BE2 Inroduction à la data science

Tariq CHELLALI

December 23, 2023

Contents

1 Travail à rendre (1): RePTree.

1.1 Question 1

1.1.1 BD "cars.arff"

Voici le résultats des premiére lignes de la methode REPtree apliquée à la BD *"cars.arff"* sans elagage.

```
REPTree
============

Surete = bas : nacc (576/0) [0/0]
Surete = moyen
|   Npers = 2.0 : nacc (192/0) [0/0]
|   Npers = 4.0
|   |   Pachat = thaut
|   |   |   Pmaint = thaut : nacc (12/0) [0/0]
|   |   |   Pmaint = haut : nacc (12/0) [0/0]
|   |   |   Pmaint = moyen
|   |   |   |   Tcoffre = petit : nacc (4/0) [0/0]
|   |   |   |   Tcoffre = moyen : nacc (4/2) [0/0]
|   |   |   |   Tcoffre = grand : acc (4/0) [0/0]
|   |   |   Pmaint = bas
|   |   |   |   Tcoffre = petit : nacc (4/0) [0/0]
|   |   |   |   Tcoffre = moyen : nacc (4/2) [0/0]
|   |   |   |   Tcoffre = grand : acc (4/0) [0/0]
|   |   Pachat = haut
|   |   |   Tcoffre = petit : nacc (16/0) [0/0]
|   |   |   Tcoffre = moyen
|   |   |   |   Nportes = 2.0 : nacc (4/0) [0/0]
|   |   |   |   Nportes = 3.0 : nacc (4/0) [0/0]
|   |   |   |   Nportes = 4.0 : acc (4/1) [0/0]
|   |   |   |   Nportes = 5plus : acc (4/1) [0/0]
|   |   |   Tcoffre = grand
```

Si on prend par exemple la première ligne,

```
Surete = bas : nacc (576/0) [0/0]
```

Surete = bas : Cela signifie que la première division (ou nœud) de l'arbre est basée sur la caractéristique "Surete" et la condition est que "Surete" doit être égal à "bas".

nacc (576/0) [0/0] : Cela indique les résultats de cette division.

"nacc" est la classe prédite lorsque la condition est vraie (Surete = bas). Ensuite, entre parenthèses, les statistiques pour cette prédiction. Dans ce cas, il y a 576 instances qui satisfont cette condition et sont correctement classées comme "nacc". Le nombre d'instances mal classées est 0. Les chiffres entre crochets ([0/0]) indique les statistiques liée à l'elagage. 0/0 est normal compte tenu du fait que nous avons désactiver l'elagage Ainsi, la première ligne de l'arbre nous dit que si la caractéristique "Surete" est égale à "bas", alors toutes les instances correspondantes (576 dans ce cas) seront classées dans la classe "nacc". Cela représente la première décision que l'arbre prend lorsqu'il évalue une nouvelle instance. En fonction de la valeur de "Surete", l'algorithme suit la branche appropriée de l'arbre pour effectuer des sous-divisions supplémentaires jusqu'à atteindre les feuilles de l'arbre où la classe finale est attribuée.

Voici le résultat de l'evaluation:

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

Correctly Classified Instances        1614               93.4028 %
Incorrectly Classified Instances       114                6.5972 %
Kappa statistic                          0.8557
Mean absolute error                      0.0352
Root mean squared error                  0.1611
Relative absolute error                 15.3719 %
Root relative squared error             47.639  %
Total Number of Instances             1728

=== Detailed Accuracy By Class ===

TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
0,970    0,075    0,968      0,970    0,969      0,896    0,977     0,986     nacc
0,862    0,032    0,885      0,862    0,873      0,838    0,957     0,856     acc
0,892    0,008    0,806      0,892    0,847      0,842    0,996     0,824     tbon
0,739    0,011    0,739      0,739    0,739      0,728    0,921     0,643     bon
0,934    0,061    0,934      0,934    0,934      0,875    0,971     0,937

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 1174   33    0    3 |    a = nacc
   37  331    5   11 |    b = acc
    0    3   58    4 |    c = tbon
    2    7    9   51 |    d = bon
```

Pour la classe a (nacc) :

- 1174 instances de la classe nacc ont été correctement classées comme nacc.
- 33 instances de la classe acc ont été incorrectement classées comme nacc.
- Aucune instance des classes tbon (c) et bon (d) n'a été classée comme nacc.

Pour la classe b (acc) :

- 331 instances de la classe acc ont été correctement classées comme acc.
- 37 instances de la classe nacc ont été incorrectement classées comme acc.
- 5 instances des classes tbon (c) et 11 instances de la classe bon (d) ont été classées comme acc.

Pour la classe c (tbon) :

- 58 instances de la classe tbon ont été correctement classées comme tbon.
- Aucune instance des classes nacc, acc, et bon n'a été classée comme tbon.

Pour la classe d (bon) :

- 51 instances de la classe bon ont été correctement classées comme bon.
- 2 instances de la classe nacc, 7 instances de la classe acc, et 9 instances de la classe tbon ont été incorrectement classées comme bon.

1.1.2 BD "vote.arff"

De la meme maniére, en desactivant l'elagage, on retrouve le résultat suivant:

```
REPTree
============

physician-fee-freeze = n
|   adoption-of-the-budget-resolution = n
|   |   synfuels-corporation-cutback = n
|   |   |   superfund-right-to-sue = n : democrat (6.14/2.03) [0/0]
|   |   |   superfund-right-to-sue = y : democrat (4.21/0.08) [0/0]
|   |   synfuels-corporation-cutback = y
|   |   |   handicapped-infants = n
|   |   |   |   crime = n : democrat (2.01/0) [0/0]
|   |   |   |   crime = y : democrat (5.11/0.05) [0/0]
|   |   |   handicapped-infants = y : democrat (8.19/0.02) [0/0]
|   adoption-of-the-budget-resolution = y
|   |   education-spending = n
|   |   |   crime = n : democrat (158.85/0.37) [0/0]
```

Pour expliquer ce résultat et les sortis associées, prenons l'exemple de cette branche:

```
crime = n : democrat (2.01/0) [0/0]
|   |   |   |   crime = y : democrat (5.11/0.05) [0/0]
```

- Pour la Condition : crime = n Si la caractéristique "crime" est égale à "non", le modèle prédit "démocrate". Les nombres entre parenthèses (2.01/0) indiquent les statistiques associées à cette prédiction. Le premier nombre (2.01) représente le poids associé à cette prédiction, ce qui peut être interprété comme une sorte de confiance ou de certitude que le modèle a dans cette prédiction. Le deuxième nombre (0) semble indiquer le nombre d'erreurs associées à cette prédiction. Cela pourrait signifier qu'aucune erreur n'a été commise pour cette prédiction dans l'ensemble d'entraînement.

- Pour la condition : crime = y Si la caractéristique "crime" est égale à "oui", le modèle prédit également "démocrate". Les nombres entre parenthèses (5.11/0.05) suivent la même logique que précédemment. Le premier nombre (5.11) représente le poids associé à cette prédiction. Le deuxième nombre (0.05) pourrait représenter une certaine mesure d'incertitude ou de variabilité associée à cette prédiction.

NOTE!!! Avec cette base de donnée on remarque qu'on a des sorties contenant des valeurs réelles où l'on attend un nombre d'instances (un entier).

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

Correctly Classified Instances         417               95.8621 %
Incorrectly Classified Instances        18                4.1379 %
Kappa statistic                          0.9127
Mean absolute error                      0.0567
Root mean squared error                  0.1843
Relative absolute error                 11.9461 %
Root relative squared error             37.8431 %
Total Number of Instances              435

=== Detailed Accuracy By Class ===
```

```
TP Rate   FP Rate   Precision   Recall   F-Measure   MCC      ROC Area    Class
0,966     0,054     0,966       0,966    0,966       0,913    0,983       democrat
0,946     0,034     0,946       0,946    0,946       0,913    0,985       republican
0,959     0,046     0,959       0,959    0,959       0,913    0,984

=== Confusion Matrix ===

   a   b   <-- classified as
 258   9 |   a = democrat
   9 159 |   b = republican
```

En résumé, le modèle REPTree semble avoir une bonne performance avec une précision globale élevée et une matrice de confusion montrant des résultats prometteurs pour la classification des différentes classes. Par contre, la taille de l'arbre indique que le modèle a une certaine complexité. D'ou l'utilité de l'élagage qui permet d'avoir un compromie de performance.

1.2 Questions 2

En appliquant le méme modèle à la BD "vin.arff", on trouve la courbe suivante:

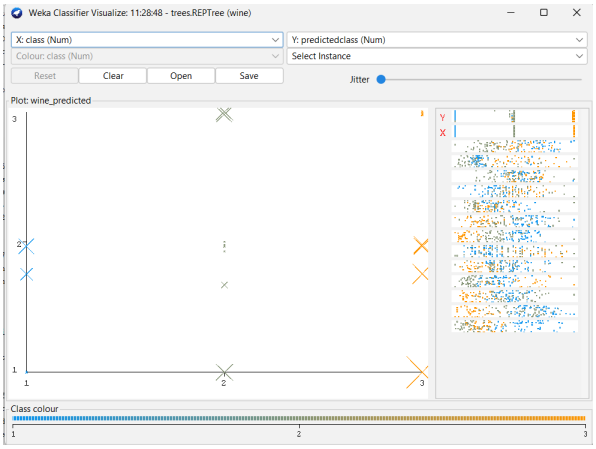


Figure 1: error classifier de RepTree avec la BD "vin"

La courbe montre les erreurs de classification dans notre modèle. En effet, pultot d'avoir une courbe y=x on remarque des clusters entre classes.

1.3 Bilan du travail (1)

- Sans élagage, les arbres de régression ont tendance à s'ajuster trop précisément aux données d'entraînement, ce qui peut conduire à une mauvaise généralisation sur de nouvelles données. Cela signifie que le modèle peut ne pas bien fonctionner sur des données qu'il n'a pas vu auparavant.

- Sensibilité aux petites variations des données : Les arbres de régression sont souvent sensibles aux petites variations dans les données d'entraînement, ce qui peut entraîner une instabilité dans la structure de l'arbre.

2 Exercice (4.2): Règles de Classification

En appliquant la méthode Ripper (JRip sous Weka) 'a la BD "meteo" (weather-nominal.arff"). On retrouve les résultats suivants:

```
=== Run information ===
```

```
Scheme:       weka.classifiers.rules.JRip -F 3 -N 2.0 -O 2 -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) ===

JRIP rules:
===========

(humidity = high) and (outlook = sunny) => play=no (3.0/0.0)
(outlook = rainy) and (windy = TRUE) => play=no (2.0/0.0)
 => play=yes (9.0/0.0)

Number of Rules : 3


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                          0.186
Mean absolute error                      0.3674
Root mean squared error                  0.5338
Relative absolute error                 77.1635 %
Root relative squared error            108.1927 %
Total Number of Instances               14

=== Detailed Accuracy By Class ===

TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
0,778    0,600    0,700      0,778    0,737      0,189    0,656     0,757     yes
0,400    0,222    0,500      0,400    0,444      0,189    0,656     0,646     no
0,643    0,465    0,629      0,643    0,632      0,189    0,656     0,717

=== Confusion Matrix ===

 a b   <-- classified as
 7 2 | a = yes
 3 2 | b = no
```

2.1 Explication

Trois règles ont été extraites par le modèle JRip à partir des données d'entraînement. Chaque règle est une condition logique qui conduit à la prédiction d'une classe (play=yes ou play=no). Par exemple, la première règle indique que si l'humidité est élevée et la perspective est ensoleillée, alors la prédiction

est "play=no". La matrice de confusion montre comment les instances ont été classées par le modèle. Par exemple, le modèle a correctement classé 7 instances comme "play=yes" et 2 instances comme "play=no", mais a mal classé 3 instances "yes" comme "no" et 2 instances "no" comme "yes".

3 Travail à rendre (2): Règles de Classification

3.1 Question 1 et 2

1- Appliquant la méthode Ripper à la BD cars.arff

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

JRIP rules:
===========

(Surete = haut) and (Pachat = bas) and (Tcoffre = grand)
and (Npers = 4.0) => Verdict=tbon (16.0/4.0)
(Surete = haut) and (Tcoffre = grand) and (Pachat = bas)
and (Npers = plus) => Verdict=tbon (16.0/4.0)
(Surete = haut) and (Pachat = moyen) and (Pmaint = moyen)
and (Npers = plus) and (Tcoffre = grand) => Verdict=tbon (4.0/0.0)
(Surete = haut) and (Pachat = moyen) and (Pmaint = bas)
and (Tcoffre = grand) and (Npers = 4.0) => Verdict=tbon (4.0/0.0)
(Surete = haut) and (Tcoffre = moyen) and (Pachat = bas)
and (Npers = plus) => Verdict=tbon (16.0/7.0)
(Surete = haut) and (Pachat = moyen) and (Npers = plus)
and (Pmaint = bas) => Verdict=tbon (12.0/5.0)
(Surete = haut) and (Npers = 4.0) and (Pmaint = moyen)
and (Pachat = moyen) and (Tcoffre = grand) => Verdict=tbon (4.0/0.0)
(Surete = haut) and (Tcoffre = moyen) and (Pachat = moyen)
and (Pmaint = moyen) and (Npers = plus) => Verdict=tbon (4.0/1.0)
(Surete = haut) and (Pachat = bas) and (Tcoffre = moyen)
and (Npers = 4.0) and (Nportes = 4.0) => Verdict=tbon (4.0/1.0)
(Surete = haut) and (Npers = 4.0) and (Tcoffre = moyen)
and (Nportes = 5plus) and (Pachat = bas) => Verdict=tbon (4.0/1.0)

....
```

3.2 Question 3

3- L'algorithme Ripper (JRip dans Weka), est conçu pour traiter des données catégorielles. Par conséquent, il peut rencontrer des difficultés lorsqu'il est confronté à des jeux de données contenant des valeurs manquantes ou des attributs numériques. Dans ces cas, Ripper pourrait ne pas être en mesure de construire des règles significatives pour ces attributs, car il ne peut pas les diviser en catégories de manière directe. Pour travailler avec des jeux de données contenant des valeurs manquantes ou des attributs numériques Pour les valeurs manquantes : - On peut envisager de prétraiter vos données en imputant ou en supprimant les valeurs manquantes. Pour les attributs numériques : - On peut convertir vos attributs numériques en attributs catégoriels en définissant des intervalles ou des plages. Cela peut se faire à l'aide de filtres dans Weka.

3.3 Question 4

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

```
Decision Table:

Number of training instances: 1728
Number of Rules : 432
Non matches covered by Majority class.
Best first.
Start set: no attributes
Search direction: forward
Stale search after 5 node expansions
Total number of subsets evaluated: 22
Merit of best subset found:   94.676
Evaluation (for feature selection): CV (leave one out)
Feature set: 1,2,4,5,6,7

Time taken to build model: 0.02 seconds

=== Predictions on test data ===

    inst#     actual  predicted error prediction
        1     1:nacc     1:nacc       0.4
        2     1:nacc     1:nacc       0.4
        3     1:nacc     1:nacc       0.5
        4     1:nacc     1:nacc       0.4
        5     1:nacc     1:nacc       0.5
        6     1:nacc     1:nacc       0.5
        7     1:nacc     1:nacc       0.5
        8     1:nacc     1:nacc       0.5

....
```

Le modèle semble avoir une bonne performance, avec un taux de classification correct de 91.49La matrice de confusion montre que la classe "nacc" a été bien classée, tandis que la classe "acc" présente quelques erreurs.

3.4 Question 5

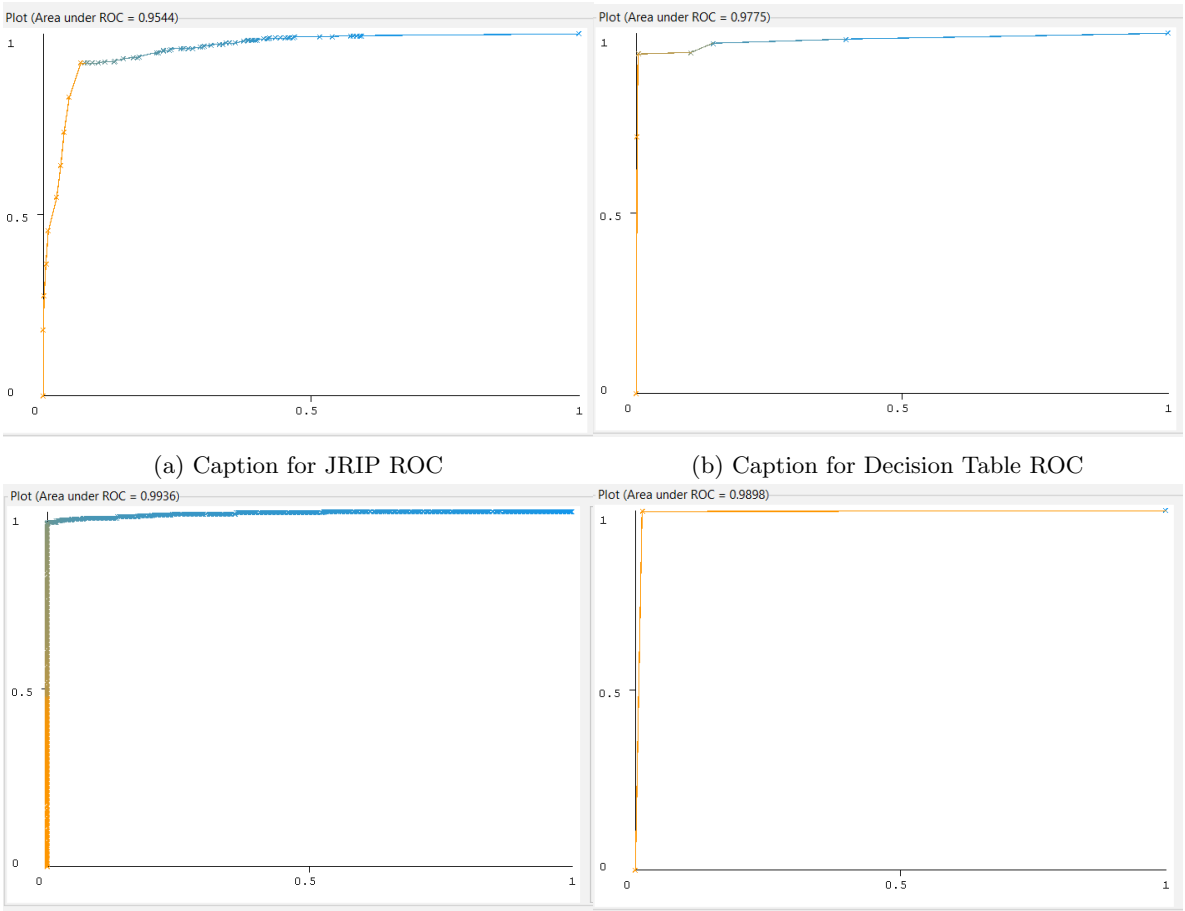
```
=== Run information ===

Scheme:       weka.classifiers.rules.DTNB -X 1
Relation:     Cars
Instances:    1728
Attributes:   7
              Pachat
              Pmaint
              Nportes
              Npers
              Tcoffre
              Surete
              Verdict
Test mode:    10-fold cross-validation

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

Decision Table:
```

```
Number of training instances: 1728
Number of Rules : 144
Non matches covered by Majority class.
Evaluation (for feature selection): CV (leave one out)
Feature set: 1,2,5,6,7

Time taken to build model: 0.13 seconds

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

Correctly Classified Instances        1646               95.2546 %
Incorrectly Classified Instances        82                4.7454 %
Kappa statistic                          0.8975
Mean absolute error                      0.1426
Root mean squared error                  0.2085
Relative absolute error                 62.2786 %
Root relative squared error             61.6522 %
Total Number of Instances             1728

=== Detailed Accuracy By Class ===

TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
0,970    0,033    0,986      0,970    0,978      0,928    0,994     0,998     nacc
0,932    0,030    0,899      0,932    0,916      0,891    0,989     0,954     acc
0,908    0,005    0,881      0,908    0,894      0,890    0,999     0,965     tbon
0,797    0,010    0,764      0,797    0,780      0,771    0,995     0,818     bon
0,953    0,030    0,954      0,953    0,953      0,912    0,993     0,980

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 1174   33    0    3 |    a = nacc
   15  358    2    9 |    b = acc
    0    1   59    5 |    c = tbon
    2    6    6   55 |    d = bon
```

3.5 Question 6

	Ripper (JRIP)	PRISM	Decision Table	DTNB
Correctly Classified Instances	95.4%	89.6%	91.49%	95.25%
Incorrectly Classified	4.59%	3.12%	8.50%	4.74%
TP Rate	95.4%	96.6%	91.5%	95.3%
Precision	95.5%	96.4%	91.3%	95.4%
F-Measure	95.4%	96.5%	91.3%	95.3%
ROC Area	94.2%	94.9%	97.3%	99.3%
Kappa Statistics	90.37%	91.5%	80.91%	89.75%

Table 1: Evaluation metrics on the four methods with BD cars



(a) Caption for JRIP ROC

(b) Caption for Decision Table ROC

(c) Caption for DTNP Table

(d) Caption for PRISM ROC

Figure 2: Overall Caption for the Figures

3.6 Question 7

Pour les courbes ROC, l'objectif est de maximiser l'aire sous la courbe de ROC plaus l'air est élévée plus notre modèle est performant. Plus précisement, Elle représente la relation entre le taux de vrais positifs (True Positive Rate ou Sensibilité) et le taux de faux positifs (False Positive Rate) pour différentes valeurs de seuil de classification.

Les figures montre que la méthode DTNP détient l'aire sous la courbe ROC la plus élévé pour cette base de donné cars. Aussi la courbe de cette méthode monte rapidement vers le coin supérieur gauche ce qui indique un modèle performant.

3.7 Question 9

Correctly Classified Instances : Decision Table a la deuxième meilleure performance (91.49 %), précédé par Ripper (95.4 %). Les deux sont des performances élevées.

TP Rate (True Positive Rate) : PRISM a le meilleur TP Rate (96.6 %), suivi de près par Ripper (95.4 %).

Kappa Statistics : PRISM a la statistique Kappa la plus élevée (91.5 %), indiquant une concordance robuste.

ROC Area : Decision Table a la ROC Area la plus élevée (97.3 %), ce qui indique une bonne capacité à discriminer entre les classes.

F-Measure : PRISM a la plus haute F-Measure (96.5 %), suivi de Ripper (95.4 %).

En résumé, pour une classification correcte globale, DTNP pourrait être préférable.

3.8 Question 10

	Ripper (JRIP)	PRISM	Decision Table	DTNB
Correctly Classified Instances	95.4%	N/A	94.9%	94.95%
Incorrectly Classified	4.59%	N/A	5.50%	5.04%
TP Rate	95.4%	N/A	94.5%	94.3%
Precision	95.5%	N/A	94.3%	95.4%
F-Measure	95.4%	N/A	94.3%	95.3%
ROC Area	94.2%	N/A	98.3%	98.3%
Kappa Statistics	90.37%	N/A	89.91%	89.75%

Table 2: Evaluation metrics on the four methods with BD vote

3.9 Question 11

Correctly Classified Instances : Decision Table a la meilleure performance dans la classification correcte des instances avec 94.9%, suivi de près par Ripper avec 95.4%. Ces deux méthodes ont des performances élevées.

Incorrectly Classified : Decision Table a le taux le plus bas d'instances mal classées avec 5.50%, ce qui indique une performance solide. Ripper suit avec 4.59%.

TP Rate (True Positive Rate) : Decision Table a un TP Rate de 94.5%, ce qui est légèrement inférieur à celui de Ripper (95.4%).

Precision : Ripper a la meilleure précision avec 95.5%, suivie de près par Decision Table avec 94.3%.

F-Measure : Ripper a la meilleure F-Measure avec 95.4%, suivie de Decision Table avec 94.3%.

ROC Area : Decision Table a la ROC Area la plus élevée avec 98.3%, indiquant une bonne capacité à discriminer entre les classes.

Kappa Statistics : Decision Table a une statistique Kappa élevée de 89.91%, ce qui suggère une concordance robuste

En se basant sur ces résultats, la méthode "Decision Table" semble être la meilleure option pour la base de données "vote".

3.10 Question 12

12- Non, on ne peut pas nécessairement dire que la même méthode l'emporte sur toutes les bases de données (d'aprés les resultats ci-dessus)

4 Règles d'association

4.1 Bonus: Règles de classification par la mèthode A PRIORI

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


Apriori
=======

Minimum support: 0.45 (196 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 11

Generated sets of large itemsets:

Size of set of large itemsets L(1): 6
```

```
Size of set of large itemsets L(2): 5

Size of set of large itemsets L(3): 1

Best rules found:

 1. adoption-of-the-budget-resolution=y physician-fee-freeze=n 219
 ==> Class=democrat 219    conf:(1)
 2. adoption-of-the-budget-resolution=y physician-fee-freeze=n
 aid-to-nicaraguan-contras=y 198 ==> Class=democrat 198    conf:(1)
 3. physician-fee-freeze=n aid-to-nicaraguan-contras=y 211
 ==> Class=democrat 210    conf:(1)
 4. physician-fee-freeze=n education-spending=n 202
 ==> Class=democrat 201    conf:(1)
 5. physician-fee-freeze=n 247
 ==> Class=democrat 245    conf:(0.99)
 6. el-salvador-aid=n aid-to-nicaraguan-contras=y 204
 ==> Class=democrat 197    conf:(0.97)
 7. el-salvador-aid=n 208 ==> Class=democrat 200    conf:(0.96)
 8. adoption-of-the-budget-resolution=y aid-to-nicaraguan-contras=y 215
 ==> Class=democrat 203    conf:(0.94)
 9. education-spending=n 233 ==> Class=democrat 213    conf:(0.91)
10. adoption-of-the-budget-resolution=y 253
==> Class=democrat 231    conf:(0.91)
```

5 Travail en bonus (1)

5.1 Question 1

On appliquant les méthodes JRIP, PRISM, Decision Table et DTNB, on trouve les résultas suivants:

Méthode	REPTree	DTNB	Prism	JRip
Précision	99.96%	99.94%	100%	100%
Erreur Globale	0.04%	0.06%	0%	0%
Kappa Statistic	0.9993	0.9988	1	1
Mean Absolute Error	0.0003	0.0052	0	0
Root Mean Squared Error	0.0166	0.039	0	0
Instances Correctes	8121	8124	8124	8124
Instances Incorrectes	3	5	0	0

Table 3: Comparaison des performances des méthodes de classification sur la base de données des champignons (tableau transposé).

En fonction de ces métriques, Prism et JRip semblent avoir les performances les plus élevées sur cet ensemble de données particulier. Par contre, on suspecte que les deux modéles sont en Overfitting. Il faut tester avec un jeu de donnée test pour s'assurer. Voici quelques règles communes qui semblent apparaître dans plusieurs modèles:

- Règle commune sur l'odeur (présente dans plusieurs modèles) :
Règle : Si odor = f, alors le champignon est vénéneux (classifié comme "p"). Présence : Apparaît dans les modèles JRip et DTNB.

- Règle commune sur la taille des lamelles (présente dans plusieurs modèles) :
Règle : Si gill-size = n et gill-color = b, alors le champignon est vénéneux. Présence : Apparaît dans les modèles JRip et DTNB.

- Règle commune sur l'habitat et la couleur du chapeau (présente dans plusieurs modèles) :

Règle : Si habitat = l et cap-color = w, alors le champignon est vénéneux. Présence : Apparaît dans les modèles Prism et JRip.

- Règle commune sur la couleur des spores (présente dans plusieurs modèles) :

Règle : Si spore-print-color = r, alors le champignon est vénéneux. Présence : Apparaît dans les modèles DTNB et Prism.

- Règle commune sur la couleur du chapeau (présente dans plusieurs modèles) :

Règle : Si cap-color = c et odor = n, alors le champignon est comestible. Présence : Apparaît dans les modèles DTNB et Prism.

On appliquant la méthode APRIORI, on retrouve que

```
       =======

Minimum support: 0.35 (2843 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 13

Generated sets of large itemsets:

Size of set of large itemsets L(1): 17

Size of set of large itemsets L(2): 52

Size of set of large itemsets L(3): 69

Size of set of large itemsets L(4): 43

Size of set of large itemsets L(5): 12

Size of set of large itemsets L(6): 1

Best rules found:

 1. odor=n ring-number=o 2928 ==> class=e 2880    conf:(0.98)
 2. odor=n veil-type=p ring-number=o 2928 ==> class=e 2880    conf:(0.98)
 3. odor=n gill-size=b 3288 ==> class=e 3216    conf:(0.98)
 4. odor=n gill-size=b veil-type=p 3288 ==> class=e 3216    conf:(0.98)
 5. odor=n gill-attachment=f gill-size=b 3096 ==> class=e 3024    conf:(0.98)
 6. odor=n gill-size=b veil-color=w 3096 ==> class=e 3024    conf:(0.98)
 7. odor=n gill-attachment=f gill-size=b veil-type=p 3096 ==> class=e 3024
 conf:(0.98)
 8. odor=n gill-attachment=f gill-size=b veil-color=w 3096 ==> class=e 3024
 conf:(0.98)
 9. odor=n gill-size=b veil-type=p veil-color=w 3096 ==> class=e 3024
 conf:(0.98)
10. odor=n gill-attachment=f gill-size=b veil-type=p veil-color=w 3096 ==>
class=e 3024    conf:(0.98)
```

En général, ces règles indiquent une confiance sur les regles de 98%.

5.2 Question 2

On appliquant la méthode APRIORI à la bd ZOO, voici le résultat:

```
Apriori
=======

Minimum support: 0.7 (71 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 6

Generated sets of large itemsets:

Size of set of large itemsets L(1): 8

Size of set of large itemsets L(2): 13

Size of set of large itemsets L(3): 2

Best rules found:

 1. venomous=false tail=true 71 ==> backbone=true 71    <conf:(1)>
 lift:(1.22) lev:(0.13) [12] conv:(12.65)
 2. tail=true 75 ==> backbone=true 74    <conf:(0.99)> lift:(1.2)
 lev:(0.12) [12] conv:(6.68)
 3. backbone=true tail=true 74 ==> venomous=false 71    <conf:(0.96)>
 lift:(1.04) lev:(0.03) [2] conv:(1.47)
 4. backbone=true 83 ==> venomous=false 79    <conf:(0.95)> lift:(1.03)
 lev:(0.03) [2] conv:(1.31)
 5. breathes=true 80 ==> fins=false 76    <conf:(0.95)> lift:(1.14)
 lev:(0.09) [9] conv:(2.69)
 6. airborne=false 77 ==> feathers=false 73    <conf:(0.95)> lift:(1.18)
 lev:(0.11) [11] conv:(3.05)
 7. tail=true 75 ==> venomous=false 71    <conf:(0.95)> lift:(1.03)
 lev:(0.02) [1] conv:(1.19)
 8. tail=true 75 ==> backbone=true venomous=false 71    <conf:(0.95)>
 lift:(1.21) lev:(0.12) [12] conv:(3.27)
 9. breathes=true venomous=false 75 ==> fins=false 71    <conf:(0.95)>
 lift:(1.14) lev:(0.09) [8] conv:(2.52)
10. breathes=true 80 ==> venomous=false 75    <conf:(0.94)> lift:(1.02)
lev:(0.01) [1] conv:(1.06)
```

L'algorithme a trouvé plusieurs règles d'association fréquentes dans la base de données Zoo en utilisant un support minimum de 0.7 et une confiance minimale de 0.9. Les règles mettent en évidence des relations intéressantes entre les attributs. Par exemple, la présence d'une queue (tail=true) semble souvent associée à la présence d'une colonne vertébrale (backbone=true), et l'absence de venin (venomous=false) est également souvent associée à ces deux caractéristiques. Les niveaux de confiance élevés (près de 1 dans plusieurs règles) indiquent une forte corrélation entre les attributs dans ces règles.

5.3 Question 3

6 Travail à rendre (3) : Bayésienne sur la BD Spam

6.1 Question 1

Le modèle semble avoir une performance globalement satisfaisante, avec un pourcentage d'instances correctement classées de 88.55%. Cela signifie que le modèle a correctement classé la catégorie correcte

(spam ou non-spam) pour environ 88.55% de l'ensemble de données lors de la validation croisée à 10 plis.

Cependant, cette performance est basée sur certaines hypothèses du modèle Naive Bayes, telles que l'indépendance conditionnelle des attributs étant donné la classe. Ces hypothèses peuvent ne pas toujours être entièrement valides dans le contexte réel des données des mails.

6.2 Question 2

Ce bon comportement du modèle, avec un taux élevé d'instances correctement classées (88.55%), peut être attribué à plusieurs facteurs :

- Caractéristiques Discernantes : Les caractéristiques utilisées pour entraîner le modèle semblent être informatives pour distinguer entre les e-mails spam et non-spam. Les fréquences de certains mots ou caractères peuvent être fortement corrélées avec la nature du contenu.
- Séparation Claire des Classes : Il semble y avoir une séparation claire entre les deux classes dans l'espace des caractéristiques. Cela signifie que les e-mails spam et non-spam présentent des différences significatives en termes de fréquences de mots et de caractères, facilitant la tâche du modèle de les distinguer.
- Adéquation du Modèle : Le modèle Naive Bayes peut bien fonctionner dans des situations où l'indépendance conditionnelle des attributs est une approximation raisonnable. Même si cette hypothèse est simplificatrice, elle peut être suffisamment valide pour ce type de données.
- Volume de Données : Avec 4601 instances, le modèle dispose d'un ensemble de données de taille décente pour l'apprentissage. Un ensemble de données suffisamment grand peut permettre au modèle d'apprendre des modèles discriminants.

6.3 Question 3

- Indépendance des attributs : Le modèle suppose que les différentes caractéristiques (fréquences de mots et caractères) sont indépendantes les unes des autres conditionnellement à la classe. Cela peut ne pas être entièrement réaliste, car certaines dépendances entre les mots peuvent exister dans les e-mails.
- Sac de mots : Le modèle considère les e-mails comme des sacs de mots, ignorant la séquence (ordre) et la structure des phrases. Cela peut ne pas capturer les relations sémantiques complexes entre les mots.
- Sac de mots : Le modèle considère les e-mails comme des sacs de mots, ignorant la séquence (ordre) et la structure des phrases. Cela peut ne pas capturer les relations sémantiques complexes entre les mots.

6.4 Question 4

```
        Naive Bayes Classifier

                                  Class
Attribute                             0      1
                                 (0.61) (0.39)
=============================================
...
```

Dans ce tableau, les valeurs entre parenthèses représentent les probabilités a priori de chaque classe. Dans cet exemple, la classe 0 a une probabilité a priori de 0.61, et la classe 1 a une probabilité a priori de 0.39.

Weka calcule ces probabilités a priori de la manière suivante:

Pour un ensemble de données donné, la probabilité a priori d'une classe est simplement la fréquence relative de cette classe parmi toutes les instances de l'ensemble de données. Cela signifie que si vous avez N instances dans votre ensemble de données et que la classe 0 apparaît n0 fois et la classe 1 apparaît n1 fois, alors la probabilité a priori de la classe 0 serait n0 / N et la probabilité a priori de la classe 1 serait n1 / N.

6.5 Question 5

```
      word_freq_3d_binarized
0                             2781.0 1775.0
1                                9.0   40.0
[total]                       2790.0 1815.0
```

Pour calculer $P(3d|spam)$:

$$P(3d|spam) = \frac{\text{Nombre d'instances spam avec 3d}}{\text{Nombre total d'instances spam}}$$

Dans le tableau, cela serait :

$$P(3d|spam) = \frac{40}{1815}$$

De même, pour $P(3d|non\text{-}spam)$:

$$P(3d|non\text{-}spam) = \frac{\text{Nombre d'instances non-spam avec 3d}}{\text{Nombre total d'instances non-spam}}$$

Dans l'exemple que vous avez donné, cela serait :

$$P(3d|non\text{-}spam) = \frac{9}{2790}$$

6.6 Question 6

6.6.1 Question 6-1

La sortie de Weka indique que le modèle Naive Bayes a correctement classé 407 instances sur un total de 461 instances dans l'ensemble de test, ce qui représente un taux de classification de 88,29%. Le modèle semble être plus précis pour la classe 0 (non-spam) que pour la classe 1 (spam), avec un taux de rappel plus élevé pour la classe 0. Cela peut être dû à plusieurs raisons, notamment un déséquilibre de classe ou des caractéristiques qui ne sont pas aussi distinctives pour la classe 1.

7 Travail en Bonus (3)

7.1 Random Forests

7.1.1 Question I

Random Forest est un algorithme d'apprentissage ensembliste utilisé pour la classification et la régression. Il construit plusieurs arbres de décision lors de l'entraînement, en utilisant des échantillons aléatoires du jeu de données et des sous-ensembles aléatoires de caractéristiques. Chaque arbre "vote" pour la prédiction finale, et le résultat est déterminé par le vote majoritaire. Cela améliore la robustesse et la précision du modèle, tout en minimisant le surajustement.

7.1.2 Question II

Validation croisée stratifiée : Résultats globaux :

- *Instances correctement classées : 7 (50%)*
- *Instances incorrectement classées : 7 (50%)*
- Statistique Kappa : -0,1395
- Erreur absolue moyenne : 0,5413
- Erreur quadratique moyenne : 0,6239
- Erreur absolue relative : 113,6824%
- Erreur quadratique relative : 126,4669%
- Nombre total d'instances : 14

Précision détaillée par classe :

- Classe "yes" : Taux de vrais positifs = 66,7%, Taux de faux positifs = 80%, Précision = 60%, Rappel = 66,7%, F-Mesure = 63,2%
- Classe "no" : Taux de vrais positifs = 20%, Taux de faux positifs = 33,3%, Précision = 25%, Rappel = 20%, F-Mesure = 22,2%

Matrice de confusion;

```
a b   <-- classifié comme
6 3 | a = yes
4 1 | b = no
```

7.2 Régression Logistique Ridge

7.2.1 Question I

La régression logistique Ridge est une variante de la régression logistique qui inclut une pénalité de régularisation de type "ridge" pour prévenir le surajustement. Elle ajoute une contrainte L2 à la fonction de coût, en ajoutant la somme des carrés des coefficients à minimiser. Cela favorise des coefficients plus petits et plus équilibrés, ce qui est utile lorsque les caractéristiques sont fortement corrélées. La force de régularisation est contrôlée par un paramètre appelé alpha. En ajustant cet alpha, on peut contrôler le compromis entre l'ajustement précis des données d'entraînement et la prévention du surajustement.

7.2.2 Question II

Dans l'ensemble, la régression logistique avec ridge semble avoir de meilleures performances sur la base meteo nominal avec un 10-XF-cross-validation. (voir table ci-dessous)

8 Travail en Bonus (4)

8.1 Régression Logistique

8.1.1 Question I

Correctly Classified Instances : 3809 92.0048 %

Métrique	Régression Logistique (Ridge)	RandomTree (Arbre de Décision)
Correctly Classified Instances	71.4286%	50%
Incorrectly Classified Instances	28.5714%	50%
Kappa statistic	0.3778	-0.1395
Mean Absolute Error	0.2858	0.5413
Root Mean Squared Error	0.5345	0.6239
Relative Absolute Error	60.009%	113.6824%
Root Relative Squared Error	108.3425%	126.4669%
TP Rate (yes)	77.8%	66.7%
FP Rate (yes)	40%	80%
Precision (yes)	77.8%	60%
Recall (yes)	77.8%	66.7%
F-Measure (yes)	77.8%	63.2%
MCC	0.378	-0.141
ROC Area	76.7%	33.3%
PRC Area	81.7%	55.8%

Table 4: Table de comparaison entre Régression Logistique (Ridge) et RandomTree

8.1.2 Question II

Effet des Coefficients : Les coefficients indiquent comment chaque variable contribue à la probabilité d'appartenir à la classe respective. Un coefficient négatif signifie une contribution négative à la probabilité, tandis qu'un coefficient positif signifie une contribution positive.

Interprétation des Coefficients : Par exemple, un coefficient élevé pour une variable comme wordfreqmeetingbinarized dans la classe 0 (positif) suggère que la présence fréquente du mot "meeting" augmente la probabilité d'appartenir à la classe 0, tandis que le même coefficient dans la classe 1 (négatif) suggère que la présence fréquente du mot "meeting" diminue la probabilité d'appartenir à la classe 1.

Comparaison des Classes : En comparant les coefficients entre les deux classes, on peut observer quelles variables ont des effets opposés sur la probabilité d'appartenir à l'une ou l'autre classe.

Impact sur la Prédiction : Les variables avec des coefficients plus élevés (positifs ou négatifs) ont un impact plus fort sur la prédiction de la classe respective.

8.1.3 Question III

- Pour *word freq hp binarized* dans la classe 1 : 1.39
- Pour *char freq $ binarized* dans la classe 1 : -0.74

8.1.4 Question IV

Oui, les coefficients de régression ont un sens significatif, même si la classe 1 correspond au spam. Dans le contexte de la régression logistique, la fonction sigmoïde est utilisée pour transformer la somme pondérée des variables (produit des coefficients et des valeurs des variables) en une probabilité entre 0 et 1 Pour une $P(y = 1) = \frac{1}{1+\exp x}$ avec $(y = 1)$ signigie que y appartient à la classe 1 (spam) et $x = \sum \beta * x$

Dans ce contexte, un coefficient positif pour une variable d'entrée signifie que la présence de cette variable augmente la probabilité de la classe 1 (spam), tandis qu'un coefficient négatif signifie que la présence de cette variable diminue la probabilité de la classe 1.

8.1.5 Question V

Si les attributs prédictifs sont à valeurs réelles plutôt que binaires, la régression logistique continue d'être applicable, mais il y a quelques implications à considérer :

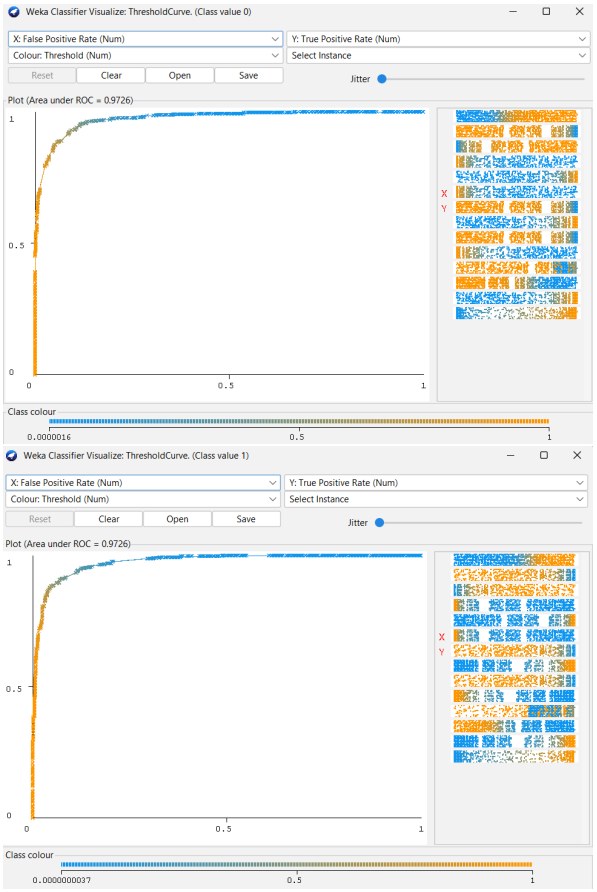


Figure 3: En haut ROC de la classe 1 et en bas ROC de la classe 0

Interprétation des Coefficients : Les coefficients de régression représentent toujours la contribution à la probabilité de la classe positive par unité de changement dans la variable indépendante. Une augmentation d'une unité dans une variable continue affectera la probabilité de la classe positive selon le coefficient associé.

Échelle des Coefficients : Les coefficients peuvent être comparés en termes de magnitude. Un coefficient plus grand indique une contribution plus importante à la probabilité.

Overfitting : L'utilisation de variables continues peut rendre le modèle plus complexe, ce qui peut augmenter le risque de surajustement (overfitting) si le nombre d'observations est limité par rapport au nombre de variables.

8.2 SVM

8.2.1 Question I

Correctly Classified Instances pour SVM: 241 (94.5098 %)

Correctly Classified Instances pour régression de ridge: 3809 (92.0048 %)

8.2.2 Question II

Les coefficients pour les attributs [word freq hp binarized] et [charfreq$ binarized] dans le modèle SVM (SMO) sont les suivants :

Coefficient pour [word freq hp binarized]: -1.3652 Coefficient pour [char freq $ binarized]: 1.1007

8.2.3 Question III

Voici la différence entre SVM et regression logistique de ridge:

- **Frontière de Décision (Hyperplan) :**

SVM Linéaire : L'objectif principal de la SVM linéaire est de trouver un hyperplan qui maximise la marge entre les différentes classes. L'hyperplan est défini comme la frontière de décision optimale qui maximise la distance entre les points les plus proches de chaque classe (vecteurs de support).

Régression Logistique : La régression logistique utilise une fonction logistique pour modéliser la probabilité qu'une instance appartienne à une classe particulière. La frontière de décision est définie par le seuil de probabilité, généralement 0.5. Si la probabilité prédite est supérieure à ce seuil, l'instance est classée dans une classe, sinon dans l'autre.

- **Fonction de Coût :**

SVM Linéaire : La SVM linéaire utilise une fonction de coût basée sur la marge. L'objectif est de minimiser la norme du vecteur de poids (coefficients de l'hyperplan) tout en maximisant la marge entre les classes. Les erreurs de classification n'ont pas d'impact tant qu'elles ne franchissent pas la marge.

Régression Logistique : La régression logistique utilise la fonction de coût de la log-vraisemblance négative. L'objectif est de maximiser la probabilité des étiquettes observées par le modèle. Les erreurs de classification sont pénalisées, mais elles n'ont pas le même impact sur le coût que dans la SVM linéaire.

- **Interprétation des Coefficients :**

SVM Linéaire : Les coefficients de l'hyperplan de la SVM linéaire ne sont pas directement interprétables en termes de probabilités. L'accent est mis sur la marge entre les classes.

Régression Logistique : Les coefficients de la régression logistique sont directement liés aux log-odds des probabilités. Chaque coefficient est associé à une variable d'entrée et mesure son impact sur la probabilité de la classe positive.

8.3 Aller plus loin (IBK, KNN, feature selection)

8.3.1 Question I

Correctly Classified Instances 2934 99.9659 % Incorrectly Classified Instances 1 0.0341 % Kappa statistic 0.9994 Précision 100%

Une précision de 100 % suggère que le modèle IBk a parfaitement classé toutes les instances dans les classes respectives.

8.3.2 Question II

Tester sur les données d'apprentissage avec un classifieur 1-NN (1-plus-proche-voisin) peut conduire à une performance biaisée et peu représentative de la capacité du modèle à généraliser à de nouvelles données. En effet, Le classifieur 1-NN est très flexible et peut mémoriser les données d'apprentissage, y compris le bruit ou les particularités spécifiques aux données d'apprentissage. Cela peut conduire à un surajustement, où le modèle s'adapte trop précisément aux données d'apprentissage mais ne généralise pas bien aux nouvelles données.

8.3.3 Question III

Dans le cas d'un classifieur 1-NN, il est souvent attendu que les performances sur l'ensemble de test soient légèrement inférieures ou égales à celles sur l'ensemble d'apprentissage

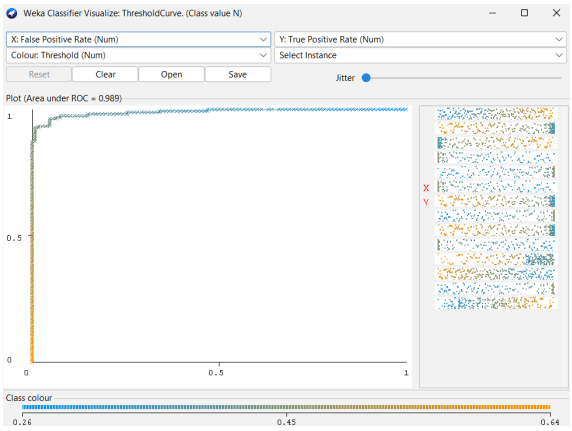
8.3.4 Question IV

La précision du classifieur si tous les points étaient étiquetés "N" peut être calculée à partir de la matrice de confusion. La précision est définie comme le nombre de vrais positifs (TP) divisé par la somme des vrais positifs et des faux positifs (FP).

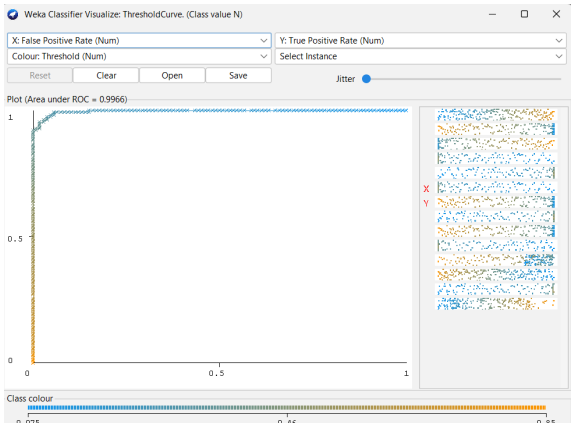
```
   a   b   c   <-- classified as
105  24  20 |   a = N
1  51   0 |   b = EI
6   4  44 |   c = IE
```

Si tous les points étaient étiquetés "N", alors tous les vrais positifs seraient dans la classe "N" (105 dans ce cas). La somme des vrais positifs et des faux positifs pour la classe "N" serait donc 105 + 24 + 20 = 149.

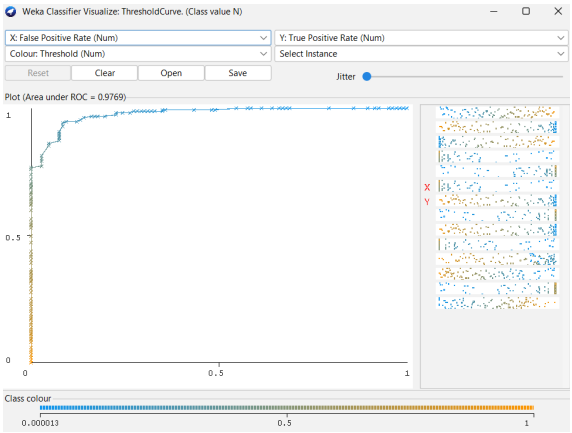
La précision serait alors calculée comme suit : $\frac{105}{149} = 0.704$



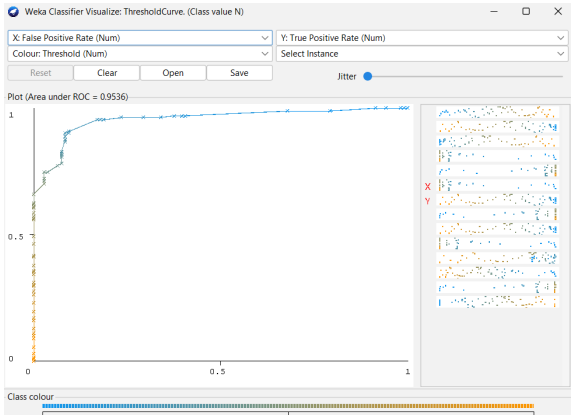
(a) ROC curve for N class for $K = 1000$



(b) ROC curve for N class for $K = 100$



(c) ROC curve for N class for $K = 10$



(d) ROC curve for N class for $K = 5$

Figure 4: Overall ROC curve for N class for different K

Plus le K est élévée plus la précision deient élévé ainsi l'air sous la courbe de ROC.