BE3 Clustering

Tariq CHELLALI

January 13, 2024

Contents

List of Figures

1 Travail A

1.1 EXERCICE I

	Outlook	Temperature	Humidity	windy	play
cluster 0	overcast	mild	high	True	yes
cluster 1	sunny	hot	high	False	no

Table 1: Centroides pour Farthest First

1.2 EXERCICE II

	Outlook	Temperature	Humidity	windy	play
cluster 0	sunny	mild	high	False	yes
cluster 1	overcast	hot	normal	True	yes

Table 2: Centroides pour SimpleKMeans

1.3 EXERCICE III

On observe que les centroides obtenus grâce aux deux algorithmes diffèrent en raison des disparités entre ces deux méthodes. FarthestFirst privilégie initialement les les centroides les plus dispersées et termine en une seule itération en affectant chaque instance au cluster ayant le centre le plus proche. En revanche, SimpleKmeans choisit aléatoirement les centres initiaux et les met à jour progressivement en assignant les instances au centre le plus proche, puis en recalculant les centres de chaque cluster.

Les valeurs de la variable "Play" en fonction de l'appartenance au cluster pour chaque instance sont représentées graphiquement. On note que, pour l'algorithme FarthestFirst, on a :

- Cluster 0 avec 8 (Play = oui) et 2 (Play = non)
- Cluster 1 avec 1 (Play = oui) et 3 (Play = non)

En ce qui concerne l'algorithme SimpleKmeans, on a :

- Cluster 0 avec 6 (Play = oui) et 4 (Play = non)
- Cluster 1 avec 3 (Play = oui) et 1 (Play = non)

Dans notre étude de cas, l'utilisation de l'algorithme FarthestFirst permet une meilleure distinction entre les instances ayant Play = oui et celles ayant Play = non.

1.4 EXERCICE IV

Outlook:	Overcast	Outlook:	Rainy
Windy:	False	Windy:	False
Outlook:	Overcast	Outlook:	Rainy
Windy:	True	Windy:	True

Table 3: Première table

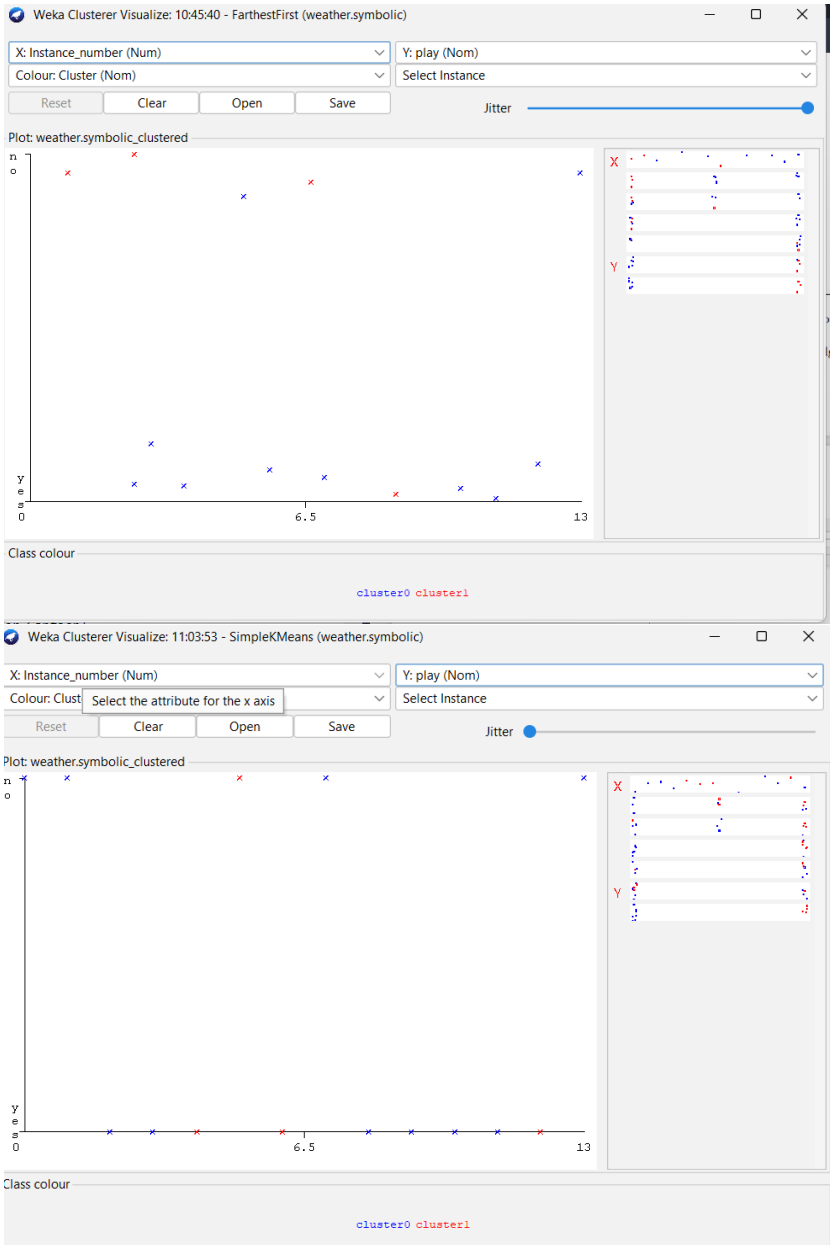


Figure 1: fig exo 4: En haut, FarthestFirst. En bas, SimpleKmeans

Outlook:	Sunny	Outlook;	Rainy
Windy:	False	False	False

Table 4: Deuxième table

1.5 EXERCICE V

On exécute l'algorithme SimpleKMeans avec seed = 20. Le centroïde du cluster 0 n'a pas subi de modification par rapport à notre première analyse avec seed = 10. Par contre, le centroïde du cluster 1 a évolué. Voici les centroides correspondant:

	Outlook	Temperature	Humidity	windy	play
cluster 0	sunny	mild	high	False	yes
cluster 1	sunny	cool	normal	True	no

Table 5: Centroides pour SimpleKMeans avec seed = 20 (exo5)

1.6 EXERCICE VI

Pour l'algorithme SimpleKMeans, le cluster 0 est constitué des instances avec les valueurs d'Outlook et Windy suivantes :

Outlook	sunny	Outlook	Overcast
Windy	False	Windy	False
Outlook	Rainy	Outlook	
Windy	False	windy	

Table 6: Table exo-6

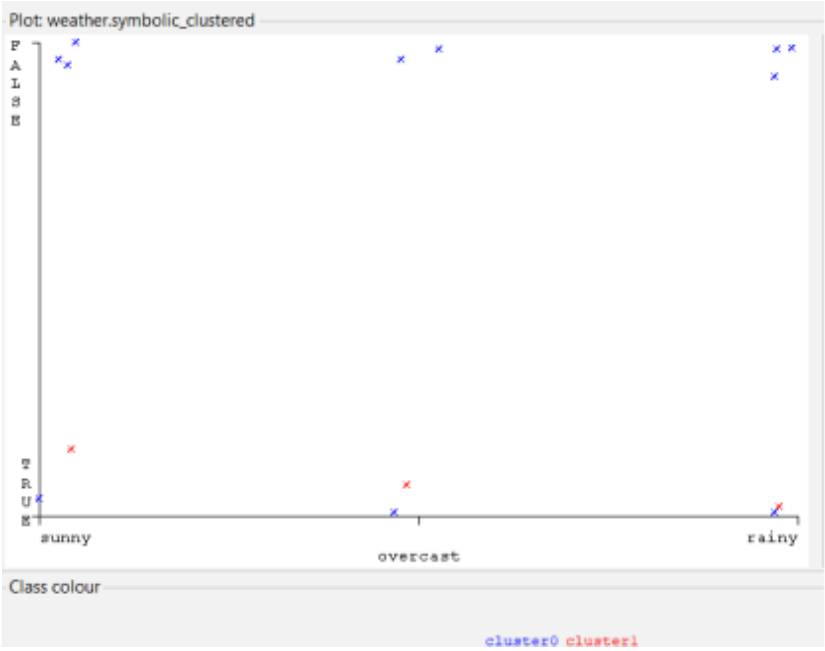


Figure 2: fig exo 6: SimpleKmeans (seed 20)

1.7 EXERCICE VII

On appliquant farthest first :

Outlook	sunny	Outlook	rainy
Windy	False	Windy	True

Table 7: fig exo 7

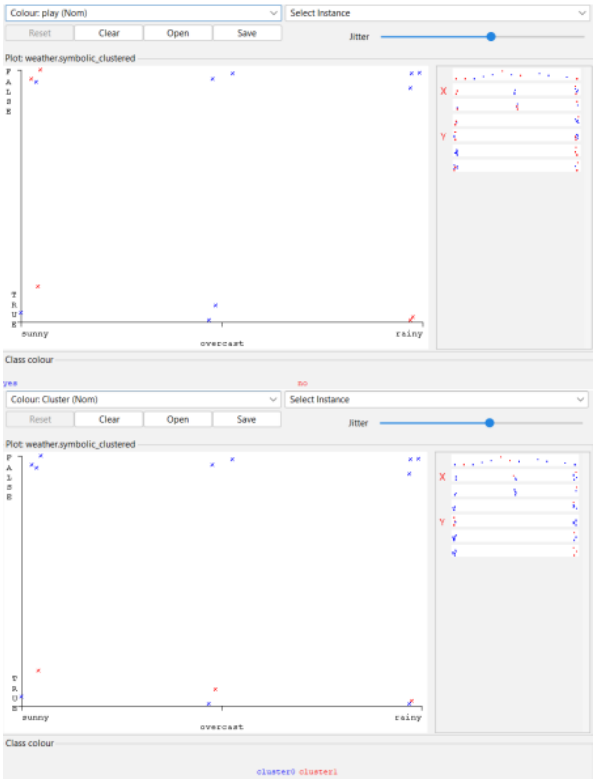


Figure 3: fig exo 7: En haut, couleur = Play. En bas, couleur = Cluster

1.8 EXERCICE VIII

De même pour l'algorithme SimpleKMeans. On obtient :

Outlook:	sunny	Outlook:	rainy
windy:	false	windy:	True
Outlook:	overcast	Outlook:	
Windy:	False	windy	

Table 8: table exo 8

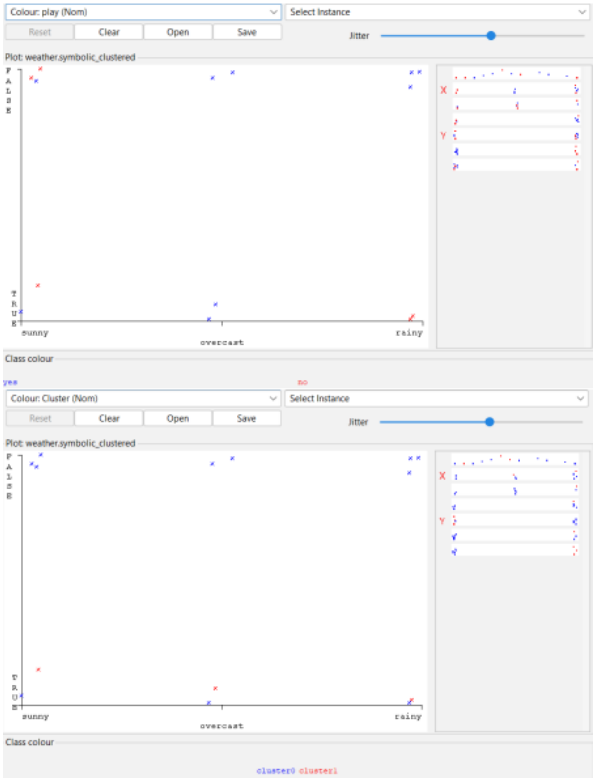


Figure 4: fig exo 8: En haut, couleur = Play. En bas, couleur = Cluster2

1.9 EXERCICE IX

On lance l'algorithme SimpleKMeans avec seed = 20 et numClusters = 2 et FathestFirst avec numClusters = 2 en sélectionnant l'option "Classes to clusters evaluation"

	FarthestFirst	SimpleKmeans
Faux Positifs	4	4
Faux négatifs Taux d'erreur	4 42,86%	4 42,86%

Table 9: table ex 9

1.10 EXERCICE X

L'algorithme SimpleKMeans avec seed = 1000 est l'algorithme qui génère le meilleur résultat avec 21% d'erreur

Algorithme	Seed	Taux d'err	play= yes Mal placées	Play=no Mal Placées
SimpleKmeans	1	36	3	2
SimpleKmeans	10	50	3	4
SimpleKmeans	20	42	4	4
SimpleKmeans	50	50	4	3
SimpleKmeans	100	36	3	2
SimpleKmeans	1000	21	2	1
FarthestFirst	1	42	4	4
FarthestFirst	10	35	2	3
FarthestFirst	20	42	4	4
FarthestFirst	50	36	3	2
FarthestFirst	100	35	2	3
FarthestFirst	1000	36	3	2

Table 10: table exo 10

1.11 EXERCICE XI

En visualisant les clusters:

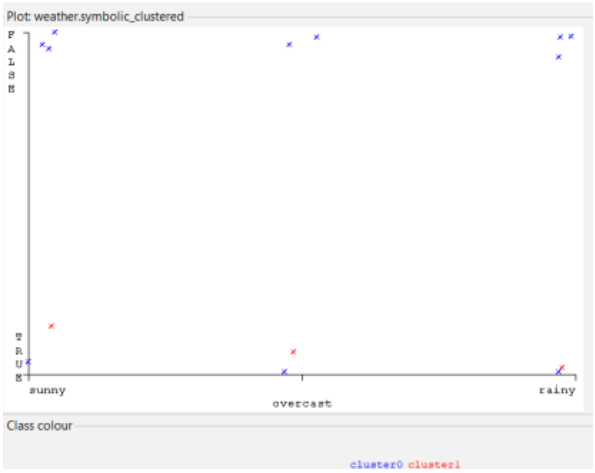


Figure 5: fig exo 11: Cluster assignement pour SimpleKMeans (seed = 1000)

Outlook	sunny	Outlook	Overcast
Windy	False	windy	False
Outlook	Rainy	Outlook	
Windy	False	Windy	

Table 11: table exo11

1.12 EXERCICE XII

La meilleure méthode semble être la 2, qui présente le taux d'erreur le plus faible (970) pour un nombre d'instances testées trois fois plus élevé (148) et donc un nombre de calculs de distance plus élevé. La méthode 1 présente un nombre élevé d'erreurs (1510) pour un grand nombre d'instances classées (400+), mais cela est normal, car l'essai a été effectué sur des données d'entraînement. Les méthodes 3 et 4 ont un taux d'erreur similaire (1300-1400) pour un nombre d'instances classées similaire (43-44).

Méthode	Train	Test	SSE
1	100%	100%	1510
	Cluster 0	214 (49%)	
	Cluster 1	221 (51%)	
2	66%	33%	970
	Cluster 0	73 (49%)	
	Cluster 1	75 (51%)	
3	90%	10%	1206
	Cluster 0	25 (58%)	
	Cluster 1	18 (42%)	
4	10%	90%	1510
	Cluster 0	170 (48%)	
	Cluster 1	181 (52%)	

Table 12: table exo 12

2 Travail B

2.1 EXERCICE XIII

Après avoir effectué les tests sur differentes valeurs de MinPoints 2, 3, 4 et 6 tout en laissant les autres paramètres par défaut , nous avons constaté que la valeur de minPoints qui donnait le meilleur taux d'erreur était 6, avec un taux d'erreur de 21% et 14 clusters.

Algo	MinPoints	Taux d'err	Class = Democrat mal placées	Class = Republican mal placées
DBSCAN	5	21%	45	50

Table 13: table exo 13

Nous remarquons qu'un grand nombre d'instances (313) n'ont pas pu être classées dans un cluster avec un MinPoint egale à . Cela peut être dû à la valeur de minPoints choisie qui est trop élevée et qui ne permet pas à ces instances de former un cluster valide

2.2 EXERCICE XIV

On fixant la valeur de minPoints à 2. On fait varier la valeur de entre 1 et 1.2. On constate que le meilleur taux d'erreur était obtenu pour = 1.2.

Algo	Epsilon	Taux d'err	Class = Democrat mal placées	Class = Republican mal placées
DBSCAN	1.2	9%	15	26

Table 14: table exo 14

2.3 EXERCICE XV

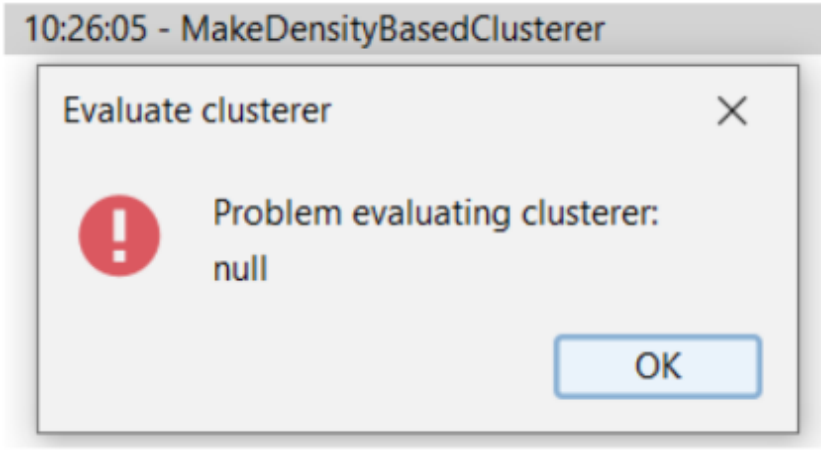
La méthode DBSCAN donne une erreur. A cause, du cluster vide.

On fait varier les paramètres des deux autres algorithmes et on note les meilleures métriques trouvées dans le tableau ci-dessous :

Algo	Paramétre	Meilleur LV	NB clusters	Taux d'erreur
SimpleKmean	numClusters = 2	-7.7	2	12.64%
FarthestFirst	numClusters = 2	-7.74	2	12.90%

Table 15: table exo 15

On note que les meilleurs algorithmes sont obtenus pour numClusters = 2.



2.4 EXERCICE XVI

Algo	Seed	Taux d'err	Play=yes Mal placées	Play=No Mal placées
EM	1000	35%	2	3

Table 16: table exo 16

Changer le seed, ne change rien. Cela s'explique par le petit nombre de clusters

2.5 EXERCICE XVII

Cluster	Outlook	Temperature	humidity	windy	Play
Cluster 0	rainy	70	81	True	No
Cluster 1	Overcast	82	84	False	Yes

Table 17: table exo 17

2.6 EXERCICE XVIII-XIX

Algo	Cutoff	Taux d'err	Play = Yes mal placées	Play = No Mal placées
CobWeb	0.3	35%	0	5

Table 18: table exo 18-19

2.7 EXERCICE XX

Les méthodes DBScan et EM ont été appliquées sur la base de données Labor. DBScan avec minpoints = 3 et epsilon = 1.1 a donné le résultat dans la figure ci-dessous. En comparant uniquement ces matrices de confusion, c'est DBScan qui nous a fourni le meilleur résultat.

```
Class attribute: class
Classes to Clusters:

  0  1  <-- assigned to cluster
  3  4 | bad
 29  0 | good

Cluster 0 <-- good
Cluster 1 <-- bad

Incorrectly clustered instances :       3.0       5.2632 %
Class attribute: class
Classes to Clusters:

  0  1  2  <-- assigned to cluster
  9  9  2 | bad
 29  1  7 | good

Cluster 0 <-- good
Cluster 1 <-- bad
Cluster 2 <-- No class

Incorrectly clustered instances :      19.0      33.3333 %
```

Figure 6: fig – En haut : DBScan. En bas : EM (exo 20)

2.8 EXERCICE XXI

Avec Hierarchical Clustering, en faisant uniquement varier le nombre de clusters, les résultats d'attribution sont un peu meilleurs (une instance mieux classée) avec un cluster plutôt que deux (toutes les instances attribuées à la même classe), même si avoir deux clusters a en réalité plus de sens

```
=== Model and evaluation on training set ===

Clustered Instances

0      14 (100%)


Class attribute: play
Classes to Clusters:

 0  <-- assigned to cluster
 9 | yes
 5 | no

Cluster 0 <-- yes

Incorrectly clustered instances :       5.0       35.7143 %
```

```
=== Model and evaluation on training set ===

Clustered Instances

0      13 ( 93%)
1       1 (  7%)


Class attribute: play
Classes to Clusters:

 0 1  <-- assigned to cluster
 8 1 | yes
 5 0 | no

Cluster 0 <-- yes
Cluster 1 <-- No class

Incorrectly clustered instances :       6.0       42.8571 %
```

3 Travail C

3.1 EXERCICE XXII

- **Groupe 1:** Cluster 0 et Cluster 1 ont des caractéristiques similaires, étant principalement composés de femmes vivant en zone rurale.
- **Groupe 2:** Cluster 4 et Cluster 5 ont des caractéristiques similaires avec des hommes principalement vivant en zone urbaine
- **Groupe3:** Les deux clusters 2 et 3 partagent des tendances similaires telles que la possession d'un compte courant et l'absence de prêt hypothécaire, mais diffèrent dans le revenu et la zone de résidence.

3.2 EXERCICE XXIII

Les détails fournis confirment en grande partie les observations précédentes, mais apportent des nuances supplémentaires. Voici quelques points clés :

Groupe 1:

- Les Clusters 0 et 1 sont principalement composés de femmes vivant en zone rurale.
- Les deux clusters ont des caractéristiques similaires en termes de sexe, région (RURAL), et certaines caractéristiques financières.

Groupe 2:

- Les Clusters 4 et 5 sont similaires, comprenant principalement des hommes vivant en zone urbaine.
- Les caractéristiques communes incluent le sexe (MALE), la région (INNER_CITY), et d'autres similitudes dans les attributs financiers.

Groupe 3:

- Les Clusters 2 et 3 partagent des similitudes, telles que la possession d'un compte courant, l'absence de prêt hypothécaire, et d'autres caractéristiques.
- La différence notable réside dans le revenu moyen, avec le Cluster 3 ayant un revenu moyen plus élevé que le Cluster 2.
- La région de résidence varie également, avec le Cluster 2 en INNER_CITY et le Cluster 3 en TOWN.

3.3 EXERCICE XXV

Parmi les 60 instances du jeu de test, 35 instances (58%) appartiennent au Cluster 0, et 25 instances (42%) appartiennent au Cluster 1.

On variant le parametre seed de 10-100, on remarque que la meilleur valeur du SSE est celle donnée par un seed égale à 50 avec un SSE

```
Number of iterations: 3
Sum of within cluster distances: 1745.336892101037
```

3.4 EXERCICE XXVI

En appliquant la méthode J48, nous retrouvons le résultat suivant:

```
J48 pruned tree
------------------

pep = YES
|   car = NO
|   |   sex = FEMALE: cluster1 (6.0)
|   |   sex = MALE: cluster0 (7.0)
|   car = YES: cluster0 (15.0)
pep = NO
|   car = NO: cluster1 (19.0)
|   car = YES
|   |   sex = FEMALE: cluster1 (7.0)
|   |   sex = MALE: cluster0 (6.0)

Number of Leaves  :  6

Size of the tree :  11


Time taken to build model: 0.03 seconds
```

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          58               96.6667 %
Incorrectly Classified Instances         2                3.3333 %
Kappa statistic                          0.9327
Mean absolute error                      0.0361
Root mean squared error                  0.1755
Relative absolute error                  7.2395 %
Root relative squared error             35.0999 %
Total Number of Instances               60

=== Detailed Accuracy By Class ===

TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area Class
0,929    0,000    1,000      0,929    0,963      0,935    0,993     0,987     0
1,000    0,071    0,941      1,000    0,970      0,935    0,993     0,989     1
0,967    0,038    0,969      0,967    0,967      0,935    0,993     0,988

=== Confusion Matrix ===

  a  b   <-- classified as
 26  2 |  a = cluster0
  0 32 |  b = cluster1
```

3.5 EXERCICE XXVII

On appliquant la méthode EM, avec un seed = 500 et nplis = 20: on trouve le résultat suivant avec une valeur de log-vraisamplance maximale:

```
    EM
==

Number of clusters selected by cross validation: 5
Number of iterations performed: 15


                 Cluster
Attribute              0          1          2          3          4
                  (0.18)     (0.29)     (0.19)     (0.05)     (0.29)
=============================================================================
age
  mean            61.1454    37.5106    25.4182    45.5993    46.5198
  std. dev.        4.7222    11.9476     5.3634      6.637     9.1348

sex
  FEMALE          56.9535    85.3787    58.2112    16.1923    88.2643
  MALE            51.9362    88.4345    60.4091    17.7938    86.4263
  [total]        108.8897   173.8133   118.6203    33.9861   174.6906
region
  INNER_CITY      51.3678    76.7333    59.6023     6.7372    79.5593
  TOWN            22.2461    54.6562    32.3035    16.9021    51.8921
  RURAL           24.2126    22.8039    17.8627     7.0485    29.0724
  SUBURBAN        13.0632    21.6199    10.8518     5.2983    16.1668
```

```
  [total]         110.8897   175.8133   120.6203    35.9861   176.6906
income
  mean          47926.8926 21474.0415 14967.6607 32108.1286 28545.0078
  std. dev.      7766.5667  7164.5079  4597.8763  7691.6908  7618.6299

married
  NO               33.3427    53.3108    40.5581    15.7436    66.0447
  YES               75.547   120.5024    78.0622    18.2425   108.6459
  [total]         108.8897   173.8133   118.6203    33.9861   174.6906
children
  mean              1.0881          0     1.6435      0.541     1.6314
  std. dev.         1.0473          0     0.8871     0.6102     0.9368

car
  NO                51.571     93.258     67.637     20.919    75.6149
  YES              57.3187    80.5552    50.9833    13.0671    99.0756
  [total]         108.8897   173.8133   118.6203    33.9861   174.6906
save_act
  NO                2.7273    78.7542    45.5862     3.6413    60.2909
  YES             106.1624     95.059    73.0342    30.3448   114.3996
  [total]         108.8897   173.8133   118.6203    33.9861   174.6906
current_act
  NO               23.3769    45.5695    25.0673    11.8629    44.1234
  YES              85.5128   128.2437    93.5531    22.1233   130.5672
  [total]         108.8897   173.8133   118.6203    33.9861   174.6906
mortgage
  NO               74.2151   110.2831    78.1268      7.871    125.504
  YES              34.6746    63.5301    40.4935    26.1152    49.1866
  [total]         108.8897   173.8133   118.6203    33.9861   174.6906
pep
  YES              66.8196    73.3686    30.3323    16.8148    91.6647
  NO               42.0701   100.4446    88.2881    17.1713    83.0259
  [total]         108.8897   173.8133   118.6203    33.9861   174.6906


Time taken to build model (full training data) : 3.8 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       72 ( 12%)
1      253 ( 42%)
2      102 ( 17%)
3        8 (  1%)
4      165 ( 28%)


Log likelihood: -17.68175
```

3.6 EXERCICE XXVIII

L'algorithme EM a automatiquement sélectionné 5 clusters en utilisant la validation croisée chacun caractérisé par des distributions spécifiques d'attributs. Avec une log likelihood de -17.68175

3.7 EXERCICE XXIX

Voici un tableau qui résume les résultats trouvées:

Algorithme de Clustering	Log Vraisemblance
SimpleKMeans	-21.99761
Canopy	-21.63505
Cobweb	-22.41547
FilteredClusterer (avec EM)	-21.99761
Hierarchical	-22.10968
EM	-21.28495

Table 19: Résumé de la log-vraisemblance pour chaque algorithme de clustering.

Dans ce cas, l'algorithme EM a la log-vraisemblance la plus élevée, suivi de SimpleKMeans, Hierarchical, Canopy, FilteredClusterer, et enfin Cobweb.

4 Travail D

4.1 EXERCICE XXXIII

On fait un clustering avec K-means en fixant le nombre de clusters à 3 clusters, sur le dataset iris-2D.arf.

Figure 7: fig exo 33: le nombre d'instances en fonction de Petal Length.

Il est notable que lorsque l'on utilise la longueur des pétales, les trois groupes sont clairement distingués, et il existe des variations minimales. De manière similaire, lorsqu'on considère la largeur des pétales, les différents groupes sont également bien séparés. Cela suggère que ces deux caractéristiques sont cruciales pour l'identification des spécimens d'iris. Si l'on représente graphiquement les trois classes en fonction des clusters, le résultat est le suivant :

Pour récapituler, presque chaque cluster correspond à une classe. Les résultats obtenus sont presque parfaits, bien qu'il existe quelques exceptions où la classe 3 est présente dans le cluster 2 (et vice versa). Dans cette analyse, l'attribut qui semble le plus distinctif entre les clusters est la largeur du pétale (petalwidth). Si nous effectuons une analyse similaire sur la base de données iris.arff, l'attribut qui semble le mieux séparer les clusters est la longueur du sépale (sepallength), même si les résultats obtenus ne sont pas totalement satisfaisants.

Figure 8: fig exo 33; bas, le nombre d'instances en fonction de Petal Width

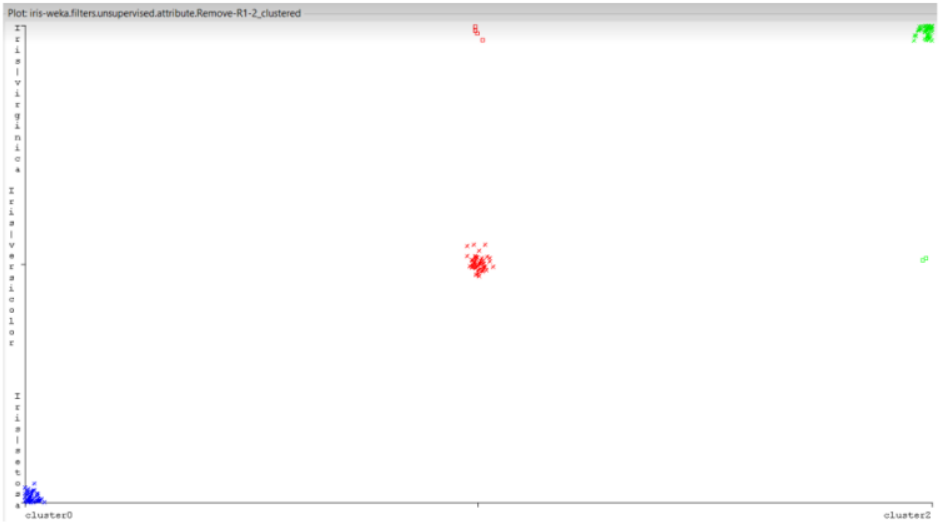


Figure 9: fig exo 33: Classes en fonction des clusters

4.2 EXERCICE XXXIV

Lorsque nous utilisons l'algorithme de clustering EM avec un nombre maximum de clusters réglé à -1 (permettant à la méthode de les trouver automatiquement), nous obtenons une valeur de Log Likelihood de -1.61. Cette valeur mesure la qualité de la répartition des données dans les clusters. Cependant, si nous réglons le nombre maximum de clusters à 3, la valeur de Log Likelihood augmente à -2.055. Cela indique qu'en utilisant un nombre maximum de clusters plus élevé, la répartition des données dans les clusters devient moins efficace.

Pour comparer les résultats obtenus avec ces deux approches, vous pouvez utiliser un tableau qui présente la correspondance entre les classes et les clusters pour chaque méthode. À gauche, vous avez la méthode avec le nombre maximum de clusters réglé automatiquement à -1, et à droite, vous avez la méthode avec un nombre maximum de clusters fixé à 3.

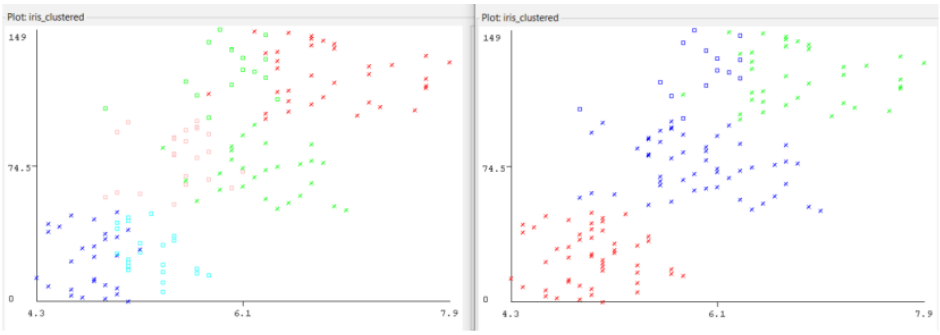


Figure 10: fig exo 34: Classes en fonction des clusters pour deux méthodes de choix du nombre de clusters

Dans les deux scénarios, on peut observer qu'un des clusters regroupe des données appartenant à deux classes différentes, ce qui signifie qu'il est composé de données provenant de deux classes distinctes. Plus précisément, le cluster 2 regroupe deux classes lorsque le nombre maximum de clusters est réglé à -1 pour la détermination automatique, tandis que le cluster 0 regroupe deux classes lorsque le nombre maximum de clusters est fixé à 3. Cette situation peut s'expliquer par la similarité entre ces deux classes, ce qui peut rendre difficile leur séparation lors de l'utilisation d'un nombre limité de clusters.

Les comparaisons des nombres d'instances par attributs en affichant les clusters produisent des résultats similaires, à l'exception du fait que les résultats sont répartis en deux clusters de plus lorsque le nombre maximum de clusters est réglé à -1 :

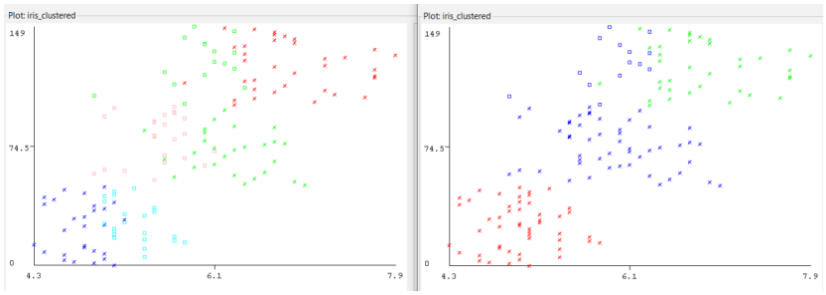


Figure 11: fig exo 34: Classes en fonction des clusters pour deux méthodes de choix du nombre de clusters

En résumé, les résultats obtenus en utilisant un nombre maximum de clusters réglé à -1 semblent préférables, car ils conduisent à une légèrement meilleure valeur de log vraisemblance. De plus, l'ajout des clusters lorsque le nombre maximum est fixé à 3 ne semble pas apporter une valeur significative supplémentaire.

4.3 EXERCICE XXXV

En appliquant la technique de clustering EM sur l'ensemble d'entraînement, qui représente 90 % des données, en utilisant l'ensemble de test fourni créé à partir de 10 % des données :

En examinant les visualisations, il est évident que toutes les instances ont été correctement classées. En d'autres termes, les instances sont réparties de manière uniforme dans les différents clusters, et il n'y a pas de valeurs aberrantes. Cela démontre que l'algorithme utilisé, à savoir le clustering EM, a bien fonctionné et a réussi à regrouper les données en clusters de similarité de manière satisfaisante.

2 - En ne prenant plus le test set de 10% mais avec la "Class to Cluster Evaluation" :

Lors de cette analyse, nous avons observé que 13 instances ont été mal classées, ce qui représente un nombre supérieur à celui de la première analyse. Toutefois, il est important de noter que la majorité de ces erreurs se sont produites au sein de la classe Iris-versicolor. Bien que ce clustering soit très performant pour les deux autres classes, il aurait été intéressant de le tester sur un nouvel ensemble

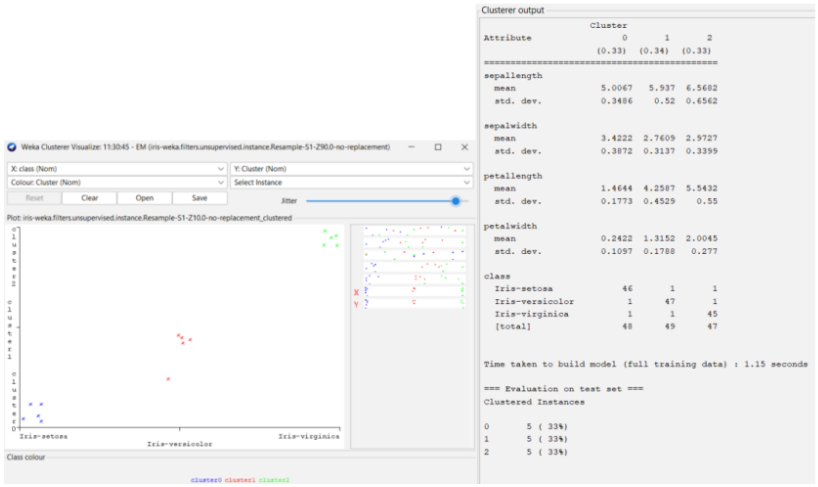


Figure 12: fig exo 35: Clusters en fonction des classes et statistiques pour 1

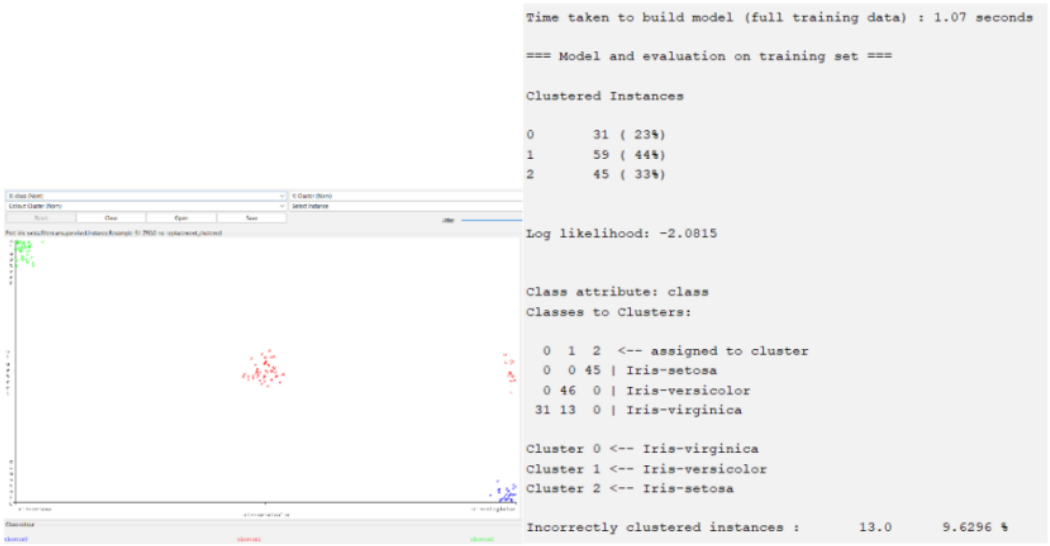


Figure 13: fig exo 35: Clusters en fonction des classes et statistiques pour 2

de données. Comparativement aux résultats précédents, il semble que les résultats obtenus ici soient meilleurs pour la classe Iris-versicolor, mais moins satisfaisants pour les deux autres classes, où quelques erreurs ont été observées dans les données d'entraînement.