Master SIDI S3

Projet Data Mining

Encadré par : Prof . Mohamed Sabiri
Réalisé par : Boutougasse Aziza

Introduction :

Dans ce travail en met en ouvres les étapes clés nécessaires pour élaborer un projet de Data mining étapes par étapes avec Weka .

Nous allons suivre le processus suivant:

1. Chargez l'ensemble de données.
2. Analysez l'ensemble de données.
3. Préparez des vues de l'ensemble de données.
4. Évaluer les algorithmes.
5. Finaliser le modèle et présenter les résultats.

1. Charger l'ensemble de données

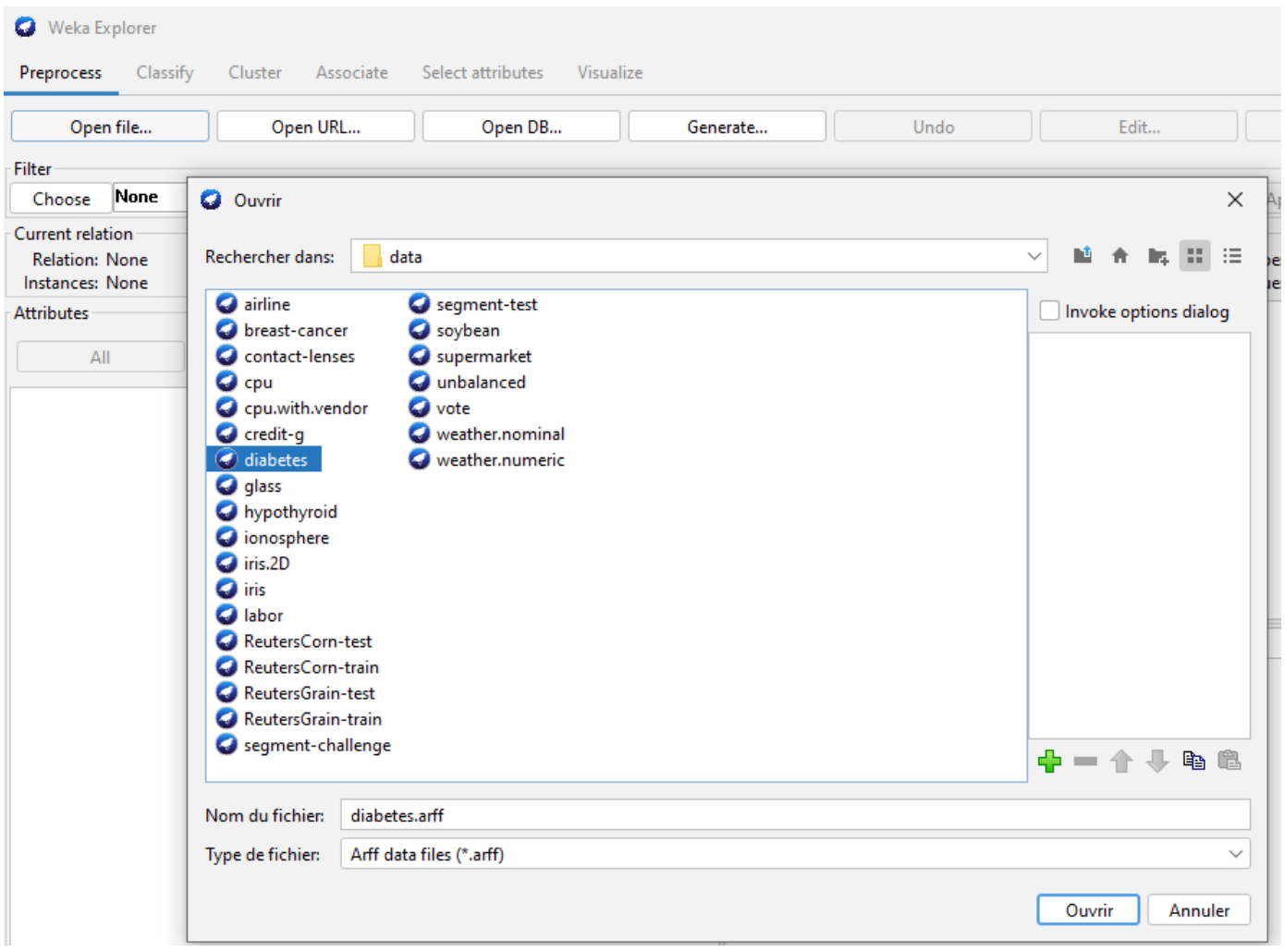
Le problème utilisé dans ce projet est l'ensemble de données sur l'apparition du diabète chez les adultes.

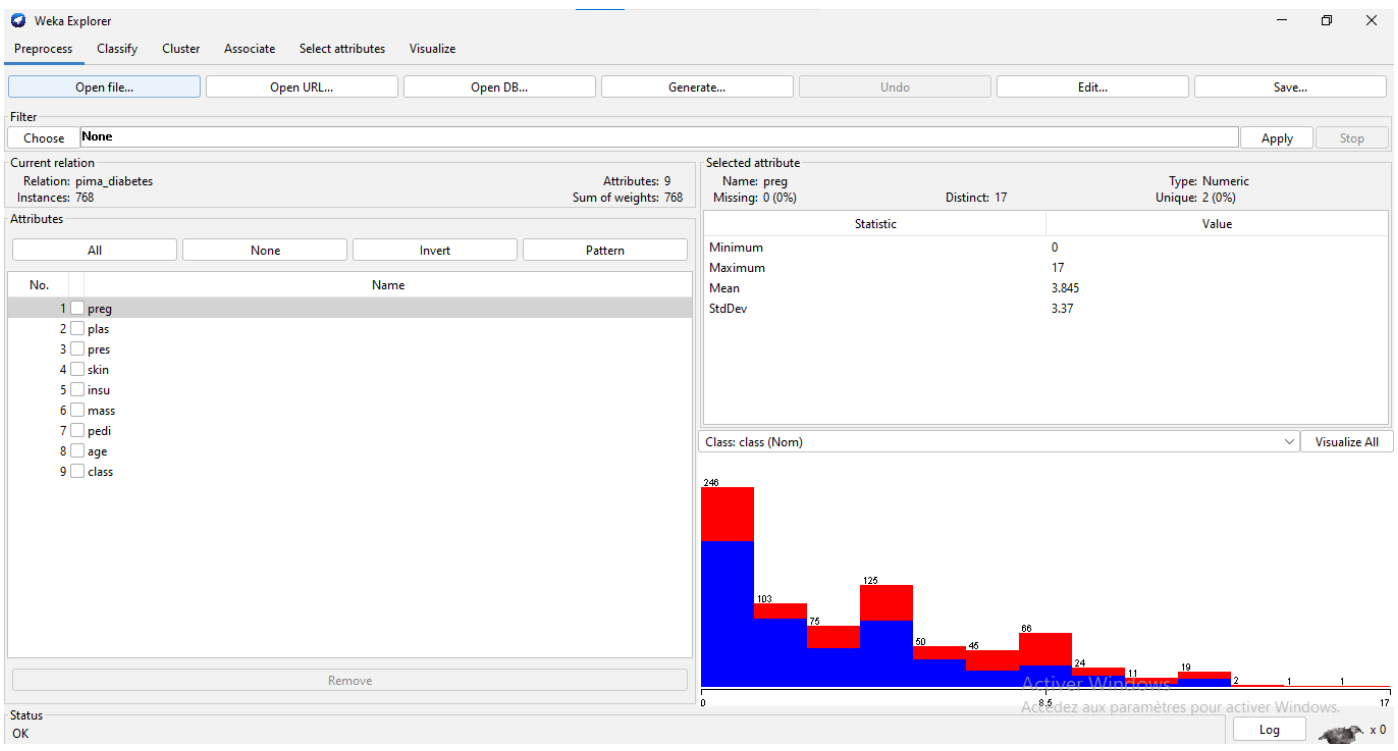
Dans cet ensemble de données, chaque instance représente des détails médicaux pour un patient et la tâche consiste à prédire si le patient aura un début de diabète au cours des cinq prochaines années. Il y a 8 variables d'entrée numériques qui ont toutes des échelles variables.

1. on Ouvre le sélecteur d'interface graphique **Weka.**

2. on clique sur le bouton" Explorer " pour ouvrir l'explorateur **Weka.**

3. puis sur le bouton " Ouvrir le fichier"", accédez au répertoire données / et sélectionnez diabète.arff.





2. Analyser l'ensemble de données

On va revoir nos données avant de commencer la modélisation.

L'examen de la distribution de chaque attribut et des interactions entre les attributs peut éclairer des transformations de données spécifiques et des techniques de modélisation spécifiques que nous pourrions utiliser.
Statistiques Sommaires

Passez en revue les détails de l'ensemble de données dans le volet “Relation actuelle”. On peut remarquer quelques petites choses:

L'ensemble de données porte le nom pima_diabetes.
Il y a 768 instances dans l'ensemble de données. Si nous évaluons les modèles en utilisant une validation croisée 10 fois, chaque pli aura environ 76 instances, ce qui est bien.
Il y a 9 attributs, 8 attributs d'entrée et un attribut de sortie.

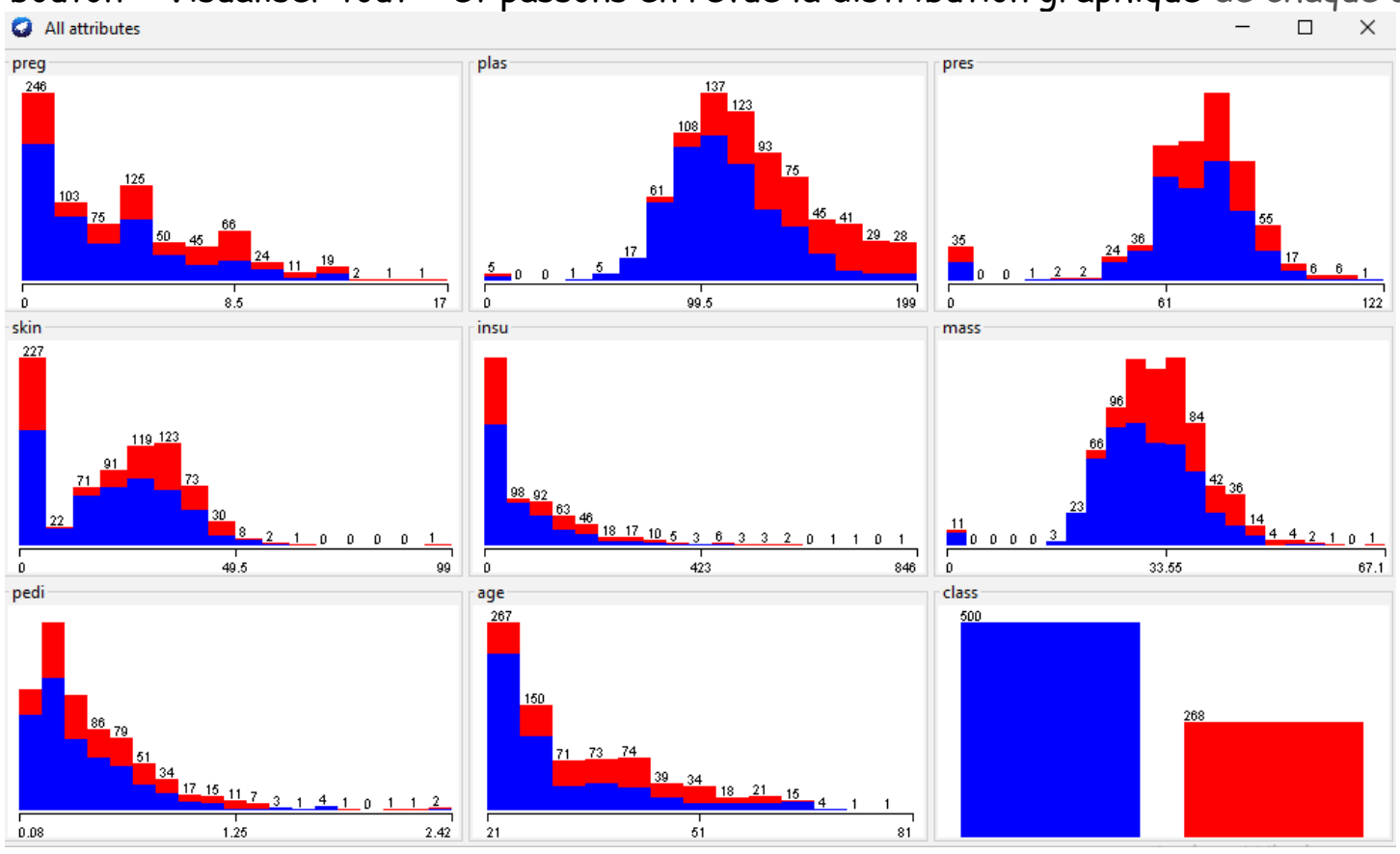
On clique sur chaque attribut dans le volet” Attributs "et examinez les statistiques récapitulatives dans le volet "Attribut sélectionné".

Nous pouvons remarquer quelques faits sur nos données:

- Les attributs d'entrée sont tous numériques et ont des échelles différentes. Nous pourrions voir certains avantages à normaliser ou à normaliser les données.
- Il n'y a pas de valeurs manquantes marquées.
- Il y a des valeurs pour certains attributs qui ne semblent pas raisonnables, en particulier: please, pres, skin, insu et mass ont des valeurs de 0. Ce sont probablement des données manquantes qui pourraient être marquées.
- L'attribut class est nominal et a deux valeurs de sortie, ce qui signifie qu'il s'agit d'un problème de classification binaire ou à deux classes.
- L'attribut de classe est déséquilibré, 1 résultat "positif" pour 1,8 résultats "négatifs", soit près du double du nombre de cas négatifs. Nous pourrions bénéficier de l'équilibrage des valeurs de classe.

Distributions d'attributs

Cliquez sur le bouton « Visualiser tout » et passons en revue la distribution graphique de chaque attribut.



Nous pouvons remarquer quelques petites choses sur la forme des données:

Certains attributs ont une distribution de type gaussienne tels que plas, pres, skin et mass, suggérant que les méthodes qui font cette hypothèse pourraient obtenir de bons résultats, comme la régression logistique et les Bayes naïfs.

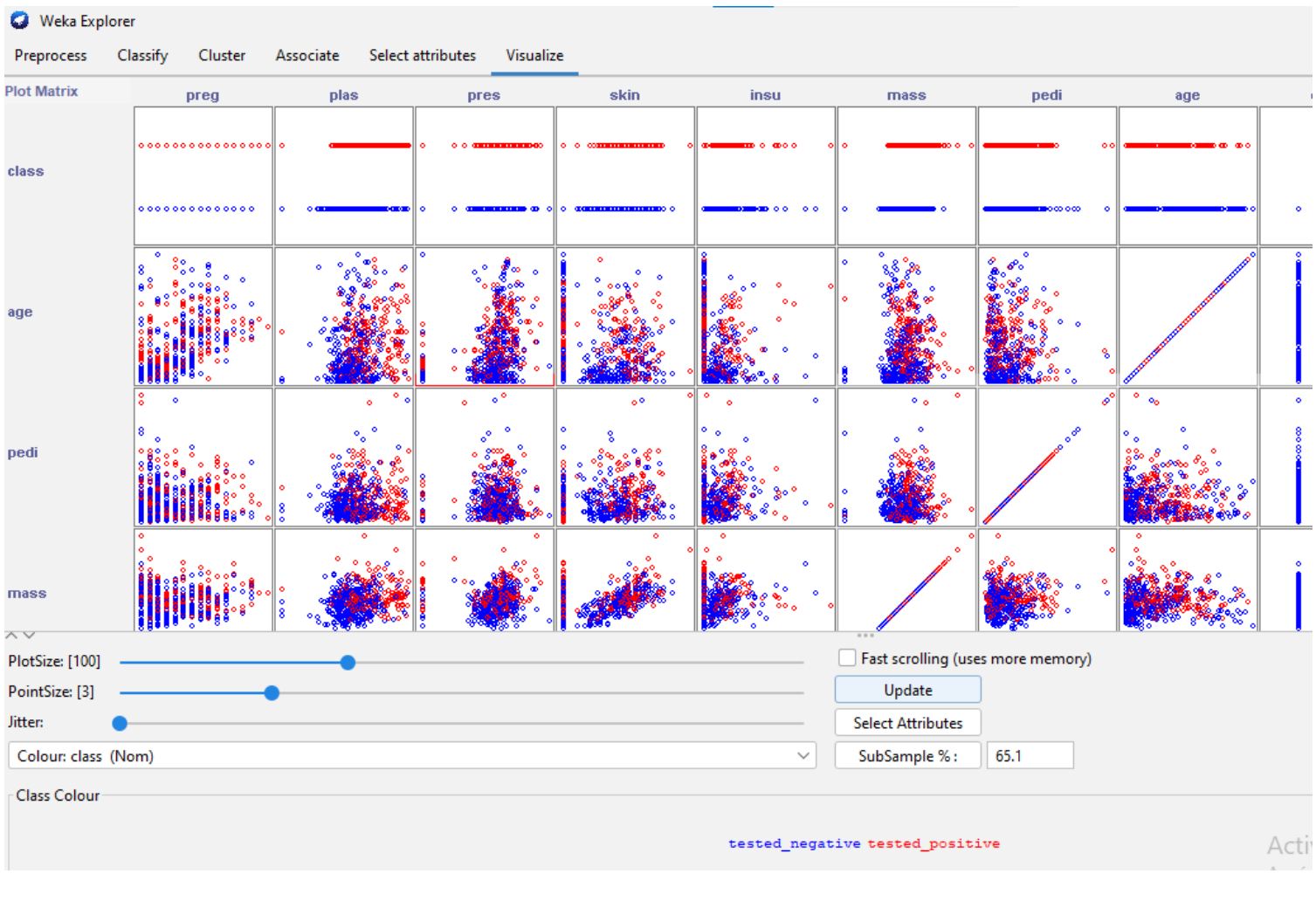
Nous voyons beaucoup de chevauchement entre les classes à travers les valeurs d'attribut. Les classes ne semblent pas facilement séparables.

Nous pouvons clairement voir le déséquilibre des classes représenté graphiquement.

Interactions d'attributs

Cliquez sur l'onglet" Visualiser " et passons en revue certaines interactions entre les attributs.

- On Augmente la taille de la fenêtre pour que tous les tracés soient visibles.
- On Augmente la "taille des points" à 3 pour rendre les points plus faciles à voir.
- Puis on Clique sur le bouton" Mettre à jour " pour appliquer les modifications.



En regardant à travers les graphiques pour les variables d'entrée, nous pouvons généralement voir une mauvaise séparation entre les classes sur les nuages de points. Cet ensemble de données ne sera pas une promenade dans le parc.

Cela suggère que nous pourrions bénéficier de bonnes transformations de données et de la création de plusieurs vues de l'ensemble de données. Cela suggère également que nous pourrions tirer des avantages de l'utilisation de méthodes d'ensemble.

3. Préparer des vues de l'ensemble de données

Dans cette section, nous allons créer des vues variées des données, de sorte que lorsque nous évaluerons les algorithmes dans la section suivante, nous pourrons avoir une idée des vues qui sont généralement les meilleures pour exposer la structure du problème de classification aux modèles.

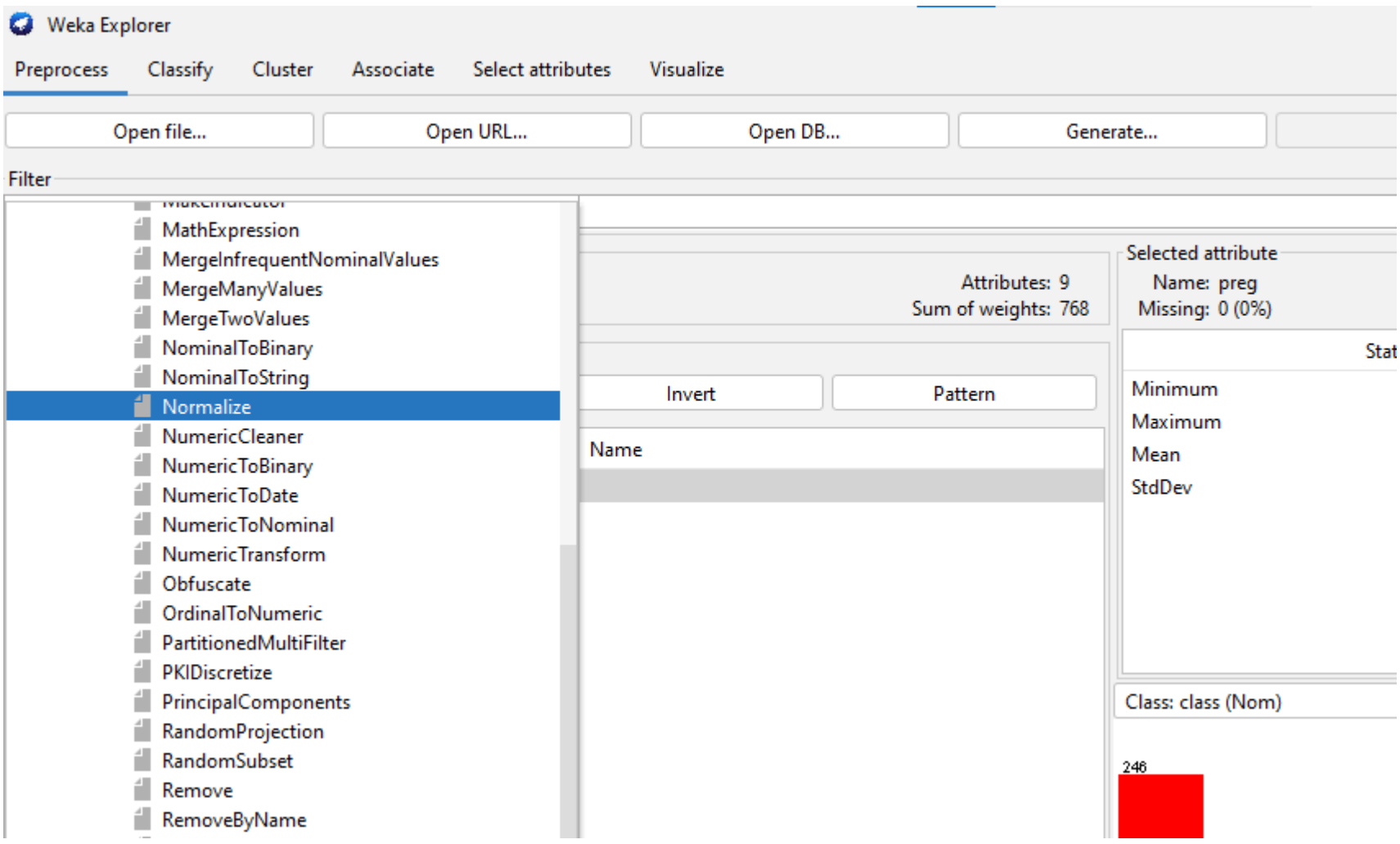
Nous allons créer 3 vues supplémentaires des données, de sorte qu'en plus des données brutes, nous aurons 4 copies différentes de l'ensemble de données au total. Nous allons créer chaque vue de l'ensemble de données à partir de l'original et l'enregistrer dans un nouveau fichier pour une utilisation ultérieure dans nos expériences

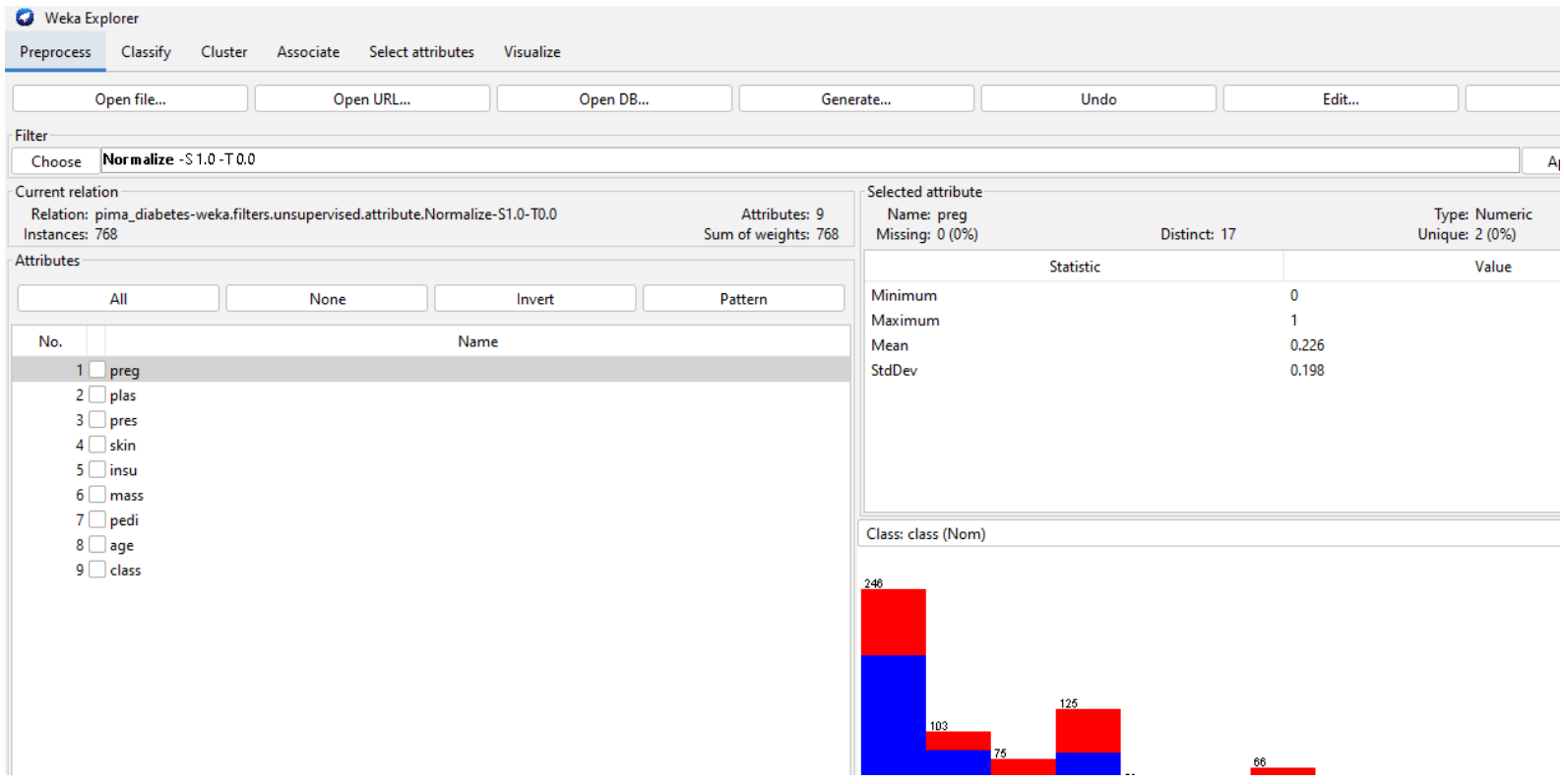
Vue Normalisée

La première vue que nous allons créer est celle de tous les attributs d'entrée normalisés entre 0 et 1.

- Dans l'explorateur avec les données / diabète.fichier arff chargé.
- On Clique sur le bouton "Choisir"dans le volet "Filtre" et choisissez “non supervisé.attribut.Filtre "Normaliser".
- puis sur le bouton” Appliquer " pour appliquer le filtre.

- On Clique sur chaque attribut dans le volet" Attributs "et examinez les valeurs min et max dans le volet "Attribut sélectionné" pour confirmer qu'elles sont 0 et 1.
- On Clique sur le bouton" Enregistrer...", accédez à un répertoire approprié et saisissez un nom approprié pour cet ensemble de données transformé, tel que "diabète-normaliser.arff".



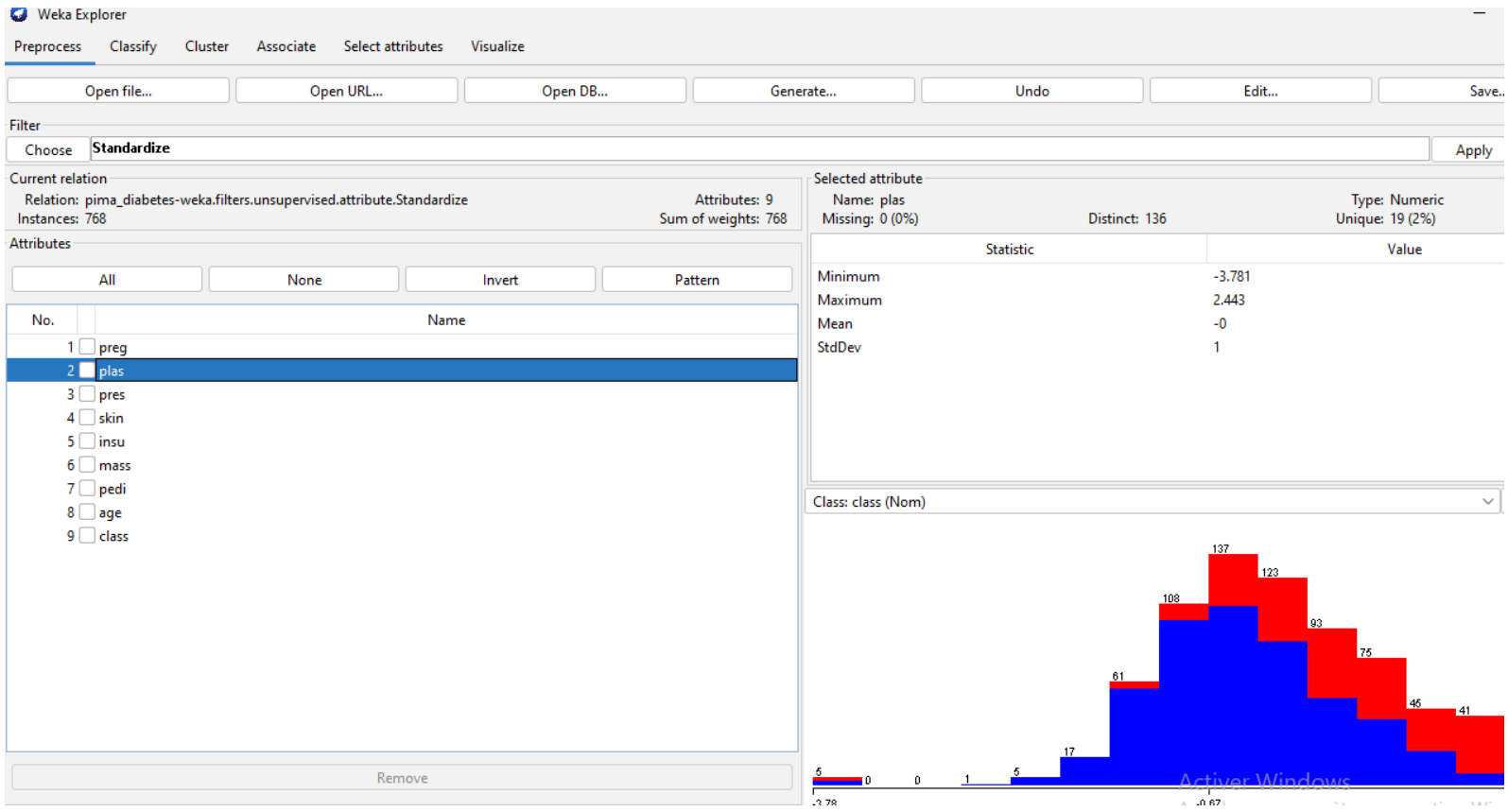


Vue standardisée

Nous avons noté dans la section précédente que certains attributs ont une distribution de type gaussienne. Nous pouvons redimensionner les données et prendre en compte cette répartition en utilisant un filtre standardisant.
Cela créera une copie de l'ensemble de données où chaque attribut a une valeur moyenne de 0 et un écart type (variance moyenne) de 1.

1. On Ouvre l'explorateur Weka.
2. On Charge l'ensemble de données sur l'apparition du diabète .
3. On Clique sur le bouton « Choisir » dans le volet « Filtrer » et choisissez le filtre « unsupervised.attribute.Standardize ».
4. On Clique sur le bouton « Appliquer » pour appliquer le filtre.

5. On Clique sur chaque attribut dans le volet « Attributs » et examinez les valeurs moyennes et d'écart type dans le volet « Attribut sélectionné » pour confirmer qu'elles sont respectivement 0 et 1.
1. On Clique sur le bouton « Enregistrer… »,sous « *diabetes-standardize.arff* ".

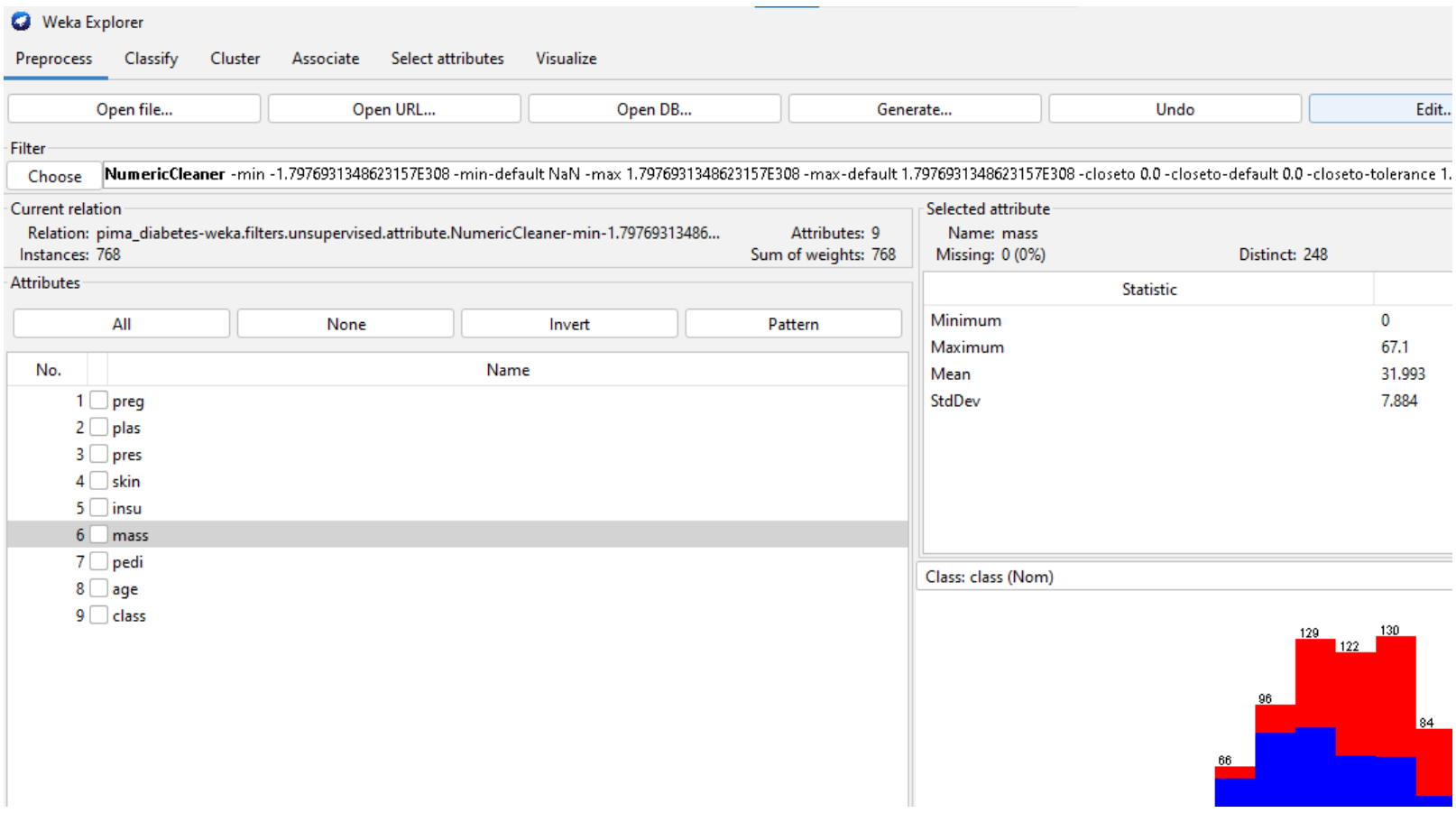


Données Manquantes :

Nous pouvons créer une nouvelle copie de l'ensemble de données avec les données manquantes marquées puis imputées avec une valeur moyenne pour chaque attribut. Cela peut aider les méthodes qui supposent un changement en douceur des distributions d'attributs, telles que la régression logistique et les méthodes basées sur les instances.

Marquons d'abord les valeurs 0 pour certains attributs comme manquantes.

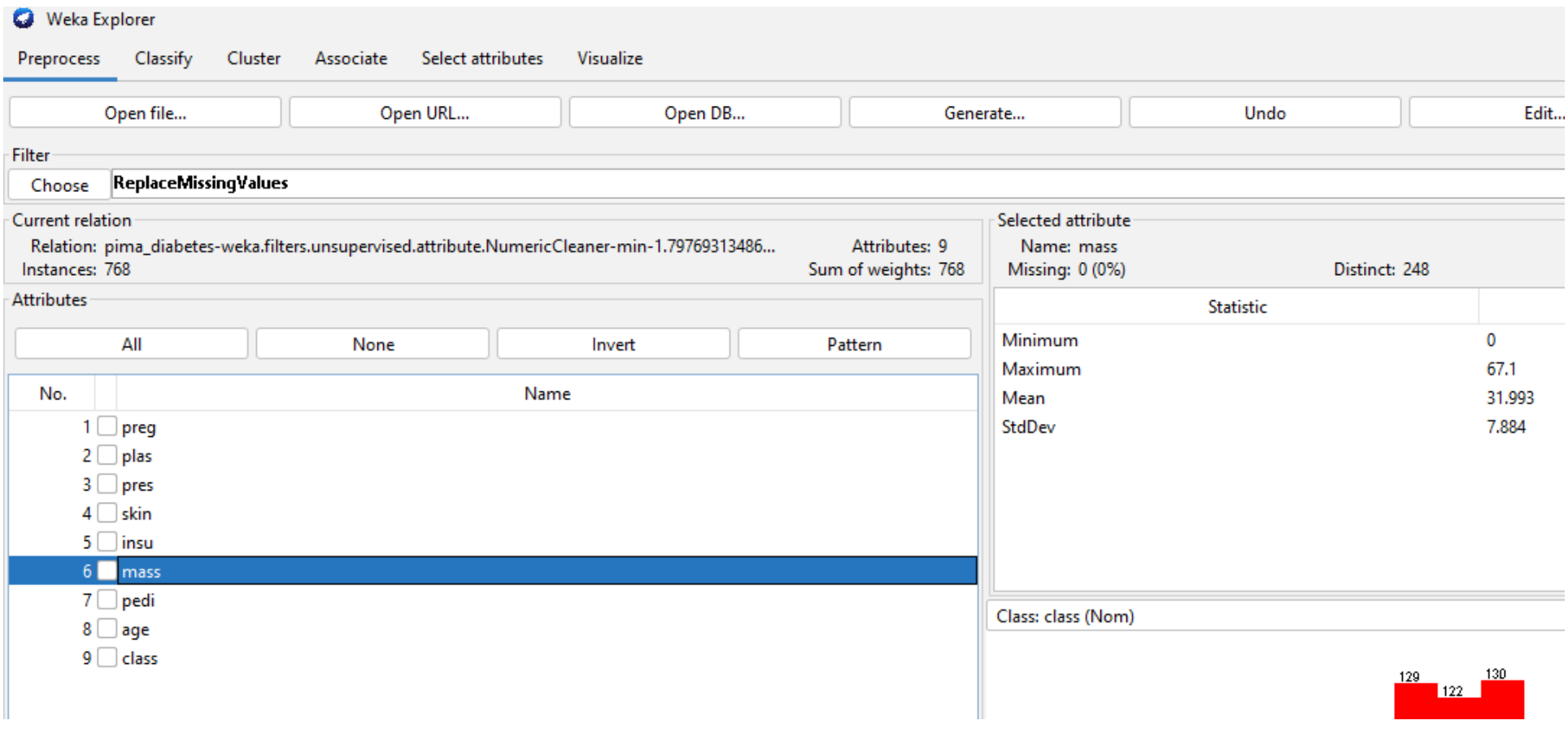
- On Ouvre l'explorateur Weka.
- On Chargel'ensemble de données sur l'apparition du diabète chez les Indiens Pima.
- On Clique sur le bouton "Choisir" pour le filtre et sélectionnez le non supervisé.attribut.Filtre de nettoyage numérique.
- On Cliquez sur le filtre pour le configurer.
- On Définisse les indices d'attribut sur 2-6
- On Définissez minThreshold sur 0,1 E-8 (proche de zéro), qui est la valeur minimale autorisée pour chaque attribut.
- On Définisse minDefault sur NaN, qui est inconnu et remplacera les valeurs inférieures au seuil.
- On Clique sur le bouton" OK " dans la configuration du filtre.
- On Clique sur le bouton" Appliquer " pour appliquer le filtre.
- On Clique sur chaque attribut dans le volet "Attributs" et vérifiez le nombre de valeurs manquantes pour chaque attribut. Vous devriez voir des nombres non nuls pour les attributs 2 à 6.



Maintenant, imputons les valeurs manquantes comme moyenne.

- On Clique sur le bouton" Choisir "dans le volet" Filtre " et sélectionnez non supervisé.attribut.Filtre de remplacement des valeurs manquantes.
- On Clique sur le bouton" Appliquer " pour appliquer le filtre à votre jeu de données.
- On Clique sur chaque attribut dans le volet "Attributs" et vérifiez le nombre de valeurs manquantes pour chaque attribut. Vous devriez voir que tous les attributs ne devraient avoir aucune valeur manquante et que la distribution des attributs 2 à 6 devrait avoir légèrement changé.
- On Clique sur le bouton" Enregistrer...", accédez à un répertoire approprié et saisissez un nom approprié pour cet ensemble de données transformé, tel que " diabète-manquant.arff".

D'autres vues des données que vous pouvez envisager d'examiner sont des sous-ensembles d'entités choisis par une méthode de sélection d'entités et une vue où l'attribut de classe est rééquilibré.



4. Évaluer Les Algorithmes

Concevons une expérience pour évaluer une suite d'algorithmes de classification standard sur les différentes vues du problème que nous avons créées.

- Cliquez sur le bouton "Expérimentateur" du sélecteur d'interface graphique Weka pour lancer l'environnement d'expérimentation Weka.
- Cliquez sur "Nouveau" pour commencer une nouvelle expérience.
- Dans le volet "Jeux de données", cliquez sur "Ajouter un nouveau..." et sélectionnez les 4 jeux de données suivants:

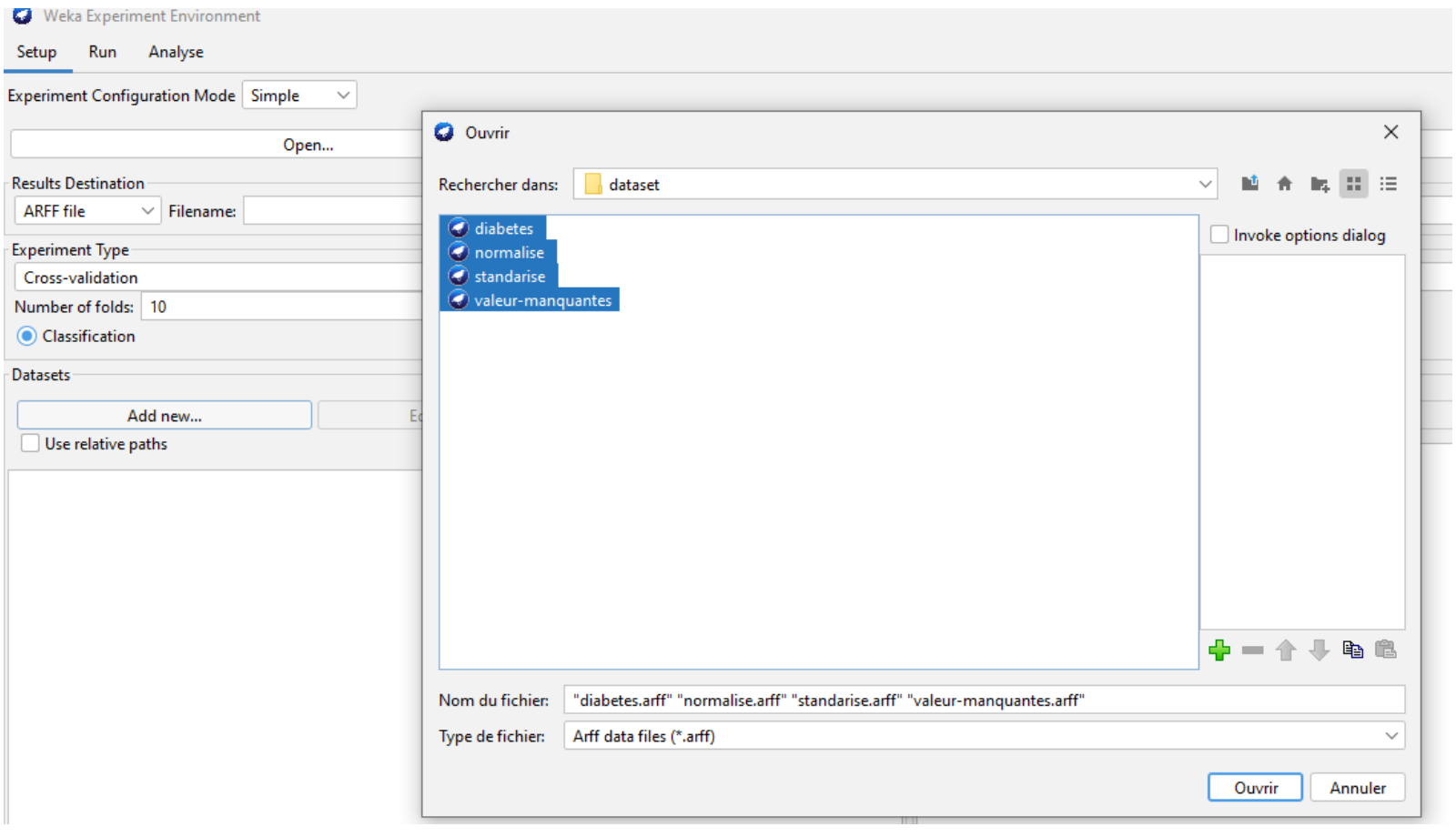
- données / diabète.arff (l'ensemble de données brutes)
- diabète-normalisé.arff
- diabète-Stondarsé.arff
- diabète-manquant.arff

Dans le volet" Algorithmes", cliquez sur "Ajouter un nouveau..." et ajoutez les 8 algorithmes de classification multiclasses suivants:

- rules.ZeroR
- bayes.NaiveBayes
- functions.Logistic
- functions.SMO
- lazy.IBk
- rules.PART
- trees.REPTree
- trees.J48

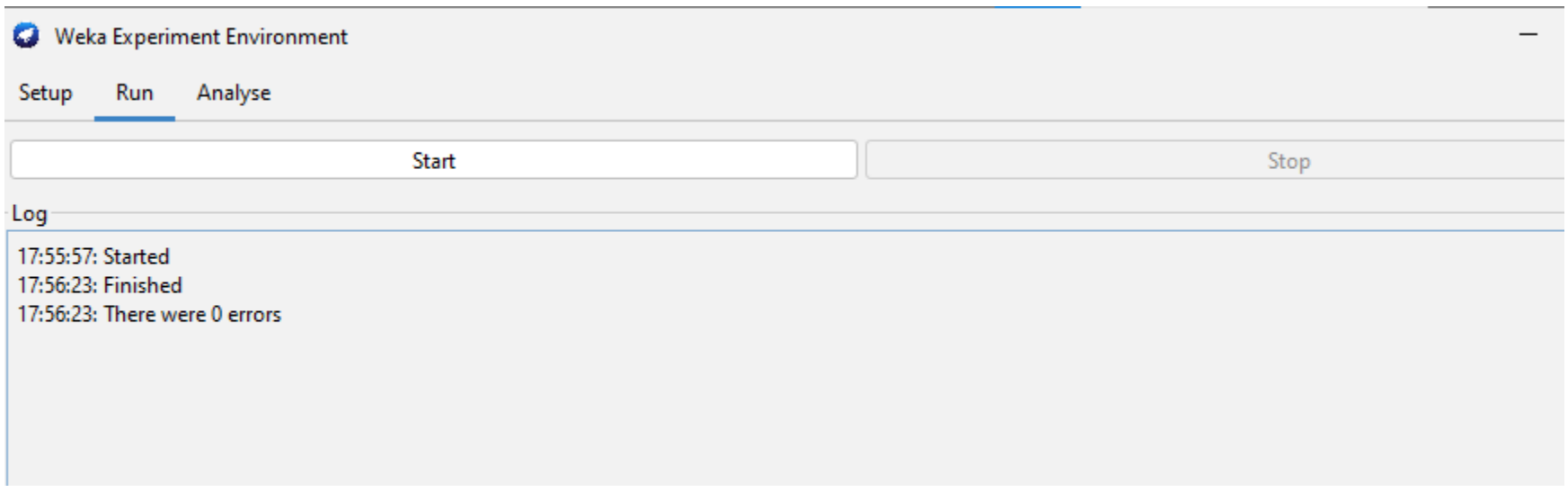
on Sélectionnne IBK dans la liste des algorithmes et cliquez sur le bouton" Modifier la sélection selected".

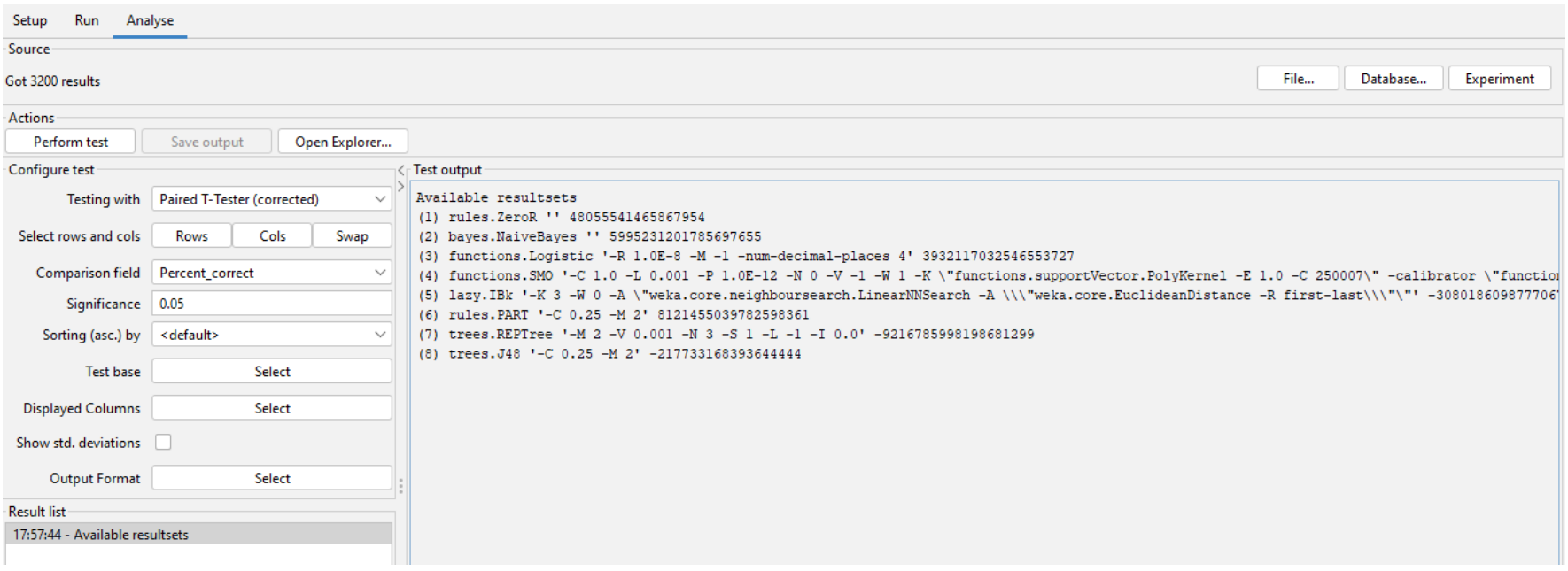
Puis dans " KNN on met " 3 " et cliquez sur le bouton" OK" pour enregistrer les paramètres.



On Clique sur "Exécuter" pour ouvrir l'onglet Exécuter et cliquez sur le bouton "Démarrer" pour exécuter l'expérience. L'expérience devrait se terminer en quelques secondes seulement.

On Clique sur "Analyser" pour ouvrir l'onglet Analyser. On Clique sur le bouton" Expérience " pour charger les résultats de l'expérience.
on Clique sur le bouton "Effectuer un test" pour effectuer un test par paires-test comparant tous les résultats aux résultats pour ZeroR.





On Clique sur le bouton "Effectuer un test" pour effectuer un test par paires-test comparant tous les résultats aux résultats pour ZeroR.

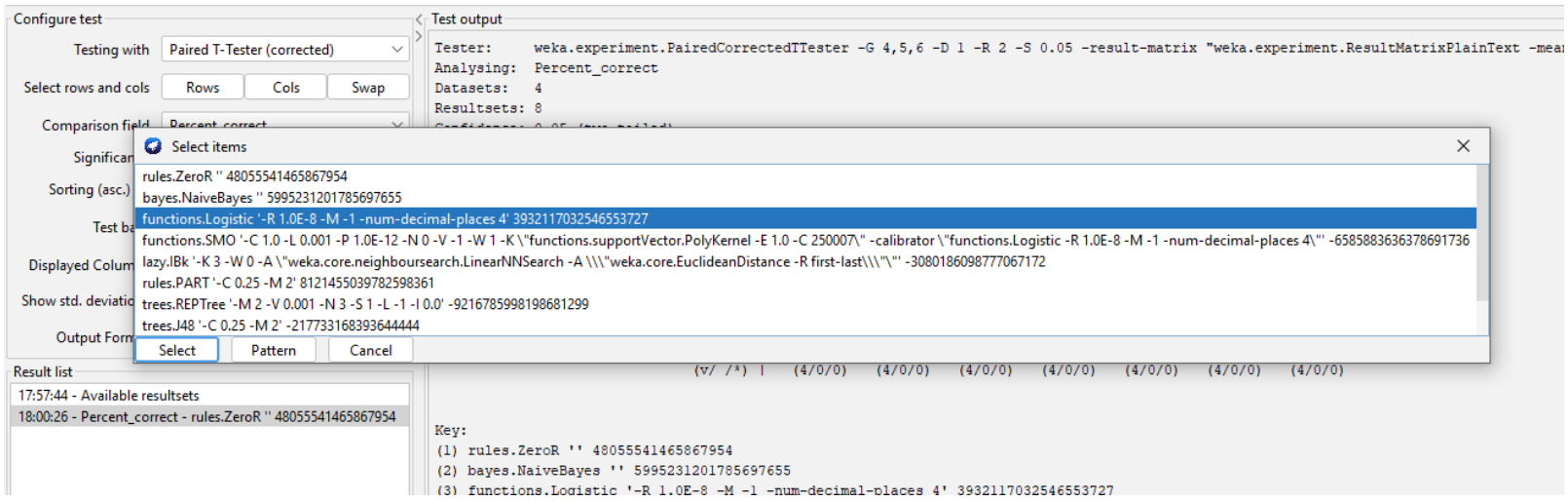
```
Tester:     weka.experiment.PairedCorrectedTTester -G 4,5,6 -D 1 -R 2 -S 0.05 -result-matrix "weka.experiment.ResultMatrixPlainText -
Analysing:  Percent_correct
Datasets:   4
Resultsets: 8
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       12/01/2024 18:00


Dataset                   (1) rules.Ze | (2) bayes (3) funct (4) funct (5) lazy. (6) rules (7) trees (8) trees
-----------------------------------------------------------------------------------------------------------------
pima_diabetes            (100)    65.11 |   75.75 v   77.47 v   76.80 v   73.86 v   73.45 v   74.46 v   74.49 v
pima_diabetes-weka.filter(100)    65.11 |   75.77 v   77.47 v   76.80 v   73.86 v   73.53 v   74.42 v   74.58 v
pima_diabetes-weka.filter(100)    65.11 |   75.65 v   77.47 v   76.81 v   73.86 v   73.49 v   74.39 v   74.52 v
pima_diabetes-weka.filter(100)    65.11 |   75.75 v   77.47 v   76.80 v   73.86 v   73.45 v   74.46 v   74.49 v
-----------------------------------------------------------------------------------------------------------------
                              (v/ /*) |   (4/0/0)   (4/0/0)   (4/0/0)   (4/0/0)   (4/0/0)   (4/0/0)   (4/0/0)
```

Nous pouvons voir que tous les algorithmes sont habiles sur toutes les vues de l'ensemble de données par rapport à Zéro. Nous pouvons également voir que notre base de référence pour la compétence est une précision de 65,11%.

En regardant simplement les précisions de classification brutes, nous pouvons voir que la vue de l'ensemble de données avec les valeurs manquantes imputées semble avoir entraîné une précision du modèle plus faible en général. Il semble également qu'il y ait peu de différence entre les résultats standardisés et normalisés par rapport aux résultats bruts autres que quelques fractions de pourcentage. Cela suggère que nous pouvons probablement nous en tenir à l'ensemble de données brutes.

Enfin, il semble que la régression logistique ait pu obtenir des résultats plus précis que les autres algorithmes, vérifions si la différence est significative.N(4)

On Clique sur le bouton "Sélectionner" pour "Base de test" et choisissez "fonctions".Logistique". puis sur le bouton" Effectuer un test " pour relancer l'analyse.



Résulta de test

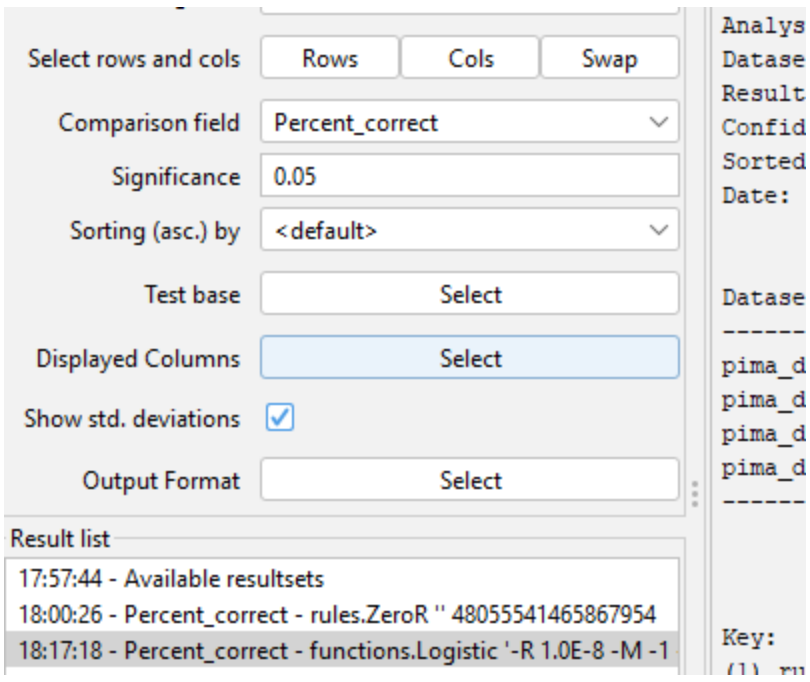
```
Tester:     weka.experiment.PairedCorrectedTTester -G 4,5,6 -D 1 -R 2 -S 0.05 -result-matrix "weka.experiment.ResultMatrixPlainText -mean-prec
Analysing:  Percent_correct
Datasets:   4
Resultsets: 8
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       12/01/2024 18:17


Dataset                   (3) function | (1) rules (2) bayes (4) funct (5) lazy. (6) rules (7) trees (8) trees
-------------------------------------------------------------------------------------------------------------
pima_diabetes            (100)   77.47 |   65.11 *   75.75     76.80     73.86 *   73.45 *   74.46 *   74.49
pima_diabetes-weka.filter(100)   77.47 |   65.11 *   75.77     76.80     73.86 *   73.53 *   74.42 *   74.58
pima_diabetes-weka.filter(100)   77.47 |   65.11 *   75.65     76.81     73.86 *   73.49 *   74.39 *   74.52
pima_diabetes-weka.filter(100)   77.47 |   65.11 *   75.75     76.80     73.86 *   73.45 *   74.46 *   74.49
-------------------------------------------------------------------------------------------------------------
                              (v/ /*) |   (0/0/4)   (0/4/0)   (0/4/0)   (0/0/4)   (0/0/4)   (0/0/4)   (0/4/0)


Key:
(1) rules.ZeroR '' 48055541465867954
(2) bayes.NaiveBayes '' 5995231201785697655
(3) functions.Logistic '-R 1.0E-8 -M -1 -num-decimal-places 4' 3932117032546553727
(4) functions.SMO '-C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K \"functions.supportVector.PolyKernel -E 1.0 -C 250007\" -calibrator \"functio
(5) lazy.IBk '-K 3 -W 0 -A \"weka.core.neighboursearch.LinearNNSearch -A \\\"weka.core.EuclideanDistance -R first-last\\\"\"' -308018609877706
(6) rules.PART '-C 0.25 -M 2' 8121455039782598361
(7) trees.REPTree '-M 2 -V 0.001 -N 3 -S 1 -L -1 -I 0.0' -9216785998198681299
(8) trees.J48 '-C 0.25 -M 2' -217733168393644444
```

Il semble que les résultats de la régression logistique soient meilleurs que certains des autres résultats, tels que IBk, PART, REPTree et ZeroR, mais pas statistiquement significativement différents de NaiveBayes, SMO ou J48.

On Coche " Afficher std. Écarts " pour afficher les écarts types.

On clique sur le bouton "Sélectionner" pour "Colonnes affichées" et choisissez "fonctions». Logistique", puis sur "Sélectionner" pour accepter la sélection. Cela ne montrera que les résultats de l'algorithme de régression logistique.
On Clique sur "Effectuer un test" pour relancer l'analyse.

Nous avons maintenant un résultat final que nous pouvons utiliser pour décrire notre modèle.



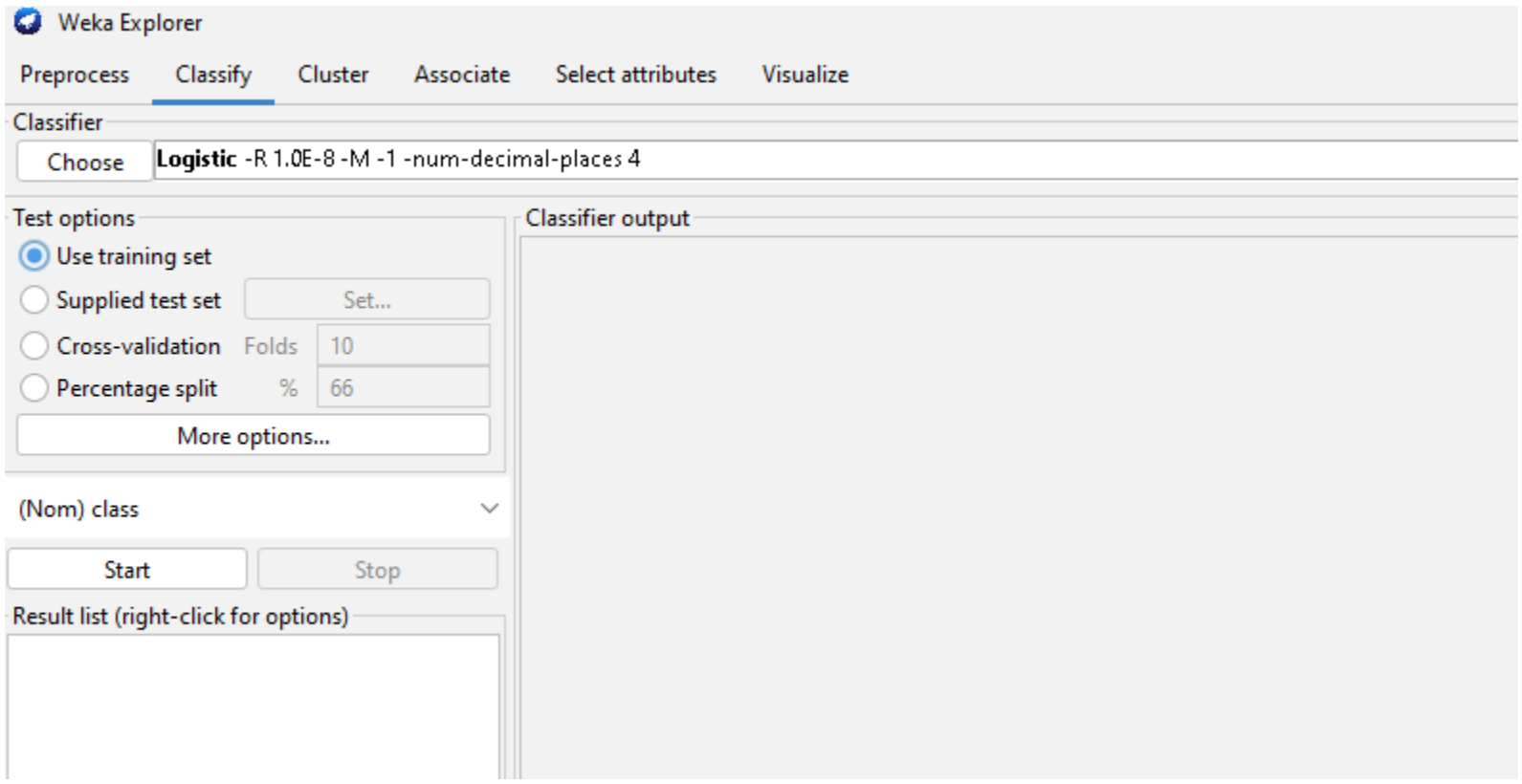

```
Tester:     weka.experiment.PairedCorrectedTTester -G 4,5,6 -D 1 -R 2 -S 0.05 -V -result-matrix "weka.experiment.Resul
Analysing:  Percent_correct
Datasets:   4
Resultsets: 8
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       12/01/2024 18:31


Dataset                   (3) functions.Logist
------------------------------------------------
pima_diabetes           (100)   77.47(4.39) |
pima_diabetes-weka.filter(100)   77.47(4.39) |
pima_diabetes-weka.filter(100)   77.47(4.39) |
pima_diabetes-weka.filter(100)   77.47(4.39) |
------------------------------------------------
(v/ /*) |


Key:
(3) functions.Logistic '-R 1.0E-8 -M -1 -num-decimal-places 4' 3932117032546553727
```

Nous pouvons voir que la précision estimée du modèle sur des données invisibles est de 77,47% avec un écart type de 4,39%.

5. Finaliser le Modèle et Présenter les Résultats

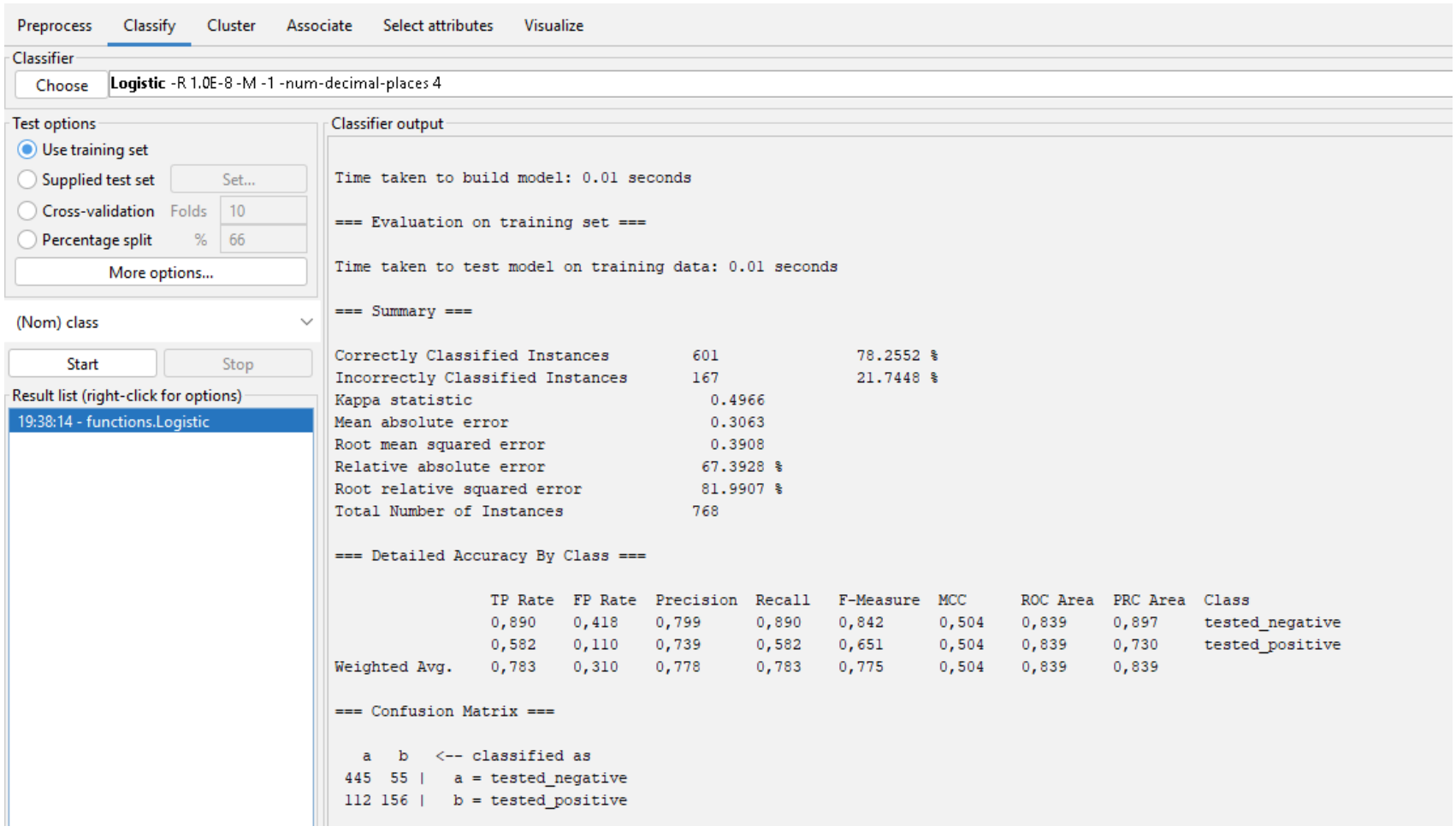
Nous pouvons créer une version finale de notre modèle entraîné sur toutes les données d'entraînement et l'enregistrer dans un fichier.

- On Ouvre l'explorateur Weka et on charge les données / diabète de données arff.
- On Clique sur le Classer.
- On Sélectionne les fonctions.Algorithme logistique.
- On Change les “Options de test” de "Validation croisée" à “Utiliser l'ensemble d'entraînement""
- On Clique sur le bouton” Démarrer " pour créer le modèle final.



Un clic droit sur l'élément de résultat dans la "Liste de résultats" et sélectionnez "Enregistrer le modèle""

Ce modèle peut ensuite être chargé ultérieurement et utilisé pour faire des prédictions sur de nouvelles données.



Conclusion

Concrètement, on appris :

- Comment analyser un ensemble de données et suggérer des techniques spécifiques de transformation et de modélisation des données qui peuvent être utiles.
- Comment concevoir et enregistrer plusieurs vues de vos données et vérifier sur place plusieurs algorithmes sur ces vues.
- Comment finaliser le modèle pour faire des prédictions sur de nouvelles données et présenter la précision estimée du modèle sur des données invisibles.