# Exercise 3: Data Mining

*Objective: Discover in the EO products relevant and application-valuable scene structures; perform semantic annotation of selected structures, generate a semantic catalogue for the observed area.*

## 1. Preliminaries

Data Mining refers to the overall interactive and iterative process of discovering useful information in the Sentinel-1/-2 product or a collection of products. The EO product is processed and the extracted image features and metadata are structured in a DB. This is actionable information.

### Active Learning and Semantic annotation

*Active Learning* is a form of supervised machine learning in which the learning algorithm is able to interactively query some information source to obtain the desired outputs at new data point. The key idea behind Active Learning is that a machine learning algorithm can achieve greater accuracy with fewer training labels if it is allowed to choose the data from which it learns. [AL]

The input is the training data sets obtained interactively from the HMI. The training dataset refers to a list of images marked as positive or negative examples. The output is the verification the Active Learning loop sent to HMI and the semantic annotation wrote in the DBMS catalogue.

Active learning methods include Relevance Feedback. The Relevance Feedback supports users to search images of interest in a large repository. The GUI allows automatically ranking the suggested images, which are expected to be grouped in the class of relevance. Visual supported ranking allows enhancing the quality of search results by giving positive and negative examples as right and left click respectively.

During the Active Learning two goals are tried to achieve: 1) learn the targeted image category as accurately and as exhaustively as possible and 2) minimize the number of iterations in the relevance feedback loop.

## 2. Environment requirement

To unzip the front-end tool which needs to be downloaded from the platform, compression software (e.g., winzip, 7-zip) is needed.

To run this exercise, the environment requirements are:

- Java Runtime Environment (JRE)
- Command-line window

## Jupyterlab workspace

In this exercise, a user's workspace contains the following needed Jupyter-notebooks and the frontend GUI tool:

- Jupyter-notebook :
  - *Launch_dmg.ipynb*
  - *Monetdb.ipynb*
- frontend :
  - kdd.zip

## 3. Short introduction of the exercises

The following exercises demonstrate the data mining tool in 2 modes:

1. EO Image Mining; users via the GUI interactively operate an Active Learning module which using EO image features performs functions as search, browse, query for image patches of interest for the user. The discovered relevant structures are semantically annotated and stored in the DB. This is the actual EO image semantics learned adapted to the user conjecture and application.
2. EO Data Mining is performed via SQL multimodal queries. It uses image features, image semantics, and selected EO product metadata.

## Step 1: Run DMG for the image of interest for Sentinel-1

Objective: Find the Sentine-1 product of interest

**Open the jupyter-notebook named *Launch_dmg.ipynb* (Part 1)**

1 Search all the Sentinel-1 data product of type GRD, choose a city from the city list[1] which is prepared in the notebook, choose a time period between 10th of July 2019 and the 20th of July 2019[2]

2 List the products; choose one product path as the input for visualization

3 Visualize the product image to be sure it contains content of interest

4 (optional) Repeat 2-3 until you are satisfied to perform data mining on it

5 Run the data model generation for the interested S-1 product

## Step 2: Run DMG for the corresponding image of interest for Sentinel-2

Objective: Find the corresponding Sentine-2 product of interest

**Open the jupyter-notebook named *Launch_dmg.ipynb* (Part 2)**

1 Search all the Sentinel-2 data product of level 1C, choose a city id from the corresponding S-2 tile id list which is prepared in the notebook, choose the same time period as in Step 1

2 List the products; choose one product path as the input for visualization

3 Visualize the product image to be sure it contains content of interest

4 (optional) Repeat 2-3 until you are satisfied to perform data mining on it

5 Run the data model generation for the interested S-2 product

## Step 3: Download the frontend to your local computer

Objective: prepare the frontend GUI tool

1 Right click on the kdd.zip file in the *frontend* folder, choose *download* to download it to your local laptop

2 Unzip the file and check the folder. It contains:

- etc

[1] Because the corresponding Sentinel-2 tile ids is prepared. So you can perform this exercise for both S-1 and S-2 products for the same area.

[2] This time period can be changed according to your interest, however, it's recommended to choose a short period in order to avoid getting a long list of product paths.

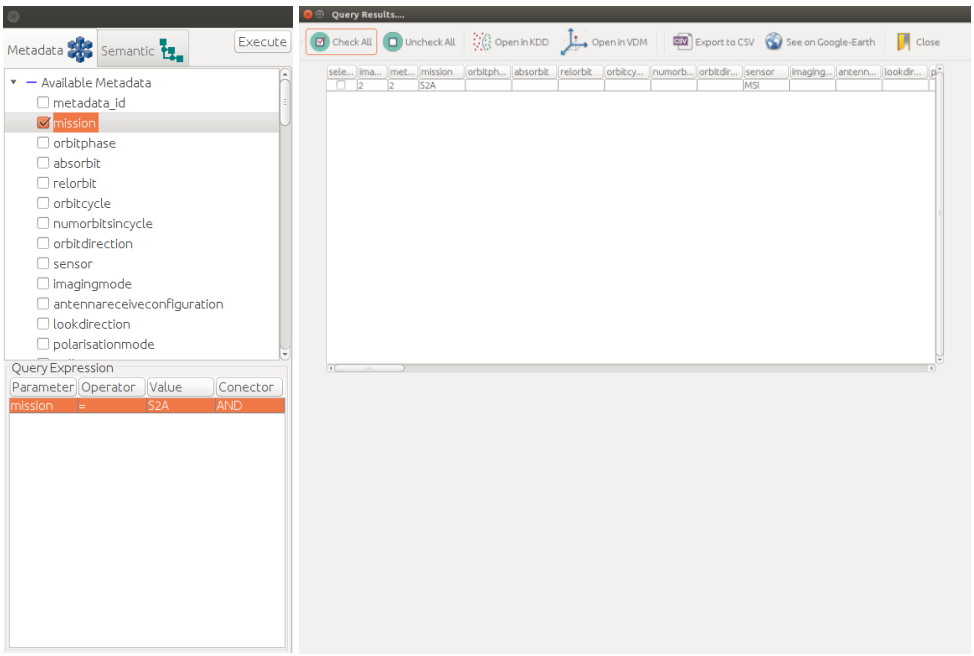
    - kdd-toobar.xml
- lib
    - kdd.jar
    - a list of .jar files
    - icon
        - a list of .png files
- run_kdd.sh

## Step 4: Run the frontend and perform Image Mining

Objective: EO image mining on ingested products which are previously populated in the database

**Run the run_kdd.sh[3] in local terminal**

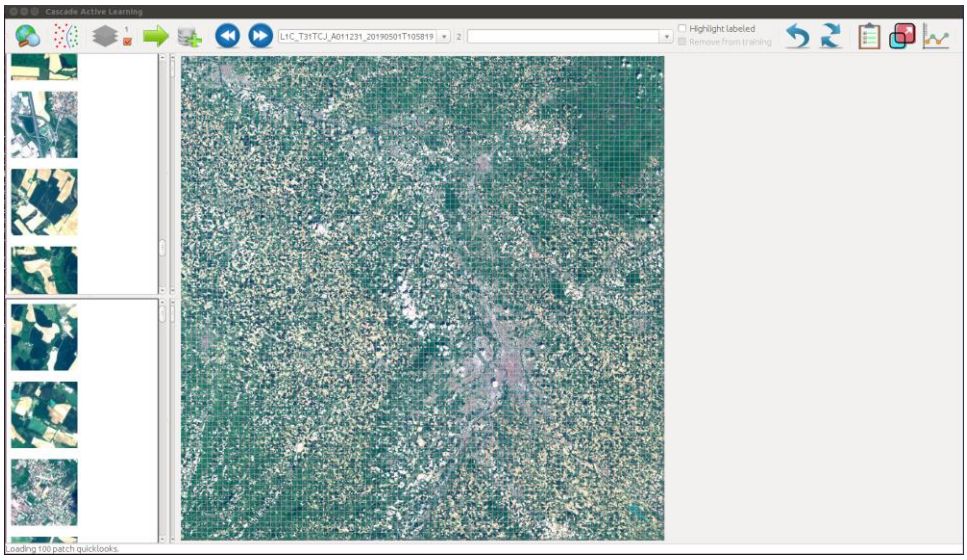
1 Users start the GUI tool on their local machines, click the query button  to perform search and query in the database, based on the mission option of the metadata. Then users click the 'Open in KDD' button to load the selected product in the interface.



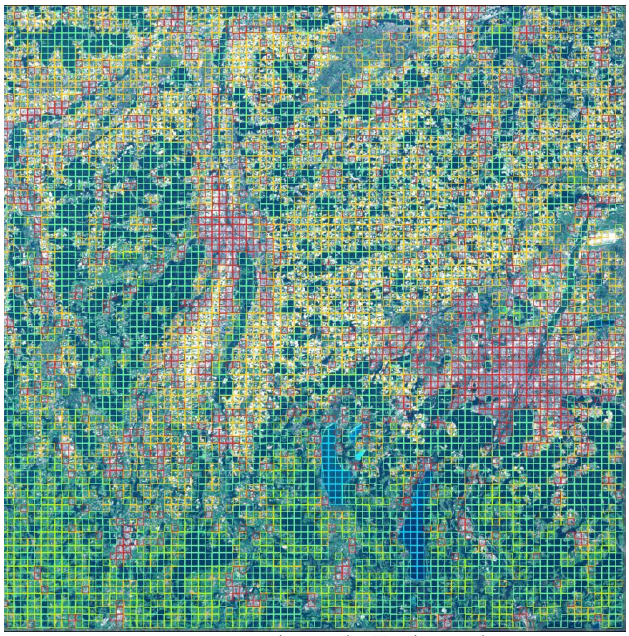

2 Users click their mouse in the GUI tool to perform semantic annotation, by viewing either S-1 or S-2 quick-look image patches. The buttons  are useful during this step. After iterations of learning which are supervised by human expertise.

[3] Or run the kdd.jar file directly in the terminal

In the KDD module, is important to have some measures of “trustworthiness” the retrieved results are ranked according their significance, as most **relevant** and most **ambiguous,** see figure.



3 Users ingest the annotations which are obtained from the previous step into the remote ‘candela’ database by pressing the ingest button  of the GUI tool. The buttons  are also useful to perform a visual and statistical analysis of the results.



| | | | | |
|---|---|---|---|---|
| Mixed urban areas | | ☑ | 2448 | Mixed urban are... |
| Stubble / bare / ploughed agric... | | ☑ | 5486 | Stubble / bare / ... |
| Grassland | | ☑ | 1314 | Grassland [1314] |
| Lakes | | ☑ | 146 | Lakes [146] |
| Mixed forest | | ☑ | 3408 | Mixed forest [34... |

## Step 5: Search and query in the database on the platform

Objective: EO Data Mining in the database

**Open the jupyter-notebook named monetdb.*ipynb***

1 Search and query in the database for the metadata of the processed products

2 Search and query in the database for the extracted features

3 Search and query in the database for the annotations

## Step 6 (optional[4]): Perform Image Mining on the image of interest for Sentinel-1 or Sentinel-2

Objective: EO image mining on fresh ingested products which are processed during Hackathon

**Run the run_kdd.sh in local terminal**

The same procedure as Step 4.

[4] It depends on the time of the DMG process for a Sentinel-1 or a Sentinel-2 product.