Applied Machine Learning

Data Access:
mlr3oml package

Learning goals

- Access OpenML from within R
- Download datasets and tasks
- Convert OpenML objects to mlr3
- Use benchmark suites and collections

WHAT IS MLR3OML?

- Benchmark experiments require high-quality data for meaningful conclusions.
- **OpenML.org**: Open-source platform for sharing entities of ML experiments.
- `mlr3oml` is an R package providing an interface to OpenML (via REST API).
  ⇒ Query, download, and publish data, tasks, and task collections.
  ⇒ Experiment results of others can also be queried and downloaded.

API Key Usage in `mlr3oml`:

- Download operations work without an API key, but may be rate limited.
- Uploading to OpenML requires an API key.
- Obtain API key by creating an account on `OpenML.org` and specify in R:
  - Set environment variable `OPENMLAPIKEY`
  - Use R option `mlr3oml.api_key` (takes precedence), e.g.:
    `options(mlr3oml.api_key="c1994bdb7ecb3c6f3c8f3b35f4b47f1f")`

OVERVIEW MLR3OML

- `mlr3oml` supports five OpenML object types (for downloading):
  - `OMLData`: Represents datasets.
  - `OMLTask`: Represents tasks (problem specifications).
  - `OMLFlow`: Represents machine learning flows (pipelines).
  - `OMLRun`: Represents the execution of flows on tasks.
  - `OMLCollection`: Represents collections of tasks or runs.

  $\Rightarrow$ Each object can be converted to its corresponding `mlr3` object.
- Find OpenML objects using `list_oml_*()` functions
- Upload datasets, create tasks, and collections (requires API key):
  - `publish_data()` - Upload dataset
  - `publish_task()` - Create task
  - `publish_collection()` - Create collection

LISTING DATA

Example of filtering datasets by properties:

```
library(mlr3oml)
odatasets = list_oml_data(
  number_features = c(10, 20),
  number_instances = c(45000, 50000),
  number_classes = 2
)
odatasets[1:5, c(1,2,9)]
##    data_id                          name NumberOfFeatures
##      <int>                        <char>            <int>
## 1:     179                         adult                15
## 2:    1461                bank-marketing                17
## 3:    1590                         adult                15
## 4:   43898                         adult                15
## 5:   44234 Bank_marketing_data_set_UCI                17
```

DOWNLOADING DATA

- Download metadata with `odt(id = 1590)` or `OMLData$new(id = 1590)`.
- Query metadata (number of rows, columns, etc.) without loading the entire data:

```
odata = odt(id = 1590)
class(odata)
## [1] "OMLData"   "OMLObject" "R6"
odata$nrow
## [1] 48842
odata$ncol
## [1] 15
```

- Download and store data by accessing the `$data` field:

```
odata$data[1:5, 1:5]
##      age workclass fnlwgt    education education.num
##    <int>    <fctr>  <int>       <fctr>         <int>
## 1:    25   Private 226802         11th             7
## 2:    38   Private  89814      HS-grad             9
## 3:    28 Local-gov 336951   Assoc-acdm            12
## 4:    44   Private 160323 Some-college            10
## 5:    18      <NA> 103497 Some-college            10
```

CONVERT DATA TO MLR3 TASKS

- `mlr3oml` Cache:
  - Data is cached in memory after first access.
  - Option to cache permanently by setting `options(mlr3oml.cache = tempfile())`.
- Convert data to `mlr3` tasks for seamless integration:

```
library(mlr3)
tsk_adult = as_task_classif(odata$data, target = "class")
tsk_adult
## <TaskClassif:odata$data> (48842 x 15)
## * Target: class
## * Properties: twoclass
## * Features (14):
##   - fct (8): education, marital.status, native.country, occupation, race,
##     relationship, sex, workclass
##   - int (6): age, capital.gain, capital.loss, education.num, fnlwgt,
##     hours.per.week
```

LISTING TASKS

- OpenML tasks specify target variable, train-test splits, etc.
- Example of filtering tasks:

```
adult_tasks = list_oml_tasks(data_id = 1590)
adult_tasks[task_type == "Supervised Classification", ]
##    task_id                 task_type data_id
##      <int>                    <char>   <int>
## 1:    7592 Supervised Classification    1590
## 2:   14947 Supervised Classification    1590
## 3:  126025 Supervised Classification    1590
## 4:  146154 Supervised Classification    1590
## 5:  146598 Supervised Classification    1590
## 6:  168878 Supervised Classification    1590
## 7:  233099 Supervised Classification    1590
## 8:  359983 Supervised Classification    1590
## 9:  361515 Supervised Classification    1590
```

DOWNLOADING TASKS AND CONVERT TO MLR3

- Load task with ID 359983 and examine data and splits:

```
otask = otsk(id = 359983) # alternative: OMLTask$new(id = 359983)
otask$data # downloads the data
otask$task_splits # downloads the resampling information
##           type rowid repeat.  fold
##         <fctr> <int>   <int> <int>
##      1:  TRAIN 32427       0     0
##      2:  TRAIN 13077       0     0
##     ---
## 488419:   TEST 25263       0     9
## 488420:   TEST 43381       0     9
```

- Convert to `mlr3`:

```
as_task(otask) # creates mlr3 task
as_resampling(otask) # creates mlr3 resampling object
## <ResamplingCustom>: Custom Splits
## * Iterations: 10
## * Instantiated: TRUE
## * Parameters: list()
```

TASK COLLECTIONS AND BENCHMARK SUITES

- Bundle tasks for benchmark suites, e.g., CC-18 benchmark suite with ID 99:

```
otask_collection = ocl(id = 99)
otask_collection$task_ids[1:5]
```

- Downloads and defines tasks and resamplings from task collection:

```
tasks = as_tasks(otask_collection)
resamplings = as_resamplings(otask_collection)
```

- Example to obtain only a subset from the collection:

```
binary_cc18 = list_oml_tasks(
  limit = 6,
  task_id = otask_collection$task_ids,
  number_classes = 2
)
otasks = lapply(binary_cc18$task_id, otsk)
tasks = as_tasks(otasks)
resamplings = as_resamplings(otasks)
```