Projet "Ensemble methods in ML" - Non-Alternants

M2 MALIA - Université Lyon 2 - 2022/2023

Responsable : Julien Ah-Pine

1 Objectif du projet

Dans le cadre du cours "Ensemble methods in ML" du Master 2 MALIA, il vous est demandé de réaliser un projet afin de compléter vos connaissances et de mettre en pratique des méthodes vues en cours dans le but de parfaire vos compétences de data scientist en Python.

2 Aspects liés à l'organisation et à la remise du dossier

Il est impératif de respecter les instructions données ci-dessous. Tout manquement pourra donner lieu à une pénalisation.

2.1 Groupes et contenu du dossier

Les projets s'effectuent par groupe de 2. Il est attendu que vous fournissiez :

1. un script python .py ou .ipynb dans lequel se trouvera tout le code (démarquez bien les deux parties -cf ci-dessous-),
2. un fichier comportant les données utilisées,
3. un rapport au format PDF accompagnant le projet.

Le code Python doit s'exécuter sans aucun problème et produire l'ensemble des résultats attendus. Il doit également être propre, bien structuré et bien commenté. Similairement, le rapport constitue un élément important de votre rendu. Il doit être bien structuré, bien rédigé et il doit mettre en perspective l'ensemble de votre travail que cela concerne vos implémentations ou vos expériences ou vos analyses. Vous trouverez dans la suite du sujet plus de détails sur ce qui est attendu.

Dans le cadre de votre rapport, une introduction générale sur l'objectif de ce projet et une conclusion sur les apports/limites de ce travail seront les bienvenues.

2.2 Constitution des groupes et des données

Une partie du projet correspond à l'étude approfondie d'un jeu de données que vous choisirez. Dans cette perspective, vous devrez m'envoyer par mail **au plus tard le 10 octobre 2022 à l'adresse : julien.ah-pine@univ-lyon2.fr**, la constitution de votre binôme, le jeu de données que vous souhaitez analyser. Ce jeu de données devra comporter au moins 5000 individus et au moins 20 variables. Dans votre message, vous m'indiquerez la source des données et une brève description du contexte et de la problématique que vous souhaitez étudier.

2.3 Modalités du rendu du dossier

Vous devrez créer une archive .zip contenant les fichiers correspondant aux éléments cités plus haut. Vous nommerez votre archive ainsi que les fichiers qu'elle contient de la manière suivante NOM1_NOM2.(zip|py|ipynb|pdf|csv|npz) où NOMi sont les noms des membres du groupe. Vous devrez soumettre cette archive **au plus tard le 21 octobre 2022 à 23h59** via la boîte de dépôt du cours Moodle. Si vos données sont trop volumineuses vous indiquerez un lien de téléchargement de celles-ci. **ATTENTION** : vous êtes responsables de votre envoi et donc toute absence de ressource ou tout problème conduisant à l'impossibilité d'accéder correctement à votre travail est de votre responsabilité.

3 Descriptif du travail à réaliser

Le projet est constitué de 2 parties :

— Une étude de cas à partir d'un jeu de données de votre choix visant à mettre en oeuvre et à comparer plusieurs méthodes d'ensemble en apprentissage supervisé.

— Une étude de cas visant l'implémentation de plusieurs fonctions permettant de mettre en oeuvre l'approche de consensus clustering vue en cours.

3.1 Partie 1 - Apprentissage supervisé

3.1.1 Programmation

L'objectif est d'étudier un cas pratique intéressant, présentant un problème de prédiction que ce soit dans le cadre de la régression ou de la catégorisation. Vous devrez pour cela implémenter le code permettant de mettre en oeuvre les points suivants :

— La lecture des données (que vous fournirez avec votre dossier en format .csv ou .xls ou .npz).

— Le prétraitement des données.

— La validation croisée permettant d'évaluer les performances des méthodes suivantes :

  — Une méthode linéaire pénalisée par norme ℓ_2, une méthode linéaire pénalisée par norme ℓ_1, une méthode basée sur les plus proches voisins, un arbre de décision.

  — La méthode ensembliste basée sur l'agrégation simple par moyenne artithmétique ou vote majoritaire des méthodes individuelles précédentes.

  — La méthode ensembliste basée sur l'agrégation par stacking des méthodes individuelles précédentes (vous choisirez comme bon vous semble la méthode de la 2ème couche).

  — La méthode ensembliste des random forest (avec le test de plusieurs valeurs pour les hyperparamètres).

  — La méthode ensembliste issue de AdaBoost (avec le test de plusieurs valeurs pour les hyperparamètres).

  — La méthode ensembliste issue du Gradient Boosting (avec le test de plusieurs valeurs pour les hyperparamètres).

— Des graphiques permettant de comparer les résultats de chaque méthode avec les différents paramétres utilisés.

3.1.2 Rapport

Le rapport devra contenir :

— Une présentation du jeu de données et de la problématique abordée.

— Une présentation succincte des méthodes testées pour faire la prédiction.

— Une présentation succincte des librairies Python utilisées pour mettre en oeuvre chaque méthode.

— Une présentation de votre protocol expérimental et des différents ensembles de paramètres utilisés pour chaque méthode le cas échéant.

— Une présentation, analyse et discussion des résultats obtenus pour chaque méthode et vis à vis de chaque ensemble de paramètres utilisés le cas échéant.

— Une présentation, analyse, comparaison et discussion des meilleurs résultats obtenus des différents modèles, conduisant à une conclusion sur le modèle que vous préconisez finalement pour traiter la problématique initiale.

3.2 Partie 2 - Apprentissage non-supervisé

3.2.1 Programmation

L'objectif est d'écrire un programme permettant de mettre en oeuvre la méthode de consensus clustering vue en cours à partir des résultats obtenus par plusieurs méthodes de classification automatique classiques. Vous devrez pour cela implémenter le code permettant de réaliser les points suivants :

- Une fonction intitulée `multiple_clustering` prenant en entrée une matrice de données `X` (individus × attributs), un nombre de clusters `k` comme hyperparamètre et donnant en sortie un ensemble de partitions issue des méthodes suivantes :
  - 3 résultats de la méthode k-means issus de 3 initialisations aléatoires différentes.
  - 1 résultat de la méthode bisecting k-means qui utilise récursivement le 2-means (c'est à dire le k-means avec k = 2) pour scinder en deux l'ensemble puis les sous-ensembles d'individus jusqu'à obtenir un nombre total de k clusters.
  - 4 résultats de la classification hiérarchique ascendante donnant k clusters avec comme mesures de dissimilarités : Ward, complete, single et average linkage.
- Une fonction intitulée `condorcet_matrix` prenant en entrée plusieurs partitions et donnant en sortie une matrice carrée indiquant pour chaque paire d'individus, le nombre de partitions les ayant mis dans le même cluster.
- Une fonction `consensus_clustering` prenant en entrée une matrice de données `X` (individus × attributs), un nombre de clusters `k` comme hyperparamètre et donnant en sortie le résultat du spectral clustering avec k clusters qui aura été appliqué à la matrice d'affinités donnée par `condorcet_matrix` elle-même issue des partitions obtenues par `multiple_clustering`.

Remarque : vous utiliserez les fonctions implémentées dans `scikit-learn` pour déterminer les solutions des méthodes de clustering de base utilisées en début de chaîne et celle du spectral clustering mis en oeuvre en bout de chaîne.

3.2.2 Rapport

Le rapport devra contenir :

- Une présentation succincte des méthodes de clustering de base et du spectral clustering utilisées dans le consensus clustering.
- Une présentation succincte du workflow des différentes fonctions implémentées.
- Une présentation, analyse et discussion des résultats obtenus pour chaque méthode de base et celle fournie par le consensus clustering sur le jeu de données que vous avez étudié en partie 1. Dans cette perspective, vous pourrez mesurer la qualité du clustering en confrontant les différentes partitions de base et celle du consensus clustering à la vérité terrain donnée par votre variable cible de la partie 1. Si précédemment, vous avez mis en oeuvre une tâche de catégorisation alors vous comparerez les différentes partitions de base et de concensus à la partition engendrée par la variable y au travers de la mesure Normalized Mutual Information [1] qui est fondée sur des mesures d'entropie (plus le NMI est proche de 1, plus les deux partitions sont similaires et meilleur est le résultat de clustering). Si vous avez étudiez un problème de régression dans la partie 1, alors vous comparerez les différents résultats de clustering de base et de consensus en calculant la moyenne pondérée des variances de y au sein de chaque cluster (il s'agit de la variance intra-classe, plus elle est petite, meilleur est le résultat du clustering puisque la partition est alors davantage en adéquation avec la distribution des valeurs de y). Cette mesure est donnée par (avec les notations du cours -cf slide 94-) :

$$\mathsf{Err}(C_1, \ldots, C_{\mathrm{k}}) = \frac{1}{\mathrm{n}} \sum_{m=1}^{\mathrm{k}} \sum_{i=1}^{\mathrm{n}} (y_i - \mu^m)^2 \mathsf{Ind}(\mathbf{x}_i \in C_m)$$

où μ^m est la moyenne des y_i des objets de C_m.

1. Voir `https://scikit-learn.org/stable/modules/generated/sklearn.metrics.normalized_mutual_info_score.html#sklearn-metrics-normalized-mutual-info-score`.