Rapport KNN – Pierre Langhade, Minji KIM

Nous avons choisi de créer une classe KNN qui prend en paramètre k, le nombre de voisins à considérer, initialisé à 11 si aucune valeur n'est renseignée.

Méthodes de classe :

- Match : prend en paramètre les données à étiqueter et les étiquettes et les enregistre en tant que variables de classe pour pouvoir apprendre

- Predict : prend en paramètre les données à étiqueter et retourne la liste de toutes les étiquettes prédites pour chaque donnée de la liste de données.

- _predict : sous-fonction de Predict, réelle fonction de prédiction.
- Dans un premier temps, on calcule la distance entre la donnée dont on veut prédire l'étiquette et la donnée d'entraînement avec la fonction distance euclidienne.
- K_index utilise argsort pour trier la liste des distances (du plus proche au plus lointain) et retourne la liste des indexes de ces distances triées.
- Pour garder uniquement les k plus proches voisins, on applique le slicing du début de k_index jusqu'à l'index k compris.
- Ensuite, on récupère les étiquettes des données d'entraînement aux indexes compris dans k_index et enfin, on retourne l'étiquette la plus utilisée parmi la liste d'étiquettes récupérées juste auparavant.
- Ce résultat est donc envoyé dans la grande fonction predict qui réitère l'opération pour chaque donnée des données de test.

Pour faire tourner l'algorithme, on crée donc un objet de la classe knn. Les données d'entraînement ainsi que leurs étiquettes associées sont données à la méthode de classe match et enfin on utilise la fonction de classe predict sur les données de test afin de récupérer les étiquettes prédites.

Pour le fichier « preTest.txt », l'algorithme a prédit 359 valeurs correctement pour k=15.