

---

# ANÀLISI D'UNA XARXA BAYESIANA

## Final d'una partida d'escacs: rei-torre vs. rei-peó en a7

---

8 d'abril de 2021

Esther Amores Gago<sup>1</sup>, Anna Costa Garrido<sup>2</sup>, and Oscar Ortiz Romero<sup>3</sup>

<sup>1</sup>Applied Statistics Student – Universitat Autònoma de Barcelona

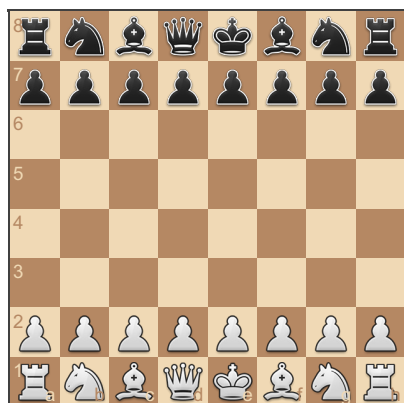
# Índex

<b>1</b>	<b>Introducció</b>	<b>3</b>
1.1	Els escacs . . . . .	3
1.2	Base de dades . . . . .	4
<b>2</b>	<b>Construcció d'un classificador Bayesià</b>	<b>6</b>
2.1	Preprocessament de les dades . . . . .	7
2.2	<i>Naive Bayes classifier</i> . . . . .	9
2.2.1	Aprenentatge d'estructura . . . . .	9
2.2.2	Aprenentatge de paràmetres . . . . .	9
2.2.3	Validació . . . . .	9
2.3	<i>Augmented Naive Bayes classifier</i> . . . . .	10
2.3.1	Aprenentatge d'estructura . . . . .	10
2.3.2	Aprenentatge de paràmetres . . . . .	10
2.3.3	Validació . . . . .	10
2.4	<i>Tria del clasificador bayesià final</i> . . . . .	10
2.4.1	Aprenentatge al conjunt d'entrenament i el de validació al conjunt test del classificador escollit . . . . .	11
<b>3</b>	<b>Aprenentatge i predicció per a noves partides amb el classificador escollit</b>	<b>13</b>
<b>4</b>	<b>Referències</b>	<b>14</b>
<b>A</b>	<b>R code</b>	<b>i</b>
A.1	Preprocessament de les dades . . . . .	i
A.2	<i>Naive Bayes classifier</i> . . . . .	i
A.3	<i>Augmented Naive Bayes classifier</i> . . . . .	i
A.4	<i>Tria del clasificador bayesià final</i> . . . . .	ii
A.5	<i>Predicció per noves partides</i> . . . . .	iii

# 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 [2]. 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 1.



**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 [3].

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 [4].

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 [5].

Per entendre les variables de la base de dades, cal tenir en compte els elements bàsics del tauler d'escacs [6]:

**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* [7]. 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 2). 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".

La *variable classe* serà **target**, i com ja s'ha esmentat, pren els valors "win" i "nowin". Cal esmentar, que la distribució que pren aquesta variable **TARGET** està força equilibrada, és a dir: en 1669 casos (52%) les blanques guanyen (valor "win"), mentre que en 1527 casos (48%) les blanques no guanyen (valor "nowin").

Per altra banda, a continuació, les *variables atributs* s'exposen de manera detallada a partir d'un exemple (veure Figura 2):

**bkbk** 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 negra amb 1 moviment

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

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

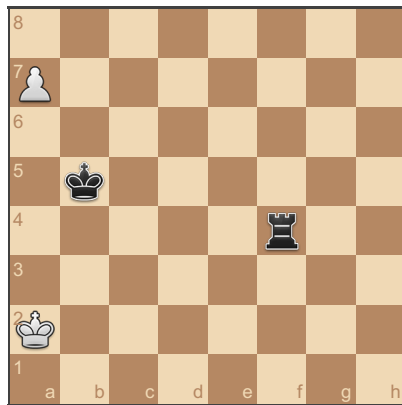
**bkxwp** el rei negre pot atacar el peó blanc

**blxwp** negres ataquen al peó blanc (torre negra 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



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

**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)

**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 que 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

**wtog** el rei blanc està a una casella de la cantonada

És a dir, l'**objectiu** és construir una Xarxa Bayesiana de manera que, donada unes posicions concretes on només tenim la torre i el rei negres i, el rei i el peó blancs (aquest últim està en la posició del tauler a7), poder estimar la probabilitat que les blanques puguin guanyar o no.

Com que aquesta Xarxa Bayesiana, es farà servir per classificar si guanyen o no les blanques, s'utilitzaran dos possibles classificadors bayesians: el *Naive Bayes* i l'*Augmented Naive Bayes*.

En definitiva, per poder assolir aquest objectiu: en primer lloc es farà un preprocessing de la base de dades, seguidament, es separarà la base de dades amb un conjunt d'entrenament i un de test. Amb el d'entrenament, es durà a terme el procediment de k-fold cross-validació per tal d'entrenar i validar tant el Naive Bayes com l'Augmented Naive Bayes (s'utilitzarà el paquet *bnlearn* per entrenar cadascuna de les xarxes). També, a partir de les mesures de comportament triarem aquell que sigui l'òptim. Per altra banda, amb la xarxa escollida i el conjunt test farem una segona validació per acabar de constatar si la xarxa escollida funciona bé. I finalment, a partir de tota la base de dades, s'aprendrà el classificador bayesià escollit, per a fer la predicció de la classe de nous casos i predigui la classe d'alguns casos que ens han semblat interessants (mitjançant el paquet *gRain*).

## 2 Construcció d'un classificador Bayesià

En aquest apartat, com ja s'ha explicat anteriorment, s'explicaran i s'ensenyan els passos que s'han seguit per assolir l'objectiu proposat. Els enumerem a continuació:

El primer pas, és el **preprocessament de les dades** on es selecciona les variables, mitjançant l'algoritme Stepwise, per posteriorment aprendre les xarxes. I també es separa la base de dades en un conjunt d'entrenament i un de test, on aquest últim s'utilitzarà per acabar de constatar si el classificador bayesià escollit funciona bé.

Seguidament, amb el conjunt d'entrenament, es fa el **procés d'aprenentatge i validació**, mitjançant k-fold cross validation, tant pel classificador Naive Bayes com per l'Augmented Naive Bayes. I també la seva comparativa amb la mesura de comportament *Accuracy*.

Tanmateix, per acabar de constatar si el classificador bayesià escollit funciona bé, es tornarà a fer la validació amb el conjunt de test.

Finalment, farem la construcció final del model amb tota la base de dades per fer la seva predicció de nous casos i aquells que s'han considerat interessants.

## 2.1 Preprocessament de les dades

En primer lloc, es carrega la base de dades al directori on es treballa i s'assegura que no hi hagi cap valor missing amb la funció `anyNA`. A més, s'assigna totes les variables com a categòriques amb la funció `factor` i es mira un resum descriptiu de cadascuna de les variables.

Aquest últim pas es pot veure a continuació:

Cal destacar que la variable `spcop` obté una distribució de les dades molt anòmla ja que només existeix un cas on aquesta variable pren el valor `TRUE`. D'aquesta manera, es tindrà en compte alhora de triar les variables que s'utilitzin per entrenar les xarxes.

Tanmateix, es pot veure que cap de les variables conté valors perduts, i per tant, és un fet destacable i que també s'ha de tenir en compte quan s'aprenen les xarxes.

Pel que fa al preprocessament de les dades, primerament s'ajusta un model de regressió logística amb la funció `glm` (veure Taula 4), ja que la variable classe del dataset és dicotòmica (0-blanques perden, 1-blanques guanyen).

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-0.0267	0.9777	-0.03	0.9782
bkbkTRUE	5.5019	0.6667	8.25	0.0000
bknwyTRUE	8.1035	1.2881	6.29	0.0000
bkon8TRUE	-6.2764	1.1419	-5.50	0.0000
bkonaTRUE	-3.6138	0.6553	-5.51	0.0000
bksprTRUE	-0.2133	0.3192	-0.67	0.5041
bkxbqTRUE	4.0130	0.4736	8.47	0.0000
bkxcrTRUE	-1.1294	0.3951	-2.86	0.0043
bkwpxTRUE	1.3558	0.7235	1.87	0.0609
blxwpTRUE	-2.3534	0.7895	-2.98	0.0029
bxqsqTRUE	-37.0606	663.2263	-0.06	0.9554
cntxtTRUE	-1.1261	0.5577	-2.02	0.0435
dsoppTRUE	1.6302	0.4747	3.43	0.0006
dwipdl	-0.1663	0.3281	-0.51	0.6121
hdchkTRUE	-22.8198	9027.1090	-0.00	0.9980
katrin	3.4570	0.5776	5.99	0.0000
katriw	8.1036	0.9507	8.52	0.0000
mulchTRUE	-10.9328	1.8880	-5.79	0.0000
qxmsqTRUE	5.4414	0.9081	5.99	0.0000
r2ar8TRUE	-0.8848	0.5035	-1.76	0.0788
reskdTRUE	-0.2835	1.6745	-0.17	0.8656
reskrTRUE	-1.3756	0.4884	-2.82	0.0049
rimmxTRUE	83.0630	1247.2067	0.07	0.9469
rkxwpTRUE	1.6233	0.7217	2.25	0.0245
rxmsqTRUE	-4.8069	0.5364	-8.96	0.0000
simplTRUE	-0.0397	0.2782	-0.14	0.8865
skachTRUE	-2.7302	1.5892	-1.72	0.0858
skewrTRUE	0.1431	0.3873	0.37	0.7118
skrxpTRUE	-2.8784	0.7466	-3.86	0.0001
spcopTRUE	-21.3896	48196.1423	-0.00	0.9996
stlmtTRUE	-10.9730	5134.8656	-0.00	0.9983
thrskTRUE	-2.4217	0.7399	-3.27	0.0011
wkctiTRUE	1.4413	0.4925	2.93	0.0034
wkna8TRUE	-13.9101	1.6674	-8.34	0.0000
wknckTRUE	-10.3333	0.6991	-14.78	0.0000
wkovlTRUE	-2.1560	0.5336	-4.04	0.0001
wkposTRUE	3.8434	0.4796	8.01	0.0000
wtoegt	0.0790	0.3903	0.20	0.8395

**Taula 4:** Model de regressió logística ajustat amb `extttglm`

	Level	N	%
bkblk	FALSE	2839	88.8
	TRUE	357	11.2
bkwny	FALSE	2971	93.0
	TRUE	225	7.0
bkon8	FALSE	3076	96.2
	TRUE	120	3.8
bkona	FALSE	2874	89.9
	TRUE	322	10.1
bkspr	FALSE	2129	66.6
	TRUE	1067	33.4
bkxbq	FALSE	1722	53.9
	TRUE	1474	46.1
bkxcr	FALSE	2026	63.4
	TRUE	1170	36.6
bkxwp	FALSE	2500	78.2
	TRUE	696	21.8
blxwp	FALSE	1980	62.0
	TRUE	1216	38.0
bxqsq	FALSE	2225	69.6
	TRUE	971	30.4
cntxt	FALSE	1817	56.9
	TRUE	1379	43.1
dsopp	FALSE	2860	89.5
	TRUE	336	10.5
dwipd	g	991	31.0
	l	2205	69.0
hdchk	FALSE	3181	99.5
	TRUE	15	0.5
katri	b	224	7.0
	n	2526	79.0
	w	446	14.0
mulch	FALSE	3040	95.1
	TRUE	156	4.9
qxmsq	FALSE	3099	97.0
	TRUE	97	3.0
r2ar8	FALSE	1000	31.3
	TRUE	2196	68.7
reskd	FALSE	3170	99.2
	TRUE	26	0.8

	Level	N	%
reskr	FALSE	2714	84.9
	TRUE	482	15.1
rimmx	FALSE	2612	81.7
	TRUE	584	18.3
rkxwp	FALSE	2556	80.0
	TRUE	640	20.0
rxmsq	FALSE	3013	94.3
	TRUE	183	5.7
simpl	FALSE	1975	61.8
	TRUE	1221	38.2
skach	FALSE	3185	99.7
	TRUE	11	0.3
skewr	FALSE	980	30.7
	TRUE	2216	69.3
skrxp	FALSE	3021	94.5
	TRUE	175	5.5
spcop	FALSE	3195	100.0
	TRUE	1	0.0
stlmt	FALSE	3149	98.5
	TRUE	47	1.5
thrsk	FALSE	3060	95.7
	TRUE	136	4.3
wkcti	FALSE	2631	82.3
	TRUE	565	17.7
wkna8	FALSE	3021	94.5
	TRUE	175	5.5
wknck	FALSE	1984	62.1
	TRUE	1212	37.9
wkovl	FALSE	1189	37.2
	TRUE	2007	62.8
wkpos	FALSE	851	26.6
	TRUE	2345	73.4
wtoeg	n	2407	75.3
	t	789	24.7
target	nowin	1527	47.8
	won	1669	52.2

**Taula 3:** Freqüències absolutes de les variables del dataset **chess** amb els corresponents percentatges de freqüències relatives de cadascuna de les seves variables



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 exclou les variables **stlmt**, **reskd**, **skewr**, **bkspr**, **simpl**, **wtoeg**, **dwipd** i **spcop**, i aquesta última ja s'ha pogut veure anteriorment que obté una distribució anòmala dels seus casos.

Per tant, aquestes variables no les considerarem al posterior entrenament de les xarxes.

I per últim, es separa la base de dades en un conjunt d'entrenament i en un de test on pel primer, es fa el **procés d'aprenentatge i validació**, mitjançant k-fold cross validation, tant pel classificador Naive Bayes com per l'Augmented Naive Bayes i pel test, s'acabarà de constatar si el classificador bayesià escollit funciona bé.

## 2.2 Naive Bayes classifier

En aquest apartat i a partir del conjunt d'entrenament es durà a terme l'aprenentatge de l'estructura i dels paràmetres del classificador Naive Bayes així com la seva validació, mitjançant el procediment de k-fold cross-validation amb  $k=10$ .

És a dir, aquest conjunt d'entrenament es separarà en 10 parts proporcionals, que tindran aproximadament el mateix nombre de casos. Llavors es començarà un procés iteratiu on repetirà 10 vegades el següent: s'aprendrà tant l'estructura com els paràmetres del classificador a partir del conjunt d'entrenament format per 9 trossos i pel tros restant, es farà la validació, és a dir, per cada cas d'aquest s'entren els inputs al classificador entrenat i fem la predicció per l'output. Llavors per a cada iteració del procés, s'anirà canviant el tros del conjunt de test per a fer la corresponent validació.

A partir d'això podrem aconseguir la matriu de confusió i calcular les mesures de comportament corresponents per cadascuna de les iteracions.

A continuació s'explica en més detall en què consisteix l'aprenentatge de l'estructura i dels paràmetres així com de la seva validació que es desenvolupa a cada iteració. També es mostrarà quines són les mesures de comportament que s'obtenen.

### 2.2.1 Aprenentatge d'estructura

Com ja s'ha esmentat, s'ha fet l'aprenentatge de 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

Un cop ja tenim l'estructura del classificador, s'ha estimat els paràmetres, és a dir, les probabilitats de cada node condicionat al seu pare per a cada iteració.

### 2.2.3 Validació

L'últim pas és fer que el classificador bayesià que hem après a cadascuna de les iteracions, predigui per a cada cas de cadascun dels folds destinats a la validació, la probabilitat que assigni si les peces blanques guanyaran o no.

D'aquesta manera, s'obté la matriu de confusió de cada iteració, així com les mesures de comportament calculades, com es pot visualitzar a continuació:

	Fold - 1	Fold - 2	Fold - 3	Fold - 4	Fold - 5	Fold - 6	Fold - 7	Fold - 8	Fold - 9	Fold - 10
Accuracy	89.22	83.53	89.62	87.64	86.18	88.49	91.67	90.70	85.19	85.20
True Positive Rate	89.17	79.25	88.72	88.73	84.87	88.39	88.99	88.03	82.35	87.18
True Negative Rate	89.29	86.71	90.55	86.47	87.40	88.57	94.12	92.91	87.90	83.46
Balanced Accuracy	89.23	82.98	89.63	87.60	86.14	88.48	91.56	90.47	85.12	85.32
Positive Predictive Value	89.92	81.55	90.77	87.50	86.32	86.09	93.27	91.15	86.73	82.26
F1 score	89.54	80.38	89.73	88.11	85.59	87.22	91.08	89.56	84.48	84.65

**Taula 5:** Mesures de comportament per cada fold del classificador Naive Bayes

## 2.3 *Augmented Naive Bayes classifier*

Tal com s'ha realitzat el classificador de Naive Bayes, s'han seguit els mateixos passos per a realitzar l'aprenentatge i la validació per l'Augmented Naive Bayes. En aquest cas, l'única cosa que varia, és com es defineix l'aprenentatge de l'estructura per a cada iteració, com s'exposarà al següent apartat.

### 2.3.1 Aprenentatge d'estructura

Quan volem que el classificador bayesià sigui un Augmented Naive Bayes, introduïm 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 afegim una *black list* perquè no ens cal evitar que les variables explicatives tinguin fletxes entre elles.

### 2.3.2 Aprenentatge de paràmetres

De manera similar a l'aprenentatge de paràmetres del Naive Bayes, s'ha estimat els paràmetres, és a dir, les probabilitats de cada node condicionat al seu pare per a cada iteració.

### 2.3.3 Validació

Com ja s'ha explicat a la validació pel classificador del Naive Bayes, 2.2.3, l'últim pas és fer que el classificador bayesià que hem après per cadascuna de les iteracions, predigui per cada cas de cadascun dels folds destinats a la validació, la probabilitat que assigni si les peces blanques guanyaran o no.

D'aquesta manera, s'obté la matriu de confusió de cada iteració, així com les mesures de comportament calculades, com es pot visualitzar a continuació:

	Fold - 1	Fold - 2	Fold - 3	Fold - 4	Fold - 5	Fold - 6	Fold - 7	Fold - 8	Fold - 9	Fold - 10
Accuracy	95.69	96.75	96.90	97.42	96.34	96.80	95.59	98.44	97.93	98.80
True Positive Rate	96.67	96.15	98.48	98.59	97.48	98.18	98.15	99.14	98.31	100.00
True Negative Rate	94.64	97.18	95.24	96.12	95.28	95.71	93.28	97.87	97.56	97.74
Balanced Accuracy	95.66	96.67	96.86	97.36	96.38	96.94	95.72	98.50	97.94	98.87
Positive Predictive Value	95.08	96.15	95.59	96.55	95.08	94.74	92.98	97.46	97.48	97.48
F1 score	95.87	96.15	97.01	97.56	96.27	96.43	95.50	98.29	97.89	98.72

**Taula 6:** Mesures de comportament per cada fold del classificador Augmented Naive Bayes

## 2.4 *Tria del clasificador bayesià final*

Una vegada ja tenim el dos clasificadors bayesians "Naive Bayes" i "Augmented Naive Bayes" hem de triar quina de les dues xarxes modelitza millor les nostres dades d'escacs, i això ho podem fer a partir de les mesures de bondat d'ajust de cadascuna de les xarxes.

Com que amb el procés de validació de cada xarxa, s'ha aconseguit una mostra de la mesura Accuracy, s'ha mirat si aquestes dues mostres segueixen una distribució de normalitat, a través del test de Shapiro Wilk. Aquest considera a la hipòtesi nul·la que la mostra en qüestió segueix la distribució normal.

S'ha vist que efectivament, els dos p-valors són majors a 0.05 (respectivament són tal i tal), i per tant, no tenim evidències per rebutjar aquesta hipòtesi de normalitat a les dues mostres.

D'aquesta manera, s'ha desenvolupat un t-test de mostres aparellades, per veure si les mitjanes de les accuracies per cada fold del Naive Bayes i Augmented Naive es poden considerar diferents. Aquest test anomena a la hipòtesi nul·la que les mitjanes de les dues mostres són iguals. No obstant, el p-valor ha resultat ser menor a 0.001 i per tant, tenim evidències per rebutjar aquesta hipòtesi nul·la i considerar que les mitjanes de les accuracies són diferents.

Com es pot observar, a la taula següent, la mitjana de l'Accuracy del Augmented Naive Bayes és major al del Naive Bayes. Per tant, el model escollit serà l'Augmented Naive Bayes.

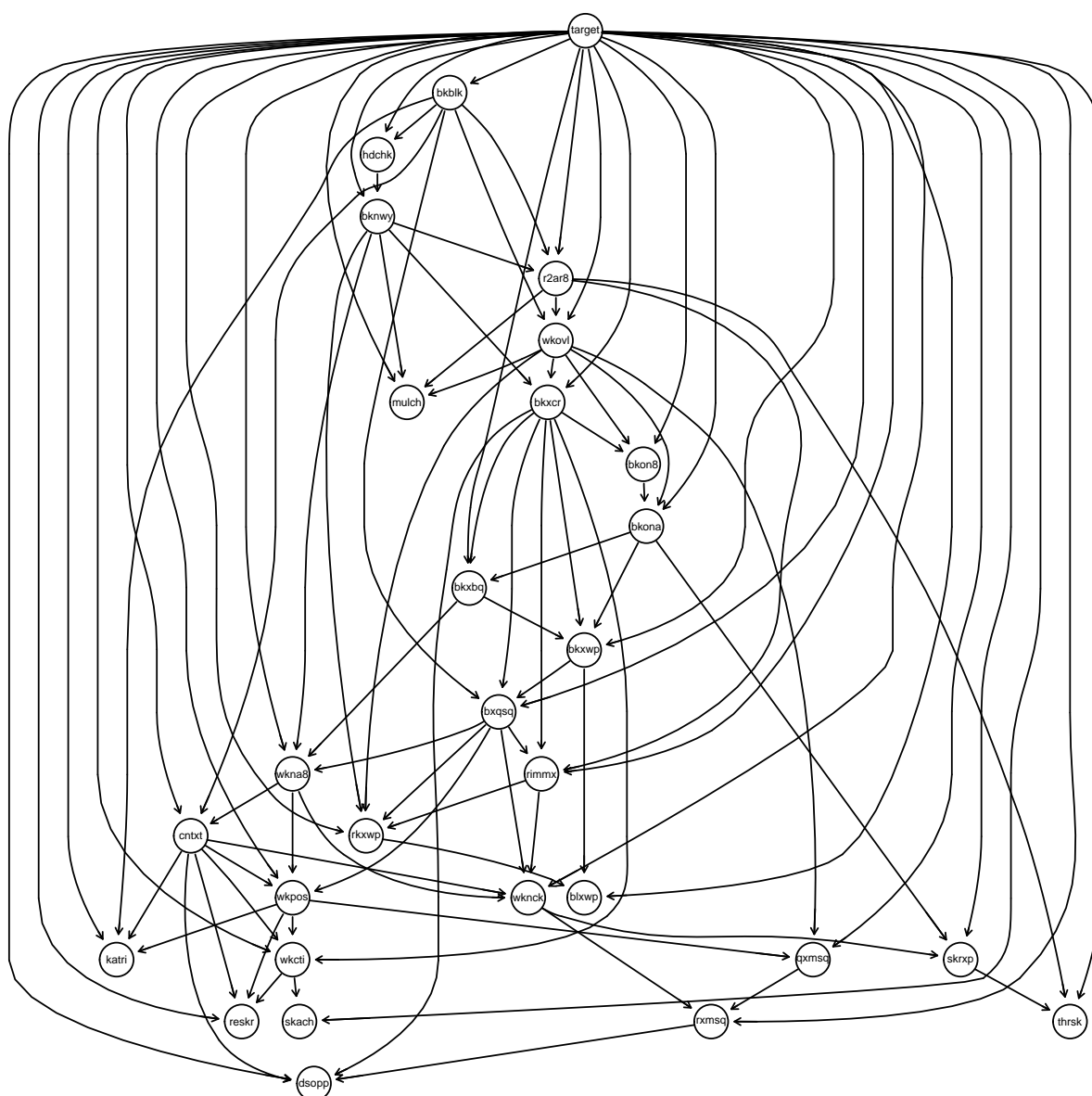
	Naive Bayes	Augmented Naive Bayes
Accuracy	87.74	97.07
True Positive Rate	86.57	98.12
True Negative Rate	88.74	96.06
Balanced Accuracy	87.65	97.09
Positive Predictive Value	87.56	95.86
F1 score	87.03	96.97

**Taula 7:** Mitjanes de les mesures de comportament del classificador Naive Bayes i Augmented Naive Bayes

Tanmateix, podem veure que totes les altres mesures (TPR, TNR, PPV, F1) que hem calculat per saber com de bé s'estima les xarxes bayesianes, les del "Augmented Naive Bayes" són més elevades (>95%) que les del "Naive Bayes".

#### 2.4.1 Aprenentatge al conjunt d'entrenament i el de validació al conjunt test del classificador escollit

Finalment el classificador escollit és l'"Augmented Naive Bayes", en aquest cas tornarem a fer l'aprenentatge de l'estructura, considerant una *white list* que forçi tot els arcs de la variable classe (**target**) cap a la resta de variables, pero sense una *black list* per que el model pugui crear arcs entre les variables explicatives. També farem l'aprenentatge dels paràmetres i ambdós es faran a partir del conjunt d'entrenament. Un cop fet aquests aprenentatge, es fa la validació sobre el conjunt tests. Aquestes són les mesures de comportament obtingudes:



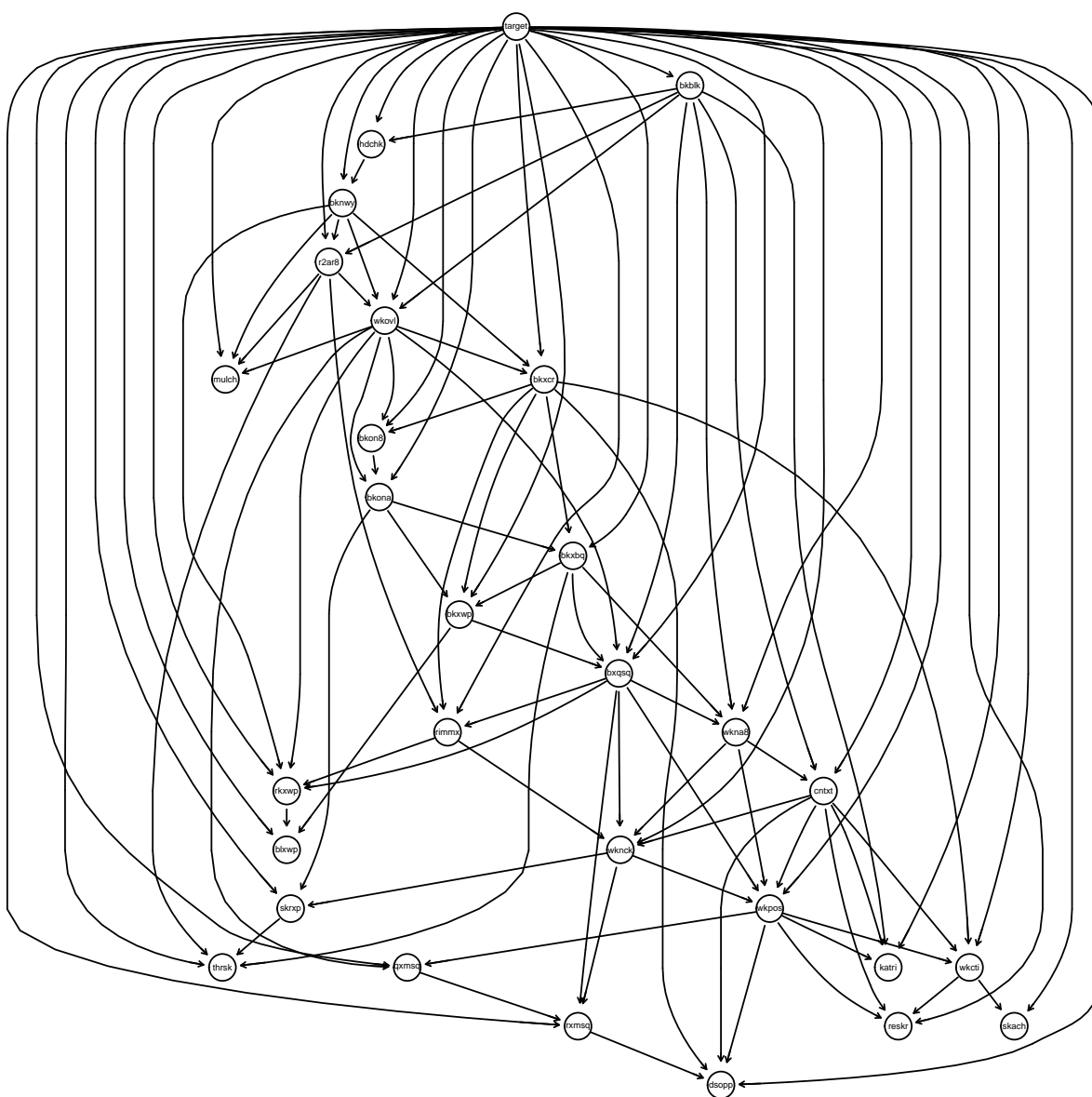
	Mètriques
Accuracy	96.71
True Positive Rate	97.76
True Negative Rate	95.71
Balanced Accuracy	96.74
Positive Predictive Value	95.62
F1 score	96.68

**Taula 8:** Mesures de comportament del classificador Augmented Naive Bayes amb l'aprenentatge sobre el conjunt d'entrenament i la validació sobre el de test

Observem, finalment, que el classificador Augmented Naive Bayes obté unes mètriques molt bones (>94%).

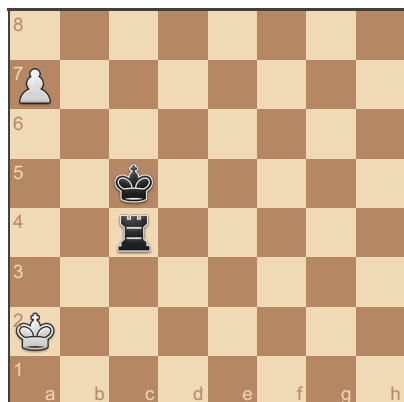
Al següent apartat ja podrem entrenar la nostra xarxa bayesiana final amb **totes** les dades fent servir un "Augmented Naive Bayes", per després procedir a fer la validació.

### 3 Aprenentatge i predicció per a noves partides amb el classificador escollit



Un cop s'ha après l'estructura i els paràmetres del classificador Augmented Naive Bayes a partir de tota la base de dades, ja estem preparats per poder simular noves partides d'escacs on es doni aquesta situació de final de partida on les peces blanques tenen el Rei i un Peó mentre que les negres tenen el Rei i un torre. Hem pensat tres propostes de partides d'escacs amb aquestes condicions i son les següents:

```
## Error in .nodes.states2evidence(nodes, states): Either states are not given or nodes
and states do not have same length
## Error in eval(expr, envir, enclos): objeto 'xarxa.evid' no encontrado
## Error in querygrain(xarxa.evid, nodes = c("target"), type = "marginal"): objeto 'xarxa.evid'
no encontrado
## Error in eval(expr, envir, enclos): objeto 'qq' no encontrado
## Error in base::which.max(x, ...): 'list' object cannot be coerced to type 'double'
## Error in max(distribucio): 'type' (list) de argumento no válido
## Error in 100 * distribucio[2]: argumento no-numérico para operador binario
```

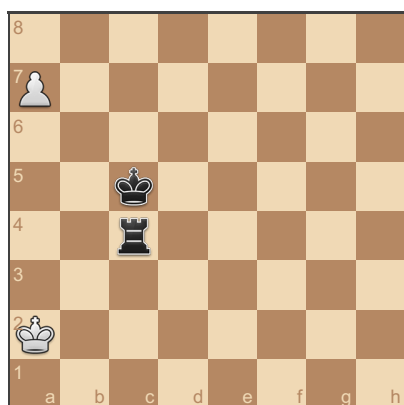


**Figura 3:** Diagrama d'escacs. Torre i rei negres contra rei i peó blancs.

```
## Error in matrix(c(predicció, CL, reincidence.risc), nrow = 1):  objeto 'reincidence.risc'
no encontrado
## Error in h(simpleError(msg, call)):  error in evaluating the argument 'x' in selecting
a method for function 'as.data.frame':  objeto 'Resultat' no encontrado
## Error in colnames(Resultat) <- c("Predicció", "Confidence Level (CL) en %", :  objeto
'Resultat' no encontrado
## Error in eval(expr, envir, enclos):  objeto 'evid' no encontrado
## Error in eval(expr, envir, enclos):  objeto 'Resultat' no encontrado
```

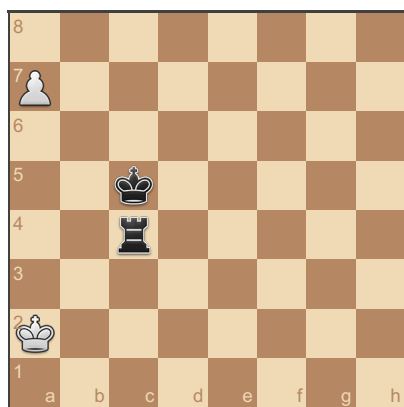
## 4 Referències

- [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]



**Figura 4:** Diagrama d'escacs. Torre i rei negres contra rei i peó blancs.

- [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\\_\(escacs\)](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]



**Figura 5:** Diagrama d'escacs. Torre i rei negres contra rei i peó blancs.



## A R code

### A.1 Preprocessament de les dades

```
#### Importar les dades: ####
dades <- read_csv("kr-vs-kp.data",
                  col_names = c("bkblk", "bkwny", "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")
print(xtable::xtable(model, caption="Model de regressió logística ajustat amb \texttt{glm}", label="t
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 <- stepwise$model
dades <- dades[c(2:ncol(dades), 1)]
```

### A.2 Naive Bayes classifier

```
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")
)

Naive <- Bayes_clasificador(training, wl, bl, 10, 666)
```

### A.3 Augmented Naive Bayes classifier

```
atributes <- colnames(dades[-ncol(dades)])
wl <- data.frame(from = rep("target", length(atributes)), to = atributes)

Augmented <- Bayes_clasificador(training, wl, bl = NULL, 10, 666)
```

```
Augmented$Metrics
```

#### A.4 Triatge del clasificador bayesià final

```
atributes <- colnames(dades[-ncol(dades)])

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

xarxa <- hc(training, score = "bic", whitelist = wl)

graphviz.plot(xarxa)
xarxa.estimada <- bn.fit(xarxa, training, method = "mle")
xarxa.grain <- suppressWarnings(as.grain(xarxa.estimada))

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

for (j in 1:nrow(test)){
  if (is.numeric(predict(xarxa.grain, response="target",
                        test[j,], predictors = attributes,
                        type = "dist")$pred[[1]][1,1])==FALSE){
    prediccio[[j]] <- NA
    CL[[j]] <- 0
    distribucio[[j]] <- c(rep(0,2))
  }
  else{
    prueba[[j]] <- predict(xarxa.grain, response="target",
                        test[j,], predictors = attributes,
                        type="dist")
    distribucio[[j]] <- prueba[[j]]$pred[[1]]
    prediccio[[j]] <- dimnames(distribucio[[j]])[[2]][which.max(distribucio[[j]])]
    CL[[j]] <- max(distribucio[[j]])
  }
}
matriu.confusio <- as.matrix(table(unlist(prediccio), test$target))
matriu.confusio

# Accuracy
Accuracy_final <- round((sum(matriu.confusio[1,1] +
                             matriu.confusio[2,2])/
                             sum(matriu.confusio))*100,2)

# True Positive Rate
TPR_final <- round((sum(matriu.confusio[1,1])/
                      sum(matriu.confusio[1,1] +
                          matriu.confusio[2,1]))*100,2)

# True Negative Rate
TNR_final <- round((sum(matriu.confusio[2,2])/
                      sum(matriu.confusio[2,2] +
                          matriu.confusio[1,2]))*100,2)
```

```

# Balanced Accuracy
baccuracy_final <- round((TPR_final+TNR_final)/2,2)

# Positive Predictive Value
PPV_final <- round((sum(matriu.confusio[1,1])/
                      sum(matriu.confusio[1,1] +
                          matriu.confusio[1,2]))*100,2)

# F1 score
F1_final <- round(2*((PPV_final*TPR_final)/(PPV_final+TPR_final)),2)

Mesures <- rbind(Accuracy_final, TPR_final, TNR_final,
                 baccuracy_final, PPV_final, F1_final)
rownames(Mesures) <- c("Accuracy", "True Positive Rate",
                      "True Negative Rate", "Balanced Accuracy",
                      "Positive Predictive Value", "F1 score")
colnames(Mesures) <- "Mètriques"
Mesures

xtable(Mesures)
#

```

## A.5 Predicció per noves partides

```

a <- lapply(dades, function(x) sample(levels(x),1))
a <- unlist(a[-c(length(a))])

a <- dades[sample(1:nrow(dades),1),]
a <- a[-c(length(a))]

a <- c(TRUE,FALSE,FALSE,FALSE,TRUE,TRUE,FALSE,FALSE,FALSE,FALSE,FALSE,FALSE,
      "g",FALSE,"n",TRUE,FALSE,FALSE,FALSE,TRUE,FALSE,FALSE,FALSE,TRUE,TRUE,
      TRUE,TRUE,FALSE,FALSE,FALSE,TRUE,FALSE,FALSE,TRUE,TRUE,"n")

a <- c(TRUE,FALSE,FALSE,FALSE,TRUE,TRUE,T,FALSE,FALSE,FALSE,FALSE,FALSE,
      "g",FALSE,"n",TRUE,FALSE,FALSE,FALSE,TRUE,FALSE,FALSE,FALSE,TRUE,TRUE,
      TRUE,TRUE,FALSE,FALSE,TRUE,TRUE,FALSE,FALSE,TRUE,TRUE,"t")

a <- c(FALSE,FALSE,FALSE,FALSE,FALSE,FALSE,FALSE,FALSE,FALSE,FALSE,FALSE,FALSE,
      "l",FALSE,"n",FALSE,FALSE,TRUE,FALSE,FALSE,FALSE,FALSE,FALSE,FALSE,
      TRUE,FALSE,FALSE,FALSE,FALSE,FALSE,FALSE,FALSE,TRUE,TRUE,"n")

xarxa.evid <- setEvidence(xarxa.grain, nodes = atributes, states = a)
evid <- xarxa.evid$evidence
qq <- querygrain(xarxa.evid, nodes = c("target"), type = "marginal")
distribucio <- qq$target
predicccio <- dimnames(distribucio)[[1]][which.max(distribucio)]
CL <- round(100*max(distribucio), 2)
reincidence.risc <- round(100*distribucio[2], 2)

Resultat <- matrix(c(predicccio, CL,reincidence.risc),nrow=1)
Resultat <- as.data.frame(Resultat)
colnames(Resultat) <- c("Predicció", "Confidence Level (CL) en %",

```

```
    "Risc de reincidència")  
evid  
Resultat
```