ISEL \ DEETC

MEIM – MEIC

AMD / MLDM (Machine Learning and Data Mining**)**

Paulo Trigo Silva

1. Case study: “the e-commerce”

Consider the problem described in the “Final Project B”.

In this practical class we will develop some tools to assist the “market basket analysis” and to automatically generate the proper “`.basket`” format for the Orange algorithms’ processing.

The tools will be implemented using exclusively PostgreSQL capabilities and Python.

a) Take a look at the “`z_dataset_sample.txt`” and notice that it contains events, taken from the “Final Project B” dataset, recorded during the first hour of the first day of January the 2012.

2. Create a database and populate it with the e-commerce dataset

a) Execute the “`00_script_CREATE_DB.txt`”, to create “`db_e_commerce_sample`” database.

b) Complete the “`01_script_CREATE_SCHEMA.txt`”, to create the “`track`” table that will be populated with the dataset (cf., “`z_dataset_sample.txt`”).

c) Adjust the “`02_script_IMPORT_POPULATE_SCHEMA.txt`” to import the dataset into the “`track`” table. Use the “pgAdmin” tool to confirm that the dataset was properly imported.

3. Build views to assist dataset analyzes (for the “market basket”)

Consider the file: “`a03_CREATE_VIEW.txt`”.

a) Create the “`v_cookie_session_number_of_events`“ view that aggregates (groups) cookies and sessions and computes the total number of events.

b) Create the “`v_cookie_number_of_sessions`“ view that aggregates cookies and computes the total number of sessions (for each cookie) and also the total sum of events for each session. *Suggestion*: build this view from the previous (“`v_cookie_session_number_of_events`“).

c) Create the “`v_number_of_cookies_number_of_sessions`“ view that aggregates the number of sessions and computes the total number of cookies (visitors) at each session. *Suggestion*: build this view from the previous.

d) Create the “`v_number_of_events_per_session_number_of_cookies`“ view that aggregates the number of events per session and computes the number of cookies (visitors). *Suggestion*: build this view from the “`v_cookie_session_number_of_events`“.

e) Use the file “`z_VIEW_expected.txt`” to test your views against the expected result.

Paulo Trigo Silva

4. Export a basket for the extraction of "association rules"

a) Consider the "`v_export`" view that is implemented at the end of the "`a03_CREATE_VIEW.txt`" file. This is an example of a possible basket that we may now build from the previous views. Remove the comments and implement your own view with the information that you consider the most appropriate to build the "basket". *Suggestion*: for now you may just use the provided view in order to validate the whole process (cf., next items).

b) Consider the "`04_script_EXPORT_DATA.txt`" and guarantee that the information to be exported adheres to the <`TID, PID`> "market basket" problem formulation.

c) If you adopted the provided "`v_export`" view you can validate your result against the expected one by comparing the result with the data in file "`z_dataset_sample_OUT_expected.txt`".

5. … info about the "`.basket`" format and "string normalization"

a) Before proceeding take a look at the "`.basket`" format as describe in "**Basket Format**" section in: "`http://orange.readthedocs.io/en/latest/reference/rst/Orange.data.formats.html`"

b) Also take a look at the "`product_gui`" data in file "`z_dataset_sample_OUT.txt`". Notice that there is some "string normalization" work to do! For example a string (in "`product_gui`") that contains the "=" character may get the "`.basket`" parser to fail. Why?

… also all strings should be converted to the same (lower or upper) case; accents and white spaces eliminated to reduce the possibility of different strings representing the same product. Identify potential problems; recall that this dataset is a small sample from a much larger one.

6. Generate the "`.basket`" format for sparse matrix representation

Consider the file: "`_goPy_transform.py`".

a) Complete the provided file in order to automatically normalize strings and generate "`.basket`" file.

b) You can validate your implementation checking your result against the expected one as provided in "`z_dataset_sample_OUT_expected.basket`". *Note*: be aware that your files may have the same data but following a different order from the one provided in the "`*_expected.*`" files.

7. Build your "market basket analysis" (e.g., with Orange workflows)

a) Open "Orange", select "Options \ Add-ons [x] Orange3-Associate" \ <Ok>. Build the workflows: Data\"File" >> Data\"Data Table" >> Associate\"Frequent Itemset", and Data\"File (1)" >> Data\"Data Table" >> Associate\"Association Rules". Explore the "`*test*.basket`" datasets. Explore "support", "confidence" and "number of rules".

b) For **programmatically implementing** the workflows you may consider the **baseline example** provided in the folder "AssociationRulesExample".