Université de Carthage
Ecole Supérieure de la Statistique et de l'Analyse de l'Information

Examen de Data Mining

3 ème année du cycle de formation d'ingénieurs

Durée de l'épreuve : 1 heure 30 - Documents non autorisés
Nombre de pages : 5 - Date de l'épreuve : 26 janvier 2021

On considère la base de données `data`. Chacune des 726 observations de cette base est décrite par la variable à prédire (`credit`), prenant les modalités 1 ou 0 selon que l'individu est classé bon ou mauvais client ainsi que les variables prédictives qui sont décrites dans le tableau à l'Annexe-Données.

Nous avons expliqué la variable `credit` en utilisant la méthode des arbres de décision. Les résultats obtenus sont présentés dans l'Annexe-ARBRE :

1) En utilisant l'arbre obtenu, classer l'individu ALPHA ayant les caractéristiques suivantes :
(`age=27 ; stab_emploi= 3 ; autre_garant=2 ; trav_etrang=2 ; montant=2000 ; etat_compte=2; duree_credit=15 ; part_mens=2 ; ressources=2 ; autres_credits=3`)
2) Indiquer la règle issue de cet arbre qui permet de classer cet individu.
3) On voudrait élaguer `arbre.full`, indiquer la procédure à suivre pour obtenir un arbre optimal à partir de cet arbre.
4) Compléter la commande `prune` par les paramètres adéquats afin d'obtenir l'arbre optimal puis indiquer le nombre de feuilles de cet arbre.

On a aussi effectué une régression logistique afin d'expliquer la variable `credit`. Pour cela nous avons effectué une sélection pas à pas *forward*. La première et la dernière étape de cette sélection pas à pas sont présentées à l'Annexe-RL.

5) Expliquer le principe de la sélection pas à pas *forward*.

6) Classer l'individu ALPHA à l'aide du modèle obtenu par la régression logistique.

7) Comparer les variables sélectionnées par l'arbre de décision `arbre.full` et celle de la sélection pas à pas *forward* de la régression logistique.

Nous avons enfin expliqué la variable `credit` en utilisant la méthode Random Forest. Les résultats obtenus sont présentés à l'Annexe-RF.

8) Expliquer le lien entre le choix de la valeur du paramètre `mtry` et le taux d'erreur réel du modèle donné par la Random Forest.

9) A partir des outputs de `modele_RF$confusion`, calculer les taux d'erreurs dans chaque classe ainsi que l'erreur OOB. Commenter ces résultats.

```
+ duree_credit    1   753.20 757.20
+ ressources      4   755.05 765.05
+ stab_emploi     4   755.75 765.75
+ autres_credits  2   764.10 770.10
+ age             1   772.37 776.37
+ trav_etrang     1   772.43 776.43
+ autre_garant    2   772.75 778.75
<none>                778.21 780.21
+ part_mens       3   777.28 785.28


## DERNIERE ETAPE
Step:  AIC=670.16
credit ~ etat_compte + montant + stab_emploi + ressources + trav_etrang +
    autres_credits + duree_credit + autre_garant

            Df Deviance    AIC
<none>           634.16 670.16
+ part_mens  3   628.69 670.69
+ age        1   633.24 671.24
> summary(pr.f.step)

Call:
glm(formula = credit ~ etat_compte + montant + stab_emploi +
    ressources + trav_etrang + autres_credits + duree_credit +
    autre_garant, family = "binomial")

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.8444   0.1970   0.4447   0.6598   1.7275

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)      4.120e-01  4.939e-01   0.834 0.404172
etat_compte2     6.924e-01  3.543e-01   1.954 0.050710 .
etat_compte3     1.435e+00  2.194e-01   6.539 6.19e-11 ***
montant         -8.494e-05  4.276e-05  -1.986 0.047014 *
stab_emploi2    -4.332e-01  4.415e-01  -0.981 0.326425
stab_emploi3    -1.139e-01  4.205e-01  -0.271 0.786401
stab_emploi4     8.495e-01  4.763e-01   1.784 0.074471 .
stab_emploi5     2.777e-01  4.411e-01   0.629 0.529033
ressources2      4.192e-02  2.909e-01   0.144 0.885421
ressources3      2.686e-01  4.166e-01   0.645 0.519108
ressources4      7.865e-01  5.046e-01   1.559 0.119097
ressources5      1.112e+00  3.079e-01   3.610 0.000306 ***
trav_etrang2     2.139e+00  1.217e+00   1.757 0.078945 .
autres_credits2 -1.153e-01  4.731e-01  -0.244 0.807529
autres_credits3  6.576e-01  2.684e-01   2.450 0.014279 *
duree_credit    -2.050e-02  1.019e-02  -2.012 0.044169 *
autre_garant2   -7.983e-01  5.084e-01  -1.570 0.116352
autre_garant3    8.418e-01  5.330e-01   1.579 0.114253
---
```

- ANNEXE-DONNÉES -

Nom de la variable	Description
age	Age du client
stab_emploi	Stabilité dans l'emploi
autre_garant	Autre garant
trav_etrang	Travailleur étranger
montant	Montant du crédit
etat_compte	Etat du compte
duree_credit	Durée du crédit
part_mens	Part des mensualités du revenu disponible
ressources	Valeur des ressources financière du client
autres_credits	Autres crédits en cours

- ANNEXE-ARBRE -

```
> summary(data)
 credit       age         etat_compte stab_emploi autre_garant trav_etrang
 0:165   Min.   :19.00   1:269       1: 41       1:669        1:704
 1:561   1st Qu.:27.00   2: 63       2:123       2: 24        2: 22
         Median :33.00   3:394       3:247       3: 33
         Mean   :35.62               4:128
         3rd Qu.:41.00               5:187
         Max.   :74.00
  duree_credit   part_mens ressources autres_credits    montant
 Min.   : 4.00   1:100     1:384      1: 97           Min.   :  250
 1st Qu.:12.00   2:177     2: 91      2: 35           1st Qu.: 1386
 Median :18.00   3:115     3: 55      3:594           Median : 2300
 Mean   :20.74   4:334     4: 42                      Mean   : 3308
 3rd Qu.:24.00             5:154                      3rd Qu.: 4028
 Max.   :72.00                                        Max.   :18424
> library(rpart)
> set.seed(1)
> arbre.full <- rpart(credit ~ ., data = data, method = "class")
> print(arbre.full)
n= 726

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

  1) root 726 165 1 (0.22727273 0.77272727)
    2) etat_compte=1 269 105 1 (0.39033457 0.60966543)
      4) montant>=12296.5 12   0 0 (1.00000000 0.00000000) *
      5) montant< 12296.5 257  93 1 (0.36186770 0.63813230)
       10) ressources=1,2 189  81 1 (0.42857143 0.57142857)
         20) duree_credit>=20.5 84  34 0 (0.59523810 0.40476190)
           40) autre_garant=2,3 8   0 0 (1.00000000 0.00000000) *
           41) autre_garant=1 76  34 0 (0.55263158 0.44736842)
             82) duree_credit>=46.5 13   2 0 (0.84615385 0.15384615) *
```

```
            83) duree_credit< 46.5 63  31 1 (0.49206349 0.50793651)
             166) montant< 2249 15   4 0 (0.73333333 0.26666667) *
             167) montant>=2249 48  20 1 (0.41666667 0.58333333)
               334) age< 28.5 16   6 0 (0.62500000 0.37500000) *
               335) age>=28.5 32  10 1 (0.31250000 0.68750000)
                 670) age>=41.5 9   3 0 (0.66666667 0.33333333) *
                 671) age< 41.5 23   4 1 (0.17391304 0.82608696) *
          21) duree_credit< 20.5 105  31 1 (0.29523810 0.70476190)
            42) autre_garant=1 84  30 1 (0.35714286 0.64285714)
              84) montant< 1961.5 51  24 1 (0.47058824 0.52941176)
               168) duree_credit>=11 27  11 0 (0.59259259 0.40740741)
                 336) stab_emploi=2,3,5 20   5 0 (0.75000000 0.25000000) *
                 337) stab_emploi=1,4 7   1 1 (0.14285714 0.85714286) *
               169) duree_credit< 11 24   8 1 (0.33333333 0.66666667) *
              85) montant>=1961.5 33   6 1 (0.18181818 0.81818182) *
            43) autre_garant=2,3 21   1 1 (0.04761905 0.95238095) *
        11) ressources=3,4,5 68  12 1 (0.17647059 0.82352941) *
     3) etat_compte=2,3 457  60 1 (0.13129103 0.86870897) *
> printcp(arbre.full)

Classification tree:
rpart(formula = credit ~ ., data = data, method = "class")

Variables actually used in tree construction:
[1] age          autre_garant duree_credit etat_compte  montant      ressources
[7] stab_emploi

Root node error: 165/726 = 0.22727

n= 726

        CP nsplit rel error  xerror     xstd
1 0.036364      0   1.00000 1.00000 0.068434
2 0.016162      4   0.83030 0.91515 0.066278
3 0.015152      9   0.73939 0.89091 0.065621
4 0.010000     13   0.67879 0.96364 0.067536
> arbre.full.prune<-prune(?,?)
```

- ANNEXE-RL -

```
> modele_simple <- glm(credit ~ 1, "binomial")
> pr.f.step<-step(modele_simple, scope = ~ age+etat_compte+stab_emploi+autre_garant
+trav_etrang+duree_credit+part_mens+ressources+autres_credits+montant, dir="forward")
Start:  AIC=780.21
credit ~ 1

                Df Deviance    AIC
+ etat_compte    2   710.61 716.61
+ montant        1   752.40 756.40
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 778.21  on 725  degrees of freedom
Residual deviance: 634.16  on 708  degrees of freedom
AIC: 670.16

Number of Fisher Scoring iterations: 6
```

- ANNEXE-RF -

```
> library(randomForest)
> # mtry : Number of variables randomly sampled as candidates at each split.
> modele_RF <- randomForest(credit~.,data=data, mtry= 5,ntree=500)
> # modele_RF$confusion : the confusion matrix of the prediction (based on OOB data).
> modele_RF$confusion
   0   1
0 52 113
1 33 528
```