Test N°1 en IA2

Consignes

Ce contrôle s'effectue entièrement sur ordinateur.

Connecter-vous à Cyberlearn pour télécharger ***les données des deux exercices (exe1data.zip et exe2data.zip)****.*

Par la suite, il n'est plus autorisé d'accéder à Internet, d'utiliser d'autres objets connectés, ni de communiquer (désactiver votre wifi). *Tout autre support (cours, exercices, documentation Python, etc.) est autorisé.*

De plus, merci de respecter les indications suivantes :

- ***La durée maximale est de 1h45 (de 12h45 jusqu'à 14h30).***
- *Toutes les questions doivent être résolus dans un seul fichier nommé <nom_prénom.py>.*
- *Inclure vos noms et prénoms à l'entête de votre fichier de code.*
- *Inclure la réponse aux questions sous forme de commentaires dans le code.*
- *À la fin du travail uniquement, reconnecter vous au réseau et déposer son fichier directement sur Cyberlearn. Ce fichier contiendra un en-tête avec votre nom complet et la version de python que vous utilisez.*

Exercice 1 : classification de texte (15 pts)

Soit les commentaires sur les films stockés dans le répertoire ***data***. Chaque commentaire est mis sur un fichier (lire un exemple). Le sous-répertoire ***pos*** contient les fichiers des commentaires positifs et le ***neg*** contient les commentaires négatifs. Le but de cet exercice est d'implémenter un classificateur positif/négatif des commentaires en texte.

1. Charger les fichiers depuis le répertoire *data* en utilisant la méthode *load_files()* du package *sklearn.datasets*.
2. Diviser les données en train et test avec une répartition de 70-30%.
3. Implémenter 2 pipelines de classification : Naïf Bayes et SVM.
4. Implémenter pour chaque pipleline un Gridsearch avec une cross validation de 5 folders comme suit :
    a. Pour Naïf Bayes varier les paramètres
        i. use_idf : (True, False),
        ii. alpha : (1e-2, 1e-3)
    b. Pour SVM varier les paramètres
        i. use_idf: (True, False),
        ii. C: [0.1,1,5]
5. Afficher les meilleurs paramètres de la cross-validation de chaque modèle.
6. Interpréter les valeurs trouvées dans *use_idf*.
7. Interpréter la valeur trouvée de *C* dans le cas de SVM.
8. Evaluer les deux modèles avec le dataset de test.
9. Afficher un rapport d'évaluation pour chaque modèle.
10. Y-avait-il du surentrainement (overfitting) ? pourquoi ?
11. Interpréter les résultats des f1_score.

Exercice 2 : Regression (12 pts)

Le but de ce régresseur est de prédire les classes de prix (0, 1, 2, 3) des téléphones portables à partir de leurs caractéristiques techniques (puissance de la batterie, couleurs, mémoire, etc.).

1. Lire les données du répertoire ***data*** dans un dataframe Pandas et afficher les premiers 10 lignes des données de *train.*
2. Enlever depuis les données de *train* les lignes où la colonne ***px_height*** est nulle.
3. Diviser les données en train et test.
4. Utiliser un KNN classifier (***KNeighborsClassifier***) pour faire une première classification. Comment vous choisissez le k.
5. Implémenter un pipeline SVM avec un ***scaler*** de votre choix.
6. Chercher avec un gridsearch de votre choix les meilleurs paramètres de votre SVM.
7. Afficher le rapport d'évaluation après l'évaluation des données de test.
8. Conclure.
9. **Facultatif** : implémenter un arbre de décision puis un Forest Tree *(from sklearn.ensemble import RandomForestClassifier) (**+3 pts**)*