Algoritmes de classificació

Cas pràctic Santa Coloma de Queralt

Pere Gelabert

Table of contents

Breu introducció a R

R és un llenguatge de programació i un entorn dissenyat per al càlcul estadístic. Facilita l'ús de moltes tècniques estadístiques (R Core Team, 2016). Amb R, pots escriure codi per crear nous mètodes, definir funcions o automatitzar tasques repetitives.

R inclou una àmplia varietat de mètodes estadístics, des dels enfocaments tradicionals fins a les tècniques més avançades. Els usuaris poden explorar i triar els paquets que millor s'adaptin a les seves necessitats.

Es pot considerar R com un conjunt d'eines per treballar amb dades, càlculs i gràfics.

En canvi **RStudio** és una interfície gràfica que ens facilita treballar en R

Molt important: RStudio no és R

Objectes

Com s'acaba de mencionar, R és un llenguatge orientat a objectes. Un objecte és un **contenidor** on s'hi enmagatzema informació (una taula, un ràster, un objecte vectorial, una llista, un vector...) i es pot declarar de les següents maneres:

```
nom = "Pere"
nom <- "Pere"
```

Errors, warnings i missatges

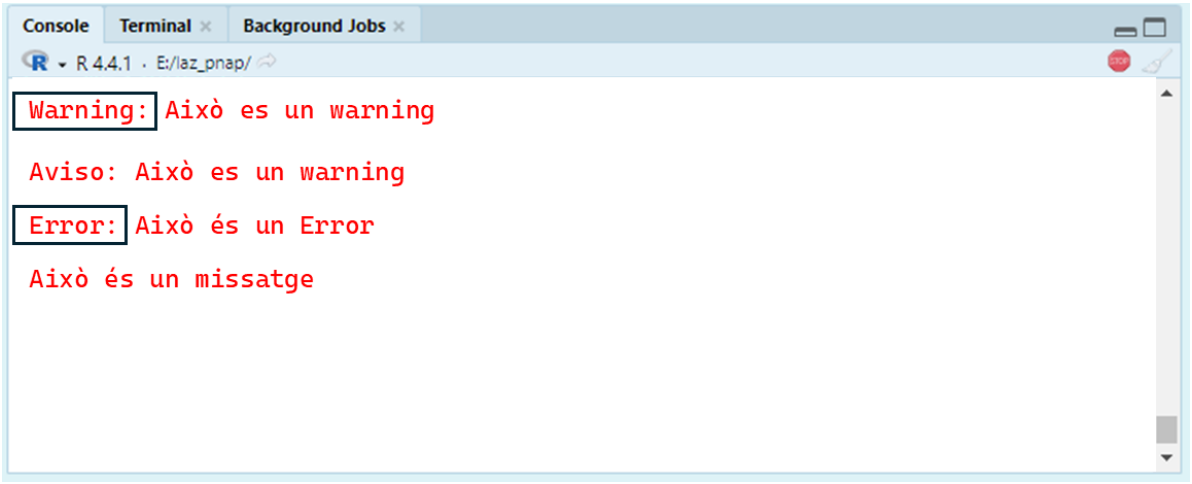


Figure 1: Font: Elaboració pròpia

Funcións

Una funció és un comando compost per arguments -separats per ,- que executa una acció concreta. Per exemple si volem saber el valor màxim dins d'un vector de números hem d'aplicar la funció `max()`.

```
max(c(2,4,5,6,7,4,23,22,11,1,8))
```

```
[1] 23
```

Exercici de classificació

Càrrega de llibreries

Un paquet és un repositori de funcions concretes, per realitzar certes accions, creades per algú i validades per la comunitat.

A la següent peça de codi carregarem les següents llibreries. No obstant, si és el primer cop que les feu servir, prèviament s'ha d'executar el comando 'install.packages("nom de la llibreria")'

```r
library(tidyverse)
library(terra)
library(caret)
library(gtools)
library(sf)
library(tidyverse)
library(tmap)
```

Dades

Imatge multi espectral de l'àrea de *l'incendi de Santa Coloma de Queralt*. En aquest cas pretenem classificar les diferents cobertes del sòl existents abans de l'incendi per tal d'avaluar quins usos han estat més afectats i sota quina severitat.

Com a imatge de referència tenim la imatge capturada pel **Sentinel 2 el dia 27 de juliol de 2021.**. Aquesta imatge està conté les següents bandes i resolució nativa:

- **B2:** 10 m - 490 nm - **Blue**
- **B3:** 10 m - 560 nm - **Green**
- **B4:** 10 m - 665 nm - **Red**
- **B8:** 10 m - 842 nm - **Visible and Near Infrared (VNIR)**
- **B11:** 20 m - 1610 nm - **Short Wave Infrared (SWIR)**

```r
imatge <- rast("./Santa Coloma/S2_21072021_preFire.tif")
```

Objectiu

L'objectiu de la següent pràctica és predir les cobertures del sòl utilitzant mètodes de classificació supervisada i no supervisada. Posteriorment en derivarem els resultats estadístics descriptius pertinents.

Classificació

La classificació d'imatges és una seqüència lògica en la qual es busca extreure informació temàtica d'una imatge *ràster* i el resultat serveix per a crear cartogràfia categòrica.

Classificació no supervisada

La classificació no supervisada (CNS) també coneguda com "*clustering*", utilitza les propietats i moments de la distribució estadística dels píxels en un espai característic de bandes espectrals per a fer una diferenciació entre grups relativament similars.

La CNS és una forma eficaç per a les imatges preses per sensors remots en un espai amb característiques espectrals, per a extreure informació útil sobre la cobertura terrestre. Aquesta classificació utilitza clústers coneguts com a tècniques de reducció de dades on es comprimeix la Informació diversa dels píxels en grups que tenen valors similars.

A diferència de la classificació supervisada la CNS no necessita que l'usuari ingressi dades d'entrenament. La CNS requereix únicament definir el nombre de grups (clúster) i les bandes que es van a utilitza, sense necessitat d'altres entrades.

Avantatges

- ***No es necessita entrenament***
- ***Abstracció/Sense biaix interpretatiu humà***

Desavantatges

- ***Desconeixement del n. de categories***
- ***Interpretació del que representa cada categoria***
- ***Major cost computacional***

Algoritme de classificació Kmeans

És un algoritme molt conegut i utilitzat per la seva eficiència i robustesa, el seu nom fa referència al numero K de classes o grups a buscar

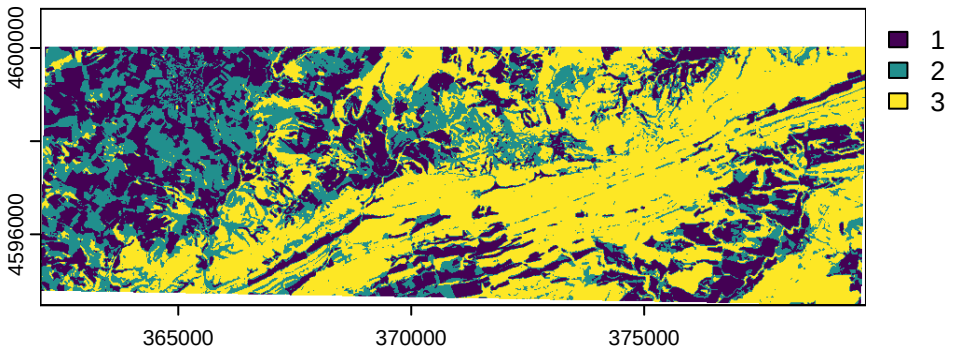
És una tècnica no jeràrquica de partició on:

- Selecciona K objectes a l'atzar del conjunt total i s'assignen com a patró o centroides de les K classes que es buscaran (prèviament definides).
- Calcula totes les distàncies euclidianes de tots els objectes restants als K centroides, i s'assigna la pertinença a cada objecte al clúster que tingui mes pròxim.
- Es recalcula el centroide de cada clúster com la mesura de tots els objectes que el componen, buscant minimitzar el valor d'una funció de cost, que és la sumatòria de totes les distàncies euclidianes dels objectes de cada classe al centroide de la seva respectiva classe.

- El pas 2 i 3 es repeteixen successivament fins que els centres de tots els grups romanguin constants o fins que es compleixi alguna altra condició de parada.

Visualització cluster kmeans

```
## Crear un vector de dades
dades <- values(imatge)

## cercar els NA´s a les dades
idx <- complete.cases(dades)

## Executar algoritme Kmeans
km <- kmeans(dades[idx,], centers = 3, iter.max = 50)

## Crear un vector per representar els resultats
kmClust <- vector(mode = "integer", length = ncell(imatge))

## Tabulació de resultats
kmClust[!idx] <- NA
kmClust[idx] <- km$cluster

## Espacialitzaciño de resultats
raster_output <- rast(imatge[[1]])
values(raster_output) <- kmClust
raster_output <- as.int(raster_output) %>% as.factor()
## Visualització de resultats
plot(raster_output)
```




```
## Per exportar i visualitzar a QGIS
# writeRaster(raster_output, "C:/.../nom.tif")
```

Classificació supervisada

La classificació supervisada, utilitza dades d'entrenament que son variables predictors mesurades en cada unitat de mostreig i assigna classes prèvies al mostreig. Al següent diagrama es mostren els passos a realitzar en una classificació supervisada:

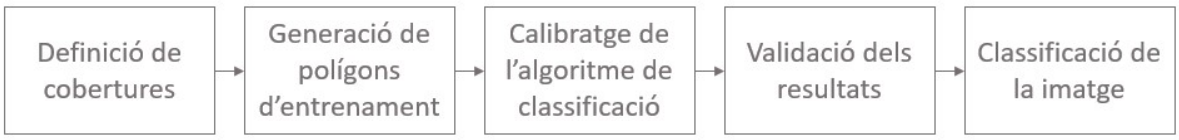


Figure 2: Font: Elaboració pròpia

Els algoritmes de *Machine Learning* ofereixen la possibilitat de crear una classificació eficaç i eficient d'imatges satèl · litals, entre els algoritmes més coneguts es troben: *Arbres de decissió (CART), random forest, Support Vector Machine i xarxes neuronals.*

Avantatges

- **Coneixement de les categories**
- **Optimització de les àrees d'entrenament**

Desavantatges

- **Requereix d'un esforç entrenament previ**

Random forest

Random Forest (RF) és un algoritme de machine learning que es fa servir per a tasques de classificació (i també de regressió). Ara t'explico com funciona amb exemples senzills:

Què és un arbre de decisió?

Un arbre de decisió és un model que pren decisions fent preguntes successives sobre les dades. Per exemple:

- Pregunta 1: És una fruita verda? (Sí/No)
- Pregunta 2: És dolça? (Sí/No) Amb aquestes preguntes, l'arbre arriba a una "decisió" final, com ara: *Aquesta fruita és una poma.*

El problema? Els arbres de decisió són molt sensibles: si canvies una mica les dades, l'arbre pot donar resultats completament diferents.

Com soluciona això el Random Forest?

RF no fa servir un sol arbre, sinó molts arbres alhora. La idea clau és aquesta:

- **Genera múltiples arbres de decisió**, cadascun creat amb una petita mostra aleatòria de les dades d'entrenament (això es diu *bagging*, o *bootstrap aggregating*).
- Cada arbre fa una predicció (com si fos un vot). En classificació, el Random Forest tria el resultat més votat per tots els arbres.

Pensa-ho com si preguntessis a un grup d'amics què opinen sobre alguna cosa. Si la majoria diu "És una poma", confiaràs en la seva opinió més que en la d'una sola persona.

Per què funciona bé?

- **Redueix l'error**: Com que fa servir moltes mostres i arbres diferents, el RF evita que els errors d'un sol arbre afectin el resultat final.
- **Evita el sobreajust**: Això vol dir que no es "memoritzarà" els detalls exactes de les dades d'entrenament, sinó que generalitzarà millor.

Paràmetres importants

Perquè RF funcioni, cal ajustar alguns paràmetres:

- **mtry**: Quantes variables considerarà cada arbre per fer les seves preguntes.
- **Mida mínima dels nodes**: Quantes dades ha de tenir un node per continuar fent divisions. Si és massa petit, l'arbre pot fer divisions inútils.

Exemple

Creació d'arbres dins el Random Forest

Suposem que construïm **5 arbres de decisió** (per simplificar) i cada arbre utilitza diferents combinacions de les variables:

Arbre 1:

- Mira el **Grau d'enmarronament de les fulles**: Si és >50%, diu "Malalt".
- Si és 50%, mira si té **veïns infectats**: Si "Sí", diu "Malalt". Si "No", diu "Salut".

Arbre 2:

- Mira la **Quantitat de fulles verdes**: Si és >60%, diu "Salut".
- Si és 60%, mira el **Grau d'enmarronament**: Si és >30%, diu "Malalt". Si no, "Salut".

Arbre 3:

- Mira si té **veïns infectats**: Si "Sí", diu "Malalt".
- Si "No", mira la **Quantitat de fulles verdes**: Si és >70%, diu "Salut". Si no, diu "Malalt".

Arbre 4:

- Mira el **Grau d'enmarronament**: Si és >40%, diu "Malalt".
- Si és 40%, mira si té **veïns infectats**: Si "No", diu "Salut". Si "Sí", diu "Malalt".

Arbre 5:

- Mira la **Quantitat de fulles verdes**: Si és >50%, diu "Salut".
- Si és 50%, diu "Malalt".

Exemple de classificació

Un arbre (*real no dedecisió*) amb aquestes característiques:

- **Quantitat de fulles verdes:** 55%.
- **Grau d'enmarronament de les fulles:** 45%.
- **Veïns infectats:** Sí.

Cada arbre fa la següent predicció:

- **Arbre 1**: **Malalt** (veïns infectats).
- **Arbre 2**: **Salut** (55% de fulles verdes, grau d'enmarronament no arriba a 50%).
- **Arbre 3**: **Malalt** (té veïns infectats).
- **Arbre 4**: **Malalt** (enmarronament >40% i veïns infectats).
- **Arbre 5**: **Salut** (55% de fulles verdes).

Decisió final del Random Forest

El Random Forest fa una **votació majoritària**:

- **Malalt:** 3 arbres (Arbre 1, Arbre 3, Arbre 4).
- **Salut:** 2 arbres (Arbre 2, Arbre 5).

Com que la majoria d'arbres diu **"Malalt"**, el Random Forest classifica aquest arbre com a **Malalt**.

Potencials del *Random Forest*?

- Cada arbre treballa amb només 2 variables a cada pas (perquè `mtry = 2`), per tant són més diversos i menys susceptibles a fer errors similars.
- Amb molts arbres, el Random Forest combina totes les opinions i evita errors greus d'un arbre individual.

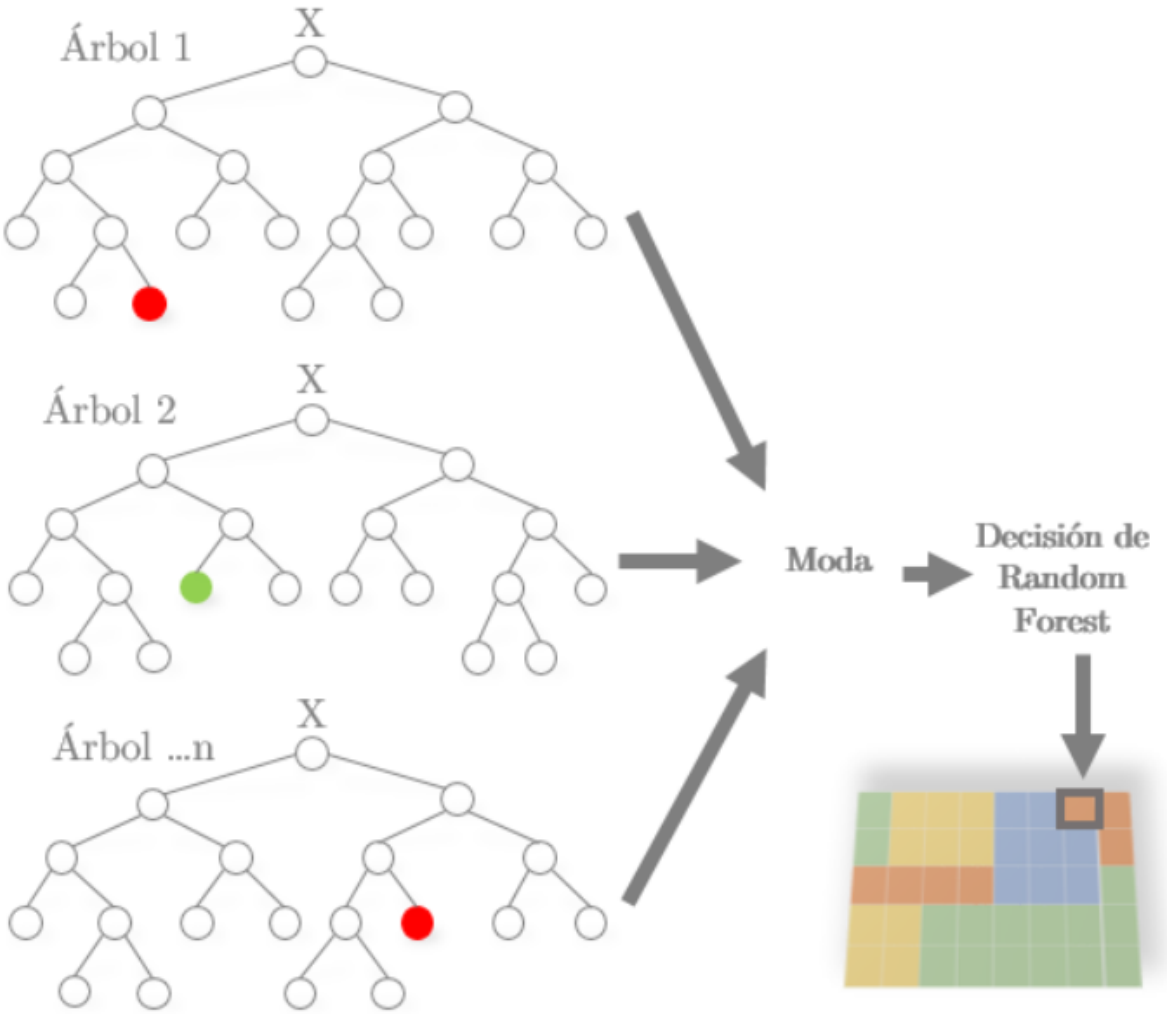


Figure 3: Font: Elaboració pròpia

Com s'ha mencionat abans, el RF té una sèrie `d'hiperparametres`. Els `hiperparametres` son aquells parametres que l'algoritme no pot aprendre de les dades i que s'han d'entrenar mitjançant altres mètodes -e.g. validació creuada.

- `mtry:` Nombre de variables que entren a cada arbre (per defecte `sqrt(nvars)` i si es testeja els valors a considerar oscil · len entre 1 i `nvars -1`
- `min node size:` Nombre d'observacions mímimes per tancar l'arbre.

Mitjançant validació creuada repetida s'iteraran diferents submostres per tal d'obtenir la hiperarametrització òptima. La validació creuada crea X particions i selecciona X-1 submostres per entrenar i la mostra restant per validar. Itera tants cops com submostres hem especificat.

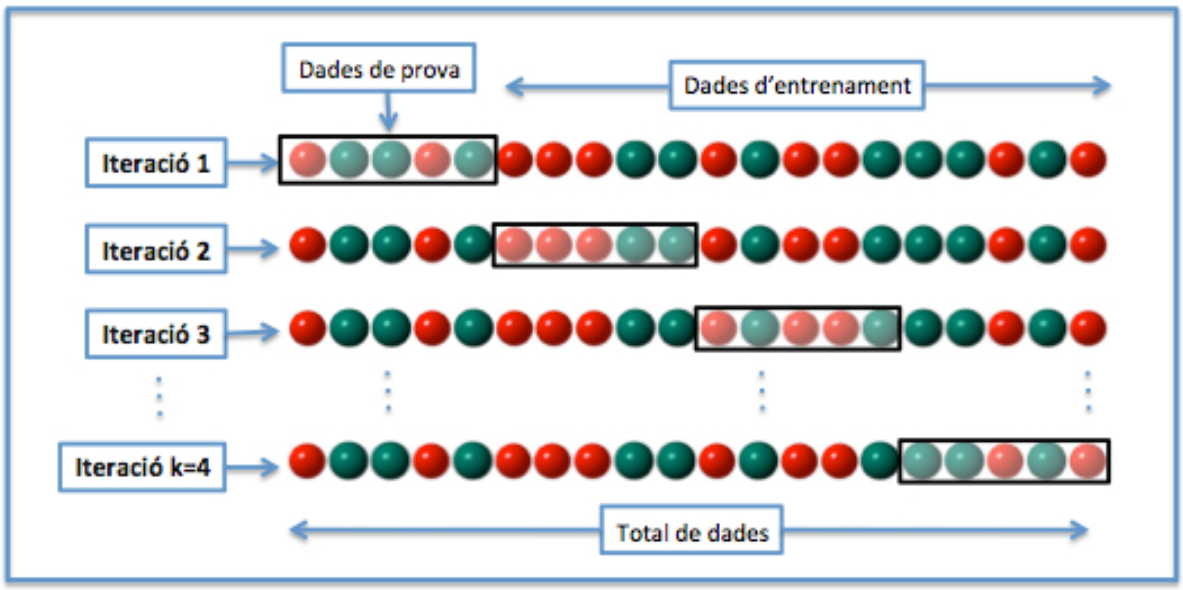


Figure 4: Font: Viquipèdia

Cas pràctic Santa Coloma de Queralt

```
# Carregar les dades d'entrenament
train_shp <- st_read("./Santa Coloma/Poligons_entrenament/Poligons.shp") %>% st_transform(32
```

```
Reading layer `Poligons' from data source
  `D:\OneDrive - udl.cat\01 - Docencia\102448 - Tècniques Avançades de Diagnòstic (DGEFCN)\24
  using driver `ESRI Shapefile'
Simple feature collection with 101 features and 2 fields
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box:  xmin: 362130.8 ymin: 4594793 xmax: 378545.7 ymax: 4600030
Projected CRS: ETRS89 / UTM zone 31N
```

```
#Establir els noms de les bandes
names(imatge) <- c("B2","B3","B4","B8","B11","B12")

#Càlcul NDVI com a nova informació
ndvi <- (imatge$B8-imatge$B4)/(imatge$B8+imatge$B4)
names(ndvi) <- "NDVI"
imatge <- c(imatge,ndvi)

#mascara dels valors d'entrenament per convertir-los a punts
imatge_mask <- mask(imatge,train_shp)

# Punts d'entrenament
punts <- as.points(imatge_mask) %>% st_as_sf()

#Punts amb tota la informació temàtica
punts_def <- st_intersection(punts,train_shp)%>%
  dplyr::select(-Id)
```

```
Warning: attribute variables are assumed to be spatially constant throughout
all geometries
```

```
#Creació de la mostra d'entrenament
train <- punts_def%>% st_drop_geometry() %>%
   sample_n(3000)

# Creació de la mostra de validació
test <- punts_def%>%
   anti_join(train) %>%
  sample_n(2000)
```

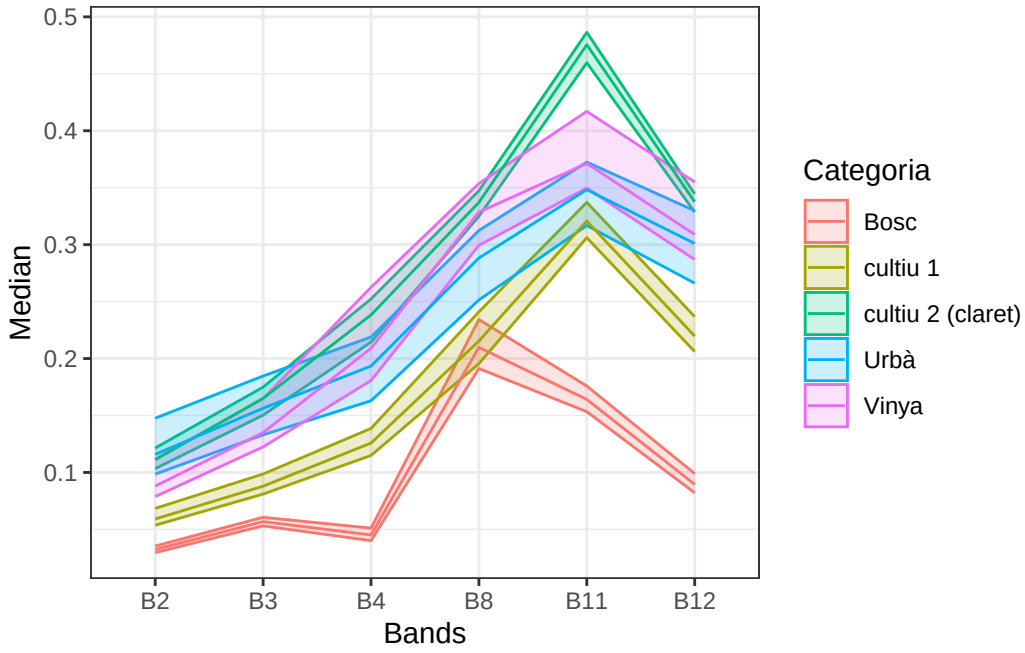
```
Joining with `by = join_by(B2, B3, B4, B8, B11, B12, NDVI, Categoria)`
```

```
# Visualització de la distrivució de les dades
train %>%
  dplyr::select(starts_with("B"),
                Categoria) %>%
  pivot_longer(!Categoria,
               names_to = "Bands",
               values_to = "value") %>%
  drop_na() %>%
  group_by(Categoria,
```

```
          Bands) %>%
summarise(Median = median(value, na.rm = T),
          q1 = quantile(value, 0.25),
          q3 = quantile(value,

mutate(Bands = factor(Bands),
       Bands = factor(Bands,
                      levels =mixedsort(levels(Bands))))%>%


ggplot(aes(group = Categoria,
           color = Categoria,
           fill = Categoria)) +
geom_ribbon(aes(x = Bands,ymax = q3,
                ymin = q1),
            alpha = 0.2) +
geom_line(aes(x = Bands,
              y = Median)) +
theme_bw()
```

0.75

```
`summarise()` has grouped output by 'Categoria'. You can override using the
`.groups` argument.
```



```r
# Definició de la Validació creuada repertida
particions  <- 10
repeticions <- 5


# # Hiperparàmetres
hiperparametres <- expand.grid(
  mtry = c(1, 3, 5, 7),
  min.node.size = c(100, 50, 25, 10),
  splitrule = "gini"
)


# Definició de l'entrenament
control_train <- trainControl(
  method = "repeatedcv",
  number = particions,
  repeats = repeticions,
  returnResamp = "final",
  verboseIter = TRUE,
  allowParallel = TRUE
)
```

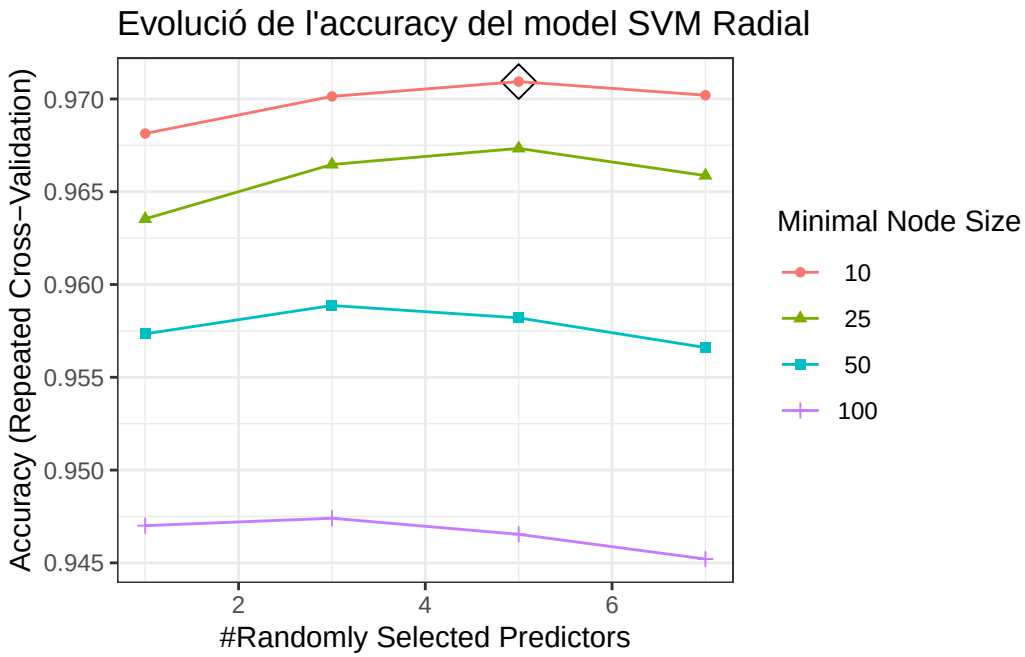
```r
# Ajust del model de classificació
model_rf <- train(
  Categoria ~ .,
  data = train,
  method = "ranger",
  tuneGrid = hiperparametres,
  metric = "Accuracy",
  trControl = control_train
)
```

```r
print(model_rf)
```

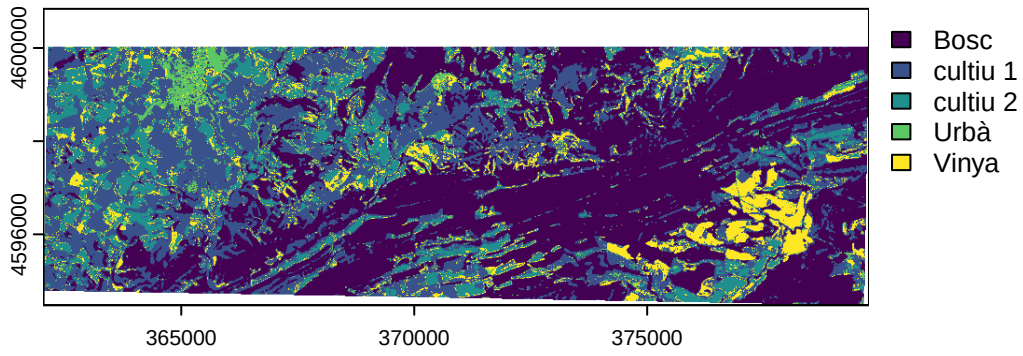
```
Random Forest

3000 samples
   7 predictor
   5 classes: 'Bosc', 'cultiu 1', 'cultiu 2 (claret)', 'Urbà', 'Vinya'
```

```
No pre-processing
Resampling: Cross-Validated (10 fold, repeated 5 times)
Summary of sample sizes: 2700, 2699, 2700, 2699, 2700, 2699, ...
Resampling results across tuning parameters:

  mtry  min.node.size  Accuracy   Kappa
  1      10            0.9681356  0.9487444
  1      25            0.9635367  0.9413173
  1      50            0.9573397  0.9312181
  1     100            0.9470043  0.9146444
  3      10            0.9701376  0.9519954
  3      25            0.9664673  0.9460849
  3      50            0.9588700  0.9337567
  3     100            0.9474039  0.9153688
  5      10            0.9709349  0.9532635
  5      25            0.9673349  0.9474762
  5      50            0.9582028  0.9326632
  5     100            0.9465376  0.9140249
  7      10            0.9702016  0.9520729
  7      25            0.9658693  0.9450921
  7      50            0.9566024  0.9300655
  7     100            0.9452054  0.9118882

Tuning parameter 'splitrule' was held constant at a value of gini
Accuracy was used to select the optimal model using the largest value.
The final values used for the model were mtry = 5, splitrule = gini
 and min.node.size = 10.
```

```r
ggplot(model_rf, highlight = TRUE) +
  labs(title = "Evolució de l'accuracy del model SVM Radial") +
  theme_bw()
```




```
# Espacialització del model
RF_class <- predict(imatge,model_rf, na.rm=T)


# Representació del resultat

plot(RF_class)
```




```
# Guardar el raster si escau
# writeRaster(raster,"C:/Users/Pere/Desktop/EX4.tif")
```

Validació

Per validar els nostres resultats compararem els valors observats i predits als punts de la mostra test.

Com a estadístics farem, servir:

Precissió Global: És refereix a la probabilitat de que una observació estigui classificada correctament

$$OA = \frac{TP + TN}{TN + TP + FP + FN}$$

On:

- **TP:** és el nombre de valors vertaders positius.
- **TN:** és el nombre de valors vertaders negatius.
- **FP:** és el Nombre de valors fals positius.

- **FN:** és el Nombre de valors fals negatius.

Index Kappa: És una mesura estadística - per a elemeents qualitatius (variables categòriques) - que ajusta el l'efecte de l'azar en la proporció de la concordància observada.

$$k = \frac{Pr_{(a)} - Pr_{(e)}}{1 - Pr_{(e)}}$$

On:

- $Pr_{(a)}$ = Probabilitat valors classificats correctament sense tenir en compte l'efecte de l'atzar.
- $Pr_{(e)}$ = Probabilitat valors classificats correctament tenint en compte l'efecte de l'atzar.

```
# Extracció de dades del raster als punts test
categories <- levels(RF_class)[[1]]
ext_res <- extract(RF_class,test) %>% left_join(categories)
```

```
Joining with `by = join_by(class)`
```

```
# Dataset de validació
validation <- data.frame(cbind(ext_res$class, test$Categoria))

colnames(validation) <- c("Pred","Obs")

# Conversió a factors
validation$Pred <- factor(validation$Pred)
validation$Obs <- factor(validation$Obs)

confusionMatrix(validation$Pred,validation$Obs)
```

```
Confusion Matrix and Statistics

                  Reference
Prediction          Bosc cultiu 1 cultiu 2 (claret) Urbà Vinya
  Bosc               981        5                 0    0     0
  cultiu 1             2      714                 0    8     3
  cultiu 2 (claret)    0        2                39    0     1
  Urbà                 0        8                 0   51     2
  Vinya                0        6                 2    7   169
```

```
Overall Statistics

               Accuracy : 0.977
                 95% CI : (0.9694, 0.9831)
    No Information Rate : 0.4915
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9626

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Bosc Class: cultiu 1 Class: cultiu 2 (claret)
Sensitivity               0.9980          0.9714                   0.9512
Specificity               0.9951          0.9897                   0.9985
Pos Pred Value            0.9949          0.9821                   0.9286
Neg Pred Value            0.9980          0.9835                   0.9990
Prevalence                0.4915          0.3675                   0.0205
Detection Rate            0.4905          0.3570                   0.0195
Detection Prevalence      0.4930          0.3635                   0.0210
Balanced Accuracy         0.9965          0.9806                   0.9748
                     Class: Urbà Class: Vinya
Sensitivity               0.7727       0.9657
Specificity               0.9948       0.9918
Pos Pred Value            0.8361       0.9185
Neg Pred Value            0.9923       0.9967
Prevalence                0.0330       0.0875
Detection Rate            0.0255       0.0845
Detection Prevalence      0.0305       0.0920
Balanced Accuracy         0.8838       0.9787
```

Sumari estadístic

En aquest apartat calcularem quina proporció de cobertura forestal s'ha cremat sota les diferents categories de severitat.

Pasos:

1. Reclassificació dels valors del dNBR
2. Retall dels resultats pel perimetre de l'incendi
3. Resum estadístic

```r
# Càrrega del raster de severitat - (dNBR CAMPUS VIRTUAL)
dNBR <- rast("./Santa Coloma/dNBR_SantaColoma.tif")

# Reclassificació de la severitat
reclass_m <- matrix(c(-5000, 100, 0,
              100, 270, 1,
             270, 440, 2,
             440, 660, 3,
             660,1200,4),ncol=3,byrow = T)

dNBR_classified <- classify(dNBR,
                     reclass_m)


# Valors no cremats com a NA
dNBR_classified[dNBR_classified==0] <- NA

# Càrrega perimetre incendi (PERIMETRE DE CAMPUS VIRTUAL)

perimetre <- st_read("./Santa Coloma/perimetre_SantaColoma/perimetre_Sentinel_210813.shp")%>%
  st_transform(32631)
```

```
Reading layer `perimetre_Sentinel_210813' from data source
  `D:\OneDrive - udl.cat\01 - Docencia\102448 - Tècniques Avançades de Diagnòstic (DGEFCN)\24
  using driver `ESRI Shapefile'
Simple feature collection with 1 feature and 5 fields
Geometry type: POLYGON
Dimension:     XY
Bounding box:  xmin: 363874 ymin: 4595794 xmax: 376806.7 ymax: 4599744
Projected CRS: ETRS89 / UTM zone 31N
```

```r
# Convertim la classificació de RF (Random Forest) en polígons,
# dissolent les categories

usos_pol <- as.polygons(RF_class, dissolve = TRUE) %>%
  st_as_sf() %>%
  st_cast("POLYGON")
```

```
Warning in st_cast.sf(., "POLYGON"): repeating attributes for all
sub-geometries for which they may not be constant
```

```r
# Filtre dels polígons d'ús de sòl que es troben dins del perímetre d'interès

usos_cremats <- st_filter(usos_pol, perimetre)

# Convertim una classificació dNBR (differenced Normalized Burn Ratio) en polígons dins del p

dNBR_pol <- as.polygons(mask(dNBR_classified, perimetre), dissolve = TRUE) %>%
  st_as_sf()

# Intersecció espacial entre els polígons d'ús de sòl cremats i
#els polígons de classificació dNBR

results <- st_intersection(usos_cremats, dNBR_pol) %>%
  mutate(area = st_area(.))
```

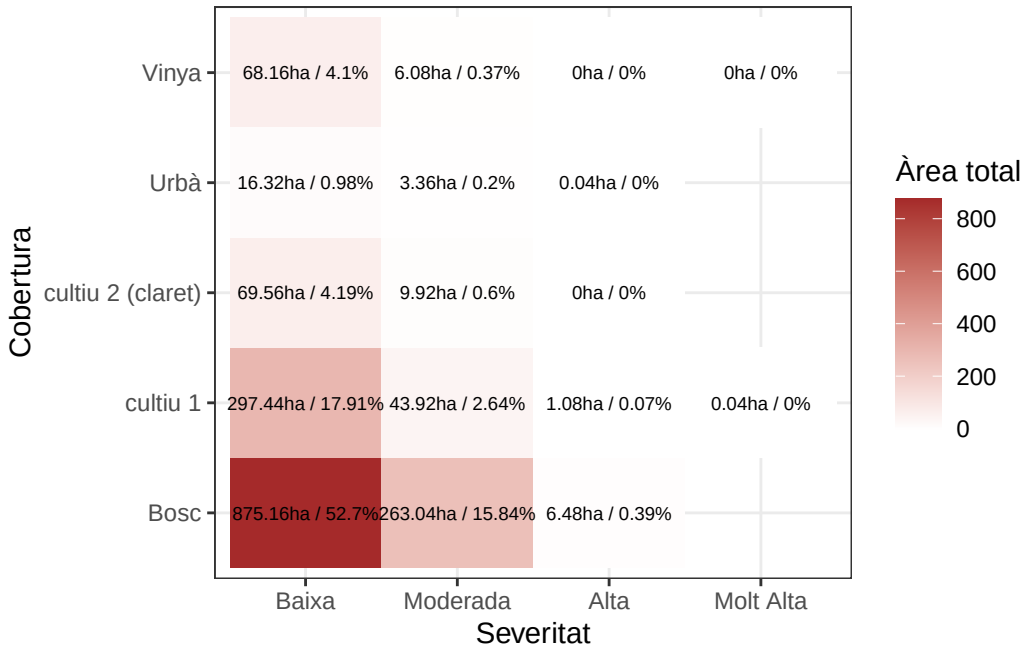
```
Warning: attribute variables are assumed to be spatially constant throughout
all geometries
```

```r
# Afegim una columna amb la classificació textual
# de severitat basada en `dNBR_SantaColoma`

results <- results %>%
  mutate(dNBR = ifelse(
    dNBR_SantaColoma == 0, "No cremat",
    ifelse(dNBR_SantaColoma == 1, "Baixa",
      ifelse(dNBR_SantaColoma == 2, "Moderada",
        ifelse(dNBR_SantaColoma == 3, "Alta",
          ifelse(dNBR_SantaColoma == 4, "Molt Alta", NA)
        )
      )
    )
  ))

# Resumim els resultats agrupant-los per classe d'ús de sòl i severitat de dNBR
summ <- results %>%
  group_by(class, dNBR) %>%
  summarise(
    Total_area = (sum(area %>%
                        units::drop_units()) / 10000),
    N = n()
  ) %>%
```

```
  ungroup() %>%
  mutate(Percentatge = round((Total_area / sum(Total_area)) * 100, 2)) %>%
  arrange(desc(Percentatge))
```

```
`summarise()` has grouped output by 'class'. You can override using the
`.groups` argument.
```

```
# Creem un gràfic de heatmap per visualitzar els resultats
ggplot(summ) +
  geom_tile(aes(x = fct_relevel(factor(dNBR),"Baixa", "Moderada","Alta","Molt Alta"),
                y=class,
                fill = Total_area)) +
  geom_text(aes(
    x = dNBR,
    y = class,
    label = paste0(round(Total_area, 2),
                   "ha / ",
                   Percentatge,
                   "%")),
    size = 2.5) +
  scale_fill_gradient(low = "white",
                      high = "brown") +
  theme_bw() +
  labs(y = "Cobertura",
       x = "Severitat",
       fill = "Àrea total")
```



Entregable

1. Esquema conceptual del procés metodològic (Només classificació supervisada)
2. Descripció i argumentació dels diferents punts de l'esquema.
3. Interpretació dels resultats per usos i severitat