ANÀLISI D'UNA XARXA BAYESIANA
Final d'una partida d'escacs: rei-torre vs. rei-peó en a7

5 d'abril de 2021

Esther Amores Gago[1], Anna Costa Garrido[2], and Oscar Ortiz Romero[3]

[1]Applied Statistics Student – Universitat Autònoma de Barcelona

Índex

1 Introducció

1.1 Els escacs

Els escacs són un dels jocs de taula més antics i populars, jugat per dos oponents que mouen 16 peces segons unes determinades regles fixes a través d'un tauler d'escacs i intenta fer escac i mat al rei de l'oponent [?]. Cada jugador té les peces d'un color, que pot ser blanc o negre, i disposa en iniciar la partida de totes les peces d'un d'aquests colors, que són un rei, una reina (o dama), dos alfils, dos cavalls, dues torres i vuit peons disposades tal i com mostra la Figura ??.

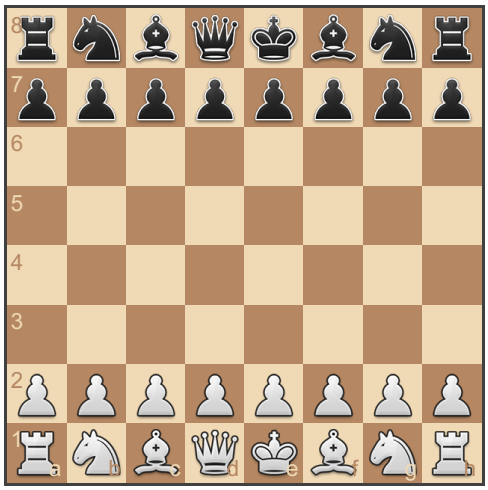


Figura 1: Diagrama d'escacs. Tauler d'escacs amb les peces col·locades inicialment

Tradicionalment es considera que si el valor d'un peó com a unitat comparativa és d'un punt, cada alfil i cada cavall en valen tres, cada torre cinc i la reina deu. Inicia la partida el jugador que té les peces blanques, i els dos jugadors van movent alternativament les pròpies peces, una cada vegada, intentant de matar el nombre més gran possible de les del contrari o bé d'aconseguir una posició dominant i que el rei d'aquest resti indefens. Si no pot fer escac i mat cap dels jugadors, la partida acaba en taules [?].

En una partida d'escacs existeixen tres fases: l'obertura, on els jugadors despleguen les seves peces en posicions útils per al joc; el joc mitjà, on les peces ja estan desplegades, es lluita per cada casella i els jugadors fan plans d'atac i de defensa; i el final, on la majoria de les peces estan fóra el tauler, els reis adopten un paper més actiu en la lluita i la promoció (o coronació) d'un peó és un factor decisiu [?].

Així doncs, la partida finalitza quan un dels jugadors fa escac i mat i guanya, o bé quan els jugadors empaten (fan taules). Els finals en escacs es classifiquen d'acord amb el tipus de peces i la quantitat de

peces que queden al tauler [?].

Per entendre les variables de la base de dades, cal tenir en compte els elements bàsics del tauler d'escacs [?]:

Fila Cadascuna de les vuit línies de vuit caselles que es numeren de l'1 al 8, començant des de la primera fila pel que fa al bàndol de les peces blanques.

Columna Cadascuna de les vuit línies de vuit caselles que s'anomenen amb lletres minúscules de la (a) a la (h), començant des de la primera columna esquerra pel que fa al bàndol de les peces blanques.

Diagonal Cadascuna de les 16 línies que es formen agrupant les caselles diagonalment. Les dues grans diagonals (a1-h8 o bé h1-a8) tenen vuit caselles.

Centre El centre del tauler són les quatre caselles centrals.

Cantonades Cadascuna de les quatre caselles situades a les cantonades del tauler.

Vores Les dues columnes (a i h) i dues files (1 i 8) situades als extrems del tauler.

1.2 Base de dades

En aquest treball s'utilitza una base de dades de l'any 1989 que té un total de 3196 registres i 36 variables, no té cap valor perdut (*missing*), i ha estat extreta del portal *UCI Machine Learning Repository* [?]. Les dades van ser originalment generades i descrites per Alen Shapiro, i el format del dataset va ser modificat el 1990 per tal que coincidís amb el format de la resta de bases de dades del repositori UCI.

Per tant, cadascun d'aquests registres representa, de les 3196 posicions de partides registrades, un tipus de final concret: torre i rei contra rei i peó. En aquest cas, torre i rei negres contra rei i peó blancs, on el peó blanc està en la posició del tauler a7 (veure Figura ??). És el torn de les blanques.Totes les variables prenen 2 valors, que són vertader (TRUE) o fals (FALSE), a excepció de les variables **dwipd** que pren els valors "g" o "l", **katri** que pren els valors "b", "n" i "w" (és la única variable que pren 3 valors) i **target**, que pren els valors "win" i "nowin".

A continuació s'hi troben de manera detallada cadascuna de les variables de la base de dades:

bkblk el rei negre no està en el camí de les peces blanques

bknwy el rei no molesta en cap moviment de la torre negra (black rook)

bkon8 el rei negre està a la fila 8 per ajudar a la torre negra

bkona el rei negre està a la columna A per ajudar a la torre negra

bkspr el rei negre pot ajudar a la torre negre amb 1 moviment

bkxbq el rei negre no està atacat de cap manera pel peó coronat

bkxcr el rei negre pot atacar la cuadrícula crítica al voltant de b7

bkxwp el rei negre pot atacar el peó blanc

blxwp negres ataquen al peó blanc (torre negre només pot en direcció x=-1, és a dir, cap a f7)

bxqsq una o més peces negres (és a dir, o el rei o la torre) controlen el quadrat de la coronació

cntxt el rei blanc està a una cantonada i no en a8

dsopp els reis es troben en oposició: es troben cara a cara en una fila o columna, amb només una casella entre ells

dwipd la distància del rei blanc al punt d'intersecció f4 és gran (*great*, g) o petita (*low*, l)

hdchk es fa escac a la descoberta, és a dir, és l'escac que fa una peça en moure's una altra peça del mateix bàndol que n'obstaculitzava l'acció.

katri el rei negre controla el punt d'intersecció de blanques (*white*, w), negres (*black*, b) o cap (*none*, n)

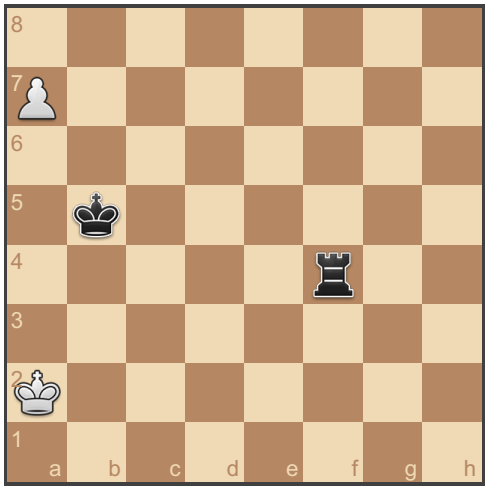


Figura 2: Diagrama d'escacs. Torre i rei negres contra rei i peó blancs.

mulch les peces negres poden tornar a fer escac per guanyar avantatge

qxmsq la casella d'escac i mat és atacada d'alguna manera pel peó blanc coronat

r2ar8 la torre negra no té accés a la fila 8 o a la columna A

reskd el rei blanc pot ser atacat doblement

reskr la torre negra en a4 faria una amenaça doble

rimmx la torre negra pot ser capturada de forma segura

rkxwp si la torre negra es mou a f7, amenaça al peó blanc

rxmsq la torre negra pot atacar de forma segura la casella per fer escac i mat

simpl hi aplica un patró molt simple perquè el peó blanc avança i corona (es converteix en una dama), aleshores la torre negra fa escac en a4 i, per tant, blanques estan forçades a menjar la torre amb la dama blanca

skach es pot fer 1 o més escacs al rei blanc per sacrificar el peó

skewr hi ha una clavada potencial fent un atac doble: el peó blanc es corona i la torre negra mou a a4

skrxp la torre negra pot aconseguir un atac doble o el rei negre pot atacar el peó blanc

spcop hi ha una oposició especial entre dos reis

stlmt el rei blanc és ofegat, és a dir: no té jugades legals per realitzar i el rei no es troba en estat d'escac. La partida acaba en taules

thrsk fer escac a la descoberta (una peça qu eestà interferint l'acció d'una segona, s'aparta del seu camí) fent un doble atac

wkcti el rei blanc no pot controlar el punt d'intersecció f4

wkna8 el rei blanc està a la casella a8

wknck el rei blanc està en escac

wkovl el rei blanc està sobrecarregat, és a dir: el rei blanc pot defensar una de les dues peces que estan sent amenaçades, per tant, sacrifica l'altre peça

wkpos el rei blanc està en una potencial posició de rebre un atac doble

wtoeg el rei blanc està a una casella de la cantonada

target blanques guanyen (*win*) o blanques perden (*nowin*)

La distribució que pren la variable classe WIN està equilibrada, és a dir: en 1669 posicions (52%) les blanques guanyen, mentre que en 1527 posiicions (48%) les blanques no guanyen.

1.3 Software

Els paquets que s'utilitzaran són els següents:

```
library(bnlearn)
library(gRain)
library(rchess)
library(Rgraphviz)
library(tidyverse)
library(MASS) # per realitzar el stepwise
library(readr)
library(xtable)

# OSCAR I ANNA: EXECUTEU AQUEST CODI PER A QUE NO US DONI PROBLEMES COMPILAR EL .Rnw
# devtools::install_github("jbkunst/rchess")
# webshot::install_phantomjs()
```

2 Construcció d'un classificador Bayesià

2.1 Preprocessament de les dades

En primer lloc, carreguem les dades al directori on es treballa, ens assegurem que no hi hagi cap valor missing amb la funció `anyNA`, assignem totes les variables com a categòriques amb la funció `factor` i mirem un resum descriptiu de cadascuna de les variables.

Pel que fa al preprocessament de les dades, primerament hi ajustem un model de regressió logística amb la funció `glm` (veure Taula **??**), ja que les la variable classe del dataset és dicotòmica (0-blanques perden, 1-blanques guanyen).

Taula 1: Model de regressió logística ajustat amb `glm`

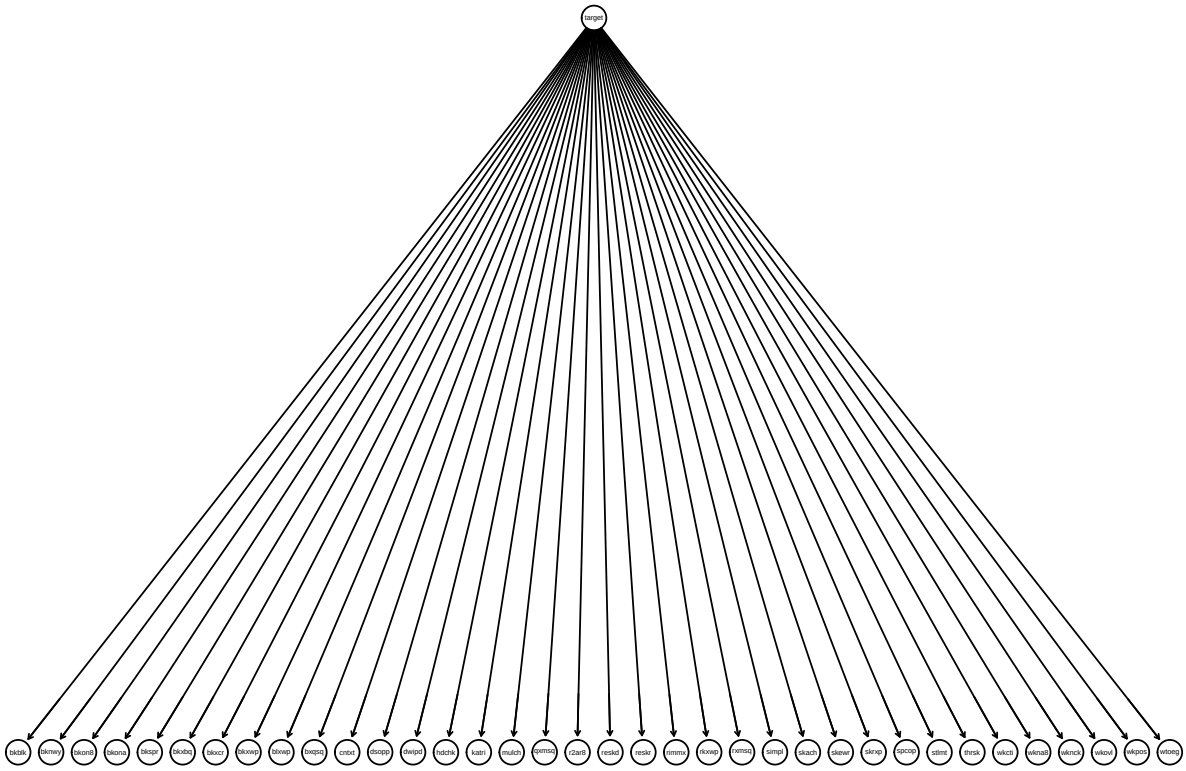
A continuació, per seleccionar les variables estadísticament significatives realitzem una selecció per mitjà del Criteri d'Informació d'Akaike (AIC) amb l'*Stepwise Algorithm*, tant el *Forward* com el *Backward*, i guardem el nou model amb les variables escollides en un nou objecte. El millor model escollit després de fer el mètode *Stepwise* segons valor del AIC exlou les variables **stlmt, reskd, skewr, bkspr, simpl, wtoeg, dwipd** i **spcop**.

2.2 *Naive Bayes classifier*

2.2.1 Aprenentatge d'estructura

Ara que ja hem fet el preprocessament de les dades, hem d'aprendre l'estructura de la xarxa mitjançant el mètode heurístic de *Search-and-score*. Aquest mètode utilitza una funció de puntuació i l'algoritme intenta trobar l'estructura que la maximitzi, per això farem servir la funció `hc` de la llibreria `bnlearn`. En aquest cas, el nostre dataset conté només dades completes, com bé hem dit abans.

Com volem que el classificador bayesià sigui un Naive Bayes, introduïrem una *white list* perquè ens interessa que els arcs vagin des de la variable classe (**target**) cap a tota la resta de variables explicatives, i una *black list* per evitar que les variables explicatives tinguin fletxes entre elles.



2.2.2 Aprenentatge de paràmetres

Ara que ja tenim l'estructura de les dades donada, estimarem els paràmetres mitjançant la base de dades per a cadascuna de les variables.

2.2.3 Classificació Bayesiana

L'últim pas és fer la nostra Xarxa Bayesiana sigui un classificador bayesià per a que assigni a nous finals determinats de partides d'escacs si les peces blanques guanyaran o no. Ara que ja tenim l'estructura i l'estimació dels paràmetres podem avaluar la bondat d'ajust del model, és a dir: com s'adapta a les dades observades.

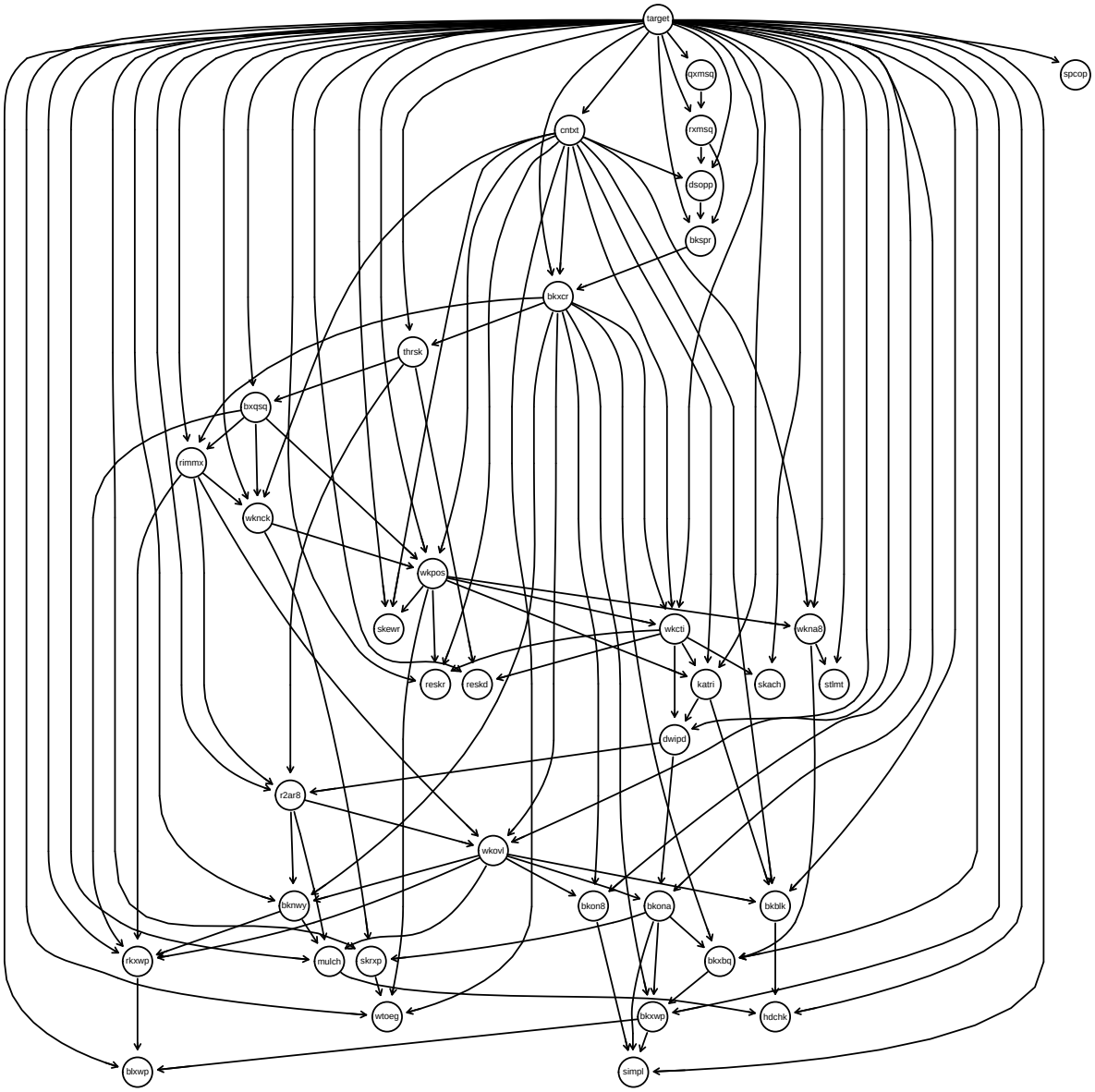
2.2.4 Validació del model

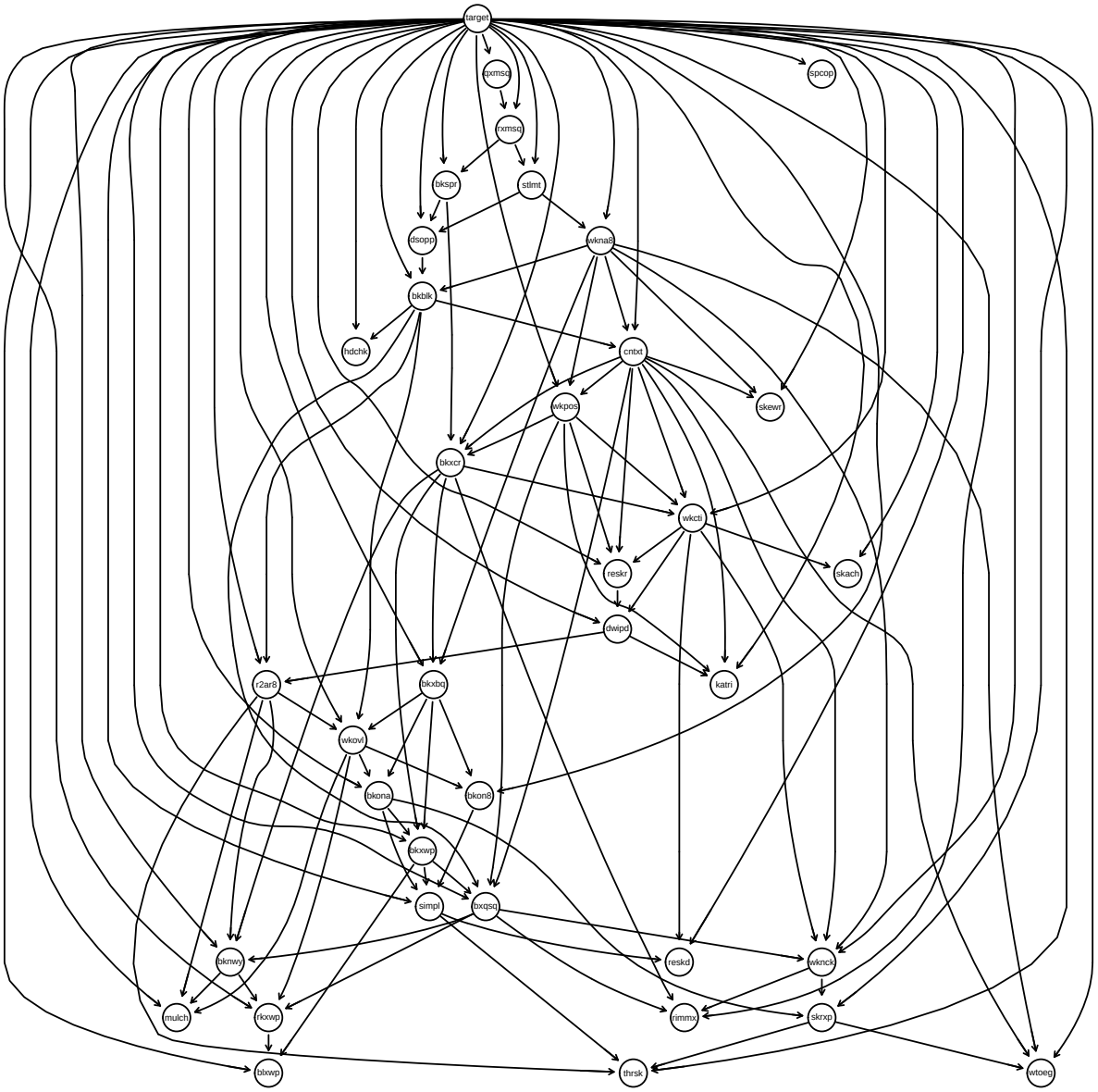
2.3 *Augmented Naive Bayes classifier*

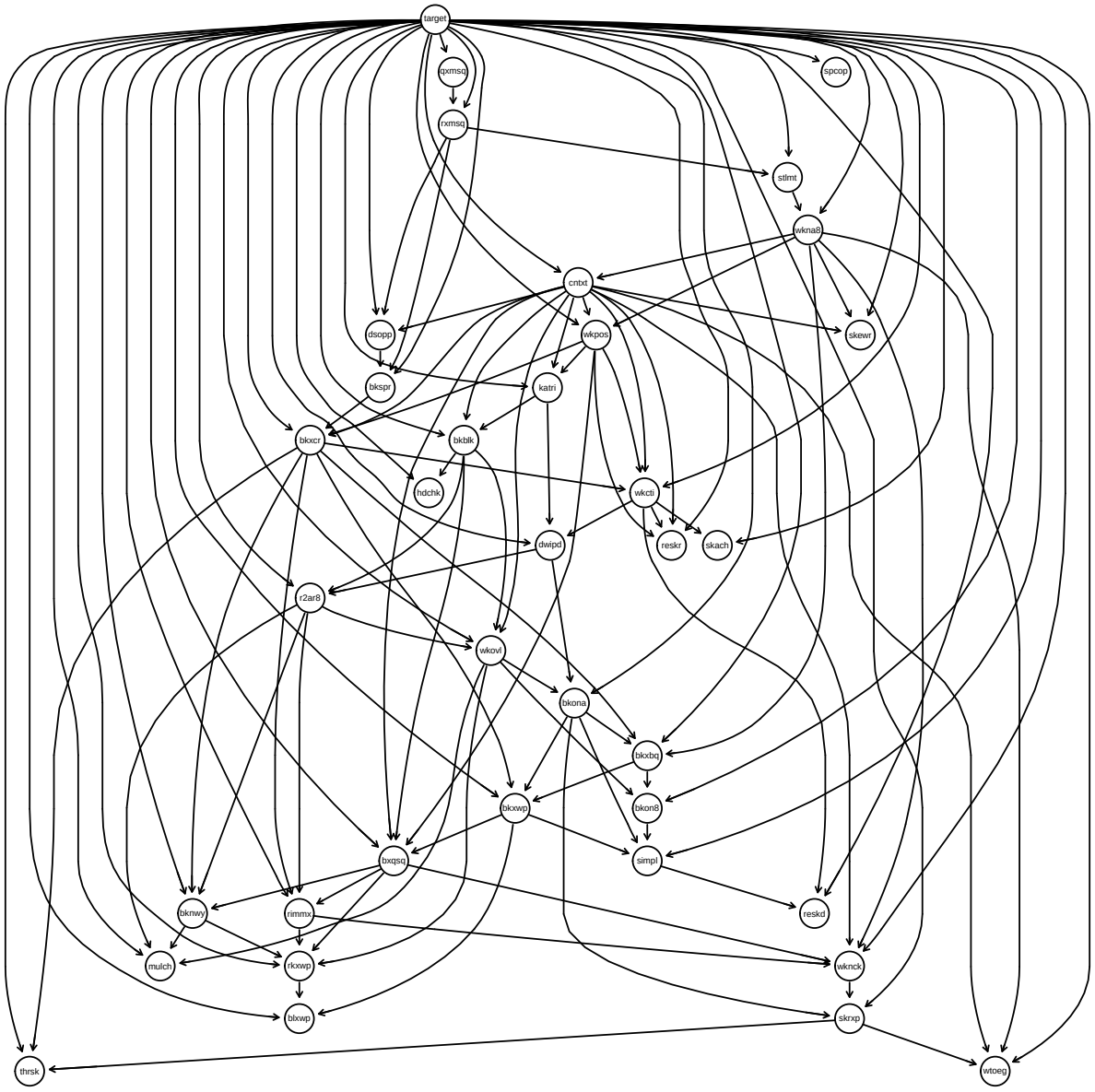
2.3.1 Aprenentatge d'estructura

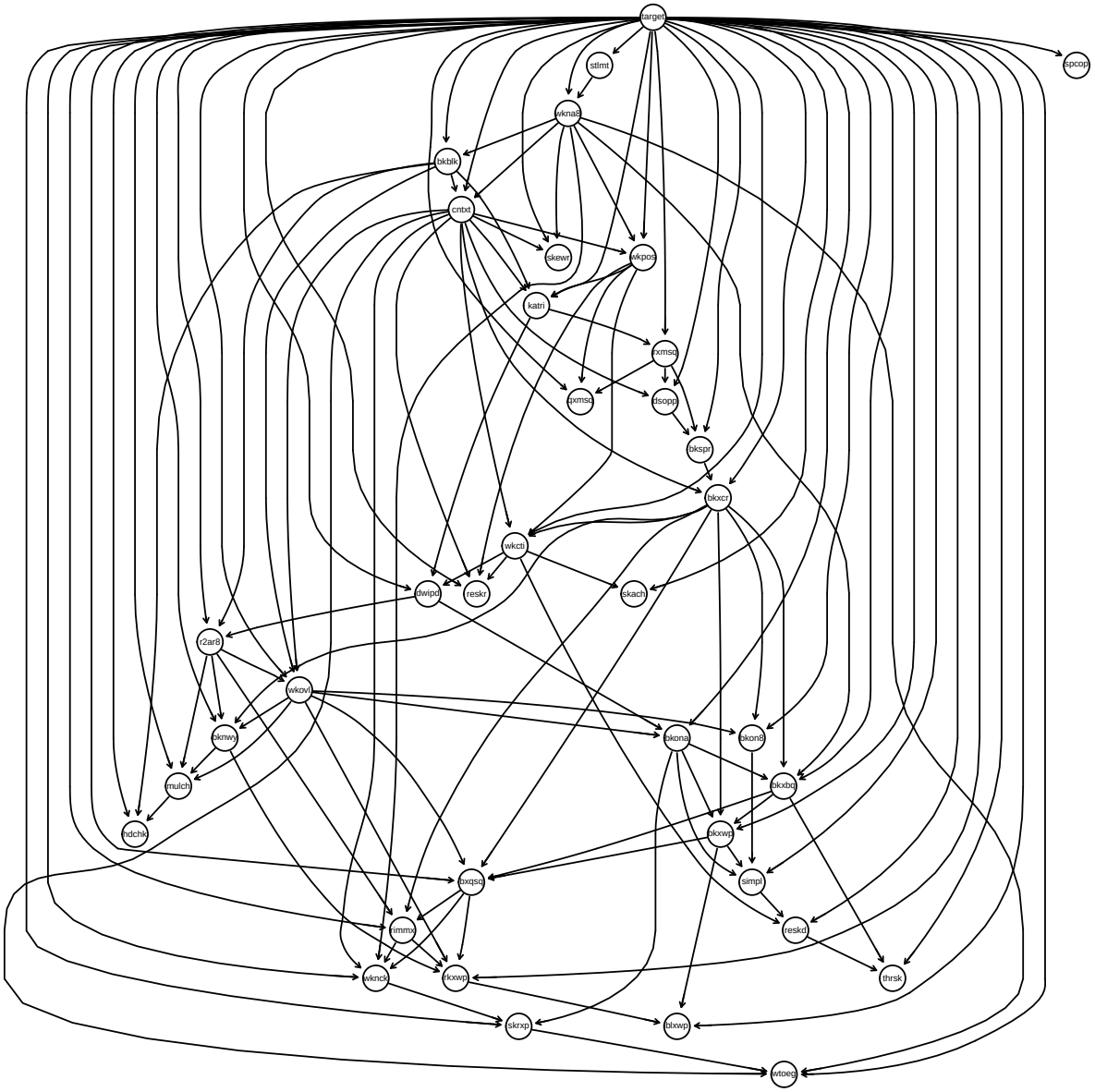
Ara que ja hem fet el preprocessament de les dades, hem d'aprendre l'estructura de la xarxa mitjançant el mètode heurístic de *Search-and-score*. Aquest mètode utilitza una funció de puntuació i l'algoritme intenta trobar l'estructura que la maximitzi, per això farem servir la funció `hc` de la llibreria `bnlearn`. En aquest cas, el nostre dataset conté només dades completes, com bé hem dit abans.

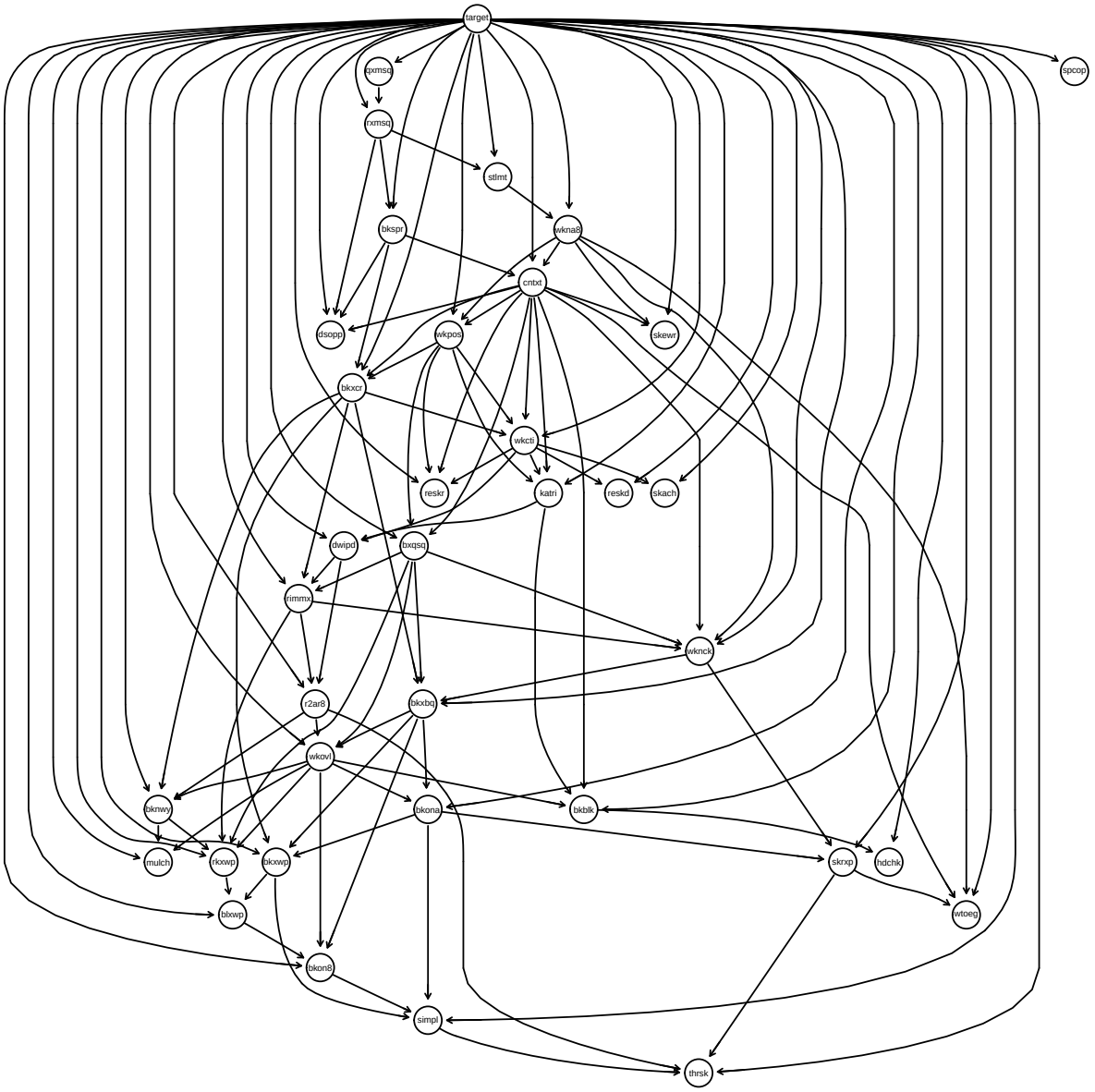
Com volem que el classificador bayesià sigui un Augmented Naive Bayes, introduïrem una *white list* perquè ens interessa que els arcs vagin des de la variable classe (**target**) cap a tota la resta de variables explicatives, però en aquest cas no afegirem una *black list* perquè no ens cal evitar que les variables explicatives tinguin fletxes entre elles.


target
spcop
qxmsq
rxmsq
cntxt
dsopp
bkspr
bkxcr
thrsk
bxqsq
rimmx
wknck
wkpos
skewr
wkcti
wkna8
reskr
reskd
katri
skach
stlmt
dwipd
r2ar8
wkovl
bknwy
bkon8
bkona
bkblk
rloowp
mulch
skrxp
blxbq
wtoeg
bkxwp
hdchk
blkwp
simpl


target
qxmsq
spcop
rxmsq
bkspr
stlmt
dsopp
wkna8
bkblk
hdchk
cntxt
wkpos
skewr
bkxcr
wkctl
reskr
skach
dwipd
r2ar8
bkxbq
katri
wkovl
bkona
bkon8
blxwp
simpl
bxqsq
bknwy
reskd
wknck
mulch
rlxwp
nimmx
skrxp
blxwp
thrsk
wtoeg


target
qpxmsq
spcop
rxmsq
stlmt
wkna8
cntxt
dsopp
wkpos
skewr
bkspr
katri
bkxcr
bkblk
hdchk
wkcti
dwipd
reskr
skach
r2ar8
wkovl
bkona
bkxbq
bkxwp
bkon8
bxqsq
simpl
bknwy
rimmx
reskd
mulch
rkxwp
wknck
blxwp
skrxp
thrsk
wtoeg


target
stlmt
spcop
wknab
bkblk
cmbxt
skewr
wkpos
katri
rxmsq
pxmsq
tsopp
bkspr
bloxcr
wkctl
dwipd
reskr
skach
r2ar8
wkovl
pkmwy
bkona
bkon8
mulch
blcxbq
ndchk
blcxwp
bxqsq
simpl
rmmxy
reskd
wknck
fkswp
thrsk
skrxp
blxwp
wtoeq



2.3.2 Aprenentatge de paràmetres

Ara que ja tenim l'estructura de les dades donada, estimarem els paràmetres mitjançant la base de dades per a cadascuna de les variables.

2.3.3 Classificació Bayesiana

L'últim pas és fer la nostra Xarxa Bayesiana sigui un classificador bayesià per a que assigni a nous finals determinats de partides d'escacs si les peces blanques guanyaran o no. Ara que ja tenim l'estructura i l'estimació dels paràmetres podem avaluar la bondat d'ajust del model, és a dir: com s'adapta a les dades observades.

2.3.4 Validació del model

3 Referències

[1] rchess *A Chess package for R* . http://jkunst.com/rchess/ [Online; visitat el 27 de març de 2021]

[2] chess *Definition of chess.* https://www.merriam-webster.com/dictionary/chess [Online; visitat el 27 de març de 2021]

[3] escacs *Escacs* https://www.enciclopedia.cat/ec-gec-0153105.xml [Online; visitat el 27 de març de 2021]

[4] *Fases de la partida.* https://www.123ajedrez.com/la-partida-de-ajedrez/fases-de-la-partida [Online; visitat el 27 de març de 2021]

[5] Final (escacs) *Finals contra rei i peó.* https://ca.wikipedia.org/wiki/Final_(escacs) [Online; visitat el 27 de març de 2021]

[6] Escacs *L'escaquer.* https://ca.wikipedia.org/wiki/Escacs [Online; visitat el 27 de març de 2021]

[7] Base de dades *Chess (King-Rook vs. King-Pawn) Data Set.* https://archive.ics.uci.edu/ml/datasets/Chess+%28King-Rook+vs.+King-Pawn%29 [Online; visitat el 27 de març de 2021]

A R code

A.1 Preprocessament de les dades

```
setwd("~/Trabajos del cole/ESTADÍSTICA APLICADA/4t curs/2n semestre/Modelització de Dades Complexes/
#### Importar les dades: ####
dades <- read_csv("kr-vs-kp.data",
                  col_names = c("bkblk","bknwy","bkon8","bkona","bkspr",
                                "bkxbq","bkxcr","bkxwp","blxwp","bxqsq",
                                "cntxt","dsopp","dwipd","hdchk","katri",
                                "mulch","qxmsq","r2ar8","reskd","reskr",
                                "rimmx","rkxwp","rxmsq","simpl","skach",
                                "skewr","skrxp","spcop","stlmt","thrsk",
                                "wkcti","wkna8","wknck","wkovl","wkpos",
                                "wtoeg", "target"))

anyNA(dades)

# Passar a factor totes les variables
dades <- lapply(dades, factor) %>%
  as.data.frame()

str(dades)
apply(dades, 2, table)
#### Preprocessing de les dades: ####
model <- glm(target ~ ., data = dades, family = "binomial")
xtable(model)
stepwise <- stepAIC(model, direction = "both")
# el millor model escollit després de fer el mètode Stepwise segons valor del AIC
# treu les variables: stlmt, reskd, skewr, bkspr, simpl, wtoeg, dwipd, spcop

# Les variables escollides les guardem en aquest nou objecte:
dades.step <- stepwise$model
```

A.2 *Naive Bayes classifier*

```
#### 1er: Aprendre l'estructura de les dades (amb totes les variables) ####

# - Introduim la black i white list:
atributes <- colnames(dades[-ncol(dades)])

wl <- data.frame(from = rep("target", length(atributes)), to = atributes)
bl <- rbind(
  expand.grid(atributes, atributes),
  data.frame(Var1 = atributes, Var2= rep("target", length(atributes))),
  data.frame(Var1 = "target", Var2= "target")
)

# - Fem el k-fold cross validation:
kfold <- function(k, seed = NULL){
  if(!is.null(seed)) set.seed(seed)
  trainingset <- list()
  testset <- list()
```

```
  for(i in 1:k){
    dades$id <- sample(1:k, nrow(dades), replace = TRUE)
    folds <- 1:k

    trainingset[[i]] <- subset(dades, id %in% folds[-i])
    testset[[i]] <- subset(dades, id %in% c(i))
  }

  return(list(training = trainingset, test = testset))
}

# Generem la llista amb tots els folds i treiem la columna dels id's
# tant pels trainings com pels tests

training <- kfold(5, 666)$training %>%
  lapply(function(x) x[,-ncol(x)])

test <- kfold(5, 666)$test %>%
  lapply(function(x) x[,-ncol(x)])


# - Aprendre l'estructura de les dades per tots els possibles folds
xarxa <- lapply(training, function(x) hc(x, score = "bic", whitelist = wl, blacklist = bl))

# Fem el plot per cada fold
lapply(xarxa, graphviz.plot)
#### 2on: Estimem els paràmetres de la xarxa, pel mètode MLE ####
xarxa.estimada <- list()
for(i in 1:5)
  xarxa.estimada[[i]] <- bn.fit(xarxa[[i]], training [[i]], method = "mle")

#### 3er: Fem la validació, estimar la classe per tots els conjunts tests ####
# passem la xarxa a format gRain
xarxa.grain <- lapply(xarxa.estimada, function(x) suppressWarnings(as.grain(x)))

distribucio <- NULL
prediccio <- NULL
CL <- NULL

for(i in 1:5){
  distribucio[[i]] <- list()
  prediccio[[i]] <- list()
  CL[[i]] <- list()
  for(j in 1:nrow(test[[i]])){
    if(is.numeric(predict(xarxa.grain[[i]],
                          response="target",
                          test[[i]][j,],
                          predictors=atributes,
                          type="dist")$pred[[1]][1,1])==FALSE)
    {
      prediccio[[i]][[j]]<-NA
      CL[[i]][[j]]<-0
      distribucio[[i]][[j]]<-c(rep(0,2))
    }
    else
```

```
    {
      distribucio[[i]][[j]] <- list(predict(xarxa.grain[[i]],
                                            response="target",
                                            test[[i]][j,],
                                            predictors=atributes,
                                            type="dist")$pred[[1]][1,])
      prediccio[[i]][[j]] <- names(distribucio[[i]][[j]][[1]])[which.max(distribucio[[i]][[j]][[1]])]
      CL[[i]][[j]]<-max(distribucio[[i]][[j]][[1]])
    }
  }
}
#### 4t: Càlcul de la matriu de confusió i les mesures d'avaluació ####
matriu.confusio <- list()
for(i in 1:5)
  matriu.confusio[[i]] <- as.matrix(table(unlist(prediccio[[i]]), test[[i]]$target))

# Accuracy
accuracy <- lapply(matriu.confusio, function(x) round((sum(x[1,1] + x[2,2])/sum(x))*100,2))
accuracy <- unlist(accuracy)

# True Positive Rate
TPR <- lapply(matriu.confusio, function(x) round((sum(x[1,1])/sum(x[1,1] + x[2,1]))*100,2))
TPR <- unlist(TPR)

# True Negative Rate
TNR <- lapply(matriu.confusio, function(x) round((sum(x[2,2])/sum(x[2,2] + x[1,2]))*100,2))
TNR <- unlist(TNR)

# Balanced Accuracy
baccuracy <- (TPR+TNR)/2

# Positive Predictive Value
PPV <- lapply(matriu.confusio, function(x) round((sum(x[1,1])/sum(x[1,1] + x[1,2]))*100,2))
PPV <- unlist(PPV)

# F1 score
F1 <- 2*((PPV*TPR)/(PPV+TPR))
```

A.3 *Augmented Naive Bayes classifier*

```
#### 1er: Aprendre l'estructura de les dades (amb totes les variables) ####

# - Introduim la white list:
atributes <- colnames(dades[-ncol(dades)])

wl <- data.frame(from = rep("target", length(atributes)), to = atributes)

# - Fem el k-fold cross validation:
kfold <- function(k, seed = NULL){
  if(!is.null(seed)) set.seed(seed)
  trainingset <- list()
  testset <- list()

  for(i in 1:k){
```

```
    dades$id <- sample(1:k, nrow(dades), replace = TRUE)
    folds <- 1:k

    trainingset[[i]] <- subset(dades, id %in% folds[-i])
    testset[[i]] <- subset(dades, id %in% c(i))
  }

  return(list(training = trainingset, test = testset))
}

# Generem la llista amb tots els folds i treiem la columna dels id's
# tant pels trainings com pels tests

training <- kfold(5, 666)$training %>%
  lapply(function(x) x[,-ncol(x)])

test <- kfold(5, 666)$test %>%
  lapply(function(x) x[,-ncol(x)])


# - Aprendre l'estructura de les dades per tots els possibles folds
xarxa <- lapply(training, function(x) hc(x, score = "bic", whitelist = wl))

# Fem el plot per cada fold
lapply(xarxa, graphviz.plot)
#### 2on: Estimem els paràmetres de la xarxa, pel mètode MLE ####
xarxa.estimada <- list()
for(i in 1:5)
  xarxa.estimada[[i]] <- bn.fit(xarxa[[i]], training [[i]], method = "mle")

#### 3er: Fem la validació, estimar la classe per tots els conjunts tests ####
# passem la xarxa a format gRain
xarxa.grain <- lapply(xarxa.estimada, function(x) suppressWarnings(as.grain(x)))

distribucio <- NULL
prediccio <- NULL
CL <- NULL

for(i in 1:5){
  distribucio[[i]] <- list()
  prediccio[[i]] <- list()
  CL[[i]] <- list()
  for(j in 1:nrow(test[[i]])){
    if(is.numeric(predict(xarxa.grain[[i]],
                          response="target",
                          test[[i]][j,],
                          predictors=atributes,
                          type="dist")$pred[[1]][1,1])==FALSE)
    {
      prediccio[[i]][[j]]<-NA
      CL[[i]][[j]]<-0
      distribucio[[i]][[j]]<-c(rep(0,2))
    }
    else
    {
```

```
      distribucio[[i]][[j]] <- list(predict(xarxa.grain[[i]],
                                            response="target",
                                            test[[i]][j,],
                                            predictors=atributes,
                                            type="dist")$pred[[1]][1,])
      prediccio[[i]][[j]] <- names(distribucio[[i]][[j]][[1]])[which.max(distribucio[[i]][[j]][[1]])]
      CL[[i]][[j]]<-max(distribucio[[i]][[j]][[1]])
    }
  }
}
#### 4t: Càlcul de la matriu de confusió i les mesures d'avaluació ####
matriu.confusio <- list()
for(i in 1:5)
  matriu.confusio[[i]] <- as.matrix(table(unlist(prediccio[[i]]), test[[i]]$target))

# Accuracy
accuracy <- lapply(matriu.confusio, function(x) round((sum(x[1,1] + x[2,2])/sum(x))*100,2))
accuracy <- unlist(accuracy)

# True Positive Rate
TPR <- lapply(matriu.confusio, function(x) round((sum(x[1,1])/sum(x[1,1] + x[2,1]))*100,2))
TPR <- unlist(TPR)

# True Negative Rate
TNR <- lapply(matriu.confusio, function(x) round((sum(x[2,2])/sum(x[2,2] + x[1,2]))*100,2))
TNR <- unlist(TNR)

# Balanced Accuracy
baccuracy <- (TPR+TNR)/2

# Positive Predictive Value
PPV <- lapply(matriu.confusio, function(x) round((sum(x[1,1])/sum(x[1,1] + x[1,2]))*100,2))
PPV <- unlist(PPV)

# F1 score
F1 <- 2*((PPV*TPR)/(PPV+TPR))
```