

Data Management

Lab 02 - Analyse de données

Robin Chappatte

Frédéric Montet

Brian Nydegger

Rendu le 27 novembre 2016

à Lausanne

Professeurs :

Dr. Laura Elena Raileanu

Fabien Dutoit

Table des matières

Introduction	1
1 Installation et configuration	2
1.1 Installation	2
1.2 Configuration	2
2 Analyse de données	4
2.1 Arbre de décisions	4
2.2 Clustering	7
2.3 Règles d'associations	7
2.3.1 Apriori	8
2.3.2 FilteredAssociator	8
3 Conclusion	10

Introduction

Pour le second laboratoire du cours de Data Management, nous allons mettre en pratique les concepts d'analyse de donnée vus en cours :

- Arbre de décisions
- Clustering
- Règles d'associations

Pour faire cela, nous utiliserons le logiciel Weka car il dispose d'une grande collection d'outils pour l'analyse de donnée.

1 | Installation et configuration

Dans notre situation nous avons décidé d'installer Weka sur une machine virtuelle Ubuntu 16.10.

1.1 Installation

Nous avons installé :

- Java JDK 1.8_111
- Weka 3.8.0
- Xampp x64 7.0.13
- MySQL connector for Java 5.1.39

Un élément à mentionner est que nous avons utilisé le connecteur dans sa version 5.1.39 alors que sur le site de MySQL¹ il est disponible en 5.1.40. Ceci est dû au fait que nous avons utilisé la version disponible dans un repository ppa. L'installation a pu être faite avec la commande `sudo apt-get install libmysql-java`.

Hormis le point cité au paragraphe ci dessus, cette étape n'a pas posé de problème particulier qu'il serait nécessaire de mentionner. Il a suffit de suivre les différents tutoriaux relatifs à chacun des logiciels à installer.

1.2 Configuration

Pour configurer la machine virtuelle, un seul point a posé problème : la connexion à la base de donnée MySQL depuis Weka en utilisant le connecteur MySQL Java.

Le problème rencontré a été la prise en compte de la variable d'environnement `CLASSPATH` lors des tentatives de connexions. Sans cette variable, Weka ne trouve pas le connecteur.

Après avoir effectué une recherche, nous nous sommes rendu compte qu'avec Ubuntu, il fallait préciser le `CLASSPATH` dans la commande pour démarrer Weka à l'aide du paramètre `-cp`. Le listing Figure 1.1 présente le `.sh` qui a été utilisé pour démarrer Weka et le connecter à MySQL avec succès.

1. <https://dev.mysql.com/downloads/connector/j/>

1.2. Configuration

```
1 # Variables
2 java8=/home/fredmontet/Desktop/lab02-weka/asset/java/jdk1.8.0_111/jre/bin/
   ↪ java
3 weka=/home/fredmontet/Desktop/lab02-weka/asset/weka-3-8-0/weka.jar
4 connector=/usr/share/java/mysql-connector-java.jar
5
6 # Command
7 export CLASSPATH=$connector:$CLASSPATH
8 $java8 -Xmx300m -cp "$weka:$connector" weka.gui.GUIChooser
```

FIGURE 1.1 – Contenu du fichier bash à exécuter pour lancer Weka

2 | Analyse de données

Les données à dispositions dans la base de donnée qui a été importée à partir du fichier *aventure2014.sql.gz* n'ont pas été utilisées complètement pour ce laboratoire. Uniquement les attributs suivants ont été utilisés :

- MaritalStatus
- Gender
- YearlyIncome
- TotalChildren ;
- EnglishEducation
- EnglishOccupation
- HouseOwnerFlag
- NumberCarsOwned

2.1 Arbre de décisions

L'algorithme choisi pour cette méthode de classification a été REPTree. Après avoir effectué plusieurs runs aux paramètres différents, nous avons choisi un des meilleurs de nos tests.

Sur le listing Figure 2.1, on observe les informations du run choisis ainsi que les attributs sur la base desquels on veut trouver les classes de l'attributs **EnglishOccupation** qui sont :

- Professional
- Management
- Skilled Manual
- Clerical
- Manual

2.1. Arbre de décisions

```
1 Scheme:      weka.classifiers.trees.REPTree -M 2 -V 0.001 -N 10 -S 42 -L -1
   ↳ -P -I 0.0 -batch-size 200
2 Relation:    QueryResult-weka.filters.unsupervised.attribute.Remove-R1
   ↳ -7,9,11,14,16,18,21-25
3 Instances:   18484
4 Attributes:  8
5              MaritalStatus
6              Gender
7              YearlyIncome
8              TotalChildren
9              EnglishEducation
10             EnglishOccupation
11             HouseOwnerFlag
12             NumberCarsOwned
13 Test mode:   split 80.0% train, remainder test
```

FIGURE 2.1 – Information du run REPTree

Le résultat de ce run a montré que la moyenne pondérée des instances correctement classifiées est 91.29%. La valeur de ce résultat est bonne et indique que notre modèle est bon. Cependant, seul ce résultat ne permet pas d'affirmer que le modèle est bon. Il est nécessaire d'observer les différentes métriques dans la partie **Detailed Accuracy By Class** du listing Figure 2.2 pour en savoir d'avantage. Sur ce dernier, on constate que 0.913 est la moyenne des métriques TP Rate, Precision, Recall et F-Measure pour les différentes classes. Notre analyse est donc confortée par la stabilité du résultat moyen de ces différentes métriques.

Maintenant, il est intéressant de savoir quelles sont les forces et faiblesses de notre modèle. Pour cela, nous baserons nos constats sur la F-Measure. Cette dernière étant robuste car elle prend en compte precision et recall. On y voit que la classe au meilleur score est **Manual** avec 0.970 et la moins bonne est **Skilled Manual** avec 0.882.

Afin d'améliorer notre modèle, on se concentrera donc sur la classes **Skilled Manual**. La matrice de confusion nous donne le détail de classification des instances de **Skilled Manual**. On y voit que **Skilled Manual** est souvent pris pour **Professional** et vis et versa. Ceci peut nous donner une piste pour de plus amples analyse ou on se questionnera sur les similarités et différences entre 2 instances de chacune des classes pour comprendre quel sont les facteurs qui rentrent en jeux.

2.1. Arbre de décisions

```

1 === Summary ===
2
3 Correctly Classified Instances      3375      91.2902 %
4 Incorrectly Classified Instances    322      8.7098 %
5 Kappa statistic                    0.8883
6 Mean absolute error                 0.0434
7 Root mean squared error            0.1549
8 Relative absolute error             13.8991 %
9 Root relative squared error         39.2271 %
10 Total Number of Instances          3697
11
12 === Detailed Accuracy By Class ===
13
14 TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area
15 ↪ Class
16 0.904    0.043    0.899      0.904    0.902      0.860    0.982    0.963
17 ↪ Professional
18 0.912    0.019    0.909      0.912    0.910      0.892    0.992    0.970
19 ↪ Management
20 0.873    0.035    0.892      0.873    0.882      0.844    0.981    0.945
21 ↪ Skilled Manual
22 0.946    0.013    0.930      0.946    0.938      0.927    0.991    0.971
23 ↪ Clerical
24 0.972    0.005    0.968      0.972    0.970      0.966    0.997    0.981
25 ↪ Manual
26
27 Weighted Avg.
28
29 0.913    0.027    0.913      0.913    0.913      0.885    0.987    0.963
30
31 === Confusion Matrix ===
32
33      a      b      c      d      e  <-- classified as
34 1002    25    69    12     0 | a = Professional
35  46   568     9     0     0 | b = Management
36  62    32   802    20     3 | c = Skilled Manual
37   4     0    15   545    12 | d = Clerical
38   0     0     4     9   458 | e = Manual

```

FIGURE 2.2 – Résultat du run REPTree

Concernant l'arbre de décision, son début est visible dans le listing Figure 2.3. Sur cet exemple, on voit que la division commence par l'apport financier, puis l'éducation, les enfants, l'état civil et la possession d'une maison.

Il est intéressant de noter que juste après la division homme-femme, l'état civil et la possession d'une maison ont été interverti dans la branche M et F. Ceci met en évidence que la meilleure manière de diviser les classes est prise en compte en fonction de chacune des instances.


```

1 YearlyIncome < 45000
2 |   YearlyIncome < 25000
3 |   |   EnglishEducation = Bachelors
4 |   |   |   YearlyIncome < 15000
5 |   |   |   |   TotalChildren < 1.5
6 |   |   |   |   |   Gender = M
7 |   |   |   |   |   |   MaritalStatus = M
8 |   |   |   |   |   |   |   HouseOwnerFlag = 1 : Manual (27/1) [0/0]
9 |   |   |   |   |   |   |   HouseOwnerFlag = 0 : Manual (3/0) [0/0]
10 |   |   |   |   |   |   |   MaritalStatus = S : Manual (11/0) [0/0]
11 |   |   |   |   |   |   |   Gender = F
12 |   |   |   |   |   |   |   |   HouseOwnerFlag = 1
13 |   |   |   |   |   |   |   |   |   MaritalStatus = M : Manual (21/1) [0/0]
14 |   |   |   |   |   |   |   |   |   MaritalStatus = S : Manual (7/0) [0/0]
15 |   |   |   |   |   |   |   |   |   HouseOwnerFlag = 0 : Manual (8/1) [0/0]

```

FIGURE 2.3 – Début du REPTree

2.2 Clustering

2.3 Règles d'associations

Pour cette section, le filtre *NumericToNominal* a été appliqué aux attributs *NumberCarsOwned*, *YearlyIncome* et *TotalChildren*.

Seul les algorithmes *Apriori* et *FilteredAssociator* ont été testé. Seul ceux-ci étaient disponible (voir 2.4).

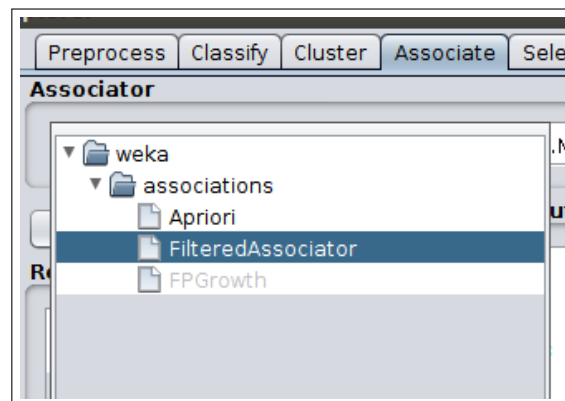


FIGURE 2.4 – Règles d'associations : Algos disponible

2.3.1 Apriori

Le seuil minimum de confiance a été modifié pour obtenir plus de règles.

Voici les paramètres :

```
1 Apriori -N 10 -T 0 -C 0.8 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c 1
```

Et le résultat (figure 2.5).

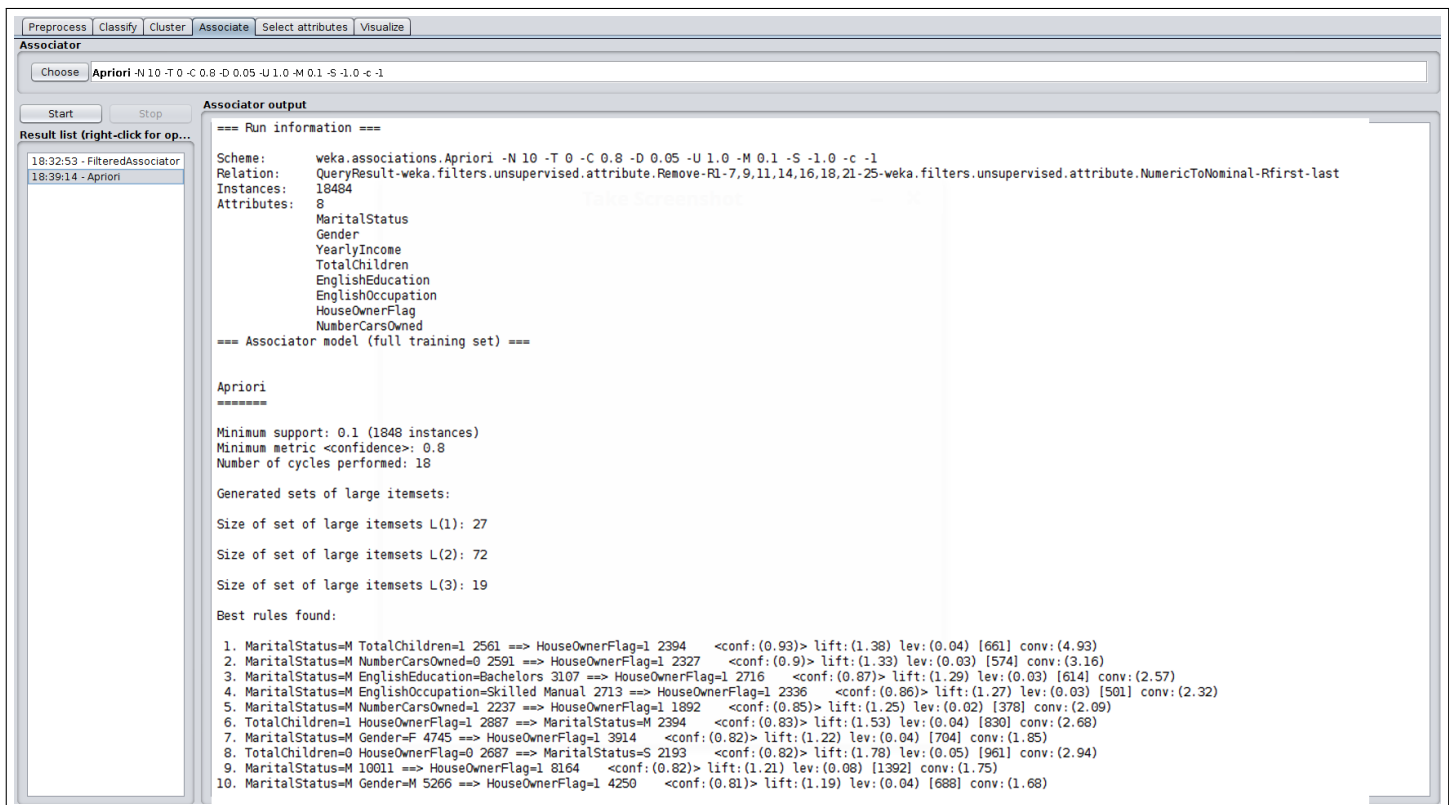


FIGURE 2.5 – Règles d'associations : Apriori résultat

2.3.2 FilteredAssociator

Cet algorithme permet de créer des règles d'association filtrées.

Les 10 règles obtenues (figure 2.6) sont identiques au résultat de l'*Apriori*.

2.3. Règles d'associations

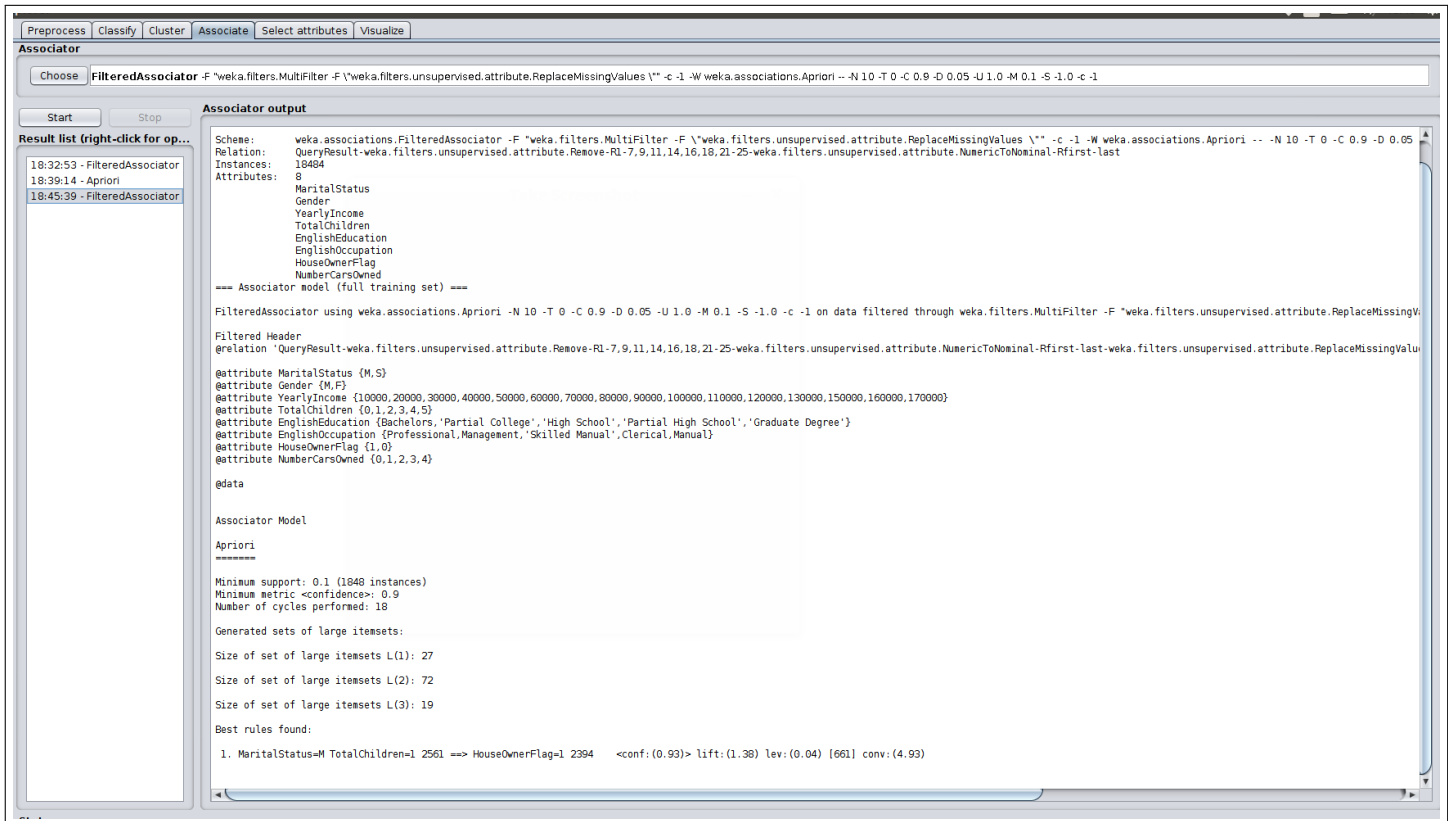


FIGURE 2.6 – Règles d'associations : FilteredAssociator résultat

3 | Conclusion