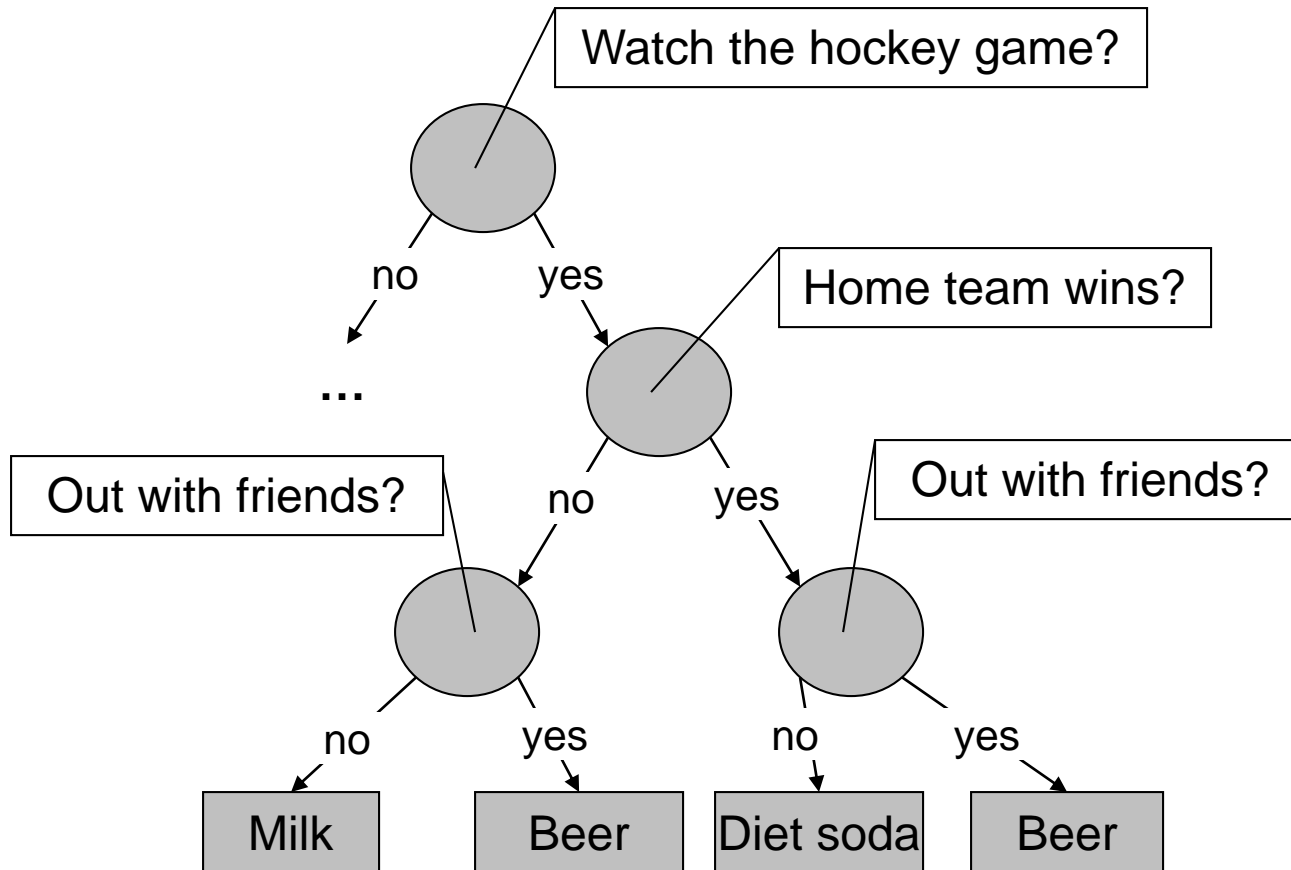


# Classification rules

Lecture 01.03

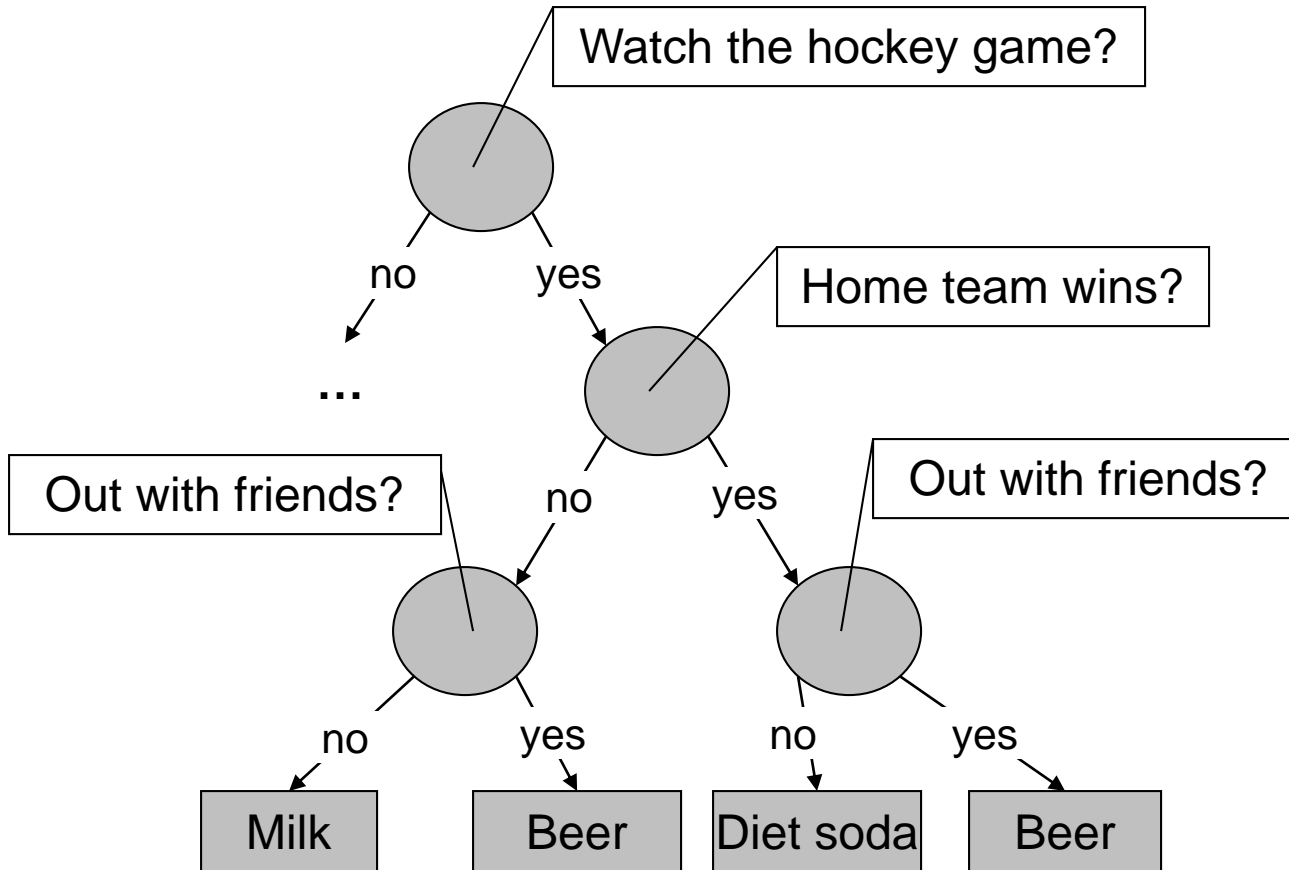
# Decision tree – as a collection of rules



- If** watch the game **and** home team wins **and** sitting at home **then** diet soda
- If** watch the game **and** home team wins **and** out with friends **then** beer
- If** watch the game **and** home team loses **and** sitting at home **then** milk
- If** watch the game **and** home team loses **and** out with friends **then** beer

# Why rules -1

## Rules can be simpler than the tree



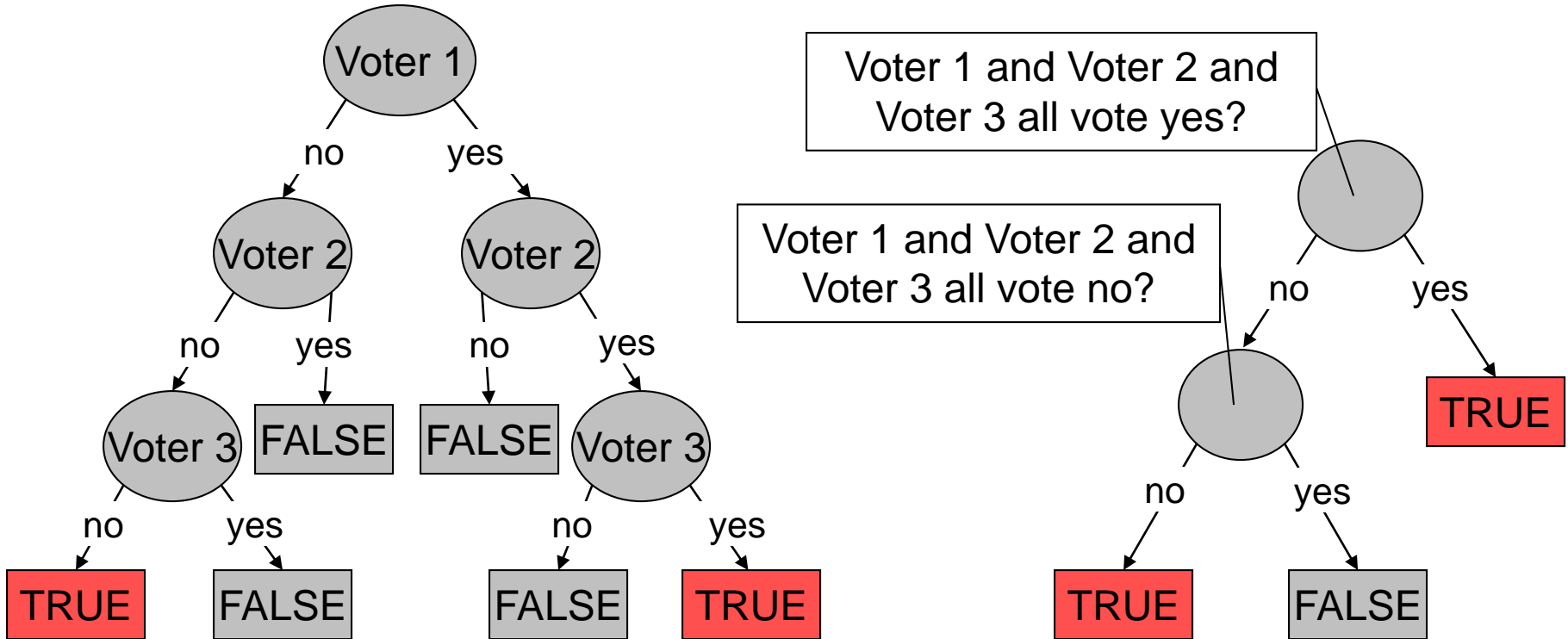
**If watch the game and home team wins and sitting at home then diet soda**  
**If watch the game and home team loses and sitting at home then milk**



**If watch the game and out with friends then beer**

# Why rules – 2?

## Rules allow Boolean conjunctions



Class label: Unanimous voting

# Classification rules – start from the class label

- Decision tree starts with attribute values
- Start with the class label:
  - (*Condition*) → *class label*
  - *LHS*: rule *antecedent* or condition
  - *RHS*: rule *consequent*
  - Examples of classification rules:
    - (Blood Type=Warm)  $\wedge$  (Lay Eggs=Yes) → Birds
    - (Taxable Income < 50K)  $\wedge$  (Refund=Yes) → Evade=No

# Example: animal classification

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
human	warm	yes	no	no	mammals
python	cold	no	no	no	reptiles
salmon	cold	no	no	yes	fishes
whale	warm	yes	no	yes	mammals
frog	cold	no	no	sometimes	amphibians
komodo	cold	no	no	no	reptiles
bat	warm	yes	yes	no	mammals
pigeon	warm	no	yes	no	birds
cat	warm	yes	no	no	mammals
leopard shark	cold	yes	no	yes	fishes
turtle	cold	no	no	sometimes	reptiles
penguin	warm	no	no	sometimes	birds
porcupine	warm	yes	no	no	mammals
eel	cold	no	no	yes	fishes
salamander	cold	no	no	sometimes	amphibians
gila monster	cold	no	no	no	reptiles
platypus	warm	no	no	no	mammals
owl	warm	no	yes	no	birds
dolphin	warm	yes	no	yes	mammals
eagle	warm	no	yes	no	birds

# Animal classification rules

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
human	warm	yes	no	no	mammals
python	cold	no	no	no	reptiles
salmon	cold	no	no	yes	fishes
whale	warm	yes	no	yes	mammals
frog	cold	no	no	sometimes	amphibians
komodo	cold	no	no	no	reptiles
bat	warm	yes	yes	no	mammals
pigeon	warm	no	yes	no	birds
cat	warm	yes	no	no	mammals
leopard shark	cold	yes	no	yes	fishes
turtle	cold	no	no	sometimes	reptiles
penguin	warm	no	no	sometimes	birds
porcupine	warm	yes	no	no	mammals
eel	cold	no	no	yes	fishes
salamander	cold	no	no	sometimes	amphibians
gila monster	cold	no	no	no	reptiles
platypus	warm	no	no	no	mammals
owl	warm	no	yes	no	birds
dolphin	warm	yes	no	yes	mammals
eagle	warm	no	yes	no	birds

R1: (Give Birth = no)  $\wedge$  (Can Fly = yes)  $\rightarrow$  Birds

R2: (Give Birth = no)  $\wedge$  (Live in Water = yes)  $\rightarrow$  Fishes

R3: (Give Birth = yes)  $\wedge$  (Blood Type = warm)  $\rightarrow$  Mammals

R4: (Give Birth = no)  $\wedge$  (Can Fly = no)  $\rightarrow$  Reptiles

R5: (Live in Water = sometimes)  $\rightarrow$  Amphibians

# Rule coverage

- A rule  $r$  **covers** an instance  $x$  if the attributes of the instance satisfy **the condition** of the rule

R1: (Give Birth = no)  $\wedge$  (Can Fly = yes)  $\rightarrow$  Birds

R2: (Give Birth = no)  $\wedge$  (Live in Water = yes)  $\rightarrow$  Fishes

R3: (Give Birth = yes)  $\wedge$  (Blood Type = warm)  $\rightarrow$  Mammals

R4: (Give Birth = no)  $\wedge$  (Can Fly = no)  $\rightarrow$  Reptiles

R5: (Live in Water = sometimes)  $\rightarrow$  Amphibians

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
hawk	warm	no	yes	no	?
grizzly bear	warm	yes	no	no	?

The rule R1 covers a hawk  $\Rightarrow$  Bird

The rule R3 covers the grizzly bear  $\Rightarrow$  Mammal



# Rule quality:

## Coverage and accuracy

- **Coverage** of a rule:
  - Fraction of records that satisfy the **condition** of a rule (over all records)
- **Accuracy** of a rule:
  - Fraction of records that satisfy **both the condition and the class** (over those that satisfy the condition)

Tid	Refund	Marital Status	Taxable Income	Class
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

**(Status=Single) → No**

**Coverage = 40%, Accuracy = 50%**

# Ordered Rule Set

- Rules are ranked ordered according to their quality (e.g. accuracy and coverage)
- An ordered rule set is known as a **decision list**
- When a test record is presented to the classifier
  - It is assigned to the class label of the highest ranked rule it has triggered
  - If none of the rules fired, it is assigned to the default class

R1: (Give Birth = no)  $\wedge$  (Can Fly = yes)  $\rightarrow$  Birds

R2: (Give Birth = no)  $\wedge$  (Live in Water = yes)  $\rightarrow$  Fishes

R3: (Give Birth = yes)  $\wedge$  (Blood Type = warm)  $\rightarrow$  Mammals

R4: (Give Birth = no)  $\wedge$  (Can Fly = no)  $\rightarrow$  Reptiles

R5: (Live in Water = sometimes)  $\rightarrow$  Amphibians

Stop here



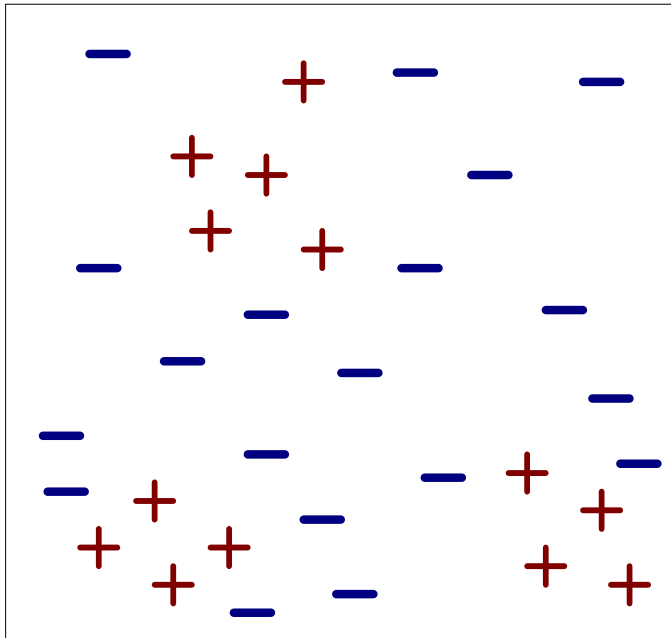
Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
turtle	cold	no	no	sometimes	?

# Algorithms for generating the rules

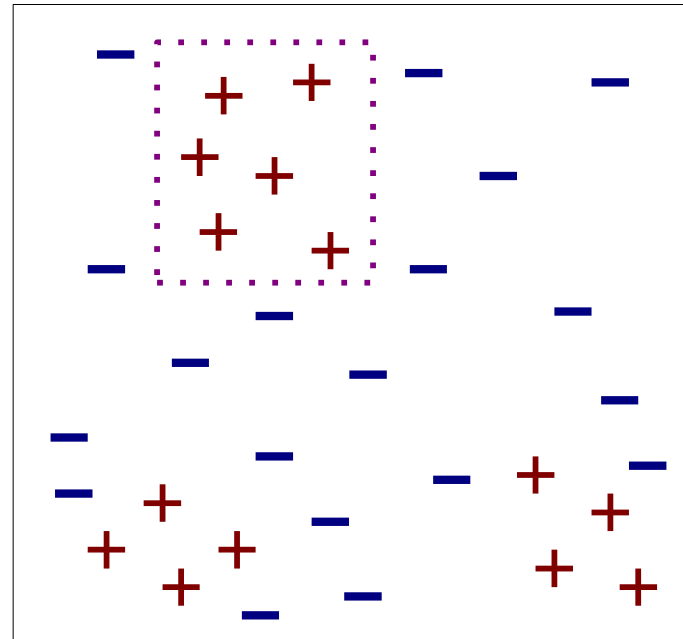
- From decision trees (*divide-and-conquer*)
- **Covering approach** (*separate and conquer*):
  - At each step – take a class and find a condition which covers instances in this class
  - The goal - to cover all instances

# Building Classification Rules: Sequential Covering

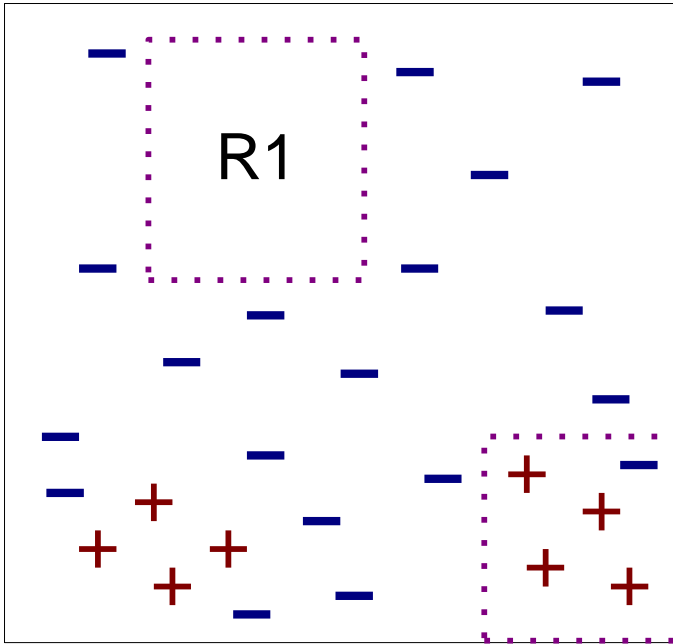
1. Start from an empty rule
2. Grow a rule using some **Learn-One-Rule** function
3. Remove training records **covered** by the rule
4. Repeat Step (2) and (3) until stopping criterion is met



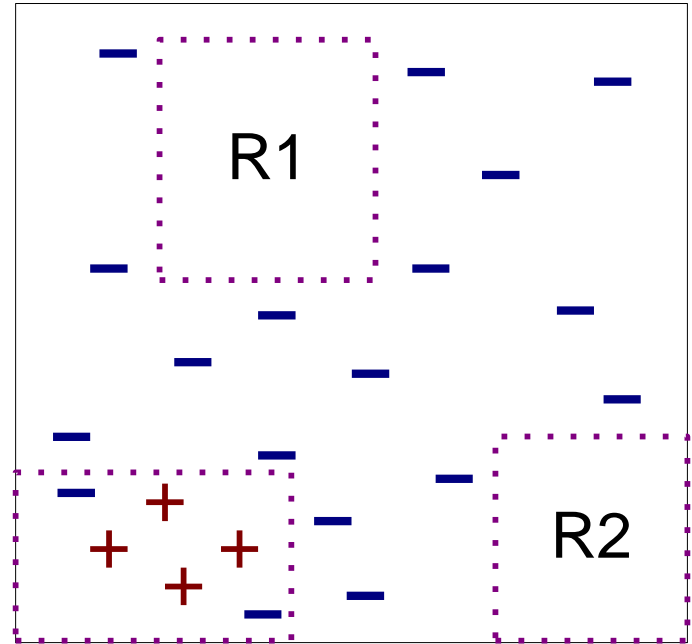
(i) Original Data



(ii) Step 1



(iii) Step 2



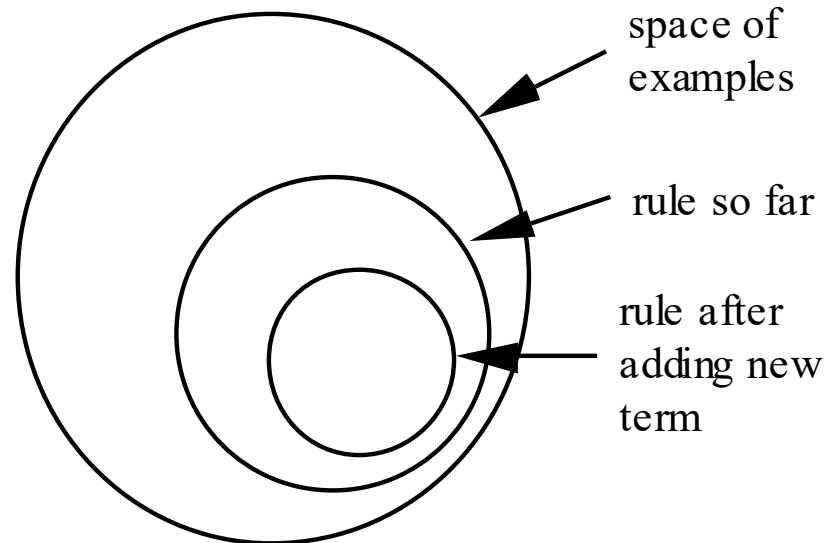
(iv) Step 3

- This approach is called a **covering** approach because at each stage a rule is identified that covers some of the instances

# A simple covering algorithm idea

- Generate a rule by adding tests that maximize rule's accuracy
  - Similar to situation in decision trees: problem of selecting an attribute to split on
  - But: decision tree inducer maximizes **overall** purity, for a rule it is important to have purity **for only a selected class**

Each new test  
(improving accuracy)  
reduces rule's coverage.



# Step-by-step example 1.

## Weather dataset

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

**If**    ?    **Then Yes**

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

Try each attribute - estimate accuracy:

If outlook=sunny then yes: 2/5

If outlook=overcast then yes: 3/4

If outlook=rainy then yes: 3/5

If temp=cool then yes: 3/4

If temp=mild then yes: 4/6

If temp=hot then yes: 2/4

If humidity=normal then yes: 6/7

If humidity=high then yes: 4/7

If windy=true then yes: 4/6

If windy=false then yes: 5/8



**If** ? **Then** no

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

If outlook=sunny then no: 3/5

If outlook=overcast then no: 1/4

If outlook=rainy then no: 2/5

If temp=cool then no: 1/4

If temp=mild then no: 2/6

If temp=hot then no: 2/4

If humidity=normal then no: 1/7

If humidity=high then no: 3/7

If windy=true then no: 2/6

If windy=false then no: 3/8

# R1: if humidity=normal then yes: 6/7

Maximum accuracy

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

If ? Then Yes

If outlook=sunny then yes: 2/5

If outlook=overcast then yes: 3/4

If outlook=rainy then yes: 3/5

If temp=cool then yes: 3/4

If temp=mild then yes: 4/6

If temp=hot then yes: 2/4

If humidity=normal then yes: 6/7

If humidity=high then yes: 4/7

If windy=true then yes: 4/6

If windy=false then yes: 5/8

# Adjust R1 to 100% accuracy

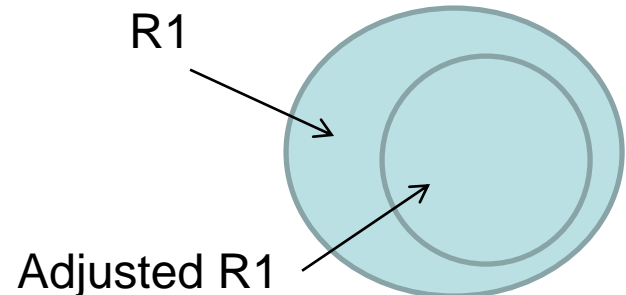
Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

R1: if humidity=normal and ? then yes

If outlook=sunny then yes: 2/2  
 If outlook=overcast then yes: 2/2  
 If outlook=rainy then yes: 2/3

If temp=cool then yes: 3/4  
 If temp=mild then yes: 2/2  
 If temp=hot then yes: 1/1

If windy=true then yes: 2/3  
If windy=false then yes: 4/4



R1: if humidity=normal and windy=false  
then yes

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

Remove all records covered by R1

# Continue with the remaining subset

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

**If** ? **Then** Yes

If outlook=sunny then yes: 1/4

If outlook=overcast then yes: 3/3

If outlook=rainy then yes: 1/3

If temp=cool then yes: 1/2

If temp=mild then yes: 3/5

If temp=hot then yes: 1/3

If humidity=normal then yes: 2/3

If humidity=high then yes: 5/7

If windy=true then yes: 4/6

If windy=false then yes: 2/4

Rules so far:

R1: if humidity=normal and windy=false → yes

# Continue with the remaining subset

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

**If** ? **Then** No

If outlook=sunny then no: 3/4  
 If outlook=overcast then no: 0/3  
 If outlook=rainy then no: 2/3

If temp=cool then no: 1/2  
 If temp=mild then no: 2/5  
 If temp=hot then no: 2/3

If humidity=normal then no: 1/3  
 If humidity=high then no: 2/7

If windy=true then no: 2/6  
 If windy=false then no: 2/4

Rules so far:

R1: if humidity=normal and windy=false → yes

# R2: if outlook=overcast then play: 3/3

100% accuracy

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

The rule has accuracy 100% - so no adjustments are needed

Rules so far:

R1: if humidity=normal and windy=false → yes

R2: if outlook=overcast → yes

# R2: if outlook=overcast then play: 3/3

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

Remove all records covered by R1, R2

Rules so far:

R1: if humidity=normal and windy=false → yes

R2: if outlook=overcast → yes



# Continue with the remaining subset

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

**If** ? **Then** Yes

If outlook=sunny then yes: 1/4

If outlook=rainy then yes: 1/3

If temp=cool then yes: 0/1

If temp=mild then yes: 2/4

If temp=hot then yes: 0/2

If humidity=normal then yes: 1/2

If humidity=high then yes: 1/5

If windy=true then yes: 1/4

If windy=false then yes: 1/3

Rules so far:

R1: if humidity=normal and windy=false → yes

R2: if outlook=overcast → yes

# Continue with the remaining subset

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

**If** ? **Then** No

If outlook=sunny then no: 3/4

If outlook=rainy then no: 2/3

If temp=cool then no: 1/1

If temp=mild then no: 2/4

If temp=hot then no: 2/2

If humidity=normal then no: 1/2

If humidity=high then no: 4/5

If windy=true then no: 3/4

If windy=false then no: 2/3

Rules so far:

R1: if humidity=normal and windy=false → yes

R2: if outlook=overcast → yes

# R3: if temp=hot then no: 2/2

100% accuracy

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

Rules so far:

R1: if humidity=normal and windy=false → yes

R2: if outlook=overcast → yes

R3: if temp=hot → no

# Continue with the remaining subset

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

**If** ? **Then** Yes

If outlook=sunny then yes: 1/2

If outlook=rainy then yes: 1/3

If temp=cool then yes: 0/1

If temp=mild then yes: 2/4

If humidity=normal then yes: 1/2

If humidity=high then yes: 1/3

If windy=true then yes: 1/3

If windy=false then yes: 1/2

Rules so far:

R1: if humidity=normal and windy=false → yes

R2: if outlook=overcast → yes

R3: if temp=hot → no

# Continue with the remaining subset

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

**If** ? **Then** no

If outlook=sunny then no: 1/2

If outlook=rainy then no: 2/3

Assuming that we do not care about rules that cover < 2 instances

If temp=cool then no: 1/1

If temp=mild then no: 2/4

If humidity=normal then no: 1/2

If humidity=high then no: 2/3

If windy=true then no: 2/3

If windy=false then no: 1/2

Rules so far:

R1: if humidity=normal and windy=false → yes

R2: if outlook=overcast → yes

R3: if temp=hot → no

# R4: if outlook=rainy then no: 2/3

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

If outlook=rainy and ? Then No

If temp=cool then no: 1/1

If temp=mild then no: 1/2

If humidity=normal then no: 1/1

If humidity=high then no: 1/2

If windy=true then no: 2/2

If windy=false then no: 0/1

Rules so far:

R1: if humidity=normal and windy=false → yes

R2: if outlook=overcast → yes

R3: if temp=hot → no

# R4: if outlook=rainy and windy=true then no: 3/3

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

Remove covered records

Etc....

Until every record is covered by some rule

Rules so far:

R1: if humidity=normal and windy=false → yes

R2: if outlook=overcast → yes

R3: if temp=hot → no

R4: if outlook=rainy and windy=true → no

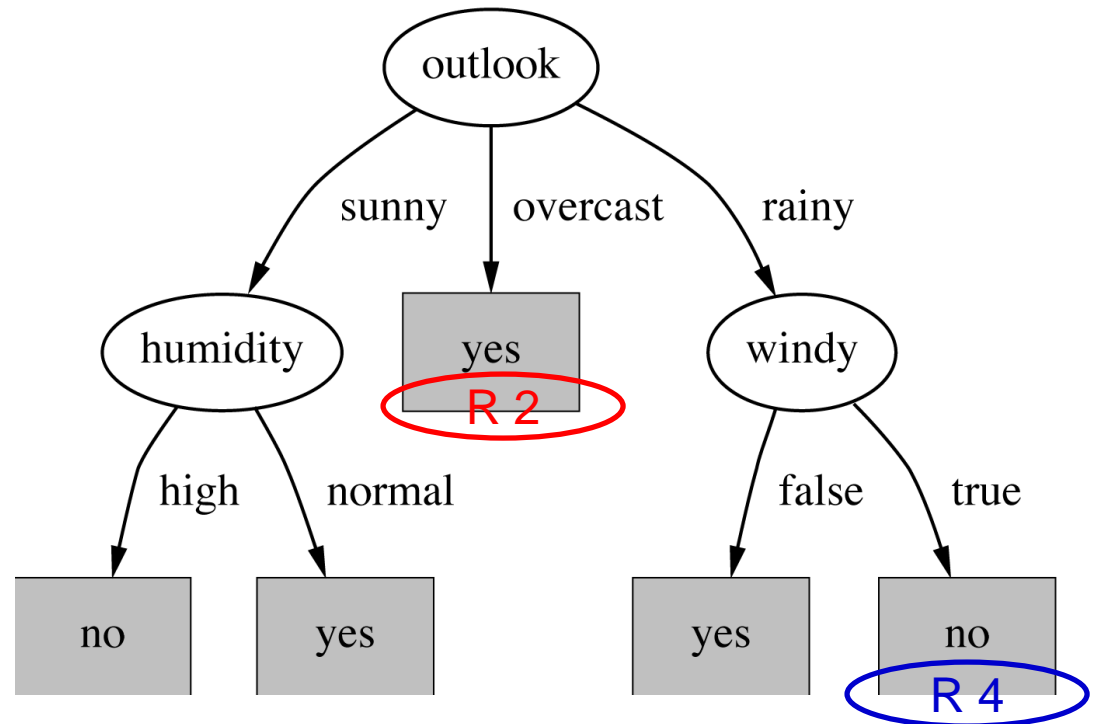
# Each rule corresponds to some path in the decision tree

R1: if humidity=normal and windy=false → yes

R2: if outlook=overcast → yes

R3: if temp=hot → no

R4: if outlook=rainy and windy=true → no





# Pseudocode for PRISM

ref

```
Initialize  $E$  to all records
Until  $E$  is not empty do:
     $E = \text{learn-one-rule}(E)$ 
```

Procedure **learn-one-rule** (set  $E$ ):

For each class  $C$

Heuristic: order  $C$  in ascending order of occurrence.

Initialize  $E_C$  to all instances with class label  $C$

Create a rule  $R$  with an empty left-hand side that predicts class  $C$

Until  $R$  is perfect (or there are no more attributes to use) do

For each attribute  $A_i$  not mentioned in  $R$ , and each attr. value  $v_j$ , consider adding the condition  $A_i = v_j$  to the left-hand side of  $R$

Select  $A_k = v_m$  to maximize the accuracy: **correct/all**

(break ties by choosing the condition with the largest **correct**)

Add  $A_k = v_m$  to  $R$

Remove the instances covered by  $R$  from  $E$

Return remaining instances

# Separate and conquer

- Methods like PRISM (for dealing with one class) are ***separate-and-conquer*** algorithms:
  - First, a rule is identified
  - Then, all instances covered by the rule are **separated out**
  - Finally, the remaining instances are “conquered”
- Difference to divide-and-conquer methods:
  - Subset covered by rule doesn't need to be explored any further

# Step-by-step example 2

## contact lenses data

Age	Spectacle prescription	Astigmatism	Tear production rate	Recommended Lenses
young	myope	no	reduced	none
young	myope	no	normal	soft
young	myope	yes	reduced	none
young	myope	yes	normal	hard
young	hypermetrope	no	reduced	none
young	hypermetrope	no	normal	soft
young	hypermetrope	yes	reduced	none
young	hypermetrope	yes	normal	hard
pre-presbyopic	myope	no	reduced	none
pre-presbyopic	myope	no	normal	soft
pre-presbyopic	myope	yes	reduced	none
pre-presbyopic	myope	yes	normal	hard
pre-presbyopic	hypermetrope	no	reduced	none
pre-presbyopic	hypermetrope	no	normal	soft
pre-presbyopic	hypermetrope	yes	reduced	none
pre-presbyopic	hypermetrope	yes	normal	none
presbyopic	myope	no	reduced	none
presbyopic	myope	no	normal	none
presbyopic	myope	yes	reduced	none
presbyopic	myope	yes	normal	hard
presbyopic	hypermetrope	no	reduced	none
presbyopic	hypermetrope	no	normal	soft
presbyopic	hypermetrope	yes	reduced	none
presbyopic	hypermetrope	yes	normal	none

# Example: contact lenses data

❖ Rule we seek:

If ?

then recommendation = hard

❖ Possible tests:

Age = Young	2/8
Age = Pre-presbyopic	1/8
Age = Presbyopic	1/8
Spectacle prescription = Myope	3/12
Spectacle prescription = Hypermetrope	1/12
Astigmatism = no	0/12
Astigmatism = yes	4/12
Tear production rate = Reduced	0/12
Tear production rate = Normal	4/12

The numbers on the right show the fraction of “correct” instances in the set singled out by that choice.

In this case, correct means that their recommendation is “hard.”

# Modified rule and resulting data

## ❖ Rule with best test added:

```
If astigmatism = yes  
then recommendation = hard
```

## ❖ Instances covered by modified rule:

Age	Spectacle prescription	Astigmatism	Tear production rate	Recommended lenses
Young	Myope	Yes	Reduced	None
Young	Myope	Yes	Normal	Hard
Young	Hypermetrope	Yes	Reduced	None
Young	Hypermetrope	Yes	Normal	hard
Pre-presbyopic	Myope	Yes	Reduced	None
Pre-presbyopic	Myope	Yes	Normal	Hard
Pre-presbyopic	Hypermetrope	Yes	Reduced	None
Pre-presbyopic	Hypermetrope	Yes	Normal	None
Presbyopic	Myope	Yes	Reduced	None
Presbyopic	Myope	Yes	Normal	Hard
Presbyopic	Hypermetrope	Yes	Reduced	None
Presbyopic	Hypermetrope	Yes	Normal	None

The rule isn't very accurate, getting only 4 out of 12 that it covers. So, it needs further refinement.

# Further refinement

## ❖ Current state:

```
If astigmatism = yes
    and ?
    then recommendation = hard
```

## ❖ Possible tests:

Age = Young	2/4
Age = Pre-presbyopic	1/4
Age = Presbyopic	1/4
Spectacle prescription = Myope	3/6
Spectacle prescription = Hypermetrope	1/6
Tear production rate = Reduced	0/6
Tear production rate = Normal	4/6

# Modified rule and resulting data

## ❖ Rule with best test added:

```
If astigmatism = yes
    and tear production rate =
normal
    then recommendation = hard
```

## ❖ Instances covered by modified rule:

Age	Spectacle prescription	Astigmatism	Tear production rate	Recommended lenses
Young	Myope	Yes	Normal	Hard
Young	Hypermetrope	Yes	Normal	hard
Pre-presbyopic	Myope	Yes	Normal	Hard
Pre-presbyopic	Hypermetrope	Yes	Normal	None
Presbyopic	Myope	Yes	Normal	Hard
Presbyopic	Hypermetrope	Yes	Normal	None

Should we stop here? Perhaps. But let's say we are going for exact rules, no matter how complex they become.

So, let's refine further.

# Further refinement

## ❖ Current state:

```
If astigmatism = yes
    and tear production rate = normal
    and ?
    then recommendation = hard
```

## ❖ Possible tests:

Age = Young	2/2
Age = Pre-presbyopic	1/2
Age = Presbyopic	1/2
Spectacle prescription = Myope	3/3
Spectacle prescription = Hypermetrope	1/3

## ❖ Tie between the first and the fourth test

- ❑ We choose the one with greater coverage



# The result

## ❖ Final rule:

```
If astigmatism = yes  
and tear production rate = normal  
and spectacle prescription = myope  
then recommendation = hard
```

## ❖ Second rule for recommending “hard lenses”: (built from instances not covered by first rule)

```
If age = young and astigmatism = yes  
and tear production rate = normal  
then recommendation = hard
```

- ## ❖ These two rules cover all “hard lenses”:
- ❑ Process is repeated with other two classes

# Rule learners

1. PRISM – as we learned. Only nominal attributes (Cendrowska)
2. Ridor - RIpple-DOWn Rule learner (Gaines and Compton)
3. PART (Eibe and Witten)
4. JRip - Repeated Incremental Pruning to Produce Error Reduction (William W. Cohen)
5. Decision table (Kohavi)