

RAPPORT D'ANALYSE DE DONNÉE

Prédiction de Défauts de Paiement

Husseini Mohamad Ali
M. NICOLAS PASQUIER
18 janvier 2025

Objectif : Développer un modèle permettant de prédire le risque de défaut de paiement des clients et l'appliquer à des données inconnues pour effectuer des prédictions.

Sommaire

I - Analyse Exploratoire

II - Pré-traitement des données

III - Clustering des données

IV - Définition de la méthode d'évaluation

V - Définition d'une fonction de perte

VI - Choix du meilleure modèle

VII - Conclusion

Analyse Exploratoire des données

L'ensemble des données dans le fichier projet.csv comprend 11 variables décrivant les caractéristiques financières et démographiques des clients. Voici une analyse détaillée de chacune de ces variables :

1. client (identifiant du client)

Cette variable agit comme un simple identifiant et ne possède aucun lien avec le risque de défaut de paiement. Par conséquent, elle sera exclue des variables prédictives.

2. age (âge en années)

- Données manquantes : 539 valeurs.
- Observations : La distribution des âges montre que la majorité des clients se situent entre 18 et 44 ans, ce qui pourrait indiquer une clientèle active, principalement en âge de travailler. Les valeurs manquantes devront être imputées ou exclues selon leur impact sur le modèle.

3. education (niveau d'éducation)

- Cette variable est ordinaire, représentant le niveau d'éducation par rapport au baccalauréat (par exemple, Bac, Bac+2, Bac+3, etc.).
- Une exploration détaillée est nécessaire pour évaluer son influence sur les défauts de paiement, en analysant la proportion de clients ayant fait défaut pour chaque catégorie de niveau d'éducation.

4. emploi (années avec l'employeur actuel)

- Variable quantitative.
- Cette variable pourrait refléter la stabilité financière des clients, car un nombre élevé d'années passées avec le même employeur peut indiquer une stabilité professionnelle.

5. categorie (catégorie bancaire)

- Observations : Tous les clients ont la même valeur (12).
- Cette absence de variabilité rend la variable inutile pour la prédiction. Elle sera donc exclue de l'analyse.

6. adresse (années passées à l'adresse actuelle)

- Données manquantes : 617 valeurs.
- Observations : La majorité des clients ont résidé entre 3 et 14 ans à leur adresse actuelle. Cette variable pourrait être corrélée avec le défaut de paiement et donc potentiellement informative pour la prédiction. Les valeurs manquantes devront être gérées lors de la phase de pré-traitement.

7. revenus (revenus du foyer en milliers de dollars)

- Une exploration approfondie est nécessaire pour détecter et gérer les anomalies éventuelles (par exemple, des revenus extrêmement élevés ou erronés). L'impact de cette variable en tant que prédicteur sera également évalué.

8. debcred (ratio débit/crédit, multiplié par 100)

- Un ratio élevé pourrait être un indicateur de difficultés financières, car il reflète une forte dépendance aux crédits par rapport aux ressources disponibles.

9. debcarte (débit carte de crédit en milliers de dollars)

- Observations : Cette variable quantitative peut refléter la tendance des clients à utiliser leur crédit de manière récurrente. Il serait pertinent de vérifier son corrélation avec les défauts de paiement.

10. autres (autres dettes en milliers de dollars)

- Décrit le montant des dettes supplémentaires au-delà des cartes de crédit, offrant une vision plus globale de la dette totale.

11. defaut (variable cible, défaut de paiement)

- Variable binaire (Oui/Non) indiquant si un défaut de paiement est survenu.
- Une attention particulière doit être accordée à la distribution des classes pour identifier d'éventuels déséquilibres (par exemple, une sur-représentation de clients n'ayant pas fait défaut).

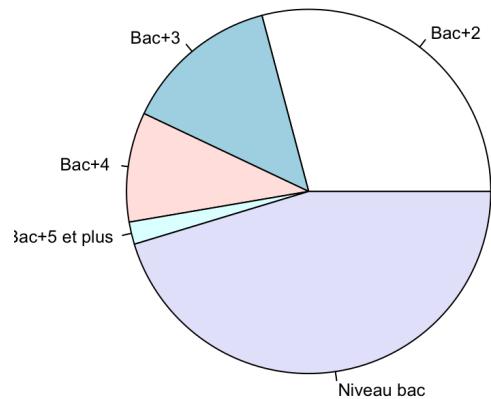
client	age	education	emploi	categorie	adresse	revenus	debcrid	debcarte	autres	defaut
X Min. :1201	Min. : 18	Bac+2 :1748	Min. : 0	Min. :12	Min. : 0	Min. : 12	Min. : 0	Min. : 0	Min. : 0	Non:4331
X.1 1st Qu.:2701	1st Qu.: 29	Bac+3 : 830	1st Qu.: 2	1st Qu.:12	1st Qu.: 3	1st Qu.: 25	1st Qu.: 5	1st Qu.: 0	1st Qu.: 1	Oui:1669
X.2 Median :4200	Median : 36	Bac+4 : 584	Median : 6	Median :12	Median : 7	Median : 36	Median : 9	Median : 1	Median : 2	NA
X.3 Mean :4200	Mean :122	Bac+5 et plus: 119	Mean : 8	Mean :12	Mean :110	Mean : 51	Mean :10	Mean : 2	Mean : 3	NA
X.4 3rd Qu.:5700	3rd Qu.: 44	Niveau bac :2719	3rd Qu.:12	3rd Qu.:12	3rd Qu.: 14	3rd Qu.: 57	3rd Qu.:14	3rd Qu.: 2	3rd Qu.: 4	NA
X.5 Max. :7200	Max. :999	NA	Max. :63	Max. :12	Max. :999	Max. :2462	Max. :45	Max. :140	Max. :417	NA

Valeurs manquantes

Valeurs extrêmes à modifier

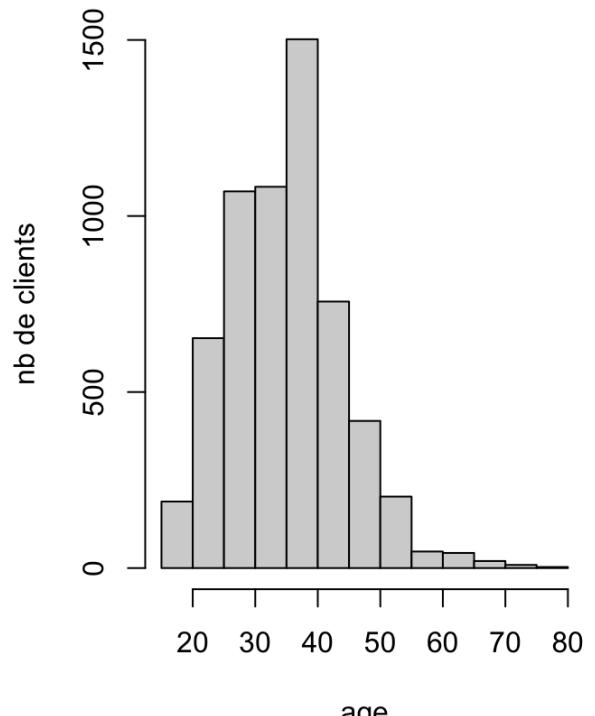
NB : Les graphiques suivants présentent les résultats après les modifications apportées pour améliorer la clarté des données.

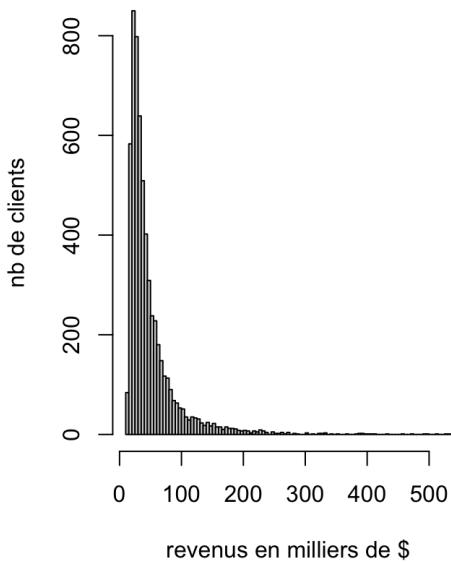
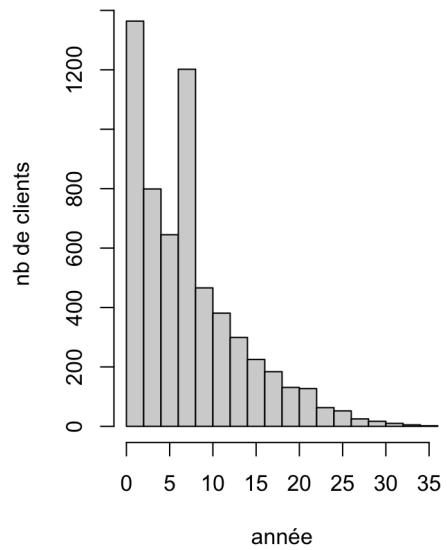
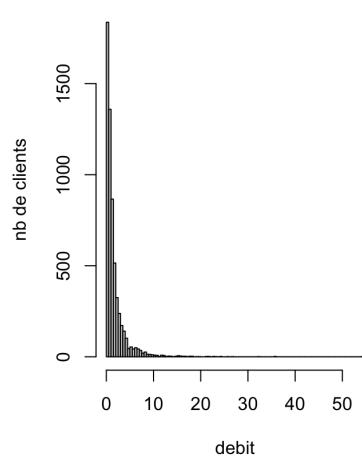
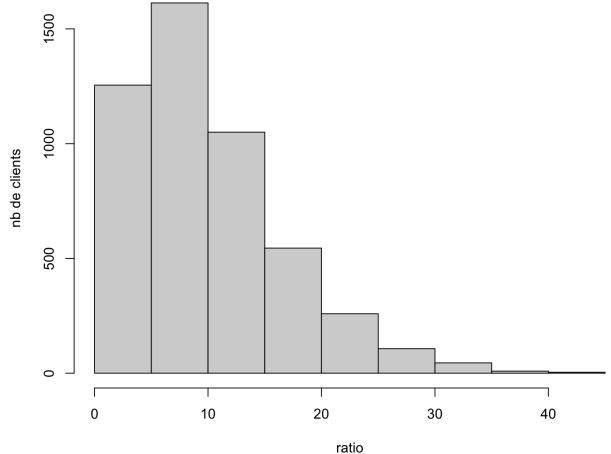
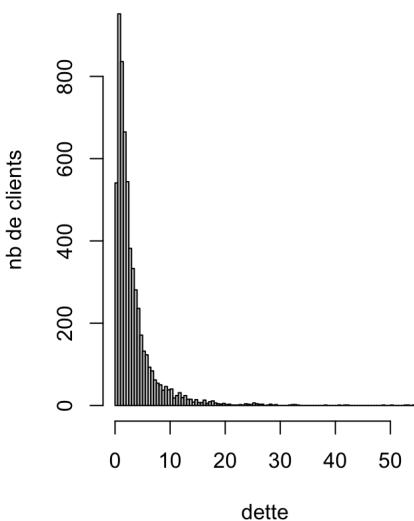
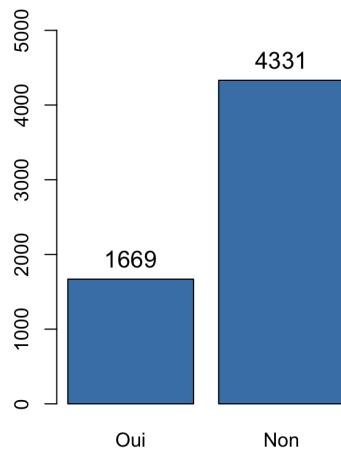
Repartition des niveaux d'educations



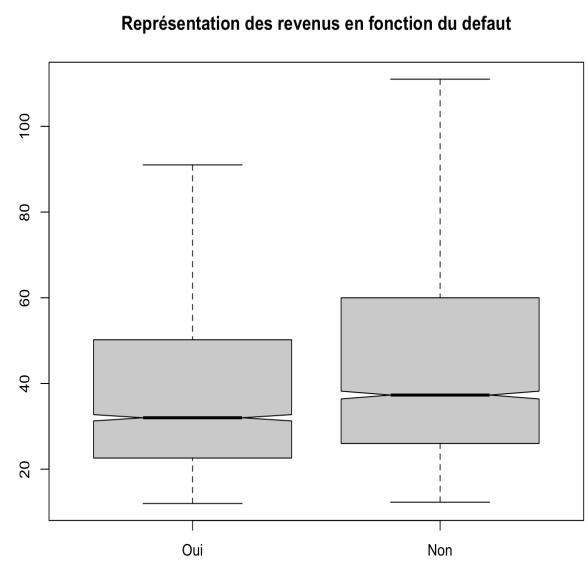
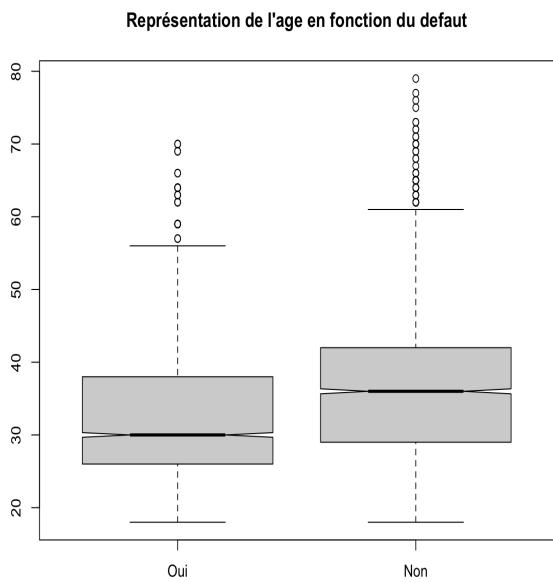
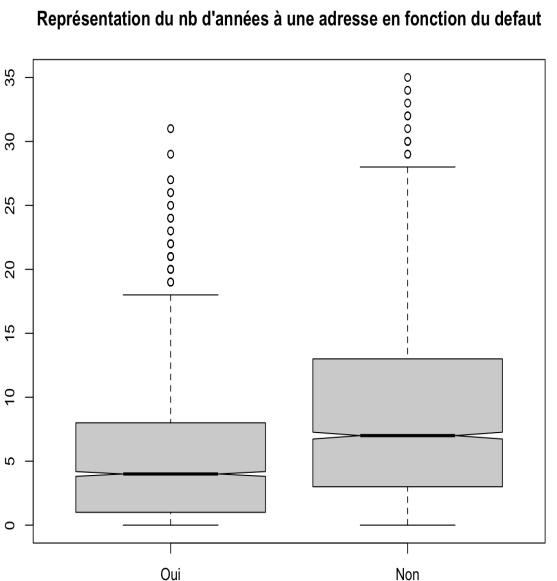
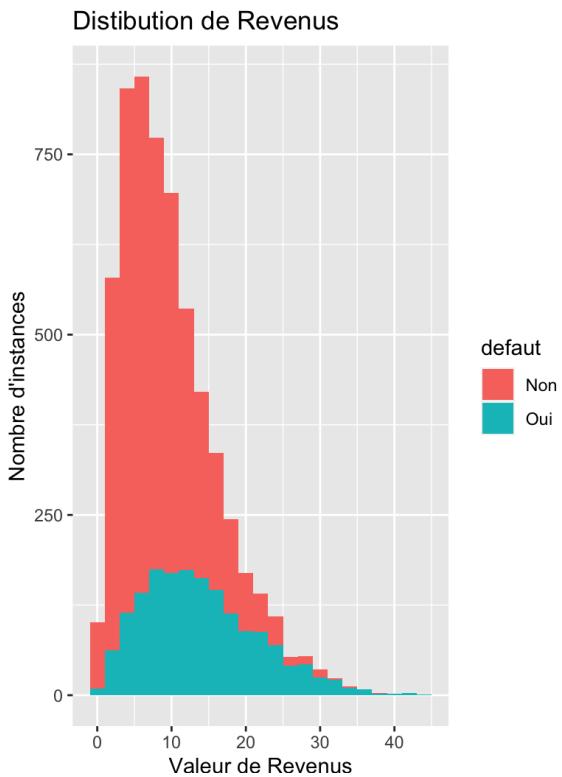
On peut voir la grande majorité ayant un niveau bac

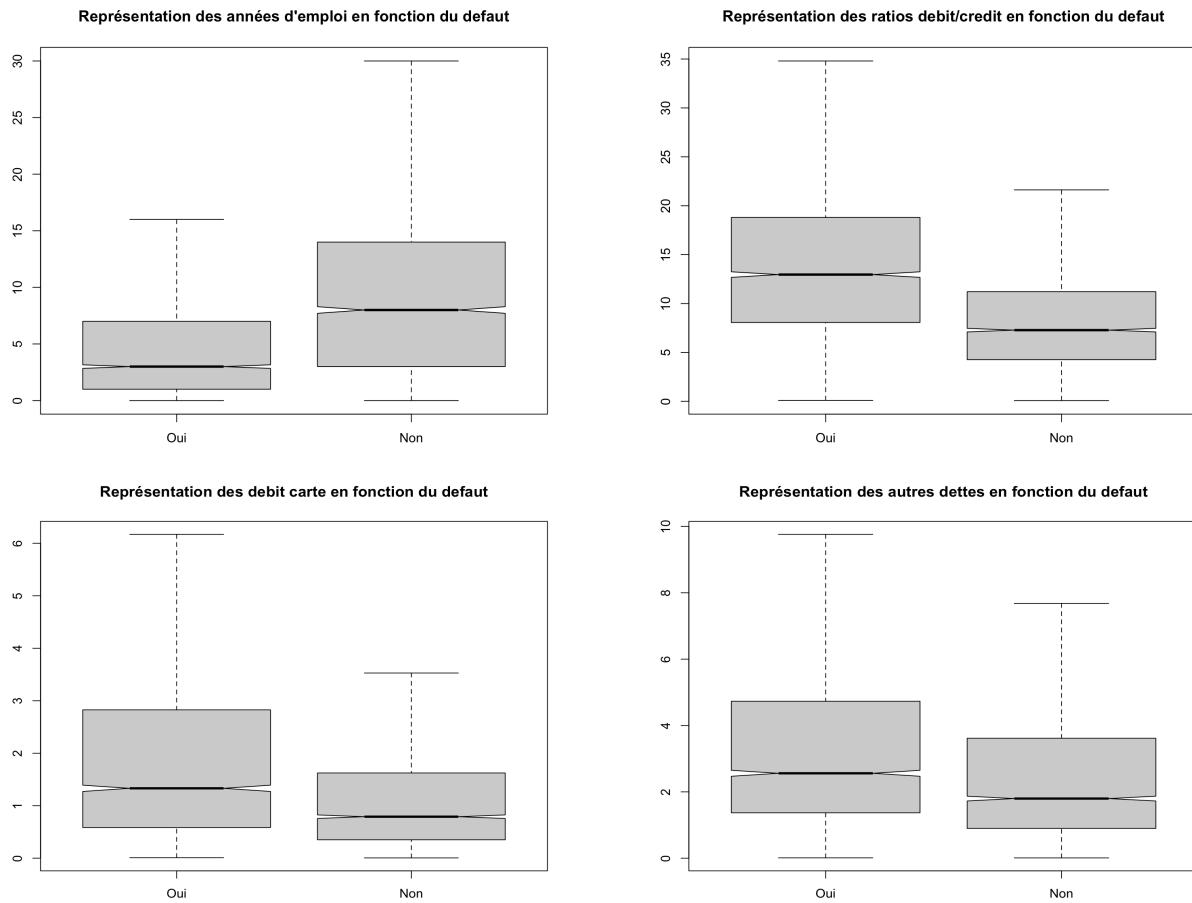
Repartition des ages



Repartition des revenus**Nombre d'années à l'adresse actuelle****Débit carte de crédit en milliers de \$****Ratio Débit/Crédit (x100)****Autres dettes en milliers de \$****Nb de default**

J'ai représenté les graphiques des données. Maintenant regardons les graphiques en fonction des défauts.





Toutes les variables représentées sont importantes pour construire notre modèle car dans chaque graphique, les deux boxplots se chevauchent peu ou ont des médianes très différentes ce qui suggère que la variable étudiée pourrait être un bon indicateur pour prédire la classe default.

Pré-traitement des données

Si nous supprimons les valeurs manquantes des deux variables, il ne nous reste que 4887 données des 6000 données initiales ce qui représente 18.5% des données. C'est mieux de supprimer une colonne entière avec beaucoup de valeurs manquantes mais elle peut être importante pour notre modèle de prédiction.

Comme j' ai 2 variables avec des valeurs manquantes (age et adresse), je dois voir pour chacune s' il faut supprimer la colonne ou supprimer / modifier les observations manquantes.

C'est pourquoi j'effectue un test de Student pour chacune des 2 variables après avoir supprimer les données manquantes avec

H0 : Les moyennes d' **adresse/age** sont égales pour **defaut = Oui** et **defaut = Non**.

H1 : Les moyennes d' adresse/age sont différentes.

Alpha = 0.05

Si la p-valeur est grande (> alpha), cela signifie qu'il n'y a pas suffisamment de preuves pour rejeter l'hypothèse H0. Par conséquent, cette variable pourrait ne pas être corrélée avec la cible pour cette répartition particulière, mais son importance pourrait être réévaluée dans d'autres contextes. Sinon, je garde la colonne i.e. la variable et je modifie les valeurs manquantes en les remplaçant par la moyenne de cette variable après filtration des données.

Test Student pour la variable adresse :

```
Welch Two Sample t-test

data: data_filter$adresse[data_filter$defaut == "Oui"] and data_filter$adresse[data_filter$defaut == "Non"]
t = -18, df = 3259, p-value <2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-3.7 -2.9
sample estimates:
mean of x mean of y
5.3      8.6
```

Comme nous pouvons le voir la p-valeur est trop petite ($<2e-16$) donc je rejette H_0 avec un risque alpha et je garde H_1 , donc la variable adresse est importante pour notre modèle. Ainsi, je modifie les valeurs manquantes (la modification est faite après).

Test Student pour la variable age:

```
Welch Two Sample t-test

data: data_filter$age[data_filter$defaut == "Oui"] and data_filter$age[data_filter$defaut == "Non"]
t = -15, df = 2655, p-value <2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-5.0 -3.8
sample estimates:
mean of x mean of y
32       36
```

Pareil avec cette variable. La p-valeur est plus petit que alpha donc je rejette H_0 avec un risque alpha et je garde H_1 .

Je peux soit remplacer ces valeurs manquantes par les moyennes ou les supprimer. Mais comme les valeurs manquantes représentent 18.5% des données, je vais les remplacer par les moyennes pour ne pas perdre autant de données.

Modification des données manquantes :

Je remplace les valeurs manquantes de la colonne age par la médiane, et de même pour la colonne adresse. J'élimine certaines valeurs extrêmes qui sont isolées, comme dans la colonne revenus. En supprimant les colonnes client et categorie, je crée deux nouvelles colonnes missing_age et missing_adresse pour suivre l'emplacement des valeurs manquantes d'origine et ces nouvelles variables seront données aux modèles.

▲	age	education	emploi	adresse	revenus	debcrid	debcarte	autres	defaut	missing_age	missing_adresse
X	Min. :18	Bac+2 :1748	Min. :0	Min. :0	Min. :12	Min. :0	Min. :0	Min. :0	Non:4329	Min. :0.00	Min. :0.0
X.1	1st Qu.:29	Bac+3 :830	1st Qu.: 2	1st Qu.: 3	1st Qu.: 25	1st Qu.: 5	1st Qu.: 0	1st Qu.: 1	Oui:1668	1st Qu.:0.00	1st Qu.:0.0
X.2	Median :36	Bac+4 :584	Median : 6	Median : 7	Median : 36	Median : 9	Median : 1	Median : 2	NA	Median :0.00	Median :0.0
X.3	Mean :35	Bac+5 et plus: 116	Mean : 8	Mean : 8	Mean : 50	Mean :10	Mean : 2	Mean : 3	NA	Mean :0.09	Mean :0.1
X.4	3rd Qu.:41	Niveau bac :2719	3rd Qu.:12	3rd Qu.:11	3rd Qu.: 57	3rd Qu.:14	3rd Qu.: 2	3rd Qu.: 4	NA	3rd Qu.:0.00	3rd Qu.:0.0
X.5	Max. :79	NA	Max. :63	Max. :35	Max. :533	Max. :45	Max. :54	Max. :63	NA	Max. :1.00	Max. :1.0

Clustering des données

J'ai choisi les variables suivantes pour le clustering : age, education, emploi, adresse, revenus, debcred, debcarte, autres.

17979006 dissimilarities, summarized :

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.00	0.14	0.20	0.20	0.26	0.74

Metric : mixed ; Types = I, N, I, I, I, I, I, I, N, I, I

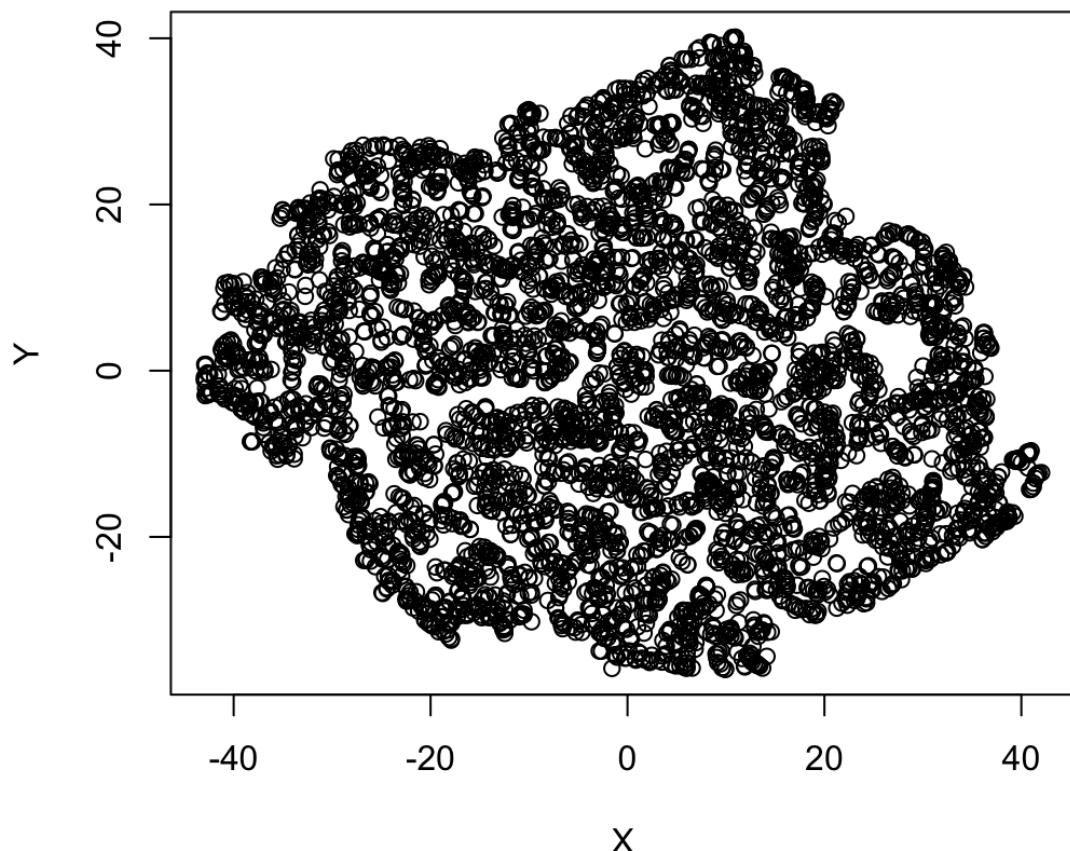
Number of objects : 5997

Voici la matrice de distance pour les instances du data frame data_filter en utilisant la fonction daisy() du package cluster qui permet de traiter les données hétérogènes.

Le nombre de dissimilarités représente 2 parmi n avec n = 5997 ce qui équivaut à $5997 \times (5996)/2 = 17979006$.

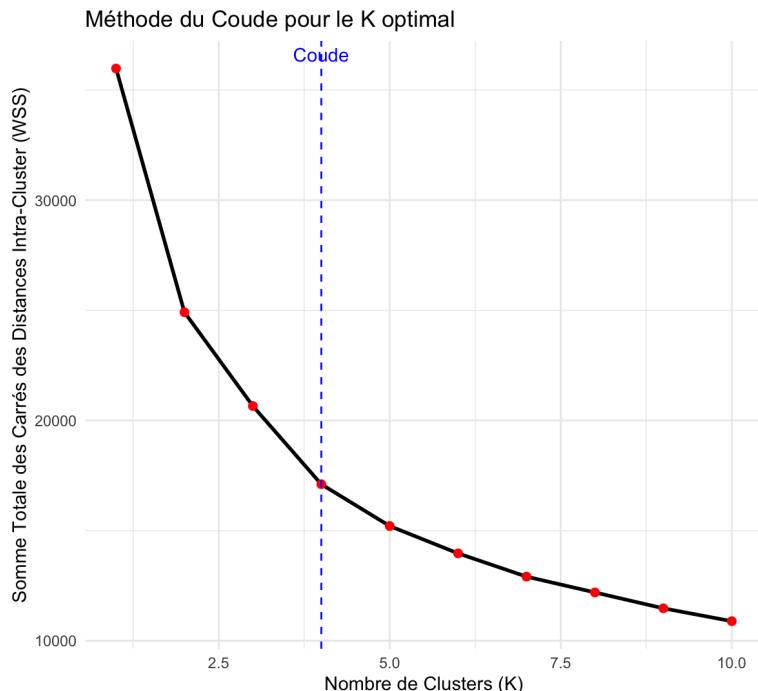
D'après le 1er et 3eme quartile, on a 25% des données qui sont à distance plus petite que 0.14 et 75% sont à distance plus petite que 0.26.

Comme les variables ont des domaines différents, je vais les normaliser (à l'aide de la fonction scale de R) pour que ça n'affecte pas les calculs des distances.



Ceci représente les données en 2D en fonction des variables choisies auparavant à l'aide de t-SNE.

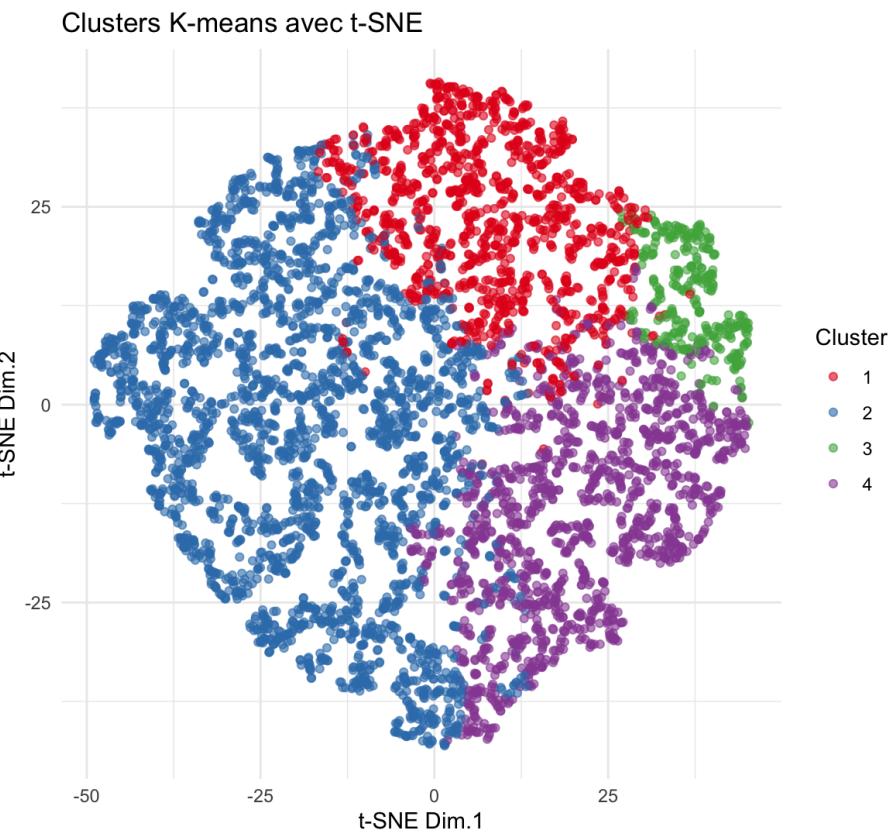
J'applique la méthode **Elbow** pour trouver le meilleure K pour le K-nearest neighbors.



Pour déterminer le nombre optimal de clusters K en utilisant **la méthode du coude (Elbow Method)** à partir de ce graphique , nous devons identifier le point où la diminution de **la somme totale des carrés des distances intra-cluster (WSS)** commence à ralentir de manière significative.

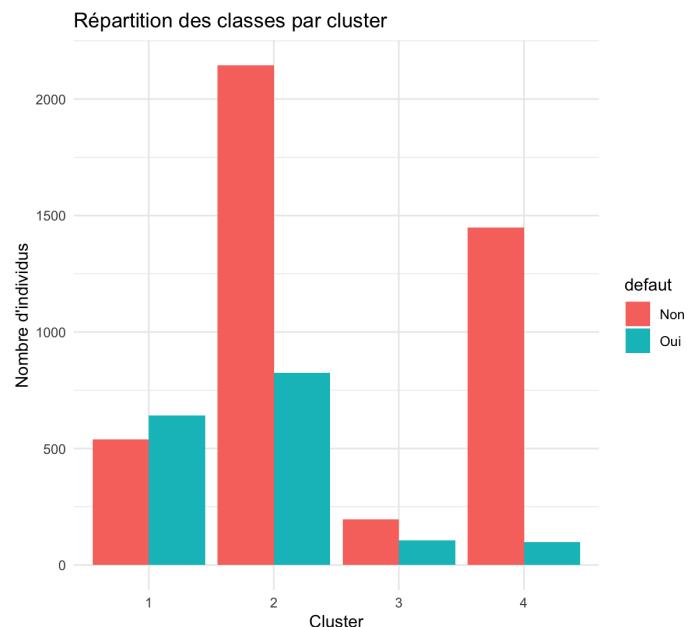
En observant le graphique, nous constatons que la diminution de la WSS est très rapide jusqu'à K = 4. Au-delà de ce point, la diminution devient plus graduelle. Par conséquent, le coude semble se situer autour de K = 4.

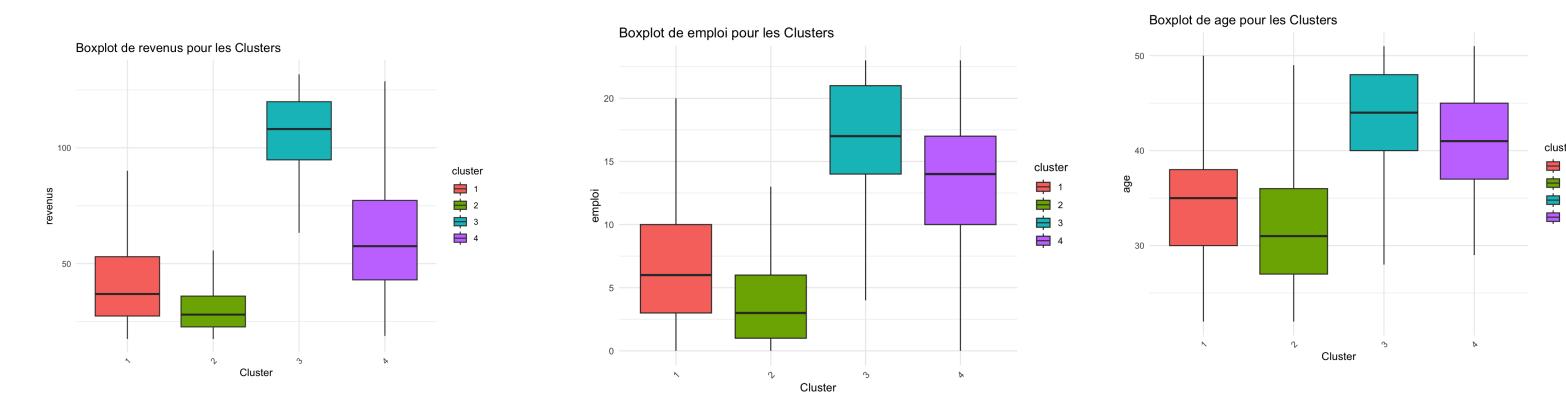
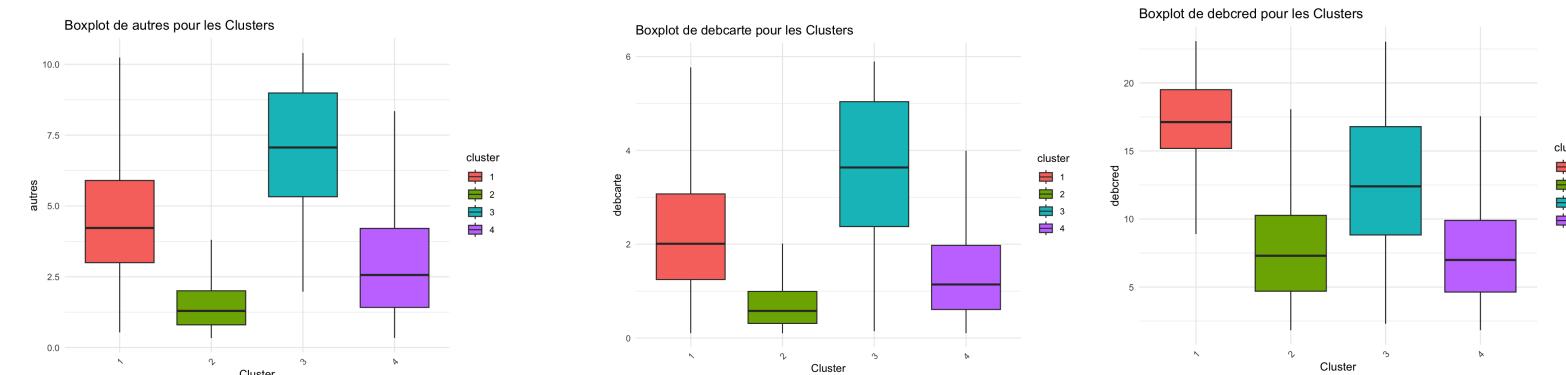
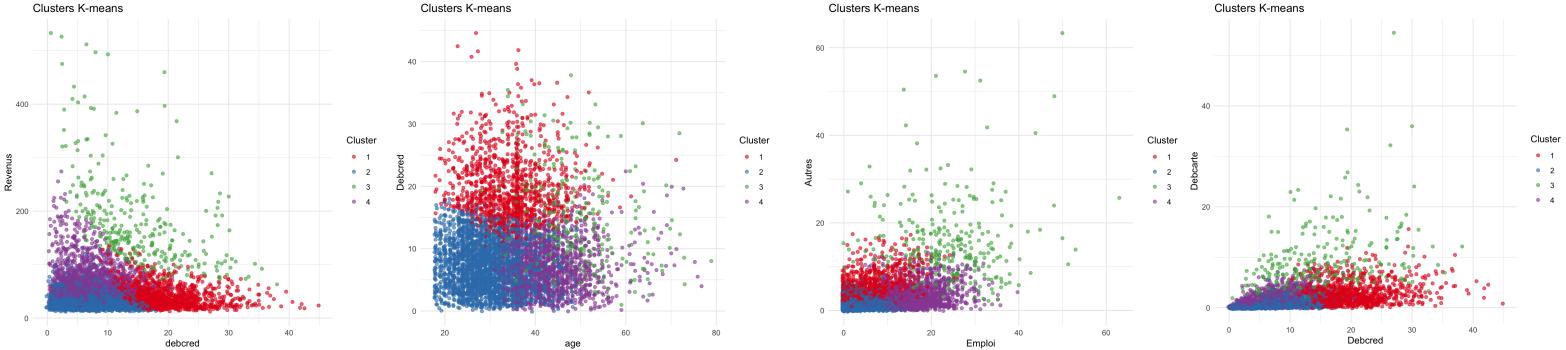
Comme j' ai le K optimal, je peux maintenant regrouper les données dans K groupes c'est-à-dire 4 groupes. Voici le graphique qui nous montre les clusters après l'application de la méthode t-SNE:



Les clusters sont relativement bien séparés, bien que certains chevauchements soient visibles, notamment entre les clusters rouge et bleu, ainsi qu'entre les clusters violet et bleu. Le deuxième cluster est le plus dominant dans ce graphique. Les points sont plus denses dans certaines régions, indiquant des zones où les individus sont plus similaires entre eux.

Regardons aussi les effectifs de oui et de non dans chaque cluster :





Clusters	Caractéristiques
Cluster 1 (Rouge)	Population avec des revenus et un emploi faibles, mais des dettes modérées à élevées. Majoritairement jeune ou d'âge moyen.
Cluster 2 (Vert)	Groupe le plus défavorisé, avec des valeurs basses pour toutes les variables (faibles revenus, emploi, et dettes).
Cluster 3 (Bleu)	Profil le plus favorisé, avec des revenus élevés, un emploi important, et des dettes modérées à élevées. Représente une population active et plutôt âgée.
Cluster 4 (Violet)	Groupe intermédiaire avec une dispersion importante. Revenus modérés, dettes et emploi moyens, et âge relativement avancé.

Ces graphiques représentent les clusters des données obtenus par **t-SNE**. Les boxplots représentent les caractéristiques de chaque cluster. Je peux passer maintenant aux modèles de prédictions.

Définition de la méthode d'évaluation

Pour commencer, je divise les données modifiées du fichier projet.csv comme indiqué dans la partie pré-traitement en 2 partie:

80% des données dans le working_set, et 20% dans le holdout.

Pour évaluer la performance des différents modèles de prédiction, nous avons utilisé **la méthode de validation croisée répétée stratifiée dans le working_set (repeated cross-validation)**. Cette méthode permet de diviser les données en plusieurs ensembles d'entraînement et de test, et de répéter ce processus plusieurs fois pour obtenir des estimations plus robustes de la performance des modèles.

Configuration de la validation croisée

Nous avons configuré la validation croisée avec les paramètres suivants :

- Méthode : **repeatedcv** (validation croisée répétée)
- Nombre de plis (folds) : **10**
- Fonction de résumé : **twoClassSummary** (pour calculer les métriques ROC et autres)
- Probabilités de classe : **TRUE** (pour calculer les probabilités de classe)
- Sauvegarde des prédictions : **all** (pour sauvegarder toutes les prédictions pour une évaluation ultérieure)
- Index : stratification des données avec 5 répétition

Modèles évalués

Nous avons évalué plusieurs modèles de prédiction en utilisant la configuration de validation croisée décrite ci-dessus. Voici les modèles évalués grâce au package caret :

Modèle	Méthode	Description	Tuning
model1	glm	Régression logistique pour la classification binaire.	-
model2	rf	Modèle d'ensemble basé sur des arbres de décision.	-
model2.2	ranger	Modèle d'ensemble basé sur des arbres de décision (tuned).	mtry, splitrule, min.node.size
model3	svmRadial	Support Vector Machine avec noyau radial.	-
model3.2	svmRadial	Support Vector Machine avec noyau radial (tuned).	tuneLength
model3.3	svmLinear	Support Vector Machine avec noyau linéaire.	-
model4	rpart	Arbre de décision basé sur des règles de décision.	-
model4.1	rpart	Arbre de décision basé sur des règles de décision (Gini split, minbucket = 10).	cp, split, minbucket
model4.2	rpart	Arbre de décision basé sur des règles de décision (Gini split, minbucket = 9).	cp, split, minbucket
model4.3	rpart	Arbre de décision basé sur des règles de décision (Information split, minbucket = 5).	cp, split, minbucket
model4.4	rpart	Arbre de décision basé sur des règles de décision (Information split, minbucket = 9).	cp, split, minbucket
model5	knn	Modèle basé sur la distance entre les instances.	tuneLength
model5.1	knn	Modèle basé sur la distance entre les instances (tuned).	tuneLength
model6	glmnet	Régression logistique avec régularisation LASSO.	alpha, lambda
model7	C5.0	Modèle d'arbre de décision avec options de boosting.	trials, model, winnow, minCases, noGlobalPruning
model7.1	C5.0	Modèle d'arbre de décision avec options de boosting (noGlobalPruning = TRUE).	trials, model, winnow, minCases, noGlobalPruning
model7.2	C5.0	Modèle d'arbre de décision avec options de boosting (minCases = 4).	trials, model, winnow, minCases, noGlobalPruning
model7.3	C5.0	Modèle d'arbre de décision avec options de boosting (minCases = 4, noGlobalPruning = TRUE).	trials, model, winnow, minCases, noGlobalPruning
model8	rpart	Arbre de décision basé sur des règles de décision.	cp, split, minbucket
model8.1	rpart	Arbre de décision basé sur des règles de décision (minbucket = 4).	cp, split, minbucket

Avec un total de **20 modèles**, nous allons évaluer les modèles.

Évaluation des modèles

Pour chaque modèle, nous avons calculé les métriques suivantes :

- AUC-ROC : Area Under the Curve - Receiver Operating Characteristic, qui mesure la capacité du modèle à distinguer entre les classes.
- Sensibilité (Sensitivity) : Proportion de vrais positifs correctement identifiés.
- Spécificité (Specificity) : Proportion de vrais négatifs correctement identifiés.

Calcul de l'AUC-ROC

Nous avons calculé l'AUC-ROC pour chaque modèle en utilisant les prédictions sauvegardées lors de la validation croisée. Voici les étapes suivies :

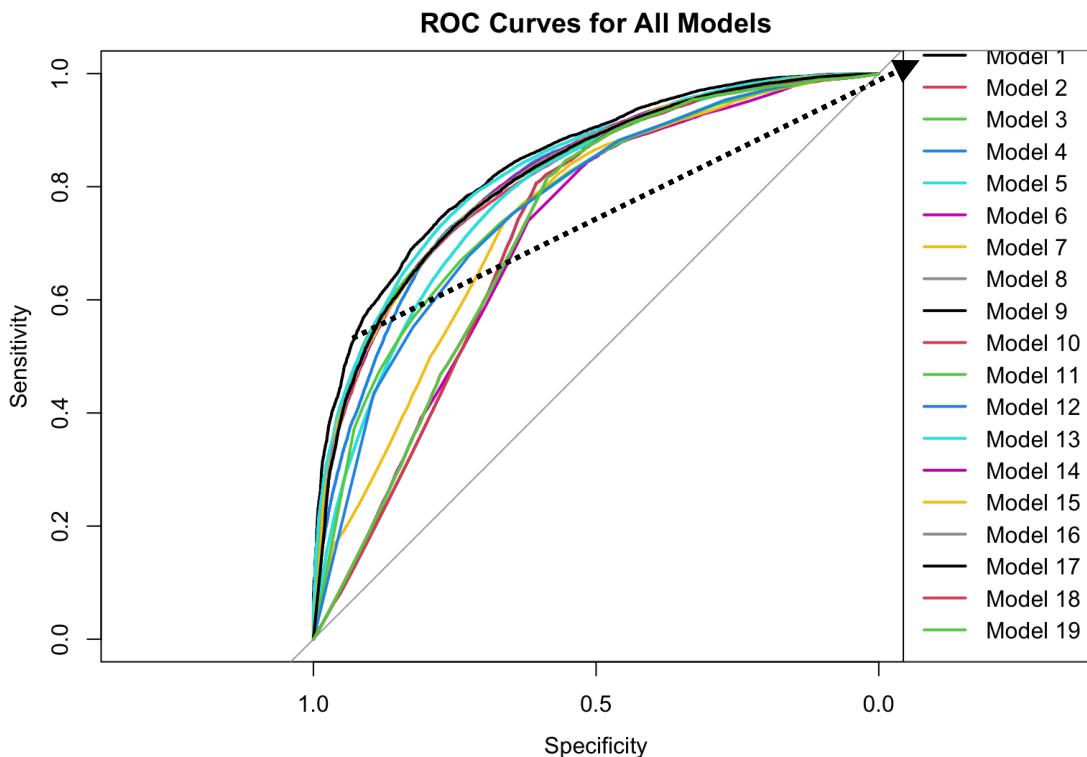
1. Extraction des prédictions et des valeurs réelles.
2. Calcul de la courbe ROC pour chaque modèle.
3. Calcul de l'AUC moyen pour chaque modèle.

L'AUC est calculé en prenant la moyenne sur tous les plis et les répétitions.

En évaluant plusieurs variations des modèles Random Forest et SVM, nous avons pu identifier les configurations optimales pour notre jeu de données. Cette approche rigoureuse nous permet de sélectionner les modèles les plus performants et robustes pour la prédiction des défauts de paiement, en tenant compte des spécificités et des complexités de nos données.

Voici les courbes **ROC** des modèles :

Modèle	AUC/ROC	Sensibilité	Spécificité	Paramètres Finaux
Generalized Linear Model	0.84	0.91	0.49	
Random Forest	0.82	0.93	0.42	mtry = 2
Random Forest	0.83	0.90	0.47	mtry = 4, splitrule = gini, min.node.size = 15
Support Vector Machines with Radial Kernel	0.82	0.93	0.41	sigma = 0.057, C = 0.25
Support Vector Machines with Radial Kernel	0.81	0.93	0.41	sigma = 0.06, C = 0.25
CART	0.72	0.91	0.43	cp = 0.013
CART	0.79	0.88	0.48	cp = 0.001
CART	0.72	0.92	0.43	cp = 0.01
CART	0.73	0.92	0.43	cp = 0.01
CART	0.73	0.92	0.42	cp = 0.01
k-Nearest Neighbors	0.78	0.89	0.43	k = 11
k-Nearest Neighbors	0.78	0.89	0.46	k = 7
glmnet	0.84	0.91	0.49	alpha = 1, lambda = 0.0022
C5.0	0.83	0.90	0.49	trials = 50, model = tree, winnow = FALSE
C5.0	0.83	0.89	0.50	trials = 50, model = tree, winnow = FALSE
C5.0	0.83	0.90	0.48	trials = 50, model = tree, winnow = FALSE
C5.0	0.83	0.90	0.49	trials = 50, model = tree, winnow = FALSE
CART	0.72	0.92	0.42	cp = 0.01
CART	0.72	0.92	0.42	cp = 0.01



Je trace **le taux de vrais positifs (sensibilité)** en fonction du **taux de faux positifs (1 - spécificité)** à divers seuils de classification. Nous privilégions le modèle qui maximise la sensibilité.

En tant que banque, les pertes financières associées à un défaut de paiement **sont significativement plus élevées** que les coûts liés à un refus de prêt à un client qui aurait pu rembourser. Par conséquent, il est crucial de minimiser les faux négatifs pour réduire les pertes financières.

Pour un prêt de **10 000 euros** avec un taux d'intérêt de 5 % sur 1 an, le revenu brut serait environ **10 500 euros en cas de non default**. Si le client fait défaut et aucune garantie n'existe, la perte est quasi-totale. Mais si la banque refuse le prêt, la banque perd **500 euros** (5 % sur 10 000 euros) qui est beaucoup moins.

C'est pourquoi nous utilisons une fonction de perte pondérée, qui prend en compte à la fois les coûts relatifs des faux positifs et des faux négatifs, ainsi que la prévalence de la classe minoritaire.

Fonction de Perte

Le modèle sélectionné après la validation croisée parmi les autres modèles est le meilleure modèle probabiliste c'est-à-dire le modèle qui génère la meilleure probabilité ($P(\text{defaut}|X)$ où X vecteur des variables). Pour générer une classe, il faudra choisir un seuil S tel que $P(\text{defaut}|X) > S$ donne la classe Oui sinon le défaut est Non. La courbe ROC nous montre les différentes sensibilités et spécificités pour chaque seuil. Pour choisir un seuil, il faudra une fonction de perte adaptée à notre problématique de prédiction de défaut. Les coûts ont été fixés comme suit :

FP : Coût d'un faux positif, fixé à 1.

FN : Coût d'un faux négatif, fixé à 3 (trois fois le coût d'un faux positif).

Le rapport de coût relatif est défini comme suit :

Coût relatif = FN/FP = 3

Nous avons également calculé la prévalence de la classe positive :

Prévalence = nombre de défauts/ nombre total d'observations = 0.28

Calcul de la Perte Attendue

La perte attendue pour chaque modèle a été calculée en utilisant la fonction de perte définie ci-dessus. Cette perte prend en compte à la fois la sensibilité et la spécificité des modèles, pondérées par les coûts des faux positifs et des faux négatifs. Nous avons utilisé la fonction de coût de Youden Index et on cherche le seuil qui maximise cette fonction de perte sur les différents plis de validation croisée.

Perte Attendu = FP x Cout FP + FN x Cout FN

critère de performance pondéré

$J = \text{sensibilité} + \text{spécificité} \times (1-\text{prévalence}) / (\text{Coût relatif} \times \text{prévalence})$

Modèle	Seuil Optimal	Perte Attendue	AUC/ROC	Description
Generalized Linear Model	0.24	0.37	0.84	Model 1
Random Forest	0.19	0.40	0.82	Model 2
Random Forest	0.23	0.39	0.82	Model 2.2 (mtry = 4, splitrule = gini, min.node.size = 15)
Support Vector Machines	0.18	0.41	0.82	Model 3
Support Vector Machines	0.21	0.41	0.82	Model 3.2 (sigma = 0.057, C = 0.25)
CART	0.28	0.49	0.72	Model 4
CART	0.21	0.47	0.79	Model 4.1 (cp = 0.001)
CART	0.28	0.49	0.72	Model 4.2 (cp = 0.01)
CART	0.26	0.47	0.73	Model 4.3 (cp = 0.01)
CART	0.26	0.47	0.73	Model 4.4 (cp = 0.01)
k-Nearest Neighbors	0.23	0.45	0.79	Model 5
k-Nearest Neighbors	0.26	0.46	0.78	Model 5.1 (k = 7)
glmnet	0.25	0.39	0.84	Model 6
C5.0	0.33	0.39	0.83	Model 7
C5.0	0.33	0.39	0.83	Model 7.1 (winnow = FALSE)
C5.0	0.28	0.40	0.83	Model 7.2 (winnow = FALSE)
C5.0	0.33	0.40	0.83	Model 7.3 (winnow = FALSE)
CART	0.28	0.49	0.72	Model 8
CART	0.28	0.49	0.72	Model 8.1 (cp = 0.01)

Pour chaque modèle, je calcule les AUC, les pertes attendus et le seuil optimal pour chaque modèle. La sensibilité est cruciale dans notre cas car elle représente la capacité du modèle à identifier correctement les cas positifs qui ont un cout plus important pour la banque.

Choix du meilleure modèle

Pour sélectionner le meilleur modèle, je compare les AUC (moyenne obtenue après la validation croisée) des modèles et je trouve que le **GLM**, **Régression Logistic** est le meilleure pour minimiser les pertes et prédire correctement la classe defaut.

1. Entraînement du Modèle sur l'Ensemble working_set:

Modèle	AUC/ROC
Model 1	0.84

Voici ce que j'obtiens après l'entraînement du modèle sur tout l'ensemble **working_set**.

On évalue le modèle sur le holdout.

2. Prédiction sur le holdout et matrice de confusion :

	Prédit Non		Prédit Oui	
Réel Non	619 / 51.6%		66 / 5.5%	
Réel Oui	247 / 20.6%		268 / 22.3%	
Modèle	AUC/ROC	Sensibilité	Spécificité	Seuil Optimale
Model 1	0.83	0.56	0.89	0.29
Modèle	AUC/ROC	Sensibilité	Spécificité	Seuil
Model 1	0.83	0.67	0.81	0.5

Le modèle sélectionné (**Model GLM**) a été évalué de manière rigoureuse en utilisant une fonction de perte qui prend en compte les coûts des faux positifs et des faux négatifs. Les résultats montrent que ce modèle offre un bon compromis entre sensibilité et spécificité, avec une perte attendue acceptable.

En comparant les résultats en prenant 2 seuils différents (le premier trouvé dans la validation croisée et le seuil normal), je trouve que l'AUC ne change pas et la sensibilité est plus élevée avec le seuil normal. **Mais nous pouvons faire mieux !!**

PARTIE 2

Dans la première partie, j'avais **1669 données** avec **défaut Oui** et **4331** avec **défaut Non**. Pour améliorer la performance de nos modèles, nous allons traiter les données manquantes et appliquer la technique **SMOTE** (Synthetic Minority Over-sampling Technique) **pour équilibrer les classes sur les données d'entraînement des modèles.**

Dans un premier lieu, nous utilisons l'**imputation par k-plus proches voisins** (kNN) pour traiter les données manquantes en appliquant **la méthode kNN** de la librairie **VIM**.

	age	education	emploi	adresse	revenus	debcrid	debcarte	autres	defaut
X	Min. :18	Min. :1.0	Min. : 0	Min. : 0	Min. : 12	Min. : 0	Min. : 0	Min. : 0	Non:4331
X.1	1st Qu.:28	1st Qu.:1.0	1st Qu.: 2	1st Qu.: 3	1st Qu.: 25	1st Qu.: 5	1st Qu.: 0	1st Qu.: 1	Oui:1669
X.2	Median :34	Median :3.0	Median : 6	Median : 6	Median : 36	Median : 9	Median : 1	Median : 2	NA
X.3	Mean :35	Mean :3.2	Mean : 8	Mean : 8	Mean : 51	Mean :10	Mean : 2	Mean : 3	NA
X.4	3rd Qu.:41	3rd Qu.:5.0	3rd Qu.:12	3rd Qu.:11	3rd Qu.: 57	3rd Qu.:14	3rd Qu.: 2	3rd Qu.: 4	NA
X.5	Max. :79	Max. :5.0	Max. :63	Max. :35	Max. :2462	Max. :45	Max. :140	Max. :417	NA

Dans un deuxième lieu, nous utilisons l'algorithme **SMOTE** sur le working_set dans le **trainControl pour équilibrer les classes en générant des exemples synthétiques pour la classe minoritaire qui est « Oui ».**

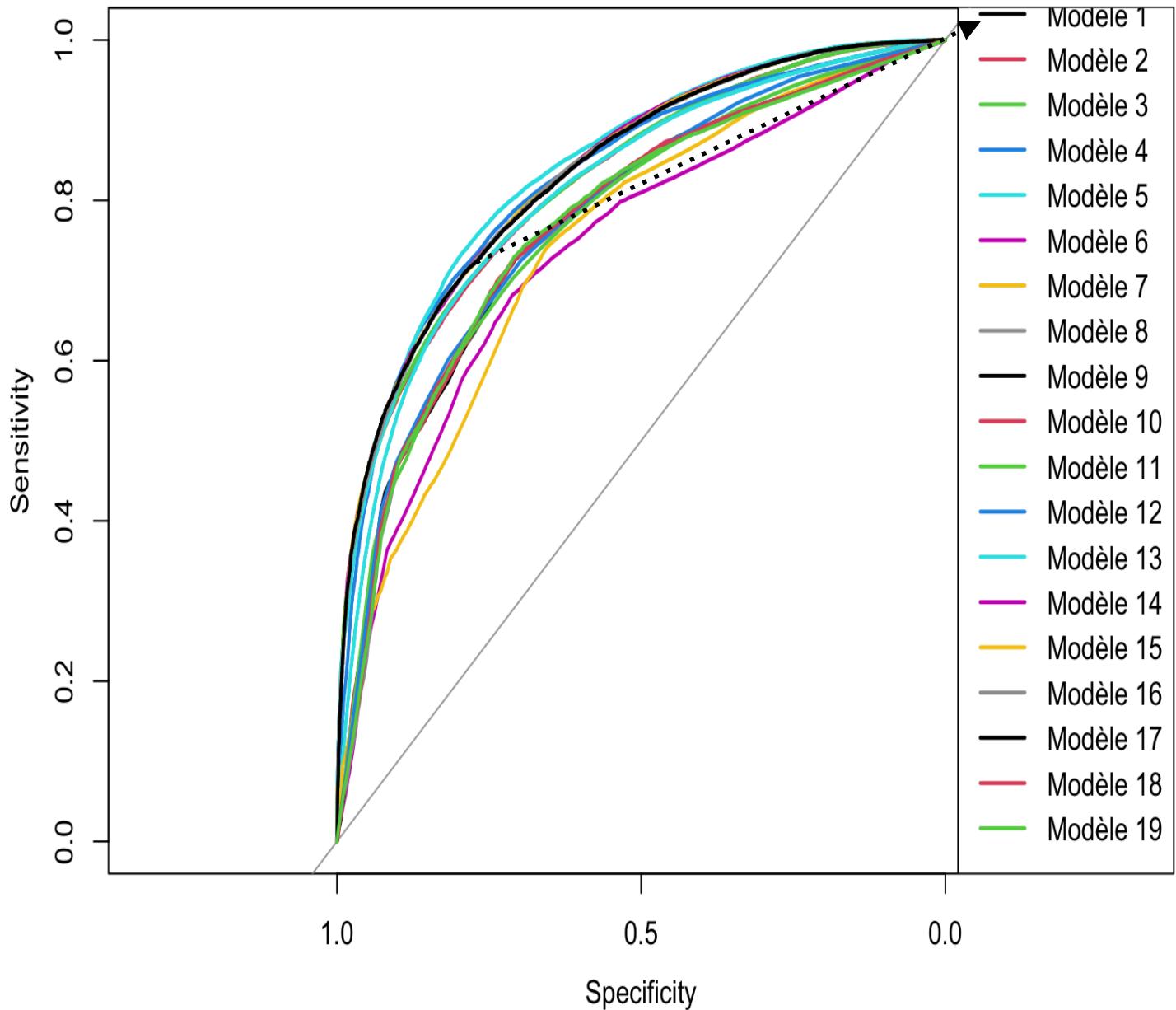
Si nous appliquons Smote sur le working_set directement, nous risquons de créer des exemples synthétiques qui pourraient ne pas être représentatifs, ce qui pourrait conduire à un **Data Leakage**. Les exemples synthétiques créés peuvent introduire un biais dans les données. Par contre, si nous l'appliquons dans le trainControl, les données de validation (test) ne sont pas modifiées, ce qui permet une évaluation plus réaliste des performances du modèle mais ceci augmente le temps de calcul, car SMOTE doit être appliqué à chaque itération de la validation croisée.

Entraînement des modèles sur le nouveau working_set

Modèle	AUC/ROC (Après SMOTE)	Sensibilité (Après SMOTE)	Spécificité (Après SMOTE)	Paramètres Finaux
Generalized Linear Model	0.85	0.75	0.78	-
Random Forest	0.83	0.83	0.62	mtry = 2
Random Forest (ranger)	0.83	0.82	0.64	mtry = 4, splitrule = gini, min.node.size = 15
Support Vector Machines with Radial Kernel	0.83	0.74	0.76	sigma = 0.18, C = 0.25
Support Vector Machines with Radial Kernel	0.83	0.74	0.76	sigma = 0.18, C = 0.25
CART	0.75	0.68	0.74	cp = 0.018
CART	0.79	0.76	0.68	cp = 0.001
CART	0.77	0.71	0.72	cp = 0.01
CART	0.77	0.73	0.70	cp = 0.01
CART	0.77	0.72	0.71	cp = 0.01
k-Nearest Neighbors	0.79	0.70	0.73	k = 11
k-Nearest Neighbors	0.78	0.73	0.71	k = 7
glmnet	0.85	0.75	0.79	alpha = 1, lambda = 0.0046
C5.0	0.84	0.84	0.64	trials = 50, model = tree, winnow = TRUE
C5.0	0.84	0.84	0.62	trials = 50, model = tree, winnow = FALSE
C5.0	0.84	0.84	0.62	trials = 50, model = tree, winnow = FALSE
C5.0	0.83	0.85	0.61	trials = 50, model = tree, winnow = TRUE
CART	0.77	0.72	0.71	cp = 0.01
CART	0.77	0.72	0.71	cp = 0.01

Voici les résultats après l'application de la technique Smote. On peut voir une augmentation générale dans l'AUC. À la page 27, je mets les anciennes et les nouvelles valeurs pour pouvoir les comparer.

Courbe ROC pour tous les modèles



Voici les résultats après entraînement des modèles sur la nouvelle working_set. **Le déséquilibre** dans la première partie entre les classes a **biaisé** les modèles vers la prédiction de la classe majoritaire, ce qui a affecté négativement la sensibilité et l'AUC. C'est pourquoi j'obtient une meilleure AUC sur ces données après l'application de l'algorithme de **Smote**. Le modèle 1 est le modèle GLM.

Modèle	AUC/ROC (Avant SMOTE)	Sensibilité (Avant SMOTE)	AUC/ROC (Après SMOTE)	Sensibilité (Après SMOTE)	Paramètres Finaux
Generalized Linear Model	0.84	0.91	0.85	0.75	-
Random Forest	0.82	0.93	0.83	0.83	mtry = 2
Random Forest (ranger)	0.83	0.90	0.83	0.82	mtry = 4, splitrule = gini, min.node.size = 5
Support Vector Machines with Radial Kernel	0.82	0.93	0.83	0.74	sigma = 0.18, C = 128
Support Vector Machines with Linear Kernel	0.81	0.93	0.83	0.74	C = 1
CART	0.72	0.92	0.75	0.68	cp = 0.018
CART	0.79	0.92	0.79	0.76	cp = 0.001
k-Nearest Neighbors	0.78	0.89	0.79	0.70	k = 7
glmnet	0.84	0.91	0.85	0.75	alpha = 1, lambda = 0.0046
C5.0	0.83	0.90	0.84	0.84	trials = 50, model = tree, winnow = FALSE
C5.0	0.83	0.89	0.84	0.84	trials = 50, model = tree, winnow = FALSE
C5.0	0.83	0.90	0.84	0.84	trials = 50, model = tree, winnow = FALSE
C5.0	0.83	0.90	0.83	0.85	trials = 50, model = tree, winnow = FALSE
CART	0.72	0.92	0.77	0.72	cp = 0.01

Dans ce tableau, je mets les différentes métriques avant et après l'application de l'algorithme smote pour les comparer. On remarque que l'**AUC est plus élevé** qu'avant **Smote** et la sensibilité **est plus élevée** pour quelque modèles.

Modèle	Méthode	Caractéristiques	Seuil Optimal	Perte Attendue	AUC/ROC
Model 1	Logistic Regression	Régression logistique standard	0.45	0.30	0.85
Model 2	Random Forest	Random Forest standard	0.31	0.22	0.83
Model 2.2	Random Forest (Tuned)	Random Forest avec tuning (mtry, splitrule, min.node.size)	0.34	0.23	0.83
Model 3	SVM (Radial Kernel)	SVM avec noyau radial, centré et scalé	0.39	0.31	0.83
Model 3.2	SVM (Radial Kernel, Tuned)	SVM avec noyau radial, centré et scalé, tuning de C	0.39	0.29	0.83
Model 3.3	SVM (Linear Kernel)	SVM avec noyau linéaire, centré et scalé	0.36	0.41	0.83
Model 4	Decision Tree	Arbre de décision standard	0.45	0.31	0.75
Model 4.1	Decision Tree (Gini, mb=10)	Arbre de décision avec split Gini, minbucket = 10	0.45	0.29	0.79
Model 4.2	Decision Tree (Gini, mb=9)	Arbre de décision avec split Gini, minbucket = 9	0.44	0.37	0.77
Model 4.3	Decision Tree (Info, mb=5)	Arbre de décision avec split Information, minbucket = 5	0.44	0.37	0.77
Model 4.4	Decision Tree (Info, mb=9)	Arbre de décision avec split Information, minbucket = 9	0.45	0.37	0.77
Model 5	K-Nearest Neighbors	K-Nearest Neighbors standard	0.41	0.28	0.79
Model 5.1	K-Nearest Neighbors (Tuned)	K-Nearest Neighbors avec tuning et scaling	0.36	0.28	0.78
Model 6	LASSO	Régression LASSO avec tuning de lambda	0.48	0.33	0.85
Model 7	C5.0	C5.0 avec minCases = 9, noGlobalPruning = FALSE	0.35	0.23	0.84
Model 7.1	C5.0	C5.0 avec minCases = 9, noGlobalPruning = TRUE	0.34	0.23	0.84
Model 7.2	C5.0	C5.0 avec minCases = 4, noGlobalPruning = FALSE	0.35	0.23	0.84
Model 7.3	C5.0	C5.0 avec minCases = 4, noGlobalPruning = TRUE	0.34	0.23	0.83
Model 8	Decision Tree (Gini, mb=9)	Arbre de décision avec split Gini, minbucket = 9	0.44	0.37	0.77
Model 8.1	Decision Tree (Gini, mb=4)	Arbre de décision avec split Gini, minbucket = 4	0.45	0.37	0.77

J'ai le choix entre GLM et LASSO mais je choisi le modèle le plus **interprétable**, avec **l'AUC le plus élevé et des pertes les plus basses**. D'après le dernier tableau, je choisis le modèle GLM avec le seuil optimal de 0.45, perte attendue de 0.30 et AUC/ROC de 0.85.

Entrainement sur le working_set et prédiction sur le holdout

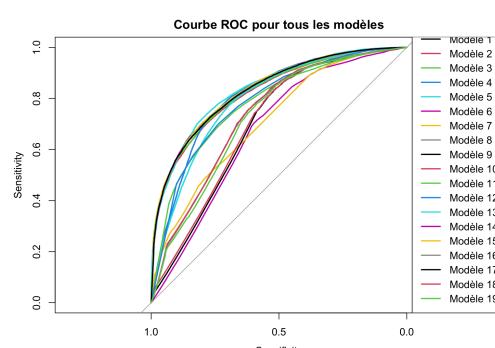
Matrice de confusion :

	Prédit Non	Prédi Oui
Réel Non	772	169
Réel Oui	94	164

Les Métrique du modèle :

Métrique	Valeur
Sensibilité (True Positive Rate)	0.64
Spécificité (True Negative Rate)	0.82
AUC	0.82
Perte Attendue	25
Seuil Optimal	0.45
Méthode	GLM (Generalized Linear Model)

En outre, il serait pertinent d'examiner les résultats obtenus en appliquant uniquement la méthode des k plus proches voisins (k-NN) sans recourir à la technique SMOTE. Cette approche permettrait de comparer les performances intrinsèques de l'algorithme k-NN en l'absence de toute manipulation préalable des données, offrant ainsi une perspective complémentaire sur l'efficacité de cette méthode dans notre contexte spécifique.



Modèle	optimal.threshold	expected.loss	ROC
Generalized Linear Model	0.26	0.34	0.85
Random Forest	0.24	0.38	0.83
Random Forest (Tuned)	0.26	0.38	0.83
Support Vector Machines with Radial Kernel	0.19	0.38	0.82
Support Vector Machines with Radial Kernel (Tuned)	0.22	0.40	0.81
CART	0.24	0.55	0.69
CART (Gini split, minbucket = 10)	0.21	0.54	0.80
CART (Gini split, minbucket = 9)	0.22	0.49	0.72
CART (Information split, minbucket = 5)	0.24	0.52	0.70
CART (Information split, minbucket = 9)	0.25	0.51	0.70
k-Nearest Neighbors	0.23	0.44	0.80
k-Nearest Neighbors (Tuned)	0.24	0.43	0.79
glmnet	0.27	0.36	0.85
C5.0 (minCases = 9, noGlobalPruning = FALSE)	0.29	0.37	0.84
C5.0 (minCases = 9, noGlobalPruning = TRUE)	0.30	0.36	0.84
C5.0 (minCases = 4, noGlobalPruning = FALSE)	0.29	0.37	0.84
C5.0 (minCases = 4, noGlobalPruning = TRUE)	0.29	0.37	0.84
CART (Gini split, mincut = 9)	0.23	0.49	0.73
CART (Gini split, mincut = 4)	0.23	0.49	0.72

Résultats après entraînement sur le working_set et prédiction sur le hold_out :

Modèle	AUC	Sensibilité	Spécificité	Perte Attendue	Seuil
Generalized Linear Model	0.82	0.53	0.90	0.39	0.25
Generalized Linear Model	0.82	0.66	0.81	0.53	0.5

J'utilise ce modèle pour prédire le risque de défaut dans le fichier data_new.csv après l'entraînement de ce modèle sur les données du fichier data.csv (working_set et hold out set).

Il y a 93 valeurs manquantes dans le tableau data_new. J'enlève les colonnes client et catégorie.

	age	education	emploi	adresse	revenus	debcred	debcarte	autres
X	Min. :18.00	Min. :1.000	Min. : 0.00	Min. : 0.000	Min. : 13.00	Min. : 0.000	Min. : 0.000	Min. : 0.000
X.1	1st Qu.:28.00	1st Qu.:1.000	1st Qu.: 2.00	1st Qu.: 2.000	1st Qu.: 25.48	1st Qu.: 5.645	1st Qu.: 0.454	1st Qu.: 1.144
X.2	Median :34.00	Median :3.000	Median : 6.00	Median : 6.000	Median : 35.00	Median : 9.200	Median : 1.040	Median : 2.283
X.3	Mean :34.77	Mean :3.212	Mean : 8.22	Mean : 7.667	Mean : 48.51	Mean :10.696	Mean : 1.720	Mean : 3.403
X.4	3rd Qu.:40.00	3rd Qu.:5.000	3rd Qu.:12.00	3rd Qu.:11.000	3rd Qu.: 57.62	3rd Qu.:15.005	3rd Qu.: 2.217	3rd Qu.: 4.272
X.5	Max. :77.00	Max. :5.000	Max. :44.00	Max. :36.000	Max. :324.60	Max. :38.170	Max. :29.255	Max. :37.731
X.6	NA's :47	NA	NA	NA's :46	NA	NA	NA	NA

Nous utilisons l'**imputation par k-plus proches voisins** (kNN) pour traiter les données manquantes en appliquant la **méthode kNN** de la librairie **VIM**.

	age	education	emploi	adresse	revenus	debcred	debcarte	autres	age_imp	adresse_imp
X	Min. :18.00	Min. :1.000	Min. : 0.00	Min. : 0.000	Min. : 13.00	Min. : 0.000	Min. : 0.000	Min. : 0.000	Mode :logical	Mode :logical
X.1	1st Qu.:29.00	1st Qu.:1.000	1st Qu.: 2.00	1st Qu.: 3.000	1st Qu.: 25.48	1st Qu.: 5.645	1st Qu.: 0.454	1st Qu.: 1.144	FALSE:453	FALSE:454
X.2	Median :34.00	Median :3.000	Median : 6.00	Median : 6.000	Median : 35.00	Median : 9.200	Median : 1.040	Median : 2.283	TRUE:47	TRUE:46
X.3	Mean :34.67	Mean :3.212	Mean : 8.22	Mean : 7.476	Mean : 48.51	Mean :10.696	Mean : 1.720	Mean : 3.403	NA	NA
X.4	3rd Qu.:40.00	3rd Qu.:5.000	3rd Qu.:12.00	3rd Qu.:10.000	3rd Qu.: 57.62	3rd Qu.:15.005	3rd Qu.: 2.217	3rd Qu.: 4.272	NA	NA
X.5	Max. :77.00	Max. :5.000	Max. :44.00	Max. :36.000	Max. :324.60	Max. :38.170	Max. :29.255	Max. :37.731	NA	NA

J'utilise le modèle GLM pour prédire les risques de défaut de chaque client. Je prend le seuil optimal qui est de 0.4587581 pour le GLM et je transforme tous les probabilités en dessous de ce seuil en Non et les autres en Oui.

Les résultats avec les probas et les identifiants des clients sont dans le nouveau fichier HUSSEINI_MohamadAli.csv.

CONCLUSION

Le modèle GLM a montré de bonnes performances avec une sensibilité de 0.67, une spécificité de 0.81, et une AUC de 0.83. La perte attendue sur l'ensemble de validation est de 0.26. Ces résultats indiquent que le modèle est efficace pour détecter les cas positifs tout en maintenant un faible taux de faux positifs.

Ce projet a mis en lumière l'importance de l'équilibrage des classes pour améliorer la performance des modèles de classification. En utilisant des techniques comme SMOTE et en évaluant les modèles de manière rigoureuse, nous avons pu identifier les modèles les plus performants pour les prédictions.

Ce projet a été une expérience enrichissante et formatrice et je vous remercie chaleureusement.