Projet Final

By Loop Eliott,de Crombrugghe Grégoire

1-Introduction

Description du Problème

Une Banque, nous demande de réaliser un modèle qui permettra de déterminer quel produit d'investissement à proposer à un client, suivant son profil. La Banque nous met à disposition une série de données, qui doit être analysée afin de déterminer les attributs dits "utiles" et de pouvoir construire ce modèle.

Instances

nombres d'instances: 4734
Attribut continu: SE1,BA1-BA7
Attribut discret: SE2,PE1-PE15, IA1-IA3
Variable cible: InvType

But de l'analyse

Le but de l'analyse est de déterminer quels sont les attributs essentiels à la construction du modèle et de déterminer quel algorithme permet de construire un modèle avec la plus grande précision. Pour pouvoir proposer un produit qui corresponde aux maximum aux données du client.

2-Analyse Exploratoire

IA3 (Attribut discret)

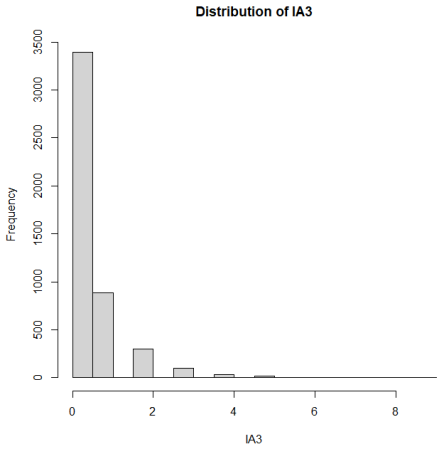
Pour cet attribut, les données que nous trouvons intéressantes sont les activités de 0 à 2 car pour ces investissements là nous possédons suffisamment de données que pour pouvoir en tirer des conclusions pertinentes.

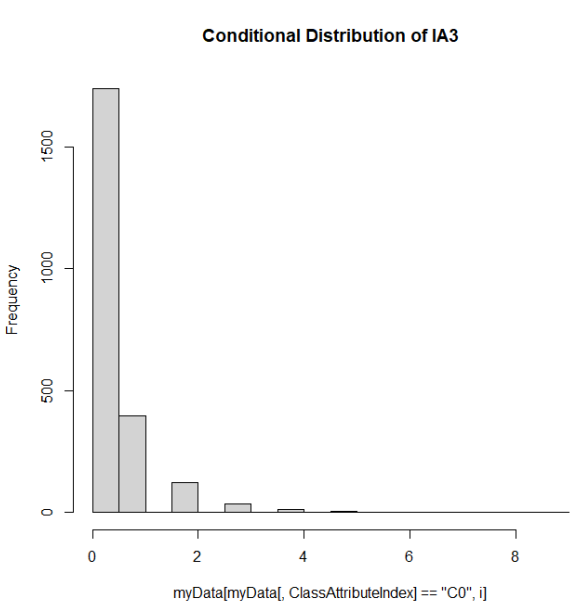
On remarque que parmi ceux qui ont investi dans l'activité 0 , il y a 51% , de ces personnes, qui ont choisi le produit « C0 », et 49% qui ont choisi le produit « C1 ».

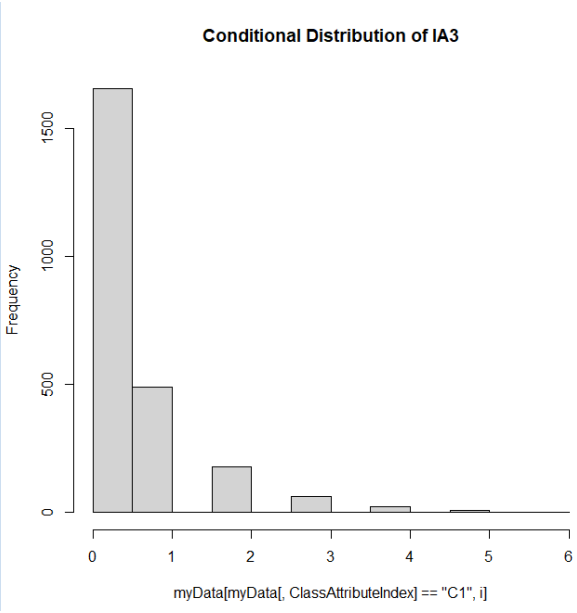
Pour l'activité 1, nous avons une proportion plus inégale, çad 45% ont choisi le produit C0, et 55% ont choisi le produit C1.

En observant l'activité 2, 60% (des investisseurs de l'activité 2) on choisit le produit C1 et 40% le produit C0.

Au vu des remarques ci-dessus, l'attribut IA3 ne nous permet pas de choisir avec certitude un produit dans lequel investir.

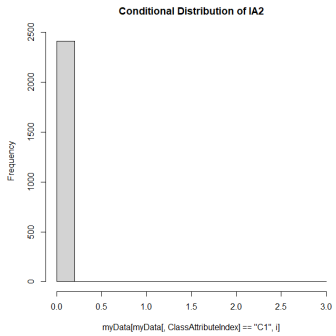






IA2 (Attribut discret)

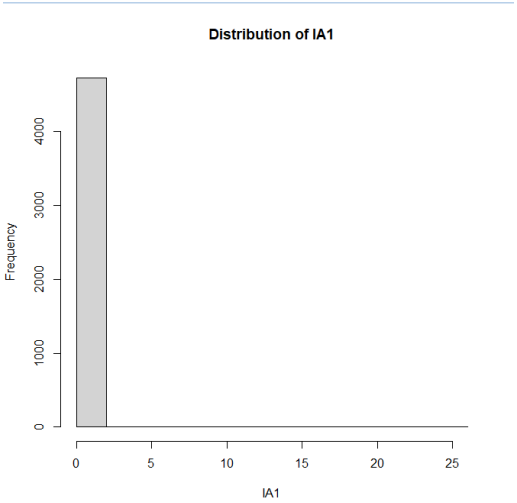
L'attribut IA2 étant constant, 99% des données ont la même valeur. Il n'est donc pas intéressant de prendre cet attribut en compte. Selon nous, cet attribut fait partie des attributs à ignorer.

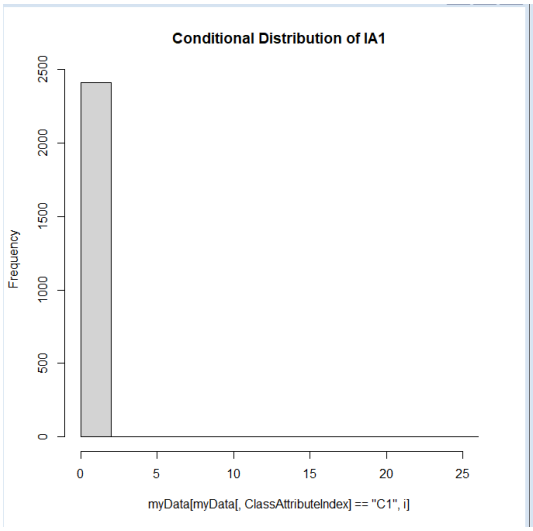


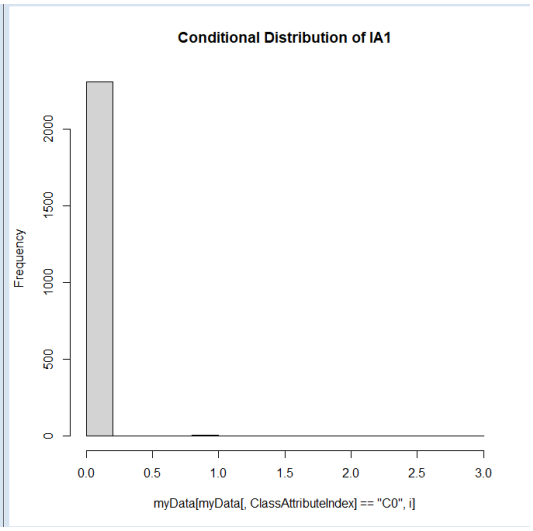
IA1 (Attribut discret)

données intéressantes : activité 0 car si nous faisons "table(myData$IA1)" dans la console R, nous observons qu'il y a 4714 personnes possède la valeur 0 pour cette attribut là.

De ces personnes qui ont pour l'attribut IA1 la valeur 0 : 51%, de personnes ont investi dans le produit C1 et 49% ont investi dans le produit C0.







<u>Tableau des attributs PE (Attribut discret)</u>

NPA-> **N**ombre de **P**ersonnes ayant **A**cheté le produit sur les 4734 personnes
A-C0-> probabilité qu'une personne, ayant acheté le produit, investisse dans le produit C0
A-C1-> probabilité qu'une personne, ayant acheté le produit, investisse dans le produit C1
NA-C0-> probabilité qu'une personne, n'ayant pas acheté le produit, investisse dans le produit C0
NA-C1-> probabilité qu'une personne, n'ayant pas acheté le produit, investisse dans le produit C1

Produit	NPA	A-C0	A-C1	NA-C0	NA-C1
PE15	4448	49%	51%	37%	63%
PE14	4349	49%	51%	51%	49%
PE13	3471	50%	50%	45%	55%

PE12	3790	51%	49%	42%	58%
PE11	3762	50%	50%	45%	55%
PE10	3314	52%	48%	43%	57%
PE9	4528	50%	50%	34%	66%
PE8	4373	50%	50%	40%	60%
PE7	4426	51%	49%	44%	56%
PE6	4134	48%	52%	55%	45%
PE5	3282	51%	49%	44%	56%
PE4	4705	49%	51%	79%	21%
PE3	4698	49%	51%	67%	33%
PE2	4709	49%	51%	40%	60%
PE1	4500	49%	51%	47%	53%

Commentaire :

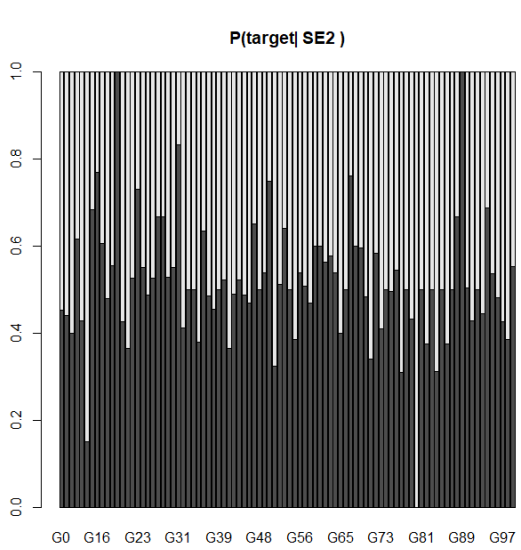
Pour chaque PE, sauf le 13,12,11,10 et le 5, il y a peu de personnes qui en possèdent. Nous trouvons que ces produits-là ne sont pas intéressants (non pertinents car trop de disparité entre ceux qui les ont achetés et ceux qui ne l'ont pas fait). Dans le cas où un client, pour lequel on doit conseiller un produit, à acheter un de ces produits, il nous est difficile de lui proposer quoi que ce soit seulement sur base de ce produit-là.

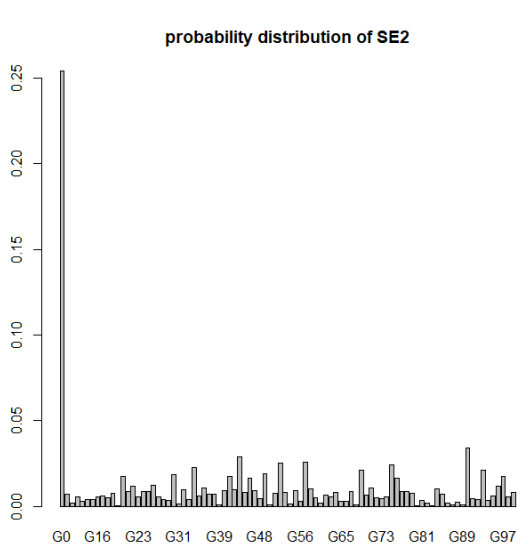
Ces attributs sont la plupart constant çad +- 50% des clients investissent dans le produit C1 et +-50% investissent dans le produit C0. Nous en déduisons que ce ne sont pas des caractéristiques "importantes" mais elles ne sont à négliger non plus.

SE2 (Attribut discret)

Selon nous la seule localisation qui pourrait être intéressante, à ce stade ci, est le G0. Car elle possède assez de données que pour pouvoir en tirer une conclusion pertinente (çad 1202 personne au total ce qui correspond à ¼ des données totales). En effet, 55% des personnes y résident ont investi dans le produit C1 et 45% ont investi dans le produit C0.

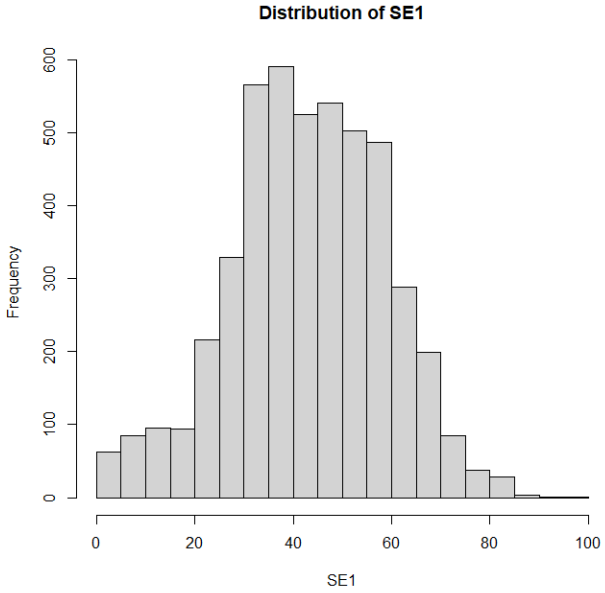
Il est compliqué d'utiliser les autres localisations, car comme dit ci-dessus, nous ne possédons pas assez de données.

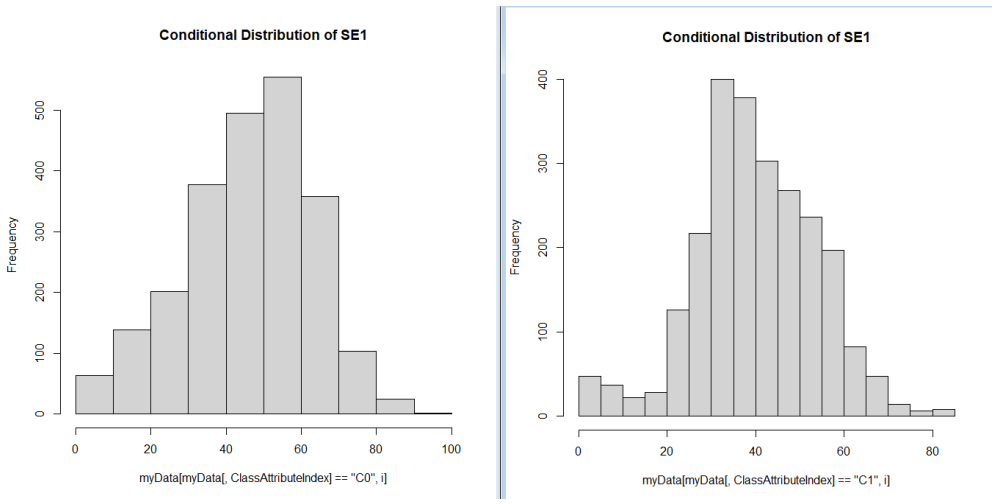




SE1(Attribut continu)

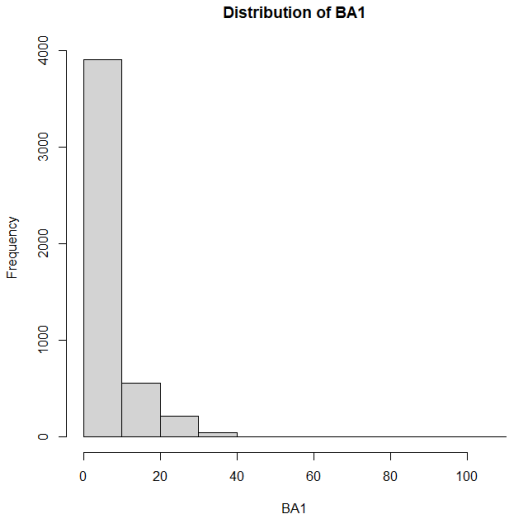
Sur base des graphes ; Nous en concluons que cet attribut est inutilisable, car nous observons que les graphes de Conditional Distribution selon CO et C1 sont très semblables. Qu'il n'y pas de, selon notre observation, une différence majeure entre les deux graphes (ils se chevauchent), qui permet d'argumenter la prise de décision du choix d'un produit.

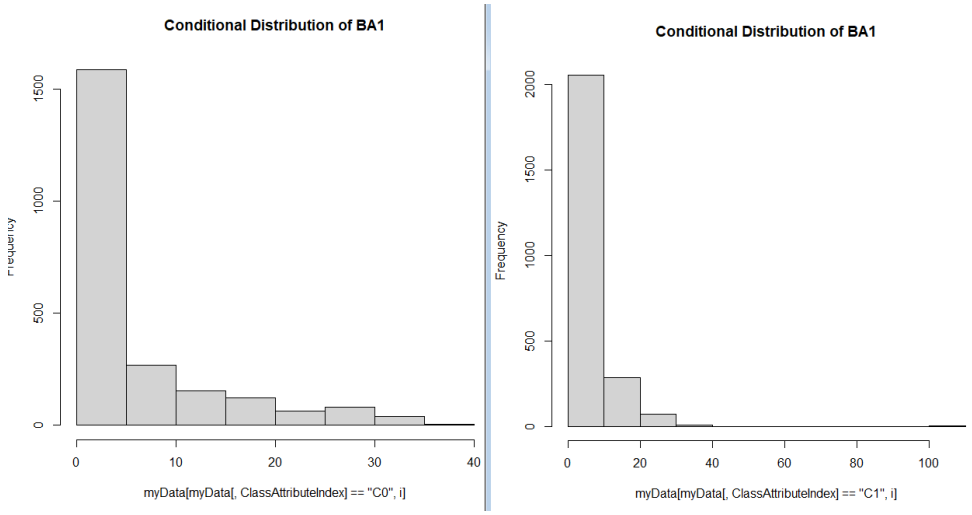




BA1(Attribut continu)

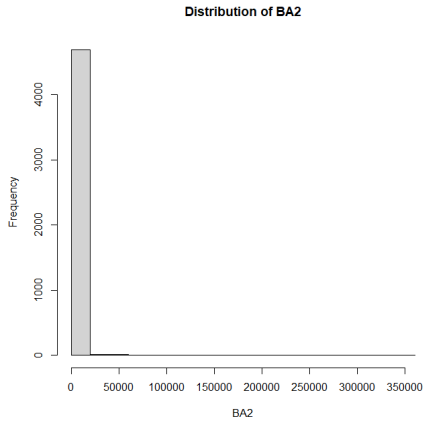
Sur base des graphes ; Nous en concluons que cet attribut est inutilisable, car nous observons que les graphes de Conditional Distribution selon CO et C1 sont très semblables. Qu'il n'y pas, selon notre observation, de différence majeure entre les deux graphes (il se chevauche), qui permet d'argumenter la prise de décision du choix d'un produit.

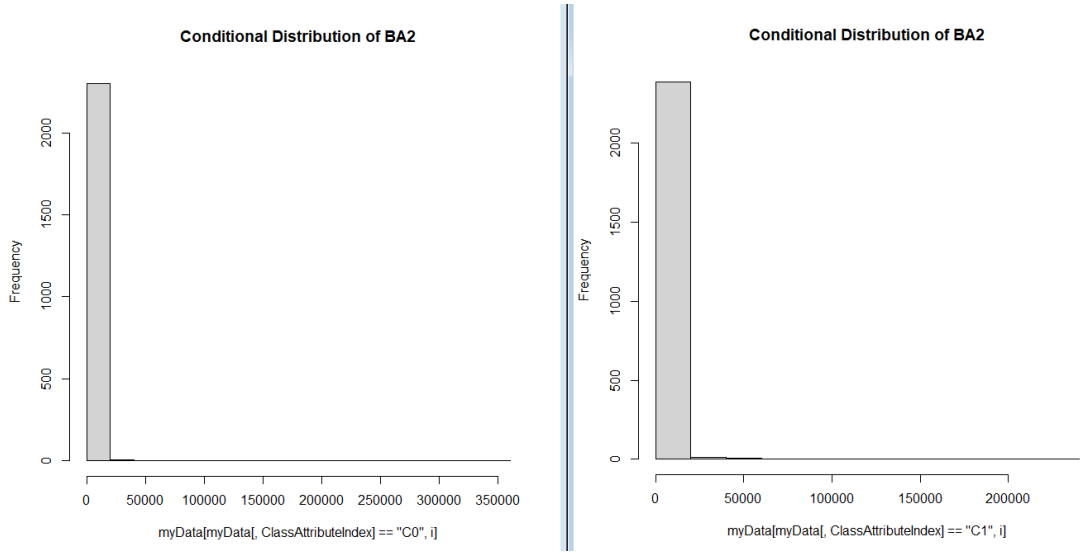




BA2 (Attribut continu)

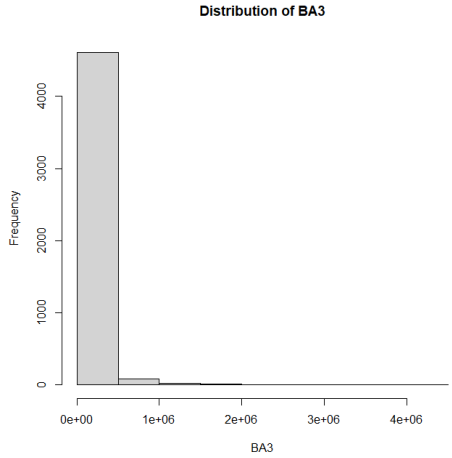
Sur base des graphes; Nous observons (visible) que les clients, ayant un montant supérieur à 50.000 (pour cette opération), ont tous investi dans le produit C1. Or, lorsque nous recherchons le montant maximum via une commande R, d'un client ayant investi dans le produit C0 nous observons que celui-ci est de 341698.2 et qu'il n'apparaît pas clairement sur le graphe. Nous en concluons que celui- ci est un cas isolé, (que l'intervalle dans lequel il se trouve à une fréquence égale à 1), et nous pensons que ce cas isolé peut-être ignoré.

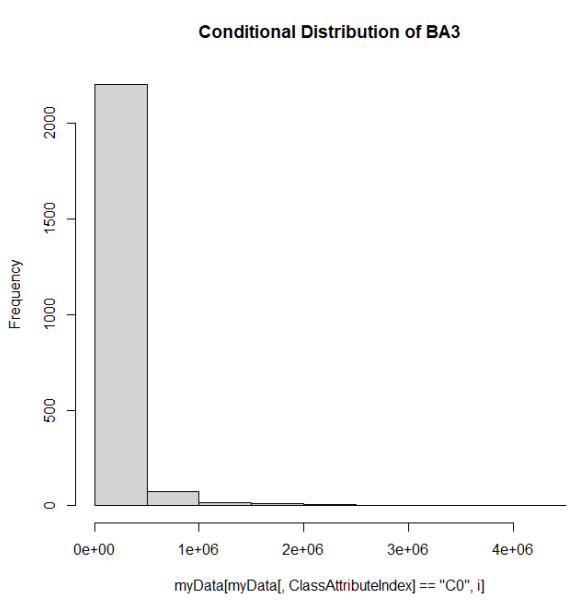


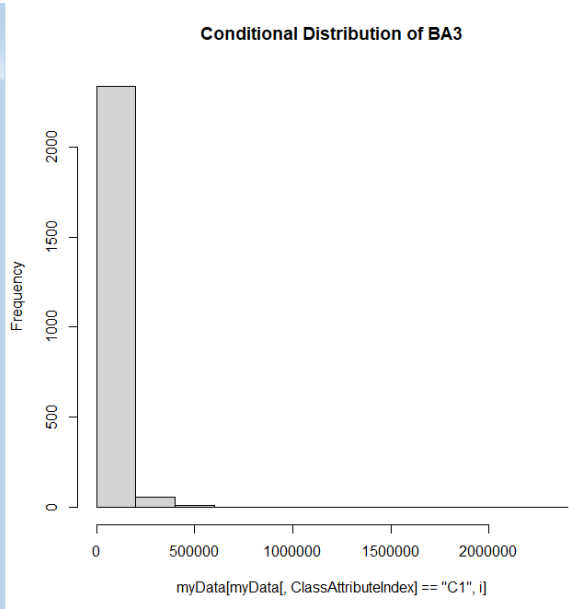


BA3 (Attribut continu)

Sur base des graphes ; Nous constatons que les clients, étant dans des tranches supérieurs à 1e+06, investissent généralement dans le produit C0. Toutefois, si nous cherchons le maximum du Conditional Distribution selon C0, nous remarquons qu'au moins un client ayant investi dans le produit C0 se trouve dans une tranche supérieure à 1e+06. Le montant maximum étant 2316357. Nous pensons que cette donnée, ce client (ayant investi dans C0) se trouvant dans "les tranches supérieures", est un cas isolé et peut-être ignorée.

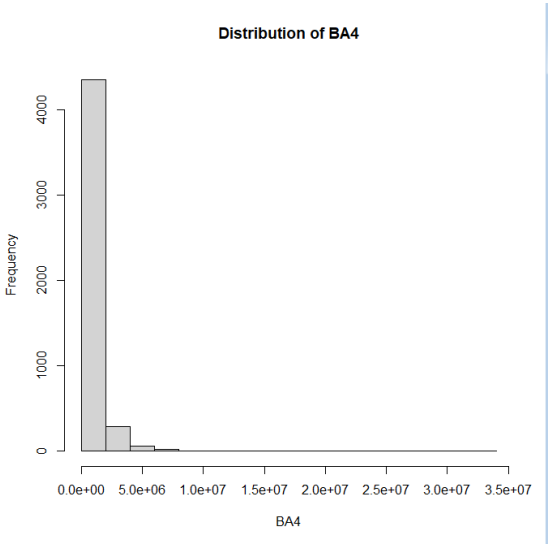


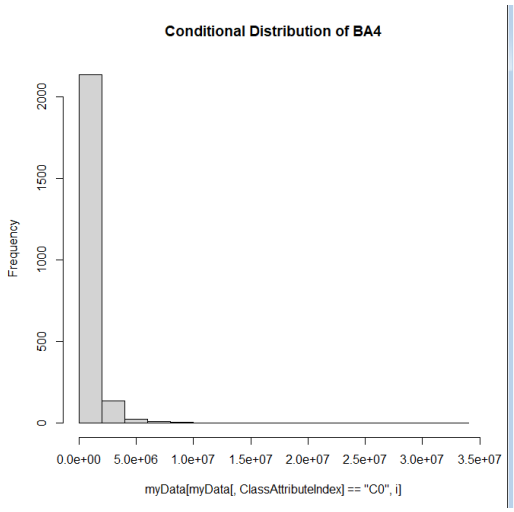


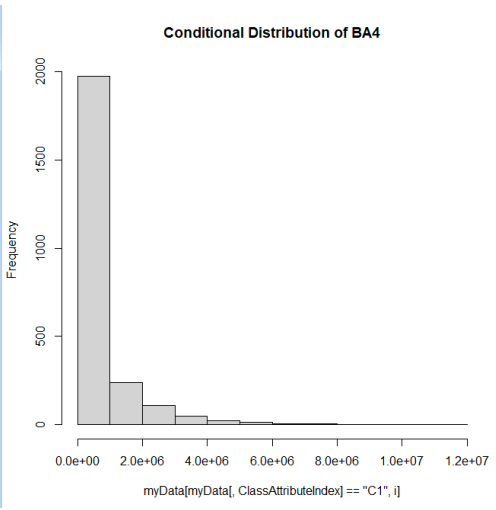


BA4 (Attribut continu)

Sur base des graphes ; Le montant des opérations, qui ont uniquement été effectuées par des clients qui ont investi dans le produit C0, est de 8.0e+06. Nous ne trouvons pas de conclusion sur base des autres tranches.

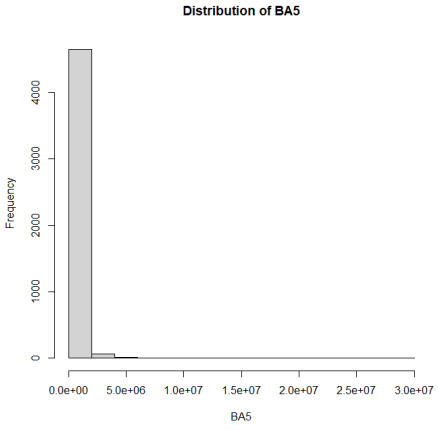


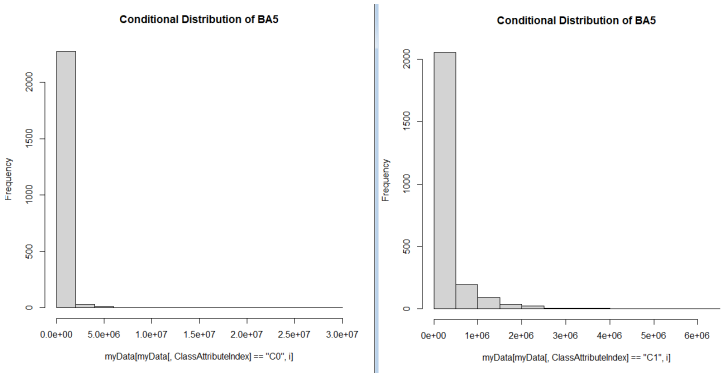




BA5 (Attribut continu)

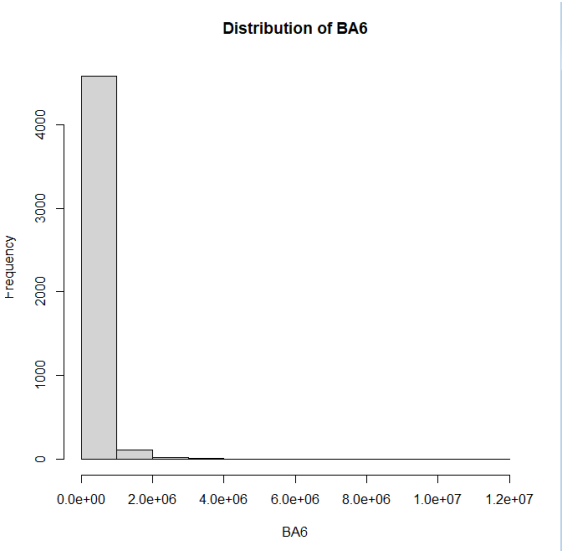
Sur base de graphes ; Toutes les opérations d'un montant supérieur à 4e+06, ont uniquement été effectuées par des clients qui ont investi dans le produit C0. Nous ne trouvons pas de conclusion sur base des autres tranches.

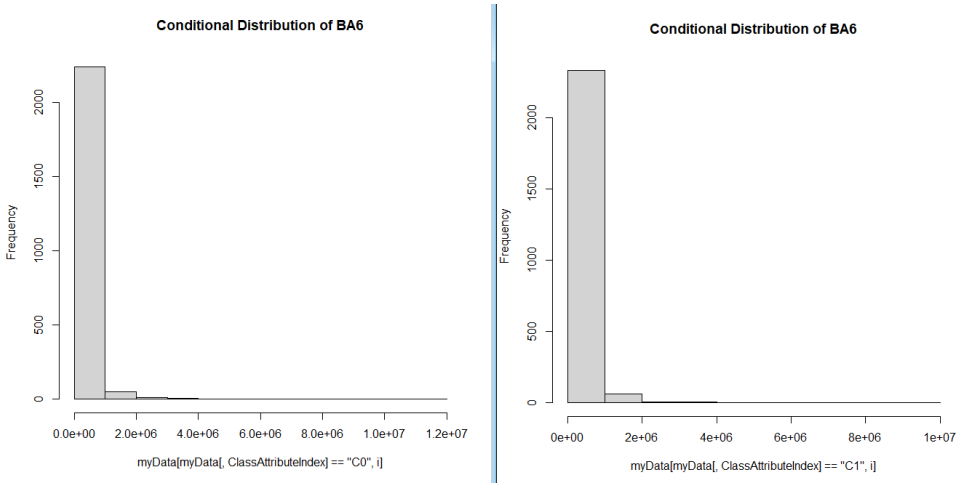




BA6 (Attribut continu)

Les graphes du Conditional Distribution selon C1 et C0 sont pratiquement identiques. Nous en concluons que cet attribut est, sur base d'une simple observation, inutile car elle ne nous permet pas de différencier les clients ayant investis dans le produit C0 de ceux qui ont investi dans le produit C1. L'attribut est par conséquent considéré comme impure.

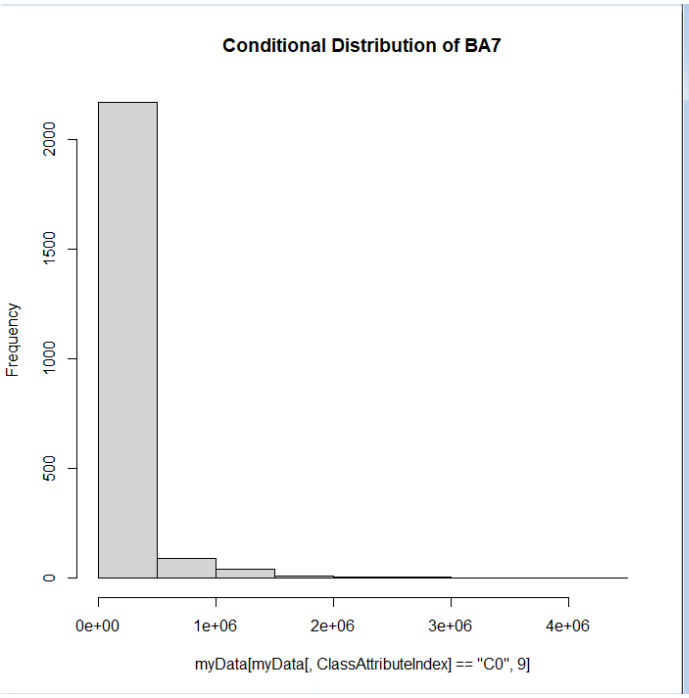


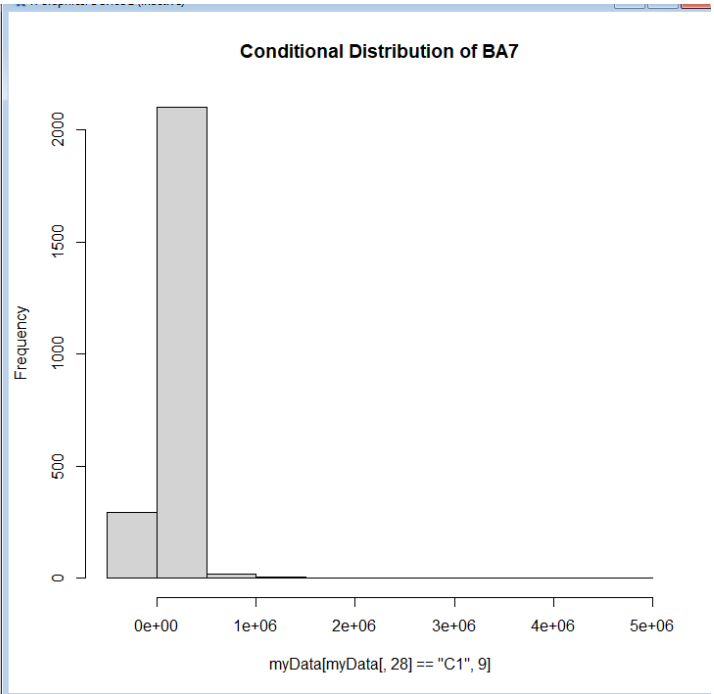


BA7 (Attribut continu)

Pour cet attribut, nous remarquons que les activité bancaire (BA7) dont le montant est

compris entre [-2 ;0[sont exclusivement fait par des clients qui ont investi dans le produit C1. La fréquence de cet intervalle étant égale à 2 celle-ci peut être ignorée, car pas assez représentative. De même que les montant supérieur à 2e+06, sont fait par des clients qui ont investis uniquement dans le produit C0. Les opérations faite avec les tranches de montant se trouvant entre les deux ne permettent pas de déterminer avec précision le produit à conseiller.





Sur base de l'analyse exploratoire nous concluons que seul l'attribut IA2 est vraiment inutile pour les raisons citées ci-dessus.
Nous ajoutons aussi qu'il est difficile de bannir certains attributs simplement sur base d'observation de graphes. Dans la suite de ce rapport nous allons calculer l'information gain de chaque attribut permettant de nous faire une idée plus précise des attributs inutiles

3-Importance des attributs

Cette partie du rapport correspond au deuxième TP effectué. Nous avons repris les mêmes données qui se trouvent dans le TP-information Gain.

Données Obtenus

Information gain=$\Delta H(y, x) = H(y) - H(y|x)$.
Information gain ratio = I(f,x)/H(f) (pour les attribut continu, f est la valeur pour laquelle l'information Gain est le plus grand. H(f) étant l'entropie normalisée).

Attribut	Information gain	Information gain ratio	Value (pour les attribut continu,la valeur pour laquelle l'information Gain est le plus grand)
BA3	0,144456733	0,14737084	18899
BA7	0,10164385	0,102630852	16371
BA5	0,055975917	0,059459887	16396
BA4	0,048956529	0,053012718	16396
SE1	0,03652936	0,041097321	53
SE2	0,021894631	0,026735939	/
BA1	0,009160626	0,038015086	24
IA3	0,005466622	0,013835623	/
PE10	0,004581686	0,00519914	/
PE12	0,003553709	0,004930615	/
PE15	0,002822064	0,008575454	/
PE9	0,002774747	0,011157604	/
PE5	0,002721656	0,003060283	/
PE8	0,00185796	0,004778081	/
PE4	0,001739443	0,032306844	/
PE6	0,001527011	0,002784296	222472

BA6	0,001479024	0,001810853	/
PE13	0,001186667	0,001418057	/
PE11	0,001138118	0,001553856	/
IA1	0,000806219	0,04549336	/
BA2	0,000750181	0,008465243	12568,2
PE3	0,000704663	0,010932269	/
PE7	0,000531	0,001529305	/
IA2	0,000205024	0,071097941	/
PE14	0,000189486	0,000465755	/
PE2	7,43E-05	0,002407872	/
PE1	3,5113041490508800000E-05	0,000123651	/

Au vu des données ci-dessus, nous constatons que ce sont BA3,BA7,BA5 qui ont l'information gain le plus élevé. Ceci nous indique que ce sont ces 3 attributs qui sont les plus pures et qui seront les attributs principaux permettant la prise de décision.

4-Evaluation des différents Algorithmes

Decision tree

Précision moyenne Information Gain

CP= "*complexity parameter. Any split that does not decrease the overall lack of fit by a factor of cp is not attempted*" (definition du cours)
minsplit="*the minimum number of observations that must exist in a node in order for a split to be attempted."* (definition du cours)

minsplit\CP	0,001	0,01	0,1	0,2
5	0,7330789	0,7368821	0,717237	0,717237
10	0,7299113	0,7368821	0,717237	0,717237
50	0,7287706	0,7368821	0,717237	0,717237
100	0,731052	0,7368821	0,717237	0,717237
200	0,7332066	0,7362484	0,717237	0,717237

Précision moyenne Gini Index

minsplit\CP	0,001	0,01	0,1	0,2
5	0,7390368	0,7362484	0,717237	0,717237
10	0,7375158	0,7362484	0,717237	0,717237
50	0,7356147	0,7362484	0,717237	0,717237
100	0,7414449	0,7362484	0,717237	0,717237
200	0,7382736	0,7357414	0,717237	0,717237

*ces tableaux ont été repris du TP sur les arbre de décision

Naive Bayes

Modèle à partire de Naive Bayes	Accuracy	Default_Classifier	Accuracy-Default_Classifier
modèle 1	0.5760456	0.5038023	0.07224335
modèle 2	0.5931559	0.5104563	0.08269962
modèle 3	0.5963245	0.5101394	0.08618504
modèle 4	0.5893536	0.5072877	0.08206591
modèle 5	0.5766793	0.5139417	0.06273764
	Average Accuracy:	0.5863118	

*ce tableaux à été repris du TP sur Naive-Bayes

Voisins Proches

K	Précision moyenne	Précision moyenne	Comparaison

	knn	default classifier	
1	0.626616	0.512801	0,113815
3	0.6494297	0.512801	0,1366287
5	0.6640051	0.512801	0,1512041
10	0.6724968	0.512801	0,1596958
20	0.6802281	0.512801	0,1674271
50	0.6811153	0.512801	0,1683143

*ce tableaux à été repris du TP sur les voisins proches

5-Evaluation croisée

Pour l'évaluation croisée nous avons posé le nombre de partition à 9, se rapprochant le plus de 10 et étant un diviseur de 4734 qui est le nombre total de nos données.

Algorithme	Pourcentage de mauvaise classification	HyperParameters
Naive Bayes	41.44487%	/
Decision Tree	25.2218%	cp=0.001 minsplit=100 critère de sélection: Gini index
Voisins Proches	32.31939%	k=50(nombre de voisin proche)
Default Classifier	48.96493%	/

Test MCNemar

P-value	Naive-Bayes	Decision Tree	Voisins Proches	Default Classifier
Naive-Bayes	/	p.value<2.2e-16	p.value=1.137e-11	p.value = 2.454e-14
Decision Tree	p.value<2.2e-16	/	p.value<2.2e-16	p.value < 2.2e-16
Voisins Proches	p.value=1.137e-11	p.value<2.2e-16	/	p.value < 2.2e-16

Default Classifier	p.value = 2.454e-14	p.value < 2.2e-16	p.value < 2.2e-16	/

Etant donné que le p.value est systématiquement plus petit que 0.05. On peut rejeter l'hypothèse Nulle. Ce qui veut dire que la différence de performance entre les deux algorithmes de classification est significative.

Voici le tableau ordonnant les différents algorithmes de classification en fonction de leurs scores obtenus lors de la comparaison des algorithmes. Un algorithme est crédité de 1 point si sa performance est significativement meilleure et est crédité de 0,5 point si la différence de performance n'est pas assez significative.

Algorithme de classification	Précision de bonne classification	Score
Decision Tree	74.7782%	3
Voisins Proches	67.68061%	2
Naive-Bayes	58.55513%	1
Default Classifier	51.03507%	0

6-Conclusion

C'est l'algorithme du Decision Tree qui est le meilleur algorithme dans notre cas de figure. En effet, nous observons que c'est l'algorithme Rpart qui possède la précision moyenne la plus élevée, et lors de tous les tests de MCnemar c'est Rpart qui possède "le meilleur score".

Voici comment, nous pensons que Rpart fonctionne. Pour aider la compréhension de notre explication nous prenons comme exemple la première ligne de donnée comme instance :

```
myData[1,]
SE1 SE2 BA1 BA2  BA3 BA4 BA5 BA6 BA7 PE1 PE2 PE3 PE4 PE5 PE6 PE7 PE8 PE9
45 G29  12   0 5934   0   0   0   0 I0  I0  I0  I0  I0  I0  I0  I0 I0
PE10 PE11 PE12 PE13 PE14 PE15 IA1 IA2 IA3 InvType
I0   I0   I0   I0   I0 1    I0   0   0   1      C1
```

Et nous utiliserons le modèle fait lors du TP3 (modèle sur l'ensemble des données selon le Gini index):

```
n= 4734

node), split, n, loss, yval, (yprob)
      * denotes terminal node

  1) root 4734 2318 C1 (0.48964935 0.51035065)
    2) BA3>=18888 2758  896 C0 (0.67512690 0.32487310)
      4) BA3>=97475.5 1050  194 C0 (0.81523810 0.18476190)
        8) SE2=G10,G12,G13,G14,G15,G16,G18,G19,G2,G20,G21,G23,G28,G29,G31,G32,G36,G37,G38,G39,G44,G45,G46,G47,G49,G50,G53,G54,G56,G58,G59,G60,G61,G63,G64,G67,G68,G69,G70,G7
        9) SE2=G0,G17,G22,G24,G25,G26,G27,G30,G33,G34,G35,G40,G42,G43,G48,G51,G52,G55,G62,G65,G66,G71,G74,G75,G77,G79,G80,G81,G82,G86,G92,G93,G96 578  154 C0 (0.73356401 0.
         18) SE1>=50.5 327   61 C0 (0.81345566 0.18654434) *
         19) SE1< 50.5 251   93 C0 (0.62948207 0.37051793)
           38) SE2=G0,G17,G22,G24,G26,G27,G30,G34,G40,G43,G48,G51,G52,G55,G71,G75,G79,G81 216   70 C0 (0.67592593 0.32407407) *
           39) SE2=G25,G33,G42,G62,G65,G66,G74,G77,G80,G82,G93,G96 35   12 C1 (0.34285714 0.65714286) *
      5) BA3< 97475.5 1708  702 C0 (0.58899297 0.41100703)
       10) BA6< 222472 1351  479 C0 (0.64544782 0.35455218)
         20) BA7>=20125 1017  291 C0 (0.71386431 0.28613569)
           40) SE2=G10,G11,G12,G15,G16,G17,G20,G22,G23,G25,G26,G27,G28,G29,G31,G37,G44,G45,G48,G49,G52,G55,G59,G61,G62,G63,G67,G68,G69,G72,G75,G78,G79,G81,G82,G86,G87,G88,G
           41) SE2=G0,G13,G14,G18,G19,G21,G24,G30,G32,G33,G34,G35,G36,G38,G40,G42,G43,G46,G47,G51,G53,G56,G58,G60,G64,G65,G70,G71,G73,G74,G76,G77,G80,G84,G85,G90,G95,G96,G9
             82) BA5< 173177 478  148 C0 (0.69037657 0.30962343)
              164) SE2=G0,G13,G19,G24,G30,G32,G34,G36,G38,G46,G51,G53,G65,G70,G71,G76,G77,G90,G95,G96,G98 387  101 C0 (0.73901809 0.26098191) *
              165) SE2=G14,G18,G21,G33,G35,G40,G42,G43,G47,G58,G60,G64,G73,G74,G84,G85 91   44 C1 (0.48351648 0.51648352) *
             83) BA5>=173177 139   64 C1 (0.46043165 0.53956835)
              166) SE2=G14,G40,G42,G43,G47,G51,G64,G71,G73,G74,G76,G85,G90 33   10 C0 (0.69696970 0.30303030) *
              167) SE2=G0,G13,G18,G21,G24,G30,G32,G34,G36,G53,G56,G58,G65,G70,G77,G80,G84,G95,G96,G98 106   41 C1 (0.38679245 0.61320755) *
         21) BA7< 20125 334  146 C1 (0.43712575 0.56287425)
           42) BA7< 268 157   50 C0 (0.68152866 0.31847134)
             84) SE2=G12,G19,G21,G22,G28,G31,G33,G37,G38,G40,G43,G47,G49,G53,G54,G56,G58,G59,G63,G65,G67,G73,G76,G78,G79,G93,G97,G99 57    3 C0 (0.94736842 0.05263158) *
             85) SE2=G0,G11,G13,G17,G20,G25,G29,G30,G34,G36,G39,G42,G44,G46,G52,G62,G69,G74,G75,G77,G82,G83,G90 100   47 C0 (0.53000000 0.47000000)
              170) SE2=G0,G20,G30,G44,G52,G62,G77 66   24 C0 (0.63636364 0.36363636) *
              171) SE2=G11,G13,G17,G25,G29,G34,G36,G39,G42,G46,G69,G74,G75,G82,G83,G90 34   11 C1 (0.32352941 0.67647059) *
           43) BA7>=268 177   39 C1 (0.22033898 0.77966102) *
       11) BA6>=222472 357  134 C1 (0.37535014 0.62464986)
         22) SE2=G10,G15,G16,G18,G23,G25,G35,G47,G50,G53,G54,G58,G59,G60,G64,G67,G68,G72,G74,G75,G92,G94,G95,G96,G99 91   30 C0 (0.67032967 0.32967033) *
         23) SE2=G0,G19,G20,G21,G22,G24,G26,G27,G30,G32,G34,G37,G38,G40,G42,G43,G44,G45,G46,G48,G49,G51,G52,G55,G56,G62,G66,G69,G70,G71,G73,G76,G77,G79,G81,G85,G90,G91,G93,
    3) BA3< 18888 1976  456 C1 (0.23076923 0.76923077)
      6) BA3< 529 127   27 C0 (0.78740157 0.21259843) *
      7) BA3>=529 1849  356 C1 (0.19253651 0.80746349)
       14) BA7>=7688 697  244 C1 (0.35007174 0.64992826)
         28) BA6< 55252 457  214 C1 (0.46827133 0.53172867)
           56) SE2=G11,G13,G15,G16,G17,G19,G22,G23,G24,G27,G28,G32,G33,G35,G36,G47,G48,G49,G50,G60,G62,G66,G72,G74,G78,G79,G88,G90,G91,G92,G94,G95,G96 109   23 C0 (0.788990
           57) SE2=G0,G10,G12,G14,G18,G20,G21,G26,G29,G30,G34,G37,G38,G40,G42,G43,G44,G45,G46,G51,G52,G53,G55,G58,G59,G61,G63,G64,G65,G67,G69,G70,G71,G73,G75,G76,G77,G81,G8
            114) SE2=G0,G10,G12,G18,G21,G26,G29,G30,G34,G40,G43,G44,G46,G51,G52,G64,G67,G69,G70,G75,G76,G84,G98,G99 265  115 C1 (0.43396226 0.56603774)
              228) BA7< 23038 214  106 C1 (0.49532710 0.50467290)
                456) SE1< 24.5 34    7 C0 (0.79411765 0.20588235) *
                457) SE1>=24.5 180   79 C1 (0.43888889 0.56111111)
                  914) SE1>=44.5 80   37 C0 (0.53750000 0.46250000) *
                  915) SE1< 44.5 100   36 C1 (0.36000000 0.64000000) *
              229) BA7>=23038 51    9 C1 (0.17647059 0.82352941) *
           115) SE2=G14,G20,G37,G38,G42,G45,G53,G55,G58,G59,G61,G63,G65,G71,G73,G77,G81,G85,G93,G97 83   13 C1 (0.15662651 0.84337349) *
         29) BA6>=55252 240   30 C1 (0.12500000 0.87500000) *
       15) BA7< 7688 1152  112 C1 (0.09722222 0.90277778) *
```

Rpart va prendre la valeur de l'attribut BA3 (qui est l'élément racine de l'arbre de décision et est l'attribut avec le gini index (indice d'impureté le plus faible) ou l'information gain la plus élevée) de notre instance, il va ensuite suivant sa valeur aller dans le sous-arbres de gauche ou de droite exemple:

myData[1,]$BA3<18888 (il ira donc au points (3) le sous-arbres de de droite)

Lorsqu'il arrive dans ce sous arbre il refait le même raisonnement mais avec l'élément racine de ce sous-arbre. Ce raisonnement, çad le la racine du sous-arbre possède le gini index le plus faible (ou information gain le plus élevé) avec les données ou BA3 est plus petit que 18888,se poursuit jusqu'à ce que rpart "tombe" sur une feuille) exemple:

myData[1,]$BA3< 529 (il ira donc au points (6) le sous-arbres de gauche)

Arrivé sur une feuille (cela signifie qu'il n'y pas de gini index plus petit (pour l'information gain c'est le contraire)) la probabilité la plus grande ,que ce soit CO ou C1, "l'emporte" exemple:

le point 6) est une feuille et don prédit le produit d'investissement C0 avec 78,74%

Pour conclure ce projet, l'analyse exploratoire ne nous a pas donné de résultats probants, nous entendons par là qu'aucun attribut "se montrait" inutile (exception part pour IA2).

lors du TP2 (ref: Point 3 de ce rapport) nous avions conclu que les attributs BA3,BA7,BA5 étaient les plus importants selon l'information gain. En observant l'image du modèle créé par Rpart, ci-dessus, nous remarquons que seuls les attributs BA3,BA6,BA7,SE2,SE1 sont nécessaires pour pouvoir proposer un produit aux clients, qui se trouve respectivement à la 1, 16, 2, 6, 5 ème place dans le tableau du de l'information gain. Toutefois, ce sont ces attributs qui possèdent le gini index le plus faible.

Pour le point 4, Evaluation des différents algorithmes, nous observons que le decision tree possède la meilleur précision, une précision de plus de 70% en moyenne.

Annexe

TP-Arbre de décision

Rapport

TP-03 DataMining

Introduction

Lors du deuxième TP, il nous a été demandé d'analyser une seconde fois les différents attributs mais, cette fois-ci, avec une approche quantitative. En effet, les consignes était d'implémenter le code R calculant l'Entropie, l'Entropie Conditionnel et l'information gain. Et d'ensuite analyser certaines données.

Pour ce troisième TP, nous devons utiliser la méthode de classification Rpart. Cette méthode permet de construire un modèle de prédiction. Le but de ce TP est de comprendre comment les arbres de décision fonctionnent. Sur base de cette méthode, nous devons construire 5 modèles avec une séparation, du jeux de données, différentes (en fonction du gini index et de l'information gain). Ainsi que d'appliquer la méthode sur la totalité des données(aussi pour le gini index et l'information gain).

Description du Problème

Une Banque, nous demande de réaliser un modèle qui permettra de déterminer quel produit d'investissement à proposer à un client, suivant son profil. La Banque nous met à disposition une série de données, qui doit être analysée afin de déterminer les attributs dits "utiles" et de pouvoir construire ce modèle.

Attribut continu: SE1,BA1-BA7
Attribut discret: SE2,PE1-PE15, IA1-IA3
Variable cible: InvType

Précision moyenne Information Gain

CP= "*complexity parameter. Any split that does not decrease the overall lack of fit by a factor of cp is not attempted*" (definition du cours)
minsplit="*the minimum number of observations that must exist in a node in order for a split to be attempted."* (definition du cours)

minsplit\CP	0,001	0,01	0,1	0,2
5	0,7330789	0,7368821	0,717237	0,717237
10	0,7299113	0,7368821	0,717237	0,717237
50	0,7287706	0,7368821	0,717237	0,717237
100	0,731052	0,7368821	0,717237	0,717237
200	0,7332066	0,7362484	0,717237	0,717237

Précision moyenne Gini Index

minsplit\CP	0,001	0,01	0,1	0,2
5	0,7390368	0,7362484	0,717237	0,717237
10	0,7375158	0,7362484	0,717237	0,717237
50	0,7356147	0,7362484	0,717237	0,717237
100	0,7414449	0,7362484	0,717237	0,717237
200	0,7382736	0,7357414	0,717237	0,717237

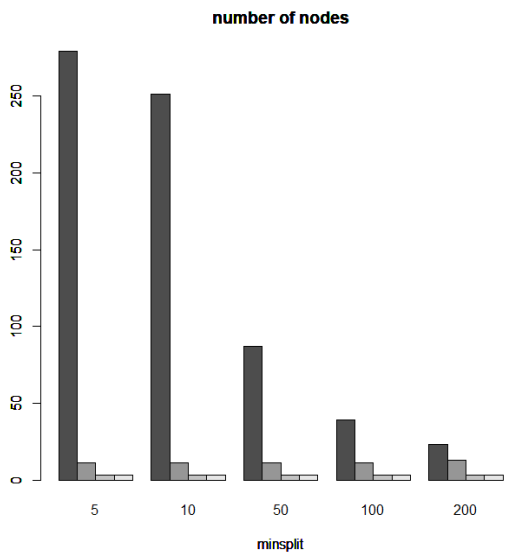
Observation:

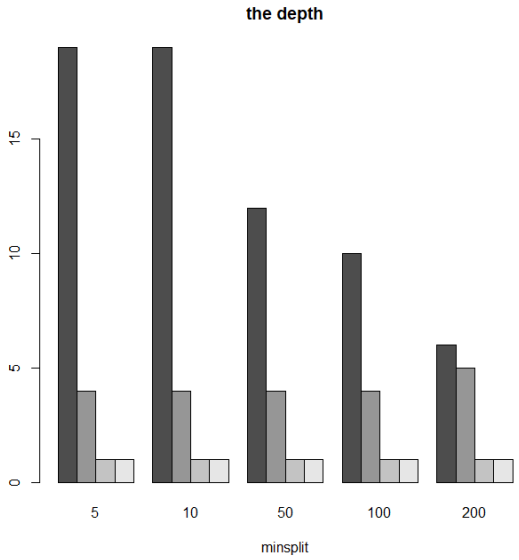
Nous observons que pour la valeur 0.001 (du CP) la précision moyenne (pour les deux tableaux séparées), en fonction du minsplit, varie. Ce qui n'est pas le cas pour les valeurs 0.01,0.1 et 0.2 (pour les minsplit données). De plus, nous observons que les valeur 0.1 et 0.2 (de CP) ne varient pas même entre les deux tableaux

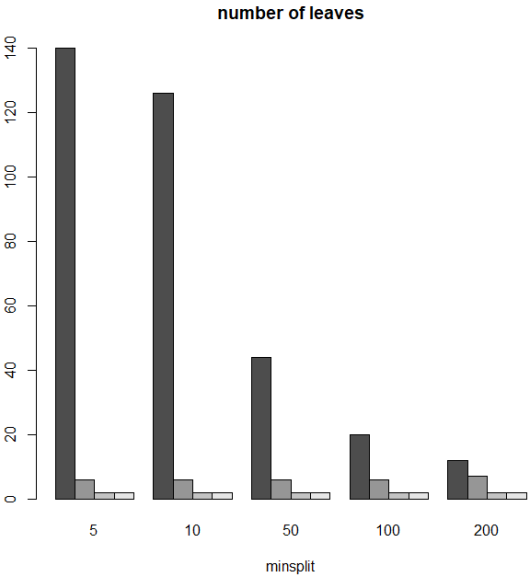
Il est annoté que pour CP=0.01 il y a une légère baisse dans la précision pour l'information gain et pour le Gini index, lorsque le minsplit vaut 200 nous abordons le sujet plus tard dans ce rapport.

Taille des arbres

Les graphes sont basés, par facilité, sur le dernier modèle créé, qui utilise l'information gain.
Légende:
les bâtonnets de Gauche à droite:
1) cp=0.001
2) cp=0.01
3) cp=0.1
4) cp=0.2







Suivant les différents valeurs que prend cp nous observons que:

1. CP=0.001

Plus le minsplit est “grand “ plus la dimension de l’arbre est petite. (Exception: dans le graphe “the depth" lorsque minsplit prends les valeurs 5 et 10 la profondeur l’arbre ne change pas)

2. CP=0.01

Nous observons, que les dimensions sont constantes (suivant le minsplit), sauf lorsque le minsplit vaut 200 où nous remarquons une légère augmentation.

Nous avons émis une hypothèse quant à cette augmentation. En effet, nous pensons que lorsque le minsplit prend la valeur 200, il rejette un nœud décisionnel qui “utilise cette” valeur (que peut prendre l’attribut). Valeur qui permettait à l’attribut d’avoir le gini index le plus faible (ou l’information gain le plus grand) et ainsi d’être un nœud de décision, dont un des enfants est une feuille (dans les modèle avec un minsplit plus faible). Or puisque dans notre situation ce nœud est “rejeté”, Rpart n’a donc pas d’autre choix que de continuer à ajouter des nœuds jusqu’à obtenir un nœud avec le gini index le plus faible.

Nous pouvons ainsi faire la corrélation avec l'augmentation de la dimension de l'arbre ainsi qu'avec la baisse de précision remarquée ci-dessus (une augmentation).

3. CP=0.1 & CP=0.2

Lorsque le CP "prend" ces valeurs là, nous observons que les dimensions de l'arbre ne varie pas en fonction du minsplit.

En conclusion le *minsplit* contrôle la taille de l'arbre en limitant le nombre minimum d'observation nécessaire avant de *split.* En nous basant sur nos observations, nous aurons tendance à dire qu'en général plus grand est le *minsplit* plus petit sera l'arbre. Toutefois, il est à noter que si le minsplit est trop élevé, nous obtiendrons un arbre plus petit MAIS une précision beaucoup plus faible. Car certaines valeurs "essentielles" pour la décision entre C0 et C1 ne seront pas prises en compte.

Analyse des deux modèles finaux

Modèle avec le Gini index(arbre1):
Paramètres: minsplit=100, cp=0.001

explication: L'élément racine est basée sur l'attribut BA3 avec une valeur décisionnelle de 18890. L'arbre est construit de 47 nœuds dont 24 sont des feuilles et ce sur 9 niveaux. Après le test de chaque nœud se trouve le nombre d'instances dans ce nœud, ensuite le nombre d'instances qui ont été mal classées et pour finir la classe attachée à ce nœud. Dans les parenthèses se trouve le pourcentage des différentes classes dans le nœud.

Modèle avec l' information gain(arbre2):
Paramètres: minsplit=100, cp=0.01

```
[1] "rpart with information gain on the full dataset"
n= 4734

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 4734 2318 C1 (0.4896493 0.5103507)
   2) BA3>=18888 2758  896 C0 (0.6751269 0.3248731)
     4) BA3>=97475.5 1050  194 C0 (0.8152381 0.1847619) *
     5) BA3< 97475.5 1708  702 C0 (0.5889930 0.4110070)
      10) BA6< 222472 1351  479 C0 (0.6454478 0.3545522)
        20) BA7>=20125 1017  291 C0 (0.7138643 0.2861357) *
        21) BA7< 20125 334  146 C1 (0.4371257 0.5628743)
          42) BA7< 268 157   50 C0 (0.6815287 0.3184713) *
          43) BA7>=268 177   39 C1 (0.2203390 0.7796610) *
      11) BA6>=222472 357  134 C1 (0.3753501 0.6246499)
        22) SE2=G10,G15,G16,G18,G23,G25,G35,G47,G50,G53,G54,G58,$
        23) SE2=G0,G19,G20,G21,G22,G24,G26,G27,G30,G32,G34,G37,G$
   3) BA3< 18888 1976  456 C1 (0.2307692 0.7692308)
     6) BA3< 529 127   27 C0 (0.7874016 0.2125984) *
     7) BA3>=529 1849  356 C1 (0.1925365 0.8074635) *
```

explication: L'élément racine est basée sur l'attribut BA3 avec une valeur décisionnelle de 18890. L'arbre est construit de 15 nœuds dont 8 sont des feuilles et ce sur 5 niveaux. Après le test de chaque nœud se trouve le nombre d'instances dans ce nœud, ensuite le nombre d'instances qui ont été mal classées et pour finir la classe attachée à ce nœud. Dans les parenthèses se trouve le pourcentage des différentes classes dans le nœud.

Les attributs les plus importants sont : BA3, BA6, BA7 et SE2. C'est sur base de ces attributs là que le modèle va prédire la classe. Par exemple pour l'arbre2, si l'attribut BA3 d'une instance est entre 529 et 18890. Le modèle prédit la classe C0.
Ces attributs ont respectivement le *Gini index* le plus bas ou *l'information gain* le plus élevée. Le critère de sélection de tous les attributs est calculé après chaque *split*. Si il n'y a pas d'attributs avec un critère de sélection plus performant, il s'agit alors d'une feuille sinon on rajoute cet attribut en tant que nouveau nœud.

Comparaison des deux modèles

Les deux arbres construits par les deux modèles finaux ne sont pas identiques. L'arbre créé avec le critère de sélection *Gini index (arbre1)* est plus grand en taille comparé à l'arbre créé avec le critère de sélection *Information Gain (arbre2)*. L'arbre1 possède 24 feuilles contre 8 pour l'arbre2 et est construit sur 9 niveaux contre 5 pour l'arbre2.
Les deux arbres utilisent en grande partie les mêmes attributs mais l'arbre1 étant plus grand, utilise plus d'attributs que l'arbre2.
Ceci est lié au fait que le *complexity parameter* est plus petit pour l'arbre1 que pour l'arbre2.

Classification d'une instance

Voici comment, nous pensons que Rpart fonctionne. Pour aider la compréhension de notre explication nous prenons comme exemple la première ligne de donnée comme instance :

myData[1,]
SE1 SE2 BA1 BA2 BA3 BA4 BA5 BA6 BA7 PE1 PE2 PE3 PE4 PE5 PE6 PE7 PE8 PE9
45 G29 12   0 5934   0   0   0   0 I0 I0 I0 I0 I0 I0 I0 I0 I0
PE10 PE11 PE12 PE13 PE14 PE15 IA1 IA2 IA3 InvType
I0   I0   I0   I0   I0 1   I0   0   0   1   C1

Et nous utiliserons le modèle ci-dessus (modèle sur l'ensemble des données selon le gini index):

```
n= 4734

node), split, n, loss, yval, (yprob)
      * denotes terminal node

  1) root 4734 2318 C1 (0.48964935 0.51035065)
    2) BA3>=18888 2758  896 C0 (0.67512690 0.32487310)
      4) BA3>=97475.5 1050  194 C0 (0.81523810 0.18476190)
        8) SE2=G10,G12,G13,G14,G15,G16,G18,G19,G2,G20,G21,G23,G28,G29,G31,G32,G36,G37,G38,G39,G44,G45,G46,G47,G49,G50,G53,G54,G56,G58,G59,G60,G61,G63,G64,G67,G68,G69,G70,G7
        9) SE2=G0,G17,G22,G24,G25,G26,G27,G30,G33,G34,G35,G40,G42,G43,G48,G51,G52,G55,G62,G65,G66,G71,G74,G75,G77,G79,G80,G81,G82,G86,G92,G93,G96 578  154 C0 (0.73356401 0.
         18) SE1>=50.5 327   61 C0 (0.81345566 0.18654434) *
         19) SE1< 50.5 251   93 C0 (0.62948207 0.37051793)
           38) SE2=G0,G17,G22,G24,G26,G27,G30,G34,G40,G43,G48,G51,G52,G55,G71,G75,G79,G81 216   70 C0 (0.67592593 0.32407407) *
           39) SE2=G25,G33,G42,G62,G65,G66,G74,G77,G80,G82,G93,G96 35   12 C1 (0.34285714 0.65714286) *
      5) BA3< 97475.5 1708  702 C0 (0.58899297 0.41100703)
       10) BA6< 222472 1351  479 C0 (0.64544782 0.35455218)
         20) BA7>=20125 1017  291 C0 (0.71386431 0.28613569)
           40) SE2=G10,G11,G12,G15,G16,G17,G20,G22,G23,G25,G26,G27,G28,G29,G31,G37,G44,G45,G48,G49,G52,G55,G59,G61,G62,G63,G67,G68,G69,G72,G75,G78,G79,G81,G82,G86,G87,G88,G
           41) SE2=G0,G13,G14,G18,G19,G21,G24,G30,G32,G33,G34,G35,G36,G38,G40,G42,G43,G46,G47,G51,G53,G56,G58,G60,G64,G65,G70,G71,G73,G74,G76,G77,G80,G84,G85,G90,G95,G96,G9
             82) BA5< 173177 478  148 C0 (0.69037657 0.30962343)
              164) SE2=G0,G13,G19,G24,G30,G32,G34,G36,G38,G46,G51,G53,G65,G70,G71,G76,G77,G90,G95,G96,G98 387  101 C0 (0.73901809 0.26098191) *
              165) SE2=G14,G18,G21,G33,G35,G40,G42,G43,G47,G58,G60,G64,G73,G74,G84,G85 91   44 C1 (0.48351648 0.51648352) *
             83) BA5>=173177 139   64 C1 (0.46043165 0.53956835)
              166) SE2=G14,G40,G42,G43,G47,G51,G64,G71,G73,G74,G76,G85,G90 33   10 C0 (0.69696970 0.30303030) *
              167) SE2=G0,G13,G18,G21,G24,G30,G32,G34,G36,G53,G56,G58,G65,G70,G77,G80,G84,G95,G96,G98 106   41 C1 (0.38679245 0.61320755) *
         21) BA7< 20125 334  146 C1 (0.43712575 0.56287425)
           42) BA7< 268 157   50 C0 (0.68152866 0.31847134)
             84) SE2=G12,G19,G21,G22,G28,G31,G33,G37,G38,G40,G43,G47,G49,G53,G54,G56,G58,G59,G63,G65,G67,G73,G76,G78,G79,G93,G97,G99 57    3 C0 (0.94736842 0.05263158) *
             85) SE2=G0,G11,G13,G17,G20,G25,G29,G30,G34,G36,G39,G42,G44,G46,G52,G62,G69,G74,G75,G77,G82,G83,G90 100   47 C0 (0.53000000 0.47000000)
              170) SE2=G0,G20,G30,G44,G52,G62,G77 66   24 C0 (0.63636364 0.36363636) *
              171) SE2=G11,G13,G17,G25,G29,G34,G36,G39,G42,G46,G69,G74,G75,G82,G83,G90 34   11 C1 (0.32352941 0.67647059) *
           43) BA7>=268 177   39 C1 (0.22033898 0.77966102) *
       11) BA6>=222472 357  134 C1 (0.37535014 0.62464986)
         22) SE2=G10,G15,G16,G18,G23,G25,G35,G47,G50,G53,G54,G58,G59,G60,G64,G67,G68,G72,G74,G75,G92,G94,G95,G96,G99 91   30 C0 (0.67032967 0.32967033) *
         23) SE2=G0,G19,G20,G21,G22,G24,G26,G27,G30,G32,G34,G37,G38,G40,G42,G43,G44,G45,G46,G48,G49,G51,G52,G55,G56,G62,G66,G69,G70,G71,G73,G76,G77,G79,G81,G85,G90,G91,G93,
    3) BA3< 18888 1976  456 C1 (0.23076923 0.76923077)
      6) BA3< 529 127   27 C0 (0.78740157 0.21259843) *
      7) BA3>=529 1849  356 C1 (0.19253651 0.80746349)
       14) BA7>=7688 697  244 C1 (0.35007174 0.64992826)
         28) BA6< 55252 457  214 C1 (0.46827133 0.53172867)
           56) SE2=G11,G13,G15,G16,G17,G19,G22,G23,G24,G27,G28,G32,G33,G35,G36,G47,G48,G49,G50,G60,G62,G66,G72,G74,G78,G79,G88,G90,G91,G92,G94,G95,G96 109   23 C0 (0.788990
           57) SE2=G0,G10,G12,G14,G18,G20,G21,G26,G29,G30,G34,G37,G38,G40,G42,G43,G44,G45,G46,G51,G52,G53,G55,G58,G59,G61,G63,G64,G65,G67,G69,G70,G71,G73,G75,G76,G77,G81,G8
            114) SE2=G0,G10,G12,G18,G21,G26,G29,G30,G34,G40,G43,G44,G46,G51,G52,G64,G67,G69,G70,G75,G76,G84,G98,G99 265  115 C1 (0.43396226 0.56603774)
              228) BA7< 23038 214  106 C1 (0.49532710 0.50467290)
                456) SE1< 24.5 34    7 C0 (0.79411765 0.20588235) *
                457) SE1>=24.5 180   79 C1 (0.43888889 0.56111111)
                  914) SE1>=44.5 80   37 C0 (0.53750000 0.46250000) *
                  915) SE1< 44.5 100   36 C1 (0.36000000 0.64000000) *
              229) BA7>=23038 51    9 C1 (0.17647059 0.82352941) *
             115) SE2=G14,G20,G37,G38,G42,G45,G53,G55,G58,G59,G61,G63,G65,G71,G73,G77,G81,G85,G93,G97 83   13 C1 (0.15662651 0.84337349) *
         29) BA6>=55252 240   30 C1 (0.12500000 0.87500000) *
       15) BA7< 7688 1152  112 C1 (0.09722222 0.90277778) *
```

Rpart va prendre la valeur de l'attribut BA3 (qui est l'élément racine de l'arbre de décision et est l'attribut avec le gini index (indice d'impureté le plus faible) ou l'information gain la plus élevée) de notre instance, il va ensuite suivant sa valeur aller dans le sous-arbres de gauche ou de droite exemple:

myData[1,]$BA3<18888 (il ira donc au points (3) le sous-arbres de de droite)

Lorsqu'il arrive dans ce sous arbre il refait le même raisonnement mais avec l'élément racine de ce sous-arbre. (Ce raisonnement se poursuit jusqu'à ce que rpart "tombe" sur une feuille) exemple:

myData[1,]$BA3< 529 (il ira donc au points (6) le sous-arbres de gauche)

Arrivé sur une feuille (cela signifie qu'il n'y pas de gini index plus petit (pour l'information gain c'est le contraire)) la probabilité la plus grande ,que ce soit CO ou C1, "l'emporte" exemple:

le point 6) est une feuille et don prédit le produit d'investissement C0 avec 78,74%

Conclusion

Pour conclure ce rapport, Rpart va, selon le min split et le CP, calculer le le gini index (ou l'information gain) de chaque attribut. l'attribut ayant le gini index le plus faible (ou l'information gain la plus élevé) sera l'élément racine de notre arbre. Il va alors recalculer le gini index de cet attribut, suivant la séparation effectuée. Si un attribut existe avec un gini index plus faible (ou information gaine la plus élevé) alors la prochaine séparation se fera sur cet attribut, avec le même raisonnement.

Au vu des observations effectuées ci-dessus, nous en avons conclu que minsplit faisait varier la taille de l'arbre. En général, plus le minsplit sera grand, plus petit sera l'arbre.

TP-Voisin Proche

Rapport

TP 5: Voisin proches

Introduction

Lors du quatrième TP, nous devions utiliser la méthode de classification Naive Bayes. Cette méthode permet de construire un modèle de prédiction. Le but de ce TP était de comprendre comment Naive Bayes fonctionne. Sur base de cette méthode, nous devions construire 5 modèles avec une séparation, du jeux de données, différentes. Ainsi que d'appliquer la méthode sur la totalité des données.

Pour cinquième TP, nous devons construire les modèles, en utilisant la library class, suivant le voisin le plus proche. Il nous est demandé de construire 5 modèles avec une répartition des données différentes pour ensuite calculer la précision moyenne de ces 5 modèles. Ceci est appliquée pour différentes “distance” (k={1,3,5,10,20,50})

Description du Problème

Une Banque, nous demande de réaliser un modèle qui permettra de déterminer quel produit d’investissement à proposer à un client, suivant son profil. La Banque nous met à disposition une série de données, qui doit être analysée afin de déterminer les attributs dits "utiles" et de pouvoir construire ce modèle.

Attribut continu: SE1,BA1-BA7
Attribut discret: SE2,PE1-PE15, IA1-IA3
Variable cible: InvType

Données obtenues

K	Précision moyenne knn	Précision moyenne default classifier	Comparaison
1	0.626616	0.512801	0,113815
3	0.6494297	0.512801	0,1366287
5	0.6640051	0.512801	0,1512041

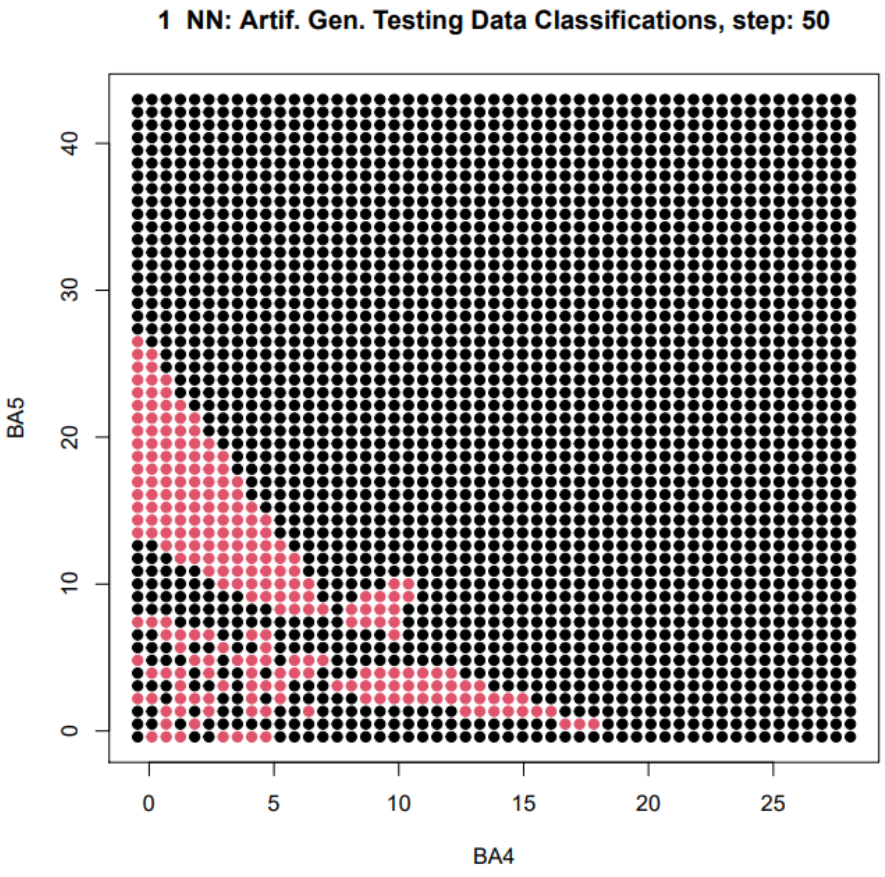
10	0.6724968	0.512801	0,1596958
20	0.6802281	0.512801	0,1674271
50	0.6811153	0.512801	0,1683143

Nous observons que plus l'algorithme k-NN prend en compte de voisin proche plus la précision moyenne de notre modèle augmente.

Visualisation de k-NN

dans les graphes ci-dessous le rouge correspond aux instances ayant comme valeur pour l'attribut cible le produit C_ et le noir au C_

K=1

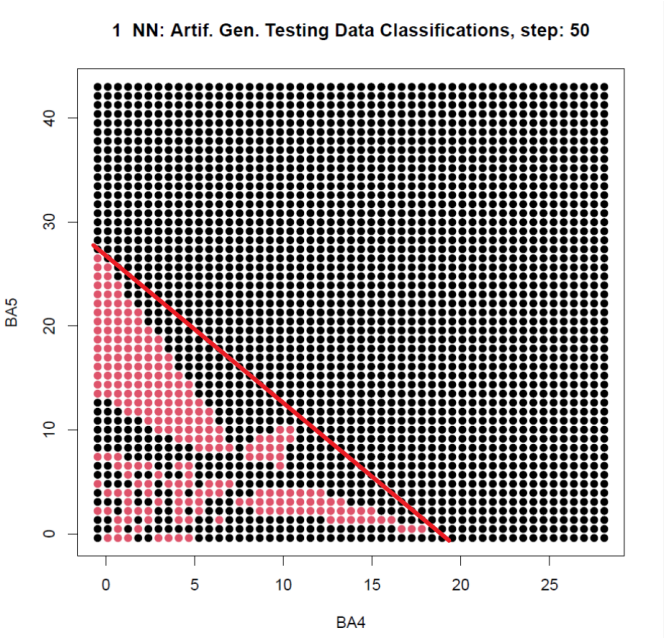


Avec 1 comme distance (de voisin proche), nous obtenons un graphe avec une "frontière de class" très complexe, çad qu'il est difficile d'établir une séparation claire entre C0 et C1. En effet, le voisin le plus proche fonctionne ainsi: lors d'une nouvelle instance, celle-ci est ajoutée aux données desquelles nous connaissons déjà les valeurs de l'attribut cible. Nous observons ensuite ces voisins les plus proches. Le modèle prédit ensuite la valeur qui est la

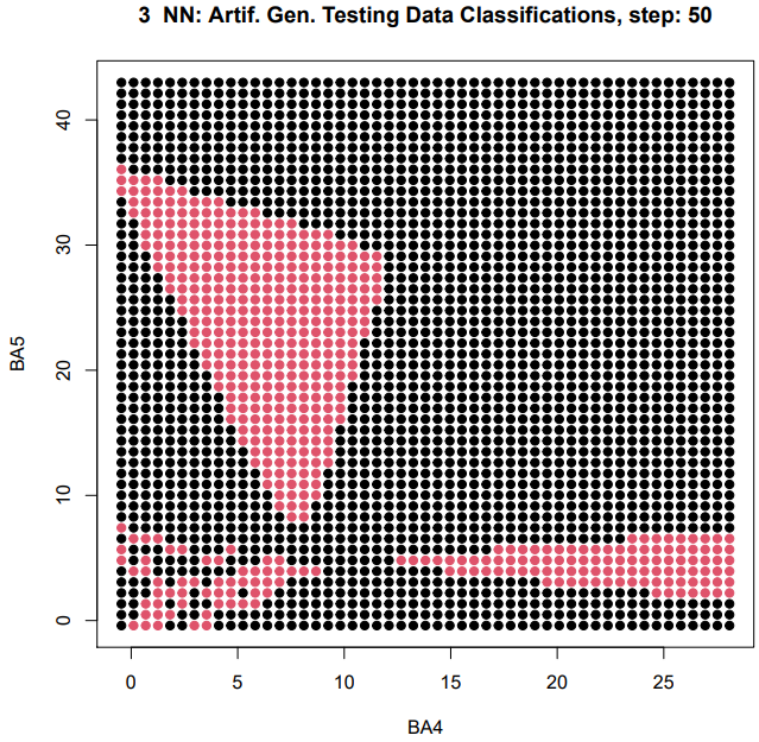
plus présente chez les voisins comme valeur de l'attribut cible. Dans notre cas puisqu'on regardera seulement le voisin le plus proche, la nouvelle instance prendra la valeur, pour l'attribut cible, de ce voisin.

Que nous dit le graphe?

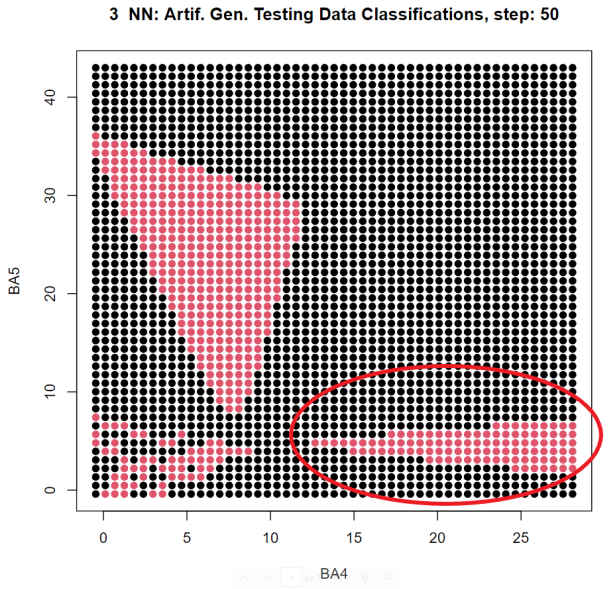
Nous observons une séparation définie par la ligne rouge (dans l'image ci-dessous), ou toutes les instances définie (au dessus de la séparation) ont été prédit avec le produit d'investissement en noir(C1 ou C0).Ce qui nous amène à émettre, à ce stade ci, l'hypothèse que dans cette zone, il y a très peu d'instance ayant comme valeur (Rouge) par rapport aux instances (Noir) dans cette tranche de valeur pour BA5 et BA4 (En ajoutant au fait que le graphe est complexe et donc très sensible aux fluctuations). Pour la zone, situé en dessous de la droite, il est difficile de dire quoi que ce soit, car il n'y a pas vraiment de couleur dominante, nous en concluons que dans cette zone il y a plus de diversité, au niveaux des deux produits, que dans la zone au-dessus.



K=3

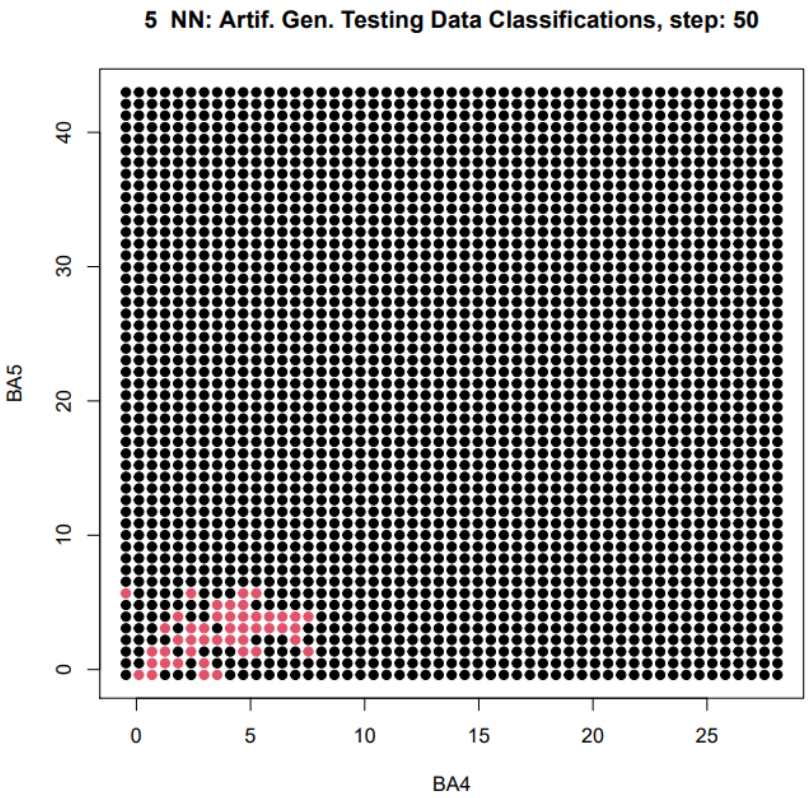


Notre première hypothèse faite ci-dessus, est déjà mise à mal:



En effet, nous observons que dans cette zone , en augmentant les voisins proches, la prédiction des instances pour l'attribut cible change. Ce qui démontre que dans cette zone, les 3 voisins les plus proches de l'instance de test prédisent majoritairement (Rouge) (lorsqu'une instance change de valeur).

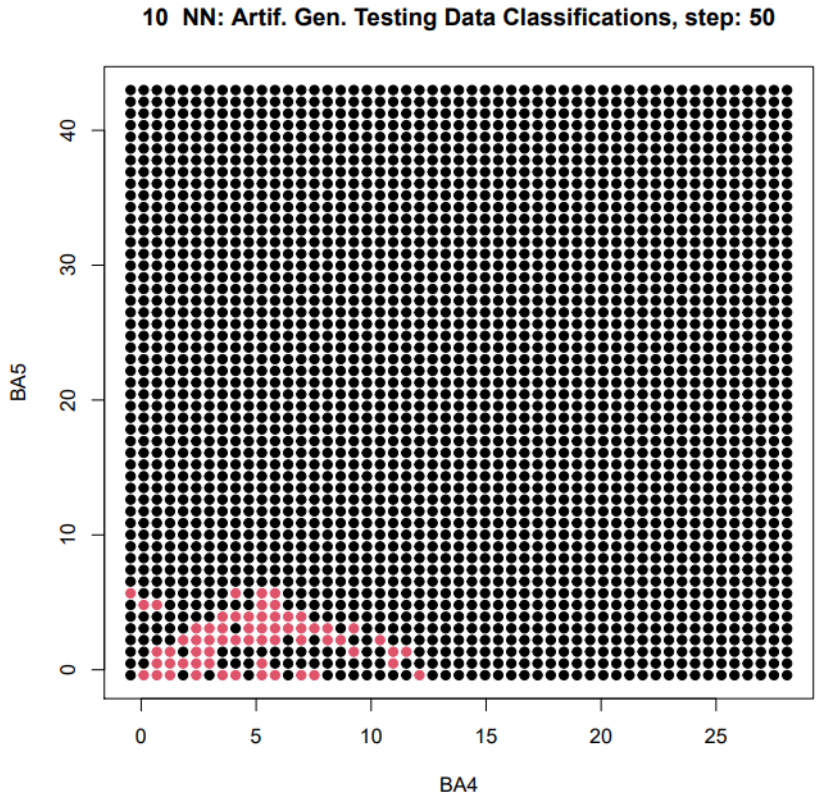
K=5



Nous observons que, lorsque nous prenons en compte les 5 voisin les plus proches, le modèle prédit majoritairement le produit associé à la couleur noir. Nous avons donc que pour chaque instance de test, les 5 voisins les plus proches predisent majoritairement NOIR.

En observant le graphe nous constatons que la précision doit se rapprocher de la précision du “*default clasifier*”. En effet dans les deux cas les modèle prédisent majoritairement le (NOIR)

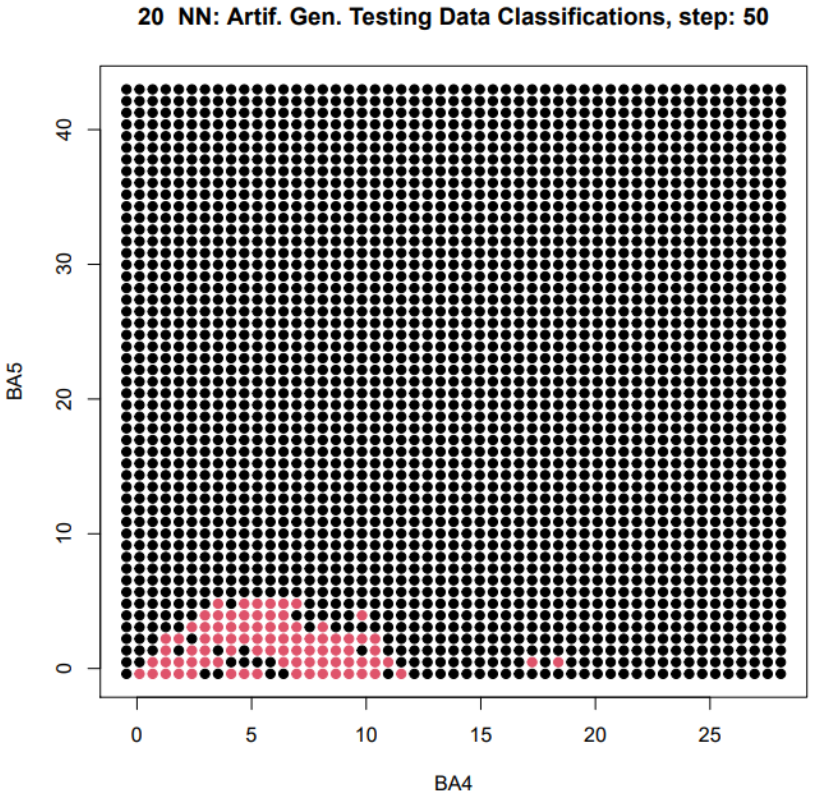
K=10



Nous observons que, lorsque nous prenons en compte les 10 voisin les plus proches, le modèle prédit en général le produit associé à la couleur noir. Nous avons donc que pour chaque instance de test, les 10 voisins les plus proches predisent majoritairement NOIR.

En observant le graphe nous constatons, une nouvelle fois, que la précision doit se rapprocher de la précision du "*default clasifier*". En effet dans les deux cas les modèle prédisent majoritairement le (NOIR)

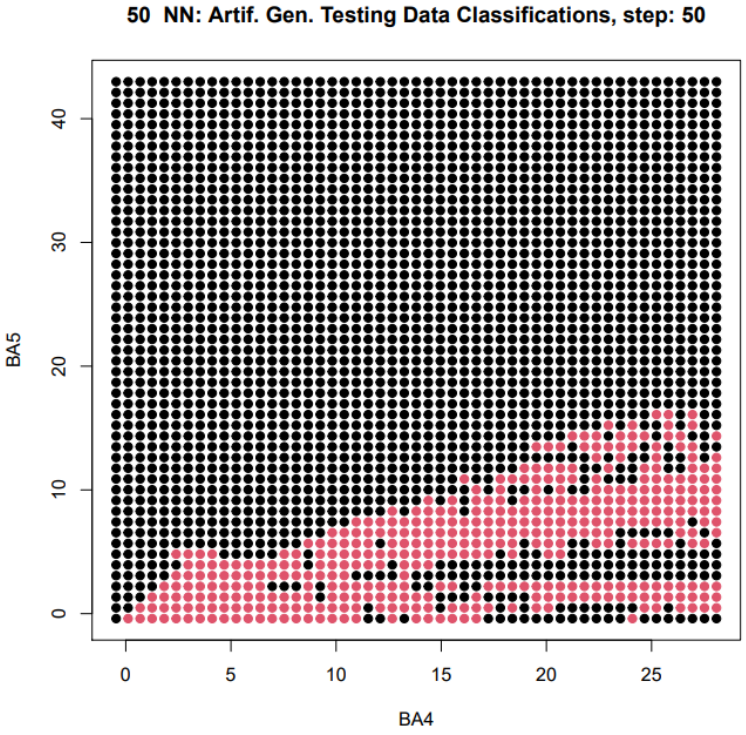
K=20



Nous observons que, lorsque nous prenons en compte les 20 voisin les plus proches, le modèle prédit majoritairement le produit associé à la couleur noir. Nous avons donc que pour chaque instance de test, les 20 voisins les plus proches ont majoritairement la valeur NOIR comme valeur pour l'attribut cible.

De plus, nous constatons que les graphes 5NN, 10NN et 20NN sont très similaires et auront par conséquent une précision qui varie peu par rapport aux deux autres.

K=50



Nous commençons à observer une constante. En effet, avec k=50 nous avons un graphe qui n'est plus aussi sensible au fluctuation que pour le graphe 1. Si nous ajoutons, une instance au jeu de test, le graphe ne changera pas autant que le graphe 1 car il doit prédire la classe majoritaire qui se trouve dans les 50 voisins les plus proches.

Conclusion

Lors de nos observations, nous avons constaté que plus on augmentait le nombre de voisins, plus la précision moyenne augmentait. Toutefois, nous pensons que si on augmente trop le nombre de voisin proche nous nous rapprocherons de la précision du *default* classifier (cas extrême où nous prenons l'ensemble des données comme voisins proches).