Master TRIED,

prepaTP-30-10 : Rapport TP EX22

Sujet:

Algorithme des k plus proches voisins (kppv)

Réalisé par:

FOUTSE YUEHGOH

Année universitaire : 2018/2019

1°) Expliquer le principe de fonctionnement, la nature et les différentes étapes d'un algorithme des k plus proches voisins, puis retrouver les différentes étapes spécifiquement associées à cet algorithme dans le programme kppvo.m fourni.

L'algorithme des k plus proches voisins est une méthode d'apprentissage supervisé qui peut être utilise aussi bien pour la régression que pour la classification. Elle ne nécessite pas d'apprentissage mais simplement le stockage des données d'apprentissage. Une donnée de classe inconnue est comparée à toutes les données stockées. On choisit pour la nouvelle donnée la classe majoritaire parmi ses K plus proches voisins (Elle peut donc être lourde pour des grandes bases de données) au sens d'une distance choisie.

Principe : Méthode très intuitive qui classe les exemples non étiquetés sur la base de leur similarité avec les exemples de la base d'apprentissage (données étiquetées) :

Pour un exemple non étiqueté x, trouver les k "plus proches exemples étiquetés de la base d'apprentissage et affecter à x la classe qui apparaît le plus souvent.

Les k-PPV nécessitent seulement :

- Un entier k
- Une base d'apprentissage
- Une métrique pour la proximité

On peut schématiser le fonctionnement de K-NN en l'écrivant en pseudo-code suivant :

Début Algorithme
Données en entrée :

- un ensemble de données. D
- une fonction de définition distance d.
- Un nombre entier K

Pour une nouvelle observation X dont on veut prédire sa variable de sortie Faire :

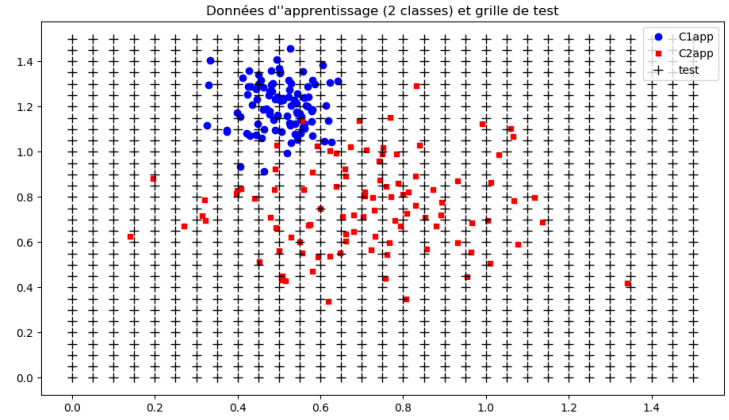
1. Calculer toutes les distances de cette observation X avec les autres observations du jeu de données D
2. Retenir les K observations du jeu de données D les proches de en utilisation la fonction de calcul de distance d
3. Prendre les valeurs de y des K observations retenues :
    1. Si on effectue une régression, calculer la moyenne (ou la médiane) de y retenues
    2. Si on effectue une classification, calculer le mode de y retenues
4. Retourner la valeur calculée dans l'étape 3 comme étant la valeur qui a été prédite par K-NN pour l'observation X.

Fin Algorithme

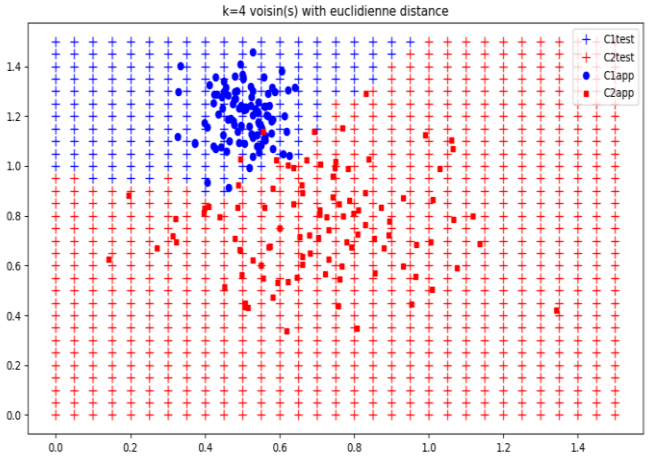
2°) Soit un ensemble de données d'apprentissage en dimension 2 labellisées en 2 classes. Les données sont contenues dans le fichier DataA.txt et les étiquettes dans le fichier labelA.txt. Cet ensemble d'apprentissage comprend 100 exemples pour chacune des classes, soit 200 au

total.
Le programme demokppvEuMa.m utilise ces données pour faire ressortir, à l'aide d'un ensemble de test sur un maillage 2D, les frontières de décision obtenues par l'algorithme kppv. On vous demande d'exécuter ce programme, et de rendre compte des résultats.

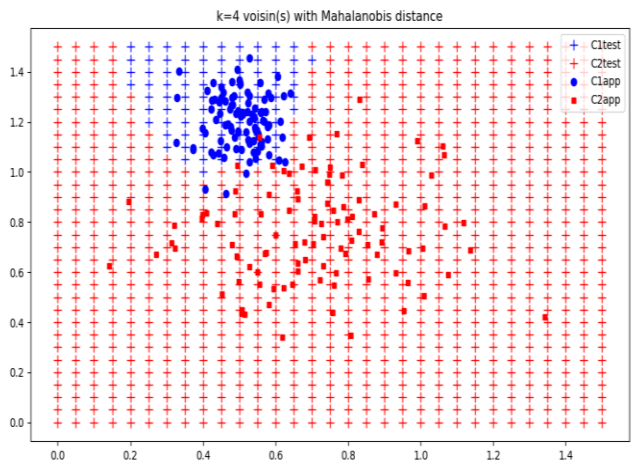
Les résultats du programme demokppvEuMa sont comme suit :



Sur cette figure on fait une présentation des données, nous avons un nuage de point des données d'apprentissage représentent deux classes et les données de test qui n'ont pas été encore différencié. On remarque que les données de la classe rouge sont beaucoup plus dispersées que ceux de la classe bleu C1app.

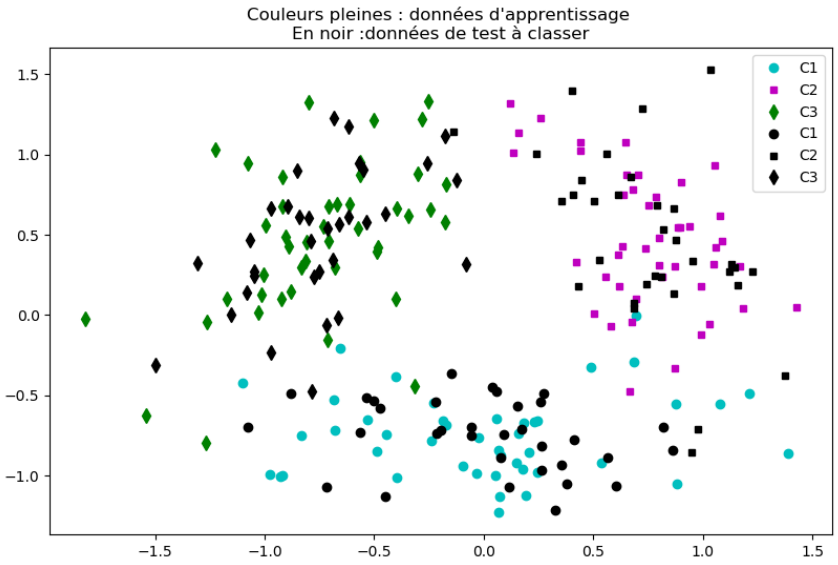


Sur cette figure, nous avons les données d'apprentissage et les champs de décisions d'appartenance à une classe. Ici le classifier a été appliqué et on a une frontière entre les deux classes qui est bien déterminée. La probabilité qu'une donnée soit classée dans la classe rouge est plus grande que la probabilité qu'elle soit dans le bleu. On constate que le classifier ne reflète pas les données d'apprentissage.



Sur cette figure, nous avons les données d'apprentissage et les champs de décisions d'appartenance à une classe. Ici le classifier a été appliqué et on a une frontière entre les deux classes qui est bien déterminée. La probabilité qu'une donnée soit classée dans la classe rouge est plus grande que la probabilité qu'elle soit dans le bleu. On constate que le classifier reflète beaucoup mieux les données d'apprentissage. Mais aussi des éléments des bleus se retrouvent dans la classe des rouges et la classe des bleus est beaucoup plus petite. Le classifier prend la forme des données bleues. On remarque là le caractère de la distance de Mahalanobis qui prend en compte la forme des données.

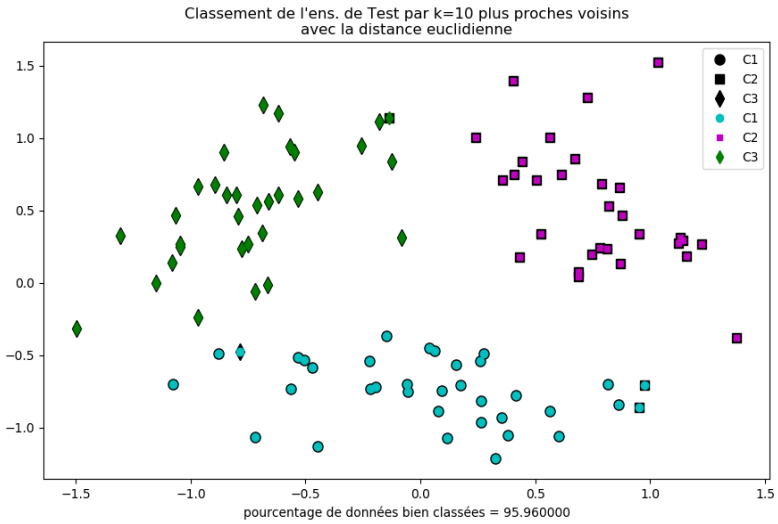
3°) On dispose maintenant d'un ensemble de 132 données d'apprentissage en dimension 2, labellisées en 3 classes d'égale taille. Les données sont contenues dans le fichier Data1.mat et les étiquettes dans le fichier Label1.mat. Avec ces données d'apprentissage, le programme demokppvDatas.m classe les 99 données contenues dans Data1test.mat avec algorithme du plus proche voisin (code kppvo.m). Les données à classer ont été étiquetées par un expert, les labels correspondants sont contenus dans Label1test.mat. Les labels sont utilisés pour calculer la matrice de confusion (MCO). On présente ces données dans la figure ci-dessous.



Nous avons là les données d'apprentissage en couleurs et les données de test à classer en noir contenues dans **Data1.mat** et **Label1.mat ; Data1test.mat et Label1test.mat**

Pour chaqu'une des distances, on ajoute après chaque classification, le calcul du pourcentage de données bien classées. Dans les résultats obtenus pour différentes valeurs de k (1, 2, 3, 4, 5, 10, 15, 20) selon la distance utilisée sont représenter dans les figures si dessous, les classifications obtenues pour le meilleur et le moins bon des résultats ainsi que la matrice de confusion.

- Pour la distance euclidienne nous avons comme meilleur et moins bon résultats les suivant :

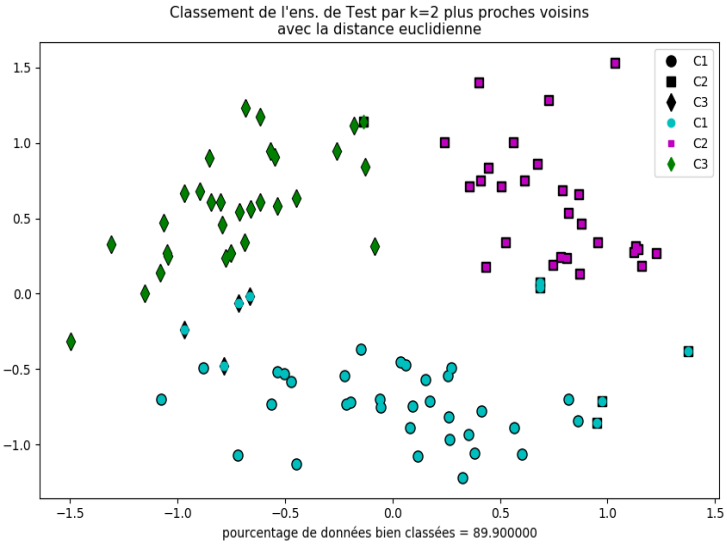


K=10

Actual\Pred	1	2	3
1	33	0	0
2	2	30	1
3	1	0	32

Accuracy : 95.96%

Dans la figure on voie bien 2 éléments de la **classe 2** et 1 élément de la **classe 3** qui sont mal classé dans **classe 1** ; et 1 élément de la **classe 2** qui est mal classé dans la **classe 3**. Nous avons également se même % pour k = 5, 15 et 20



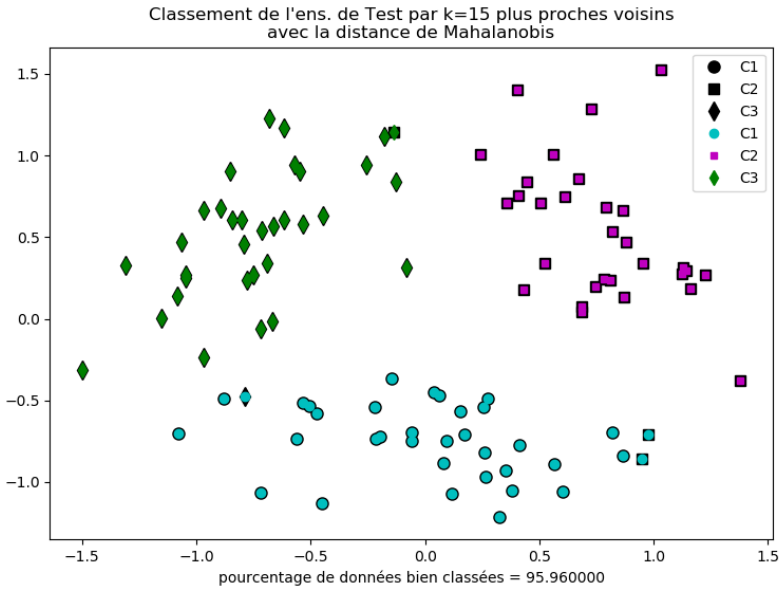
K=2

Actual\Pred	1	2	3
1	33	0	0
2	5	27	1
3	4	0	29

Accuracy : 89.90

Dans cette figure, on voie 4 éléments de **classe 3** et 5 éléments de **classe 2** qui sont mal classé dans la **classe 1** ; 1 élément de **classe 2** qui est mal classé dans la **classe 3.**

- Pour la distance de Mahalanobis, nous avons comme meilleur et moins bon résultats les suivant :

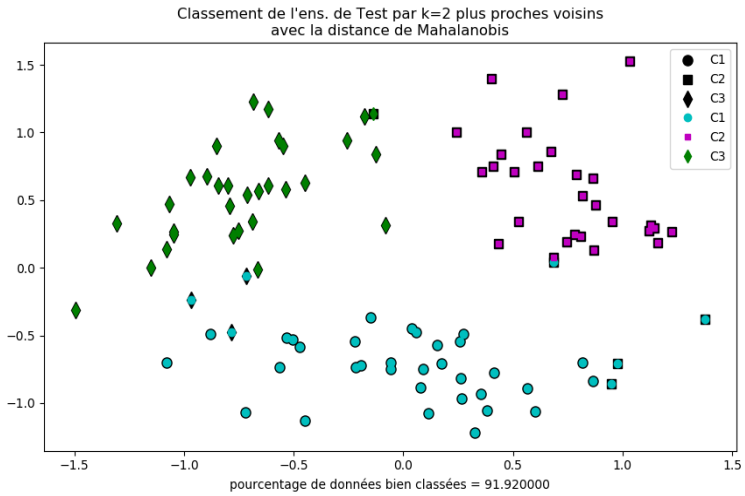


K=15

Actual\Pred	1	2	3
1	33	0	0
2	2	30	1
3	1	0	32

Accuracy : 95.96%

Dans la figure on voie bien 2 éléments de la **classe 2** et 1 élément de la **classe 3** qui sont mal classé dans **classe 1** ; et 1 élément de la **classe 2** qui est mal classé dans la **classe 3**. Nous avons également se même % pour k = 20



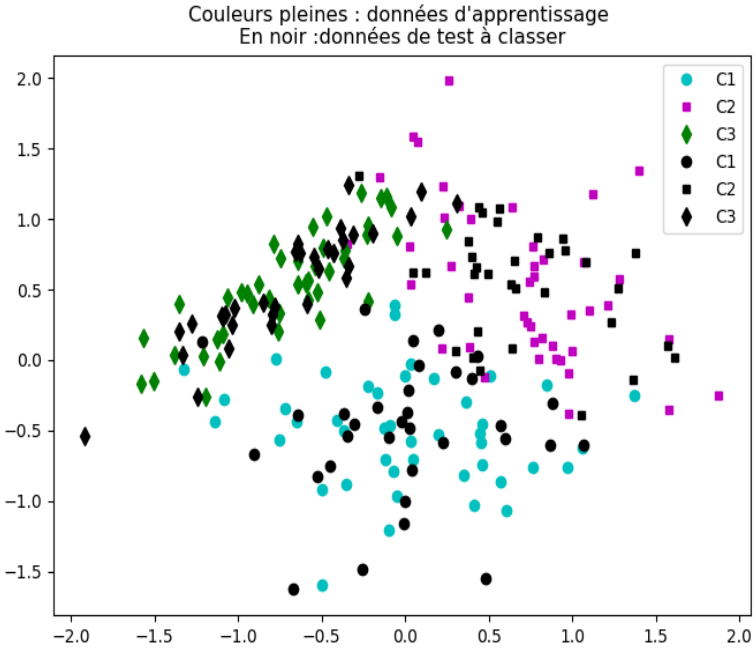
K=2

Actual\Pred	1	2	3
1	33	0	0
2	4	28	1
3	3	0	30

Accuracy : 91.92%

Dans cette figure, on voie 3 éléments de **classe 3** et 4 éléments de **classe 2** qui sont mal classé dans la **classe 1** ; 1 élément de **classe 2** qui est mal classé dans la **classe 3.**

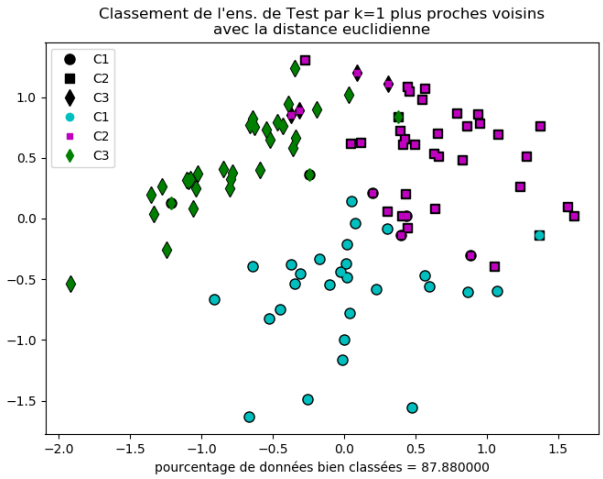
Q) On nous demande de refaire l'étude avec les fichiers de données suivants : Data2.mat, Label2.mat, Data2test.mat, Label2test.mat.

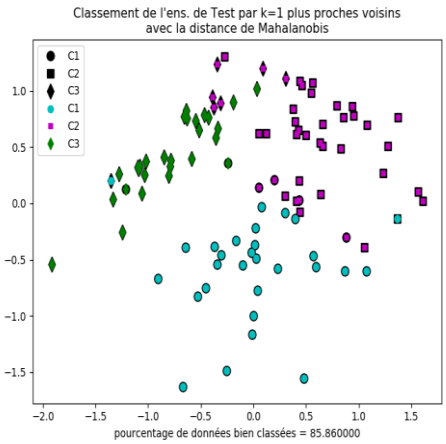


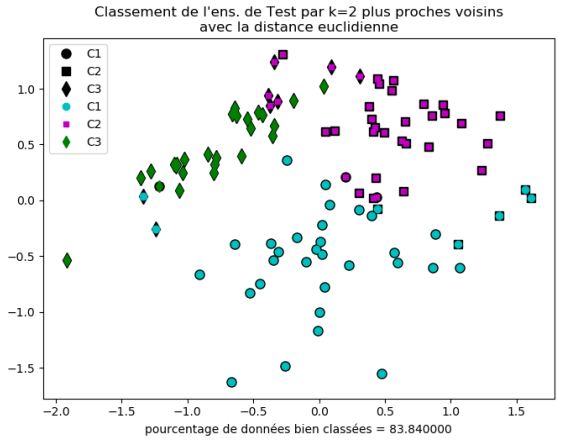
Dans ce graphique, nous avons les données d'apprentissage et les données de test à classer en noir qui nous a été donné pour cet exercice : **Data2.mat, Label2.mat, Data2test.mat, Label2test.mat**

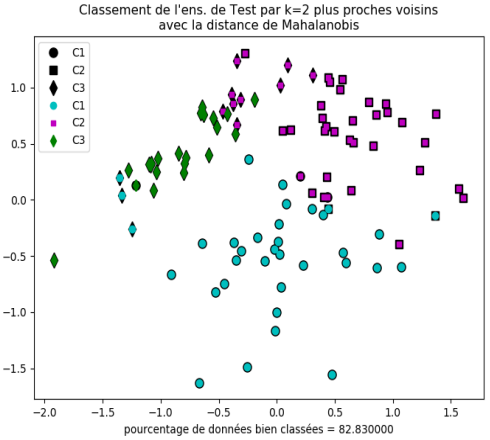
Nous observons la dispersion de ces données et nous allons effectuer ce qui nous a été demandé de faire.

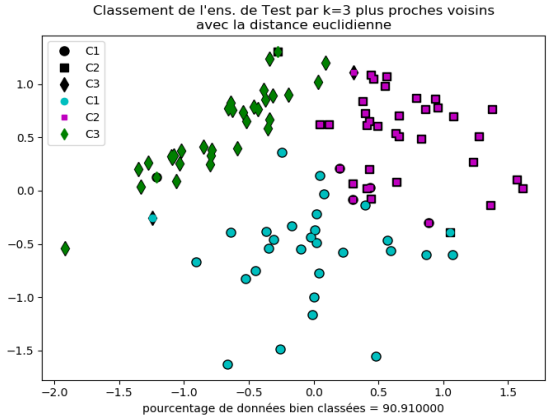
Pour chaqu'une des distances, on ajoute après chaque classification, le calcul du pourcentage de données bien classées. Les résultats obtenus pour différentes valeurs de k (1, 2, 3, 4, 5, 10, 15, 20) selon la distance utilisée sont représenter dans les figures si dessous.

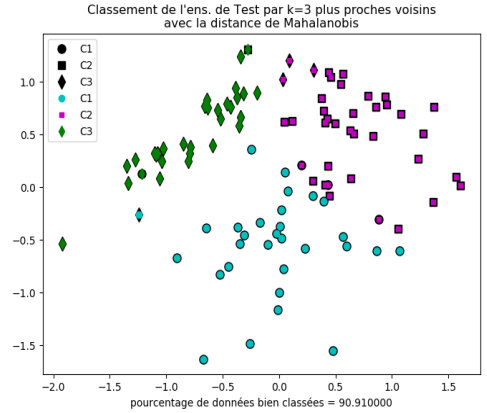


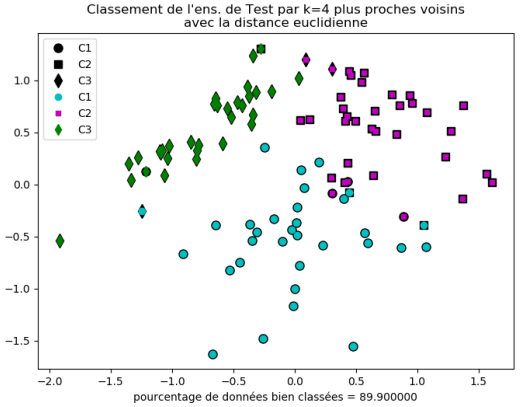


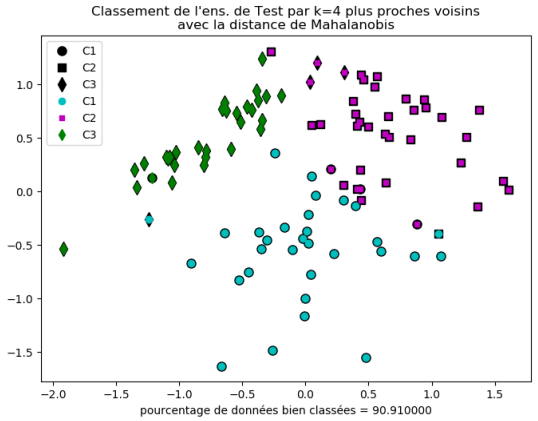

Classement de l'ens. de Test par k=2 plus proches voisins
avec la distance euclidienne
C1
C2
C3
C1
C2
C3
pourcentage de données bien classées = 83.840000

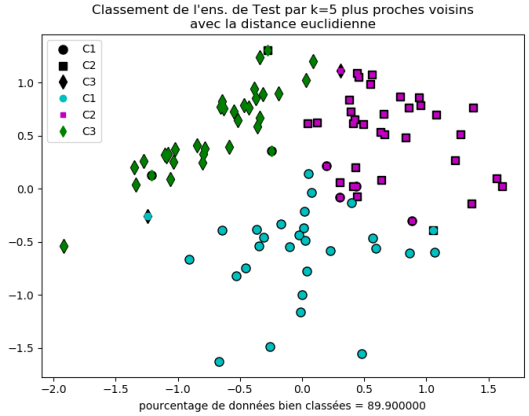

Classement de l'ens. de Test par k=2 plus proches voisins
avec la distance de Mahalanobis
C1
C2
C3
C1
C2
C3
pourcentage de données bien classées = 82.830000

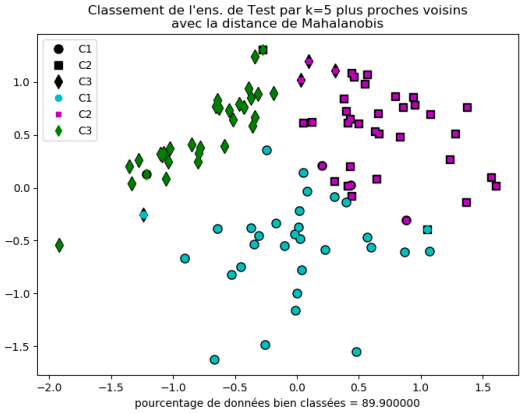

Classement de l'ens. de Test par k=3 plus proches voisins
avec la distance euclidienne
C1
C2
C3
C1
C2
C3
pourcentage de données bien classées = 90.910000

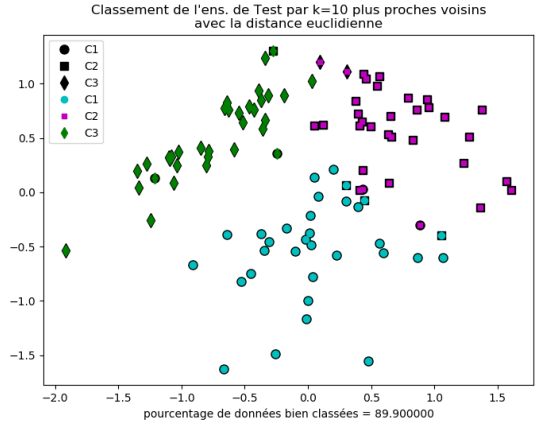

Classement de l'ens. de Test par k=3 plus proches voisins
avec la distance de Mahalanobis
C1
C2
C3
C1
C2
C3
pourcentage de données bien classées = 90.910000

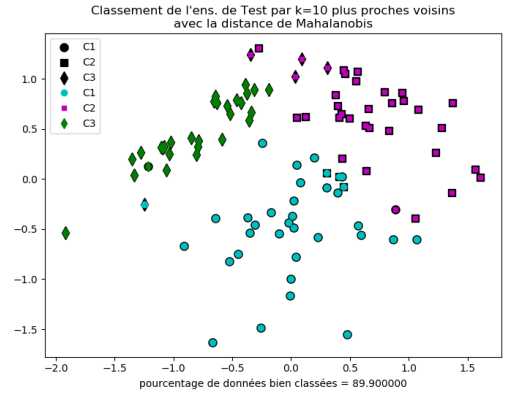

Classement de l'ens. de Test par k=4 plus proches voisins
avec la distance euclidienne
C1
C2
C3
C1
C2
C3
pourcentage de données bien classées = 89.900000

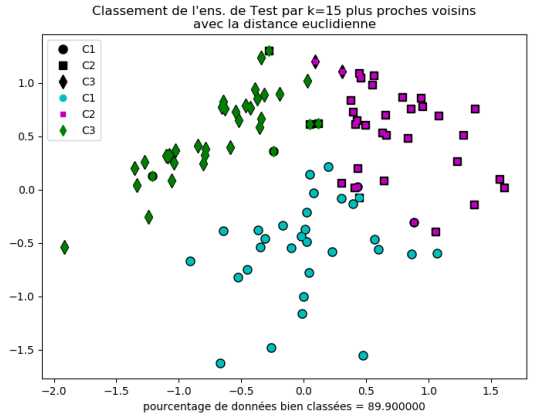

Classement de l'ens. de Test par k=4 plus proches voisins
avec la distance de Mahalanobis
C1
C2
C3
C1
C2
C3
pourcentage de données bien classées = 90.910000

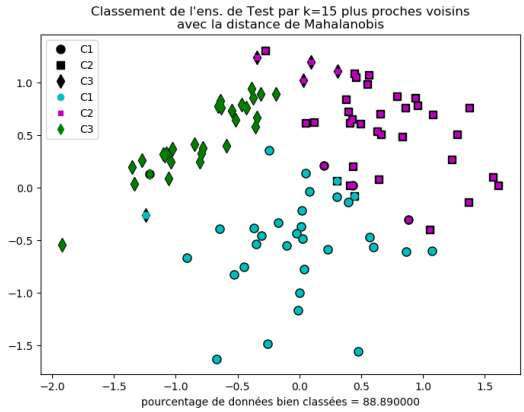

Classement de l'ens. de Test par k=5 plus proches voisins
avec la distance euclidienne
C1
C2
C3
C1
C2
C3
pourcentage de données bien classées = 89.900000

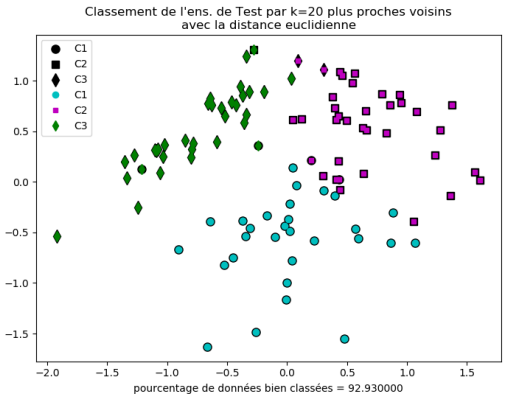

Classement de l'ens. de Test par k=5 plus proches voisins
avec la distance de Mahalanobis
C1
C2
C3
C1
C2
C3
pourcentage de données bien classées = 89.900000

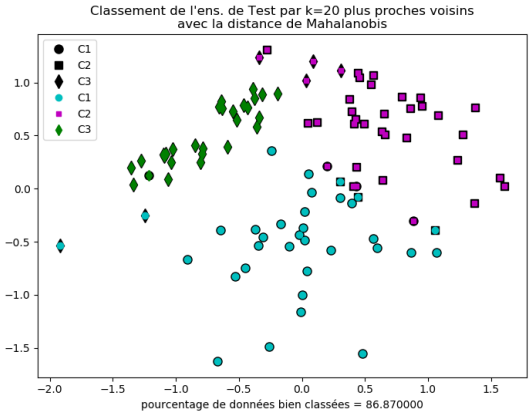

Classement de l'ens. de Test par k=10 plus proches voisins
avec la distance euclidienne
C1
C2
C3
C1
C2
C3
pourcentage de données bien classées = 89.900000


Classement de l'ens. de Test par k=10 plus proches voisins
avec la distance de Mahalanobis
C1
C2
C3
C1
C2
C3
pourcentage de données bien classées = 89.900000


Classement de l'ens. de Test par k=15 plus proches voisins
avec la distance euclidienne
C1
C2
C3
C1
C2
C3
pourcentage de données bien classées = 89.900000


Classement de l'ens. de Test par k=15 plus proches voisins
avec la distance de Mahalanobis
C1
C2
C3
C1
C2
C3
pourcentage de données bien classées = 88.890000





Q) Pour chacune des distances, on nous demande de reproduire les figures de classification obtenues pour le meilleur et le moins bon des résultats ainsi que la matrice de confusion.

- De ces figures on remarque que : la meilleure classification obtenue **pour la distance euclidienne** est celle obtenu avec k = 20 et la moins bonne est pour k = 2 avec les matrices de confusion suivante :

K=20

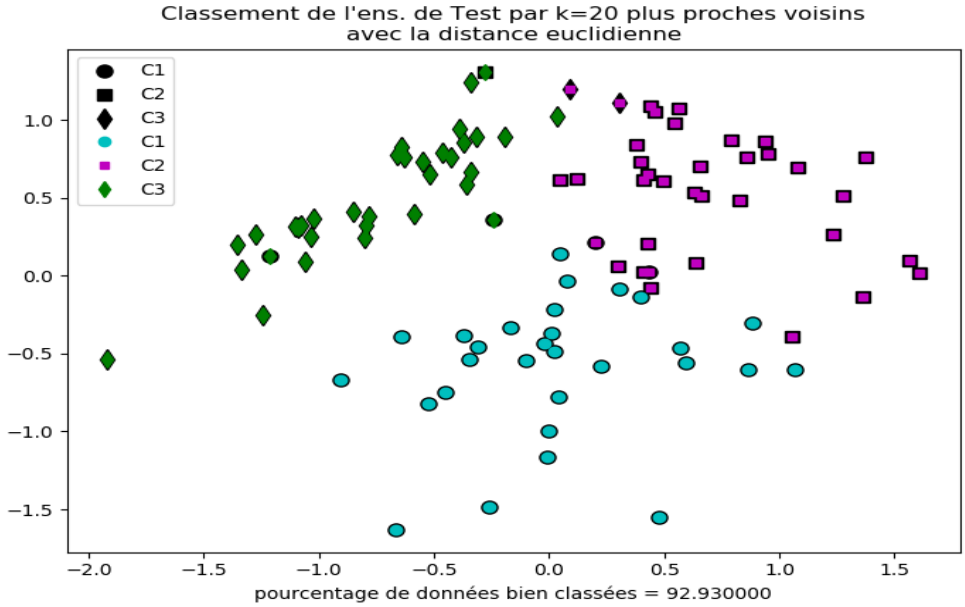
Actual\Pred	1	2	3
1	29	2	2
2	0	32	1
3	0	2	31

Accuracy : 92.93

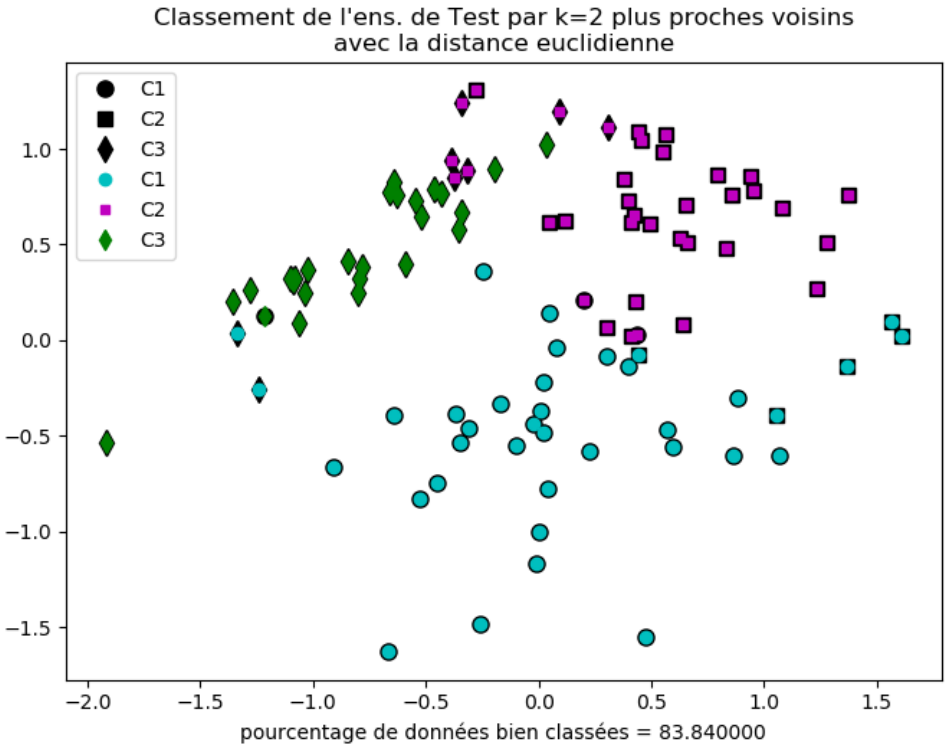
K=2

Actual\Pred	1	2	3
1	30	2	1
2	5	28	0
3	2	6	25

Accuracy : 83.84



On vois bien que on a 1 element de la **classe 2** et 2 elements de la **classe 1** qui se trouvent dans la **classe 3 ;** 2 element de **classe 3 et** 2 element de **classe 1** se trouve dans la **classe 2.**



On vois bien que on a 1 elements de la **classe 1** qui se trouvent dans la **classe 3 ;** 6 element de **classe 3 et** 2 element de **classe 1** se trouve dans la **classe 2 ;** 5 elements de **classe 2** et 2 elements de **classe 3** se trouve dans la **classe 1.**

- La meilleure classification obtenue **pour la distance de Mahalanobis** est celle obtenu avec k = 3 ou 4 et la moins bonne est pour k = 2 avec les matrices de confusion suivante :

K=3

Actual\Pred	**1**	**2**	**3**
1	29	3	1
2	0	32	1
3	1	3	29

Accuracy : 90.91

K=2

Actual\Pred	**1**	**2**	**3**
1	30	2	1
2	2	31	0
3	3	9	21

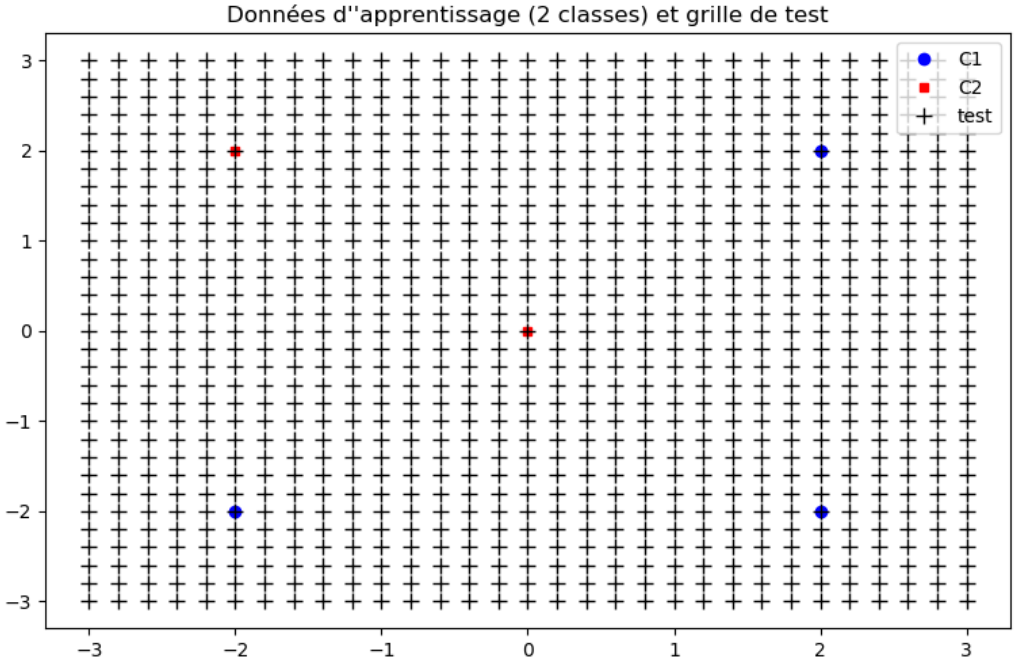
Accuracy : 82.83

Les informations de ces matrices de confusion, sont également observer sur les figures correspondantes à ces valeurs de k, dans les graphiques donnés plus haut.

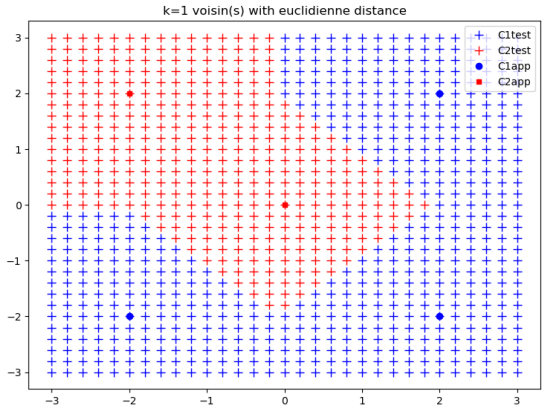
4°) On considère un ensemble de 5 points :

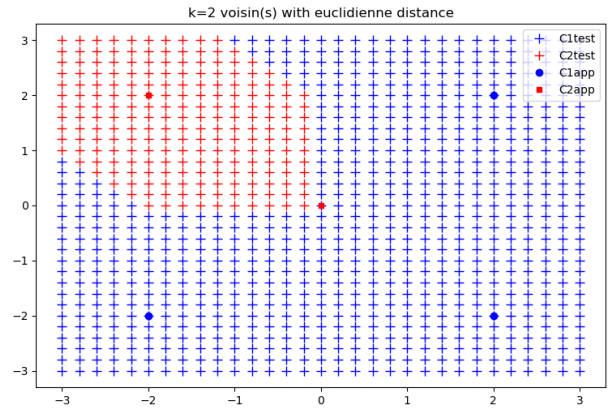
x1=(2,2), x2 = (2,-2), x3 = (-2,-2)	**qui sont de classe 1 et**
x4=(-2,2), x5=(0,0)	**qui sont de classe 2.**

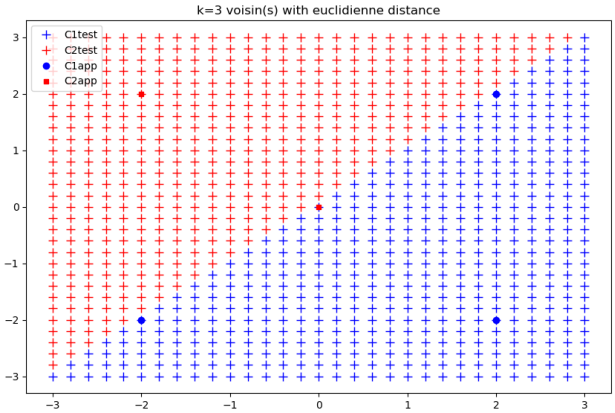
On nous demande d'écrire un script qui permet de représenter graphiquement les frontières de décision, en utilisant la distance euclidienne. Pour se faire, nous avons définir un grillage 2D. Ce grillage est défini en abscisse et en ordonnés de -3 à 3 par pas de 0.2. Les points de ce maillage constituent l'ensemble de test qui sera utilisé pour faire apparaitre les frontières de décision. La figure si dessous présente les données d'apprentissage et le grill de test.

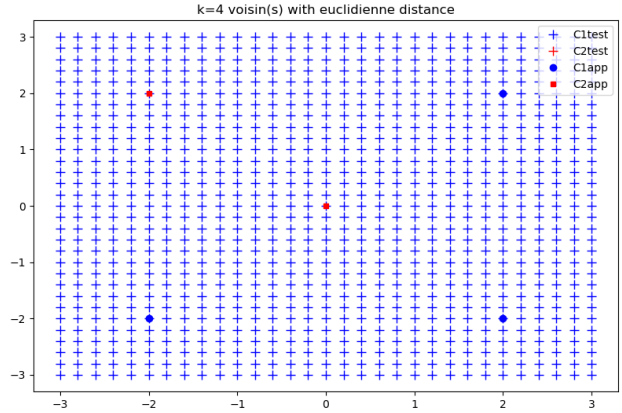


On fait apparaître graphiquement les frontières de décision en même temps que les 5 points d'apprentissage pour k = 1, 2, 3, 4. Nous avons les graphiques suivant :



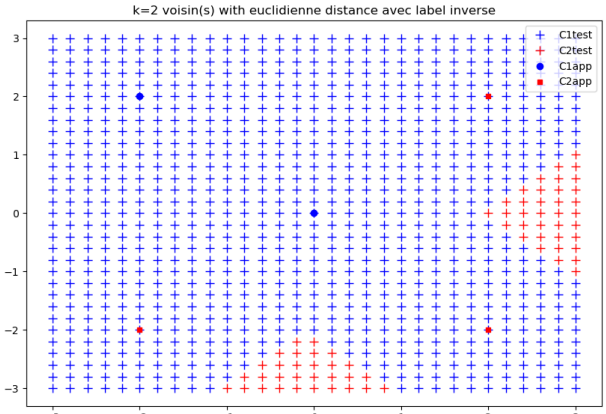


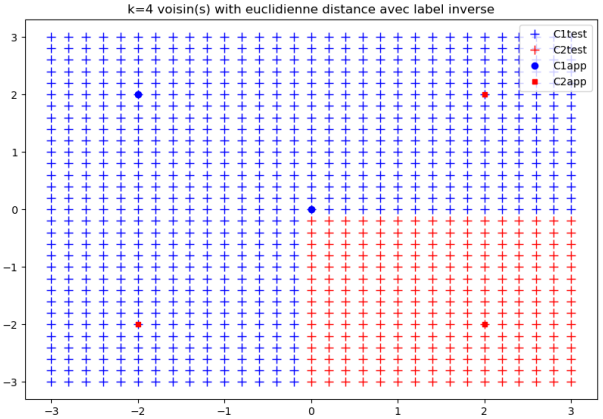




Que se passe-t-il si l'on échange les noms des classes 1 et 2:
classe 2 : x1 = (2,2), x2 = (2,-2), x3 = (-2,-2).
classe 1 : x4 = (-2,2), x5 = (0,0).

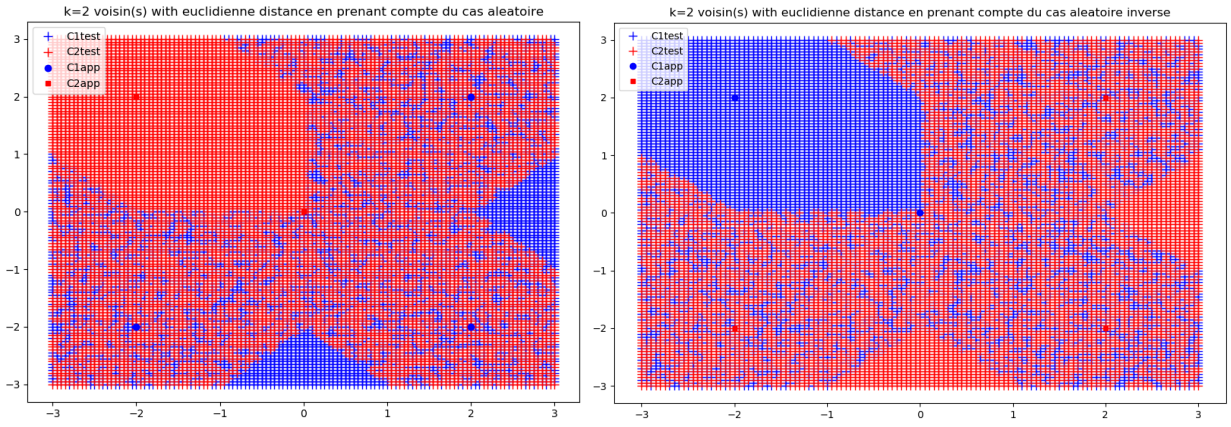
On présente si dessous les frontières de décision qui sont différentes des cas précédents.





On remarque une difference de classement quand on echange les classes pour les valuers de k qui sont pair. Donc l'algorithme vary les figures en fonction du label dans les cas ou on a égalité de vote.

On recommence maintenant en choisissant d'une manière aléatoire la classe à laquelle est affecté un point de la grille en cas d'égalité du vote. Pour montre l'effet de notre intervention, on définit un maillage plus resserré (-3:0.05:3), et on est demandé de montrer la figure de la frontière de décision pour k = 2 voisins uniquement et commentez.



On remarque ici une même forme de figure avec juste une permutation de coleurs qui correspond au fait que on a permuté les labels. Donc l'algorithme ne vary pas les figures en fonction du label dans les cas ou elle choisit d'une manière aléatoire la classe à laquelle est affecté un point de la grille en cas d'égalité du vote.