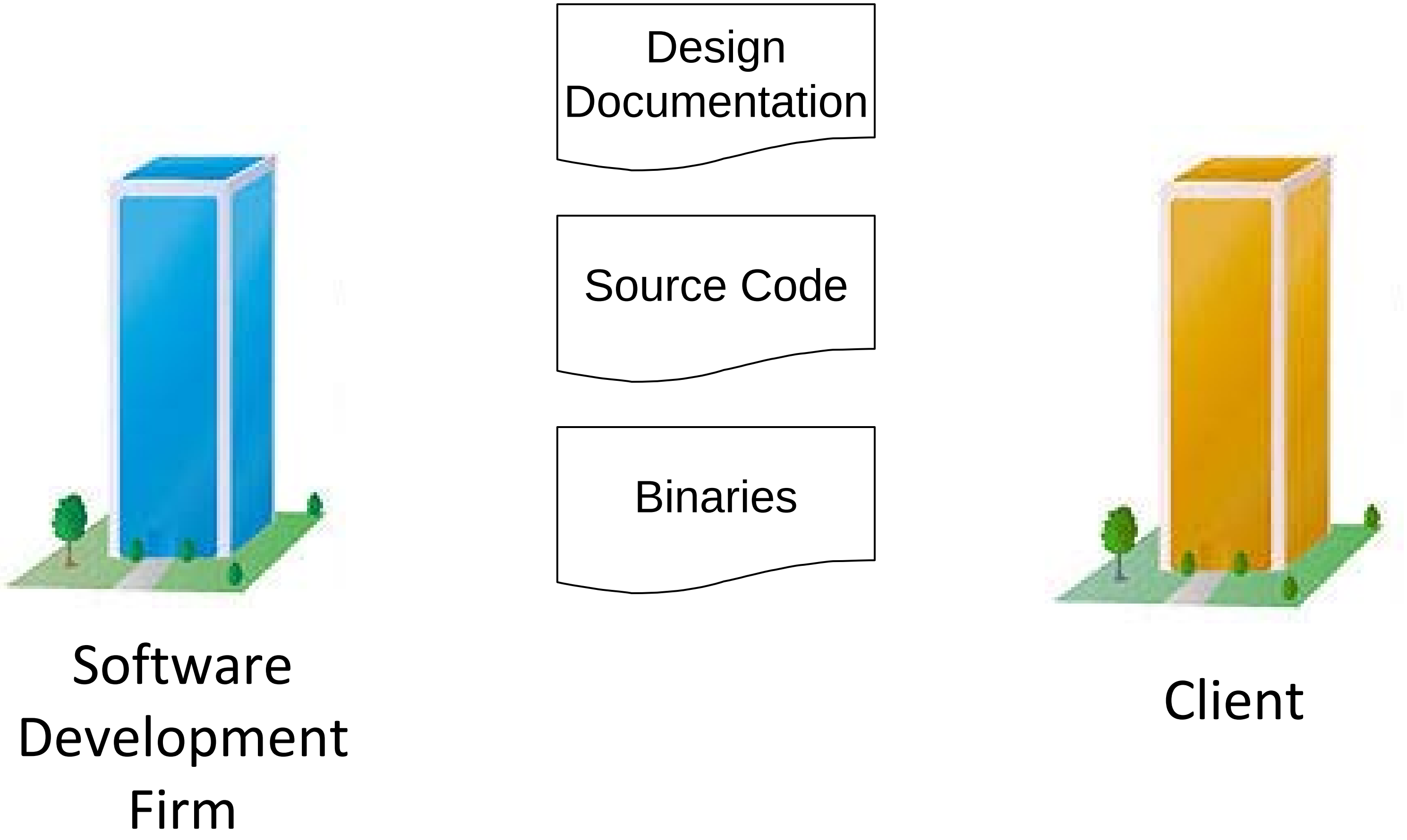
[1] Guillaume Obozinski, "Multi-Class and Structured Classification"

Multiclass composed of **binary** classifiers



Problem:
Source Code is not always available

- Question of Ownership



Design Documentation

Source Code

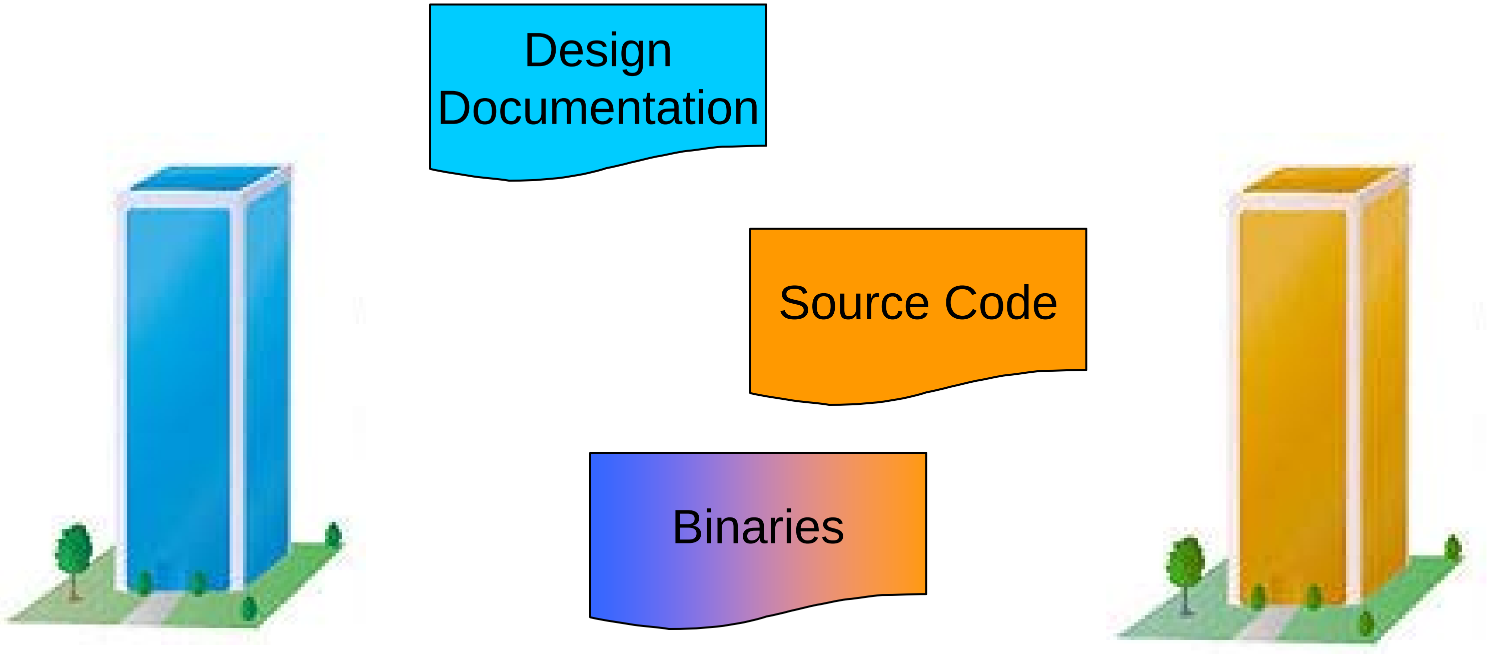
Binaries

Software Development Firm

Client

Problem:
Source Code is not always available

- **Client** owns the Source Code



Design Documentation

Source Code

Binaries

Software Development Firm

Client

Our Solution

- Use only **API calls** from binaries as attributes
- API calls can be extracted from binaries as dependencies
- API calls define critical functionality

APIs Appear Everywhere

Example API package:

com.sun.java_cup.internal

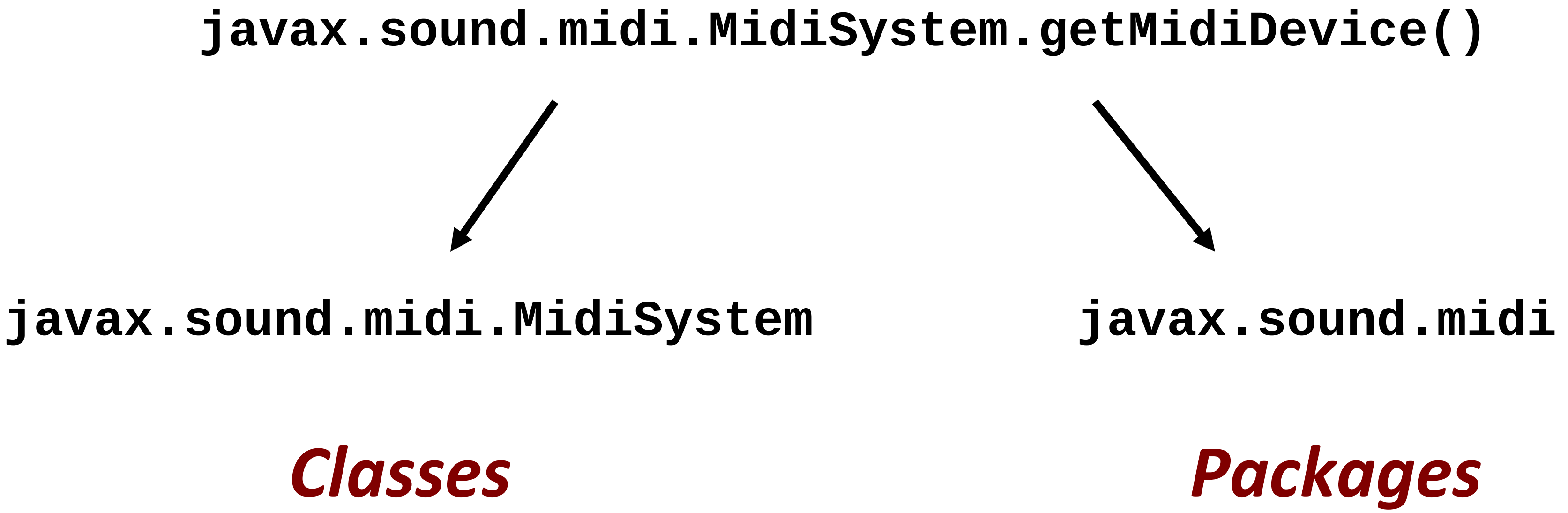
Used over **3000** times in **600 of 8000** different applications from Sourceforge.

```java
import java.awt.event.*;
import javax.swing.*;
import javax.sound.midi.*;

/**
 * Illustrates general MIDI melody instruments and MIDI controllers.
 *
 * @version @(#)MidiSynth.java    1.15 99/12/03
 * @author Brian Lichtenwalter
 */
public class MidiSynth extends JPanel implements ControlContext {
    public void open() {
        try {
            if (synthesizer == null) {
                if ((synthesizer = MidiSystem.getSynthesizer()) == null) {
                    System.out.println("getSynthesizer() failed!");
                    return;
                }
            }
            synthesizer.open();
            sequencer = MidiSystem.getSequencer();
            sequence = new Sequence(Sequence.PPQ, 10);
        } catch (Exception ex) { ex.printStackTrace(); return; }

        Soundbank sb = synthesizer.getDefaultSoundbank();
   if (sb != null) {
            instruments =
   synthesizer.getDefaultSoundbank().getInstruments();
            synthesizer.loadInstrument(instruments[0]);
        }
        MidiChannel midiChannels[] = synthesizer.getChannels();
```

Two API-based Attributes



`javax.sound.midi.MidiSystem.getMidiDevice()`

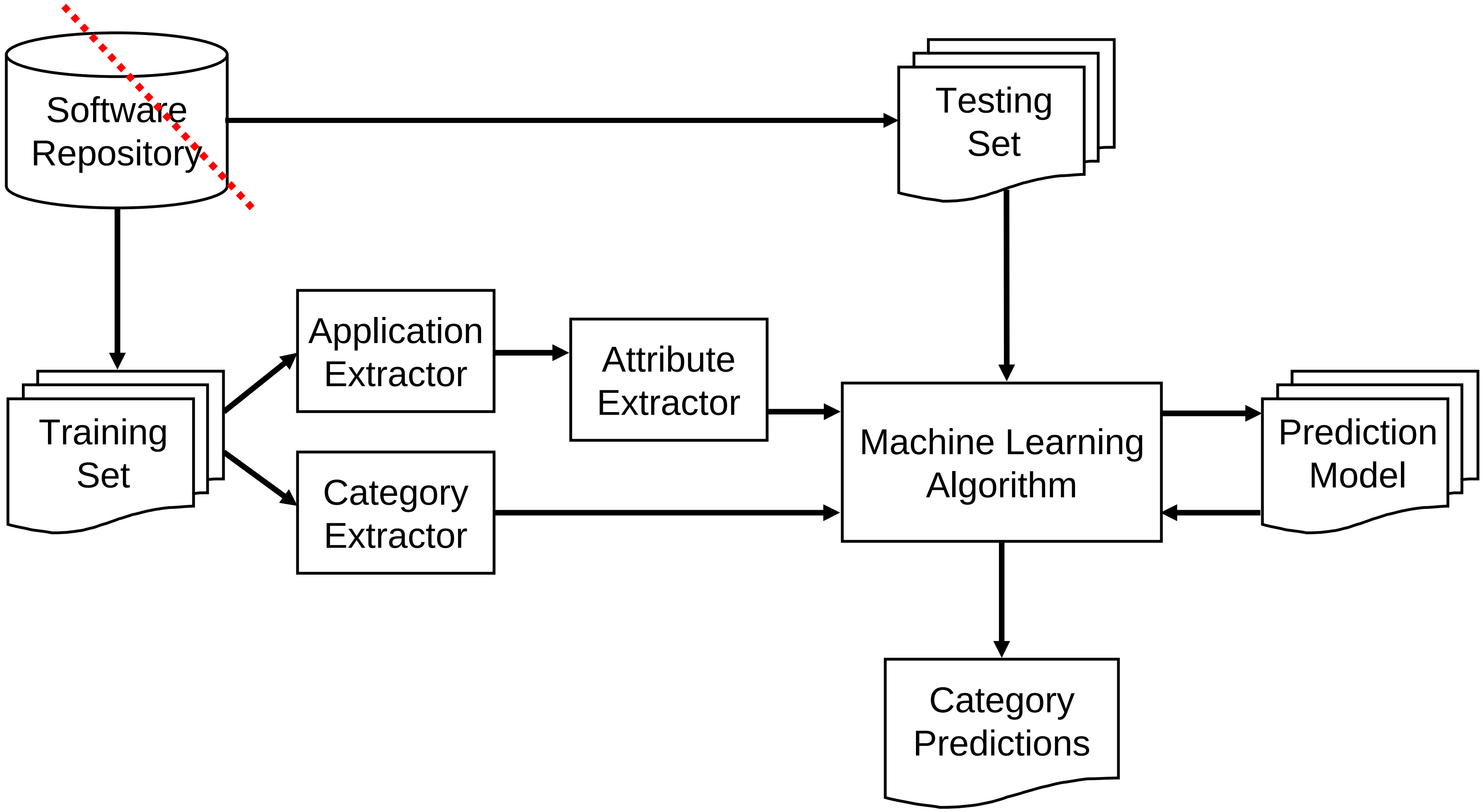
`javax.sound.midi.MidiSystem`

Classes

`javax.sound.midi`

Packages

Cross Validation Experiment



Key Design Questions

- Which **Machine Learning Algorithm** to use?
  - Support Vector Machines (SVM)
  - Decision Trees
  - Naïve Bayesian

- Which **Attributes** to select?
  - Terms
  - API calls

Different Configurations

	State-of-the-Art	Our Work
Attributes		
Terms	✓	✓
API Classes		✓
API Packages		✓
ML Algorithms		
SVM	✓	✓
Decision Trees		✓
Naïve Bayes		✓
Number of Apps	1683	4031

Software Repositories

SourceForge *(3,286 apps)*

Category	Count	Category	Count
Bio-Informatics	323	Indexing	329
Chat	504	Internet	1061
Communication	699	Interpreters	303
Compilers	309	Mathmatics	373
Database	988	Networking	360
Education	775	Office	522
Email	366	Scientific	326
Frameworks	1115	Security	349
Front-Ends	584	Testing	907
Games	607	Visualization	456
Graphics	313	Web	534

ShareJar *(745 apps)*

Category	Count
Chat & SMS	320
Dictionaries	30
Education	90
Free Time	120
Internet	180
Localization	20
Messengers	50
Music	50
Science	20
Utilties	190
Emulators	30
Programming	10
Sports	40

Research Questions

RQ_1 Which **machine learning algorithm** is most effective for software categorization?

RQ_2 Which level of API granularity, **classes or packages**, is more effective for categorization?

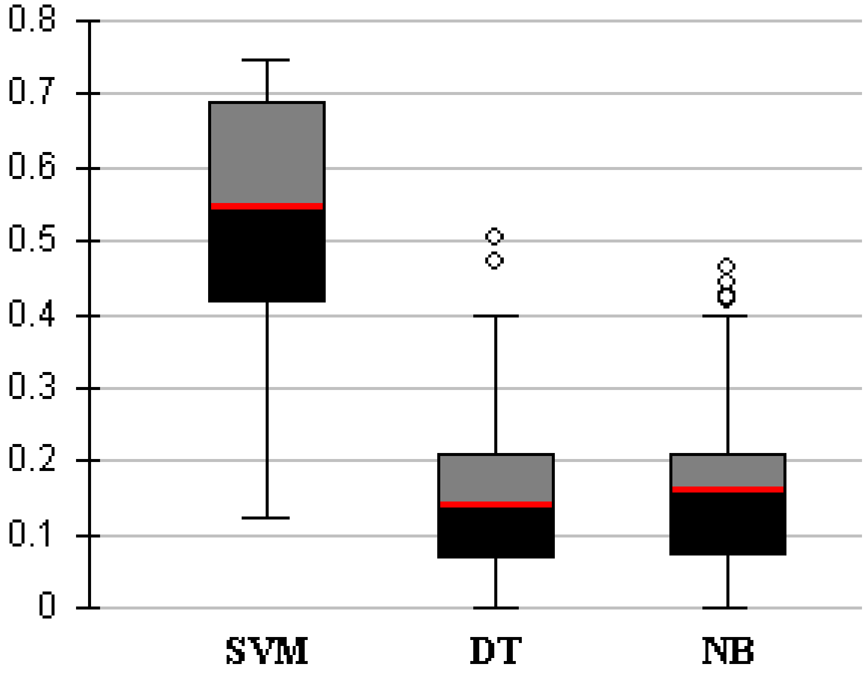
RQ_3 Are the API classes or API packages as effective as **words from source code** for categorization?

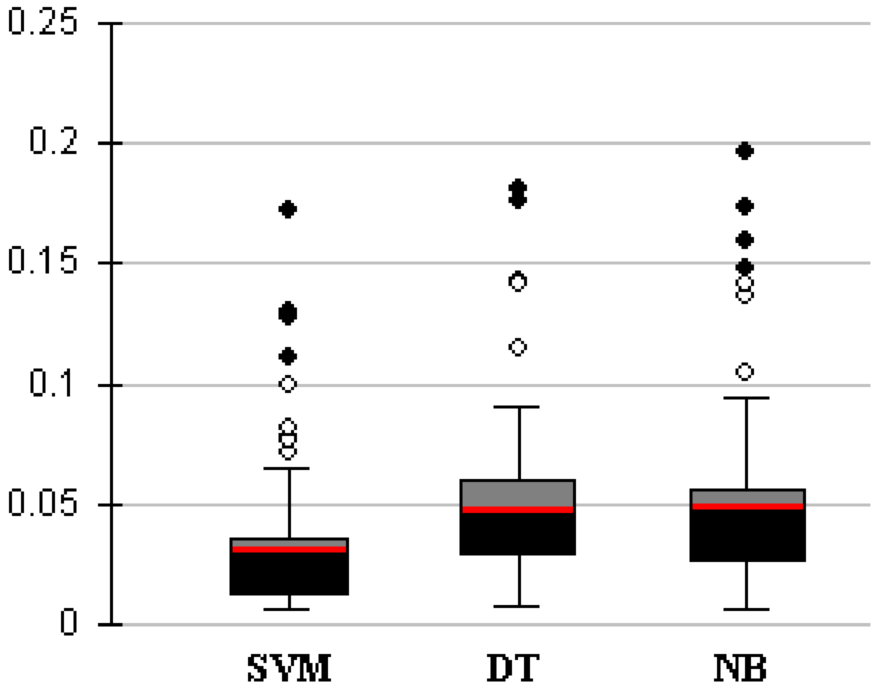
Evaluation Metrics

- True Positive Rate
  - Proportion of **correct** links that were found
  - Analogous to Recall

- False Positive Rate
  - Proportion of **incorrect** links that were found
  - Analogous to Fall-Out

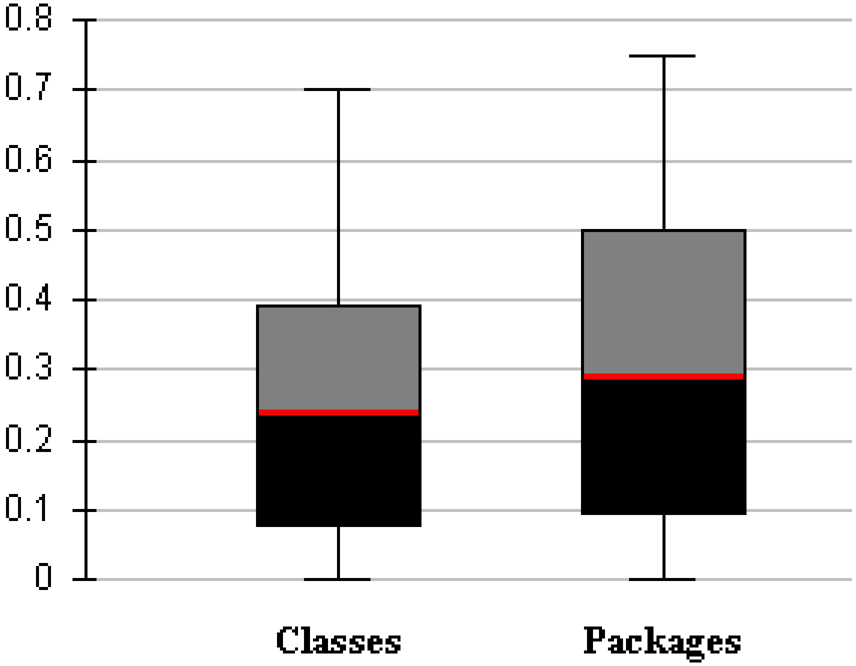
RQ$_1$: Machine Learning Algorithms


True Positive Rate

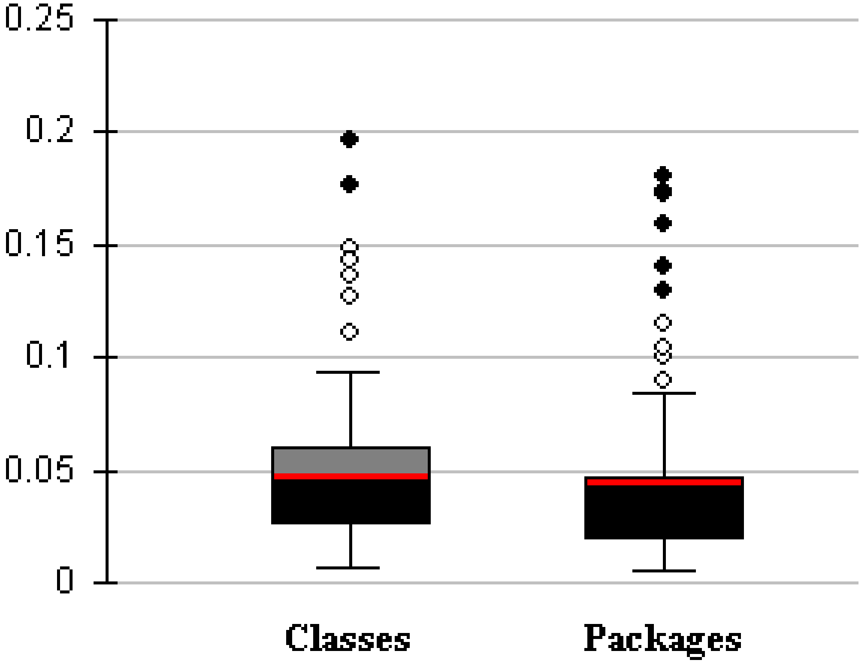

False Positive Rate

SVM outperforms DT and NB.

RQ$_2$: API Classes vs. Packages



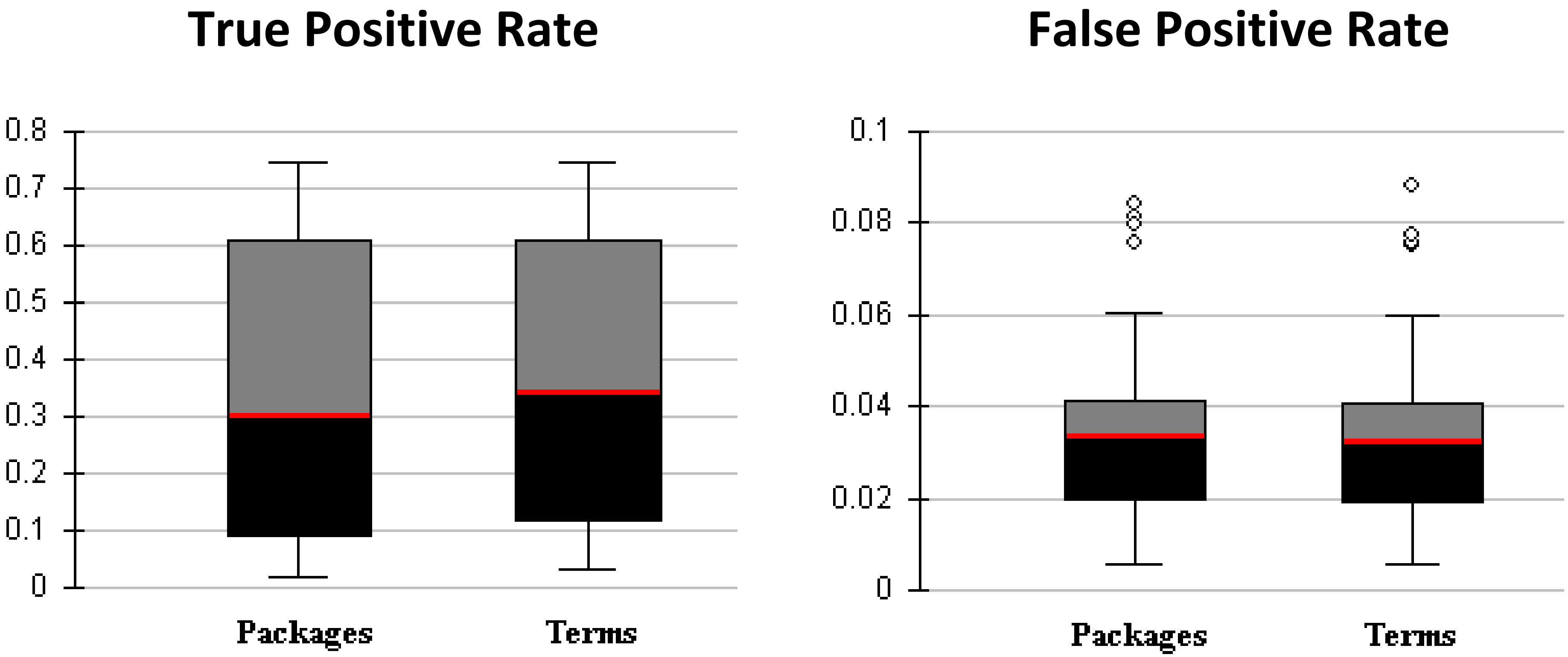
True Positive Rate



False Positive Rate

API packages outperforms API classes.

RQ$_3$: API Packages vs. All Terms



True Positive Rate

False Positive Rate

API packages performs nearly as well as Terms.

Statistical Tests

- Friedman Test with Nemenyi's Post-Hoc Procedure

  H_0 There is no statistically-significant difference between the **TPR** of **SVM** and **DT**.

  H_1 There is no statistically-significant difference between the **TPR** of **SVM** and **NB**.

  H_2 There is no statistically-significant difference between the **FPR** of **SVM** and **DT**.

  H_3 There is no statistically-significant difference between the **FPR** of **SVM** and **NB**.

H	$q_{critical}$	$q_{observed}$	Decision
H_0	26.59	140.5	Reject
H_1	26.59	132.5	Reject
H_2	26.59	141.5	Reject
H_3	26.59	118.0	Reject

Anecdotal Example

Top **term**, **API class**, and **API package** in *Email* category of Sourceforge.

Type of Feature	Feature	Apps in Category with Feature	Total Apps with Feature
Term	replyto	8	33
Package	sun.net.www	8	300
Class	com.sun.jlex.internal.CEmit	8	300

Conclusions

- We present an approach for software categorization
- Our approach categorizes using API calls
- We replicated a state-of-the-art study and showed:
  - **SVM** is the best of three selected ML algorithms
  - **API packages** outperform API classes as attributes
  - API packages perform **as well as terms** for categorization
- Our approach **does not rely on source code**

http://www.cs.wm.edu/semeru/catml/