Data Mining I – HT15 – Lab 1

Knowledge Discovery with RapidMiner Studio

The objective of this lab is two-fold:

1. To learn how to **design a knowledge discovery process**.
2. To learn how to **use RapidMiner Studio**, one of the most popular tools for data analytics.

The examination takes place directly in the lab, where your teaching assistant will check that you correctly execute the assigned tasks.

WARM UP: RapidMiner Studio basics

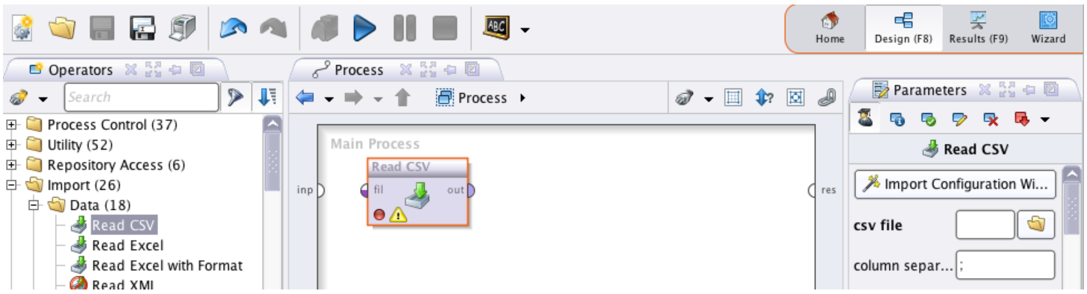
When you open RapidMiner Studio you may be asked to specify a license key, in case it is the first time the software is used on your computer. If this is the case, use the key available on the student portal. This is a free key that you can also get from RapidMiner's home page if you want to install the software on your laptop. (It is also possible to get a free academic license to unlock advanced features, e.g., reading XML input files, but this is not needed for this course.)

Then, start a new process. The **design perspective** contains:

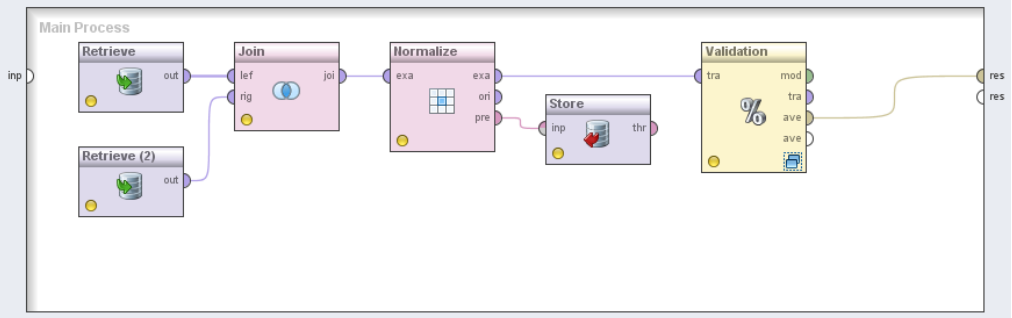
- An **Operators** panel, where you can select operators to drag and drop into the Process panel. You can also search for the needed operators by name.
- The **Process** panel, where you build your process by connecting and configuring operators.
- A **Parameters** panel, where you can set up the operators.

You can execute your process using the "play" icon in the top menu, which triggers the opening of the results perspective. You can move from one perspective to the other using the top-right menu.

After executing each task, we recommend saving your temporary results to a local file. In RapidMiner Studio, this is done using the **file > export** menu item. (The **file > save** menu item is used to store your process into a remote repository, a function not used in our labs.)



In RapidMiner Studio, processes are designed using operators to retrieve data, pre-process them, execute data mining algorithms, learn, apply and evaluate models. Operators have input and output ports that can be connected to design complex processes. As an example, consider the process:



The *tra* input port of the Validation operator receives a training set, the *mod* port outputs the classification model generated inside the operator using this training set, and *ave* returns a summary of the evaluation computed by this operator. On the right there are global ports, notated as *res*: all the information connected to these ports (like models, results of data mining algorithms, data sets and statistics) are visualized after the process has been executed.

PART 1: Data preparation

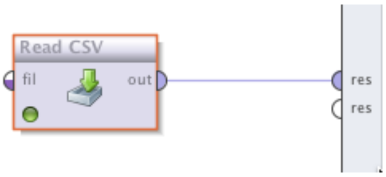
Data are seldom ready to be analyzed, because we often analyze data collected for other purposes. The objective of the first part of the lab is to prepare some data for the analysis. We use the IRIS data already mentioned in the lectures, stored into two files available on the student portal:

- iris_data.csv, containing the id of each flower and measurements for petal length, petal width, sepal length and sepal width.
- iris_labels.csv, indicating for each flower its species and the person who assigned the label.

In this part you will create a single dataset containing the following attributes:

- id, petal length, petal width, sepal length, sepal width, label.

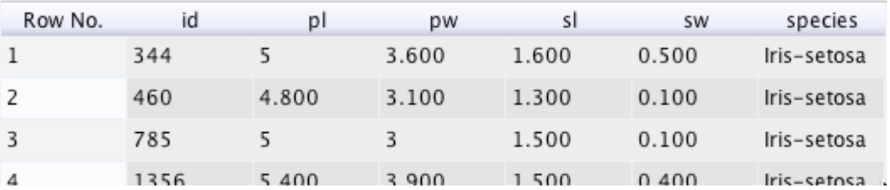
TASK 1: reading data From the operators panel of the design view, choose the "Import > Data > Read CSV" operator and drag and drop it into the empty process panel. You can also find the operator using the search box. Set up the Read CSV operator so that it reads the iris_data.csv file, then connect its out port to a result port and execute this "minimalistic" process using the "play" button. To set up the operator you can use the Import Configuration Wizard, or manually specify the values of the parameters.



TASK 2: checking the data After executing the process, the Result panel will open automatically. Quickly check that the data have been read correctly, e.g., it contains 5 columns, and change the settings of the Read CSV operator if this is not the case. When the data has been read correctly, go back to the Design perspective.

TASK 3: pre-processing Now that you can read data, modify the current minimalistic process to transform the data as follows. You can think of this activity as a visual specification of an extended Relational Algebra expression. In particular:
3.1) Add another operator to read also the file with the labels (iris_labels.csv).
3.2) Rename the two Read CSV operators to make the diagram more readable.
3.3) Join their outputs.
3.4) Filter out the attribute "examiner", that is not needed for the analysis.
3.5) Sample 150 instances (that is, flowers) uniformly at random.
3.6) Order the result by the "species" attribute.
While looking for these operators, also have a look at the different categories of operators available in this software.

TASK 4: Groups for assignments Execute your new process and check that if your final data looks like it is expected, e.g., it contains 150 records with the following attributes:



Row No.	id	pl	pw	sl	sw	species
1	344	5	3.600	1.600	0.500	Iris-setosa
2	460	4.800	3.100	1.300	0.100	Iris-setosa
3	785	5	3	1.500	0.100	Iris-setosa
4	1356	5.400	3.900	1.500	0.400	Iris-setosa

Data

Statistics

Charts

The Data view opens automatically after the execution. If the data has not been retrieved correctly, get back to Task 3. Then, explore your data using other views:
4.1) Statistics, to see a summary of the values for each attribute.
4.2) Charts, to visualize the data. Set the visualization so that each class is represented using a different color. Visualize single attribute pairs, and also all combinations using a grid plot.

PART II: CLASSIFICATION

Now that your data is ready, you can use it to classify previously unseen records. In this lab we will still use different data mining algorithms as black boxes inside larger processes, to focus on the general design process. In the next labs we will then focus on fine-tuning the algorithms and optimizing their parameters.

TASK 5: k-NN classification Add a k-NN classifier to the process. Set k=1 and Manhattan distance as parameters. Connect its input port to the output of the data pre-processing sub-process.

TASK 6: prepare the data for classification When you try to execute the new process, you may get an error message from the software informing you that the data must be further pre-processed before being ready to be classified. In particular, you might not have specified which column in the data contains the labels. If this is the case, follow the instructions from the system to fix this and execute the process, inspecting the output of the mod(el) port of k-NN.

TASK 7: classify previously unseen records The file iris_unseen.csv contains six unlabeled flowers. Read it (renaming the new CSV Read operator appropriately) and add an Apply Model operator to classify them.

TASK 8: verify the prediction Inspect the predictions, compare them with the labeled data to verify that they are reasonable. Plot the previously unseen records showing the predicted species with different colors, to visually validate them. (notice that you can connect multiple outputs to the res ports, e.g., the result of the prediction and the original data.)

TASK 9: check the impact of the parameters Execute again the process using different parameters, for example:

- k=12.
- Euclidean distance.

TASK 10: Model validation Instead of using the model to classify new records, test its performance. In particular, add a Split Validation operator.
The Validation operator is a sub-process that can be expanded. Show its internal structure by double-clicking on it.

You will find two phases inside it: training and testing (we will discuss these in detail during the lectures). You will also see input ports for training and test datasets (the operator will automatically take care of splitting the original data into these two sets). Use k-NN in the training phase, and Apply Model in the testing phase. Inside the Testing phase use also a Performance operator to evaluate the labels predicted by Apply Model.
The result of the Performance operator can be connected to the *ave* output port. In this way, the evaluation will be repeated multiple times and the average result returned outside the Validation sub-process. Try to execute it and see if the classification result is satisfactory.

TASK 11 (optional): Testing multiple algorithms Execute the current process replacing k---NN with other classification algorithms, in particular: Decision tree, Neural network, Naïve Bayes. In all cases also output and check the produced model. In the following, note the accuracy of each algorithm:

k-NN _______________

Decision tree _______________

Neural network _______________

Naïve Bayes _______________

PART III (optional): Independent design – clustering

To conclude this overview of RapidMiner Studio, and check that you are now familiar with the modeling activity, modify the current process to execute a clustering algorithm. As a clustering algorithm you can use k-means, with k=3. We remind you that clustering groups the records without looking at any class attribute. After having executed the process, check its result using the Charts view and plotting the different clusters using different colors: if you have done everything correctly, you will see again the division of the flowers into (more or less) the three groups for the three species – but without letting the system know about the labels!