

Régression logistique: prédiction et classification

Analyse multidimensionnelle appliquée

Léo Belzile

HEC Montréal

automne 2022

La régression logistique spécifie un modèle pour la probabilité de succès

$$p = \Pr(Y = 1 \mid \mathbf{x}) = \frac{1}{1 + \exp(-\eta)}$$

où $\eta = \beta_0 + \dots + \beta_p x_p$.

En substituant l'estimation $\hat{\beta}_0, \dots, \hat{\beta}_p$, on calcule

- le prédicteur linéaire $\hat{\eta}_i$ et
- la probabilité de succès \hat{p}_i

pour chaque ligne de la base de données.

Choisir un point de coupure c :

- si $\hat{p} < c$, on assigne $\hat{Y} = 0$.
- si $\hat{p} \geq c$, on assigne $\hat{Y} = 1$.
- Un point de coupure de $c = 0.5$ revient à assigner l'observation à la classe (catégorie) la plus probable.
- Si $c = 0$, on catégorise toutes les observations en succès avec $\hat{Y}_i = 1$ ($i = 1, \dots, n$).

L'erreur quadratique pour un problème de classification est

$$(Y - \hat{Y})^2 = \begin{cases} 1, & Y \neq \hat{Y}; \\ 0, & Y = \hat{Y}. \end{cases}$$

et donc on obtient le **taux de mauvaise classification** si on calcule la moyenne.

Plus le taux de mauvaise classification est petit, meilleure est la capacité prédictive du modèle.

Utiliser les mêmes données pour l'ajustement et l'estimation de la performance n'est (toujours) pas recommandé.

Plutôt, considérer la validation croisée ou la division de l'échantillon.

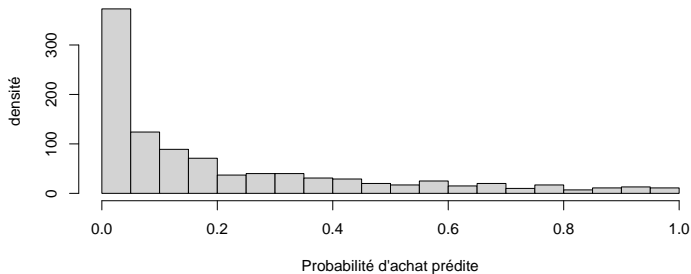
On considère un modèle pour yachat, le fait qu'une personne achète suite à l'envoi d'un catalogue.

```
1 data(dbm, package = "hecmulti")
2 formule <- formula("yachat ~ x1 + x2 + x3 +
3                   x4 + x5 + x6 + x7 + x8 + x9 + x10")
4 dbm_class <- dbm |>
5   dplyr::filter(test == 0) |>
6   # pour caret, convertir 0/1 en facteurs
7   dplyr::mutate(yachat = factor(yachat))
```

Estimation avec validation croisée

On utilise la fonction `train` du paquet `caret`, avec le modèle linéaire généralisé.

```
1 set.seed(202209)
2 cv_glm <-
3   caret::train(form = formule,
4                 data = dbm_class,
5                 method = "glm",
6                 family = binomial(link = "logit"),
7                 trControl = caret::trainControl(
8                   method = "cv",
9                   number = 10))
```

Répartition des probabilités de succès prédites par validation croisée.

On peut varier le point de coupure et regarder pour chaque valeur de c la classification résultante.

```
1 # predict retourne une matrice n x 2
2 # avec [P(Y=0), P(Y=1)]
3 predprob <- predict(cv_glm, type = "prob")[,2]
4 classif <- with(dbm, yachat[test == 0])
5 # Tableau de la performance
6 hecmulti::perfo_logistique(
7   prob = predprob,
8   resp = classif)
```

On peut classifier les observations dans un tableau pour un point de coupure donné.

Table 1: Matrice de confusion avec point de coupure 0.465.

	$Y = 1$	$Y = 0$
$\hat{Y} = 1$	109	52
$\hat{Y} = 0$	101	738

Classification et mesures de performance

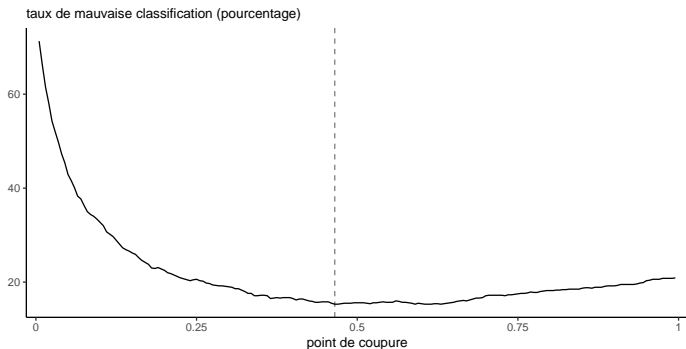
Les estimés empiriques sont simplement obtenus en calculant les rapports du nombre d'observations dans chaque classe.

	$Y = 1$	$Y = 0$		$Y = 1$	$Y = 0$
$\hat{Y} = 1$	109	52	$\hat{Y} = 1$	VP	FP
$\hat{Y} = 0$	101	738	$\hat{Y} = 0$	FN	VN

- La **sensibilité** est le taux de succès correctement classés, $\Pr(Y = 1, \hat{Y} = 1 \mid Y = 1)$, soit $VP / (VP + FN)$.
- La **spécificité** est le taux d'échecs correctement classés, $\Pr(Y = 0, \hat{Y} = 0 \mid Y = 0)$, soit $VN / (VN + FP)$.
- Le taux de **faux positifs** est $\Pr(Y = 0, \hat{Y} = 1 \mid \hat{Y} = 1)$.
- Le taux de **faux négatifs** est $\Pr(Y = 1, \hat{Y} = 0 \mid \hat{Y} = 0)$.

Choix d'un point de coupure.

On peut faire varier le point de coupure et choisir celui qui maximise le taux de bonne classification, $\widehat{\Pr}(Y = \hat{Y})$.



Avec $c = 0.465$, on obtient un taux de mauvaise classification de 15.3%.

Il est également possible d'assigner un poids différent à chaque événement selon le scénario et chercher à maximiser le gain.

Table 2: Matrice de gain (cas général)

	$Y = 1$	$Y = 0$
$\hat{Y} = 1$	c_{11}	c_{10}
$\hat{Y} = 0$	c_{01}	c_{00}

On calcule le gain en faisant la somme des entrées fois les poids, soit

$$\text{gain} = c_{11}\text{VP} + c_{10}\text{FP} + c_{01}\text{FN} + c_{00}\text{VN}.$$

Si on cherche à maximiser le taux de bonne classification, cela revient à assigner les poids suivants.

Table 3: Matrice de gain pour le taux de bonne classification.

	$Y = 1$	$Y = 0$
$\hat{Y} = 1$	1	0
$\hat{Y} = 0$	0	1

Coûts et bénéfices du ciblage marketing

- Si on n'envoie pas de catalogue, notre gain est nul.
- Si on envoie le catalogue
 - à un client qui n'achète pas, on perd 10\$ (le coût de l'envoi).
 - à un client qui achète, notre revenu net est de 57\$ (revenu moyen moins coût de l'envoi).

n	moyenne	écart-type	minimum	maximum
210	67.29	13.24	25	109

Statistiques descriptives des montants d'achats pour la base de données marketing.

Table 4: Matrice de gain pour ciblage marketing.

	$Y = 1$	$Y = 0$
$\hat{Y} = 1$	57	-10
$\hat{Y} = 0$	0	0

Point de coupure avec gain

```
1 formule = formula(yachat ~ x1 + x2 + x3 +  
2               x4 + x5 + x6 + x7 +  
3               x8 + x9 + x10)  
4 modele <- glm(formule,  
5               family = binomial,  
6               data = hecmulti::dbm)  
7 coupe <- hecmulti::select_pcoupe(  
8   modele = modele,  
9   c00 = 0,  
10  c01 = 0,  
11  c10 = -10,  
12  c11 = 57,  
13  plot = TRUE)
```

La fonction `select_pcoupe` estime le gain pour différents points de coupures, avec probabilités estimées par validation croisée avec `ncv` groupes, répétée `nrep` fois.

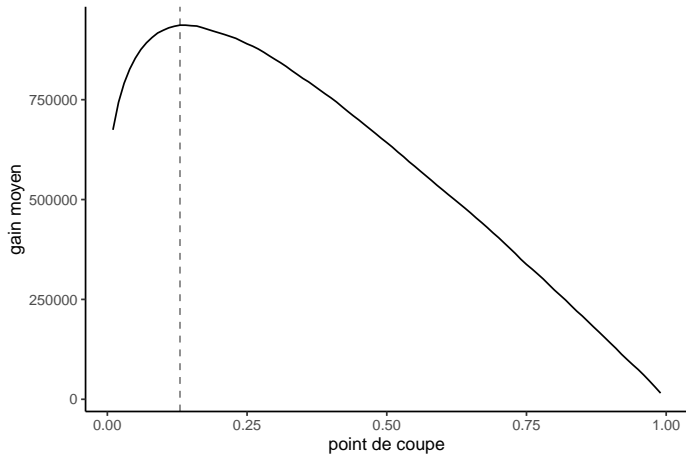


Figure 1: Estimation du gain moyen en fonction du point de coupure pour l'exemple de base de données marketing.

Dans l'exemple, le point de coupure qui maximise le gain est 0.13. Avec ce point de coupure, on estime que

- le taux de bonne classification est de 70.3
- la sensibilité est de 90.95.

Ainsi, on va détecter environ 90.95% des clients qui achètent.

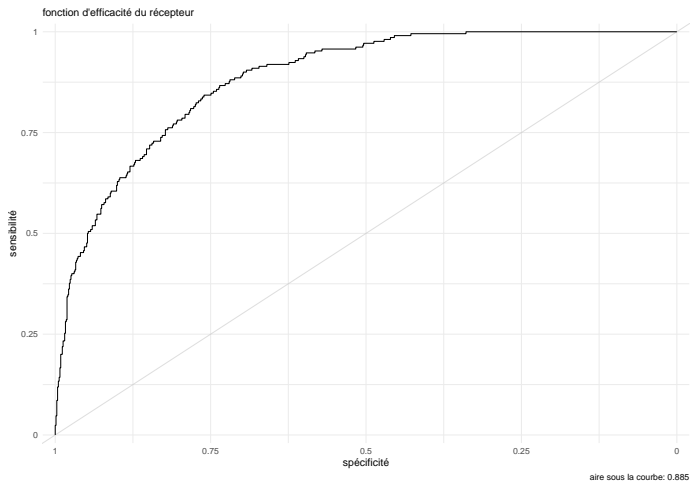
Il est coûteux de rater un client potentiel, donc la stratégie optimale est d'envoyer le catalogue à plus de clients quitte à ce que plusieurs d'entre eux n'achètent rien.

Graphique de la sensibilité en fonction de un moins la spécificité, en faisant varier le point de coupure, souvent appelé courbe ROC (de l'anglais *receiver operating characteristic*).

La fonction `hecmulti::courbe_roc` permet de tracer la courbe et de calculer l'aire sous la courbe.

```
1 roc <- hecmulti::courbe_roc(  
2   resp = classif,  
3   prob = predprob,  
4   plot = TRUE)  
5 print(roc)  
6 ## Pour extraire l'aire sous la courbe, roc$aire
```

Courbe ROC



- Plus la courbe se rapproche de $(0, 1)$ (coin supérieur gauche), meilleure est la classification.
- Autrement dit, plus l'aire sous la courbe est près de 1, mieux c'est.
- Une aire sous la courbe de 0.5 (ligne diagonale) correspond à la performance d'une allocation aléatoire.

À quelle point notre modèle est meilleur qu'une assignation aléatoire?

- Ordonner les probabilités de succès estimées par le modèle, \hat{p} , en ordre croissant.
- Regarder quelle pourcentage de ces derniers seraient bien classifiés (le nombre de vrais positifs sur le nombre de succès). La référence est la ligne diagonale, qui correspond à une détection aléatoire.

Code pour produire la courbe lift

```
1 tab_lift <- hecmulti::courbe_lift(  
2   prob = predprob,  
3   resp = classif,  
4   plot = TRUE)  
5 tab_lift
```


Courbe lift

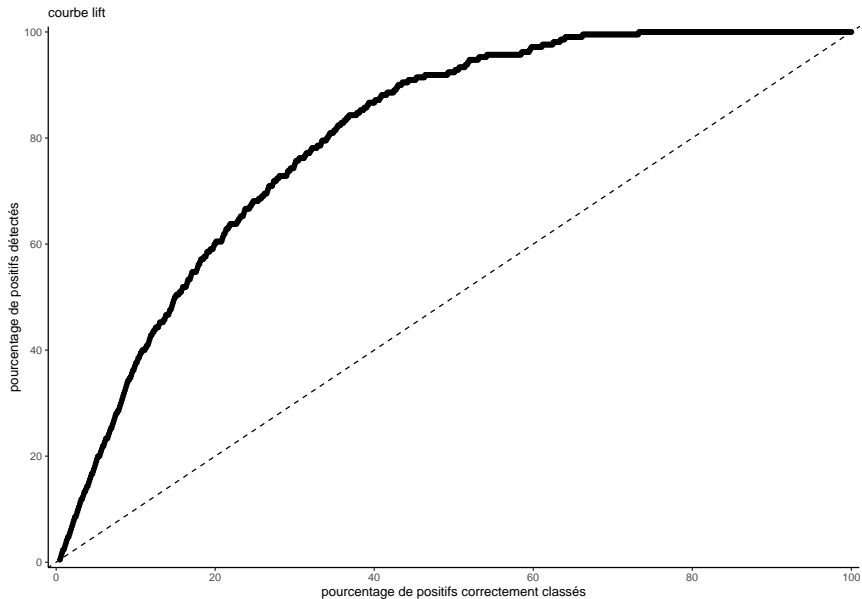


Table 5: Tableau du lift (déciles).

	hasard	modèle	lift
10%	21	81	3.86
20%	42	127	3.02
30%	63	159	2.52
40%	84	183	2.18
50%	105	195	1.86
60%	126	204	1.62
70%	147	209	1.42
80%	168	210	1.25
90%	189	210	1.11

Si on classifiait comme acheteurs les 10% qui ont la plus forte probabilité estimée d'achat, on détecterait 81 des 210 clients.

Le lift est le nombre détecté par le modèle sur la proportion détectée par hasard.

Certains modèles sont trop confiants dans leurs prédictions (surajustement).

une statistique simple proposée par Spiegelhalter (1986) peut être utile à cette fin. Pour une variable Bernoulli $Y \in \{0, 1\}$, l'erreur quadratique moyenne s'écrit

$$\begin{aligned}\overline{B} &= \frac{1}{n} \sum_{i=1}^n (Y_i - p_i)^2 \\ &\quad \text{erreur quadratique moyenne} \\ &= \frac{1}{n} \sum_{i=1}^n (Y_i - p_i)(1 - 2p_i) + \frac{1}{n} \sum_{i=1}^n p_i(1 - p_i). \\ &\quad \text{manque de calibration} \qquad \qquad \text{variabilité}\end{aligned}$$

Si notre modèle était parfaitement calibré, $E_0(Y_i) = p_i$ et $\text{Va}_0(Y_i) = p_i(1 - p_i)$.

On peut calculer la moyenne et la variance attendue sous l'hypothèse nulle et construire une statistique de test (Spiegelhalter, 1986).

$$Z = \frac{\overline{B} - E_0(\overline{B})}{sd_0(\overline{B})}$$

Sous l'hypothèse nulle de calibration parfaite, $Z \sim \text{No}(0, 1)$ en grand échantillon.

```
1 hecmulti::calibration(  
2   prob = predprob,  
3   resp = classif)
```

Test de calibration de Spiegelhalter (1986)

Statistique de test: 0.28

valeur-p: 0.782

Il n'y a pas de preuve ici que le modèle est mal calibré.