Classification Advanced Machine Learning

LOPEZ Fabien

05/03/2021

Classification

Afin de mieux visualiser ce que symbolise exactement la classification, partons sur quelques exemples. Tout d'abord, chargeons un jeu de données. Par exemple, le jeu de données MNIST,

```
require(class)
```

```
## Loading required package: class
```

```
library(class)
data("iris")
set.seed(42)
summary(iris)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
##        Species
##  setosa    :50
##  versicolor:50
##  virginica :50
##
##
##
```

Comme on peut le voir ici, nous avons exactement 3 races d'iris distinctes : les setosas, les versicolors et les vifinicas avec pour chaque iris (peu importe l'espèce) la longueur et largeur du sépale et du pétale. L'objectif dans ce RMarkdown va être d'identifier les espèces en fonction des données d'entrées.

Classification des moindres carrées

Pour cet exemple, nous n'utiliserons que 2 types d'iris. Ici nous préparons tout d'abord les données avec un dataset pour l'entrainement (trainset) et un pour le test (testset). Nous les décomposerons chacun en 2, un premier repectivement x_train et x_test qui contiendront les valeurs des variables que nous supposons liées au résultat à trouver et ensuite y_train et y_test qui contiennent chacun le résultat, c'est-à-dire la classe à laquelle appartient l'iris dans notre cas.

```
dataset <- subset(iris, Species == "setosa" | Species == "versicolor")
dt <- sample(nrow(dataset), nrow(dataset) * 0.8)
```

```
trainset<-dataset[dt,]
testset <- dataset[setdiff(1:nrow(dataset), dt),]
nrow(trainset)
```

```
## [1] 80
```

```
nrow(testset)
```

```
## [1] 20
```

```
y_train <- factor(trainset$Species)
x_train <- cbind.data.frame(trainset$Sepal.Length, trainset$Sepal.Width, trainset$Petal.Length, trainse
y_test <- testset$Species
x_test <-cbind.data.frame(testset$Sepal.Length, testset$Sepal.Width, testset$Petal.Length, testset$Peta
```

On peut voir qu'aillant découpé mon dataset en 80% pour le train et 20% pour le test, ne travaillant plus qu'avec 100 éléments (50 pour chaque espèce), j'obtiens effectivement 80 individus dans le train pour 20 dans le test. Maintenant que nos données sont prêtes nous pouvons lancer le modèle :

```
names(x_train) <- c("Sepal.Length", "Sepal.Width", "Petal.Length", "Pepal.Length")
names(x_test) <- c("Sepal.Length", "Sepal.Width", "Petal.Length", "Pepal.Length")
names(y_train) <- "Species"
#names(y_test) <- "Species"
model <- lm(trainset$Species~., trainset, family = binomial)
prevision <- predict(model, testset)
res <- ifelse(prevision>1.5, "versicolor", "setosa")
res
```

```
##           1           7          11          13          19          31
##     "setosa"    "setosa"    "setosa"    "setosa"    "setosa"    "setosa"
##          46          53          59          60          62          64
##     "setosa" "versicolor" "versicolor" "versicolor" "versicolor" "versicolor"
##          73          75          77          82          85          86
## "versicolor" "versicolor" "versicolor" "versicolor" "versicolor" "versicolor"
##          90          98
## "versicolor" "versicolor"
```

Ici, on peut voir que notre résultat a bien été classé suivant les classe Versicolor et Setosa. On se retrouve donc avec 7 setosas et 13 versicolors.

Classification logistique

Passons à la classification logistique. Les datasets étant déjà opérationnels, l'exécution de l'exemple se basera dessus. nous n'avons ici qu'à appliquer la commande *glm*.

```
modelglm <- glm(y_train~., x_train, family = binomial(logit))
resglm <- predict(modelglm, x_test)
ifelse(resglm>0.5,"versicolor", "setosa")
```

```
##           1           2           3           4           5           6
##     "setosa"    "setosa"    "setosa"    "setosa"    "setosa"    "setosa"
##           7           8           9          10          11          12
##     "setosa" "versicolor" "versicolor" "versicolor" "versicolor" "versicolor"
##          13          14          15          16          17          18
## "versicolor" "versicolor" "versicolor" "versicolor" "versicolor" "versicolor"
##          19          20
## "versicolor" "versicolor"
```

On retrouve ici encore 7 setosas ainsi que 13 versicolors

Classification multi-classe

La classification multi-classe peut être utilisé à partir du moment ou 2 classes ou plus sont en jeu bien que généralement on ne l'utilisera que lorsque plus de 2 classes apparaissent. Ici nous utiliserons la méthode la plus simple afin d'effectuer une classification multi-classe : l'algorithme des K plus proches voisins (ou usuellement K-nn : K nearest neighbours) Nous utiliserons néanmoins toujours le dataset Iris mais avec les 3 types d'iris :

```
datasetmc <- iris
dtmc <- sample(nrow(datasetmc), round(nrow(datasetmc) * 0.7))
trainsetmc<- datasetmc[dtmc,]
testsetmc <- datasetmc[setdiff(1:nrow(datasetmc), dtmc),]
nrow(trainsetmc)
```

```
## [1] 105
```

```
nrow(testsetmc)
```

```
## [1] 45
```

```
y_trainmc <- factor(trainsetmc$Species)
x_trainmc <- cbind.data.frame(trainsetmc$Sepal.Length, trainsetmc$Sepal.Width, trainsetmc$Petal.Length,
y_testmc <- testsetmc$Species
x_testmc <-cbind.data.frame(testsetmc$Sepal.Length, testsetmc$Sepal.Width, testsetmc$Petal.Length, tests


Species = c(y_trainmc, y_testmc)



previsionmc <- knn(train = x_trainmc, test = x_testmc, cl = y_trainmc, k=3)
previsionmc
```

```
##  [1] setosa     setosa     setosa     setosa     setosa     setosa
##  [7] setosa     setosa     setosa     setosa     setosa     setosa
## [13] setosa     setosa     setosa     setosa     versicolor versicolor
## [19] versicolor versicolor versicolor versicolor virginica  versicolor
## [25] versicolor versicolor versicolor versicolor versicolor versicolor
## [31] versicolor virginica  virginica  virginica  virginica  virginica
## [37] versicolor virginica  virginica  virginica  virginica  virginica
## [43] virginica  virginica  virginica
## Levels: setosa versicolor virginica
```

```
summary(previsionmc)
```

```
##     setosa versicolor  virginica
##         16         15         14
```

```
summary(y_testmc)
```

```
##     setosa versicolor  virginica
##         16         15         14
```

```
previsionsmc <- knn(train = x_train, test = x_test, cl = y_train, k=3)
summary(previsionsmc)
```

```
##     setosa versicolor
##          7         13
```

```
summary(y_test)
```

```
##     setosa versicolor  virginica
##          7         13          0
```

On obtient dans notre cas 16 setosas, 15 versicolors et 14 virginicas grâce au knn ainsi que le même résultat lorsqu'on affiche le résultat sur ce dataset (y_testmc)

A titre de comparaison, j'affiche aussi le résultat du knn avec seulement 2 classes et j'obtiens les mêmes résultats que précédemment ce qui semble comfirmer nos résultats.

Mon Advisor c'est trouvé être Mme. Thiam Fatim qui m'a aidé pour certaines erreurs avec R ainsi que certaines erreurs de grammaire.