Compte-rendu du TP04 de SY09
Discrimination

NGO Sy-Toan, Elliot BARTHOLME

28 juin 2017

1 Introduction

Ce quatrième et dernier Travail Pratique dans l'UV SY09 aborde plusieurs méthodes de discrimination et de régression, dans la continuité du TP03.

1.1 Analyses discriminantes quadratique et linéaire

Les premières qui seront abordées sont l'Analyse Discriminante Linéaire (*ADL*), l'Analyse Discriminante Quadratique (*ADQ*) et le Classifieur Bayésien Naïf (*NBA*).
L'utilisation de ces méthodes de discrimination repose sur la supposition suivante : le vecteur de caractéristiques $\boldsymbol{X}$ suit conditionnellement à chaque classe ω_k une loi normale multidimensionnelle d'espérance $\boldsymbol{\mu}_k$ et de variance Σ_k.
En effectuant différentes hypothèses sur les paramètres de ces lois, notamment sur les matrices de variance, on obtient différentes expressions de la règle de Bayes (*cf. TP03*), grâces auxquelles on déduit des règles de décision en remplaçant les paramètres théoriques par leurs estimations.
Ces méthodes présentent donc d'autres modèles d'apprentissage grâce auxquels on peut prédire l'appartenance à une classe (variable à expliquer) avec une approche probabiliste.

1.2 Régression logistique

Nous utiliserons aussi la méthode de régression logistique (modèle binaire) ainsi que la régression logistique quadratique. Ces méthodes constituent des modèles de régression binomiale. Il s'agit aussi de modéliser l'effet d'un vecteur de variables aléatoires (caractéristiques) sur une variable binomiale à prédire (la classe de l'individu), mais contrairement aux modèles de fonctions discriminantes qui font des hypothèses sur la distribution conditionnelle de chaque classe, on chercher ici à estimer directement les probabilités a posteriori d'appartenance aux classes.

1.3 Arbres binaires

Finalement, nous terminerons avec l'étude des arbres de décision binaires, qui permettent de résoudre aussi bien des problèmes de discrimination que de régression. Ces méthodes consistent à partitionner de manière récursive l'espace des caractéristiques en régions homogènes, c'est-à-dire qui ne contiennent qu'une seule valeur de la variable Z à expliquer.

Ces différents modèles de classification supervisée seront, comme dans le *TP03*, testés sur plusieurs échantillons de données simulées, puis sur des données réelles. L'on s'attachera à l'interprétation des différents taux d'erreurs de chacun.

2 Programmation

Il s'agit ici de définir les modèles d'apprentissage présentés précédemment, avec R. Nous définissions pour cela les fonctions *adl.app*, *adq.app*, *nba.app*, *log.app*, *log.app*, *tree.app* qui permettent d'apprendre les paramètres de chacun sur un jeu de données.

Les analyses discriminantes supposent une répartition des données conditionnellement à chaque classe sur le modèle gaussien. Ainsi, on a deux densités $f_1(x)$ et $f_2(x)$ qui correspondent à la densité d'une réalisation x d'une observation sachant que l'on appartienne à la classe 1 ou 2. Chacune de ces lois de probabilité $f_k(x)$ possède donc des paramètres μ_k et Σ_k selon ω_k. En fonction des hypothèses faites sur ces paramètres et sur leurs estimations, on peut obtenir différentes expressions de la règle de Bayes qui pour rappel minimise la probabilité d'erreur d'une décision en affectant un individu x à la classe de plus grande probabilité à posteriori :

$$\delta^*(x) = \begin{cases} a_1 & \text{si } \mathbb{P}(\omega_2|x) < \mathbb{P}(\omega_1|x), \\ a_2 & \text{sinon.} \end{cases} \tag{1}$$

2.1 Analyse discriminante quadratique

Rappelons la densité conditionnelle $f_k(x)$:

$$f_k(x) = \frac{1}{(2\pi)^{\frac{p}{2}}(det\Sigma_k)^{\frac{1}{2}}} exp(-\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)) \tag{2}$$

La règle de Bayes s'écrit $\delta^*(x) = a_{k^*}$ avec

$$k^* = arg\max_k \mathbb{P}(\omega_k|x) = arg\max_k g_k(x), \tag{3}$$

et

$$g_k(x) = lnf_k(x) + ln\pi_k = -\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k) - \frac{1}{2}ln(det\Sigma_k) + ln\pi_k - \frac{p}{2}ln(2\pi). \tag{4}$$

Cette fonction $g_k(x)$ sert à définir la règle de décision que l'on appelle fonction discriminante. On a ici des formes quadratiques : les régions de décision associées à chaque classe sont séparées par des frontières d'équations $g_k(x) = g_l(x)$ lorsque $k \neq l$.

adq.app est donc une fonction qui détermine les estimateurs (sans biais) du maximum de vraisemblance des paramètres.

2.2 Analyse discriminante linéaire

En faisant une hypothèse d'homoscédasticité, c'est-à-dire que les matrices de variance sont communes à toutes les classes, on obtient des fonctions discriminantes linéaires telles que :

$$h_k(x) = (\Sigma^{-1}\mu_k)^T x - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k + ln\pi_k \tag{5}$$

La règle de Bayes devient une règle de décision linéaire avec une frontière de décision qui est un hyperplan qui passe par le centre du segment $[\mu_k, \mu_l]$ si $\pi_k = \pi_l$. On a en fait une généralisation du classifieur euclidien vu au TP03 avec la distance de Mahalanobis qui est une distance affectant moins de poids au variables dont la variance est forte.

adl.app est donc la fonction qui détermine les estimateurs sans biais.

2.3 Classifieur bayésien naïf

Ici on fait l'hypothèse que les variables sont indépendantes entre elles conditionnellement à la classe Z, on a alors des matrices Σ_k diagonales. Dans ce cas on obtient une variante de l'ADQ, qu'il est même possible de conjuguer avec l'hypothèse d'homoscédasticité.

On utilisera pour ce modèle la fonction *nba.app*.

En utilisant chacun de ces modèles, on peut déterminer les résultats et probabilités à posteriori sur un ensemble d'individus avec la fonction *ad.val*. On rappelle que la probabilité à postériori d'appartenance à une classe ω_k s'exprime avec la relation suivante :

$$\mathbb{P}(Z = \omega_k | X = x) = \frac{f_k(x)}{f(x)} = \frac{f_k(x)}{\pi_1 f_1(x) + \pi_2 f_2(x)} \tag{6}$$

2.4 Régression logistique et régression logistique quadratique

Les analyses discriminantes quadratiques et linéaires, ainsi que le classifieur bayésien naïf, consistent à utiliser les fonctions de densités $f_k(x)$ dans lesquelles on remplace les paramètres par leurs estimateurs du maximum de vraisemblance (EMV). On utilise ensuite les valeurs des fonctions de densité pour calculer les probabilités a posteriori et ainsi affecter x à la classe ayant la probabilité la plus élevée. La régression logistique consiste elle à estimer directement les probabilités $\mathbb{P}(Z = \omega_k | X = x)$ d'appartenance aux classes.

On implémente une fonction *log.app* chargée d'apprendre les paramètres du modèle et une fonction *log.val* chargée de classer un ensemble d'individus en fonction des résultats de *log.app* (probabilités estimées les plus fortes). Dans ce TP, nous utiliserons le *modèle logit* qui consiste à exprimer le logarithme du ratio des probabilités à posteriori $\mathbb{P}(\omega_k|x)$ et $\mathbb{P}(\omega_g|x)$ comme un fonction de x pour tout $k = 1, ..., g-1$:

$$ln(\frac{\mathbb{P}(\omega_k|x)}{\mathbb{P}(\omega_g|x)}) = w_k^T x$$

Le modèle de régression logistique quadratique généralise le concept précédant en l'apprenant sur un espace de plus haute dimension contenant les données initiales. Par exemple,

$$X = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

deviendra :

$$X_2 = \begin{pmatrix} 1 & 2 & 2 & 1 & 4 \\ 3 & 4 & 12 & 9 & 16 \end{pmatrix}$$

L'espace ainsi obtenu est de dimension $\frac{p(p+3)}{2}$.

2.5 Arbres

Lors de la construction d'arbres de décision binaires, il est important d'avoir une phase de régularisation souvent appelée *élagage* qui permet de les simplifier afin d'éviter les problèmes d'adaptation trop forte aux données d'apprentissage, ce qui traduit une situation *d'overfitting*.

On peut calculer la pénalisation de la complexité avec la formule suivante :

$$\eta\lambda(A) = \varepsilon(A) + \lambda\xi(A) \tag{7}$$

λ est un terme de compromis entre le taux d'erreur ε par rapport à la complexité en séparations ξ ; le but étant de rechercher les nœuds qui n'améliorent pas assez l'erreur commise par rapport à la complexité qu'ils engendrent.

Cette régularisation peut se faire avec R grâce à la fonction *prune.misclass*.

Les arbres fournissent des frontières de décision caractéristiques en forme d'escaliers (combinaisons de frontières linéaires suivant chaque variable). Nous avons réussi à les dessiner grâce aux librairies *plyr*, *rpart* et *ggplot2*.

3 Application

3.1 Test sur données simulées

Dans cette partie les données que nous utilisons sont des données simulées selon une distribution normale multivariée déterminée. Le protocole est absolument le même qu'au dernier TP, on rappelle que le taux d'erreur suit une loi normale : $\varepsilon \sim \mathcal{N}(\mu, \sigma^2)$. μ et σ étant inconnus, l'intervalle de confiance sur ε s'écrit :

$$I_c = \left[\overline{\varepsilon} - t_{1-\alpha/2} \frac{s^*}{\sqrt{N}}, \overline{\varepsilon} + t_{1-\alpha/2} \frac{s^*}{\sqrt{N}} \right] \tag{8}$$

Les paramètres estimés pour chacune des classes sont les suivants pour chaque jeu de données :

	Synth1-1000	Synth2-1000	Synth3-1000
π_1	0.51	0.505	0.525
π_2	0.49	0.495	0.475
μ_1	$\begin{pmatrix} -1.03 \\ -1.95 \end{pmatrix}$	$\begin{pmatrix} -0.9 \\ -1.92 \end{pmatrix}$	$\begin{pmatrix} -1.06 \\ -1.91 \end{pmatrix}$
μ_2	$\begin{pmatrix} 1.1 \\ 2.07 \end{pmatrix}$	$\begin{pmatrix} 0.97 \\ 2.17 \end{pmatrix}$	$\begin{pmatrix} 0.92 \\ 2.19 \end{pmatrix}$

Voici les estimations de variance selon les différentes hypothèses faites avec chaque méthode :

Données	Sigma	ADQ	ADL	NBA
$Synth1_{1000}$	Σ_1	$\begin{pmatrix} 2.89 & -1.53 \\ -1.53 & 1.96 \end{pmatrix}$	$\begin{pmatrix} 2.5 & -0.78 \\ -0.78 & 2.38 \end{pmatrix}$	$\begin{pmatrix} 2.89 & 0 \\ 0 & 1.96 \end{pmatrix}$
	Σ_2	$\begin{pmatrix} 2.09 & 0 \\ 0 & 2.81 \end{pmatrix}$	$\begin{pmatrix} 2.5 & -0.78 \\ -0.78 & 2.38 \end{pmatrix}$	$\begin{pmatrix} 2.09 & 0 \\ 0 & 2.81 \end{pmatrix}$
$Synth2_{1000}$	Σ_1	$\begin{pmatrix} 2.81 & -0.19 \\ -0.19 & 0.90 \end{pmatrix}$	$\begin{pmatrix} 1.87 & -0.12 \\ -0.12 & 2.73 \end{pmatrix}$	$\begin{pmatrix} 2.81 & 0 \\ 0 & 0.9 \end{pmatrix}$
	Σ_2	$\begin{pmatrix} 0.90 & -0.04 \\ -0.04 & 4.6 \end{pmatrix}$	$\begin{pmatrix} 1.87 & -0.12 \\ -0.12 & 2.73 \end{pmatrix}$	$\begin{pmatrix} 0.9 & 0 \\ 0 & 4.60 \end{pmatrix}$
$Synth3_{1000}$	Σ_1	$\begin{pmatrix} 2.88 & -1.55 \\ -1.55 & 3.5 \end{pmatrix}$	$\begin{pmatrix} 2.90 & -1.78 \\ -1.78 & 3.81 \end{pmatrix}$	$\begin{pmatrix} 2.88 & 0 \\ 0 & 3.5 \end{pmatrix}$
	Σ_2	$\begin{pmatrix} 2.92 & -2.02 \\ -2.02 & 4.17 \end{pmatrix}$	$\begin{pmatrix} 2.90 & -1.78 \\ -1.78 & 3.81 \end{pmatrix}$	$\begin{pmatrix} 2.92 & 0 \\ 0 & 4.17 \end{pmatrix}$

TABLE 1 – Estimations des paramètres de distribution conditionnelle de chacune des classes de chaque jeu de données *Synth*

On voit bien l'hypothèse d'homoscédasticité sur l'ADL et l'indépendance des variables sur le NBA au niveau des matrices de covariance Σ_k.

Dans la suite, les résultats avec chacun des modèles seront fournis pour chaque jeu de données. On présentera les estimations du taux d'erreur de chaque méthode, ainsi que l'intervalle de confiance associé à ce taux. Quelques frontières de décision significatives seront tracées à l'aide des foncions *prob.ad* et *prob.log/prob.log2* ou de celles que nous avons implémentées pour les arbres. Elles permettent d'afficher la courbe de niveau de probabilité à posteriori $\hat{\mathbb{P}}(\omega_1|x)$ estimée. Nous choisirons à chaque fois un seuil de probabilité $\hat{\mathbb{P}}(\omega_1|x) = 0.5$, ce qui signifie que les individus au niveau de cette frontière ont autant de chance d'être placés dans la classe 1 que la classe 2 par la méthode de discrimination.

Notons que nous ne présenterons pas les valeurs de variance de l'estimation du taux d'erreur, celles -ci étant très faibles, on peut dire que les tests sont reproductibles (les valeurs de nos estimations ne varieront que très peu et sont donc viables pour nos interprétations).

3.1.1 Synth1-1000

Résultats

Voici l'estimation (moyenne sur $N = 20$) du taux d'erreur de chaque méthode de discrimination sur les données $Synth1 - 1000$:

	ADL	ADQ	NBA	LOG	LOGQ	TREE
Estimation de ε	0.041	0.034	0.040	0.032	0.036	0.049
Intervalle sur ε	[0.037, 0.045]	[0.030, 0.038]	[0.036, 0.045]	[0.028, 0.035]	[0.031, 0.041]	[0.043, 0.055]

TABLE 2 – Estimation du taux d'erreur et intervalle de confiance de ε, avec chacune des méthodes de discrimination ; données *Synth1-1000*.

Interprétation :

Les taux d'erreur sont relativement bas comparé au dernier TP (moins de 5%) car les distributions sont meilleures et les données nombreuses.
L'analyse des paramètres de distribution de ce premier jeu de données révèle qu'aucune des hypothèses (homoscédasticité et indépendance des variables) que font l'analyse discriminante linéaire et le classifieur bayésien naïf ne sont vérifiées. Il est donc naturel que l'analyse discriminante quadratique fournisse de meilleurs résultats ici.

The error test of tree is also small (less than 5%), however it is still the largest value of error among the 6 methods. Moreover, for the tree method, we take all the tree that construct and do not refine it. Therefore, that tree is very easy to get overfit on the apprentissage set. This indicates the mal-performence of the tree on the test set.

Frontières

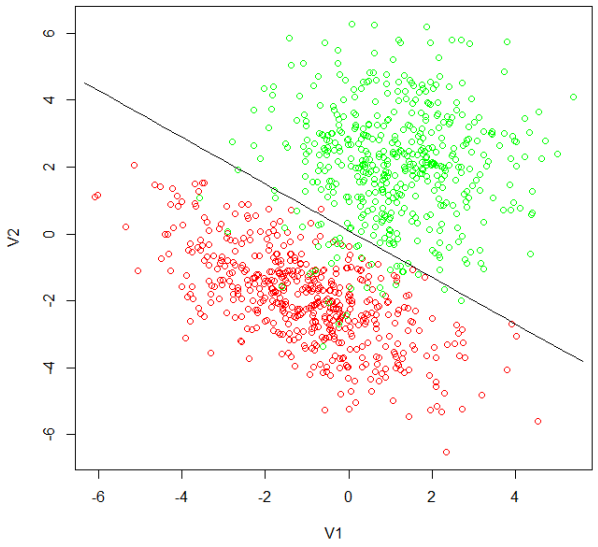

FIGURE 1 – ADL

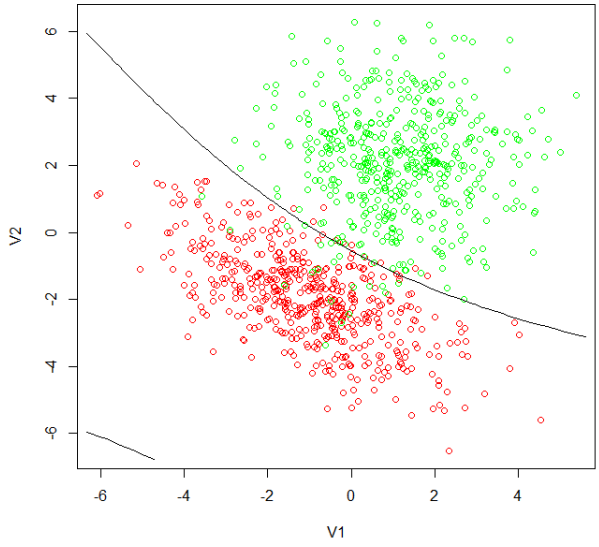

FIGURE 2 – ADQ

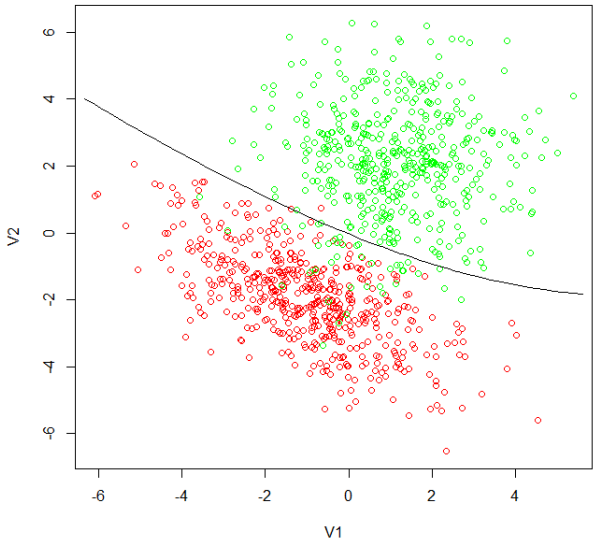

FIGURE 3 – NBA

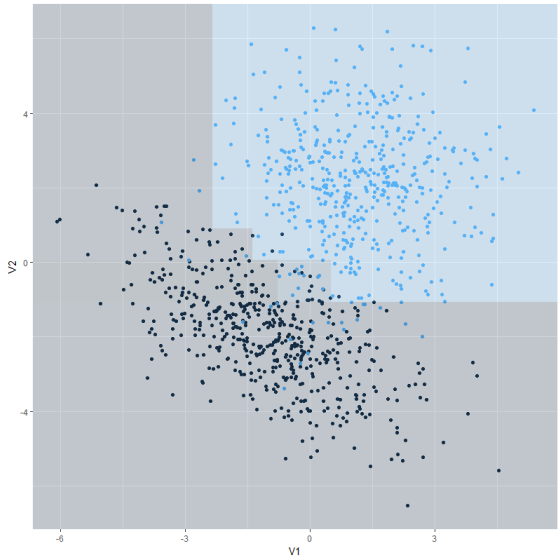

FIGURE 4 – TREE

On voit bien sur les figures 1 et 2 que l'ADQ prend mieux en compte la forme de chaque nuage de points, leurs inerties étant différentes, alors que l'ADL suppose le contraire.

3.1.2 Synth2-1000

Résultats

Voici l'estimation (moyenne sur $N = 20$) du taux d'erreur de chaque méthode de discrimination sur les données $Synth2 - 1000$:

	ADL	ADQ	NBA	LOG	LOGQ	TREE
Estimation de ε	0.078	0.064	0.062	0.073	0.068	0.080
Intervalle sur ε	[0.073, 0.082]	[0.058, 0.069]	[0.058, 0.066]	[0.067, 0.080]	[0.064, 0.072]	[0.074, 0.086]

TABLE 3 – Estimation du taux d'erreur et intervalle de confiance de ε, avec chacune des méthodes de discrimination ; données *Synth2-1000*.

Interprétations

Cette fois, on obtient après estimation par les *EMV*, sans hypothèse, deux matrices de covariance qui ne se ressemblent pas mais admettent des éléments non diagonaux très proches de zéro (covariances nulles). Il semble donc logique que l'analyse discriminante quadratique avec hypothèse d'indépendance (classifieur bayésien naïf) fonctionne bien, et c'est le cas : il offre le taux d'erreur le plus faible sur ces données.

The tree this time also gives back the bad result in comparaison with others methods. It is slightly as good as ADL.

Frontières

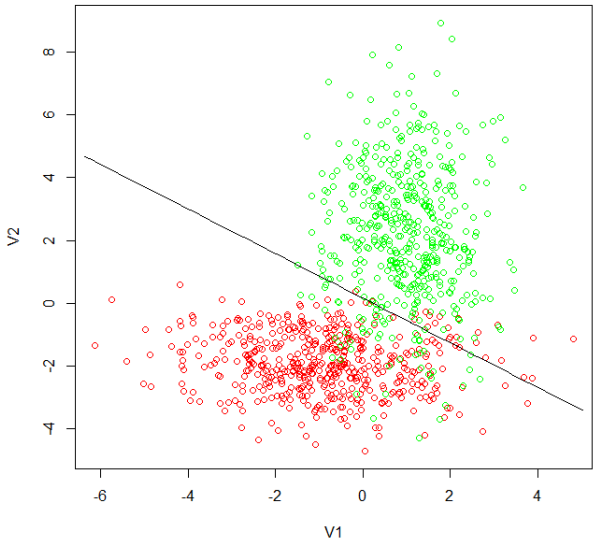

FIGURE 5 – ADL

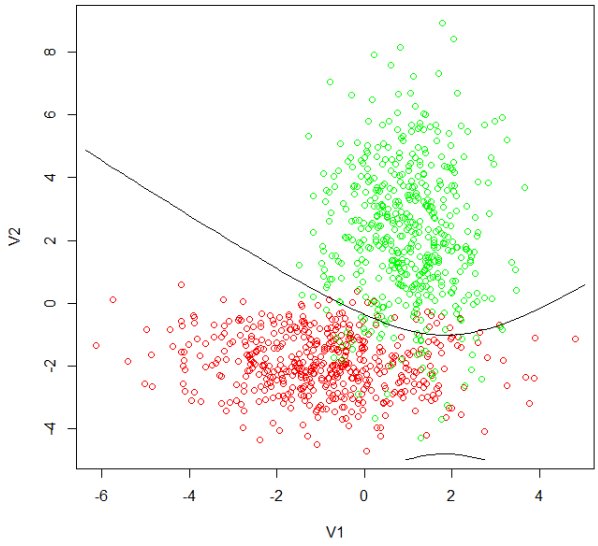

FIGURE 6 – NBA

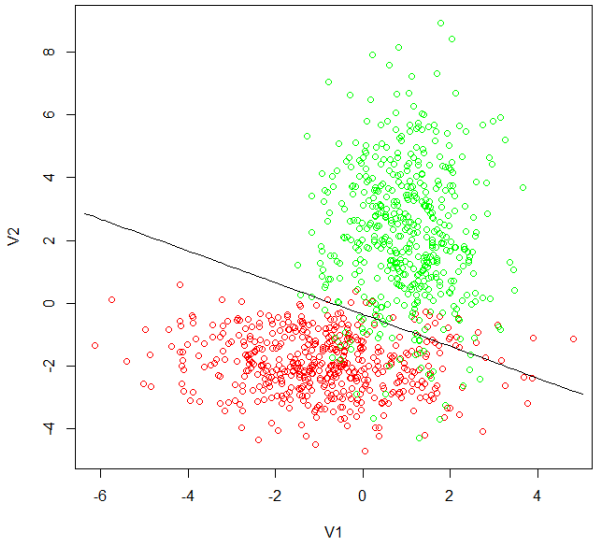

FIGURE 7 – LOG

FIGURE 8 – TREE

On voit bien sur la figure 6 que la frontière de forme concave permet d'épouser la forme des nuages qui sont orientés à l'inverse l'un de l'autre (ω_1 distribuée selon V_2 et ω_2 selon V_1), quand l'ADL n'offre qu'une approximation très moyenne puisqu'elle ne s'adapte pas bien à de telles distributions.

3.1.3 Synth3-1000

Résultats

Voici l'estimation (moyenne sur $N = 20$) du taux d'erreur de chaque méthode de discrimination sur les données $Synth3 - 1000$:

	ADL	ADQ	NBA	LOG	LOGQ	TREE
Estimation de ε	0.043	0.042	0.052	0.041	0.039	0.064
Intervalle sur ε	[0.039, 0.046]	[0.039, 0.046]	[0.046, 0.058]	[0.037, 0.046]	[0.034, 0.044]	[0.058, 0.069]

TABLE 4 – Estimation du taux d'erreur et intervalle de confiance de ε, avec chacune des méthodes de discrimination ; données *Synth3-1000*.

Interprétations

Finalement, pour ce troisième jeu de données simulées, on observe que les matrices de covariances conditionnelles aux classes sont très semblables. Une hypothèse d'homoscédasticité peut donc se faire, et l'analyse discriminante linéaire devrait être performante. C'est confirmé, puisqu'elle donne cette fois de bons résultats tout comme l'analyse quadratique, contrairement au classifieur bayésien naïf (variables non indépendantes conditionnellement aux classes...)

Les modèles de régression logistique fournissent aussi de bons résultats grâce à la séparation linéaire des deux nuages (*logit* : fonction linéaire qui suppose $\frac{\mathbb{P}(\omega_k|x)}{\mathbb{P}(\omega_g|x)} = exp(w_k^T x)$).

This time tree algorithm draws a better result in comparaison with synth1 and synth2, however it is still not as good as others methods.

Frontières

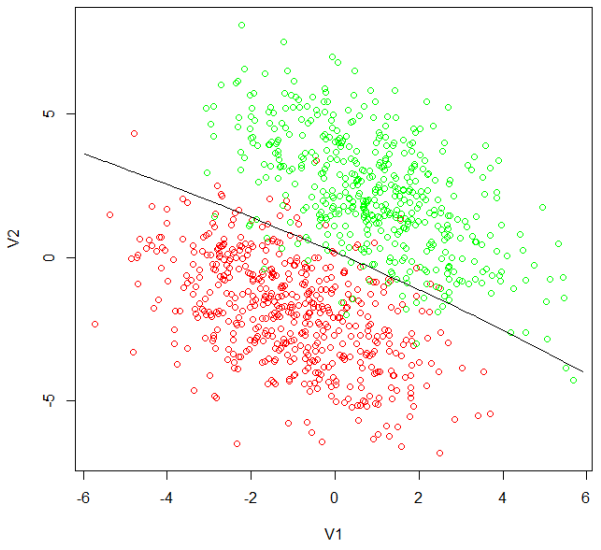

FIGURE 9 – NBA

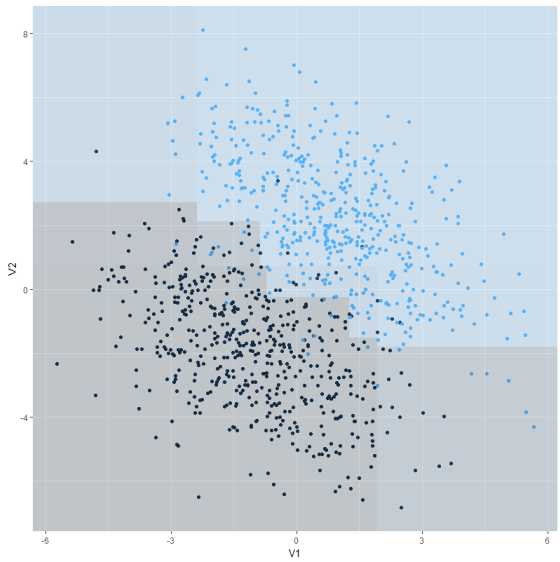

FIGURE 10 – TREE

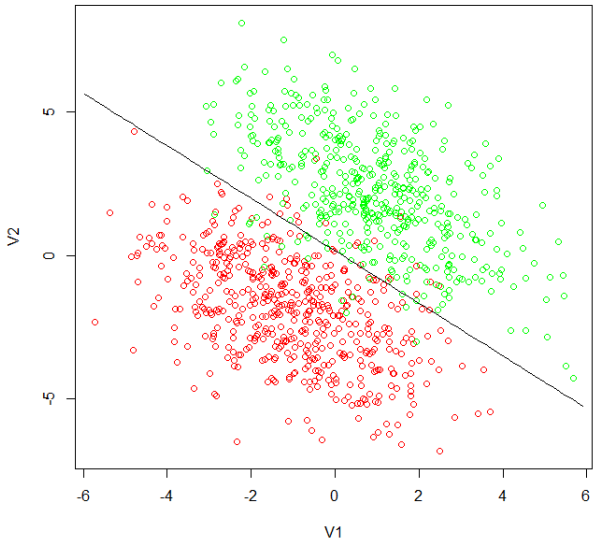

FIGURE 11 – LOG

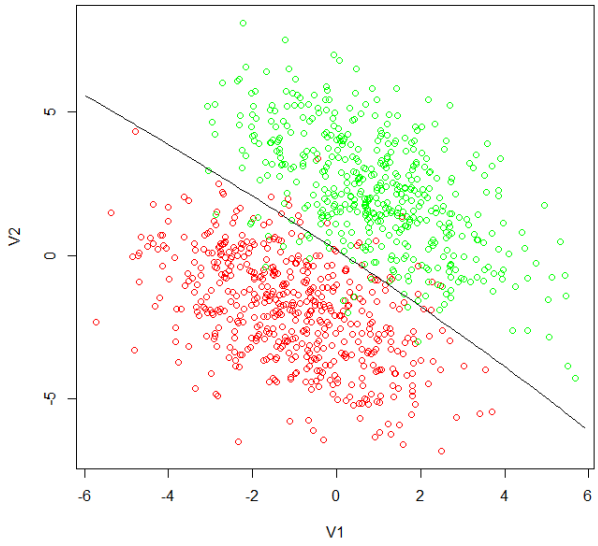

FIGURE 12 – ADQ

3.1.4 Conclusion de la discrimination sur données simulées

Dans cette première partie d'analyse de données *artificielles*, nous avons pu mettre en évidence l'influence des hypothèses sur les performances de chaque discriminant. Chacun possède ses particularités qui lui permettent de s'adapter préférablement à un jeu de données particulier.

De manière générale, les modèles réduisent de plus en plus le nombre de paramètres à estimer avec des hypothèses pour gagner en robustesse, mais exagérer ces hypothèses ou leur nombre conduit à des taux d'erreurs plus élevés. À l'inverse, choisir un modèle très paramétrique permet d'être beaucoup plus flexible et de s'adapter à des distributions plus atypiques.

On peut dire que globalement que l'ADL est plus robuste car il ne s'adaptera pas aux spécificités : il sera donc moins bon sur les données d'apprentissage. À contrario, l'ADQ donnera moins d'erreurs d'apprentissage car il s'adaptera plus au données utilisées grâce à son nombre de paramètre conséquent ; les prédictions sur des données de test non utilisées pour l'apprentissage du modèle seront en revanche bien plus complexes. La simplification du nombre de paramètres à estimer permet de s'affranchir des phénomènes de sur-apprentissage et donc d'obtenir de bons taux de discrimination si les hypothèses sont vérifiées.

L'idéal est donc de trouver un compromis avec un modèle de complexité adapté à la taille de l'ensemble d'apprentissage.

D'autres résultats intéressants peuvent être trouvés avec des tests sur la distribution des données : l'hypothèse de normalité des analyses discriminantes peut jouer sur le taux d'erreur si elle n'est pas vérifiée.

On peut réaliser pour cela un test de Mardia qui est un test empirique d'adéquation d'un *set* de données avec une distribution normale multivariée. Son hypothèse nulle est : *les données suivent une loi multivariée*. Pour des *p-value* suffisamment petites on rejette donc cette hypothèse :

— $Synth1_{1000}$: $p\text{-}value = 2.84e-33$
— $Synth2_{1000}$: $p\text{-}value = 5.34-73$
— $Synth3_{1000}$: $p\text{-}value = 0.39$

À chaque fois les données sont déclarées comme ne suivant pas une distribution multivariée normale :

```
   Mardia's Multivariate Normality Test
---------------------------------------
   data : donn

   g1p            : 0.06357995
   chi.skew       : 10.59666
   p.value.skew   : 0.3897925

   g2p            : 12.72349
   z.kurtosis     : -6.571708
   p.value.kurt   : 4.974132e-11

   chi.small.skew : 10.6444
   p.value.small  : 0.385885

   Result          : Data are not multivariate normal.
---------------------------------------
```

FIGURE 13 – Résultat d'un test de normalité de Mardia

Malgré une simulation de distribution les critères de loi multivariée n'ont pas pu être atteints. Ce qui est probant, c'est que la *p-value* de $Synth2_1000$, est largement la plus petite et le taux d'erreur associé est le plus grand : on peut éventuellement supposer que l'hypothèse de normalité possède bien une influence.

3.2 Test sur des données réelles

Dans cette partie on abordera l'étude de données réelles associées à des problématiques (diabète, prédiction du cancer du sein).

3.2.1 Pima

Nous commençons par regarder la distribution des données conditionnellement à chaque classe et chaque variable des données Pima :

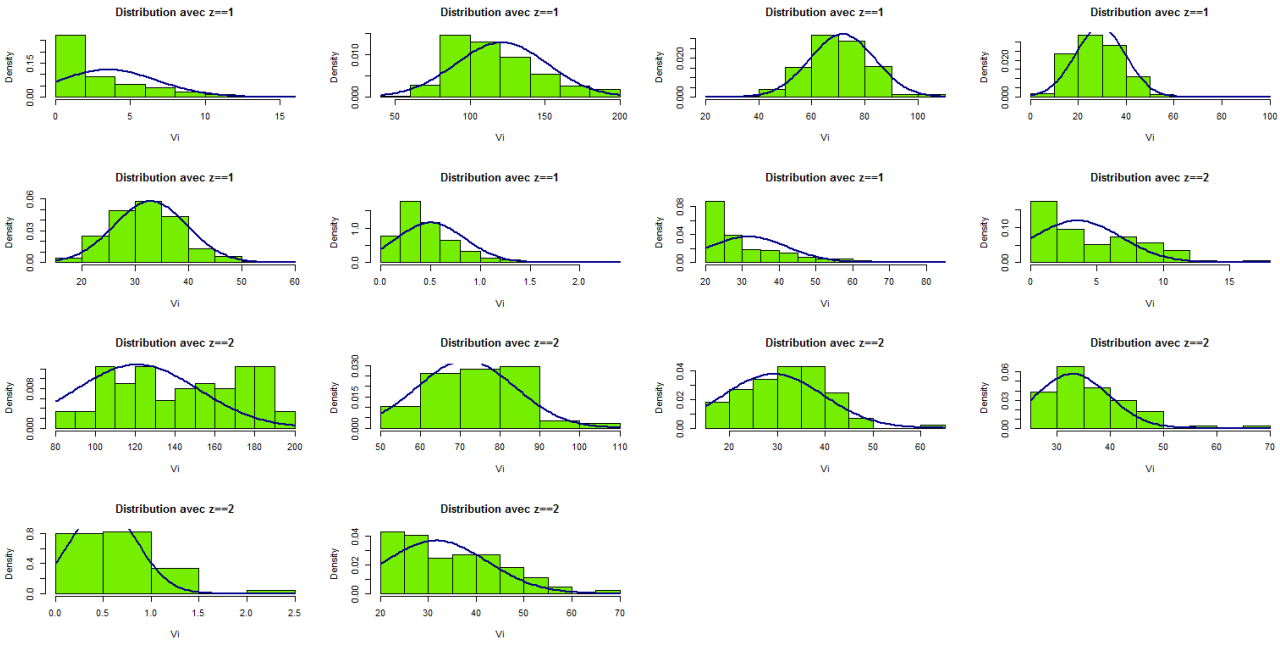

FIGURE 14 – Distribution de chaque variable de Pima conditionnellement aux classes

Ces distributions révèlent parfois une normalité unidimensionnelle de chaque variable numérique, cependant on ne peut pas grâce à cette méthode vérifier la *multinormalité*. Le test de Mardia est justement formel : on ne se trouve pas dans le cas d'une distribution multivariée normale ($p\text{-}value = 2.25e-198$).

Si la plupart de nos modèles d'analyse la supposent, ce n'est cependant pas toujours un gage de réussite ou inversement d'échec. La discrimination peut très bien être bonne sans distribution gaussienne.

Voici les résultats obtenus avec ce jeu de données :

	ADL	ADQ	NBA	LOG	LOGQ	TREE
Estimation de ε	0.223	0.237	0.240	0.219	0.237	0.290
Intervalle sur ε	[0.218, 0.229]	[0.232, 0.242]	[0.234, 0.246]	[0.215, 0.224]	[0.232, 0.244]	[0.272, 0.308]

TABLE 5 – Estimation du taux d'erreur et intervalle de confiance de ε, avec chacune des méthodes de discrimination ; données *Pima*.

Les taux d'erreur sont très mauvais pour chacun des modèles. Une représentation grâce à l'ACP des données montre comme à chaque fois qu'il est très difficile de séparer les groupes de diabétiques et non diabétiques :

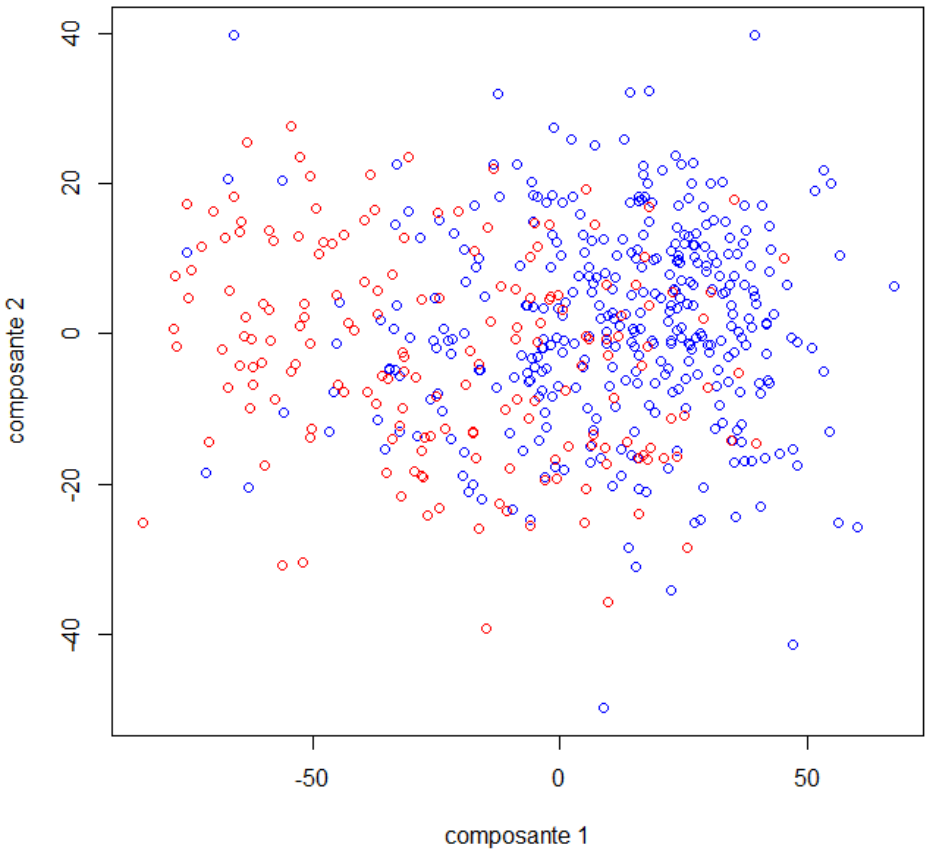

FIGURE 15 – Représentation des données dans le premier plan factoriel

Même si l'inertie expliquée par cet espace vectoriel est mauvaise, on voit difficilement comment discriminer un nouvel individu selon les variables prédictives qui caractérisent les données (*glu, npreg...*). Notons qu'effectuer une discrimination avec les variables de l'ACP comme variables prédictives donne sensiblement les mêmes résultats.

Un moyen efficace d'obtenir des bons taux de classification serait de réduire l'ensemble à une variable seulement, qui serait la plus prédictive au niveau des classes. Cependant on perdrait l'intérêt des lois multivariées et on ne peut pas y appliquer les fonctions discriminantes développées dans cette étude, encore faut-il trouver cette variable.

Frontières

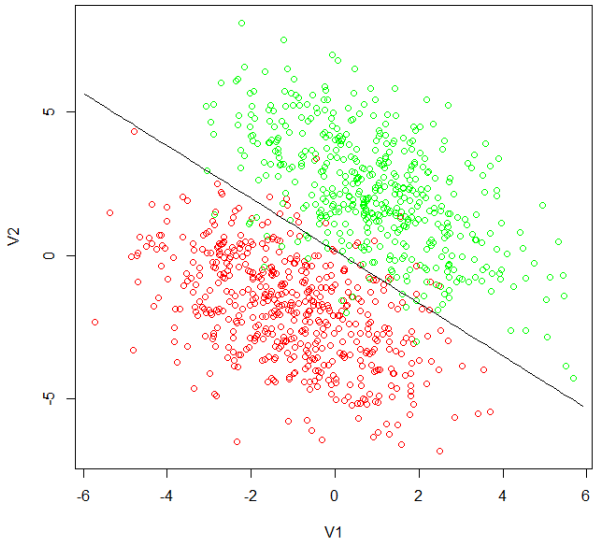

FIGURE 16 – LOG

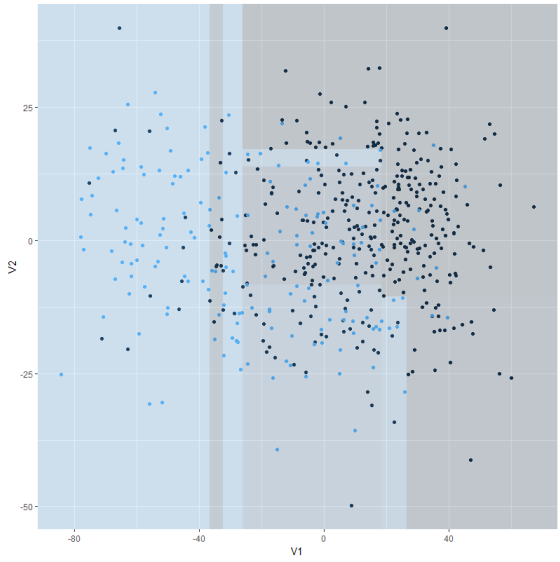

FIGURE 17 – TREE

As we have seen with the PCA, it is really difficult to distinguish 2 groups, the frontier traited by tree method looks really anmbigue and also have 2 parts that blend to each other, it is much confident on the left hand side but not for the right hand side. There are also the points could not be seperated by the tree at the upper-right part.

Nous n'avons donc finalement trouvé aucun moyen de prédire correctement le diabète en fonction des variables explicatives fournies.

3.2.2 Breast cancer Wisconsin

	ADL	ADQ	NBA	LOG	TREE
Estimation de ε	0.046	0.050	0.039	0.041	0.039
Intervalle sur ε	[0.044, 0.048]	[0.048, 0.053]	[0.036, 0.041]	[0.038, 0.04]	[0.033, 0.044]

TABLE 6 – Estimation du taux d'erreur et intervalle de confiance de ε, avec chacune des méthodes de discrimination ; données *Breast cancer Wisconsin*.

Les taux obtenus ici sont bien meilleurs par rapport à Pima.

Frontières

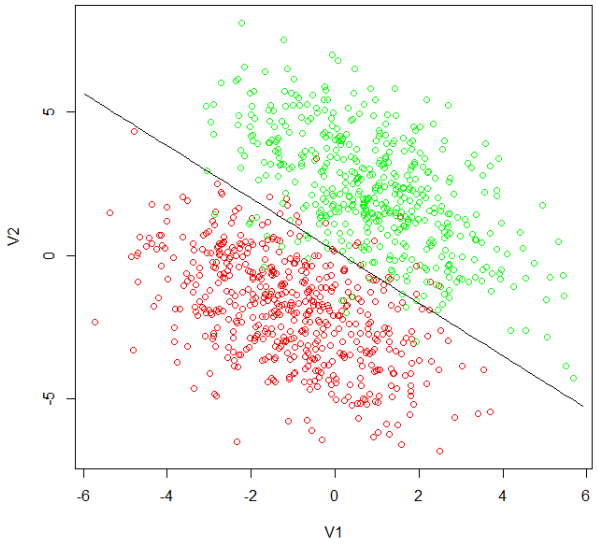

FIGURE 18 – LOG

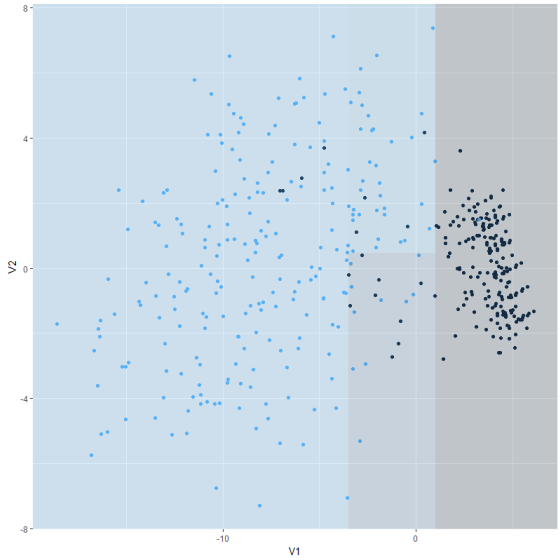

FIGURE 19 – TREE

This time, the taux d'erreur is small so the tree algorithm is much better on the breast cancer wisconsin dataset. Even though les points with z=1 is "plus disperse", z=2 is "plus centre", the left part of the graph is separate well with the the right part. The only part that ambigue is the small region in between. However, it is only a small portion.

La prédiction du cancer du sein est meilleure que celle du diabète. On peut considérer soit que le cancer du sein est un phénomène qui s'explique mieux par les indicateurs que l'on connaît de cette maladie, soit que ceux du diabète étaient incomplets, insuffisants ou trop nombreux.

3.2.3 Conclusion sur les données réelles

On remarque que lorsque l'on dispose de beaucoup de variables et de peu d'individus, les prédictions sont en général mauvaises. Moins le modèle d'apprentissage est complexe et paramétrique, plus il pourra se contenter de peu d'observations, mais en généra,l un modèle sera d'autant meilleur que les données sur lesquelles on l'entraîne sont nombreuses. Cependant, leur distribution et leur répartition conditionnellement à leur classe influent tout autant sur les performances des modèles qui posent beaucoup d'hypothèses.

Finalement, si les variables étudiées ne sont tout simplement pas assez explicatives (elles ne permettent pas de prédire l'appartenance à une classe d'un individu), alors il sera très difficile de faire des prédictions. Les processus de sélection de variables constituent donc une facette très importante des procédés de classification automatique supervisée. C'est l'objet de la dernière partie.

4 Challenge des données Spam

Spam est un jeu de données contenant 57 variables explicatives et une variable de classe à prédire, qui caractérise des mails en tant que *spam* ou non. On suppose à la vue des variables que les indicateurs d'entrée $V_1 ... V_{57}$ sont des fréquences ou des scores qui peuvent caractériser des mots associés aux spams, des expressions, répétitions etc. Les hypothèses peuvent être multiples sans description fournie au préalable.

Avec 4601 observations pour 58 variables au total, on comprend vite que le challenge principal de ce *dataset* est de réduire sa dimension, c'est-à-dire son nombre d'indicateurs d'entrée. Le *fléau du dimensionnement*, qui désigne un déséquilibre entre le nombre d'individus d'apprentissage et le nombre de variables explicatives d'entrée, affecte les modèles de discrimination et de régression. Il porte préjudice au modèle en leur apportant de la confusion et du *bruit*, ce qui dégrade considérablement la qualité de leurs prédictions et la variance de leurs

résultats, qui deviennent beaucoup plus hasardeux (perte de reproductibilité).

Nous avons tout de même fait de l'apprentissage supervisé avec les modèles étudiés, sans traitement :

	ADL	ADQ	NBA	LOG	LOGQ	TREE
Estimation de ε	0.154	0.155	0.091	0.066	0.077	0.064

TABLE 7 – Estimation du taux d'erreur de discrimination sur les données *Spam*.

Les taux d'erreurs sont encore acceptables pour les modèles de régression logistique, mais il est sûrement possible de faire mieux au vu de la problématique exposée précédemment.

De nombreux algorithmes ont été étudié pour trouver des solutions de sélection de variables, encore appelées *feature selection methods*, afin de réduire l'espace dimensionnel des données d'entrée aux indicateurs les plus prédictifs. Une méthode que nous avons étudiée est celle des forêts aléatoires (*randomForest* avec R), qui permettent de générer aléatoirement plusieurs arbres de décision et de les tester sur des échantillons de données pour faire de la prédiction. On peut l'utiliser facilement dans la *feature selection* puisque le calcul de critère d'impureté apporté par une variable est simple. Rappelons que le critère d'impureté le plus utilisé est celui de Gini :

$$G(p) = \sum_{k=1}^{g} p_k(1 - p_k) \tag{9}$$

Celui-ci calcule l'homogénéité des ensembles de données séparés lors de la prédiction des classes, avec la fréquence conditionnelle p_k.

Le processus de sélection associé à cette méthode est donc d'induire des arbres aléatoires de décision qui mesureront la décroissance du critère de Gini à chaque nouveau nœud, pour déterminer les variables les plus influentes en terme de prédiction. La construction de ces arbres utilise le *Bootstrap* (inférence statistique avec des échantillons de *bootstrap* : sous-ensembles des données).
En définissant plusieurs seuils d'impact sur l'indice de Gini, on réduit les variables utilisées. Un seuil assez haut (impact supérieur à 15 au minimum) sélectionne 24 variables. De nouvelles estimations du taux d'erreur donnent alors 0.035 avec une régression logistique ou un arbre de décision, ce qui divise pratiquement par deux les taux d'erreurs obtenus initialement (tableau 7).

La confusion générée par un espace dimensionnel trop grand pour les modèles de prédiction peut donc être réduite avec des processus de sélection de variables. D'autres méthodes peuvent être citées comme celles de régression et de régularisation d'indicateurs, l'analyse de corrélations, les algorithmes génétiques...

5 Conclusion

Ce quatrième et dernier Travail Pratique a été l'occasion de découvrir de nouveaux processus d'apprentissage supervisé. Les modèles d'analyse discriminante permettent de prédire une variable en utilisant des hypothèses sur la distribution des données et sur les estimations de leurs paramètres, quand les modèles de régression logistique utilisent des expressions des probabilités à *poseriori*. Les arbres permettent aussi de construire des prédictions ou des régressions à partir d'un ensemble de variables explicatives le plus souvent numériques, mais constituent aussi un procédé intéressant de sélection de variables pour pallier au *fléau des dimensions*...

Toutes ces méthodes possèdent leurs caractéristiques propres et n'ont pas les mêmes performances en fonction des paramètres qui sont estimés mais aussi des particularités probabilistes de la répartition des données. Ils font parti de l'outil de base de l'analyste de données qui veut construire des modèles de prédiction.