Supervised Methods

in

Methods to integrate multiple tables in biomedical studies to detect biomarkers and stratify individuals

Instituto de Salud Carlos III. Centro Nacional de Epidemiología
September, 2017

Juan R Gonzalez

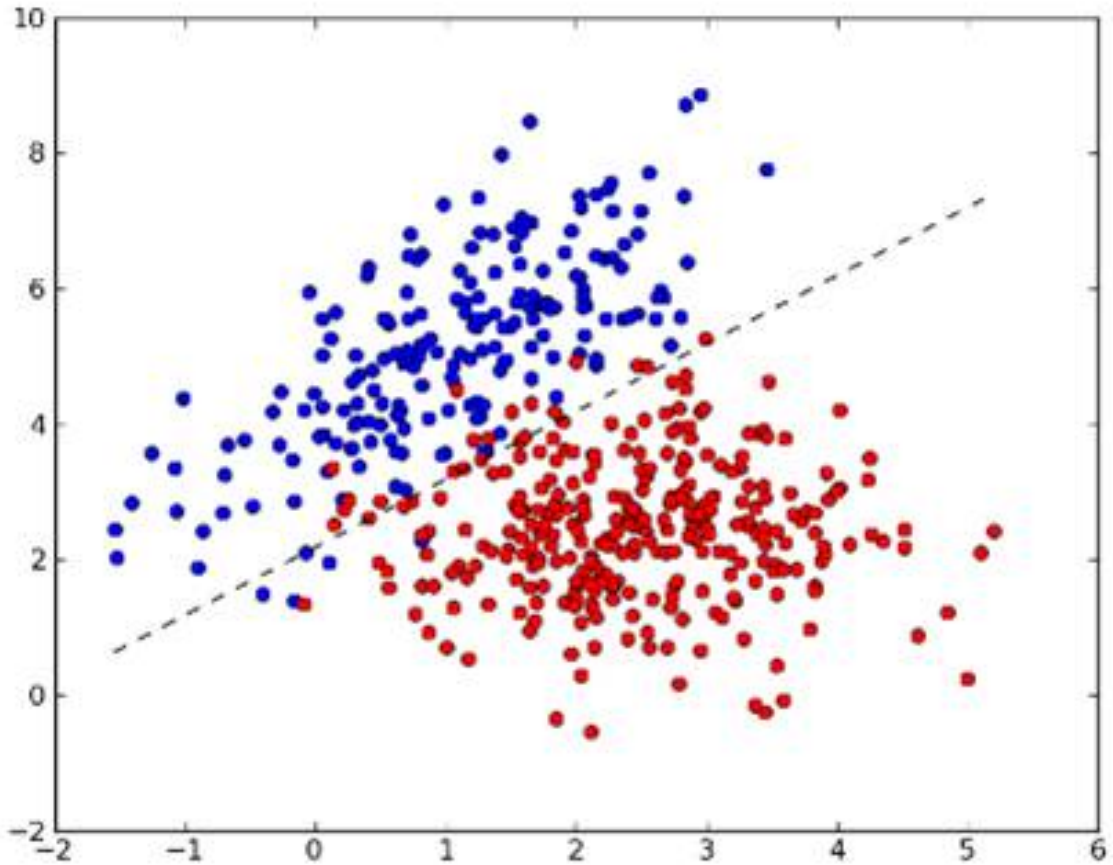
BRGE - Bioinformatics Research Group in Epidemiology
Barcelona Institute for Global Health (ISGlobal)
e-mail:juanr.gonzalez@isglobal.org
http://www.creal.cat/brge
and Departament of Mathematics, UAB

Outline

Outline

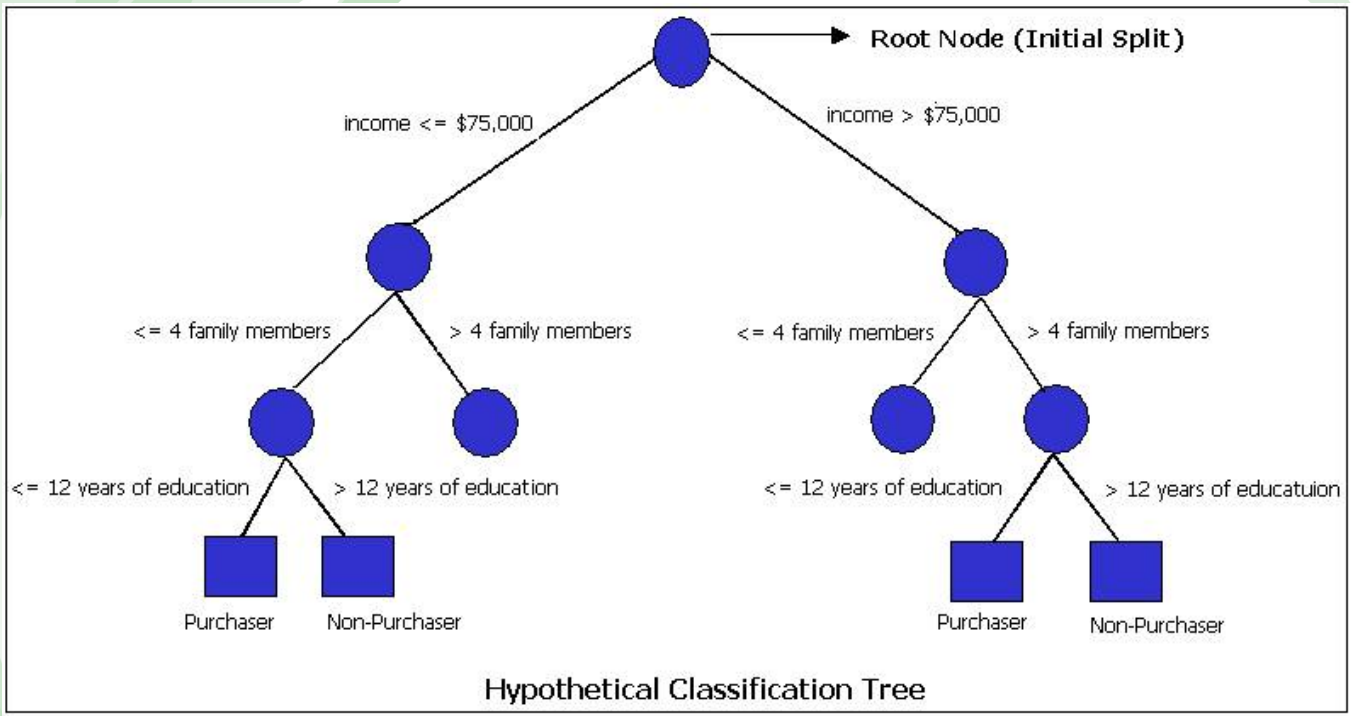
Supervised Methods

Logistic Regression: LR Uses a model to predict the probability of having one characteristic or not. Linear Discriminant Analysis (LDA) can be as an extension of LR (more than two categories in the outcome)



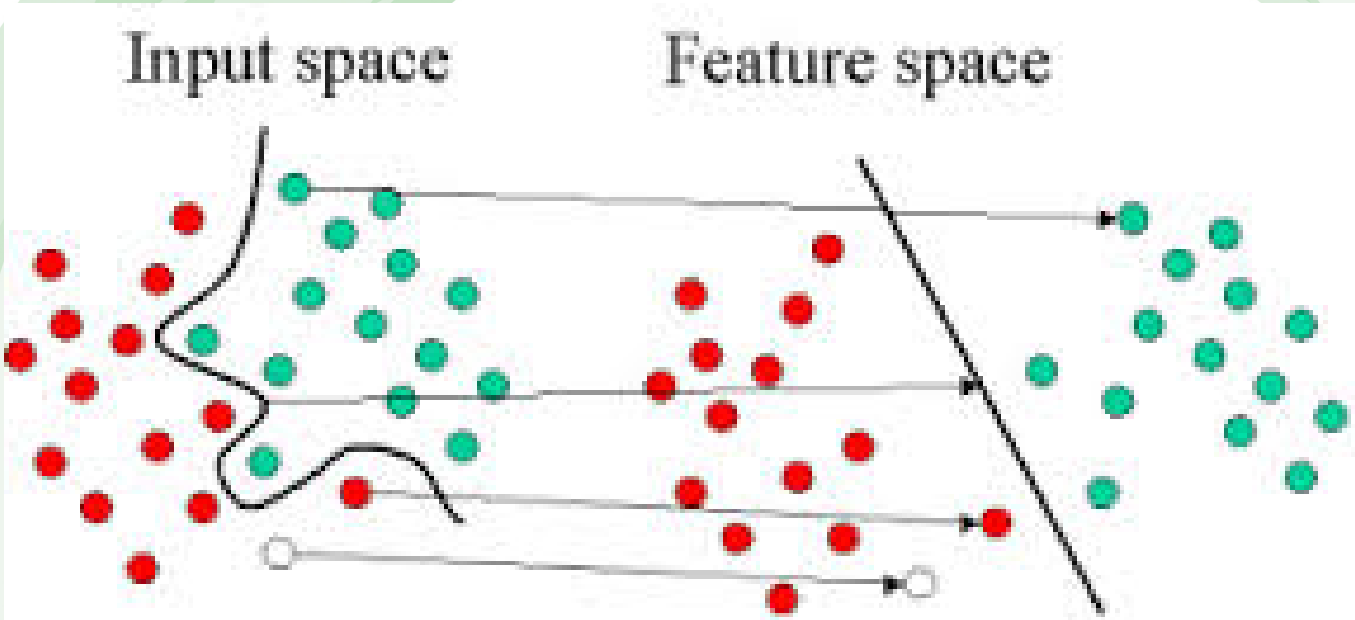
Supervised Methods

Classification Trees: A tree model resembles that of a linear model, where the criterion is the factor indicating class membership and the predictor variables are the observed values for each variable.



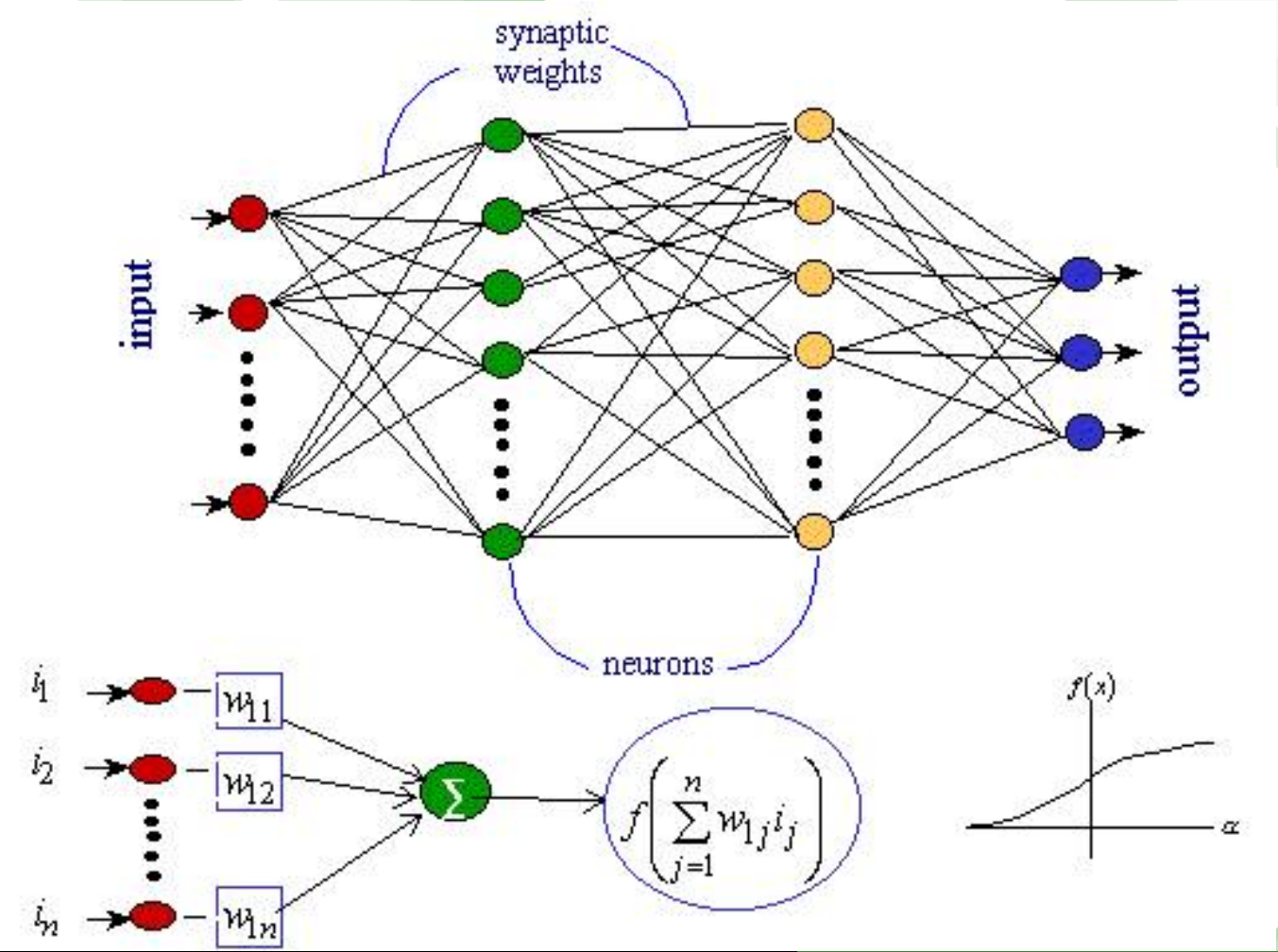
Supervised Methods

Support Vector Machine: SVM finds separating lines (hyper planes) between groups of points.



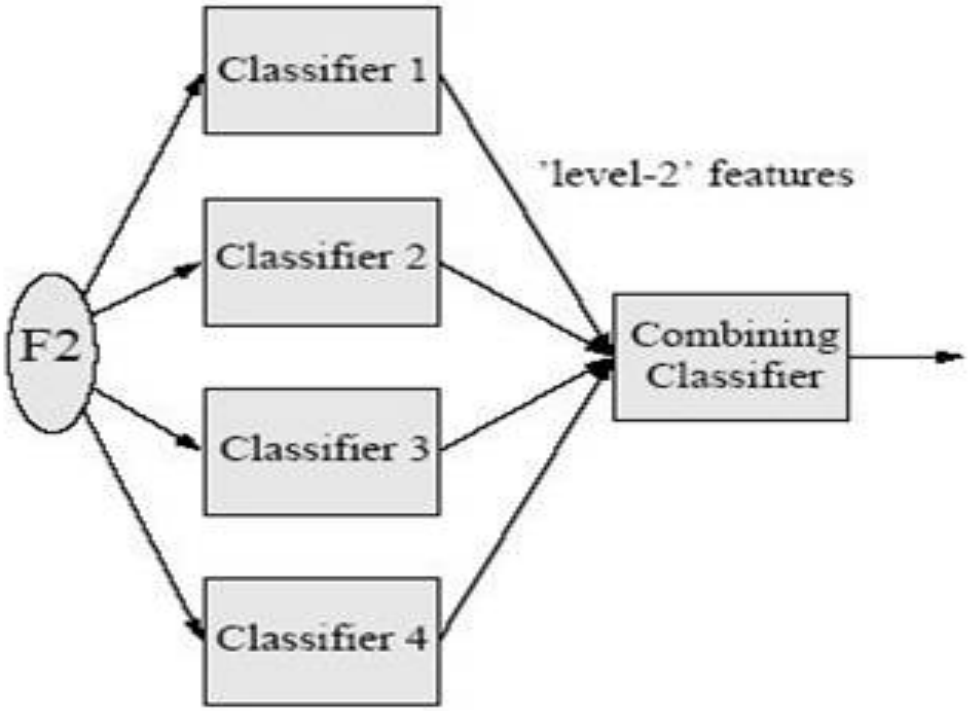
Supervised Methods

Neural Networks: Nonlinear models consisting of hyperplanes around classes of objects given a set of prediction variables finds separating lines (hyper planes) between groups of points.



Supervised Methods

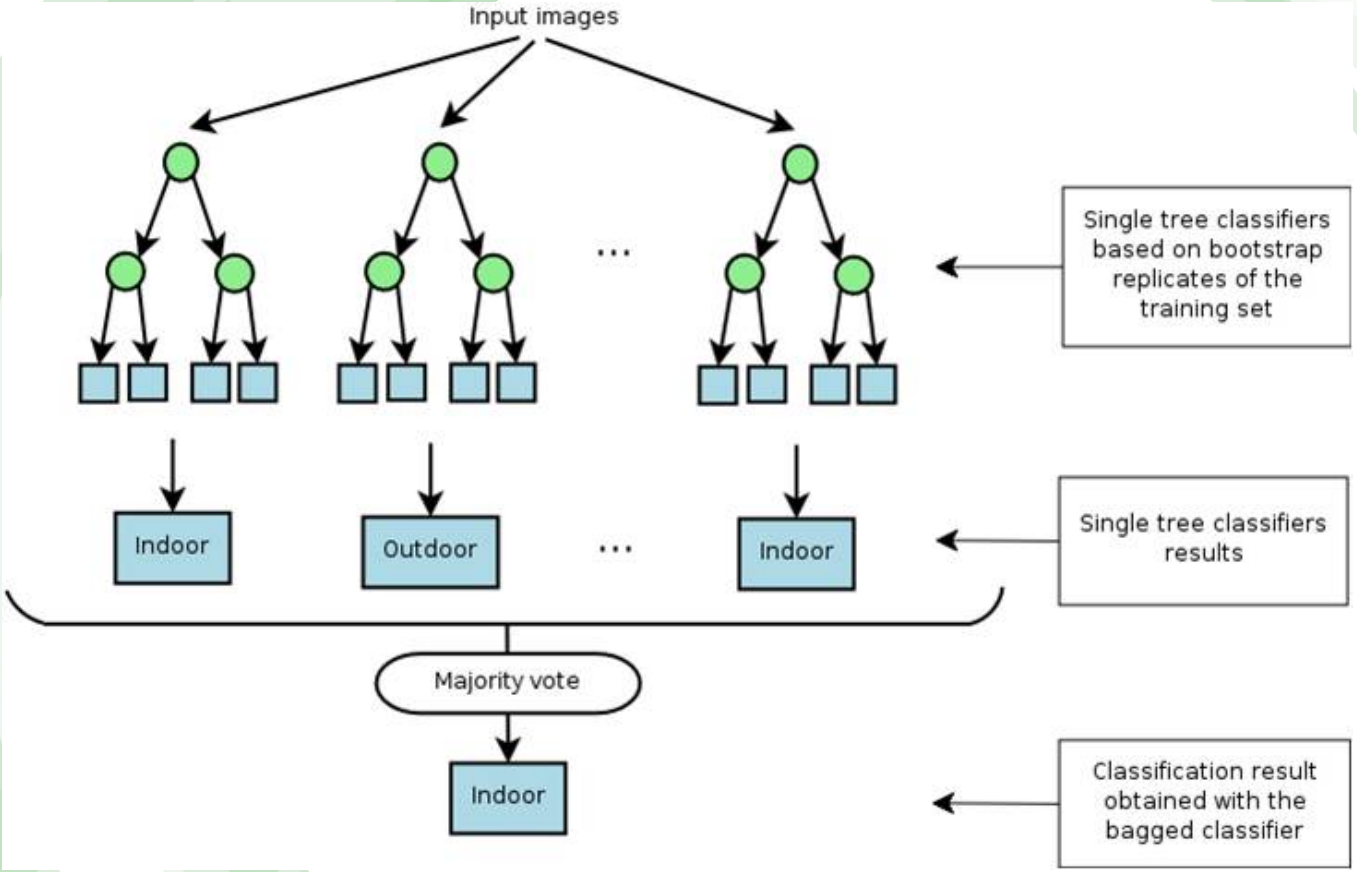
Boosting: Boosting is a combination of weak classifiers to produce a powerful committee.



Single feature set, different classifiers

Supervised Methods

Random Forest: It can be seen as an extension of Boosting when using trees as a classifiers.



Supervised methods

Example: `oliveoil` data set represents eight chemical measurements on different specimen of olive oil produced in various regions in Italy (northern Apulia, southern Apulia, Calabria, Sicily, inland Sardinia and coast Sardinia, eastern and western Liguria, Umbria) and further classifiable into three macro-areas: Centre-North, South, Sardinia.

```
library(pdfCluster)
data(oliveoil)
head(oliveoil)
```

```
##   macro.area       region palmitic palmitoleic stearic oleic linole
## 1      South Apulia.north     1075          75     226  7823      6
## 2      South Apulia.north     1088          73     224  7709      7
## 3      South Apulia.north      911          54     246  8113      5
## 4      South Apulia.north      966          57     240  7952      6
## 5      South Apulia.north     1051          67     259  7771      6
## 6      South Apulia.north      911          49     268  7924      6
##   arachidic eicosenoic
## 1        60         29
## 2        61         29
## 3        63         29
```

Supervised methods

```
set.seed(1234)
ss <- sample(1:nrow(oliveoil), 200)
train <- oliveoil[-ss,-2]
test <- oliveoil[ss,-2]
```

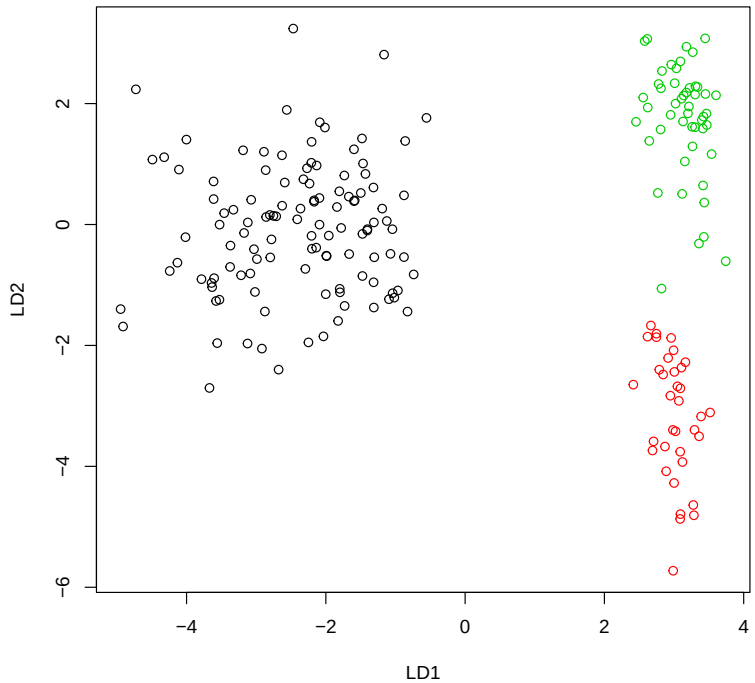
Linear discriminant analysis

```
library(MASS)
olive.lda <- lda(macro.area~., train)
pregion.lda <- predict(olive.lda, test)$class
table(test[,1], pregion.lda)

##                pregion.lda
##                 South Sardinia Centre.North
##   South           116        0            0
##   Sardinia          0       35            0
##   Centre.North      0        1           48
```

Linear discriminant analysis

```
plot(predict(olive.lda, test)$x,
     col=as.numeric(test[,1]))
```



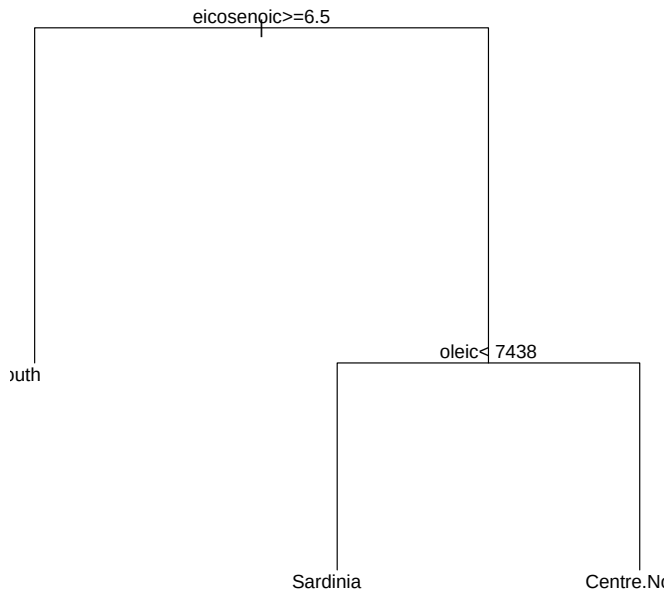
Classification Trees

```
library(rpart)
olive.rp <- rpart(macro.area~., train,
          method="class")
olive.rp

## n= 372
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
## 1) root 372 165 South (0.5564516 0.1693548 0.2741935)
##   2) eicosenoic>=6.5 207   0 South (1.0000000 0.0000000 0.0000000)
##   3) eicosenoic< 6.5 165  63 Centre.North (0.0000000 0.3818182 0.61
##     6) oleic< 7438.5 63   0 Sardinia (0.0000000 1.0000000 0.0000000
##     7) oleic>=7438.5 102   0 Centre.North (0.0000000 0.0000000 1.00
```

Classification Trees

```
plot(olive.rp)
text(olive.rp)
```



Linear discriminant analysis

```
temp <- predict(olive.rp, test)
head(temp)

##     South Sardinia Centre.North
## 66      1        0            0
## 356     0        1            0
## 348     0        1            0
## 355     0        1            0
## 489     0        0            1
## 364     0        1            0

pregion.rp <- apply(temp, 1, function(x) which(x==1))
```

Linear discriminant analysis

```
table(test[,1], pregion.rp)

##                pregion.rp
##                   1   2   3
##   South         116   0   0
##   Sardinia        0  34   1
##   Centre.North    0   2  47
```

Support Vector Machine

```
library(e1071)
olive.svm <-  svm(macro.area ~. , data = train)
pregion.svm <- predict(olive.svm, test)
table(test[,1], pregion.svm)

##                pregion.svm
##                 South Sardinia Centre.North
##   South           116        0            0
##   Sardinia          0       35            0
##   Centre.North      0        0           49
```

Neural Network

```
library(nnet)
olive.nnet <-  nnet(macro.area ~. , data = train,
                    size=2)

## # weights:  27
## initial  value 463.503444
## final  value 365.191009
## converged

pregion.nnet <- predict(olive.nnet, test, type="class")
table(test[,1], pregion.nnet)

##                pregion.nnet
##                 South
##   South           116
##   Sardinia         35
##   Centre.North     49
```

Neural Network

```
olive.nnet <-  nnet(macro.area ~. , data = train,
               size=4)

## # weights:  51
## initial  value 539.582101
## final  value 365.191009
## converged

pregion.nnet <- predict(olive.nnet, test, type="class")
table(test[,1], pregion.nnet)

##                pregion.nnet
##                 South
##   South           116
##   Sardinia         35
##   Centre.North     49
```

Boosting

```
library(adabag)
olive.boost <-  boosting(macro.area ~. , data = train,
                control = rpart.control(maxdepth = 2))
pregion.boost <- predict(olive.boost, test, type="class")$class
table(test[,1], pregion.boost)

##                pregion.boost
##                 Centre.North Sardinia South
##    South                   0        0   116
##    Sardinia                1       34     0
##    Centre.North           47        2     0
```

Random Forest

```
library(randomForest)
olive.rf <-  randomForest(macro.area ~. , data = train)
pregion.rf <- predict(olive.rf, test, type="class")
table(test[,1], pregion.rf)

##                pregion.rf
##                 South Sardinia Centre.North
##   South           116        0            0
##   Sardinia          0       35            0
##   Centre.North      0        0           49
```

Outline

Model performance

- Rand Index (categorical biomarker)
- ROC curve (continuous biomarker)

Model performance

Rand Index: used in the class prediction problem

```
library(flexclust)
randIndex(table(test[,1], pregion.rf))

## ARI
##   1

randIndex(table(test[,1], pregion.lda))

##       ARI
## 0.9914602

randIndex(table(test[,1], pregion.rp))

##       ARI
## 0.9749979

randIndex(table(test[,1], pregion.boost))

##       ARI
## 0.9749979
```

Model performance

Let us assume that we want to use different biomarkers (continuous) to predict and outcome. For instance, researchers want to use several clinical and one laboratory variable to predict 6-month outcome (Good and Poor) after having an aneurysmal subarachnoid haemorrhage (aSAH). These are the variables the collected at hospital admission

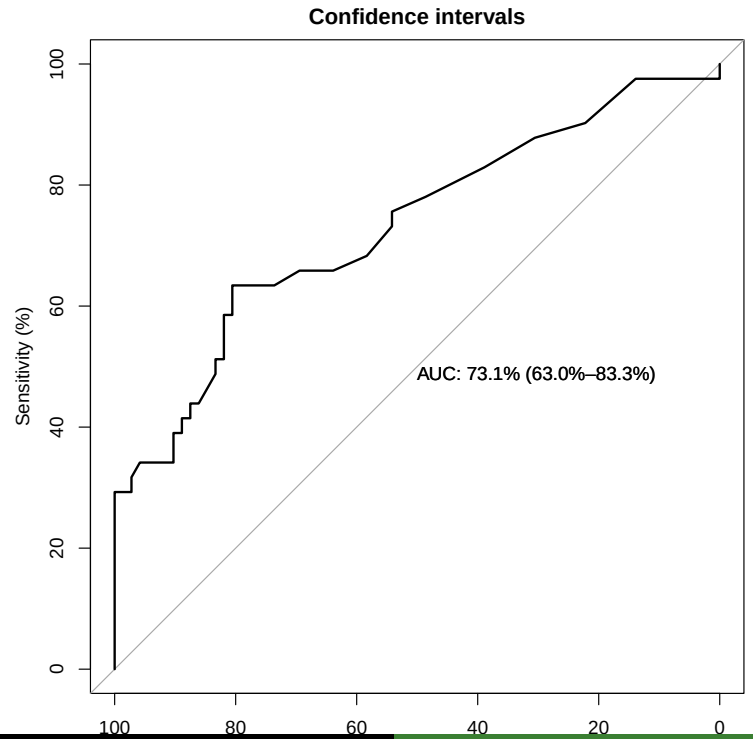
```
library(pROC)
data(aSAH)
head(aSAH)
```

```
##    gos6 outcome gender age wfns s100b  ndka
## 29    5    Good Female  42    1  0.13  3.01
## 30    5    Good Female  37    1  0.14  8.54
## 31    5    Good Female  42    1  0.10  8.09
## 32    5    Good Female  27    1  0.04 10.42
## 33    1    Poor Female  42    3  0.13 17.40
## 34    1    Poor   Male  48    2  0.10 12.75
```

Model performance

Let us assume that we want to compute the AUC and its confidence interval for a given biomarker

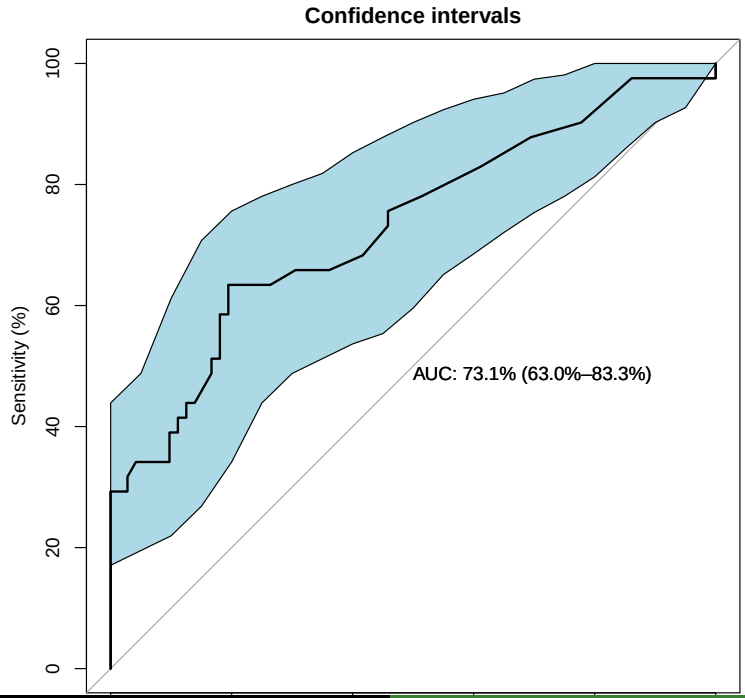
```
rocobj <- plot.roc(aSAH$outcome, aSAH$s100b,
                   main="Confidence intervals",
                   percent=TRUE,
                   ci=TRUE,
                   print.auc=TRUE)
```



Model performance

A confidence band can be added

```
rocobj <- plot.roc(aSAH$outcome, aSAH$s100b,
                   main="Confidence intervals", percent=TRUE,
                   ci=TRUE,
                   print.auc=TRUE)
ciobj <- ci.se(rocobj, progress = "none",
                  specificities=seq(0, 100, 5)) #This can be selected (gr
plot(ciobj, type="shape", col="lightblue") # plot as a blue shape
```



Model performance

Two biomarkers can be compared by

```
rocobj1 <- plot.roc(aSAH$outcome, aSAH$s100,
                    main="Statistical comparison", percent=TRUE, col="
rocobj2 <- lines.roc(aSAH$outcome, aSAH$ndka, percent=TRUE, col="red")
testobj <- roc.test(rocobj1, rocobj2)
text(50, 50, labels=paste("p-value =", format.pval(testobj$p.value)),
legend("bottomright", legend=c("S100B", "NDKA"), col=c("blue", "red"),
```