UNIVERSITÉ
CÔTE D'AZUR

PROJET
ANALYSE DE DONNÉES

MASTER 1
INGÉNIERIE
MATHÉMATIQUE

PRÉDICTION DE L'APPÉTENCE DE CLIENTS

Rédigé par :

KOULONI Yannick
BEDJA MROINKODO SAID Awadi

Année Universitaire 2022 - 2023

Introduction

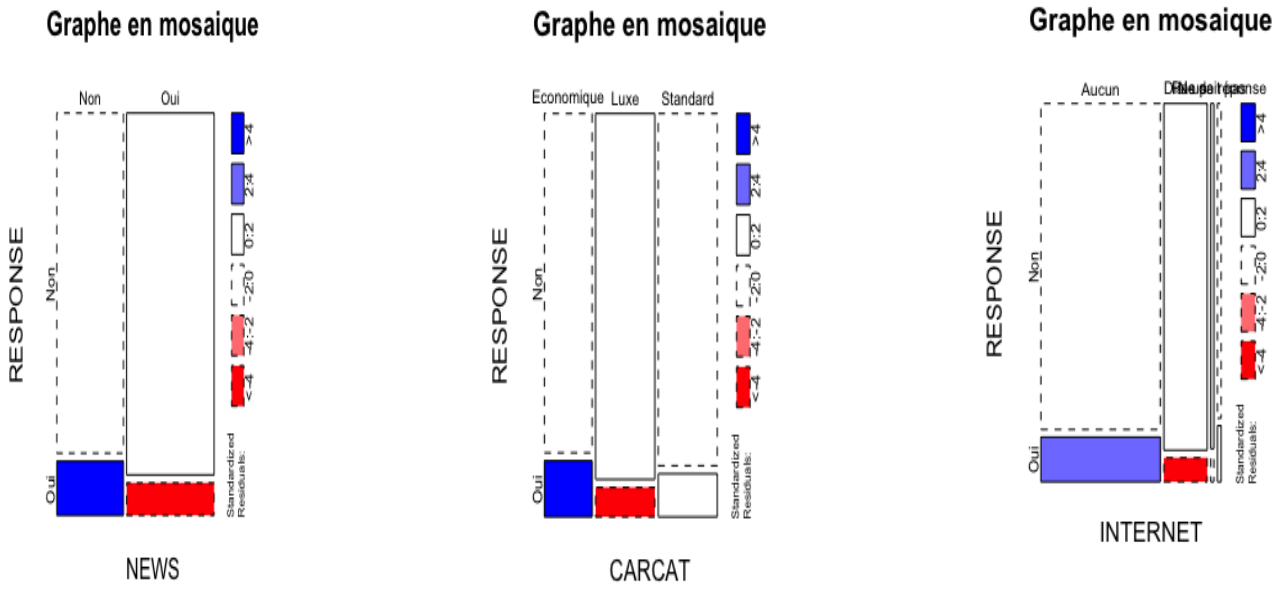
Faisant l'étude et l'analyse d'un ensemble de données contenant le choix des clients à donner une réponse positive ou négative à l'issue d'une offre promotionnelle commerciale, l'objectif de notre projet est de comprendre les critères principaux associés à une réponse positive ou négative de ces clients à l'offre et de construire un modèle de prédiction de la réponse de ces derniers pour l'appliquer à un ensemble de nouveaux clients . Définir des offres promotionnelles commerciales efficaces nécessite la compréhension des différents groupes de clients afin de mieux les cibler et de maximiser le taux de réponses positives pour limiter les coûts de diffusions. De fait, pour mener à bien notre projet, on applique premièrement les techniques de clustering afin de générer des clusters qui différencient les deux classes de clients et ensuite, les techniques de classification supervisée afin de concevoir un modèle de prévision de la classe des clients.

I - Exploration des données, interprétations des résultats et Pré-traitements appliqués aux données

1) Test de dépendance entre variables discrètes

Pour évaluer la dépendance entre nos variables discrètes dans notre ensemble d'étude, on considère les fonctions chisq.test() et fisher.test() de la librairie stats de R Base permettant de calculer respectivement la corrélation de Pearson(test de chi-2) et celle de Fisher entre deux variables discrètes.
Par exemple, on souhaite examiner la dépendance entre la variable NEWS, CARCAT et INTERNET par rapport à RESPONSE.
Le test de chi 2, nous donne pour un seuil de 5%, des p-values respectives 2.032e-12, 2.225e-11 et 1.403e-09. Celui de Fisher, on a 2.595e-12, 1.818e-11 et 2.043e-10.
On remarque, ainsi, qu'avec ces tests que ce soit Pearson ou Fisher, on a des p-values très faibles. Donc avec une seule variable, on ne peut pas être certain que le client donne une réponse positive par rapport à l'offre proposée.

On peut s'intéresser au test de dépendance entre valeurs de variables discrètes. Ainsi, on introduit la fonction mosaicplot(variable1 ~ variable2, data = data_frame) de R Base qui permet d'afficher un graphique en mosaïque d'interprétation des valeurs des résidus du test chi 2. Les dépendances positives sont affichées en bleu et en rouge pour les dépendances négatives. Pour identifier la force du lien de dépendance, on observe le niveau d'intensité de la couleur(foncé ou clair). L'absence de couleur indique la non-dépendance significative.



Illustrations pour les exemples des tests de chi2 réalisés précédemment

2) Test de dépendance entre variables discrètes

On considère la fonction cor.test() de la librairie stats de R Base qui permet de calculer les corrélations de Pearson, Spearman et Kendall qui sont les principales mesures de corrélation le plus utilisées. Ainsi, l'objectif est de déterminer si les valeurs de deux variables sont significativement éloignées de l'indépendance. Lorsqu'on souhaite mesurer la corrélation respectivement entre AGE, INCOME, ADDRESS et EMPLOY avec CAR, on a des p-values <

2.2e-16 et des tau respectivement 0.2980293, 0.975565, 0.1859606 et 0.4568693.

3) Mesures d'évaluation d'utilité prédictive de variables hétérogènes

On souhaite s'intéresser à la quantité d'information que chaque variable prédictive est en mesure de quantifier sa contribution à la prédiction de l'appétence des clients. Nous allons utiliser, ici, la librairie CORElearn qui fournit les fonctions encore() et attrEval() pour calculer ces mesures d'évaluations sur les attributs plus précisément le coefficient de Gini, l'algorithme du Relief et le principe de la longueur de description minimale qui sont les principales mesures basées sur l'entropie de Shanon. On s'intéresse plus particulièrement aux mesures d'Informations Gain qui nous a permises d'identifier les dix variables hétérogènes les plus utiles pour la prédiction qui sont telles que INCCAT, CAR,CARCAT, INCOME, INTERNET, NEWS, ED, OWNPC, CALLWAIT, MULTLINE.

II - Clustering des données

- Méthode de clustering utilisée : **Clustering par partitionnement**
- Approche algorithmique : Algorithme des K-means
  En partitionnant notre espace de données en 4 clusters (K=4), on constate qu'il y a un déséquilibre entre les classes RESPONSE=Oui et RESPONSE=Non; la classe positive Oui est inégalement répartie.
  En variant la valeur de K jusqu'à 10 par le biais d'une boucle for dans la fonction élaborée, on constate que ce déséquilibre persiste toujours.
  Nous avons pu résoudre ce problème par une méthode de rééquilibrage des classes d'une manière aléatoire .

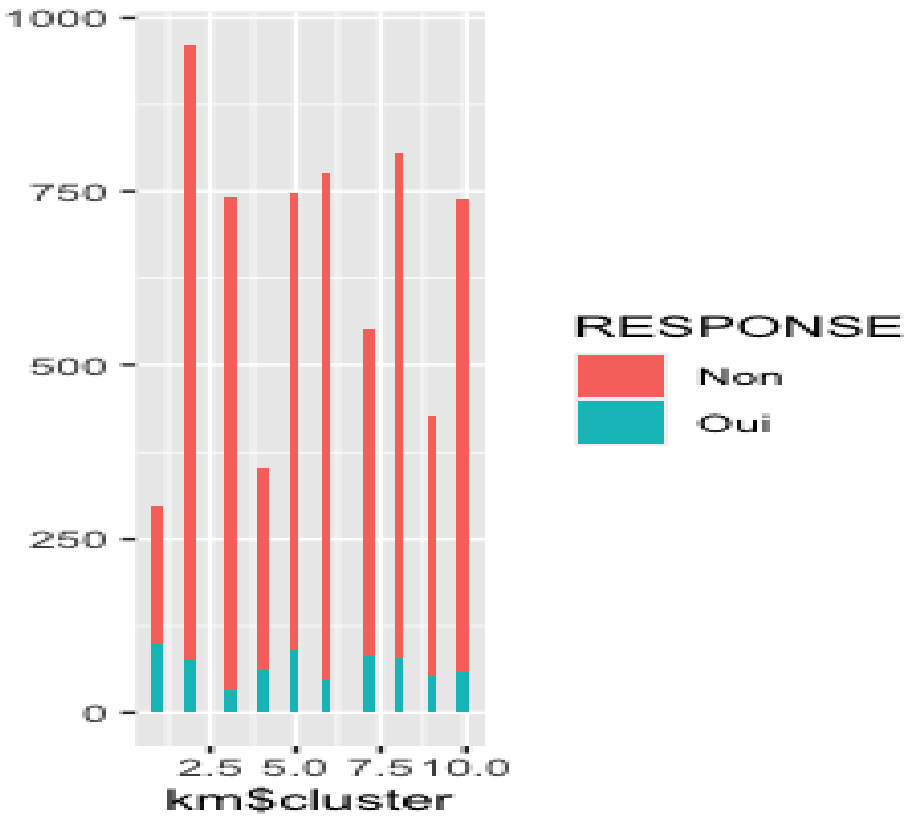


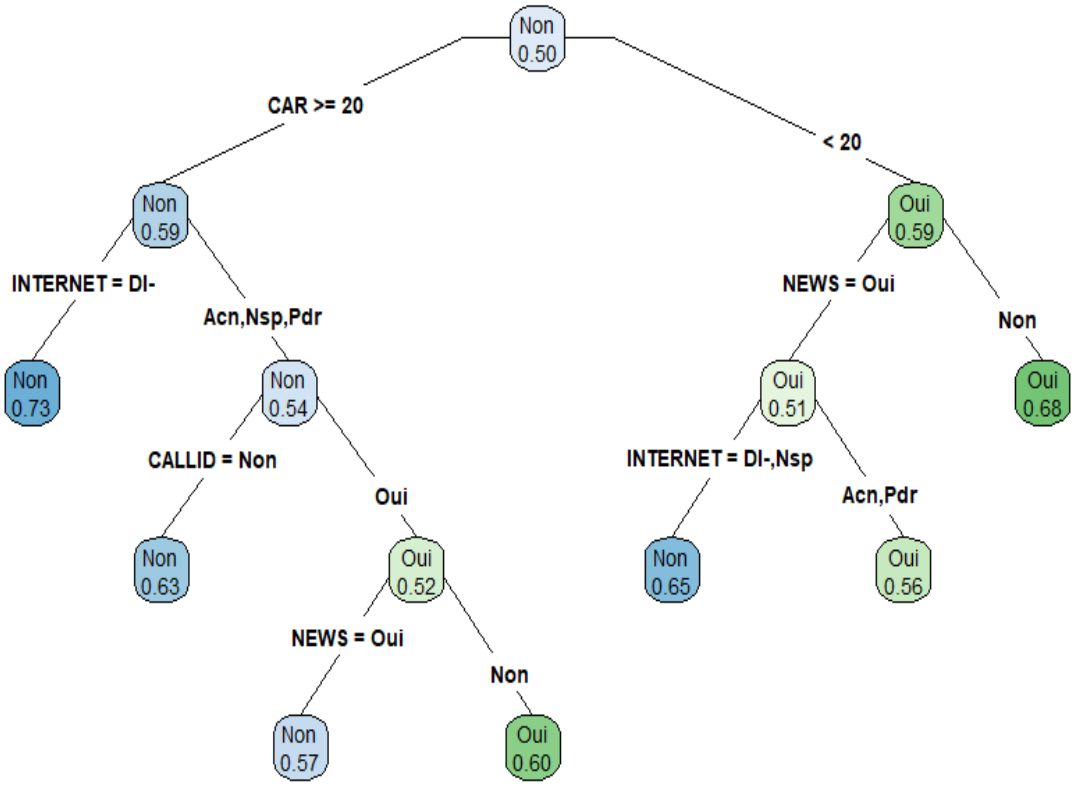
Illustration cluster K-means avec K = 10

III - Classification supervisée

D'après les histogrammes affichés précédemment et la méthode de clustering appliquée ci-dessus , on a observé un problème de déséquilibre des classes. On constate que la classe de "Non" est très fréquente contrairement à celle de "Oui". De ce fait, on souhaite trouver une méthode qui consiste à rééquilibrer les classes de manière que la classe minoritaire soit suffisamment représentée dans l'ensemble d'apprentissage afin que les algorithmes d'apprentissage de classifieurs puissent la reconnaître et la prédire en la distinguant de la classe majoritaire.
On souhaite prendre pour l'ensemble d'apprentissage 3⁄4 de l'ensemble de nos données. Donc, initialement on a 6100 observations réparties en 652 "oui" et 5448 "non". Pour résoudre à ce déséquilibre, on va jouer avec la longueur de chaque classe en utilisant la fonction sample() de R Base qui permet de prendre l'échantillon de la taille spécifiée(taille de "Oui" ou taille de "Non") à partir des éléments(échantillon de "Non" ou échantillon de "Oui") en utilisant soit

avec ou sans remplacement. Avec la fonction sample(), on obtient 1304 observations dans l'ensemble d'apprentissage.

1) Le classifieur Rpart()

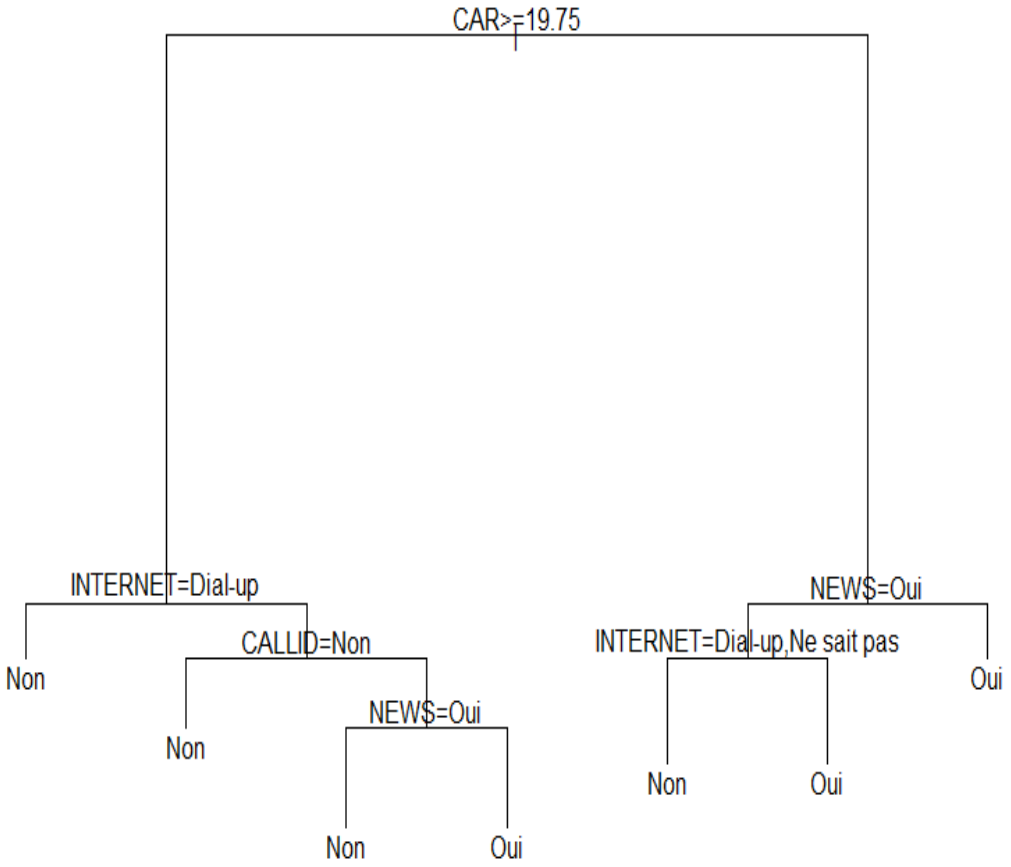


Graphe de l'arbre Rpart()

L'application de la prédiction, avec le classifieur Rpart(), nous donne un taux de succès égale de 53,67 %. On construit la matrice de confusion qui permet de quantifier numériquement l'importance des différents types de succès et erreurs et détaille le nombre de prédictions correctes et incorrectes pour chaque classe prédite et chaque réelle. Ainsi, on observe qu'en utilisant les mesures d'évaluations des succès et échecs, on a :

- Mesure du rappel (sensibilité) = 59,25 %
- Mesure de spécificité = 53,11%
- Mesure de précision = 11,11%
- Mesure du taux de vrais négatifs = 0,9294 (proportion)

2) Le classifieur C5.0 ()

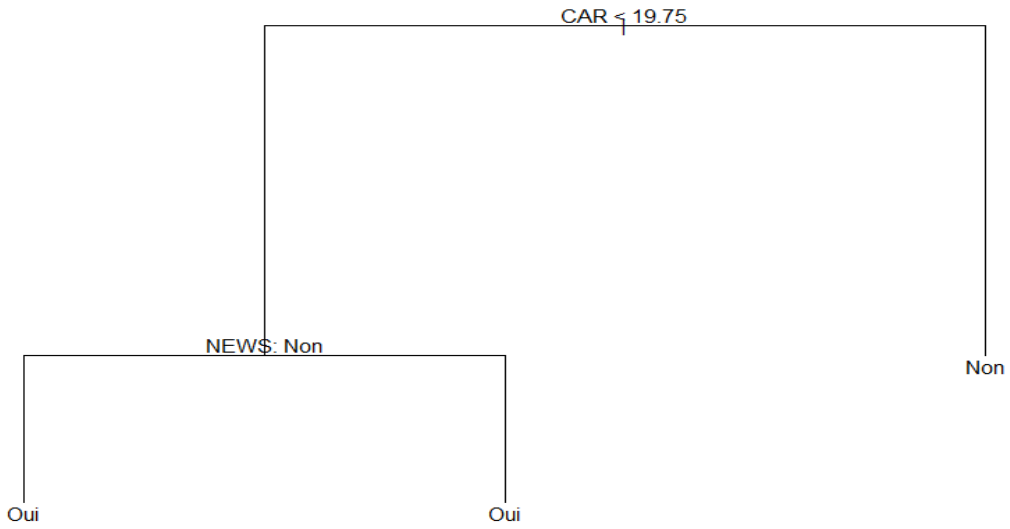


Graphe de l'arbre de décision C5.0 ()

De même, avec le classifieur C5.0(), on a:

- Taux de succès égale à 58,33%
- Mesure du rappel(sensibilité) = 48,14%
- Mesure de spécificité = 59,34 %
- Mesure de précision = 9,02%
- Mesure du taux de vrais négatifs = 1,0384 (proportion)

3) Classifieur tree()

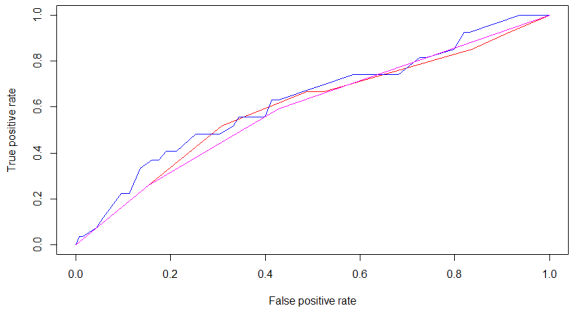


Graphe de l'arbre de décision tree ()

Pour le classifieur tree(), on a :

- Taux de succès égale à 28,33 %
- Mesure du rappel(sensibilité) = 74,07 %
- Mesure de spécificité = 23,80%
- Mesure de précision = 13,86%
- Mesure du taux de vrais négatifs = 0,41667(proportion)

4) Courbes ROC



- Rpart()
- C5.0()
- Tree()

En calculant les indicateurs AUC(Area Under the Curve), on trouve :

- pour l'arbre Rpart(), l'indice AUC vaut *0,54*
- pour l'arbre C5.0(), l'indice AUC vaut *0,62*
- pour l'arbre Tree(), l'indice AUC vaut *0,53*

Conclusion

Cette étude de cas servant d'application de différents classifieurs d'arbres de décisions nous a permis de savoir que les différentes méthodes d'évaluation comme taux de succès / taux d'échecs, précision à l'aide de matrices de confusion et le calcul de l'indice AUC peuvent être presque égaux pour montrer la pertinence ou non des classifieurs utilisés. Comme difficultés rencontrées, on n'a pas pu utiliser la méthode de rééquilibrage sous-échantillonnage mentionnée dans les ressources complémentaires du cours du fait de l'incompatibilité des fonctions de rééquilibrage sous-échantillonnage aléatoire des classes comme neater(), RandUnderclassif () et ubUnder avec notre version de logiciel R . De plus, nous avons été contraints de supprimer les accents pour les variables MARITAL, EMPCAT, JOBSAT, INCCAT et INTERNET et nous avons pu constater aussi qu'étant donné que la rééquilibrage des classes s'est faite de façon aléatoire, les résultats varient après chaque nouvelle compilation ; les résultats inversés dépendent donc de l'implémentation . Toutefois, on peut noter que le classifieur d'arbre de décision C5.0 () ayant l'indice AUC le plus élevé est le plus performant à conserver afin de pouvoir minimiser avant tout le risque de ne pas prévoir l'appétence d'un client, c'est-à-dire que l'on souhaite avant tout éviter de prédire à tort un client comme non susceptible de répondre positivement à l'offre.