Examen mi-parcours

Master parcours SSD / C2ES - UE Analyse Fouille de Données

Automne 2019

Cet examen mi-parcours prend la forme d'un devoir maison, à réaliser **en binôme** pour le **lundi 20 janvier**. Vous devrez me rendre (à l'adresse `pierre.mahe@biomerieux.com`) :

1. un rapport de 6 pages maximum (sans le code) décrivant les analyses réalisées (2 pages maximum pour l'exercice 1, 4 pages maximum pour l'exercice 2),
2. un fichier texte nommé `votre-nom_exo-2.txt` contenant les prédictions obtenues à l'issue de l'exercice 2,
3. deux notebooks Jupyter (commentés un minimum) permettant de reproduire vos analyses.

1 Exercice 1

On considère qu'on dispose d'un jeu de données `(X,y)` lié à une problématique de classification binaire et qu'on a créé un objet `GridSearchCV` pour optimiser les paramètres d'un classifieur donné.

1. Expliquer la différence entre les deux portions de code ci-dessous :
   - code # 1 :

     ```
     > grid_search.fit(X, y)
     > acc = grid_search.best_score_
     ```

   - code # 2 :

     ```
     > cv_res = cross_val_score(grid_search, X, y, cv = 10)
     > acc = cv_res.mean()
     ```

2. Implémenter une procédure basée sur un classifier SVM à noyau RBF et le jeu de données "moons" pour illustrer cette différence. Vous générez le jeu de données grâce à la commande ci-dessous :

   ```
   > X, y = make_moons(500, noise = 0.6, random_state = 27)
   ```

2 Exercice 2

Dans cet exercice nous allons travailler sur une problématique de classification visant à reconnaître une propriété d'une bactérie à partir d'une matrice de descripteurs. Vous aurez à disposition un jeu d'apprentissage à partir duquel vous devrez fournir (à l'aveugle) des prédictions sur un jeu de test. J'évaluerai ensuite les prédictions et leur qualité contribuera à la note.

1. Charger le jeu de données et fournir quelques éléments d'analyse exploratoire (e.g., statistiques descriptives, analyse ACP) illustrant votre prise en main du jeu de données.
   - Le jeu d'apprentissage est stocké dans le fichier texte `train-data.txt`. La première colonne contient un code définissant la propriété à reconnaître : il s'agit du Gram de la bactérie, qui peut être positif ou négatif. Les colonnes suivantes sont les descripteurs disponibles.
   - Les données de test sont stockées dans le fichier `test-data.txt`, qui est formaté de la même manière (sans le Gram).
2. Construire un modèle de prédiction visant à maximiser l'aire sous la courbe ROC.
   - Vous avez toute liberté quant au choix des modèles à considérer, leurs hyperparamètres et les éventuels pré-traitements à appliquer au jeu de données.
   - Les codes `POS` et `NEG` définissent respectivement les catégories positive et négative à considérer pour définir les critères de sensibilité et spécificité.

3. Représenter sur une même figure l'évolution de la spécificité et la précision de votre meilleur modèle quand on fait varier sa sensibilité. Quelles valeurs de spécificité et de précision obtient-on pour une sensibilité de 80% ? Qu'est ce qui explique cette différence ?
   — Réaliser pour cela une expérience de validation croisée basée sur votre meilleur modèle en utilisant la fonction `cross_val_predict` (pour avoir accès aux prédictions obtenues sur le jeu de données).
   — Utiliser ensuite les fonctions `scikit-learn` permettant de calculer des courbes ROC et de "precision/recall".
4. Calculer enfin les prédictions obtenues sur le jeu de test et les enregistrer dans un fichier texte nommé `votre-nom_exo-2.txt`. Notez que leur qualité sera évaluée en terme d'aire sous la courbe ROC.