

OpenML Cheat Sheet (Python)

config

Find your API key (required for uploads):

- www.openml.org > Your profile > API Authentication

Main OpenML servers:

- Public: <https://www.openml.org/api/v1> (default)
- Test: <https://test.openml.org/api/v1>

datasets

`list_datasets(offset=None, size=None, tag=None)`

- `offset` and `size` for paging results
- `tag` to filter datasets (e.g. 'uci')
- `status`: active, in_preparation, deactivated
- `data_name`, `data_version`, `number_instances`

`get_dataset(dataset_id)`

- returns **OpenMLDataset** object
- automatically downloads and caches the data itself

OpenMLDataset

- `.features` : list of features and their properties
- `.qualities` : list of all dataset properties
- `.get_data` (target,return_attribute_names=False,return_categorical_indicator=False) returns data as numpy arrays, attribute names, and which are categorical
- `.retrieve_class_labels(target_name='class')` : return all class labels for the given target attribute

Upload new datasets

- Create a new OpenML dataset with all relevant information
- `datasets.functions.create_dataset` for uploading pandas dataframes or numpy arrays
- Call `.publish()` to upload

tasks

`list_tasks(task_type_id=None, offset=None, size=None, tag=None)`

- `offset` and `size` for paging results, `tag` to filter tags
- `task_type_id`: 1=Classification, 2=Regression,...
- Task IDs do not match dataset IDs

`get_task(task_id)`

- returns **OpenMLTask** object
 - includes estimation procedure, target name, cost matrix,...
- automatically caches the task description

OpenMLTask

- `.get_dataset()` : downloads associated dataset
- `.download_split()` : downloads train/test splits

```
# General imports
from openml import datasets, tasks, runs, flows, config, evaluate
import os, pandas, sklearn, arff, pprint, numpy, seaborn
```

Set server, API key and cache directory (default: `~/openml/cache`)

```
config.apikey = 'qxlpbe...ebairtd'
config.server = 'https://...'
config.set_cache_directory(os.path.expanduser('~/.mycache'))
```

Or, create a config file called `~/openml/config` and add these lines:

```
server=https://www.openml.org/api/v1
apikey=qxlpbeaudtprb23985hcqlfoebairtd
cachedir=/homedir/.openml/cache
```

```
dlist = datasets.list_datasets(size=100)
pandas.DataFrame.from_dict(dlist, orient='index')[
    ['name', 'NumberOfInstances', 'NumberOfFeatures'][:3]
```

	name	NumberOfInstances	NumberOfFeatures
2	anneal	898	39
3	kr-vs-kp	3196	37
4	labor	57	17

```
odata = datasets.get_dataset(1471)
print(odata.name, "Target: "+ odata.default_target_attribute,
      odata.description[260:308], sep='\n')
```

```
eeg-eye-state
Target: Class
All data is from one continuous EEG measurement
```

```
X, y, attribute_names = odata.get_data(
    target=odata.default_target_attribute,
    return_attribute_names=True)
pandas.DataFrame(X, columns=attribute_names)[:2]
```

	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

```
md = datasets.OpenMLDataset(data_file='dataset.arff', name='t',
    description='t', version='1', format='ARFF', licence='CC0',
    visibility='public', default_target_attribute='class')
data_id = md.publish()

print("New dataset ID: " + str(data_id))
```

New dataset ID: 6677

```
tlist = tasks.list_tasks(task_type_id=1, size=100)
pandas.DataFrame.from_dict(tlist, orient='index')[
    ['name', 'estimation_procedure'][:3]
```

	name	estimation_procedure
2	anneal	10-fold Crossvalidation
3	kr-vs-kp	10-fold Crossvalidation
4	labor	10-fold Crossvalidation

```
task = tasks.get_task(14951)
pprint.pprint(task.estimation_procedure)
```

```
{'data_splits_url': 'https://www.openml.org/api_spl
Task_14951_splits.arff',
 'parameters': {'number_folds': '10',
                'number_repeats': '1',
                'percentage': '',
                'stratified_sampling': 'true'},
 'type': 'crossvalidation'}
```

Create new tasks

...
Under development

