OpenML

FRICTIONLESS, COLLABORATIVE, AUTOMATED MACHINE LEARNING

JOAQUIN VANSCHOREN, TU/E

www.openml.org

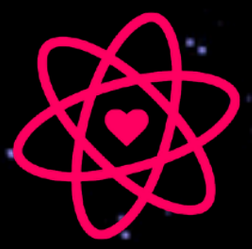
@open_ml

Machine learning: lots of unnecessary friction

Scattered, ill-described, datasets
Manual searching, reformatting, making assumptions

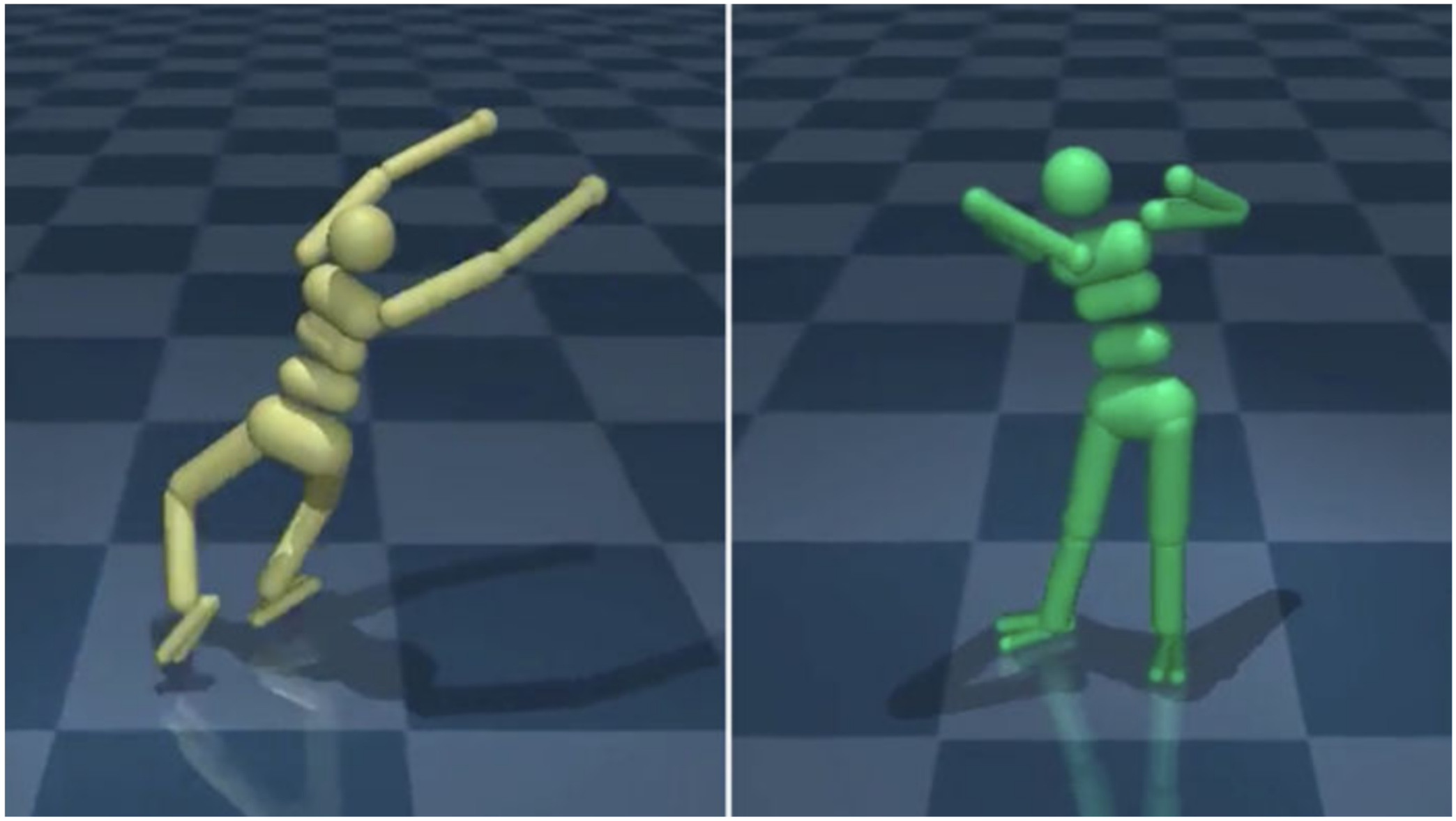
Myriad algorithms, versions, languages
Manually setting up experiments, storing results,…

Reproducibility is hard
Manually tracking every detail is error-prone



Hard to discover and build on prior results
No standards / hubs for sharing and organizing results

Hard to automate model building end-to-end
Requires automated data organization, clean APIs,…



The same algorithm can learn to walk in wildly different ways. YUVAL TASSA

Missing data hinder replication of artificial intelligence studies

By Matthew Hutson | Feb. 15, 2018 , 12:30 PM

Other sciences have figured this out

NETWORKED SCIENCE

move data out of people's labs and minds, and into online tools that help us structure and reuse the information in new ways - M. Nielsen

Designed serendipity

Broadcast ***data***, so people may use it in unexpected ways
Broadcast ***questions***, combine many minds to solve it
Somewhere, someone has just the right data or ideas to help

Requires frictionless collaboration

Open, organized, interconnected data (collective working memory)
Contribute new data/ideas quickly, effortlessly
Reputation building, incentives that reward openness

WHAT IF WE CAN ANALYSE DATA

COLLABORATIVELY

WHAT IF WE CAN ANALYSE DATA

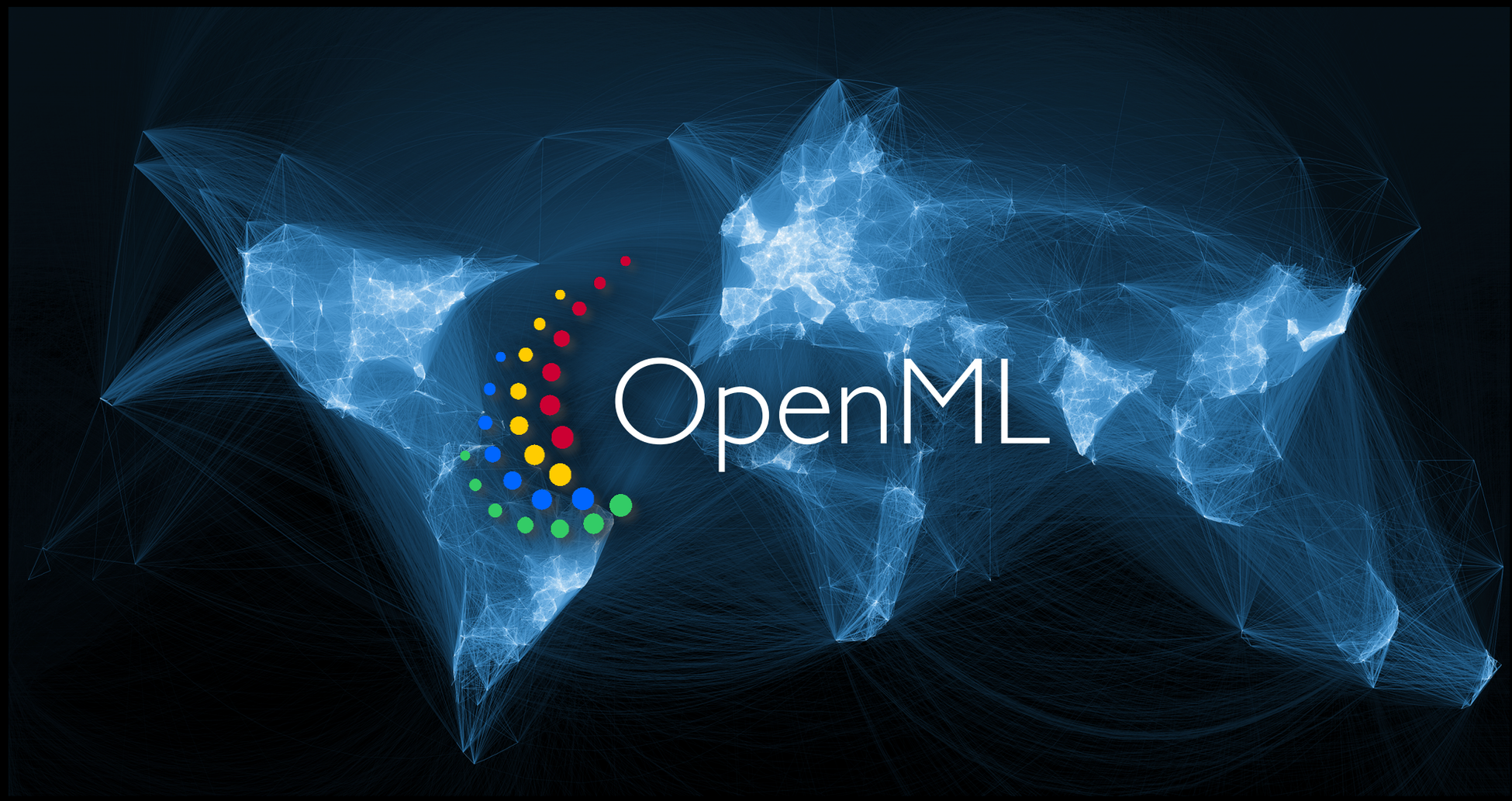
COLLABORATIVELY

ON WEB SCALE

WHAT IF WE CAN ANALYSE DATA

COLLABORATIVELY

ON WEB SCALE IN REAL TIME



Easy to use: Integrated in many ML tools/environments

Easy to contribute: Automated sharing of data, code, results

Organized data: Meta-data, reproducible models, link to people

Reward structure: Track your impact, build reputation

Self-learning: Learn from many experiments to help people

OpenML: Components

Datasets: Auto-annotated, organized, well-formatted
Find the datasets you need, share your own

Tasks: Auto-generated, machine-readable
Everyone's results are directly comparable

Flows: Pipelines/code that build ML models
Run locally (or wherever), auto-upload all results

Runs: All results from running flows on tasks
All details needed for tracking and reproducibility
Evaluations can be queried, compared, reused

OpenML Search GUIDE SIGN IN

www.openml.org

OpenML beta

Machine learning, better, together

19903 data sets

Find or add **data** to analyse

46553 tasks

Download or create scientific **tasks**

4539 flows

Find or add data analysis **flows**

5913832 runs

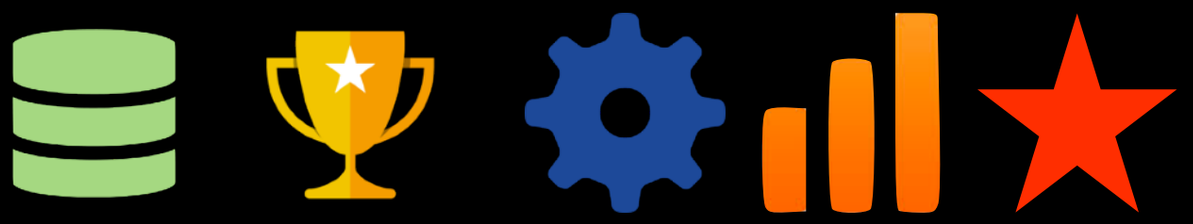
Upload and explore all **results** online.

HACKATHON

Bring your own data, bring your own algorithms, or build cool new features.

Next location: 9-14 October in Leiden, the Netherlands

OPENML API



REST (XML/JSON), PYTHON, R, JAVA,...

Interact with OpenML any way you want. All data is open.

- Search Data, Flows, Models, Evaluations
- Download data, meta-data and runs
- Upload new data, flows, runs
- Program against OpenML (e.g. easy benchmarking)
- Build your own bots that automate processes
  - Data cleaning, annotation, model building, evaluation

It starts with data

It starts with data

Data (tabular) easily uploaded or referenced (URL)

OpenML

```
from openml import datasets
my_data = datasets.OpenMLDataset(data_file='omg.arff',
                                 name='OMG_data',
                                 description='We want to...',
                                 licence='Public')
response = my_data.publish()
```

Name	Version

interoperability

Data can remain in existing repositories

-> registered via URL, transparent to users

For now: only tabular data

-> ARFF or CSV import (auto-annotate features)

-> FrictionlessData support in the works

auto-versioned, analysed, organised online

OpenML Search

19587 results

Only showing active datasets (public or shared with you).

FILTERS SORT: **MOST RUNS** ID'S TABLE + ADD NEW

credit-g (1)
This dataset classifies people described by a set of attributes as good or bad c...
★ **439801 runs** ♥ **2 likes** **39 downloads** **41 reach** **11 impact**
1000 instances - 21 features - 2 classes - 0 missing values

blood-transfusion-service-c...
Data taken from the Blood Transfusion Service Center in Hsin-Chu City in Taiw...
★ **424540 runs** ♥ **1 likes** **21 downloads** **22 reach** **13 impact**
748 instances - 5 features - 2 classes - 0 missing values

Search

```
import openml as oml
openml_list = oml.datasets.list_datasets()
```

```
3
4 datalist = pandas.DataFrame.from_dict(openml_list, orient='index
5 datalist = datalist[['did','name','NumberOfInstances','NumberOfF
6 print("Found %s datasets" % len(datalist))
7 datalist.head(n=10)
```

```
Found 2522 datasets
```

Out[4]:

	did	name	NumberOfInstances	NumberOfFeatures	NumberOfClasses
2	2	anneal	898	39	5
3	3	kr-vs-kp	3196	37	2
4	4	labor	57	17	2
5	5	arrhythmia	452	280	13
6	6	letter	20000	17	26
7	7	audiology	226	70	24

Search

```
import openml as oml
openml_list = oml.datasets.list_datasets()
```

```
datalist = pandas.DataFrame.from_dict(openml_list
```

```
datalist.query('name == "eeg-eye-state"')
```

```
datalist.query('NumberOfClasses > 50')
```

```
datalist[datalist.NumberOfInstances>10000
        ].sort_values(['NumberOfInstances'])
```

Get

In [6]:

```
dataset = oml.datasets.get_dataset(1471)
```

```
# Print a summary
print("This is dataset '%s', the target feature is '%s'" %
      (dataset.name, dataset.default_target_attribute))
print("URL: %s" % dataset.url)
print(dataset.description[:500])
```

```
This is dataset 'eeg-eye-state', the target feature is 'Class'
URL: https://www.openml.org/data/download/1587924/eeg-eye-state
**Author**: Oliver Roesler, it12148'@'lehre.dhbw-stuttgart.de
**Source**: [UCI](https://archive.ics.uci.edu/ml/datasets/EEG+E
rsity (DHBW), Stuttgart, Germany
**Please cite**:

All data is from one continuous EEG measurement with the Emotiv
 117 seconds. The eye state was detected via a camera during th
 after analysing the video fr
```

Download

```
X, y, attribute_names = dataset.get_data(
    target=dataset.default_target_attribute,
    return_attribute_names=True)
```

```
5 print(eeg[:10])
```

```
          V1           V2           V3           V4           V5  \
0  4329.229980  4009.229980  4289.229980  4148.209961  4350.259766
1  4324.620117  4004.620117  4293.850098  4148.720215  4342.049805
2  4327.689941  4006.669922  4295.379883  4156.410156  4336.919922
3  4328.720215  4011.790039  4296.410156  4155.899902  4343.589844
4  4326.149902  4011.790039  4292.310059  4151.279785  4347.689941
5  4321.029785  4004.620117  4284.100098  4153.330078  4345.640137
6  4319.490234  4001.030029  4280.509766  4151.790039  4343.589844
7  4325.640137  4006.669922  4278.459961  4143.080078  4344.100098
8  4326.149902  4010.770020  4276.410156  4139.490234  4345.129883
9  4326.149902  4011.280029  4276.919922  4142.049805  4344.100098
```

Fit

In [9]:

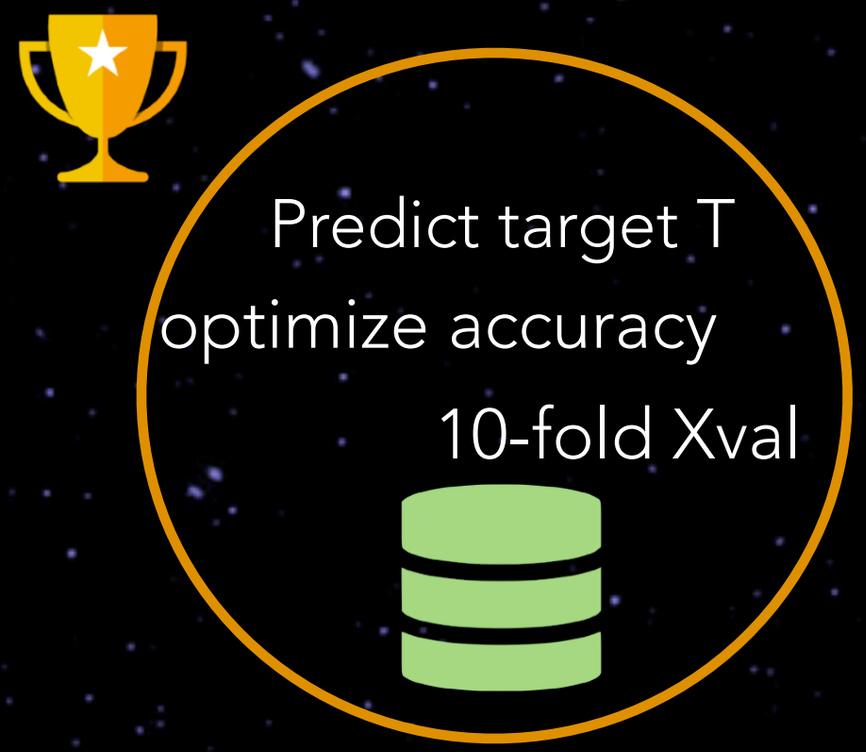
```
1  from sklearn import neighbors
2
3  dataset = oml.datasets.get_dataset(1471)
4  X, y = dataset.get_data(target=dataset.default_targe
5  clf = neighbors.KNeighborsClassifier(n_neighbors=1)
6  clf.fit(X, y)
```

Out[9]:

```
KNeighborsClassifier(algorithm='auto', leaf_size=30, met
                     metric_params=None, n_jobs=1, n_neighbors=1,
                     weights='uniform')
```

Doesn’t tell you how to evaluate, do what you want

Why analyze this data?



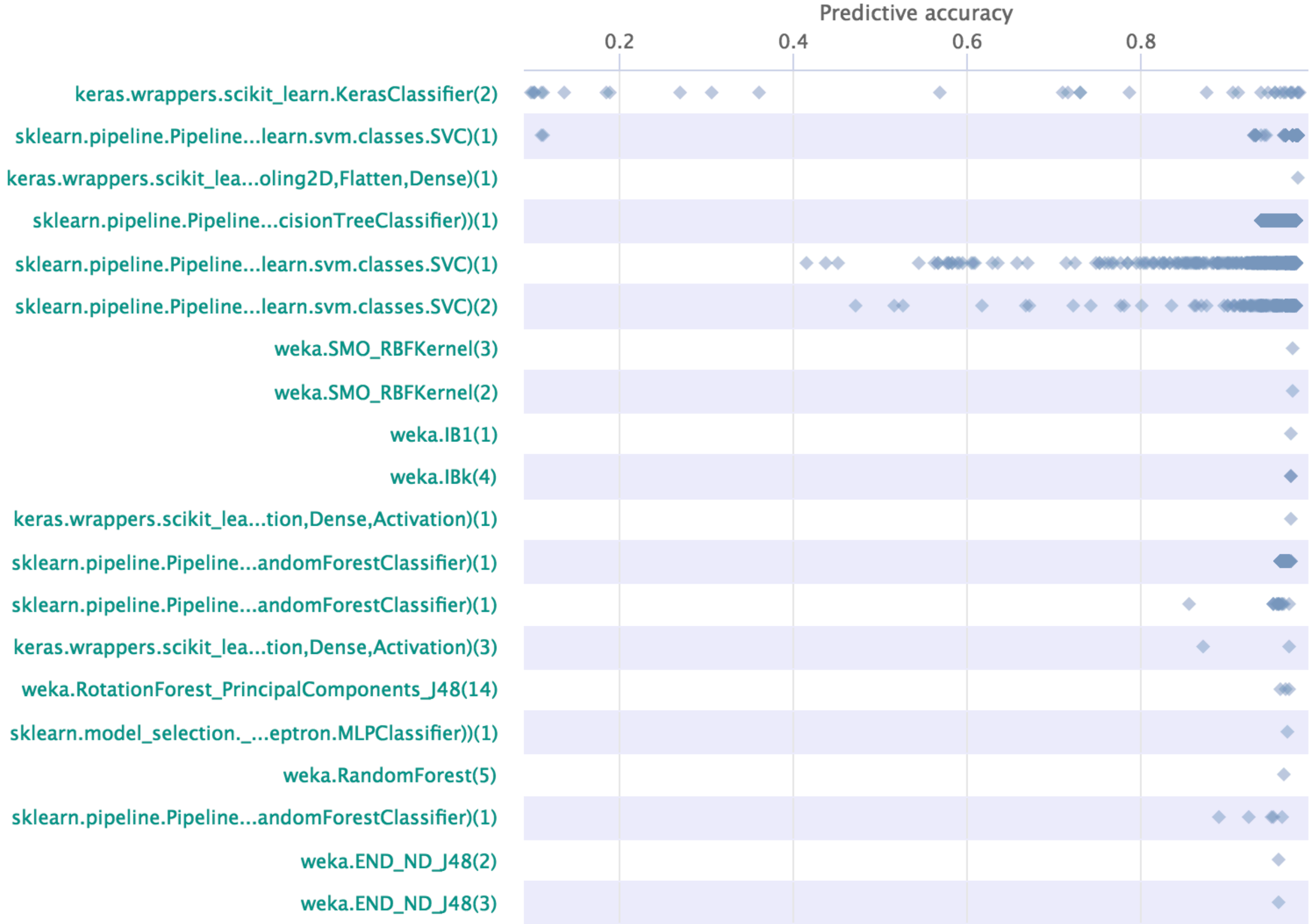
Tasks contain data, goals, procedures.

Auto-build + evaluate models correctly

All evaluations are directly comparable

Evaluations per flow (multiple parameter settings)

every point is a run, click for details



Search

In [11]:

```
task_list = oml.tasks.list_tasks()
```

```
3 mytasks = pd.DataFrame.from_dict(task_list, orient='index')
4 mytasks = mytasks[['tid','did','name','task_type','estimation_pr
5 print("First 5 of %s tasks:" % len(mytasks))
6 mytasks.head()
```

```
First 5 of 5000 tasks:
```

Out[11]:

	tid	did	name	task_type	estimation_procedure	evaluation_measures
2	2	2	anneal	Supervised Classification	10-fold Crossvalidation	predictive_accuracy
3	3	3	kr-vs-kp	Supervised Classification	10-fold Crossvalidation	predictive_accuracy
4	4	4	labor	Supervised Classification	10-fold Crossvalidation	predictive_accuracy
5	5	5	arrhythmia	Supervised Classification	10-fold Crossvalidation	predictive_accuracy
6	6	6	letter	Supervised Classification	10-fold Crossvalidation	predictive_accuracy

Get

```
task = oml.tasks.get_task(14951)
```

```
2 task = oml.tasks.get_task(14951)
3 pprint(vars(task))
```

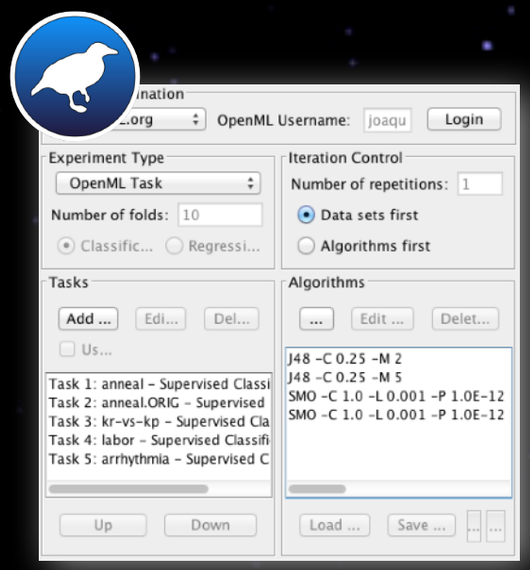
```
{'class_labels': ['1', '2'],
 'cost_matrix': None,
 'dataset_id': 1471,
 'estimation_parameters': {'number_folds': '10',
                           'number_repeats': '1',
                           'percentage': '',
                           'stratified_sampling': 'true'},
 'estimation_procedure': {'data_splits_url': 'https://www.ope
```

Easily train many algorithms

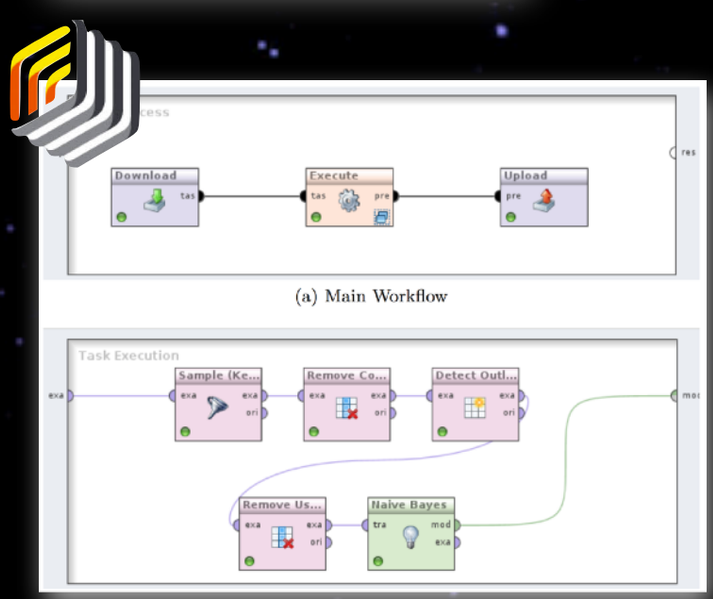
Auto-run algorithms/workflows on any task

Integrated in many machine learning tools (+ APIs)

Integrated in many machine learning tools (+ APIs)




```python
from sklearn import ensemble
from openml import tasks, flows, runs
task = tasks.get_task(3954)
clf = ensemble.RandomForestClassifier()
flow = flows.sklearn_to_flow(clf)
run = runs.run_flow_on_task(task, flow)
result = run.publish()
```



(a) Main Workflow

```r
library(OpenML)
library(mlr)
task = getOMLTask(10)
lrn = makeLearner("classif.rpart")
res = runTaskMlr(task, lrn)
run.id = uploadOMLRun(res)
```

Integrated in many machine learning tools (+ APIs)

Run locally (or wherever you want)

```
from sklearn import ensemble
from openml import tasks, flows, runs
task = tasks.get_task(3954)
clf = ensemble.RandomForestClassifier()
flow = flows.sklearn_to_flow(clf)
run = runs.run_flow_on_task(task, flow)
result = run.publish()
```

Pipelines, Code

```
from sklearn import pipeline, ensemble, preprocessing
from openml import tasks,runs, datasets
task = tasks.get_task(59)
pipe = pipeline.Pipeline(steps=[
              ('Imputer', preprocessing.Imputer(strategy='median')),
              ('OneHotEncoder', preprocessing.OneHotEncoder(sparse=F
              ('Classifier', ensemble.RandomForestClassifier())
             ])
flow = oml.flows.sklearn_to_flow(pipe)

run = oml.runs.run_flow_on_task(task, flow)
myrun = run.publish()
print("Uploaded to http://www.openml.org/r/" + str(myrun.run_id))
```

```
Uploaded to http://www.openml.org/r/7943199
```

REST API allows uploading code (or GitHub repo's)

Python API not yet

Flows

sklearn.model_selection._search.RandomizedSea

Visibility: public Uploaded 05-10-2017 by Matthias Feurer sklearn==0.18.1 numpy>=1.6.1 scipy>=0.9 100 runs

0 likes downloaded by 0 people 0 issues 0 downvotes , 0 total downloads

openml-python python scikit-learn sklearn sklearn_0.18.1 study_14 Verified_Supervised_Classification + Add tag

Loading wiki

Automatically created scikit-learn flow.

Components

estimator	sklearn.pipeline.Pipeline(Imputer=openmlstudy14.preprocessing.ConditionalImputer,OneHotEncoder=sklearn.preproces (1)

Parameters

Keras

Parameters

batch_size	default: 32
build_fn	default: {"oml-python:serialized_object": "function", "value": "__main__.lenet_fmnist"}
epochs	default: 5
layer0	default: {"class_name": "Reshape", "config": {"batch_input_shape": [null, 784], "dtype": "float32", "name": "reshape_1", "target_shape": [28, 28, 1], "trainable": true}}
layer1	default: {"class_name": "Conv2D", "config": {"activation": "linear", "activity_regularizer": null, "batch_input_shape": [null, 1, 28, 28], "bias_constraint": null, "bias_initializer": {"class_name": "Zeros", "config": {}}, "bias_regularizer": null, "data_format": "channels_last", "dilation_rate": [1, 1], "dtype": "float32", "filters": 6, "kernel_constraint": null, "kernel_initializer": {"class_name": "VarianceScaling", "config": {"distribution": "uniform", "mode": "fan_avg", "scale": 1.0, "seed": null}}, "kernel_regularizer": null, "kernel_size": [5, 5], "name": "conv2d_1", "padding": "valid", "strides": [1, 1], "trainable": true, "use_bias": true}}
layer10	default: {"class_name": "Dense", "config": {"activation": "linear", "activity_regularizer": null, "bias_constraint": null, "bias_initializer": {"class_name": "Zeros", "config": {}}, "bias_regularizer": null, "kernel_constraint": null, "kernel_initializer": {"class_name": "VarianceScaling", "config": {"distribution": "uniform", "mode": "fan_avg", "scale": 1.0, "seed": null}}, "kernel_regularizer": null, "name": "dense_1", "trainable": true, "units": 84, "use_bias": true}}

What works?

Experiments auto-uploaded, evaluated online

reproducible, linked to **data, flows, authors**

and **all other experiments**

Experiments auto-uploaded, evaluated online

Result files

Description

XML file describing the run, including user-defined evaluation measures.

Model readable

A human-readable description of the model that was built.

Model serialized

A serialized description of the model that can be read by the tool that generated it.

Predictions

ARFF file with instance-level predictions generated by the model.

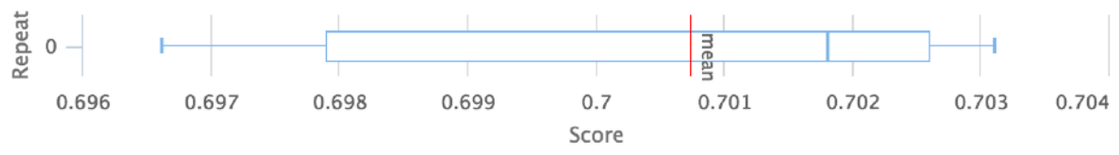
Area under ROC curve

0.7007 ± 0.0023

Per class

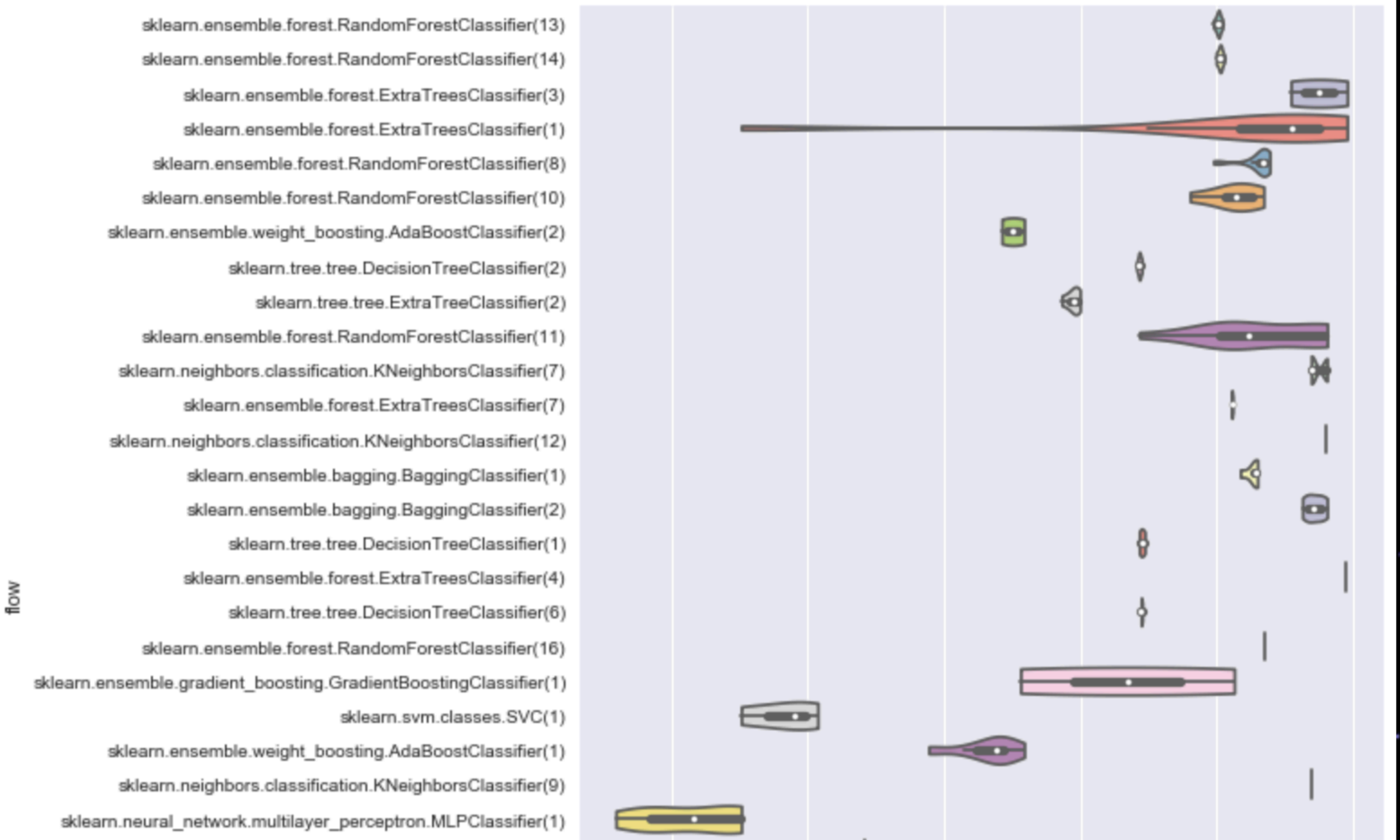
0	1
0.7007	0.7007

Cross-validation details (10-fold Crossvalidation)



★ Search, download, reuse

```
1  fig, ax = pyplot.subplots(figsize=(8, 12))
2  sns.violinplot(x="score", y="flow", data=pd.DataFrame(scores), scale="width", palette="Se
```



Publishing, impact tracking

Heidi Seibold

PhD student in Computational Biostatistics at the University of Zurich. I am into R, open science and reproducible research.

University of Zurich Joined 2016-01-27

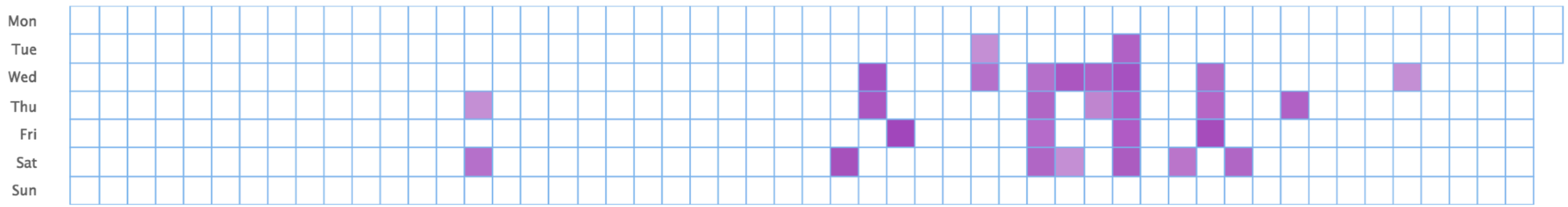
Activity 1649.5 | Reach 12 | Impact 195 | Uploads 1 | 35 | 0 | 1605

EDIT PROFILE

	Activity	Reach		Impact
Data Sets	1	4	0	0
Flows	35	7	0	195
Tasks	0	0	0	
Runs	1605	1	0	

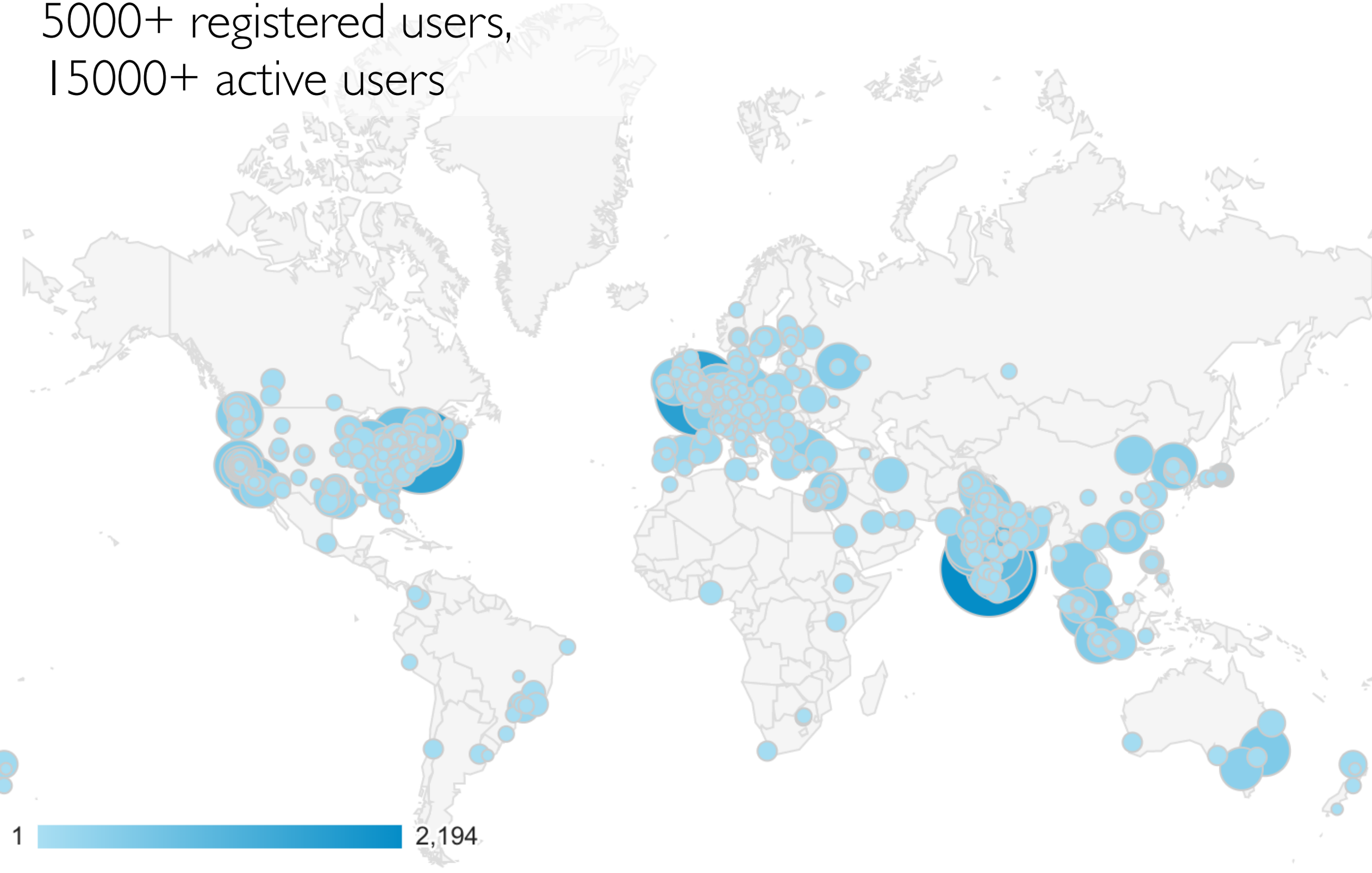
Activity: 1.65K | 1.64K | 0 | 17

Activity from Sunday 2016-05-29 to Monday 2017-05-29



OpenML Community

5000+ registered users,
15000+ active users



In progress (all help welcome!)

Sharing settings
Share datasets, flows, studies with certain people.

Studies
Online counterpart of a paper, reproducible results

Integrations with Jupyter, GitHub,...
Link to notebooks, export to GitHub

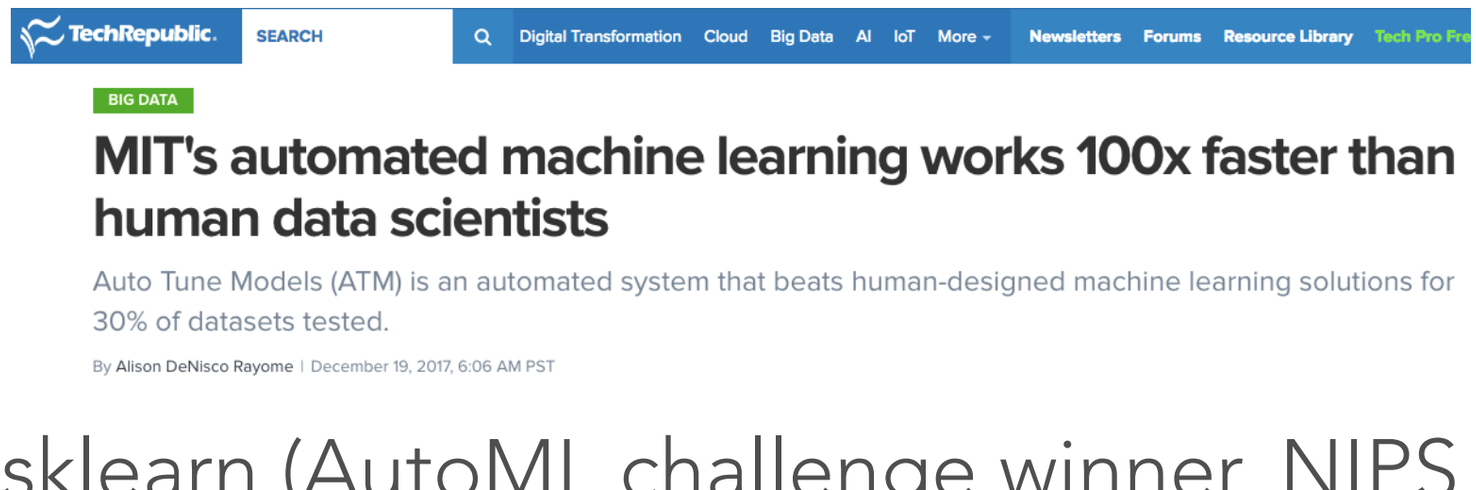
Learning to learn
Bots that learn from all prior experiments
Clean data, design pipelines, optimize models

REUSING OPENML (META)DATA

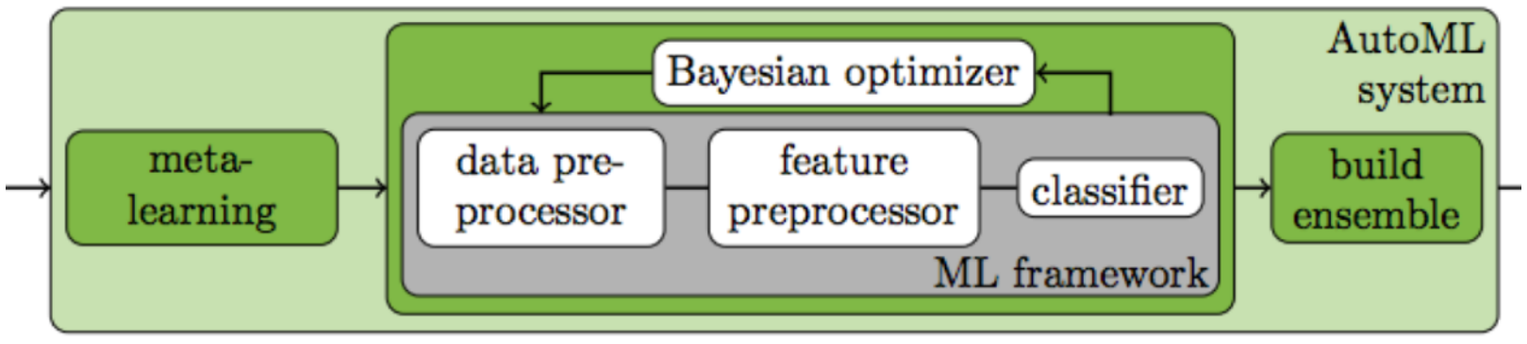
- **Find similar datasets**
  - 20,000+ versioned datasets, with 130+ meta-features
- **Reuse (millions of) prior model evaluations:**
  - Benchmark new algorithms against state-of-the-art
  - Meta-models: E.g. predict performance or training time
- **Reuse results on many hyperparameter settings**
  - Surrogate models: predict best hyperparameter settings
  - Study hyperparameter effects/importance
- **Deeper understanding of algorithm performance:**
  - WHY models perform well (or not)
  - How is performance related to dataset properties
- **Never-ending Automatic Machine Learning**:
  - AutoML methods built on top of OpenML get increasingly better as more meta-data is added

OPENML FOR AUTOML RESEARCH

- MIT AutoML system (ICBD 2017)
  - Uses and compares against OpenML results



- Auto-sklearn (AutoML challenge winner, NIPS 2016)
  - Lookup similar datasets, start with best pipelines



Matthias Feurer et al. (2016) NIPS

- Amazon's multi-task learning AutoML (NIPS 2017)
  - Pre-trains task-specific models on OpenML results

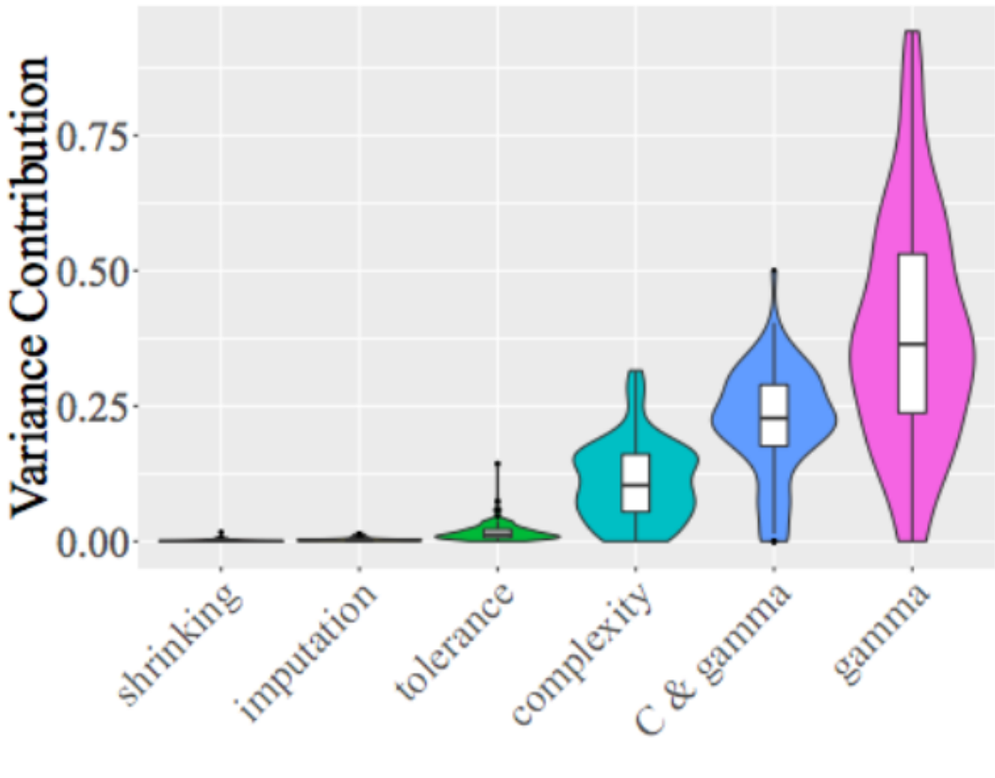
Valerio Perrone et al. (2017) MetaLearning@NIPS

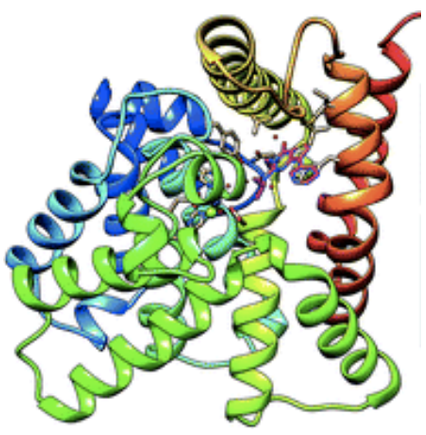
OPENML FOR ML RESEARCH

Hyperparameter importance

Use OpenML data to learn which hyperparameters to tune

Jan van Rijn et al. (2017) AutoML@ICML





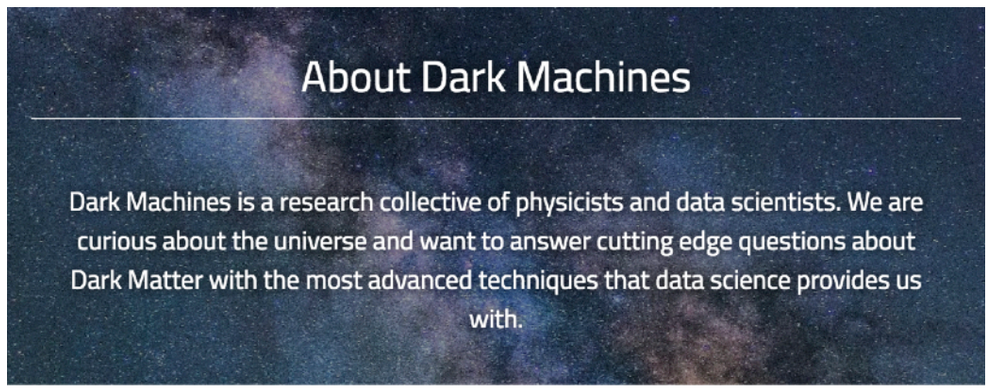
Faster drug discovery (QSAR)

- Meta-learning to build better models that recommend drug candidates for rare diseases

Olier et al. (2018) Machine Learning

Dark Machines

- Organize data and experiments in dark matter research



PIPELINE EVOLUTION

- Evolve complex pipelines with meta-learning

```
from tpot import TPOTClassifier
from sklearn.datasets import load_digits
from sklearn.model_selection import train_test_split

digits = load_digits()
X_train, X_test, y_train, y_test = train_test_split(digits.data, digits.target,
                                                    train_size=0.75, test_size=0.25)

tpot = TPOTClassifier(generations=5, population_size=50, verbosity=2, n_jobs=-1)
tpot.fit(X_train, y_train)
```

```
Optimization Progress:   0%|          | 0/300 [00:00<?, ?pipeline/s]
```

```
print(tpot.score(X_test, y_test))
```

Learning to learn

Bots that learn from all prior experiments

Automate drudge work, help people build models

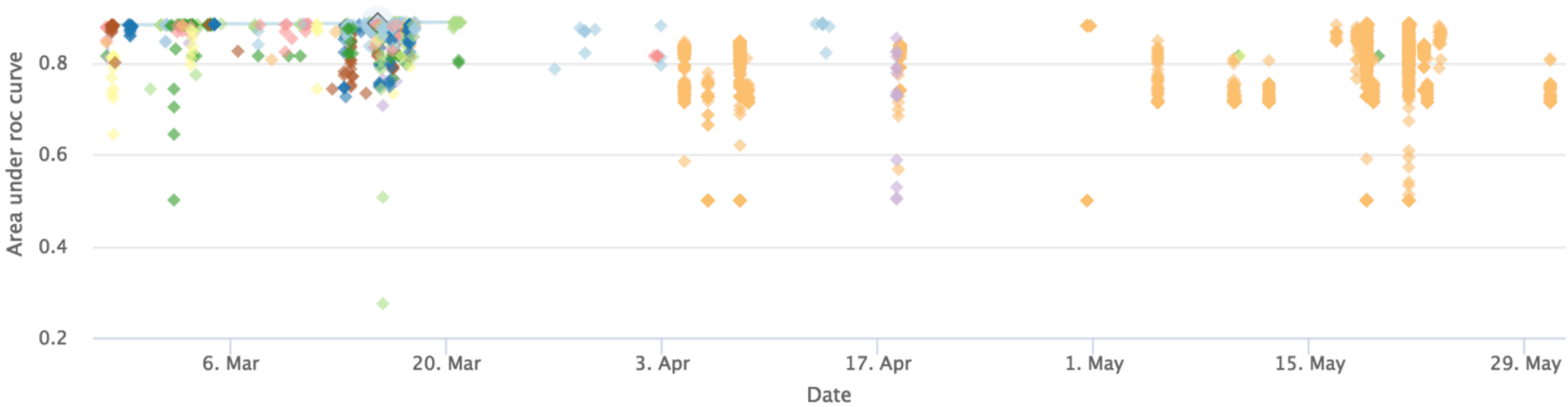
Timeline

Metric: AREA UNDER ROC CURVE

Flow: **DeepForest.DeepForest(4)**
Area under roc curve: **0.8858**
Training time (seconds): **null**
Uploader: Jeroen van Hoof
Click for more info

Contributions over time

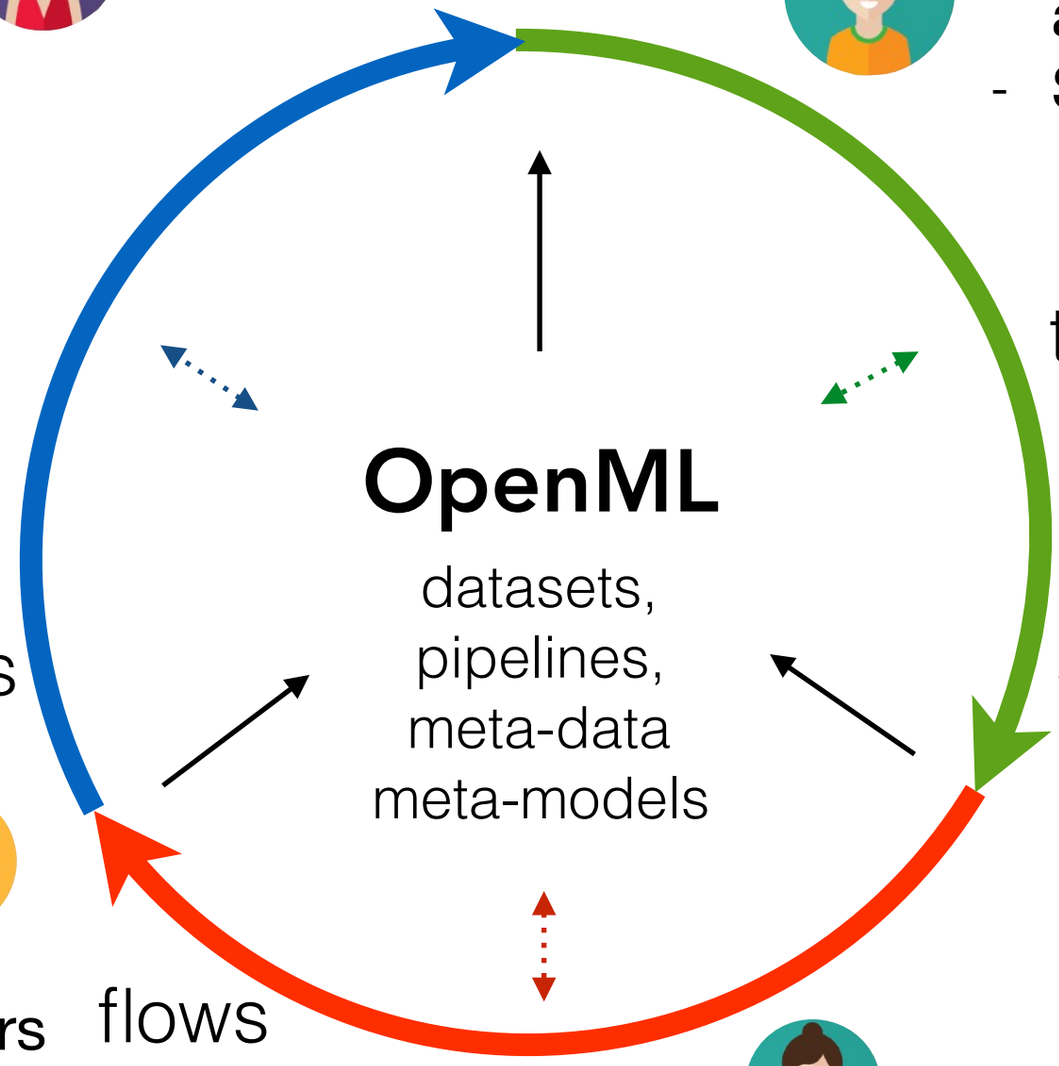
every point is a run, click for details



frontier Joaquin Vanschoren Hilde Weerts edorigatti Joel Goossens Niels Hellinga Mingpeiyu Zhang
Evertjan Peer stevens jethefer Hongliang Qiu Yezi Zhu János Szedelényi Chin-Fang Lin Wenting Xiong
M de Roode Tianyu Zhou Lirong Zhang Ruud Andriessen Stefan Majoor Angelo Majoor Changbin Lu
Irfan Nur Afif Nan Yang Niels de Jong Thomas Hagebols Stanley Clark Joost Visser Jeroen van Hoof
Xiaolei Wang Timothy Aerts Lieuwe Stooker Corbin Joosen Jos Mangnus Luis Armando Perez Rey
Jet van den Broek Thijs Ledeboer Brent van Strien Arun Tom Skariah Sako Arts Xuqiang Fang Yongyu Fan
Suraj Iyer Filip Obers Laurens Reulink Kevin van Eenige Tong Wu Jan van Rijn y q OpenML_Bot R
Raphaël Couronné Mikaël Le Bars

HUMAN-ROBOT SYMBIOSIS

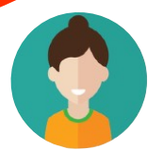
Discuss results
- explain models
- scientific interpretations
- statistical analysis
- visualize results
- collect better data
- write papers



Human expert
- Formulate the problem as an ML problem
- Suggest solutions

AI-assist
- Show similar problems
- Auto-encoding
- Auto-balancing
- Feature importance
- Detect outliers, missing values,…
- Recommend algorithms
- Predict training time

AI-assist (tuning)
- Build pipelines
- Optimize hyperparameters
- Compare results online



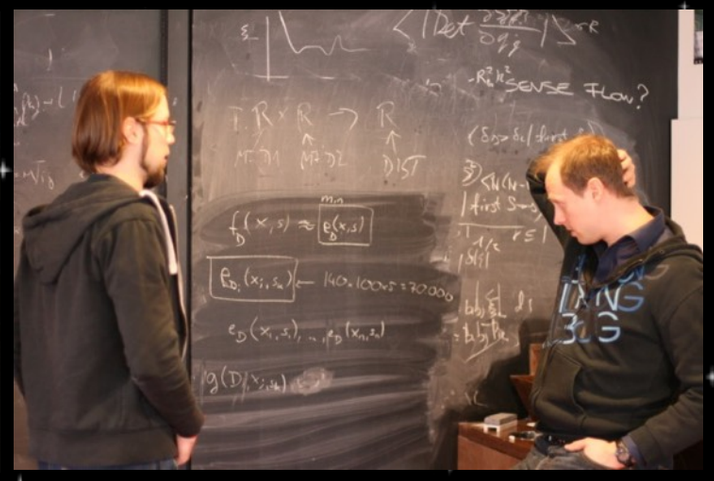
Build models
- Preprocess data
- Write code

Join us! (and change the world)

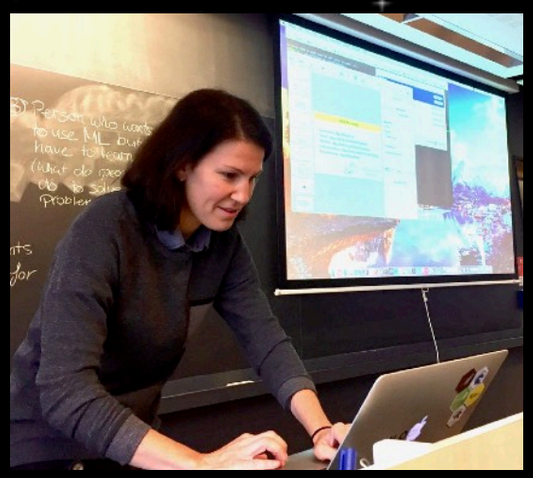
Active open source community
We need more bright people

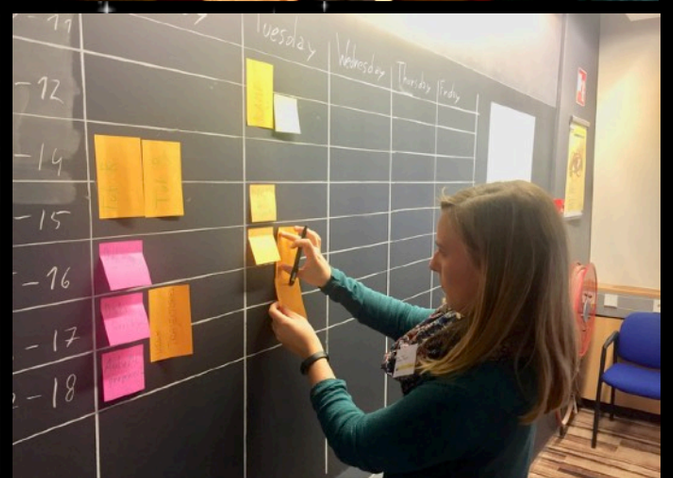
- ML/DB experts
- Developers
- UX












@open_ml

OpenML

Thank You

All support is welcome!