ID3 Algorithm

ID3(*Examples*, *Target_attribute*, *Attributes*)

Examples are the training examples. Target_attribute is the attribute whose value is to be predicted by the tree. Attributes is a list of other attributes that may be tested by the learned decision tree. Returns a decision tree that correctly classifies the given Examples.

target:
PlayTennis (yes, no)

attributes:
Humidity (high, normal)
Wind (strong, weak)
Outlook (sunny, overcast, rain)

examples:
Outlook=rain ∧ Wind=strong ∧ PlayTennis=no
...

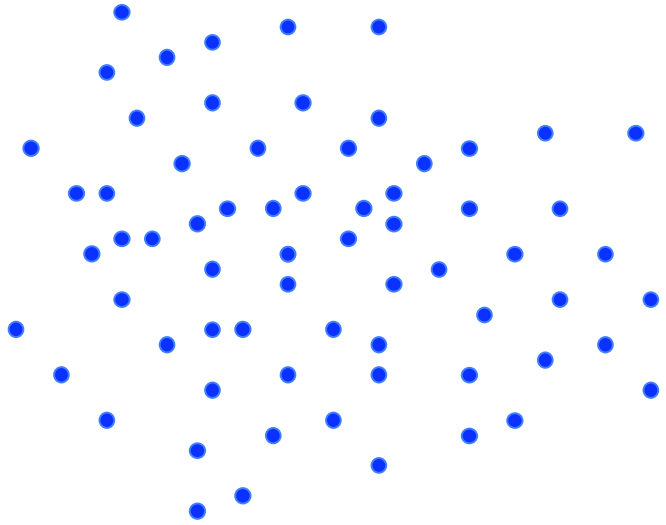
ID3 Algorithm

ID3(*Examples*, *Target_attribute*, *Attributes*)

Examples are the training examples. Target_attribute is the attribute whose value is to be
utes is a list of other attributes that may be tested by the learned
ision tree that correctly classifies the given Examples.

attributes divide up the space of objects (days)

examples:



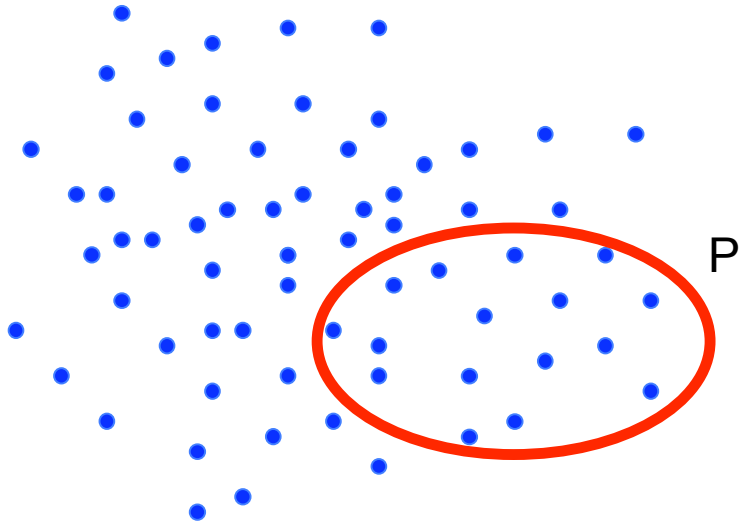
ID3 Algorithm

ID3(*Examples*, *Target_attribute*, *Attributes*)

Examples are the training examples. Target_attribute is the attribute whose value is to be
utes is a list of other attributes that may be tested by the learned
ision tree that correctly classifies the given Examples.

attributes divide up the space of objects (days)

examples:



PlayTennis=yes

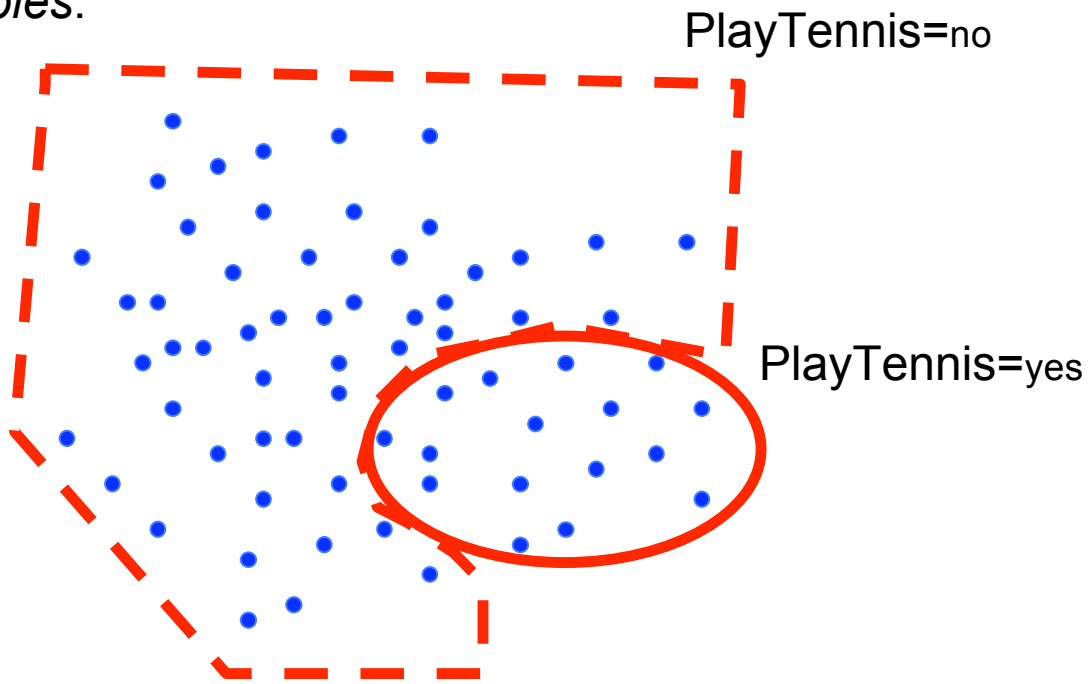
ID3 Algorithm

ID3(*Examples*, *Target_attribute*, *Attributes*)

Examples are the training examples. Target_attribute is the attribute whose value is to be
utes is a list of other attributes that may be tested by the learned
ision tree that correctly classifies the given Examples.

attributes divide up the space of objects (days)

examples:



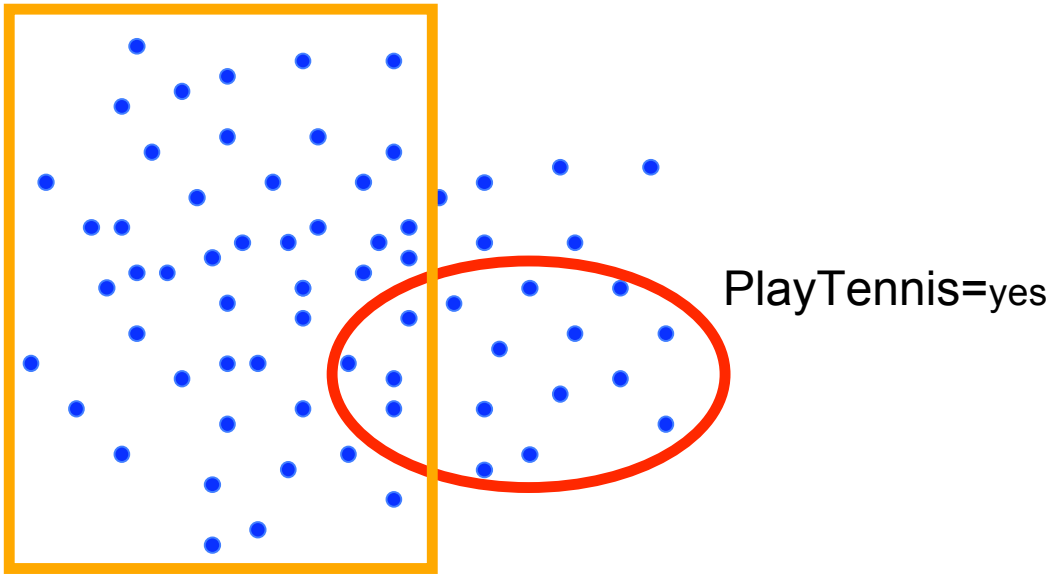
ID3 Algorithm

ID3(*Examples*, *Target_attribute*, *Attributes*)

Examples are the training examples. Target_attribute is the attribute whose value is to be
utes is a list of other attributes that may be tested by the learned
ision tree that correctly classifies the given Examples.

attributes divide up the space of objects (days)

examples:



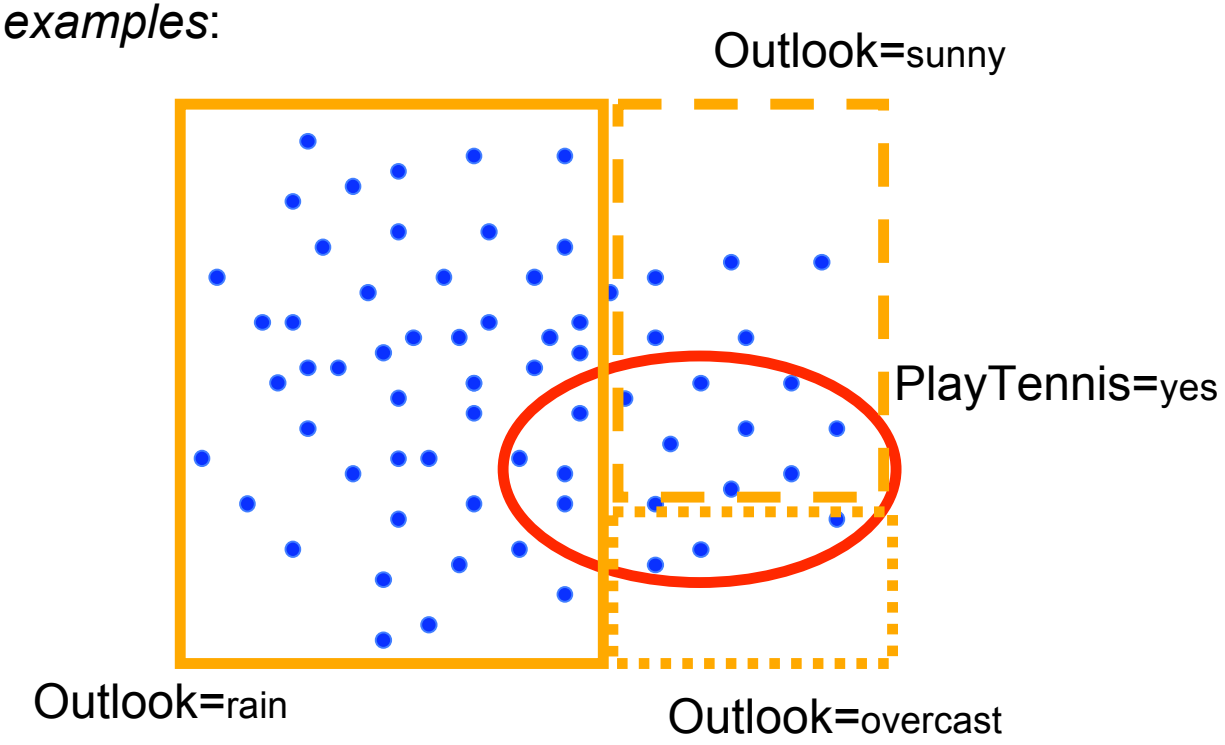
ID3 Algorithm

ID3(*Examples*, *Target_attribute*, *Attributes*)

Examples are the training examples. Target_attribute is the attribute whose value is to be
utes is a list of other attributes that may be tested by the learned
ision tree that correctly classifies the given Examples.

attributes divide up the space of objects (days)

examples:



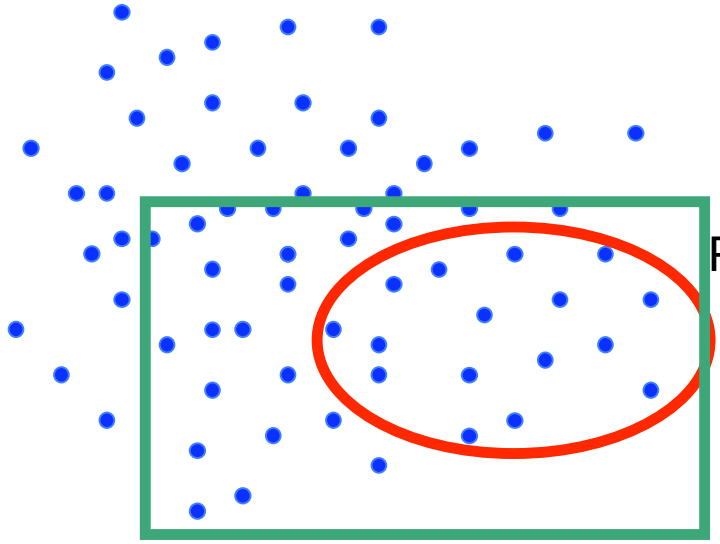
ID3 Algorithm

ID3(*Examples*, *Target_attribute*, *Attributes*)

Examples are the training examples. Target_attribute is the attribute whose value is to be
utes is a list of other attributes that may be tested by the learned
ision tree that correctly classifies the given Examples.

attributes divide up the space of objects (days)

examples:



PlayTennis=yes

Humidity=normal

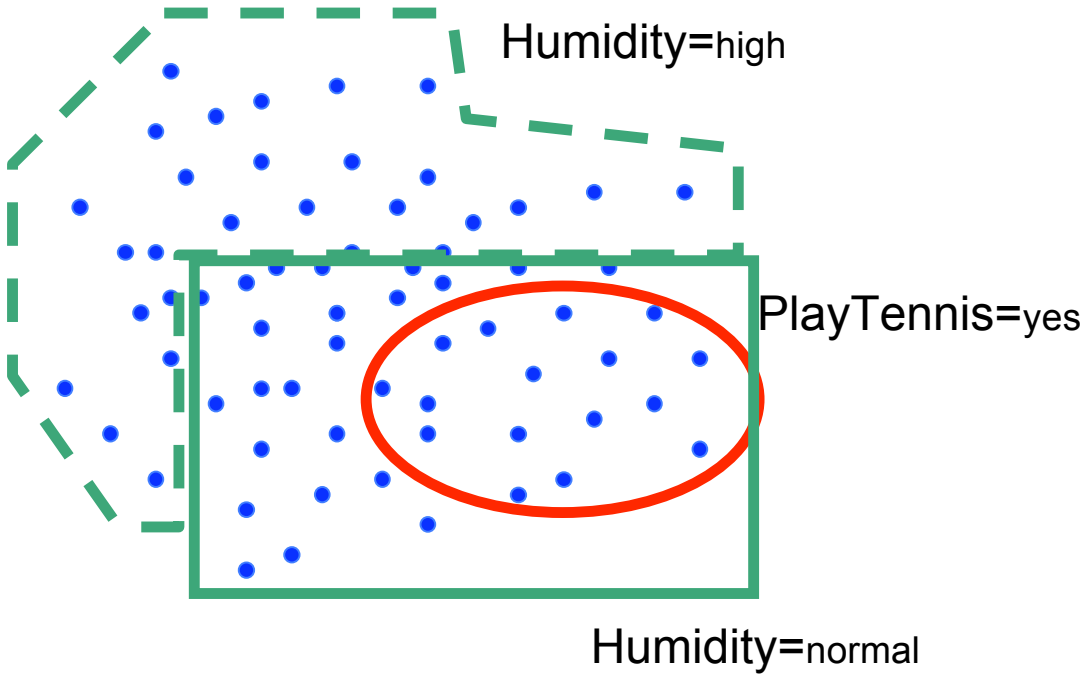
ID3 Algorithm

ID3(*Examples*, *Target_attribute*, *Attributes*)

Examples are the training examples. Target_attribute is the attribute whose value is to be
utes is a list of other attributes that may be tested by the learned
ision tree that correctly classifies the given Examples.

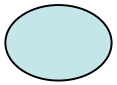
attributes divide up the space of objects (days)

examples:



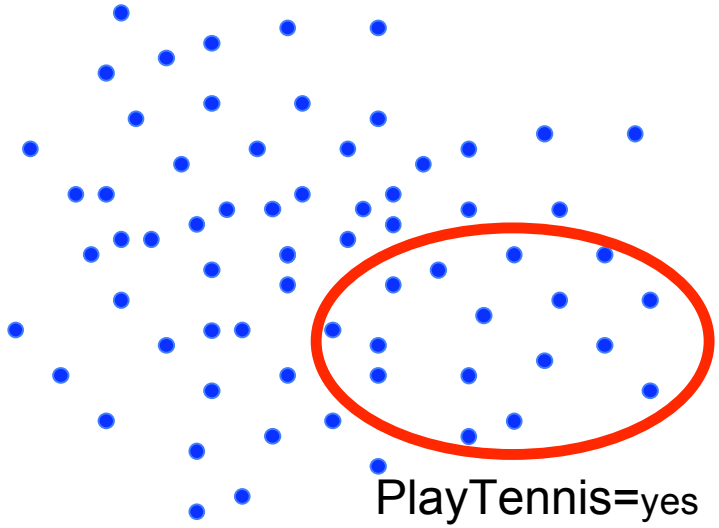
ID3 Algorithm

- Create a *Root* node for the tree



attributes:
Humidity (high, normal)
Wind (strong, weak)
Outlook (sunny, overcast, rain)

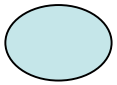
examples:



ID3 Algorithm

- If all *Examples* are positive, Return the single-node tree *Root*, with label = +
- If all *Examples* are negative, Return the single-node tree *Root*, with label = −
- If *Attributes* is empty, Return the single-node tree *Root*, with label = most common value of *Target_attribute* in *Examples*

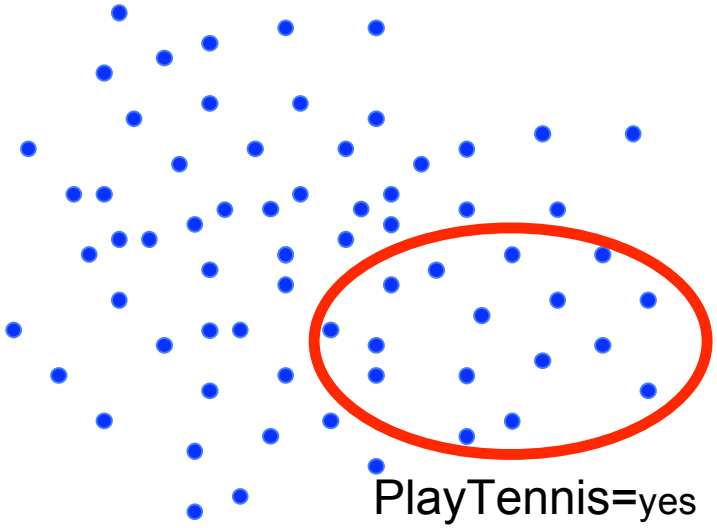
(these basically handle degenerate cases)



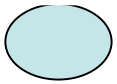
attributes:

Humidity (high, normal)
Wind (strong, weak)
Outlook (sunny, overcast, rain)

examples:

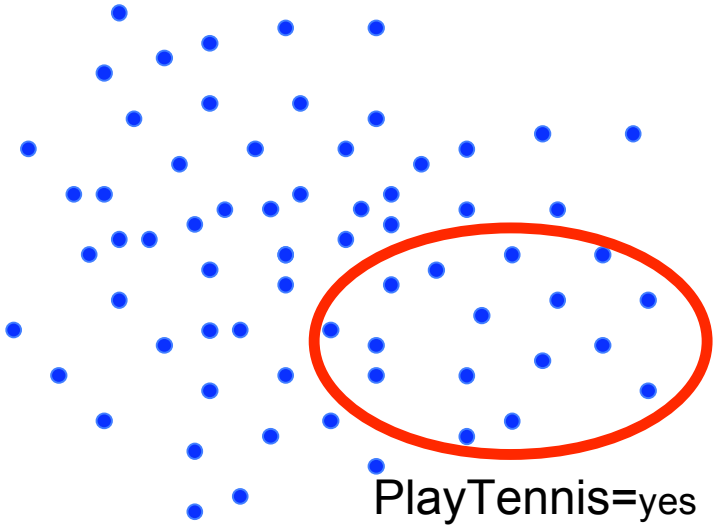


- Otherwise Begin
  - $A \leftarrow$ the attribute from *Attributes* that best* classifies *Examples*



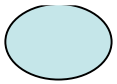
attributes:
- Humidity (high, normal)
- Wind (strong, weak)
- Outlook (sunny, overcast, rain)

examples:



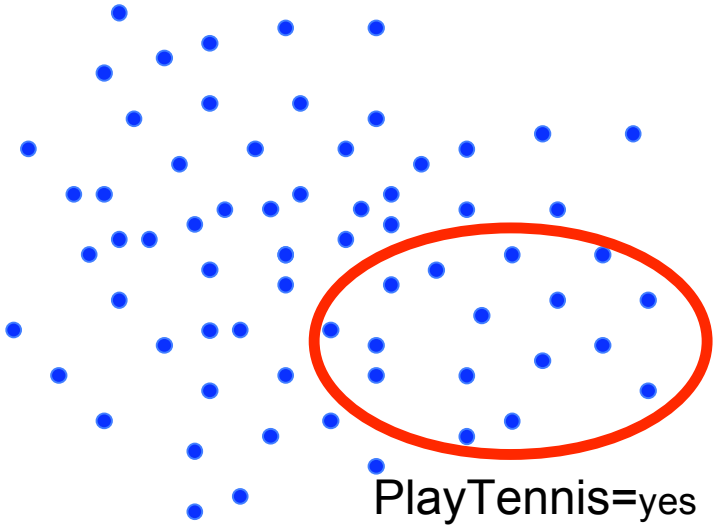
- Otherwise Begin
  - $A \leftarrow$ the attribute from *Attributes* that best* classifies *Examples*

we'll talk about this later
for now, assume some attribute is more strongly related to whether we "PlayTennis" than others



attributes:
- Humidity (high, normal)
- Wind (strong, weak)
- Outlook (sunny, overcast, rain)

examples:



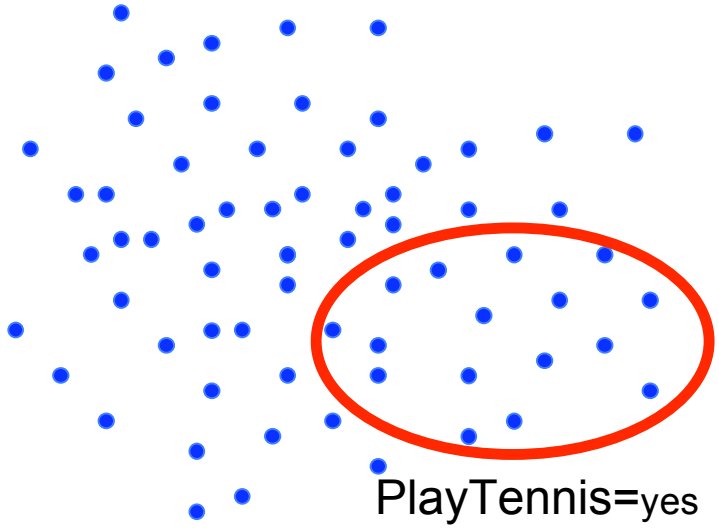
- Otherwise Begin
  - $A \leftarrow$ the attribute from *Attributes* that best* classifies *Examples*
  - The decision attribute for *Root* $\leftarrow A$

we'll talk about this later
for now, assume some attribute is more strongly related to whether we "PlayTennis" than others

Outlook

attributes:
Humidity (high, normal)
Wind (strong, weak)
Outlook (sunny, overcast, rain)

examples:

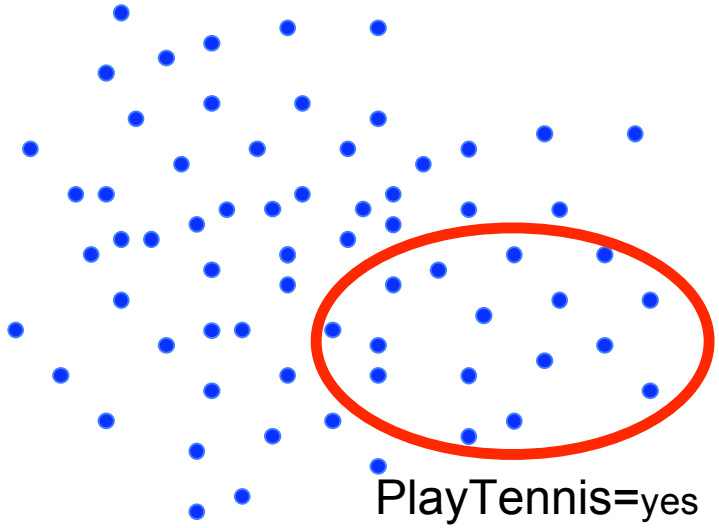


- Otherwise Begin
  - $A \leftarrow$ the attribute from *Attributes* that best* classifies *Examples*
  - The decision attribute for *Root* $\leftarrow A$
  - For each possible value, v_i, of A,

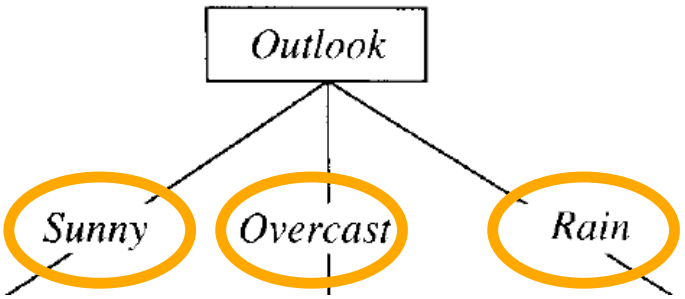
Outlook

attributes:
Humidity (high, normal)
Wind (strong, weak)
Outlook (sunny, overcast, rain)

examples:

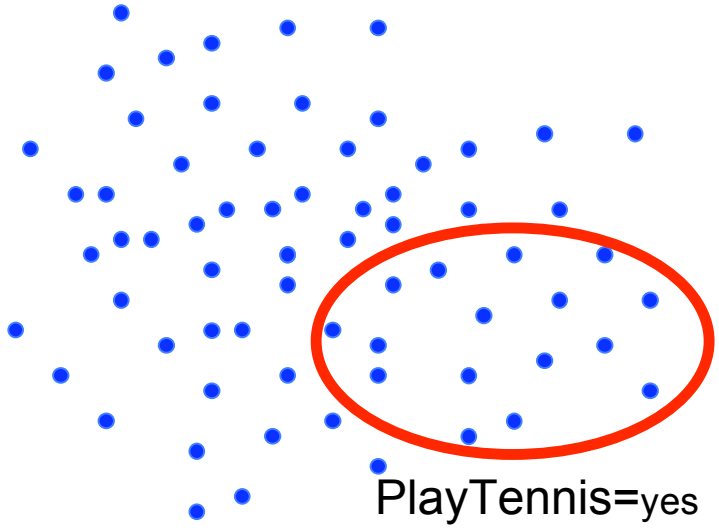


- Otherwise Begin
  - $A \leftarrow$ the attribute from *Attributes* that best* classifies *Examples*
  - The decision attribute for *Root* $\leftarrow A$
  - For each possible value, v_i, of A,
    - Add a new tree branch below *Root*, corresponding to the test $A = v_i$

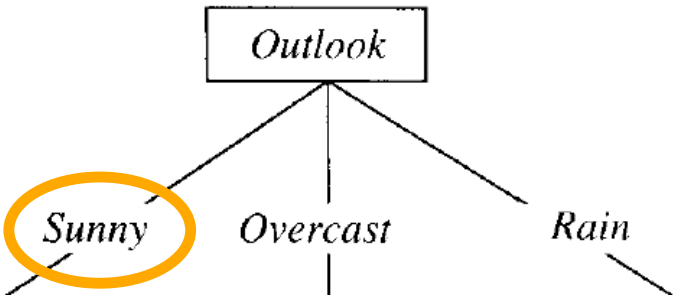


attributes:
Humidity (high, normal)
Wind (strong, weak)
Outlook (sunny, overcast, rain)

examples:

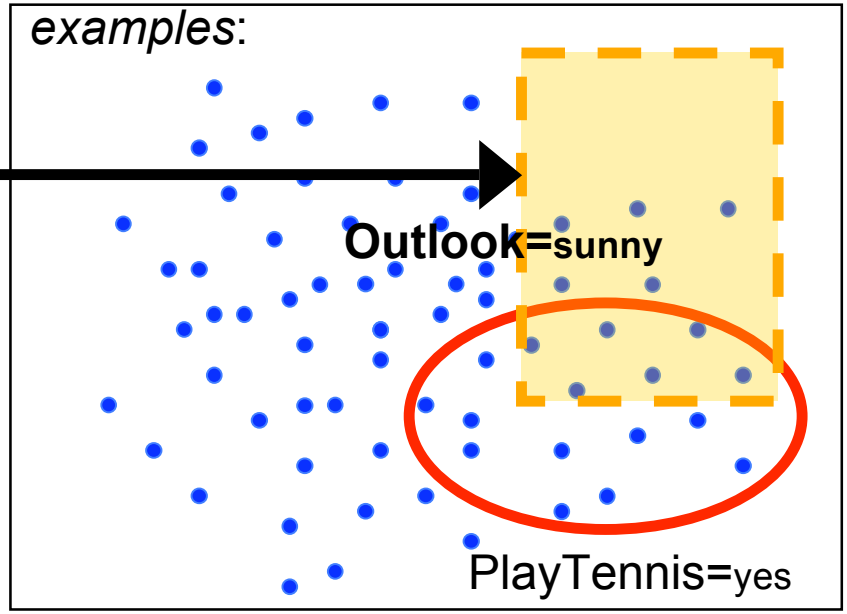


- Otherwise Begin
  - $A \leftarrow$ the attribute from *Attributes* that best* classifies *Examples*
  - The decision attribute for *Root* $\leftarrow A$
  - For each possible value, v_i, of A,
    - Add a new tree branch below *Root*, corresponding to the test $A = v_i$
    - Let $Examples_{v_i}$ be the subset of *Examples* that have value v_i for A

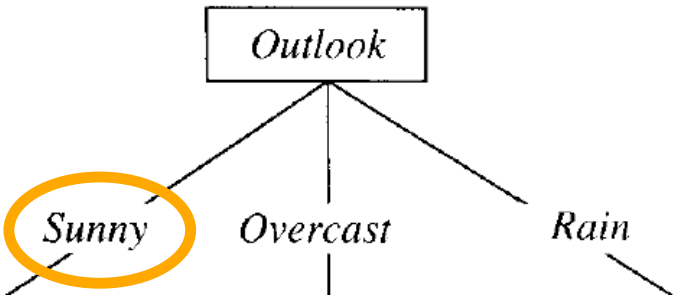


$Examples_{sunny}$

attributes:
Humidity (high, normal)
Wind (strong, weak)
Outlook (sunny, overcast, rain)

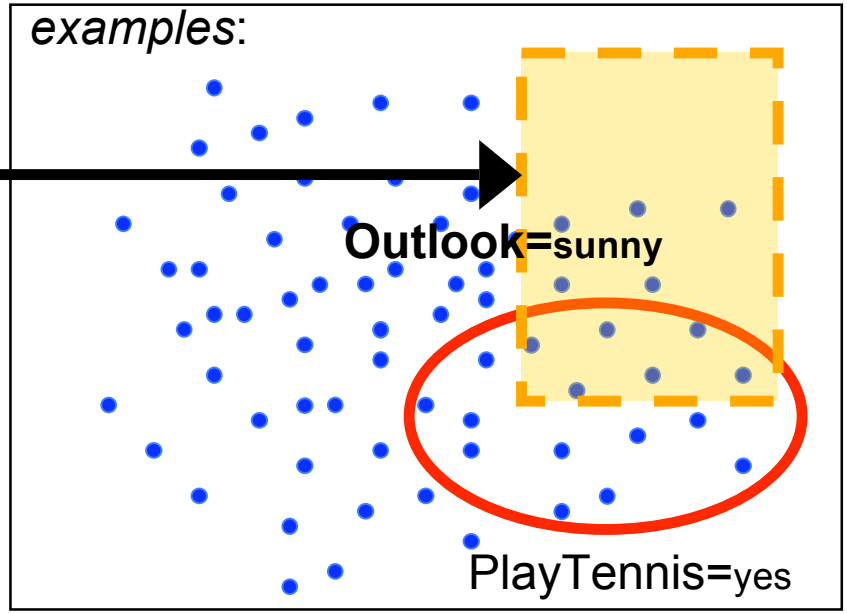


- Otherwise Begin
  - $A \leftarrow$ the attribute from *Attributes* that best* classifies *Examples*
  - The decision attribute for *Root* $\leftarrow A$
  - For each possible value, v_i, of A,
    - Add a new tree branch below *Root*, corresponding to the test $A = v_i$
    - Let $Examples_{v_i}$ be the subset of *Examples* that have value v_i for A
    - If $Examples_{v_i}$ is empty



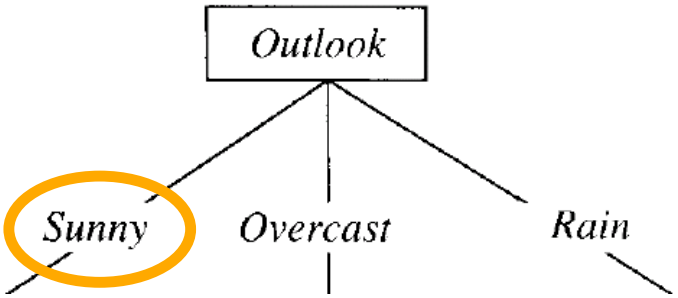
$Examples_{sunny}$

attributes:
Humidity (high, normal)
Wind (strong, weak)
Outlook (sunny, overcast, rain)



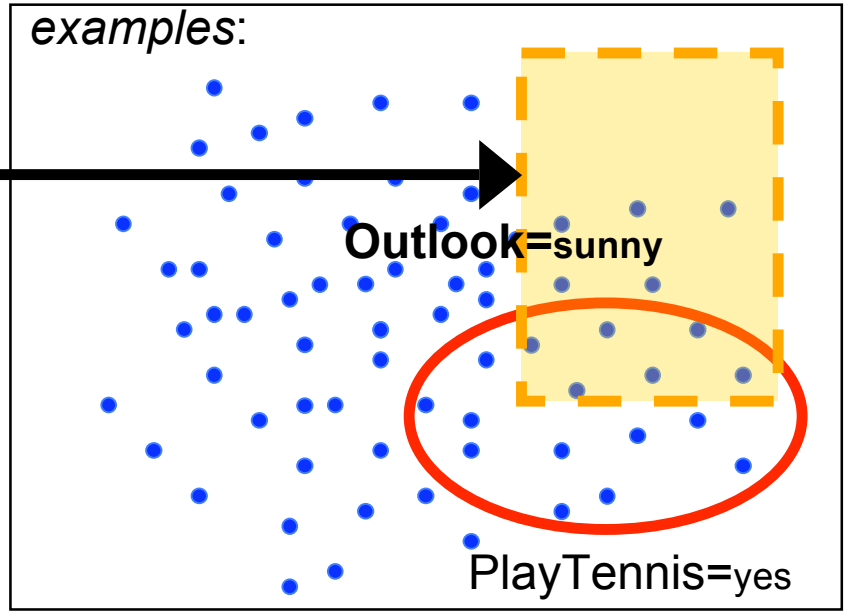
- Otherwise Begin
  - $A \leftarrow$ the attribute from *Attributes* that best* classifies *Examples*
  - The decision attribute for *Root* $\leftarrow A$
  - For each possible value, v_i, of A,
    - Add a new tree branch below *Root*, corresponding to the test $A = v_i$
    - Let $Examples_{v_i}$ be the subset of *Examples* that have value v_i for A
    - If $Examples_{v_i}$ is empty

      - Else below this new branch add the subtree
        ID3($Examples_{v_i}$, *Target_attribute*, *Attributes* – {A}))
- End



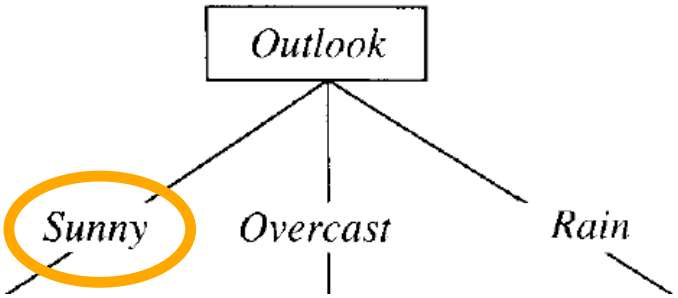
$Examples_{sunny}$

attributes:
Humidity (high, normal)
Wind (strong, weak)
Outlook (sunny, overcast, rain)



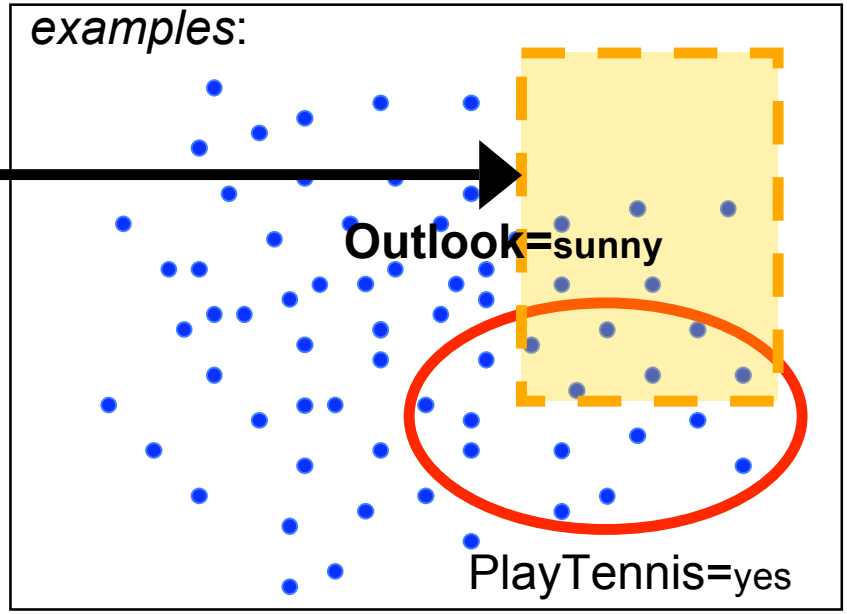
- Otherwise Begin
  - $A \leftarrow$ the attribute from *Attributes* that best* classifies *Examples*
  - The decision attribute for *Root* $\leftarrow A$
  - For each possible value, v_i, of A
    - Add a new tree branch
    - Let $Examples_{v_i}$ be the
    - If $Examples_{v_i}$ is empt
    - Else below this new branch add the subtree
      ID3($Examples_{v_i}$, *Target_attribute*, *Attributes* – {A}))
- End

recursive call to ID3, with
- remaining set of examples ($Examples_{sunny}$)
- set of attributes MINUS "Outlook"

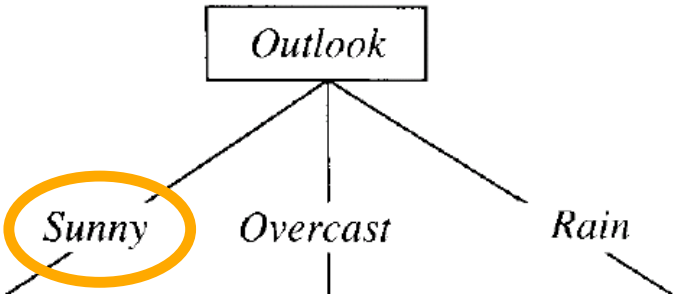


$Examples_{sunny}$

attributes:
Humidity (high, normal)
Wind (strong, weak)
Outlook (sunny, overcast, rain)

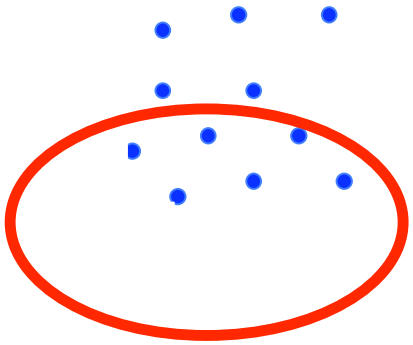


- Otherwise Begin
  - $A \leftarrow$ the attribute from *Attributes* that best* classifies *Examples*



attributes:
Humidity (high, normal)
Wind (strong, weak)

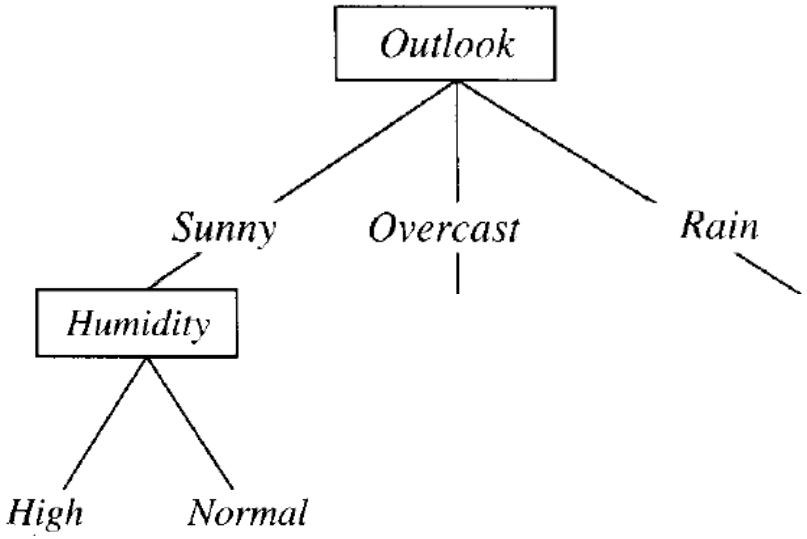
examples:



PlayTennis=yes

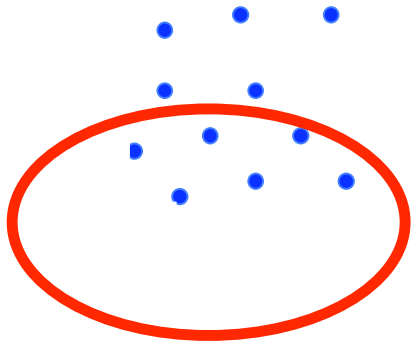
- Otherwise Begin
  - $A \leftarrow$ the attribute from *Attributes* that best* classifies *Examples*
  - The decision attribute for *Root* $\leftarrow A$
  - For each possible value, v_i, of A,
    - Add a new tree branch below *Root*, corresponding to the test $A = v_i$

just repeat the steps...



attributes:
Humidity (high, normal)
Wind (strong, weak)

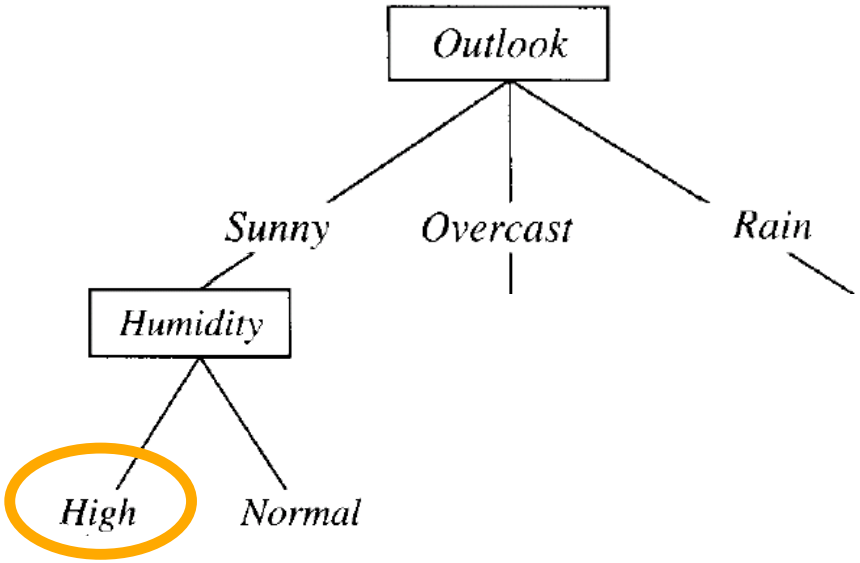
examples:



PlayTennis=yes

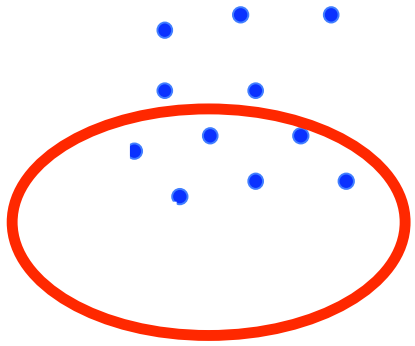
- Otherwise Begin
  - $A \leftarrow$ the attribute from *Attributes* that best* cl
  - The decision attribute for *Root* $\leftarrow A$
  - For each possible value, v_i, of A,
    - Add a new tree branch below *Root*, corresponding to the test $A = v_i$
    - Let $Examples_{v_i}$ be the subset of *Examples* that have value v_i for A
    - If $Examples_{v_i}$ is empty
      - Then below this new branch add a leaf node with label = most common value of *Target_attribute* in *Examples*

just repeat the steps...



attributes:
Humidity (high, normal)
Wind (strong, weak)

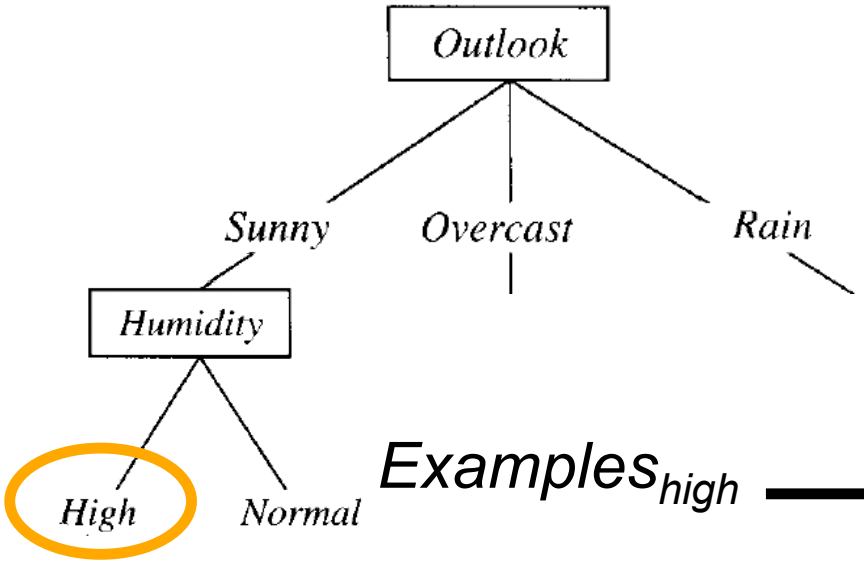
examples:



PlayTennis=yes

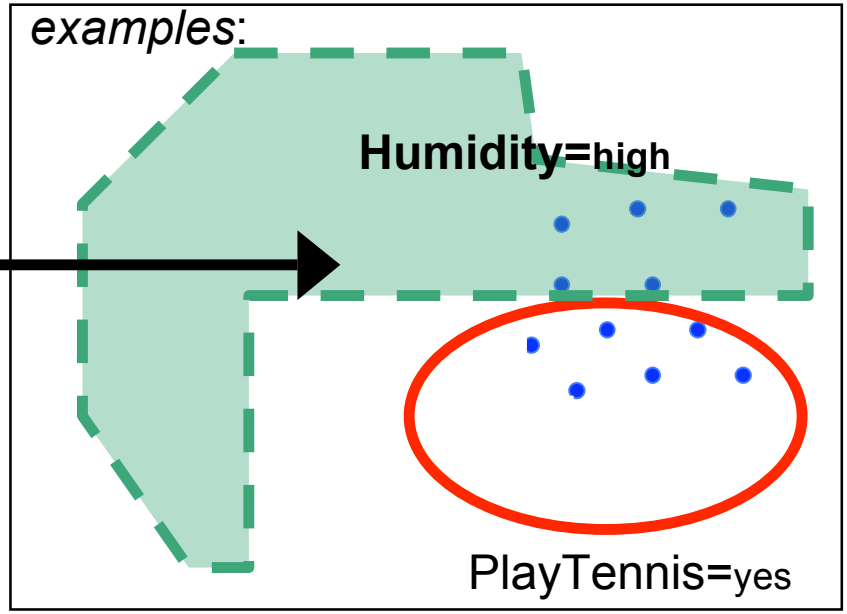
- Otherwise Begin
  - $A \leftarrow$ the attribute from *Attributes* that best* cl
  - The decision attribute for *Root* $\leftarrow A$
  - For each possible value, v_i, of A,
    - Add a new tree branch below *Root*, corresponding to the test $A = v_i$
    - Let $Examples_{v_i}$ be the subset of *Examples* that have value v_i for A
    - If $Examples_{v_i}$ is empty
      - Then below this new branch add a leaf node with label = most common value of *Target_attribute* in *Examples*

just repeat the steps...



$Examples_{high}$

attributes:
Humidity (high, normal)
Wind (strong, weak)

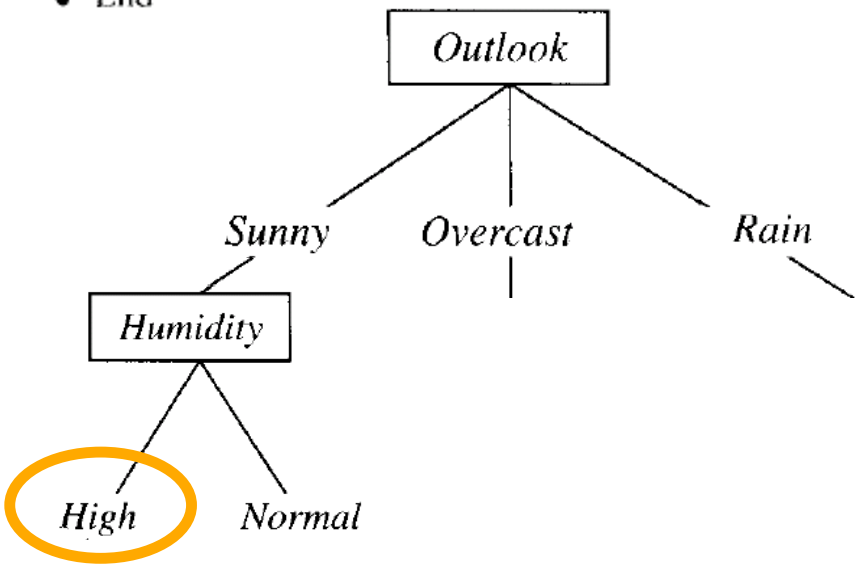


- Otherwise Begin
  - $A \leftarrow$ the attribute from *Attributes* that best* classifies *Examples*
  - The decision attribute for *Root* $\leftarrow A$
  - For each possible value, v_i, of A
    - Add a new tree branch
    - Let $Examples_{v_i}$ be the
    - If $Examples_{v_i}$ is empt

recursive call to ID3, with
- remaining set of examples ($Examples_{high}$)
- set of attributes MINUS "Humidity"

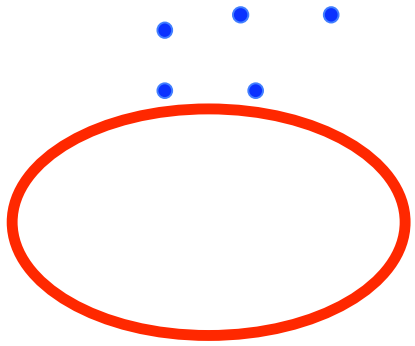
$ID3(Examples_{v_i}, Target\_attribute, Attributes - \{A\})$

- End



attributes:
Humidity (high, normal)
Wind (strong, weak)

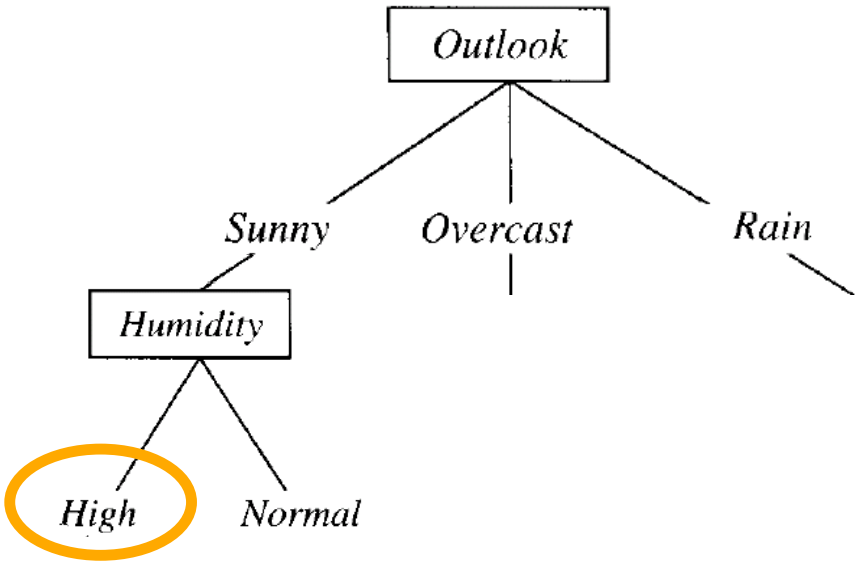
examples:



PlayTennis=yes

- If all *Examples* are positive, Return the single-node tree *Root*, with label = +
- If all *Examples* are negative, Return the single-node tree *Root*, with label = −
- If *Attributes* is empty, Return the single-node tree *Root*, with label = most common value of *Target_attribute* in *Examples*

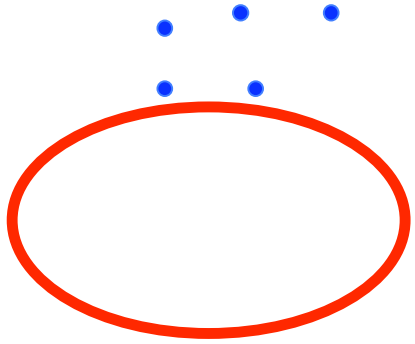
All examples are negative – return "No"



attributes:

Wind (strong, weak)

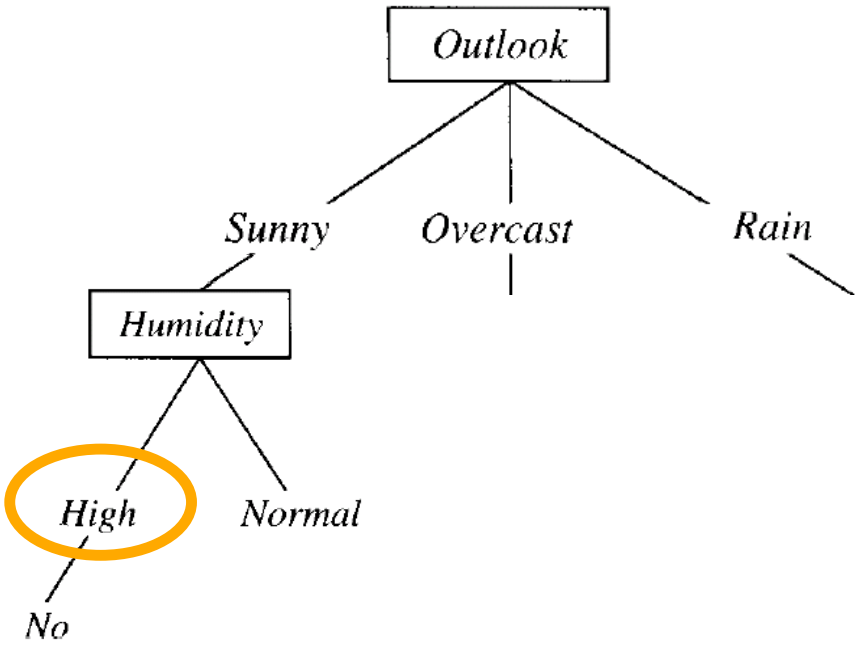
examples:



PlayTennis=yes

- If all *Examples* are positive, Return the single-node tree *Root*, with label = +
- If all *Examples* are negative, Return the single-node tree *Root*, with label = −
- If *Attributes* is empty, Return the single-node tree *Root*, with label = most common value of *Target_attribute* in *Examples*

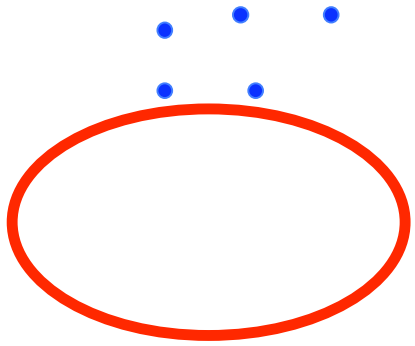
All examples are negative – return "No"



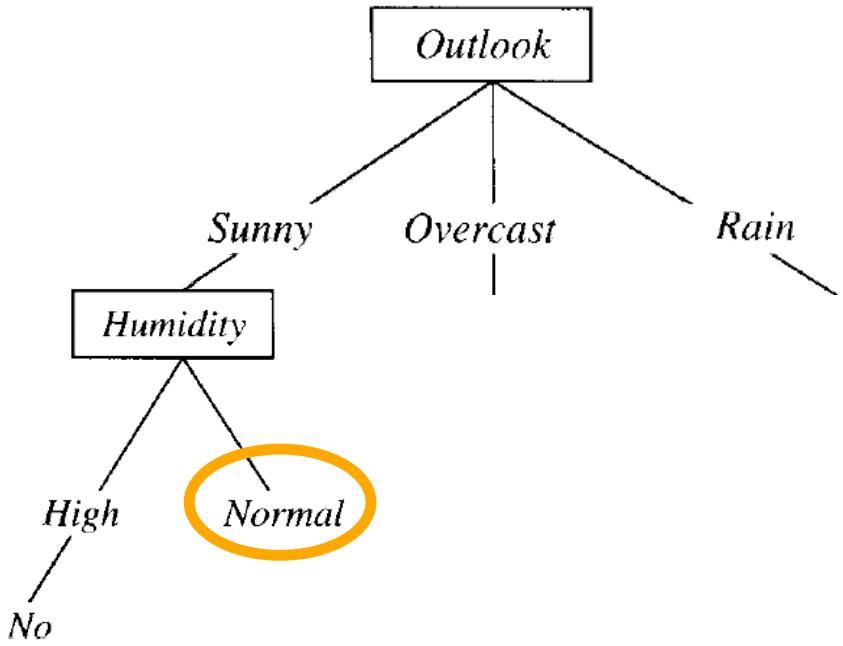
attributes:

Wind (strong, weak)

examples:

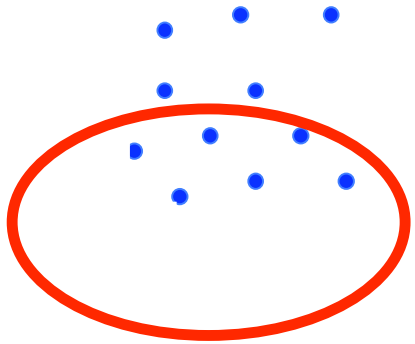


PlayTennis=yes



attributes:
Humidity (high, normal)
Wind (strong, weak)

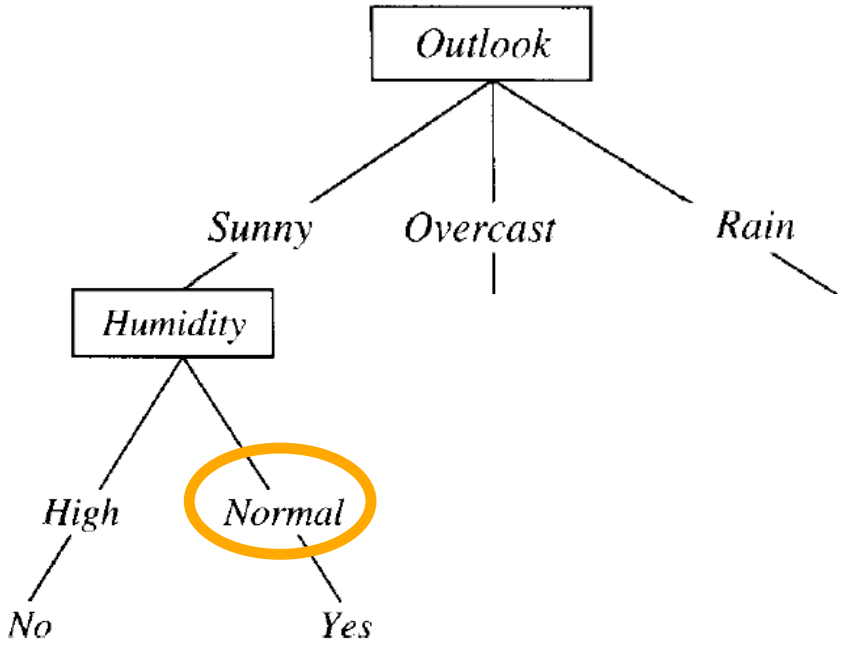
examples:



PlayTennis=yes

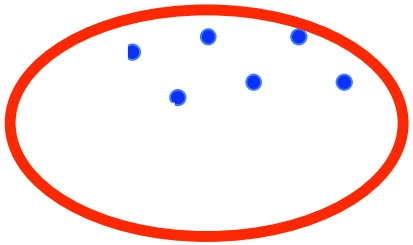
now just repeat the algorithm

here I'll assume something similar happens with "Normal" but positive



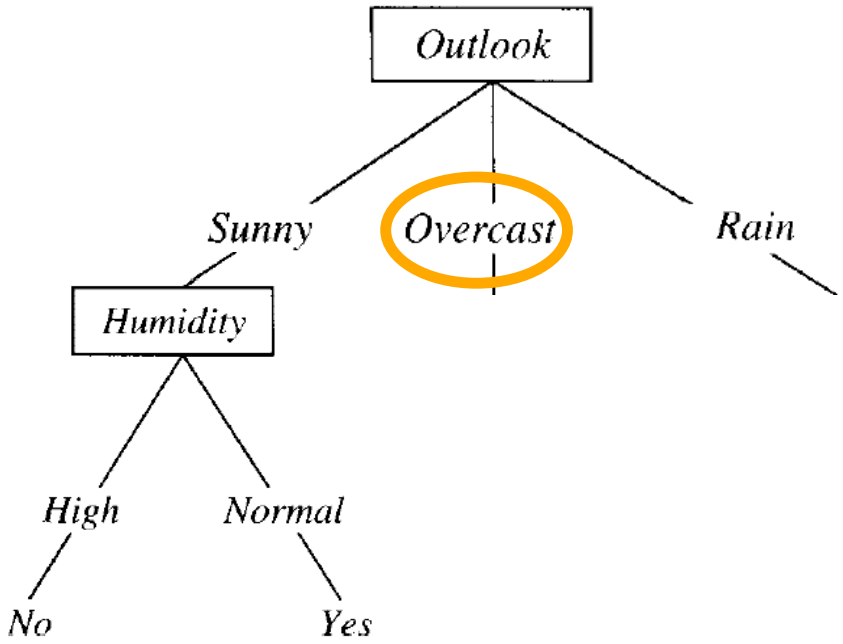
attributes:
Humidity (high, normal)
Wind (strong, weak)

examples:



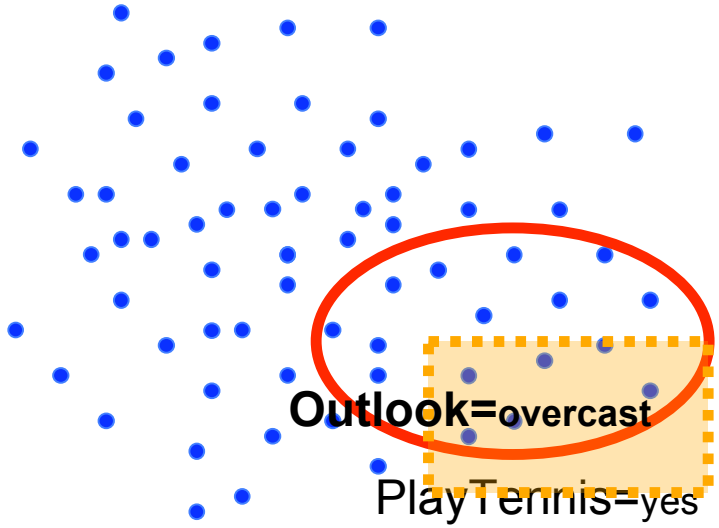
PlayTennis=yes

now just repeat the algorithm...

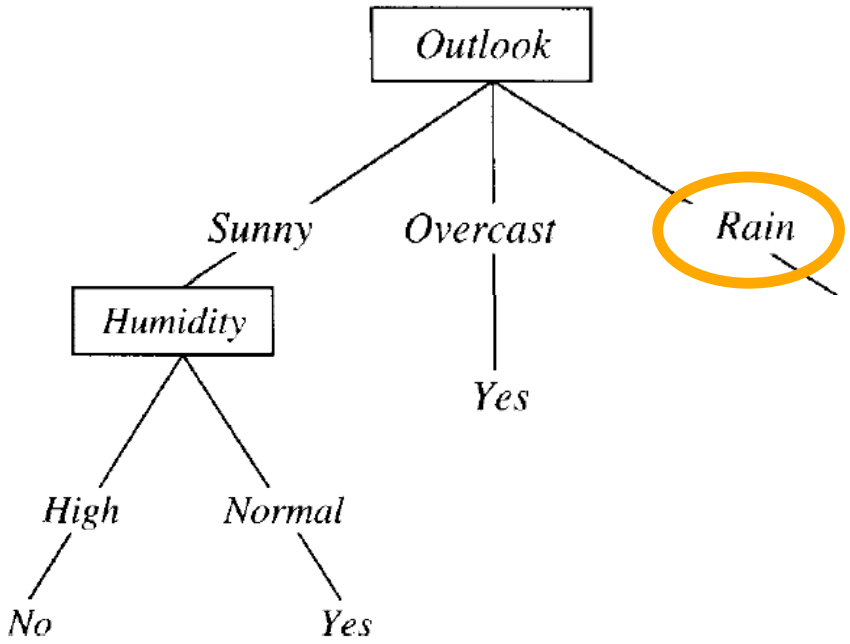


attributes:
- Humidity (high, normal)
- Wind (strong, weak)
- Outlook (sunny, overcast, rain)

examples:

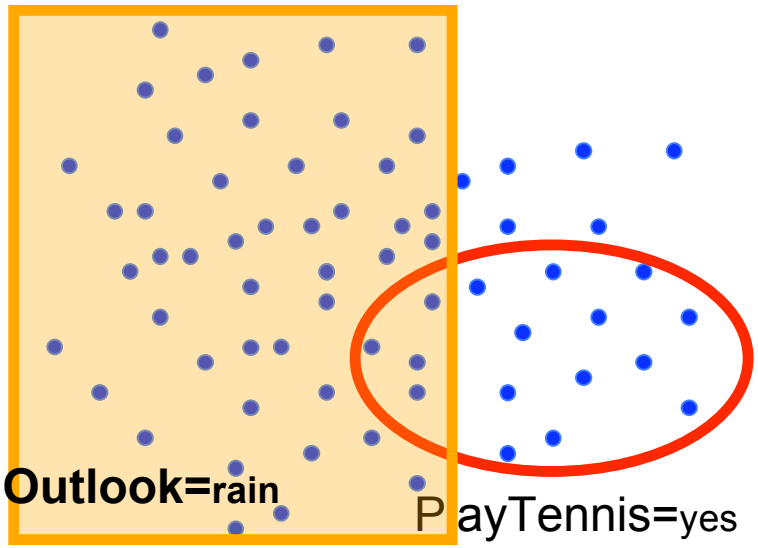


now just repeat the algorithm...

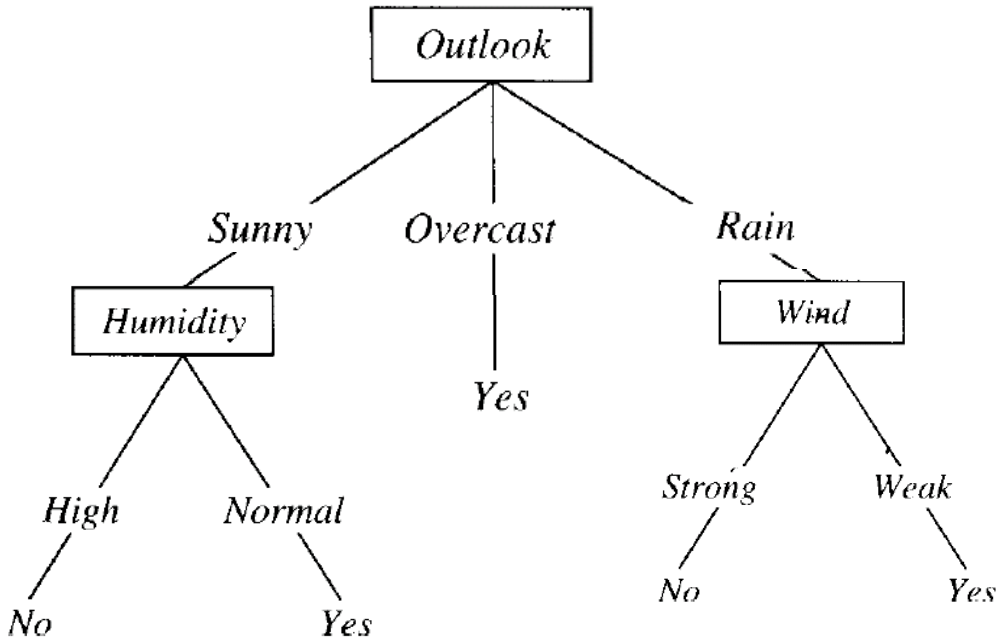


attributes:
- Humidity (high, normal)
- Wind (strong, weak)
- Outlook (sunny, overcast, rain)

examples:



(eventually) done!



attributes:
- Humidity (high, normal)
- Wind (strong, weak)
- Outlook (sunny, overcast, rain)

examples: