TD Apprentissage supervisé à partir de données personnelles

L'objectif de ce TD est d'illustrer les enjeux de l'apprentissage supervisé à partir de données personnelles. Concrètement, nous allons essayer d'estimer si il est possible de prédire le genre et l'animal préféré d'une personne à partir de données personnelles numériques simples (collectées en classe) telles que la taille, l'âge, la pointure de chaussure, etc.

1. **Environnement de travail et chargement des données**
    a. Téléchargez le notebook sur Google collab (Fichier>télécharger le fichier .ipynb): https://colab.research.google.com/drive/1in7KLJD192u3THJxJ6eAaIW5UjRfu5jH
    b. Ouvrir le fichier .ipynb via jupyter-notebook

2. **Régression logistique**
    a. Ecrivez un code permettant d'apprendre une régression logistique sur un split 75% / 25% (cf. séance précédente) pour prédire la classe "genre", puis la classe "animal" .
    b. Affichez les labels prédits et attendus, l'accuracy et les poids du modèle linéaire appris.

3. **Leave-P-out cross validation avec Régression logistique**
    a. Ecrivez un code permettant de faire de la validation croisée de type leave-p-out avec les fonctions cross_val_score et LeavePOut du package model_selection.
    b. Affichez l'accuracy moyenne (pour P=1) pour les deux tâches de classification ("genre" et "animal"). Est-elle meilleure ou moins bonne que le meilleur prédicteur constant ?
    c. Affichez l'écart type de l'accuracy sur l'ensemble des splits.
    d. Faites de même avec un classifieur knn (KNeighborsClassifier) et faites varier la valeur de k

4. **Sélection de modèles par cross-validation et grid search**
    a. Utilisez la fonction GridSearchCV du package model_selection pour automatiser la sélection du paramètre C d'un svm C-régularisé (voir exemples sur la page de GridSearchCV). Affichez l'accuracy moyenne obtenu par le meilleur modèle (meilleure hyper-paramétrisation) .

b. Faites de même pour optimiser les hyper-paramètres de KNeighborsClassifier (*n_neighbors*), RidgeClassifier (*alpha*), RandomForestClassifier (*n_estimators*, *max_depth*)

5. **Interprétabilité par tests d'ablation ou randomisation de variables**
    a. Effectuez des tests d'ablation des variables d'entrée (une par une) pour estimer leur importance (en utilisant le meilleur modèle)
    b. Effectuez des tests de randomisation des variables d'entrée (une par une) pour estimer leur importance (en utilisant le meilleur modèle). Vous pouvez utiliser la fonction "sklearn.inspection.permutation_importance".

6. **Conclusion**
    a. Parmis tous les modèles de classification appris lequel est le meilleur ?
    b. Est-il possible de prédire l'animal préféré d'une personne à partir des données utilisées ?