TP Classification

Objectifs

Dans ce TP, nous allons travailler sur la base des iris de Fisher. Il s'agit de prédire l'espèce d'iris en fonction de différentes caractéristiques végétales. Nous allons utiliser les algorithmes des k-plus proches voisins ainsi que la régression logistique afin de réaliser notre classification et de comparer leurs performances.

Commençons par récupérer la base de données des iris. Nous allons la charger dans un dataframe pandas à l'aide de la fonction read_csv.

Avec la bibliothèque pandas, il existe plusieurs fonctions permettant une première analyse simple des données:

- L'attribut shape permet de connaitre les dimensions du dataframe.
- La fonction info permet d'avoir un résumé rapide des données.
- La fonction describe permet d'avoir des statistiques sur différentes tendances sur les données.
- La fonction head permet d'afficher les premières lignes du dataframe.
- …

1. Utiliser ces fonctions pour répondre à ces questions :
   - Combien de classes ?
   - Combien de caractéristiques descriptives ? De quels types ?
   - Combien d'exemples ?
   - Combien d'exemples de chaque classe ?
2. Séparer des données en bases d'apprentissage et de test

La bibliothèque sklearn fournit la fonction train_test_split qui permet de séparer la base. Pour cela, nous allons utiliser from sklearn.model_selection import train_test_split. Cette fonction a l'avantage de randomiser l'ensemble avant de faire le split, ce qui est très important avec cette base des iris.

3. Il nous faut maintenant apprendre les modèles. Commencez par créer un knn (KNeighborsClassifier()) puis LogisticRegression(). Ensuite il faut l'entrainer sur la base

d’apprentissage avec la fonction fit. On peut mesurer la précision de notre modèle avec accuracy et f1-score.

- Quel résultats obtenez-vous en apprentissage ? Et en test ?
- Afficher ensuite la matrice de confusion. Qu’observez vous ?
- Jouer avec le paramètre k. Etudiez l’influence du paramètre k.

4. En utilisant une validation croisée sur le jeu d’entrainement en 5 folds. Optimisation les performances des modèles étudiés.