Apprentissage non-supervisé

1 Catégorisation : K-Means

La catégorisation, ou clustering ou classification non-supervisée, est une méthode pour construire des regroupements d'individus proches, en groupe d'individus distincts. Chaque groupe d'individus est alors une catégorie ayant des caractéristiques spécifiques. L'algorithme des K-Means, K-Moyennes ou nuées dynamiques, est un algorithme classique pour déterminer des catégories de personnes.

1.1 Catégorisation des étudiants

Exercice 1 - Catégorisation des étudiants

Lors des cours, les étudiants remplissent des évaluations du cours en répondant à des questions sur : l'appréciation globale, le sujet, les supports, la clarté et si l'enseignant était intéressant. Les notes négatives (-1 et -2) indiquent des évaluations respectivement "pas bon" et "pas bon du tout". Les notes positives (1 et 2) indiquent des évaluations respectivement "bon" et "très bon". Le 0 indique la neutralité dans l'évaluation. Les données ont été recueillies pour 4 cours différents sur plusieurs années.

On cherche à identifier des "profils" d'étudiants par rapport à leur réponses au questionnaire.

Question a) *Chargement et visualisation des données*

Créer une nouvelle expérimentation dans RapidMiner. Dans un premier temps, ajouter uniquement le bloc de chargement des données et configurer-le :

- les données sont disponibles dans le fichier `EvaluationsCours.csv`,
- le symbole des décimals est la virgule : il faut bien penser à modifier le séparateur de décimal par défaut (qui est le point).

Lancer l'expérimentation et visualiser les données ("plot view") en choisissant deux des quatres questions comme axes du graphique et, par exemple, en visualisant une couleur par cours. En utilisant le curseur *Jitter*, vous ajouterez un "bruit" dans l'affichage des données qui permettra de mieux visualiser l'ensemble de celles-ci (évitera les superpositions).

- Tous les étudiants qui ont trouvé l'enseignant "très clair" et "très intéressant" ont-ils trouvé le cours globalement "très bon" ?
- Est-il nécessaire à l'enseignant de produire de très bon supports de cours pour avoir son cours estimé comme "bon" ou "très bon" ?
- Y a-t-il des cours pour lesquels l'enseignant semblent plus "clair" et plus "intéressant" ?

Question b) *Catégorisation automatique*

Nous allons maintenant configurer l'expérimentation pour utiliser l'algorithme K-Means. Le processus à construire est illustré dans la Figure 1. L'utilisation de cet algorithme ne peut se faire que sur des données numériques et qui sont sans données manquantes. Les données d'enquêtes comportent des données manquantes et définissent des attributs qui ne nous intéressent pas : "Cours" qui est qualitatif (il ne peut pas être utilisé dans un K-Means), "Année" est peu informatif pour notre classification.

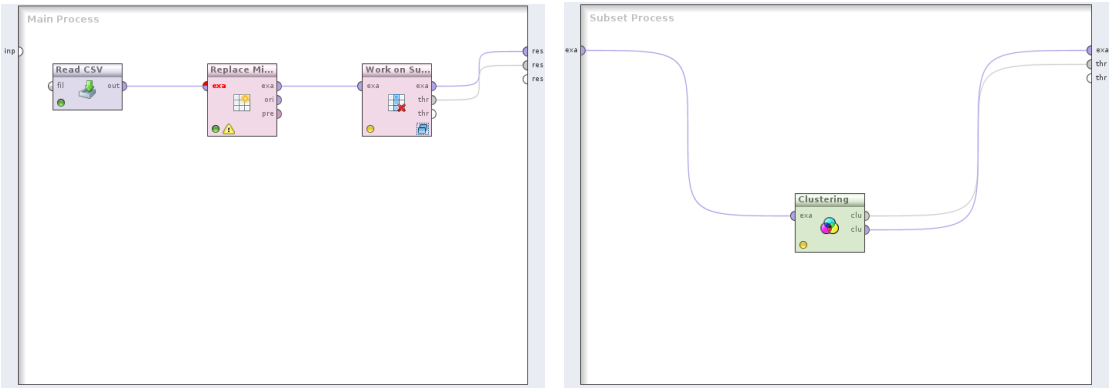


FIGURE 1 – Illustration du processus de classification des données d'évaluation de cours. Sur la gauche, le processus générale, sur la droite, l'intérieur du bloc `Work on subset`

1. Traitement des données manquantes : ajouter un bloc `Replace Missing Value` (accessible depuis le menu `Data Transformation> data cleaning`) et utiliser un remplacement par défaut par la moyenne des autres valeurs (`average`).
2. Masquage des attributs inutiles à la classification : ajouter une bloc `Work on subset`. Configurez-le :
   - filtrer un *subset* d'attributs,
   - définisser les attributs à supprimer ("Cours" et "Année") en cliquant sur le boutont `Select Attributes` puis en donnant ces noms d'attributs dans la colonne de droite (attention aux majuscules),
   - sélectionner l'option `invert_selection` pour indiquer qu'on va supprimer un attribut (plutôt que le conserver).
3. Brancher la sortie du `Work on subset` sur la sortie du processus.

Le bloc `Work on subset` va contenir la partie du processus qui n'utilisera que les attributs sélectionnés par la bloc. Les autres attributs seront masqués à l'intérieur, mais vous les récupérer à la sortie du bloc.

- entrer dans le bloc `Work on subset`
- ajouter un bloc pour l'algorithme *K-Means* (accessible depuis le menu `Modeling> Clustering and segmentation`)
- configurer le bloc avec `k` égal à 3 (le nombre de classes).
- prendre les données en entrée du `Work on subset`, et récupérer la seconde sortie du *K-Means* (en bleue)

Une fois les résultats obtenus, vous pourrez observer :

- le *centroid plot view* qui donne la valeurs des attributs pour les centroides des classes. Il s'agit donc des caractéristiques des différentes classes qui ont été trouvées.
- le *scatter plot* des données en visualisant le *cluster* comme couleur de point (attributs ajoutés automatiquement pour indiquer la classe de chaque individu).

À partir des caractéristiques de classe, donner un nom ou une phrase pour décrire les classes d'étudiants.

Modifier le paramètre `k` de la l'algorithme *K-Means* (essayer 2, 3 et 4 classes) et comparer les résultats :

– toutes les catégories changent-elles de caractéristiques en fonction du nombre de classes ?
– quel nombre de classe vous semble le plus adapté pour ces données ?

Question c) *Analyse de la répartition des classes dans les années* En utilisant un *scatter plot*, visualiser les clusters en fonction des années (ajouter du *jitter* pour faciliter la visualisation). La répartition des étudiants dans les différentes années est-elle la même pour toutes les années ? Que peut-on en conclure ?

Question d) *Comparaison de deux classifications*

On cherche maintenant à comparer les résultats de classification en construisant une *sorte de matrice de confusion*. Les classifications qui peuvent être comparées sont les suivantes :

– classifications obtenues avec un algorithme pour 3 classes ou 4 classes
– classifications obtenues avec deux algorithmes différents (tester `K-Means` et `W-HierarchicalClustering`

Pour cela, réaliser les opérations suivantes dans l'ordre. Le résultat final est illustré dans la Figure 2.

1. Ajouter un meta-bloc `Work on subset`,
2. Dans ce bloc, insérer :
   – un bloc de classification à utiliser
   – un bloc `set role` pour changer le role de l'attribut `cluster` en sortie de la classification en `regular`
   – un bloc `rename` pour changer le nom de l'attribut `cluster` en `cluster_kmeans` par exemple
   – un bloc `select attribute` pour supprimer l'attribut `id` (utiliser une sélection "single" en choisisant l'attribut "id" et valider les cases *invert selection* et *include special attributes*). L'attribut `id` est ajouté par les algorithmes de classification, mais ne nous servent pas après.
3. Faire de même pour un autre algorithme dans un autre meta-bloc `Work on subset` à mettre à la suite du précédent (**faire un copier-coller du bloc`Work on subset` puis modifier l'algorithme de classification ou ses paramètres**)
4. Dans les paramètres du second meta-bloc `Work on subset`, indiquer de masquer l'attribut `cluster_kmeans` pour ne pas le prendre en compte dans l'apprentissage,
5. Finalement, récupérer le jeu d'exemples en sortie.

Lancer l'expérimentation et visualiser dans un *scatter plot* `cluster_kmeans` en fonction de `cluster_cah` (ajouter du *jitter*).

Une fois cette expérimentation construite, vous pouvez faire varier également les algorithmes de catégorisation (*EMClustering*, *SupportVectorClustering*) et en comparant les résultats d'attribution de classes.

1.2 Application à la classification d'images

Pour cet exercice, on utilise des images MODIS 250m, indice NDVI, de la région de Diourbel (Sénégal) en 2002. Pour chaque pixel de l'image on dispose d'une valeur de NDVI tous les 15 jours. On peut alors parler de SITS (Séries Temporelles d'Image Satellite) L'objectif est de classer les pixels de l'images.

Le processus général à construire est décrit dans la Figure 1.2.

Le bloc `GeoTiffSampler` charge une image au format GeoTIFF. Des coordonnées de l'étendue sont requises par ce bloc pour échantillonner l'image :

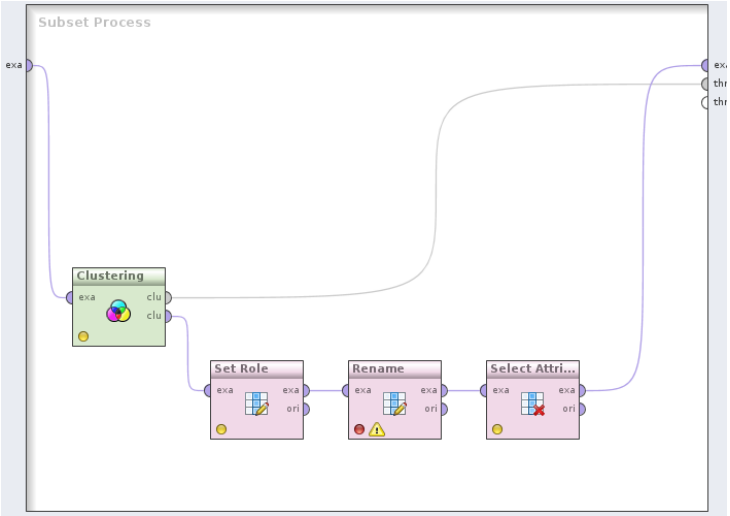


FIGURE 2 – Intérieur du meta-bloc `Work on subset`, avec transformation en sortie de la classification

- CRS : "EPSG :32628"
- West limit : 325106
- East limit : 427838
- North limit : 1658078
- South limit : 1603957

Ensuite, on donne les données à un bloc `Work on Subset` qui va permettre de travailler en masquant les attributs de coordonnées (attributs `X-Coordinate` et `Y-Coordinate`). Vous pourrez mettre un algorithme de catégorisation dans ce meta-bloc et sortir le résultat de la classification (exemples classés et modèle).

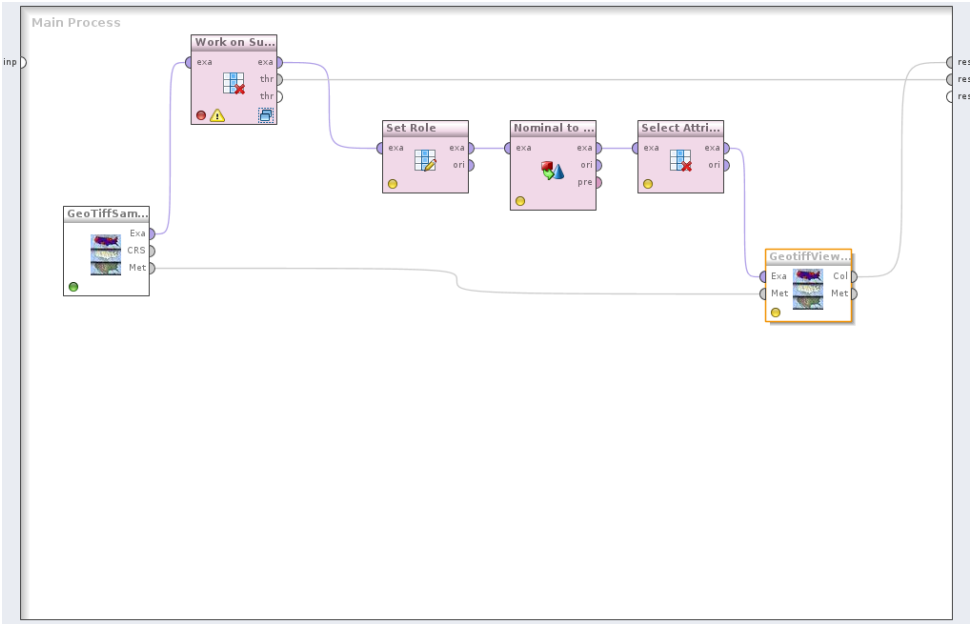
À la sortie de ce bloc, on récupère les données par la sortie `exa`. Le reste du traitement permet de sélectionner l'attribut `cluster` pour le donner au bloc "`GeoTiffWriter`" ou "`GeoTiffViewer`" permettant de visualiser le résultat. Dans notre cas, on utilisera le bloc `GeoTiffViewer`. Ce bloc ne permet d'afficher qu'une seule couche, il faut donc lui donner une seule valeur d'attribut. Les trois blocs de prétraitements permettent de faire cela :

- le bloc `set role` transforme l'attribut `cluster` en un attribut `regular`,
- le bloc `Nominal to numeric` transforme les attributs catégoriel, comme la classe, en attribut numérique. Vous pouvez laisser la transformation de "tous" les attributs ou ne transformer que l'attribut `cluster`
- le bloc `select attribute` ne doit conserver que les attributs `cluster`, `X-Coordinate` et `Y-Coordinate`.

Dans un premier temps, vous utiliserez une catégorisation par K-Means avec $k = 5$.

Dans le bloc `GeoTiffViewer`, vous configurerez les couleurs d'affichage dans la liste des couleurs (*Edit list*). Pour la cohérence de l'afficahge à venir, il est conseiller d'utiliser les couleurs suivantes :

- 0 : bleu marine
- 1 : bleu ciel
- 2 : vert
- 3 : jaune
- 4 : rouge



Exécuter votre processus.

Vous pourrez utiliser le *Centroid Plot View* pour visualiser les centroïdes des classes (profils annuels moyens). Le bloc `GeoTiffViewer` affichera une image comme résultat.

Question e) *Sachant que 2002 a été une année d'importante sècheresse, interpréter les différents centroïdes de classes.*

Question f) *Modifier les algorithmes ou le nombre de classes pour comparer "visuellement" les résultats.*

1.3 Evaluation quantitative des performances du clustering

RAPIDMINER n'est pas très riche en indices performances des clustering. On trouve le bloc `Cluster Distance Performances` qui donne des informations sur les distances moyennes au sein des classes et calcule l'indice de Davies-Bouldin.

Pour l'un ou l'autre des jeux de données précédents, construire un processus qui automatise le teste de plusieurs valeurs de k pour des K-Means en récupérant à chaque étape un indice sur la densité des classes (prendre la distance moyenne au centroid, *avg distance to centroid*).

Rappel : Utiliser un bloc Optimisation Grid dans le `Work on subset`.