Data mining - Crédit à la consommation

Hervier Timothy, Ekuba Bénicia, Mulot Lucile

Table des matières

1) Préparation des données

A Nettoyage de la base

Notre base de données "crédit à la consommation" comprend 10 000 observations de 41 variables. Notre but est de prédire si le client sera solvable ou non.

En étudiant les variables, nous nous sommes vite rendu compte que certaines étaient redondantes. En effet, elles étaient présentes en facteurs et en numériques. Nous avons donc décidé d'enlever les facteurs. Par la suite, nous avons remarqué que certaines variables pouvaient être fusionner car elles donnaient les mêmes informations (par exemple, une variable donnait la partie constante d'une note et l'autre sa partie décimale). Enfin, une variable était constante et une était en double, nous les avons également enlevé. Nous avons donc à présent une base de données propres comprenant 10 000 observations de 34 variables.

Pour réaliser nos futurs études nous avons besoin de diviser la base de données en deux. Nous choisissons de diviser la base de données en proportion $\frac{1}{3}$ / $\frac{2}{3}$: $\frac{1}{3}$ pour les données tests et $\frac{2}{3}$ pour les données d'entrainements. Nous avons donc 3 333 observations dans la base de données test et 6 667 dans les données d'entrainement.

B Gestion des données manquantes

Nous décidons à présent de vérifier si nous avons des données manquantes. Nous n'en avons aucune dans la base apprentissage, par contre, nous en avons 14 dans la base test.

Pour gérer ces données nous décidons d'utiliser la technique de Knn. Nous enlevons donc la variable à expliquer de notre base puis lançons la commande Knn avec k=10. Par la suite, nous enlevons alors les 33 variables créées par le Knn et ajoutons notre variable à expliquer pour créer notre base test finale.

2) LDA-QDA-Logit

A LDA

Pour commencer notre analyse nous créons une base de données d'apprentissage et une base de données test uniquement avec les variables numériques. Nous lançons alors nos commandes pour réaliser notre analyse linéaire discriminante. Un message d'erreur apparait nous indiquant que des variables colinéaires sont présentes. Pour décorréler ces variables nous décidons de procéder à une ACP. Nous procédons alors de nouveau à la LDA mais en utilisant les coordonnées des individus obtenues avec l'ACP. Nous commençons par utiliser un seuil à 0.5. Sachant que pour une banque le plus important est qu'il y ait peu de faux négatifs, nous décidons de baisser le seuil à 0.2 en prenant le risque d'augmenter l'erreur globale. Cette LDA nous donne comme résultat :

```
Prior probabilities of groups:
        N         Y
0.8107095 0.1892905
```

C'est à dire que 81% des observations correspondent à non dans BadLoan et que 18% des observations correspondent à des prêts problématiques.

B QDA

Pour l'analyse quadratique discriminante nous procédons comme pour la LDA, c'est à dire que nous utilisons les coordonnées des individus obtenues lors de l'ACP ainsi qu'un seuil à 0.2.

C Logit

Nous utilisons également l'ACP pour réaliser un modèle Logit. Nous obtenons le modèle suivant :

TABLE 1 – Modèle de régression logistique

	Dependent variable :
	BadLoan
Dim.1	-0.204^{***}
	(0.017)
Dim.2	-0.219^{***}
	(0.021)
Dim.3	0.302^{***}
	(0.023)
Dim.4	0.635^{***}
	(0.026)
Dim.5	0.527^{***}
	(0.029)
Constant	-1.908^{***}
	(0.043)
Observations	6,667
Log Likelihood	−2,526.545
Akaike Inf. Crit.	5,065.089
Note :	*p<0.1 ; **p<0.05 ; ***p<0.01

Nous pouvons noter que la dimension 1 (contenant des variables comme funded_amnt, loan_amnt, funded_amnt_inv...) et 2 (contenant des variables comme pub_rec_zero, last_record_none et pub_rec) influence négativement BadLoan. C'est à dire que si ces dimensions augmentent, le prêt devient plus sur. À l'inverse, les autres dimensions influencent négativement BadLoan.

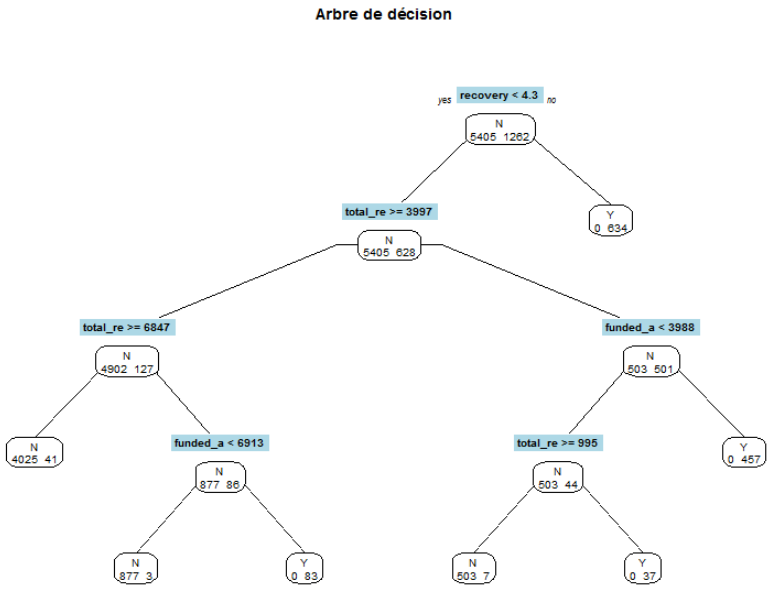
3) Knn-Arbres-Forêts aléatoires-Boosting

A Knn

Nous commençons par réaliser une méthode Knn avec un k=5. Par la suite, nous décidons d'améliorer cette prédiction avec la fonction tune. Nous trouvons alors un k optimal égal à 1. Nous refaisons alors notre prédiction avec ce nouveau paramètre.

B Arbres

Nous commençons par les arbres. Nous décidons de réaliser un arbre sur toutes les données d'apprentissages. Nous décidons ensuite d'utiliser le package tune pour trouver le meilleur paramètre de complexité afin d'améliorer notre arbre et de procéder à l'élagage. Par la suite, nous décidons de refaire notre arbre en nous aidant des résultats précédemment obtenus. Nous obtenons l'arbre suivant :

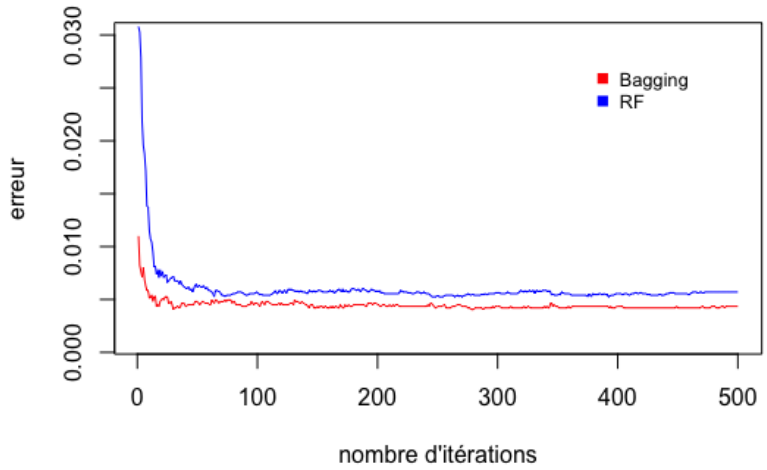


On peut déduire de cet arbre les conditions pour que le client soit à risque. Comme par exemple :

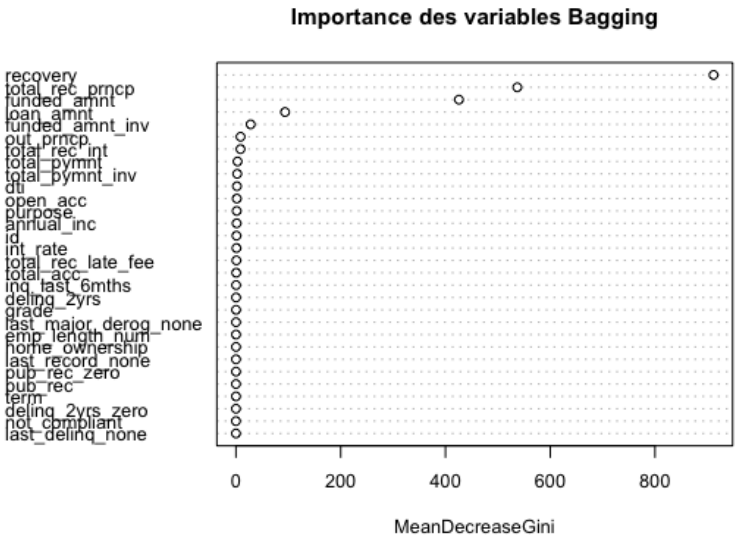
- Si le coût de recouvrement est supérieur à 4.3
- Si le coût de recouvrement est inférieur à 4.3, que le capital reçu est inférieur à 3997 et que le montant total engagé dans le prêt est supérieur 3988

C Bagging-Forêts aléatoires

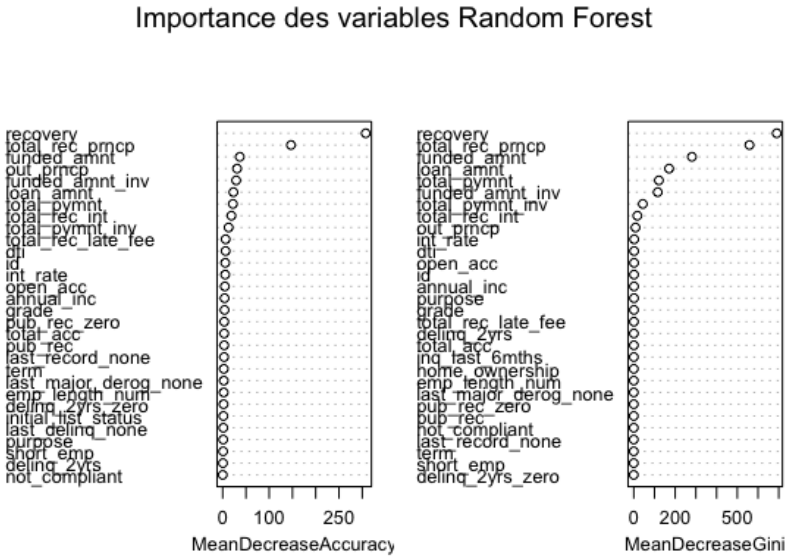
Nous commençons par réaliser une forêt aléatoire et un bagging. Nous décidons de visualiser l'erreur OOB de ces deux techniques. Nous obtenons ce résultat :



Nous décidons ensuite d'améliorer notre forêt aléatoire en choisissant le nombre de variable adapté (qui est de 20) ainsi que la taille minimale d'un noeud terminal (qui est de 13). Pour améliorer notre bagging nous décidons de modifier la taille minimale des noeuds terminaux. Nous la fixons à 10. Nous obtenons comme importance des variables pour le bagging :



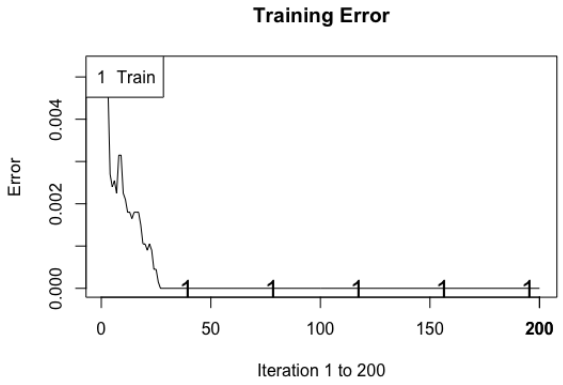
Nous pouvons conclure que les variables les plus importantes dans notre bagging sont recovery, total_rec_prnc, funded_amnt et loan_amnt.
Et comme importance des variables pour la forêt aléatoire :

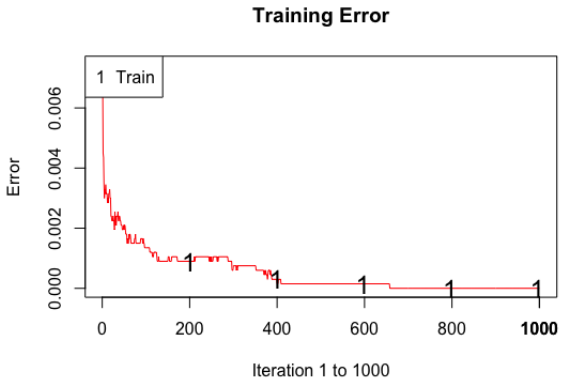


Les variables importantes pour la forêt aléatoire sont recovery, total_rec_prnc, funded_amnt et loan_amnt, total_pymnt et funded_amnt_inv.

D Boosting

Nous commençons par réaliser un simple boosting. Nous décidons ensuite d'améliorer ce boosting en changeant la profondeur maximale d'un noeud, le nombre d'itération ainsi que le paramètre de pénalisation. Nous obtenons comme meilleur boosting celui avec une profondeur maximale d'un noeud terminal à 10, 1000 itérations et un paramètre de pénalisation à 0.01. Nous décidons de comparer l'évolution de l'erreur avec nos différents boosting en fonction du nombre d'itérations :





4) Présentation des résultats

A LDA - QDA - Logit

Nous obtenons les trois matrices de confusions suivantes :

Prévision LDA

Réalité		Négatif	Positif	Total
	Négatif	2025	646	2671
	Positif	218	444	662
	Total	2243	1090	3333

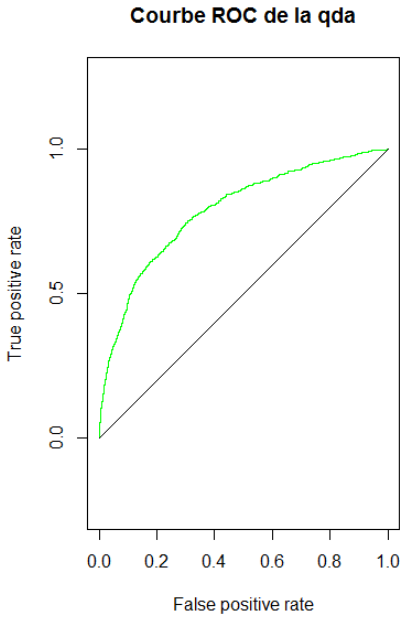
Prévision QDA

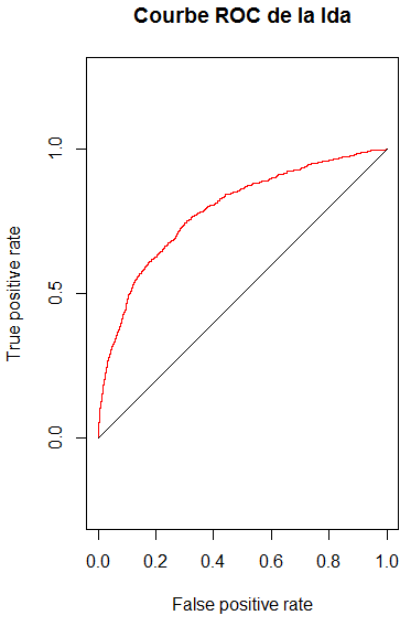
Réalité		Négatif	Positif	Total
	Négatif	2009	662	2671
	Positif	219	443	662
	Total	2228	1105	3333

Prévision Logit

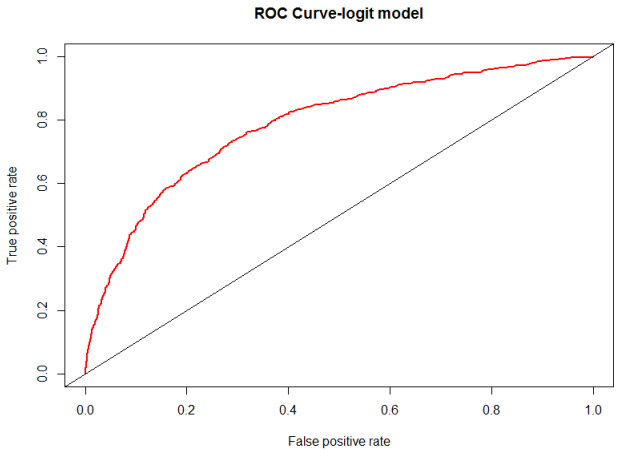
Réalité		Négatif	Positif	Total
	Négatif	1984	687	2671
	Positif	205	457	662
	Total	2189	1144	3333

Nous obtenons également les trois courbes ROC suivantes :





L'aire sous la courbe ROC de la LDA est de 0.7868 et celle de la QDA est de 0.7860.



L'aire sous la courbe ROC du modèle logit est de 0.7862.
Nous avons également les taux d'erreurs suivants :

Modèle	Moyenne des erreurs
LDA	0.2592
QDA	0.2643
Logit	0.2676

Au vu des aires sous les courbes ROC, des taux de faux négatifs et des taux d'erreurs globales, le meilleur modèle paramétrique est le modèle obtenu par LDA.

B Knn - Arbres - RF - Bagging - Boosting

En choisissant nos meilleurs modèles obtenus précédemment, nous obtenons les matrices de confusions suivantes :

Prévision Knn

		Négatif	Positif	Total
Réalité	Négatif	2493	178	2671
	Positif	356	306	662
	Total	2849	484	3333

Prévision Arbre

		Négatif	Positif	Total
Réalité	Négatif	2671	40	2711
	Positif	0	622	622
	Total	2671	662	3333

Prévision RF

		Négatif	Positif	Total
Réalité	Négatif	2667	14	2681
	Positif	4	648	652
	Total	2671	662	3333

Prévision Bagging

		Négatif	Positif	Total
Réalité	Négatif	2665	651	3316
	Positif	6	11	17
	Total	2671	662	3333

Prévision Boosting

		Négatif	Positif	Total
Réalité	Négatif	2671	649	3320
	Positif	0	13	13
	Total	2671	662	3333

Nous décidons à présent de prédire nos erreurs sur les données tests. Nous obtenons les moyennes d'erreurs suivantes :

Modèle	Moyenne des erreurs
Knn	0.1602
Arbres	0.012
Forêt	0.0054
Bagging	0.0051
Boosting	0.0039

Au vu des matrices de confusions et des taux de faux négatifs, le meilleur modèle non paramétrique est le modèle obtenu par boosting.

En conclusion, notre meilleur modèle paramétrique est la LDA et notre meilleur modèle non paramétrique est le boosting. Il apparait que le modèle ayant le moins de faux négatifs et ayant un taux d'erreur le plus bas est le boosting. Nous le considérons donc comme le meilleur modèle pour prédire si un emprunt sera à risque ou non.