

1 Calendrier

Vous devez déposer un document pdf sur le dépôt git de votre projet contenant les éléments décrits dans la section suivante au plus tard le 29 novembre 2024, avant minuit.

2 Description des éléments attendus dans le rendu pour l'UE2 (développement efficace)

Pour évaluer votre travail du point de vue algorithmique, nous attendons un document décrivant les éléments suivants :

Description des éléments attendus :

- Implémentation de k-NN** : Une description de votre implémentation de l'algorithme k-NN : classe implémentant l'algorithme, méthode(s) de cette classe implémentant le calcul de la distance, traitement de la normalisation, méthode(s) de cette classe implémentant la classification, méthode(s) évaluant la robustesse. N'hésitez pas à mettre en avant l'efficacité de ces méthodes (approprié pour un grand volume de données, normalisation efficace des distances).
- Validation croisée** : Une explication de votre méthode de validation croisée (comment sont calculés les pourcentages que vous donnez ?)
- Choix du meilleur k** : Les résultats pour différents k et les deux distances (Manhattan et Euclidienne), pour les iris et les pokemons. Eventuellement, distances basées sur différents attributs et/ou pondération. Conclusion sur le meilleur choix à faire.
Pour les données pokemon, la robustesse sera établie pour plusieurs (au moins 2) classifications (différents choix de catégorie).

Efficacité : Vous préciserez la ou les structures de données que vous avez utilisé pour l'implémentation de votre algorithme.

Vous expliquerez en quoi les choix que vous avez fait en terme d'algorithme et de structure de données sont efficaces (en terme de nombre d'opération, en rapidité d'exécution...) par rapport à d'autres possibilités.

3 Notation

Dans le barème, nous tiendrons compte :

- De l'orthographe
- De la clarté des éléments présentés
- Du recul pris sur votre implémentation
- De la présence des éléments décrits dans ce rapport dans les classes citées.

4 Correction et barème

— Implémentation de k-NN : 8 points

Bien vérifier que les éléments décrits sont bien présents dans le code.

2 pt Calcul de la distance : 1 pour le calcul simple, 1 pour la prise en compte d'un sous-ensemble d'attributs et d'une pondération.

2 pt Gestion de la normalisation **efficace**, avec renormalisation si nécessaire lors de l'ajout d'un point.

2 pt Classification : méthode qui détermine la catégorie d'un point étant donné une distance, un ensemble de données et un k .

2 pt Robustesse : méthode qui détermine le meilleur k pour une distance et un jeu de données.

— Validation : 1,5 points

1,5 pts Méthode de validation : description textuelle de la méthode suivie. Vérifier la cohérence avec la méthode vu dans le code au point précédent.

— Choix du meilleur k : 4,5 Vérifier la cohérence des résultats (100% partout c'est louche)

1,5 pt Pour les Iris : tableaux avec les résultats.

3 pt Pour les Pokemons : tableaux avec les résultats. 1,5 pour la deuxième catégorie.

— Efficacité 2 pts

1 pt Description des structures de données : comment sont stockées les données ? Comment sont gérés les types des attributs ?

1 pt Evaluation de l'efficacité de l'algorithme d'apprentissage implémenté : vérifier le nombre de parcours des données, la gestion de la normalisation (parcours après ou stockage dès le chargement des données).

— **Orthographe 2 pt**

— **Qualité globale, soin apporté 2 pts**