

# 2.1 Open Source Interpretability Tools

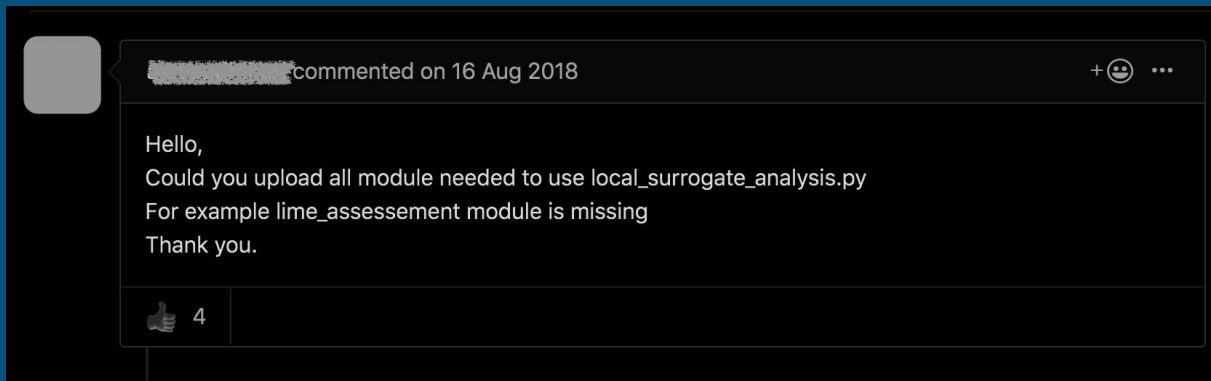
Using the Example of  
FAT Forensics

- Research software and reproducibility.
- Paperware.
- Interpretability, explainability and transparency toolkits.
- FAT Forensics and its design principles.

# Research Software

---

- Lucky to have it.
- Often not maintained.
- Sometimes lengthy Python scripts.
- Heavy and unnecessary dependencies (or simply unavailable).
- Over (or under) engineered.



# Reproducibility

---

Software underpins much of research, but it tends to lack a good foundation.

- *Vapourware* -- promised but not delivered, i.e., non-existent.
- **Paperware** -- available but difficult to use by people outside of the core research (and/or development) team.
- *Software* -- documented, tested, usable and welcoming.

We need `scikit-learn` for Fairness, Accountability and **Transparency**.

# Algorithmic Transparency

---

## Individual explainers:

- LIME  
<https://github.com/marcotcr/lime>
- Local surrogates  
<https://github.com/axa-rev-research/locality-interpretable-surrogate>
- Anchor  
<https://github.com/marcotcr/anchor>
- PyCEbox  
<https://github.com/AustinRochford/PyCEbox>

## Transparency/Interpretability/Explainability **packages**:

- Microsoft's InterpretML  
<https://github.com/interpretml/interpret>
- IBM's AI Explainability 360  
<https://github.com/IBM/AIX360>
- Oracle's Skater  
<https://github.com/oracle/Skater>
- ELI5  
<https://github.com/TeamHG-Memex/eli5>
- Yellowbrick  
<https://github.com/DistrictDataLabs/yellowbrick>

# FAT Forensics

---

Algorithmic Fairness, Accountability and Transparency Toolkit



# Origin

---

Creating a piece of software that covers fairness, accountability and transparency.



University of  
BRISTOL

THALES

Team:

- Kacper Sokol -- Lead Developer
- Alex Hepburn -- Core Developer
- Peter Flach -- Principal Investigator
- Rafael Poyiadzi -- Developer
- Matthew Clifford -- Developer
- Raul Santos-Rodriguez -- Co-Investigator

# Design and Development Principles

- Open sourced under the BSD 3-Clause licence.
- Minimal dependencies.
- Good software engineering practices:
  - unit testing;
  - continuous integration; and
  - consistent code styling and formatting.
- Complete and diverse documentation:
  - API reference;
  - online tutorials;
  - how-to guides; and
  - code examples.

Software	<a href="#">license</a> <a href="#">BSD-3-Clause</a> <a href="#">release</a> <a href="#">v0.1.0</a> <a href="#">pypi</a> <a href="#">v0.1.0</a> <a href="#">python</a> <a href="#">3.5</a>
Docs	<a href="#">homepage</a> <a href="#">read</a>
CI	<a href="#">build</a> <a href="#">passing</a> <a href="#">codecov</a> <a href="#">100%</a>
Try it	<a href="#">launch</a> <a href="#">binder</a>
Contact	<a href="#">mailing list</a> <a href="#">Google Groups</a> <a href="#">chat</a> <a href="#">on gitter</a>
Cite	<a href="#">cite</a> <a href="#">BibTeX</a> <a href="#">JOSS</a> <a href="#">10.21105/joss.01904</a> <a href="#">DOI</a> <a href="#">10.5281/zenodo.3833199</a>

# Scope

## Fairness

### Data

Do some data points share the same unprotected features but different protected features?

### Models

Is there demographic parity between certain sub-groups?

### Predictions

Are two data points that differ only in protected features treated differently?

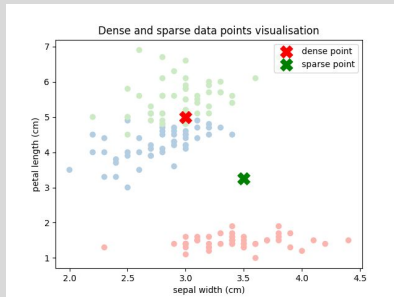
## Accountability

### Data

Is there a sampling bias in the data according to some sub-groups?

### Models

Is there a systematic performance bias in the model?



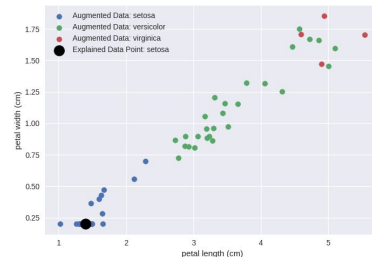
## Transparency

### Predictions

Why is a decision made?

### Models

What influence does each feature have on the model?

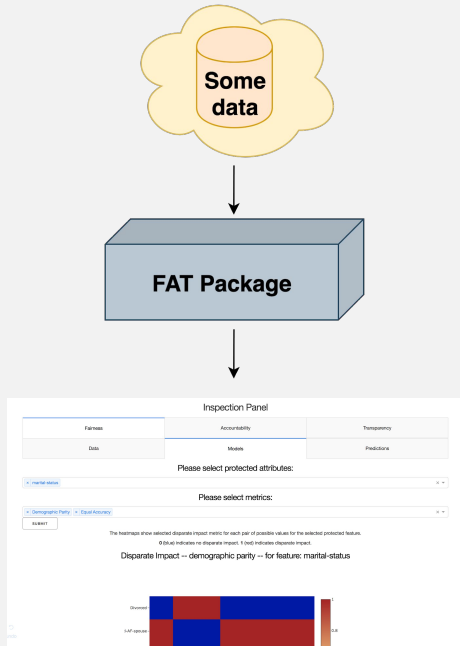




# Use Modes

## Deployment Mode

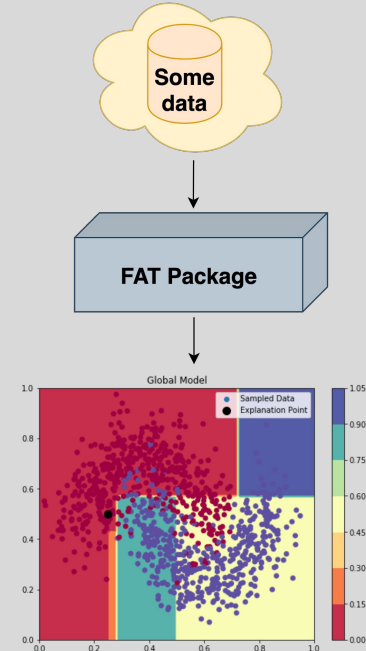
data in -- data out



<https://fatf.herokuapp.com/>

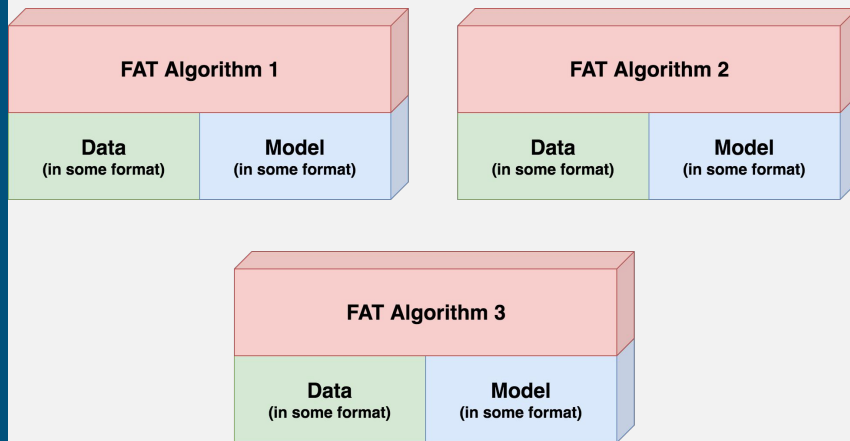
## Research Mode

data in -- visualisations out

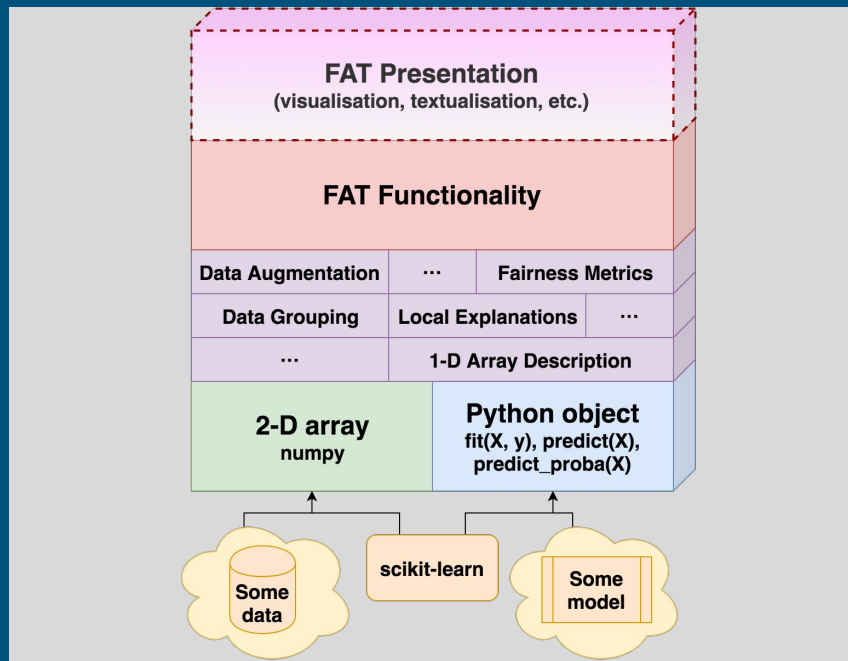


# Modularity

## Bespoke Code



## Modular Design



# Implemented Functionality

---

	Fairness	Accountability	Transparency
Data/ Features	<ul style="list-style-type: none"><li>• Systemic Bias (disparate treatment labelling).</li><li>• Sample size disparity (e.g., class imbalance).</li></ul>	<ul style="list-style-type: none"><li>• Sampling bias.</li><li>• Data Density Checker.</li></ul>	<ul style="list-style-type: none"><li>• Data description.</li></ul>
Models	<ul style="list-style-type: none"><li>• Group-based fairness (disparate impact).</li></ul>	<ul style="list-style-type: none"><li>• Systematic performance bias.</li></ul>	<ul style="list-style-type: none"><li>• Partial dependence.</li><li>• Individual conditional expectation.</li></ul>
Predictions	<ul style="list-style-type: none"><li>• Counterfactual fairness (disparate treatment).</li></ul>		<ul style="list-style-type: none"><li>• Counterfactuals.</li><li>• Tabular bLIMEy (LIME alternative).</li></ul>

# Planned Transparency Features

---

	Fairness	Accountability	Transparency
Data/ Features			<ul style="list-style-type: none"><li>• Bespoke Interpretable Representations</li></ul>
Models			<ul style="list-style-type: none"><li>• Permutation Importance</li><li>• Decision Tree Explainer</li></ul>
Predictions			<ul style="list-style-type: none"><li>• Anchors</li><li>• Image and Text Surrogates</li><li>• Tree-specific Counterfactuals</li></ul>

# <https://github.com/fat-forensics/fat-forensics>

The screenshot shows the GitHub repository page for `fat-forensics/fat-forensics`. The repository is owned by `fat-forensics` and has 4 watchers, 32 stars, and 9 forks. The main branch is `master`, with 8 branches and 3 tags. The repository description is "Modular Python Toolbox for Fairness, Accountability and Transparency Forensics". The repository is licensed under the BSD-3-Clause License. The repository has 39 commits, with the latest commit being `6fa252a` on 19 May. The repository is categorized under `machine-learning`, `fairness`, `accountability`, `transparency`, `interpretability`, `explainability`, `explainable-ai`, and `interpretable-ai`. The repository has 2 releases, with the latest release being `FAT-Forensics 0.1.0` on 19 May. The repository has 3 packages.

Search or jump to...

Pull requests Issues Marketplace Explore

fat-forensics / fat-forensics

Unwatch 4 Star 32 Fork 9

<> Code Issues Pull requests 7 Actions Security Insights Settings

master 8 branches 3 tags

Go to file Add file Code

**So-Cool Updated citation guidelines** 6fa252a on 19 May 39 commits

.github	Preparation for open-sourcing	12 months ago
build_tools	0.0.2 documentation update and documentation deployment fix (remove o...	10 months ago
doc	Updated citation guidelines	3 months ago
examples	0.1.0 release and JOSS publication (#32)	4 months ago
fatf	0.1.0 release and JOSS publication (#32)	4 months ago
.coveragerc	Travis yaml, linting and flake8 (close #4)	2 years ago
.editorconfig	Dev environment setup (#20)	2 years ago
.flake8	Package documentation for version 0.0.1 release (#28)	13 months ago
.gitignore	Tabular Surrogates (#29 and #26)	10 months ago
.mypy.ini	0.1.0 release and JOSS publication (#32)	4 months ago
.pylintrc	Dev environment setup (#20)	2 years ago
.style.yapf	Dev environment setup (#20)	2 years ago
.travis.yml	Travis CI skip_cleanup instead of cleanup -- still deploy v1	4 months ago

About

Modular Python Toolbox for Fairness, Accountability and Transparency Forensics

fat-forensics.org

machine-learning fairness accountability transparency interpretability explainability explainable-ai interpretable-ai

Readme


BSD-3-Clause License


Releases 3


FAT-Forensics 0.1.0 Latest on 19 May + 2 releases

Packages

# <https://fat-forensics.org/>

 **FAT Forensics** [Home](#) [Documentation](#) [FAT User Guide](#)

ENHANCED BY 



## Welcome to FAT Forensics!

FAT Forensics is a Python toolkit for evaluating Fairness, Accountability and Transparency of Artificial Intelligence systems. It is built on top of [SciPy](#) and [NumPy](#), and distributed under the 3-Clause BSD license (new BSD).

In addition to the code documentation, this web page also includes a detailed [User Guide](#) that describes FAT algorithms on a more theoretical level and talks about best practices when using them.

---

A great way to get yourself familiar with the package and where it comes from is the [Getting Started](#) page.

## Source Code

For hosting the source code we use the [FAT-Forensics](#) organisation on GitHub, with the source code for the FAT Forensics packed being held in the [fat-forensics](#) repository.

## Communication

We use a range of platforms for communication in the project:

- for issues with the source code or the documentation please open an issue on our [GitHub issue tracker](#);
- the code-related development discussion should happen on our [gitter](#) channel;
- the discussion about the project's future and the direction of the development happens on our [slack](#) channel and [mailing list](#).


## Acknowledgement

The project has started as an academic collaboration between the [University of Bristol](#) and Thales. You can find all of our contributors and more information about the support we receive [here](#).

---

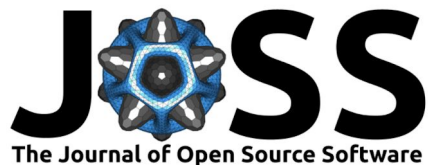
Please remember to [cite us](#) if you use any part of the package or its documentation.

© 2018–2020, Kacper Sokol et al.

 **THALES**  
More information on our contributors

[Show this page source](#)

<https://joss.theoj.org/papers/10.21105/joss.01904>



# FAT Forensics: A Python Toolbox for Implementing and Deploying Fairness, Accountability and Transparency Algorithms in Predictive Systems

**Kacper Sokol<sup>1</sup>, Alexander Hepburn<sup>2</sup>, Rafael Poyiadzi<sup>2</sup>, Matthew Clifford<sup>2</sup>, Raul Santos-Rodriguez<sup>2</sup>, and Peter Flach<sup>1</sup>**

<sup>1</sup> Department of Computer Science, University of Bristol <sup>2</sup> Department of Engineering Mathematics, University of Bristol

DOI: [10.21105/joss.01904](https://doi.org/10.21105/joss.01904)

<https://arxiv.org/abs/1909.05167>

---

**FAT Forensics:  
A Python Toolbox for Algorithmic Fairness,  
Accountability and Transparency**

Kacper Sokol  
K.Sokol@bristol.ac.uk  
Department of Computer Science,  
University of Bristol  
Bristol, United Kingdom

Raul Santos-Rodriguez  
enrsr@bristol.ac.uk  
Department of Engineering  
Mathematics, University of Bristol  
Bristol, United Kingdom

Peter Flach  
Peter.Flach@bristol.ac.uk  
Department of Computer Science,  
University of Bristol  
Bristol, United Kingdom



Next Up

---

# Hands-on Session Preparation

(Alex Hepburn)