KNN Demare Delgado

Introduction

Nous avons implémenté en python un simple KNN tel que présenté en TD. L'article de Medium Create a K-Nearest Neighbors Algorithm from Scratch in Python a été très utile pour compléter les recherches.

L'algorithme est simple car nous calculons seulement les distances entre l'individu testé et le dataset source. Nous avons testé k sur les dataset fourni pour comparer la précision.

Choix de la distance et tests

Une des premières étapes clef du KNN est de déterminer le meilleur k , en prenant en compte différents calculs de distances de la librairie scipy.spatial.distances. Nous avons choisis la distance euclidienne car elle est à une précision satisfaisante et est plus rapide que les autres options que nous avions étudiées que sont Tchebychev, Manhattan ou encore Minkowski.

Ainsi avec cette algorithme nous avons obtenu une précision de près de 98% sur les dataset data et preTest.

Idée d'amélioration

Malgré de bon résultat de précision assez rapidement, nous nous sommes penchées sur des pistes pour approfondir notre algorithme. Une piste trouvé était d'implémenter un préprocessing sur le dataset pour éliminer les mauvais représentants et garder seulement les meilleurs et ainsi améliorer la précision de l'algorithme. Cependant nous n'avons pas eu le temps d'implémenter un algorithme d'undersampling.

Comparaison avec le KNN de Scikit

Le fichier knn_Sklearn contient l'implémentation du KNN en utilisant la bibliothèque scikit afin de comparer les résultats avec le KNN « maison ». On obtient une précision maximale de 98,5% ce quest très proche de la précision maximale de notre KNN « maison ». Ci-dessous la précision du modèle scikit selon le dataset et le k.