Rapport du BE1: Indroduction à la data science

Tariq CHELLALI

November 7, 2023

1 Exercice (1.5)

Appliquez les méthodes 0R et 1R à la base de données "banque2-app.arff" et consignez les résultats. Testez vos modèles à la base de données "banque2-test.arff" et consignez les résultats

1.1 Méthode 0R sur la base de donnée "banque2-app.arff

On important les résultats de classification 0R à la base de données "banque2-app.arff", on génere le résultat suivant:

=== Run information ===

Scheme: weka.classifiers.rules.ZeroR Relation: internet Instances: 53 Attributes: 5 Montant Age Residence Etudes Internet Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===

```
ZeroR predicts class value: oui

Time taken to build model: 0 seconds

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

Correctly Classified Instances          32               60.3774 %
Incorrectly Classified Instances        21               39.6226 %
Kappa statistic                          0
Mean absolute error                      0.4799
Root mean squared error                  0.4897
Relative absolute error                100      %
Root relative squared error            100      %
Total Number of Instances               53

=== Detailed Accuracy By Class ===

TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
1,000    1,000    0,604      1,000    0,753      ?        0,455     0,582     oui
0,000    0,000    ?          0,000    ?          ?        0,455     0,375     non
0,604    0,604    ?          0,604    ?          ?        0,455     0,500     WA
WA = Weighted Avg.
=== Confusion Matrix ===

  a  b   <-- classified as
 32  0 |  a = oui
 21  0 |  b = non
```

Explication du résultat:

L'évaluation des performances est effectuée en utilisant une validation croisée à 10 plis pour mesurer la capacité du modèle à généraliser. 32 sur 53 Instances correctement classées et 21 sur 53 Instances incorrectement classées. La matrice de confusion montre que toutes les instances ont été classées comme "oui" (32 vrais positifs) et aucune comme "non" (0 vrais négatifs).

En résumé, le modèle 0R est très basique, car il prédit toujours la classe majoritaire, ce qui, dans ce cas, est "oui". Il obtient une précision de 60,3774%.

1.2 Méthode 1R sur la base de donnée "banque2-app.arff

```
    === Run information ===

Scheme:       weka.classifiers.rules.OneR -B 6
Relation:     internet
Instances:    53
Attributes:   5
              Montant
              Age
              Residence
              Etudes
              Internet
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

Residence:
village -> non
bourg -> oui
ville -> oui
banlieue -> oui
(38/53 instances correct)


Time taken to build model: 0 seconds

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

Correctly Classified Instances          38               71.6981 %
Incorrectly Classified Instances        15               28.3019 %
Kappa statistic                          0.35
Mean absolute error                      0.283
Root mean squared error                  0.532
Relative absolute error                 58.9793 %
Root relative squared error            108.6381 %
Total Number of Instances               53

=== Detailed Accuracy By Class ===

 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
 0,938    0,619    0,698      0,938    0,800      0,398    0,659     0,692     oui
 0,381    0,063    0,800      0,381    0,516      0,398    0,659     0,550     non
 0,717    0,399    0,738      0,717    0,688      0,398    0,659     0,636
WA = Weighted Avg.
=== Confusion Matrix ===

  a  b   <-- classified as
```

```
 30  2 |  a = oui
 13  8 |  b = non
```

Explication du résultat:

- La méthode 1R a été appliquée aux données avec un focus sur la caractéristique "Residence". Cette méthode a généré une règle simple qui associe certaines valeurs de cette caractéristique à des classes spécifiques. Par exemple, elle a associé "village" à "non", "bourg", "ville", et "banlieue" à "oui".

- Sur les 53 instances testées, 38 ont été correctement classées en utilisant cette règle, ce qui représente une précision de 71,6981 %. Cela signifie que la règle 1R a bien fonctionné pour la classification, mais il y avait encore 15 instances mal classées.

- La statistique Kappa a été calculée à 0,35, indiquant un certain niveau d'accord entre les prédictions et les étiquettes réelles.

- Les erreurs absolues moyennes et quadratiques moyennes ont été calculées respectivement à 0,283 et 0,532, ce qui mesure la magnitude des erreurs de classification.

- La matrice de confusion montre que la méthode 1R a correctement classé 30 instances comme "oui" et 13 instances comme "non". Cependant, elle a également commis des erreurs, classant 2 instances "non" comme "oui" et 8 instances "oui" comme "non".

En résumé, la méthode 1R a généré une règle simple basée sur la caractéristique "Residence" qui a bien fonctionné pour classer la plupart des instances, mais elle a tout de même commis des erreurs de classification. La précision globale était de 71,6981 %, ce qui indique une performance modérée.

2 Exercice (4.1)

On appliquant la methode SimpleCART à la base de donnée "weather-nominal.arff", nous retrouvons les résultats suivant :

```
    === Run information ===

Scheme:       weka.classifiers.trees.SimpleCart -M 2.0 -N 5 -U -C 1.0 -S 1
Relation:     weather.symbolic
Instances:    14
Attributes:   5
              outlook
              temperature
              humidity
              windy
              play
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

CART Decision Tree

outlook=(sunny)|(rainy)
|  humidity=(high)
|  |  outlook=(sunny)|(overcast): no(3.0/0.0)
|  |  outlook!=(sunny)|(overcast): yes(1.0/1.0)
|  humidity!=(high)
|  |  windy=(TRUE): yes(1.0/1.0)
|  |  windy!=(TRUE): yes(3.0/0.0)
outlook!=(sunny)|(rainy): yes(4.0/0.0)

Number of Leaf Nodes: 5
```

```
Size of the Tree: 9

Time taken to build model: 0 seconds

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

Correctly Classified Instances           9               64.2857 %
Incorrectly Classified Instances         5               35.7143 %
Kappa statistic                          0.2553
Mean absolute error                      0.4345
Root mean squared error                  0.5812
Relative absolute error                 91.25   %
Root relative squared error            117.8043 %
Total Number of Instances               14

=== Detailed Accuracy By Class ===

   TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
   0,667    0,400    0,750      0,667    0,706      0,258    0,622     0,692     yes
   0,600    0,333    0,500      0,600    0,545      0,258    0,622     0,504     no
WA 0,643    0,376    0,661      0,643    0,649      0,258    0,622     0,625 Weighted Avg.

=== Confusion Matrix ===

 a b   <-- classified as
 6 3 | a = yes
 2 3 | b = no
```

Le résultat nous montre que la précision pour la classe "yes" est de 75%, et la précision pour la classe "no" est de 50%. La table ci-dessous nous montre une comparaison entre les statistiques de chaque méthode :

Méthode d'arbre de décision	Précision (WA)	MAE	RMSE
J48	0,521	0,4167	0,5984
ID3	0,857	0,1429	0,378
SimpleCART	0,661	0,4345	0,5812

Table 1: Comparaison entre J48, ID3 et SimpleCART

En somme, les statistiques nous montres que la melleure méthode est celle avec une précision élevé, la méthode ID3 avec une précision égale à 0,857.

3 Exercices : la pratique de Weka sur l'exemple Banque

3.1 Banque: choix d'attribut

On suivant les instructions de l'exercice on retrouve le résultat suivant pour la méthode "InfoGainAttributeEval" (Entropie):

```
    === Run information ===

Evaluator:    weka.attributeSelection.InfoGainAttributeEval
Search:       weka.attributeSelection.Ranker -T -1.7976931348623157E308 -N -1
```

```
Relation:     internet
Instances:    53
Attributes:   5
              Montant
              Age
              Residence
              Etudes
              Internet
Evaluation mode:    evaluate on all training data



=== Attribute Selection on all input data ===

Search Method:
Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 5 Internet):
Information Gain Ranking Filter

Ranked attributes:
 0.15302  3 Residence
 0.05379  2 Age
 0.03947  4 Etudes
 0.00679  1 Montant

Selected attributes: 3,2,4,1 : 4
```

Explication: L'analyse de sélection d'attributs a été effectuée sur l'ensemble de données "internet" avec un total de 53 instances et 5 attributs. L'objectif était de déterminer les attributs les plus importants pour la prédiction de la classe "Internet". L'analyse a révélé que les attributs les plus importants, classés par ordre d'importance décroissante en fonction de leur gain d'information, sont les suivants :

Residence (Importance : 0.15302) Age (Importance : 0.05379) Etudes (Importance : 0.03947) Montant (Importance : 0.00679)

Comparaison: Si on applique l'analyse de sélection d'attributs en utilisant cette fois une méthode de sélection "GainRatioAttributeEval" et on compare le résultat avec la méthode "InfoGainAttributeEval" (Entropie) dans le tableau suivant:

Méthode de sélection d'attribut	**"Residence"**	**"Age"**	**"Études"**	**"Montant"**
InfoGainAttributeEval (Entropie)	0.15302	0.05379	0.03947	0.00679
GainRatioAttributeEval	0.07713	0.03482	0.02563	0.00473
Correlation Ranking Filter	0.1711	0.141	0.1217	0.05

Table 2: Comparaison entre les méthodes de sélection d'attributs dans Weka

InfoGainAttributeEval (Entropie)

- "Residence" est l'attribut le plus important avec un score de 0.15302.
- Suivi de près par "Age" avec un score de 0.05379.
- Ensuite, "Études" avec un score de 0.03947.
- Enfin, "Montant" avec un score de 0.00679.

GainRatioAttributeEval

- Les scores de GainRatioAttributeEval sont généralement plus faibles que ceux d'InfoGainAttributeEval, ce qui signifie que cette méthode considère les attributs comme moins importants.
- "Residence" est également le plus important avec un score de 0.07713.
- Les autres attributs ont des scores plus faibles par rapport à InfoGainAttributeEval.

Correlation Ranking Filter

- Cette méthode donne des scores différents par rapport aux deux premières méthodes, avec "Residence" étant l'attribut le plus important avec un score de 0.1711.
- "Age" et "Études" sont également évalués plus haut que dans les deux premières méthodes.
- "Montant" a un score relativement faible, bien qu'il soit un peu plus élevé que dans GainRatioAttributeEval.

Conclusion:
En conclusion, la méthode de sélection d'attributs "Correlation Ranking Filter" semble donner des scores plus élevés à la plupart des attributs par rapport aux autres méthodes, ce qui suggère qu'elle considère ces attributs comme plus importants pour la prédiction de la classe "Internet".

3.2 Exercice : trouver le meilleur attribut 'à la main'

Nous allons appliquer la formule du Gain sur la banque en utilisant le code fourni et un dénombrement à la main.

En calculant le nombre d'instance associé à chaque classe (yes/no)

- L'attribut Montant: [[17, 12], [8,6], [7, 3]]
- L'attribut Age: [[10, 2], [12,9], [10, 10]]
- L'attribut Residence: [[2, 8], [9,5], [12, 2], [9,6]]
- L'attribut Etudes: [[6, 8], [11,4], [15,9]]
- L'attribut Internet: [21, 32]

Le tableau ci-dessous résume le calcul du gain pour chaque attribut:

	Montant	Age	Residence	Etudes	Internet
Gain	0.006792	0.053793	0.15302181	0.039469	** Entropy égale à 0.940285

Table 3: Resultats du calcul sur la banque

Notons que nous retrouvons les meme résultats numériques trouvé à partir du logitiel Weka.

3.3 Exercice : arbre de d´ecision de l'exemple Banque

Les résultats de cette exercice sont pareils que l'exercice précédent.

4 Travail 1

4.1 ID3, J48 et CART sur la BD Titanic

4.1.1 Méthode ID3

```
    === Run information ===

Scheme:       weka.classifiers.trees.Id3
Relation:     relation
Instances:    2201
```

```
Attributes:   4
              class
              age
              sex
              survived
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

Id3


sex = male
|  class = 1st
|  |  age = adult: no
|  |  age = child: yes
|  class = 2nd
|  |  age = adult: no
|  |  age = child: yes
|  class = 3rd
|  |  age = adult: no
|  |  age = child: no
|  class = crew: no
sex = female
|  class = 1st
|  |  age = adult: yes
|  |  age = child: yes
|  class = 2nd
|  |  age = adult: yes
|  |  age = child: yes
|  class = 3rd
|  |  age = adult: no
|  |  age = child: no
|  class = crew: yes

Time taken to build model: 0.01 seconds

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

Correctly Classified Instances        1737               78.9187 %
Incorrectly Classified Instances       464               21.0813 %
Kappa statistic                          0.431
Mean absolute error                      0.3091
Root mean squared error                  0.394
Relative absolute error                 70.6739 %
Root relative squared error             84.2568 %
Total Number of Instances             2201

=== Detailed Accuracy By Class ===

TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
0,381    0,016    0,919      0,381    0,539      0,501    0,754     0,691     yes
0,984    0,619    0,769      0,984    0,863      0,501    0,754     0,829     no
0,789    0,424    0,817      0,789    0,759      0,501    0,754     0,784
```

```
=== Confusion Matrix ===

    a    b   <-- classified as
  271  440 |    a = yes
   24 1466 |    b = no
```

4.1.2 Méthode J48

```
=== Run information ===

Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     relation
Instances:    2201
Attributes:   4
              class
              age
              sex
              survived
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

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

sex = male
|   class = 1st
|   |   age = adult: no (175.0/57.0)
|   |   age = child: yes (5.0)
|   class = 2nd
|   |   age = adult: no (168.0/14.0)
|   |   age = child: yes (11.0)
|   class = 3rd: no (510.0/88.0)
|   class = crew: no (862.0/192.0)
sex = female
|   class = 1st: yes (145.0/4.0)
|   class = 2nd: yes (106.0/13.0)
|   class = 3rd: no (196.0/90.0)
|   class = crew: yes (23.0/3.0)

Number of Leaves  :  10

Size of the tree :  15


Time taken to build model: 0.04 seconds

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

Correctly Classified Instances        1737               78.9187 %
Incorrectly Classified Instances       464               21.0813 %
Kappa statistic                          0.429
Mean absolute error                      0.312
```

```
Root mean squared error                  0.3959
Relative absolute error                 71.3177 %
Root relative squared error             84.6545 %
Total Number of Instances             2201

=== Detailed Accuracy By Class ===

TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
0,376    0,013    0,930      0,376    0,535      0,503    0,746     0,680     yes
0,987    0,624    0,768      0,987    0,864      0,503    0,746     0,822     no
0,789    0,427    0,820      0,789    0,758      0,503    0,746     0,777

=== Confusion Matrix ===

    a    b   <-- classified as
  267  444 |    a = yes
   20 1470 |    b = no
```

Nous pouvons visualiser l'arbre de décision pour cette méthode:

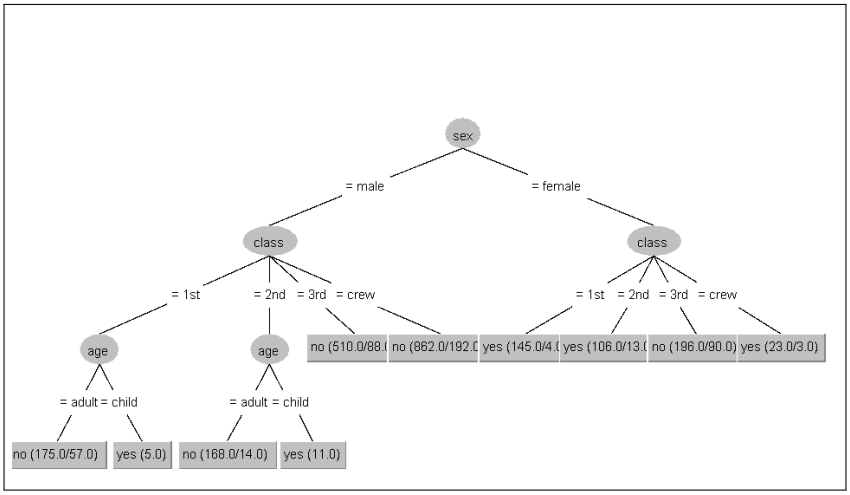

Figure 1: Arbre de décision de la BD Titanic via la méthode J48

4.1.3 Methode CART

```
=== Run information ===

Scheme:       weka.classifiers.trees.SimpleCart -M 2.0 -N 5 -C 1.0 -S 1
Relation:     relation
Instances:    2201
Attributes:   4
              class
              age
              sex
              survived
Test mode:    10-fold cross-validation
```

```
=== Classifier model (full training set) ===

CART Decision Tree

sex=(male)
|  age=(adult): no(1329.0/338.0)
|  age!=(adult)
|  |  class=(3rd)|(crew): no(35.0/13.0)
|  |  class!=(3rd)|(crew): yes(16.0/0.0)
sex!=(male)
|  class=(3rd): no(106.0/90.0)
|  class!=(3rd): yes(254.0/20.0)

Number of Leaf Nodes: 5

Size of the Tree: 9

Time taken to build model: 0.24 seconds

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

Correctly Classified Instances        1740               79.055  %
Incorrectly Classified Instances       461               20.945  %
Kappa statistic                          0.4334
Mean absolute error                      0.3138
Root mean squared error                  0.3973
Relative absolute error                 71.73   %
Root relative squared error             84.9592 %
Total Number of Instances             2201

=== Detailed Accuracy By Class ===

TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
0,380    0,013    0,931      0,380    0,539      0,506    0,726     0,655     yes
0,987    0,620    0,769      0,987    0,864      0,506    0,726     0,797     no
0,791    0,424    0,821      0,791    0,759      0,506    0,726     0,751

=== Confusion Matrix ===

    a    b   <-- classified as
  270  441 |    a = yes
   20 1470 |    b = no
```

4.2 ID3, J48 et CART sur la BD Cars

4.2.1 Méthode ID3

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

Correctly Classified Instances        1544               89.3519 %
Incorrectly Classified Instances        61                3.5301 %
Kappa statistic                          0.9071
Mean absolute error                      0.019
```

```
Root mean squared error                  0.1379
Relative absolute error                  9.4937 %
Root relative squared error             45.0502 %
UnClassified Instances                 123               7.1181 %
Total Number of Instances             1728

=== Detailed Accuracy By Class ===

TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
0,974    0,017    0,994      0,974    0,984      0,939    0,977     0,985     nacc
0,936    0,026    0,898      0,936    0,917      0,897    0,868     0,736     acc
1,000    0,006    0,830      1,000    0,907      0,909    0,836     0,574     tbon
0,787    0,008    0,755      0,787    0,771      0,764    0,764     0,423     bon
0,962    0,018    0,964      0,962    0,963      0,925    0,946     0,909

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 1171   28    0    3 |    a = nacc
    7  292    4    9 |    b = acc
    0    0   44    0 |    c = tbon
    0    5    5   37 |    d = bon
```

4.2.2 Méthode J48

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

Correctly Classified Instances        1596               92.3611 %
Incorrectly Classified Instances       132                7.6389 %
Kappa statistic                          0.8343
Mean absolute error                      0.0421
Root mean squared error                  0.1718
Relative absolute error                 18.3833 %
Root relative squared error             50.8176 %
Total Number of Instances             1728

=== Detailed Accuracy By Class ===

TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
0,962    0,064    0,972      0,962    0,967      0,892    0,983     0,992     nacc
0,867    0,047    0,841      0,867    0,854      0,811    0,962     0,859     acc
0,892    0,011    0,763      0,892    0,823      0,818    0,995     0,808     tbon
0,594    0,011    0,695      0,594    0,641      0,629    0,918     0,593     bon
0,924    0,056    0,924      0,924    0,924      0,861    0,976     0,940

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 1164   43    0    3 |    a = nacc
   33  333    7   11 |    b = acc
    0    3   58    4 |    c = tbon
    0   17   11   41 |    d = bon
```

4.2.3 Méthode SimpleCART

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

Correctly Classified Instances        1596               92.3611 %
Incorrectly Classified Instances       132                7.6389 %
Kappa statistic                          0.8343
Mean absolute error                      0.0421
Root mean squared error                  0.1718
Relative absolute error                 18.3833 %
Root relative squared error             50.8176 %
Total Number of Instances             1728

=== Detailed Accuracy By Class ===

TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
0,962    0,064    0,972      0,962    0,967      0,892    0,983     0,992     nacc
0,867    0,047    0,841      0,867    0,854      0,811    0,962     0,859     acc
0,892    0,011    0,763      0,892    0,823      0,818    0,995     0,808     tbon
0,594    0,011    0,695      0,594    0,641      0,629    0,918     0,593     bon
0,924    0,056    0,924      0,924    0,924      0,861    0,976     0,940

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 1164   43    0    3 |    a = nacc
   33  333    7   11 |    b = acc
    0    3   58    4 |    c = tbon
    0   17   11   41 |    d = bon
```

5 Travail 2: Méthode Bayésienne

5.1 Question 1

La raison pour laquelle le lissage de Laplace a été appliqué est que les occurrences de la combinaison "skin=hairy" et "class=0" étaient nulles dans la base de données d'entraînement, ce qui a conduit à une probabilité conditionnelle nulle ($P(\text{class} = 0|\text{skin} = \text{hairy})$).

Le lissage de Laplace a résolu ce problème en ajoutant une constante aux occurrences de chaque combinaison attribut-classe, garantissant ainsi qu'aucune probabilité conditionnelle ne soit nulle. Cela permet d'éviter les problèmes liés à la multiplication de probabilités nulles dans le calcul des probabilités conditionnelles. En ajoutant cette constante, le modèle a pu attribuer une probabilité non nulle à la combinaison "skin=hairy" et "class=0", ce qui a permis d'éviter l'annulation de la somme des probabilités.

```
                   Class
Attribute              0      1
                   (0.44) (0.56)
================================
skin
  hairy                1.0    5.0
  smooth               3.0    1.0
  [total]              4.0    6.0

colour
```

```
  brown               1.0    2.0
  green               2.0    3.0
  red                 3.0    1.0
  [total]             6.0    6.0

size
  small               3.0    1.0
  large               2.0    4.0
  [total]             5.0    5.0

flesh
  soft                4.0    2.0
  hard                1.0    4.0
  [total]             5.0    6.0

eats_shortbread
  0                   3.0    2.0
  1                   2.0    4.0
  [total]             5.0    6.0

length
  mean             1.9667 3.8104
  std. dev.        0.4014 0.4077
  weight sum            3      4
  precision        0.4917 0.4917
```

5.2 Question 2

Dans cette exemple, la matrice de confusion trouvée est:

```
=== Confusion Matrix ===

 a b   <-- classified as
 1 1 | a = 0
 0 1 | b = 1
```

Le modèle a prédit deux instances au total : une dans la classe 0 et une dans la classe 1. Cependant, une instance qui appartenait réellement à la classe "a" a été mal classée comme "0", tandis qu'une autre instance qui appartenait réellement à la classe "b" a été correctement classée comme "1".

Pour résumer :

- Vrai positif (VP) : Le modèle a correctement prédit la classe "1" pour un échantillon réel de classe "b".
- Faux positif (FP) : Le modèle a prédit la classe "1" pour un échantillon réel de classe "a".
- Vrai négatif (VN) : Le modèle a correctement prédit la classe "0" pour un échantillon réel de classe "a".
- Faux négatif (FN) : Il n'y a pas de faux négatif dans ce cas car toutes les instances de la classe "b" ont été correctement prédites.

5.3 Question 3

5.3.1 Question 3-1

Les données manquantes sont :

- Instance 2 attribut size
- Instance 3 attribut skin
- Instance 7 attribut colour

No.	1: skin Nominal	2: colour Nominal	3: size Nominal	4: flesh Nominal	5: eats_shortbread Nominal	6: length Numeric	7: **is_haggis** Nominal
1	hairy	brown	large	hard	1	3.25	1
2	hairy	green		hard	1	4.22	1
3		red	small	soft	0	1.27	0
4	hairy	green	large	hard	1	3.55	1
5	smooth	red	small	soft	0	2.13	0
6	smooth	green	large	soft	1	2.67	0
7	hairy		large	soft	0	3.77	1

Figure 2: Jeux de données avec les données manquantes

5.3.2 Question 3-2

Lorsque des données manquantes sont présentes dans un ensemble de données, weka les traite en les ignorant lors du calcul des probabilités conditionnelles. Le lissage de laplace n'est pas une méthode pour traiter les données manquantes.

5.3.3 Question 3-3

Weka traite les attributs numériques en effectuant une normalisation ou une standardisation, sans faire d'hypothèse spécifique sur la distribution des données numériques.

5.4 Question 4

Données fournies :

$$P(\text{classe} = 0) = \frac{3}{7}$$
$$P(\text{classe} = 1) = \frac{4}{7}$$

Attributs de l'instance x :

skin=smooth, color=red, size=large, flesh=hard, eats shortbread=yes, length=3.25

Calcul des probabilités conditionnelles :

$$P(\text{skin=smooth}|\text{classe} = 0) = \frac{3}{4}$$
$$P(\text{skin=smooth}|\text{classe} = 1) = \frac{1}{6}$$
$$P(\text{color=red}|\text{classe} = 0) = \frac{3}{6}$$
$$P(\text{color=red}|\text{classe} = 1) = \frac{1}{6}$$
$$P(\text{size=large}|\text{classe} = 0) = \frac{2}{5}$$
$$P(\text{size=large}|\text{classe} = 1) = \frac{4}{5}$$
$$P(\text{flesh=hard}|\text{classe} = 0) = \frac{1}{5}$$
$$P(\text{flesh=hard}|\text{classe} = 1) = \frac{2}{5}$$
$$P(\text{eats shortbread=yes}|\text{classe} = 0) = \frac{2}{5}$$
$$P(\text{eats shortbread=yes}|\text{classe} = 1) = \frac{4}{6}$$

5.5 Question 5

On utilisant les résultats:

```
                          Class
    Attribute              0      1
                       (0.44) (0.56)
    ================================
length
      mean             1.9667 3.8104
      std. dev.        0.4014 0.4077
      weight sum            3      4
      precision        0.4917 0.4917
```

Nous allons calculés la densité de probabilité en utilisant la formule du cours.
Pour la classe "No Haggis" ou "class 0"

$$FDP(\text{length} = 3.25|\text{Haggis}) = \frac{1}{0.4014 \cdot \sqrt{2\pi}} \cdot \exp\left(-\frac{(3.25 - 1.9667)^2}{2 \cdot (0.4014)^2}\right)$$

$$FDP(\text{length} = 3.25|\text{Haggis}) \approx 0.00596$$

Pour la classe "Haggis" ou "class 1" :

$$FDP(\text{length} = 3.25|\text{No Haggis}) = \frac{1}{0.4077 \cdot \sqrt{2\pi}} \cdot \exp\left(-\frac{(3.25 - 3.8104)^2}{2 \cdot (0.4077)^2}\right)$$

$$FDP(\text{length} = 3.25|\text{No Haggis}) \approx 0.38045$$

Par suite, on normalise:

$$P(\text{length} = 3.25|\text{Haggis}) = \frac{0.00596}{0.00596 + 0.38045}$$

$$P(\text{length} = 3.25|\text{Haggis}) \approx 0.015$$

et:

$$P(\text{length} = 3.25|\text{ No Haggis}) = \frac{0.38045}{0.00596 + 0.38045}$$

$$P(\text{length} = 3.25|\text{No Haggis}) \approx 0.980$$

Calcul des probabilités postérieures conditionnelles :

$$\begin{aligned}
P'(\text{classe} = 0|x) &= P(\text{classe} = 0)\times \\
P(\text{skin=smooth}|\text{classe} = 0)&\times \\
P(\text{color=red}|\text{classe} = 0)&\times \\
P(\text{size=large}|\text{classe} = 0)&\times \\
P(\text{flesh=hard}|\text{classe} = 0)&\times \\
P(\text{eats shortbread=yes}|\text{classe} = 0)&\times \\
P(\text{length=3.25}|\text{classe} = 0)& \\
&= \left(\frac{3}{7}\right) \times \left(\frac{3}{4}\right) \times \left(\frac{1}{2}\right) \times \left(\frac{2}{5}\right) \times \left(\frac{1}{5}\right) \times \left(\frac{2}{5}\right) \times 0.980 \\
&\approx 5.04 \times 10^{-3} \\
P'(\text{classe} = 1|x) &= P(\text{classe} = 1)\times \\
P(\text{skin=smooth}|\text{classe} = 1)&\times \\
P(\text{color=red}|\text{classe} = 1)&\times \\
P(\text{size=large}|\text{classe} = 1)&\times \\
P(\text{flesh=hard}|\text{classe} = 1)&\times \\
P(\text{eats shortbread=yes}|\text{classe} = 1)&\times \\
P(\text{length=3.25}|\text{classe} = 1)& \\
&= \left(\frac{4}{7}\right) \times \left(\frac{1}{6}\right) \times \left(\frac{1}{6}\right) \times \left(\frac{4}{5}\right) \times \left(\frac{2}{5}\right) \times \left(\frac{4}{6}\right) \times 0.015 \\
&\approx 5.079 \times 10^{-5}
\end{aligned}$$

On normalisant :

$$P(\text{classe} = 1|x) = \frac{P'(\text{classe} = 1|x)}{P'(\text{classe} = 1|x) + P'(\text{classe} = 0|x)} = 0.99$$

$$P(\text{classe} = 0|x) = \frac{P'(\text{classe} = 0|x)}{P'(\text{classe} = 1|x) + P'(\text{classe} = 0|x)} = 0.0099$$

Comparaison des probabilités postérieures conditionnelles : Puisque $P(\text{classe} = 1|x)$ est plus élevé que $P(\text{classe} = 0|x)$, l'instance x serait classée dans la classe haggis.

5.6 Question 6

5.6.1 Question 6-a

Weka prédit l'instance x comme étant haggis.

5.6.2 Question 6-b

L'analyse des résultats de notre modèle de classification bayésienne naïve révèle les éléments suivants:

- Taux de succès de la prédiction : Le modèle a correctement classé 3 instances sur un total de 4, ce qui représente un taux de succès de 75 %.

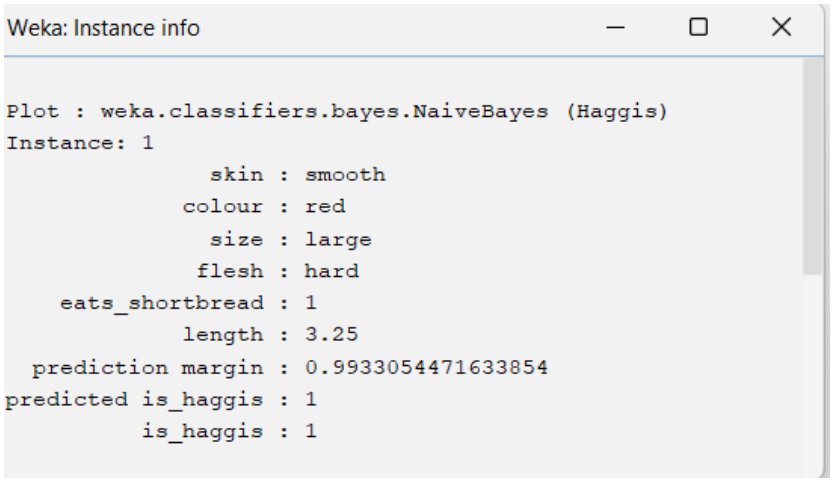

Figure 3: Résultats de prédiction de l'instance x par Weka

- Matrice de confusion :

  a = 0 : Cela signifie qu'il y a un total de 1 échantillon appartenant à la classe "a" dans le jeu de données, et le modèle a correctement prédit cet échantillon comme appartenant à la classe "a".

  b = 1 : Cela signifie qu'il y a un total de 3 échantillons appartenant à la classe "b" dans le jeu de données. Le modèle a correctement prédit 2 de ces échantillons comme appartenant à la classe "b", mais il a fait une prédiction incorrecte en classant un échantillon de la classe "a" comme appartenant à la classe "b".

a	b	¡– classifié comme
1	0	$a = 0$
1	2	$b = 1$

Conclusion En conclusion, notre modèle de classification bayésienne naïve a obtenu un taux de succès de 75 % sur l'ensemble de test.

6 Bilan du BE

Les méthodes ID3 et C4.5 sont toutes deux des algorithmes d'arbres de décision, mais C4.5 est une amélioration de l'ID3 qui introduit plusieurs améliorations et fonctionnalités supplémentaires. Voici les différences clés entre ces deux méthodes :

ID3 (Iterative Dichotomiser 3) :

- **Manipulation d'attributs :** ID3 est conçu pour les attributs catégoriels. Il fonctionne exclusivement avec des attributs catégoriels (discrétisation nécessaire pour les attributs numériques).
- **Gestion des valeurs manquantes :** L'ID3 ne gère pas bien les valeurs manquantes. Si des valeurs manquantes sont présentes, il peut interrompre le processus de construction de l'arbre.
- **Critère de sélection :** L'ID3 utilise le gain d'information comme critère de sélection des attributs pour diviser les données.

C4.5 (Successor of ID3) / J48 (implémentation de C4.5 dans Weka) :

- **Manipulation d'attributs :** C4.5 prend en charge à la fois les attributs catégoriels et numériques. Contrairement à l'ID3, il peut gérer nativement les attributs numériques sans nécessiter de discrétisation préalable.

- **Gestion des valeurs manquantes :** C4.5 est capable de gérer les valeurs manquantes dans les attributs en utilisant des techniques spécifiques pour prendre en compte ces valeurs manquantes dans la construction de l'arbre de décision.
- **Pruning (élagage) :** C4.5 intègre une étape d'élagage dans le processus de construction de l'arbre pour réduire le surajustement (overfitting) et améliorer la généralisation du modèle.
- **Autres critères de sélection :** En plus du gain d'information utilisé par l'ID3, C4.5 utilise également le gain ratio comme critère de sélection des attributs, ce qui prend en compte la taille des ensembles d'attributs.
- **Gestion des coûts :** C4.5 peut prendre en compte les coûts inégaux des erreurs de classification pour améliorer la performance du modèle.
- **Sélection des feuilles :** Contrairement à l'ID3, C4.5 utilise des mécanismes plus sophistiqués pour déterminer quand arrêter la croissance de l'arbre, ce qui contribue à éviter le surajustement.

En résumé, C4.5 (J48 dans Weka) présente des améliorations significatives par rapport à l'ID3, notamment en termes de manipulation des attributs numériques, de gestion des valeurs manquantes, d'élagage de l'arbre et de critères plus sophistiqués pour la construction de l'arbre.