Validation Croisée

Rindra LUTZ / Nicolas ALLIX

17/01/2021

R Markdown

Introduction

Le monde d'aujourd'hui est un monde connecté. Cette connectivité apporte son lot de d'informations diverses : ce sont les données, ou data en anglais.

Pour traiter ces données, de nouveaux métiers ont vu le jour. Ces nouveaux métiers font appel à des traitements spécifiques dans le domaine des données comme par exemple la data analyse, la data science ou encore le datamining.

Le datamining regroupe des méthodes scientifiques destinées à l'exploration et l'analyse de données, à partir de grands volumes d'informations/de données, dans le but de créer de la valeur, comprendre des phénomènes, comprendre notre monde, et en particulier afin d'aider à prendre des décisions, anticiper des événements et agir.

Dans le datamining, il existe principalement deux grandes familles de méthodes. Ce sont les méthodes descriptives et les méthodes prédictives.

Méthodes descriptives

Une méthode descriptive (dite non supervisée) correspond à la recherche de structure, de relation, de corrélation.

- Permet de mettre en évidence des informations non visibles simplement
- Permet de résumer, synthétiser les données
- Sans variable ou phénomène à expliquer a priori

Les principales méthodes descriptives sont :

- les analyses factorielles
- les analyses typologiques
- modèles combiantoires
- les modèles à base de règles

Méthodes prédictives

Une méthode prédictive (dite supervisée) correspond à la modélisation et la prédiction d'un phénomène donné.

- Permet de définir un pattern (un modèle/une relation) pour expliquer un événement
- Permet d'extrapoler la cible
- Avec une variable/un événement à expliquer

Les principales méthodes prédictives sont :

- Régressions (linéaire, logistique, etc)
- Arbres de décision
- Réseaux de neuroness
- Analyse discriminante
- SVM - Support Vecteur Machine

Validation Croisée

Supposons que nous ayons un modèle statistique avec un ou plusieurs paramètres inconnus et un ensemble de données d'apprentissage sur lequel le modèle peut être «entraîné». Le processus de formation optimise les paramètres du modèle pour correspondre autant que possible aux données de formation. Si nous utilisons ensuite un échantillon de validation indépendant (de la même population que l'échantillon de formation), nous constatons généralement que le modèle ne modélise pas les données de validation ainsi que les données de formation : on parle de surapprentissage. Cependant, il n'y a pas toujours un échantillon de vérification indépendant. De plus, les performances de vérification du modèle peuvent varier d'un échantillon de vérification à l'autre. La validation croisée permet d'obtenir plusieurs intégrations de vérification à partir de la même base de données, de sorte qu'une estimation des performances de vérification du modèle plus robuste peut être obtenue par biais et variance.

Une fois que le modèle a été établi grâce à différents outils statistiques, il est alors nécessaire de valider sa fiabilité.

La Validation Croisée pour mesurer la fiabilité du modèle

Lors de toute modélisation, il est nécessaire de définir :

• Une population d'apprentissage (Train) : pour entrainer le modèle
• Une population de test (Test) : pour mesurer, tester la performance et robustesse du modèle

Or, une vérification via la validation croisée demanderait de travailler ici avec 3 types d'échantillons :

• Train
• Valid
• Test

Cet échantillon complémentaire (Valid) permet par exemple de tester plusieurs modèles (en faisant varier les paramètres du modèle ou les variables) : on essaie plusieurs modèles sur le Train et on identifie le plus performant sur le Valid. On teste enfin sur l'échantillon de Test (totalement vierge) le pouvoir de généralisation/sur-apprentissage sur des données toutes fraîches.

Il est possible de sophistiquer la structure des échantillons de Train/Validation au travers de quelques méthodes de validation croisée. En effet, les résultats dépendent de la manière dont ont été construits les 3 sous-ensembles Train/Valid/Test.

Les 3 principales méthodes de validation croisée sont :
• LOOCV (leave-one-out cross-validation)
• LKOCV (leave-k-out cross-validation)
• k-fold cross-validation

En résumé

La Validation non-croisée est également une possibilité, et est plutôt simple : on divisons l'échantillon de taille n en deux sous-échantillons. Le premier est appelé apprentissage (généralement supérieur à 60% de l'échantillon), et le second appelé échantillon de validation ou de test. Le modèle est construit sur des échantillons d'apprentissage et vérifié sur les échantillons de test par le score de performance que nous choisissons.

La validation croisée à k blocs (k-fold cross-validation) suit le modèle suivant : on divise l'échantillon original en k échantillons, puis on sélectionne un des k échantillons comme ensemble de validation pendant que les k-1 autres échantillons constituent l'ensemble d'apprentissage. Après apprentissage, on peut calculer une performance de validation, puis on répète l'opération en sélectionnant un autre échantillon de validation parmi les blocs de départ. À l'issue de ces procédures nous obtenons ainsi k scores de performances, soit un par bloc. La moyenne et l'écart type des k scores de performances peuvent être calculés pour estimer le biais et la variance de la performance de validation.

La validation croisée d'un contre tous (leave-one-out cross-validation : LOOCV) suit quant à elle ce modèle il s'agit d'un cas particulier de la validation croisée à k blocs où k=n. A chaque itération d'apprentissage-validation, l'apprentissage se fait donc sur n-1 observations et la validation sur l'unique observation restante.

La validation croisée permet aussi de déterminer des paramètres du modèle : on met en compétition k « sous-modèles » dont on mesure la performance pour déterminer le paramètre testé dont la performance du modèle est la meilleure.

Parlons rapidement du problème du sur-apprentissage, problème largement présent en modélisation.

Le sur apprentissage : problématique majeure en modélisation

Un modèle trop complexe, intégrant trop d'inputs et « épousant » trop les données d'apprentissage amènera donc une très bonne performance sur l'échantillon d'apprentissage (par construction), mais aura trop appris, notamment les bruits ou cas aberrants.
Il sera alors moins performant sur des données qui n'ont pas servi à la construction du modèle, c'est-à-dire sur les données sur lesquelles on souhaite faire la prédiction.
L'enjeu est donc de trouver le bon niveau de sophistication pour obtenir un bon niveau de performance sur l'échantillon d'apprentissage et sur l'échantillon de test.

Il n'y a pas sur-apprentissage lorsque la performance du modèle en Test est légèrement plus faible que celle en Train. Un écart trop grand est signe de **sur-apprentissage**.

Gestion des bases de données non-équilibrées

Dans les tâches de classification, la répartition des classes dans la base de données peut être déséquilibrée, c'est-à-dire que le nombre d'observations par classe peut ne pas être le même d'une classe à l'autre : si l'on dénote n_i le nombre d'observations de la i-ème classe, alors il existe {i,j} tel que n_i soit différent de n_j. Dans ce cas, pour éviter que la performance de validation (et d'apprentissage) ne soit biaisée par une répartition changeante des classes d'un ensemble de validation (resp. d'apprentissage) à un autre, il est recommandé d'utiliser une validation croisée stratifiée (« stratified cross validation »). La stratification consiste à s'assurer que la répartition des classes soit la même dans tous les ensembles d'apprentissage et de validation utilisés. C'est-à-dire que si la base de données initiale présente, par exemple, 3 observations de la classe 1 pour 7 observations de la classe 2, alors chaque ensemble de validation (resp. d'apprentissage) devra présenter ce ratio de 3 pour 7.

Dans le cas de la validation croisée à k blocs, il s'agit simplement de répartir les classes de la même manière d'un bloc à un autre. Les ensembles de validation et d'apprentissage qui en dériveront hériteront de cette répartition.