Université des Sciences et de la Technologie Houari Boumediene

Data Mining

Rapport Projet Partie 02
Technique de Data Mining

HEBBACHE Imad eddine,

KHOUCHA Madani,

MEZIANI Mohamed Imad,

Mme. Drias

Mr. Khennak

1er janvier 2022

Introduction

Le Data Mining est un domaine très large qui consiste à appliquer un ensemble de processus ou de techniques sur les données brutes dont le but d'explorer et analyser ces données pour les rendre utiles.

Dans cette deuxième partie de projet on va implémenter ces techniques qui permettent en premier lieu de normaliser et de discrétiser les donner, ensuite d'extraire des motifs fréquents de ces données et à la fin d'appliquer une classification à partir des données initiales.

Toute ces techniques seront appliquer à partir d'une IHM conçu avec l'utilisation du langage java.

1 Objectifs

Tout cela nous a mené à fixer les objectifs suivants :

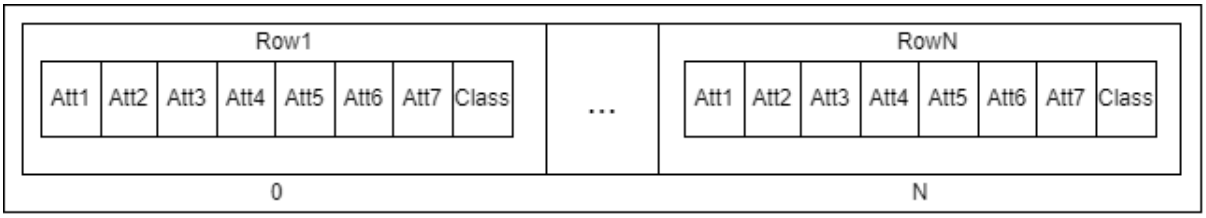
- Implémenter les deux algorithmes de normalisation :
  - La normalisation Min-Max.
  - La normalisation Z-score
- Développer et appliquer les deux type de discrétisation suivantes :
  - La discrétisation en classes d'effectifs égaux.
  - La discrétisation en classes d'amplitudes égales.
- Appliquer les deux algorithmes (Apriori et Eclat) d'extraction de motif fréquents sur les données discrétisées.
- Classification des instances du dataset par l'implémentation de l'algorithme de la classification Naïve Bayésienne et KNN.

2 Pré-traitement des données

Le pré-traitement des données est une étape importante du processus d'exploration et d'analyse des données qui prend les données dans leur format initial et les transforme en un format pouvant être compris et analysé par les ordinateurs.
On dit souvent que les données pré-traitées sont plus importants que les algorithmes les plus puissants.

2.1 La normalisation

1. Description :
   La normalisation permet de mettre les données à l'échelle dans un intervalle régularisé spécifique à fin de donner une précision pour la comparaison des donnée.
2. Structures utilisées :
   Le résultat est retourner sous forme de tableau de lignes (Class Row).



Normalisation avec Min-Max

C'est une technique de normalisation de données, une transformation linéaire est effectuée sur les données. La valeur minimale et maximale sont extraites des données et chaque valeur est remplacée selon la formule suivante :

$$\text{VALEUR}_{(i,nouvelle)} = \frac{\text{VALEUR}_{(i,courante)} - \text{VALEUR}_{(min,courante)}}{\text{VALEUR}_{(max,courante)} - \text{VALEUR}_{(min,courante)}} \left(\text{VALEUR}_{(max,nouvelle)} - \text{VALEUR}_{(min,nouvelle)}\right) + \text{VALEUR}_{(min,nouvelle)}$$

La normalisation Min-Max préserve les relations entre les valeurs de données d'origine.
Algorithm

---

input : Un tableau d'attributs $Attr$,Valeur Minimale Min,Valeur Maximale Max
output : Un tableau de lignes (Row) $Normalisedb$

while *Le tableau d'attributs non vide* **do**
  for *Chaque ligne* **do**
    $Valeur(i, nouvelle) \leftarrow \frac{Valeur(i,courante) - Valeur(min,courante)}{Valeur(max,courante) - Valeur(min,courante)} \times (Valeur(max, nouvelle) - Valeur(min, nouvelle)) + Valeur(min, nouvelle)$
  end
end
while *Attribut non vide* **do**
  Créer une nouvelle ligne avec les valeurs de chaque attribut avec le même indice Ajouter la ligne dans le tableau de lignes $Normalised$
end

Normalisation avec Z-score

Un score Z est une mesure numérique qui décrit la relation d'une valeur avec la moyenne d'un ensemble de valeurs.
La normalisation avec Z-score exprime l'écart par rapport à la valeur de la moyenne selon la formule suivante :

$$\text{VALEUR}_{(i,nouvelle)} = \frac{\text{VALEUR}_{(i,courante)} - \text{VALEUR}_{(moyenne,courante)}}{S}$$

$$S = \frac{1}{N}\sum_{i=1}^{N} \left|\text{VALEUR}_{(i,courante)} - \text{VALEUR}_{(moyenne,courante)}\right|$$

S représente formule de calcule de l'écart-type.

output : Un tableau de lignes (Row) *Normalised*

while *Le tableau d'attributs non vide* **do**

  moyenne← `Moyenne(`*Attribut*`)` $S = \frac{1}{N}\sum_{i=0}^{N}$Valeur (i,courante)−moyenne ;

  for *Chaque ligne* **do**

    $Valeur(i, nouvelle) \leftarrow \frac{Valeur(i,courante)-moyenne}{S}$

  end

end

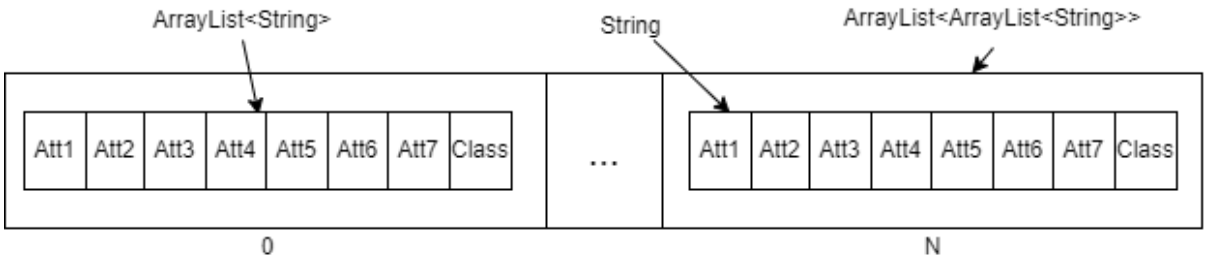
while *Attribut non vide* **do**

  Créer une nouvelle ligne avec les valeurs de chaque attribut avec le même indice Ajouter la ligne dans le tableau de lignes *Normalised*

end

2.2 La discrétisation

1. Description :
   La discréditation regroupe les données en intervalles plus petits, c'est un peut similaire au groupement des données par classe mais cela se fait après le nettoyage des données.
2. Structures utilisées :
   La structure utilisée est un tableau de tableaux de chaîne de caractères.



Discrétisation en classes d'effectifs égaux.

Cette méthode permet de crée des intervalles avec le même nombre d'attributs.

input : Un tableau d'attributs $Attr, NombredequantillesQ$

output : Un tableau de tableau de chaine de caractere *Dicretised*

while *Le tableau d'attributs non vide* **do**

  $NombreIntervals \leftarrow \frac{N}{Q}$ Creer des intervales

  for *Chaque ligne* **do**

    Attribuer a chaque valeur la classe de l'intervalle auquel elle appartient

  end

end

while *Attribut non vide* **do**

  Cree un tableau de chaîne de caractère avec la classe de chaque attribut avec le même indice Ajouter le tableau dans le tableau *Dicretised*

end

Discrétisation en classes d'amplitudes égales

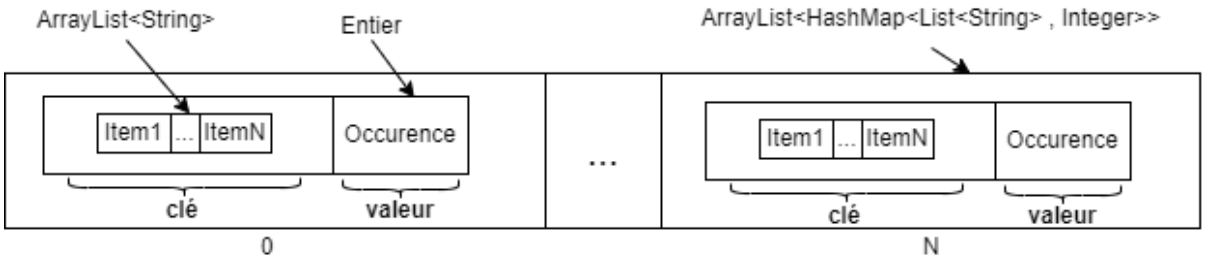
Cette méthode permet de crée des intervalles avec la même entendu.

input : Un tableau d'attributs $Attr, Nombred'intervallesQ$
output : Un tableau de tableau de chaine de caractere *Dicretised*
while *Le tableau d'attributs non vide* **do**
  $LongeurInterval \leftarrow \frac{Valeur(max)-Valeur(min)}{Q}$ Créer des intervales
  for *Chaque ligne* **do**
    Attribuer a chaque valeur la classe de l'intervalle auquel elle appartient
  end
end
while *Attribut non vide* **do**
  Cree un tableau de chaîne de caractère avec la classe de chaque attribut avec le même indice Ajouter le tableau dans le tableau *Dicretised*
end

3 Extraction des motifs fréquents

3.1 Les itemsets fréquents

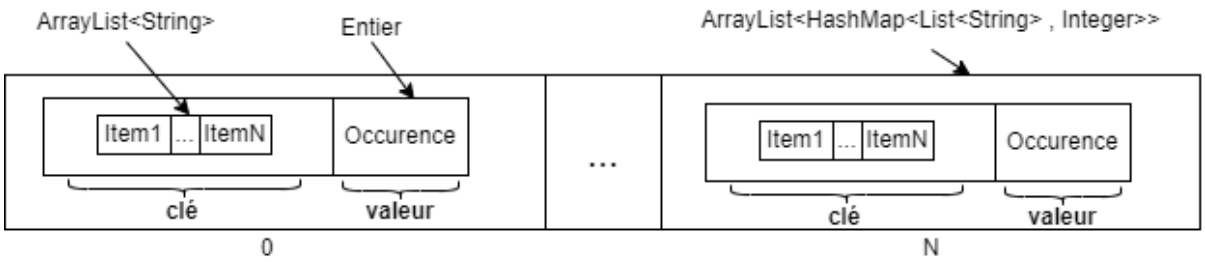
1. Description :
   Les itemsets fréquents représentent les items du dataset avec une fréquence supérieur au support minimal qui sera définit comme paramètre dans l'algorithme.
2. Structures utilisées :
   La structure utilisée est un tableau (ArrayList) de HashMap qui est une structure de données utilisée pour stocker les objets paire clé-valeur qui a comme clé un tableau de chaînes de caractères qui contient les items et comme valeur un entier qui représente le support de la clé.



3.2 Apriori

(a) Description :
L'algorithme Apriori est le premier algorithme propose pour l'extraction des motifs fréquents, une approche itérative pour découvrir les itemsets les plus fréquents. Pour chaque itération il réduit l'espace de recherche en détectant les motifs les plus fréquents par rapport une valeur dite Support minimum.

(b) Structures utilisées :
La structure utilisée est un tableau (ArrayList) de HashMap qui est une structure de données utilisée pour stocker les objets paire clé-valeur qui a comme clé un tableau de chaînes de caractères qui contient les items et comme valeur un entier qui représente le support de la clé.



(c) Pseudo-Code :

input : *Un tableau d'attributs discretisee, Support Minimal*
output : *La liste des itemsets frequents*
$L1 \leftarrow$ (*Itemsets frequents*) **while** L_{new} *nonvide* **do**
  $C_{new} \leftarrow$ *generer toute les combinaison possible*
  for *Chaque ligne de* C_{new} **do**
    Calculer le nombre de transactions dans les quelles figure l'intersection des items de la ligne
    if *la fréquence de ligne* $>$ *support minimal* **then**
      $L_{new} \leftarrow L_{new}$ + *les itemsets de cette ligne*
    end
  end
  $L \leftarrow L + L_{new}$
end

3.3 Eclat

(a) Description :

L'algorithme ECLAT signifie Equivalence Class Clustering and bottom-up Lattice Traversal, il est considéré comme une version plus améliorée de Apriori. Il utilise l'approche de recherche en profondeur d'abord alors que Apriori utilise l'approche en largeur d'abord ce qui rend Eclat plus rapide parce qu'il utilise moins de mémoire.

(b) Pseudo-code :

input : Support Minimale *supMin*,Nombre de transactions
output : Items-fréquents
2. *table verticale* $\leftarrow$ *inverse la BD*
3. Éliminé les linges avec support inférieur a *supMin*
4. .3 **for** *Toutes les combinaisons possible des items* **do**
5. .a créer une nouvelle ligne contenant
   i.l'union du contenu des premières colonnes de *table verticale*
   ii.l'intersection des contenus des deuxièmes colonnes de *table verticale*
end
6. .4 répétez les étapes 2 et 3 jusqu'à ce qu'aucun nouvel ensemble d'éléments ne puisse être créé.

4 Classification supervisée des instances du dataset

4.1 Classification Naïve Bayésienne.

La classification Naïve Bayésienne est une classification qui se base sur les probabilités, elle s'applique sur des donnes discrétisée,Cette classification est peu fiable car elle traite chaque caractéristique indépendamment mais requiert peu de donnes d'entraînement.

input : $Untableaud'attributsdiscretisee, Donnesaclassifier$
output : $Laclasse$
for *Chaque Attribut k* **do**
  for *Chaque Classe i* **do**
    $P(X_k|Classe_i) \leftarrow lenombred'apparitiondeX_kaveclaclassei$ **end**
  end
  for *Chaque Classe i* **do**
    $P(X|Classe_i) \leftarrow \prod P(X_j|Classe_i)$ **end**
    Choisir La classe avec la plus haute probabilite

4.2 Classification K-nearest neighbors(KNN).

La classification Knn est une classification qui se base sur les k plus proches voisins, elle peut être appliquée sur des donnes quantitatives en calculant la distance avec les mesures euclidienne,Manhattan ou plusieurs autres mesures , comme il peut s'appliquer sur des donnes discrétisée avec comme exemple la mesure de hamming.

input : Un tableau d'attributs discrétisée,Donnes a classifier
output : La classe
for *Chaque ligne du dataset* **do**
  Calculer la distance entre nos donnes et la ligne
end
Trier les résultats selon la distance par ordre DESC ;
Récupérer les K premiers instances;
Choisir La classe avec la plus fréquente entre les K instances ;

5 Interfaces

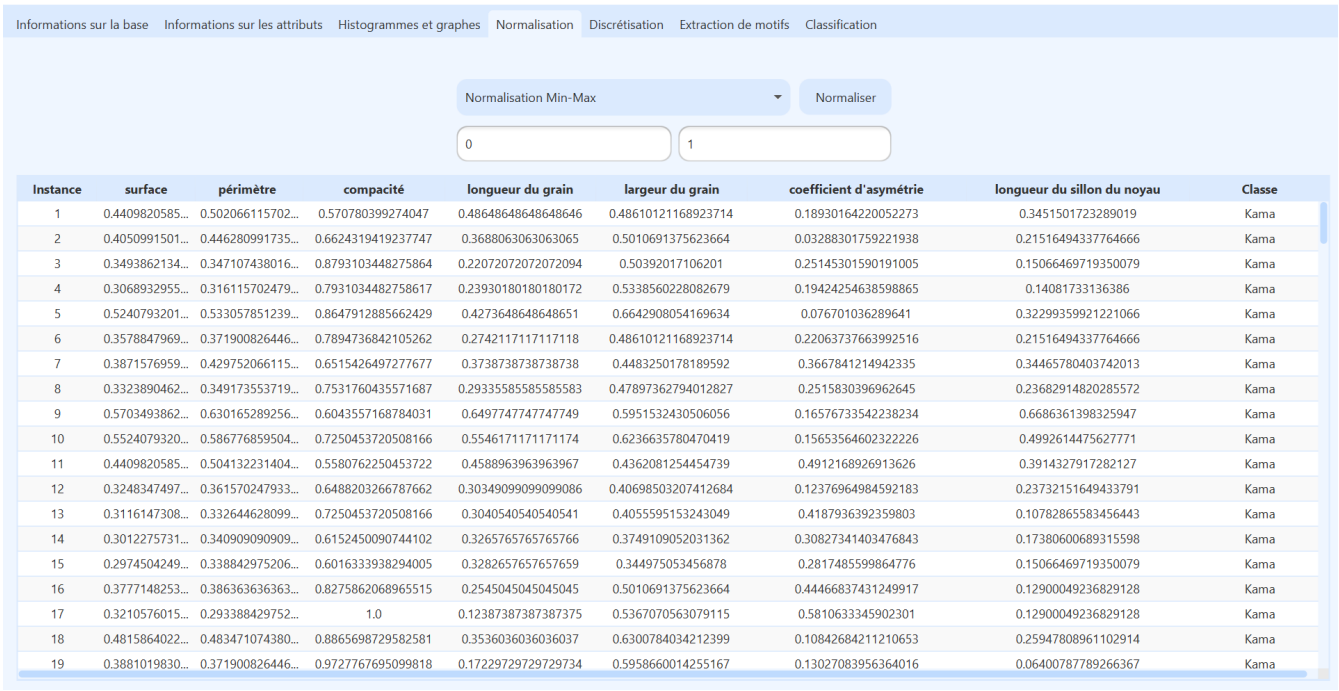
Pour visualisé nos résultats nous avons conçu un IHM simple a utilisé qui facilite cette opération on utilisant la bibliothèque **javaFX** de JAVA, ci-dessous les différentes fonctionnalités de notre interface.

5.1 Interfaces de normalisation

l'onglet ci-dessous permet la normalisation avec les méthodes Min-Min ou Z-Score.

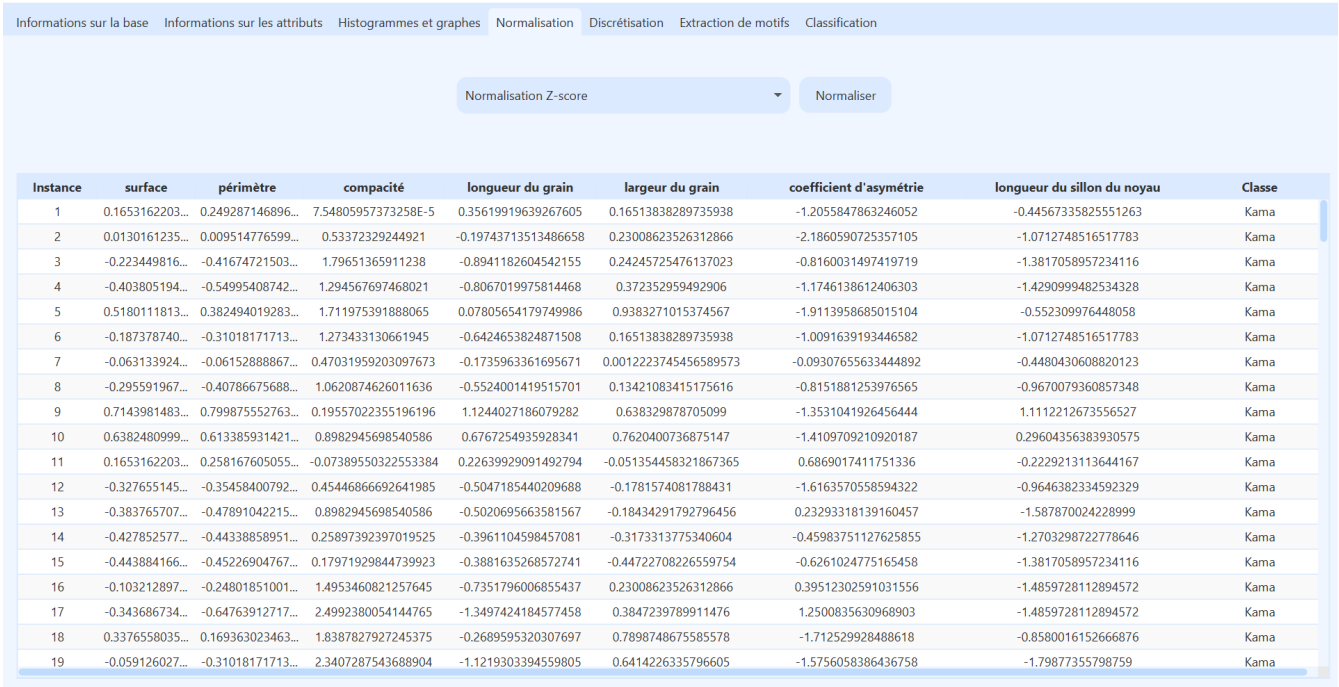
Interface de normalisation Min-Max

Pour la normalisation avec Min-Max il faut introduire les deux valeur Min et MAX.



Instance	surface	périmètre	compacité	longueur du grain	largeur du grain	coefficient d'asymétrie	longueur du sillon du noyau	Classe
1	0.4409820585...	0.502066115702...	0.570780399274047	0.48648648648648646	0.48610121168923714	0.18930164220052273	0.3451501723289019	Kama
2	0.4050991501...	0.446280991735...	0.6624319419237747	0.3688063063063065	0.5010691375623664	0.03288301759221938	0.21516494337764666	Kama
3	0.3493862134...	0.347107438016...	0.8793103448275864	0.22072072072072094	0.503920171062 01	0.25145301590191005	0.15066469719350079	Kama
4	0.3068932955...	0.316115702479...	0.7931034482758617	0.23930180180180172	0.5338560228082679	0.19424254638598865	0.14081733136386	Kama
5	0.5240793201...	0.533057851239...	0.8647912885662429	0.4273648648648651	0.6642908054169634	0.076701036289641	0.32299359921221066	Kama
6	0.3578847969...	0.371900826446...	0.7894736842105262	0.2742117117117118	0.48610121168923714	0.22063737663992516	0.21516494337764666	Kama
7	0.3871576959...	0.429752066115...	0.6515426497277677	0.3738738738738738	0.4483250178189592	0.3667841214942335	0.34465780403742013	Kama
8	0.3323890462...	0.349173553719...	0.7531760435571687	0.29335585585585583	0.47897362794012827	0.2515830396962645	0.23682914820285572	Kama
9	0.5703493862...	0.630165289256...	0.6043557168784031	0.6497747747747749	0.5951532430506056	0.16576733542238234	0.6686361398325947	Kama
10	0.5524079320...	0.586776859504...	0.7250453720508166	0.5546171171171174	0.6236635780470419	0.15653564602322226	0.4992614475627771	Kama
11	0.4409820585...	0.504132231404...	0.5580762250453722	0.4588963963963967	0.4362081254454739	0.4912168926913626	0.3914327917282127	Kama
12	0.3248347497...	0.361570247933...	0.6488203266787662	0.30349099099099086	0.40698503207412684	0.12376964984592183	0.23732151649433791	Kama
13	0.3116147308...	0.332644628099...	0.7250453720508166	0.3040540540540541	0.4055595153243049	0.4187936392359803	0.10782865583456443	Kama
14	0.3012275731...	0.340909090909...	0.6152450090744102	0.3265765765765766	0.3749109052031362	0.30827341403476843	0.17380600689315598	Kama
15	0.2974504249...	0.338842975206...	0.6016333938294005	0.3282657657657659	0.344975053456878	0.2817485599864776	0.15066469719350079	Kama
16	0.3777148253...	0.386363636363...	0.8275862068965515	0.2545045045045045	0.5010691375623664	0.44466837431249917	0.12900049236829128	Kama
17	0.3210576015...	0.293388429752...	1.0	0.12387387387387375	0.5367070563079115	0.5810633345902301	0.12900049236829128	Kama
18	0.4815864022...	0.483471074380...	0.8865698729582581	0.3536036036036037	0.6300784034212399	0.10842684211210653	0.25947808961102914	Kama
19	0.3881019830...	0.371900826446...	0.9727767695099818	0.17229729729729734	0.5958660014255167	0.13027083956364016	0.06400787789266367	Kama

Interface de normalisation Z-score



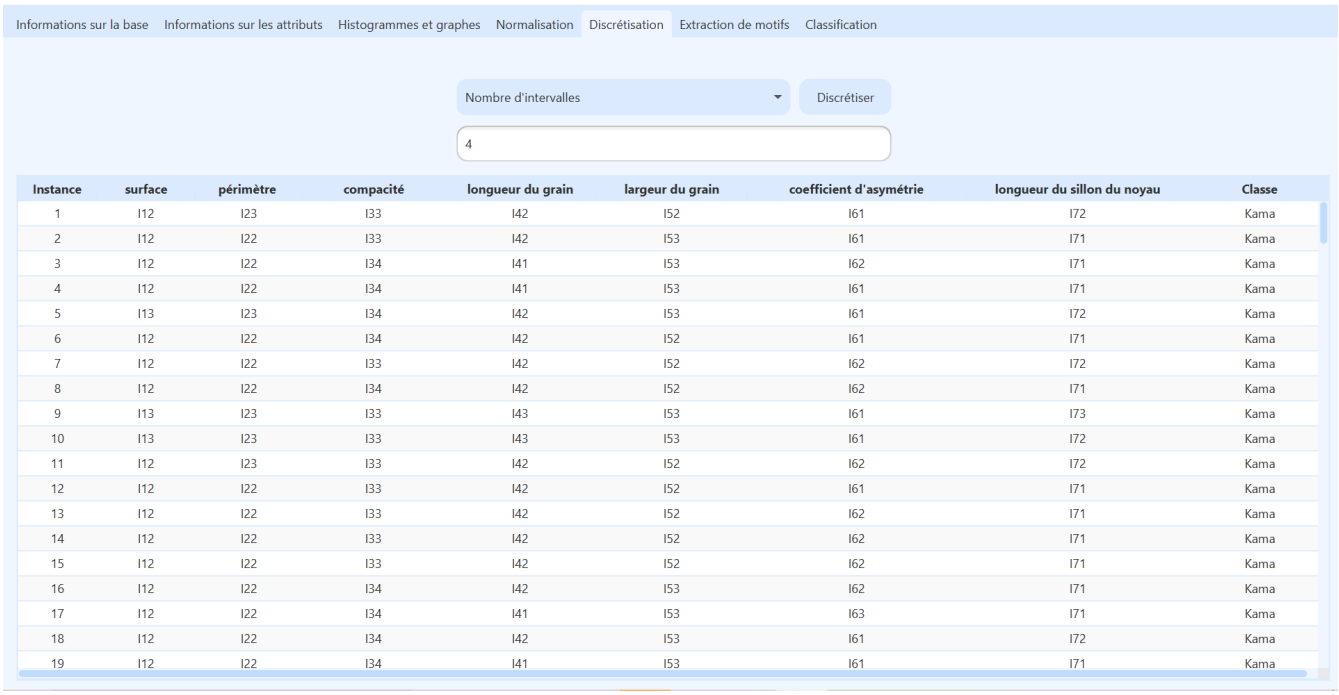
Instance	surface	périmètre	compacité	longueur du grain	largeur du grain	coefficient d'asymétrie	longueur du sillon du noyau	Classe
1	0.1653162203...	0.249287146896...	7.548059573732 58E-5	0.35619919639267605	0.16513838289735938	-1.2055847863246052	-0.44567335825551263	Kama
2	0.0130161235...	0.009514776599...	0.53372329244921	-0.19743713513486658	0.23008623526312866	-2.1860590725357105	-1.0712748516517783	Kama
3	-0.223449816...	-0.41674721503...	1.79651365911238	-0.8941182604542155	0.24245725476137023	-0.8160031497419719	-1.3817058957234116	Kama
4	-0.403805194...	-0.54995408742...	1.2945676974680 21	-0.8067019975814468	0.372352959492906	-1.1746138612406303	-1.4290999482534328	Kama
5	0.5180111813...	0.382494019283...	1.711975391888065	0.07805654179749986	0.9383271015374567	-1.9113958685015104	-0.552309976448058	Kama
6	-0.187378740...	-0.31018171713...	1.273433130661945	-0.6424653824871508	0.16513838289735938	-1.0091639193446582	-1.0712748516517783	Kama
7	-0.063133924...	-0.06152888867...	0.47031959203097673	-0.1735963361695671	0.0012223745456589573	-0.09307655633444892	-0.4480430608820123	Kama
8	-0.295591967...	-0.40786675688...	1.0620874626011636	-0.5524001419515701	0.13421083415175616	-0.8151881253976565	-0.9670079360857348	Kama
9	0.7143981483...	0.799875552763...	0.19557022355196196	1.1244027186079282	0.638329878705099	-1.3531041926456444	1.1112212673556527	Kama
10	0.6382480999...	0.613385931421...	0.8982945698540586	0.6767254935928341	0.7620400736875147	-1.4109709210920187	0.29604356383930575	Kama
11	0.1653162203...	0.258167605055...	-0.07389550322553384	0.22639929091492794	-0.051354458321867365	0.6869017411751336	-0.2229213113644167	Kama
12	-0.327655145...	-0.35458400792...	0.45446866692641985	-0.5047185440209688	-0.1781574081788431	-1.6163570558594322	-0.9646382334592329	Kama
13	-0.383765707...	-0.47891042215...	0.8982945698540586	-0.5020695663581567	-0.18434291792796456	0.23293318139160457	-1.587870024228999	Kama
14	-0.427852577...	-0.44338858951...	0.25897392397019525	-0.3961104598457081	-0.3173313775340604	-0.45983751127625855	-1.2703298722778646	Kama
15	-0.443884166...	-0.45226904767...	0.17971929844739923	-0.3881635268572741	-0.44722708226559754	-0.6261024775165458	-1.3817058957234116	Kama
16	-0.103212897...	-0.24801851001...	1.4953460821257645	-0.7351796006855437	0.23008623526312866	0.39512302591031556	-1.4859728112894572	Kama
17	-0.343686734...	-0.64763912717...	2.4992380054144765	-1.3497424184577458	0.3847239789911476	1.2500835630968903	-1.4859728112894572	Kama
18	0.3376558035...	0.169363023463...	1.8387827927245375	-0.2689595320307697	0.7898748675585578	-1.712529928488618	-0.8580016152666876	Kama
19	-0.059126027...	-0.31018171713...	2.3407287543688904	-1.1219303394559805	0.6414226335796605	-1.5756058386436758	-1.79877355798759	Kama

5.2 Interfaces de discrétisation

l'onglet ci-dessous permet la discrétisation avec les méthodes Intervalles-égaux ou Quantiles.

Discrétisation avec Intervalles-égaux

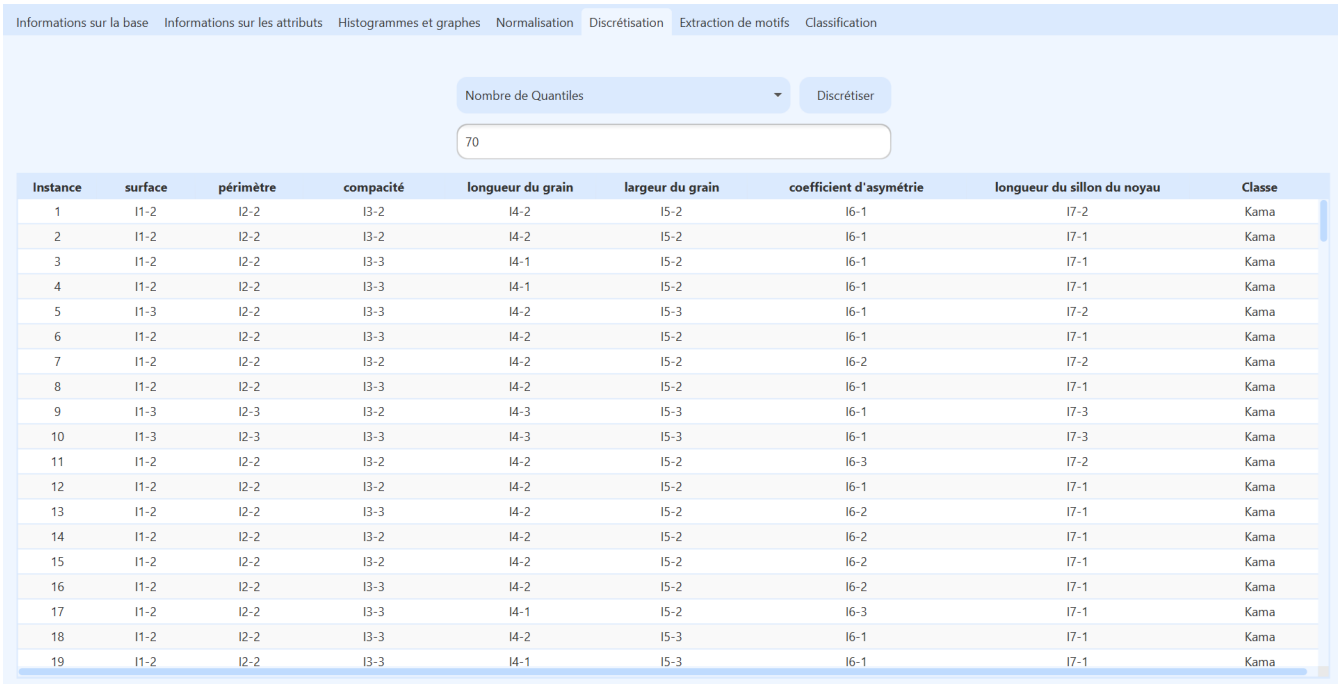
Pour la discrétisation avec Intervalles-égaux il faut introduire le nombre d'intervalles.



Instance	surface	périmètre	compacité	longueur du grain	largeur du grain	coefficient d'asymétrie	longueur du sillon du noyau	Classe
1	I12	I23	I33	I42	I52	I61	I72	Kama
2	I12	I22	I33	I42	I53	I61	I71	Kama
3	I12	I22	I34	I41	I53	I62	I71	Kama
4	I12	I22	I34	I41	I53	I61	I71	Kama
5	I13	I23	I34	I42	I53	I61	I72	Kama
6	I12	I22	I34	I42	I52	I61	I71	Kama
7	I12	I22	I33	I42	I52	I62	I72	Kama
8	I12	I22	I34	I42	I52	I62	I71	Kama
9	I13	I23	I33	I43	I53	I61	I73	Kama
10	I13	I23	I33	I43	I53	I61	I72	Kama
11	I12	I23	I33	I42	I52	I62	I72	Kama
12	I12	I22	I33	I42	I52	I61	I71	Kama
13	I12	I22	I33	I42	I52	I62	I71	Kama
14	I12	I22	I33	I42	I52	I62	I71	Kama
15	I12	I22	I33	I42	I52	I62	I71	Kama
16	I12	I22	I34	I42	I53	I62	I71	Kama
17	I12	I22	I34	I41	I53	I63	I71	Kama
18	I12	I22	I34	I42	I53	I61	I72	Kama
19	I12	I22	I34	I41	I53	I61	I71	Kama

Discrétisation avec Quantiles

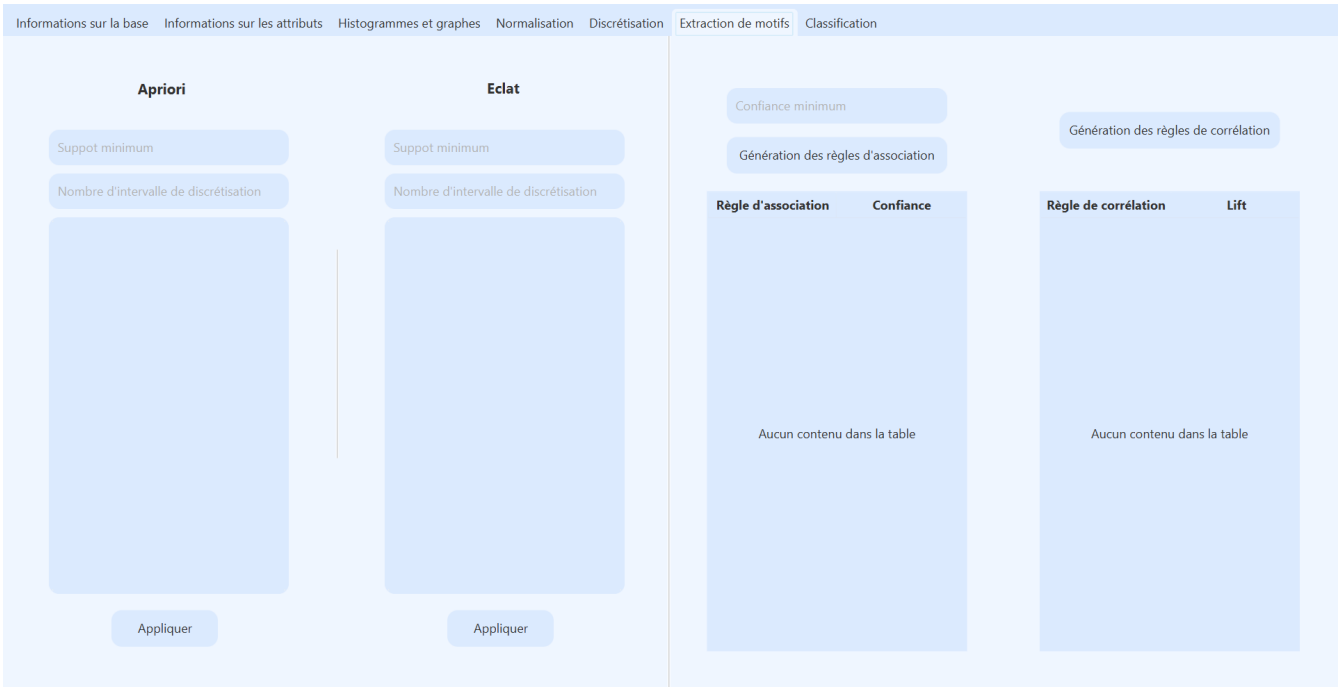
Pour la discrétisation avec Quantiles il faut introduire le nombre de quantiles.



Instance	surface	périmètre	compacité	longueur du grain	largeur du grain	coefficient d'asymétrie	longueur du sillon du noyau	Classe
1	I1-2	I2-2	I3-2	I4-2	I5-2	I6-1	I7-2	Kama
2	I1-2	I2-2	I3-2	I4-2	I5-2	I6-1	I7-1	Kama
3	I1-2	I2-2	I3-3	I4-1	I5-2	I6-1	I7-1	Kama
4	I1-2	I2-2	I3-3	I4-1	I5-2	I6-1	I7-1	Kama
5	I1-3	I2-2	I3-3	I4-2	I5-3	I6-1	I7-2	Kama
6	I1-2	I2-2	I3-3	I4-2	I5-2	I6-1	I7-1	Kama
7	I1-2	I2-2	I3-2	I4-2	I5-2	I6-2	I7-2	Kama
8	I1-2	I2-2	I3-3	I4-2	I5-2	I6-1	I7-1	Kama
9	I1-3	I2-3	I3-2	I4-3	I5-3	I6-1	I7-3	Kama
10	I1-3	I2-3	I3-3	I4-3	I5-3	I6-1	I7-3	Kama
11	I1-2	I2-2	I3-2	I4-2	I5-2	I6-3	I7-2	Kama
12	I1-2	I2-2	I3-2	I4-2	I5-2	I6-1	I7-1	Kama
13	I1-2	I2-2	I3-3	I4-2	I5-2	I6-2	I7-1	Kama
14	I1-2	I2-2	I3-2	I4-2	I5-2	I6-2	I7-1	Kama
15	I1-2	I2-2	I3-2	I4-2	I5-2	I6-2	I7-1	Kama
16	I1-2	I2-2	I3-3	I4-2	I5-2	I6-2	I7-1	Kama
17	I1-2	I2-2	I3-3	I4-1	I5-2	I6-3	I7-1	Kama
18	I1-2	I2-2	I3-3	I4-2	I5-3	I6-1	I7-1	Kama
19	I1-2	I2-2	I3-3	I4-1	I5-3	I6-1	I7-1	Kama

5.3 Interfaces d'extraction de motifs fréquents

l'onglet ci-dessous permet l'extraction des motifs fréquents avec les différentes algorithmes Apriori et Eclat ainsi l'extraction des règles d'association et de corrélation

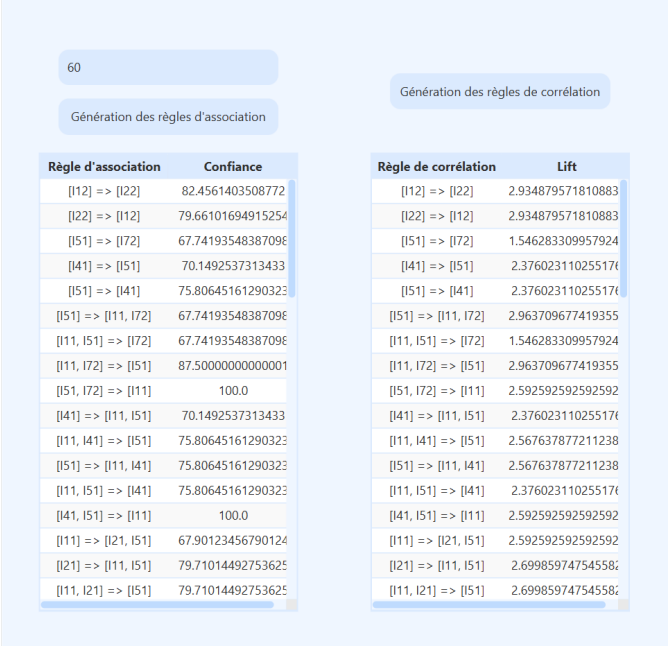


Interface de l'application des algorithmes Apriori et Eclat

Tous les deux demande en entrée le support minimun ainsi que la valeur de l'intervalle de discrétisation.

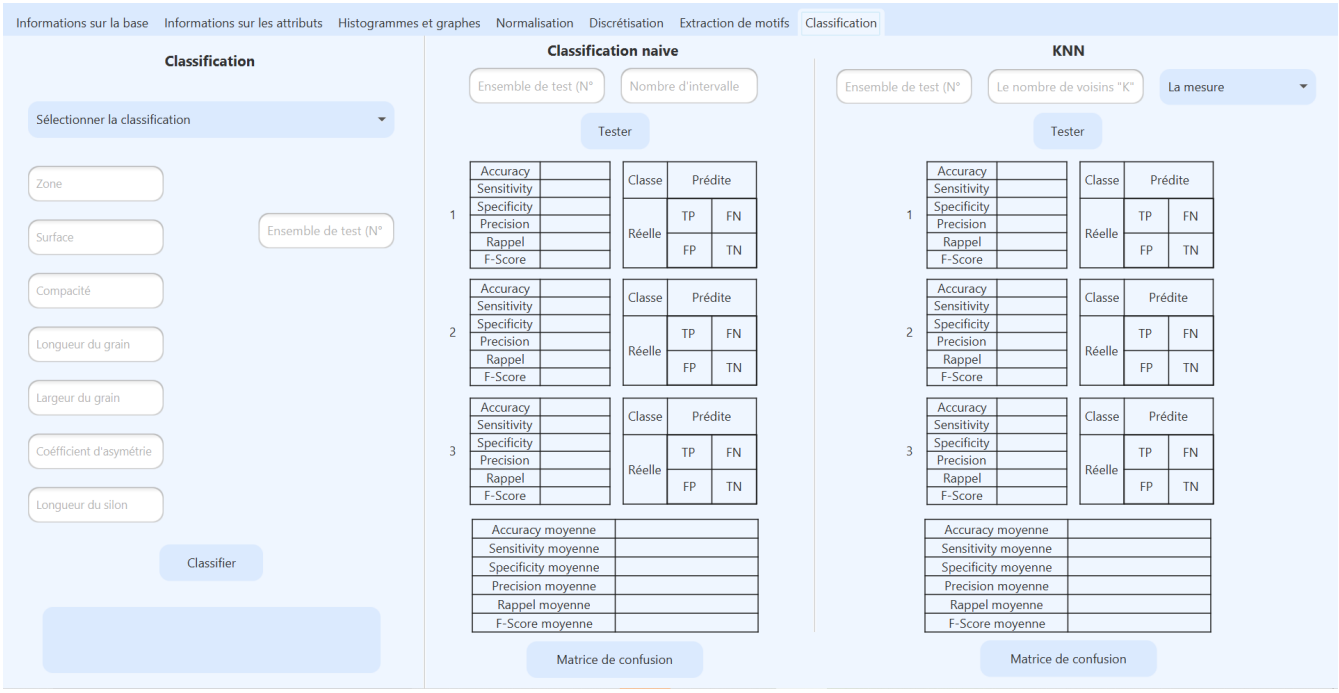
Extraction des règles d'association et de corrélation

Après l'extraction des motifs fréquents en utilisant l'un des deux algorithmes en peut générer les règles d'association et les règles de corrélation en introduisant la confiance minimun.



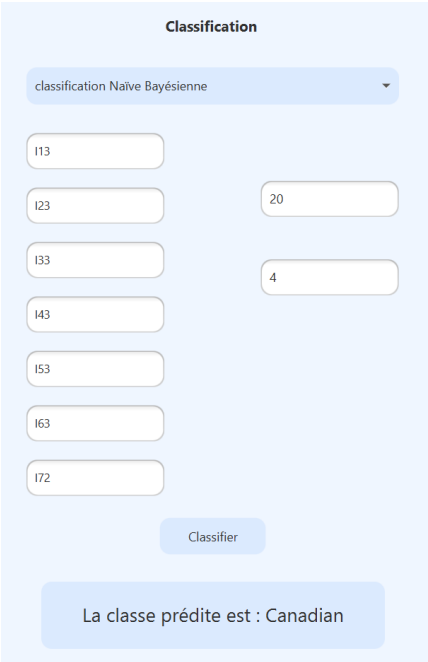
5.4 Interfaces pour la classification

l'onglet ci-dessous permet la classification des instances avec les différentes algorithmes de classification Naïve Bayésienne ou K-nearest neighbors(KNN) ainsi que la comparaison entre les deux algorithmes.



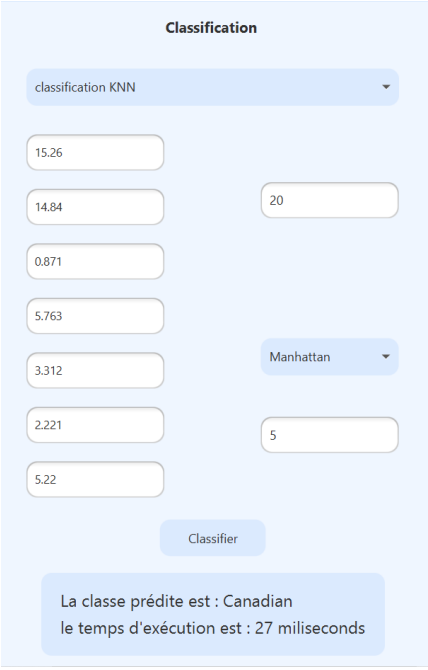
classification Naïve Bayésienne

Pour classifier une instance on doit introduire les items de chaque attribut et le pourcentage de l'ensemble de test (T), et le nombre d'intervalles.



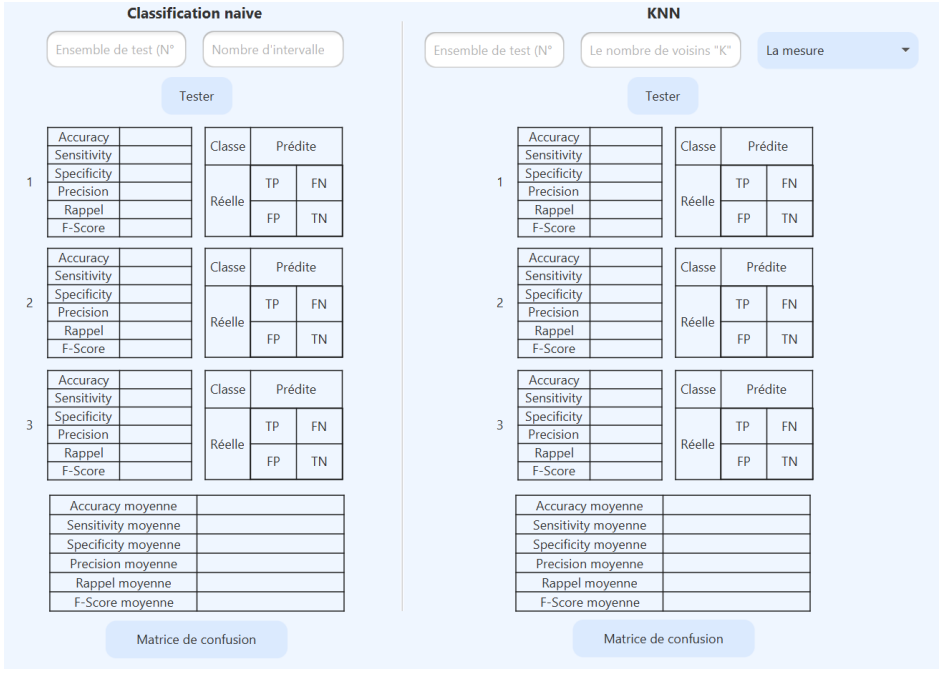
classification KNN

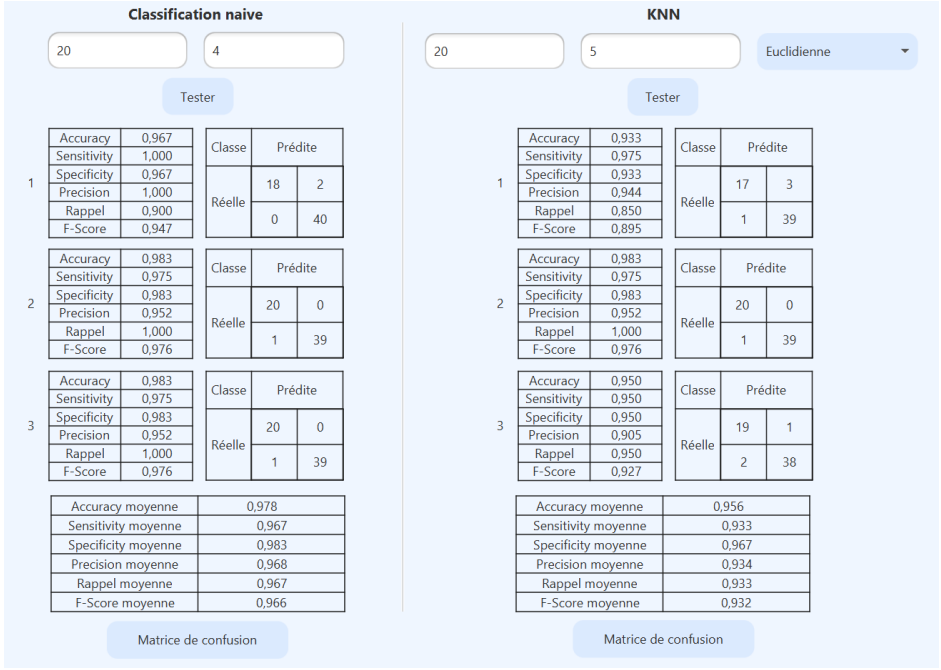
Pour classifier une instance on doit introduire les valeurs des items de chaque attribut et le pourcentage de l'ensemble de test (T), sélectionnée une mesures de distance (euclidienne ou Manhattan) et introduire le nombre de voisins.



5.5 Interface pour calculs des mesures d'evaluation d'un algorithme de classification

l'onglet ci-dessous permet les calcules des différentes mesures d'évaluation des classifieurs ainsi que la matrice de confusion pour chaque méthodes.





6 Résultats

D'après les différentes exécutions des algorithmes d'extraction des items fréquents on peut remarquer que l'algorithme Eclat et plus rapide que Apriori (La mesure du temps est en milliseconds). Le tableau ci-dessous montre les resultats.

Support min et intervalles	Apriori	Eclat
sup-min = 20 et int = 4	**47**	**16**
sup-min = 30 et int = 4	**29**	**12**
sup-min = 40 et int = 4	**11**	**8**
sup-min = 20 et int = 5	**34**	**18**
sup-min = 30 et int = 5	**13**	**9**
sup-min = 40 et int = 5	**8**	**7**

TABLE 1 – Tableau comparatif entre les algorithmes des items fréquents Apriori et Eclat

Par rapport aux résultats de la classification on remarque la classification Naïve Bayésienne et plus rapide que la classification KNN.

Conclusion

A partir de ce projet nous avons pu mieux comprendre les taches du data mining les plus importantes tel que l'extraction de règles d'associations et la classification. Ce projet nous a permis d'appliquer les algorithmes d'extraction des motifs fréquents A-Priori et Eclat, ainsi que la classification supervisé avec les deux algorithmes Naïve Bayésienne et KNN, voir l'utilité et l'importance de ces derniers et comprendre les avantages et inconvénients de chaque méthode.