#OpenML

openml.org

OpenML

COLLABORATIVE MACHINE LEARNING

BERND BISCHL - JAKOB BOSSEK - GIUSEPPE CASALICCHIO - BENJAMIN HOFNER - PASCAL KERSCHKE - DOMINIK KIRCHHOFF - MICHEL LANG - HEIDI SEIBOLD - **JOAQUIN VANSCHOREN**

OpenML beta

Exploring machine learning better, together

19590 data sets

Find or add **data** to analyse

46701 tasks

Download or create scientific **tasks**

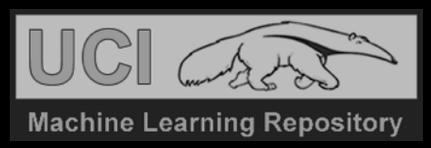
2691 flows

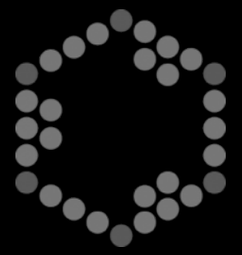
Find or add data analysis **flows**

454945 runs

Upload and explore all **results** online.

Download and share data, flows and runs through:



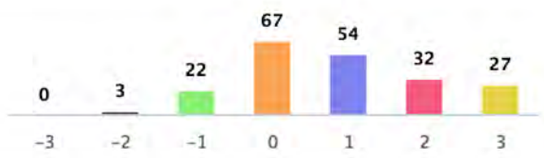
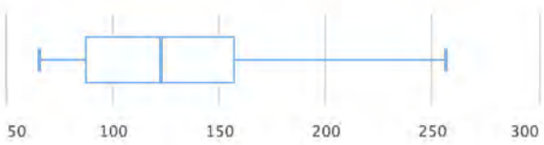
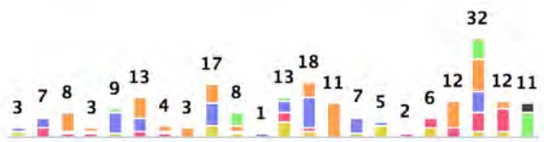


Data (ARFF) uploaded or referenced, versioned

analyzed, characterized, organized online

analyzed, characterized, organized 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

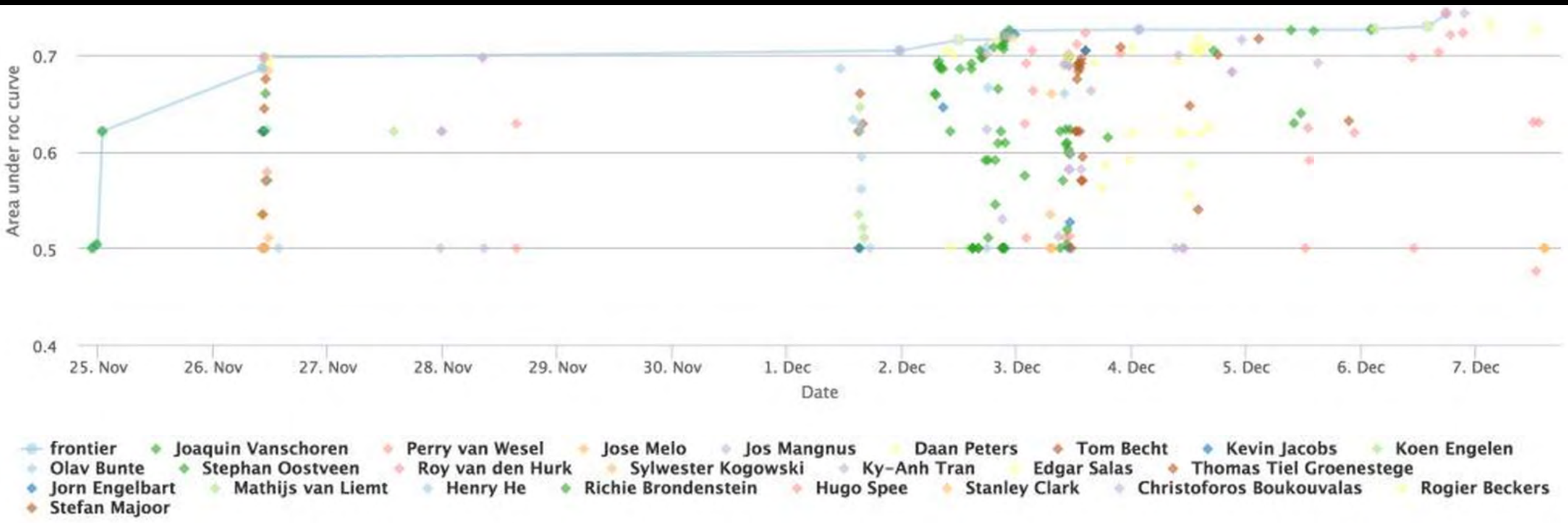
DefaultAccuracy	**0.33**	The predictive a
NumberOfClasses	**7**	The number of c
NumberOfFeatures	**26**	The number of f
NumberOfInstances	**205**	The number of i
NumberOfMissingVal...	**59**	Counts the total

Tasks contain data, goals, procedures.

Readable by tools, automates experimentation

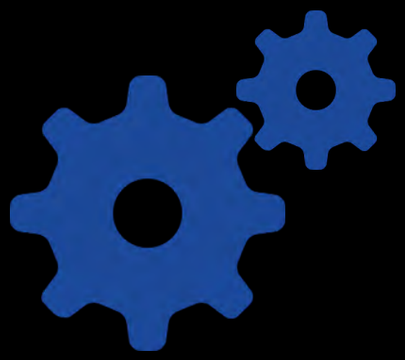
All results organized online: **realtime overview**

All results organized online: **realtime overview**

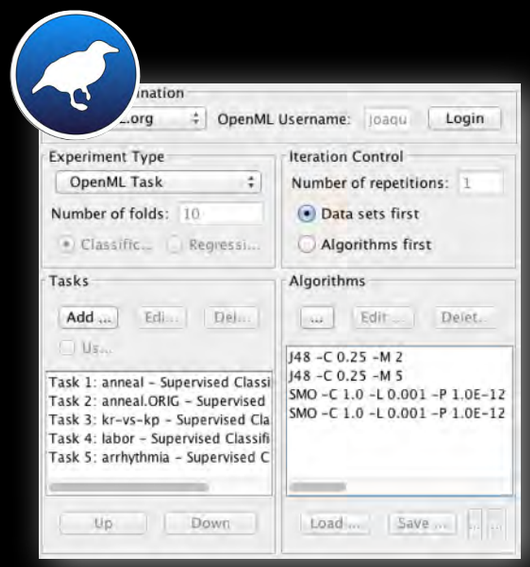


Flows (code) run locally, auto-registered by tools

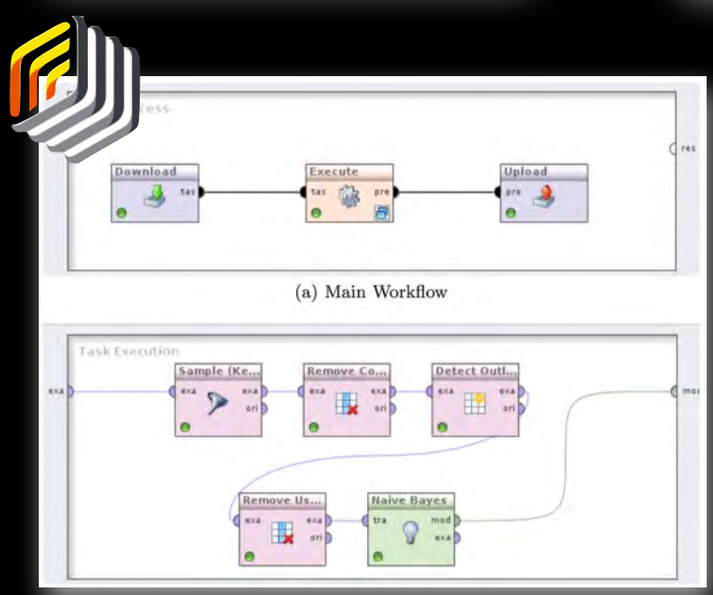
Integrations + APIs (REST, R, Python, Java,…)



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

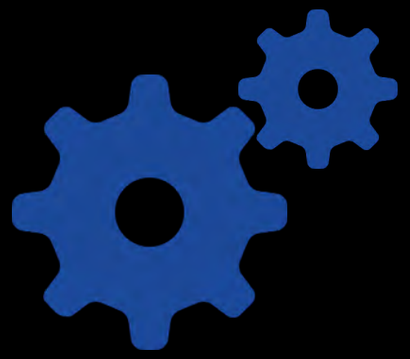


```
from sklearn import tree
from openml import tasks,runs
task = tasks.get_task(14951)
clf = tree.DecisionTreeClassifier()
run = runs.run_task(task, clf)
return_code, response = run.publish()
```



(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,...)

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

Experiments auto-uploaded, evaluated online

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

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.

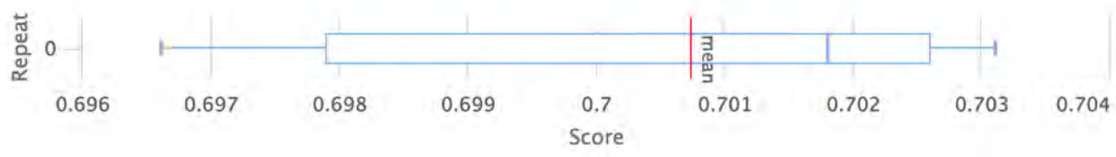
Area under ROC curve

0.7007 ± 0.0023

Per class

0	1
0.7007	0.7007

Cross-validation details (10-fold Crossvalidation)



Join OpenML

- Open Source, on GitHub
- Regular hackathons