OpenML

DEMOCRATIZING AND AUTOMATING MACHINE LEARNING

JOAQUIN VANSCHOREN, TU EINDHOVEN, 2017

@open_ml

OpenML

Be a part of this talk:

- Log in / create an account on www.openml.org
  - *You also need a GitHub account*
- Click the  or  icon
- Click ‘Launch Demo’
  - Launches notebook hosted by YSDA

Democratization

Empower anybody to do data science well
Provide easy access to reproducible data/code/models
Build easily on prior work

Allow frictionless collaboration
Crowdsource hard problems, share results easily
Organized, reproducible, open results

Simplify, speed up data science process
Automate drudge work (e.g. hyperparameter optimization)
Learn from past, build ever better algorithms, models

WHAT IF WE CAN ANALYSE DATA

COLLABORATIVELY

WHAT IF WE CAN ANALYSE DATA

COLLABORATIVELY

WHAT IF WE CAN ANALYSE DATA

COLLABORATIVELY

WHAT IF WE CAN ANALYSE DATA

COLLABORATIVELY

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

WHAT IF WE CAN ANALYSE DATA

COLLABORATIVELY

ON WEB SCALE

WHAT IF WE CAN ANALYSE DATA

COLLABORATIVELY

ON WEB SCALE

WHAT IF WE CAN ANALYSE DATA

COLLABORATIVELY

ON WEB SCALE

WHAT IF WE CAN ANALYSE DATA

COLLABORATIVELY

ON WEB SCALE

WHAT IF WE CAN ANALYSE DATA

COLLABORATIVELY

ON WEB SCALE

WHAT IF WE CAN ANALYSE DATA

COLLABORATIVELY

ON WEB SCALE

WHAT IF WE CAN ANALYSE DATA

COLLABORATIVELY

ON WEB SCALE

WHAT IF WE CAN ANALYSE DATA

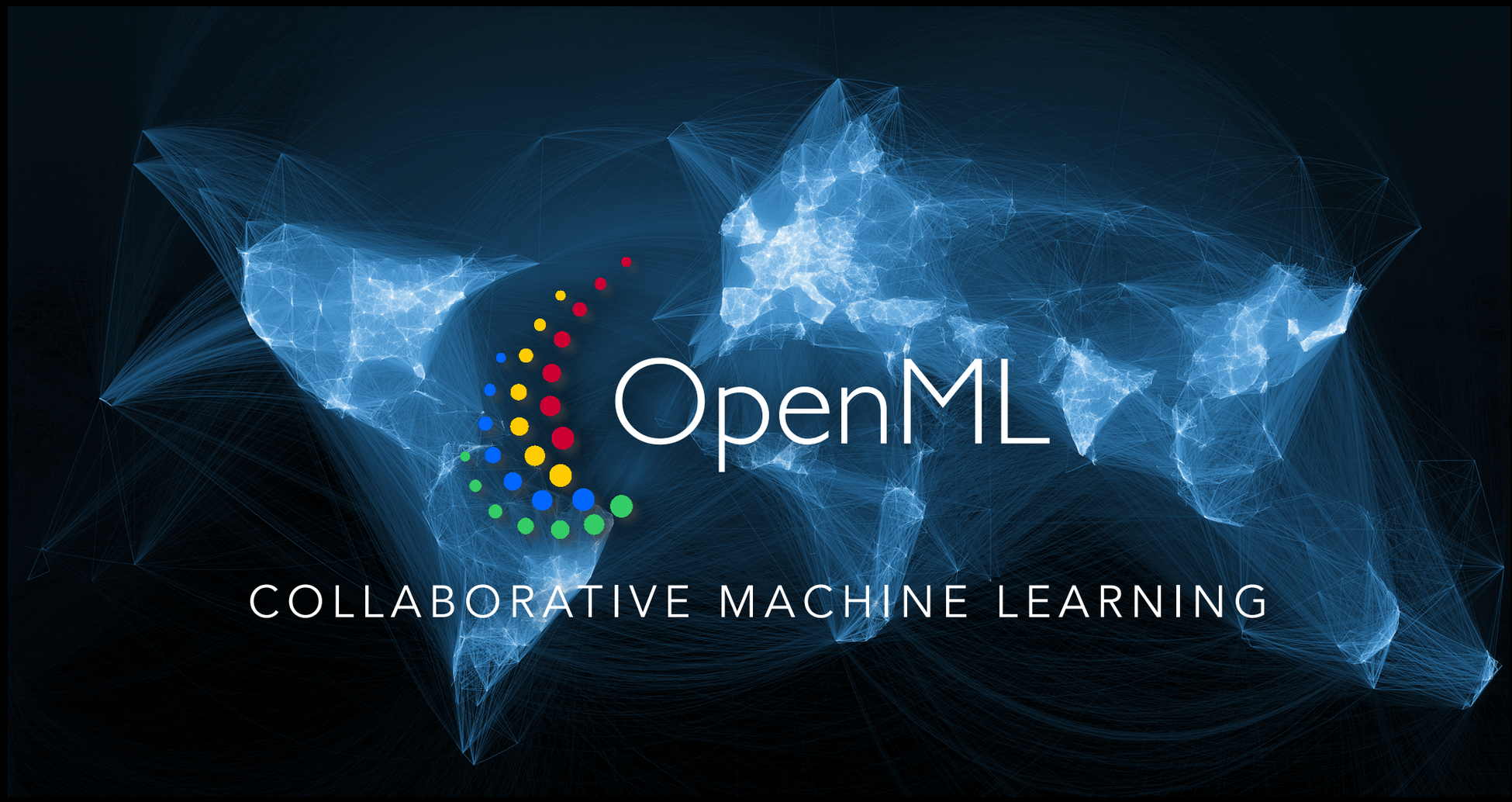
COLLABORATIVELY

ON WEB SCALE IN REAL TIME

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

UCI Machine Learning Repository

mldata.org machine learning data set repository

figshare

zenodo

kaggle

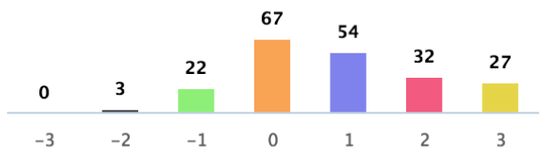
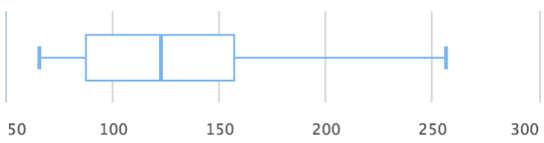
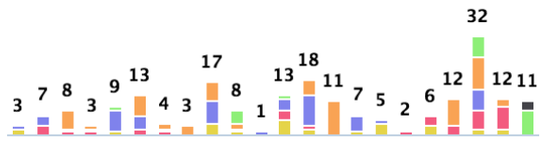
Data (ARFF) uploaded or referenced (URL)

auto-versioned, analysed, meta-data extracted, organised online

auto-versioned, analysed, meta-data
extracted, organised online

auto-versioned, analysed, meta-data
extracted, organised online

26 features

symboling **(target)**	nominal	6 unique values 0 missing	
normalized-losses	numeric	51 unique values 41 missing	
make	nominal	22 unique values 0 missing	

Show all 26 features

72 properties

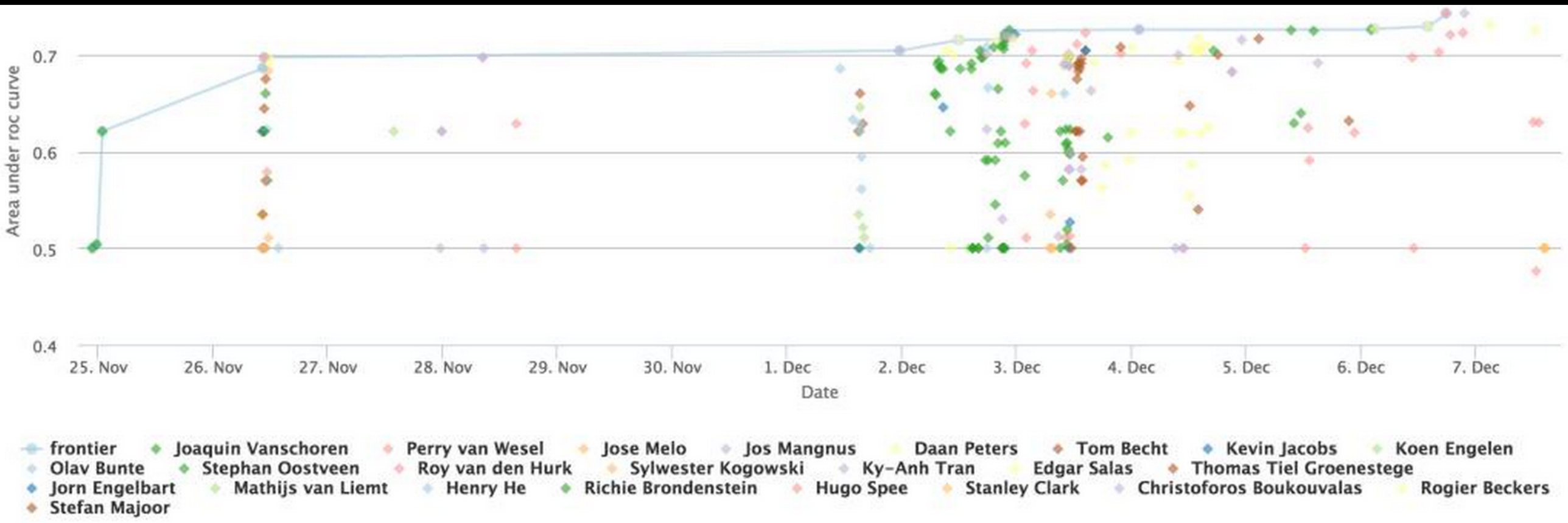
DefaultAccuracy	**0.33**	The predictive a
NumberOfClasses	**7**	The number of c
NumberOfFeatures	**26**	The number of f
NumberOfInstances	**205**	The number of i
NumberOfMissingVal...	**59**	Counts the total

Tasks contain data, goals, procedures.

Readable by tools, automates experimentation

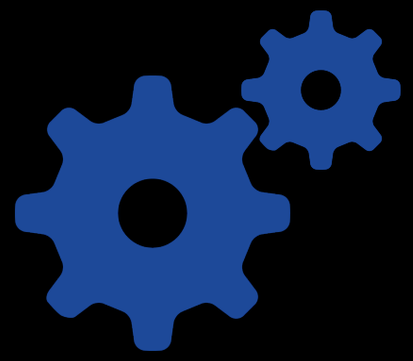
All results organized online: **realtime overview**

All results organized online: **realtime overview**

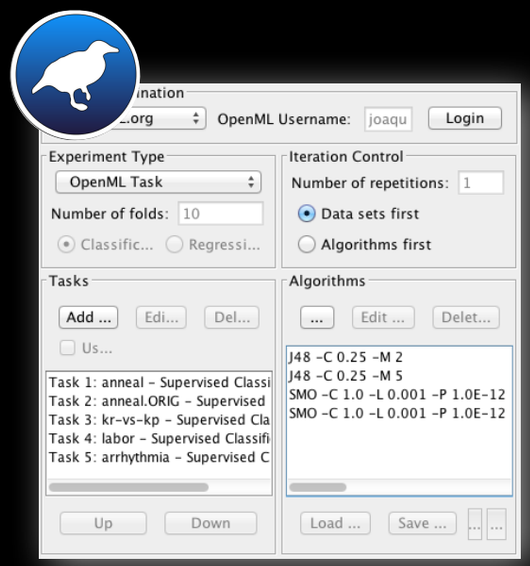


Flows (workflows, scripts) can run anywhere (locally)

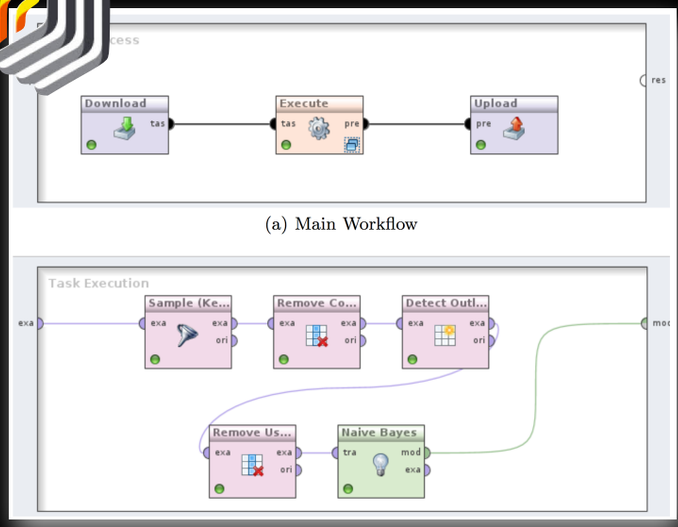
Tool integrations + APIs (REST, R, Python, Java,…)



Integrations + APIs (REST, R, Python, Java,...)

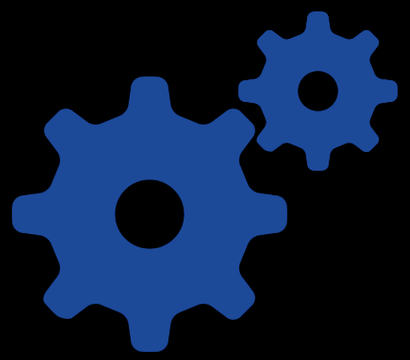


```
from sklearn import tree
from openml import tasks,runs
task = tasks.get_task(14951)
clf = tree.DecisionTreeClassifier()
run = runs.run_task(task, clf)
return_code, response = run.publish()
```



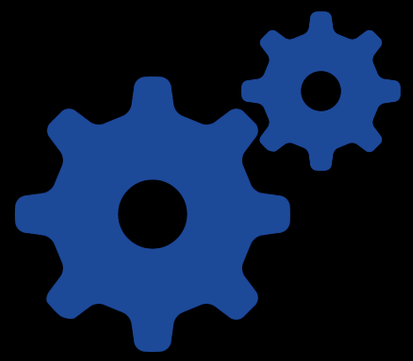
(a) Main Workflow

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



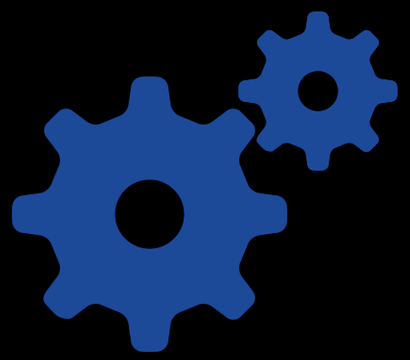
Integrations + APIs (REST, R, Python, Java,...)

```python
from sklearn import tree
from openml import tasks,runs
task = tasks.get_task(14951)
clf = tree.DecisionTreeClassifier()
run = runs.run_task(task, clf)
return_code, response = run.publish()
```



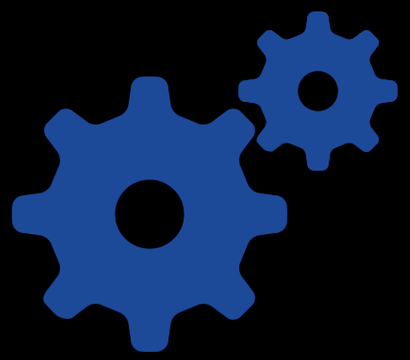
Integrations + APIs (REST, R, Python, Java,...)

```
from sklearn import tree
from openml import tasks,runs
task = tasks.get_task(14951)
clf = tree.DecisionTreeClassifier()
run = runs.run_task(task, clf)
return_code, response = run.publish()
```



Integrations + APIs (REST, R, Python, Java,…)

```python
from sklearn import tree
from openml import tasks,runs
task = tasks.get_task(14951)
clf = tree.DecisionTreeClassifier()
run = runs.run_task(task, clf)
return_code, response = run.publish()
```



Integrations + APIs (REST, R, Python, Java,...)

```
from sklearn import tree
from openml import tasks,runs
task = tasks.get_task(14951)
clf = tree.DecisionTreeClassifier()
run = runs.run_task(task, clf)
return_code, response = run.publish()
```

Experiments auto-uploaded, evaluated online

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

and **all other experiments**

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.

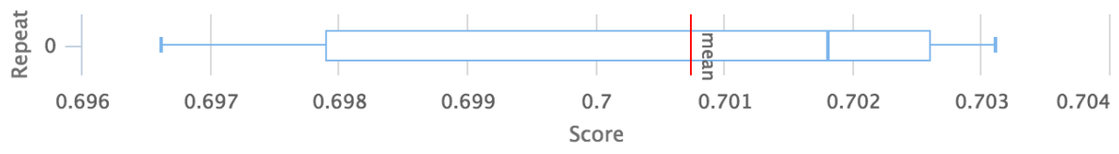
Area under ROC curve

0.7007 ± 0.0023

Per class

0	1
0.7007	0.7007

Cross-validation details (10-fold Crossvalidation)



Demo

OpenML beta

Exploring machine learning better, together

19626 data sets

Find or add **data** to analyse

25941 tasks

Download or create scientific **tasks**

3353 flows

Find or add data analysis **flows**

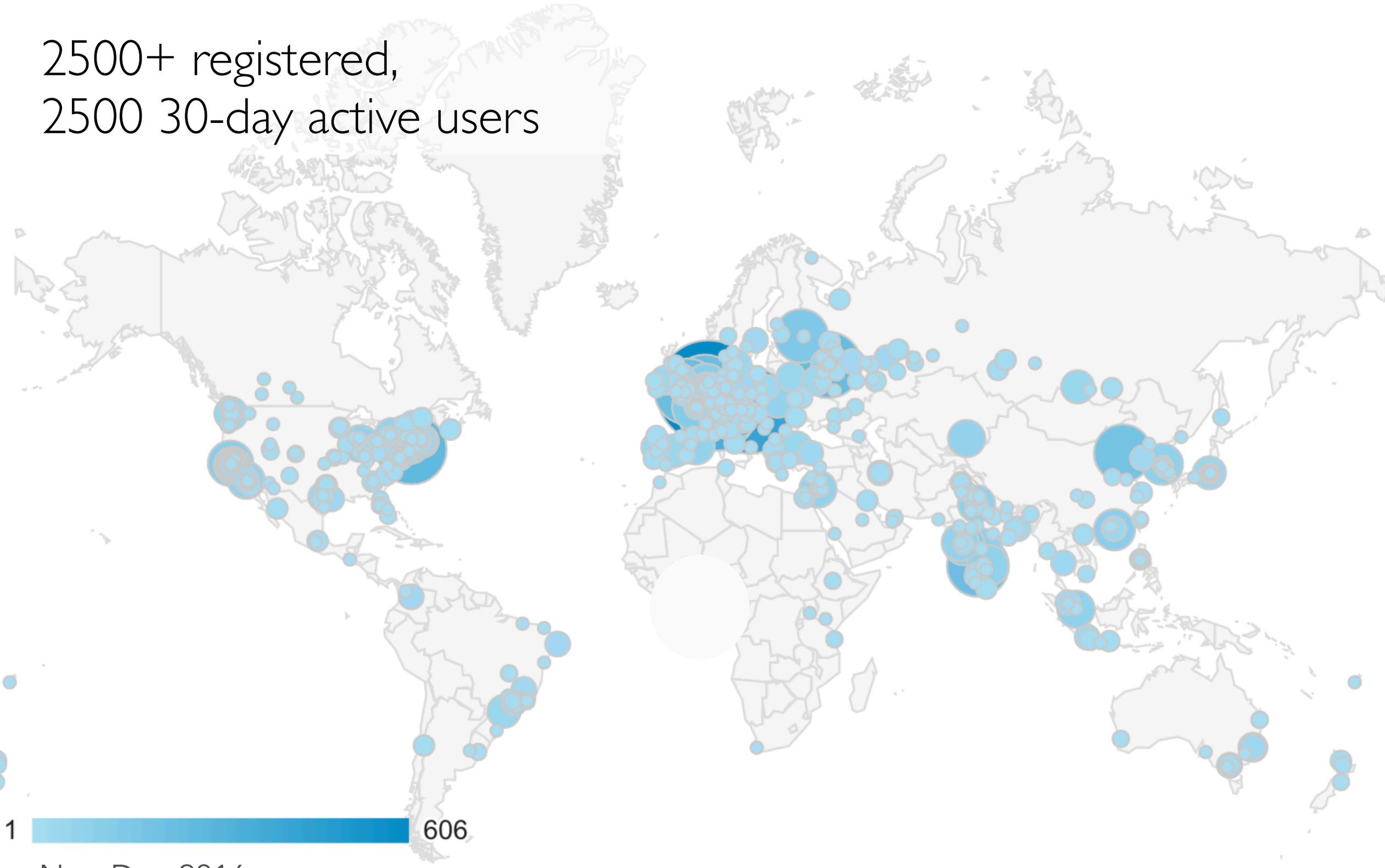
1698629 runs

Upload and explore all **results** online.

Download and share data, flows and runs through:

OpenML Community

2500+ registered,
2500 30-day active users



Nov-Dec 2016

Collaboration tools (in progress)

Circles
Create collaborations with trusted researchers

Studies (e-papers)
Online counterpart of a paper, chat/comments
Linked to GitHub repo, Jupyter notebooks

Code submissions
Sharing versioned code, docker images, archiving
GitHub integration

Collaborative challenges
Rapid Analytics and Model Prototyping
Code submission, streamlined collaboration

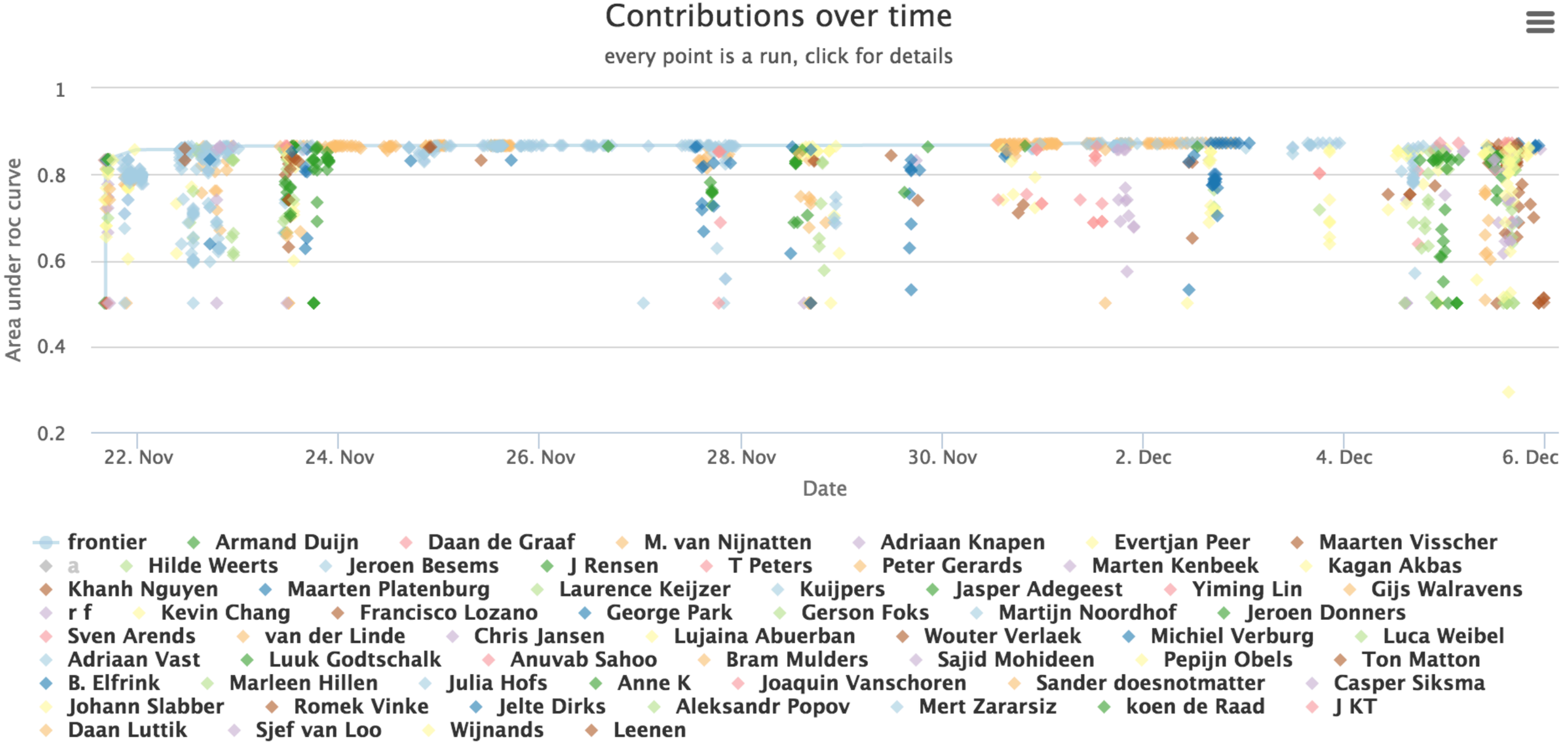
Classroom challenges

Results appear in real-time, full details available immediately

Students can learn from each other, but also have to think to do better

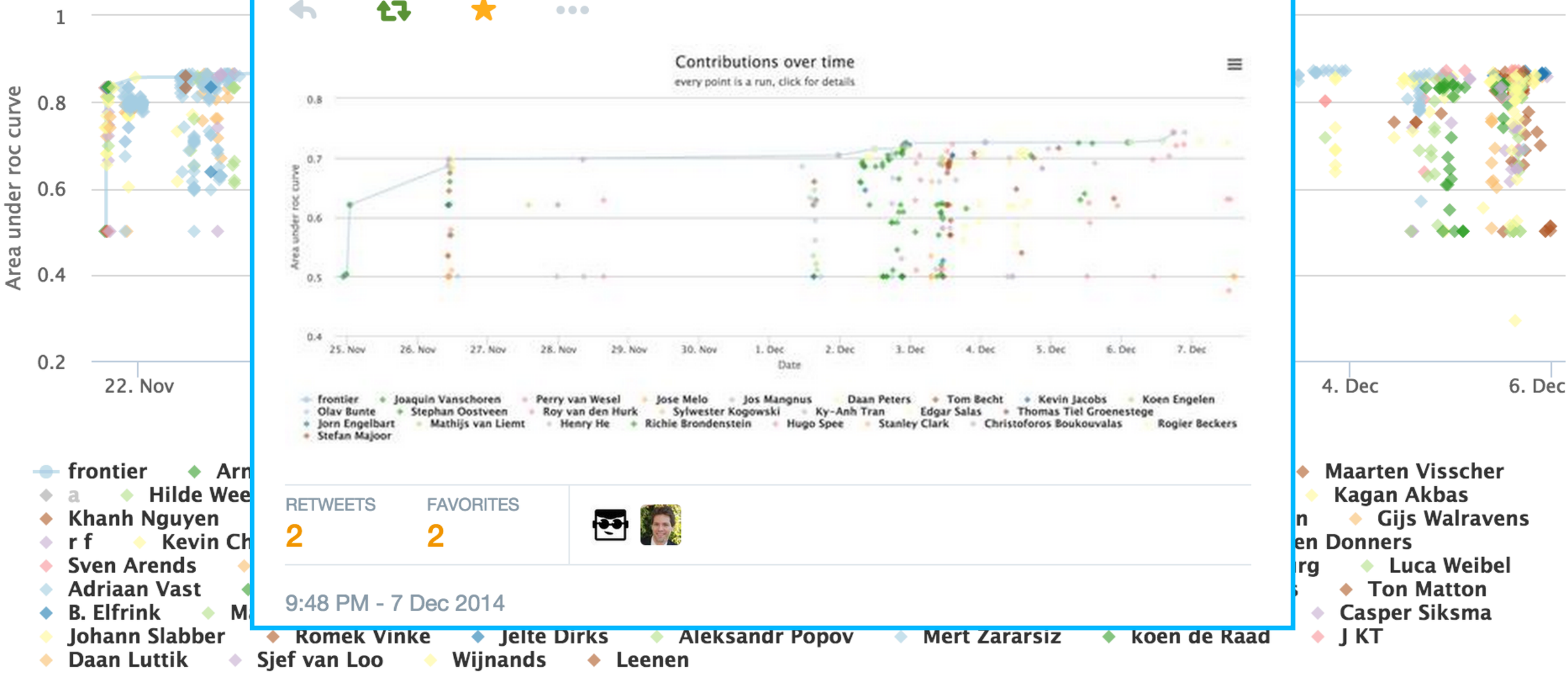
Simple resubmissions do not count

Hidden test set is also possible



Class

Results app
Students ca ter
Simple resu
Hidden test



Rogier Beckers
@RogierBeckers

Follow

Het bewijs dat ik studeer op zondag!
“@joavanschoren: #Machinelearning students on a #collaborative data mining ”

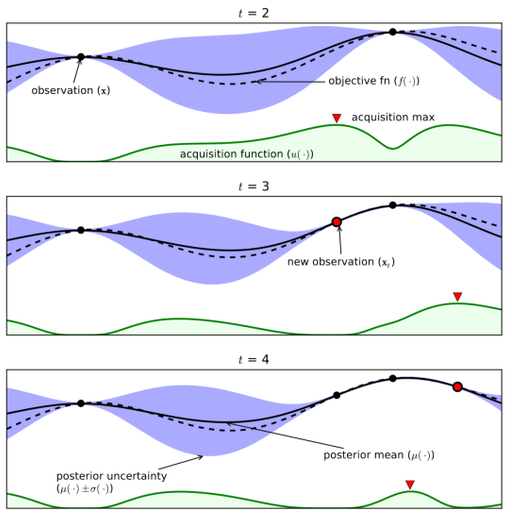
View translation

Lauradorp, Landgraaf

RETWEETS 2 FAVORITES 2

9:48 PM - 7 Dec 2014

Realtime challenges (coming up)

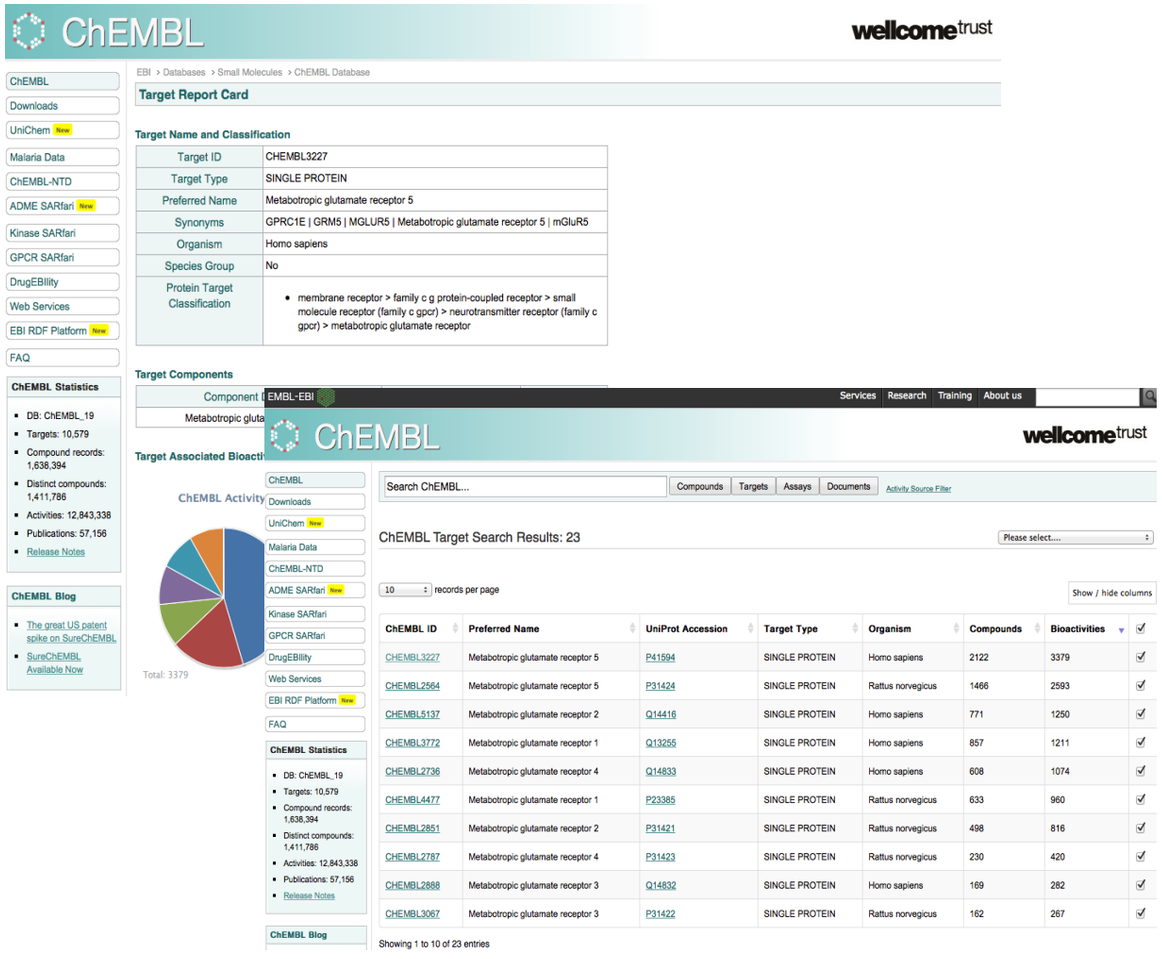


Hyperparameter optimization challenge
Every iteration can be uploaded, prior ones downloaded

Predict energy usage in smart energy
API streams data, predictions uploaded hourly

OpenML in drug discovery

Predict which drugs will inhibit certain proteins (and hence viruses, parasites,…)



SMILES

Molecular properties (e.g. MW, LogP)

Fingerprints (e.g. FP2, FP3, FP4, MACSS)

ChEMBL database

1.4M compounds,10k proteins,

12,8M activities

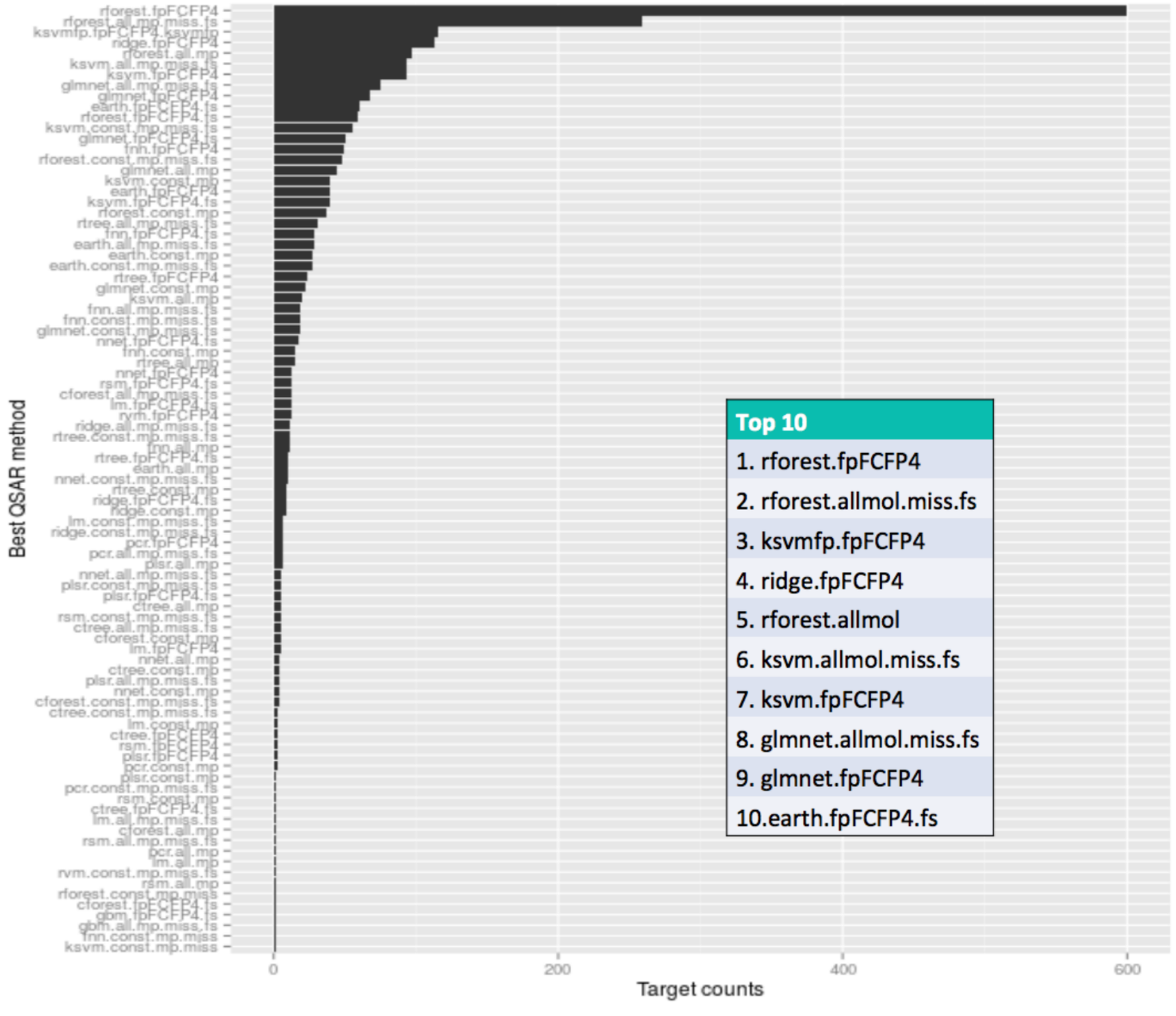
MW	LogP	TPSA	b1	b2	b3	b4	b5	b6	b7	b8	b9	
377.435	3.883	77.85	1	1	0	0	0	0	0	0	0	
341.361	3.411	74.73	1	1	0	1	0	0	0	0	0	…
197.188	-2.089	103.78	1	1	0	1	0	0	0	1	0	
346.813	4.705	50.70	1	0	0	1	0	0	0	0	0	
							⋮					

16.000+ QSAR datasets

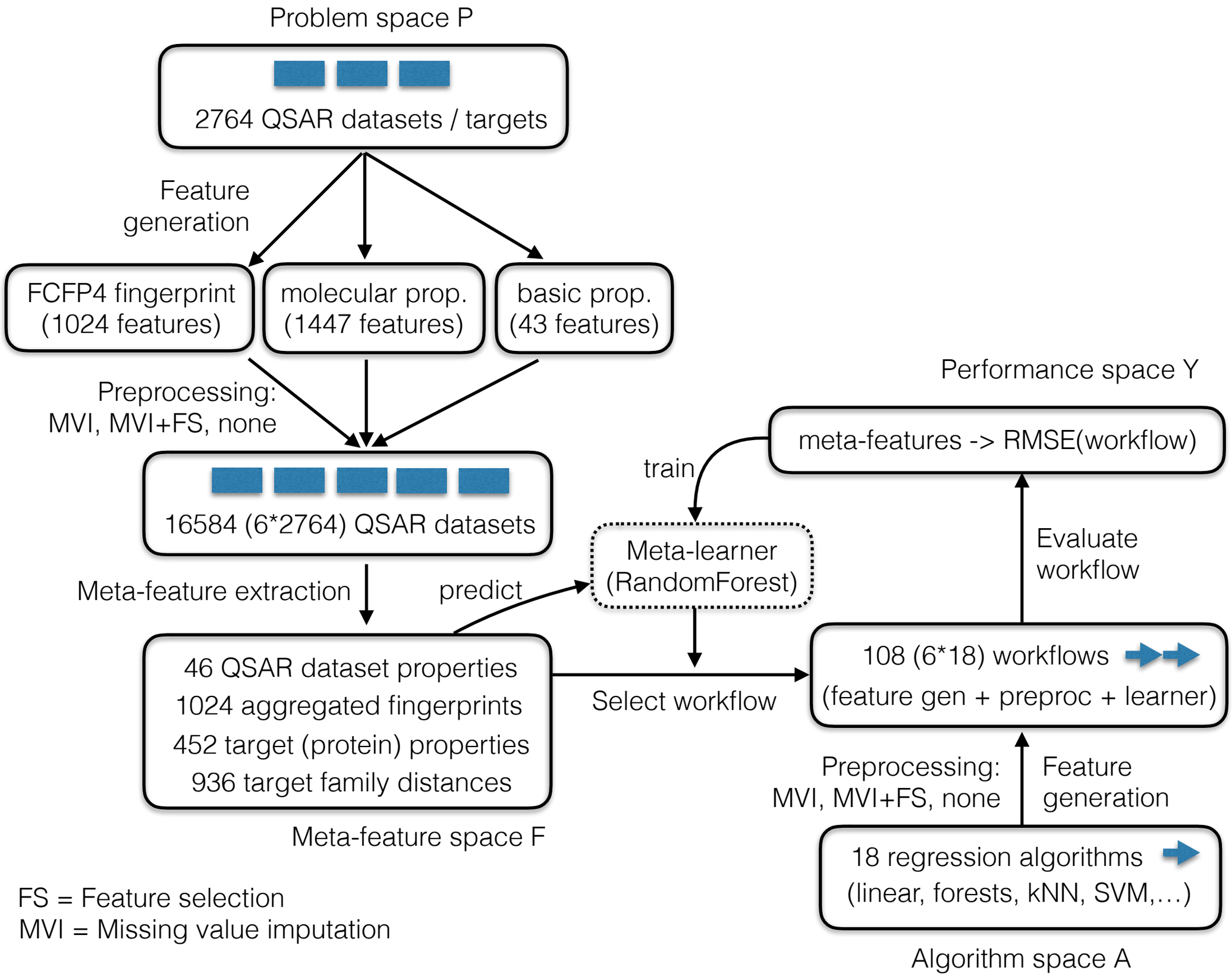
2750 targets (proteins), x 6 feature representations

OpenML in drug discovery

Many algorithms
Many feature representations



Predicting workflows with MetaLearning



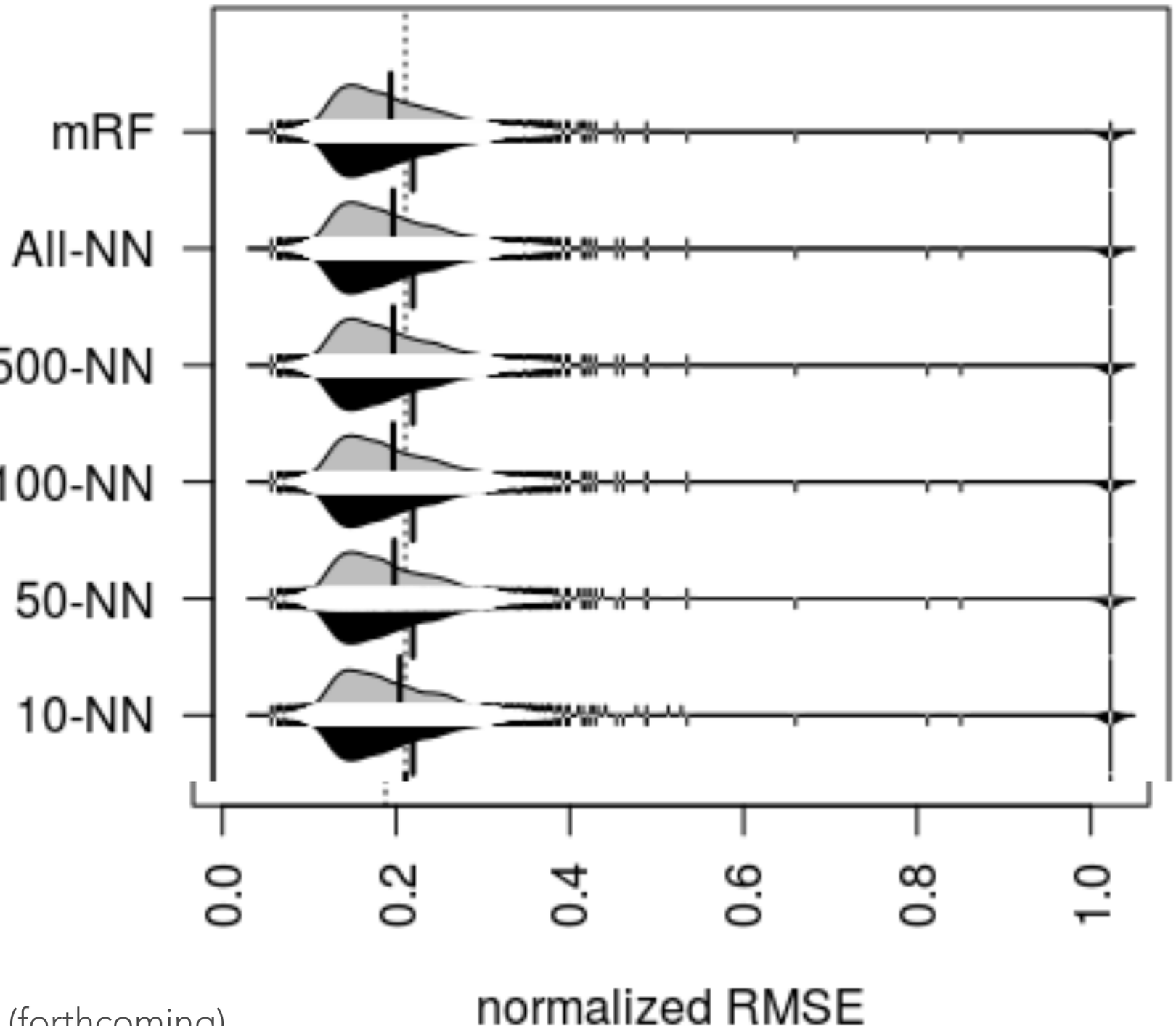
OpenML in drug discovery

Random forest (black) vs meta-learner (grey)

Meta-learner:

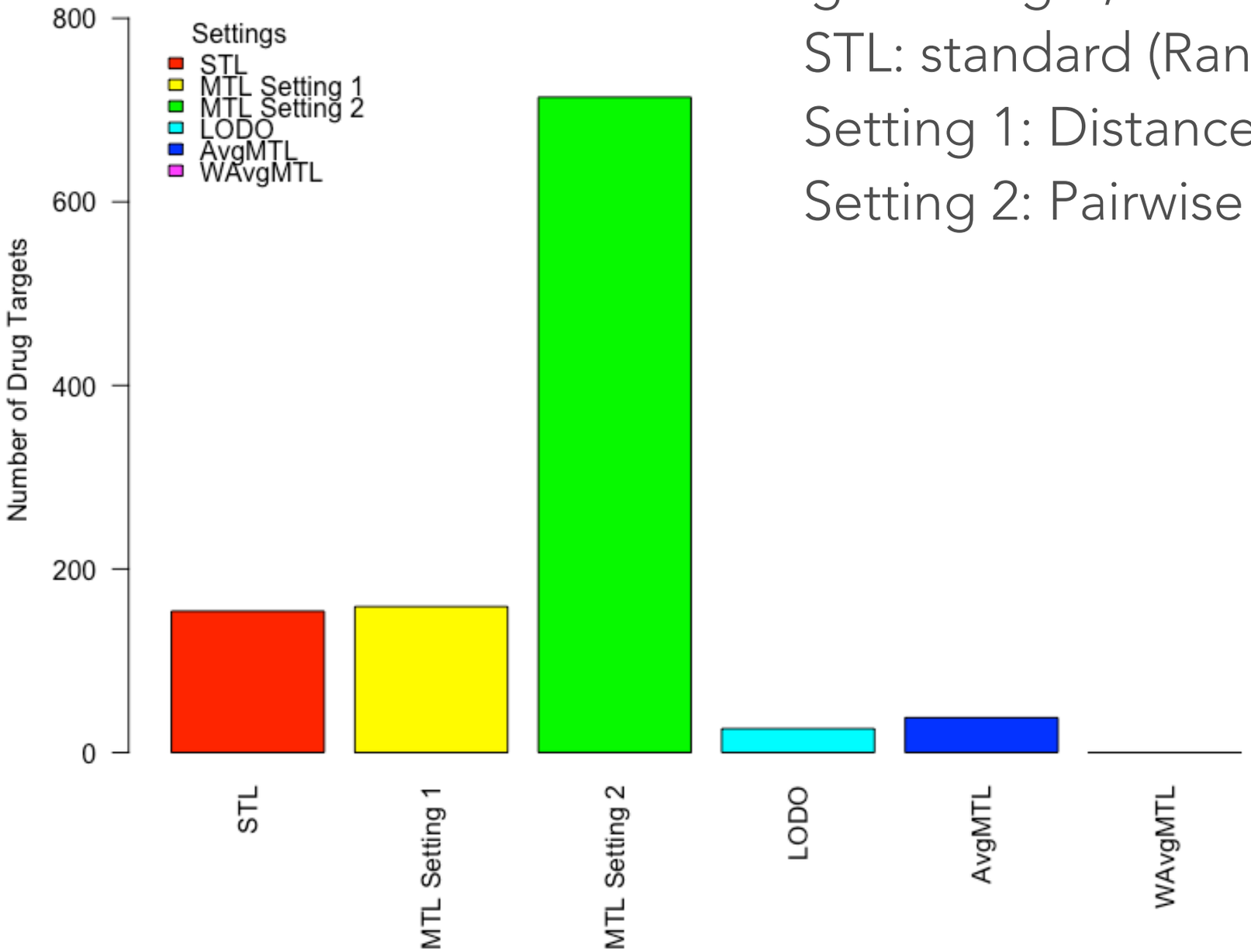
mRF: Random forest

k-NN: kNN



I. Olier et al. (2017) Machine Learning (forthcoming)

OpenML in drug discovery



Multi-target learning: if few drugs are tested on a given target, include data on 'related' targets:

STL: standard (Random Forests)

Setting 1: Distance is taxonomy (ChEMBL)

Setting 2: Pairwise sequence alignment

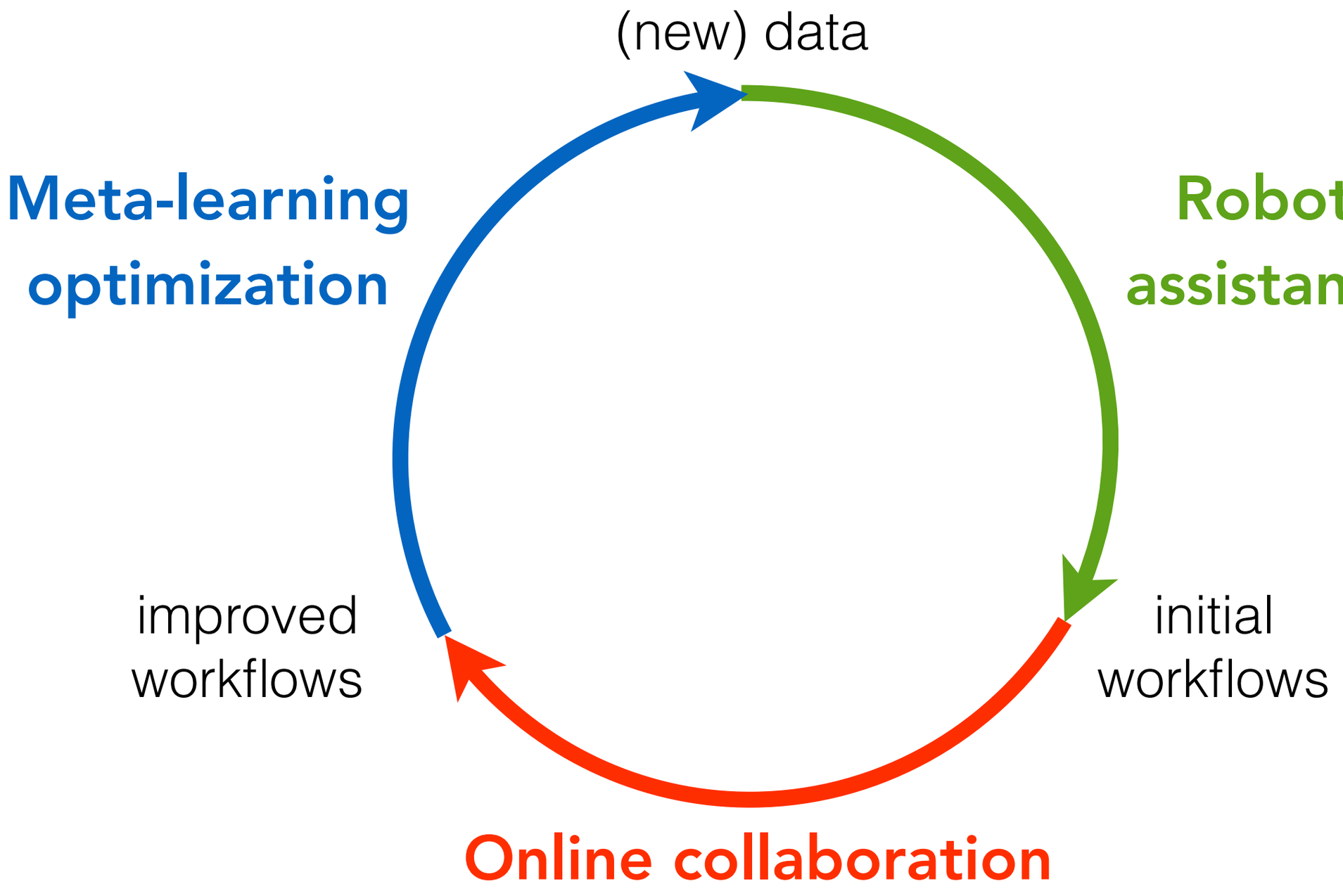
We just scratched the surface. All data is available on OpenML.

Automating machine learning

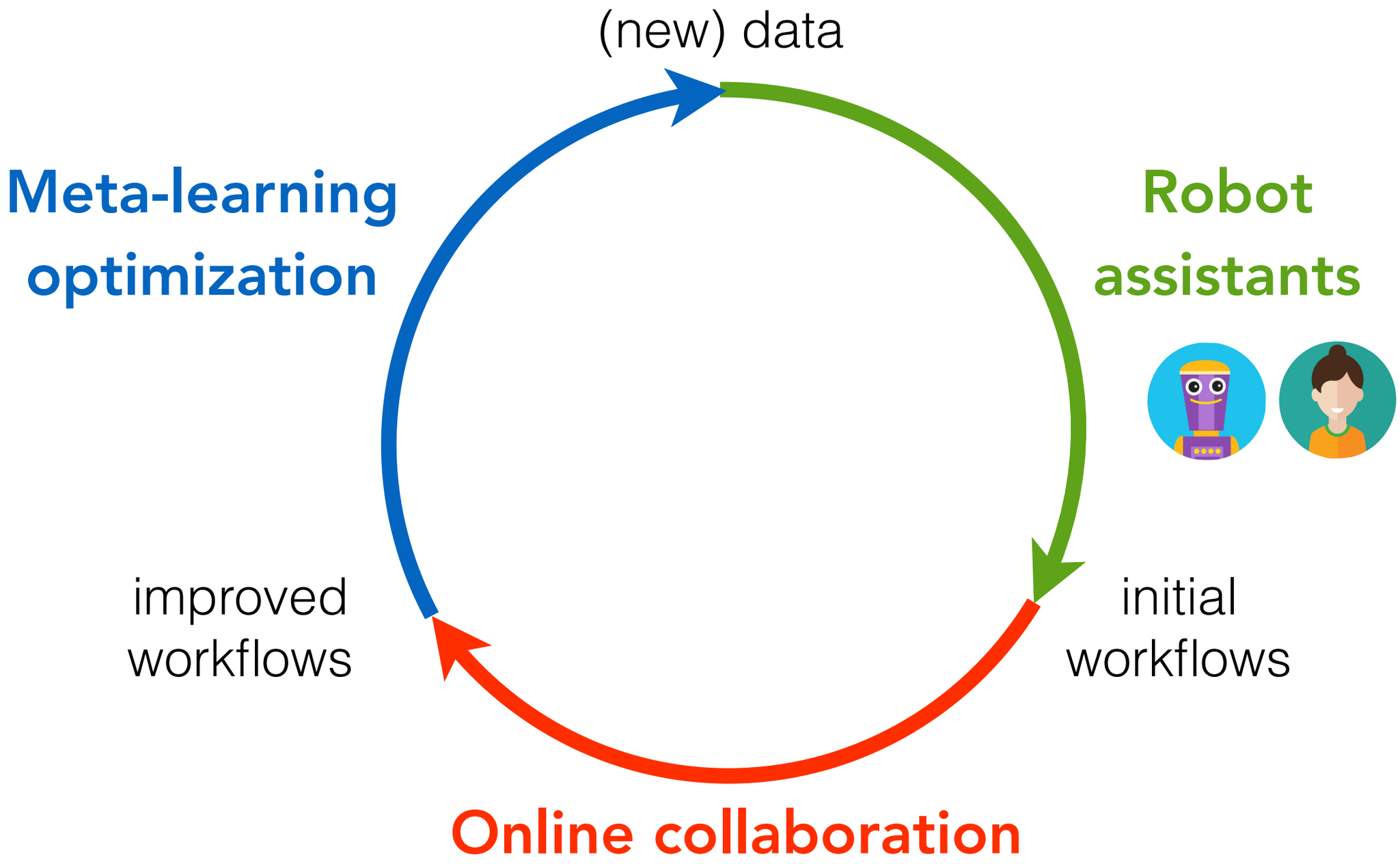
Data-driven modelling
Learn how to build models based on many prior experiments

AI-human interaction
Bots/services to simplify work
Find data, construct pipelines, optimize hyperparameters,…

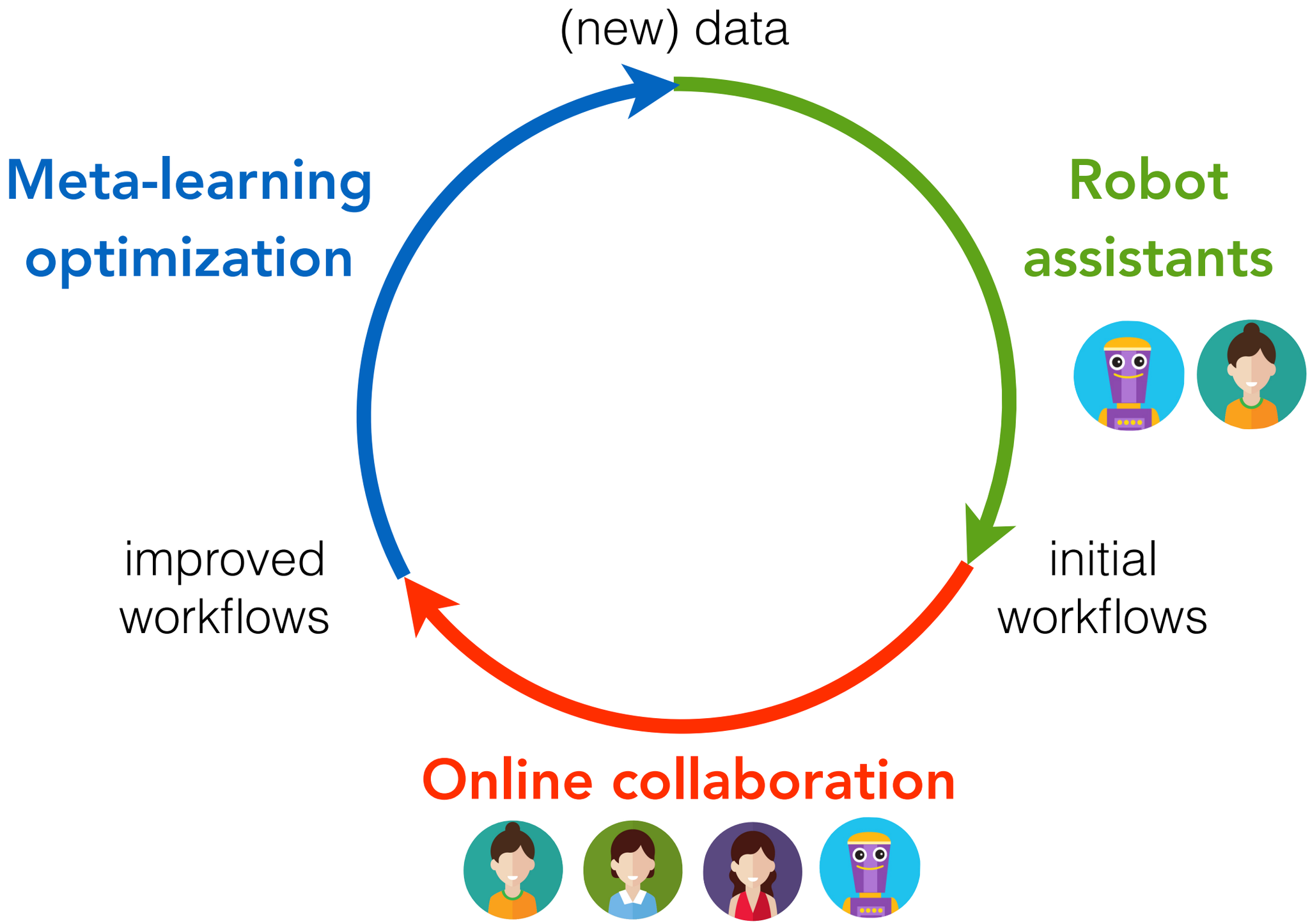
Automating machine learning: a human-robot symbiosis



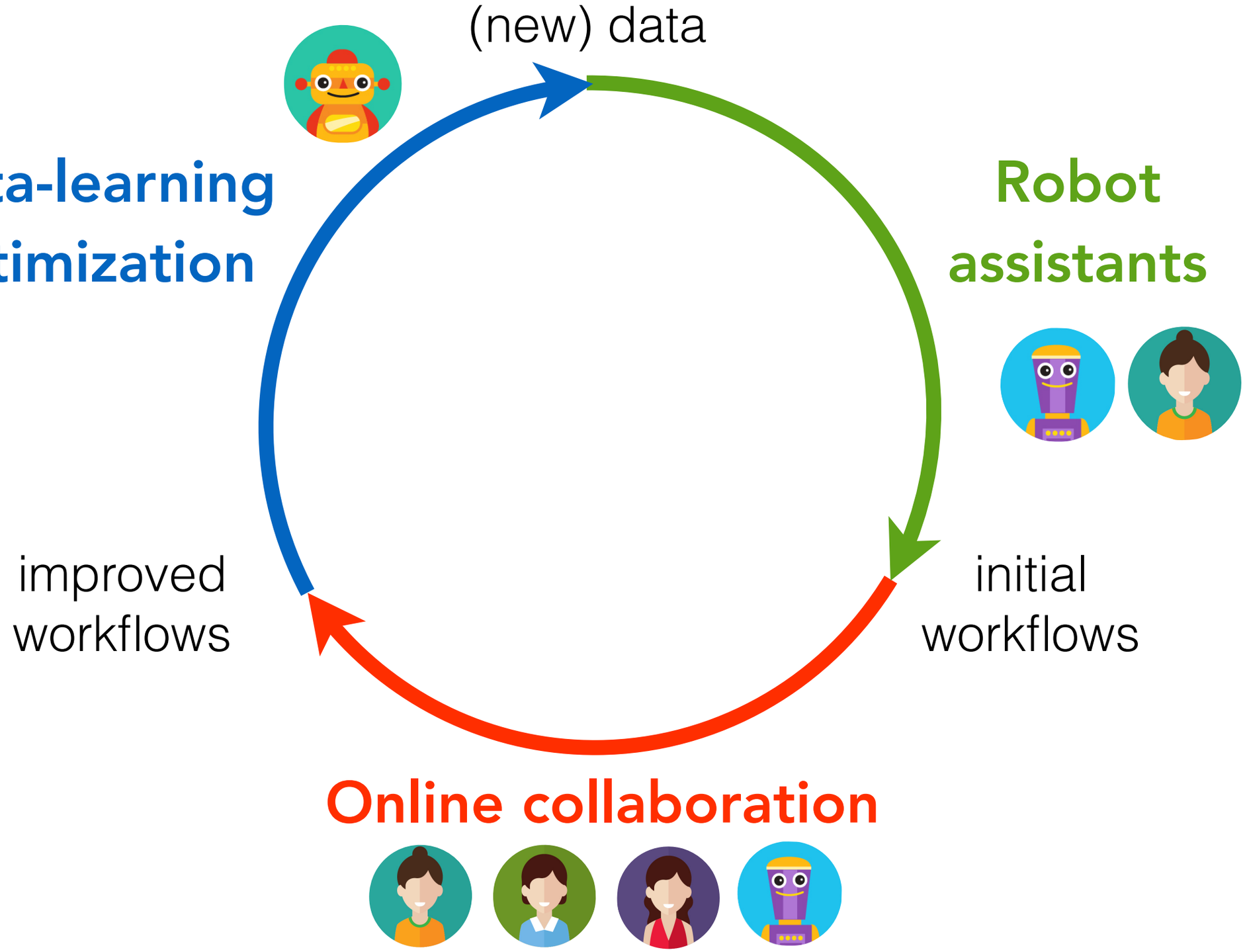
Automating machine learning: a human-robot symbiosis



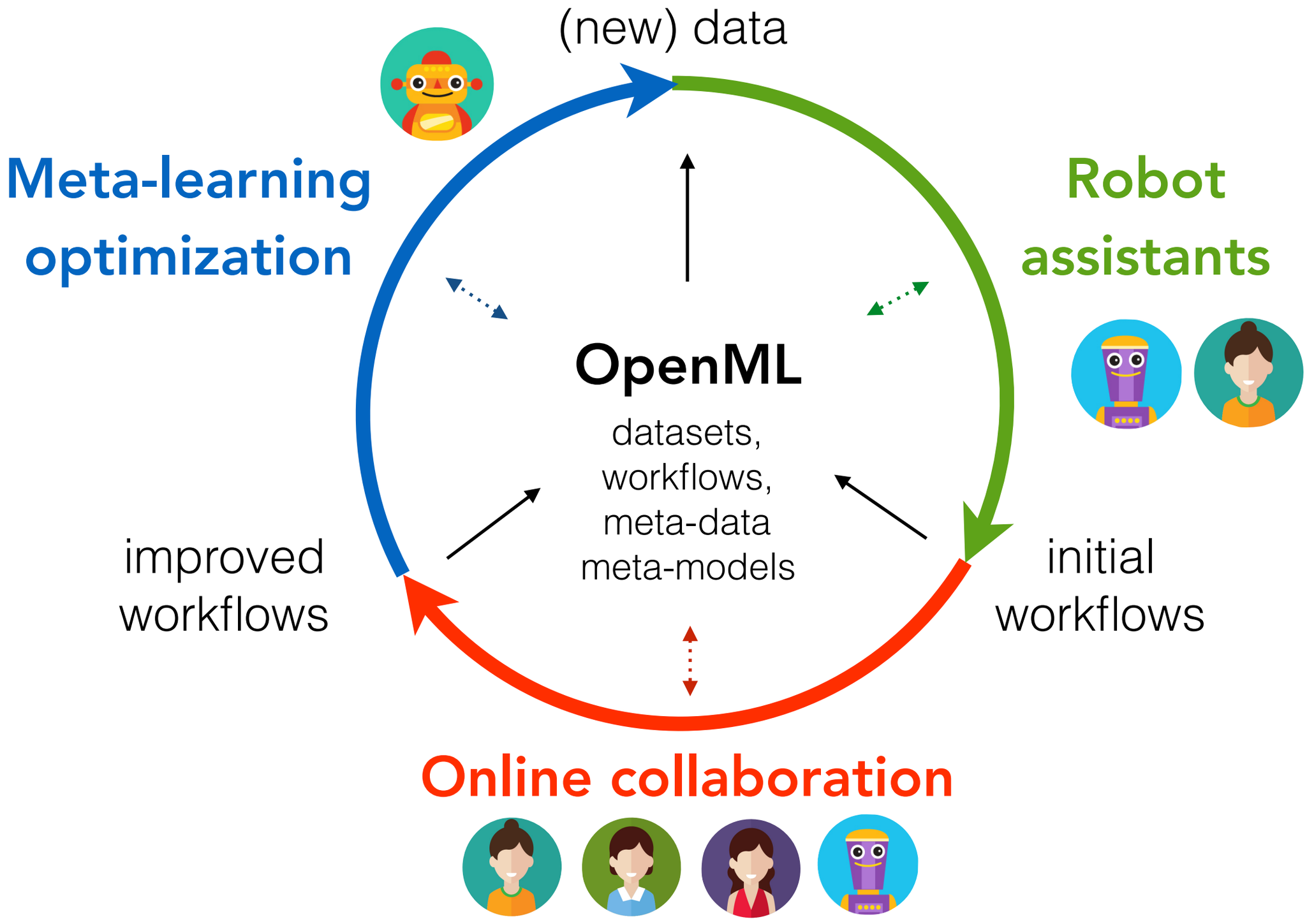
Automating machine learning: a human-robot symbiosis



Automating machine learning: a human-robot symbiosis

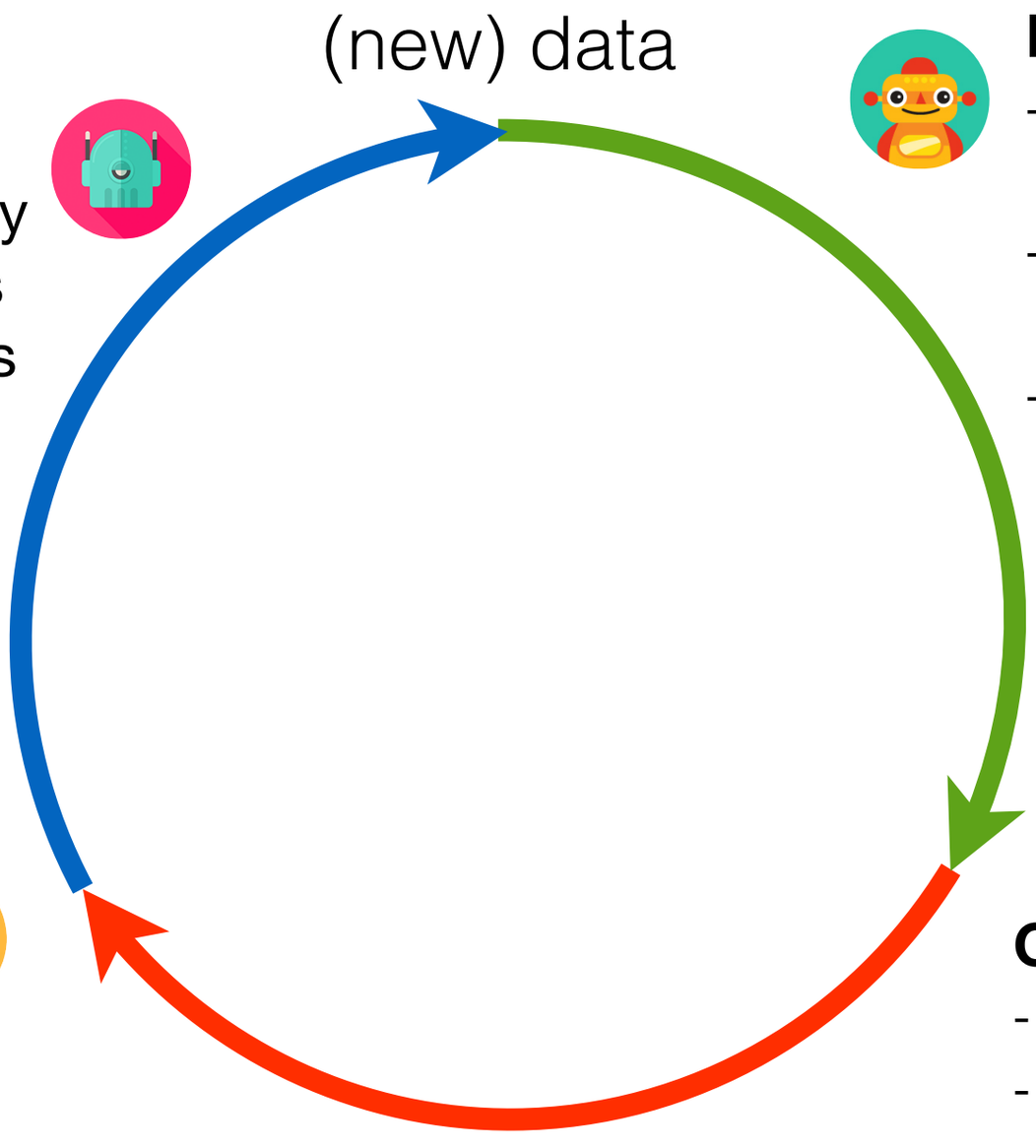


Automating machine learning: a human-robot symbiosis



Automating machine learning: a human-robot symbiosis

Meta-learning
- find similar datasets
- predict runtime, accuracy
- predict hyperparameters
  - good ranges, defaults
- trained meta-models
- recommend pipeline components



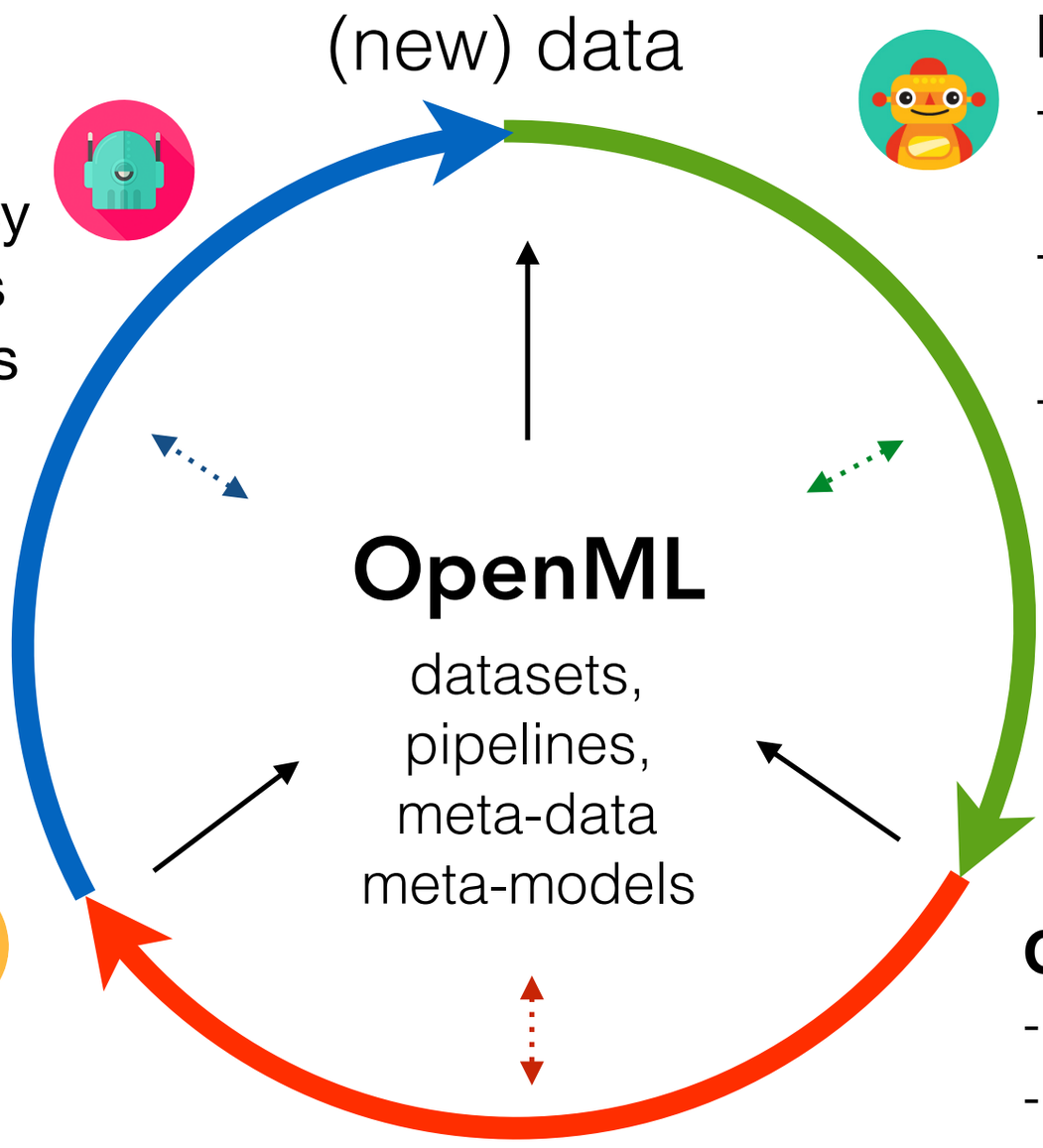
Extract meta-data
- (domain-specific) meta-features
- detect outliers, missing values,…
- Recommend primitives

Construct pipelines
- from (learned) templates
- generative methods
  - genetic programming
- by humans

Optimize pipelines
- Different strategies
  - random, model-based, bandits,…
- Learn from priors (meta-learning)
- *Upload all results*

Automating machine learning: a human-robot symbiosis

Meta-learning
- find similar datasets
- predict runtime, accuracy
- predict hyperparameters
  - good ranges, defaults
- trained meta-models
- recommend pipeline components

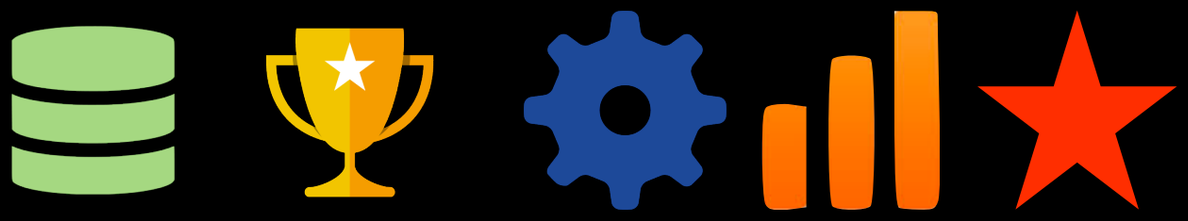


Extract meta-data
- (domain-specific) meta-features
- detect outliers, missing values,…
- Recommend primitives

Construct pipelines
- from (learned) templates
- generative methods
  - genetic programming
- by humans

Optimize pipelines
- Different strategies
  - random, model-based, bandits,…
- Learn from priors (meta-learning)
- *Upload all results*

OPENML API

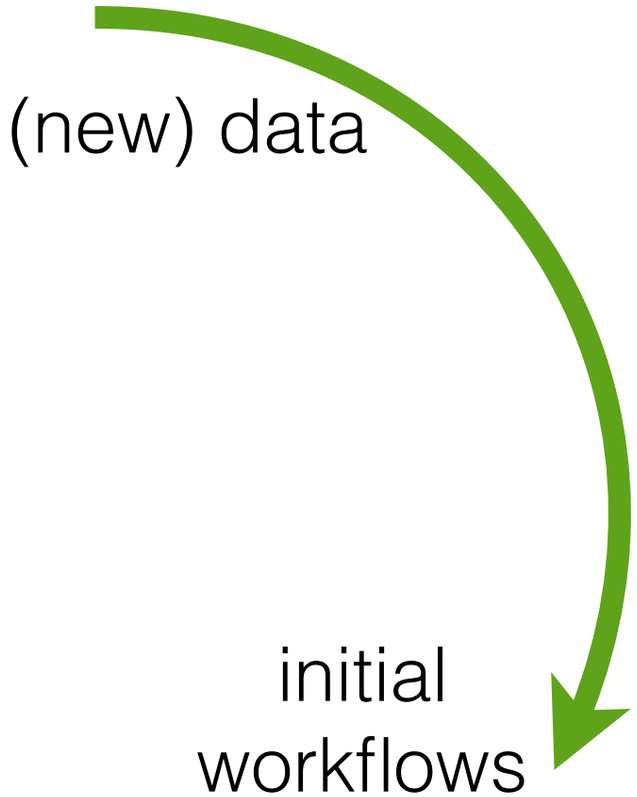


REST (XML/JSON), PYTHON, R, JAVA,...

Springboard for AutoML services

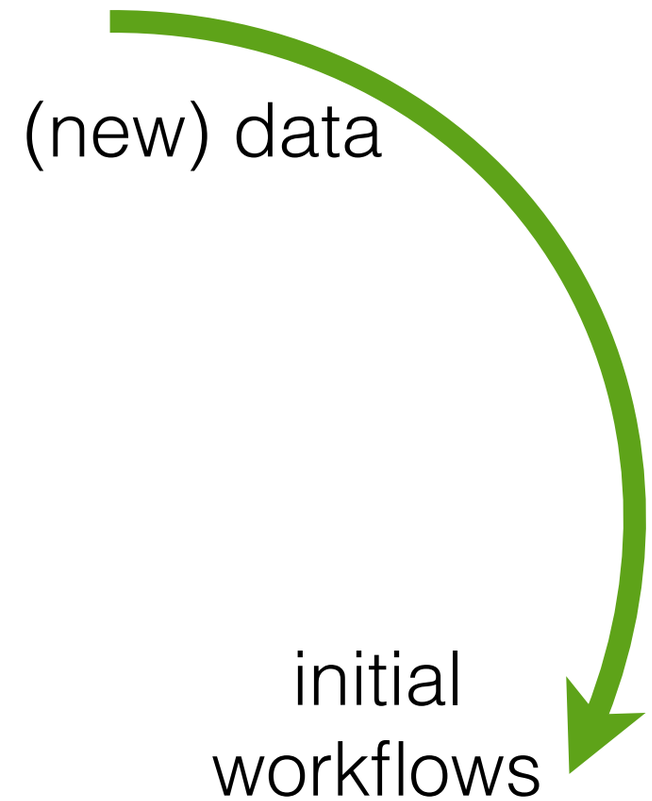
- Get Data, Flows, Models, Evaluations
- Download data, meta-data and run files
- Upload new data, flows, runs
- Upload/download optimization traces
- Upload/download meta-models
- Build AutoML bots that run against OpenML
- Compare AutoML approaches across datasets

Robot assistants: data handling



Robot assistants: data handling

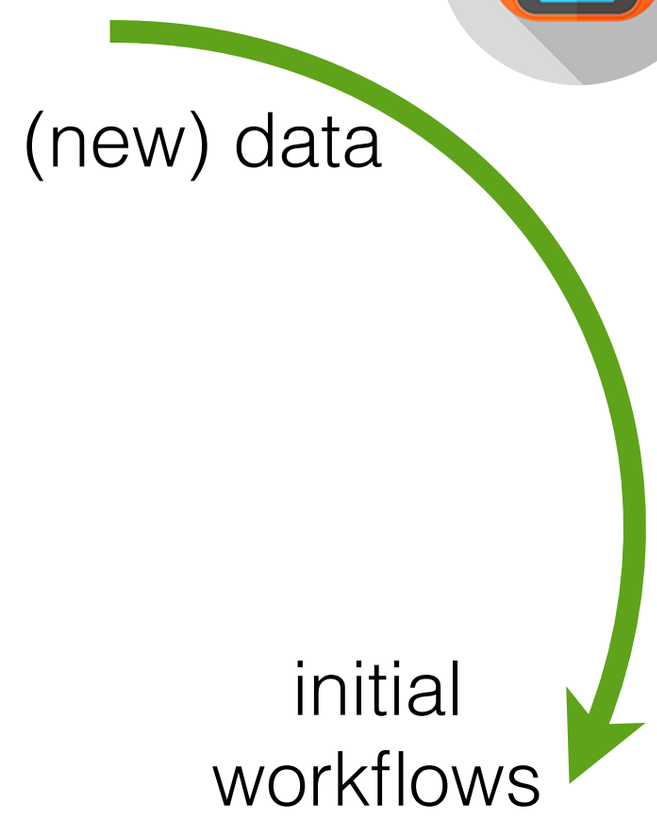
Data similarity bot: finds datasets similar to yours (meta-features)



Robot assistants: data handling

Data similarity bot: finds datasets similar to yours (meta-features)

Label imbalance bot: detects/ reduces class imbalance (e.g. SMOTE)



Robot assistants: data handling

Data similarity bot: finds datasets similar to yours (meta-features)

Label imbalance bot: detects/ reduces class imbalance (e.g. SMOTE)

(new) data

Encoding bot: converts to numeric data depending on ML algo (SVM, kNN, NN)

initial workflows

Robot assistants: data handling

Data similarity bot: finds datasets similar to yours (meta-features)

Label imbalance bot: detects/ reduces class imbalance (e.g. SMOTE)

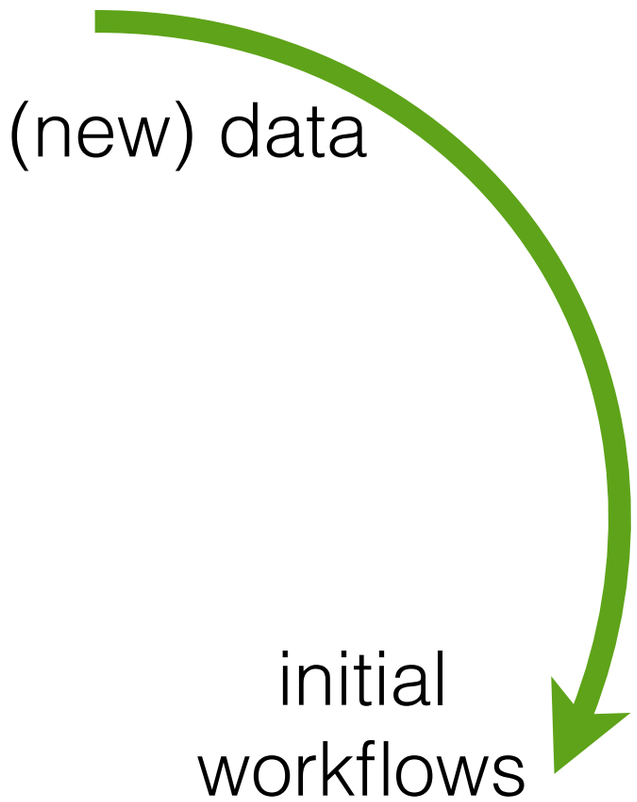
(new) data

Encoding bot: converts to numeric data depending on ML algo (SVM, kNN, NN)

Data leak bot: detects if test data leaks into the training set

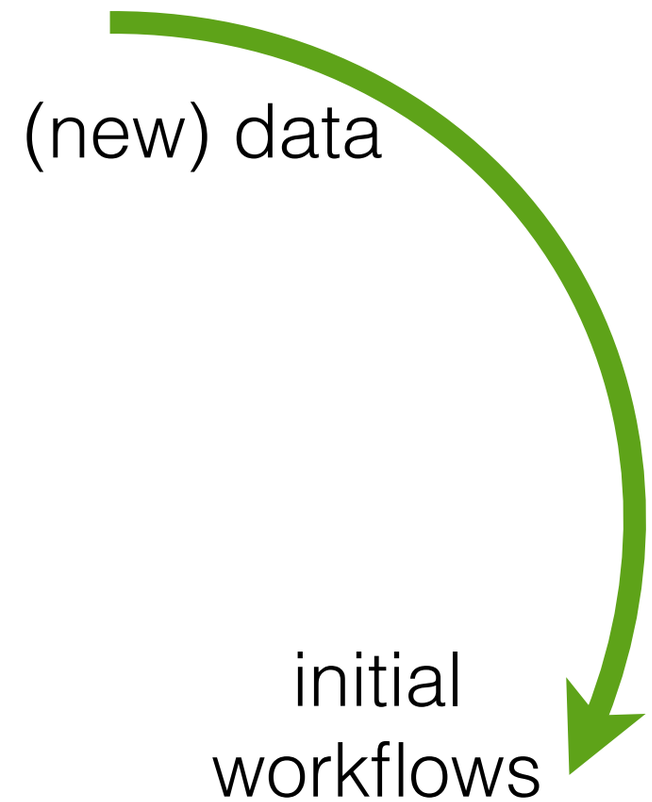
initial workflows

Robot assistants: preprocessing



Robot assistants: preprocessing

Runtime prediction bot: predicts how long an ML algorithm will run on your data



Robot assistants: preprocessing

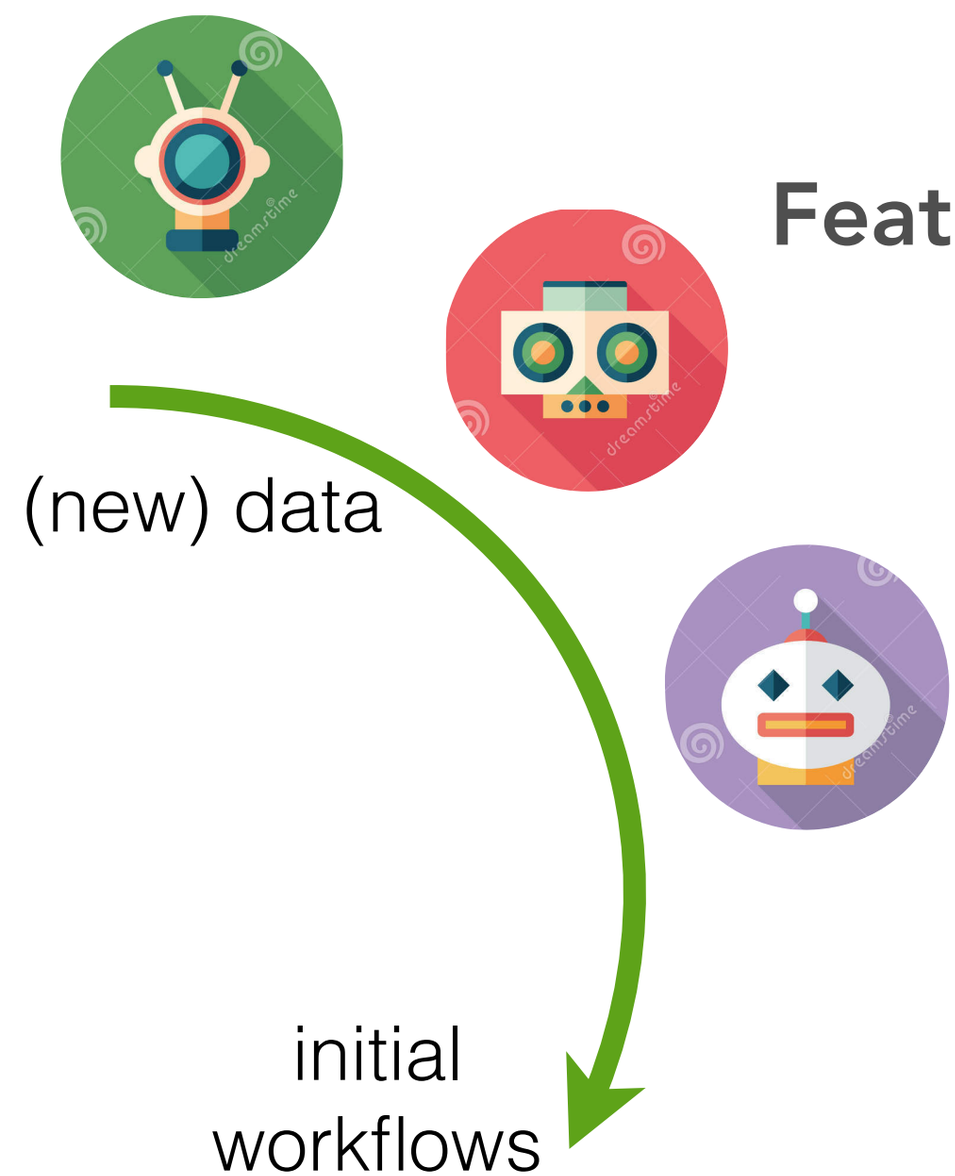
Runtime prediction bot: predicts how long an ML algorithm will run on your data

Feature selection bot: recommends/runs feature selection techniques

(new) data

initial workflows

Robot assistants: preprocessing

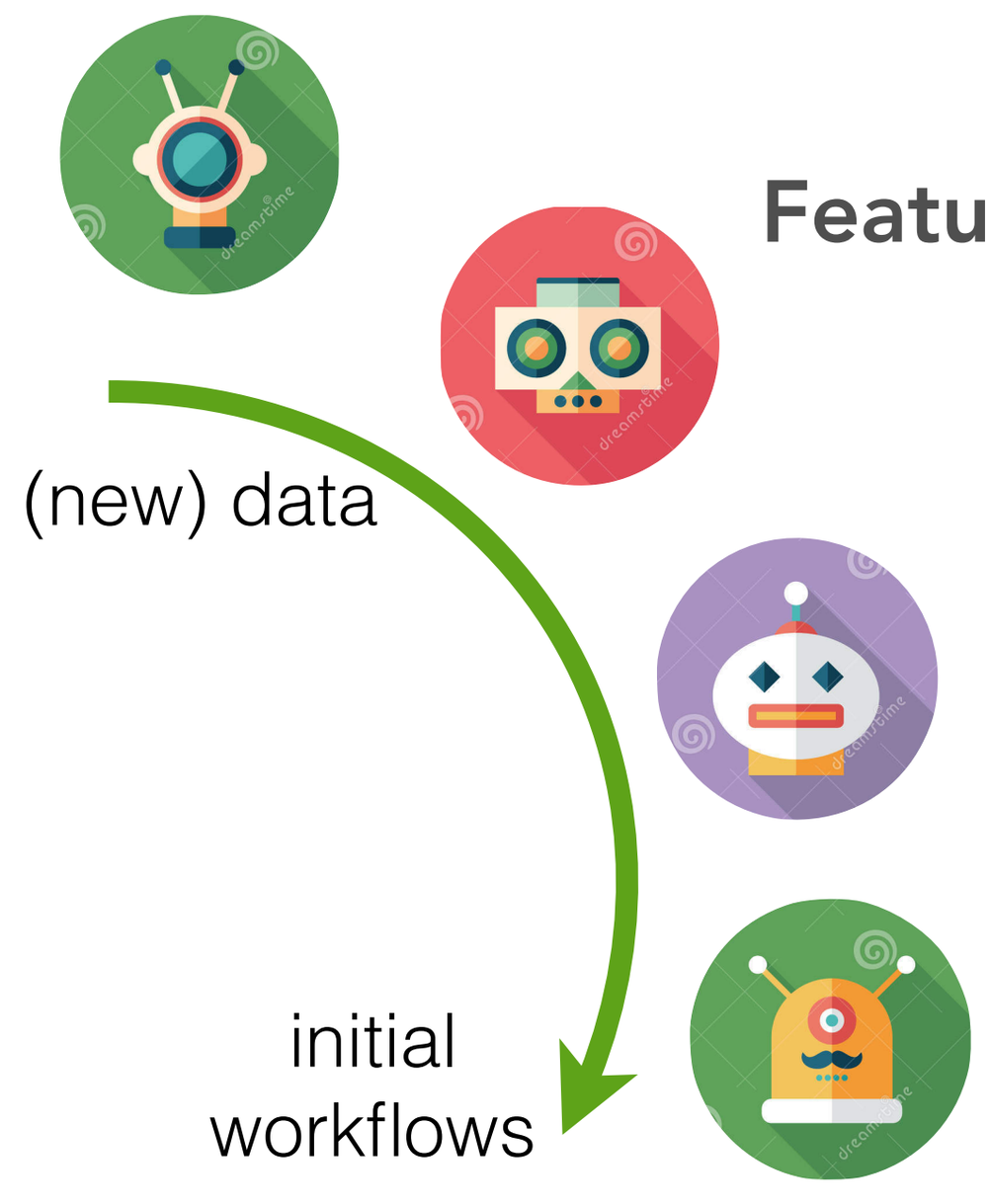


Runtime prediction bot: predicts how long an ML algorithm will run on your data

Feature selection bot: recommends/runs feature selection techniques

Imputation bot: recommends/runs missing value imputation techniques

Robot assistants: preprocessing



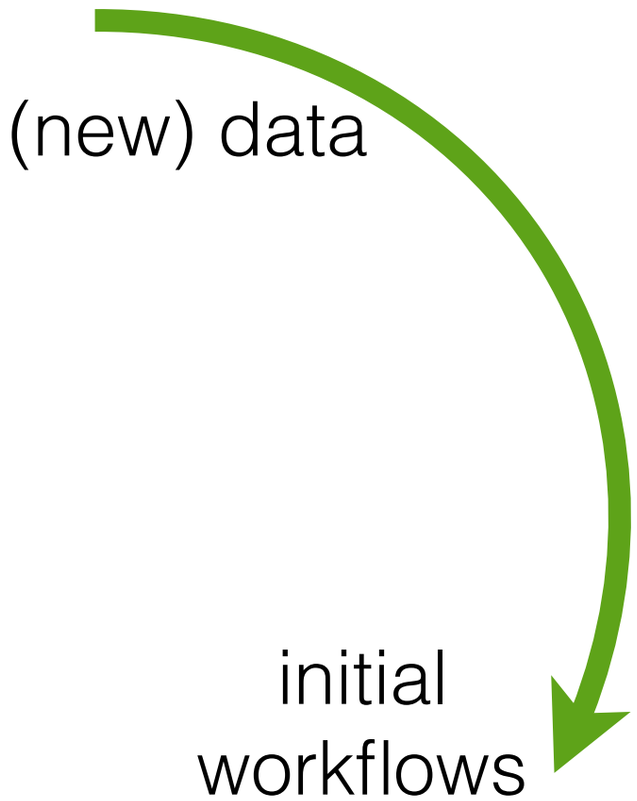
Runtime prediction bot: predicts how long an ML algorithm will run on your data

Feature selection bot: recommends/runs feature selection techniques

Imputation bot: recommends/runs missing value imputation techniques

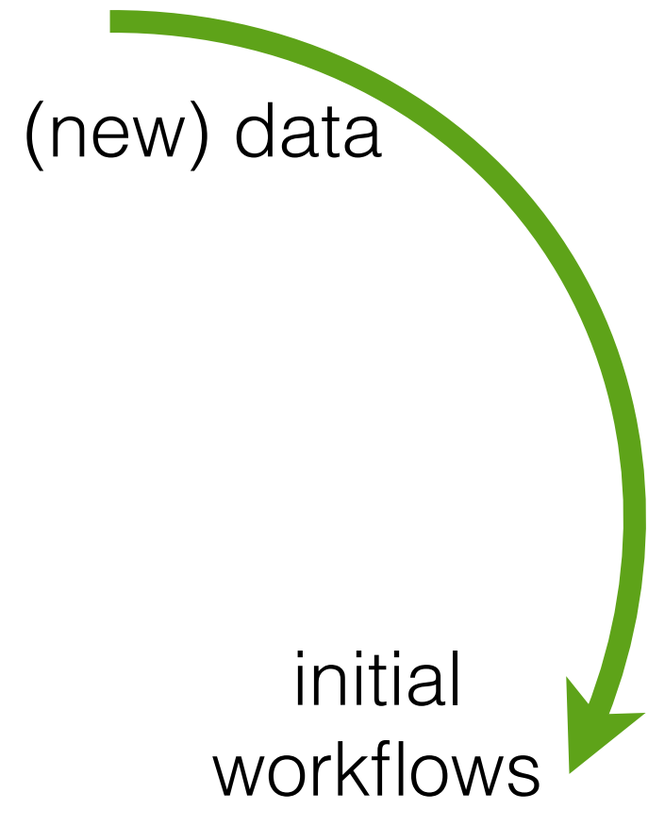
Outlier detection bot: recommends/runs outlier detection techniques

Robot assistants: model selection



Robot assistants: model selection

Random Bot: runs random search given a hyperparameter space



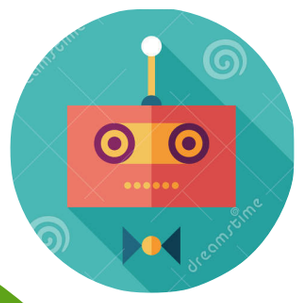
Robot assistants: model selection

Random Bot: runs random search given a hyperparameter space

Greedy Bot: learns key algorithms, hyper-parameters, ranges. Tries those first.

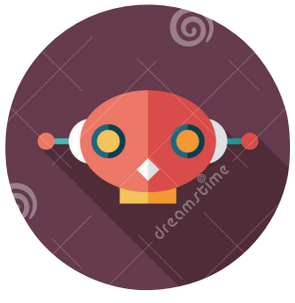
Robot assistants: model selection

Random Bot: runs random search given a hyperparameter space



Greedy Bot: learns key algorithms, hyper-parameters, ranges. Tries those first.

(new) data

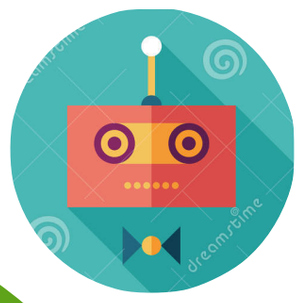


Optimization bots: runs advanced hyperparameter optimization

initial workflows

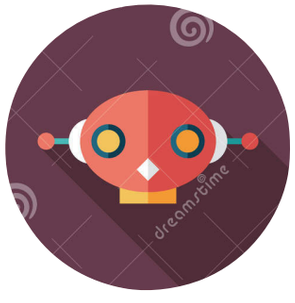
Robot assistants: model selection

Random Bot: runs random search given a hyperparameter space



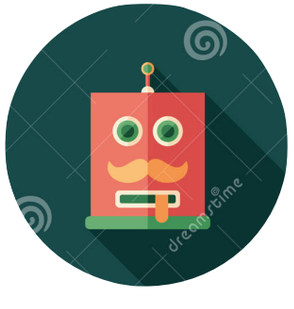
Greedy Bot: learns key algorithms, hyper-parameters, ranges. Tries those first.

(new) data



Optimization bots: runs advanced hyperparameter optimization

initial workflows



Workflow bot: build ML workflows, in collaboration with other bots

Random Bot running on OpenML

People Search

OpenML_Bot R

Joined 2017-03-07

Activity	Reach	Impact	Uploads			
98058.5	0	0	0	8	0	98050

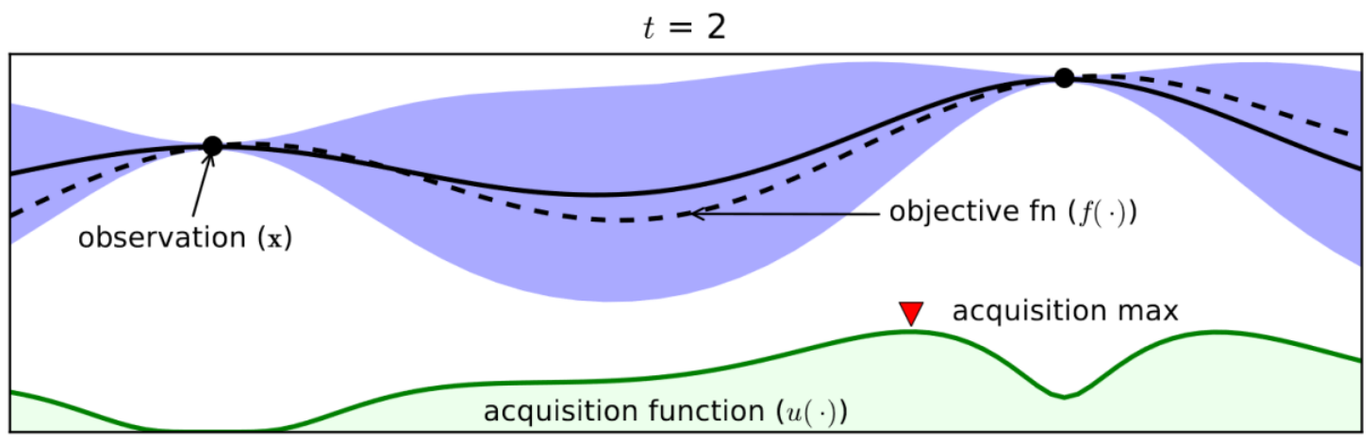
EDIT PROFILE

	Activity	Reach		Impact
Data Sets	0	0	0	0
Flows	8	0	0	0
Tasks	0	0	0	
Runs	98050	0	0	

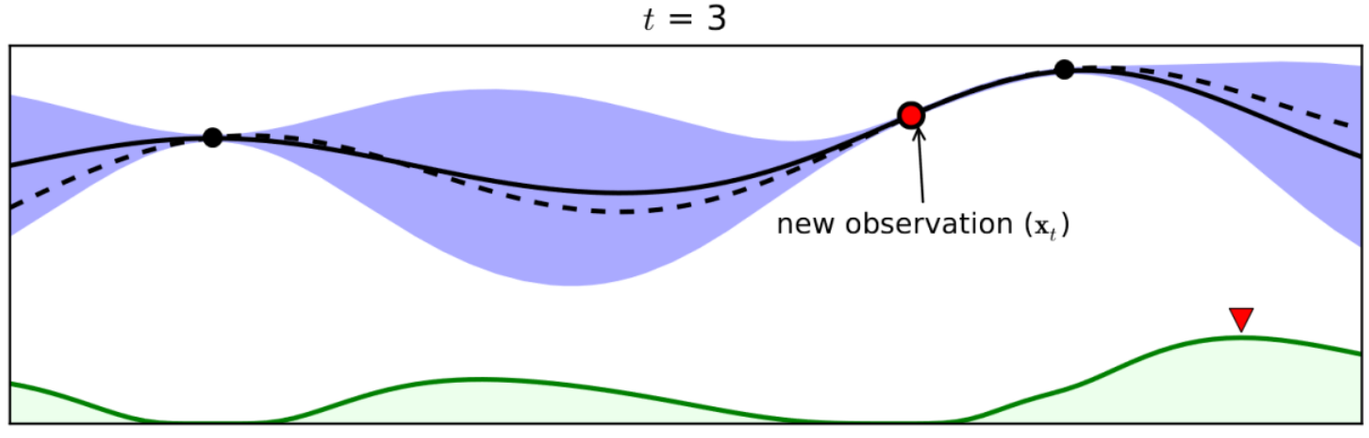
Optimization + metalearning

Include meta-data from prior datasets

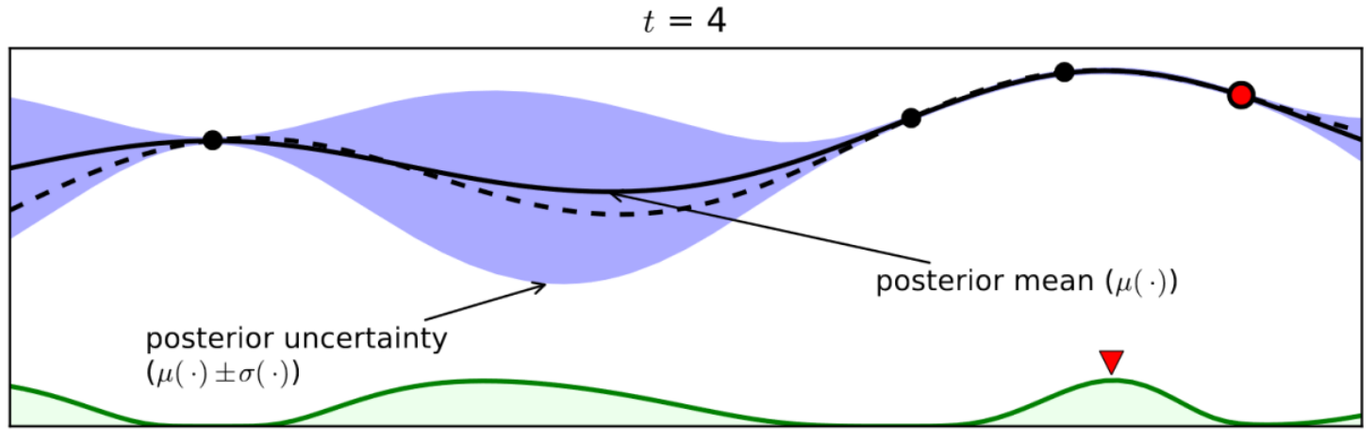
- Warm start: initialize search with promising configurations (*AutoML challenge winner*)



- Surrogate models with prior (focus on best parameters, ranges)

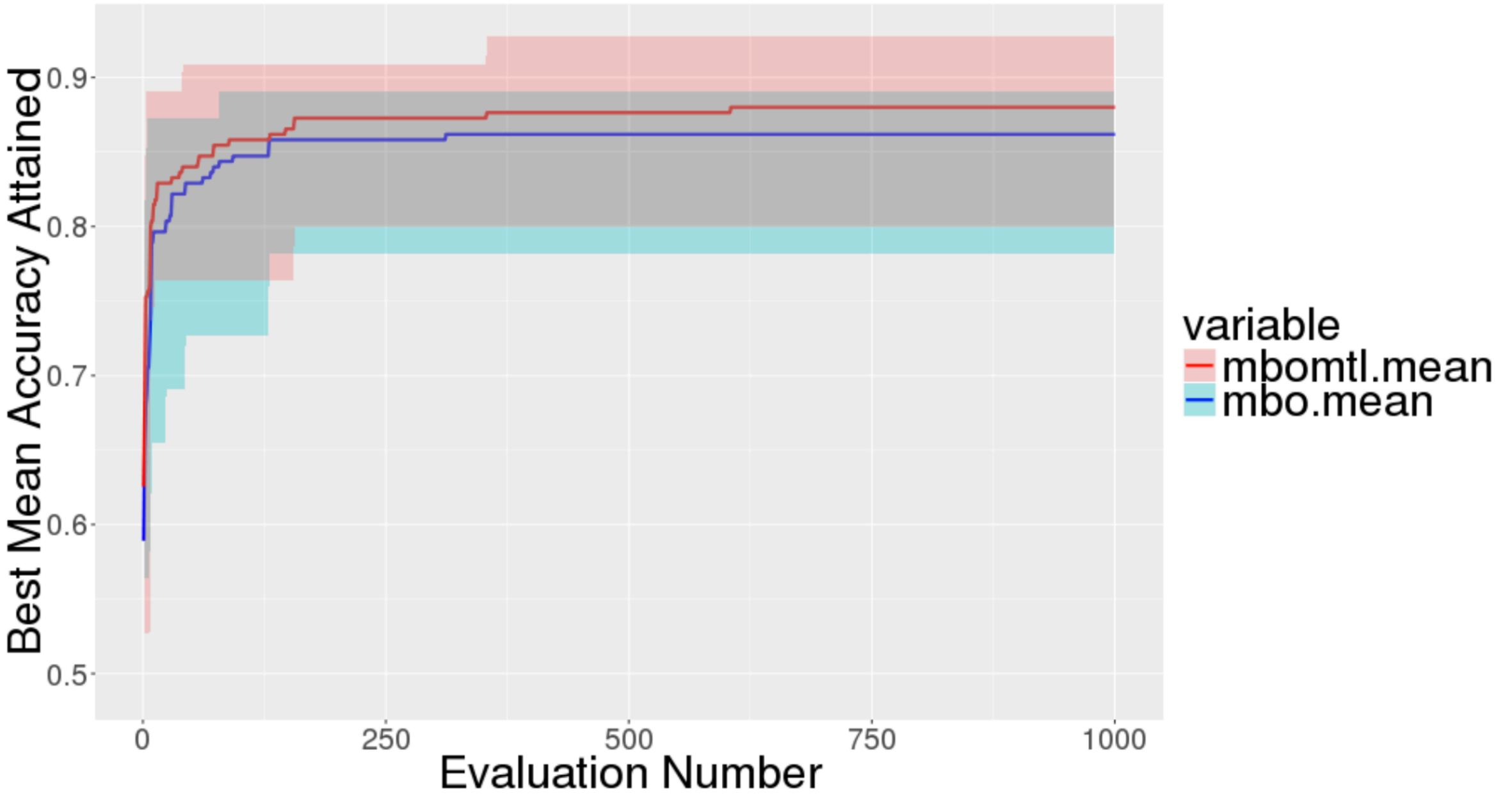


- Acquisition functions based on meta-models (predict performance, trained on prior datasets)



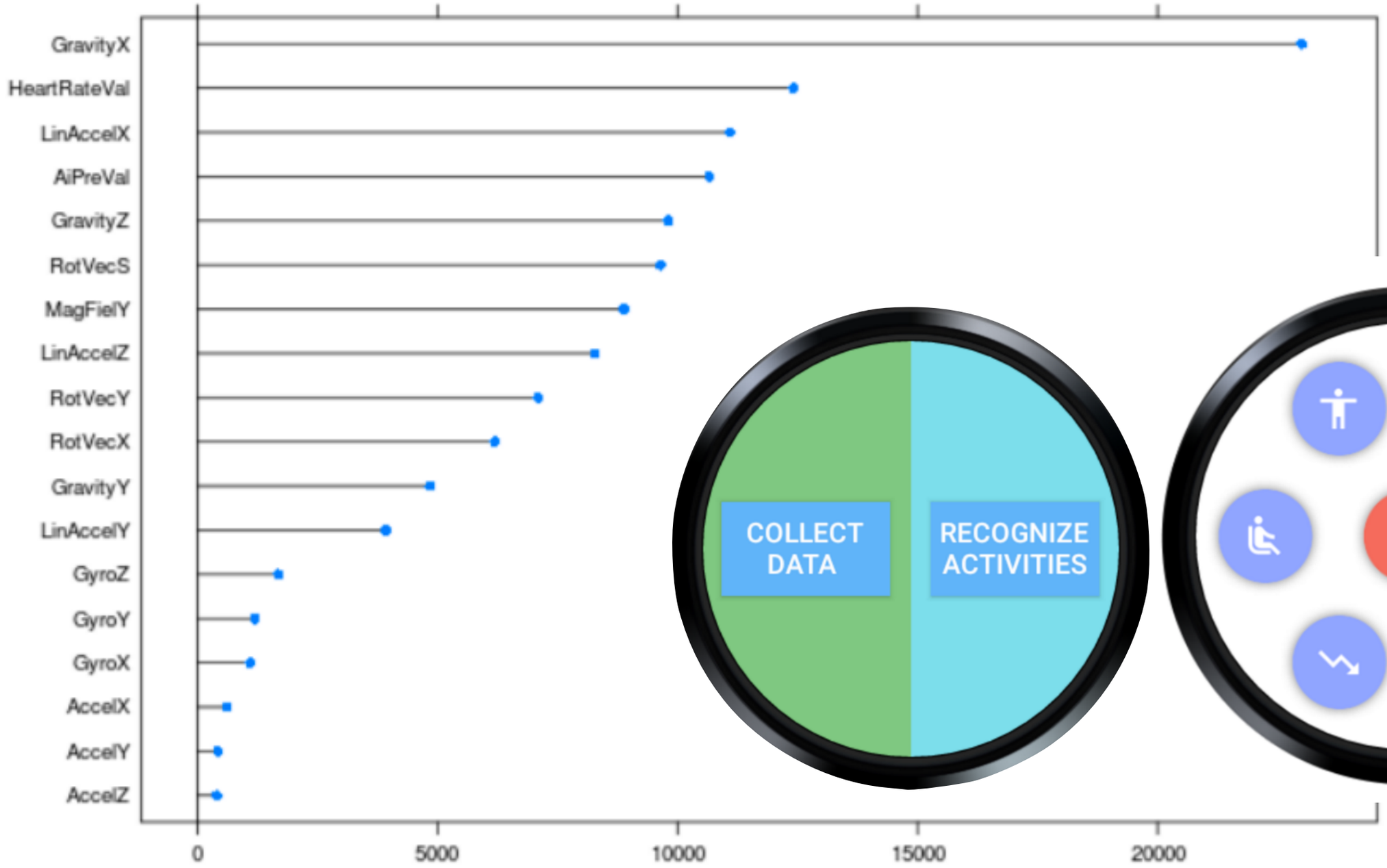
Optimization + metalearning

- Acquisition functions based on meta-models



Frugal learning

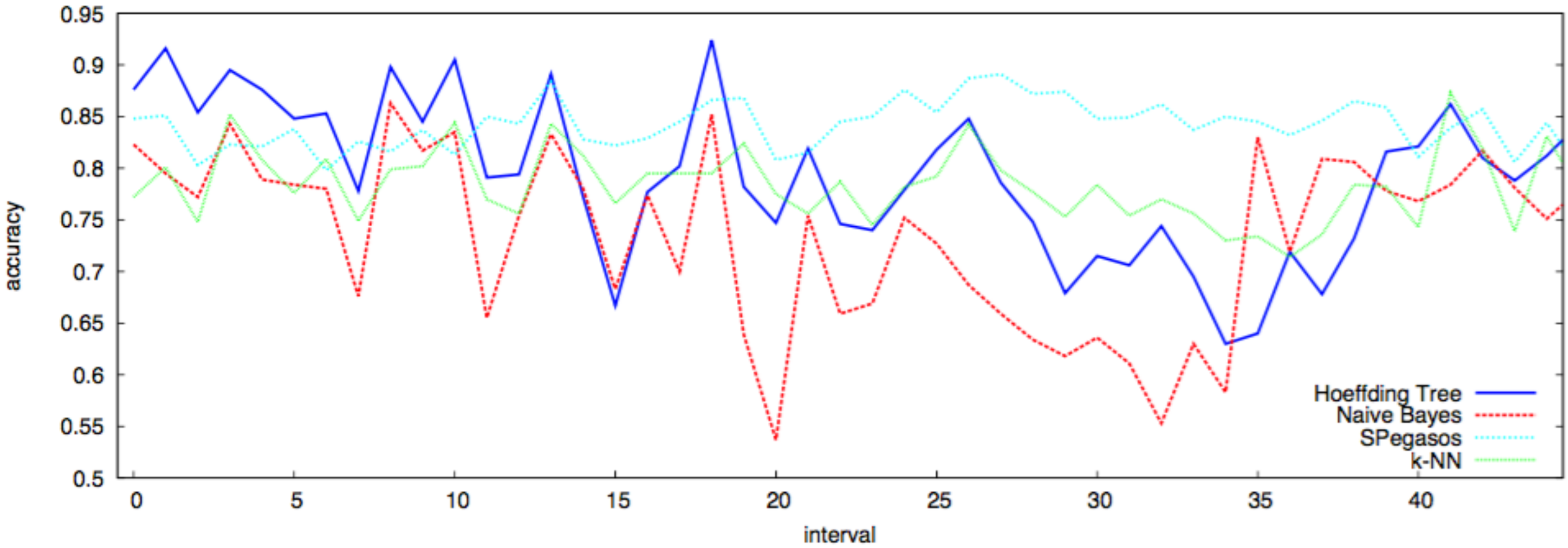
- Can we run ML on wearable devices instead of transmitting data?
- Which algorithms are very fast and highly accurate?
- Best algorithms implemented on smartwatch for activity recognition



Meta-learning on streams

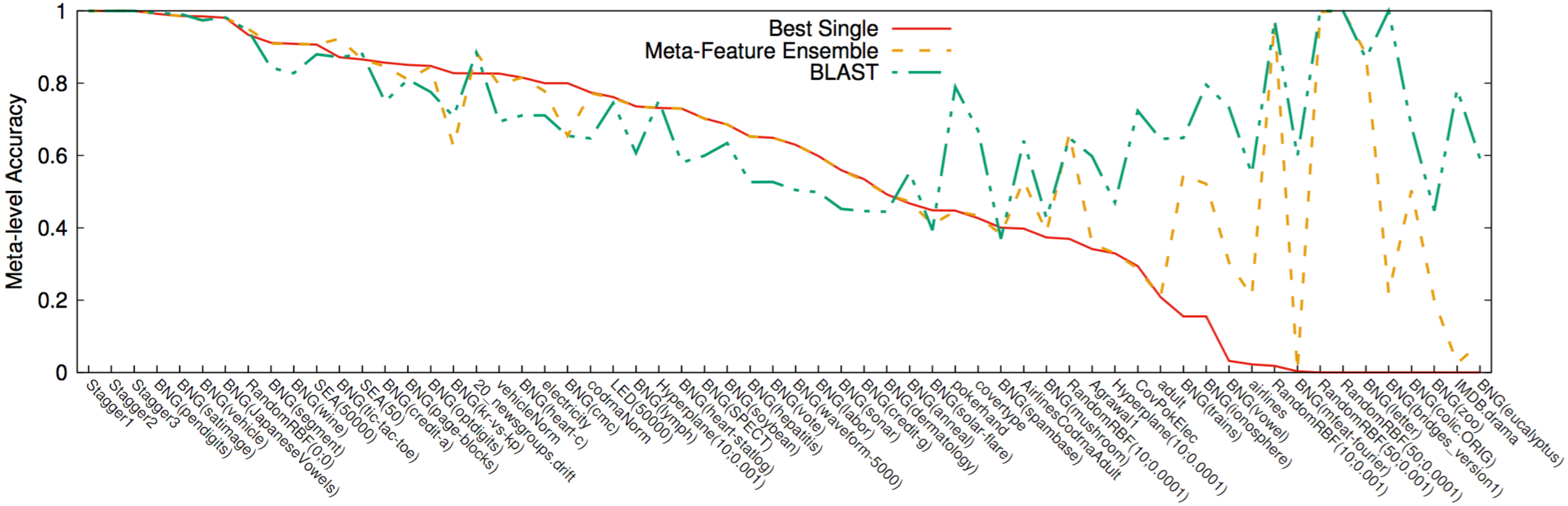
Stream data in OpenML: 'best' algorithm changes over time

Concept drift



- Use meta-learning to select the best models at each point in time

Meta-learning on streams



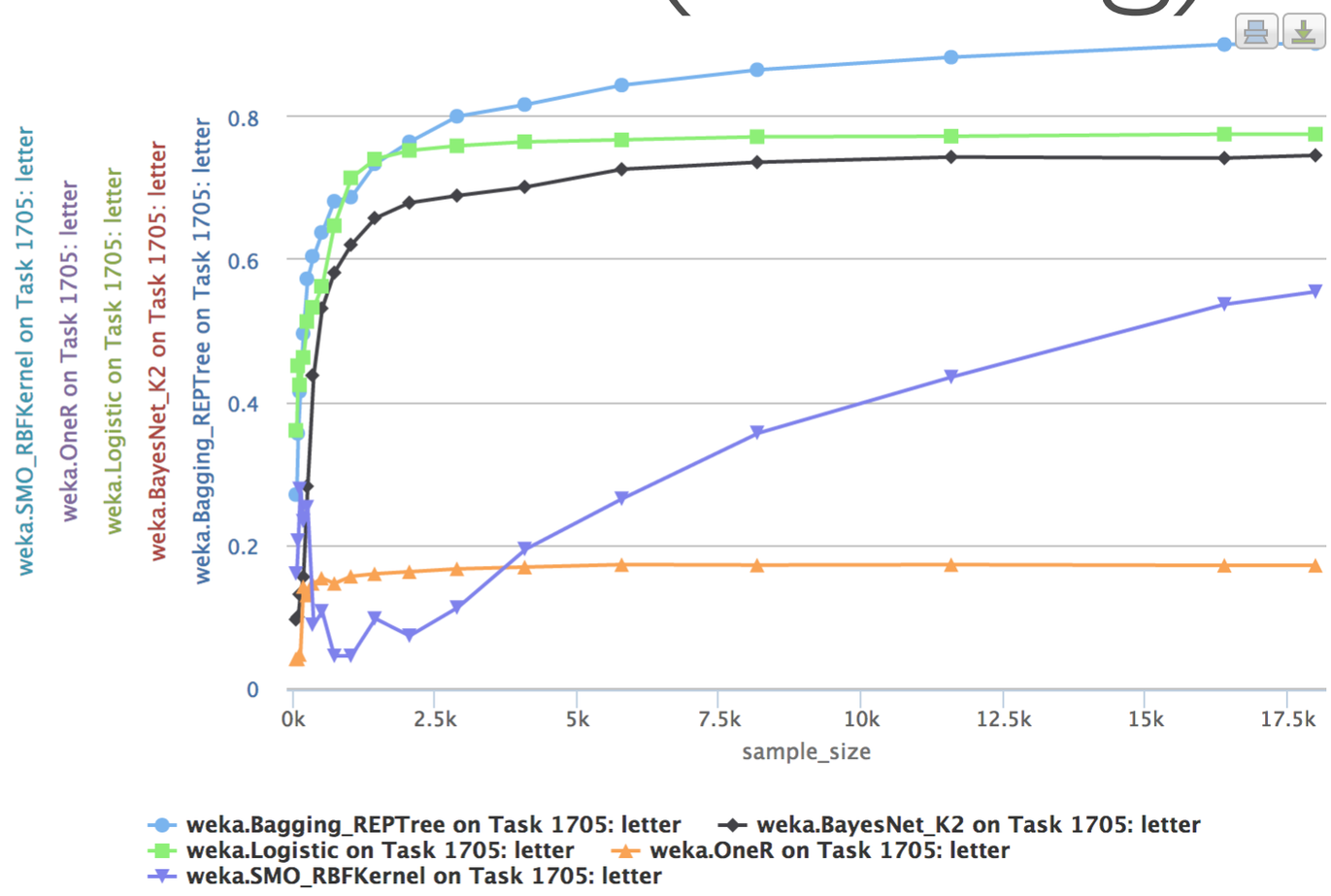
- Streaming ensembles
  - Train multiple models, use meta-learner to weight the votes of all learners for the next window
- BLast (Best-Last)
  - Choose models that performed best in previous window. Equivalent to state-of-the-art, but much faster.

J. van Rijn et al. ICDM 2015

Fast algorithm selection (ranking)

Learning curves in OpenML

For new dataset: build partial learning curves up to T (e.g. 256 instances)
Use learning curves to compute dataset similarity



- Start with strong learner a_{best}
- Choose and evaluate competitor algorithm $a_{competitor}$
- Identify k nearest prior datasets by distance between partial curves:

$$dist(d_i, d_j, a_p, a_q, T) = \sum_{t=1}^{T}(P_{p,i,s_t} - P_{p,j,s_t})^2 + \sum_{t=1}^{T}(P_{q,i,s_t} - P_{q,j,s_t})^2$$

- Choose $a_{competitor}$ that beats a_{best} on most similar prior datasets

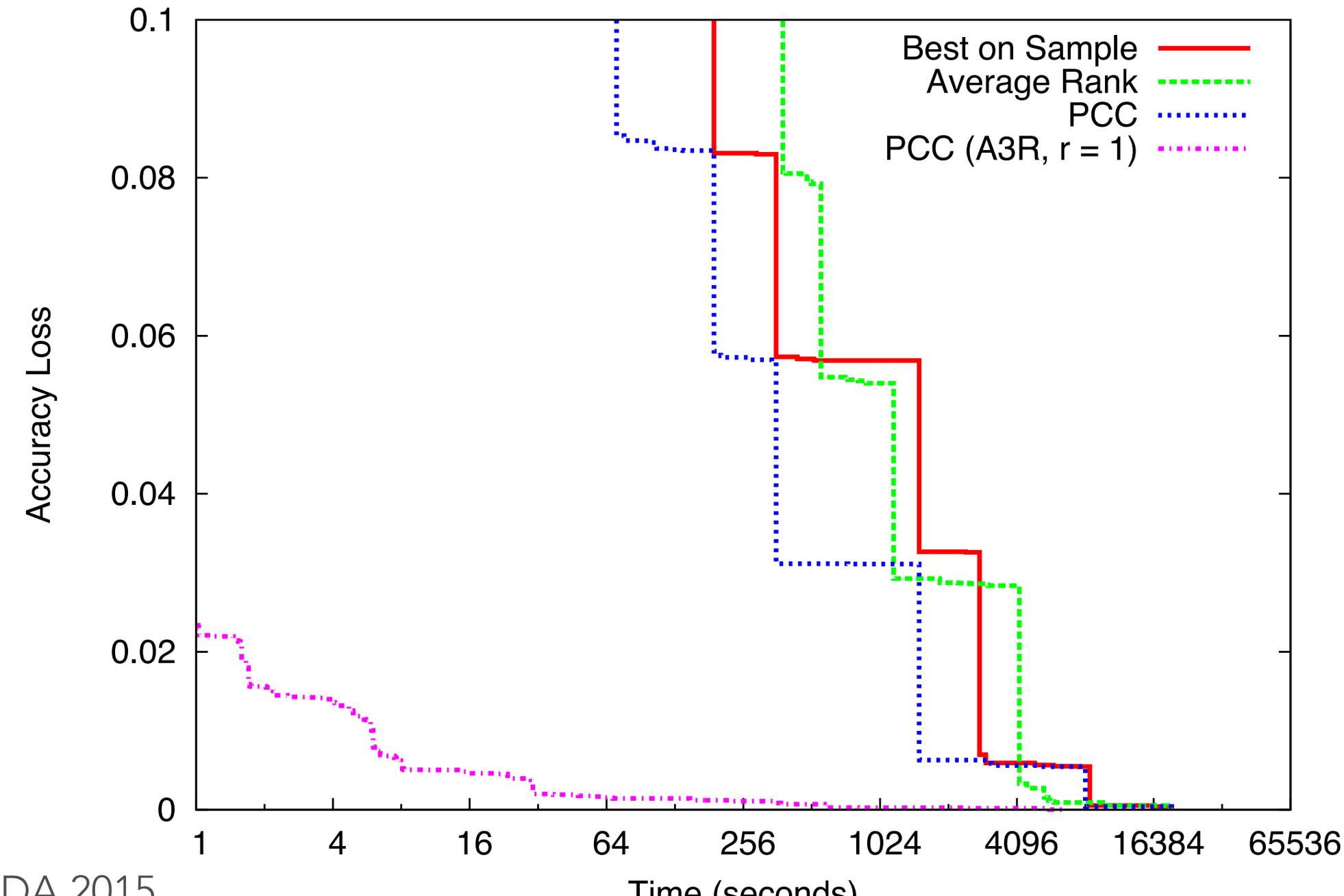
J. van Rijn et al. IDA 2015

Fast algorithm selection

Pairwise Curve Comparison (PCC), with multi-objective measure A3R

- Trade-off high expected performance and fast training time

Results for 53 classifiers on 39 datasets



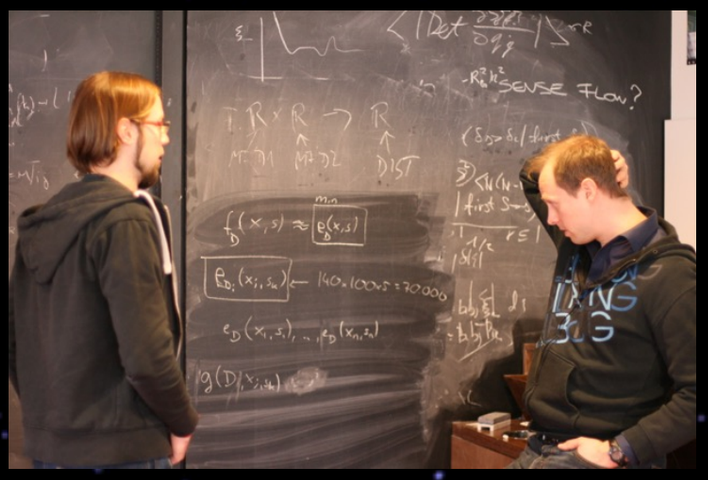
J. van Rijn et al. IDA 2015

Join Us!

www.openml.org

Join our hackathons

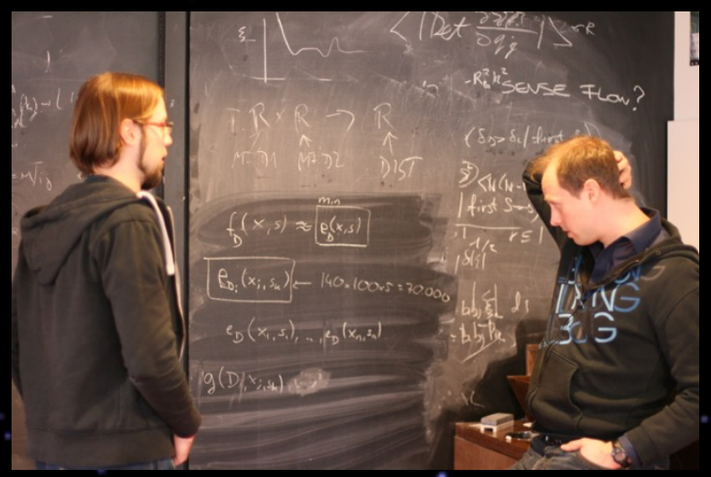
- June, Heidelberg
- Oct 9, Leiden



Help us :)

We are always looking for:

- Code contributions (open source)
  - New tool/platform integrations
    - E.g. Keras/TensorFlow
  - New bots
  - Your own ideas
- Interesting datasets
- Computing resources (!)
- Funding ideas



@open_ml

OpenML

Thank You