

Arbres de décision et SVM

Jean-Michel Marin

octobre 2016

Contents

1	Algorithme CART	1
1.1	Construction avec la fonction rpart de la bibliothèque rpart	1
1.2	Elagage	2
1.3	Prédiction	4
2	Fôrets aléatoires	4
2.1	Construction avec la fonction randomForest de la bibliothèque randomForest	4
2.2	Prédiction	5
3	SVM	5
3.1	Construction avec la fonction svm de la bibliothèque e1071	5
3.2	Prédiction	5

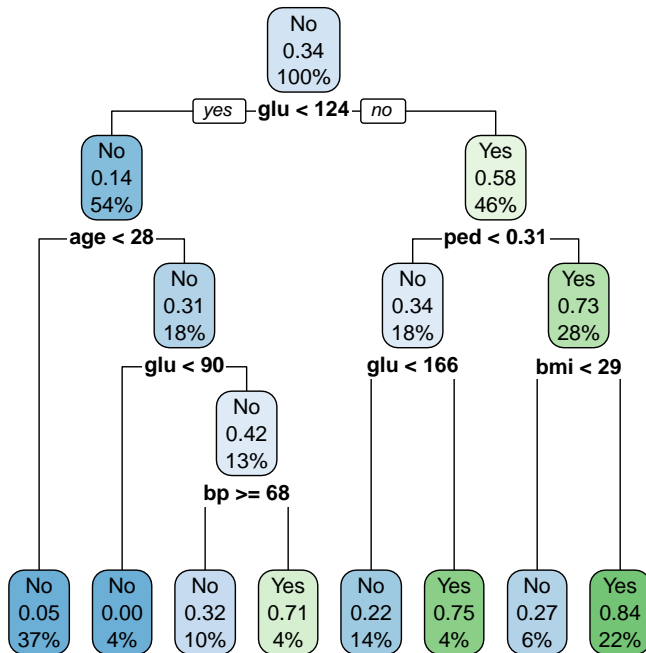
Nous allons traiter les données pima.

```
library(MASS)
data("Pima.te")
data("Pima.tr")
```

1 Algorithme CART

1.1 Construction avec la fonction rpart de la bibliothèque rpart

```
library(rpart)
model.cart <- rpart(type ~ ., data=Pima.tr)
library(rpart.plot)
rpart.plot(model.cart)
```



1.2 Elagage

```
printcp(model.cart)
```

```
##
## Classification tree:
## rpart(formula = type ~ ., data = Pima.tr)
##
## Variables actually used in tree construction:
## [1] age bmi bp glu ped
##
## Root node error: 68/200 = 0.34
##
## n= 200
##
##      CP nsplit rel error  xerror   xstd
## 1 0.220588    0  1.00000 1.00000 0.098518
## 2 0.161765    1  0.77941 0.85294 0.094370
## 3 0.073529    2  0.61765 0.76471 0.091224
## 4 0.058824    3  0.54412 0.75000 0.090647
## 5 0.014706    4  0.48529 0.61765 0.084709
## 6 0.010000    7  0.44118 0.70588 0.088822
```

```
library(caret)
```

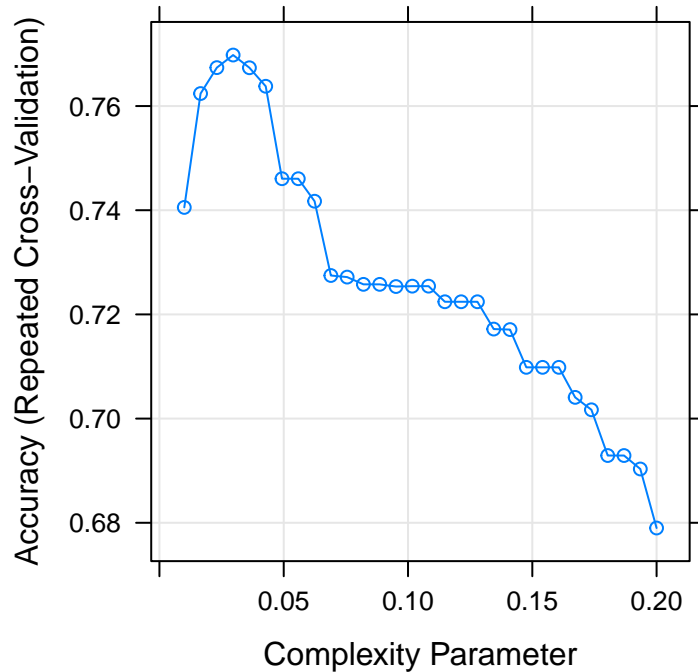
```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```

cv.cart <- train(type~.,data=Pima.tr,method="rpart",
  metric="Accuracy",trControl=trainControl(method="repeatedcv",
    repeats=50,number=10),tuneGrid=data.frame(cp=seq(0.01,0.2,length=30)))
plot(cv.cart)

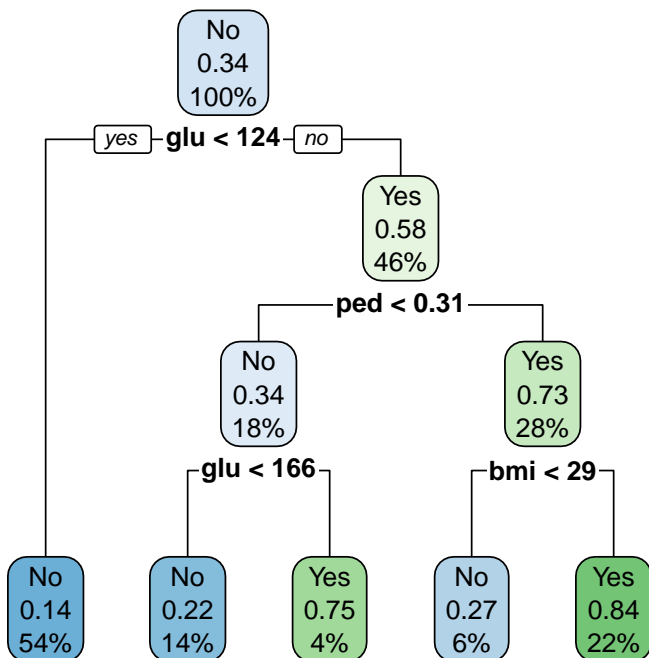
```



```

model.cart <- prune(model.cart,cp=as.numeric(cv.cart$best))
rpart.plot(model.cart)

```



1.3 Prédiction

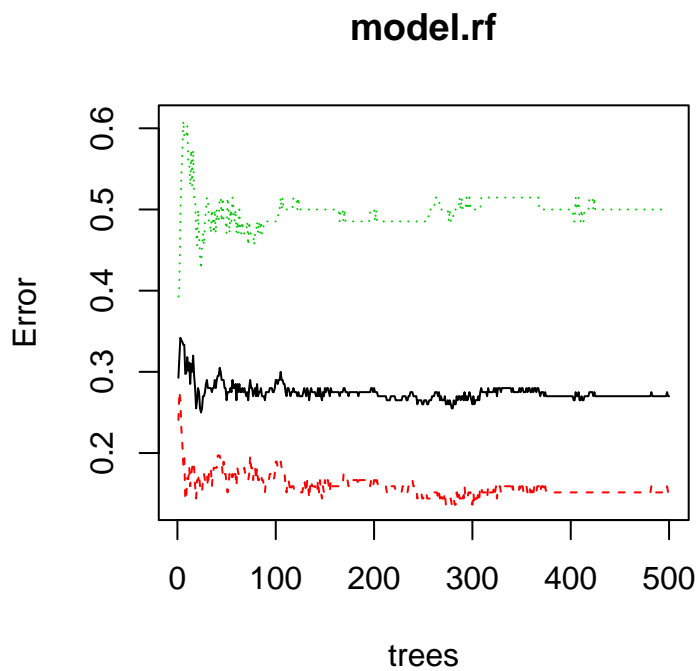
```
pred.cart <- predict(model.cart,Pima.te,type="class")  
mean(Pima.te$type!=pred.cart)
```

```
## [1] 0.2439759
```

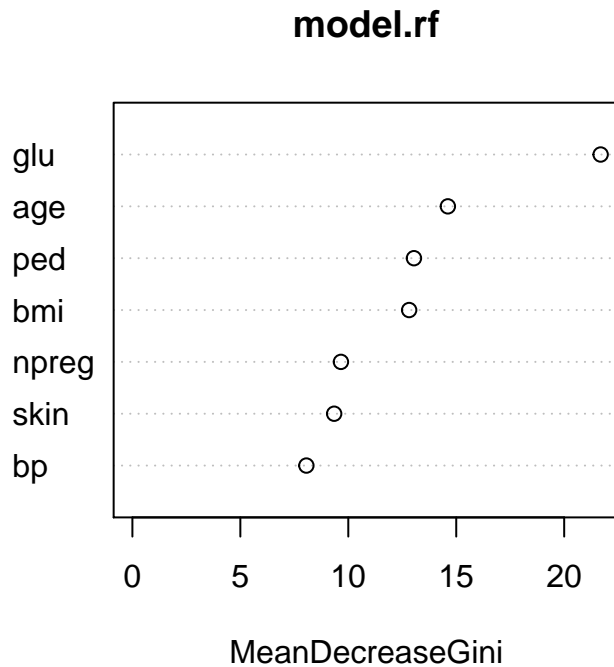
2 Fôrets aléatoires

2.1 Construction avec la fonction randomForest de la bibliothèque randomForest

```
library(randomForest)  
model.rf <- randomForest(type~.,data=Pima.tr,ntree=500)  
plot(model.rf)
```



```
varImpPlot(model.rf)
```



2.2 Prédiction

```
pred.rf <- predict(model.rf,Pima.te,type="class")  
mean(Pima.te$type!=pred.rf)
```

```
## [1] 0.2319277
```

3 SVM

3.1 Construction avec la fonction svm de la bibliothèque e1071

```
library(e1071)  
calibration <- tune.svm(type~.,data=Pima.tr,gamma=seq(0.001,1,by=0.01))  
model.svm <- svm(type~.,data=Pima.tr,gamma=calibration$best.parameters)
```

3.2 Prédiction

```
pred.svm <- predict(model.svm,Pima.te)  
mean(Pima.te$type!=pred.svm)
```

```
## [1] 0.2108434
```