

CHALLENGE IA

Pour ce challenge, nous avons donc repris le code que nous avons utilisé pour le TD dédié au knn. Nous avons changé quelques petits détails.

Nous avons utilisé une autre formule pour la distance : on a utilisé la formule de la distance de Hamming à la place. En effet après quelques tests entre cette formule de distance et la formule euclidienne, il s'est avéré qu'on avait de meilleur résultat avec celle de Hamming.

Pour la fonction qui permet de trouver les k plus proches voisins nous n'avons pas changé de méthode : on crée une liste avec les distances de l'objet qu'on est entrain d'évaluer, on l'a trie, on renvoie l'étiquette de l'objet associé à chaque distance puis on crée une autre liste qui récupère les k premiers éléments.

C'est dans la fonction `predire_classe` (qui nous permet de prédire l'étiquette de l'objet évalué comme son nom l'indique) que nous avons changé quelque chose également. De base lorsqu'on parcourait la liste des k voisins pour savoir qu'elle était l'étiquette qui revenait le plus souvent, on créait un tableau qui accordait un point pour une étiquette lorsqu'elle matchait avec la liste des k voisins. Mais cette fois-ci, en fonction de la distance avec l'objet à évaluer, nous avons instauré une échelle de point : plus un voisin était proche de l'objet, plus l'étiquette qui correspondait avec de points. Cette méthode nous a permis de gagner 3% dans notre précision.

Ainsi avec le fichier `data.csv` en apprentissage et le fichier `preTest.csv` en évaluation, nous avons une précision de 88%.