esiea
ECOLE D'INGENIEURS
DU MONDE NUMERIQUE

5A : Machine Learning 2

Mini-projet – Ensemble Learning

Attention :

- les parties ne sont pas indépendantes. Si vous tirez des conclusions dans la partie *p*, ne les oubliez pas quand vous commencez la partie *p+1*.
- Pensez à mesurer les temps d'apprentissage et d'inférence de chaque solution.

A. Base des données

La base de données « *beer quality* » comporte 1600 exemples décrits par 11 caractéristiques quantitatives. L'objectif est d'évaluer, sachant ces variables, la qualité d'une bière, évaluée de 1 (sans commentaire !) à 10 (particulièrement excellente).

Charger la base et la séparer en deux : la matrice *X* des observations et le vecteur *y* des labels. Analyser rapidement les variables prédictives *X* et la variable à prédire *y* (effectifs par score).

Diviser la base de données en deux sous-ensembles d'apprentissage et de test (70/30).

B. Classification binaire

1) Créer une nouvelle variable quantitative *ybin* à deux modalités :
   - 0 : mauvaise qualité : $y < m$
   - 1 : bonne qualité : $y >= m$

   en fonction de la médiane *m* de la variable *y*.

2) Optimiser rapidement un arbre de décision pour réaliser la classification

3) Entraîner un ensemble d'arbres de décision « faibles » (peu, voire très peu profonds) à l'aide de l'algorithme *AdaBoost* :

   - Analyser les performances en fonction des différents paramètres
   - En particulier, tracer les courbes *accuracy* en fonction de *n_estimators* pour *max_depth = 1*, en apprentissage et en test.
   - Tracer la courbe *accuracy* en fonction de *n_estimators* pour *max_depth = 5*, en apprentissage et en test.
   - Peut-on mesurer l'importance d'une caractéristique dans la décision AdaBoost ? Expliquez. Afficher les variables par ordre d'importance.
   - Conclure sur le biais et la variance de l'algorithme.

C. Classification multiclasse

1) Créer une nouvelle variable quantitative *ymulti* discrète à 3 modalités : qualité basse (0), moyenne (1) ou élevée (2).

2) Déterminer les effectifs des différentes classes. Si nécessaire, équilibrer les données d'apprentissage (voir SMOTE). Dans la suite, on présentera les résultats obtenus avec et sans équilibrage.

Partie 1

3) Entraîner un réseau de neurones à une couche cachée pour effectuer cette tâche de classification, avec *early stopping* sur la base de validation. L'optimiser rapidement en prenant soin d'éviter l'over-fitting.

4) Faire un bagging en utilisant comme classifieur de base le réseau de neurones.

   - Tracer la courbe *accuracy* en fonction de *n_estimators*, en apprentissage et en test.
   - Conclure sur le biais et la variance.

Partie 2

5) Entraîner une forêt aléatoire :

   - Faire une recherche aléatoire (*random search*) pour optimiser les paramètres *max_depth* et *n_estimators*. Choisir les paramètres optimaux et donner les performances en apprentissage et en test.
   - Afficher les variables par ordre d'importance.
   - Conclure sur le biais et la variance.

D. Conclusion générale sur les méthodes d'ensemble.

Comparer les différentes techniques mises en œuvre en termes de performances (*accuracy*, temps d'apprentissage et d'inférence). Conclure.