

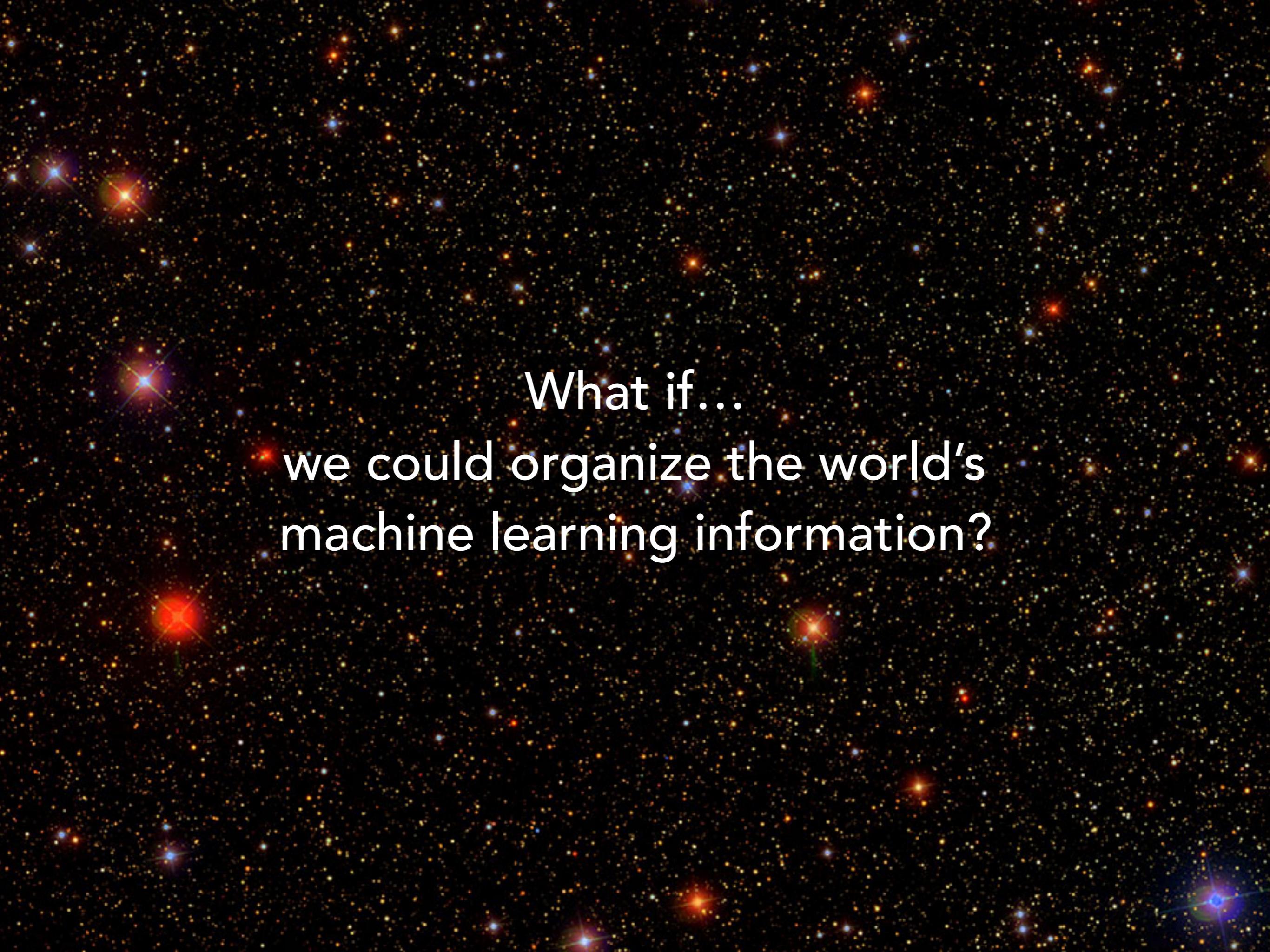


machine learning, better, together

Joaquin Vanschoren (TU/e) & the OpenML team



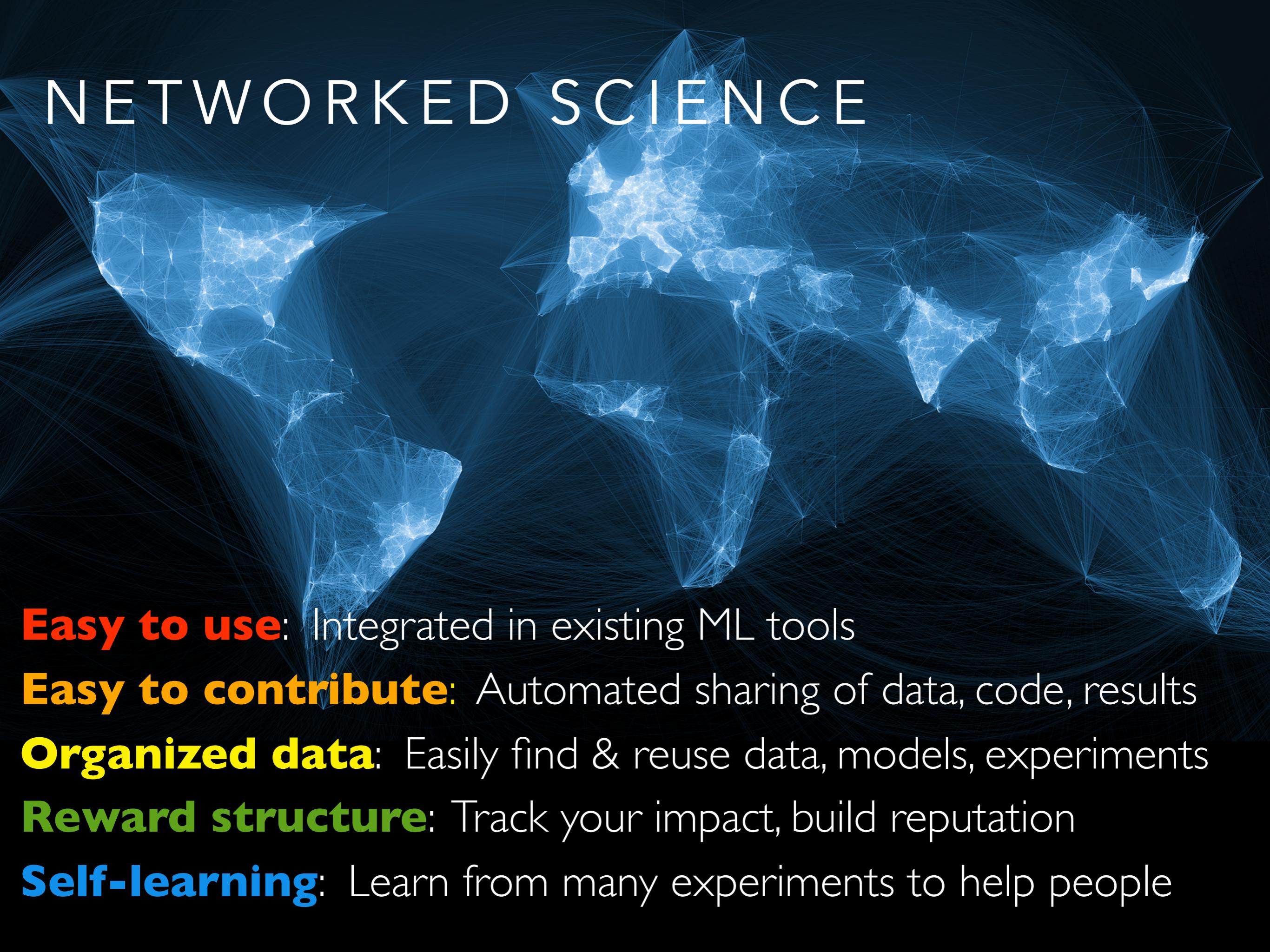
What if...
we could map the entire universe?



What if...
we could organize the world's
machine learning information?

What if...
we could organize all machine learning
datasets, experiments, and models?

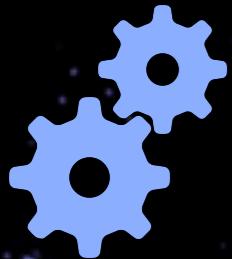
NETWORKED SCIENCE

- 
- Easy to use:** Integrated in existing ML tools
 - Easy to contribute:** Automated sharing of data, code, results
 - Organized data:** Easily find & reuse data, models, experiments
 - Reward structure:** Track your impact, build reputation
 - Self-learning:** Learn from many experiments to help people

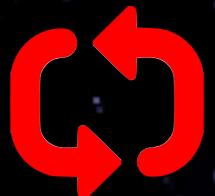
Enhance research by removing friction



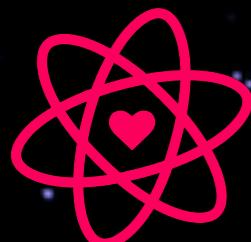
Searchable, uniformly formatted, semantically annotated datasets



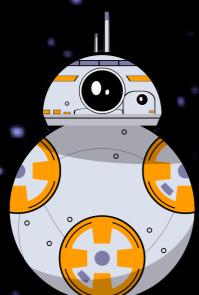
Uniform algorithm/pipeline descriptions



Better reproducibility by auto-logging all experiment details



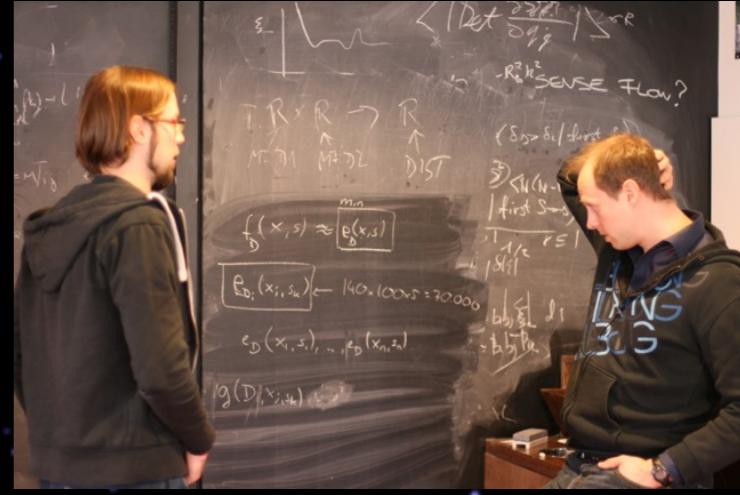
Organized, searchable, reusable experiments



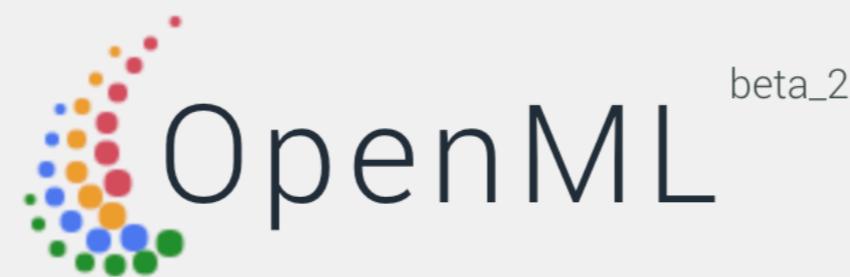
Automate machine learning and experimentation through clean APIs

Building an open source team

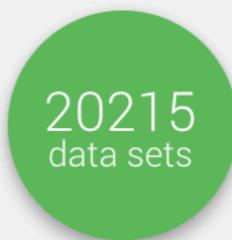
20+ active developers
Hackathons + GitHub
Open to existing projects
Academia + industry



www.openml.org



Machine learning, better, together



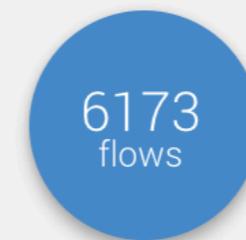
20215
data sets

Find or add **data** to analyse



68210
tasks

Download or create
scientific **tasks**



6173
flows

Find or add data analysis
flows



9494287
runs

Upload and explore all
results online.



HACKATHON

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

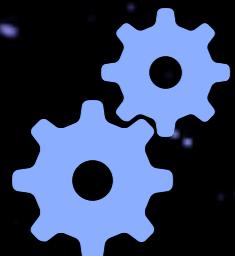
OpenML: Components



Datasets: Auto-annotated, organized, well-formatted
Easily find and load datasets + meta-data



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



Flows: Uniform description of ML pipelines
Run locally (or wherever), auto-upload globally



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

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)

The screenshot shows the OpenML web interface. At the top, there is a navigation bar with a menu icon, the text "OpenML", a search icon, and a user profile icon. A dropdown menu is open from the user profile icon, containing three items: "New data" (with a green cylinder icon), "New task" (with an orange trophy icon), and "New study" (with a purple flask icon). Below the navigation bar, the main content area has a title "Add data" with a cylinder icon. It includes fields for "Data files" (with a "Browse..." button) and "Or URL (not both)". At the bottom, there is a table header with columns "Name" and "Version".



Add data

Data files

Browse...

Or URL (not both)



New data



New task



New study

Name

Version

Description

Short description (can still be edited online). Use #tags to label it. Describe where the data originates from, and whether it was processed in any way.

Data format ([see specifications](#))

ARFF

Author(s)

Licence - [Learn more](#)

Publicly available

Mark a work that is free of known copyright restrictions. [More info](#)

Citation requests

How to reference this data in future work (e.g., publication, DOI).

Data (tabula

enced (URL)

It starts with data



Data (tabular) easily uploaded or referenced (URL)

```
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()
```



auto-versioned, analysed, organised online

OpenML

Search



19587 results

FILTERS

SORT: MOST RUNS ▾

ID'S

TABLE

+ ADD NEW

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



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

	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
	wilt (1)	High-resolution Remote Sensing data set (Quickbird). Small number of training ... ★ 326436 runs ❤ 0 likes 📁 32 downloads 32 reach 13 impact 4839 instances - 6 features - 2 classes - 0 missing values
	steel-plates-fault (1)	A dataset of steel plates' faults, classified into 7 different types. The goal was t... ★ 234209 runs ❤ 1 likes 📁 23 downloads 24 reach 13 impact 1941 instances - 34 features - 2 classes - 0 missing values
	qsar-biodeg (1)	QSAR biodegradation Data Set * Abstract: Data set containing values for 41 att... ★ 215860 runs ❤ 1 likes 📁 12 downloads 13 reach 13 impact 1055 instances - 42 features - 2 classes - 0 missing values
	Australian (3)	This dataset was retrieved 2014-11-14 from the libSVM site. It was normalized ... ★ 188737 runs ❤ 0 likes 📁 11 downloads 11 reach 4 impact 690 instances - 15 features - 2 classes - 0 missing values
	wdbc (1)	Current dataset was adapted to ARFF format from the UCI version. Sample cod... ★ 186776 runs ❤ 1 likes 📁 28 downloads 29 reach 13 impact 569 instances - 31 features - 2 classes - 0 missing values
	kr-vs-kp (1)	1. Title: Chess End-Game – King+Rook versus King+Pawn on a7 (usually abbrevi... ★ 180569 runs ❤ 0 likes 📁 24 downloads 24 reach 12 impact 3196 instances - 37 features - 2 classes - 0 missing values
	kc2 (1)	One of the NASA Metrics Data Program defect data sets. Data from software f...



auto-versioned, analysed, organised online

OpenML

Search



cardiotocography



V. 1 ▾

active

ARFF

Publicly available

Visibility: public

Uploaded 21-05-2015 by Rafael G. Mantovani

Edit

0 likes

downloaded by 21 people, 31 total downloads

0 issues

0 downvotes

21 reach

45 impact

OpenML100

study_14

study_34

study_7

+ Add tag

Help us complete this description →

Edit

Author: J. P. Marques de Sá, J. Bernardes, D. Ayers de Campos.

Source: UCI

Please cite: Ayres de Campos et al. (2000) SisPorto 2.0 A Program for Automated Analysis of Cardiotocograms. J Matern Fetal Med 5:311-318, UCI

Author: J. P. Marques de Sá, J. Bernardes, D. Ayers de Campos.

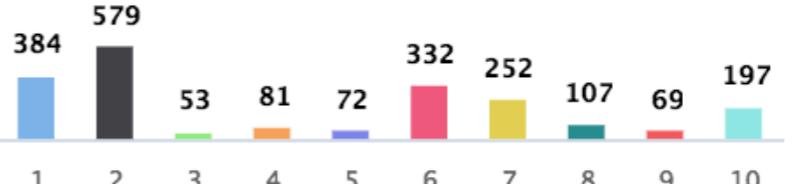
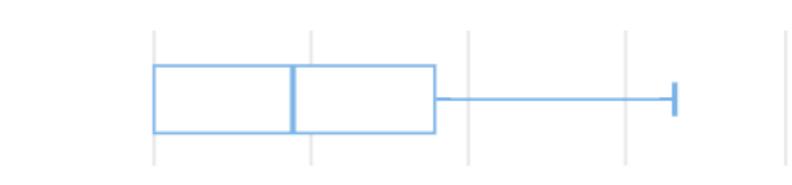
Source: UCI

Please cite: Ayres de Campos et al. (2000) SisPorto 2.0 A Program for Automated Analysis of Cardiotocograms. *J Matern Fetal Med* 5:311-318, [UCI](#)

2126 fetal cardiotocograms (CTGs) were automatically processed and the respective diagnostic features measured. The CTGs were also classified by three expert obstetricians and a consensus classification label assigned to each of them. Classification was both with respect to a morphologic pattern (A, B, C, ...) and to a fetal state (N, S, P). Therefore the dataset can be used either for 10-class or 3-

▼ Show all

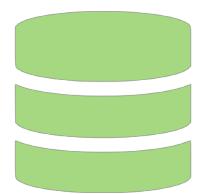
36 features

Class (target)	nominal	10 unique values 0 missing	
V1	numeric	48 unique values 0 missing	
V2	numeric	979 unique values 0 missing	

▼ Show all 36 features

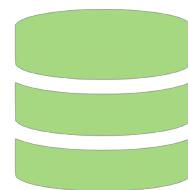


auto-versioned, analysed, organised online



Search (API)

Python, R, Java, C#



Search (API)

Python, R, Java, C#

```
import openml as om1
openml_list = om1.datasets.list_datasets()
```



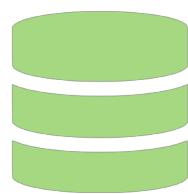
Search (API)

Python, R, Java, C#

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

Found 2522 datasets

	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

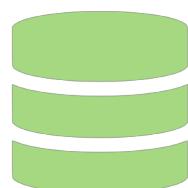


Search (API)

Python, R, Java, C#

Found 11 datasets

	did	name	NumberOfInstances	NumberOfFeatures	NumberOfClasses
6	6	letter	20000	17	26
32	32	pendigits	10992	17	10
216	216	elevators	16599	19	61
846	846	elevators	16599	19	2
977	977	letter	20000	17	2
1019	1019	pendigits	10992	17	2
1120	1120	MagicTelescope	19020	12	2
1199	1199	BNG(echoMonths)	17496	10	17491
1222	1222	letter-challenge-unlabeled.arff	20000	17	3



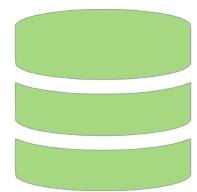
Search (API)

Python, R, Java, C#

```
openml_list = openml.datasets.list_datasets(  
    number_instances = '10000..20000',  
    number_features = '10..20')
```

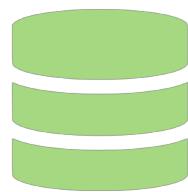
Found 11 datasets

	did	name	NumberOfInstances	NumberOfFeatures	NumberOfClasses
6	6	letter	20000	17	26
32	32	pendigits	10992	17	10
216	216	elevators	16599	19	61
846	846	elevators	16599	19	2
977	977	letter	20000	17	2
1019	1019	pendigits	10992	17	2
1120	1120	MagicTelescope	19020	12	2
1199	1199	BNG(echoMonths)	17496	10	17491
1222	1222	letter-challenge-unlabeled.arff	20000	17	3



Search (API)

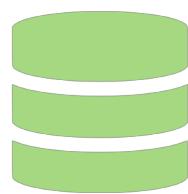
Python, R, Java, C#



Search (API)

Python, R, Java, C#

```
openml_list = omл.datasets.list_datasets(  
    data_name = 'eeg-eye-state')
```

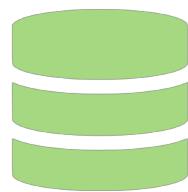


Search (API)

Python, R, Java, C#

```
openml_list = oml.datasets.list_datasets(  
    data_name = 'eeg-eye-state')
```

did	name	NumberOfInstances	NumberOfFeatures	NumberOfClasses
1471	1471 eeg-eye-state	14980.0	15.0	2.0

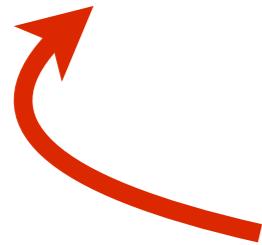


Search (API)

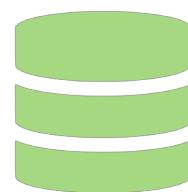
Python, R, Java, C#

```
openml_list = oml.datasets.list_datasets(  
    data_name = 'eeg-eye-state')
```

did	name	NumberOfInstances	NumberOfFeatures	NumberOfClasses
1471	1471 eeg-eye-state	14980.0	15.0	2.0

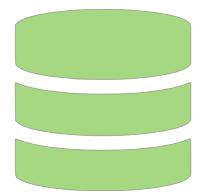


data id



Get (API)

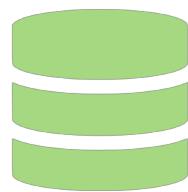
Python, R, Java, C#



Get (API)

Python, R, Java, C#

```
dataset = oml.datasets.get_dataset(1471)  
dataset.description[:500]
```



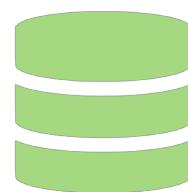
Get (API)

Python, R, Java, C#

```
dataset = oml.datasets.get_dataset(1471)
dataset.description[:500]
```

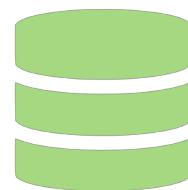
****Author**:** Oliver Roesler, it12148'@'lehre.dhbw-stuttgart.de
****Source**:** [UCI] (<https://archive.ics.uci.edu/ml/datasets/EEG+ERSITY>)
University (DHBW), Stuttgart, Germany
****Please cite**:**

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



Get (API)

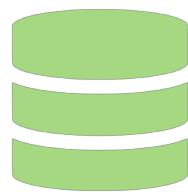
Python, R, Java, C#



Get (API)

Python, R, Java, C#

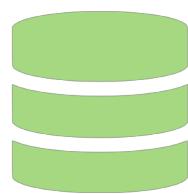
```
x, y, attribute_names = dataset.get_data(  
    return_attribute_names=True)
```



Get (API)

Python, R, Java, C#

```
x, y, attribute_names = dataset.get_data(  
    return_attribute_names=True)  
eeg = pd.DataFrame(x, columns=attribute_names)  
eeg['class'] = y
```



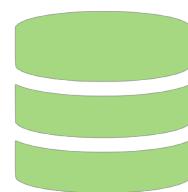
Get (API)

Python, R, Java, C#

```
x, y, attribute_names = dataset.get_data(  
    return_attribute_names=True)  
eeg = pd.DataFrame(x, columns=attribute_names)  
eeg['class'] = y
```

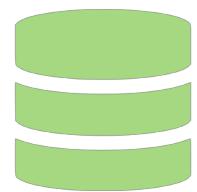
	v1	v2	v3	v4	v5	\
0	4329.229980	4009.229980	4289.229980	4148.209961	4350.259766	
1	4324.620117	4004.620117	4293.850098	4148.720215	4342.049805	
2	4327.689941	4006.669922	4295.379883	4156.410156	4336.919922	
3	4328.720215	4011.790039	4296.410156	4155.899902	4343.589844	
4	4326.149902	4011.790039	4292.310059	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	

	v6	v7	v8	v9	v10	\
0	4586.149902	4096.919922	4641.029785	4222.049805	4238.459961	



Get (API)

Python, R, Java, C#

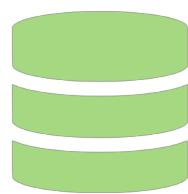


Get (API)

Python, R, Java, C#

eeg.plot()

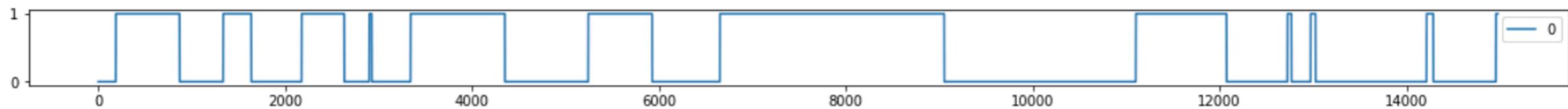
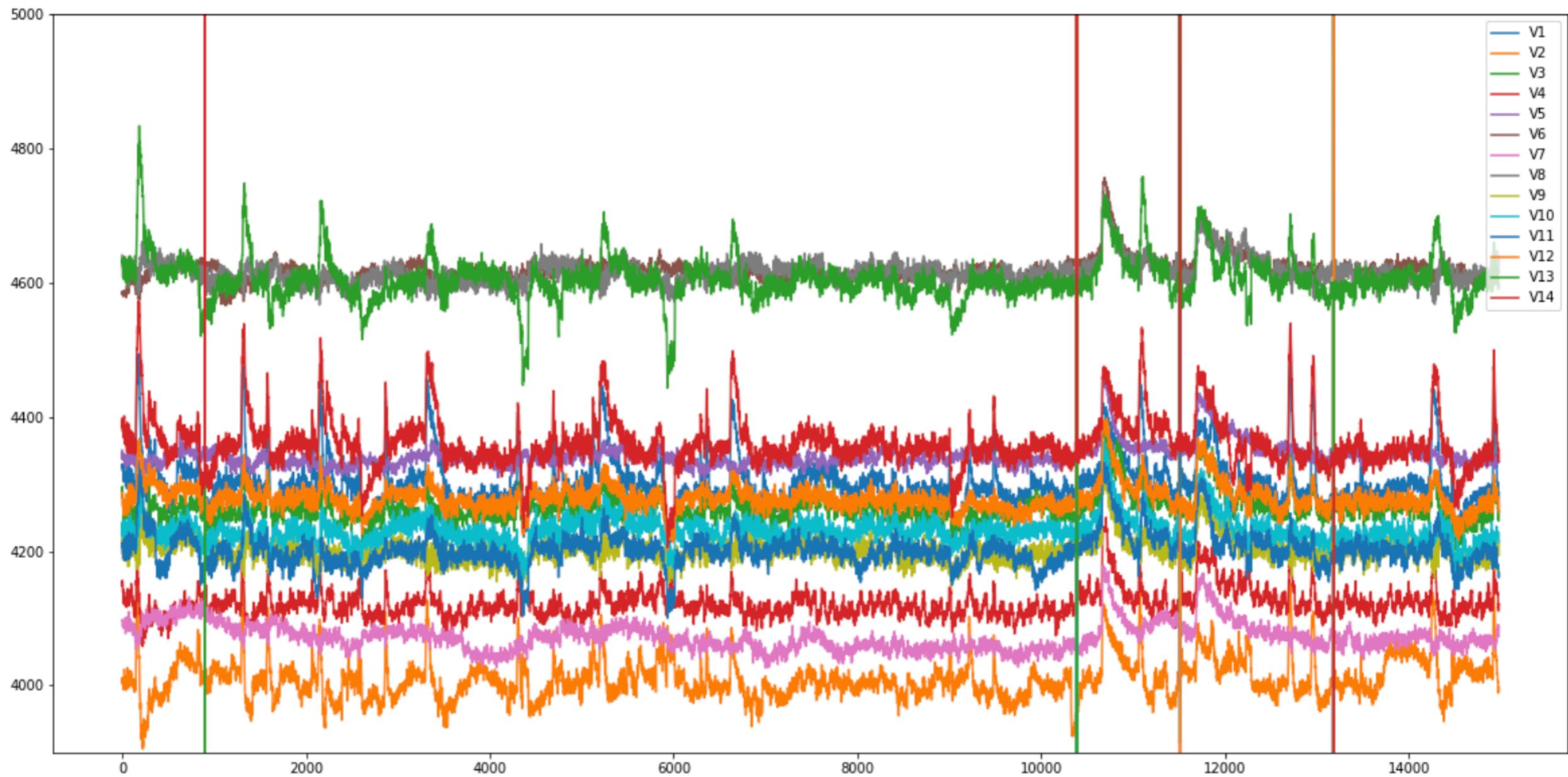
pd.DataFrame(y).plot()

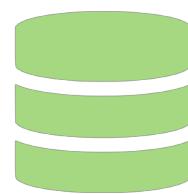


Get (API)

Python, R, Java, C#

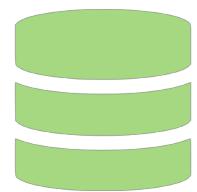
eeg.plot()
pd.DataFrame(y).plot()





Get (API)

Python, R, Java, C#



Get (API)

Python, R, Java, C#

dataset.qualities



Get (API)

Python, R, Java, C#

dataset.qualities

```
'NumberOfBinaryFeatures': 1.0,
'NumberOfClasses': 2.0,
'NumberOfFeatures': 15.0,
'NumberOfInstances': 14980.0,
'NumberOfInstancesWithMissingValues': 0.0,
'NumberOfMissingValues': 0.0,
'NumberOfNumericFeatures': 14.0,
'NumberOfSymbolicFeatures': 1.0,
'PercentageOfBinaryFeatures': 6.66666666666667,
'PercentageOfInstancesWithMissingValues': 0.0,
'PercentageOfMissingValues': 0.0,
'PercentageOfNumericFeatures': 93.3333333333333,
'PercentageOfSymbolicFeatures': 6.66666666666667,
'Quartile1AttributeEntropy': nan,
'Quartile1KurtosisOfNumericAtts': 2712.4725059647635,
'Quartile1MeansOfNumericAtts': 4193.079256508685,
'Quartile1MutualInformation': nan,
'Quartile1SkewnessOfNumericAtts': 22.47202621060797,
'Quartile1StdDevOfNumericAtts': 37.98467231888507,
```



Get (API)

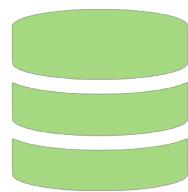
Python, R, Java, C#

dataset.qualities

```
'NumberOfBinaryFeatures': 1.0,  
'NumberOfClasses': 2.0,  
'NumberOfFeatures': 15.0,  
'NumberOfInstances': 14980.0,  
'NumberOfInstancesWithMissingValues': 0.0,  
'NumberOfMissingValues': 0.0,  
'NumberOfNumericFeatures': 14.0,  
'NumberOfSymbolicFeatures': 1.0,  
'PercentageOfBinaryFeatures': 6.66666666666667,  
'PercentageOfInstancesWithMissingValues': 0.0,  
'PercentageOfMissingValues': 0.0,  
'PercentageOfNumericFeatures': 93.3333333333333,  
'PercentageOfSymbolicFeatures': 6.66666666666667,  
'Quartile1AttributeEntropy': nan,  
'Quartile1KurtosisOfNumericAtts': 2712.4725059647635,  
'Quartile1MeansOfNumericAtts': 4193.079256508685,  
'Quartile1MutualInformation': nan,  
'Quartile1SkewnessOfNumericAtts': 22.47202621060797,  
'Quartile1StdDevOfNumericAtts': 37.98467231888507,
```



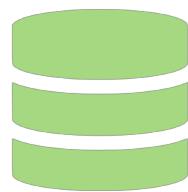
**130+ data properties
(meta-features)**



Via sklearn (0.20+)

Python, R, Java, C#

```
{"id": "40966",
"name": "MiceProtein",
'version': '4',
'format': 'ARFF',
'upload_date': '2017-11-08T16:00:15',
'licence': 'Public',
'url': 'https://openml2.win.tue.nl/data/v1/download/17928620/
MiceProtein.arff',
'file_id': '17928620',
'default_target_attribute': 'class',
'row_id_attribute': 'MouseID',
```



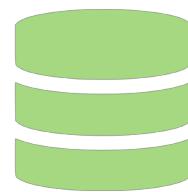
Via sklearn (0.20+)

Python, R, Java, C#

```
from sklearn.datasets import fetch_openml  
mice_data = fetch_openml(name='miceprotein',  
                           version=4)
```

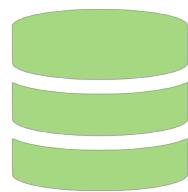
```
mice_data.details
```

```
{'id': '40966',  
 'name': 'MiceProtein',  
 'version': '4',  
 'format': 'ARFF',  
 'upload_date': '2017-11-08T16:00:15',  
 'licence': 'Public',  
 'url': 'https://openml2.win.tue.nl/data/v1/download/17928620/  
 MiceProtein.arff',  
 'file_id': '17928620',  
 'default_target_attribute': 'class',  
 'row_id_attribute': 'MouseID',
```



Fit (API)

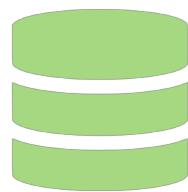
Python, R, Java, C#



Fit (API)

Python, R, Java, C#

```
from sklearn import neighbors  
clf = neighbors.KNeighborsClassifier()  
clf.fit(X, y)
```

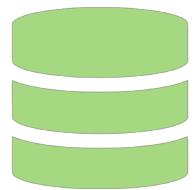


Fit (API)

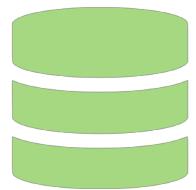
Python, R, Java, C#

```
from sklearn import neighbors
clf = neighbors.KNeighborsClassifier()
clf.fit(X, y)

KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=1, n_neighbors=1, p=2,
                     weights='uniform')
```



**Complete code to build a model,
automatically, anywhere**



Complete code to build a model, automatically, anywhere

```
import openml as oml
from sklearn import neighbors, tree

dataset = oml.datasets.get_dataset(1471)
X, y = dataset.get_data()
clf = neighbors.KNeighborsClassifier()
clf.fit(X, y)
```

Tasks

auto-benchmarking and collaboration



Tasks contain data, goals, procedures.

Auto-build + evaluate models correctly

All evaluations are directly comparable



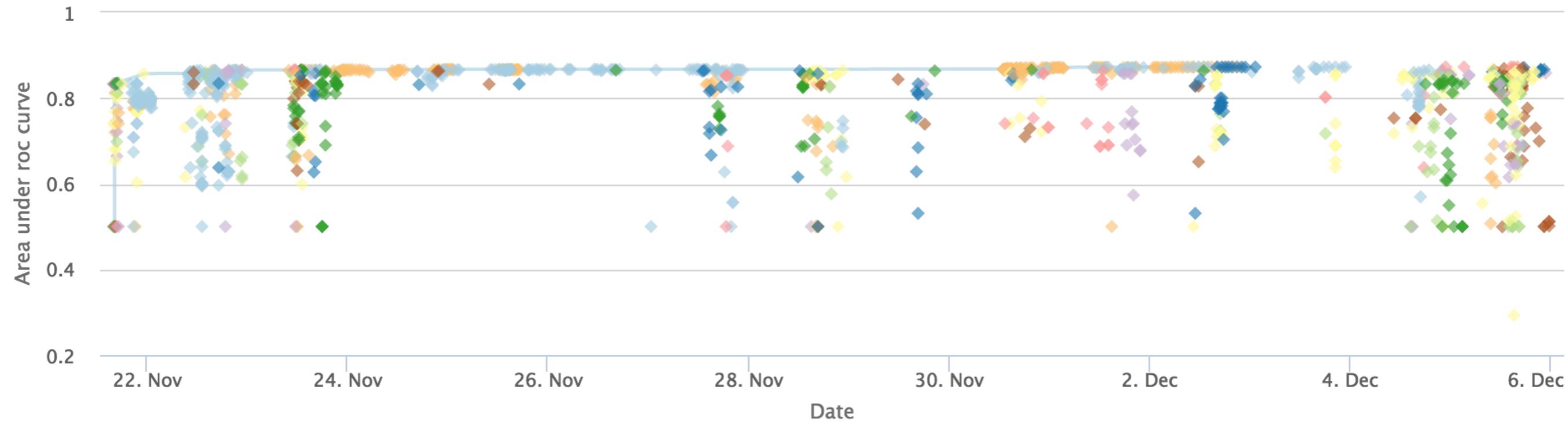
Predict target T
optimize accuracy
10-fold Xval



Collaborate in real time online



Contributions over time
every point is a run, click for details

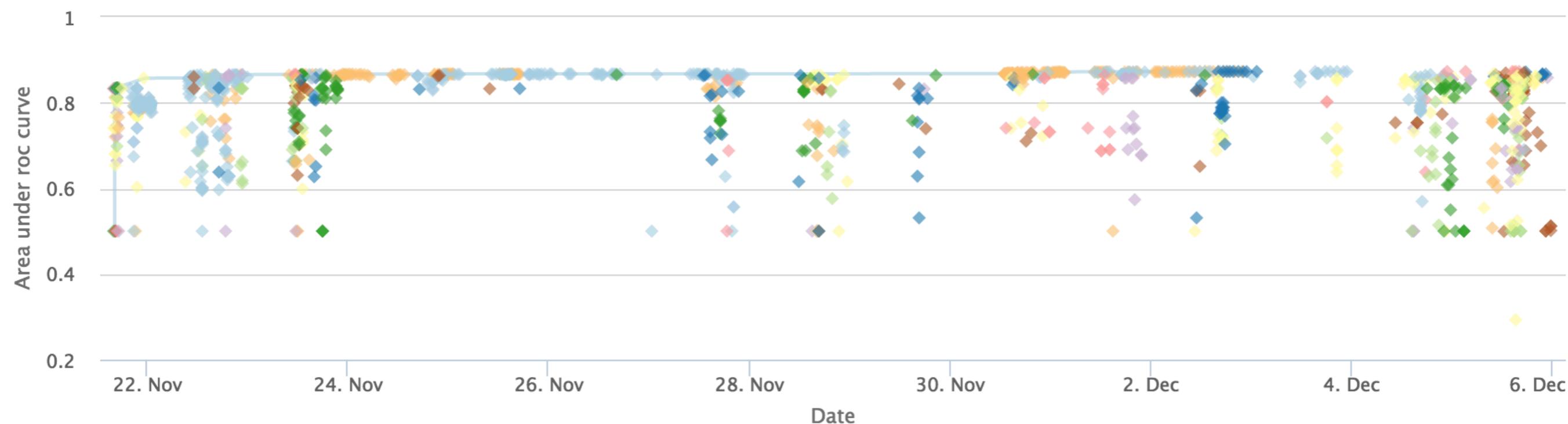




Predict target T
optimize accuracy
10-fold Xval

Contributions over time

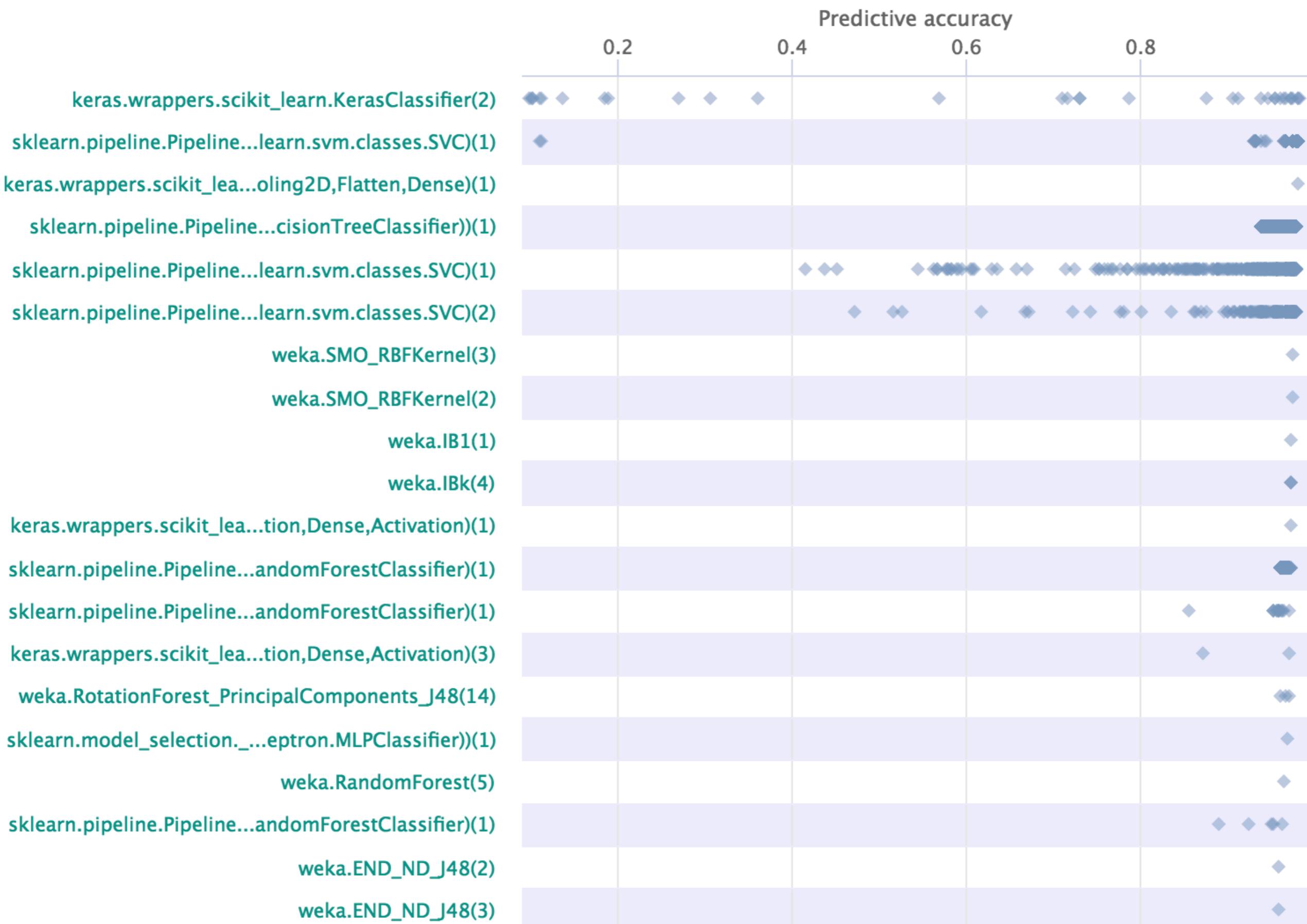
every point is a run, click for details



- frontier
- a
- r f
- Sven Arends
- Adriaan Vast
- B. Elfrink
- Johann Slabber
- Daan Luttik
- Armand Duijn
- Hilde Weerts
- Khanh Nguyen
- Kevin Chang
- van der Linde
- Luuk Godtschalk
- Marleen Hillen
- Romek Vinke
- Daan de Graaf
- Jeroen Besems
- Francisco Lozano
- Anuvab Sahoo
- Julia Hofs
- Jelte Dirks
- Wijnands
- M. van Nijnatten
- J Rensen
- Laurence Keijzer
- George Park
- Chris Jansen
- Brum Mulders
- Anne K
- Aleksandr Popov
- T Peters
- Kuijpers
- Gerson Foks
- Lujaina Abuerban
- Joaquin Vanschoren
- Sander doesnotmatter
- Leenen
- Adriaan Knapen
- Peter Gerards
- Martijn Noordhof
- Sajid Mohideen
- Mert Zararsiz
- Evertjan Peer
- Marten Kenbeek
- Yiming Lin
- Wouter Verlaek
- Pepijn Obels
- Pepijn Obels
- Michiel Verburg
- Luca Weibel
- Jeroen Donners
- Gijs Walravens
- Kagan Akbas
- Ton Matton
- Casper Siksma
- J KT
- Maarten Visscher

Evaluations per flow (multiple parameter settings)

every point is a run, click for details





Search



```
task_list = om1.tasks.list_tasks(size=5000)
```



Search

```
task_list = oml.tasks.list_tasks(size=5000)
```

First 5 of 5000 tasks:

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

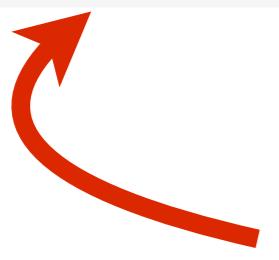


Search

```
task_list = om1.tasks.list_tasks(size=5000)
```

First 5 of 5000 tasks:

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



task id



Search



Search

```
mytasks = pd.DataFrame.from_dict(task_list)
mytasks.query('name=="eeg-eye-state"')
```



Search

```
mytasks = pd.DataFrame.from_dict(task_list)
mytasks.query('name=="eeg-eye-state"')
```

	tid	did	name	task_type	estimation_procedure	evaluation_measures
9983	9983	1471	eeg-eye-state	Supervised Classification	10-fold Crossvalidation	predictive_accuracy
14951	14951	1471	eeg-eye-state	Supervised Classification	10-fold Crossvalidation	NaN



Search

```
mytasks = pd.DataFrame.from_dict(task_list)  
mytasks.query('name=="eeg-eye-state"')
```

	tid	did	name	task_type	estimation_procedure	evaluation_measures
9983	9983	1471	eeg-eye-state	Supervised Classification	10-fold Crossvalidation	predictive_accuracy
14951	14951	1471	eeg-eye-state	Supervised Classification	10-fold Crossvalidation	NaN



task id





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



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

```
{'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
```

Flows

Run experiments *locally*, share them *globally*

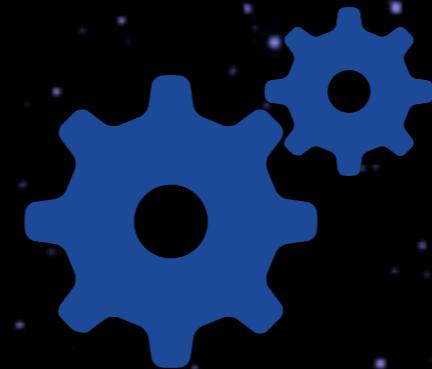


Auto-run algorithms/workflows on any task

Integrated in many machine learning tools (+ APIs)



dmlc
XGBoost



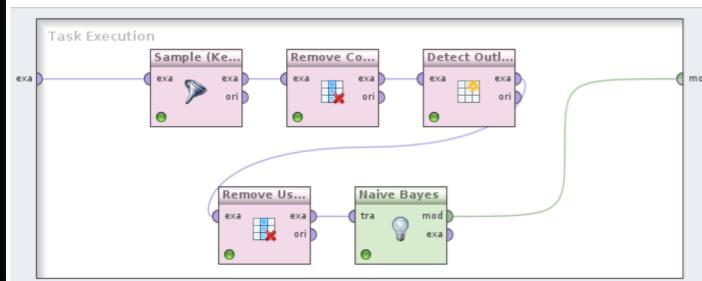
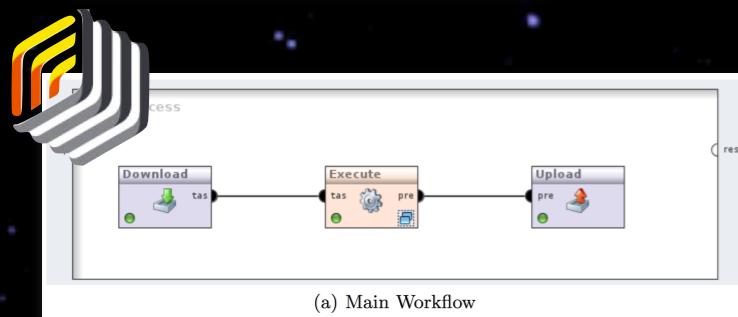
Integrated in many machine learning tools (+ APIs)

A screenshot of the OpenML web interface. It shows a sidebar with a logo of a bird in a blue circle, followed by fields for 'Experiment Type' (set to 'OpenML Task'), 'Number of folds' (set to 10), and classification status. Below this is a 'Tasks' section with buttons for 'Add...', 'Edit...', 'Delete...', and 'Us...'. A list of tasks is shown: 'Task 1: anneal - Supervised Classification', 'Task 2: anneal.ORIG - Supervised Classification', 'Task 3: kr-vs-kp - Supervised Classification', 'Task 4: labor - Supervised Classification', and 'Task 5: arrhythmia - Supervised Classification'. At the bottom are 'Up' and 'Down' buttons.



```
import openml as oml  
from sklearn import tree
```

```
task = oml.tasks.get_task(14951)  
clf = tree.ExtraTreeClassifier()  
flow = oml.flows.sklearn_to_flow(clf)  
run = oml.runs.run_flow_on_task(task, flow)  
myrun = run.publish()
```



```
library(OpenML)
```

```
library(mlr)
```

```
task = getOMLTask(10)
```

```
lrn = makeLearner("classif.rpart")
```

```
res = runTaskMlr(task, lrn)
```

```
run.id = uploadOMLRun(res)
```



Fit and share (*complete code*)

```
import openml as oml
from sklearn import tree

task = oml.tasks.get_task(14951)
clf = tree.ExtraTreeClassifier()
flow = oml.flows.sklearn_to_flow(clf)
run = oml.runs.run_flow_on_task(task, flow)
myrun = run.publish()
```



Fit and share (*complete code*)

```
import openml as oml
from sklearn import tree

task = oml.tasks.get_task(14951)
clf = tree.ExtraTreeClassifier()
flow = oml.flows.sklearn_to_flow(clf)
run = oml.runs.run_flow_on_task(task, flow)
myrun = run.publish()
```

Uploaded to <http://www.openml.org/r/9204488>



Fit and share pipelines

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



Fit and share pipelines

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

Uploaded to <http://www.openml.org/r/7943199>



Fit and share deep learning models

```
import keras
from keras.models import Sequential,
from keras.layers import Dense, Dropout, Flatten, Conv2D, MaxPooling
from keras.layers.core import Activation

model = Sequential()
model.add(Reshape((28, 28, 1), input_shape=(784,)))
model.add(Conv2D(20, (5, 5), padding="same", input_shape=(28,28,1),
activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
model.add(Conv2D(50, (5, 5), padding="same", activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
model.add(Flatten())
model.add(Dense(500))
model.add(Activation('relu'))
model.add(Dense(10))
model.add(Activation('softmax'))
model.compile(loss=keras.losses.categorical_crossentropy,
optimizer=keras.optimizers.Adadelta(),
metrics=['accuracy'])
```



Fit and share *deep learning models*

```
task = tasks.get_task(3573) #MNIST
flow = oml.flows.keras_to_flow(model)
run = oml.runs.run_flow_on_task(task, flow)
myrun = run.publish()
```



Fit and share *deep learning models*

```
task = tasks.get_task(3573) #MNIST
flow = oml.flows.keras_to_flow(model)
run = oml.runs.run_flow_on_task(task, flow)
myrun = run.publish()
```

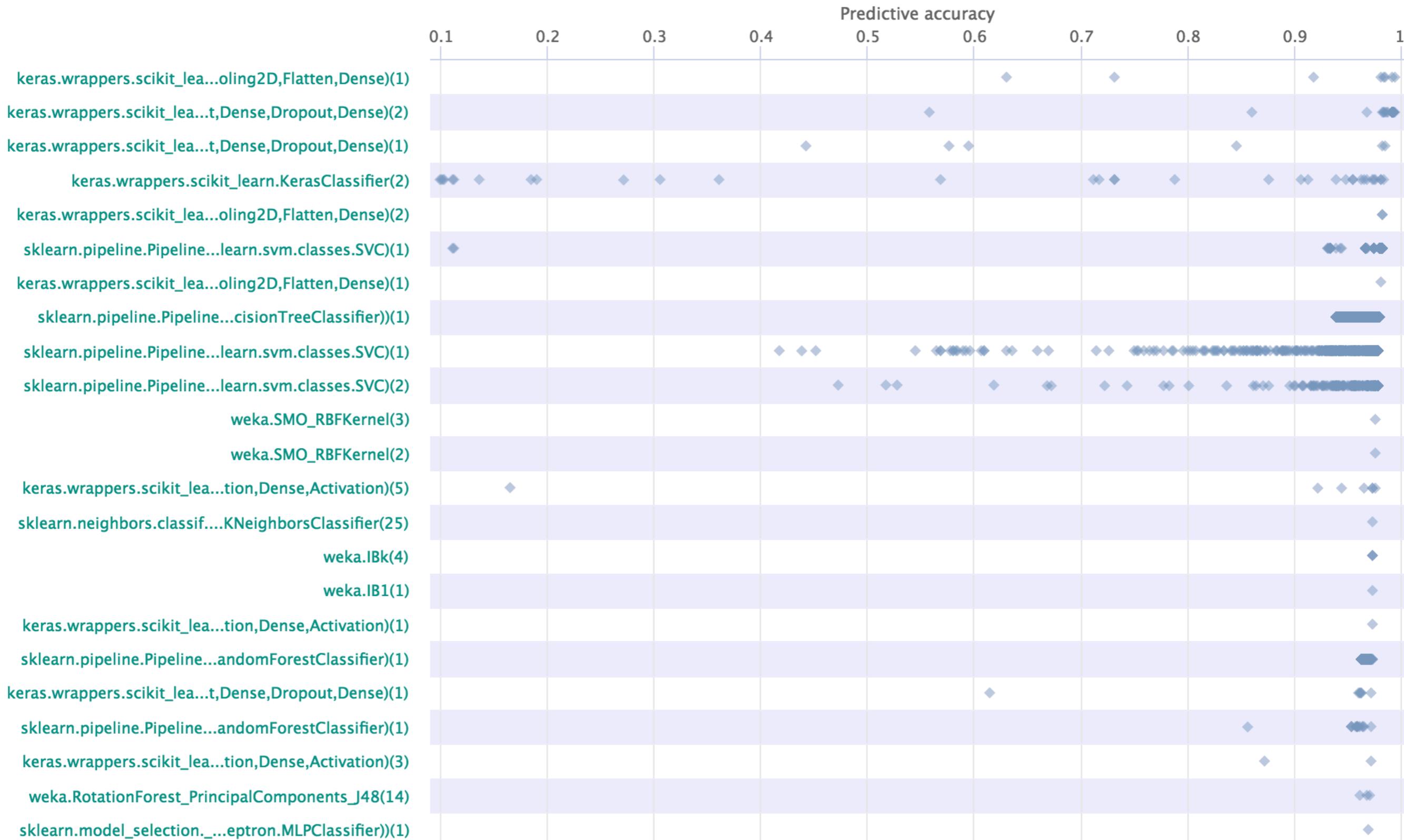
Uploaded to <https://www.openml.org/r/9204337>



Compare to state-of-the-art

Evaluations per flow (multiple parameter settings)

every point is a run, click for details



Runs

Share and reuse results



**Experiments auto-uploaded, evaluated online
reproducible, linked to data, flows, authors
and all other experiments**

Runs

Share and reuse results



**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.

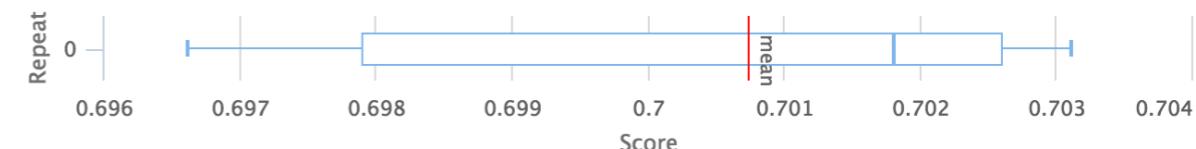
Area under ROC curve

0.7007 \pm 0.0023

Per class

0	1
0.7007	0.7007

Cross-validation details (10-fold Crossvalidation)





Download, reuse runs

★ Download, reuse runs

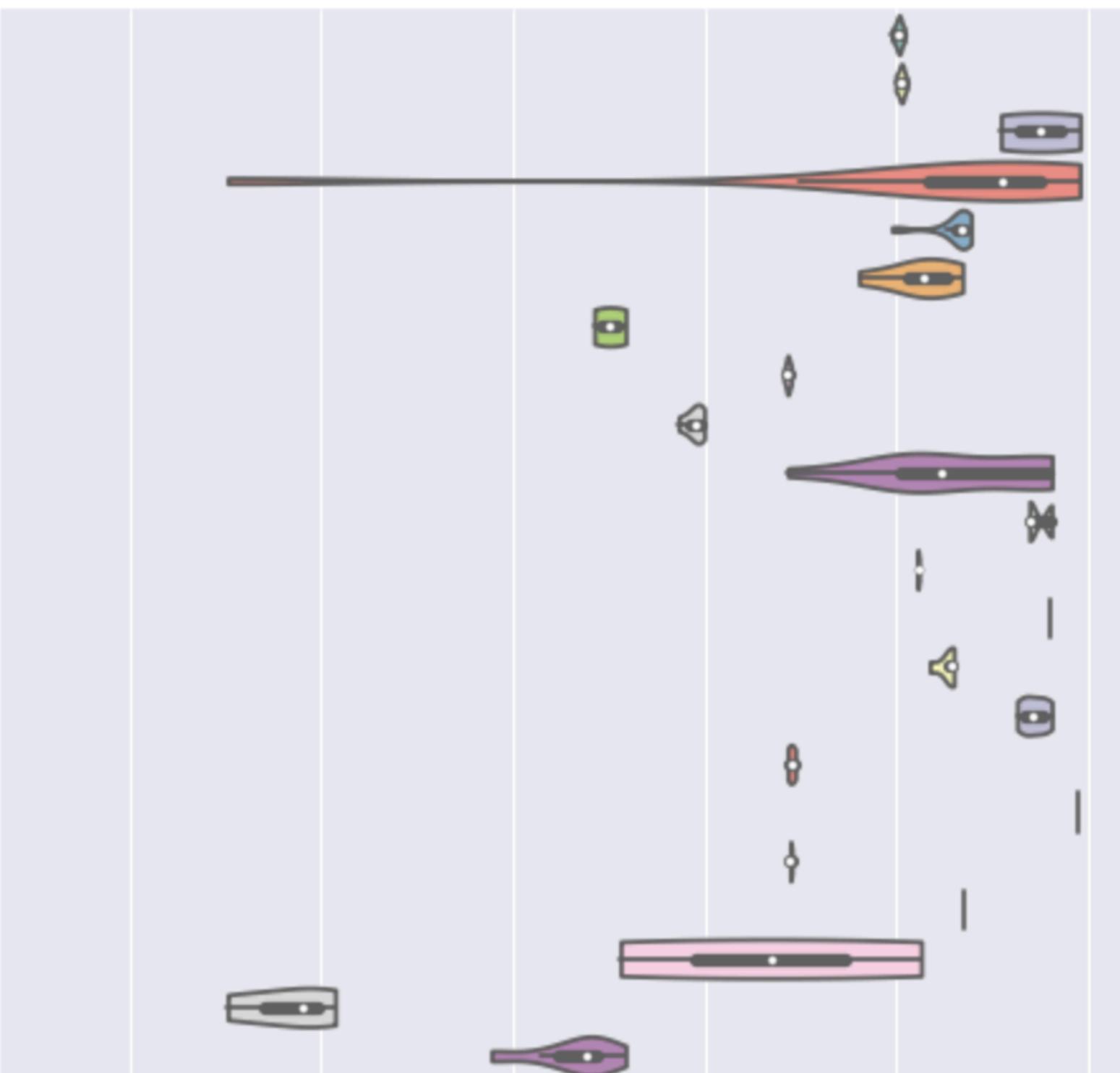
```
myruns = oml.runs.list_runs(task=[14951])  
sns.violinplot(x="score", y="flow", data=pd.DataFrame(
```



Download, reuse runs

```
myruns = oml.runs.list_runs(task=[14951])  
sns.violinplot(x="score", y="flow", data=pd.DataFrame(
```

```
sklearn.ensemble.forest.RandomForestClassifier(13)  
sklearn.ensemble.forest.RandomForestClassifier(14)  
    sklearn.ensemble.forest.ExtraTreesClassifier(3)  
    sklearn.ensemble.forest.ExtraTreesClassifier(1)  
    sklearn.ensemble.forest.RandomForestClassifier(8)  
    sklearn.ensemble.forest.RandomForestClassifier(10)  
sklearn.ensemble.weight_boosting.AdaBoostClassifier(2)  
    sklearn.tree.tree.DecisionTreeClassifier(2)  
        sklearn.tree.tree.ExtraTreeClassifier(2)  
    sklearn.ensemble.forest.RandomForestClassifier(11)  
sklearn.neighbors.classification.KNeighborsClassifier(7)  
    sklearn.ensemble.forest.ExtraTreesClassifier(7)  
sklearn.neighbors.classification.KNeighborsClassifier(12)  
    sklearn.ensemble.bagging.BaggingClassifier(1)  
    sklearn.ensemble.bagging.BaggingClassifier(2)  
        sklearn.tree.tree.DecisionTreeClassifier(1)  
    sklearn.ensemble.forest.ExtraTreesClassifier(4)  
        sklearn.tree.tree.DecisionTreeClassifier(6)  
    sklearn.ensemble.forest.RandomForestClassifier(16)  
sklearn.ensemble.gradient_boosting.GradientBoostingClassifier(1)  
    sklearn.svm.classes.SVC(1)  
sklearn.ensemble.weight_boosting.AdaBoostClassifier(1)
```





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

website called OpenML. It hosts not only algorithms, but also data sets and more than 8 million experimental runs with all their attendant details. "The exact way that you run your experiments is full of undocumented assumptions and decisions," Vanschoren says. "A lot of this detail never makes it into papers."

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 Reach Impact Uploads
 1649.5 12 195 1 35 0 1605

[EDIT PROFILE](#)

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

Activity: 1.65K

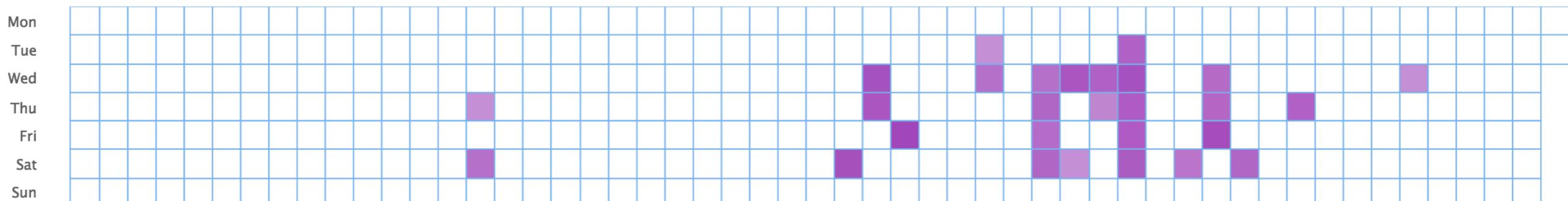
1.64K

0

17

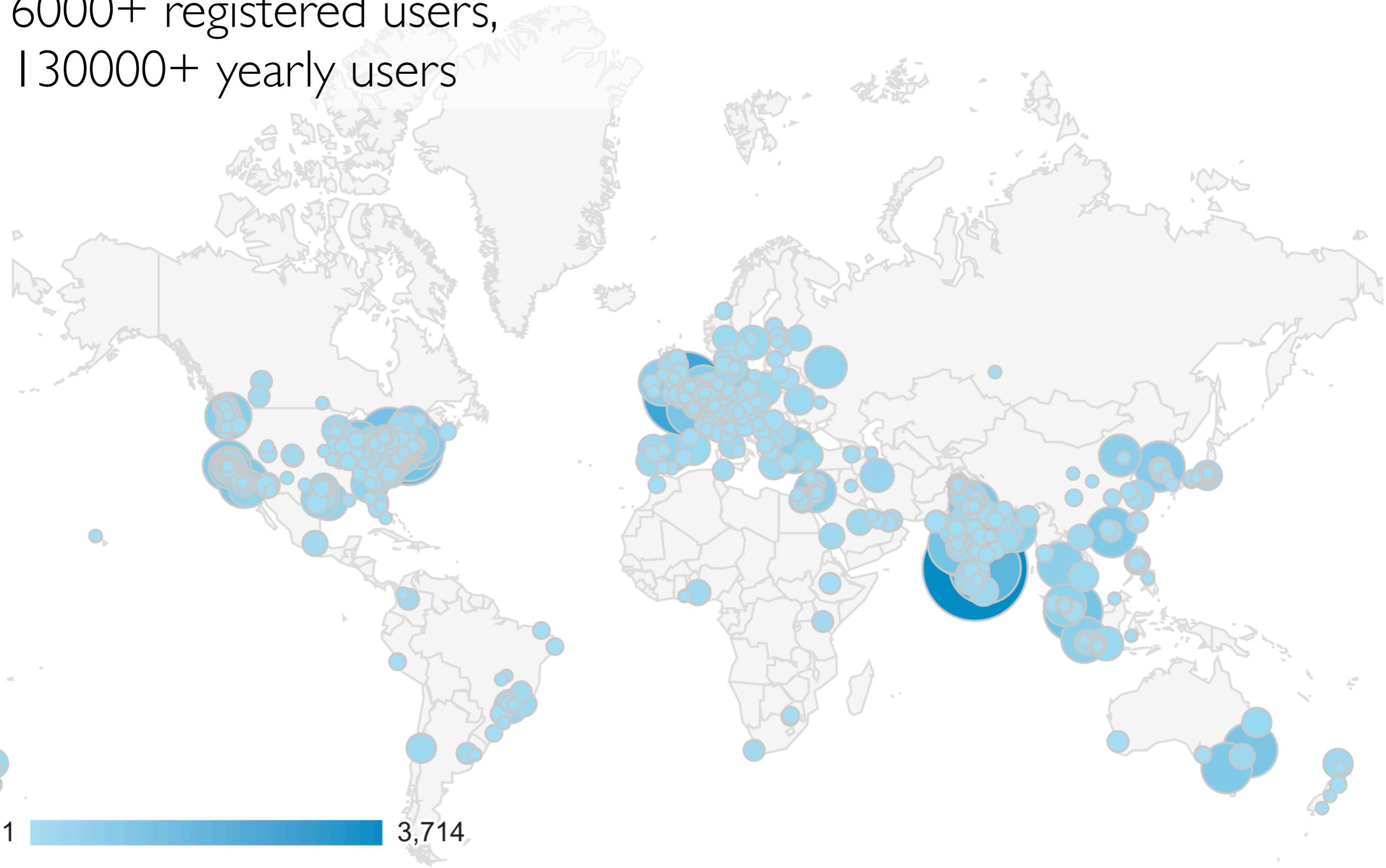
-

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



OpenML Community

6000+ registered users,
13000+ yearly users



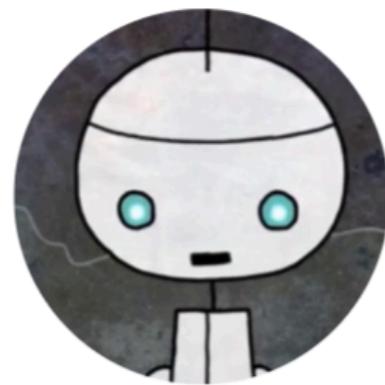
AUTOMATING MACHINE LEARNING

- **Find similar datasets**
 - See what works well on similar datasets
- **Reuse (millions of) prior model evaluations:**
 - Learn how to learn: choose/create algorithms for new tasks
- **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)
- **Never-ending Automatic Machine Learning:**
 - AutoML methods built on top of OpenML get increasingly better as more meta-data is added

Learning to learn

AutoML bots learn from models shared by humans (on OpenML)

Humans learn from models built by humans and bots (AI assist)



OpenML_Bot R

Can I help? You can find me here: <https://github.com/ja-thomas/OMLbots>

Joined 2017-03-07

Uploads
0 ⚙️ 40 🏆 0 ★ 5692477

models built

by humans

models built

by AutoML bots

Data Sets

0

0

Flows

40

33

1

2

Tasks

0

0

0

Runs

5692477

8

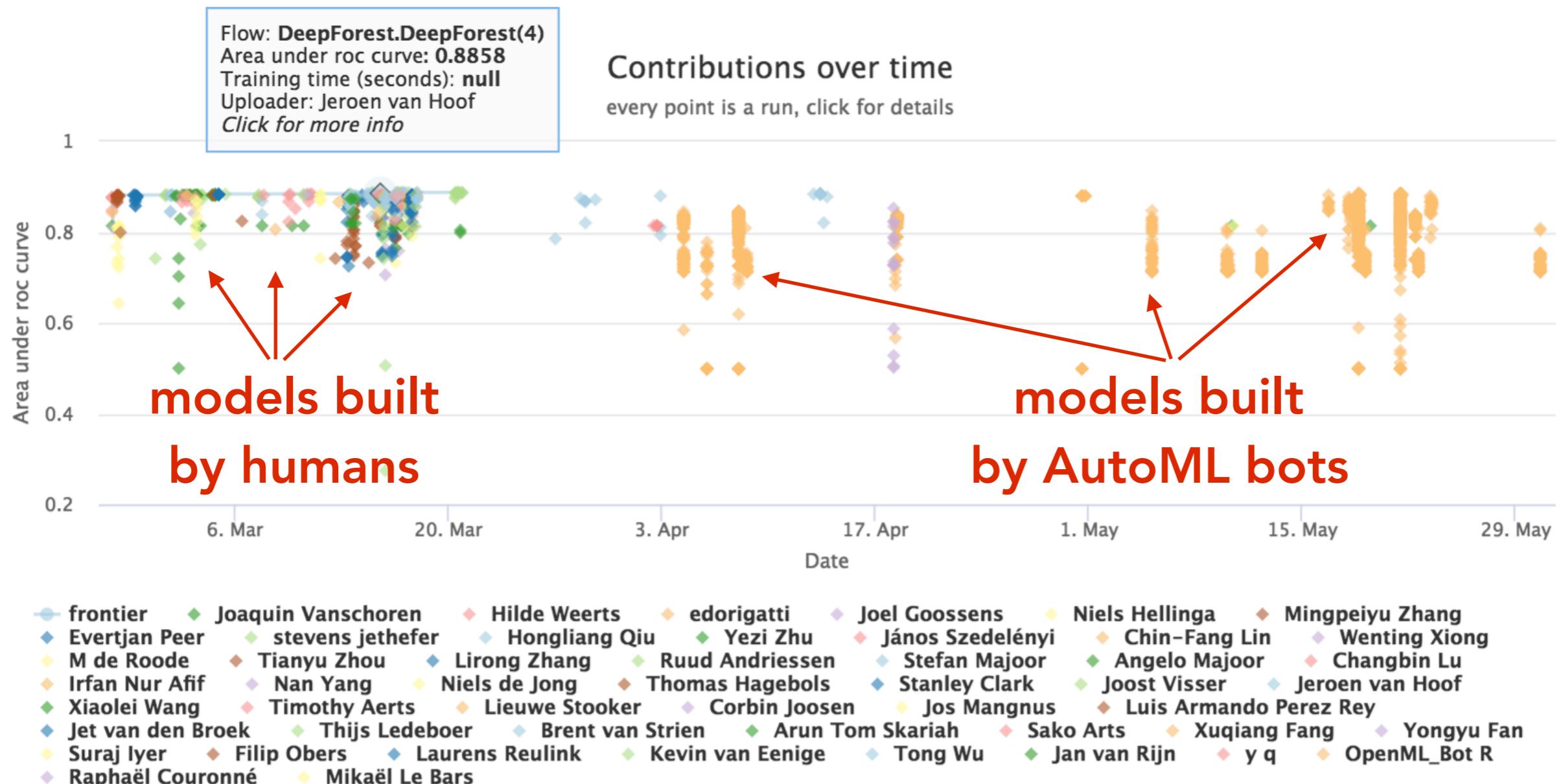
1

Learning to learn

AutoML bots learn from models shared by humans (on OpenML)
Humans learn from models built by humans and bots (AI assist)

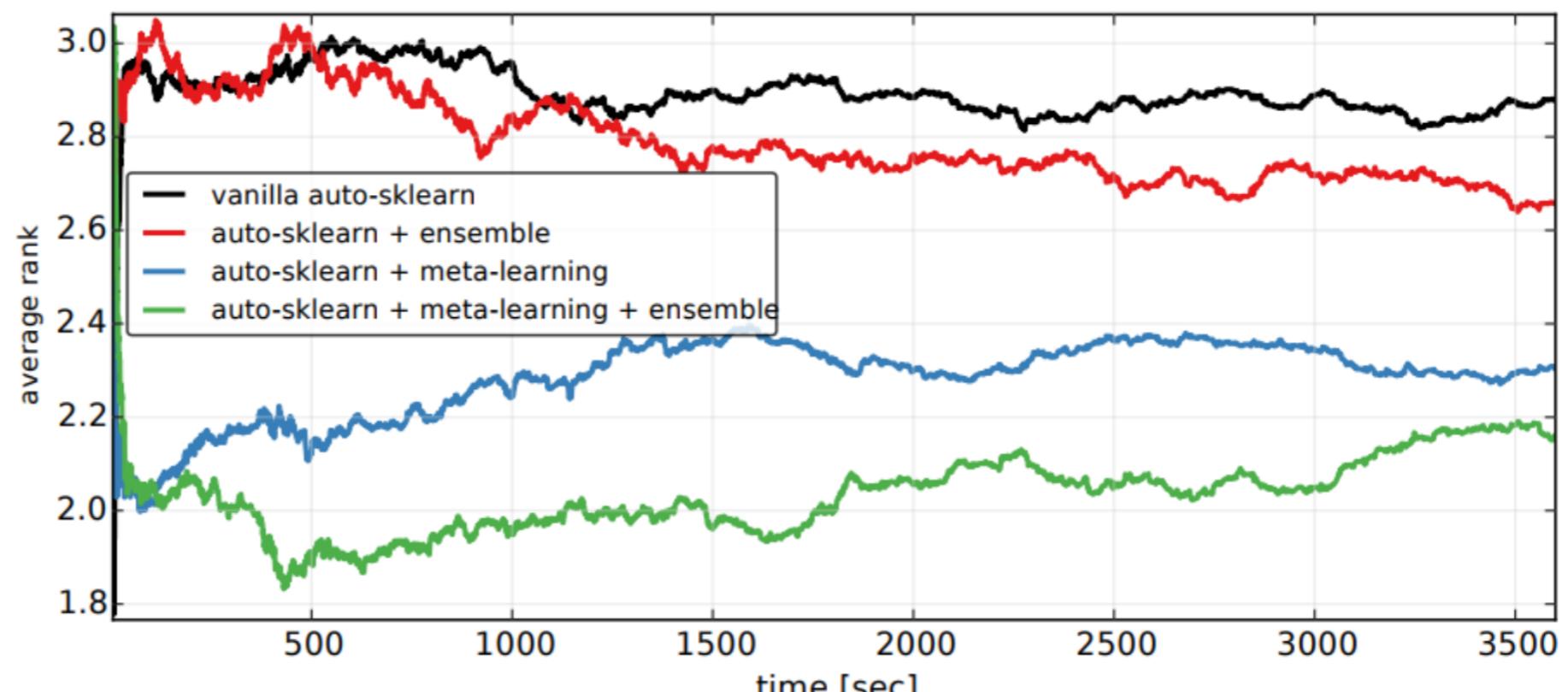
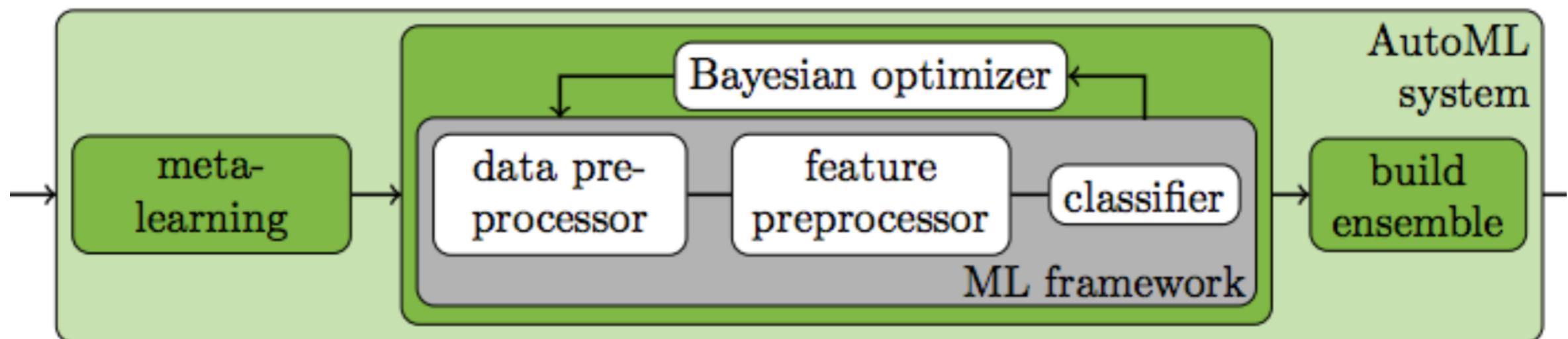
Timeline

Metric: AREA UNDER ROC CURVE



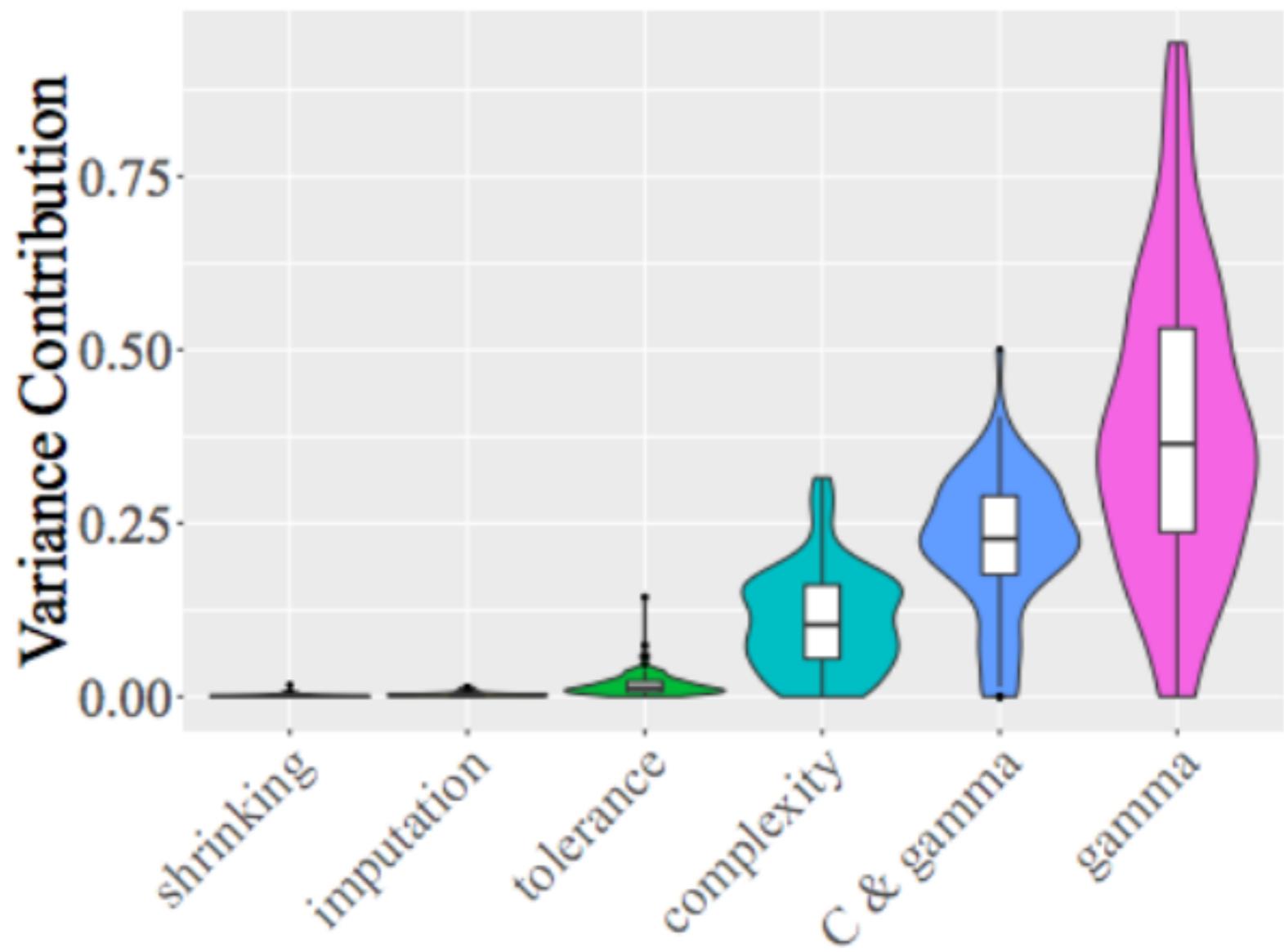
EXAMPLES

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



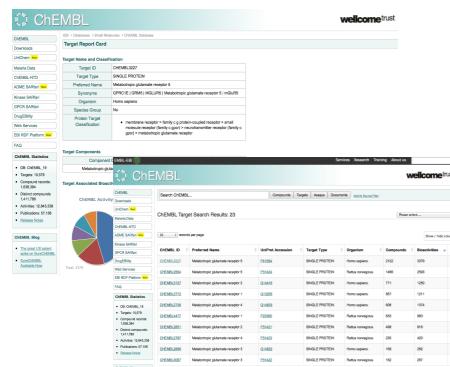
EXAMPLES

- **Hyperparameter space design**
 - Use OpenML data to learn which hyperparameters to tune



EXAMPLES

- **Faster drug discovery (QSAR)**
 - Meta-learning to build better models that recommend drug candidates for rare diseases

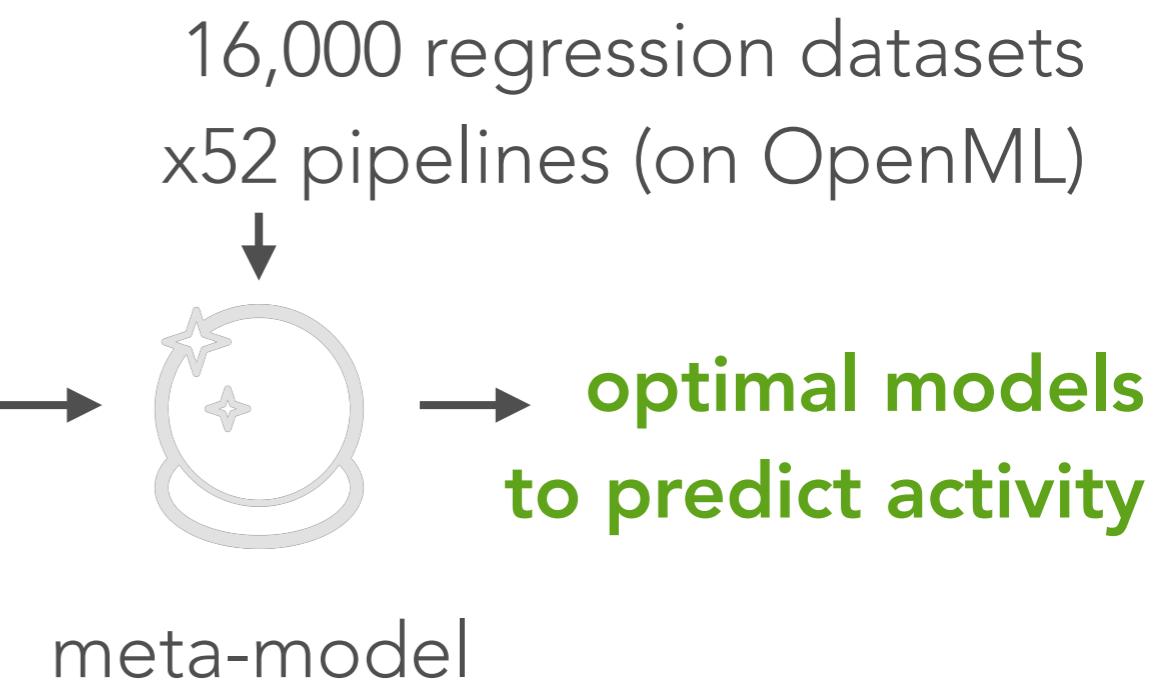


Molecule
representations

MW	LogP	TPSA	b1	b2	b3	b4	b5	b6	b7	b8	b9
377.435	3.883	77.85	1	1	0	0	0	0	0	0	0
341.361	3.411	74.73	1	1	0	1	0	0	0	0	0
197.188	-2.089	103.78	1	1	0	1	0	0	0	1	0
346.813	4.705	50.70	1	0	0	1	0	0	0	0	0
...											
:											

ChEMBL DB: 1.4M compounds,
10k proteins, 12.8M activities

all data on
new protein



Thanks to the entire OpenML team



Jan
van Rijn



Guiseppe
Casalicchio



Mattias
Feurer



Bernd
Bischl



Andreas
Mueller



Heidi
Seibold



Bilge
Celik



Pieter
Gijsbers



Andrey
Ustyuzhanin



William
Raynaut



Erin
LeDell



Tobias
Glasmachers



Jakub
Smid

**And many
more!**

Thanks to the entire OpenML team



Jan
van Rijn



Guiseppe
Casalicchio



Mattias
Feurer



Bernd
Bischl



Andreas
Mueller



Heidi
Seibold



Bilge
Celik



Pieter
Gijsbers



Andrey
Ustyuzhanin



William
Raynaut



Erin
LeDell



Tobias
Glasmachers



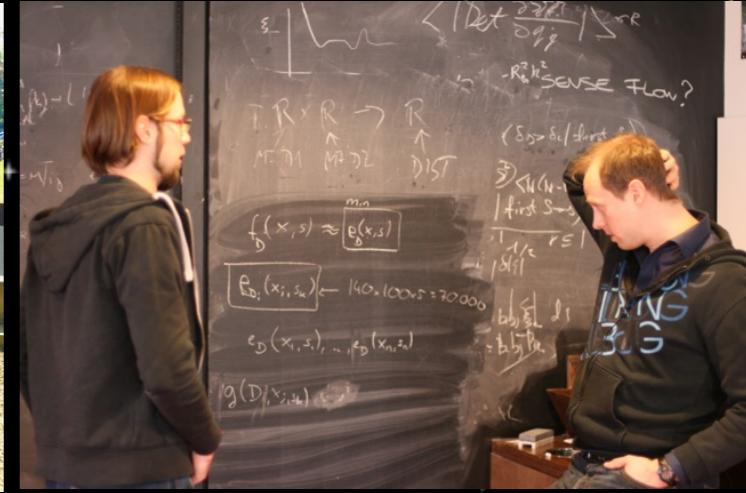
Jakub
Smid

**And many
more!**

Join us! (and change the world)

Active open source community
We need more bright people

- ML/DB experts
- Developers
- UX

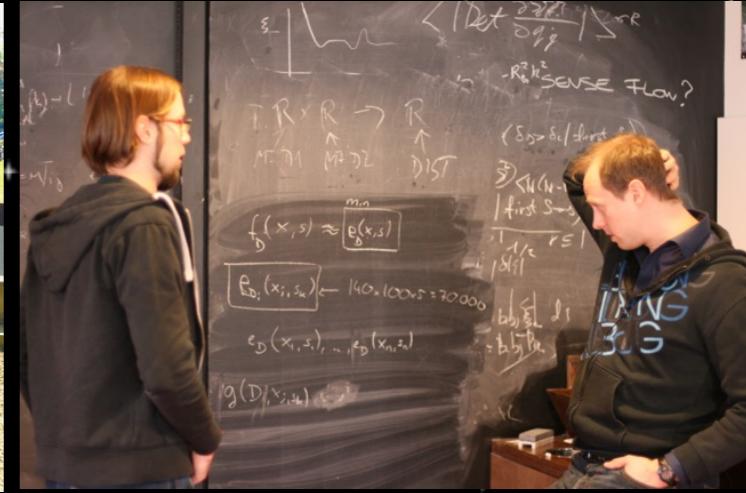


Join us! (and change the world)

Active open source community

We need more bright people

- ML/DB experts
- Developers
- UX

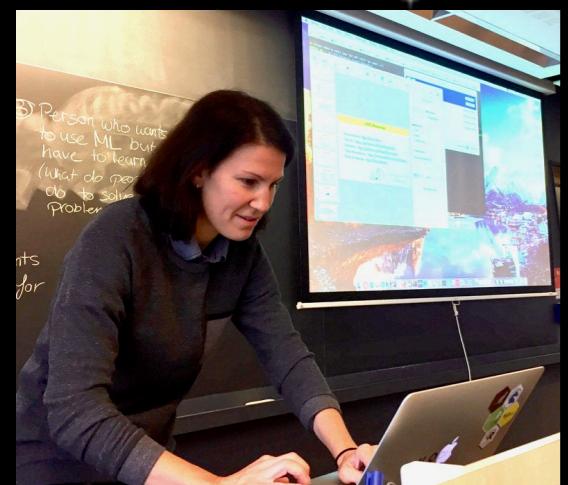
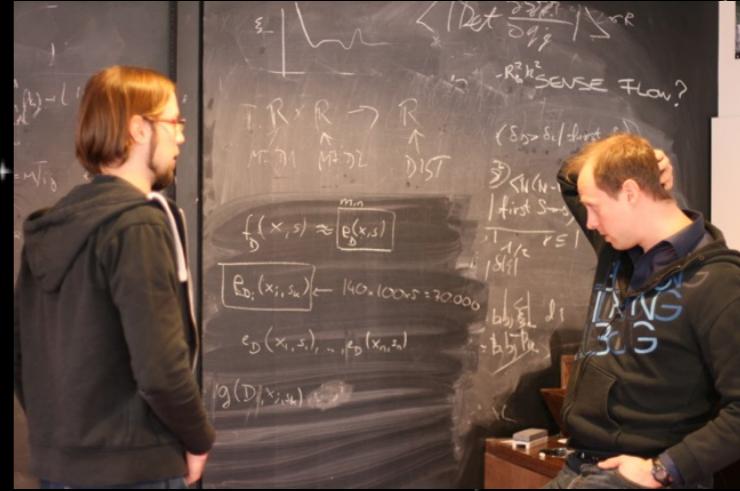


Support is welcome!

Workshop sponsorship (hackathons 2x/year)

Donations: OpenML foundation

Compute time
Project ideas

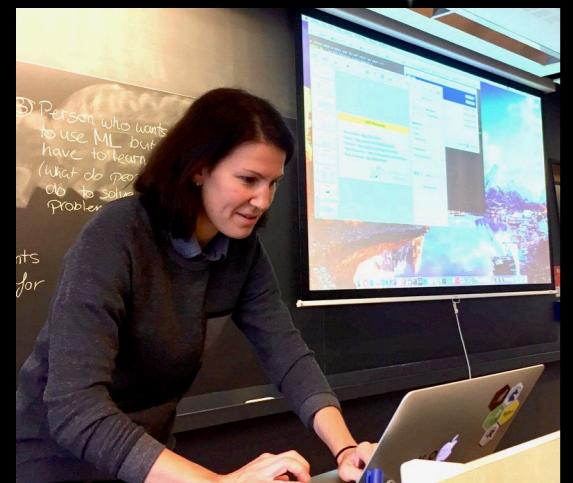
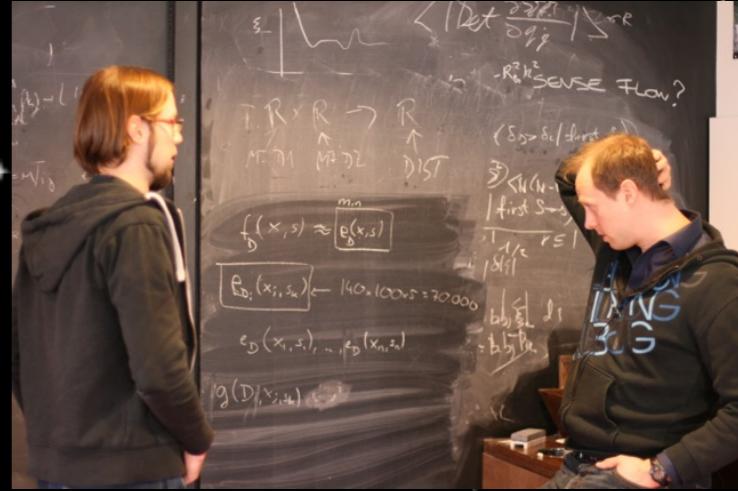


Support is welcome!

Workshop sponsorship (hackathons 2x/year)

Donations: OpenML foundation

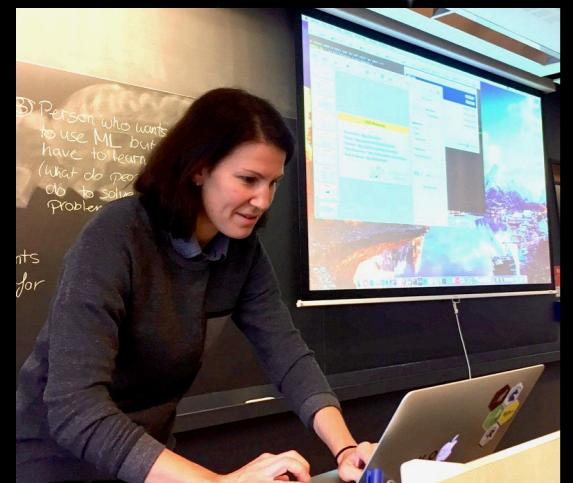
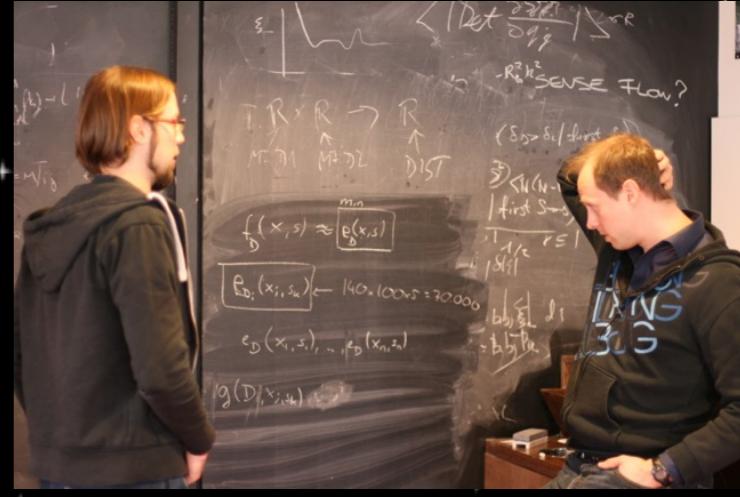
Compute time
Project ideas





Thank you!

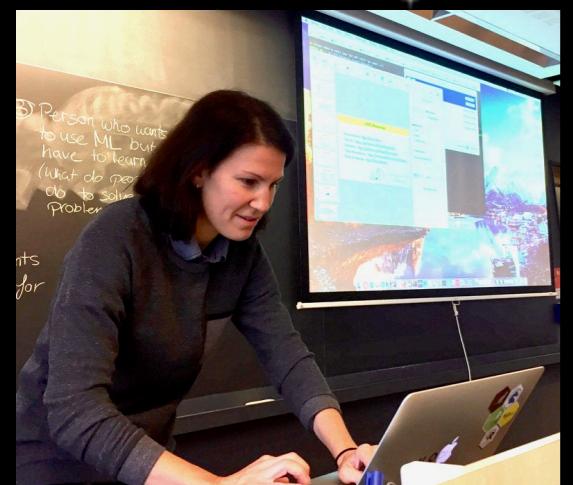
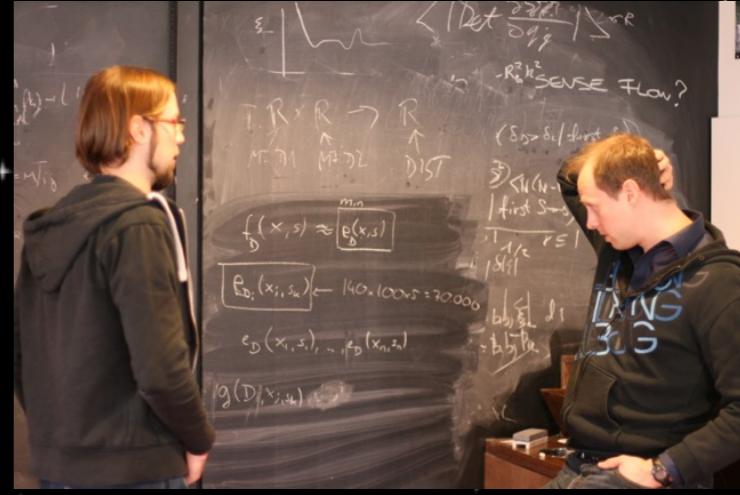
谢谢



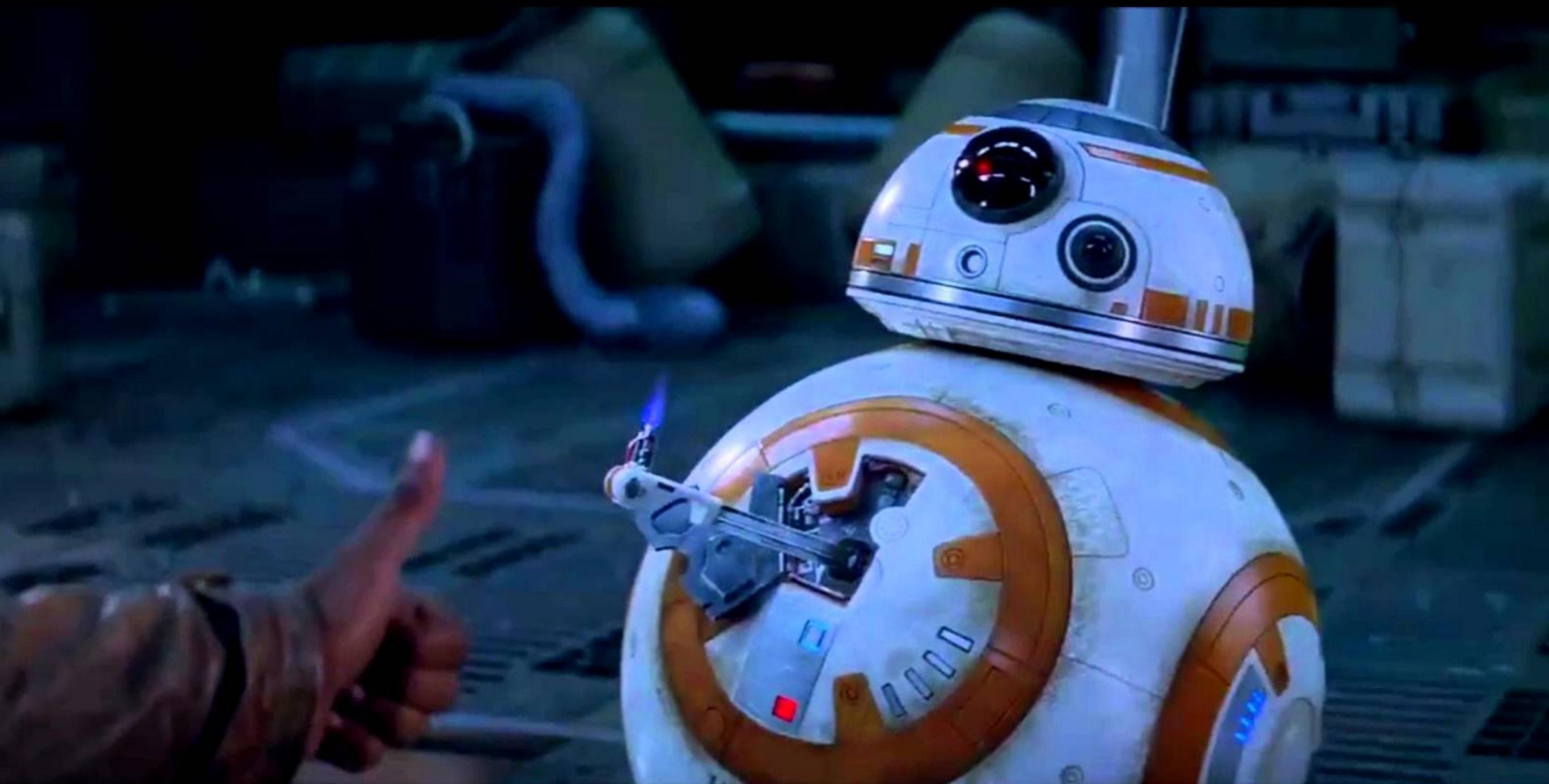


Thank you!

谢谢



Questions?



JOIN US

Looking for:

- PhD Students
- Scientific programmer

