Validation croisée (Cross-Validation)

1. Concepts de base :

Qu'est-ce que la validation croisée et pourquoi est-elle importante ?
La validation croisée est une méthode d'évaluation de la performance d'un modèle de machine learning. Elle consiste à diviser les données en plusieurs sous-ensembles (ou folds) pour que chaque partie soit utilisée à tour de rôle pour entraîner le modèle (*training set*) et pour évaluer sa performance (*validation set*).
Elle est importante car :

- Elle fournit une évaluation plus fiable de la performance du modèle, en réduisant la dépendance à un seul découpage des données.
- Elle aide à détecter le surapprentissage (*overfitting*) et à choisir un modèle généralisable.

Différence entre validation simple (train/test split) et validation croisée :

- **Validation simple** : Divise une seule fois les données en un ensemble d'entraînement et un ensemble de test. Cela peut introduire de la variance dans les résultats selon le découpage choisi.
- **Validation croisée** : Permet d'utiliser plusieurs découpages en répétant l'entraînement/test sur différents sous-ensembles, offrant ainsi une évaluation plus robuste.

2. Types de validation croisée :

Différences entre k-fold, LOOCV et stratified k-fold cross-validation :

- **K-Fold Cross-Validation** : Divise les données en k parties égales. Chaque partie est utilisée une fois comme ensemble de validation, et les autres $k-1$ parties comme ensemble d'entraînement.
- **Leave-One-Out Cross-Validation (LOOCV)** : Cas particulier de k-fold où $k = n$ (nombre total de données). Chaque point est utilisé comme validation, ce qui peut être coûteux sur le plan computationnel pour de grands ensembles.
- **Stratified K-Fold Cross-Validation** : Une version de k-fold qui maintient la proportion des classes dans chaque *fold*. Cela est crucial pour des ensembles de données déséquilibrés.

Quand utiliser stratified k-fold ?
Lorsque les données sont déséquilibrées (classes minoritaires/majoritaires), stratified k-fold garantit que chaque pli contient une représentation proportionnelle des classes, ce qui évite des biais dans l'évaluation.

3. Applications et limites :

Avantages et inconvénients pour des ensembles de données déséquilibrés :

- **Avantages** : Maintient une évaluation robuste même si les données sont déséquilibrées. En combinant stratified k-fold avec des métriques adaptées (ex. F1-score), on peut mieux gérer les biais liés à la classe majoritaire.
- **Inconvénients** : Peut être moins performant si les données sont très petites ou si la stratification introduit trop de sous-ensembles similaires.

Comment éviter le surapprentissage grâce à la validation croisée ?

- En évaluant le modèle sur plusieurs découpages, la validation croisée détecte si le modèle surapprend (performances élevées sur l'entraînement mais faibles sur la validation).
- Elle aide également à choisir les hyperparamètres en évitant un ajustement excessif aux données d'entraînement.

4. Métriques et résultats :

Que représente le score moyen ?
Le score moyen lors d'une validation croisée est une estimation de la performance généralisée du modèle sur des données inconnues.

Interprétation de la variance des scores :

- **Faible variance** : Le modèle est robuste et généralisable.
- **Forte variance** : Le modèle est instable et sensible aux variations des données d'entraînement.

Optimisation des hyperparamètres (GridSearchCV et RandomizedSearchCV)

1. Concepts de base :

Différence entre paramètres et hyperparamètres :

- **Paramètres** : Estimés par le modèle lors de l'entraînement (ex. coefficients dans une régression).
- **Hyperparamètres** : Fixés avant l'entraînement (ex. profondeur d'un arbre, taux d'apprentissage) et nécessitent une optimisation externe.

Pourquoi une optimisation séparée ?
Les hyperparamètres influencent directement la performance du modèle. Une optimisation séparée, souvent via validation croisée, permet de trouver la combinaison optimale pour un modèle généralisable.

2. Approches d'optimisation :

Comment fonctionne GridSearchCV ?

- Explore de manière exhaustive toutes les combinaisons possibles des hyperparamètres spécifiés.
- **Avantages** : Approche exhaustive, garantit de trouver le meilleur paramètre si la grille est bien définie.
- **Inconvénients** : Coût computationnel élevé, particulièrement pour de grandes grilles ou des modèles complexes.

RandomizedSearchCV vs GridSearchCV :

- **RandomizedSearchCV** explore un nombre défini de combinaisons aléatoires dans la grille.
- **Avantages** : Moins coûteux en temps de calcul, souvent suffisant pour approcher une performance optimale.
- **Quand l'utiliser ?** : Lorsque l'espace des hyperparamètres est vaste ou que les calculs sont coûteux.

Facteurs influençant le choix de méthode :

- **Taille des données** : RandomizedSearchCV pour des ensembles volumineux.

- **Coût computationnel** : RandomizedSearchCV pour réduire la complexité.
- **Importance de l'exploration complète** : GridSearchCV pour des espaces restreints d'hyperparamètres.

3. Configuration et choix :

Paramètre cv dans GridSearchCV :

- Définit le type de validation croisée (ex. k-fold). Un choix adapté est crucial pour refléter correctement la diversité des données tout en gérant les biais.

Choisir les hyperparamètres et plages de valeurs :

- Basé sur une compréhension du modèle (par ex., profondeur d'arbre pour éviter le surajustement).
- Utilisation de tests préliminaires pour affiner les plages.

4. Problèmes courants :

Risques si la validation croisée est mal configurée :

- Surévaluation ou sous-évaluation du modèle (ex. *folds* non représentatifs des données).
- Longs temps d'exécution inutiles.

Data leakage dans l'optimisation :
Se produit si des informations des données de validation/test influencent l'entraînement (par ex., si les transformations des données sont faites avant le découpage). Pour l'éviter :

- Séparer clairement les pipelines pour les données d'entraînement et de test.
- Effectuer la validation croisée après toutes les prétraitements nécessaires.

5. Métriques et performance :

Évaluation des modèles optimisés :
Utiliser un ensemble de test indépendant pour vérifier les performances après optimisation.

Privilégier une métrique spécifique :

- **Accuracy** : Pour des données équilibrées.
- **F1-score** : Pour des données déséquilibrées où la balance entre précision et rappel est critique.