Université de Nice Sophia Antipolis

RAPPORT DE PROJET

DATA MINING

Projet réalisé par

Viktoriia KUDYMOVSKA

Année universitaire 2018/2019

1. ANALYSE EXPLORATOIRE

On affiche **l'histogramme d'effectifs** de la variable LOANS (*Figure 1*) en affichant la proportion de chacune des classes pour chaque barre (en posant color=RISK). On fixe aussi la largeur des barres à 1 en utilisant le paramètre binwidth :
>qplot(LOANS, data=risk, color=RISK, binwidth = 1)

Le risque associé à la demande de carte de crédit pour les clients n'ayant pas d'emprunts en cours est plus fréquemment de la classe GOOD RISK. Les clients ayant au moins un emprunt en cours dans la plupart des cas se retrouvent dans la classe BAD PROFIT.

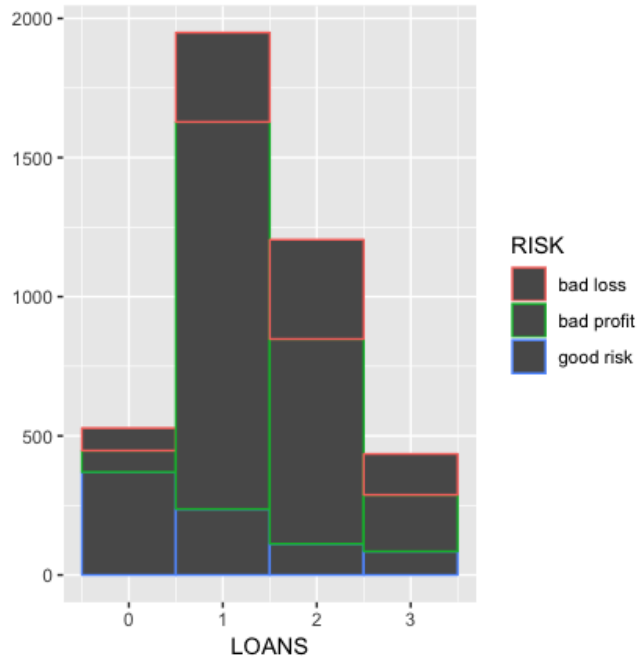

Figure 1

Afin de vérifier ses observations, on affiche les effectifs de chacune des valeurs la variable LOANS pour chaque valeur de la variable RISK par la commande suivante :
>table(LOANS, RISK)

On fait de même pour la variable INCOME *(Figure 2)* :
>qplot(INCOME, data=risk, color=RISK, binwidth = 10000)

Pour les clients dont le revenu annuel est plus important, le risque associé à la demande de carte de crédit diminue.

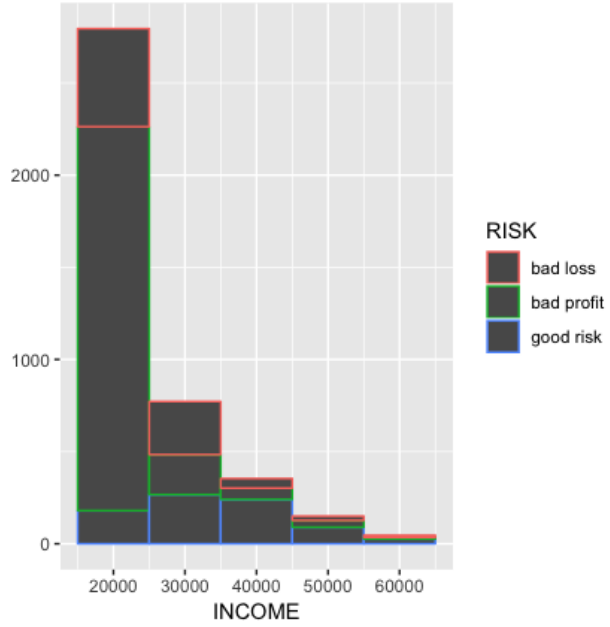

Figure 2

La majorité des instances avec la valeur « Yes » pour MORTGAGE *(Figure 3)* sont de la classe BAD PROFIT, le risque étant visiblement plus élevé que pour la valeur « No ».
> qplot(MORTGAGE, data=risk, color=RISK)

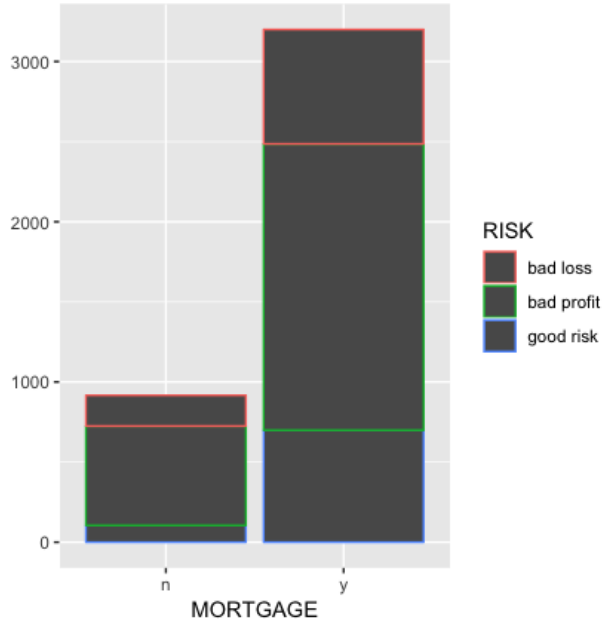

Figure 3

On crée ensuite une **boîte à moustaches** (*Figure 4*) afin de comparer la distribution des valeurs de la variable INCOME pour chaque valeur de RISK. On utilise le paramètre col=c(« red », « blue », « green ») afin d'afficher en bleu la boîte pour RISK=BAD PROFIT, en rouge la boîte pour RISK=BAD LOSS et en vert pour RISK=GOOD RISK.
>boxplot(INCOME~RISK, data=risk, col=c("red","blue", "green"), main="Revenus selon Risque", xlab="Risque", ylab="Revenus")

On peut observer des distributions différentes des valeurs de INCOME pour les trois classes car les positions verticales des 3 boîtes et de leurs moustaches sont décalées.
A l'aide d'une commande *tapply(INCOME, RISK, summary)* on affiche les statistiques élémentaires de INCOME pour chacune des trois classes (les valeurs minimale et maximale, les quartiles, la médiane).

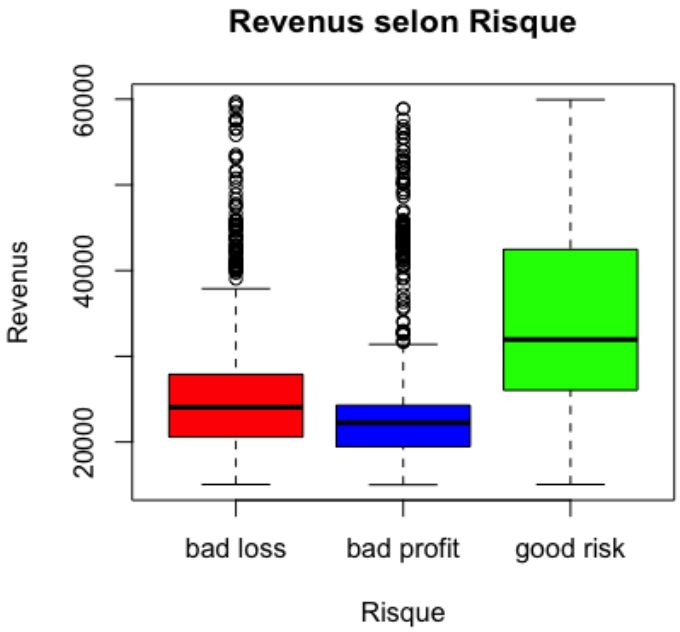

Figure 4

Par exemple, la valeur médiane pour BAD LOSS est égale a 24054.

Enfin, on affiche un **nuage de points** avec pour axe des abscisses la variable INCOME et pour axe des ordonnées la variable AGE (*Figure 5*) :
>qplot(INCOME, AGE, data=risk, color=RISK) + geom_jitter()

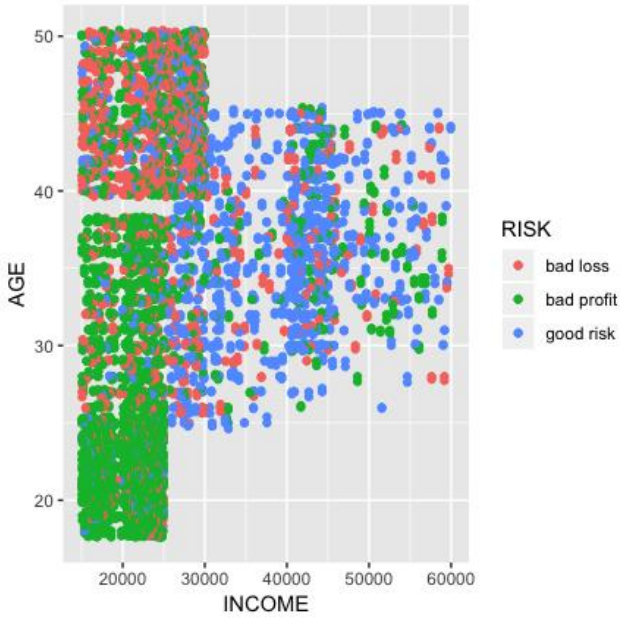

Figure 5

L’instruction + *geom_jitter()* permet de distinguer les points confondus sur le graphique.
Pour les lignes 18-25 approximativement, on observe une zone constituée quasi-exclusivement de points de la même couleur (vert=BAD PROFIT). Après, pour les lignes 25-45 à partir d’une certaine valeur de INCOME la majorité des instances appartiennent à la même classe (GOOD RISK).

On procède de façon identique pour les autres variables (*cf fichier de commandes*).

2. CONSTRUCTION ET SELECTION DU CLASSIFIEUR

En utilisant la **méthode par partitionnement**, on construit l'Ensemble d'Apprentissage (2/3 des exemples de l'ensemble de données) et l'Ensemble de Test :
>risk_EA <- risk[1:2745,]
>risk_ET <- risk[2746:4117,]

Le numéro identifiant les clients ne constitue pas une information utile pour les prédictions, donc on supprime la variable correspondante ID de l'ensemble d'apprentissage et de l'ensemble de test :
>risk_EA <- subset(risk_EA, select = -ID)
>risk_ET <- subset(risk_ET, select = -ID)

2. 1. Premier classifieur

On construit un **arbre de décision « rpart »** *tree1* afin de réaliser la prédiction de la variable RISK à partir du data frame risk_EA par la commande :
> tree1 <- rpart(RISK ~ ., risk_EA)

Afin d'évaluer cet arbre de décision on va l'appliquer à un ensemble de test, et comparer la prédiction faite par l'arbre avec la valeur «réelle» :
>pred.tree1 <- predict(tree1, risk_ET, type="class")

On construit ensuite la *matrice de confusion* avec en colonnes la classe prédite (objet pred.tree1) et en lignes la classe réelle (colonne RISK du data frame risk_ET) par la commande :
>table(pred.tree1, risk_ET$RISK)

Notre **classe** cible dite **« positive »** est BAD LOSS.

pred.tree1	bad loss	bad profit	good risk
bad loss	106	12	1
bad profit	161	1002	90
good risk	0	0	0

Taux de succès global = 80.8 %
Rappel = 50.7 %
Précision = 40 %

2. 2. Deuxième classifieur

On construit un **arbre de décision « party »** *tree2* par la commande :
> tree2 <- ctree(RISK~., risk_EA)

Afin d'évaluer cet arbre de décision on va l'appliquer à un ensemble de test, et comparer la prédiction faite par l'arbre avec la valeur «réelle» :
>pred.tree2 <- predict(tree2, risk_ET)

On construit ensuite la *matrice de confusion* avec en colonnes la classe prédite (objet pred.tree2) et en lignes la classe réelle (colonne RISK du data frame risk_ET) par la commande :
>table(risk_ET$RISK, pred.tree2)

Notre **classe** cible dite **« positive »** est BAD LOSS.

	bad loss	bad profit	good risk
bad loss	105	162	0
bad profit	12	1002	0
good risk	1	87	3

Taux de succès global = 80.9 %
Rappel = 29.7 %
Précision = 89 %

2. 3. Troisième classifieur

On construit un **arbre de décision « tree »** *tree3* par la commande :
> tree3 <- tree(RISK~., data=risk_EA)

Afin d'évaluer cet arbre de décision on va l'appliquer à un ensemble de test, et comparer la prédiction faite par l'arbre avec la valeur «réelle» :
>pred.tree3 <- predict(tree3, risk_ET, type="class")

On construit ensuite la *matrice de confusion* avec en colonnes la classe prédite (objet pred.tree3) et en lignes la classe réelle (colonne RISK du data frame risk_ET) par la commande :
>table(risk_ET$RISK, pred.tree3)

Notre **classe** cible dite **« positive »** est BAD LOSS.

	bad loss	bad profit	good risk
bad loss	106	161	0
bad profit	12	1002	0
good risk	1	90	0

Taux de succès global = 80.8 %
Rappel = 29.7 %
Précision = 89 %

2.4. Sélection du classifieur

Le deuxième classifieur parait plus performant que les deux autres selon les critères d'évaluation utilisés (taux de succès global, rappel, précision).

3. APPLICATION DU CLASSIFIEUR

On va alors appliquer l'arbre de décision sélectionné *tree2* pour prédire la classe des instances de l'ensemble de données Projet_New.csv qui ne contient pas la variable RISK.
>class.tree2 <- predict(tree2, risk_new, type="response")

On génère les probabilités pour chacune des trois classes :
>prob.tree2 <- predict(tree2, risk_new, type = "prob")

Transformons la liste de probabilités obtenue en vecteur pour pouvoir ensuite créer un data frame :
>prob.tree2 <- c(do.call("rbind",prob.tree2))

On construit un data frame **resultat** :
resultat <- data.frame(ID, class.tree2, prob.tree2[1:20],prob.tree2[21:40], rob.tree2[41:60])

Les statistiques des variables correspondant aux probabilités des classes BAD LOSS, BAD PROFIT et GOOD RISK dans le data frame resultat :
>summary(subset(resultat, resultat$Prediction=="bad loss")$"Bad Loss")

```
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.8833  0.8833  0.8833  0.8833  0.8833  0.8833
```

>summary(subset(resultat, resultat$Prediction=="bad profit")$"Bad Profit")

Min. 1st Qu. Median Mean 3rd Qu. Max.
0.4617 0.8367 0.8777 0.7893 0.8777 0.9000

>summary(subset(resultat, resultat$Prediction=="good risk")$"Good Risk")

Min. 1st Qu. Median Mean 3rd Qu. Max.
0.7981 0.7981 0.7981 0.7981 0.7981 0.7981

Résultats d'application :

ID	Prediction	Bad Loss	Bad Profit	Good Risk
1	213500	bad profit	0.07500000	0.90000000	0.02500000
2	213501	bad profit	0.08535179	0.87773933	0.03690888
3	213502	bad profit	0.16326531	0.83673469	0.00000000
4	213503	bad profit	0.16326531	0.83673469	0.00000000
5	213504	bad profit	0.08535179	0.87773933	0.03690888
6	213505	bad profit	0.08535179	0.87773933	0.03690888
7	213506	bad loss	0.88326848	0.08171206	0.03501946

ID	Prediction	Bad Loss	Bad Profit	Good Risk
8	213507	bad profit	0.08535179	0.87773933	0.03690888
9	213508	bad profit	0.16326531	0.83673469	0.00000000
10	213509	bad profit	0.33000000	0.46166667	0.20833333
11	213510	bad loss	0.88326848	0.08171206	0.03501946
12	213511	good risk	0.08653846	0.11538462	0.79807692
13	213512	bad profit	0.08535179	0.87773933	0.03690888
14	213513	bad profit	0.33000000	0.46166667	0.20833333
15	213514	bad profit	0.07500000	0.90000000	0.02500000
16	213515	bad profit	0.08535179	0.87773933	0.03690888
17	213516	good risk	0.08653846	0.11538462	0.79807692
18	213517	bad profit	0.08535179	0.87773933	0.03690888
19	213518	bad profit	0.33000000	0.46166667	0.20833333
20	213519	bad loss	0.88326848	0.08171206	0.03501946

4. CONCLUSION

Ayant évalué les performances de trois classifieurs selon différents points de vue en fonction de la classe positive BAD LOSS, on a sélectionné le deuxième classifieur, l'arbre de décision « party ».

Les probabilités associées aux prédictions faites nous indiquent que les prédictions pour BAD LOSS, par exemple, sont vérifiées au minimum à 88,33 % dans l'EA.

Ainsi, on a bien appris un modèle de prédiction de la classe des instances à partir d'exemples dont la classe est connue et l'appliqué pour classer de nouvelles instances de classe inconnue.