

TP2 : Apprentissage supervisé - Random Forest

Exercice 1: Classification avec des Forêts Aléatoires

Objectif : Dans cet exercice, vous utiliserez des forêts aléatoires pour classer des données sur des fleurs en différentes catégories.

Données : Nous utiliserons le jeu de données Iris, qui contient des mesures de différentes caractéristiques de trois espèces de fleurs. Vous pouvez le charger en utilisant scikit-learn.

Instructions :

1. Importez les bibliothèques nécessaires, y compris scikit-learn.
2. Chargez le jeu de données Iris.
3. Divisez les données en un ensemble d'entraînement et un ensemble de test.
4. Créez un modèle de forêt aléatoire avec les paramètres par défaut.
5. Entraînez le modèle sur l'ensemble d'entraînement.
6. Faites des prédictions sur l'ensemble de test.
7. Évaluez les performances du modèle en utilisant des mesures telles que la précision, le rappel et la F-mesure.
8. Expérimentez en ajustant les hyperparamètres du modèle (par exemple, le nombre d'arbres, la profondeur

Exercice 2 : Optimisation des Forêts Aléatoires

Objectif : Dans cet exercice, vous explorerez les techniques d'optimisation des forêts aléatoires pour améliorer les performances d'un modèle.

Données : Nous utiliserons le jeu de données CIFAR-10, qui contient 60 000 images réparties en 10 classes différentes. Vous pouvez le télécharger depuis la **bibliothèque torchvision de PyTorch**.

Instructions :

1. Importez les bibliothèques nécessaires, y compris scikit-learn et PyTorch.
2. Chargez le jeu de données CIFAR-10.
3. Prétraitez les données en les normalisant et en les divisant en ensembles d'entraînement, de validation et de test.
4. Créez un modèle de forêt aléatoire pour la classification des images. Vous pouvez utiliser la classe `RandomForestClassifier` de scikit-learn.

5. Entraînez le modèle sur l'ensemble d'entraînement et évaluez ses performances sur l'ensemble de validation en utilisant une métrique de votre choix (par exemple, l'exactitude).
6. Explorez les hyperparamètres du modèle, tels que le nombre d'arbres, la profondeur maximale des arbres, et d'autres paramètres pertinents. Utilisez la recherche en grille ou la recherche aléatoire pour trouver les meilleures combinaisons d'hyper paramètres qui améliorent les performances du modèle sur l'ensemble de validation.
7. Une fois que vous avez obtenu les meilleurs hyperparamètres, évaluez les performances du modèle sur l'ensemble de test pour obtenir une estimation réaliste de ses performances en situation réelle.