

HEC MONTRÉAL

Projet MATH60603

Prévisions de perte de clientèle

DAVID GARSON, 11262368
JEAN GUIRGUIS, NUM-MATRICULE

Projet présenté à

PROFESSEURE AURÉLIE LABBÉ

Table des mati  res

1	Introduction : pr��sentation des objectifs de l��tude	2
2	Exploration des donn��es	2
2.1	Pr��sentation du jeu de donn��e	2
2.2	Pr��sentation des observations	3
2.3	R��encodage ou traitement pr��paratoire des donn��es	6
3	Mod��le de simulation	7
3.1	R��gression logistique (validation crois��) — a corriger la courbe roc	7
3.2	Arbre de classification ��lag��	10
3.3	for��t al��atoire approche bagging	14
3.4	Boosting	17
3.5	SVM	18
4	R��sultats : pr��sentation des r��sultats sous forme de tableaux et figures (ne mettez pas de sortie R)	18
5	Conclusion/discussion : conclusion g��n��rale, limites de votre ��tude, qu��avez vous appris ?	18

Note :

baggin + svm + regression lin  aire

  quilibrer VS non   quilibr   (3 m  thode)

https://www.slideshare.net/yogesh_khandelwal/churn-modelling

https://www.erpublication.org/published_paper/IJETR032129.pdf

1. Introduction : pr  sentation des objectifs de l  tude

L'objectif de cette   tude est de faire une pr  diction sur le d  sabonnement d'un client pour un service de t  l  communication. Cette simulation a   t   fait    partir d'une base de donn  es pr  sent sur kaggle    l'adresse suivante : <https://www.kaggle.com/mnassrib/telecom-churn-datasets?select=churn-bigml-80.csv>. L'analyse a   t   effectu  e sur les clients d'Orange Telecom aux   tats-Unis et plusieurs variables ont   t   analys  es tels que lieu g  ographique, le type de plan que le client poss  de, le nombre de minutes utilis   pour des appels durant le jour et le soir, etc.

Ce rapport contient en premier lieu, une pr  sentation d  taill  e de notre m  thode d'analyse. Cette section contiendra d'abord une exploration d  taill  e des donn  es poss  d  es, un traitement et un nettoyage pr  paratoire des donn  es. Puis, plusieurs m  thodes et strat  gies de pr  diction seront test  es afin de d  terminer le mod  le le mieux adapt      notre   chantillon d'utilisateurs.

Une fois le mod  le choisi, la deuxi  me section du rapport pr  sentera les r  sultats du mod  le de pr  diction.

... ..

2. Exploration des donn  es

Cette section pr  sentera d'abord les variables du jeu de donn  e. La r  partition des observations dans chacune des variables est rapidement survol  e. Cette section explique aussi les diff  rents traitements, r  encodage et validation ex  cut  s sur les variables et les observations avant de d  buter l'analyse des diff  rents mod  les de simulations.

2.1. Pr  sentation du jeu de donn  e

Ci-dessous est pr  sent   l'ensemble des variables. L'ensemble des donn  es est donc constitu   de 20 variables   num  r  es ci-dessous. Le jeu de donn  e est donc initialement constitu   de 4 variables de type cha  ne de caract  res ("State", "International.plan", "Voice.mail.plan" et "Churn"). La variable Area.code est cat  gorielle, tandis que tous les autres varibales sont continuent.

- State :   tats am  ricaine de l'observation, valeur de type "character"
- Account.length : Valeur enti  re depuis combien de temps le compte existe-t-il
- Area.code : Indicatif r  gionale (415, 408 ou 510)
- International.plan : Adh  sion au plan international (Yes ou No)
- Voice.mail.plan : Adh  sion au plan de messagerie (Yes ou No)
- Number.vmail.messages : Nombre de message vocal
- Total.day.minutes : Nombre de minute utilis   durant le jour
- Total.day.calls : Nombre d'appel ex  cut   durant le jour
- Total.day.charge : Co  t total pour l'utilisation de jour
- Total.eve.minutes : Nombre de minute utilis   durant le soir
- Total.eve.calls : Nombre d'appel ex  cut   durant le soir
- Total.eve.charge : Co  t total pour l'utilisation de soir
- Total.night.minutes : Nombre de minute utilis   durant la nuit

- Total.night.calls : Nombre d'appel ex  cut   durant la nuit
- Total.night.charge : Co  t total pour l'utilisation la nuit
- Total.intl.minutes : Nombre de minute utilis      l'international
- Total.intl.calls : Nombre d'appel ex  cut      l'international
- Total.intl.charge : Co  t total pour l'utilisation    l'international
- Customer.service.calls : Nombre d'appel ex  cut   pour le service au client
- Churn : Est-ce que le client    quitt   (True, False)

```
summary(train)
```

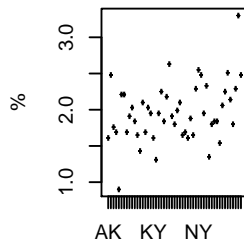
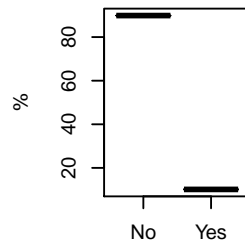
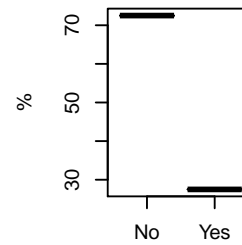
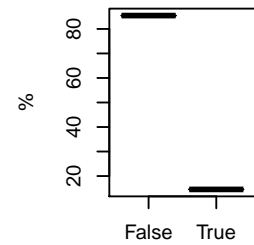
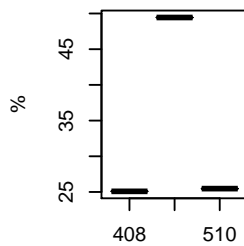
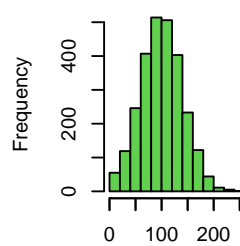
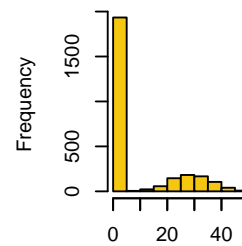
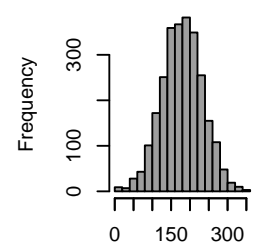
```
##      State      Account.length      Area.code      International.plan
## Length:2666      Min.   : 1.0      Min.   :408.0      Length:2666
## Class :character 1st Qu.: 73.0      1st Qu.:408.0      Class :character
## Mode  :character Median :100.0      Median :415.0      Mode  :character
##                      Mean  :100.6      Mean  :437.4
##                      3rd Qu.:127.0      3rd Qu.:510.0
##                      Max.   :243.0      Max.   :510.0
## Voice.mail.plan  Number.vmail.messages Total.day.minutes Total.day.calls
## Length:2666      Min.   : 0.000      Min.   : 0.0      Min.   : 0.0
## Class :character 1st Qu.: 0.000      1st Qu.:143.4      1st Qu.: 87.0
## Mode  :character Median : 0.000      Median :179.9      Median :101.0
##                      Mean  : 8.022      Mean  :179.5      Mean  :100.3
##                      3rd Qu.:19.000      3rd Qu.:215.9      3rd Qu.:114.0
##                      Max.   :50.000      Max.   :350.8      Max.   :160.0
## Total.day.charge Total.eve.minutes Total.eve.calls Total.eve.charge
## Min.   : 0.00      Min.   : 0.0      Min.   : 0      Min.   : 0.00
## 1st Qu.:24.38      1st Qu.:165.3      1st Qu.: 87      1st Qu.:14.05
## Median :30.59      Median :200.9      Median :100      Median :17.08
## Mean   :30.51      Mean   :200.4      Mean   :100      Mean   :17.03
## 3rd Qu.:36.70      3rd Qu.:235.1      3rd Qu.:114      3rd Qu.:19.98
## Max.   :59.64      Max.   :363.7      Max.   :170      Max.   :30.91
## Total.night.minutes Total.night.calls Total.night.charge Total.intl.minutes
## Min.   : 43.7      Min.   : 33.0      Min.   : 1.970      Min.   : 0.00
## 1st Qu.:166.9      1st Qu.: 87.0      1st Qu.: 7.513      1st Qu.: 8.50
## Median :201.2      Median :100.0      Median : 9.050      Median :10.20
## Mean   :201.2      Mean   :100.1      Mean   : 9.053      Mean   :10.24
## 3rd Qu.:236.5      3rd Qu.:113.0      3rd Qu.:10.640      3rd Qu.:12.10
## Max.   :395.0      Max.   :166.0      Max.   :17.770      Max.   :20.00
## Total.intl.calls Total.intl.charge Customer.service.calls Churn
## Min.   : 0.000      Min.   :0.000      Min.   :0.000      Length:2666
## 1st Qu.: 3.000      1st Qu.:2.300      1st Qu.:1.000      Class :character
## Median : 4.000      Median :2.750      Median :1.000      Mode  :character
## Mean   : 4.467      Mean   :2.764      Mean   :1.563
## 3rd Qu.: 6.000      3rd Qu.:3.270      3rd Qu.:2.000
## Max.   :20.000      Max.   :5.400      Max.   :9.000
```

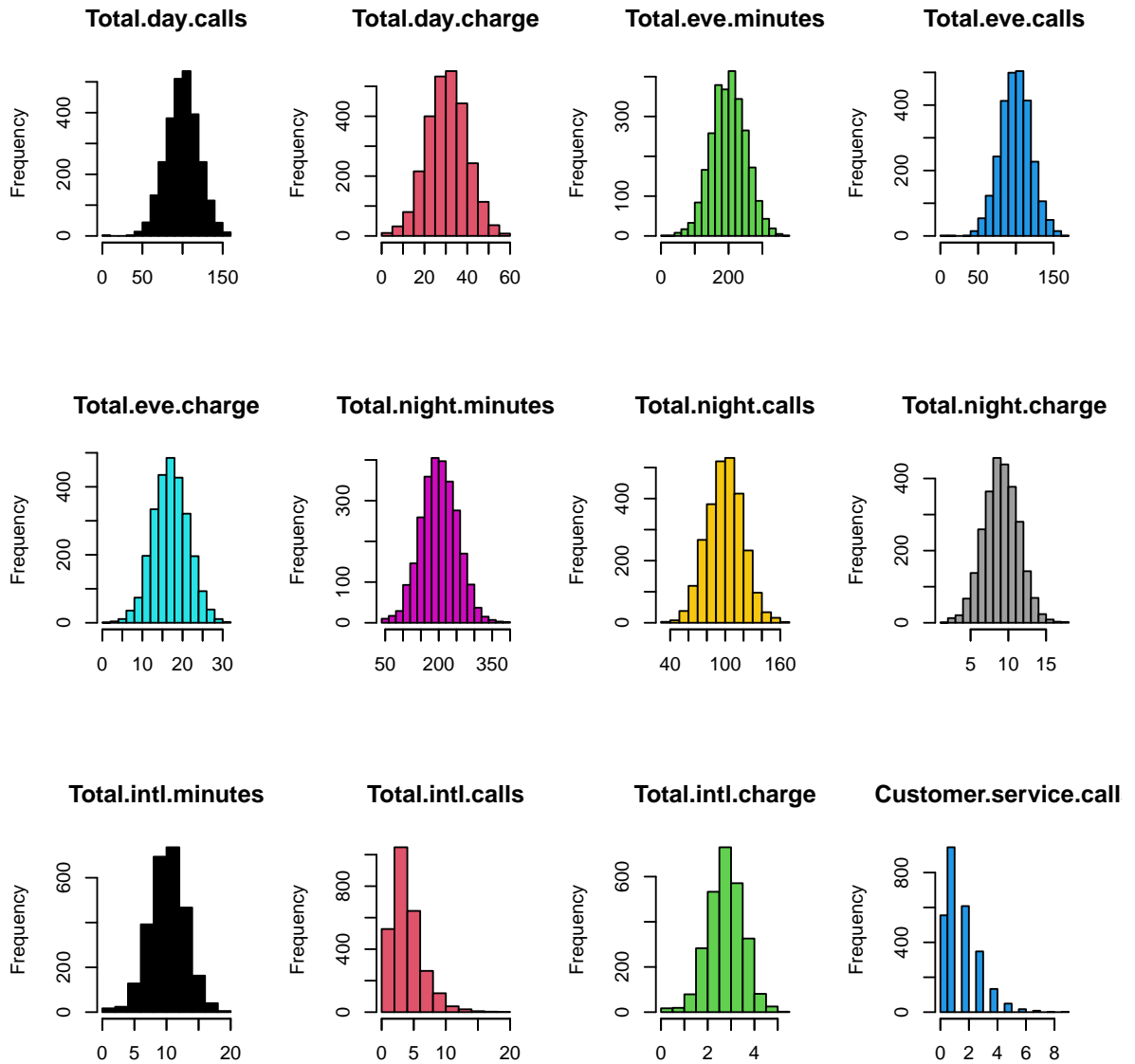
2.2. Pr  sentation des observations

Ci-dessous les graphiques pr  sente la r  partition des valeurs pour chacune des variables du jeu de donn  e. Les 5 premiers graphique pr  sentes la fr  quence en % des observations pour les variabes de type cha  ne de caract  res et cat  gorielle. Tandis que les variables de type continue sont pr  sent   sous forme d'histogramme.

   partir des graphiques ci-dessous quelques interpr  tations int  ressantes peuvent   tre faites :

- Le nombre d'observation fait sont similaire d'un   tats    l'autre
- Le jeu de donn  es est principalement compos   d'observation sur des personnes n'ayant pas adh  r      un plan international et ni    un plan de messagerie vocal.
- La grande majorit   des observation n'ont pas quitt   leurs compagnie de t  l  communication et la majorit   des observations habitent le secteur 415.
-    partir des graphiques d'histogrammes, toutes les variables de type continue suivent une courbe normale    l'ex  ption des variables Customer.service.calls et total.intl.calls

Proportion States**Proportion International****Proportion Voice.mai****Proportion Churn****Proportion Area code****Account.length****Number.vmail.message****Total.day.minutes**



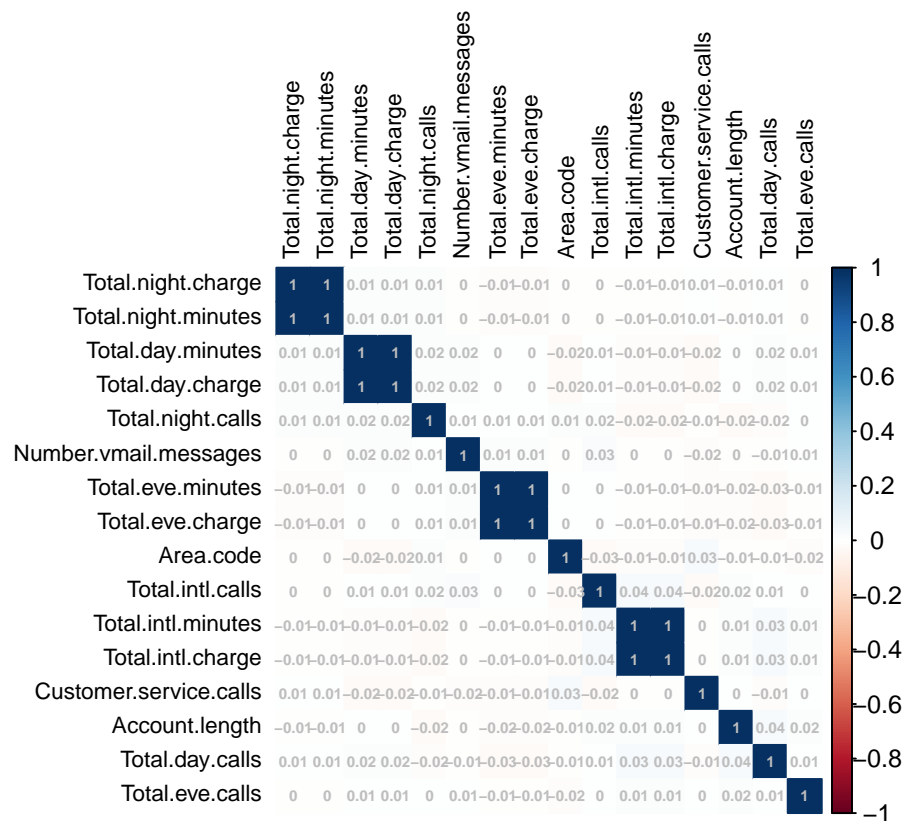
2.3. R  encodage ou traitement pr  paratoire des donn  es

Tel que pr  sent   dans le jeu de donn  e et pour la suite des simulations plusieurs valeurs ont   t   d'abord r  encod  :

- La variable "State" a   t   convertie en une variable Facteur
- Les variables "International.plan" , "Voice.mail.plan" et "Churn" en variable logique.
- Le dataset d'entra  nement ne contient aucune valeur manquante.
- Le log normale des variables a   t   calcul   sur les variables Customer.service.calls et total.intl.calls, car leurs courbent d'histogramme   taient d  centr   vers la gauche.

2.3.1 Pr  sentation de la corr  lation entre les variables

Le graphique ci-dessous pr  sente la corr  lation entre les variables. Les variables du dataset sont faiblement corr  l  es    l'exception des variables indiquant le nombre de minutes consomm  es et les frais charg  s associ  s, comme les variables : "Total.night charge" et "Total.night.minutes". Comme ces variables ont une corr  lation parfaites, nous d  cisons de supprimer du dataset les variables "charge". Ce qui revient    supprimer 4 variables du dataset.

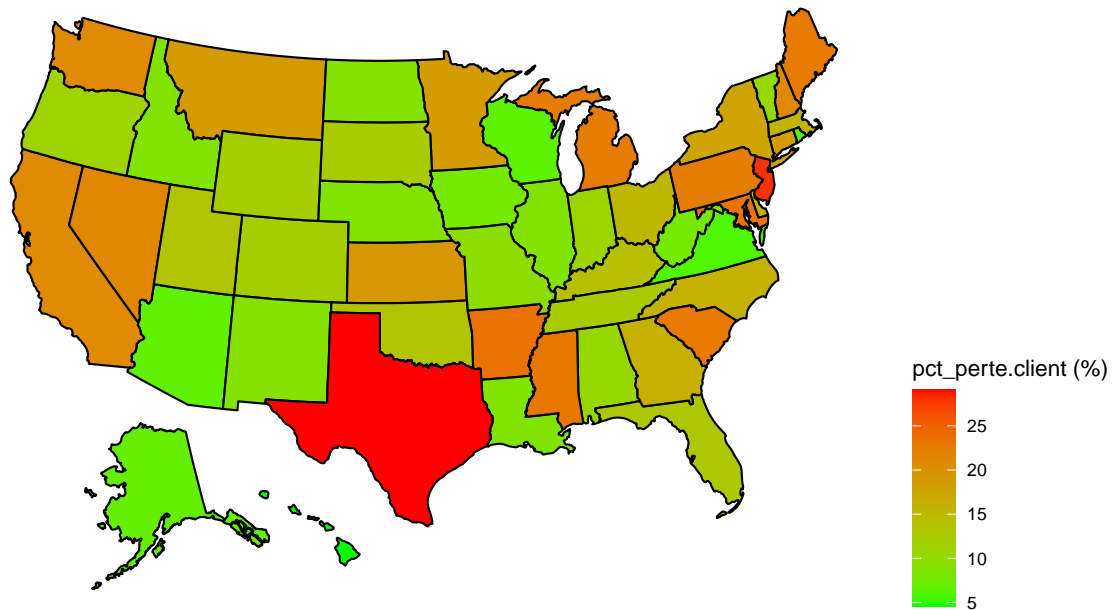


2.3.2 Pr  sentation du taux de Churn selon l'  tat

Ci-dessous un map est pr  sent   avec son taux de perte de client. On constate que les   tats du Texas et New Jersey sont les   tats ayant perdus le plus de client  le, avec un pourcentage de perte sup  rieur    25%.

Perte client  le   tats-Unis

Op  rateur Orange t  l  com



3. Mod  le de simulation

3.1. R  gression logistique (validation crois  ) — a corriger la courbe roc

```
train = train0
test = train0

set.seed(1234)

train$Churn[train$Churn == "False"] = 0
train$Churn[train$Churn == "True"] = 1
train$Churn = as.numeric(train$Churn)

test$Churn[test$Churn == "False"] = 0
test$Churn[test$Churn == "True"] = 1
test$Churn = as.numeric(test$Churn)

model1=glm(Churn~ ., family="binomial",data=train)
summary(model1)

##
## Call:
## glm(formula = Churn ~ ., family = "binomial", data = train)
```



```
##
## Deviance Residuals:
##      Min        1Q      Median        3Q        Max
## -1.9878   -0.4968   -0.3075   -0.1580    3.1228
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -8.920e+00  1.260e+00  -7.079 1.45e-12 ***
## StateAL        2.075e-01  7.925e-01   0.262  0.79346
## StateAR        8.758e-01  7.711e-01   1.136  0.25607
## StateAZ        1.406e-01  9.246e-01   0.152  0.87916
## StateCA        1.309e+00  8.860e-01   1.477  0.13969
## StateCO        3.176e-01  7.994e-01   0.397  0.69114
## StateCT        1.150e+00  7.500e-01   1.533  0.12524
## StateDC        7.592e-01  8.254e-01   0.920  0.35770
## StateDE        6.462e-01  7.753e-01   0.833  0.40463
## StateFL        5.189e-01  7.931e-01   0.654  0.51291
## StateGA        7.058e-01  7.902e-01   0.893  0.37175
## StateHI       -7.328e-01  1.002e+00  -0.731  0.46457
## StateIA        3.493e-01  9.146e-01   0.382  0.70252
## StateID        3.175e-01  8.344e-01   0.381  0.70355
## StateIL       -4.100e-01  8.949e-01  -0.458  0.64683
## StateIN        2.688e-01  8.043e-01   0.334  0.73824
## StateKS        9.024e-01  7.661e-01   1.178  0.23883
## StateKY        5.636e-01  8.182e-01   0.689  0.49096
## StateLA        5.682e-01  9.068e-01   0.627  0.53091
## StateMA        1.069e+00  7.853e-01   1.361  0.17346
## StateMD        9.575e-01  7.461e-01   1.283  0.19937
## StateME        1.375e+00  7.587e-01   1.813  0.06987 .
## StateMI        1.285e+00  7.462e-01   1.722  0.08506 .
## StateMN        1.175e+00  7.426e-01   1.583  0.11353
## StateMO        3.961e-01  8.306e-01   0.477  0.63344
## StateMS        1.505e+00  7.654e-01   1.967  0.04920 *
## StateMT        1.645e+00  7.562e-01   2.176  0.02957 *
## StateNC        2.904e-01  7.996e-01   0.363  0.71650
## StateND        2.940e-03  8.741e-01   0.003  0.99732
## StateNE        2.949e-01  8.536e-01   0.345  0.72973
## StateNH        1.305e+00  7.888e-01   1.654  0.09811 .
## StateNJ        1.398e+00  7.471e-01   1.871  0.06134 .
## StateNM        3.212e-01  8.655e-01   0.371  0.71052
## StateNV        1.040e+00  7.450e-01   1.395  0.16289
## StateNY        1.111e+00  7.467e-01   1.488  0.13671
## StateOH        7.216e-01  7.650e-01   0.943  0.34559
## StateOK        5.418e-01  7.982e-01   0.679  0.49728
## StateOR        3.802e-01  7.912e-01   0.481  0.63080
## StatePA        1.244e+00  8.055e-01   1.544  0.12260
## StateRI       -9.942e-01  9.569e-01  -1.039  0.29882
## StateSC        1.726e+00  7.675e-01   2.249  0.02451 *
## StateSD        4.039e-01  8.078e-01   0.500  0.61703
## StateTN        3.497e-01  8.477e-01   0.412  0.67998
## StateTX        1.809e+00  7.353e-01   2.461  0.01386 *
## StateUT        9.522e-01  7.754e-01   1.228  0.21945
## StateVA       -6.646e-01  8.765e-01  -0.758  0.44833
## StateVT       -2.364e-01  8.448e-01  -0.280  0.77962
## StateWA        1.326e+00  7.692e-01   1.724  0.08473 .
```

```
## StateWI          -1.862e-01  8.740e-01  -0.213  0.83128
## StateWV          2.495e-01  7.771e-01   0.321  0.74819
## StateWY          2.085e-01  7.812e-01   0.267  0.78952
## Account.length   1.482e-03  1.630e-03   0.909  0.36343
## Area.code        -3.446e-04  1.532e-03  -0.225  0.82202
## International.planYes 2.278e+00  1.704e-01  13.373 < 2e-16 ***
## Voice.mail.planYes -2.159e+00  6.811e-01  -3.170  0.00152 **
## Number.vmail.messages 4.076e-02  2.143e-02   1.902  0.05721 .
## Total.day.minutes -3.491e-01  3.826e+00  -0.091  0.92729
## Total.day.calls    3.608e-03  3.246e-03   1.111  0.26643
## Total.day.charge   2.132e+00  2.251e+01   0.095  0.92455
## Total.eve.minutes  1.419e+00  1.921e+00   0.738  0.46029
## Total.eve.calls    -1.615e-03  3.220e-03  -0.501  0.61609
## Total.eve.charge   -1.662e+01  2.260e+01  -0.735  0.46223
## Total.night.minutes 4.749e-04  1.023e+00   0.000  0.99963
## Total.night.calls  1.735e-03  3.294e-03   0.527  0.59839
## Total.night.charge  5.718e-02  2.272e+01   0.003  0.99799
## Total.intl.minutes -2.985e+00  6.178e+00  -0.483  0.62895
## Total.intl.calls   -1.228e-01  3.005e-02  -4.086  4.38e-05 ***
## Total.intl.charge  1.141e+01  2.288e+01   0.499  0.61795
## Customer.service.calls 5.463e-01  4.662e-02  11.718 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 2212.2  on 2665  degrees of freedom
## Residual deviance: 1641.2  on 2597  degrees of freedom
## AIC: 1779.2
##
## Number of Fisher Scoring iterations: 6
```

```
#BIC
round(BIC(model1), digit = 2)
```

```
## [1] 2185.52
```

```
#AIC
round(model1$aic, digit = 2)
```

```
## [1] 1779.22
```

```
library(ROCR)
prob.predict=predict.glm(model1,test,type="response")
cutoff=0.5
test.pred = rep(0, nrow(test))
test.pred[prob.predict > cutoff] = 1
M=table(test.pred, test$Churn,dnn=c("Prediction","Observation"))

a=M[1,1]
b=M[1,2]
c=M[2,1]
d=M[2,2]
```

```

# Sensibilit  
Sensibilit   = d/(b+d)

# Sp  cificit  
Sp  cificit   = a/(a+c)

# Taux d'erreurs de classification (%)
((b+c)/(a+b+c+d))*100

```

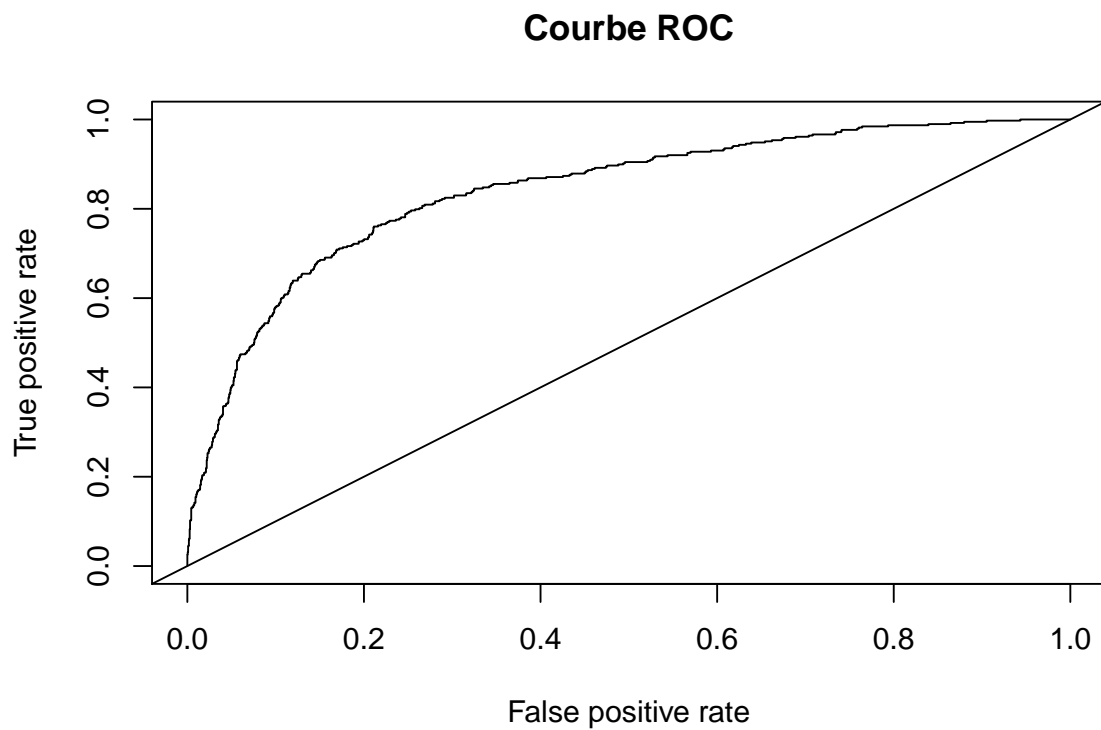
```
## [1] 12.90323
```

```

pred=prediction(prob.predict,test$Churn)
perf=performance(pred,measure="tpr",x.measure="fpr")
auc.perf = performance(pred, measure = "auc")

plot(perf, main = "Courbe ROC")
abline(a=0,b=1)

```



3.2. Arbre de classification   lag  

```

train = train0
test = train0

train$Churn[train$Churn == "False"] = 0
train$Churn[train$Churn == "True"] = 1

```

```
train$Churn = as.numeric(train$Churn)

test$Churn[test$Churn == "False"] = 0
test$Churn[test$Churn == "True"] = 1
test$Churn = as.numeric(test$Churn)

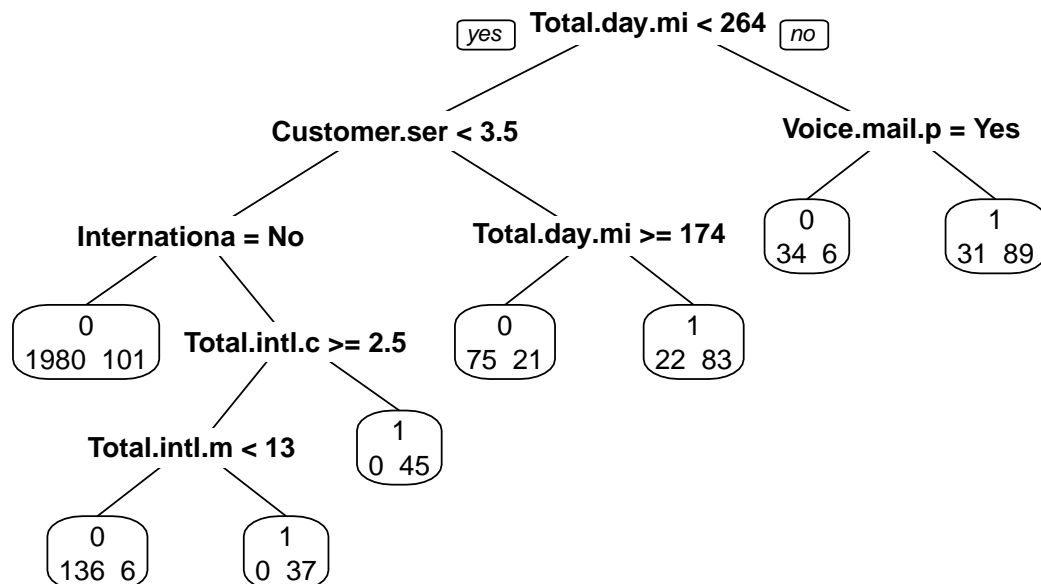
set.seed(400)

# Création de l'arbre :
library(rpart.plot)

## Loading required package: rpart

mytree = rpart(Churn~., data=train , method = "class")
cp_optimal=mytree$cptable[which.min(mytree$cptable[,4]),1]
mytree_optimal = prune(mytree,cp=cp_optimal)
prp(mytree_optimal,extra=1,roundint=FALSE, main = "Arbre avec élagage")
```

Arbre avec élagage



```
mytable=table(test$Churn, predict(mytree_optimal,test, type="class"))
names(dimnames(mytable))= c("Observed", "Predicted")
M = mytable

a=M[1,1]
b=M[1,2]
c=M[2,1]
d=M[2,2]
```

```
# Taux de mauvaises classifications
((b+c)/(a+b+c+d))*100
```

```
## [1] 7.014254
```

```
# Taux de faux positifs
round(((b+c)/(a+b+c+d))*100, digit = 2)
```

```
## [1] 7.01
```

```
# Taux faux n  gatif
round((1-(d/(b+d)))*100, digit = 2)
```

```
## [1] 17.26
```

```
# Param  tre de complexit   maximale
cp_optimal
```

```
## [1] 0.02835052
```

```
# Boucle for
```

```
nb_boucle = 20
```

```
Taux_mauvaise_classificaion=matrix(0,nb_boucle,1)
Taux_faux_positifs=matrix(0,nb_boucle,1)
Taux_faux_negatif=matrix(0,nb_boucle,1)
valeur_seed = matrix(0,nb_boucle,1)
```

```
par(mfrow=c(4,3))
```

```
for (i in 1:nb_boucle)
{
  n=nrow(mydata)
  set.seed(i*15231)
  id.train=sample(1:n,size=nrow(train))
  id.test=setdiff(1:n,id.train)
```

```
mydata.train= mydata[id.train,]
mydata.test = mydata[id.test,]
```

```
library(rpart.plot)
set.seed(i*15231)
valeur_seed[i]=i*15231
```

```
mytree = rpart(Churn~., data=mydata.train, method = "class")
```

```
cp_optimal=mytree$cptable[which.min(mytree$cptable[,4]),1]
mytree_optimal = prune(mytree,cp=cp_optimal)
```

```
#prp(mytree_optimal,extra=1,roundint=FALSE)
```

```

mytable=table(mydata.test$Churn, predict(mytree_optimal,mydata.test, type="class"))
names(dimnames(mytable))= c("Observed", "Predicted")
M = mytable

a=M[1,1]
b=M[1,2]
c=M[2,1]
d=M[2,2]

# Taux de mauvaises classifications
Taux_mauvaise_classificaion[i]= ((b+c)/(a+b+c+d))
# Taux faux positifs
Taux_faux_positifs[i]= round((1-(a/(a+c))), digit = 2 )
# Taux faux n  gatif
Taux_faux_n  gatif[i] = round((1-(d/(b+d))), digit = 2)

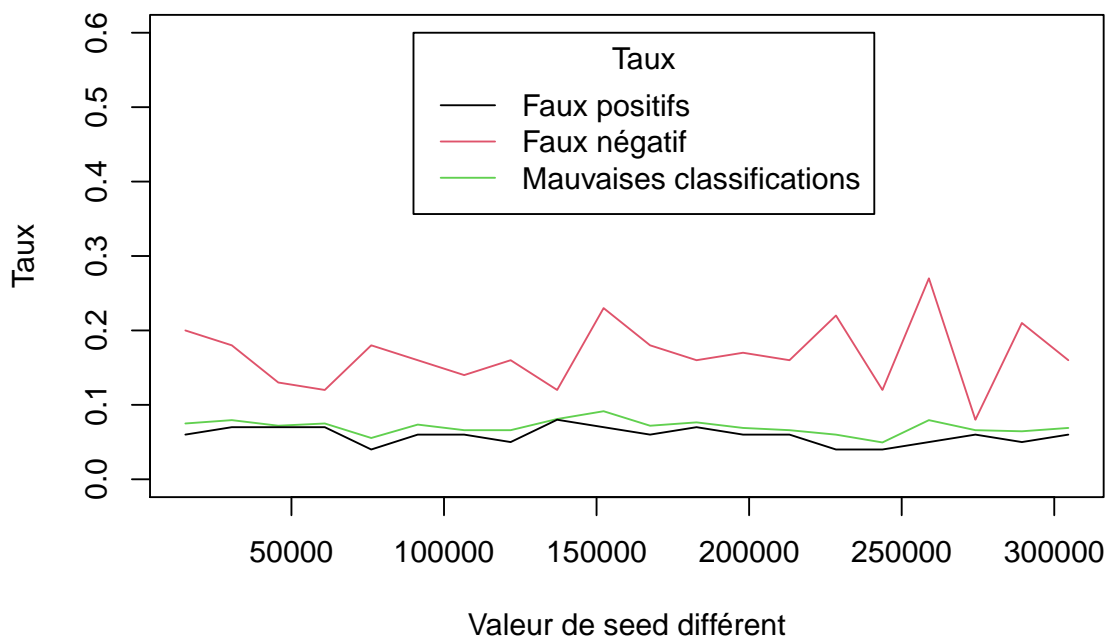
}

par(mfrow=c(1,1))

plot(valeur_seed,Taux_mauvaise_classificaion, type = "l", ylim= c(0,0.6), col = 3, ylab = "Taux")
lines(valeur_seed,Taux_faux_positifs,col=1)
lines(valeur_seed,Taux_faux_n  gatif,col=2)
legend(90000, 0.6, legend=c("Faux positifs","Faux n  gatif","Mauvaises classifications"), col=1:3)

```

Variation du taux de mauvaises clasification, de faux positif et de faux n  gatif



```
# Taux de faux positif
## Maximum
round(max(Taux_faux_positifs), digit =2)
```

```
## [1] 0.08
```

```
## Minimum
round(min(Taux_faux_positifs), digit =2)
```

```
## [1] 0.04
```

```
# Taux de mauvaises classifications
## Maximum
round(max(Taux_mauvaise_classificaion), digit =2 )
```

```
## [1] 0.09
```

```
## Minimum
round(min(Taux_mauvaise_classificaion) , digit = 2)
```

```
## [1] 0.05
```

```
# Taux faux n  gatif
## Maximum
round(max(Taux_faux_negatif), digit = 2)
```

```
## [1] 0.27
```

```
## Minimum
round(min(Taux_faux_negatif), digit = 2)
```

```
## [1] 0.08
```

3.3. for  t al  atoire approche bagging

```
train = train0
test = train0

train$Churn[train$Churn == "False"] = 0
train$Churn[train$Churn == "True"] = 1
train$Churn = as.numeric(train$Churn)

test$Churn[test$Churn == "False"] = 0
test$Churn[test$Churn == "True"] = 1
test$Churn = as.numeric(test$Churn)

library(randomForest)
```

```
## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

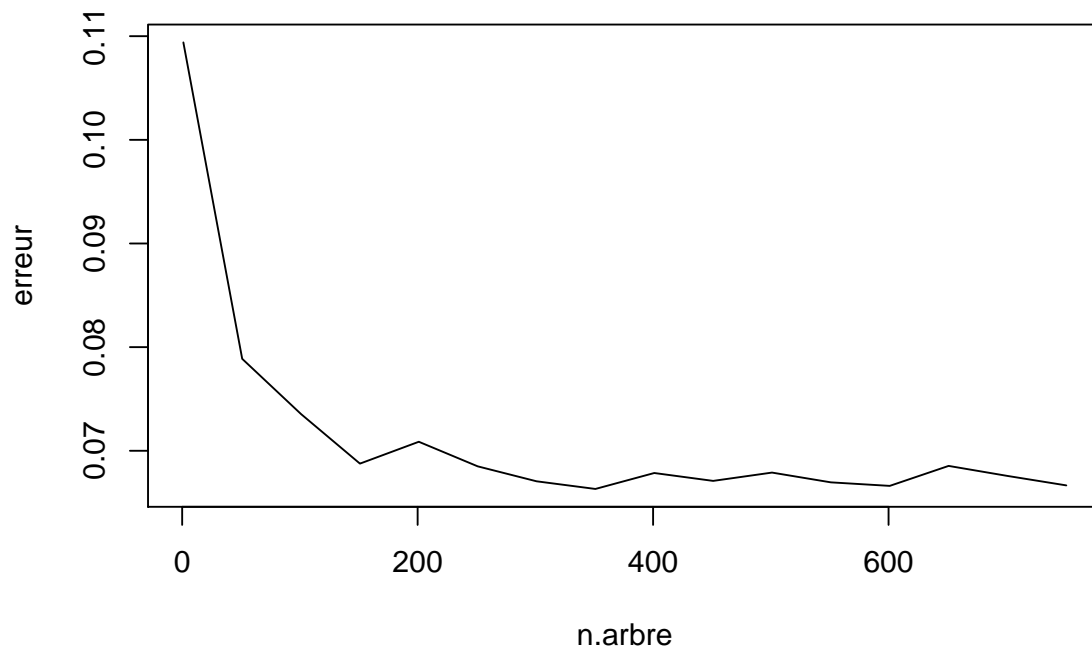
## The following object is masked from 'package:dplyr':
##
##      combine

## The following object is masked from 'package:ggplot2':
##
##      margin

## The following object is masked from 'package:gridExtra':
##
##      combine

mydata$Churn <- as.factor(mydata$Churn)

set.seed(1234)
n.arbre=seq(1,800,by=50)
erreur=NULL
for (i in n.arbre)
{
  rf=randomForest(Churn~.,data=mydata,ntree=i,mtry=length(colnames(mydata))-1)
  erreur=c(erreur,sum(rf$err.rate[,1])/rf$ntree)
}
plot(n.arbre, erreur,type="l")
```

```
rf=randomForest(Churn~.,data=mydata,ntree=600,mtry=length(colnames(mydata))-1)
rf
```

```
##
## Call:
## randomForest(formula = Churn ~ ., data = mydata, ntree = 600,      mtry = length(colnames(mydata))-1)
##              Type of random forest: classification
##              Number of trees: 600
## No. of variables tried at each split: 15
##
##      OOB estimate of  error rate: 6.78%
## Confusion matrix:
##      FALSE TRUE class.error
## FALSE  2759   91  0.03192982
## TRUE    135  348  0.27950311
```

```
importance(rf,type=1)
```

```
##
## State
## Account.length
## Area.code
## International.plan
## Voice.mail.plan
## Number.vmail.messages
## Total.day.minutes
## Total.day.calls
## Total.eve.minutes
```

```
## Total.eve.calls
## Total.night.minutes
## Total.night.calls
## Total.intl.minutes
## Total.intl.calls
## Customer.service.calls
```

3.4. Boosting

```
library(randomForest)
library(adabag)
```

```
## Warning: package 'adabag' was built under R version 4.0.3

## Loading required package: caret

## Loading required package: foreach

## Warning: package 'foreach' was built under R version 4.0.3

## Loading required package: doParallel

## Loading required package: iterators

## Warning: package 'iterators' was built under R version 4.0.3

## Loading required package: parallel
```

```
train = train0
test = train0

train$Churn[train$Churn == "False"] = 0
train$Churn[train$Churn == "True"] = 1
train$Churn = as.numeric(train$Churn)

test$Churn[test$Churn == "False"] = 0
test$Churn[test$Churn == "True"] = 1
test$Churn = as.numeric(test$Churn)

train$Churn=as.factor(train$Churn)
test$Churn=as.factor(test$Churn)

set.seed(1234)

# boosting with trees of depth 10
myboost=boosting(Churn~., data=train, mfinal = 10, coeflearn = 'Freund', control=rpart.control(
myboost$importance
```

```
##      Account.length      Area.code Customer.service.calls
##      4.4688932          0.3117404          7.4183039
##      International.plan  Number.vmail.messages      State
##      5.0942122          0.4453988          34.5424419
##      Total.day.calls      Total.day.charge      Total.day.minutes
##      2.9945890          0.0000000          16.6550028
##      Total.eve.calls      Total.eve.charge      Total.eve.minutes
##      0.6688840          0.0000000          8.2465733
##      Total.intl.calls      Total.intl.charge      Total.intl.minutes
##      4.7813379          0.0000000          4.2572276
##      Total.night.calls      Total.night.charge      Total.night.minutes
##      3.3159080          0.0000000          4.2649704
##      Voice.mail.plan
##      2.5345167
```

```
pred=predict(myboost, newdata=test)

pred$error
```

```
## [1] 0.009002251
```

```
M=pred$confusion
```

```
#mytable=table(myboost$class,mydata$Churn)
# names(dimnames(mytable))= c("Predicted", "Observed")
# M = mytable
# M
# a=M[1,1]
# b=M[1,2]
# c=M[2,1]
# d=M[2,2]

# Taux de faux positifs
M[2,1]/(M[2,1]+M[1,1])
```

```
## [1] 0.0008779631
```

```
# Taux de faux n  gatifs
M[1,2]/(M[2,2]+M[1,2])
```

```
## [1] 0.05670103
```

3.5. SVM

4. R  sultats : pr  sentation des r  sultats sous forme de tableaux et figures (ne mettez pas de sortie R)

5. Conclusion/discussion : conclusion g  n  rale, limites de votre   tude, qu'avez vous appris ?