Tests as Classifiers

Moshe Zadka – https://cobordism.com

Acknowledgement of Country

Belmont (in San Francisco Bay Area Peninsula)
Ancestral homeland of the Ramaytush Ohlone

What is a classifier?

Input: Something

What is a classifier?

Input: Something
Output: True/False

What is a classifier?

Input: Something
Output: True/False
Belongs to set / Does not belong to set

Classifier example: chihuahua or muffin

Input: Image of chihuahua or muffin

Classifier example: chihuahua or muffin

Input: Image of chihuahua or muffin
Output: Is it a dog?

Classifier failure: False alarm

Classifier says ”yes”, should say ”no”
Example: Picture of muffin, classifier says ”dog”

Classifier failure: Missing alarm

Classifier says ”no”, should say ”yes”
Example: Picture of dog, classifier says ”muffin”

Test (suites) as classifiers

Input: Code change
Output: Is the code buggy?

Test (suites) as classifiers

Tests suite failure: ”Code buggy”
Test suite success: ”Code not buggy”

Simple classifiers

Always alarm: "Yes" regardless of input
Never alarm: "No" regardless of input

Always alarm: a test

```python
def test_always_alarm():
    assert 1 == 0
```

Never alarm: a test suite

```
# Empty file
```

Why not simple classifiers?

Writing tests is hard work!

Why not simple classifiers?

Writing tests is hard work!
Can we quantify the value?

Precision

Rewards not alarming:

```
precision = (
    true_alarms /
    (true_alarms + false_alarms)
)
```

Recall

Rewards alarming:

```
recall = (
    true_alarms /
    (true_alarms + missing_alarms)
)
```

Balancing Precision and Recall: F score

Harmonic mean:

```
precision_inv = 1 / precision
recall_inv = 1 / recall
mean_inv = (precision_inv + recall_inv) / 2
f_score = 1 / mean_inv
```

Balancing Precision and Recall: F beta score

What if it's not equally important?

Balancing Precision and Recall: F beta score

What if it's not equally important?

```
precision_inv = 1 / precision
recall_inv = 1 / recall
mean_inv = (
    (beta ** 2 * precision_inv + recall_inv)
    /
    (beta ** 2 + 1)
f_beta_score = 1 / mean_inv
```

Balancing Precision and Recall: Who is beta?

F score is F beta score when beta is 1

Balancing Precision and Recall: Who is beta?

F score is F beta score when beta is 1

The "beta" parameter encodes utility: which error hurts harder?

Balancing Precision and Recall: Who is beta?

F score is F beta score when beta is 1
The "beta" parameter encodes utility: which error hurts harder?
A business decision!

Beta: a Meaning

Beta is 2: Missing alarms are twice as painful as false alarms

Beta: a Meaning

Beta is 2: Missing alarms are twice as painful as false alarms
Beta is 0.5: Missing alarms are half as painful as false alarms

Tests and the F score

What makes a test bring down the F score?

False Alarm: Flakey test

Fails ”randomly”

False Alarm: Implementation test

Testing implementation details:

False Alarm: Implementation test

Testing implementation details:

```
def test_implementation():
    with mock.patch(
        ”subprocess.check_output”
    ) as check_output:
        run_code()
    assert some_stuff
```

What happens when it uses ”subprocess.run”?

Missing Alarm: Non-covered code

What is not run does not affect result of tests

Missing Alarm: Loose assertions

Asserting something is greater than 5,

Missing Alarm: Loose assertions

Asserting something is greater than 5,
not equal to 6

Estimating F score: Flakey tests

Run known-good main branch

Estimating F score: Flakey tests

Run known-good main branch
check for failures

Estimating F score: Implementation tests

Rough measure: changes to tests

Estimating F score: Implementation tests

Rough measure: changes to tests
Heuristics to compensate for legitimate changes

Estimating F score: Mutation testing

Percentage of surviving mutants: missing alarms

Estimating F score: Check reported bugs

Bugs with added tests

Estimating F score: Check reported bugs

Bugs with added tests
but not features

Estimating F score: Check reported bugs

Bugs with added tests
but not features
Heuristics

Estimating F score: True alarms

Weighing by PR/branch

Estimating F score: True alarms

Weighing by PR/branch
Try and get data from dev machines too!

Estimating F score: Lagging indicator

Merged PRs / main develoment

F score cautions

Check heuristics

F score cautions

Check heuristics

Goodhart's law

F score usage

Revealed preferences: beta

F score usage

Revealed preferences: beta
Calibrate time investment in improving tests

Improving tests

F score: lagging guide

Improve: Flakey tests

Better isolation

Improve: Implementation tests

Clear contracts

Improve: Untested code

Coverage

Improve: Under-tested code

Mutation testing

Summary: Goal

Prevent bugs

Summary: Goal

Prevent bugs
minimal cost

Summary: Decide

Cost of bug

Summary: Measure

Expectations vs. Reality

Summary: Improve

Align reality