7

L'évaluation d'un modèle

Introduction

Si vous avez lu la partie précédente, vous savez désormais comment construire de beaux modèles de *machine learning*. Vous l'avez vu, ce n'est pas si difficile et vous vous sentez certainement prêt à aller en découdre sur l'un des challenges *Kaggle* en cours. Mais attention, prenez encore quelques instants pour lire ce qui suit afin d'éviter de tomber dans l'un des pièges classiques du débutant. Souvenez-vous d'un phénomène que nous avons évoqué précédemment, notamment lorsque nous avons parlé de la régression polynomiale : l'*overfitting*.

En effet, il est souvent très facile de construire un modèle qui restitue très bien les données utilisées pour son estimation. Il est néanmoins bien plus difficile de faire en sorte que ce modèle puisse se généraliser, c'est-à-dire qu'il soit capable de prédire de façon satisfaisante de nouvelles observations, non utilisées lors du calcul du modèle. Pour trouver un juste équilibre entre apprentissage du modèle et capacité prédictive, il est indispensable de mettre en place un dispositif qui permette d'évaluer globalement la qualité d'un modèle.

La présentation de ce dispositif est l'objet de ce chapitre, composé de deux parties. La première introduit la notion de validation croisée, qui est un dispositif d'évaluation d'un modèle ; la seconde présente un ensemble d'indicateurs (aussi appelés métriques de performance) que vous pourrez utiliser pour mesurer effectivement la qualité de vos modèles.

La validation croisée

De la nécessité de diviser vos données

À partir d'un jeu de données initial, que feriez-vous pour à la fois constituer un modèle et tester sa capacité prédictive sur des données non utilisées pour la modélisation (sans attendre de nouvelles observations, bien sûr !) ? La première réponse qui vient à l'esprit est assez évidente : diviser les données en deux groupes. L'un des groupes est utilisé pour la modélisation, l'autre est utilisé pour effectuer une prévision sur des données « fraîches ». C'est effectivement l'approche de base que l'on peut adopter. On crée un échantillon d'entraînement, sur lequel on va constituer le modèle, et un échantillon de test, sur lequel on va tester le modèle. Pour évaluer la qualité du modèle et de sa performance en prévision, on utilise une métrique de performance P (nous en reparlerons dans la deuxième partie de ce chapitre). Bien évidemment, on se doute que P_{test} sera inférieur à $P_{entraînement}$. En pratique, on a l'habitude de prendre 70 % des données pour l'échantillon d'entraînement (appelons-le $m_{entraînement}$) et 30 % des données pour l'échantillon de test (m_{test}).

Voilà pour l'approche de base… Mais si on allait plus loin ? En effet, on pourrait avoir envie d'utiliser cette séparation des données pour faire le meilleur modèle possible. On pourrait ainsi essayer différents choix de variables, plusieurs paramétrages d'un modèle (rappelez-vous les différentes manières de customiser les modèles) sur $m_{entraînement}$ et voir lequel performe le mieux sur m_{test}. C'est une idée effectivement perspicace, puisqu'elle nous permettrait de trouver celui, parmi tous les possibles, qui va maximiser P_{test} (car c'est généralement ça que l'on attend d'un modèle). De plus, comme l'indique Hyndman dans son blog[1], c'est une approche pragmatique pour choisir un modèle : efficace, concrète, et bien plus simple que l'emploi de tests statistiques de comparaison de modèles.

Néanmoins, pourrait-on alors dire à juste titre qu'on a bien testé que le modèle se généralise bien ? Pas vraiment, puisqu'il aurait été choisi de façon à maximiser la qualité de prévision sur m_{test}, donc il ne serait plus complètement vrai d'affirmer qu'il a été testé sur des données toutes fraîches et innocentes !

Pour sortir de ce dilemme, le *data scientist* choisit généralement de diviser ses données en trois :

- un jeu d'entraînement, bien sûr ($m_{entraînement}$) ;
- un jeu dit de validation ($m_{validation}$) : celui-ci va être utilisé pour tester les différents modèles paramétrés sur $m_{entraînement}$ (il remplace le m_{test} précédent) ;
- et un vrai jeu de test (m_{test}), qu'on garde de côté et qui ne sera utilisé que tout à la fin du processus de modélisation, afin de tester le plus honnêtement possible la capacité de généralisation du modèle retenu.

La qualité de l'ajustement ou de la prévision est calculée pour chacun des jeux de données, à partir de la métrique P retenue. En pratique, on prend souvent 60 % des données pour $m_{entraînement}$, 20 % pour $m_{validation}$ et 20 % pour m_{test}. Ces principes sont résumés dans la figure 14-1.

1. http://robjhyndman.com/hyndsight/crossvalidation

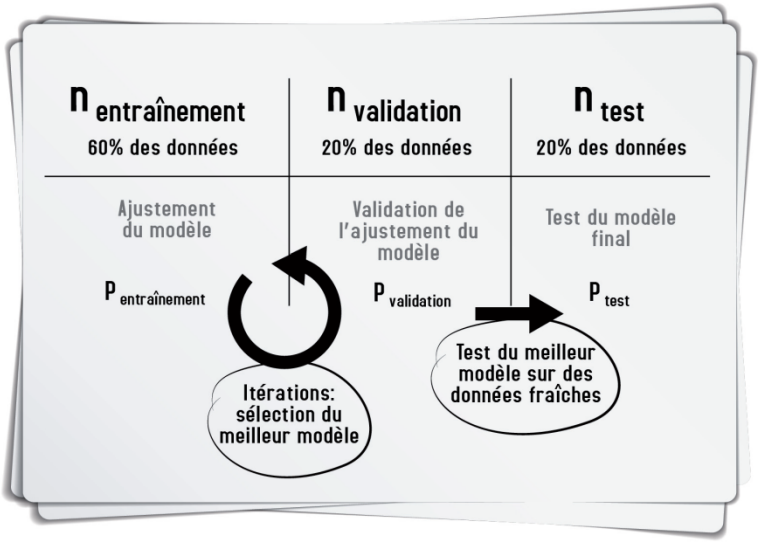

Figure 14-1 – *Les notions de jeux d'entraînement, de validation, de test*

Comme l'explique Hyndman dans son blog déjà cité, ces questions de séparation des données préoccupent plus les praticiens du *machine learning* que les statisticiens plus traditionnels. Cela peut s'entrendre : l'objectif du statisticien est avant tout de comprendre les processus stochastiques à l'œuvre dans les données, en essayant de contrôler les effets des variables du modèle. En *machine learning*, on se préoccupe moins de ces questions que de la capacité du modèle à faire la meilleure prédiction possible sur de nouvelles données, quitte à utiliser un modèle boîte noire.

La mécanique de la validation peut sembler bien évidente, pour ne pas dire basique : on coupe le jeu de données en trois paquets de données pour entraîner, valider et tester. En réalité, il existe beaucoup d'alternatives permettant de sophistiquer cette approche : on parle alors de validation croisée.

La validation croisée

Les questions liées à la validation d'un modèle peuvent vite devenir très complexes. Nous n'en aborderons quelques techniques que très superficiellement ici. Notre objectif est avant tout de vous faire comprendre que cette étape est très importante pour la bonne résolution d'un problème d'analyse de données. Il est donc nécessaire de lui consacrer un moment de réflexion non négligeable lors de tout projet de *data science*.

Par rapport à la version naïve de la validation présentée juste avant, on a généralement recours à une approche plus exhaustive, visant à ce que les données, à l'exception de celles utilisées pour le test du modèle, soient plusieurs fois utilisées pour faire partie de $m_{entraînement}$ et de $m_{validation}$. On arrive ainsi à mesurer de façon bien plus générale la qualité du modèle. Cette approche est qualifiée de validation croisée. Plusieurs méthodes de validation croisée existent, en voici les principales.

- La méthode LOOV (*leave-one-out cross-validation*) consiste à sortir une observation i de l'ensemble du jeu de données (rappel : à l'exception des données de test) et à calculer le modèle sur les m-1 données restantes. On utilise ce modèle pour prédire i et on calcule l'erreur de prévision. On répète ce processus pour toutes les valeurs de i = 1, …, m. Les m erreurs de prévision peuvent alors être utilisés pour évaluer la performance du modèle en validation croisée ($P_{Xvalidation}$).
- La méthode LKOV (*leave-k-out cross-validation*) fonctionne selon le même principe que la LOOV, sauf que l'on sort non pas une, mais k observations à prédire à chaque étape (donc LOOV est équivalent à LKOV pour k = 1). Le processus est répété de façon à avoir réalisé tous les découpages possibles en données de modélisation/de prévision.
- Enfin, avec la méthode *k-fold cross-validation*, les données sont aléatoirement divisées en k sous-échantillons de tailles égales, dont l'un est utilisé pour la prévision et les k-1 restants pour l'estimation du modèle. Contrairement à la LKOV, le processus n'est répété que k fois. À noter que la *k-fold cross-validation* permet de faire en sorte que la distribution de la variable à prédire soit équivalente dans chacun des sous-échantillons, ce qui est particulièrement intéressant dans le cas des jeux de données déséquilibrées. On parle alors de *stratified k-fold cross-validation* (remarque : chercher à équilibrer cette répartition est également une bonne pratique dans la séparation des données de test du reste du jeu de données).

Ces méthodes peuvent être regroupées en deux grandes familles : LOOV et LKOV sont des validations croisées dites « exhaustives », car une fois terminées, elles ont divisé le jeu de données en fonction de toutes les combinaisons possibles. La *k-fold cross-validation* et ses variantes sont « non exhaustives », dans le sens où elles séparent les données en un nombre limité de sous-ensembles d'observations possibles. Les méthodes non exhaustives sont des approximations des méthodes exhaustives, mais elles nécessitent moins de temps de calcul.

Il existe d'autres approches de la validation croisée, basées sur les théories de rééchantillonnage statistique, comme le *bootstrap* par exemple (dont certains principes ont déjà été abordés lorsque nous avons parlé du *bagging*). Nous n'en dirons pas plus dans le cadre de cet ouvrage, laissant soin au lecteur d'approfondir ce sujet par lui-même, par exemple en se référant à la bibliographie suggérée.

Choix de la métrique de performance (P)

Pour les problèmes de régression

Nombreuses sont les mesures disponibles pour évaluer la qualité d'un modèle de régression. Elles se basent toutes sur de savants calculs réalisés à partir de trois grandeurs :

- la valeur observée d'une série à prédire (y_i) ;
- la valeur prédite par le modèle pour cette même valeur observée ($\widehat{y_i}$) ;
- et une prévision naïve de référence, qui est la moyenne de la valeur observée ($\overline{y}$)[2].

2. C'est l'une des méthodes de prévision que pourrait employer quelqu'un qui n'a aucune notion de modélisation… très basique, mais pas forcément idiot !

Elles permettent de calculer, pour tout *i* des *m* observations :

- l'erreur de prédiction du modèle : $y_i - \widehat{y_i}$;
- l'erreur de prédiction naïve : $y_i - \overline{y}$.

Tout cela permet de définir des indicateurs de performance du modèle. Les plus connus sont les suivantes :

- l'erreur moyenne absolue (MAE, *Mean Absolute Error*) :

$$\frac{1}{m}\sum_{i=1}^{m} | y_i - \widehat{y_i} |$$

- la racine carrée de la moyenne du carré des erreurs (RMSE, *Root Mean Squared Error*) :

$$\sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(y_i - \widehat{y_i}\right)^2}$$

- le coefficient de détermination (R^2) :

$$1 - \frac{\sum_{i=1}^{m}\left(y_i - \widehat{y_i}\right)^2}{\sum_{i=1}^{m}\left(y_i - \overline{y}\right)^2}$$

- le critère d'information d'Akaike (AIC). Celui-ci n'utilise pas les grandeurs mentionnées dans forme générale (qu'on ne détaillera pas dans ce livre), mais on les retrouve néanmoins dans le cas d'erreurs distribuées normalement.
  - Formulation générale :

$$-2\log\left(L\right)+2\left(k+1\right)$$

  - Formulation dans le cas d'erreurs distribuées normalement :

$$n\log\left(\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \widehat{y_i}\right)^2\right) + 2\left(k+1\right)$$

MAE est RMSE sont assez faciles à comprendre : ils correspondent à une indication agrégée de l'erreur de prévision. Par rapport à MAE, RMSE permet de punir plus sévèrement les grandes erreurs. Ces indicateurs ont l'avantage d'être dans l'unité de la variable à expliquer, ce qui permet d'en avoir une interprétation fonctionnelle. Mais ceci est aussi un désavantage, car l'évaluation du modèle reste idiosyncrasique.

Le R^2 permet quant à lui d'avoir une idée générale de la performance du modèle. En effet, il permet de comparer l'écart à la moyenne de la variable *y* à l'écart de la prévision dans le cadre d'un

modèle conditionnel. Il peut donc être interprété comme la part de la variation de *y* attribuable au modèle ou, formulé plus simplement, comme une mesure de l'adéquation entre le modèle et les données observées. Sa valeur est comprise entre 0 et 1, 0 indiquant une adéquation nulle et 1 une adéquation parfaite. Pour comprendre plus intuitivement ce que représente cette valeur, sachez que dans le cadre d'une régression linéaire simple, $R^2 = r^2$ (le carré du coefficient de corrélation de Pearson, présenté plus en détail dans le chapitre concernant l'analyse en composantes principales). Sans entrer plus en dans les détails, évoquons tout de même l'existence d'un R^2 ajusté, qui tient du nombre de variables explicatives du modèle : son objectif est de pénaliser les modèles avec trop de variables, parfois *overfittés*.

Le R^2 n'est cependant valable que dans le cadre de modèles géométriques de type régression linéaire, supposant des erreurs distribuées selon une loi normale. C'est pourquoi l'on utilise souvent l'indicateur global de l'AIC, qui n'est pas contraint par ces hypothèses. Sa formulation générale s'appuie alors sur la vraisemblance *L* du modèle. Qu'est-ce que la vraisemblance ? Euh… nous vous renvoyons à n'importe quel bon livre de statistique, qui saura y consacrer le nombre de pages nécessaires ! Dans le cadre de ce livre, retenez que c'est un tour de passe-passe mathématique qui permet de mesurer l'adéquation entre une distribution observée sur un échantillon et une loi de probabilité supposée décrire la population dont est issu l'échantillon. Donc, grossièrement, l'AIC évalue un modèle en fonction de sa vraisemblance, tout en pénalisant les modèles avec trop de variables. Meilleur est un modèle, plus petit est l'AIC. Généralement, on associe AIC au critère d'information de Bayes (BIC, aussi appelé critère de Schwartz), similaire dans l'esprit à l'AIC, mais qui pénalise plus fortement les modèles trop complexes.

Il est difficile de dire a priori si un indicateur est supérieur à un autre. En pratique, nous avons l'habitude d'en utiliser plusieurs simultanément, afin de qualifier le modèle selon plusieurs dimensions.

Pour les problèmes de classification

Les indicateurs basiques

L'évaluation d'un problème de classification se base sur une matrice de confusion, qui met en regard des données prédites et des données observées, comme le montre le tableau 14-1 (Tufféry, 2011).

Tableau 14-1. Un exemple de matrice de confusion (Tufféry, 2011)

		Observations		
		+	-	Total
Prédictions	+	250	150	400
	-	50	550	600
	Total	300	700	1000

(Les termes « + »/« - » peuvent être remplacés par toutes les sorties possibles d'un problème de classification binaire : vrai/faux, présent/absent, oui/non, sain/malade, etc.)

Cette matrice permet de calculer une première mesure très intuitive : le taux d'erreur, c'est-à-dire le taux de mauvaise classification (50 + 150)/1000 = 20 % (prédictions « - » alors que « + » plus prédictions « + » alors que « - », divisé par nombre total d'individus). Mais d'autres mesures

peuvent être tirées à partir de cette matrice, afin de décrire plus généralement le comportement du modèle par rapport aux données réelles. Pour cela, définissons les termes suivants (tableau 14-2).

Tableau 14-2. **Les termes définis par la matrice de confusion**

		Observations		
		+	-	Total
Prédictions	+	Vrais positifs (VP)	Faux positifs (FP)	Total des positifs prédits (VP + FP)
	-	Faux négatifs (FN)	Vrais négatifs (VN)	Total des négatifs prédits (FN + VN)
	Total	Total des vrais positifs observés (VP + FN)	Total des vrais négatifs observés (FP + VN)	Taille totale de l'échantillon (N)

Ainsi, le taux d'erreur évoqué précédemment peut être défini par : (FN + FP)/N.

Tout un ensemble d'autres indicateurs peut être calculé à partir de ces mesures. En général, pour évaluer un modèle, on utilise conjointement les deux indicateurs suivants :

- le taux de vrais positifs VP/(VP + FN), aussi appelé rappel (*recall*) ou sensibilité ;
- et la précision VP/(VP + FP).

Il en existe d'autres, mais le rappel et la précision permettent déjà de se faire une bonne idée générale de la qualité d'un modèle. Le rappel permet de mesurer la proportion de positifs prédits parmi tous les positifs de la population. La précision permet de mesurer la proportion de positifs de la population parmi tous les positifs prédits. Ainsi, un modèle parfait aura un rappel égal à 1 (il prédit la totalité des positifs) et une offre précision égale à 1 (il ne fait aucune erreur : tous les positifs prédits sont des vrais positifs). En pratique, les modèles sont plus ou moins performants suivant ces deux dimensions. Par exemple, on peut avoir un modèle très précis, mais avec un faible rappel : il prédira peu de positifs, mais les positifs prédits seront justes dans la plupart des cas. À l'inverse, un modèle très sensible, mais peu précis va prédire beaucoup de vrais positifs, mais également beaucoup de faux positifs.

En reprenant les chiffres du tableau précédent, nous avions un rappel de 0,83 (250 vrais positifs prédits sur 300 positifs réels) pour une précision de 0,63 (250 vrais positifs prédits sur 400 positifs prédits). On peut facilement augmenter le rappel, en prédisant systématiquement « + », comme indiqué dans le tableau 14-3.

Tableau 14-3. **Un autre exemple de matrice de confusion (Tufféry, 2011)**

		Observations		
		+	-	Total
Prédictions	+	300	700	1000
	-	0	0	0
	Total	300	700	1000

Dans ce cas, le rappel sera parfait : tous les positifs réels seront prédits comme positifs (300/300). Par contre, la précision va fortement être détériorée (300/1000 = 0,3) : ce modèle va générer beaucoup de faux positifs.

En pratique, c'est au *data scientist* de trouver le bon compromis entre rappel et précision lors du choix de son modèle. Par exemple, si l'on souhaite prédire « + » uniquement si l'on est vraiment sûr de la justesse de la prédiction, on aura tendance à favoriser la précision au détriment du rappel. À l'inverse, si l'on préfère prédire plus de « + » et réduire le nombre de faux négatifs, au risque de générer plus de faux positifs. C'est le rappel qui sera privilégié (rappel du chapitre sur la régression logistique : sommes-nous en train de nous amuser à prédire les survivants du Titanic ou diagnostique-t-on un cancer ?).

Pour comparer plusieurs modèles, on utilise également un indicateur agrégé, composé à partir du rappel et de la précision : le *F1 score*. On calcule pour cela la moyenne harmonique de la précision et du rappel (cela permet de pondérer les deux mesures de façon équivalente).

$$F_1 = \frac{2\left(précision * rappel\right)}{précision + rappel}$$

Autrement écrit, à partir des termes définis plus haut :

$$F_1 = \frac{2VP}{2VP + FP + FN}$$

Introduisons pour finir un dernier indicateur de performance, qui nous sera utile lorsque nous aborderons la courbe ROC : la spécificité. À l'opposé de la sensibilité, il mesure la proportion de négatifs prédits parmi tous les négatifs de la population : VN/(FP + VN).

Nous avons désormais tous les ingrédients pour nous attaquer à la description de la courbe ROC, qui est certainement la méthode d'évaluation des problèmes de classification la plus utilisée.

La courbe ROC

Principe

Nous l'avons évoqué plus haut : la classification repose sur un arbitrage entre rappel et précision. Cet arbitrage se base sur le choix d'un seuil de décision qui va favoriser l'un ou l'autre. Rappelez-vous par exemple le chapitre sur la régression logistique. La prédiction d'une valeur binaire est faite à partir d'une règle de décision et d'un seuil s :

$$Y = \begin{cases} «+» \; si \; P\left(Y = 1\right) \geq s \\ «-» \; sinon \end{cases}$$

Avec $s \in [0,1] \in \mathbb{R}$

La matrice de confusion dépend donc de la valeur de s. La courbe ROC va permettre de systématiser l'analyse des résultats d'un classifieur, en fournissant une vue synthétique de sa performance

pour toutes les valeurs de s possibles. Une courbe ROC peut être construite pour tout type de classifieur : c'est donc un outil puissant qui permet de comparer plusieurs modèles.

Pour la petite histoire, ROC signifie *Receiver Operating Characteristic*. On pourrait traduire par fonction d'efficacité du récepteur (mais c'est le terme anglo-saxon, beaucoup plus *ROC'n roll*, qui est généralement utilisé, même au sein de la communauté des *data scientists* francophones). En effet, avant d'être adoptées par le monde du *machine learning*, les courbes ROC ont été développées lors de la Deuxième Guerre mondiale dans le cadre de travaux sur le traitement du signal, pour séparer les signaux radars du bruit de fond.

Construction de la courbe ROC

Pour construire une courbe ROC, deux indicateurs de performance sont requis (voir plus haut) :

- la sensibilité α, c'est-à-dire le taux de vrais positifs ;
- la spécificité β, c'est-à-dire le taux de vrais négatifs.

La courbe ROC est alors tracée dans un espace de deux dimensions définies par α en ordonnée et 1-β en abscisse : cela revient à tracer le taux de vrais positifs en fonction du taux de faux positifs. Ces valeurs sont tracées pour les différentes valeurs d'un seuil de décision s. La courbe ROC est donc le graphique $(\alpha(s), 1\text{-}\beta(s))$; $s \in \mathbb{R}$, les valeurs maximale et minimale de s étant situées respectivement aux points de coordonnées (0,0) et (1,1). Ce graphique permet d'identifier un ensemble de zones et de points remarquables, indiqués dans la figure 14-2.

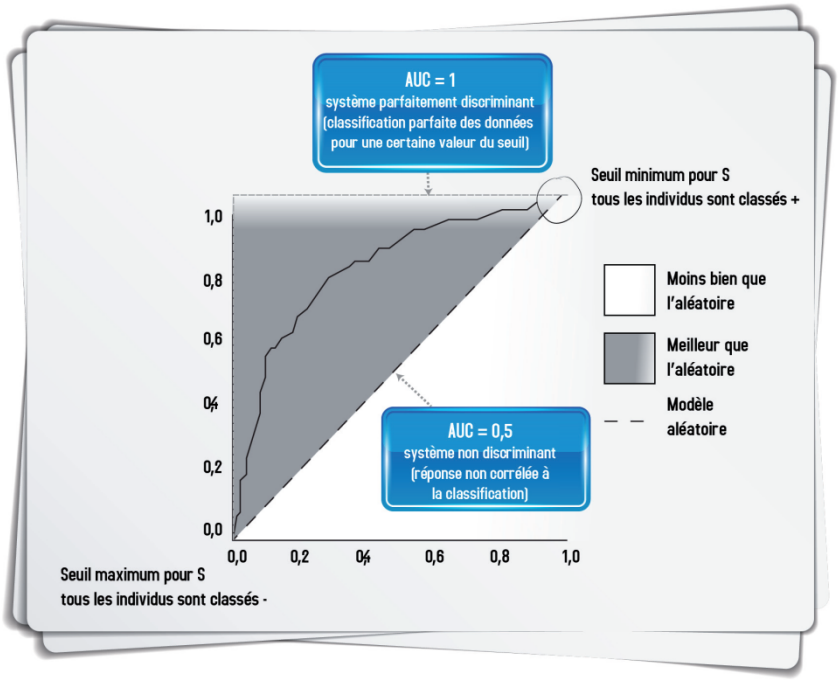

Figure 14-2 – *Courbe ROC*

En pratique, un classifieur va produire des courbes intermédiaires, entre le modèle non discriminant et le modèle parfaitement discriminant. L'analyse de la courbe va permettre de choisir le seuil de décision optimal. Ce choix peut se faire analytiquement, mais on peut tout aussi bien appliquer une règle empirique simple : le seuil optimal est celui qui est au point le plus proche de l'idéal (1,1) et au plus loin de la diagonale.

Interprétation probabiliste de la courbe ROC

La courbe permet une analyse probabiliste de la classification. Considérons les observations comme issues de deux populations : celle des cas positifs et celle des cas négatifs, chacun étant caractérisée par une distribution donnée. Fixons alors un seuil de décision : au-delà de ce seuil, une observation est considérée comme positive, en deçà, une observation est considérée comme négative (figure 14-3).

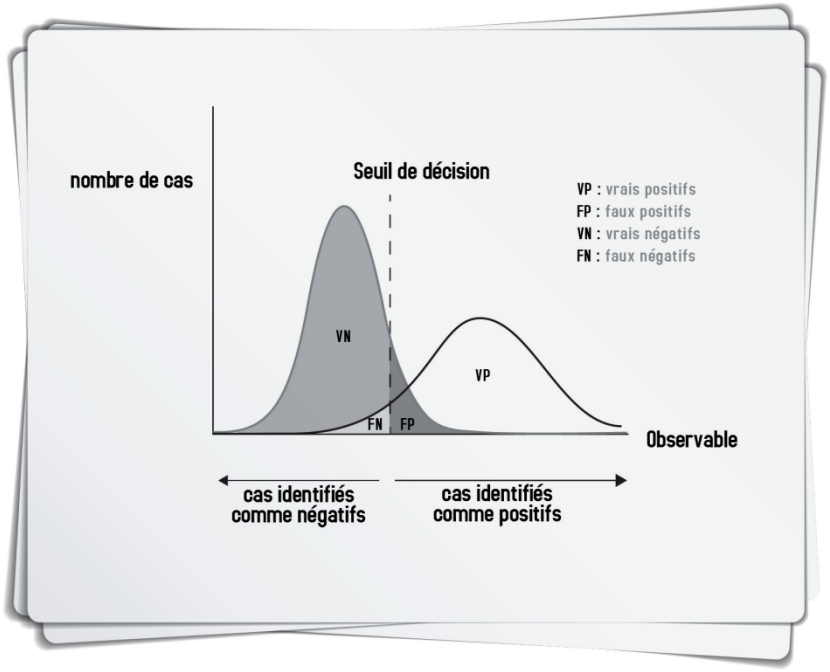

Figure 14-3 – *Impact du seuil de décision sur les vrais/faux positifs et les vrais/faux négatifs*

En faisant varier ce seuil, on modifie la sensibilité et la spécificité du modèle. Plus le seuil est bas, plus la sensibilité est élevée au détriment de la spécificité (on déclarera plus de vrais positifs, au risque de produire plus de faux positifs également), et inversement.

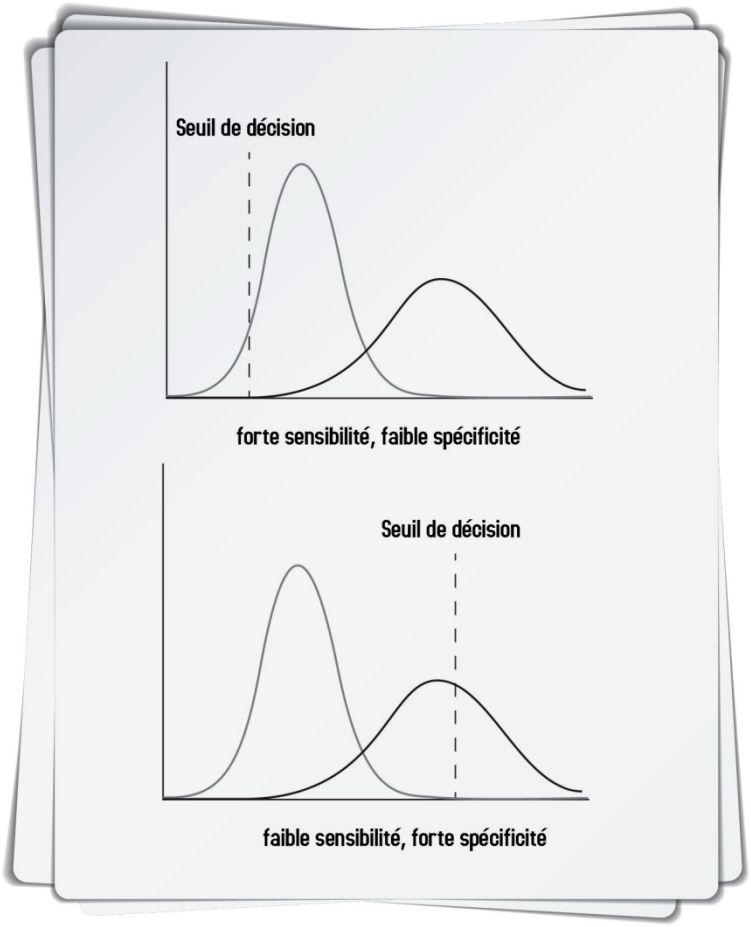

Figure 14-4 – *Impact du seuil de décision sur la sensibilité et la spécificité du modèle*

On peut faire un parallèle avec la théorie des tests statistiques : en augmentant la sensibilité, on diminuera le risque de première espèce, mais on augmentera le risque de deuxième espèce[3].

Pour les lecteurs qui aimeraient se faire une idée intuitive du lien entre cette interprétation probabiliste et la courbe ROC, nous leur recommandons de jouer avec le simulateur mis à disposition ici : http://www.navan.name/roc.

Comparaison de modèles avec la courbe ROC

Le grand attrait de la courbe ROC est qu'elle offre un cadre commun qui permet de comparer des modèles de différents types. Cette comparaison peut se faire localement (pour certains seuils de décision donnés), ou globalement, quel que soit le seuil de décision. Pour ce dernier cas, on considère la surface sous la courbe ROC (l'AUC pour les intimes, *Area Under the Curve*). Pour les matheux, cela signifie :

$$AUC = \int_{s=+\infty}^{s=-\infty} \left(1-\beta(s)\right) d\alpha(s)$$

La comparaison est d'une simplicité déconcertante : plus grande est l'AUC, meilleur est le modèle. Cette valeur s'interprète en effet comme la probabilité de classer un exemple positif choisi au hasard comme positif. Les statisticiens effectueront cette comparaison par l'intermédiaire d'un test, en comparant le rapport entre la différence entre deux AUC et l'écart-type de cette différence à une loi normale, mais nous ne développerons pas cela ici. Les lecteurs statisticiens seront aussi très excités en apprenant que l'AUC est directement liée à la (relativement) célèbre statistique de Wilcoxon (ou Mann-Whitney). Mais nous commençons à déborder du périmètre de ce livre, il est donc temps de clore ce chapitre !

À RETENIR Évaluer un modèle

Pour définir le meilleur modèle possible, on procède par validation croisée. Ceci consiste à diviser les données en trois sous-ensembles :

- un jeu d'entraînement pour entraîner plusieurs modèles ;
- un jeu de validation pour tester les modèles et sélectionner le meilleur modèle ;
- un jeu de test pour évaluer la performance finale du meilleur modèle.

Diverses méthodes existent pour exploiter au mieux les données, afin de constituer plusieurs jeux d'entraînement et de validation.

La quantification de la performance des modèles s'appuie sur diverses métriques, selon que l'on est en présence d'un problème de régression (RMSE, R^2) ou de classification (F1 et surtout courbe ROC).

Références

Pour une excellente revue des techniques de validation croisée, jetez-vous vite sur cet article :

- Arlot S., Celisse A. 2010. A survey of cross-validation procedures for model selection. *Statistics Survey*, 4, p. 1-274.

Si vous souhaitez en savoir plus au sujet de l'usage du *bootstrap* dans le cadre de la validation croisée, n'hésitez pas à lire deux stars de la statistique :

- Efron B., Tibshirani R. 1997. Improvements on cross-validation. The .632+ bootstrap method. *Journal of the American Statistical Association*, 92:438, p. 548-560.