

Dossier Developpement efficace

Implémentation de K-NN

Pour implémenter l'algorithme knn, nous avons créée une méthode knn qu'on utilise dans la class : ControleurDistance.java

Méthode : knn

Classifie un objet en fonction de k-voisins les plus proches.

```
public static Object knn(Object newObj, int k, String typeDist,
VuePoint vue)
```

Methode de calcul de distance :

```
public static Double distanceEuclidienne(Object obj1, Object
obj2, VuePoint vue)
```

- Normalisation : Les axes sont normalisés entre 0 et 1 à l'aide des valeurs minimales et maximales extraites des données.

```
public static Double distanceManhattan(Object obj1, Object obj2,
VuePoint vue)
```

- Normalisation : Les axes sont normalisés entre 0 et 1 à l'aide des valeurs minimales et maximales extraites des données.
-

Traitement de la normalisation

Méthode : getMinMax

Récupère les valeurs minimales et maximales pour un axe donné.

```
private static double[] getMinMax(VuePoint vue, String axe)
```

- **Parcours** : Parcourt tous les objets de la liste pour identifier le min et le max.

Méthode : normalize

Normalise une valeur en fonction de son intervalle [min,max].

```
private static double normalize(double value, double min, double
max)
```

- Retourne 0 si max = min
-

Validation croisée

pas implémenté

Choix du meilleur k

pas implémenté

Efficacité

1. **Normalisation des distances** : Optimisée avec un seul parcours des données pour calculer les bornes $[\min, \max][\min, \max][\min, \max]$.
2. **Classification rapide** :
 - Identification des k plus proches voisins.
3. **Modularité** : La méthode **knn** s'adapte à différents types de données (Iris ou Pokemon)