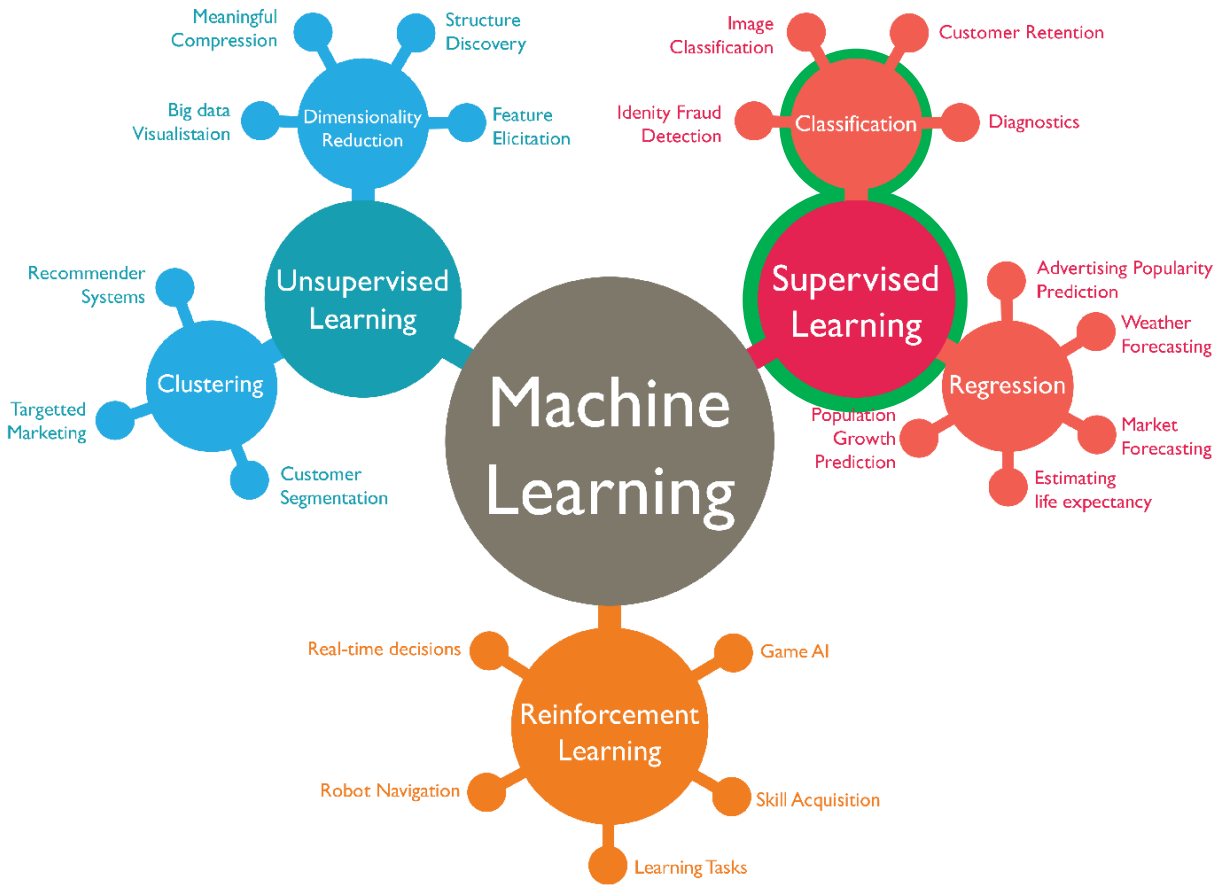
Induction by Decission Trees

Ramon Baldrich

Universitat Autònoma de Barcelona

http://www.cs.utexas.edu/~ear/nsc110/ScienceAndSociety/Lectures/AI-long.ppt
http://www.cs.utexas.edu/users/ear/nsc110/Mirrors/DSMirrorsArtificialIntelligence.ppt
http://decisiontrees.net/decision-trees-tutorial/

Another ML introduction:

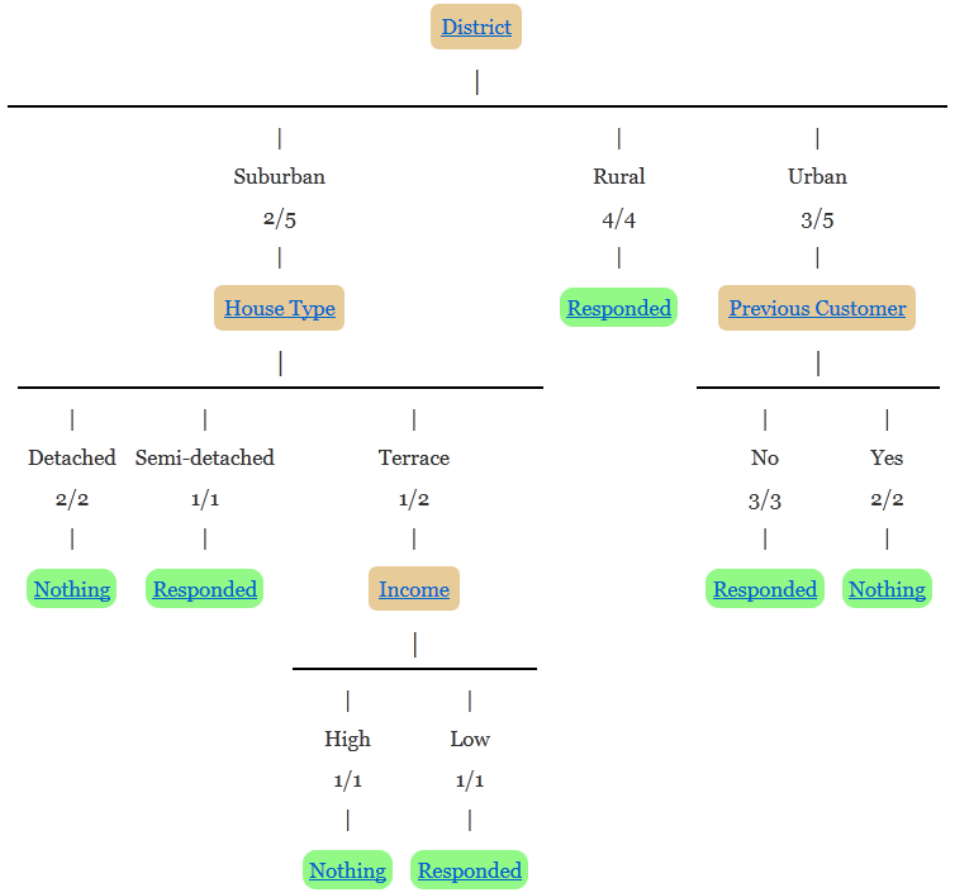


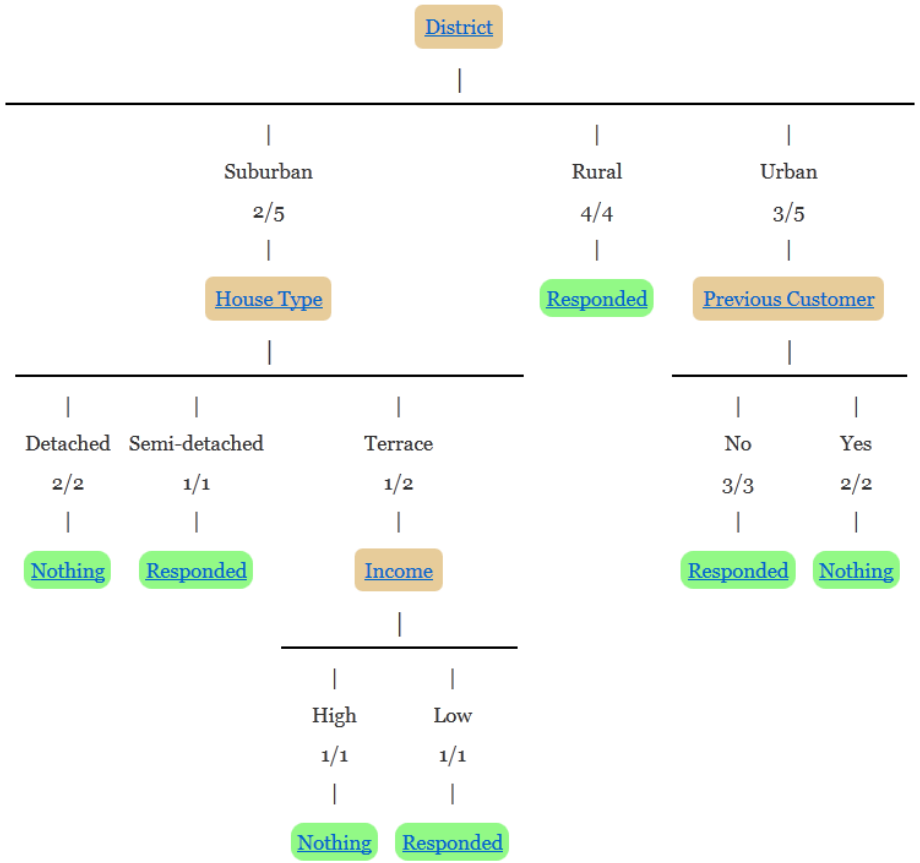
Funcionarà la publicitat?

District	House Type	Income	Previous Customer	Outcome
Suburban	Detached	High	No	Nothing
Suburban	Detached	High	Yes	Nothing
Rural	Detached	High	No	Responded
Urban	Semi-detached	High	No	Responded
Urban	Semi-detached	Low	No	Responded
Urban	Semi-detached	Low	Yes	Nothing
Rural	Semi-detached	Low	Yes	Responded
Suburban	Terrace	High	No	Nothing
Suburban	Semi-detached	Low	No	Responded
Urban	Terrace	Low	No	Responded
Suburban	Terrace	Low	Yes	Responded
Rural	Terrace	High	Yes	Responded
Rural	Detached	Low	No	Responded
Urban	Terrace	High	Yes	Nothing

Funcionarà la publicitat?

District	House Type	Income	Previous Customer	Outcome
Suburban	Detached	High	No	Nothing
Suburban	Detached	High	Yes	Nothing
Rural	Detached	High	No	Responded
Urban	Semi-detached	High	No	Responded
Urban	Semi-detached	Low	No	Responded
Urban	Semi-detached	Low	Yes	Nothing
Rural	Semi-detached	Low	Yes	Responded
Suburban	Terrace	High	No	Nothing
Suburban	Semi-detached	Low	No	Responded
Urban	Terrace	Low	No	Responded
Suburban	Terrace	Low	Yes	Responded
Rural	Terrace	High	Yes	Responded
Rural	Detached	Low	No	Responded
Urban	Terrace	High	Yes	Nothing



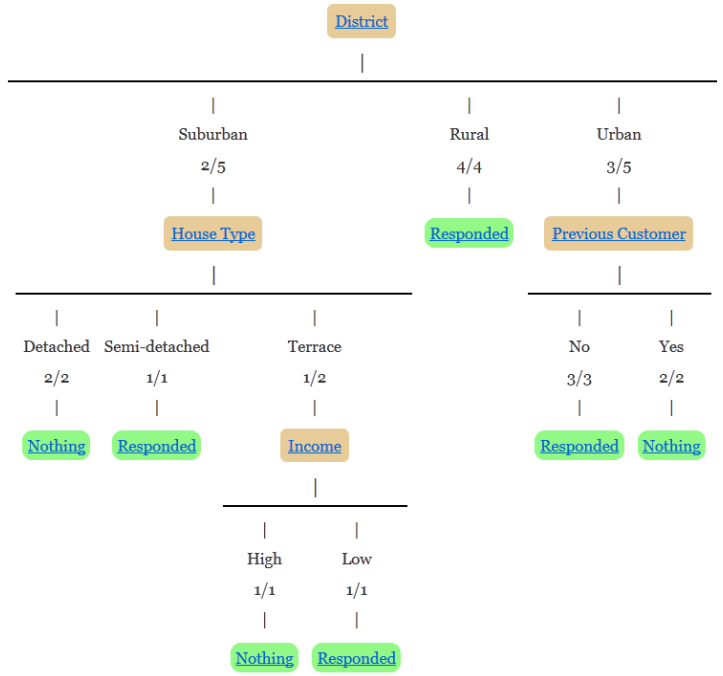


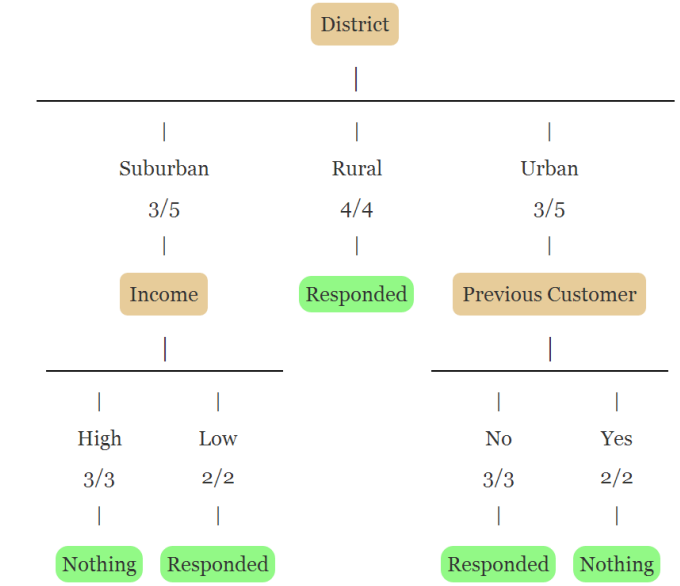
De arbre a Regles (CONEIXEMENT)

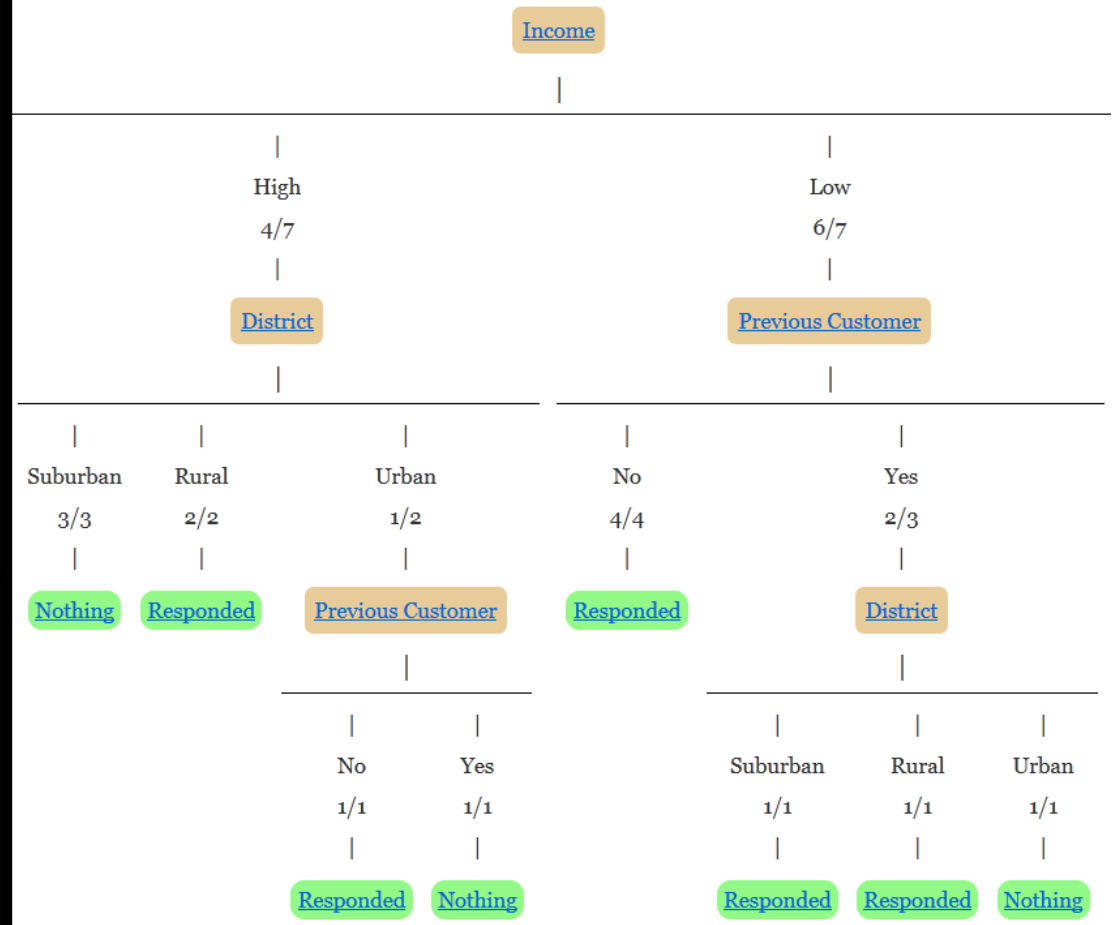
1. (District=Suburban) AND (House Type=Detached) => (Outcome = Nothing)
2. (District=Suburban) AND (House Type=Semi-Detached) => (Outcome = Reponded)
3. (District=Suburban) AND (House Type=Terrace) AND (Income=High) => (Outcome = Nothing)
4. (District=Suburban) AND (HouseType=Terrace) AND (Income=Low) => (Outcome = Responded)
5. (District=Rural) => (Outcome = Responded)
6. (District=Urban) AND (Previous Customer=No) => (Outcome = Responded)
7. (District=Urban) AND (Previous Customer=Yes) => (Outcome = Nothing)

Obtenim un classificador !!!!!!!!!!!!!!

Múltiples solucions



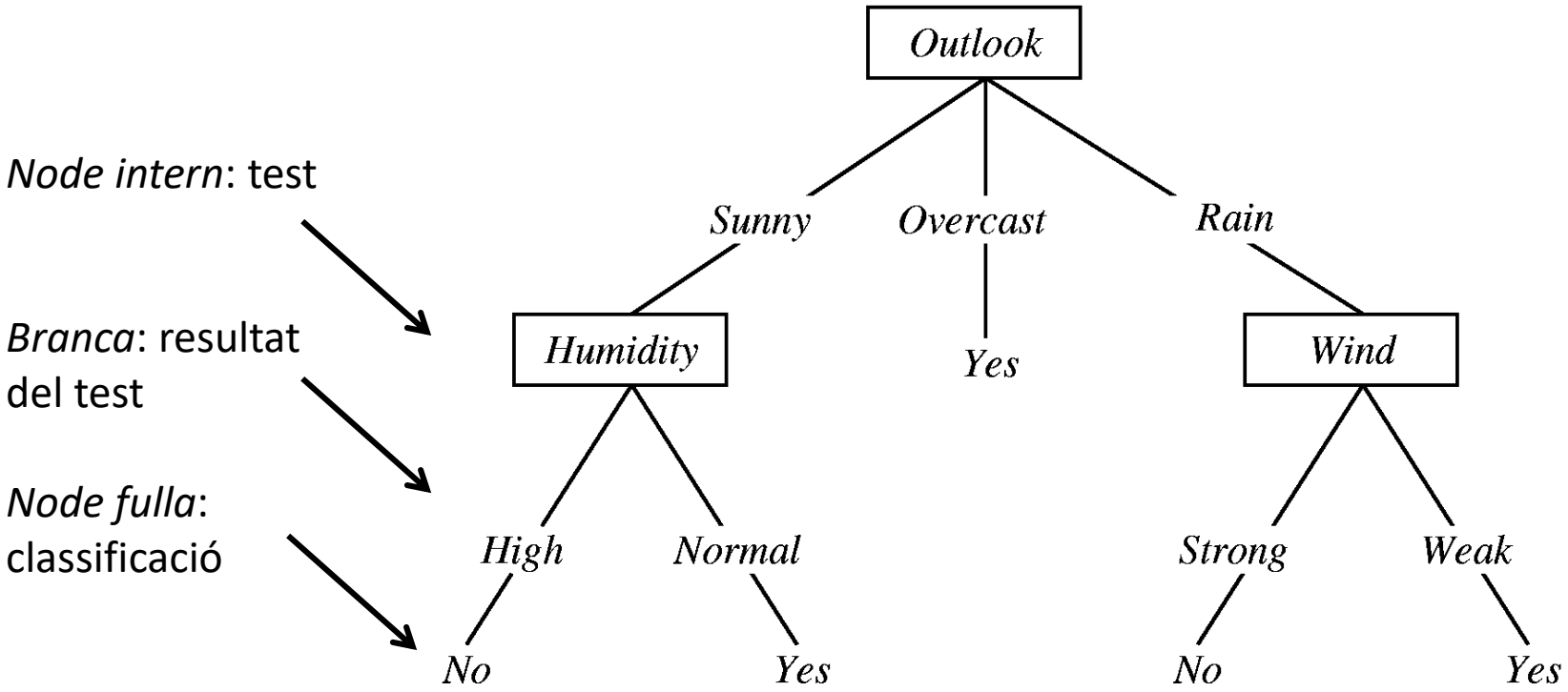




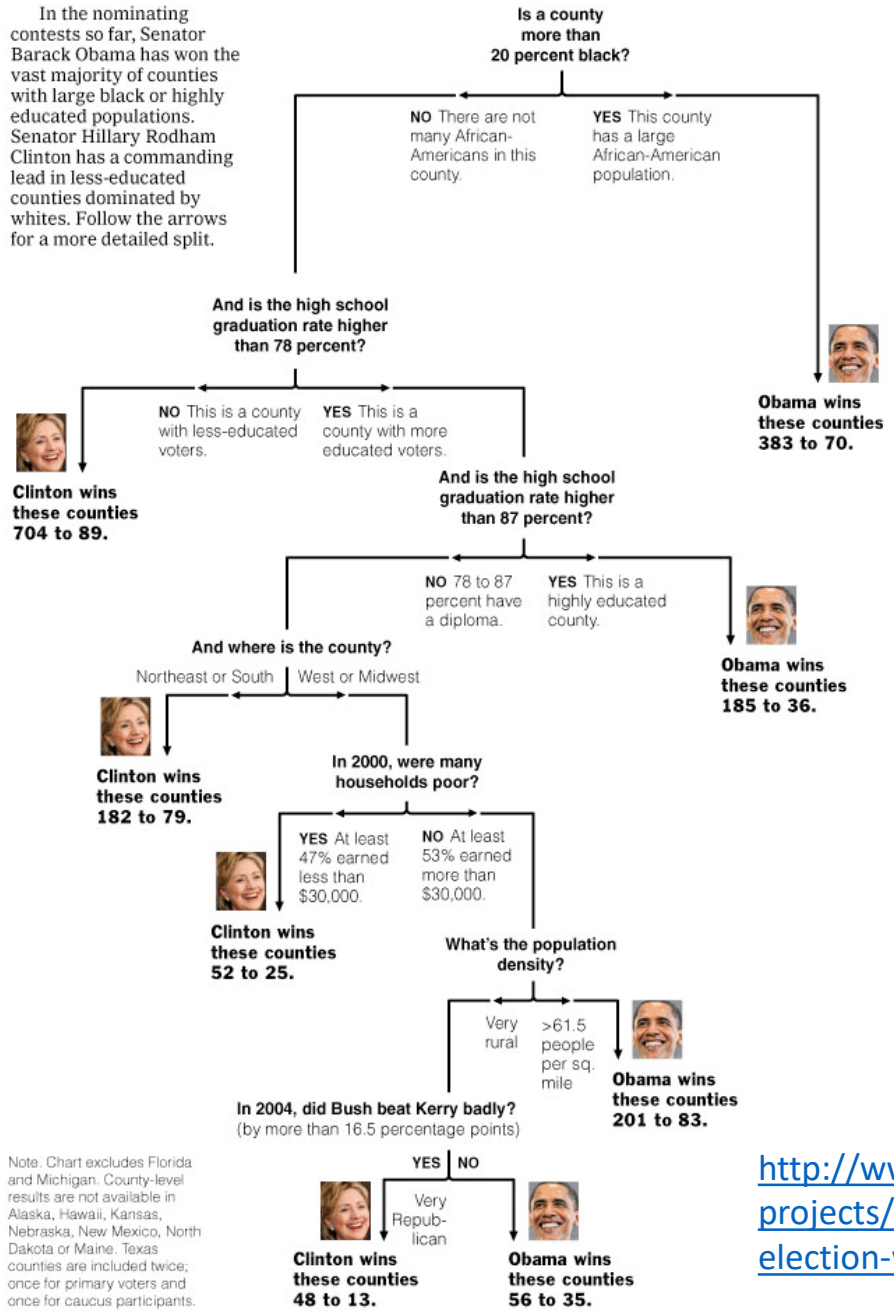
http://www.20q.net/

Arbres de Decisió

Un arbre de decisió permet processar una entitat que ha estat descrita per un conjunt de propietats i prendre una decisió.



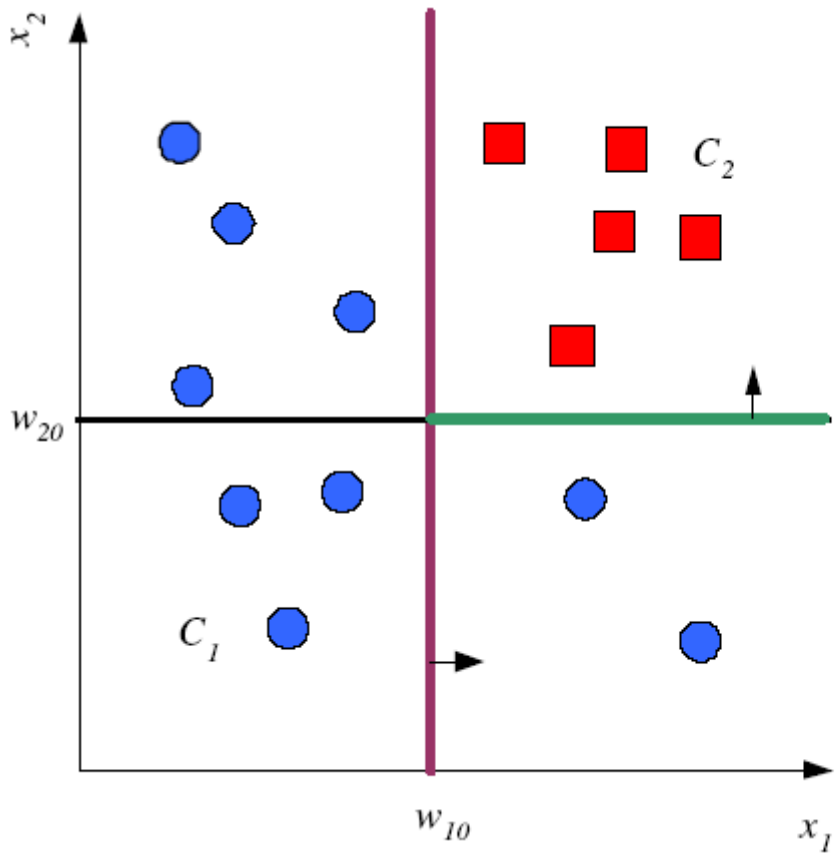
Decision Tree: The Obama-Clinton Divide

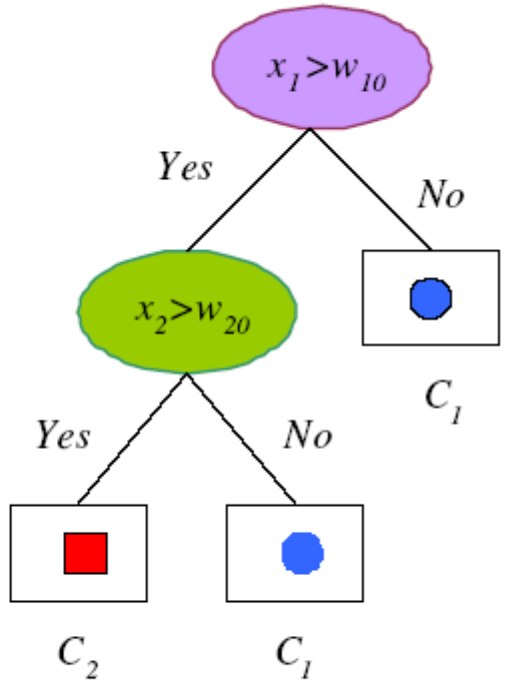


http://www.nytimes.com/imagepages/2008/04/16/us/20080416_OBAMA_GRAPHIC.html

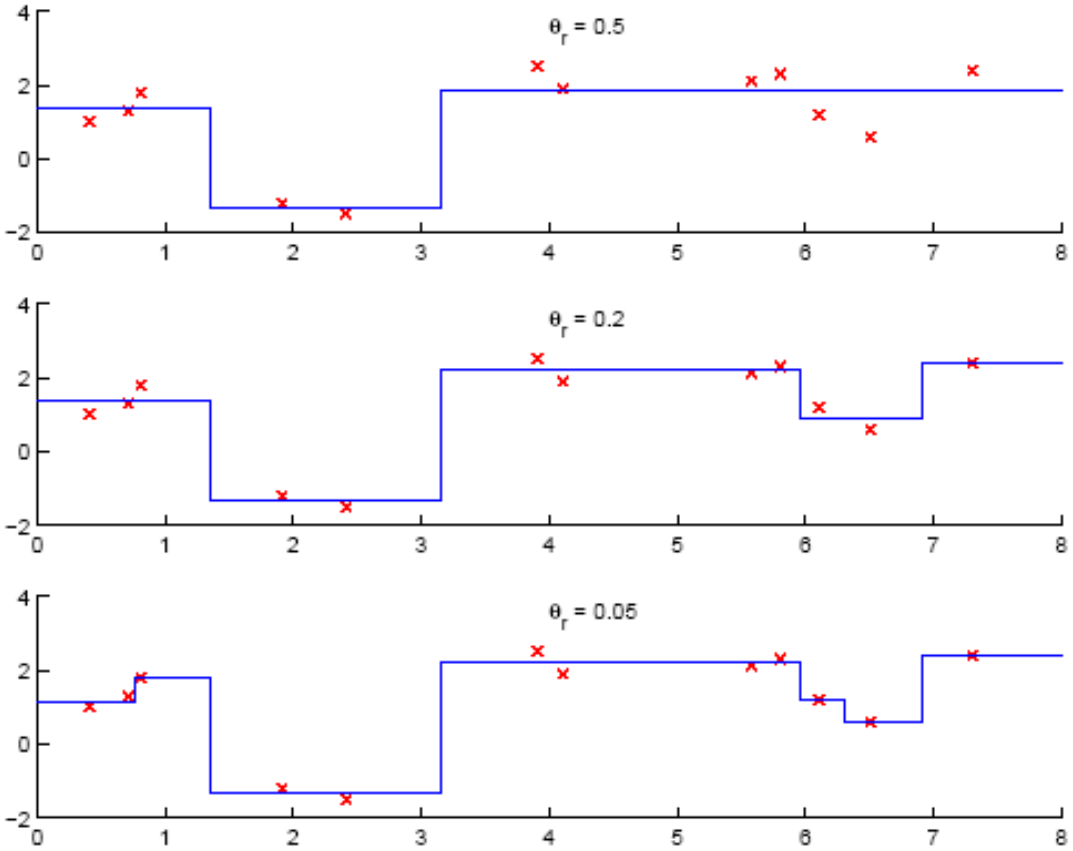
http://www.ryan-peach.com/school-projects/2017/5/22/describing-the-2016-election-with-machine-learning

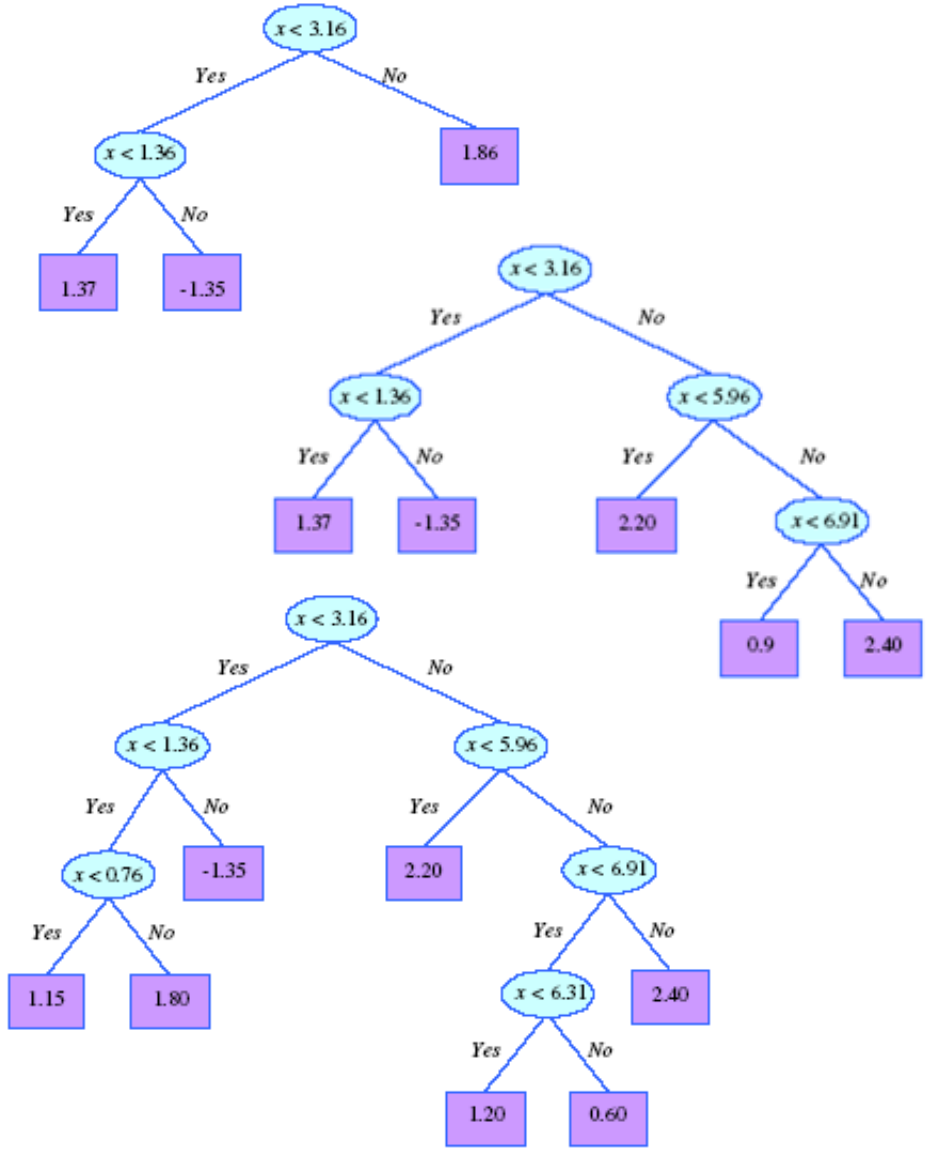
Arbres de Decisió per a la classificació





Arbres de Decisió per a la regresssió





Arbres de Decisió

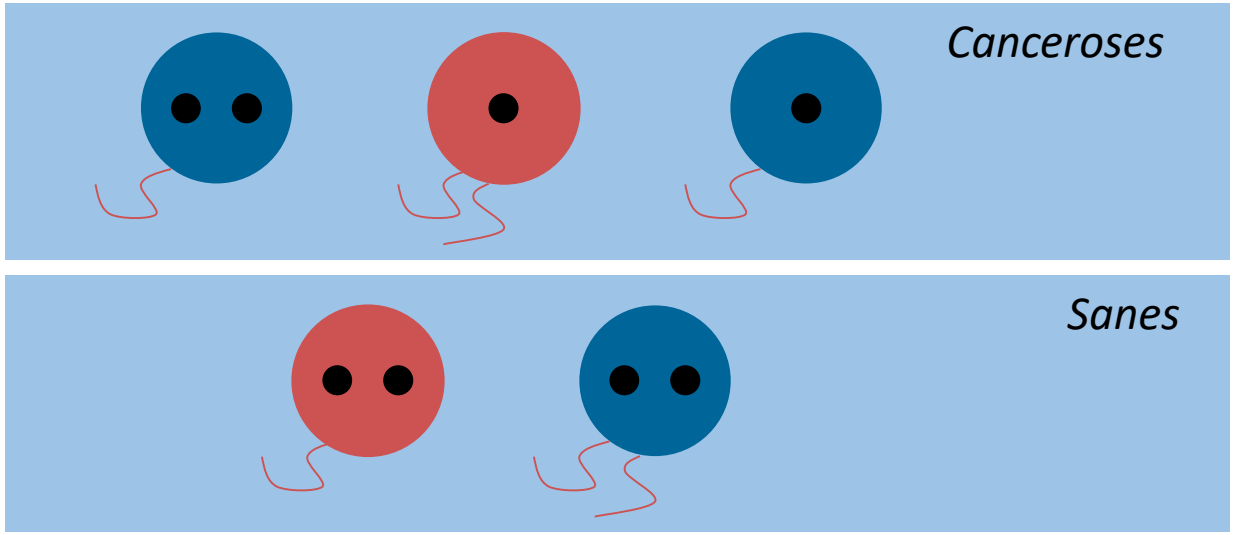
Exemple: cèl·lules cancerosa.

Classes = { sana, cancerosa }

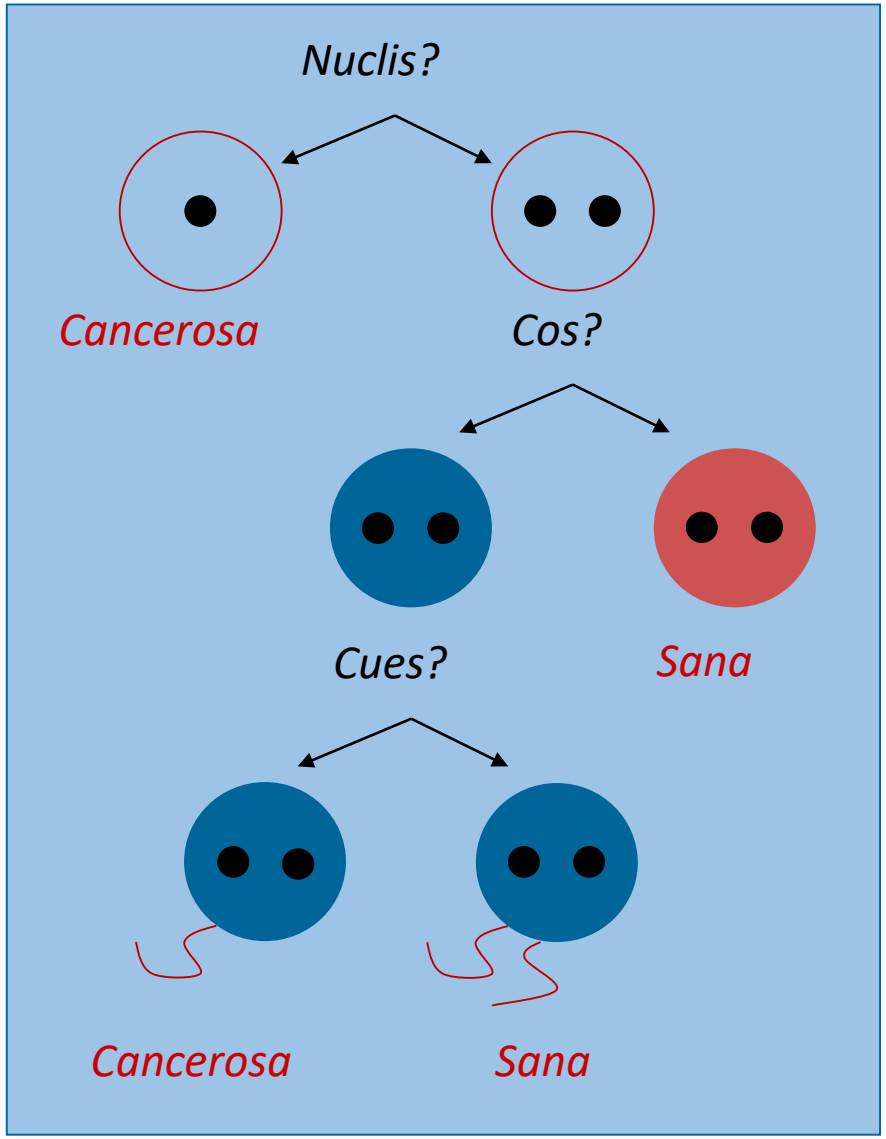
Característiques:

Cos = { blau, roig }
#Nuclis = { 1,2 }
#Cues = { 1,2 }

Exemples:



Arbres de Decisió



Arbres de Decisió

Per al cas de classificació binària, poden representat qualsevol funció booleana

$$\forall\ c,\ nucli(c,2),\ cos(c,blau),\ cua(c,1) \rightarrow cancer\ (c).$$

La taula de veritat d'una funció booleana de *n* característiques té 2^n files, i per tant hi ha 2^{2^n} funcions booleanes diferents.

Inferir un arbre de decisió és trobar una funció booleana *h* consistent dins d'aquest espai.

at1	at2	at3	at4	Class
a1	a2	a3	a4	Yes
a1	a2	a3	b4	Yes
a1	b2	a3	a4	Yes
a1	b2	b3	b4	No
a1	c2	a3	a4	Yes
a1	c2	a3	b4	No
b1	b2	b3	b4	No
c1	b2	b3	b4	No

Es generalitzable a una funció multi-classe

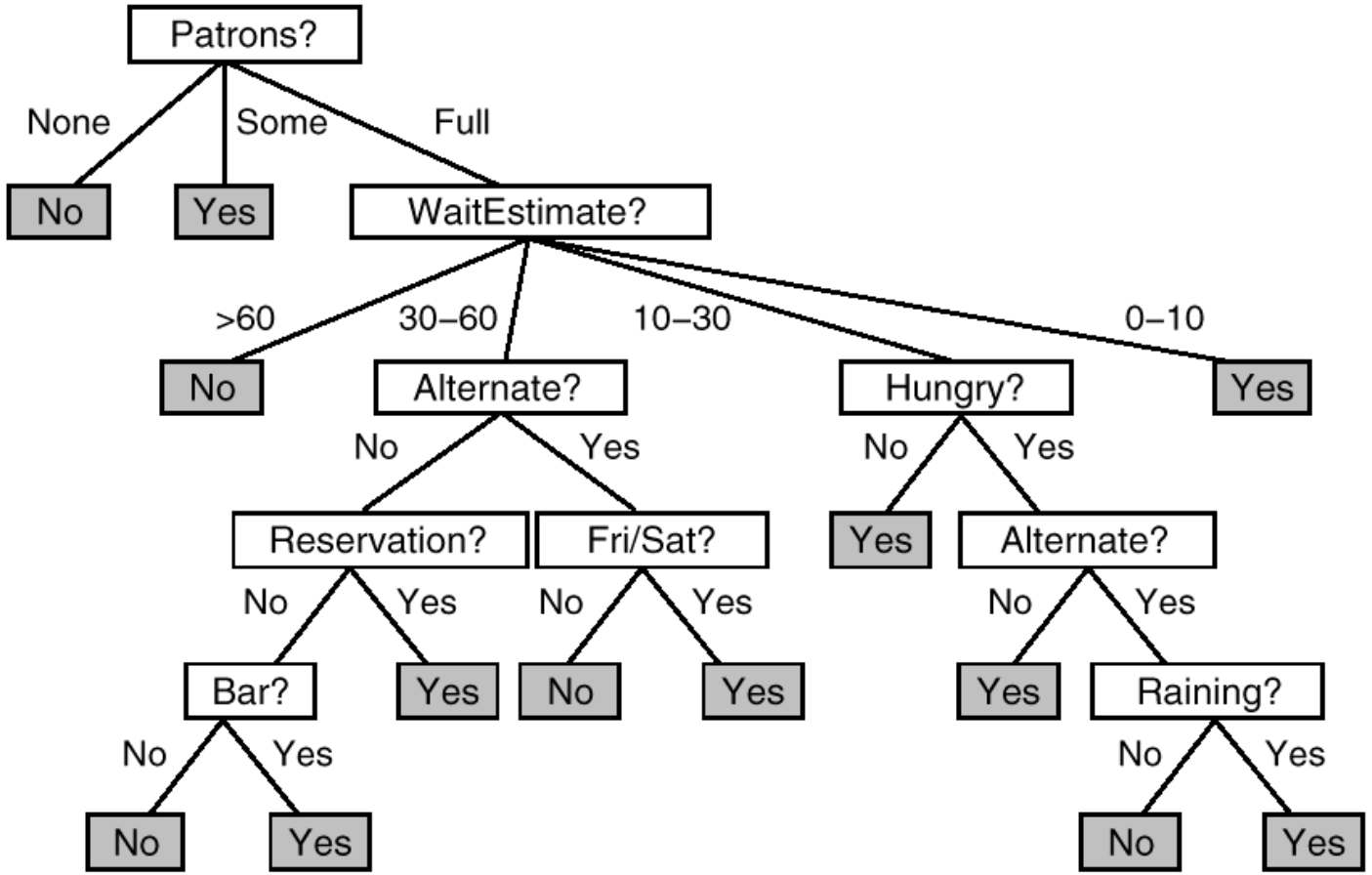
Inducció en Arbres de Decisió

Exemple 1

Ex#	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	WillWait
x1	yes	no	no	yes	some	$$$	no	yes	french	0-10	yes
x2	yes	no	no	yes	full	$	no	no	thai	30-60	no
x3	no	yes	no	no	some	$	no	no	burger	0-10	yes
x4	yes	no	yes	yes	full	$	no	no	thai	10-30	yes
x5	yes	no	yes	no	full	$$$	no	yes	french	>60	no
x6	no	yes	no	yes	some	$$	yes	yes	italian	0-10	yes
x7	no	yes	no	no	none	$	yes	no	burger	0-10	no
x8	no	no	no	yes	some	$$	yes	yes	thai	0-10	yes
x9	no	yes	yes	no	full	$	yes	no	burger	>60	no
x10	yes	yes	yes	yes	full	$$$	no	yes	italian	10-30	no
x11	no	no	no	no	none	$	no	no	thai	0-10	no
x12	yes	yes	yes	yes	full	$	no	no	burger	30-60	yes

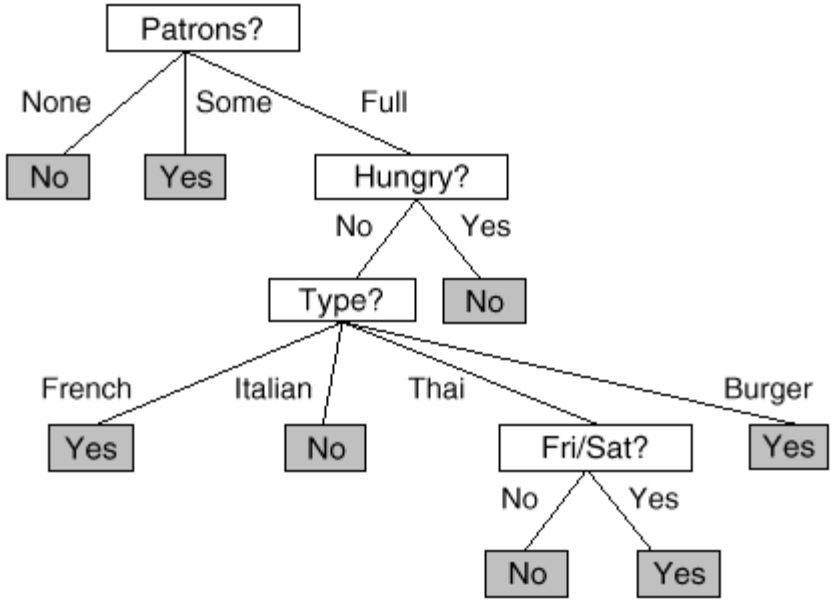
Inducció en Arbres de Decisió

Exemple 1



Inducció en Arbres de Decisió

Exemple 1



És un arbre diferent de l'anterior, però consistent i més curt !

Inducció en Arbres de Decisió

Hi ha molts arbres consistents...

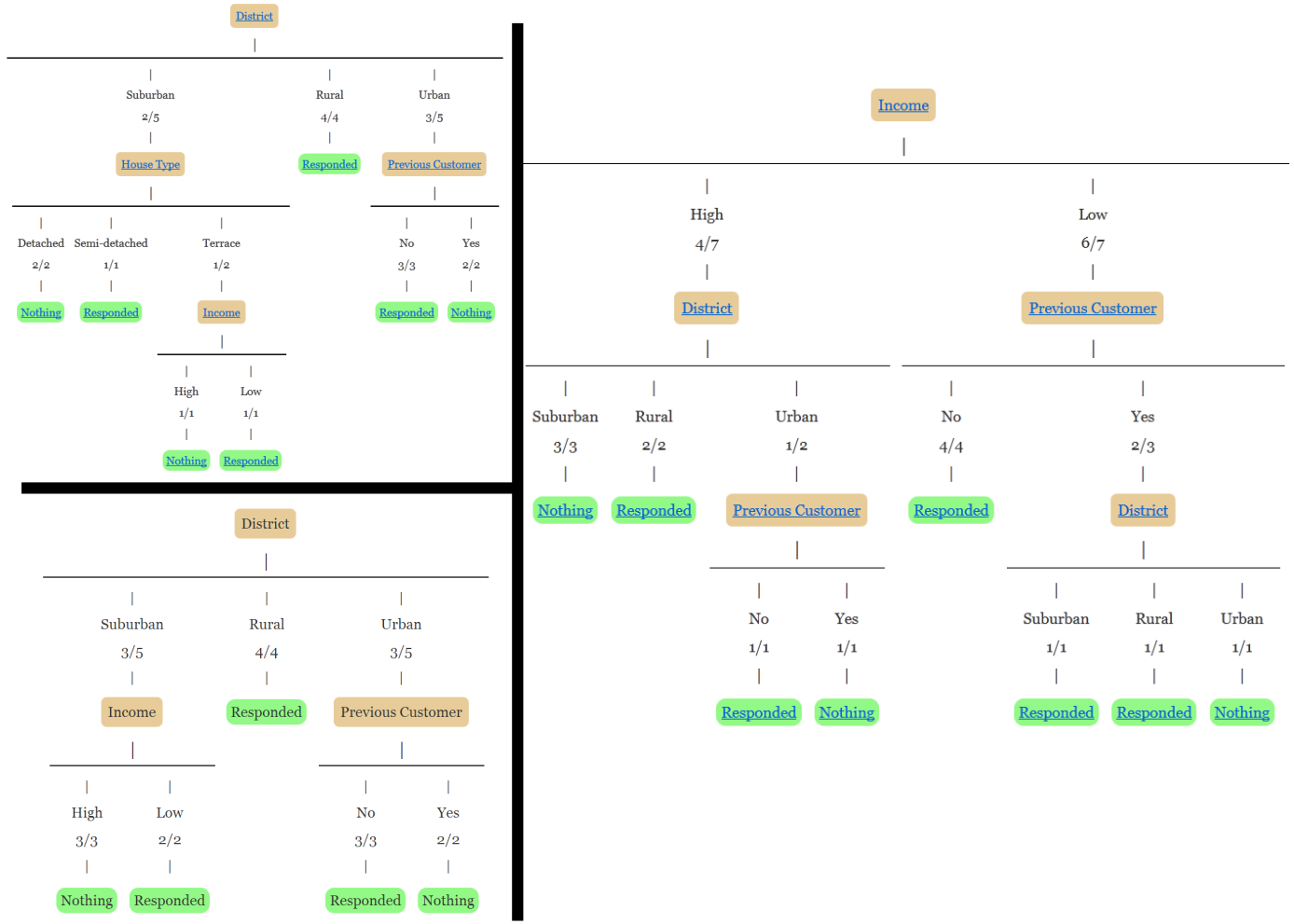
Trobar un arbre de decisió que sigui consistent amb els exemples i,
al mateix temps, el més petit possible.

... però trobar-lo és NP complert (Quinlan 1986):

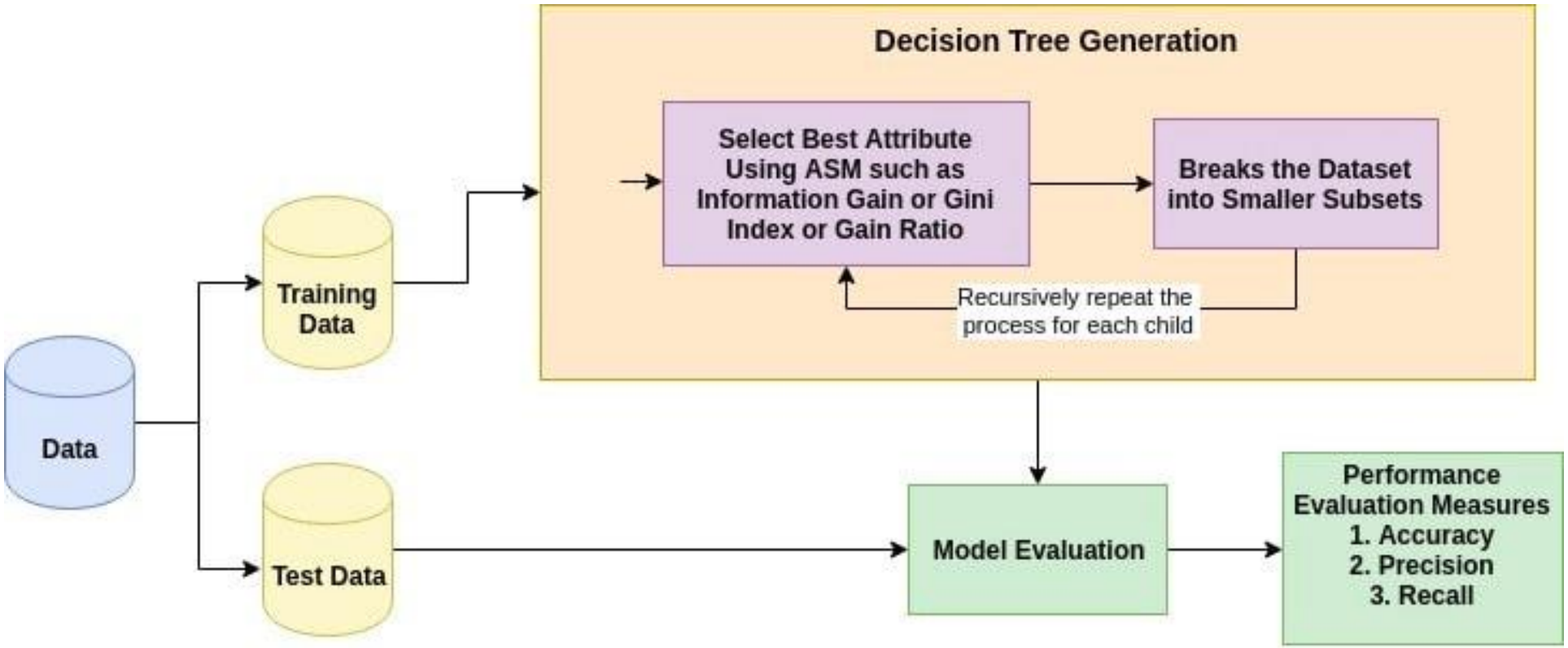
- aplicar algorismes de cerca local per trobar una solució (raonable)
- L'aprenentatge és ***greedy****:* trobar la millor partició de l'arbre recursivament
- però, per on començar?

Inducció en Arbres de Decisió

Navalla d'Ockham: La hipòtesi o solució a un problema **més simple** hauria de ser la més versemblant.



Once again


Decision Tree Generation
Select Best Attribute Using ASM such as Information Gain or Gini Index or Gain Ratio
Breaks the Dataset into Smaller Subsets
Recursively repeat the process for each child
Data
Training Data
Test Data
Model Evaluation
Performance Evaluation Measures
1. Accuracy
2. Precision
3. Recall

Inducció en Arbres de Decisió

Construcció dels arbres de decisió

Estratègia greedy (problema NP)

Algorisme “divide & conquer”:

- Comencem amb tots els exemples d’entrenament a l’arrel de l’arbre.
- Els exemples es van dividint en funció del atribut que es selecciona per a ramificar l’arbre en cada node.
- Els atributs que s’usen per a ramificar es trien en funció d’una heurística.

```
TreeGrowing (S,A,y,SplitCriterion,StoppingCriterion)
   on:
         S - Training Set (conjunt entrenament)
         A – Input Feature Set (Conjunt de caracteristiques d'entrada)
         y - Target Feature (característica objectiu)
         SplitCriterion – mètode per avaluar la divisió de nodes
         StoppingCriterion – el criteri per parar el process de construcció

   Crear un now arbre T amb un sol node arrel.
   IF StoppingCriterion(S) THEN
      Marcar T com una fulla amb el valor més comú de y en S com a etiqueta.
   ELSE
      Trobar l'atribut a que obté el millor SplitCriterion(a_i,S).
      Etiquetar t amb a
      FOR cada valor v_i de a:
         Set Subtree_i= TreeGrowing (S{a= v_i}, A, y)
         Conectar el node arrel de t_T a Subtree_i amb una aresta etiquetada com v_i
      END FOR
   END IF
   RETURN TreePruning (S,T,y)

TreePruning (S,T,y)
   on:
         S - Training Set
         y - Target Feature
         T – l'arbre per a podar
   DO
      Seleccionar un node t en T de manera que al podarlor es millora maximament
      algun criteri d'avaluació
      IF t =Ø THEN T = pruned(T, t)
   UNTIL t = Ø
   RETURN T
```

```
TreeGrowing (S,A,y,SplitCriterion,StoppingCriterion)
   on:
         S - Training Set (conjunt entrenament)
         A – Input Feature Set (Conjunt de caracteristiques d'entrada)
         y - Target Feature (característica objectiu)
         SplitCriterion – mètode per avaluar la divisió de nodes
         StoppingCriterion – el criteri per parar el process de construcció

   Crear un now arbre T amb un sol node arrel.
   IF StoppingCriterion(S) THEN
      Marcar T com una fulla amb el valor més comú de y en S com a etiqueta.
   ELSE
      Trobar l'atribut a que obté el millor SplitCriterion(a_i,S).
      Etiquetar t amb a
      FOR cada valor v_i de a:
         Set Subtree_i= TreeGrowing (S{a= v_i}, A, y)
         Conectar el node arrel de t_T a Subtree_i amb una aresta etiquetada com v_i
      END FOR
   END IF
   RETURN TreePruning (S,T,y)

TreePruning (S,T,y)
   on:
         S - Training Set
         y - Target Feature
         T – l'arbre per a podar
   DO
      Seleccionar un node t en T de manera que al podarlor es millora maximament
      algun criteri d'avaluació
      IF t =Ø THEN T = pruned(T, t)
   UNTIL t = Ø
   RETURN T
```

Inducció en Arbres de Decisió

Construcció dels arbres de decisió

Criteris de parada: quan hem de parar la construcció del arbre de decisió?

- Quan tots els exemples que quedin pertanyin a la mateixa classe (s'afegeix una fulla a l'arbre amb l'etiqueta de la classe).
- Quan no quedin atributs pels que ramificar (s'afegeix una fulla etiquetada amb la classe més freqüent en el node).
- Quan no quedin dades per classificar.

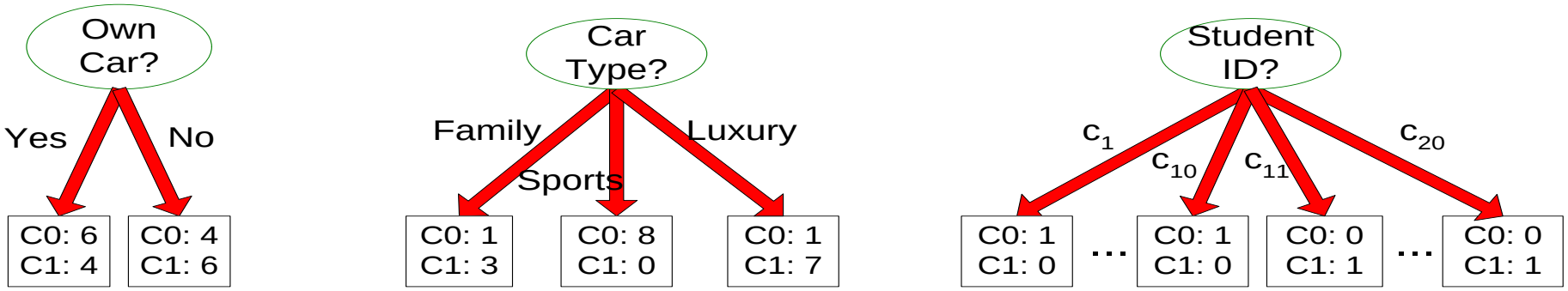
```
TreeGrowing (S,A,y,SplitCriterion,StoppingCriterion)
   on:
         S - Training Set (conjunt entrenament)
         A – Input Feature Set (Conjunt de caracteristiques d'entrada)
         y - Target Feature (característica objectiu)
         SplitCriterion – mètode per avaluar la divisió de nodes
         StoppingCriterion – el criteri per parar el process de construcció

   Crear un now arbre T amb un sol node arrel.
   IF StoppingCriterion(S) THEN
      Marcar T com una fulla amb el valor més comú de y en S com a etiqueta.
   ELSE
      Trobar l'atribut a que obté el millor SplitCriterion(a_i,S).
      Etiquetar t amb a
      FOR cada valor v_i de a:
         Set Subtree_i= TreeGrowing (S{a= v_i}, A, y)
         Conectar el node arrel de t_T a Subtree_i amb una aresta etiquetada com v_i
      END FOR
   END IF
   RETURN TreePruning (S,T,y)

TreePruning (S,T,y)
   on:
         S - Training Set
         y - Target Feature
         T – l'arbre per a podar
   DO
      Seleccionar un node t en T de manera que al podarlor es millora maximament
      algun criteri d'avaluació
      IF t =Ø THEN T = pruned(T, t)
   UNTIL t = Ø
   RETURN T
```

Inducció en Arbres de Decisió

Construcció dels arbres de decisió

Quines heurísitques es poden utilitzar per a decidir com ramificar l'arbre?

Quina és millor?



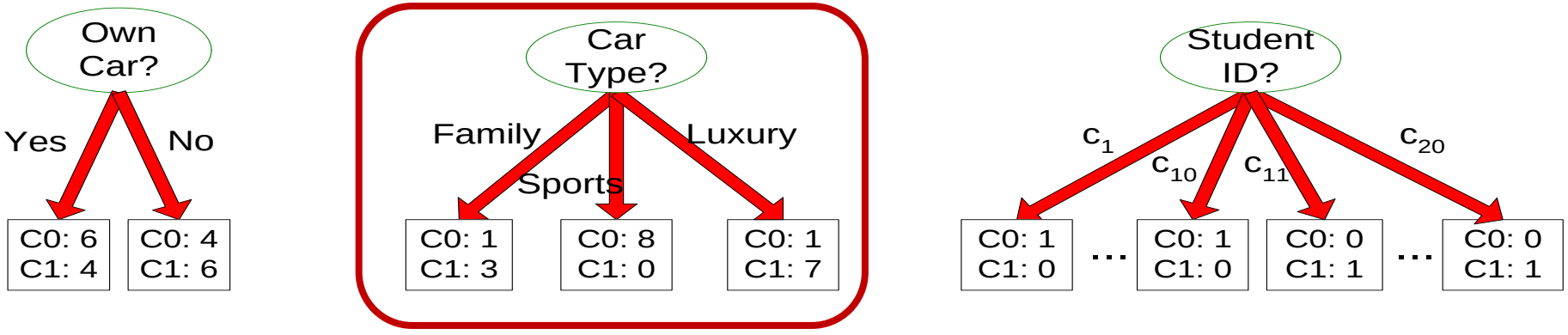
Inducció en Arbres de Decisió

Construcció dels arbres de decisió

Quines heurísitques es poden utilitzar per a decidir com ramificar l'arbre?

Quina és millor?

La que ens proporciona nodes més homogenis



Necesitem mesurar la impuresa d'un node

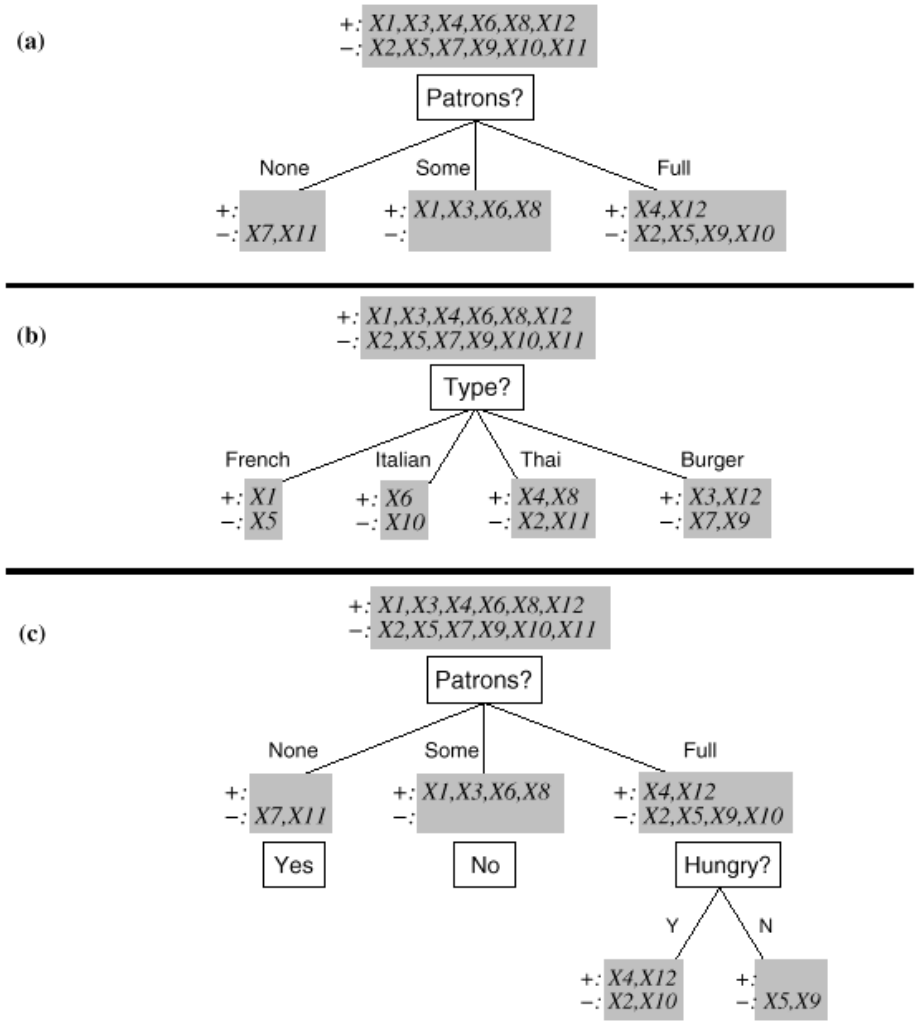
Inducció en Arbres de Decisió

Exemple 1

Bon Atribut!

Mal Atribut!

Bon Atribut
donat el primer!



Inducció en Arbres de Decisió

Construcció d'arbres de decissió

Criteris de divisió (heurístiques per a la selecció d'atributs):

- Guany d'informació (ID3, C4.5)
- Índex de Gini (CART, SLIQ, SPRINT)
- Existeixen altres regles de divisió:
  χ^2, MDL (Minimum Description Length)…

Inducció en Arbres de Decisió

Mètode recursiu:

1. Si resten exemples positius i negatius, escollir el millor atribut per dividir-los.

2. Si tots els exemples són positius o negatius ja hem acabat i podem respondre.

3. Si no hi ha exemples vol dir que no hem observat cap exemple d'aquest tipus: retornem un valor per defecte calculat a partir de la classificació majoritària del node pare.

4. Si no hi ha atributs i queden exemples positius i negatius tenim un problema: els exemples tenen la mateixa descripció però diferent classificació. Hi ha soroll a les dades!

Inducció en Arbres de Decisió: Algorisme ID3

Criteri de divisió: **Guany d'Informació** basat en l'entropia:

P(x) Estimació de la probabilitat que un exemple de S pertany a la classe C_x

Entropía

$$Entropia(S) = -\sum_{x=1}^{m} p(x)\log_2(p(x))$$

(informació necessaria per a classificar un exemple en S)

Informació necessaria per a classificar S després d'usar l'atribut A per a dividir S en n particions:

$$Entropia(S,A) = \sum_{v \in A} \frac{|S_v|}{|S|} Entropia(S_v)$$

Guany obtingut al ramificar usant l'atribut A:

$$Gain(S,A) = Entropia(S) - Entropia(S,A)$$

Entropia: mesura d'incertesa

Exemple de la teoria de codificació

Varible aleatoria x discreta amb 8 possibles estats; quants bits es necessiten per transmetre l'estat de x?

1. Tots els estats son equiprobables?

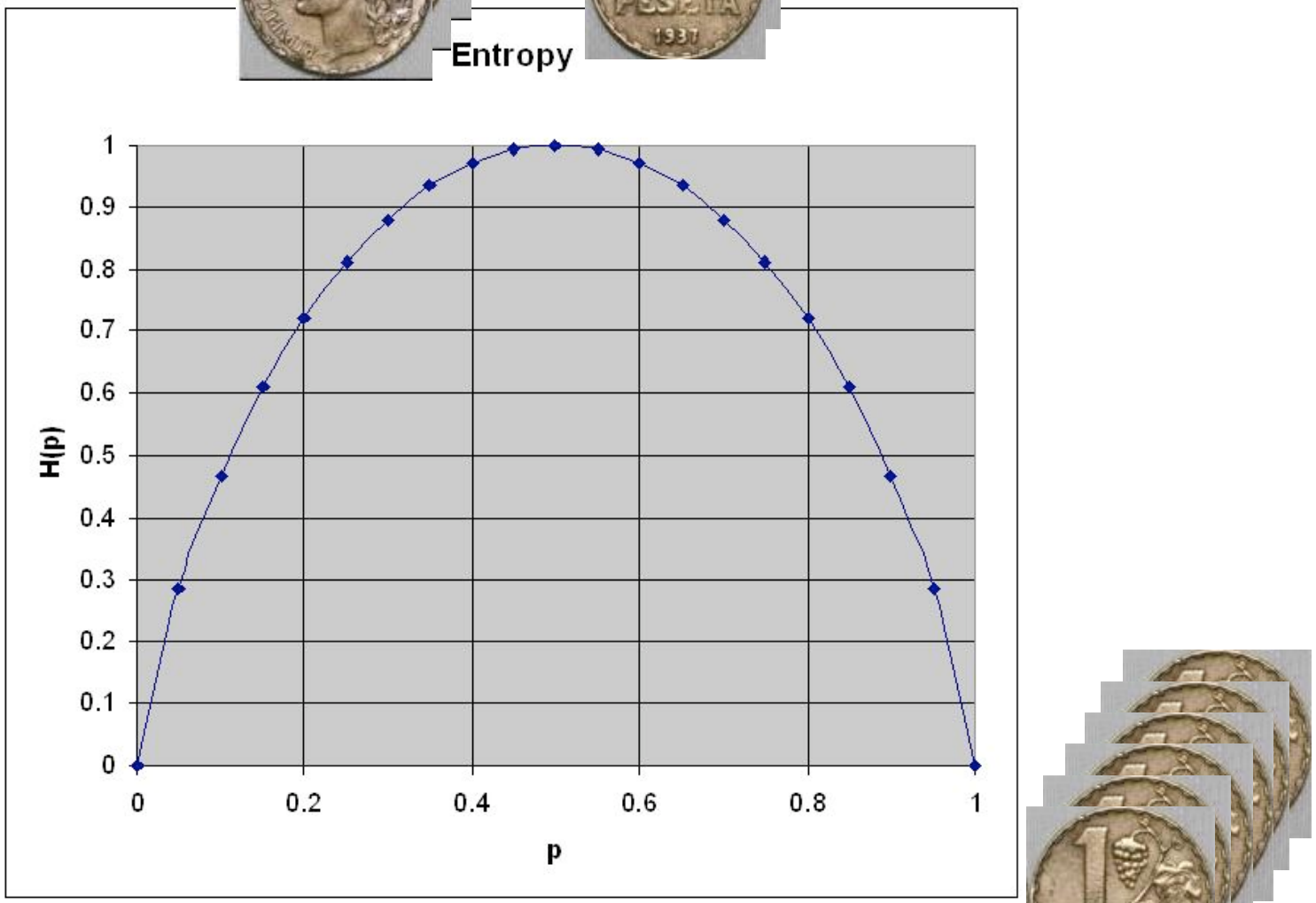
$$\mathrm{H}[x] = -8 \times \frac{1}{8} \log_2 \frac{1}{8} = 3 \text{ bits.}$$

2. Tenim la següent distribució per a x?

x	a	b	c	d	e	f	g	h
$p(x)$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{64}$	$\frac{1}{64}$	$\frac{1}{64}$	$\frac{1}{64}$
code	0	10	110	1110	111100	111101	111110	111111

$$\begin{aligned} \mathrm{H}[x] &= -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{4}\log_2\frac{1}{4} - \frac{1}{8}\log_2\frac{1}{8} - \frac{1}{16}\log_2\frac{1}{16} - \frac{4}{64}\log_2\frac{1}{64} \\ &= 2 \text{ bits} \end{aligned}$$


1
PESETA
1937


Entropy
H(p)
p
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
0.2
0.4
0.6
0.8
1

1
PESETA
1937

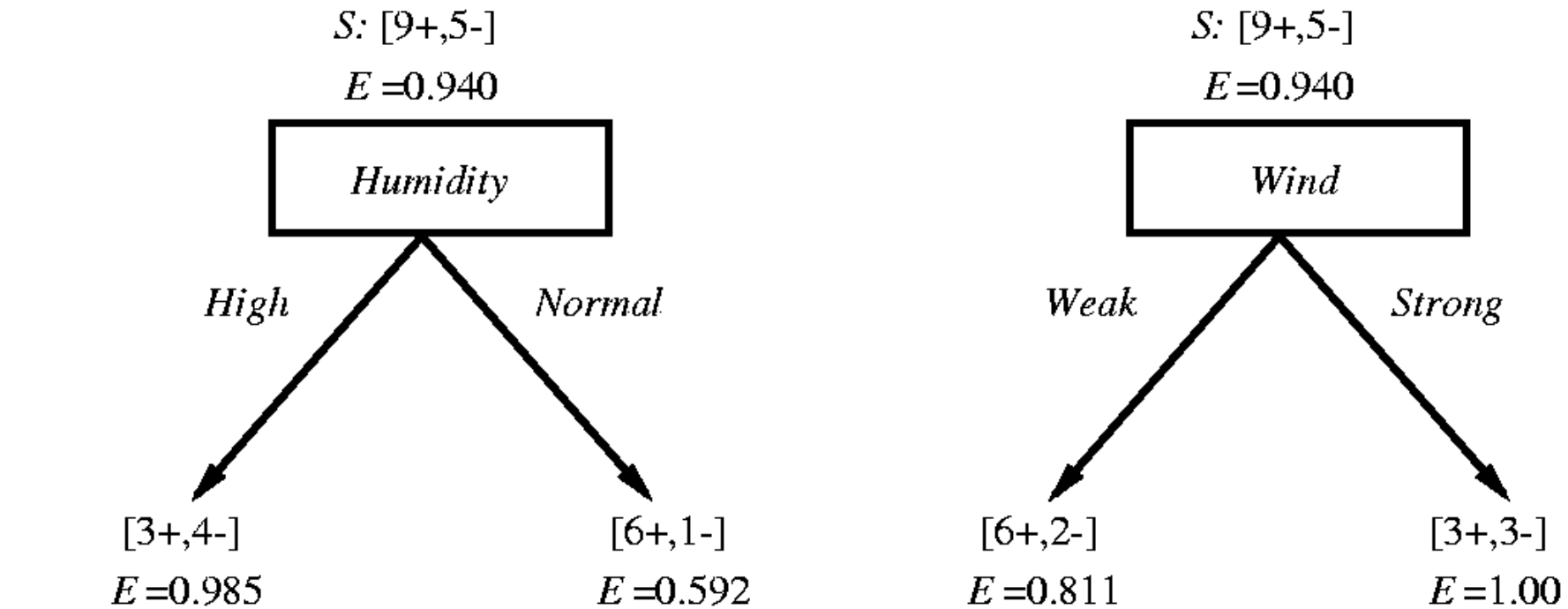
Inducció en Arbres de Decisió: Algorisme ID3

Exemple

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Inducció en Arbres de Decisió: Algorisme ID3

Quin atribut és el millor classificador?



Gain (S, Humidity)

= .940 - (7/14).985 - (7/14).592
= .151

Gain (S, Wind)

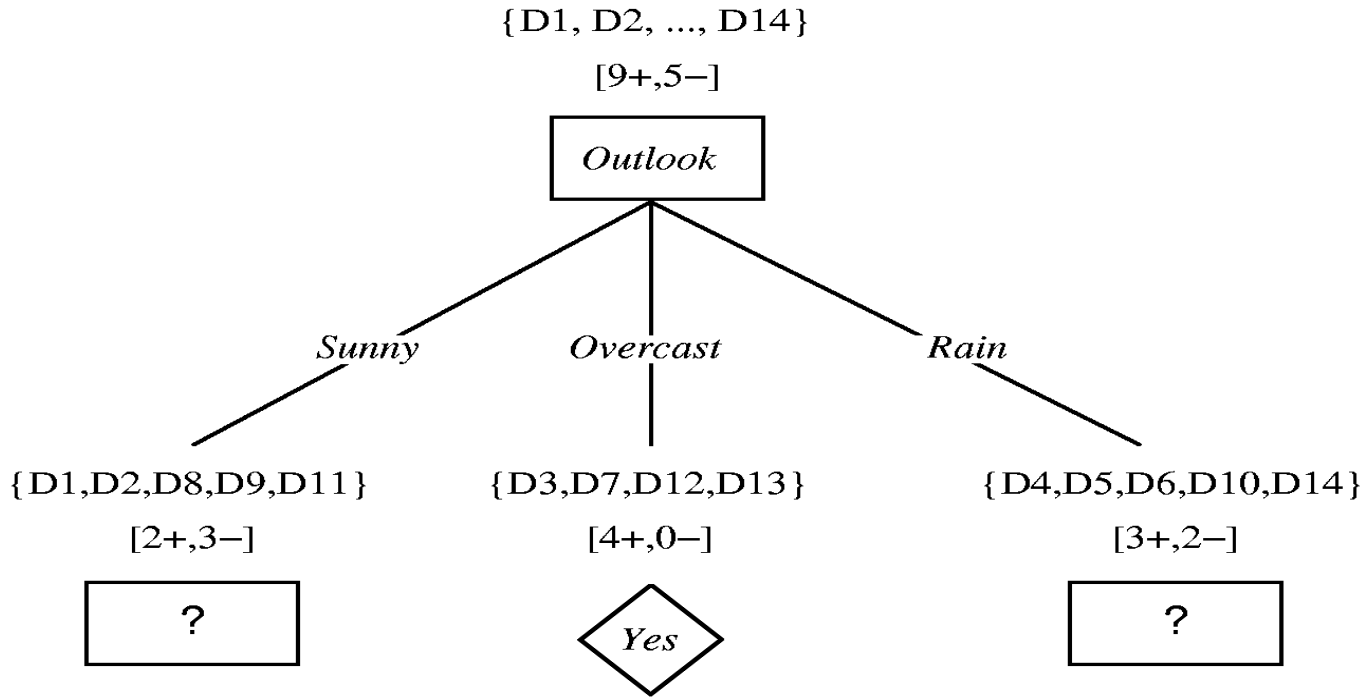
= .940 - (8/14).811 - (6/14)1.0
= .048

Selecccionariem l'atribut *Humitat* per a dividir el node arrel ja que té més guany de d'informació

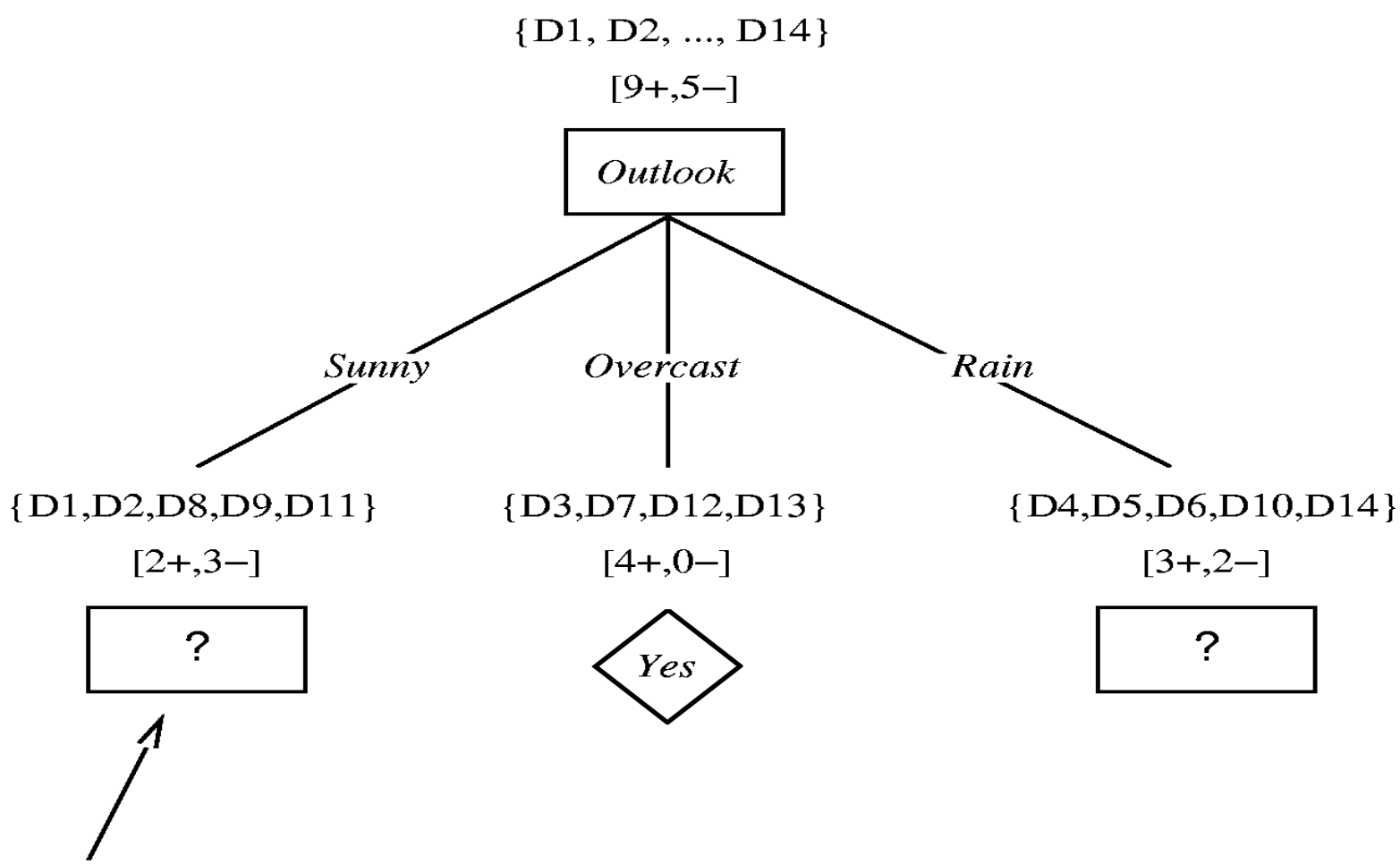
Inducció en Arbres de Decisió: Algorisme ID3

Seleccionem el següent atribut

- Calculem el guany d'informació per a cada atribut, seleccionem l'atribut *Outlook* com a primer test, resultant en el següent arbre:



- Repetim el mateix procés recursivament, fins que es compleixi algun criteri de parada.



Quin atribut hauríem de probar aquí?

S_{sunny} = {D1,D2,D8,D9,D11}

$Gain\ (S_{sunny}, Humidity) = .970 - (3/5)\ 0.0 - (2/5)\ 0.0 = .970$

$Gain\ (S_{sunny}, Temperature) = .970 - (2/5)\ 0.0 - (2/5)\ 1.0 - (1/5)\ 0.0 = .570$

$Gain\ (S_{sunny}, Wind) = .970 - (2/5)\ 1.0 - (3/5)\ .918 = .019$

Inducció en Arbres de Decisió: Algorisme C4.5

Criteri de divisió: **proporció del guany (Gain Ratio, C4.5)**

ID3 tendeix a ramificar l'arbre utilizant els atributs que tenen més valors diferents, per lo que es "normalitza" el guany d'informació usant l'entropía de la partició (que serà major com més particions petites hi hagi):

$$SplitInfo(S,A) = -\sum_{v \in A} \frac{|S_v|}{|S|} \log_2 \left(\frac{|S_v|}{|S|} \right)$$

$$GainRatio(S,A) = \frac{Gain(S,A)}{SplitInfo(S,A)}$$

Inducció en Arbres de Decisió:

Algorismes CART, SLIQ, SPRINT

Criteri de divisió: **Índex de Gini**

És una mesura estadística d'impuresa

$$Gini(y,S) = 1 - \sum_{v \in dom(y)} p(v \mid S)^2 = 1 - \sum_{v \in dom(y)} \left(\frac{|S_v|}{|S|} \right)^2$$

$$GiniGain(S,A) = Gini(y,S) - \sum_{v \in A} \frac{|S_v|}{|S|} Gini(y,S_v)$$

Per a construir l'arbre, triem l'atribut que proporciona major reducció d'impuresa

C1	**0**
C2	**6**
Gini=0.000	

C1	**1**
C2	**5**
Gini=0.278	

C1	**2**
C2	**4**
Gini=0.444	

C1	**3**
C2	**3**
Gini=0.500	

Inducció en Arbres de Decisió

Comparació de criteris de divisió

- Criteri de Guany d'informació (Gain)

  Esbiaixat cap a atributs amb molts valors diferents.

- Criteri de proporció de guany (Gain Ratio)

  Tendeix a preferir particions poc balancejades (amb una partició molt més gran que les altres)

- Criteri d'Índex de Gini

  Funciona pitjor quan hi ha moltes classes i tendeix a afavorir particions de tamany i impuresa similars.

Cap criteri de divisió és significativament millor que els demés

Inducció en Arbres de Decisió

Altres aspectes

- Arbres binàris o n-aris?
  (CART binari; C4.5 n-ari per a atributs categòrics, binari per a atributs continuus)

- Manipulació d'atributs continuus
  (selecció del conjunt de tests candidats per a ramificar l'arbre, p.ex. discretització previa)

- Manipulació de valors nuls
  (com es tracten els valors nuls/desconeguts)

Avaluació d'un classificador. Són bones les decisions?

ID code	Outlook	Temperature	Humidity	Windy	Play
a	Sunny	Hot	High	False	No
b	Sunny	Hot	High	True	No
c	Overcast	Hot	High	False	Yes
d	Rainy	Mild	High	False	Yes
e	Rainy	Cool	Normal	False	Yes
f	Rainy	Cool	Normal	True	No
g	Overcast	Cool	Normal	True	Yes
h	Sunny	Mild	High	False	No
i	Sunny	Cool	Normal	False	Yes
j	Rainy	Mild	Normal	False	Yes
k	Sunny	Mild	Normal	True	Yes
l	Overcast	Mild	High	True	Yes
m	Overcast	Hot	Normal	False	Yes
n	Rainy	Mild	High	True	No

- Gain(*outlook*) = 0.940 bits – 0.693 bits = 0.247 bits.
- Gain(*temperature*) = 0.029 bits.
- Gain(*humidity*) = 0.152 bits.
- Gain(*windy*) = 0.048 bits.

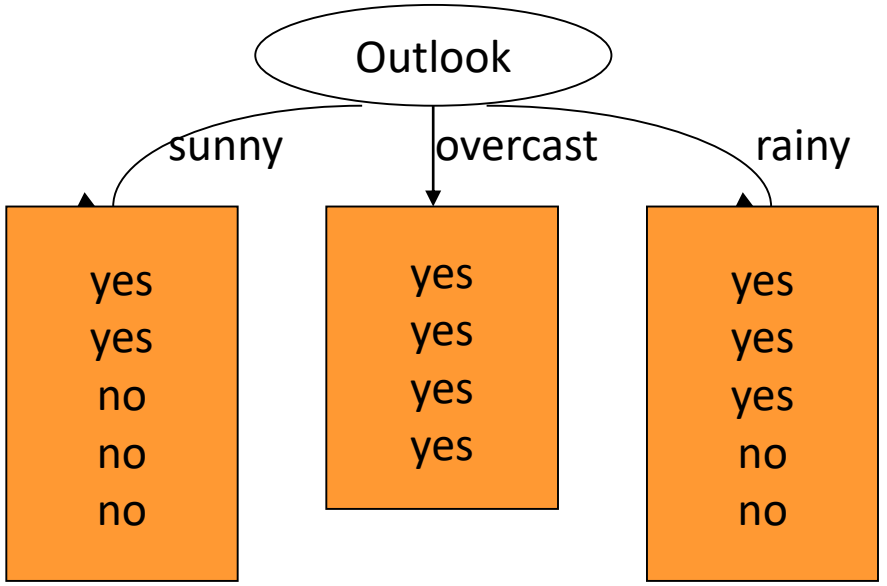
Si coneixem el resultat:

$$\frac{9}{14}\times\left(-\log\frac{9}{14}\right)+\left(\frac{5}{14}\right)\times\left(-\log\frac{5}{14}\right)=0.940\text{ bits.}$$

Considerant *Outlook*:

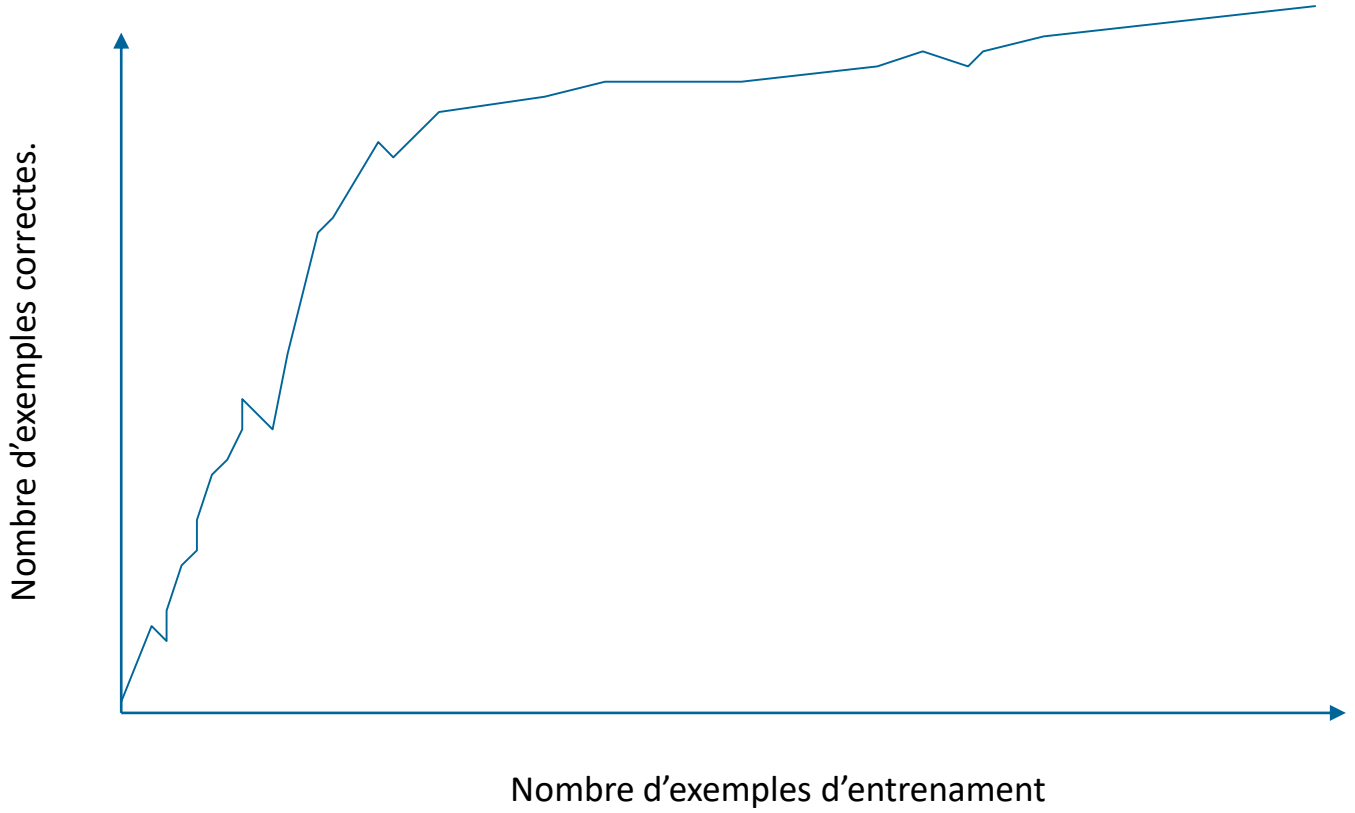
$$\text{Information further required} =$$

$$\left(\frac{5}{14}\right)\times 0.971+\left(\frac{4}{14}\right)\times 0+\left(\frac{5}{14}\right)\times 0.971=0.693 bits.$$



Avaluació d'un classificador.

Ha funcionat bé si:



Avaluació d'un classificador.

Metodologia:

1. Recollirem un nombre gran d'exemples.

2. Els dividirem en dos conjunts disjunts: aprenentatge i test.

3. Generem la hipòtesi amb el conjunt d'aprenentatge.

4. Mesurem el percentatge d'exemples en el conjunt de test que estan correctament classificats per la hipòtesi.

5. Repetim els passos 1-4 amb diferents mides del conjunt d'aprenentatge, escollint aleatòriament els seus elements.

Avaluació d'un classificador: Avaluació.

Mètriques

Com avaluar la "qualitat" d'un model de clasificació

Mètodes

Com estimar, de forma fiable, la qualitat d'un model.

Comparació

Com comparar el rendiment relatiu de dos models de classificació alternatius

Avaluació d'un classificador: Avaluació - Mètriques.

Matriu de confusió
(confusion matrix)

		Predicció	
		C_P	**C_N**
Classe real	**C_P**	**TP**: True positive	**FN**: False negative
	C_N	**FP**: False positive	**TN**: True negative

Precisió del classificador

accuracy = (TP+TN)/(TP+TN+FP+FN)

Avaluació d'un classificador: Avaluació - Mètriques.

Matriu de confusió per multiclasse (confusion matrix)

Ex: entrenament de gossos

	play	fight	alone	stranger	walk	ball
play	6%	13%	4%	25%	40%	13%
fight	1%	74%	0%	14%	6%	6%
alone	4%	7%	15%	43%	16%	16%
stranger	1%	13%	4%	63%	6%	13%
walk	11%	11%	5%	30%	30%	12%
ball	2%	7%	5%	38%	12%	36%

Avaluació d'un classificador: Avaluació - Mètriques.

Limitacions de la precisió ("accuracy") :

Suposem un problema amb 2 classes:

9990 exemples de la classe 1

10 exemples de la classe 2

Si el model de clasificació sempre diu que els exemples són de la classe 1, l'accuracy és

9990/10000 = 99.9%

Porta a engany, ja que mai detectarem cap exemple de la classe 2

Avaluació d'un classificador: Avaluació - Mètriques.

Altres mesures

		Predicció	
		C_P	**C_N**
Classe real	**C_P**	**TP**: True positive	**FN**: False negative
	C_N	**FP**: False positive	**TN**: True negative

accuracy = (TP+TN)/(TP+TN+FP+FN)

precision = TP/(TP+FP)

True positive recognition rate

recall = sensitivity = TP/P = TP/(TP+FN)

True negative recognition rate

specificity = TN/N = TN/(TN+FP)

Avaluació d'un classificador: Avaluació - Mètriques.

Més mesures

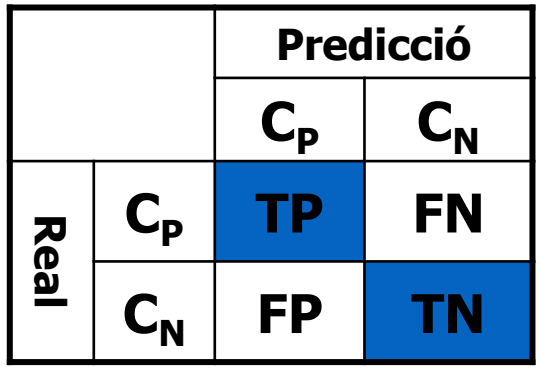
		Predicció	
		C_P	**C_N**
Classe real	**C_P**	**TP**: True positive	**FN**: False negative
	C_N	**FP**: False positive	**TN**: True negative

F-measure

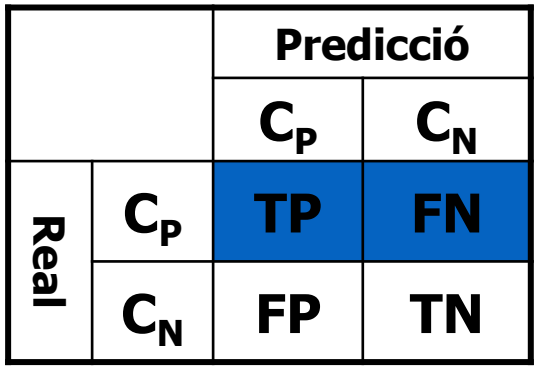
F = 2*precision*recall / (precision+recall)

F= 2TP / (2TP+FP+FN)

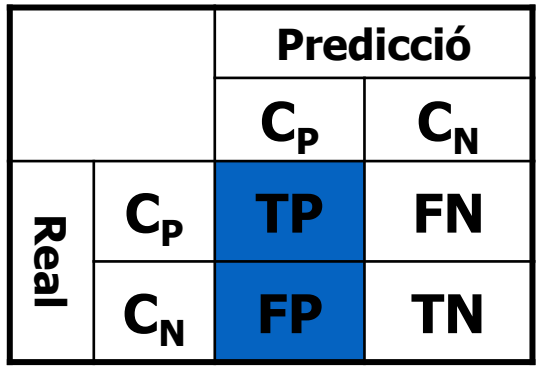
Avaluació d'un classificador: Avaluació - Mètriques.



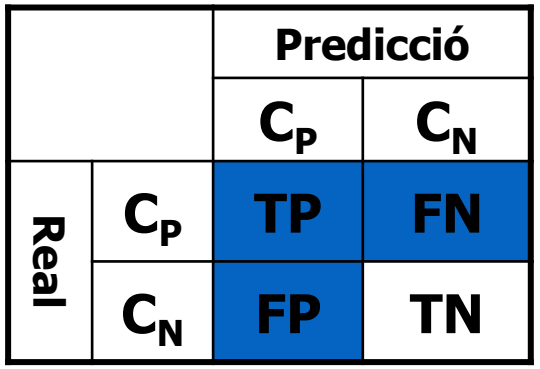
Accuracy



Recall



Precision



F-measure

Avaluació d'un classificador: Avaluació - Mètodes.

Per a avaluar la precisió d'un model de classificació mai hem d'usar el conjunt d'entrenament (ens donaria "l'**error de restitució**" del classificador), sino un conjunt de proba independient:

Com seleccionar els ?

Training set

Test Set

Mètode **Holdout:**

Per exemple, podríem reservar 2/3 dels exemples disponibles per a construir el clasificador i el 1/3 restant l'usariem de conjunt de proba per a estimar la precisió del classificador (extreure la matriu de confusió).

Avaluació d'un classificador: Avaluació - Mètodes.

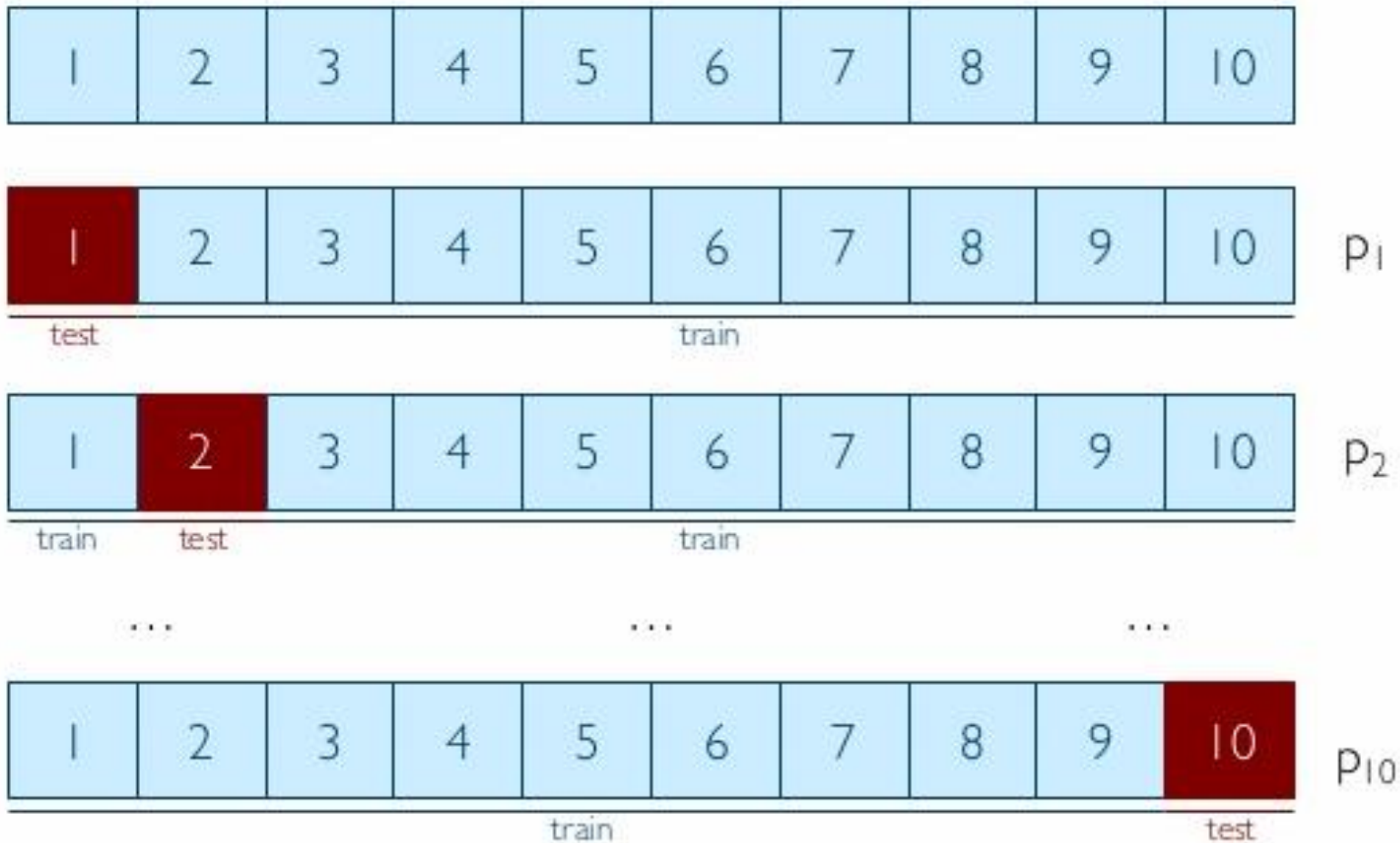
Mètode de **Validació creuada** (k-CV: k-fold Cross-Validation)

- Es divideix aleatoriament (per a no esbiaixar el resultat) el conjunt de dades en k subconjunts d'intersecció buida (més o menys de la mateixa mida).

- En la iteració i, s'usa el subconjunt i com a conjunt de proba i els k-1 restants com conjunt d'entrenamient.

- Com a mesura d'avaluació del mètodo de classificació s'agafa la mitjana aritmètica de les k iteracions realitzadas.

Ten-fold Crossvalidation



El rendiment final es calcula com el promig de les p_i

Avaluació d'un classificador: Avaluació - Mètodes.

Mètode de **Validació creuada:** Variants

- **"Leave one out"**: Es realitza una validació creuada amb k particions del conjunt de dades, on k coincideix amb el número d'exemples disponibles.

- **Validació creuada estratificada**: Les particions es realitzen intentant mantenir en totes elles la mateixa proporció de classes que apareixen en el conjunt de dades complet.

Avaluació d'un classificador: Avaluació - Mètodes.

Mètode de **Bootstrapping**

Mostreig uniforme amb reemplaçament dels exemples disponibles (un cop s'escull un exemple, es torna a deixar en el conjunt d'entrenament i pot ser que es torni a escollir).

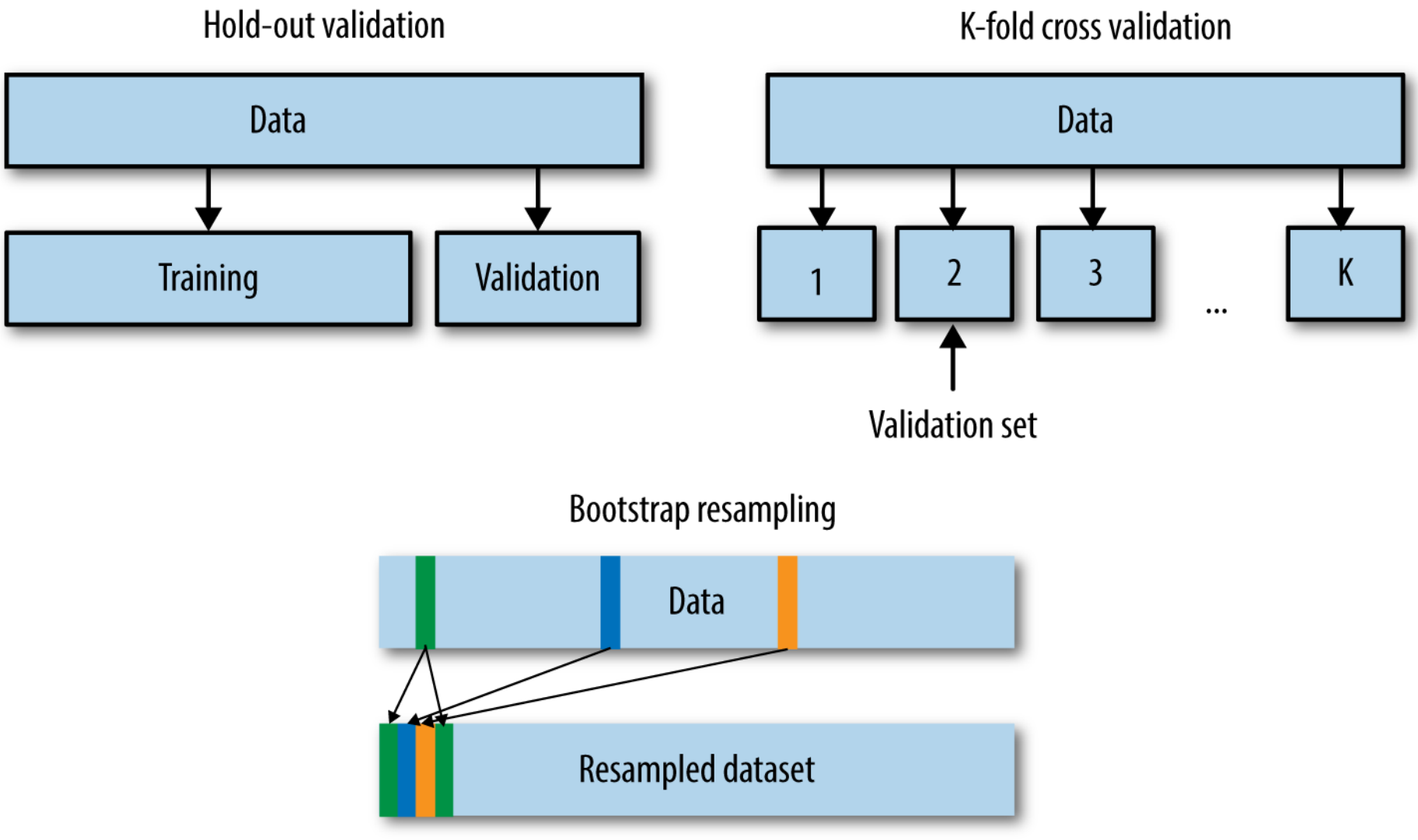
0.632 bootstrap: Donat un conjunt de x dades, es prenen x mostres. Les dades que no es trien formaran part del conjunt de proba.

Al voltant del 63.2% de les mostres estaran en el "bootstrap" (el conjunt d'entrenament) i el 36.8% en el conjunt de proba ja que

$$(1\text{-}1/x)^x \approx e^{-1} = 0.368$$

Si repetim el proces k vegades, tindrem:

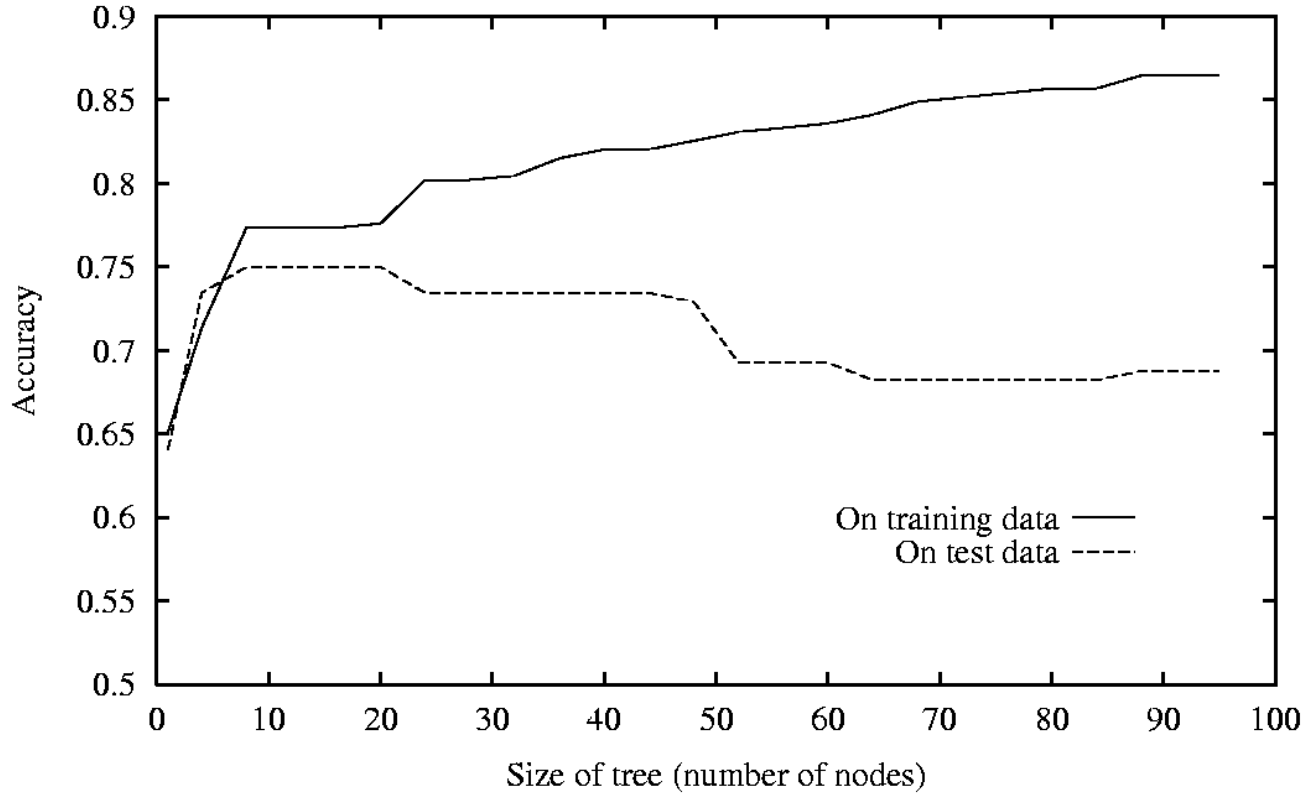
$$acc(M) = \sum_{i=1}^{k} (0.632 \times acc(M_i)_{test\_set} + 0.368 \times acc(M_i)_{train\_set})$$


Hold-out validation
Data
Training
Validation
K-fold cross validation
Data
1
2
3
...
K
Validation set
Bootstrap resampling
Data
Resampled dataset

Avaluació d'un classificador

Overfitting: una hipòtesi *h* sobrerepresenta les dades d'aprenentatge si:

- existeix alguna hipòtesi alternativa *h'*
- *h* té menor error que *h'* sobre el conjunt d'aprenentatge
- però *h'* té un error menor que *h* sobre totes les instàncies possibles.



Inducció d'Arbres de Decisió.

Rendiment baix:

Per què ?

1. Soroll a les dades (exemples mal etiquetats)
2. Pocs exemples en relació a la complexitat de la funció

Com solucionar-ho ?

1. **Aturant-se** abans de classificar perfectament les dades.
2. Post-processar l'arbre: **pruning**

Inducció d'Arbres de Decisió.

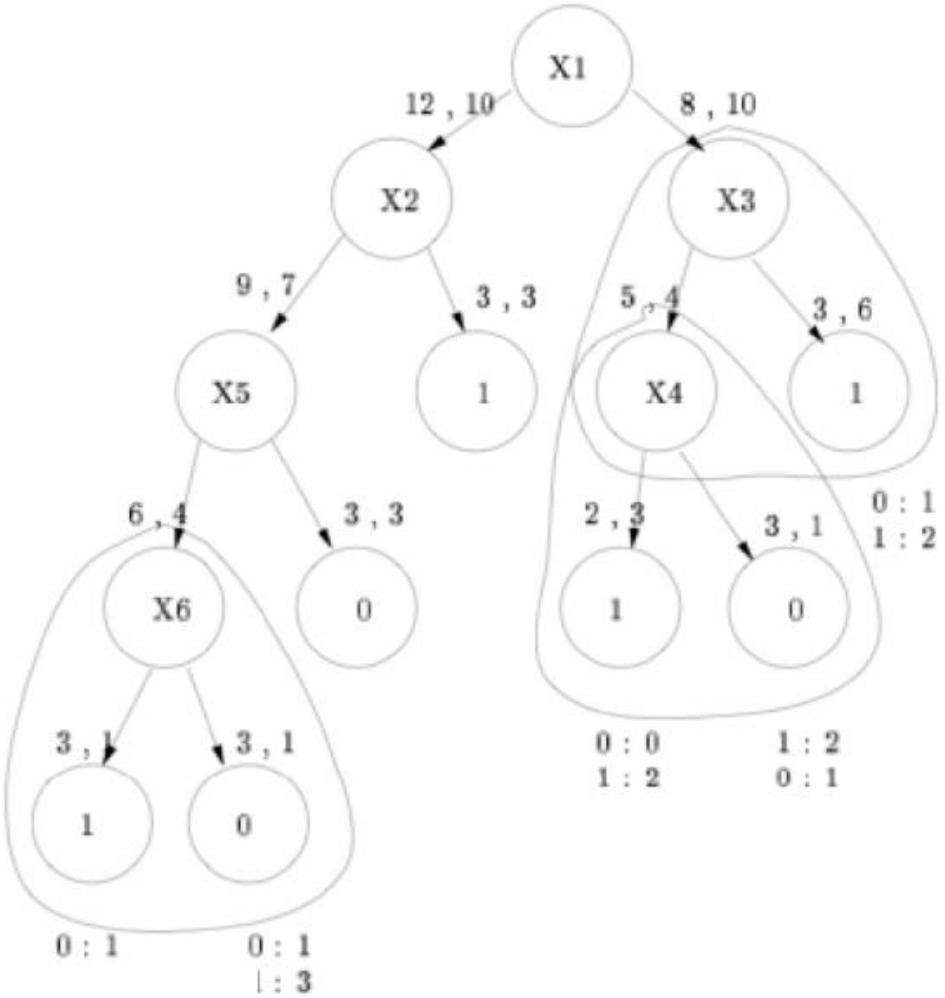
Pruning:

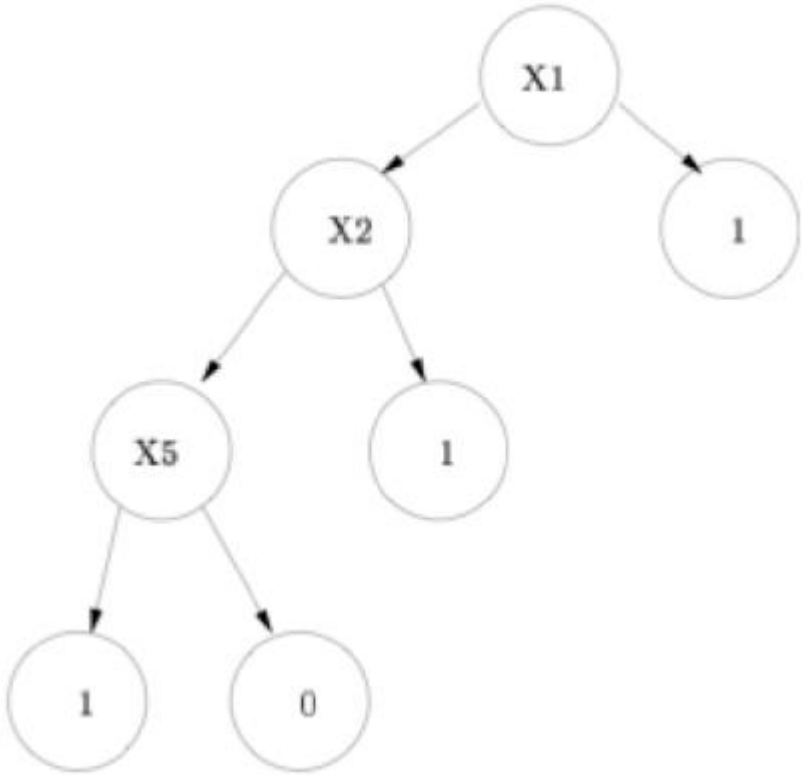
1. Tots els nodes de decisió són candidats

2. Un node és eliminat només si l'arbre resultant no és pitjor que l'anterior sobre el conjunt de **validació**

3. Eliminar el subarbre que penja del node, convertir-lo en fulla, i assignant el valor majoritari dels exemples que hi pengen com a valor de classificació.

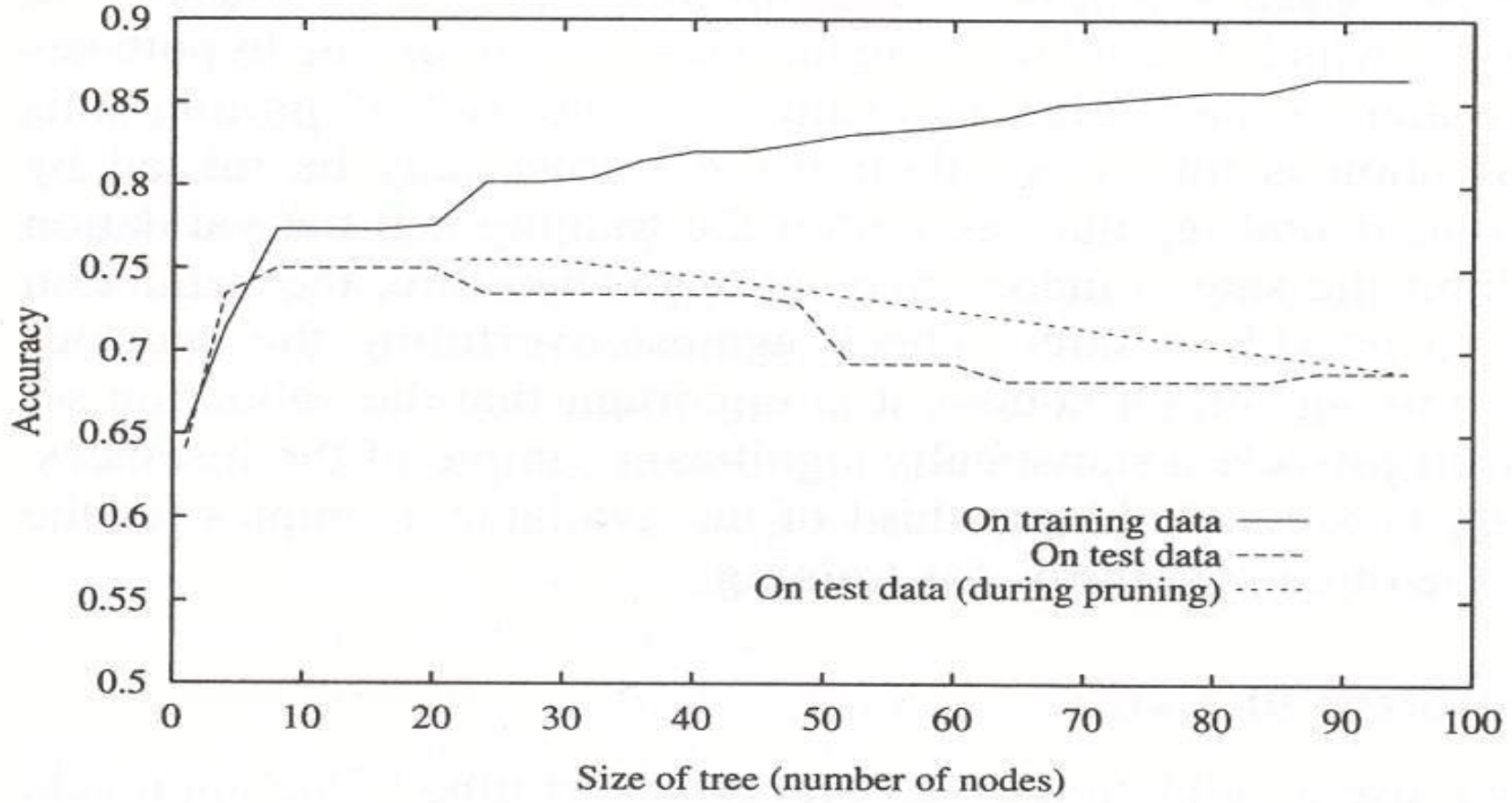
Pruning d’Arbres de Decisió.





Pruning d'Arbres de Decisió.

Resultat del pruning:



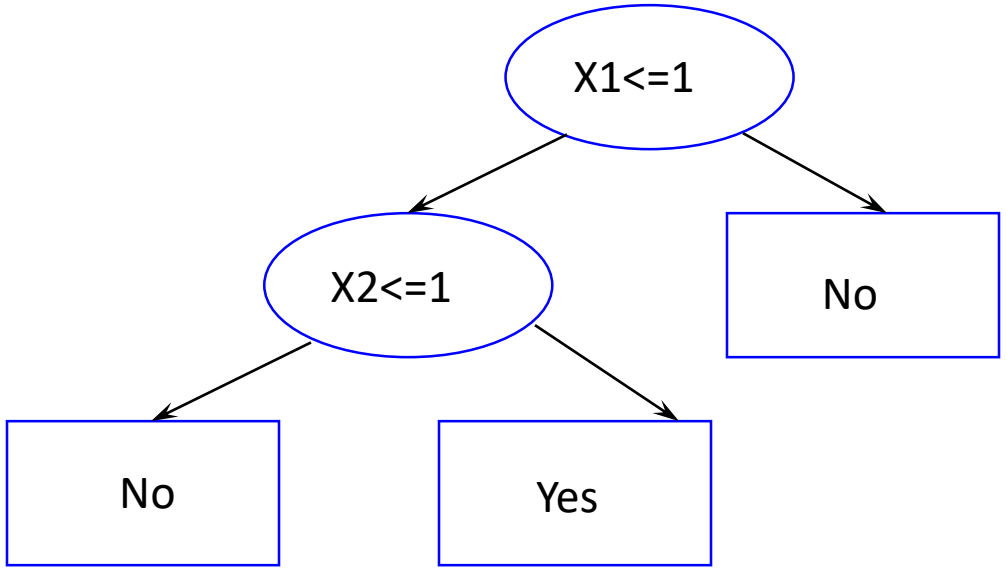
Pruning d'Arbres de Decisió.

Reduced-Error Pruning (Poda per reducció d'error)

- Classificar els exemples en el conjunt de validació – alguns podran ser erronis
- Per a cada node:
  - Sumar els errors en tot el subarbre
  - Calcular l'error amb els mateixos exemples si es converteix el subarbre en una fulla amb l'etiqueta majoritaria
- Podar el node amb la major reducció d'error
- Repetir fins que l'error no es redueix

Test set:

X1	X2	Class
1	1	Yes
1	2	Yes
1	2	Yes
1	2	Yes
1	1	Yes
1	2	No
2	1	No
2	1	No
2	2	No
2	2	No



Només arrel: 10% error

Arbre sencer: 30% error

Pruning d'Arbres de Decisió.

Pruning Pesimistic

- Evita usar el conjunt d'avaluació (podrem entrenar en més exemples)
- Estima de forma conservadora l'error real a cada node, basant-se en els exemples d'entrenament
- "Correció de continuitat" a la taxa d'error a cada node: afegim 1/2N als errors observats, on N es el número de fulles en el sub-arbre
- Podem el node a menys que l'error estimat del sub-arbre és mes petit de 1 error standard per sota del error estimat per al podat: $r'_{subtree} < r'_{pruned} + SE$

$$\varepsilon'(T,S) = \varepsilon(T,S) + \frac{|fulles(T)|}{2|S|} \qquad \varepsilon'(poda(T,t),S) \leq \varepsilon'(T,S) + \sqrt{\frac{\varepsilon'(T,S)(1-\varepsilon'(T,S))}{|S|}}$$

Pruning d'Arbres de Decisió.

"**Cost-complexity Pruning**"

- "cost-complexity" – és una mesura de l'error promig reduit per fulla
- Calcula la mesura α per a cada node si es substitueix per una fulla
- Compara els errors en les fulles, i podar el de mínima α

$$\alpha = \frac{\varepsilon(poda(T,t),S) - \varepsilon(T,S)}{|fulles(T)| - |fulles(poda(T,t))|}$$

- S'itera fins que no hi ha cap node amb α < llindar

Inducció d'Arbres de Decisió.

Atributs continus: crear atributs discrets que particionin els valors

$$A \in \{0,1\} \dashrightarrow A_c = \textit{true} \text{ si } A \geq c\text{; sino } \textit{false}$$

La qüestió és com triar *c*:

Temperature:	40	48	60	72	80	90
PlayTennis:	No	No	Yes	Yes	Yes	No

Cal escollir la *c* que doni el guany d'informació més gran!

Tenim dos candidats naturals a *c*: (48+60)/2 i (80+90)/2. El millor és

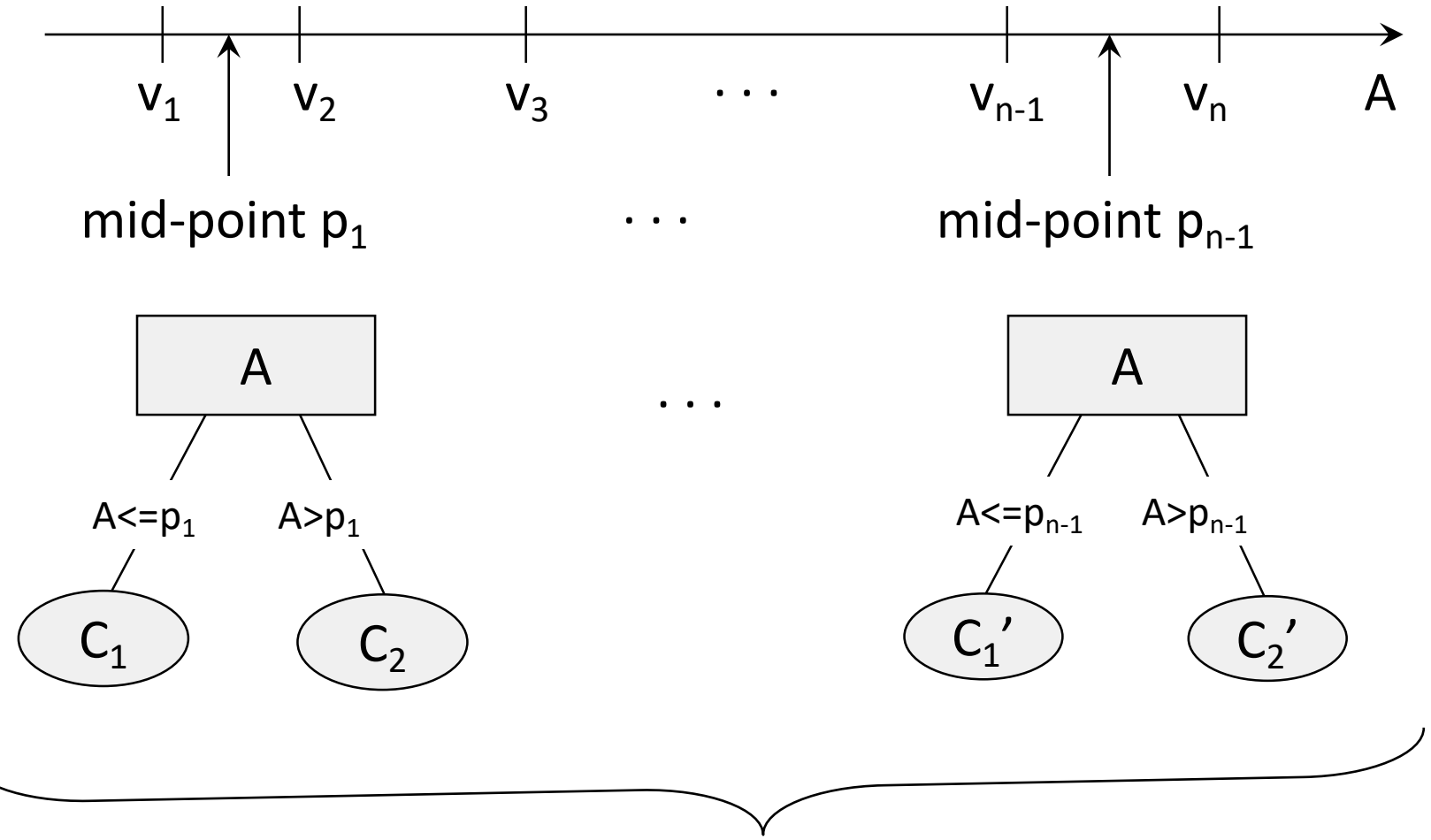
Temperature > 54.

Inducció d'Arbres de Decisió.

Atributs continus: crear atributs discrets que particionin els valors

1. Intervals pre-especificats.
2. Particions binàries (2-way partition):
   - Ordenar els valors del atribut en ordre ascendent v_1, v_2, ...,v_n (assumim que tenim n valurs per l'atribut continu A).
   - Fer, exhaustivament, una partició binària a cada punt mig (i.e., $(v_i + v_{i+1})/2$).
   - Per a cada punt mig p, calcular la entropia (o altre indicador) del arbre parcial si A es particionat al punt mig p (i.e., A <= p and A > p).
   - Seleccionar el millor punt de partició (i.e., que resulta en el màxim guany) per A.

Inducció d'Arbres de Decisió.



Seleccionar el millor punt de partició per a A que reulta en el màxim de guany en 'information gain' o 'gain-ratio'

Inducció d'Arbres de Decisió.

Valors desconeguts: En aplicacions reals sovint trobem casos on alguns valors no són coneguts.

Els valors desconeguts causen 3 problemes.

1. La selecció d'un test per a particionar el conjunt d'aprenentatge pot requerir comparar test basats en atributs amb diferent número de casos.
2. Un cop s'ha seleccionat un test (basat en l'atribut A, p exemple), els casos d'aprenentatge amb valor desconegut de A no es poden asociar al test de resultat. Com ho hem de tractar en la divisió del conjunt d'aprenentatge en subconjunts?
3. Quan l'arbre de decisió s'usa per a clasificar un cas 'no vist', com ho fem quan trobem un test sobre un atribut desconegut?

Inducció d'Arbres de Decisió.

Valors desconeguts:

Solucions:

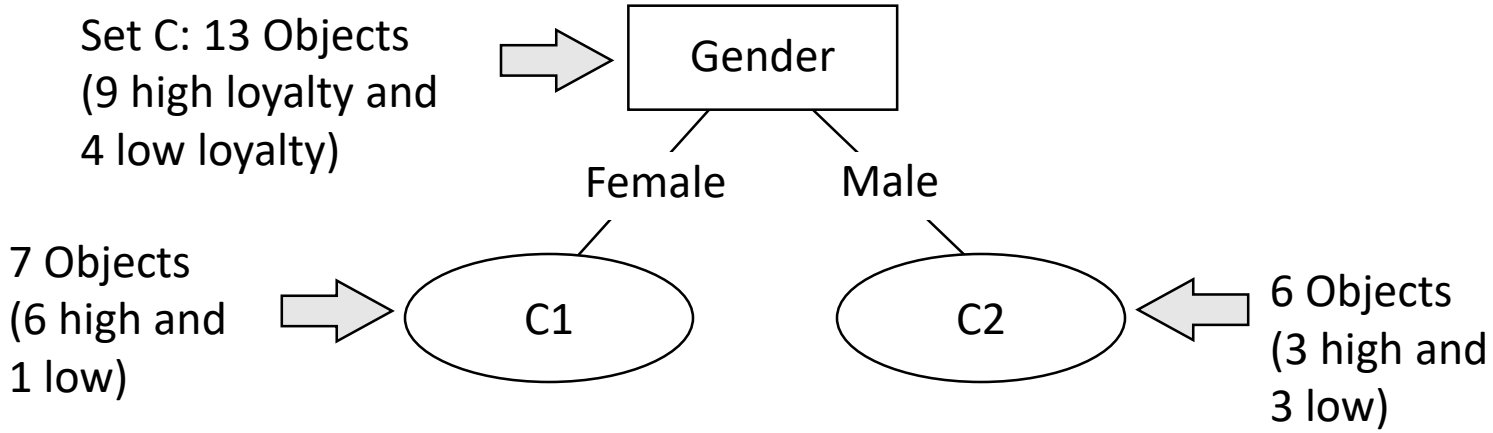
1. "Replenar" els valors perduts de A amb el valor conegut més comú abans de calcular el guany d'informació de A.

2. Ignorar els casos en el conjunt d'aprenentage amb valors desconeguts de A.
    1. Tractar la variable amb una classe més (p.ex: Si/No/?)
    2. Pesar la mesura de divisió per la proporció de casos no desconeguts.

Exemple

No.	Attributes				Class
	Location	Age	Marriage status	Gender	(Loyalty)
1	Urban	Below 21	Married	?	Low
2	Urban	Below 21	Married	Male	Low
3	Suburban	Below 21	Married	Female	High
4	Rural	Between 21 and 30	Married	Female	High
5	Rural	Above 30	Single	Female	High
6	Rural	Above 30	Single	Male	Low
7	Suburban	Above 30	Single	Male	High
8	Urban	Between 21 and 30	Married	Female	Low
9	Urban	Above 30	Single	Female	High
10	Rural	Between 21 and 30	Single	Female	High
11	Urban	Between 21 and 30	Single	Male	High
12	Suburban	Between 21 and 30	Married	Male	High
13	Suburban	Below 21	Single	Female	High
14	Rural	Between 21 and 30	Married	Male	Low

Quan avaluem el genere, tenim 13 casos amb Informació de genere per a calcular information gain o gain-ratio.



$$E(C) = -\frac{9}{13}\log_2\frac{9}{13} - \frac{4}{13}\log_2\frac{4}{13} = 0.8905$$

$$E(Gender) = \frac{7}{13}(-\frac{6}{7}\log_2\frac{6}{7} - \frac{1}{7}\log_2\frac{1}{7}) +$$

$$\frac{6}{13}(-\frac{3}{6}\log_2\frac{3}{6} - \frac{3}{6}\log_2\frac{3}{6}) = 0.7801$$

Guany d'informació de 'Genere' per a 13 casos coneguts de Genere:

$$G(Gender) = E(C) - E(Gender) = 0.1104$$

Ajustem el guany d'informació de Genere per reflectir la informació Desconeguda:

$$G'(Gender) = \frac{13}{14} G(Gender) = 0.1025$$

Ajustem el valor d'informació de 'Genere' per a reflectir 'genere desconegut' :

$$IV(Gender) = -\frac{7}{14}\log_2\frac{7}{14} \quad \text{(per a dona)}$$
$$-\frac{6}{14}\log_2\frac{6}{14} \quad \text{(per a home)}$$
$$-\frac{1}{14}\log_2\frac{1}{14} \quad \text{(per a ?)}$$
$$= 1.2958$$

Gain ratio de genere qaun consider un cas del que desconeixem el seu genere:

$$GR(Gender) = \frac{G'(Gender)}{IV(Gender)} = \frac{0.1025}{1.2958} = 0.0791$$

Inducció d'Arbres de Decisió.

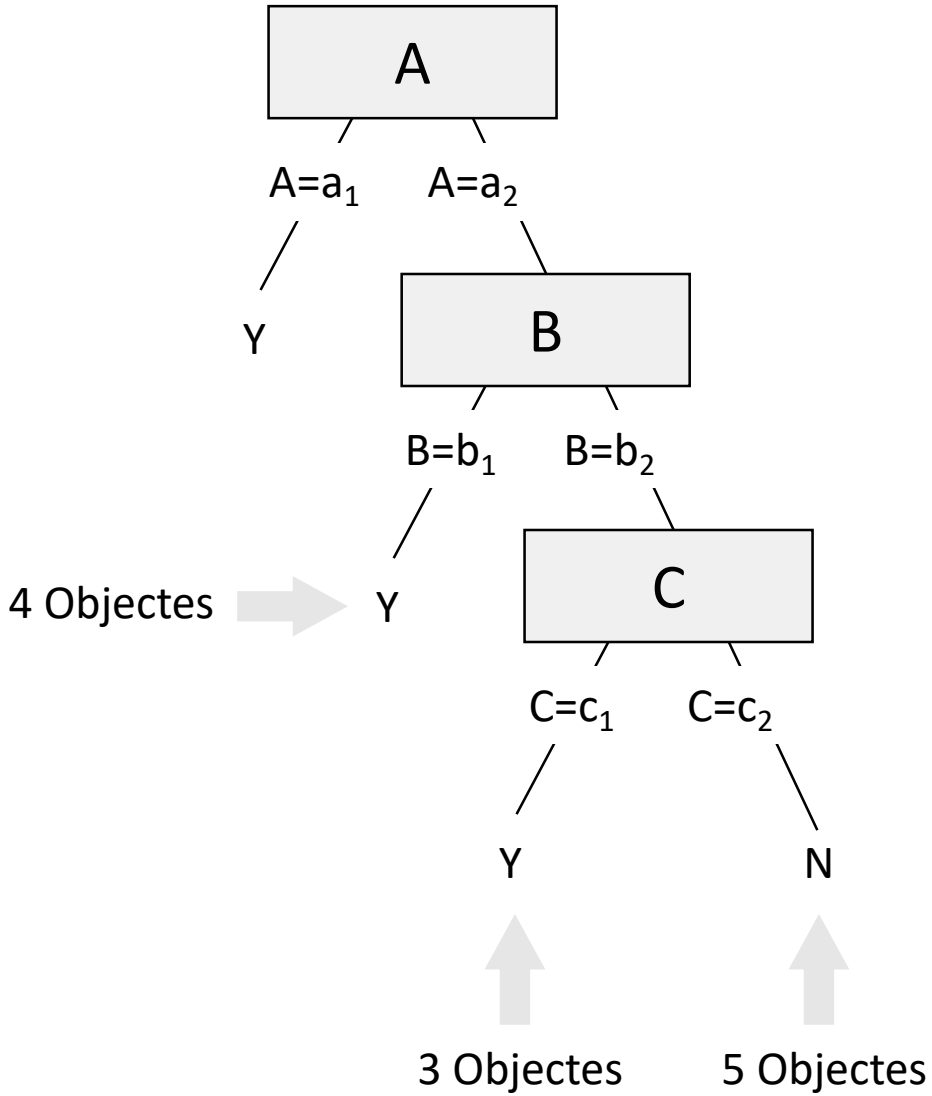
Classificació de Valors desconeguts:

Quan classifiquem un nou cas amb valor de A desconegut:

Solucions:

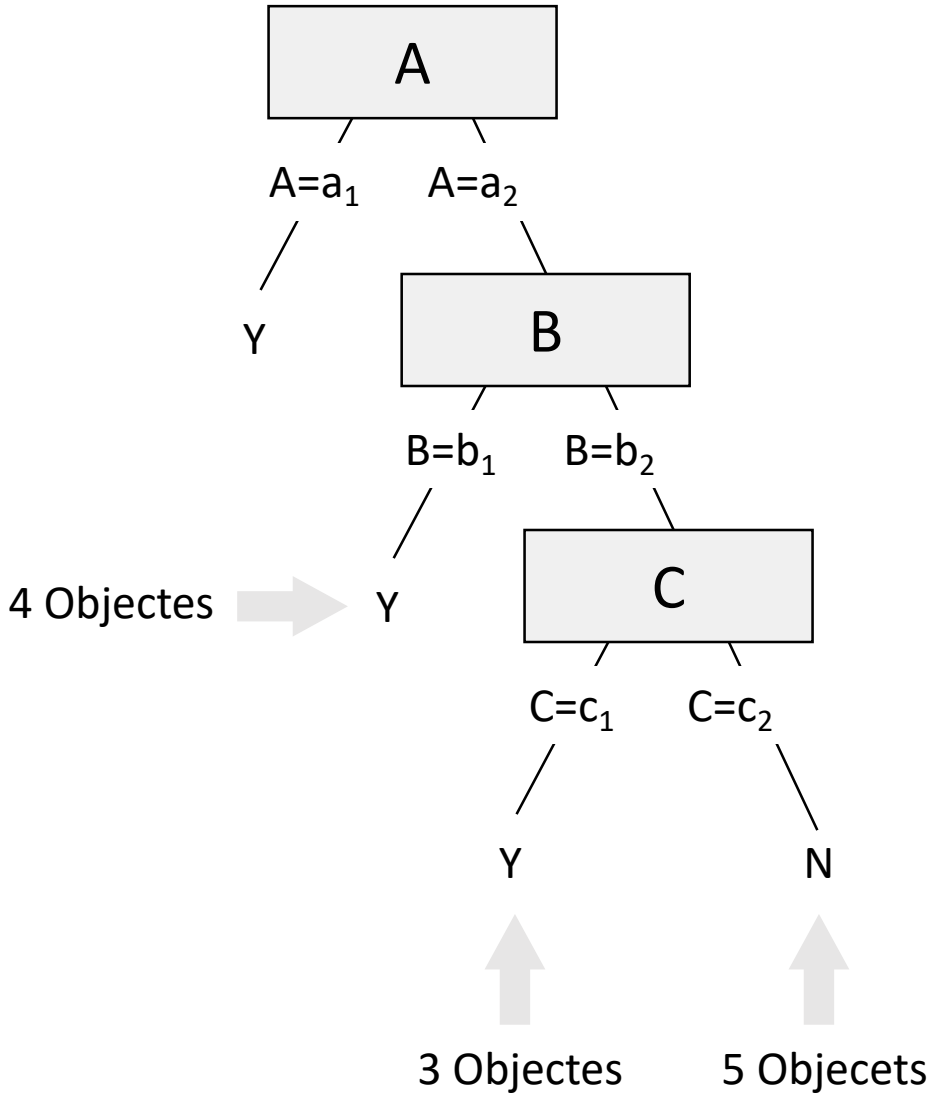
1. Si hi ha una branca per a 'valor desconegut' l'agafem..
2. Ho parem en aquest punt i assignem el cas a la classe més probable.
3. Explorem totes les branques I combinem els resultats pe a reflectir les probabilitats relatives dels diferents resultats. Asignem la classe amb més alta probabilitat de ser predita (adoptat per C4.5).

Exemple de l'aproximació per probabilitats



Quina és la classe assignada al cas ($A=a_2$, $B=?$, $C=c_2$)?

Exemple de l'aproximació per probabilitats

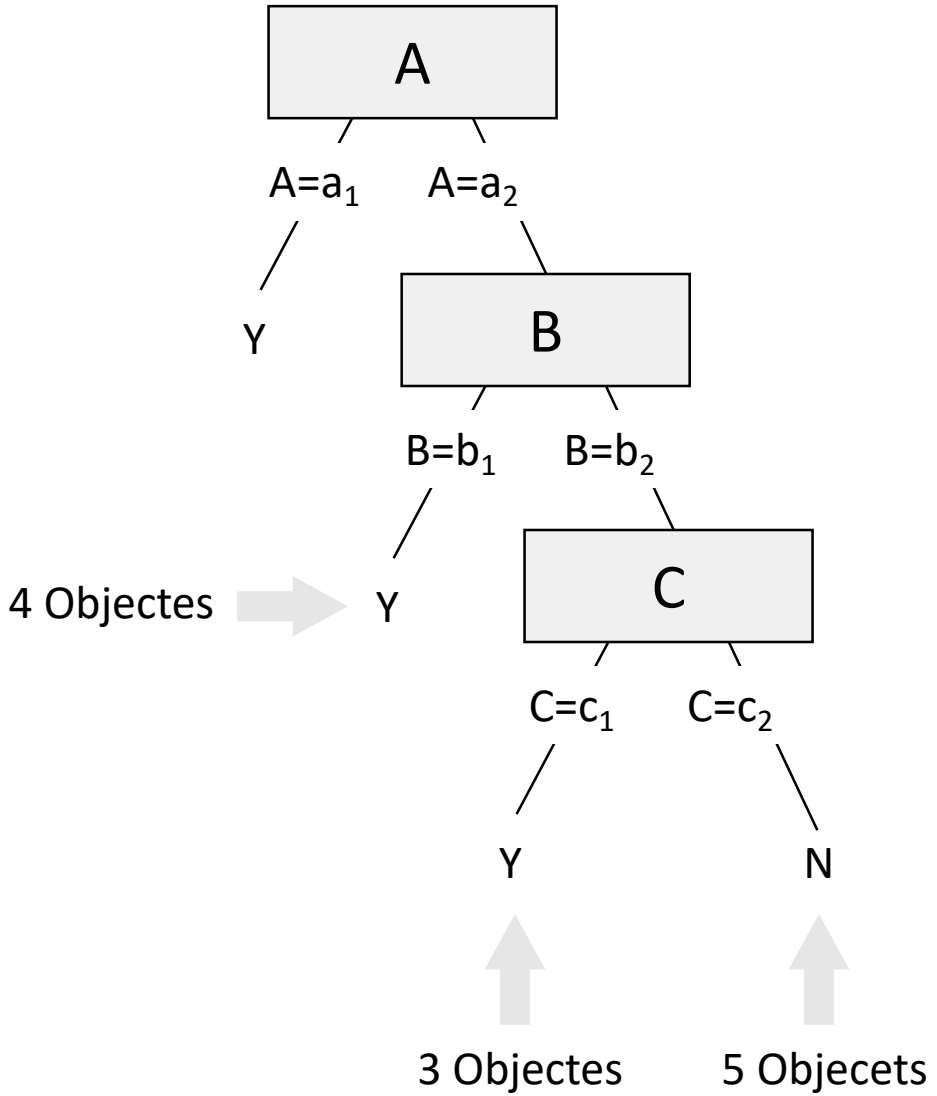


Quina és la classe assignada al cas ($A=a_2$, B=?, $C=c_2$)?

prob(Y)=4/12
prob(N)=8/12
Per tant, la classe és No.

Quina és la classe assignada al cas($A=a_2$, B=?, C=?)?

Exemple de l'aproximació per probabilitats



Quina és la classe assignada
al cas ($A=a_2$, B=?, $C=c_2$)?

prob(Y)=4/12
prob(N)=8/12
Per tant, la classe és No.

Quina és la classe assignada
al cas($A=a_2$, B=?, C=?)?

prob(Y)=4/12+(8/12)*(3/8)=7/12
prob(N)=(8/12)*(5/8)=5/12
Per tant, la classe és Yes.