Exploration Numérique 3

13 octobre 2024

Nous allons étudier le jeu de données Breast Cancer Wisconsin (Diagnostic). Ce jeu de données est associé à un problème de classification binaire, où la variable cible correspond au diagnostic : "M" pour malignant (malin) et "B" pour benign (bénin). Il comporte $d = 32$ attributs numériques et un total de $n = 569$ observations.

1. Visualiser les boxplots des 10 premiers attributs (de 1 à 10). Commentez les distributions ainsi que la présence éventuelle de valeurs atypiques.
2. Visualiser la matrice de corrélation sous forme de *heatmap* (vous pouvez utiliser la fonction `heatmap` de la bibliothèque `seaborn`).
3. Analyser et commenter les corrélations mises en évidence par cette matrice.

Dans la suite, nous recentrons les régresseurs et les normalisons par leur écart-type empirique.

Afin de réduire le nombre d'attributs, nous appliquerons une méthode de réduction de dimensionnalité. L'approche classique dans ce cadre est l'Analyse en Composantes Principales (ACP), qui permet de simplifier les données tout en préservant autant que possible leur variance. Le processus est décrit comme suit :

1. Calculer la matrice de covariance $\hat{\mathbf{\Sigma}}$ de taille $d \times d$ sur l'ensemble des données.
2. Extraire les vecteurs propres et les valeurs propres de cette matrice, en triant les valeurs propres dans l'ordre décroissant.
3. Soit $\mathbf{e}_i$ le vecteur propre associé à la valeur propre λ_i. Nous choisissons les $\tilde{d}$ plus grandes valeurs propres ainsi que leurs vecteurs propres correspondants.
4. Former une matrice $d \times \tilde{d}$, notée $\mathbf{A}$, dont les colonnes sont les vecteurs propres sélectionnés, c'est-à-dire $\mathbf{A} = [\mathbf{e}_1, \ldots, \mathbf{e}_{\tilde{d}}]$.
5. Les données projetées dans cet espace de dimension réduite $\tilde{d}$ sont alors définies par :

$$\mathbf{x}'_i = \mathbf{A}^\top \mathbf{x}_i \quad \text{pour } i \in \{1, \ldots, n\}.$$

Dans un premier temps, nous fixerons $\tilde{d} = 3$ pour l'analyse.

6. Visualiser les boxplots des attributs après l'ACP.
7. Visualiser la *heatmap* des corrélations entre les nouvelles composantes.
8. Visualiser les données $\{(y_i, \mathbf{x}'_i)\}_{i=1}^n$ [avec des labels distincts pour les deux classes].
9. Visualiser la répartition des classes en projetant les données sur les composantes principales deux à deux : $[e_1, e_2]$, $[e_1, e_3]$, $[e_2, e_3]$.

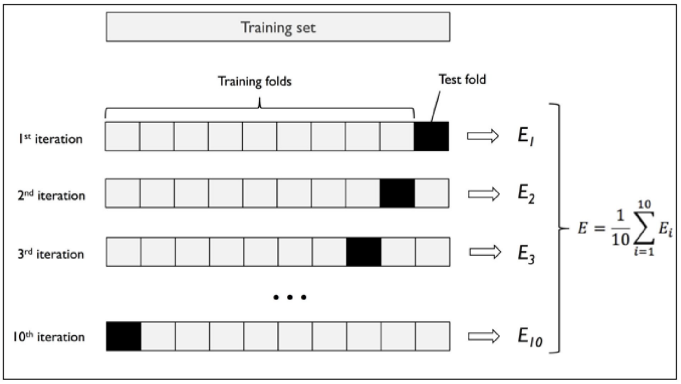

FIGURE 1 – Validation croisée à k *folds*, avec $k = 10$

10. Réaliser une Analyse Discriminante Linéaire (LDA) en considérant les trois composantes.
11. Visualiser la frontière de décision obtenue avec la LDA pour chaque paire de cmposante (trois tracés attendus).
12. Réaliser une Analyse Discriminante Quadratique (QDA) en considérant les trois composantes..
13. Visualiser la frontière de décision obtenue avec la QDA pour chaque paire de composantes (trois tracés attendus).

Nous allons maintenant évaluer les performances de ces deux classifieurs en utilisant la méthode de validation croisée.

Dans la validation croisée en "k-fold", l'ensemble de données est divisé aléatoirement en k sous-ensembles (*folds*) sans remplacement. À chaque itération, $k-1$ *folds* sont utilisés pour l'apprentissage du modèle, tandis que le *fold* restant sert à évaluer les performances. Ce processus est répété k fois afin d'obtenir k modèles et autant d'estimations de performance, que nous pourrons ensuite agréger pour obtenir une mesure globale de la performance du modèle.

Nous calculons ensuite la performance moyenne des modèles sur les différents *folds* indépendants afin d'obtenir une estimation globale de la performance. Étant donné que la validation croisée *k-fold* est une technique de rééchantillonnage sans remplacement, chaque échantillon est utilisé exactement une fois pour la formation et une fois pour la validation. La figure ci-dessous illustre le concept de la validation croisée *k-fold* avec $k = 10$.

Une amélioration par rapport à l'approche standard de la validation croisée *k-fold* est la validation croisée stratifiée, qui offre de meilleures estimations du biais et de la variance, notamment lorsque les classes sont déséquilibrées ; voir [**?**]. Dans la validation croisée stratifiée, les proportions de classe sont préservées dans chaque *fold*, garantissant que chaque sous-ensemble est représentatif des proportions observées dans l'ensemble complet de données d'apprentissage. Vous pouvez utiliser `StratifiedKFold` de `scikit-learn` pour implémenter cette méthode.

Nous définissons ensuite :

— la *précision* (*accuracy*) comme le rapport entre le nombre de "vrais positifs" et "vrais négatifs", et le nombre total de cas ;

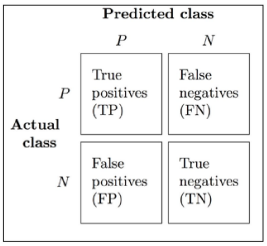

FIGURE 2 – Matrice de confusion

— le *rappel* (*recall*) comme le rapport entre le nombre de "vrais positifs" et la somme des "vrais positifs" et des "faux positifs".

Nous construisons également la matrice de confusion (voir Figure 2) pour mieux visualiser les performances des classifieurs.

14. Calculer, via validation croisée, la précision et le rappel pour les analyses discriminantes linéaires (LDA) et quadratiques (QDA) sur un k-fold avec $k = 10$.
15. Construire la matrice de confusion pour les deux classifieurs.