Rapport

Statistique en grande dimension

Soheil Salmani & Komi Agblodoe

20 janvier 2020

Table des matières

Notre rapport concerne le chapitre 9 intitulé *Additive Models, Trees, and Related Methods* du livre *The Element of Statistical Learning : Data Mining, Inference, and Prediction* écrit par Trevor Hastie, Robert Tibshirani et Jerome Friedman.

Dans ce rapport, nous présenterons trois techniques permettant d'estimer une fonction de regression inconnue : le modèle **GAM** (pour *Generalized Additive Model*, ou *modèle additif généralisé*), les **arbres** et le modèle **MARS** (pour *Multivariate Adaptative Regression Splines* ou *régression multivariée par spline adaptative*). Chaque méthode implique un compromis dans la construction du modèle que nous étudierons au cas par cas.

1 Modèle additif généralisé

Les modèles de régression linéaires échouent très souvent en pratique, du fait que la plupart du temps, les effets observés ne sont pas linéaires. Le modèle additif généralisé introduit ici est une solution flexible et automatique permettant d'estimer un modèle de régression non-linéaire de manière non-paramétrique. Les modèles additifs généralisés constituent une classe de modèles statistiques pour lesquelles les relations linéaires entre prédicteurs et réponses sont remplacés par plusieurs fonctions de lissages non-linéaires, afin de capturer les effets non-linéaires dans les données.

Un modèle additif généralisé a la forme suivante :

$$E(Y|X_1, X_2, \ldots, X_p) = \alpha + f_1(X_1) + f_2(X_2) + \cdots + f_p(X_p)\,.$$

Les p fonctions sont ici des fonctions de lissage locales (*scatterplot smoothers*). Un algorithme est utilisé pour simultanément estimer toutes les p fonctions.

Nous allons ici présenter cette classe de modèles statistiques à travers un exemple en utilisant le package `gam` sous R.

1.1 Modèle additif généralisé avec le package **gam**

L'idée ici est que nous allons ajuster des fonctions non linéaires de lissage sur un groupe de prédicteurs X_i pour capturer et apprendre les relations non linéaires entre les variables du modèle (c'est-à-dire X) et Y.

1.1.1 Données

Nous utiliserons le dataset `Wage` du package `ISLR`. L'objectif étant de prédire le salaire en fonction de l'âge (`age`), l'année (`year`) et le niveau d'éducation (`education`).

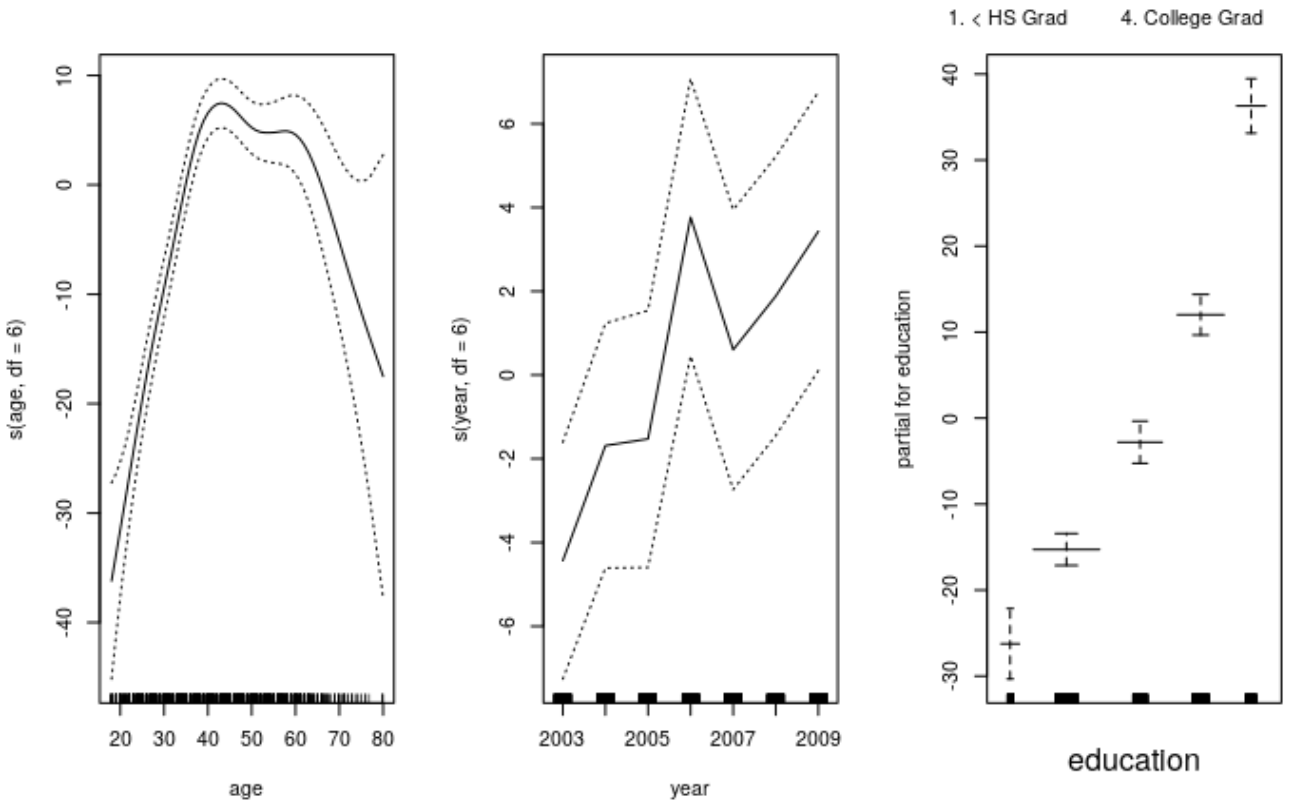
1.1.2 Modèle obtenu avec la package **gam**

Nous construisons un modèle avec la package `gam` dont la sortie est donnée ci-dessous.

```
Call: gam(formula = wage ~ s(age, df = 6) + s(year, df = 6) + education,
    data = Wage)
Deviance Residuals:
    Min      1Q  Median      3Q     Max
-119.89  -19.73   -3.28   14.27  214.45

(Dispersion Parameter for gaussian family taken to be 1235.516)

    Null Deviance: 5222086 on 2999 degrees of freedom
Residual Deviance: 3685543 on 2983 degrees of freedom
AIC: 29890.31

Number of Local Scoring Iterations: 2

Anova for Parametric Effects
                  Df  Sum Sq Mean Sq F value    Pr(>F)
s(age, df = 6)     1  200717  200717 162.456 < 2.2e-16 ***
s(year, df = 6)    1   22090   22090  17.879 2.425e-05 ***
education          4 1069323  267331 216.372 < 2.2e-16 ***
Residuals       2983 3685543    1236
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Anova for Nonparametric Effects
               Npar Df  Npar F  Pr(F)
(Intercept)
s(age, df = 6)        5 26.2089 <2e-16 ***
s(year, df = 6)       5  1.0144 0.4074
education
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Dans le modèle ci-dessus, nous avons un modèle additif généralisé (ou GAM) qui est non-linéaire pour les variables `age` et `year` en utilisant des splines lissantes avec 6 degrés de liberté, alors qu'elle reste linéaire pour la variable `education`.

Représentons graphiquement le modèle obtenu :



L'image ci-dessus présente un graphe pour chaque variable inclus dans le modèle. Nous voyons que le salaire augmente avec l'âge, et retombe à partir de 60 ans. De même, le salaire semble augmenter avec l'année, mais nous observons une descente vers 2007 ou 2008. Ce sont ces effets non-linéaires que nous souhaitons expliquer. Enfin, le salaire semble évoluer linéairement avec le niveau d'éducation. La forme des courbes des variables `age` et `year` sont dues aux splines lissantes qui modélisent les non-linéarités des données. Les lignes en pointillés autour de la courbe correspondent à la bande d'erreur type.

Les GAMs sont donc un moyen très efficace pour ajuster des fonctions non-linéaires sur plusieurs variables, et pour produire leurs graphes afin d'étudier l'effet de chacune d'entre-elles sur la réponse.

1.2 Régression logistique et GAM

Nous pouvons également adapter un modèle de régression logistique utilisant des GAMs afin de prédire les probabilités des valeurs de réponse binaire.

Pour une classification binaire, le modèle de régression logistique se modélise de la manière suivante :

$$\log\left(\frac{\mu(X)}{1-\mu(X)}\right) = \alpha + \beta_1 X_1 + \cdots + \beta_p X_p\,.$$

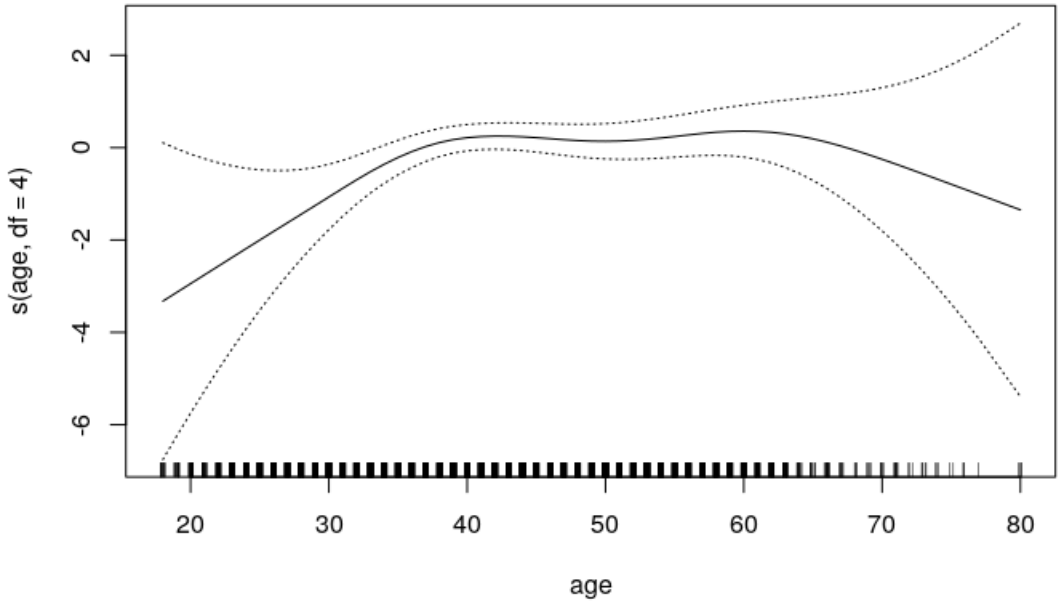
Avec un modèle additif généralisé, la modèle se formalise ainsi :

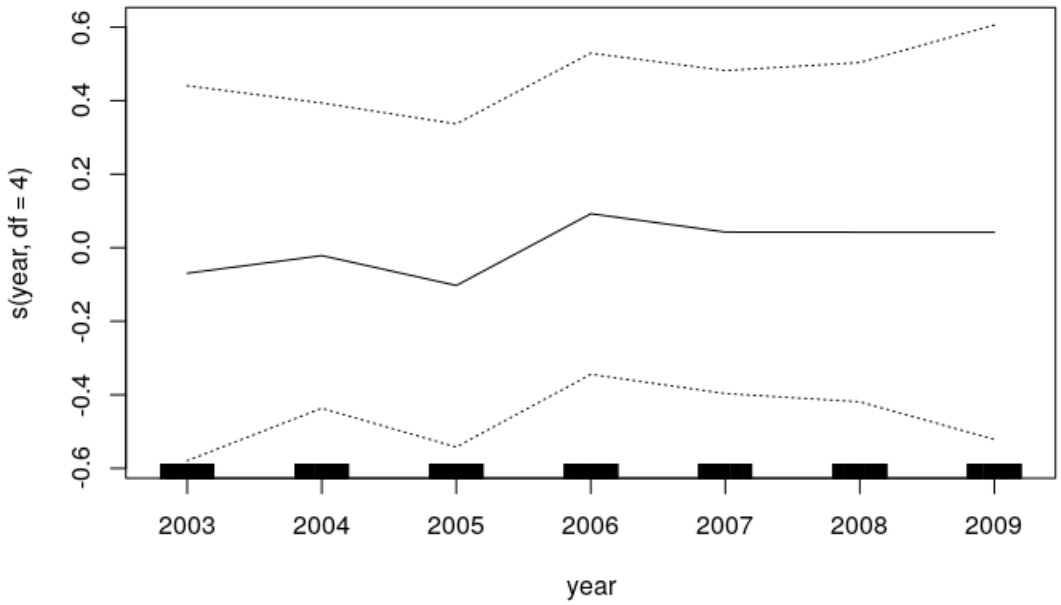
$$\log\left(\frac{\mu(X)}{1-\mu(X)}\right) = \alpha + f_1(X_1) + \cdots + f_p(X_p)\,.$$

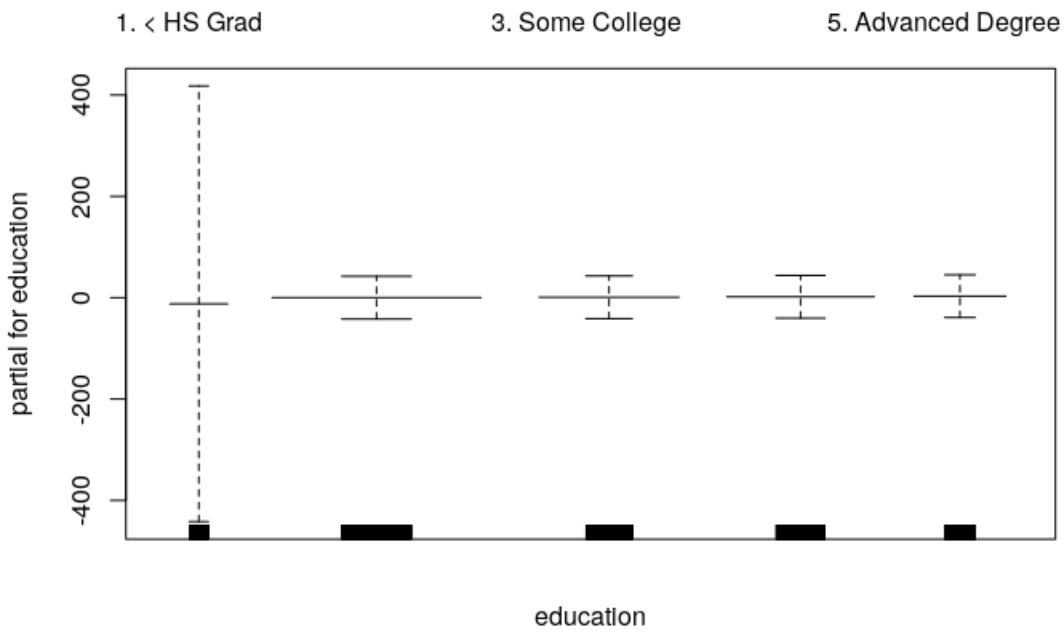
où, comme décrit précédemment, chaque fonction f_i est une fonction de lissage non spécifiée.

Nous reprenons ainsi notre exemple, et nous essayons de prédire $P(\texttt{wage} > 250|X_i)$ à partir d'un modèle de régression logistique utilisant des GAMs. Nous utiliserons également des splines lissantes.

Nous construisons le modèle et nous obtenons les graphiques suivants :







Dans le graphique ci-dessus pour la variable `year`, nous pouvons voir que les bande d'erreur est assez large, ce qui pourrait indiquer que notre fonction non linéaire ajustée pour la variable `year` n'est pas significative.

1.3 Conclusion

Les modèles additifs généralisés sont un moyen efficace d'ajuster des modèles linéaires qui dépendent de fonctions non linéaires sur certains prédicteurs afin d'expliquer des effets non linéaires dans les données. Nous pouvons facilement mélanger des termes dans les GAMs, certains termes linéaires et d'autres non linéaires, puis comparer ces modèles à l'aide d'un test ANOVA pour vérifier la qualité de l'ajustement. Les termes non linéaires des prédicteurs X_i peuvent être, par exemple, des splines lissantes, des splines cubiques, des fonctions polynomiales etc. Les GAMs sont de nature additive, ce qui signifie qu'il n'y a pas de terme d'interaction dans le modèle.

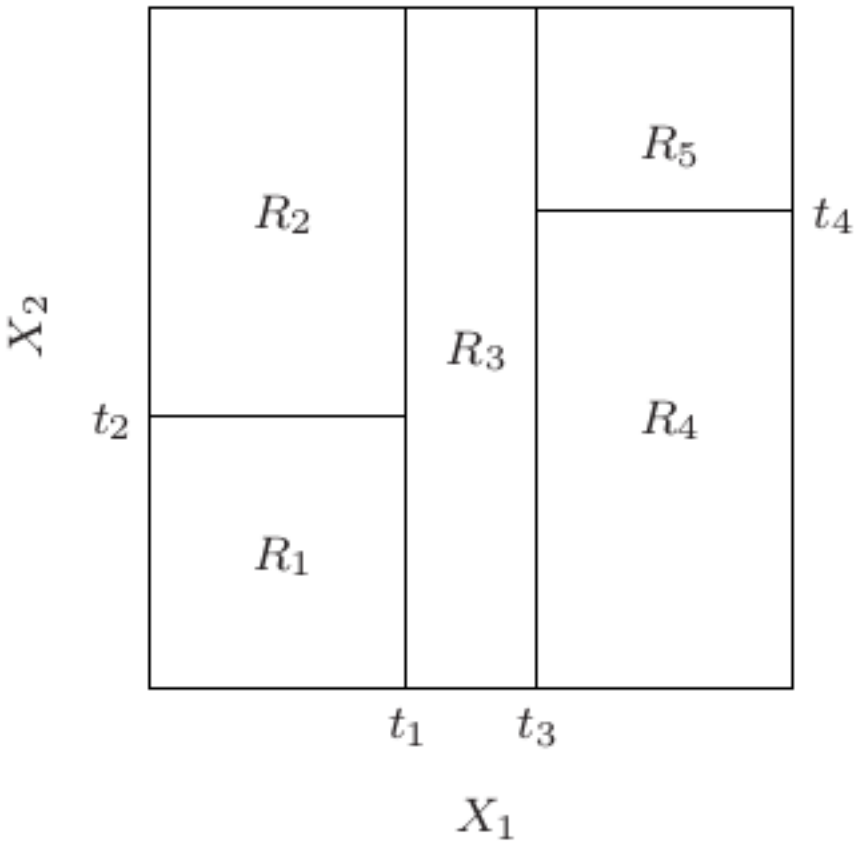
2 Méthodes à base d'arbres et MARS

Les méthodes à base d'arbres sont des méthodes qui partitionnent l'espace des co-variables en un ensemble de rectangles, et font ensuite correspondre un modèle (par exemple une constante) pour chaque élément. Ces modèles sont simples mais très efficaces.

Dans cette section, nous introduirons la méthode CART pour la classification et régression, ainsi que MARS.

Considérons un problème de régression dans lequel nous avons deux variables X_1 et X_2, et une variable cible Y. Une première méthode de partitionnement consisterait à diviser récursivement l'espace des co-variables en deux. Ainsi, on commence par diviser l'espace en deux, puis nous modélisons la réponse par la moyenne de Y pour chaque région. Ensuite, chaque région est divisée en deux également, et ceci de manière récursive jusqu'à ce qu'une règle d'arrêt est appliquée.

Considérons l'exemple ci-dessous :



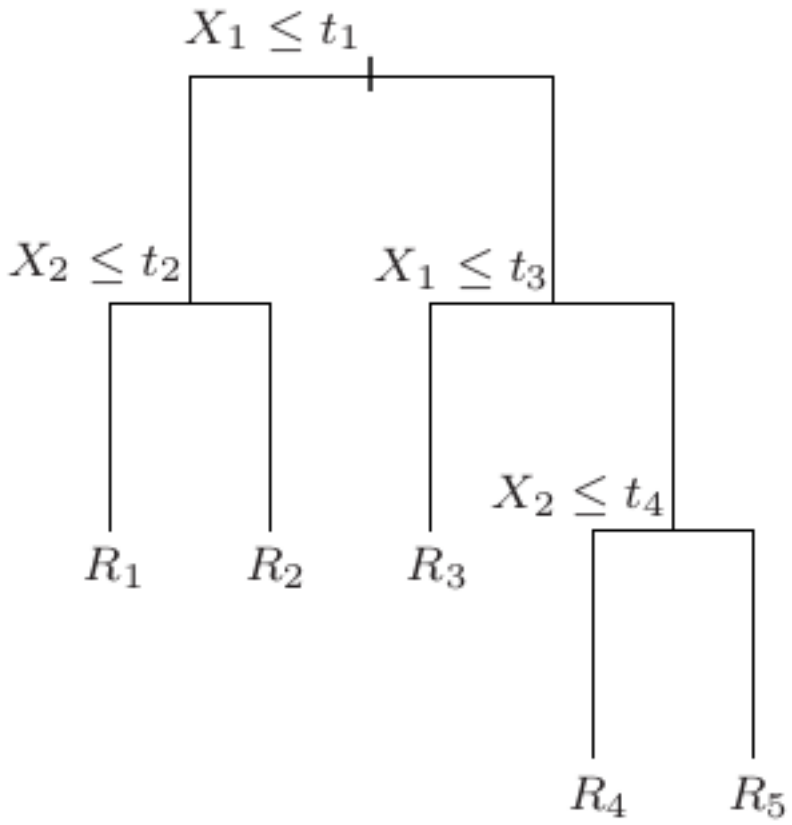
Ce partitionnement a été obtenu récursivement de la manière suivante :

1. nous avons divisé tout d'abord l'espace en $X_1 = t_1$;
2. la région $X_1 \leq t_1$ est ensuite divisée en $X_2 = t_2$, et la région $X_1 > t_1$ est divisé en $X_1 = t_3$;
3. puis enfin, la région $X_1 > t_3$ est divisée en $X_2 = t_4$.

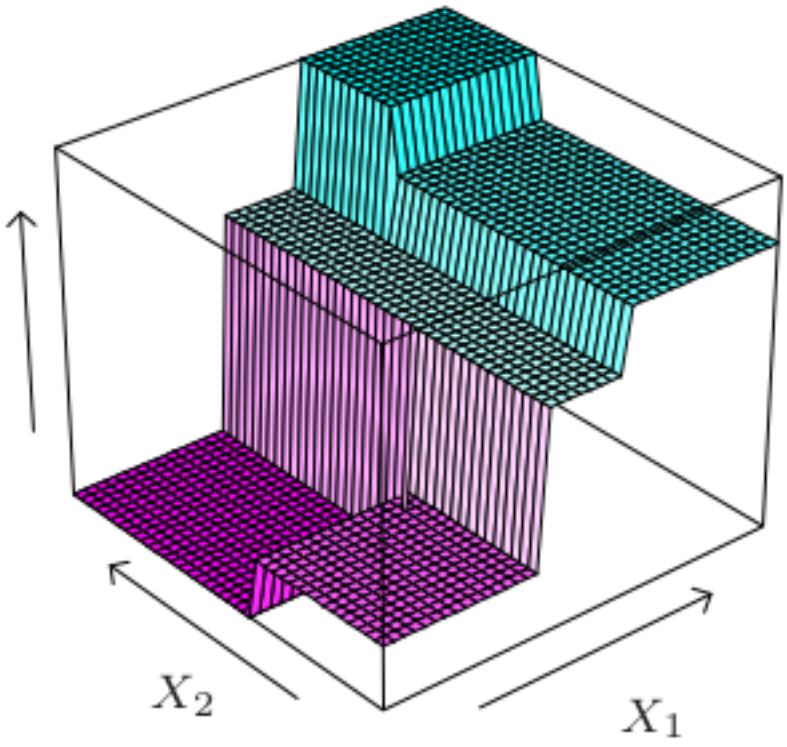
Nous obtenons alors cinq régions $R_1, R_2, \ldots, R_5$. Nous obtenons alors un modèle de régression qui prédit Y par une constante c_m pour chaque région R_m. Formellement, nous avons donc :

$$\hat{f}(X) = \sum_{m=1}^{5} c_m I\{(X_1, X_2) \in R_m\}\,.$$

L'avantage de cette modélisation est qu'un tel modèle est facilement interprétable, ainsi dans l'exemple précédent, nous pouvons représenter le modèle construit par l'arbre ci-dessous :



De plus, le modèle est également facilement représentable graphiquement (du moins, en considérant deux covariables). Ainsi, dans l'exemple précédent, si nous associons les constantes $c_1 = -5$, $c_2 = -7$, $c_3 = 0$, $c_4 = 2$ et $c_5 = 4$ aux régions $R_1, R_2, \ldots, R_5$, nous pouvons proposer un graphique agrégeant toutes ces informations, comme celui ci-dessous :



Le principal avantage d'un tel modèle est donc son **interprétabilité**.

2.1 Arbres de régression

Nous avons vu précédemment comment représenter un modèle sous forme d'arbre de régression. La question est maintenant de savoir comment déterminer un bon partitionnement de l'espace des co-variables, ou encore, comment choisir pour chaque itération, la variable à considérer pour le partitionnement et le seuil de division.

Trouver la meilleur partition en considérant comme critère de minimisation la somme des carrés, est généralement infaisable informatiquement. Par conséquent, nous procédons par un algorithme glouton. Nous commençons avec l'ensemble des données, est nous choisissons le meilleur variable et seuil de partitionnement. Nous utilisons cette méthode récursivement jusqu'à ce qu'un critère d'arrêt soit appliqué.

Un arbre de régression peut devenir très rapidement grand, et donc difficilement interprétable, d'où le choix d'un critère d'arrêt. La taille de l'arbre est un exemple d'un tel hyperparamètre, et doit être choisi minutieusement selon le jeu de données considéré. Une autre approche consisterait à diviser l'espace uniquement si le critère de minimisation dépasse un certain seuil de réduction. Enfin, un autre critère consisterait à fixer au préalable le nombre de feuilles à considérer pour le modèle final.

2.2 Arbres de classification

Dans le cas des arbres de classification, la seule différence à considérer par rapport au cas de la régression est le critère de minisation à considérer. Ainsi, dans le cas de la régression, nous avons utilisé la somme des carrés, et dans le cas de la classification, nous devons trouver un critère mesurant l'impureté de chaque région dans l'espace des co-variables. Une région correspond correspond à un nœud dans l'arbre, il faut donc une méthode pour mesure l'impureté de chaque nœud.

Ainsi, dans un nœud m, représentant une région R_m avec N_m observations, nous notons $\hat{p}_{mk}$ la proportion d'observations de classe k dans la région R_m, soit :

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k),$$

Nous classons donc une observation de R_m par la classe majoritaire du nœud m, soit formellement, par la classe $k(m) = \arg\max_k \hat{p}_{mk}$.

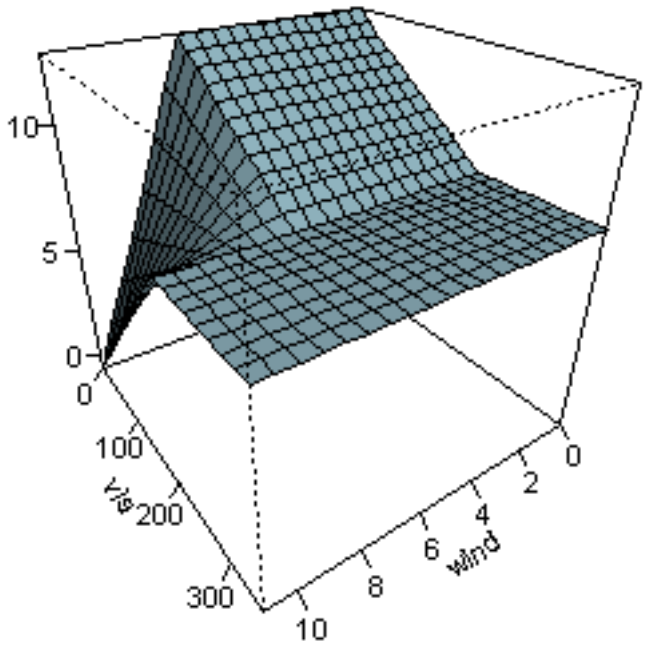
Différents critères existent pour mesure l'impureté au niveau de chaque nœud, nous avons par exemple le :

- *misclassification error* : $\frac{1}{N_m} \sum_{i \in R_m} I(y_i \neq k(m)) = 1 - \hat{p}_{mk(m)}$;
- *Gini index* : $\sum_{k \neq k'} \hat{p}_{mk} \hat{p}_{mk'} = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$;
- *cross-entropy* ou *deviance* : $-\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}$.

2.3 Problème : modèle non lisse

Un des problème au niveau des modèles à base d'arbres est que la fonction de prédiction est non lisse. Cela n'a pas trop d'importance dans le cas de la classification, mais peut dégrader significativement les performances dans le cas de la régression, où l'on souhaiterait avec une fonction de prédiction lisse.

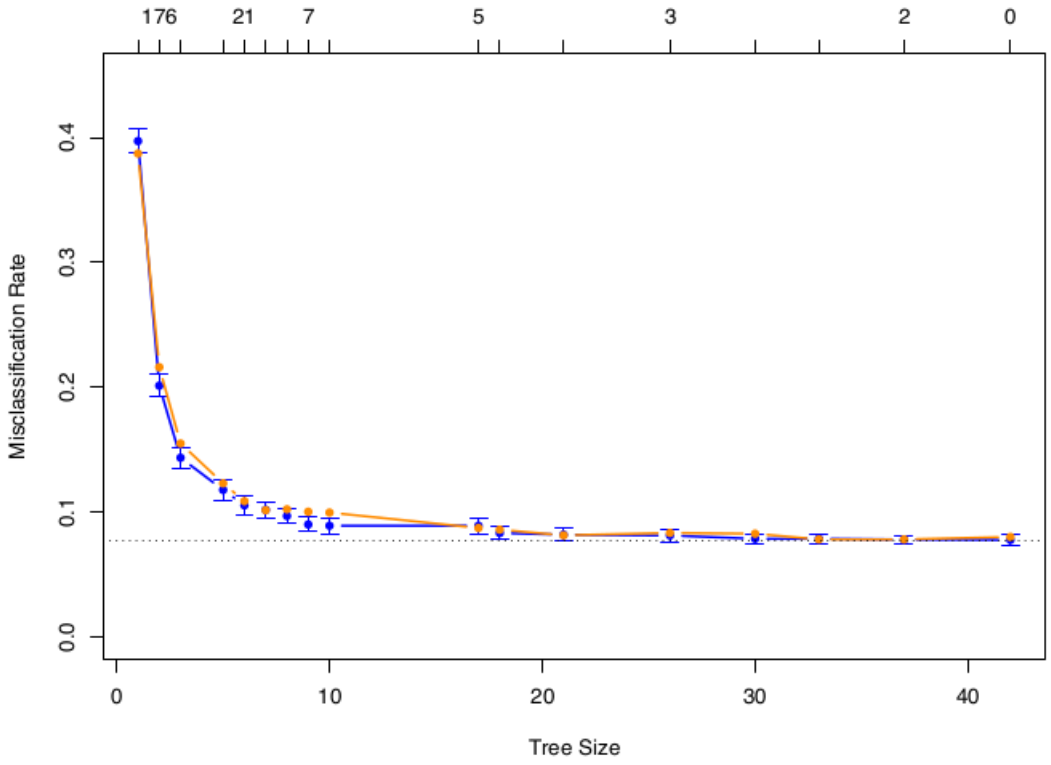
Une solution est d'appliquer la procédure MARS (MARS pour *Multivariate Adaptative Regression Splines*) qui peut être vu comme une modification de CART pour gérer justement ce manque de douceur dans la fonction de régression à l'aide de splines. MARS permet ainsi d'obtenir des modèles moins « rigides ». Un exemple graphique de modèle obtenu avec MARS est présenté ci-dessous :



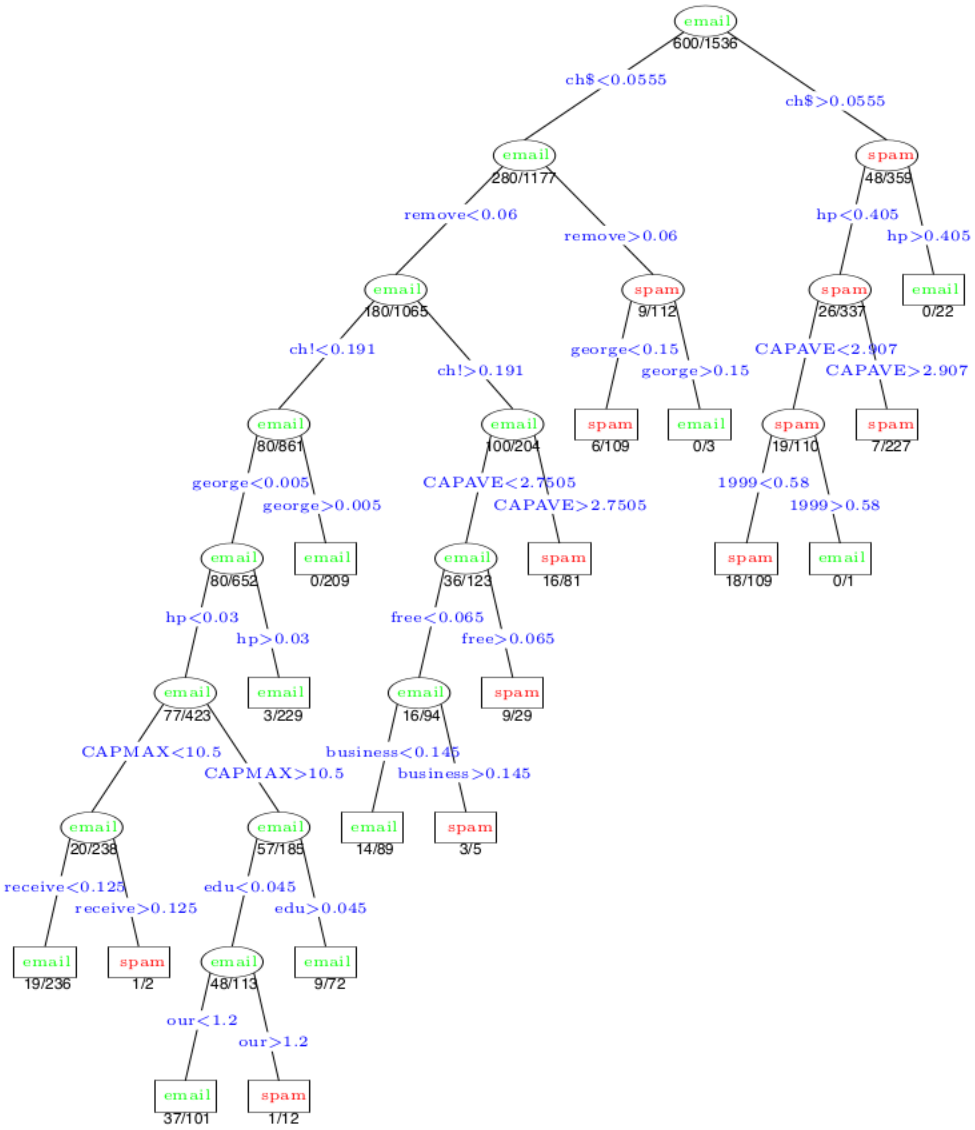
2.4 Exemple pour la classification d'e-mails

Nous souhaitons construire un arbre de classification pour les e-mails afin de déterminer si un e-mail doit être considérer comme « spam », ou non. Nous utilisons l'algorithme de classification présenté précédemment en utilisant le *cross-entropy* comme critère pour développer l'arbre de classification, et le *misclassification error* pour l'élaguer.

Nous appliquons une méthode de validation croisée à 10 folds pour construire l'arbre et nous traçons le taux d'erreur en fonction de la taille de l'arbre. Le taux d'erreur pour le jeu de données de test est présenté en orange dans le graphe ci-dessous :



Le voyons que le taux d'erreur se stabilise autour de 17 nœuds terminaux, donnant l'arbre de classification suivant :

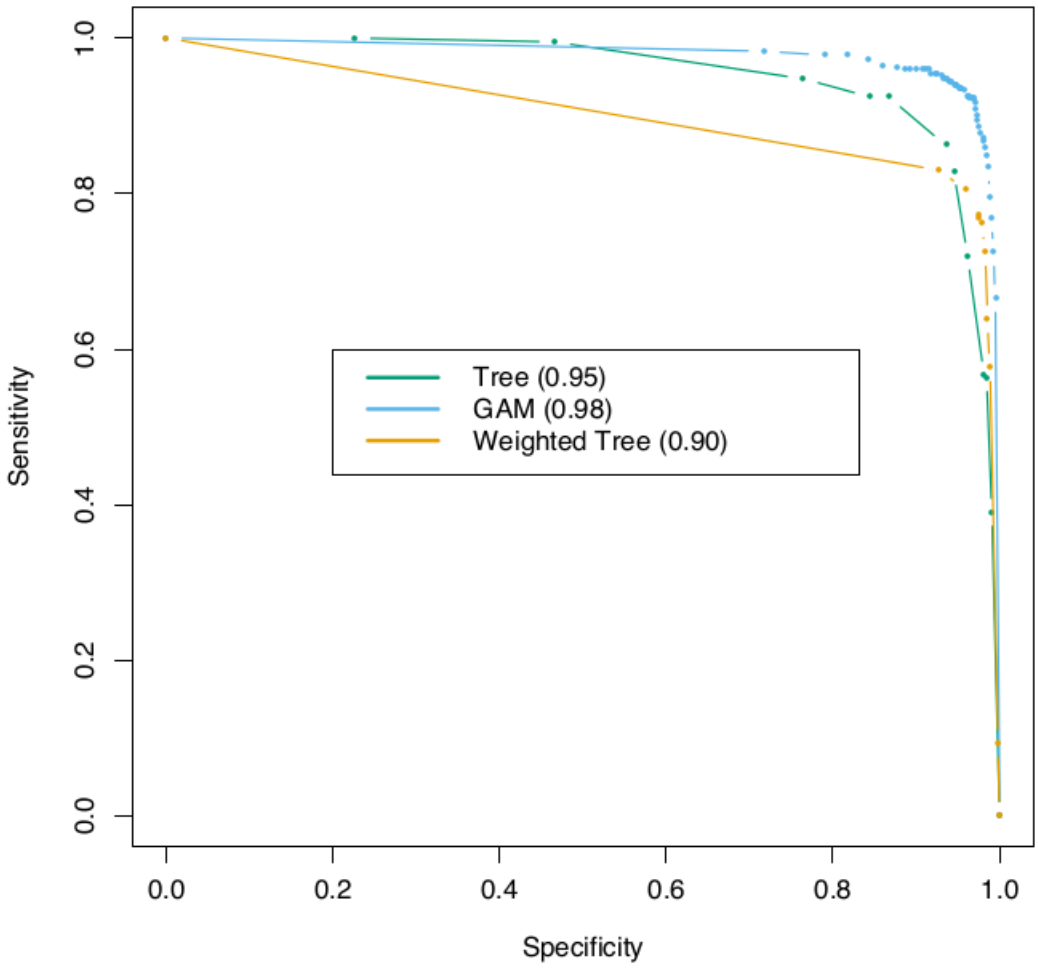


Les résultats de notre arbre de classification sur notre jeu de test sont résumés ci-dessous :

	Predicted	
True	email	spam
email	57.3%	4.0%
spam	5.3%	33.4%

Nous avons donc un taux d'erreur de 9.3%, une sensibilité de $100 \times \frac{33.4}{33.4+5.3} = 86.3\%$, et une spécificité de $100 \times \frac{57.3}{57.3+4.0} = 93.4\%$.

À titre de comparaison, en traçant la courbe ROC de différents modèles pour ce jeu de données de test, nous voyons que le modèle additif généralisé reste plus performant que l'arbre de classification pour ce cas précis d'application. Les courbes ROC des différents modèles comparés sont représentés ci-dessous.


Sensitivity
Specificity
0.0
0.2
0.4
0.6
0.8
1.0
Tree (0.95)
GAM (0.98)
Weighted Tree (0.90)