

# Data\_mining

Axelle Barou & Amélie Picard & Tanguy Delzant

06/01/2021

## Rapport

#Présentation de la problématique et du jeu de données

Nous avons choisi un jeu de données sur Kaggle : <https://www.kaggle.com/volodymyrgavrysh/bank-marketing-campaigns-dataset>

Ce dernier contient 41188 observations et 21 variables.

Il s'agit d'un jeu de données sur les campagnes marketing des banques portugaises. Ces campagnes, qui se basaient sur des appels téléphoniques, permettaient d'offrir aux clients de déposer un dépôt à terme, c'est-à-dire de prêter de l'argent à une banque sur une durée fixe, avec un taux d'intérêt connu à l'avance ou variable.

Notre problématique sera donc d'essayer de prédire, en fonction des données du client, des données récoltées au cours de l'entretien téléphonique de la campagne et d'autres attributs sociaux et économiques si ces clients seront enclins à souscrire un dépôt à terme.

```
#chargement des donnees
#setwd("C:/Users/ameli/Desktop/R/data_mining")

#le package neuralnet permet de réaliser des reseau de neurones :
library(neuralnet)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:neuralnet':
##
##      compute

## The following objects are masked from 'package:stats':
##
##      filter, lag

## The following objects are masked from 'package:base':
##
##      intersect, setdiff, setequal, union

library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.0-2
```

```
library(caTools)
```

```
data<-read.csv2("bank.csv")
```

```
#Transformation de y en variable binaire :
data$y = recode(data$y, "no" = 0, "yes" = 1)
```

```
#Réalisation du réseau de neurones
```

Nous allons voir le protocole expérimentale, puis nous allons le décrire plus en détail le déroulement de l'analyse réalisé ainsi que les conclusions.

```
##Protocole expérimental
```

```
##Transformation des données
```

Avant de réaliser les réseaux de neurones, nous devons transformer les données. Pour les variables quantitatives, nous allons les normaliser pour qu'elle ai la même importance. Pour les variables qualitatives il faut les convertir en ordonné en leur donnant une valeur numeric de 1 à k ou k est le nombre de facteur. Avant de paramétrer le réseau de neuronne, il faut normaliser les données, les variables explicatives, dans notre cas toutes les variables du jeu de données sauf "y" la variables cible.

```
#on récupère les données de base
datann<-data
dataquali<-data[,c(2,3,4,5,7,8,9,11,16)]

#quali en quali numeric
#fonction qui change pour une variable les noms des modalité par un nombre :
encoding<-function(var) {
  cpt=0
  for (i in levels(as.factor(var))) {
    for (j in length(var)) {
      var[var==i]<-cpt
    }
    cpt=cpt+1
  }
  return(as.numeric(var))
}
dataquali<-lapply(dataquali, encoding)

#donnée avec toute les variables numérique :
datann[,c(2,3,4,5,7,8,9,11,16)]<-dataquali

#normalisation des variables
scaleddata<-scale(datann[,-17])
datann<-data.frame(scaleddata,datann[17])
```

De plus, pour la vérification et la validation du meilleur reseau de neurone, nous découpons nos données en trois échantillons. Nous réalisons un premier découpage dans les données pour obtenir un échantillon d'apprentissage qui est constitué de 80% de la base et un échantillon test qui est constitué de 20% de la base. Puis nous re-découpons l'échantillon d'apprentissage en deux, pour récupérer un échantillon de validation qui est constitué de 30% de l'échantillon d'apprentissage.

```
#Echantillonnage des données
ech = sort(sample(nrow(datann), nrow(datann)*.8))

train = datann[ech, ]
test = datann[-ech, ]

#Definition variables explicatives/variable cible de test
```

```

Xtest = test[, -17]
ytest = test[, 17]

#Découpage de l'échantillon d'apprentissage en ech de validation :
#Echantillonnage des données
ech2 = sort(sample(nrow(train), nrow(train)*.7))

valid = train[-ech2, ]
train = train[ech2, ]

#Definition variables explicatives/variable cible
Xtrain = train[, -17]
ytrain = train[, 17]
Xvalid = valid[, -17]
yvalid = valid[, 17]

```

##Paramétrage du réseau de neurone

Nous allons réaliser un réseau de neurones à l'aide de la fonction “neuralnet” qui permet de réaliser tout type de réseau de neurones. Nous allons paramétrer cette méthode grâce à de nombreux paramètres : - hidden : spécifie le nombre de neurones dans la couche caché - err.fct : fonction utilisé pour déterminer le calcul de l'erreur (ce : entropie croisée, sse : somme des erreurs au carré) - linear.output = FALSE : Ne pas réaliser de regression linéaire - algorithm : contient une chaine définissant le type d'algorithme, par défaut vaut “rprop+” qui signifie rétropropagation résiliente avec retour de poids - act.fct : permet de lisser le résultat du produit croisé des neurones et des poids, par défaut “logistic” qui se rapporte à la fonction logistique.

Cette fonction utilise d'autre paramètres comme : - startweights : contient la valeurs de départ des poids (par défaut une initialisation aléatoire) - rep : Nombre d'entraînement du réseau (pas utiliser pour éviter le sur-apprentissage)

Dans le cas d'une variable y catégorielle binaire, nous choisissons comme fonction d'erreur l'entropie croisée.

Premier modèle :

```

nn=neuralnet(ytrain~., data = Xtrain, hidden=10, err.fct="ce",linear.output = FALSE)
attributes(nn)

```

```

## $names
## [1] "call"                "response"           "covariate"
## [4] "model.list"          "err.fct"             "act.fct"
## [7] "linear.output"       "data"               "exclude"
## [10] "net.result"          "weights"            "generalized.weights"
## [13] "startweights"        "result.matrix"
##
## $class
## [1] "nn"

nn$result.matrix

```

```

##                                [,1]
## error                        3.078520e+02
## reached.threshold            9.673217e-03
## steps                        2.253400e+04
## Intercept.to.1layhid1       -3.480255e+01
## age.to.1layhid1              2.295956e+01
## job.to.1layhid1             -2.940216e+01
## marital.to.1layhid1         1.074378e+02

```

```

## education.to.1layhid1 4.798784e+00
## default.to.1layhid1 -2.249173e+01
## balance.to.1layhid1 2.149169e+01
## housing.to.1layhid1 -3.655356e+01
## loan.to.1layhid1 -3.149917e+01
## contact.to.1layhid1 2.109583e+01
## day.to.1layhid1 2.947933e+00
## month.to.1layhid1 -1.380521e+01
## duration.to.1layhid1 1.254669e+02
## campaign.to.1layhid1 -5.590069e+01
## pdays.to.1layhid1 1.016081e+01
## previous.to.1layhid1 1.158116e+02
## poutcome.to.1layhid1 5.957877e+01
## Intercept.to.1layhid2 3.565246e+01
## age.to.1layhid2 1.000722e+02
## job.to.1layhid2 2.165464e+01
## marital.to.1layhid2 9.704625e+01
## education.to.1layhid2 5.309267e+00
## default.to.1layhid2 -3.210832e+00
## balance.to.1layhid2 3.553859e+01
## housing.to.1layhid2 5.211941e+01
## loan.to.1layhid2 3.801682e+01
## contact.to.1layhid2 1.720934e+01
## day.to.1layhid2 -2.069541e+01
## month.to.1layhid2 7.904101e+00
## duration.to.1layhid2 -1.262771e+02
## campaign.to.1layhid2 2.649451e+01
## pdays.to.1layhid2 -4.105018e+01
## previous.to.1layhid2 6.787993e+01
## poutcome.to.1layhid2 -4.475789e+00
## Intercept.to.1layhid3 -4.204467e+01
## age.to.1layhid3 -1.430348e+01
## job.to.1layhid3 7.733736e+00
## marital.to.1layhid3 -1.798746e+01
## education.to.1layhid3 -2.778296e+00
## default.to.1layhid3 2.837841e+01
## balance.to.1layhid3 4.330298e+01
## housing.to.1layhid3 9.834120e+00
## loan.to.1layhid3 9.945742e+00
## contact.to.1layhid3 3.432378e+01
## day.to.1layhid3 -7.430213e+00
## month.to.1layhid3 7.677598e+00
## duration.to.1layhid3 -1.350633e+02
## campaign.to.1layhid3 8.159580e+01
## pdays.to.1layhid3 1.238030e+01
## previous.to.1layhid3 -3.184778e+00
## poutcome.to.1layhid3 -1.855263e+01
## Intercept.to.1layhid4 9.861878e+00
## age.to.1layhid4 4.265444e+01
## job.to.1layhid4 -4.659241e+01
## marital.to.1layhid4 4.493561e+01
## education.to.1layhid4 -3.871765e+01
## default.to.1layhid4 -1.618845e+01
## balance.to.1layhid4 1.434099e+01

```

```

## housing.to.1layhid4      -4.684226e+01
## loan.to.1layhid4         -1.609591e+02
## contact.to.1layhid4      -6.016378e-01
## day.to.1layhid4          8.933127e+01
## month.to.1layhid4         3.247550e+01
## duration.to.1layhid4     -1.132389e+01
## campaign.to.1layhid4     1.048670e+02
## pdays.to.1layhid4        1.265018e+01
## previous.to.1layhid4     1.528459e+01
## poutcome.to.1layhid4     5.660980e+01
## Intercept.to.1layhid5    4.133733e-01
## age.to.1layhid5          3.243959e+01
## job.to.1layhid5          -1.505959e+01
## marital.to.1layhid5      1.485876e+02
## education.to.1layhid5    -5.310260e+01
## default.to.1layhid5      -4.524210e+01
## balance.to.1layhid5      -2.742683e+00
## housing.to.1layhid5      -2.121372e+00
## loan.to.1layhid5         4.700715e+00
## contact.to.1layhid5      -2.683075e+01
## day.to.1layhid5          7.664267e+01
## month.to.1layhid5        -2.191478e+01
## duration.to.1layhid5     9.083408e+01
## campaign.to.1layhid5     5.796776e+01
## pdays.to.1layhid5        -3.063196e+00
## previous.to.1layhid5     -5.097387e-01
## poutcome.to.1layhid5     2.445029e+00
## Intercept.to.1layhid6    2.056531e+01
## age.to.1layhid6          -2.438840e+01
## job.to.1layhid6          1.450424e+02
## marital.to.1layhid6      -5.172984e+00
## education.to.1layhid6    7.423944e+01
## default.to.1layhid6      -3.987934e+01
## balance.to.1layhid6      -1.554390e+01
## housing.to.1layhid6      -2.041637e+01
## loan.to.1layhid6         5.249259e+01
## contact.to.1layhid6      -6.820390e+01
## day.to.1layhid6          1.388811e+01
## month.to.1layhid6        -1.050924e+02
## duration.to.1layhid6     6.382614e+01
## campaign.to.1layhid6    -1.049221e+01
## pdays.to.1layhid6        -9.059126e+00
## previous.to.1layhid6     -5.618946e+01
## poutcome.to.1layhid6     -7.326031e+01
## Intercept.to.1layhid7    5.239230e+01
## age.to.1layhid7          5.290039e+01
## job.to.1layhid7          -1.344640e+01
## marital.to.1layhid7      -3.193677e+00
## education.to.1layhid7    -3.121724e+01
## default.to.1layhid7      -7.190914e+00
## balance.to.1layhid7      -4.583682e+00
## housing.to.1layhid7      -3.952299e+01
## loan.to.1layhid7         -4.627710e+01
## contact.to.1layhid7      -7.077869e+00

```

```

## day.to.1layhid7      3.615153e+01
## month.to.1layhid7    4.917966e+01
## duration.to.1layhid7 -2.234110e+01
## campaign.to.1layhid7 2.182087e+01
## pdays.to.1layhid7    3.697003e+01
## previous.to.1layhid7 -1.245696e+01
## poutcome.to.1layhid7 -2.654268e+01
## Intercept.to.1layhid8 -3.314241e+01
## age.to.1layhid8      6.436639e+01
## job.to.1layhid8      -1.016020e+02
## marital.to.1layhid8  -1.256900e+01
## education.to.1layhid8 -9.961317e+00
## default.to.1layhid8  -2.266020e+01
## balance.to.1layhid8   5.206390e+01
## housing.to.1layhid8   1.565258e+02
## loan.to.1layhid8      8.135157e+01
## contact.to.1layhid8   5.173071e+01
## day.to.1layhid8      3.817257e+01
## month.to.1layhid8    -5.367505e+01
## duration.to.1layhid8  4.291662e+01
## campaign.to.1layhid8 -7.290438e+01
## pdays.to.1layhid8     4.005975e+00
## previous.to.1layhid8 -1.134125e+02
## poutcome.to.1layhid8  7.051237e+01
## Intercept.to.1layhid9  6.182667e-01
## age.to.1layhid9      -1.609121e+00
## job.to.1layhid9      -1.427038e+00
## marital.to.1layhid9  -1.968421e+00
## education.to.1layhid9 1.641744e+00
## default.to.1layhid9  -6.243004e+00
## balance.to.1layhid9   1.551286e-01
## housing.to.1layhid9   4.020245e-01
## loan.to.1layhid9      -2.030789e-01
## contact.to.1layhid9   1.709863e+00
## day.to.1layhid9      1.936191e+00
## month.to.1layhid9    -2.239389e+00
## duration.to.1layhid9  3.155774e+00
## campaign.to.1layhid9  3.349251e-02
## pdays.to.1layhid9    -8.884095e-01
## previous.to.1layhid9 -6.972679e+00
## poutcome.to.1layhid9  7.728509e-01
## Intercept.to.1layhid10 -3.471373e+01
## age.to.1layhid10     -4.030681e+01
## job.to.1layhid10      6.230538e+01
## marital.to.1layhid10  3.266923e+01
## education.to.1layhid10 -3.903447e+00
## default.to.1layhid10  -6.498548e+00
## balance.to.1layhid10  -2.859671e+02
## housing.to.1layhid10  1.511434e+01
## loan.to.1layhid10     1.257730e+01
## contact.to.1layhid10  1.043909e+02
## day.to.1layhid10     4.078798e+01
## month.to.1layhid10    -7.156833e+01
## duration.to.1layhid10 -1.003785e+02

```

```
## campaign.to.1layhid10 -1.056451e+02
## pdays.to.1layhid10 2.039813e+01
## previous.to.1layhid10 -2.297472e+02
## poutcome.to.1layhid10 6.384360e+00
## Intercept.to.ytrain 2.500930e-01
## 1layhid1.to.ytrain 3.462923e+00
## 1layhid2.to.ytrain -4.003738e+00
## 1layhid3.to.ytrain -5.119591e+00
## 1layhid4.to.ytrain 4.615915e+00
## 1layhid5.to.ytrain -3.970115e+00
## 1layhid6.to.ytrain 3.310489e+00
## 1layhid7.to.ytrain -3.346698e+00
## 1layhid8.to.ytrain 2.665247e+00
## 1layhid9.to.ytrain -5.590634e+00
## 1layhid10.to.ytrain -2.755179e+00
```

Validation du modèle :

```
prob <- predict(nn,valid)
pred <- ifelse(prob>0.5, 1, 0)
pred
```

```
##      [,1]
## 2      0
## 7      1
## 11     0
## 15     0
## 26     0
## 36     0
## 37     0
## 46     0
## 57     0
## 62     1
## 65     0
## 67     0
## 70     0
## 71     1
## 81     0
## 84     1
## 85     0
## 86     0
## 87     0
## 97     0
## 116    0
## 124    1
## 126    1
## 128    0
## 129    0
## 131    0
## 132    0
## 140    0
## 141    0
## 142    1
## 143    0
## 146    0
```

## 151	0
## 153	0
## 154	0
## 161	0
## 168	0
## 171	0
## 180	0
## 183	0
## 184	0
## 187	1
## 190	0
## 197	0
## 199	0
## 210	0
## 220	0
## 232	0
## 240	0
## 241	0
## 244	0
## 261	0
## 262	0
## 263	0
## 268	0
## 282	0
## 283	0
## 285	0
## 287	0
## 288	0
## 301	1
## 304	1
## 305	0
## 306	0
## 308	0
## 328	1
## 329	0
## 341	0
## 342	0
## 356	0
## 358	0
## 363	0
## 368	0
## 371	0
## 375	0
## 377	0
## 380	0
## 383	0
## 386	0
## 389	0
## 390	0
## 394	0
## 395	0
## 399	0
## 400	1
## 401	0



## 417	0
## 418	0
## 419	0
## 420	0
## 421	0
## 427	0
## 429	0
## 432	0
## 435	0
## 444	1
## 450	0
## 455	0
## 463	0
## 467	0
## 472	0
## 479	0
## 481	0
## 496	0
## 497	0
## 500	0
## 502	0
## 511	0
## 516	0
## 519	0
## 520	1
## 526	0
## 553	0
## 564	0
## 565	0
## 567	0
## 577	0
## 588	0
## 589	0
## 592	0
## 597	0
## 609	0
## 619	0
## 621	1
## 624	0
## 626	0
## 628	0
## 629	0
## 639	0
## 650	0
## 653	0
## 655	0
## 669	0
## 670	0
## 672	0
## 677	0
## 685	0
## 691	0
## 700	0
## 703	0

## 706	0
## 707	0
## 712	0
## 714	0
## 717	0
## 720	0
## 728	0
## 732	0
## 733	0
## 738	1
## 739	0
## 749	1
## 752	0
## 754	0
## 760	1
## 761	0
## 765	0
## 766	1
## 772	0
## 774	0
## 775	0
## 776	0
## 780	0
## 782	0
## 790	0
## 793	1
## 795	1
## 801	1
## 808	0
## 809	0
## 813	0
## 822	0
## 825	0
## 826	0
## 836	1
## 843	0
## 844	0
## 846	0
## 852	0
## 853	0
## 861	0
## 863	0
## 867	0
## 880	0
## 881	0
## 883	0
## 884	0
## 893	0
## 896	0
## 912	0
## 915	0
## 916	0
## 921	0
## 932	0

## 933	0
## 934	0
## 938	0
## 941	0
## 948	0
## 954	0
## 964	0
## 965	1
## 968	0
## 969	0
## 971	0
## 972	0
## 974	0
## 981	1
## 982	0
## 987	0
## 989	0
## 995	0
## 998	0
## 999	0
## 1000	0
## 1001	0
## 1003	0
## 1004	0
## 1008	0
## 1010	0
## 1020	0
## 1025	0
## 1029	0
## 1031	0
## 1032	0
## 1038	0
## 1044	0
## 1049	1
## 1063	0
## 1068	0
## 1073	0
## 1074	0
## 1084	0
## 1085	0
## 1091	0
## 1096	0
## 1101	0
## 1102	0
## 1110	0
## 1123	0
## 1126	0
## 1128	0
## 1133	0
## 1146	0
## 1148	0
## 1178	1
## 1183	0
## 1186	0

##	1191	0
##	1194	0
##	1198	0
##	1202	0
##	1203	0
##	1205	0
##	1206	1
##	1207	1
##	1213	0
##	1214	0
##	1218	0
##	1222	0
##	1226	0
##	1230	0
##	1231	0
##	1232	0
##	1234	0
##	1238	0
##	1243	0
##	1245	0
##	1246	0
##	1253	0
##	1257	0
##	1267	0
##	1268	0
##	1269	0
##	1274	0
##	1279	0
##	1282	0
##	1298	0
##	1307	0
##	1309	0
##	1313	0
##	1318	0
##	1321	0
##	1324	0
##	1326	0
##	1343	1
##	1344	0
##	1345	0
##	1351	0
##	1355	0
##	1356	0
##	1358	0
##	1371	0
##	1374	0
##	1376	0
##	1378	0
##	1385	0
##	1387	0
##	1407	0
##	1432	0
##	1441	1
##	1445	1

##	1446	1
##	1448	0
##	1452	0
##	1467	0
##	1472	0
##	1474	1
##	1475	0
##	1490	0
##	1497	0
##	1502	0
##	1505	1
##	1506	0
##	1509	0
##	1510	1
##	1512	0
##	1513	0
##	1517	0
##	1520	0
##	1522	0
##	1523	0
##	1524	0
##	1537	0
##	1540	0
##	1549	0
##	1554	0
##	1555	0
##	1566	0
##	1569	0
##	1574	0
##	1575	0
##	1581	0
##	1583	0
##	1586	0
##	1589	0
##	1591	1
##	1599	0
##	1607	0
##	1609	0
##	1613	0
##	1615	0
##	1619	0
##	1620	0
##	1622	0
##	1626	0
##	1627	0
##	1630	0
##	1636	0
##	1637	0
##	1640	0
##	1647	0
##	1649	0
##	1660	0
##	1663	0
##	1668	0

## 1672	0
## 1676	0
## 1679	0
## 1680	0
## 1684	0
## 1685	0
## 1689	0
## 1691	0
## 1696	1
## 1698	0
## 1700	0
## 1703	0
## 1707	1
## 1710	0
## 1715	0
## 1716	0
## 1719	0
## 1721	0
## 1731	0
## 1735	0
## 1737	1
## 1739	1
## 1740	0
## 1743	0
## 1749	0
## 1758	0
## 1760	0
## 1763	0
## 1767	0
## 1768	1
## 1774	1
## 1779	0
## 1780	0
## 1781	0
## 1788	0
## 1791	0
## 1792	0
## 1798	0
## 1802	0
## 1810	0
## 1811	0
## 1813	0
## 1820	0
## 1821	0
## 1822	0
## 1823	1
## 1828	0
## 1829	0
## 1831	0
## 1836	0
## 1844	0
## 1850	0
## 1851	0
## 1852	0

##	1853	0
##	1859	0
##	1862	0
##	1868	0
##	1869	0
##	1870	1
##	1871	0
##	1876	1
##	1877	0
##	1879	0
##	1884	0
##	1885	0
##	1889	0
##	1891	0
##	1893	0
##	1894	0
##	1899	0
##	1902	0
##	1907	0
##	1908	0
##	1920	0
##	1923	0
##	1924	0
##	1926	0
##	1927	0
##	1928	0
##	1932	0
##	1936	0
##	1937	0
##	1947	1
##	1952	0
##	1954	0
##	1957	0
##	1958	0
##	1959	0
##	1965	1
##	1969	0
##	1970	0
##	1973	0
##	1982	0
##	1983	0
##	1985	0
##	1986	0
##	1989	0
##	1991	0
##	1994	0
##	1997	0
##	2000	0
##	2003	0
##	2005	1
##	2007	0
##	2009	0
##	2022	0
##	2030	0

##	2033	1
##	2039	1
##	2042	0
##	2044	0
##	2046	0
##	2049	0
##	2050	0
##	2052	0
##	2053	0
##	2057	0
##	2059	0
##	2062	0
##	2063	0
##	2064	0
##	2065	0
##	2069	0
##	2077	0
##	2080	0
##	2081	0
##	2084	0
##	2085	0
##	2099	0
##	2109	0
##	2113	0
##	2117	0
##	2120	0
##	2123	1
##	2125	0
##	2130	0
##	2132	0
##	2137	0
##	2142	0
##	2145	0
##	2147	0
##	2154	0
##	2159	0
##	2166	0
##	2169	1
##	2170	0
##	2180	0
##	2181	0
##	2188	0
##	2189	0
##	2191	0
##	2193	0
##	2200	0
##	2208	0
##	2209	0
##	2211	0
##	2213	1
##	2215	0
##	2219	0
##	2222	0
##	2225	0



##	2228	0
##	2230	0
##	2239	0
##	2241	0
##	2243	0
##	2245	0
##	2252	0
##	2253	0
##	2255	0
##	2262	0
##	2263	0
##	2264	0
##	2267	0
##	2277	0
##	2284	0
##	2285	0
##	2291	0
##	2293	0
##	2309	0
##	2310	0
##	2317	0
##	2324	0
##	2325	0
##	2337	0
##	2349	0
##	2350	0
##	2351	0
##	2355	0
##	2359	0
##	2360	0
##	2362	0
##	2378	0
##	2379	0
##	2383	0
##	2387	0
##	2392	0
##	2394	0
##	2398	0
##	2401	0
##	2409	0
##	2411	0
##	2414	0
##	2416	0
##	2421	0
##	2422	0
##	2423	0
##	2426	0
##	2427	0
##	2433	0
##	2436	0
##	2438	1
##	2441	0
##	2447	0
##	2453	0

##	2461	0
##	2463	0
##	2465	1
##	2467	0
##	2472	0
##	2476	0
##	2478	1
##	2482	0
##	2486	0
##	2496	0
##	2499	0
##	2506	0
##	2513	0
##	2521	0
##	2526	1
##	2533	0
##	2536	0
##	2537	0
##	2542	0
##	2545	0
##	2553	0
##	2558	0
##	2561	0
##	2567	0
##	2568	0
##	2569	0
##	2580	0
##	2583	0
##	2584	0
##	2593	0
##	2596	0
##	2598	0
##	2603	0
##	2607	0
##	2608	1
##	2609	0
##	2621	0
##	2622	0
##	2624	0
##	2631	0
##	2633	0
##	2639	0
##	2642	0
##	2647	0
##	2651	0
##	2656	0
##	2657	0
##	2664	0
##	2671	0
##	2673	0
##	2674	0
##	2676	0
##	2680	0
##	2690	0

##	2691	0
##	2692	0
##	2693	1
##	2694	0
##	2697	0
##	2698	1
##	2699	0
##	2702	0
##	2704	0
##	2706	0
##	2708	0
##	2710	0
##	2711	0
##	2719	0
##	2736	0
##	2745	0
##	2747	0
##	2748	0
##	2755	0
##	2756	1
##	2762	0
##	2763	0
##	2765	0
##	2766	0
##	2769	0
##	2770	0
##	2774	0
##	2775	0
##	2783	0
##	2784	0
##	2788	0
##	2794	0
##	2798	0
##	2801	0
##	2802	0
##	2805	0
##	2812	0
##	2814	0
##	2817	0
##	2822	0
##	2824	1
##	2827	1
##	2831	0
##	2834	0
##	2837	0
##	2838	1
##	2841	0
##	2843	0
##	2851	0
##	2852	0
##	2854	0
##	2860	0
##	2863	0
##	2869	0

##	2872	1
##	2876	1
##	2877	0
##	2880	1
##	2885	0
##	2886	0
##	2889	0
##	2890	0
##	2894	0
##	2925	0
##	2927	0
##	2928	0
##	2930	1
##	2931	0
##	2933	0
##	2939	0
##	2948	0
##	2951	0
##	2954	0
##	2968	0
##	2969	0
##	2973	0
##	2979	0
##	2981	1
##	2989	0
##	2997	0
##	3009	1
##	3013	0
##	3017	0
##	3018	0
##	3020	0
##	3022	1
##	3024	0
##	3031	0
##	3035	0
##	3039	0
##	3040	1
##	3051	0
##	3052	0
##	3055	0
##	3057	0
##	3058	0
##	3062	0
##	3067	1
##	3069	0
##	3071	0
##	3072	0
##	3073	0
##	3075	1
##	3078	0
##	3086	0
##	3091	0
##	3093	0
##	3097	0

##	3106	0
##	3110	0
##	3118	1
##	3125	0
##	3129	0
##	3130	0
##	3136	0
##	3140	0
##	3143	0
##	3146	0
##	3157	0
##	3164	0
##	3168	0
##	3170	0
##	3173	0
##	3174	0
##	3176	0
##	3177	0
##	3178	0
##	3179	0
##	3184	0
##	3189	0
##	3192	0
##	3195	0
##	3202	0
##	3205	0
##	3209	0
##	3213	0
##	3216	0
##	3222	0
##	3228	0
##	3230	0
##	3232	0
##	3241	0
##	3244	1
##	3246	0
##	3257	1
##	3259	0
##	3263	0
##	3266	0
##	3267	0
##	3270	0
##	3271	0
##	3277	0
##	3286	0
##	3287	0
##	3305	0
##	3306	0
##	3310	0
##	3315	1
##	3317	0
##	3324	0
##	3329	0
##	3340	0

##	3341	0
##	3342	1
##	3347	0
##	3352	0
##	3353	0
##	3359	0
##	3364	0
##	3367	0
##	3369	0
##	3372	0
##	3374	0
##	3376	1
##	3381	0
##	3388	0
##	3389	0
##	3390	0
##	3411	0
##	3414	0
##	3420	0
##	3426	0
##	3429	0
##	3437	0
##	3441	0
##	3443	0
##	3449	0
##	3453	0
##	3460	0
##	3465	0
##	3471	0
##	3477	0
##	3483	0
##	3485	0
##	3494	0
##	3496	0
##	3497	0
##	3503	1
##	3505	0
##	3506	0
##	3508	0
##	3520	1
##	3524	0
##	3526	1
##	3538	0
##	3540	0
##	3545	0
##	3553	0
##	3558	0
##	3560	0
##	3566	0
##	3571	0
##	3574	0
##	3576	0
##	3583	0
##	3590	1

##	3595	0
##	3596	0
##	3600	0
##	3602	0
##	3603	0
##	3607	0
##	3610	0
##	3646	0
##	3647	0
##	3653	0
##	3659	0
##	3661	0
##	3668	0
##	3675	0
##	3681	0
##	3687	0
##	3693	0
##	3697	0
##	3698	0
##	3701	0
##	3703	1
##	3705	0
##	3706	0
##	3708	0
##	3709	0
##	3717	0
##	3729	0
##	3733	0
##	3741	0
##	3743	0
##	3744	0
##	3746	0
##	3749	0
##	3750	0
##	3752	0
##	3756	0
##	3757	0
##	3759	0
##	3760	0
##	3762	0
##	3763	0
##	3767	0
##	3768	0
##	3770	0
##	3772	0
##	3773	0
##	3775	0
##	3776	0
##	3777	0
##	3780	0
##	3787	0
##	3791	0
##	3800	0
##	3807	0

##	3811	0
##	3813	0
##	3820	0
##	3827	0
##	3835	0
##	3847	1
##	3849	0
##	3856	0
##	3859	0
##	3861	0
##	3862	1
##	3863	0
##	3866	0
##	3872	0
##	3879	0
##	3884	1
##	3886	1
##	3887	0
##	3890	0
##	3892	1
##	3894	0
##	3895	0
##	3897	1
##	3900	0
##	3908	0
##	3909	0
##	3921	0
##	3926	0
##	3930	0
##	3935	1
##	3936	1
##	3937	1
##	3942	0
##	3943	0
##	3946	0
##	3947	0
##	3948	0
##	3950	0
##	3951	0
##	3953	0
##	3954	0
##	3957	1
##	3962	0
##	3966	0
##	3967	0
##	3971	1
##	3976	0
##	3986	0
##	3988	0
##	3994	0
##	3996	0
##	4000	0
##	4003	0
##	4004	0



##	4006	0
##	4007	0
##	4009	0
##	4010	0
##	4011	0
##	4012	0
##	4017	0
##	4026	0
##	4031	0
##	4034	0
##	4038	0
##	4042	0
##	4047	0
##	4049	0
##	4051	0
##	4057	1
##	4059	0
##	4062	1
##	4063	0
##	4065	0
##	4078	1
##	4081	0
##	4085	0
##	4086	1
##	4094	1
##	4096	0
##	4097	0
##	4100	0
##	4103	0
##	4104	0
##	4105	0
##	4110	0
##	4113	0
##	4115	0
##	4119	0
##	4121	0
##	4124	1
##	4128	0
##	4129	0
##	4133	0
##	4134	1
##	4138	0
##	4141	0
##	4143	0
##	4155	0
##	4156	0
##	4160	1
##	4163	0
##	4165	0
##	4170	0
##	4173	0
##	4177	0
##	4178	0
##	4190	0

##	4192	0
##	4194	0
##	4195	0
##	4196	0
##	4205	0
##	4206	0
##	4210	1
##	4215	0
##	4218	1
##	4219	0
##	4225	0
##	4226	0
##	4230	0
##	4233	1
##	4234	0
##	4235	0
##	4238	0
##	4248	0
##	4249	0
##	4251	0
##	4254	1
##	4257	0
##	4260	0
##	4264	0
##	4266	1
##	4270	0
##	4272	0
##	4273	0
##	4274	0
##	4278	0
##	4281	0
##	4285	0
##	4295	0
##	4303	0
##	4305	0
##	4306	0
##	4309	0
##	4319	0
##	4323	0
##	4333	0
##	4334	0
##	4335	1
##	4337	0
##	4338	0
##	4341	0
##	4342	0
##	4347	0
##	4349	0
##	4350	0
##	4354	0
##	4356	0
##	4357	0
##	4361	0
##	4366	0

```
## 4367 0
## 4368 0
## 4390 0
## 4393 1
## 4408 0
## 4418 0
## 4423 0
## 4428 0
## 4438 0
## 4440 0
## 4441 0
## 4444 0
## 4452 1
## 4454 0
## 4463 0
## 4473 0
## 4475 0
## 4476 0
## 4480 0
## 4483 0
## 4489 0
## 4496 1
## 4502 0
## 4505 1
## 4507 0
## 4517 0
## 4519 0
```

Voici la matrice de confusion entre les valeurs prédites et les valeurs test et le taux d'erreur :

```
m<-table(pred,yvalid)
m

##      yvalid
## pred  0   1
##    0 882  88
##    1  66  49

#ta d'erreur :
tx<-(m[1,2]+m[2,1])/length(yvalid)
tx

## [1] 0.1419355
```

Le taux d'erreur de ce premier modèle est de 8,5%.

Deuxième modèle :

```
nn2=neuralnet(ytrain~., data = Xtrain, hidden=10, err.fct="ce",linear.output = FALSE,likelihood = TRUE)

prob2 <- predict(nn2,valid)
pred2 <- ifelse(prob2>0.5, 1, 0)

m2<-table(pred2,yvalid)
tx2<-(m2[1,2]+m2[2,1])/length(yvalid)
tx2

## [1] 0.1465438
```

Le modèle deux est plus performant avec le “likelihood” (taux d’erreur = 8,1% < 8,5%).

Troisième modèle : augmentation du nombre de couche de la couche caché de 10 à 17 (nombre de variable d’entrée)

```
nn3=neuralnet(ytrain~., data = Xtrain, hidden=17, err.fct="ce",linear.output = FALSE)

prob3 <- predict(nn3,valid)
pred3 <- ifelse(prob3>0.5, 1, 0)

m3<-table(pred3,yvalid)
tx3<-(m3[1,2]+m3[2,1])/length(yvalid)
tx3
```

```
## [1] 0.1419355
```

Taux d’erreur un peu meilleur (7,9%<8,1%)

Quatrième modèle : augmentation du nombre de couche de la couche caché de 17 à 34 (deux fois le nombre de variable d’entrée)

```
nn4=neuralnet(ytrain~., data = Xtrain, hidden=34, err.fct="ce",linear.output = FALSE)

prob4 <- predict(nn4,valid)
pred4 <- ifelse(prob4>0.5, 1, 0)

m4<-table(pred4,yvalid)
tx4<-(m4[1,2]+m[2,1])/length(yvalid)
tx4
```

```
## [1] 0.1493088
```

Taux d’erreur un peu élevé, donc moins intéressant (7,6%<7,9%)

On garde la prédiction n°3 pour le nombre de neurones de la couche caché. Et nous décidons de garder 17 neurones dans la couche caché.

Cinquième modèle : On utilise une couche caché composé de 34 neurones et on passe l’argument “likelihood” à vrai.

```
nn5<-neuralnet(ytrain~., data = Xtrain, hidden=34, err.fct="ce",linear.output = FALSE, likelihood = TRUE)

prob5 <- predict(nn5,valid)
pred5 <- ifelse(prob5>0.5, 1, 0)

m5<-table(pred5,yvalid)
tx5<-(m5[1,2]+m[2,1])/length(yvalid)
tx5
```

```
## [1] 0.1447005
```

Cette fois ci le taux d’erreur est plus élevé (8,0%)

Nous testons un dernier modèle.

Sixième modèle : Nous augmentons encore une fois le nombre de neurones de la couche cachée (3 fois le nombre de variable d’entrée)

```
nn6<-neuralnet(ytrain~., data = Xtrain, hidden=51, err.fct="ce",linear.output = FALSE, likelihood = TRUE)

prob6 <- predict(nn6,valid)
pred6 <- ifelse(prob6>0.5, 1, 0)
```

```
m6<-table(pred6,yvalid)
tx6<-(m6[1,2]+m[2,1])/length(yvalid)
tx6
```

```
## [1] 0.1391705
```

Cette fois le taux d'erreur est de 7,8% ce qui est faiblement plus élevé que pour le quatrième modèle (7,6%). Mais ce modèle utilise un plus grand nombre de neurone ce qui peut engendrer du sur-apprentissage.

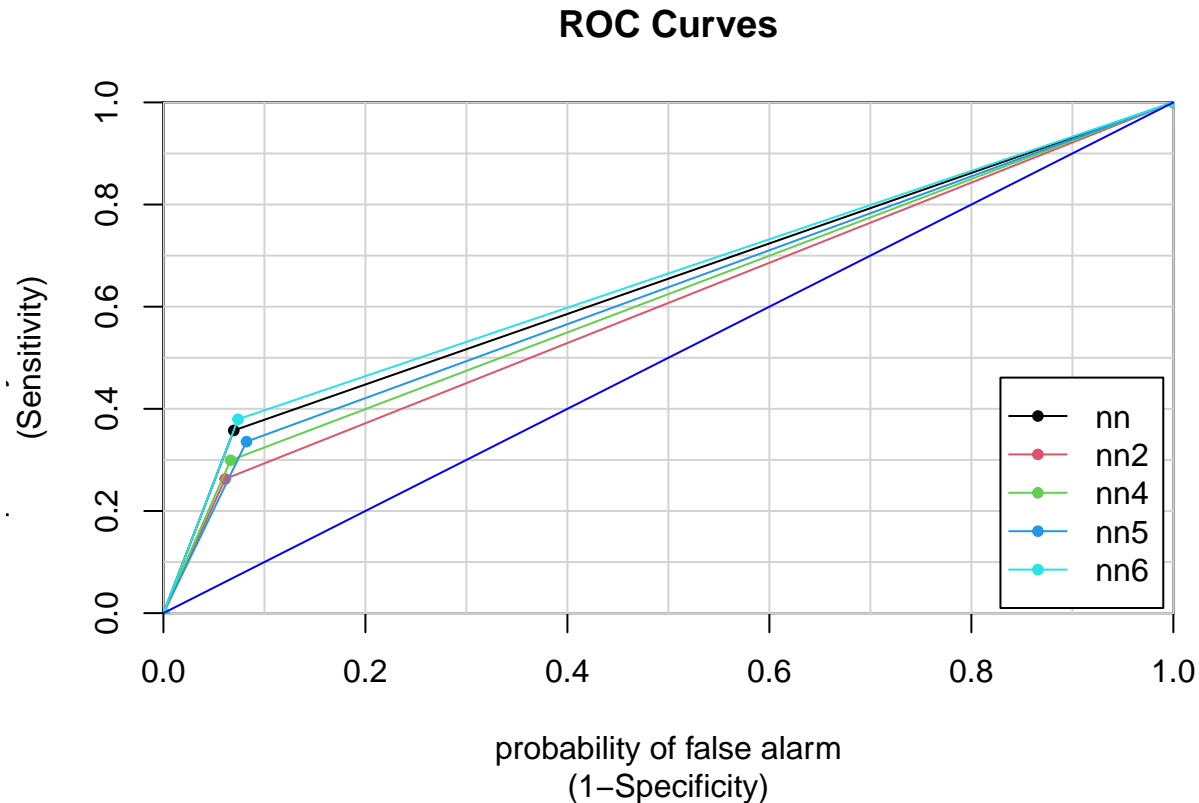
Avec la library “neuralnn” et la variable cible y correspondant à une classification binaire, il n’est pas possible de paramétrer le réseau de neurone à l’infinie. Les paramétrages pour une telle variable sont faible. Le paramètre “algorithmne” ne fonctionne qu’avec “rprop +”, la fonction d’erreur ne peut pas être modifier et la demande d’une régression logistique est impossible. Nous aurions pu changer les poids les variables d’entrée, mais le choix de l’aléatoire nous paraissait plus juste. Néanmoins, avec les réseaux de neurones que nous avons construit nous avons un taux d’erreur plutôt faible, ce qui nous paraissait être un bon résultat. Pour confirmer cette avis, nous décidons de comparer les courbes ROC des six modèles.

courbe ROC :

```
all_pred = data.frame(pred, pred2, pred4, pred5, pred6)
colnames(all_pred) = c("nn", "nn2", "nn4", "nn5", "nn6")
colAUC(all_pred, yvalid, plotROC = TRUE)
```

```
##          nn          nn2          nn4          nn5          nn6
## 0 vs. 1 0.644022 0.6007961 0.6164072 0.626744 0.6528612
```

```
abline(0,1, col = "blue")
```



A l’aide de la Courbe ROC ainsi que du taux d’erreur, nous considérons le meilleur réseau de neurone étant

le quatrième modèle construit. C'est à dire que nous utilisons 17 neurones caché et que nous plaçons à vrai l'argument "likelihood". C'est avec ce modèle que nous allons vérifier la bonne prédiction des valeurs test.

##Prédiction de la variable cible y

Représentation graphique du réseau de neurones sélectionné :

```
plot(nn4)
```

Voici la distribution de la variable y prédite par le réseau de neurone :

```
prob <- predict(nn4,test)
pred <- ifelse(prob>0.5, 1, 0)
pred
```

```
##      [,1]
## 5      0
## 13     0
## 16     0
## 17     0
## 18     0
## 25     0
## 27     0
## 30     0
## 32     0
## 34     0
## 45     0
## 47     0
## 48     0
## 53     0
## 55     0
## 56     0
## 58     0
## 63     0
## 64     0
## 73     0
## 75     0
## 76     0
## 78     0
## 79     1
## 83     0
## 91     0
## 94     1
## 100    0
## 102    0
## 106    1
## 108    0
## 109    0
## 120    0
## 135    0
## 136    0
## 137    0
## 147    0
## 156    0
## 162    0
## 164    0
```

## 169	0
## 173	0
## 181	0
## 186	0
## 198	0
## 201	1
## 204	1
## 206	0
## 217	0
## 225	0
## 228	0
## 231	0
## 238	0
## 242	0
## 243	0
## 245	0
## 249	0
## 251	0
## 265	1
## 267	0
## 269	0
## 272	0
## 277	0
## 280	0
## 294	0
## 297	0
## 307	0
## 311	0
## 314	0
## 333	0
## 339	1
## 343	0
## 344	0
## 345	1
## 349	0
## 352	0
## 357	0
## 361	0
## 362	0
## 364	0
## 365	0
## 370	0
## 373	0
## 379	0
## 384	0
## 391	0
## 402	1
## 405	0
## 407	0
## 408	0
## 410	0
## 415	0
## 424	0
## 431	0

## 433	0
## 434	0
## 440	0
## 447	0
## 456	0
## 464	0
## 466	0
## 468	0
## 470	0
## 475	0
## 477	0
## 478	1
## 485	0
## 489	0
## 506	0
## 507	0
## 522	0
## 530	0
## 535	0
## 537	0
## 542	0
## 547	1
## 549	0
## 555	0
## 558	1
## 559	0
## 570	0
## 571	0
## 572	0
## 573	0
## 574	0
## 575	0
## 578	0
## 580	0
## 582	1
## 590	1
## 606	0
## 620	1
## 622	0
## 636	0
## 637	0
## 644	0
## 646	1
## 654	0
## 657	0
## 660	0
## 665	0
## 667	0
## 686	1
## 687	0
## 692	0
## 696	0
## 701	0
## 702	0



## 704	0
## 705	0
## 727	1
## 729	0
## 731	0
## 735	0
## 736	0
## 740	0
## 746	0
## 753	0
## 762	0
## 770	0
## 771	0
## 777	0
## 785	0
## 794	0
## 798	0
## 799	0
## 802	0
## 804	1
## 819	0
## 824	1
## 827	0
## 832	0
## 834	0
## 838	0
## 845	0
## 849	0
## 850	1
## 857	0
## 862	0
## 864	0
## 869	0
## 872	0
## 874	0
## 882	0
## 888	0
## 891	0
## 892	0
## 895	0
## 901	0
## 902	0
## 906	0
## 907	0
## 913	0
## 914	0
## 917	0
## 919	0
## 925	0
## 926	0
## 945	0
## 946	0
## 957	1
## 961	0

## 967	0
## 977	0
## 979	1
## 980	0
## 997	0
## 1005	0
## 1017	0
## 1022	0
## 1026	0
## 1033	0
## 1034	0
## 1035	0
## 1040	0
## 1047	0
## 1051	0
## 1055	1
## 1061	1
## 1070	0
## 1077	0
## 1080	1
## 1082	0
## 1086	0
## 1092	0
## 1094	0
## 1097	0
## 1104	0
## 1106	0
## 1111	1
## 1122	0
## 1130	0
## 1136	0
## 1140	1
## 1141	0
## 1147	0
## 1150	0
## 1155	0
## 1158	0
## 1168	0
## 1170	0
## 1172	1
## 1175	0
## 1204	0
## 1208	0
## 1210	0
## 1217	0
## 1221	0
## 1229	0
## 1236	0
## 1244	0
## 1249	0
## 1263	0
## 1270	1
## 1283	0
## 1285	0

##	1286	0
##	1292	0
##	1304	1
##	1305	0
##	1306	0
##	1312	0
##	1317	0
##	1320	0
##	1333	0
##	1337	0
##	1338	0
##	1340	0
##	1346	0
##	1348	0
##	1352	0
##	1353	0
##	1375	0
##	1380	0
##	1381	0
##	1382	0
##	1386	0
##	1388	0
##	1393	0
##	1396	0
##	1397	0
##	1399	0
##	1402	0
##	1405	0
##	1408	0
##	1413	0
##	1415	0
##	1418	0
##	1422	0
##	1433	0
##	1434	0
##	1435	0
##	1436	0
##	1442	0
##	1443	1
##	1449	0
##	1450	0
##	1456	0
##	1457	0
##	1458	1
##	1460	1
##	1471	0
##	1473	0
##	1476	0
##	1477	0
##	1478	0
##	1479	1
##	1530	1
##	1539	0
##	1546	0

##	1547	0
##	1552	0
##	1559	0
##	1563	0
##	1567	0
##	1570	0
##	1571	0
##	1572	0
##	1582	0
##	1584	0
##	1585	0
##	1587	0
##	1593	0
##	1595	0
##	1606	0
##	1616	0
##	1621	0
##	1639	1
##	1643	0
##	1646	0
##	1648	0
##	1651	0
##	1652	0
##	1656	0
##	1659	0
##	1671	0
##	1675	0
##	1686	0
##	1687	0
##	1690	0
##	1692	0
##	1693	0
##	1695	0
##	1705	0
##	1706	0
##	1720	1
##	1732	0
##	1734	1
##	1741	0
##	1742	0
##	1745	1
##	1747	0
##	1752	0
##	1757	0
##	1759	0
##	1772	0
##	1773	0
##	1776	0
##	1782	0
##	1786	0
##	1790	0
##	1794	0
##	1795	0
##	1797	0

##	1801	0
##	1807	0
##	1808	0
##	1826	0
##	1834	0
##	1841	0
##	1842	0
##	1846	0
##	1855	1
##	1860	0
##	1861	0
##	1863	1
##	1866	0
##	1867	0
##	1883	0
##	1898	0
##	1905	1
##	1914	0
##	1933	0
##	1942	0
##	1945	0
##	1951	0
##	1953	1
##	1961	0
##	1962	0
##	1963	1
##	1966	0
##	1971	0
##	1976	0
##	1988	0
##	1992	0
##	1995	1
##	1996	0
##	2004	0
##	2010	0
##	2012	0
##	2015	1
##	2019	1
##	2020	0
##	2027	0
##	2029	0
##	2043	0
##	2054	0
##	2058	0
##	2061	0
##	2073	0
##	2086	0
##	2090	1
##	2091	0
##	2103	0
##	2108	0
##	2110	0
##	2111	0
##	2118	0

##	2131	0
##	2133	0
##	2136	0
##	2143	0
##	2150	0
##	2151	0
##	2156	0
##	2157	0
##	2158	0
##	2173	0
##	2174	0
##	2176	0
##	2184	0
##	2186	0
##	2205	0
##	2206	0
##	2217	0
##	2223	0
##	2224	0
##	2227	0
##	2233	0
##	2235	0
##	2237	0
##	2244	0
##	2248	0
##	2259	1
##	2271	0
##	2281	0
##	2282	0
##	2289	0
##	2292	0
##	2294	0
##	2297	0
##	2298	0
##	2300	0
##	2303	0
##	2304	0
##	2319	0
##	2323	0
##	2329	0
##	2333	0
##	2334	0
##	2336	0
##	2342	0
##	2344	0
##	2347	0
##	2358	0
##	2361	0
##	2365	0
##	2366	1
##	2367	0
##	2369	0
##	2373	0
##	2375	0

##	2380	0
##	2382	0
##	2391	0
##	2396	0
##	2397	1
##	2406	0
##	2428	0
##	2431	0
##	2432	0
##	2434	0
##	2437	1
##	2439	0
##	2442	0
##	2451	0
##	2455	0
##	2459	0
##	2462	0
##	2479	0
##	2485	1
##	2488	0
##	2493	0
##	2502	0
##	2507	0
##	2508	0
##	2511	0
##	2516	0
##	2522	0
##	2524	0
##	2531	0
##	2532	0
##	2541	0
##	2544	0
##	2556	0
##	2560	0
##	2571	0
##	2572	0
##	2573	0
##	2576	0
##	2586	0
##	2587	0
##	2594	0
##	2600	0
##	2605	1
##	2611	1
##	2619	0
##	2620	0
##	2635	0
##	2655	1
##	2662	0
##	2670	0
##	2677	0
##	2682	0
##	2683	0
##	2686	0

##	2700	0
##	2713	0
##	2716	0
##	2717	0
##	2720	0
##	2722	0
##	2724	0
##	2728	0
##	2729	0
##	2733	0
##	2738	0
##	2743	0
##	2750	0
##	2778	0
##	2782	0
##	2786	0
##	2789	0
##	2790	0
##	2796	0
##	2797	0
##	2807	1
##	2816	0
##	2820	1
##	2821	0
##	2826	0
##	2832	0
##	2835	0
##	2842	0
##	2845	0
##	2848	0
##	2853	1
##	2864	0
##	2867	0
##	2870	0
##	2874	0
##	2881	0
##	2895	0
##	2897	1
##	2898	0
##	2899	0
##	2901	0
##	2908	0
##	2909	0
##	2912	0
##	2920	0
##	2922	0
##	2926	0
##	2932	0
##	2941	0
##	2943	0
##	2944	0
##	2946	0
##	2947	0
##	2950	0



##	2956	0
##	2960	0
##	2964	0
##	2965	0
##	2966	0
##	2977	0
##	2986	0
##	2990	0
##	2992	0
##	2993	0
##	3006	0
##	3023	0
##	3025	0
##	3026	0
##	3028	0
##	3036	0
##	3037	1
##	3038	0
##	3041	0
##	3056	0
##	3060	0
##	3076	0
##	3081	0
##	3083	0
##	3087	0
##	3092	1
##	3094	0
##	3096	0
##	3100	0
##	3101	0
##	3105	0
##	3107	0
##	3108	0
##	3113	0
##	3114	0
##	3115	0
##	3121	0
##	3122	0
##	3123	0
##	3137	0
##	3138	0
##	3139	0
##	3144	0
##	3145	0
##	3153	1
##	3155	0
##	3156	1
##	3158	0
##	3159	0
##	3162	0
##	3166	0
##	3167	0
##	3171	0
##	3183	0

##	3185	0
##	3186	0
##	3187	0
##	3190	0
##	3193	0
##	3196	0
##	3197	0
##	3201	0
##	3203	1
##	3211	1
##	3212	0
##	3214	0
##	3218	0
##	3220	1
##	3221	0
##	3229	0
##	3233	0
##	3235	0
##	3247	0
##	3251	1
##	3262	0
##	3264	0
##	3265	0
##	3268	0
##	3276	0
##	3281	0
##	3282	1
##	3285	0
##	3308	0
##	3313	1
##	3314	0
##	3319	1
##	3320	0
##	3322	0
##	3325	0
##	3334	1
##	3335	0
##	3336	0
##	3348	0
##	3349	0
##	3356	0
##	3357	0
##	3362	0
##	3363	0
##	3368	0
##	3370	0
##	3375	0
##	3378	0
##	3383	0
##	3385	0
##	3393	0
##	3394	0
##	3396	0
##	3397	0

##	3400	0
##	3405	0
##	3408	0
##	3409	0
##	3413	0
##	3415	0
##	3422	0
##	3423	0
##	3431	0
##	3440	0
##	3461	0
##	3463	0
##	3470	0
##	3473	0
##	3479	0
##	3480	0
##	3487	0
##	3489	0
##	3502	0
##	3507	0
##	3510	0
##	3511	0
##	3513	0
##	3518	0
##	3519	0
##	3527	0
##	3531	0
##	3539	1
##	3542	0
##	3543	0
##	3546	0
##	3548	0
##	3549	0
##	3550	0
##	3552	0
##	3555	0
##	3557	0
##	3561	0
##	3563	0
##	3575	0
##	3587	1
##	3588	1
##	3589	0
##	3593	1
##	3598	0
##	3604	0
##	3612	0
##	3616	0
##	3621	0
##	3624	0
##	3628	0
##	3632	0
##	3636	0
##	3640	0

##	3641	0
##	3643	1
##	3648	0
##	3650	0
##	3655	0
##	3673	0
##	3678	0
##	3680	0
##	3688	0
##	3691	0
##	3696	1
##	3704	0
##	3715	1
##	3716	0
##	3718	0
##	3720	0
##	3724	0
##	3732	0
##	3745	0
##	3753	0
##	3764	0
##	3766	1
##	3769	0
##	3771	0
##	3790	1
##	3794	0
##	3798	1
##	3799	0
##	3815	0
##	3817	0
##	3824	0
##	3837	0
##	3839	0
##	3844	0
##	3850	0
##	3857	0
##	3858	0
##	3860	0
##	3868	0
##	3873	0
##	3874	1
##	3877	0
##	3880	0
##	3883	0
##	3885	0
##	3888	1
##	3896	0
##	3898	0
##	3913	0
##	3914	0
##	3920	0
##	3922	0
##	3923	0
##	3932	0

## 3933	1
## 3944	0
## 3949	0
## 3952	0
## 3959	0
## 3960	0
## 3984	0
## 3997	0
## 4001	0
## 4002	0
## 4019	0
## 4021	0
## 4027	0
## 4032	0
## 4033	0
## 4039	0
## 4043	0
## 4044	0
## 4045	0
## 4048	1
## 4052	0
## 4053	0
## 4066	0
## 4090	0
## 4092	0
## 4098	0
## 4101	0
## 4111	0
## 4135	0
## 4144	0
## 4147	0
## 4157	0
## 4159	0
## 4162	0
## 4164	0
## 4166	0
## 4171	0
## 4175	0
## 4184	0
## 4187	0
## 4197	1
## 4211	0
## 4212	1
## 4221	0
## 4228	0
## 4252	0
## 4261	1
## 4262	0
## 4263	0
## 4265	0
## 4279	0
## 4280	0
## 4284	0
## 4287	0

## 4291	0
## 4292	0
## 4296	0
## 4298	0
## 4304	0
## 4310	0
## 4313	0
## 4317	0
## 4318	0
## 4320	0
## 4324	1
## 4325	0
## 4327	1
## 4332	0
## 4339	0
## 4344	0
## 4355	0
## 4365	1
## 4369	0
## 4371	1
## 4373	0
## 4375	0
## 4376	1
## 4381	0
## 4383	0
## 4384	0
## 4385	0
## 4387	1
## 4392	1
## 4397	1
## 4405	1
## 4407	1
## 4413	0
## 4414	1
## 4416	0
## 4420	0
## 4421	0
## 4430	0
## 4433	0
## 4439	0
## 4446	0
## 4448	0
## 4449	0
## 4453	0
## 4455	0
## 4457	0
## 4485	1
## 4491	0
## 4497	0
## 4499	0
## 4500	0
## 4508	0
## 4510	0
## 4511	0

```
## 4516    0
```

Voici la matrice de confusion entre les valeurs prédite et les valeurs test :

```
m<-table(pred,ytest)
m
```

```
##      ytest
## pred  0   1
##      0 740  63
##      1  63  39
```

Voici le taux d'erreur de cette prédiction :

```
#ta d'erreur :
tx<-(m[1,2]+m[2,1])/length(ytest)
tx
```

```
## [1] 0.1392265
```

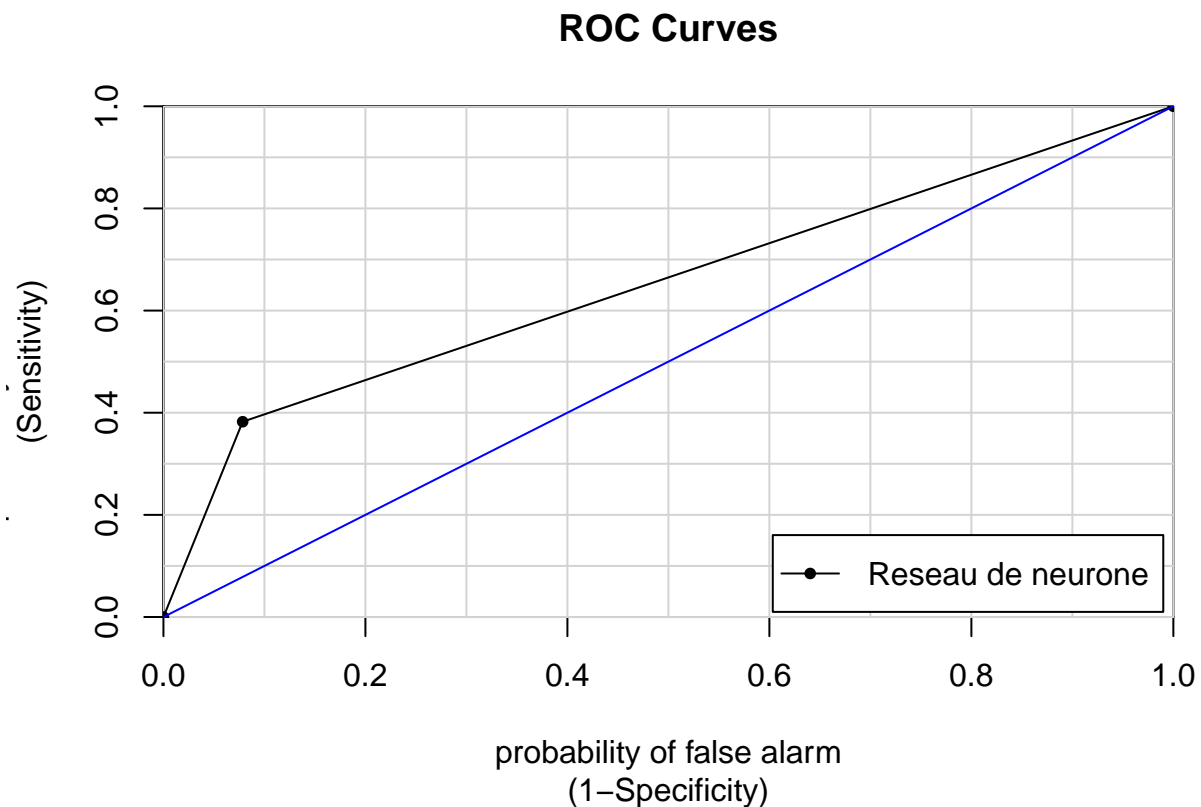
Le taux d'erreur est de 7,1%.

courbe ROC représentant :

```
colnames(pred)="Reseau de neurone"
colAUC(pred, ytest, plotROC = TRUE)
```

```
##           Reseau de neurone
## 0 vs. 1           0.6519486
```

```
abline(0,1, col = "blue")
```



L'air sous la courbe représentant le reseau de neurones est supérieur à celle de la courbe représentant l'aléatoire.