ISEL \ DEETC

MEIM | MEIC

AMD (MLDM – Machine Learning and Data Mining) – Module of Practice

Paulo Trigo Silva

1. Install and test "Orange Data Mining Framework" and "Python"

The software to install (Windows/OSX) is available in "distribution_02_OrangeDM.zip" (cf., "moodle", folder "`SoftwareDistributions`"). Nevertheless, steps to download/install Orange are as follows.

a) If you have a **64bit** architecture download directly from the Orange site:

`http://orange.biolab.si/download/`
in Windows, choose: "`Orange3-3.23.0-Miniconda-x86_64.exe`".
in OSX, choose: "`Orange3-3.23.0.dmg`"

to use (**64bit** version) follow the instructions in "`https://orange.biolab.si/download/`"; for more details see "`https://github.com/biolab/orange3/blob/master/README.md`"

b) **ONLY** If you have a **32bit** architecture (*this was my case!*):

1. download a 32bit older version from:

   `https://download.biolab.si/download/files/`
   "`Orange3-3.21.0-Python36-win32.exe`" seems to be the last 32bit version.

2. this Orange version version (3.21.0) is bundled with Python3.6.4

   this Orange version installs the framework within a "python virtual environment"; for more information on "virtual environment": `https://virtualenv.pypa.io/en/stable/`.
   So, you will find the installation in "(…)\Orange" folder.
   note: there are useful example datasets in "(...)\Orange\Lib\site-packages\Orange\datasets"

3. test if your installation can properly use the Orange libraries:

   open a new "Command Prompt" window
   change current folder to the installation; cd c:\(…)\Orange
   initiate the Orange virtual environment; execute "Orange Command Prompt" file
   launch Python interpreter (i.e., write python):

   ```
   >>> import Orange
   >>>
   ```

   To quit the Python environment write: `quit()`

   In case you still problems with the Orange installation visit:
   `https://datascience.stackexchange.com/tags/orange/info/`

4. follow the instructions in "**`_USAGE_virtual-environment-for-python.txt`**" file regarding the usage of "Python Virtual Environment:

2. Read the dataset using “Orange Canvas” – the graphical mode

a) Copy the file, “dataset_3RowHeader.txt”, built in the previous practical class to this class’s folder.

b) Execute the “Orange Canvas” application: “\Start\Orange3\Orange Canvas”.

c) Select the “Data” separator. Drag the “File” icon into the white part of the canvas.

d) Make a “double-click” in the “File” icon and choose the “Data File” to point to the folder that contains the “dataset_3RowHeader.txt” file. Notice that the file is not available for selection! To fix the problem change the file extension “.txt” to “.tab” and select the file.

e) Take a look at the file “dataset_finalFormat.tab” provided with this practical lesson. If necessary adapt the script developed in previous practical lesson in order to generated a file with this format.

f) Drag the “Data Table” icon into the white part of the canvas; “double-click” on the icon and notice the “info \ no data on input”.

g) Make a connection between “File” and “Data Table” icons. To connect two processes, represented by icon A and icon B, select a connector on the right side (output pipe) of A and drag it into a left connector of B (input pipe). A connection is a data flow from an output pipe into an input pipe.

h) See the “Data Table” output. Connect the “Data Table” process to a “Save Data” process.

i) Point to the connector line and “double-click” (or “right-click” and choose “Reset Signals”). Notice that “Data Table” process has two output available ports; explore deleting established connection and activating the other one.

j) Configure the “Save” process so that it generates a “.csv” (“comma separated values”) after some data transformations (e.g., via a pipeline of the processes explored in the previous items).

k) Use Excel to read the file that was generated in the previous item.

l) Explore the “Select Columns” process.

m) Explore the “Select Rows” process.

n) Explore the “Purge Domain” process.

o) Explore the “Discretize” process.

p) Save your work (\File \ Save As…) into an “Orange Schema” so that it can be used again later.

3. Reuse an “Orange Schema”

a) Open the “`z01_schema.ows`”; double-click on the file or open it within the ”Orange Canvas”.

b) Explore the schema and make sure that the “*Save (z_out_xx)*” widgets generate “.tab” files with the same information as in the provided “`z_out_xx.txt`” files. *Suggestion*: explore the options available in each widget and each link (right-click over the link and “Reset Signals”).

4. Start working with “`lenses`” dataset in Orange – graphic mode

Consider the dataset: “`lenses.tab`” (in folder “`_dataset`”).

a) Execute the “Orange Canvas”, choose the “File” icon and read the dataset.

Note: we may consider an “Orange“ icon as an “operator” of a visual programming language; so, instead of calling “icon” we may call “operator”.

b) Connect the “File” operator with “Data Table” and visualize all the information about the dataset.

5. [before-proceeding] – a Python “crash-course”

Consider the file: “`a00_python_crash_course.py`”.

a) Open the “`a00_python_crash_course.py`” and comment everything. *Hint*: if you are using IDLE, in order to comment a block of lines, select the lines and then select the menu option “\Format\Comment Out Region”.

b) Uncomment each section (e.g., Strings, Numbers, Boolean, Multiple Assignment, No value, etc) at a time and explore the concept e that section. We can search for additional information for example in the “p01_aByteOfPython_v3.pdf” document.

c) In the end make sure that the value computed in the “main” function gets printed!

d) Now, encapsulate each section (e.g., Strings, Numbers, Boolean, etc) in a function and make sure that each of those new functions are called from the “main” function.

6. Read a dataset with Orange – programmatic mode

Consider the file: “`a01_datasetRead.py`”.

e) Open the “`a01_datasetRead.py`” and comment everything except for the first lines of code where a dataset is read. Execute the code. *Hint*: if you are using IDLE, in order to comment a block of lines, select the lines and then select the menu option “\Format\Comment Out Region”.

f) Add to the code a line with `print( dataset )` and execute.

g) Now add a line to print `dataset.domain` and execute.

h) Now add a line to print `dataset.domain.variables` and execute.

i) Now add a line to print `dataset.domain.attributes` and execute.

j) Eliminate the comments on the code that analyses the `dataset.domain.attributes`. Execute and explain the difference from the `dataset.domain.variables` structure.

k) Eliminate the remaining comments, execute and explain the result.

7. The (second) “Kick-Off” of Final Project A

a) Consider “final project A” (cf., moodle, “Final Project” folder). Develop “Project Item: 5”.

b) Use the Orange Canvas to create a process that loads and (eventually) applies transformations to the dataset exported from PostgreSQL database. Save your work as an “Orange schema”.

c) Create a Python automatism to generate the dataset. *Suggestion*: extend the "`_goPy.py`" script provided in the previous practical class (cf. folder "`scripts`").

d) Create a Python application to extract the metadata (features and class attributes along with the corresponding domains and values in the dataset) that is necessary to implement the 1R method.