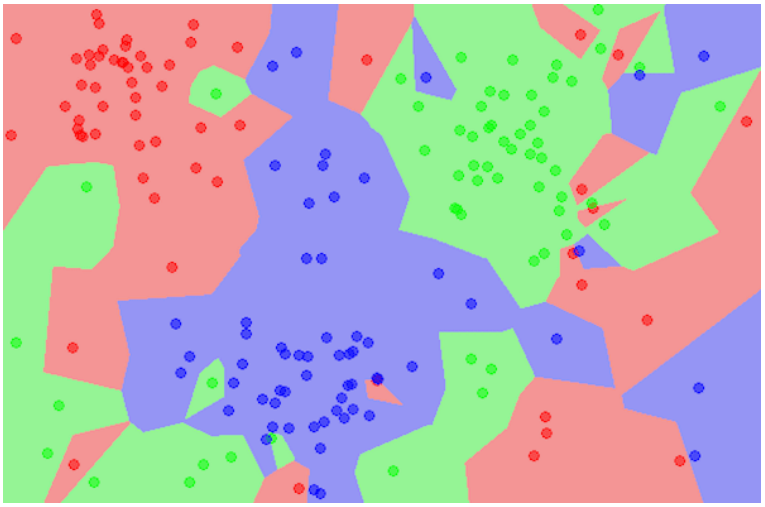
machine learning : KNN

Ben Amira Rawia

11 décembre 2022



Université de Toulon, UFR Sciences et Techniques

M1 DID

2022-2023

Table des matières

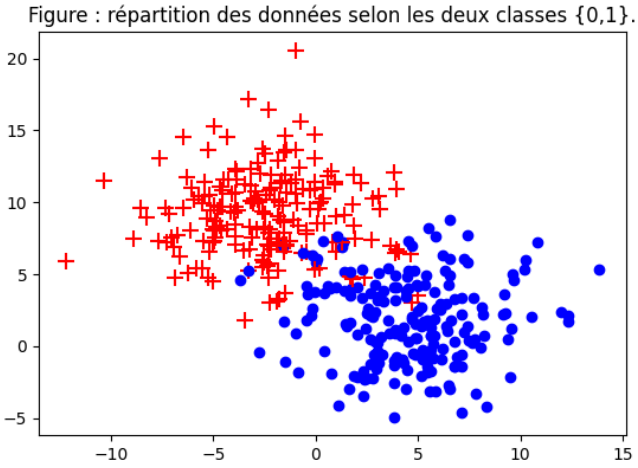
1 Introduction

Ce TP est elaboré dans le module base de l'apprentissage et qui traite l'algorithme KNN. L'algorithme des K plus proches voisins ou K-nearest neighbors (kNN) est un algorithme de Machine Learning qui appartient à la classe des algorithmes d'apprentissage qui peut être utilisé pour résoudre les problèmes de classification et de régression.

2 Exercice 1

Pour tester le code de KNN et sa viariante DWNN, j'ai testé les deux algorithme avec differentes valeurs de K = 1, 3, 7

Pour K= 1 on obtient les affichages suivants (KNN ensuite DWNN à chaque fois) :



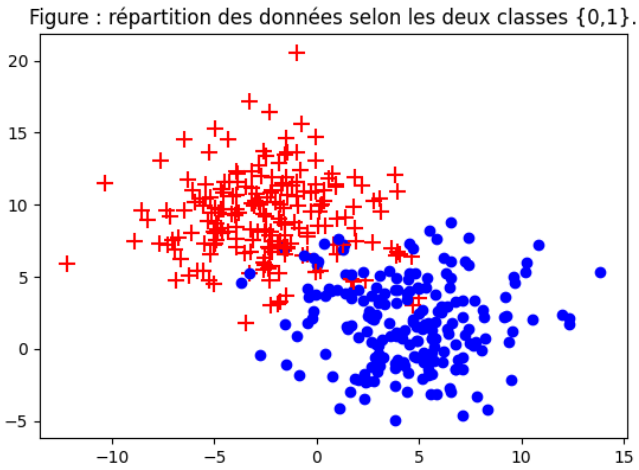


Figure : répartition des données selon les deux classes {0,1}.
20
15
10
5
0
−5
−10
−5
0
5
10
15

Pour K = 3

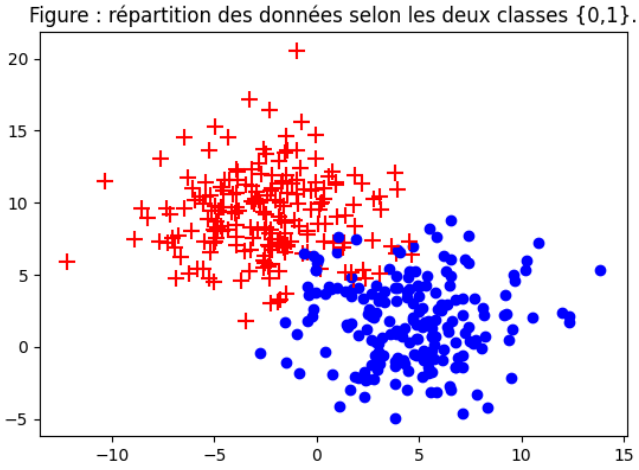


Figure : répartition des données selon les deux classes {0,1}.
20
15
10
5
0
−5
−10
−5
0
5
10
15

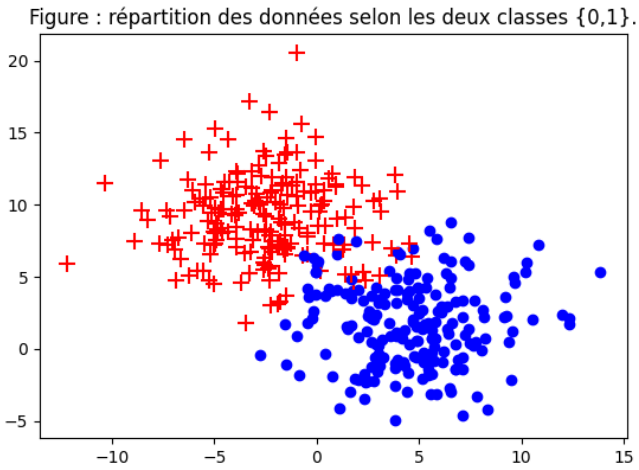


Figure : répartition des données selon les deux classes {0,1}.
20
15
10
5
0
−5
−10
−5
0
5
10
15

Pour K = 7

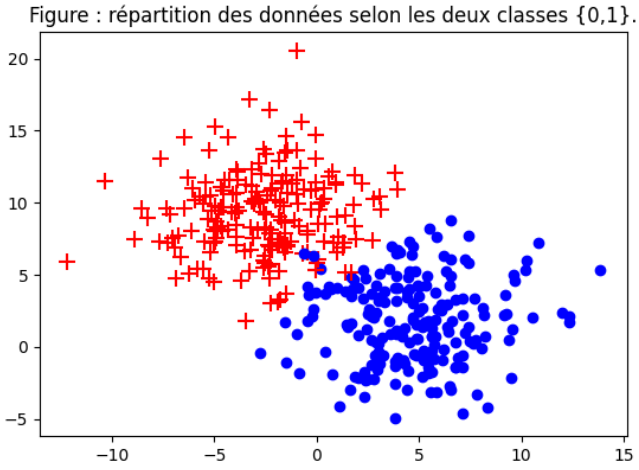
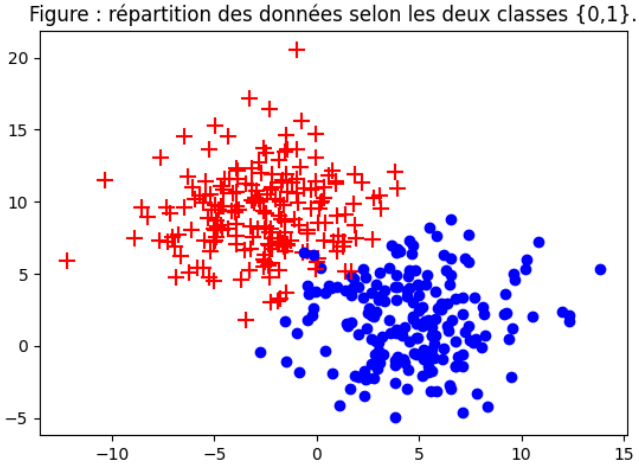


Figure : répartition des données selon les deux classes {0,1}.
20
15
10
5
0
−5
−10
−5
0
5
10
15



Si on regarde bien et on compare à chaque fois pour chaque K les deux versions, on remarque des differnces minimes. Pour mieux voir les choses on teste en calculaut l'Accurracy, la precision et le Recall.

Le calcul renvoie ces valeurs en pourcentage :

Pour KNN :

pour K=1

Accuracy : 92.75 precision : 93.0 Recall : 92.5

pour K=3

Accuracy : 92.5 precision : 93.5 Recall : 91.6

pour k=7

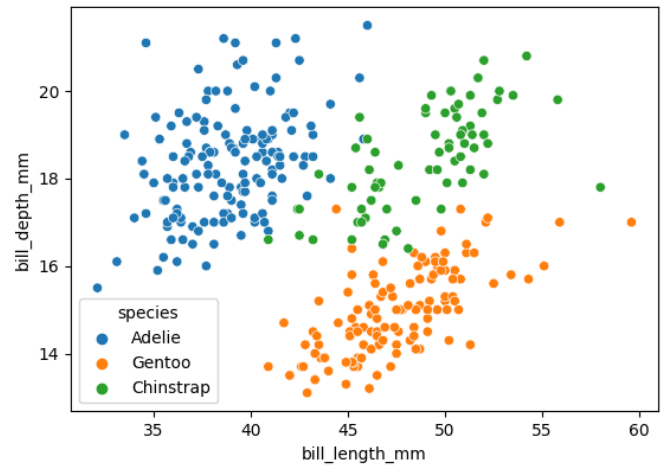
Accuracy : 94.0 precision :94.5 Recall :93.56

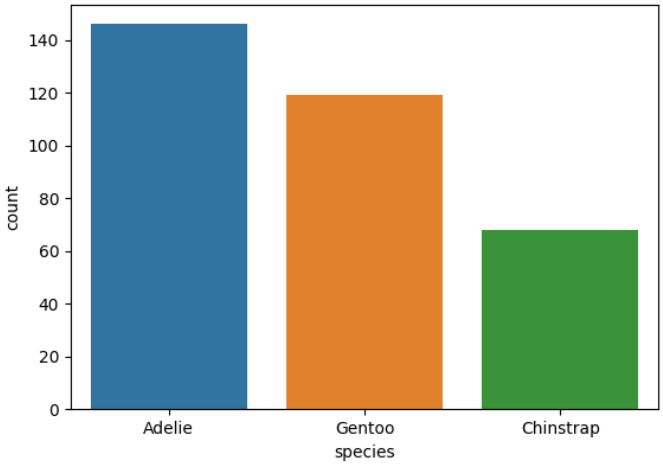
Ces valeurs ont été verifiées avec la bibliotheque scikit-learn Donc la justesse du code est bien verifiée, les valeurs retourné ne sont pas en pourcentage mais la multiplication par 100 renvoie les memes valeurs.

3 Exercice 2

dans cet exercice nous etudions des ensemble de pingouins

l analyse des donnees nous donne ces figures :





Pour etudier nos donnees j'ai decoupé mon dataSet en 3 ensembles avec pour le trainset la majorite des elements : 222, validTest : 55 et testSet 56 avec pour le trainSet 80 pour cent et le reste 10 pou cent pour chaque ensemble.

On a effacé une espece pour travailler : la fonction effacerespece efface l especе adeli de y puisque auparavant il existait 119 espece de type Gento et 136 d Adelie et Chinstrap on a fusionné les deux especes adelie et christoph pour obtenir deux especes 119 et 136 1 : Adelie + Chinstrap 0 : Gento

4 Conclusion

En conclusion pour trouver un K pertinent il faut un K qui renvoie une meilleur matrice de confusion.

pour tester les codes : python3 Classificatione_x1.py

python3 Classificatione_x2.py