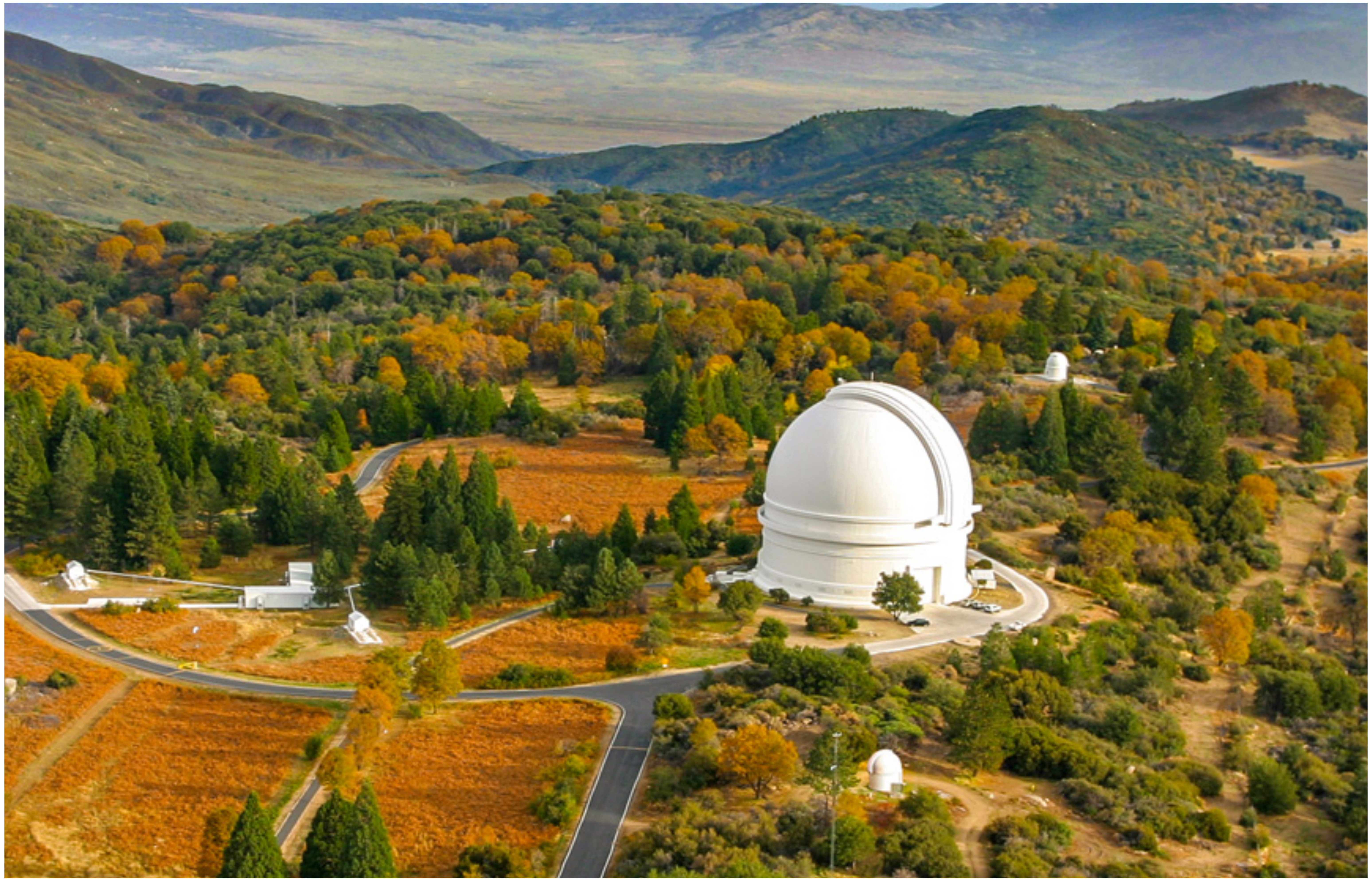
ZTF Source Classification Project (SCoPe)

Brian Healy
Postdoctoral Associate, University of Minnesota

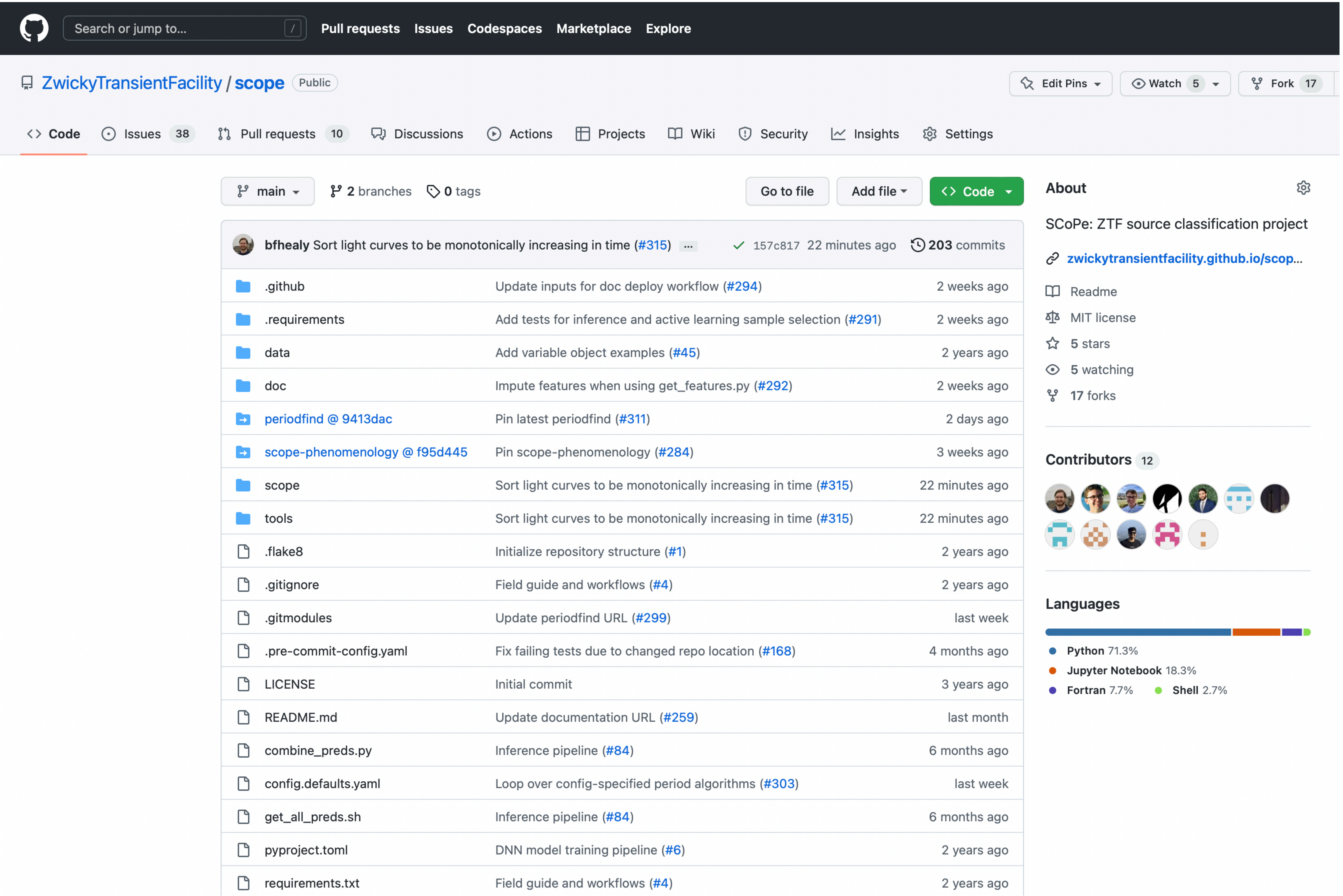
A3D3 Hackathon topic intro (MMA)

Zwicky Transient Facility (ZTF)



(Palomar Observatory/Caltech)

ZTF Source Classification Project (SCoPe)



ZwickyTransientFacility / scope Public

bfhealy Sort light curves to be monotonically increasing in time (#315) ... ✓ 157c817 22 minutes ago 203 commits

.github	Update inputs for doc deploy workflow (#294)	2 weeks ago
.requirements	Add tests for inference and active learning sample selection (#291)	2 weeks ago
data	Add variable object examples (#45)	2 years ago
doc	Impute features when using get_features.py (#292)	2 weeks ago
periodfind @ 9413dac	Pin latest periodfind (#311)	2 days ago
scope-phenomenology @ f95d445	Pin scope-phenomenology (#284)	3 weeks ago
scope	Sort light curves to be monotonically increasing in time (#315)	22 minutes ago
tools	Sort light curves to be monotonically increasing in time (#315)	22 minutes ago
.flake8	Initialize repository structure (#1)	2 years ago
.gitignore	Field guide and workflows (#4)	2 years ago
.gitmodules	Update periodfind URL (#299)	last week
.pre-commit-config.yaml	Fix failing tests due to changed repo location (#168)	4 months ago
LICENSE	Initial commit	3 years ago
README.md	Update documentation URL (#259)	last month
combine_preds.py	Inference pipeline (#84)	6 months ago
config.defaults.yaml	Loop over config-specified period algorithms (#303)	last week
get_all_preds.sh	Inference pipeline (#84)	6 months ago
pyproject.toml	DNN model training pipeline (#6)	2 years ago
requirements.txt	Field guide and workflows (#4)	2 years ago

About

SCoPe: ZTF source classification project

zwickytransientfacility.github.io/scop...

Readme

MIT license

5 stars

5 watching

17 forks

Contributors 12

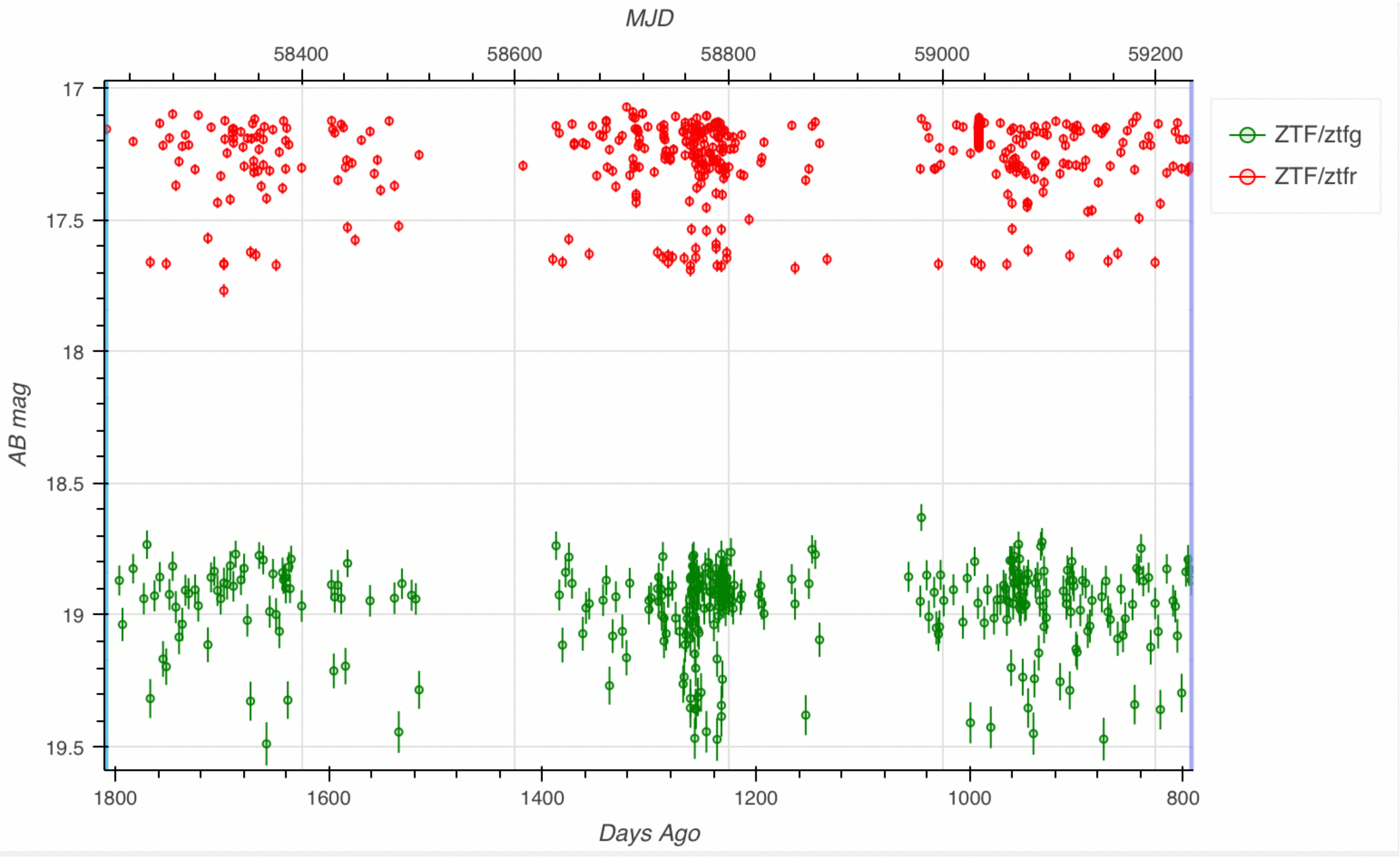
Languages

- Python 71.3%
- Jupyter Notebook 18.3%
- Fortran 7.7%
- Shell 2.7%

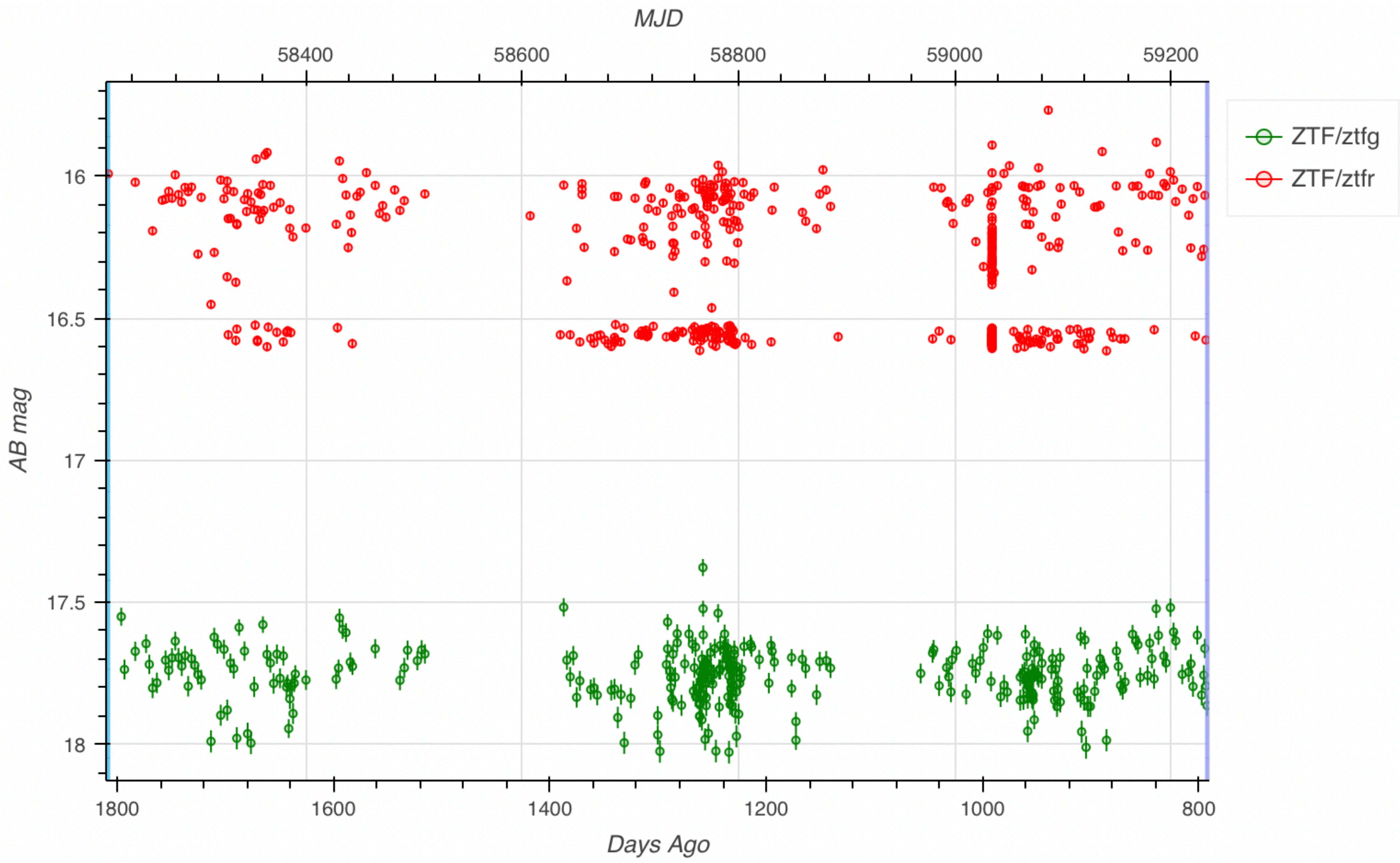
- Open-source
- Python-based
- CI/CD pipeline
- Regularly updated docs
- Hackathon topic!

(van Roestel et al. 2021, Coughlin et al. 2021)

Example ZTF Data

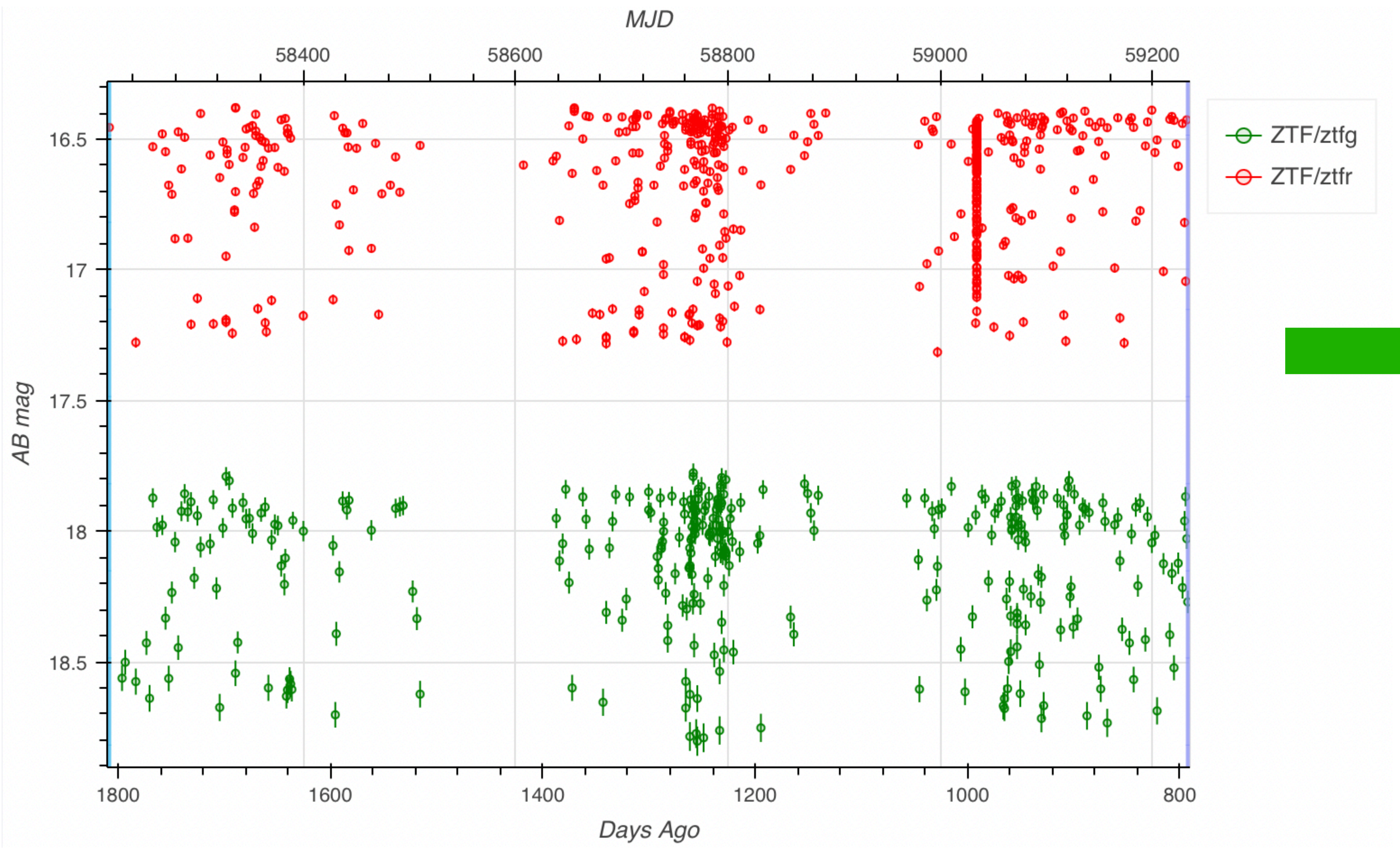


Eclipsing binary



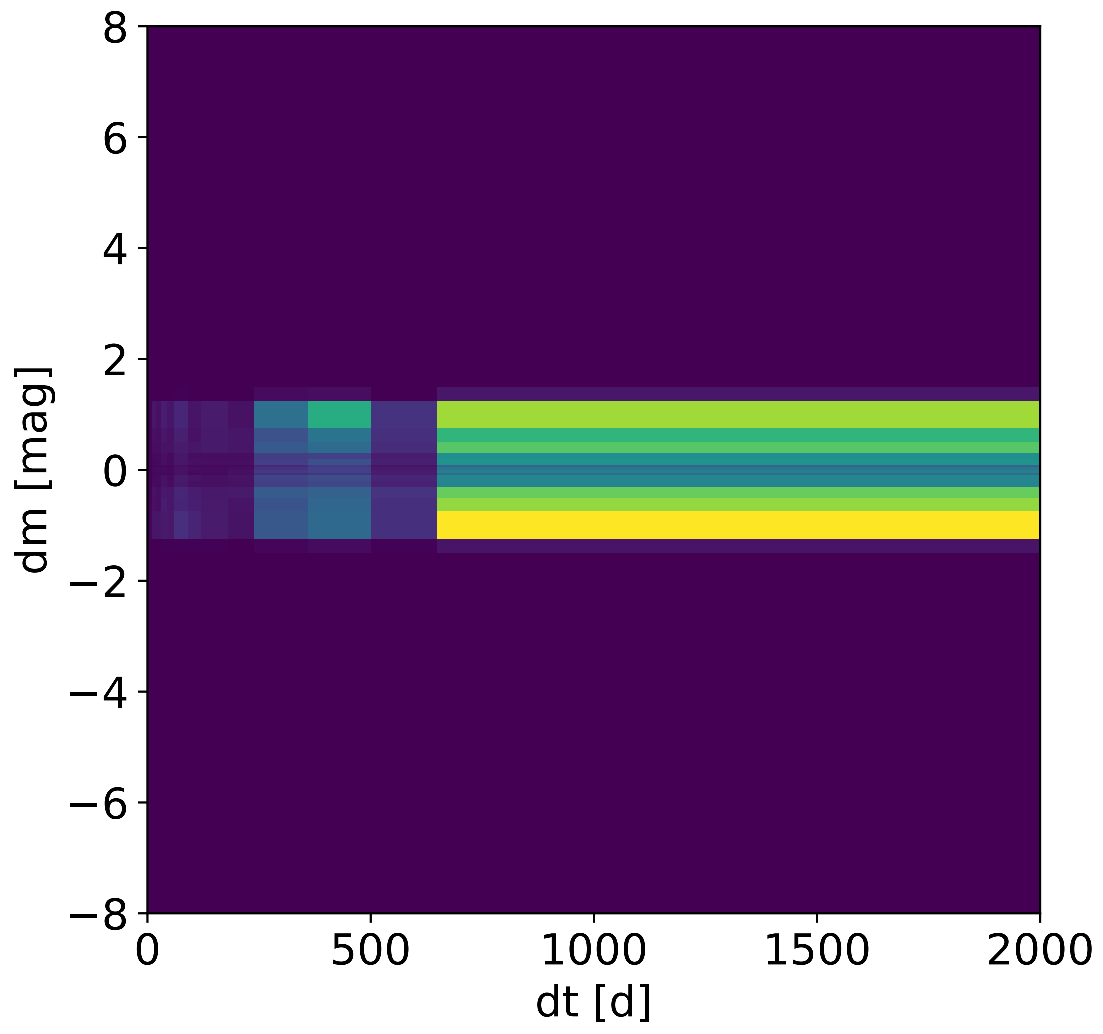
Blend

Input Features



Summary statistics

```
f1_a                        18.137365
f1_amp                       0.284065
f1_b                        -0.000034
f1_phi0                     -0.098306
f1_power                      0.93193
f1_relamp1                   0.082111
f1_relamp2                   0.024026
f1_relamp3                        0.0
f1_relamp4                        0.0
f1_relphi1                  -0.075452
f1_relphi2                   -0.03388
f1_relphi3                        0.0
f1_relphi4                        0.0
field                             853
i60r                            0.316
i70r                           0.3674
i80r                           0.4464
i90r                           0.5692
inv_vonneumannratio           0.71049
iqr                              0.25
mean_ztf_alert_braai          0.99176
median                         18.014
median_abs_dev                  0.101
n                               137.0
n_ztf_alerts                      406
norm_excess_var              0.000093
norm_peak_to_peak_amp        0.016749
pdot                              0.0
period                       0.175088
quad                                1
ra                         355.275604
roms                         3.267903
significance               167.419434
skew                        67.574478
smallkurt                  826.596439
stetson_j                    -1.10259
stetson_k                    0.815673
sw                           0.882065
welch_i                      71.02061
wmean                       18.050412
```



Magnitude-time histograms

SCoPe workflow

upload
expert review
query
initial
labeling
Database
Light Curves
Ext. Catalogs
Features
Classifications
Active learning
neural
network /
XGBoost
query
x-match
Fritz
Trained models
Inference
queries/
compute

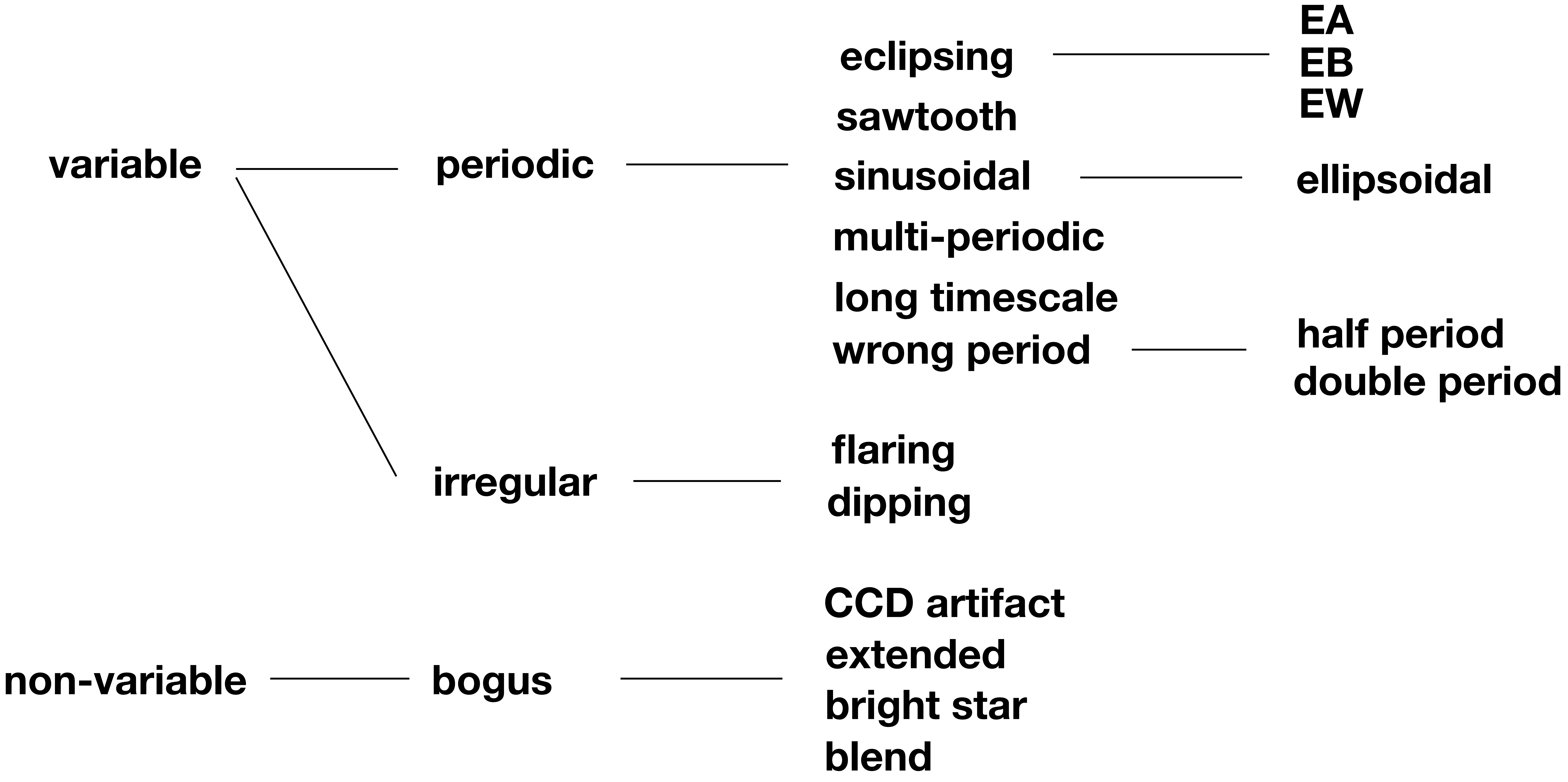
SCoPe Details

- **Supervised, active learning:** training set built up over time (w/human input)
- **Two taxonomies:** ontological (intrinsic), phenomenological (light curve shape)
    - Provides useful information for anomalous sources
    - Avoids complications of overlapping classes

SCoPe Details

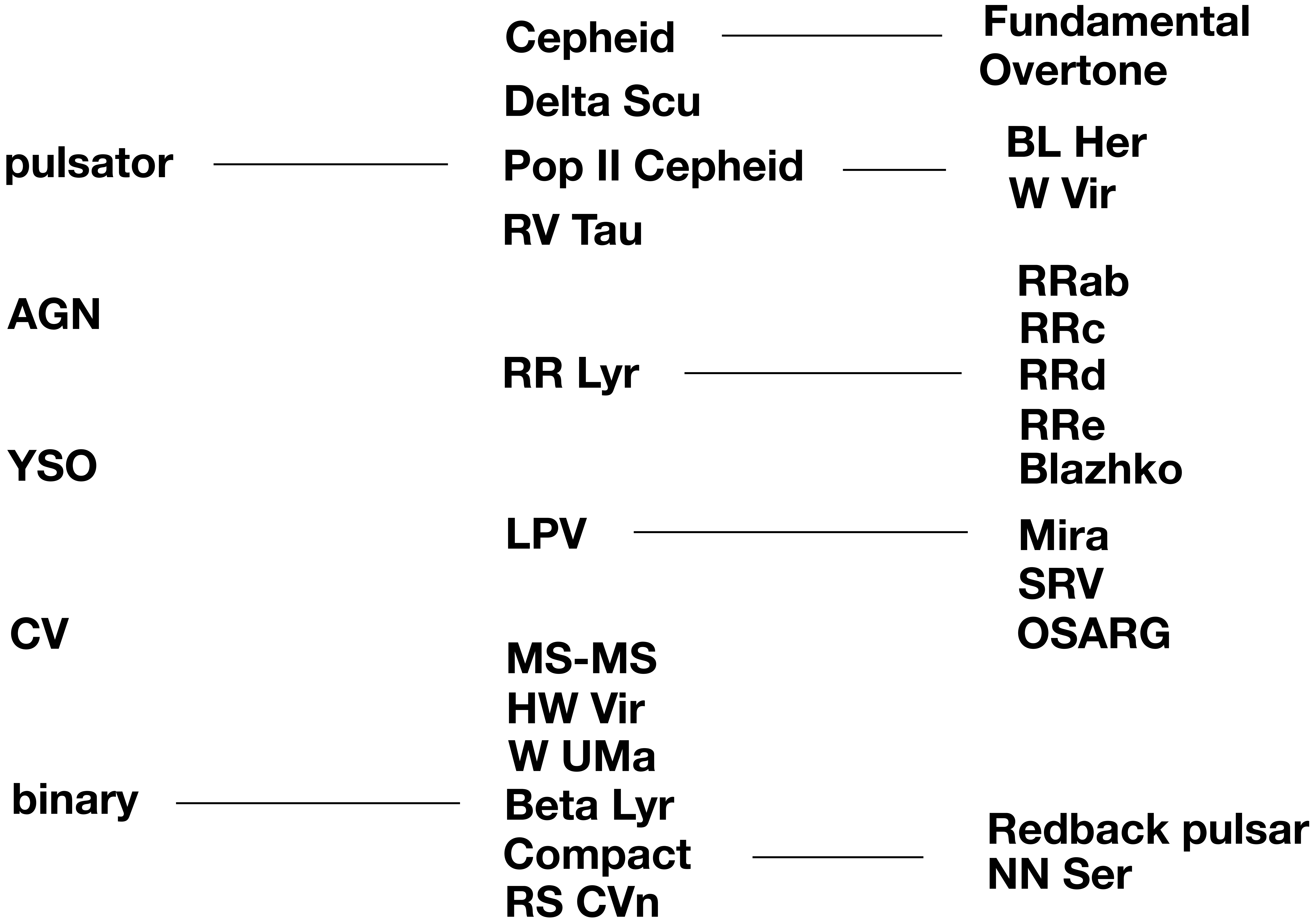
- **Binary classification:** independent predictions for each class
    - Supports multiple labels per source (varying specificity)
    - Flexible to new labels
- Train **convolutional neural network** and **XGBoost** algorithms on each label

Phenomenological Taxonomy



- variable
  - periodic
    - eclipsing
      - EA
      - EB
      - EW
    - sawtooth
    - sinusoidal
      - ellipsoidal
    - multi-periodic
    - long timescale
    - wrong period
      - half period
      - double period
  - irregular
    - flaring
    - dipping
- non-variable
  - bogus
    - CCD artifact
    - extended
    - bright star
    - blend

Ontological Taxonomy



pulsator

AGN

YSO

CV

binary

Cepheid

Delta Scu

Pop II Cepheid

RV Tau

RR Lyr

LPV

MS-MS

HW Vir

W UMa

Beta Lyr

Compact

RS CVn

Fundamental

Overtone

BL Her

W Vir

RRab

RRc

RRd

RRe

Blazhko

Mira

SRV

OSARG

Redback pulsar

NN Ser

Hackathon goals

- Modify/introduce ML classifiers
  - Adjust architecture of DNN, XGB (see scope/nn.py, scope/xgb.py)
  - Incorporate RNN or other algorithms (follow pattern of scope/nn.py, or start from scratch)
  - Training plots and results are saved to evaluate performance (models_dnn, models_xgb)
    - Compare these to current performance stats

SCoPe data on Google Drive

https://drive.google.com/drive/folders/13cm3Tf3RtudlVA5fBaMcHVi8anOnn8Gy?usp=sharing

- **Full training set (170632 rows × 247 columns)**
  - Parquet file (see SCoPe docs for comparison with CSV, HDF5)
  - Columns: obj_id, ra, dec; labels; features
  - See **SCoPe training set column guide** Google Sheet
- **10% training set (17063 rows × 247 columns)**

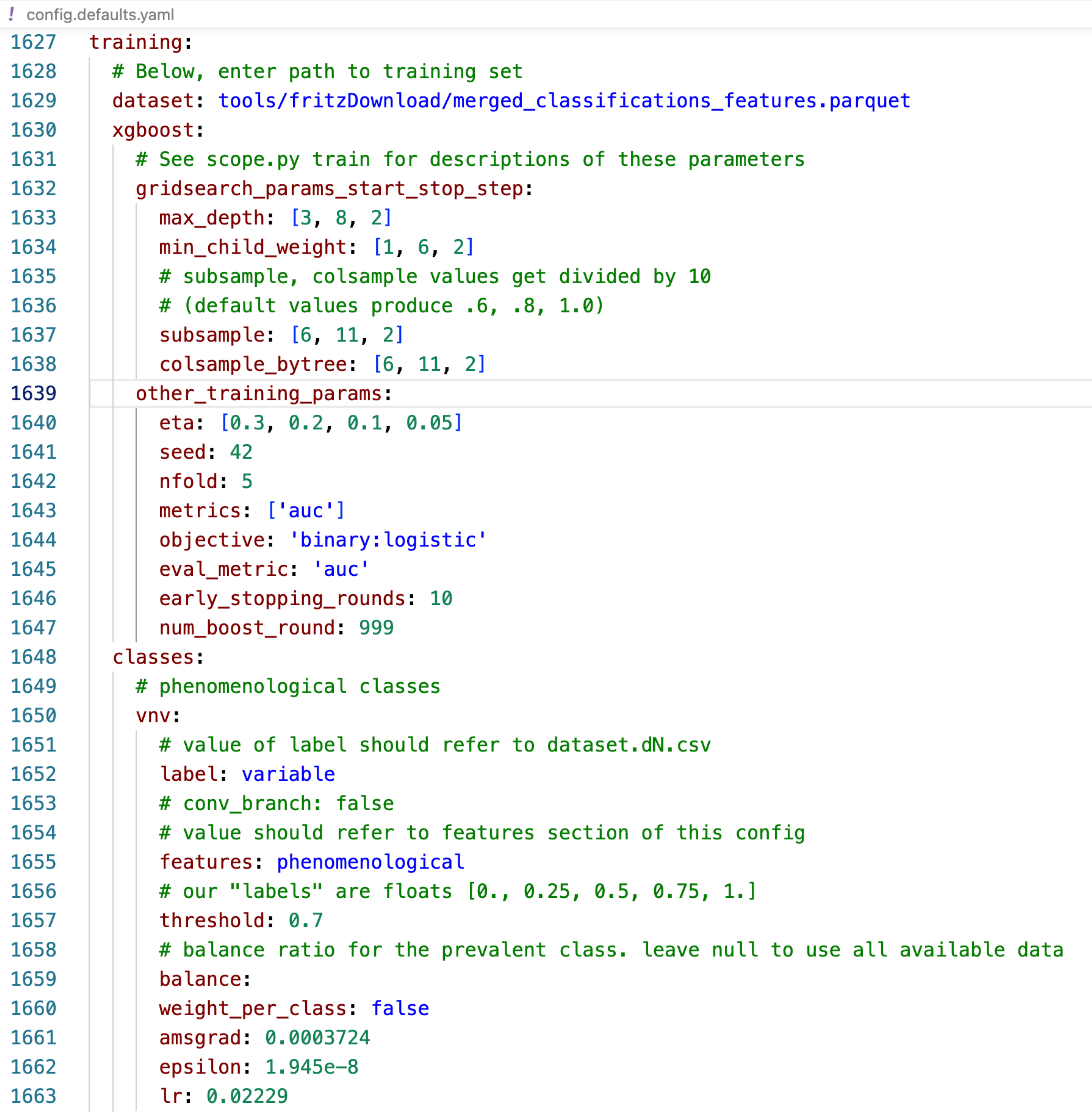
  -
    ```
    full_training_set.sample(frac=0.1, \
    random_state=9).reset_index(drop=True)
    ```
- **SCoPe model performance JSON files (DNN and XGB)**

SCoPe Installation

- Follow documentation here: https://zwickytransientfacility.github.io/scope-docs/
- Create your own fork of the SCoPe repo: https://github.com/ZwickyTransientFacility/scope
    - Also a version on the hackathon organization, but may not always be up-to-date
- `git clone` <link to your forked repo>
- `git remote add upstream` <link to ZTF SCoPe repo>
- Use conda/pip to install requirements in a new `scope-env`
- Test by running `./scope.py test_limited`

Use config.yaml to specify training details

- Copy config.defaults.yaml to a new config.yaml file
- Use `training:` config section to set and modify training details
    - Training set path, features to include, classification thresholds, hyperparameters, etc.



! config.defaults.yaml

```yaml
training:
  # Below, enter path to training set
  dataset: tools/fritzDownload/merged_classifications_features.parquet
  xgboost:
    # See scope.py train for descriptions of these parameters
    gridsearch_params_start_stop_step:
      max_depth: [3, 8, 2]
      min_child_weight: [1, 6, 2]
      # subsample, colsample values get divided by 10
      # (default values produce .6, .8, 1.0)
      subsample: [6, 11, 2]
      colsample_bytree: [6, 11, 2]
    other_training_params:
      eta: [0.3, 0.2, 0.1, 0.05]
      seed: 42
      nfold: 5
      metrics: ['auc']
      objective: 'binary:logistic'
      eval_metric: 'auc'
      early_stopping_rounds: 10
      num_boost_round: 999
  classes:
    # phenomenological classes
    vnv:
      # value of label should refer to dataset.dN.csv
      label: variable
      # conv_branch: false
      # value should refer to features section of this config
      features: phenomenological
      # our "labels" are floats [0., 0.25, 0.5, 0.75, 1.]
      threshold: 0.7
      # balance ratio for the prevalent class. leave null to use all available data
      balance:
      weight_per_class: false
      amsgrad: 0.0003724
      epsilon: 1.945e-8
      lr: 0.02229
```

Training ML algorithms with SCoPe

- Default train/val/test split: 0.81/0.09/0.10

-
  ```
  ./scope.py train --tag=vnv --path_dataset=/path/to/
  trainingset.parquet --verbose --save --plot
  --group=group_name --period_suffix=ELS_ECE_EAOV
  ```

-
  ```
  ./scope.py train --tag=vnv --algorithm=xgb
  --path_dataset=/path/to/trainingset.parquet
  --verbose --save --plot --group=xgb_group_name
  --period_suffix=ELS_ECE_EAOV
  ```

Other notes

- SCoPe has utilities to read parquet files, manage datasets
  - `from scope.utils import read_parquet, Dataset`
- Some labels in training set can are inconsistently applied
  - Many sources with incorrect periods are not labeled “wrong period”
- Use caution when improving stats by selectively reducing size of training data (e.g. the `balance` hyperparameter)
  - Can get deceptively good results because classifier is not exposed to enough types of negative examples