Data mining: concepts and algorithms

Practice – Data mining

Objective

Exploit data mining algorithms to analyze a real dataset using the RapidMiner machine learning tool. The practice session is organized in two parts. The first part focuses on classification algorithms while the second one focuses on clustering algorithms.

Dataset

The ***Users*** dataset (Users.xls, downloadable at http://dbdmg.polito.it/wordpress/teaching/data-mining-concepts-and-algorithms/) collects census data about American users of a given company. Users are classified as "basic" or "premium" according to their commonly asked services. Each dataset record corresponds to a different user. The dataset collects around 1,000 different users, including some personal user characteristics (e.g., age, sex, workclass) as well as their corresponding class. The class attribute, which will be used as class attribute throughout the practice, is reported as the last record attribute.

The complete list of dataset attributes is reported below.

(1) Age
(2) Workclass
(3) FlnWgt
(4) Education
(5) Education-num
(6) Marital status
(7) Occupation
(8) Relationship
(9) Race
(10) Sex
(11) Capital Gain
(12) Capital loss
(13) Hours per week
(14) Native country
(15) **class (class attribute)**

Part I: Classification problem

Context

Analysts want to predict the class of new users, according to the already classified user characteristics. To this purpose, analysts exploit three different classification algorithms: a decision tree (Decision Tree), a Bayesian classifier (Naïve Bayes), and a distance-based classifier (K-NN). The ***Users*** dataset is used to train classifiers and to validate their performance.

Goal

The aim of this part of the practice is to generate and analyze different classification models and validate their performance on the ***Users*** dataset using the Rapid Miner tool. Different Rapid Miner processes have to be developed. To evaluate classification performance, different configuration settings have to be tested and compared with each other. A 10-fold Stratified Cross-Validation process must be used to validate classifier performance. Results achieved by each algorithm should be analyzed in order to analyze the impact of the main input parameters.

Questions

Answer to the following questions:

1. Learn a Decision Tree using the whole dataset as training data and the default configuration setting for algorithm Decision Tree. (a) Which attribute is deemed to be the most discriminative one for class prediction? (b) What is the height of the generated Decision Tree? (c) Find an example of pure partition in the Decision Tree generated.
2. Analyze the impact of the minimal gain (using the gain ratio splitting criterion) parameter on the characteristics on the Decision Tree model learnt from the whole dataset (keep the default configuration for all the other parameters).
3. Use a 10-fold Stratified Cross-Validation approach to validate the accuracy of the generated classification model. What is the impact of the minimal gain parameter on the average accuracy achieved by the generated Decision Tree? Compare the confusion matrices achieved using different parameter settings (keep the default configuration for all the other parameters).
4. Considering the K-Nearest Neighbor (K-NN) classifier and performing a 10-fold Stratified Cross-Validation, what is the impact of parameter **K** (number of considered neighbors) on the classifier performance? Compare the confusion matrices achieved using different **K** parameter values. Perform a 10-fold Stratified Cross-Validation with the Naïve Bayes classifier. Does K-NN perform on average better or worse than the Naïve Bayes classifier on the analyzed data?

Part II: Clustering problem

Context

Analysts want to identify group of similar users. More specifically, they want to segment the users in a set of groups (clusters). For each cluster an ad-hoc advertising campaign will be designed. To this purpose, analysts exploit two different clustering algorithms: a k-Means clustering algorithm (**K-Means**) and a density-based algorithm (**DBScan**). The ***Users*** dataset contains the users to analyze. Pay attention that only the numerical attributes are used during this second part of the practice.

Goal

The aim of this second part of the practice is to generate and analyze different clustering models and validate their performance on the ***Users*** dataset using the Rapid Miner tool. To evaluate clustering performance, different configuration settings have to be tested and compared with each other. The average within cluster distance will be used to validate clustering performance. Results achieved by each algorithm should be analyzed in order to analyze the impact of the main input parameters.

Questions

Answer to the following questions:

1. Apply the **k-Means** algorithm to cluster the users. Analyze the characteristics of the generated clusters (e.g., the size of the extracted clusters).
2. Analyze the impact of parameter **k** (number of generated cluster) on the generated clusters. More specifically, perform an empirical analysis by using the average within cluster distance measure (a Cluster Cohesion measure) to evaluate the impact of the value of k on the quality of the generated clusters. What is the impact of **k** on the generated clusters?
3. Consider the **DBScan algorithm** (a density based algorithm) and compare its performance with that of the **k-Means** algorithm. What is the impact of parameter **epsilon** on the performance of DBScan (in terms of average within cluster distance)? Does **DBScan** perform better than **k-Means**?

Program setup

- Run the Rapid Miner application under Windows XP

Process building and analysis

- Create a new Rapid Miner process.
- Build the data mining flow by dragging the operators available on the left-hand side menu and dropping them into the main process window.

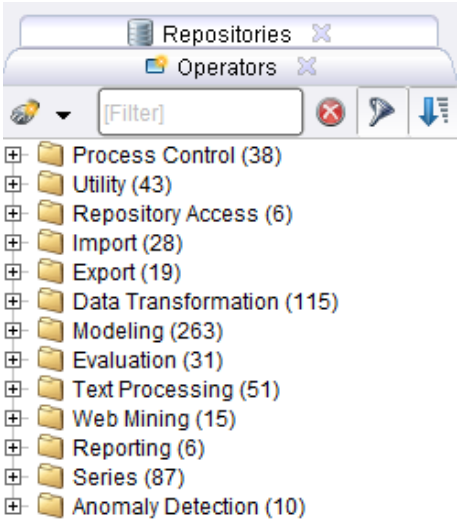

Figure 1. Operators

- To handle process execution, use the Start/Stop/Pause buttons. To view the results, change the perspective from *Design* to *Results*.

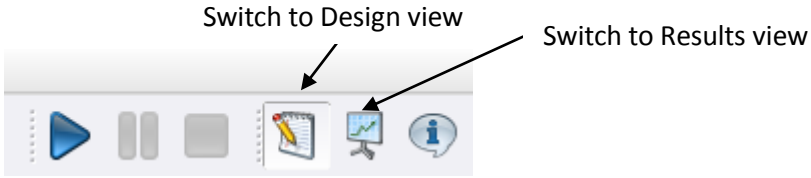

Figure 2. Execution/perspective change buttons

- Look into the content of the ***Users*** dataset, which is available in the Excel format (.xls).

Classification task

- Import the source data into the Data Mining process by using the operator “Read Excel”. To import data use the *Data Import Wizard* as follows:
  - Select the source file (Step 1).
  - Select all the spreadsheet content (Step 2).
  - Annotate the first row as the attribute name (label “name”), while keeping all the remaining rows unlabeled (“-”) at Step 3.
  - Bind the data import block with the data source. Identify the role of attribute “class” as “label” attribute (Step 4).

- Include the “Decision Tree” classifier at the end of the data mining flow. The currently generated process looks like the following one:

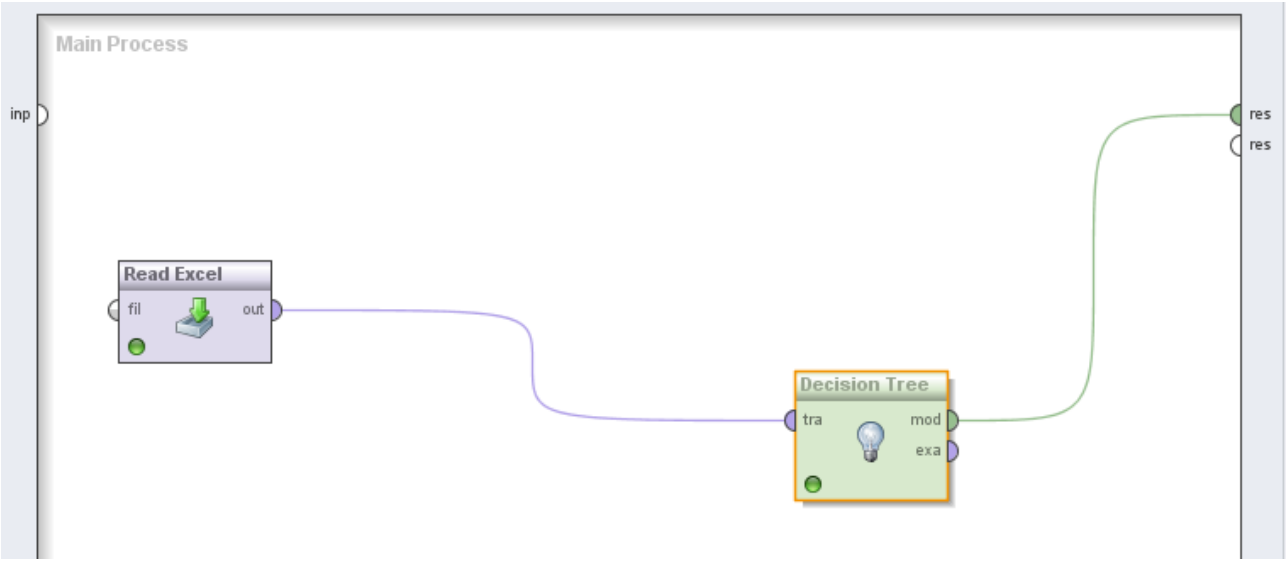

Figure 3. Decision tree classification process

- Execute the process and analyze the Decision Tree generated through the Results perspective.
- Change the configuration setting for algorithm Decision Tree clicking on the corresponding operator and using the right-hand side menu in the Design perspective. Specifically, vary the minimal gain threshold value to analyze its impact on the characteristics of the classification model.
- Modify the process flow in order to perform a 10-fold Stratified Cross-Validation. To this aim, include the “Validation” operator in place of Decision Tree into the main process first.

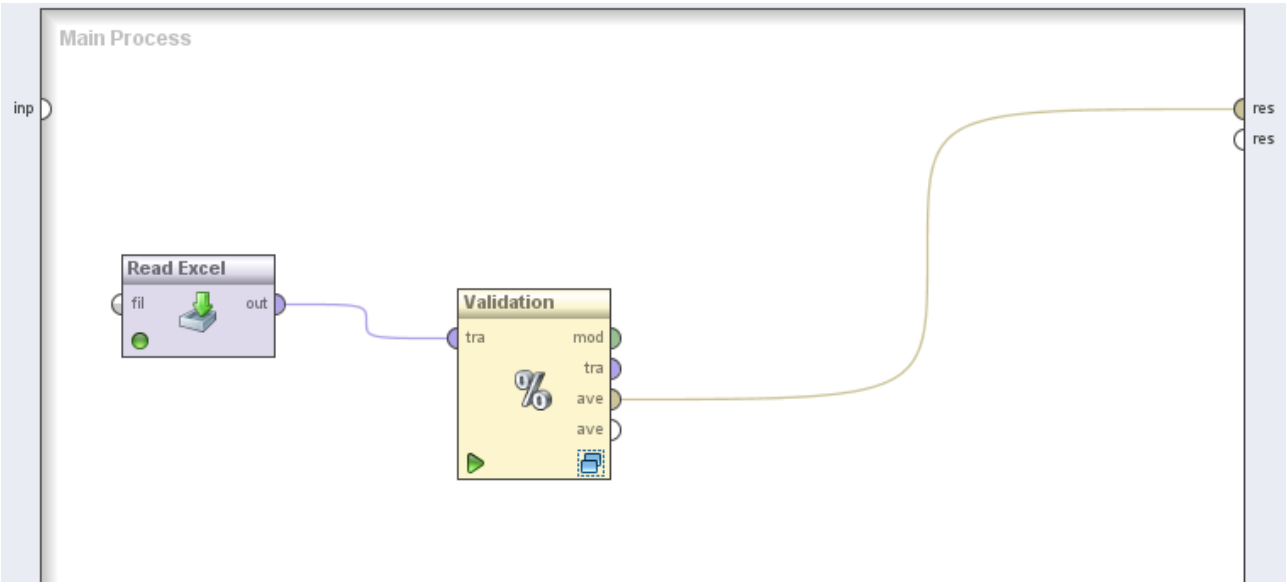

Figure 4. 10-Fold Cross-Validation process.

Next, double-click operator “Validation” and create a nested process as the one reported below:

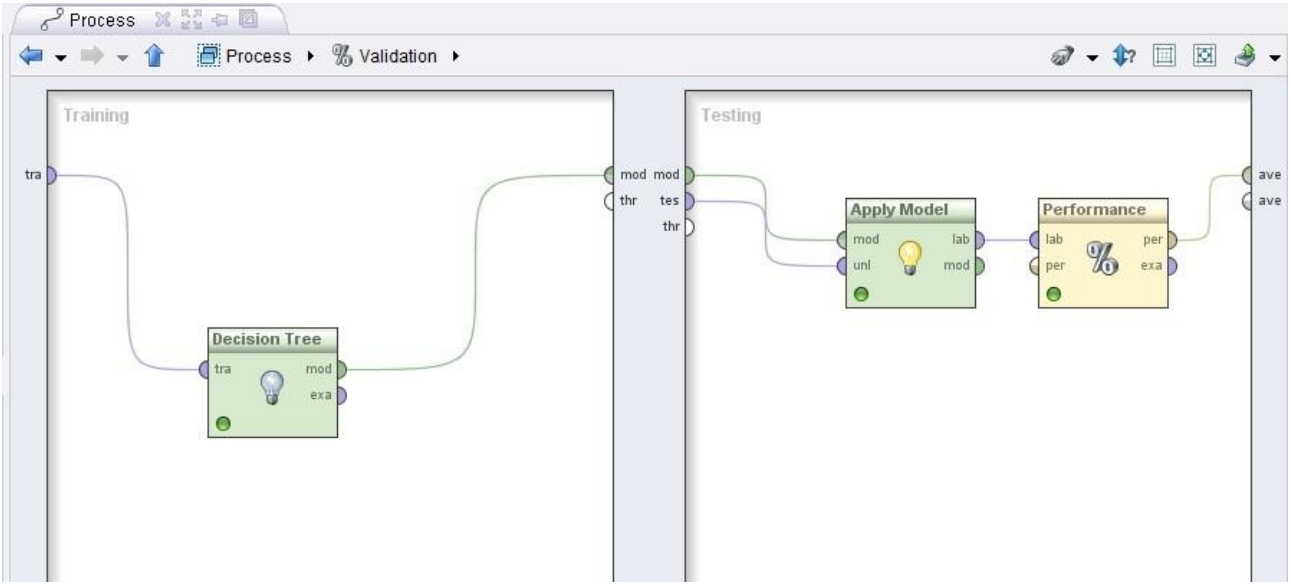

Figure 5. Validation subprocess

- Moving to the Results perspective, analyze the confusion matrix generated by the validation process.
- Substitute the classifier operator with the Naïve Bayes classifier first and with the K-NN classifier next and analyze the achieved results.
- Compare the performance of K-NN and Naïve Bayes performance in terms of average accuracy by analyzing the corresponding confusion matrices. For the K-NN classifier, vary parameter K values using the right-hand side menu in the Design perspective.

Clustering task

- Import the source data into the Data Mining process by using the operator "Read Excel". To import data use the *Data Import Wizard* as follows:
    - Select the source file (Step 1).
    - Select all the spreadsheet content (Step 2).
    - Annotate the first row as the attribute name (label "name"), while keeping all the remaining rows unlabeled ("-") at Step 3.
- Select exclusively the set of numerical attributes (i.e., exclude non-numerical attributes) by means of the "Select Attributes " operator. Set the "attribute filter type" parameter of the operator to "subset" and then click on the "Select attributes" button and select the subset of numerical attributes (age, FlnWgt, Education-Num, Capital gain, Capital loss, and Hours-per-week).
- Normalize data values by means of the operator "Normalize". Set "attribute filter type" to all and "method" to Z-transformation.
- Include the "k-Means" clustering operator at the end of the data mining flow. The currently generated process looks like the following one:

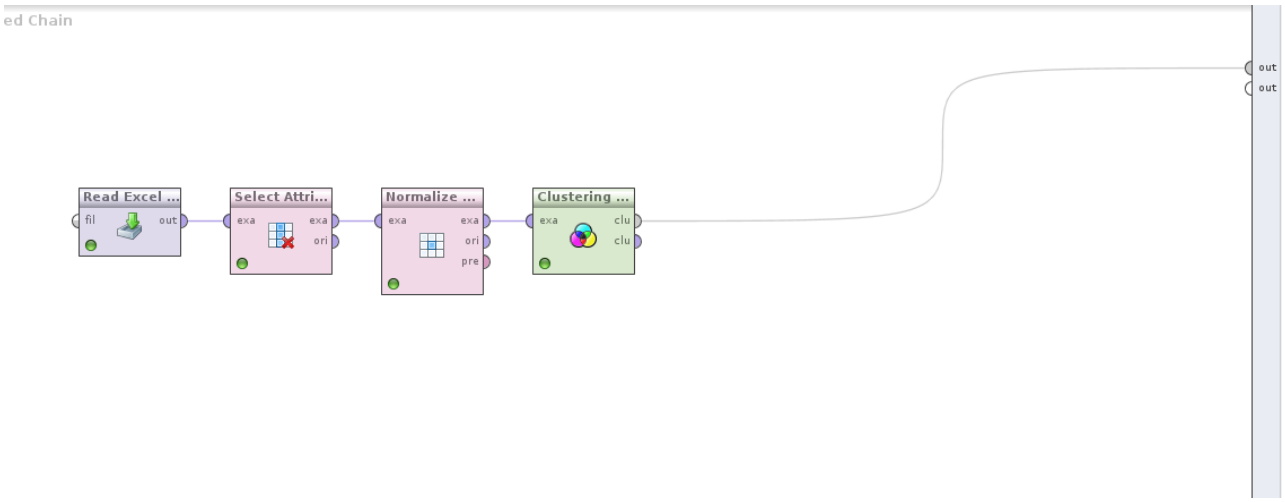

Figure 6. k-means (clustering) process

- Execute the process and analyze the generated clusters (number of clusters, number of data per cluster).
- To validate the quality of the generated clusters in terms of cluster cohesion two other operators must be included in the process. Specifically, include the "Data to similarity" similarity operator and then the "Cluster density performance" operator. The "Data to similarity" has one input (the example dataset) and two outputs (the distance between each pair of objects of the example dataset and the dataset itself). It computes the distance between the objects of the input datasets. Select "numerical measure" as measure type and "EuclideanDistance" as measure. The "Cluster density performance" computes the average within cluster distance and hence the cluster cohesion of a set of clusters. It is computed by averaging all distances between each pair of examples of a cluster. The "Cluster density performance" has three mandatory inputs: (i) the cluster model (i.e., the output of the clustering algorithm), (ii) the set of considered objects (the dataset of users in our case), and (iii) the distance between the considered objects (i.e., the first output of the "Data to similarity" operator). The currently generated process looks like the following one:

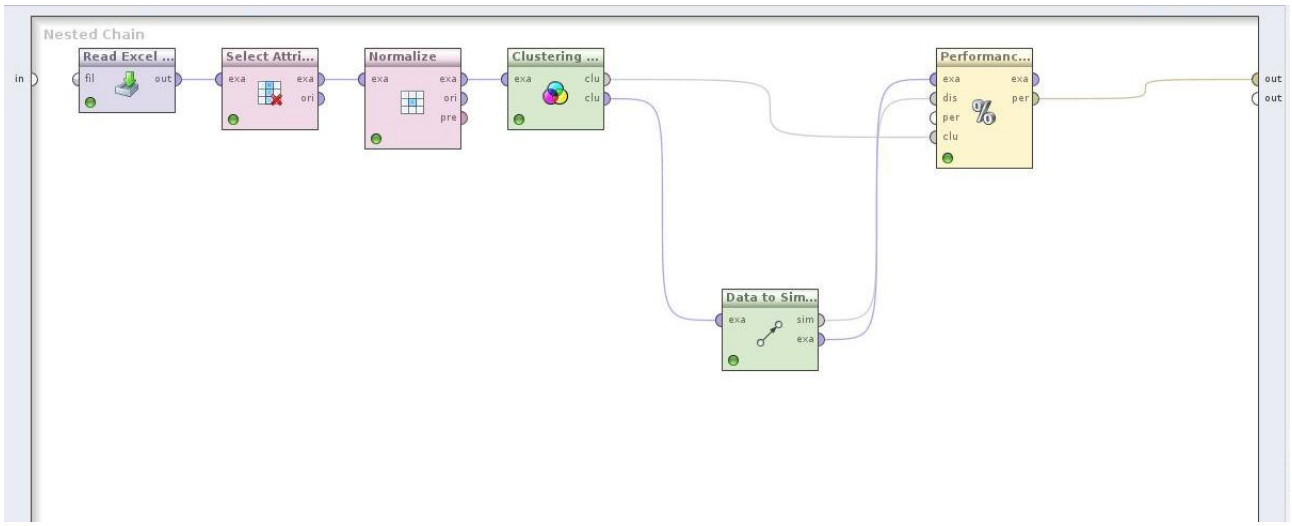

Figure 7. k-means (clustering) process + performance evaluation

- Change the configuration setting for the k-Means algorithm clicking on the corresponding operator and using the right-hand side menu in the Design perspective. Specifically, vary the value of parameter k and analyze the impact of its value on the quality the generated clusters (i.e., the impact on the average within cluster distance measure).
- Substitute the K-Means clustering operator with the DBscan clustering algorithm/operator and analyze the achieved results. Consider different values of epsilon.
- Compare the performance of K-Means and DBScan in terms of average within cluster distance measure.