



Down The Dependency Rabbit Hole

Machine Learning as a first line of defense in Intel's Dependency Review Process

```
$ env
```

```
NAME=John Andersen
```

```
HOME=Portland OR
```

```
USER=pdxjohnny
```

```
WORK=Intel
```

```
ROLE=Open Source Security Software Engineer
```

```
EMAIL=john.s.andersen@intel.com
```

```
INTERESTS=Embedded, Linux, Containers, Concurrency, Web Apps, Python
```

```
THESIS=Machine Learning
```

Dependency Evaluation Review Form

SECTION	GRADE	GRADING GUIDELINES
First look		<p>A - Mentions security audit or other proactive security activity.</p> <p>B - No major warning signs, and code is used professionally.</p> <p>C - No major warning signs, but not widely used or not well-supported.</p> <p>D - Code has minor warning signs that need to be investigated in more detail.</p> <p>F - Code has known issues, major warning signs, or is abandoned</p>
Contributors and activity		<p>A - At least five significant, active contributors.</p> <p>B - More than two significant, active contributors.</p> <p>C - Only one major contributor who is active.</p> <p>D - Project has been inactive for nine months or less.</p> <p>F - Project has been inactive for more than one year.</p>
Security issues		<p>A - Project has had previous security issues and handled them quickly and well. Bonus if they also mention doing proactive security such as fuzz testing, static analysis, or security audits.</p> <p>B - Project has a plan for handling security issues but hasn't had to use it much yet.</p> <p>C - Project does not have a plan for security issues but at least has an active bug tracker and issues get resolved.</p> <p>D - Project does not seem to resolve many open bugs.</p> <p>F - Project has open security issues that are not in the process of being resolved.</p>
Test suite		<p>A - Project has test suite with good coverage of positive and negative test cases set up as part of continuous integration, and test results are published for each build.</p> <p>B - Project has test suite with good coverage but no continuous integration.</p> <p>C - Test suite mostly covers positive test cases; very few or no error cases.</p> <p>D - Test suite has very low coverage or is only a few examples.</p> <p>F - No test suite.</p>

Automation Attempt One

- Initial dataset is made up of
 - URL of source repo
 - Security team's classification (Good / Bad)
 - Review form data
- Plan
 - Train model on dataset
 - Assess accuracy
 - Given URL, collect data to answer form questions
 - Predict classification by feeding collected data to model

Reviewers Rarely Fill Out Evaluation Form

SECTION	GRADE	GRADING GUIDELINES
First look		<ul style="list-style-type: none">A - Mentions security audit or other proactive security activity.B - No major warning signs, and code is used professionally.C - No major warning signs, but not widely used or not well-supported.D - Code has minor warning signs that need to be investigated in more detail.F - Code has known issues, major warning signs, or is abandoned
Contributors and activity		<ul style="list-style-type: none">A - At least five significant, active contributors.B - More than two significant, active contributors.C - Only one major contributor who is active.D - Project has been inactive for nine months or less.F - Project has been inactive for more than one year.
Security issues		<ul style="list-style-type: none">A - Project has had previous security issues and handled them quickly and well. Bonus if they also mention doing proactive security such as fuzz testing, static analysis, or security audits.B - Project has a plan for handling security issues but has not used it much yet.C - Project does not have a plan for security issues but at least has an active bug tracker and issues get resolved.D - Project does not seem to resolve many open bugs.F - Project has open security issues that are not in the process of being resolved.
Test suite		<ul style="list-style-type: none">A - Project has test suite with good coverage of positive and negative test cases set up as part of continuous integration, and results are published for each build.B - Project has test suite with good coverage but no continuous integration.C - Test suite mostly covers positive test cases; very few or no error cases.D - Test suite has very low coverage or is only a few examples.F - No test suite.



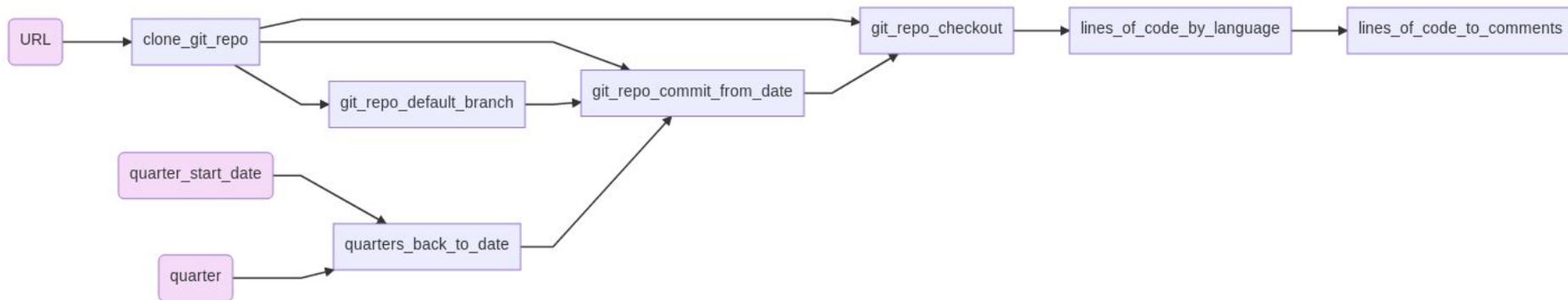
Automation Attempt One

- Initial dataset is made up of
 - URL of source repo
 - Security team's classification (Good / Bad)
 - Forms mostly not filled out 🙄
- Plan
 - Train model on dataset
 - Assess accuracy
 - Given URL, collect data to answer form questions
 - Predict classification by feeding collected data to model
- ~60% Accuracy 🚫

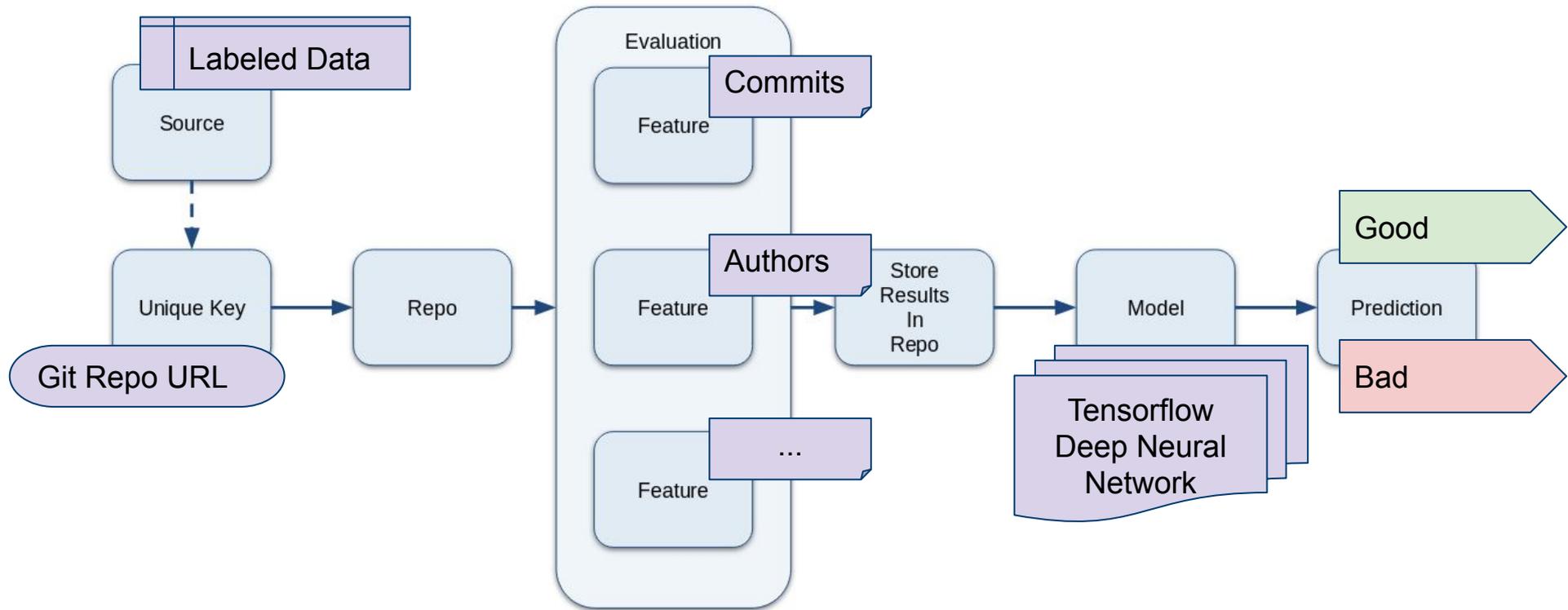
Automation Attempt Two

- Initial dataset is made up of
 - URL of source repo
 - Security team's classification (Good / Bad)
- Plan
 - Given URL, collect data
 - Train model on dataset
 - Assess accuracy
 - Predict classification by feeding collected data to model

Quarterly Ratio of Lines of Comments to Code



Prediction Data Flow



Request Classification Estimation

Project main page URL (Please provide a Public URL) *

`http://www.openssl.org/`

URL to the upstream source code development repository * Please use a valid upstream VCS repository, no "tarball", "rpm", "jar", etc files

`http://github.com/openssl/openssl`

Machine Learning prediction presented for estimation purposes only.

Machine Learning prediction: Error requesting evaluation

If you are sure this source URL is valid click [here](#).

Submit

Data Flow Facilitator for Machine Learning

Machine Learning made easy



Abstractions DFFML Provides

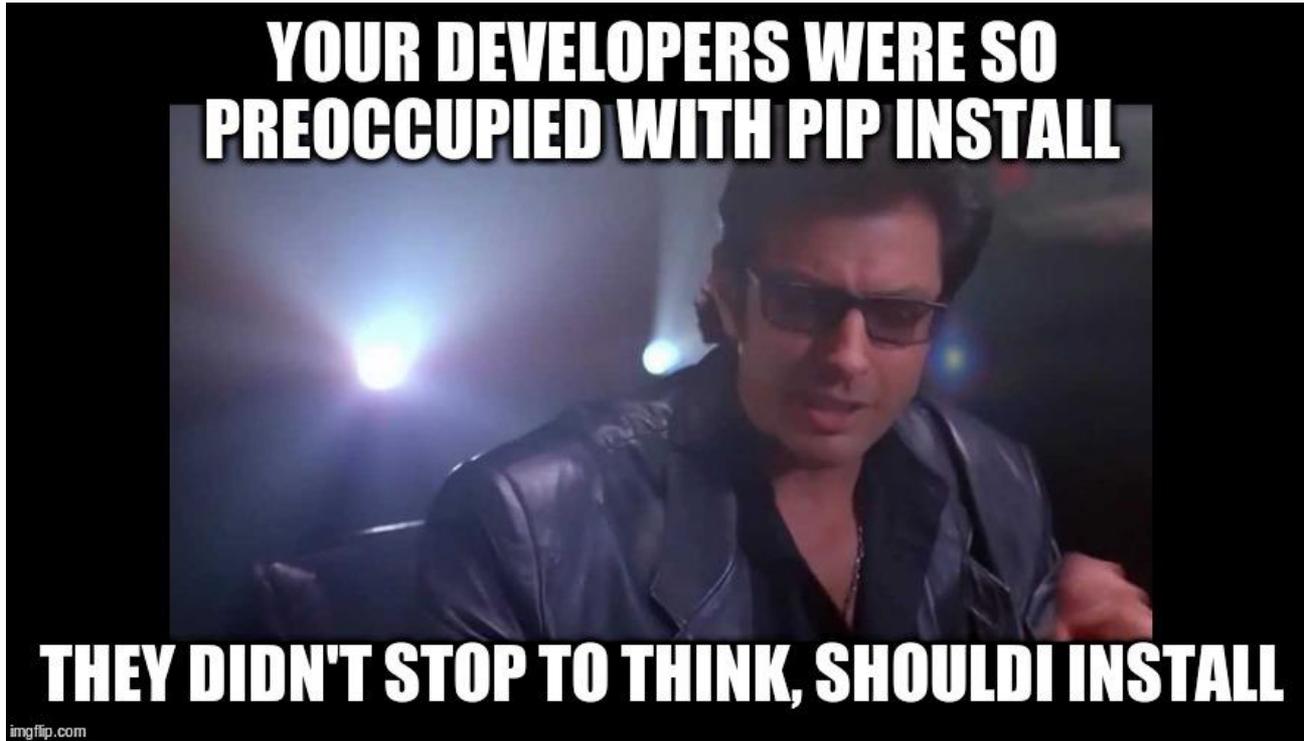
- Data Flow
 - Dataset generation
 - Concurrency without dealing with locking
- Sources
 - CSV
 - JSON
 - MySQL
- Models
 - Tensorflow
 - SciKit

Consistent API

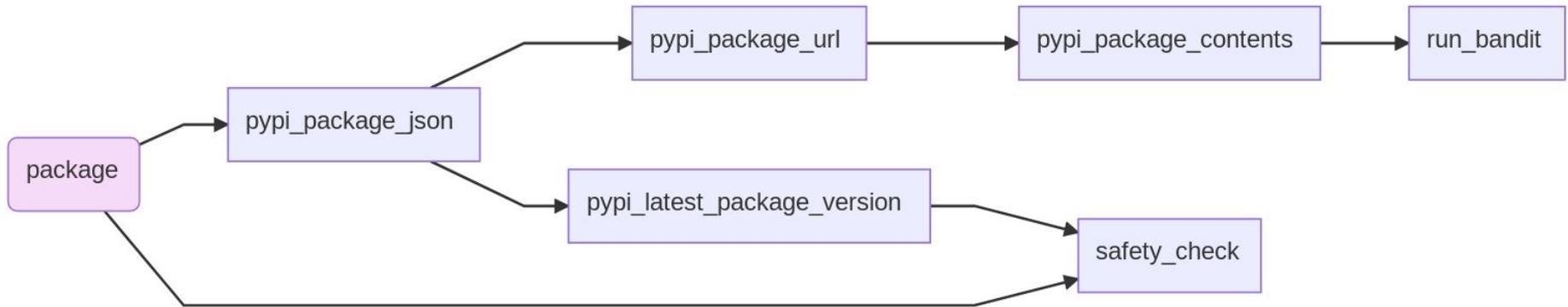
- Python Library
- Command Line Interface
- HTTP API
 - JavaScript Client

```
Created api ▶ Object { endpoint: "http://localhost:8080" }
Created training_source ▶ Object { plugin_type: "source", context_cls: DFFMLHTTPA
The character encoding of the HTML document was not declared. The document will l
declared in the document or in the transfer protocol.
Uploaded my_training_dataset.csv
Configured training_source ▶ Object { plugin_type: "source", context_cls: DFFMLHT
Created training_sctx ▶ Object { api: {...}, parent: {...}, label: "my_training_datas
Created test_source ▶ Object { plugin_type: "source", context_cls: DFFMLHTTAPISc
Uploaded my_test_dataset.csv
Configured test_source ▶ Object { plugin_type: "source", context_cls: DFFMLHTTPAF
Created test_sctx ▶ Object { api: {...}, parent: {...}, label: "my_test_dataset_conte
Training repos ▶ Object(4) [ {...}, {...}, {...}, {...} ]
Array of training repos ▶ Array(4) [ {...}, {...}, {...}, {...} ]
Created model ▶ Object { plugin_type: "model", context_cls: DFFMLHTTAPIModelCont
Configured model ▶ Object { plugin_type: "model", context_cls: DFFMLHTTAPIModelC
Created model context ▶ Object { api: {...}, parent: {...}, label: "mymodel_context"
Trained model context ▶ Object { api: {...}, parent: {...}, label: "mymodel_context"
Model context accuracy undefined
Model context predict ▶ Object { mish_the_smish: {...} }
Success!
```

Should I Be Installing This?



What is a Data Flow?



Deploy Anywhere - Command Line



Deploy Anywhere - HTTP

```
deploy $ tree
├── df
│   └── shouldi.yaml
└── mc
    └── http
        └── shouldi.yaml

3 directories, 2 files
```

Deploy Anywhere - HTTP

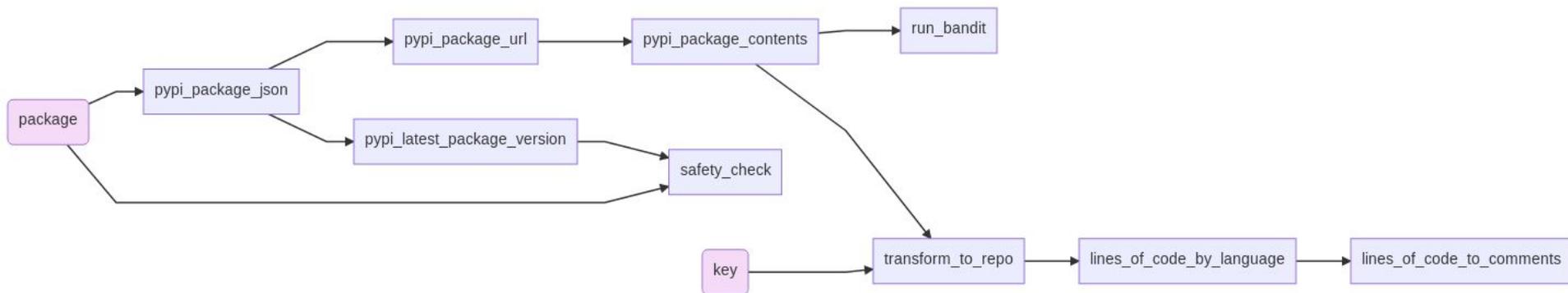
```
deploy $ cat mc/http/shouldi.yaml
path: /shouldi
presentation: json
asynchronous: false
deploy $
```

```
flow:
  bandit:
    pkg:
      - pkg.contents.directory
  pkg.cleanup:
    directory:
      - pkg.contents.directory
  pkg.contents:
    url:
      - pkg.url.url
  pkg.json:
    package:
      - seed
  pkg.url:
    response_json:
      - pkg.json.response_json
  pkg.version:
```

Deploy Anywhere - HTTP



Extend Without Writing Code - Modify DataFlow



Where To Go From Here

- How to Integrate Machine Learning Tutorial
 - <https://intel.github.io/dffml/usage/integration.html>
- shouldi
 - `pip install shouldi && shouldi install some-package-name`
 - <https://intel.github.io/dffml/tutorials/operations.html>
- Use and Contribute!
 - Weekly Meetings: Tuesdays at 9 AM
 - Gitter, Mailing List, and Meeting Links: <https://intel.github.io/dffml/community.html>
 - Documentation: <https://intel.github.io/dffml>
- Q&A