

Introduction à l'Apprentissage Artificiel

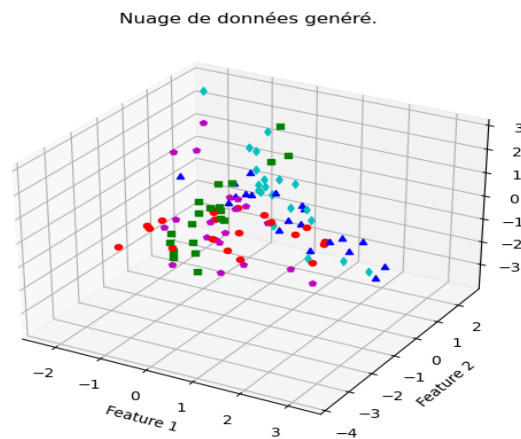
RIAZI Ibrahim

TDM2 : Apprentissage multi-classes

1. Evaluation de l'apprentissage multi-classes

1.1. Génération de jeux de données artificiels

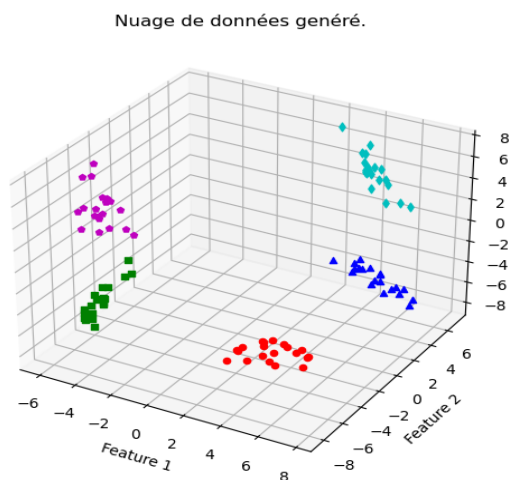
1.1.1.



S1

Pour ajouter de la confusion entre les classes dans le jeu de données **S1**, j'ai diminué la valeur de "**class_sep**" jusqu'à **0,5**. Cela peut rendre les classes plus proches les unes des autres et plus difficiles à séparer.

On constate dans la visualisation du jeu de données **S1** que **S1** a une certaine confusion entre les classes.

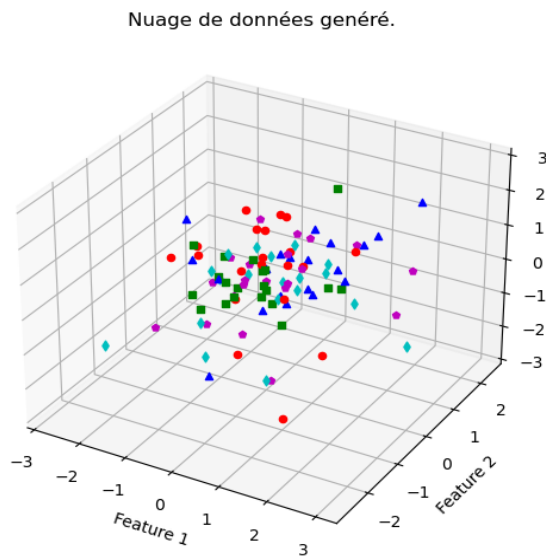


S2

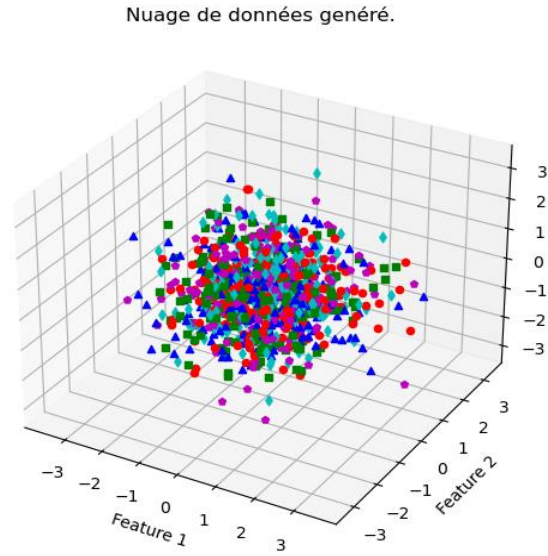
Introduction à l'Apprentissage Artificiel

Pour enlever la confusion entre les classes dans le jeu de données **S2**, j'ai augmenté la valeur de "**class_sep**" jusqu'à **6**. Cela peut rendre les classes bien séparées les unes des autres.

On constate dans la visualisation du jeu de données **S2** que les classes de **S2** bien séparées.



S3



S4

1.1.2.

- Le jeu de données S2 devrait être le plus facile à classer, car il n'y a pas de confusion entre les classes, toutes les caractéristiques sont informatives et il n'y a pas de caractéristiques redondantes.
- Le jeu de données S1 devrait être plus difficile à classer que S2, car il y a de la confusion entre les classes.
- Les jeux de données S3 et S4 sont plus complexes que S1 et S2 en raison du nombre plus élevé de dimensions. Cependant, le jeu de données S4 avec 1000 exemples peut potentiellement fournir plus de données pour apprendre un classifieur précis, ce qui pourrait le rendre plus facile à classer que S3 avec seulement 100 exemples.

Donc l'ordre relatif des performances des classifieurs qui seraient appris sur ces jeux de données est probablement : **S2** (le plus facile) > **S1** > **S4** > **S3**

1.2. Expérimentations du one versus all

1.2.1. Score moyenné (10_hold-out)

10 hold-out

KNeighborsClassifier

S1 : 0.36 +/- 0.0796155763654324

S2 : 1.0 +/- 0.0

S3 : 0.32 +/- 0.08787272614412277

S4 : 0.392 +/- 0.02416450289163838

Introduction à l'Apprentissage Artificiel

DecisionTreeClassifier

S1 : 0.43 +/- 0.0786446183791364

S2 : 0.9400000000000001 +/- 0.04111330684826993

S3 : 0.33999999999999997 +/- 0.09109256830279844

S4 : 0.5239999999999999 +/- 0.022595323409944808

Le classifieur **KNeighborsClassifier** a une performance relativement faible sur S1 et S3 avec des scores de 0.36 et 0.32 respectivement, tandis que le classifieur **DecisionTreeClassifier** a des scores plus élevés pour S1 et S3 avec des scores de 0.43 et 0.34 respectivement. Le classifieur **KNeighborsClassifier** a une performance parfaite sur S2 avec un score de 1.0 tandis que le classifieur **DecisionTreeClassifier** a une performance un peu inférieure sur S2 avec un score de 0.94. Pour S4, les performances des deux classifieurs sont plus proches avec un score de 0.39 pour le classifieur **KNeighborsClassifier** et 0.52 pour le classifieur **DecisionTreeClassifier**.

1.2.2. La matrice de confusion moyenne sans diagonale

CMNwD

KNeighborsClassifier

S1 : 0.3083333333333334 +/- 0.05431907788449858

S2 : 0.0 +/- 0.0

S3 : 0.265 +/- 0.04039322825535103

S4 : 2.0533333333333337 +/- 0.0961088641998113

DecisionTreeClassifier

S1 : 0.2833333333333334 +/- 0.0550508451847522

S2 : 0.025 +/- 0.020788939366884498

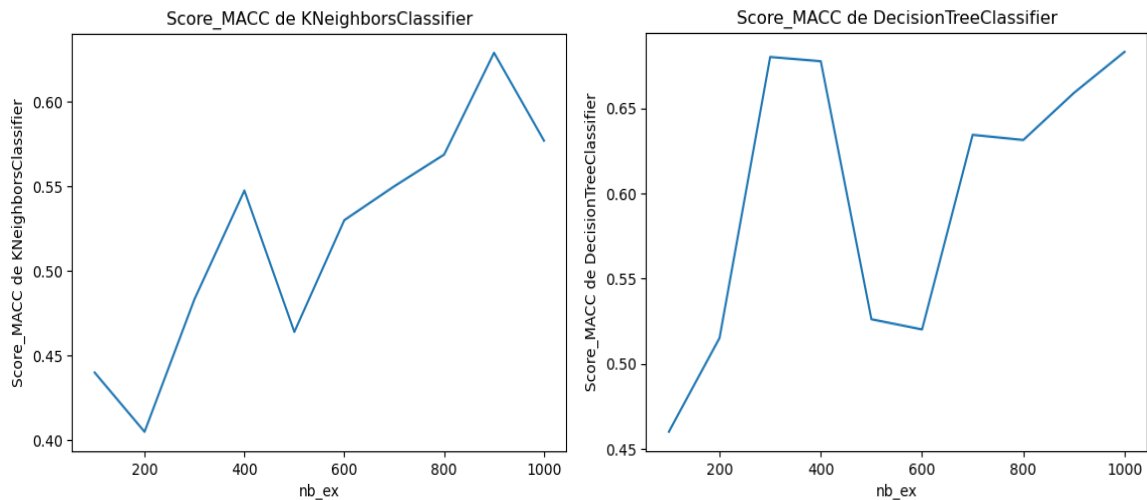
S3 : 0.31833333333333336 +/- 0.053167446180283406

S4 : 1.7416666666666667 +/- 0.11514710398249516

Le classifieur KNeighborsClassifier a des performances similaires pour S1, S3 et S4 avec une CMNwD élevée, ce qui suggère qu'il y a une confusion importante entre les classes. Le classifieur KNeighborsClassifier a une performance parfaite sur S2, ce qui explique une CMNwD de 0 pour S2. Le classifieur DecisionTreeClassifier a des performances similaires pour S1, S3 et S4, avec une CMNwD plus faible que le classifieur KNeighborsClassifier. En revanche, pour S2, le classifieur DecisionTreeClassifier a une CMNwD non nulle de 0.025, ce qui suggère une certaine confusion entre les classes.

1.2.3.

Introduction à l'Apprentissage Artificiel



D'après les graphes, on peut voir que :

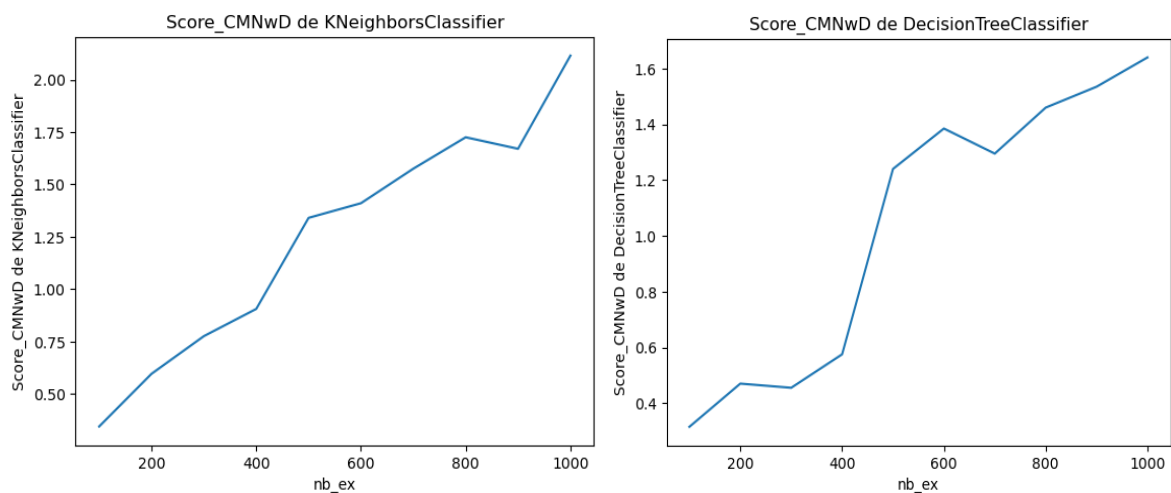
Pour l'**arbre de décision**, le nombre d'exemples où les performances sont les meilleures est **1000**. Pour le modèle **k-NN**, les performances sont les meilleures pour le nombre d'exemples **900**.

Pour le nombre d'exemples **900** le meilleur modèle est l'**arbre de décision**.

KNeighborsClassifier - nb_ex = 900 : 0.6288888888888889 +/- 0.019527272476471732

DecisionTreeClassifier - nb_ex = 900 : 0.6644444444444444 +/- 0.023616591782206993

1.2.4.



D'après les graphes, on constate que les performances des deux classifieurs diminuent lorsque le nombre d'exemples augmente,

Pour le nombre d'exemples **900** le meilleur modèle est l'**arbre de décision**.

KNeighborsClassifier - nb_ex = 900 : 1.67 +/- 0.08787272614412277

Introduction à l'Apprentissage Artificiel

DecisionTreeClassifier - nb_ex = 900 : 1.5900000000000003 +/- 0.07437677056716029

2. Perceptron multi-classes

2.1.

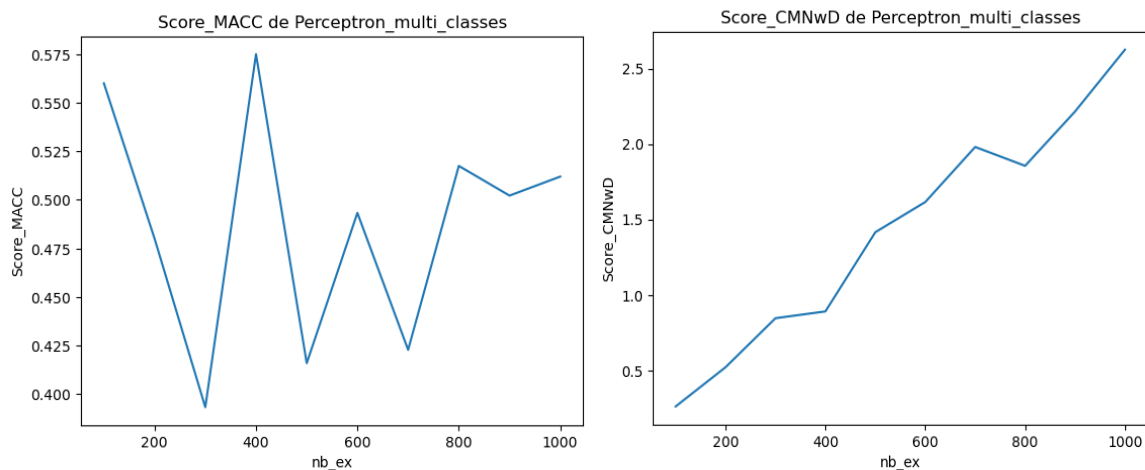
2.2.

2.3.

S1 ==> Score moyen = 0.28 CMNwD = 0.4083333333333333 Temps d'apprentissage moyen = 0.7193820476531982s

S2 ==> Score moyen = 1.0 CMNwD = 0.0 Temps d'apprentissage moyen = 0.47391324043273925s

2.4.



Le graphique montre que le score et la **CMNwD** augmentent avec le nombre d'exemples, mais il n'y a pas de corrélation linéaire claire entre les deux. Le score peut augmenter rapidement au début avec peu d'exemples, mais il semble atteindre un plateau pour un grand nombre d'exemples, tandis que la **CMNwD** continue d'augmenter. Cela peut être dû à une meilleure généralisation du modèle avec plus d'exemples, mais la **CMNwD** mesure également la qualité de la séparation des classes, ce qui peut expliquer pourquoi elle continue d'augmenter même lorsque le score ne le fait pas.

2.5.

Rapport de classification :

	precision	recall	f1-score	support
0	1.00	0.96	0.98	23
1	1.00	0.94	0.97	18
2	1.00	1.00	1.00	18
3	1.00	0.95	0.97	19

Introduction à l'Apprentissage Artificiel

4	0.95	0.95	0.95	19
5	0.95	0.95	0.95	21
6	0.95	1.00	0.98	21
7	1.00	0.94	0.97	18
8	0.83	0.91	0.87	11
9	0.86	1.00	0.92	12
accuracy			0.96	180
macro avg	0.95	0.96	0.96	180
weighted avg	0.96	0.96	0.96	180