Projet Scoring 21 novembre 2022

ROBIN Alexandre

2022-11-19

Contents

Les données

Nous disposons d'un jeu de données pour 100 individus comprenant 5 variables : "cash", "flow", "saving", "consume", et "risk". Cette dernière variable est binaire et indique si le client en question appartient à la classe de risque 1 ou à la classe de risque 2.

On note : $x_1, ..., x_n \in X$ les observations réparies en groupes. $G = \{1, 2\}$ les groupes et $y_i \in \{1, 2\}$ indique le groupe de l'observation x_i.

```
## Extrait du jeu de données :
```

```
##   cash flow saving consume risk
## 1  7.0  3.2    4.7     1.4    1
## 2  6.4  3.2    4.5     1.5    1
## 3  6.9  3.1    4.9     1.5    1
## 4  5.5  2.3    4.0     1.3    1
## 5  6.5  2.8    4.6     1.5    1
## 6  5.7  2.8    4.5     1.3    1
```

(1) Calcul de lerreur en LOOCV d'un classifieur Gaussien hétéroscédastique

Le principe de la méthode Leave-One-Out est le suivant : on enlève tour à tour un point de D l'ensemble des informations dont on dispose ($D = \{(x_i, y_i); i = 1, ...100\}$), et on teste sur ce point le classifieur construit sur les autres points. L'erreur LOO est alors le taux de mal classés. Nous utilisons ici un classifieur Gaussien hétéroscadastique.

On note :

$$\in_{LOO} = \frac{1}{n} \sum_{i=1}^{k} \mathbb{1}_{(\hat{y}_i \neq y_i)}$$

avec $\mathbb{1} = 1$ quand $\hat{y}_i \neq y_i$, 0 sinon.

```
error = NULL

for (j in 1:length(client$cash)) {
  tmp = client[-j,]
  new = client[j,]
  learn = mixmodLearn(data=tmp[,1:4],knownLabels=as.factor(tmp[,5]),
                      models=mixmodGaussianModel(listModels =
                                                   c("Gaussian_pk_Lk_Ck")),
                                                   criterion=c('BIC'))
  pred = mixmodPredict(data=new[,1:4],classificationRule=learn['bestResult'])
  if (pred@partition == new$risk){
    error[j] = 0
  } else if (pred@partition != new$risk) {
    error[j] = 1
  }
}
error_loocv = sum(error)/length(client$cash)
cat('Erreur leave-one-out :',error_loocv,'\n')
```

```
## Erreur leave-one-out : 0.04
```

4 de nos 100 individus ont été mal classés selon la technique LOO. Nous avons isolé 1 observation parmi 100 cent fois, et en créant un modèle apprenant des 99 individus, le classement a été mauvais à 4 reprises sur 100.

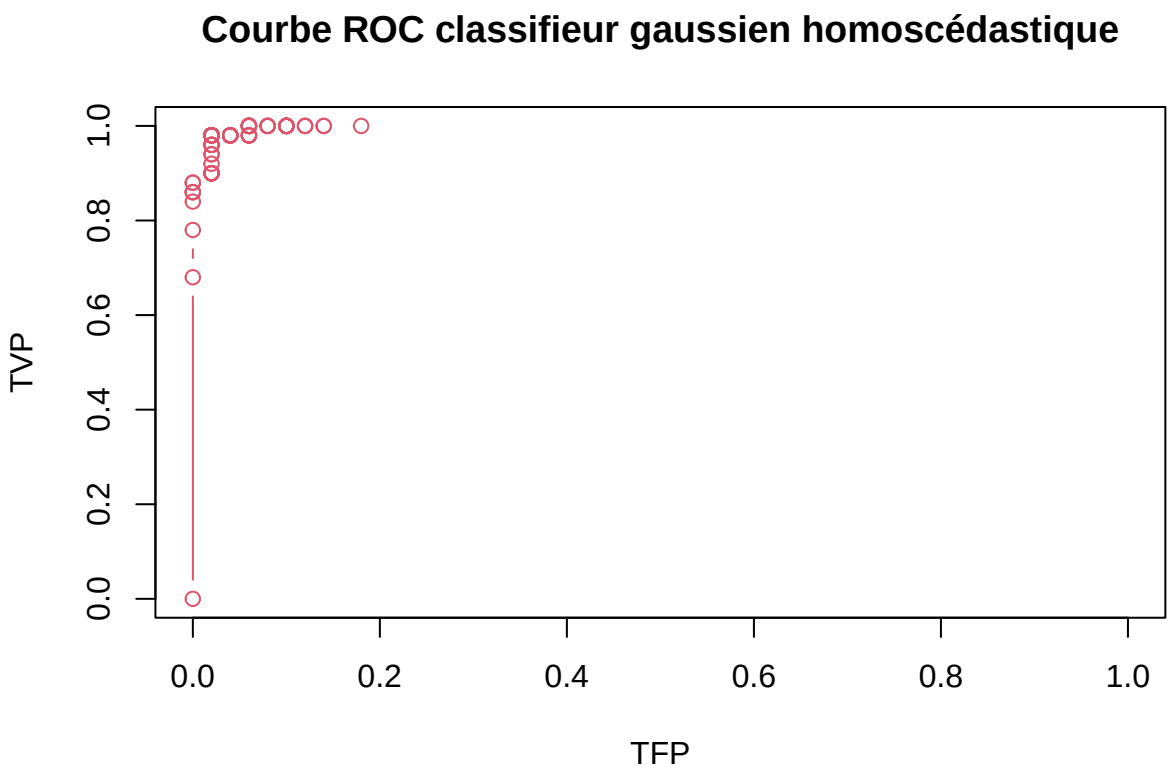
(2) Comparer la courbe ROC de 3 classifieurs

La courbe ROC permet d'évaluer la précision de notre prédiciton. Un bon classifieur est un classifieur dont la courbe ROC est concave et éloignée de la première bissectrice. Nous préfèrerons un modèle avec la plus grande aire sous la courbe ROC.

(i) Gaussien homoscédastique

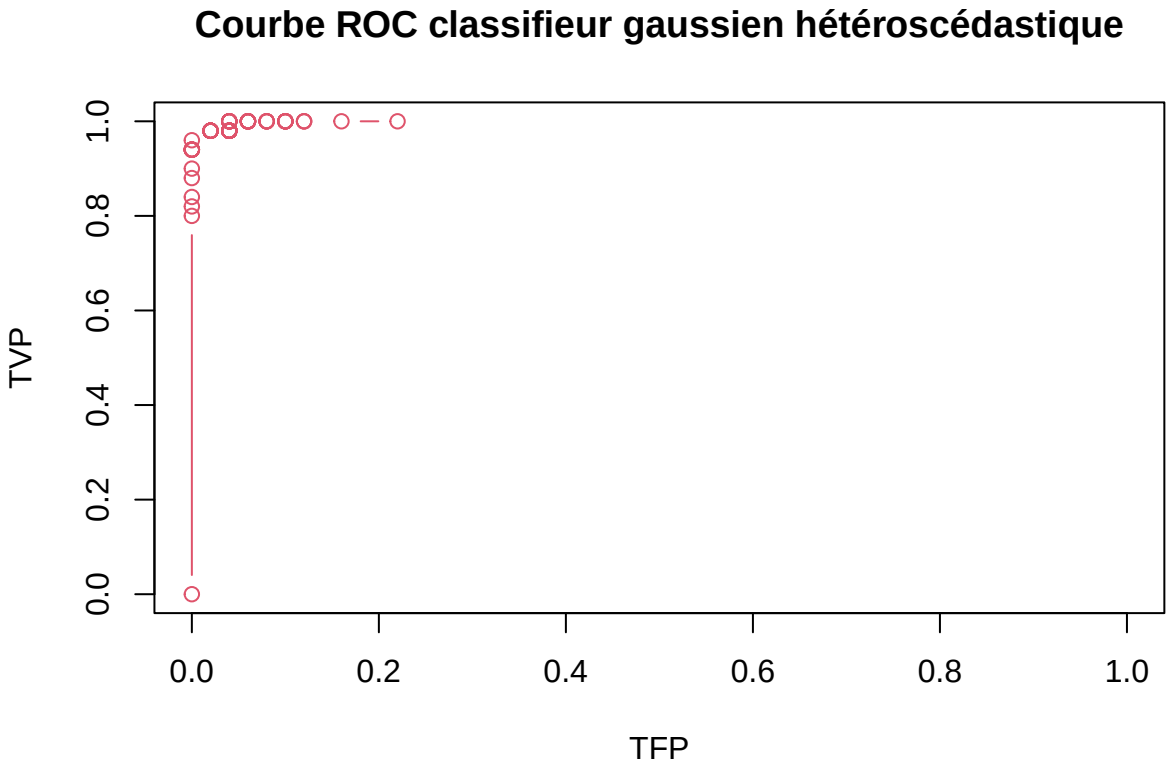
Nous allons comparer chaque résultats par seuil avec client$risk

- les faux positifs (FP) ; ils sont dans la classe 1 mais sont affectés à la classe 2;
- les vrais positifs (VP) ; ils sont dans la classe 2 et sont affectés à la classe 2;
- les faux négatifs (FN) ; ils sont dans la classe 2 mais sont affectés à la classe 1;
- les vrais négatifs (VN) ; ils sont dans la classe 1 et sont affectés à la classe 1 et;
- le taux de faux positifs $TFP = FP/N$ avec $N = FP + VN$ et
- le taux de vrais positifs $TVP = VP/P$ avec $P = VP + FN$.



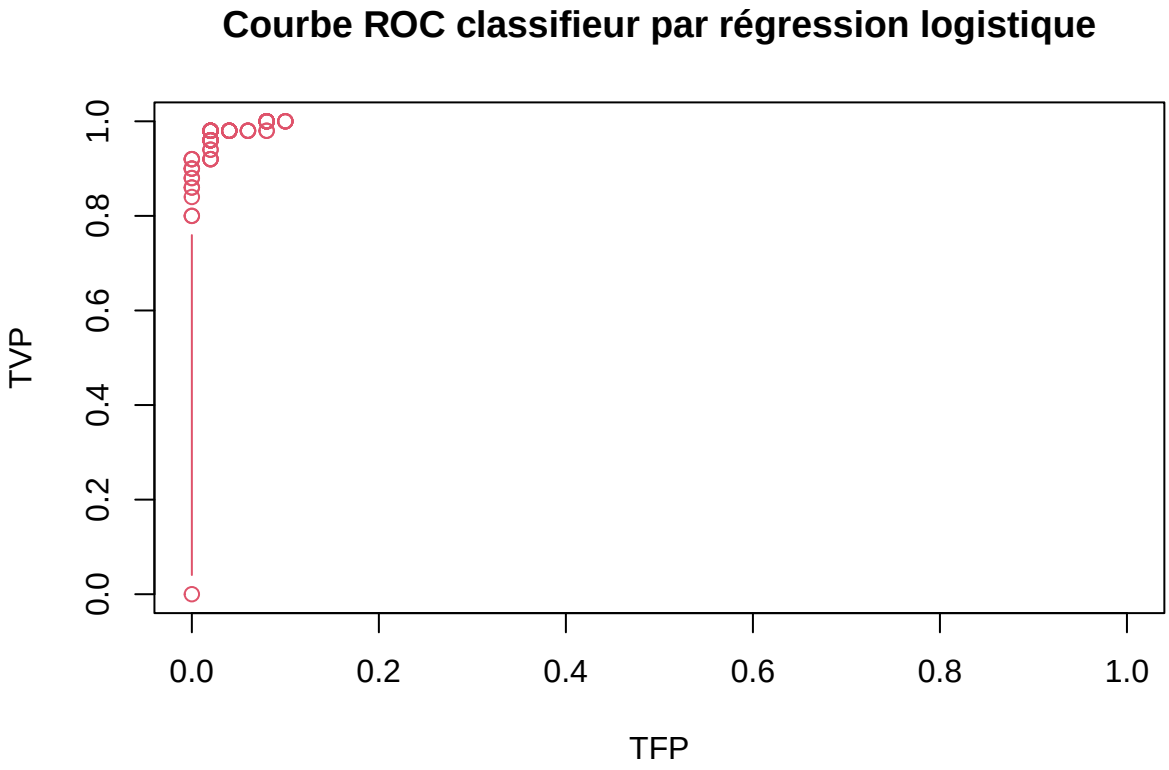

```
## Aire sous la courbe ROC du classifieur homoscédastique : 0.1682
```

(ii) Gaussien hétéroscédastique




```
## Aire sous la courbe ROC du classifieur hétéroscédastique : 0.2094
```

(iii) Régression logistique




```
## Aire sous la courbe ROC du classifieur par régression logistique : 0.088
```

Conclusion

Le modèle qui classe le mieux nos observations parmi ces 3 est le classifieur gaussien hétéroscédastique puisqu'il a la plus grande aire sous la courbe ROC ($0.2094 > 0.1682 > 0.088$).

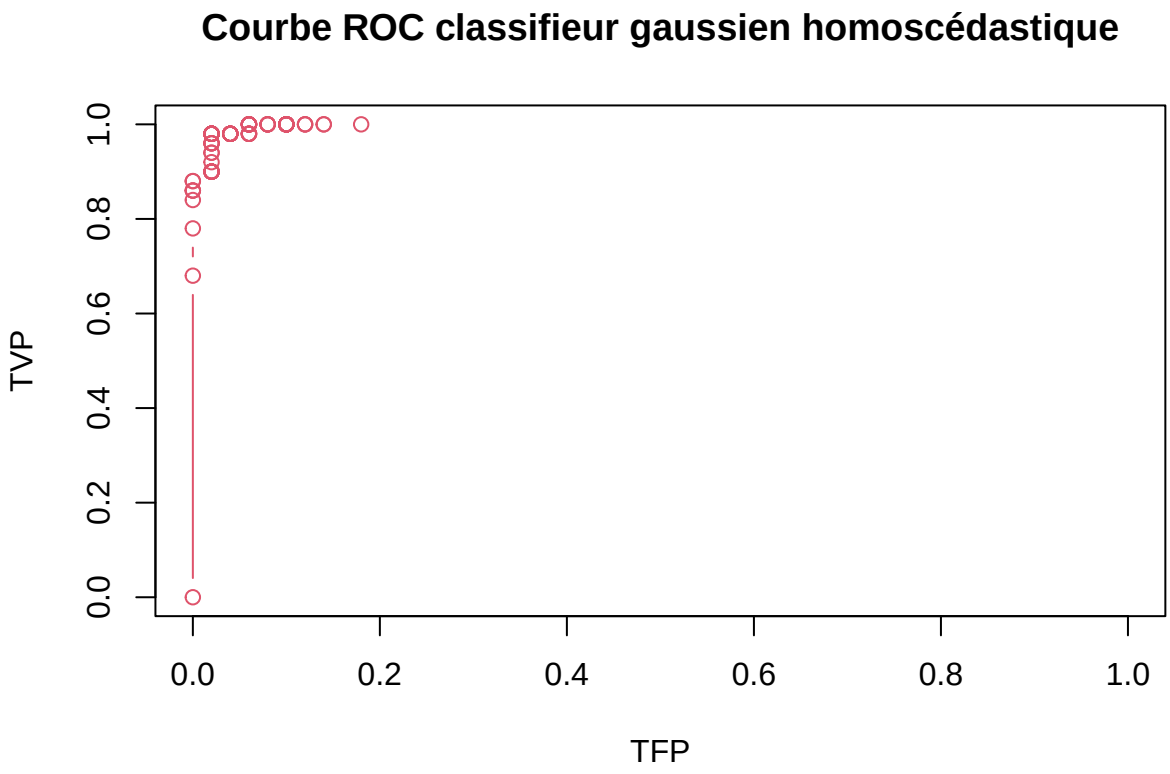
Annexes

(i) Gaussien homoscédastique

```
learn_homo = mixmodLearn(data=client[,1:4],knownLabels=as.factor(client[,5]),models=mixmodGaussianModel
pred_homo = mixmodPredict(data=client[,1:4],classificationRule=learn_homo['bestResult'])
proba_homo = pred_homo@proba
```

```
#estimer la classe en variant le seuil
classe_estimee_homo = NULL
classe_estimee_homo = as.data.frame(classe_estimee_homo)
for (i in 1 : length(client$cash)){
  for (j in 1 : 100) {
   if (proba_homo[i,2] > 0.01*j) {
      classe_estimee_homo[i,j] = 2
    } else if (proba_homo[i,2] <= 0.01*j) {
      classe_estimee_homo[i,j] = 1
    }
  }
}
TFP = NULL
TVP = NULL

for (j in 1 : 100){
  FP = VP = FN = VN = 0
  for (i in 1:length(classe_estimee_homo)) {
    if (classe_estimee_homo[i,j] == 1 && classe_estimee_homo[i,j] == client[i,5]){
      VN = VN + 1
    } else if (classe_estimee_homo[i,j] == 2 && classe_estimee_homo[i,j] == client[i,5]){
      VP = VP + 1
    } else if (classe_estimee_homo[i,j] == 1 && classe_estimee_homo[i,j] != client[i,5]){
      FN = FN + 1
    } else if (classe_estimee_homo[i,j] == 2 && classe_estimee_homo[i,j] != client[i,5]){
      FP = FP + 1
    }
  }
  TFP[j] = FP / (FP + VN)
  TVP[j] = VP / (VP + FN)
}
```

```
plot(TFP,TVP, type='b', col=2, main = "Courbe ROC classifieur gaussien homoscédastique", ylim=c(0,1), x
```




```
id <- order(TFP)
AUC_homo <- sum(diff(TFP[id])*rollmean(TVP[id],2)) # area under ROC curve
cat('Aire sous la courbe ROC du classifieur homoscédastique :',AUC_homo,'\n')
```

```
## Aire sous la courbe ROC du classifieur homoscédastique : 0.1682
```

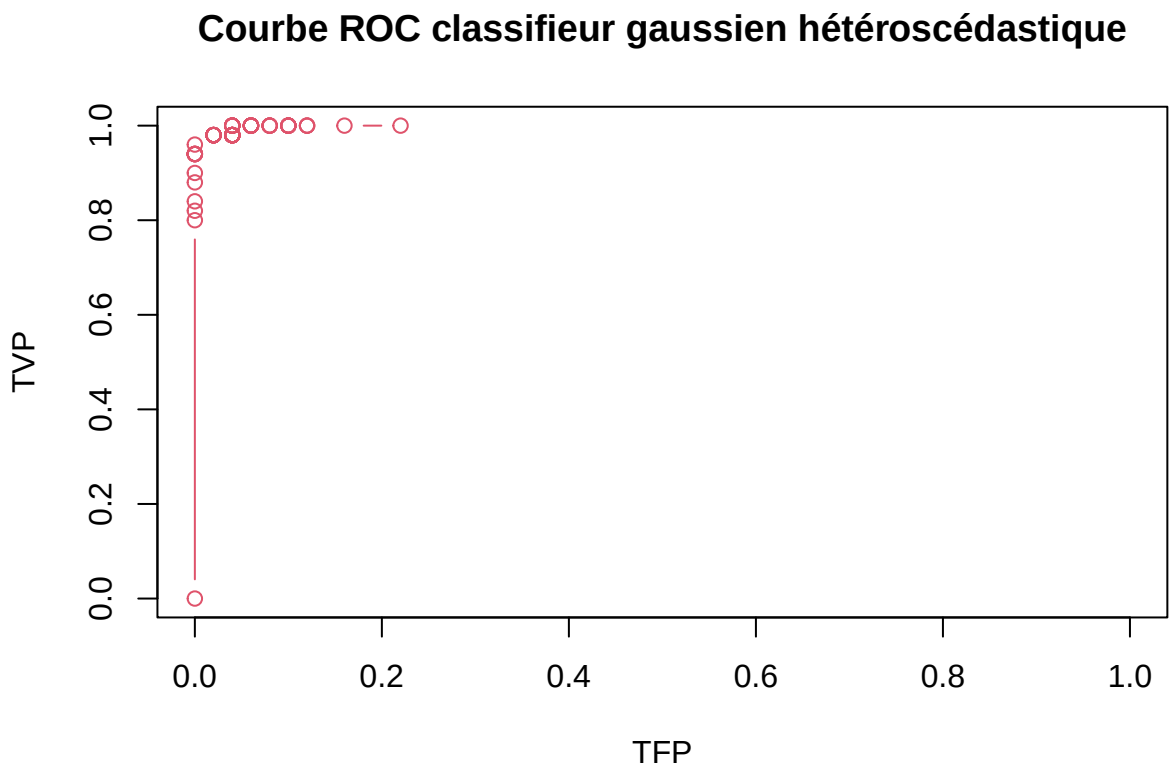
(ii) Gaussien hétéroscédastique

```
learn_hetero = mixmodLearn(data=client[,1:4],knownLabels=as.factor(client[,5]),models=mixmodGaussianMod
pred_hetero = mixmodPredict(data=client[,1:4],classificationRule=learn_hetero['bestResult'])
proba_hetero = pred_hetero@proba
```

```
#estimer la classe en variant le seuil
classe_estimee_hetero = NULL
classe_estimee_hetero = as.data.frame(classe_estimee_hetero)
for (i in 1 : length(client$cash)){
  for (j in 1 : 100) {
    if (proba_hetero[i,2] > 0.01*j) {
      classe_estimee_hetero[i,j] = 2
    } else if (proba_hetero[i,2] <= 0.01*j) {
      classe_estimee_hetero[i,j] = 1
    }
  }
```

```
}
TFP = NULL
TVP = NULL

for (j in 1 : 100){
  FP = VP = FN = VN = 0
  for (i in 1:length(classe_estimee_hetero)) {
    if (classe_estimee_hetero[i,j] == client[i,5] && classe_estimee_hetero[i,j] == 1){
      VN = VN + 1
    } else if (classe_estimee_hetero[i,j] == client[i,5] && classe_estimee_hetero[i,j] == 2){
      VP = VP + 1
    } else if (classe_estimee_hetero[i,j] != client[i,5] && classe_estimee_hetero[i,j] == 1){
      FN = FN + 1
    } else if (classe_estimee_hetero[i,j] != client[i,5] && classe_estimee_hetero[i,j] == 2){
      FP = FP + 1
    }
  }
  TFP[j] = FP / (FP + VN)
  TVP[j] = VP / (VP + FN)
}
```

```
plot(TFP,TVP, type='b', col=2, main = "Courbe ROC classifieur gaussien hétéroscédastique", ylim=c(0,1),
```



```r
id <- order(TFP)
AUC_hetero <- sum(diff(TFP[id])*rollmean(TVP[id],2)) # area under ROC curve
cat('Aire sous la courbe ROC du classifieur hétéroscédastique :',AUC_hetero,'\n')
```

```
## Aire sous la courbe ROC du classifieur hétéroscédastique : 0.2094
```

(iii) Régression logistique

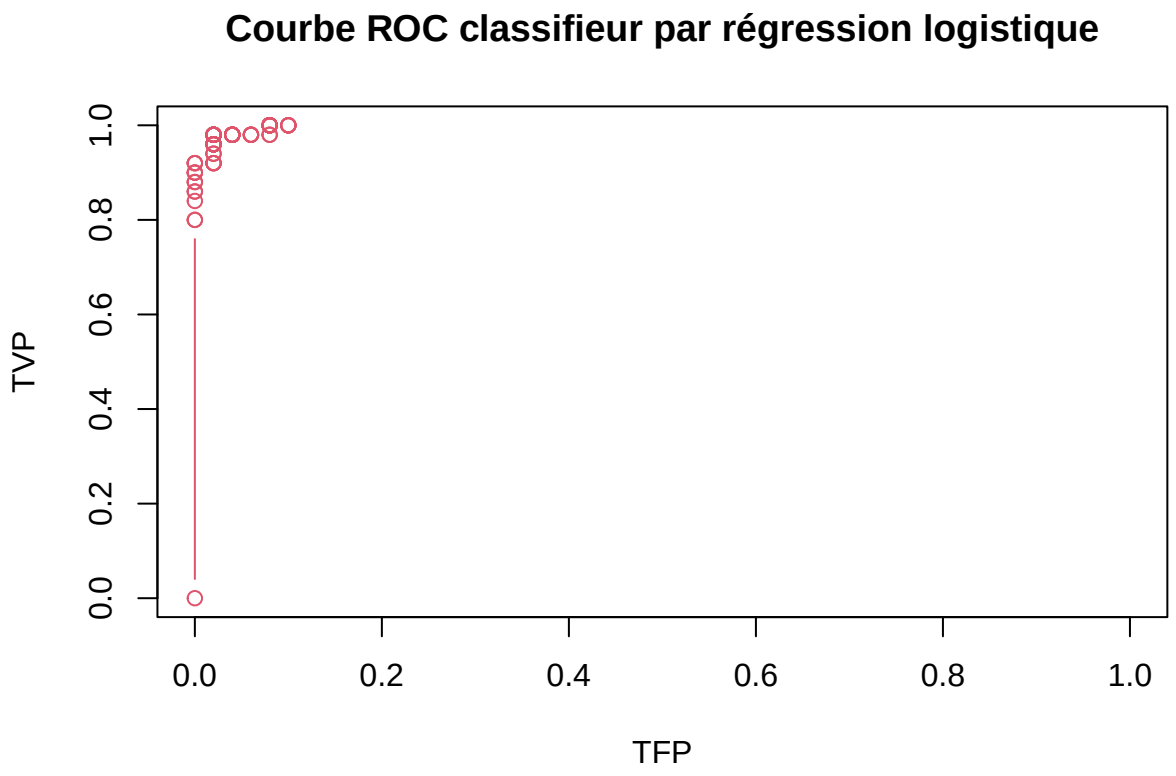
```r
rm(list=ls(all=TRUE))
client <- read.table(file='http://alexandrelourme.free.fr/scoring_data_client',sep=',',dec='.',header=TR
rule=glm(as.factor(client$risk)~., data = client[,1:4], family=binomial(link='logit'))
score = predict(rule, client[,1:4])
table(score<0, client$risk)
```

```
##
##          0  1
##   FALSE  1 49
##   TRUE  49  1
```

```r
# création des scores tels que t2 = 1 - t1
t1 = NULL
t2 = NULL

for (i in 1:length(score)){
  t1[i] = exp(score[i]) / (1 + exp(score[i]))
  t1[i] = 1 - t1[i]
  t2[i] = 1 - t1[i]
}

classe_estimee_rl = NULL
classe_estimee_rl = as.data.frame(classe_estimee_rl)
for (i in 1 : length(score)){
  for (j in 1 : 100) {
    if (t2[i] > 0.01*j) {
      classe_estimee_rl[i,j] = 2
    } else if (t2[i] <= 0.01*j) {
      classe_estimee_rl[i,j] = 1
    }
  }
}

TFP = NULL
TVP = NULL
client$risk = client[,5]+1
for (j in 1 : 100){
  FP = VP = FN = VN = 0
  for (i in 1:100) {
    if (classe_estimee_rl[i,j] == client[i,5] && classe_estimee_rl[i,j] == 1){
      VN = VN + 1
    } else if (classe_estimee_rl[i,j] == client[i,5] && classe_estimee_rl[i,j] == 2){
```

```
      VP = VP + 1
    } else if (classe_estimee_rl[i,j] != client[i,5] && classe_estimee_rl[i,j] == 1){
      FN = FN + 1
    } else if (classe_estimee_rl[i,j] != client[i,5] && classe_estimee_rl[i,j] == 2){
      FP = FP + 1
    }
  }
  TFP[j] = FP / (FP + VN)
  TVP[j] = VP / (VP + FN)
}
```

```
plot(TFP,TVP, type='b', col=2, main = "Courbe ROC classifieur par régression logistique", ylim=c(0,1), x
```




```
id <- order(TFP)
AUC_rl <- sum(diff(TFP[id])*rollmean(TVP[id],2)) # area under ROC curve
cat('Aire sous la courbe ROC du classifieur par régression logistique :',AUC_rl,'\n')
```

```
## Aire sous la courbe ROC du classifieur par régression logistique : 0.088
```