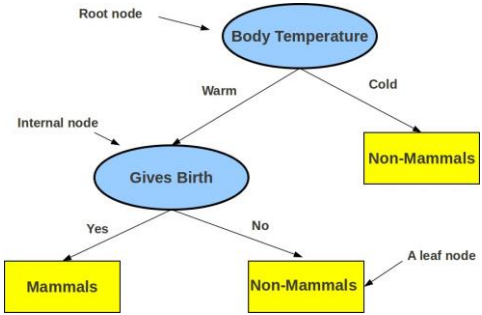
Decision Trees for Classification and Regression

Slides credit: Piyush Rai, IIT Kanpur
Edited by: Dr. Allah Bux Sargana

Decision Trees
(An example of rule-based classification system)

Decision Tree

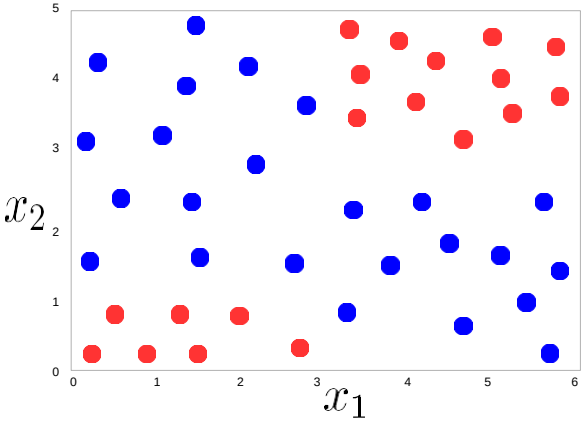
- Defines a tree-structured hierarchy of rules: can be considered a set of if-then rules for final prediction
- Consists of a root node, internal nodes, and leaf nodes



- Root and internal nodes contain the rules Leaf nodes define the predictions
- Decision Tree (DT) learning is about learning such a tree from labeled training data

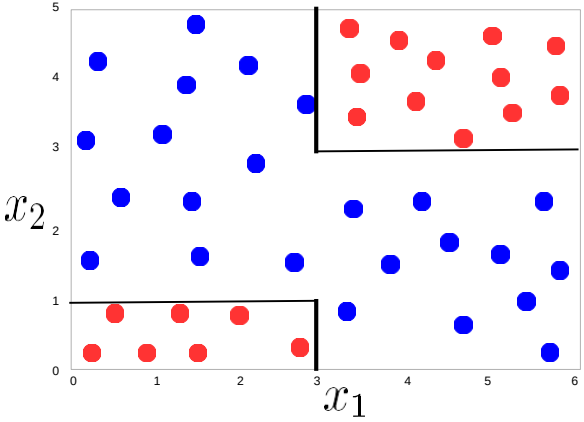
A Classification Problem

Consider binary classification. Assume training data with each input having 2 features (x_1,x_2)

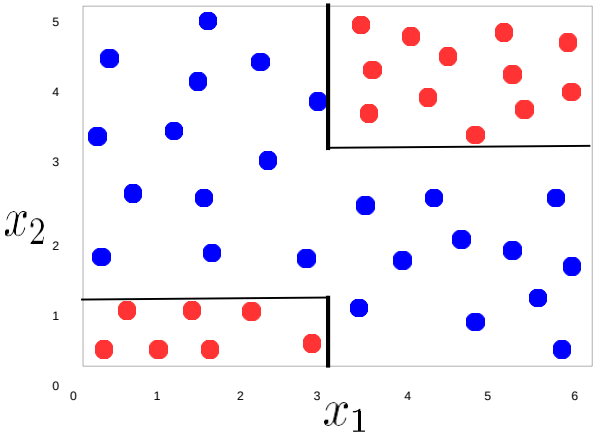


A Classification Problem

The “expected” decision boundary given this training data.



A Classification Problem

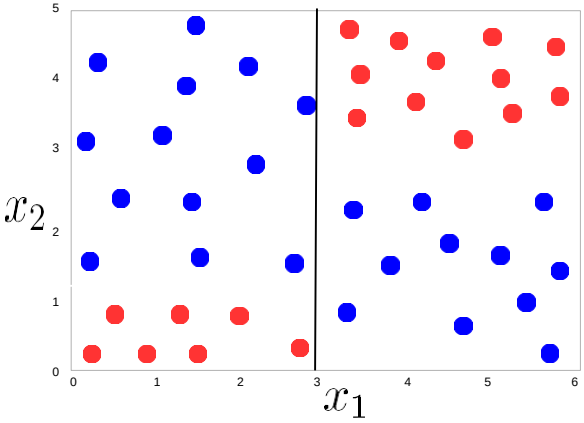


The “expected” decision boundary given this training data.

Let’s learn this in the form of a set of if-then rules!

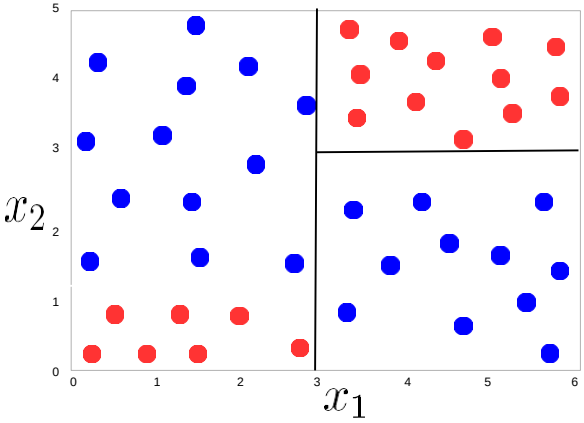
Learning by Asking Questions!

Is x_1 (feature 1) greater than 3 ?



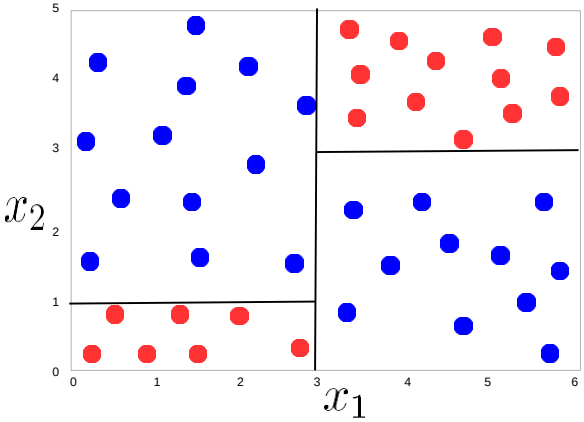
Learning by Asking Questions!

Given $x_1 > 3$, is feature 2 (x_2) greater than 3?



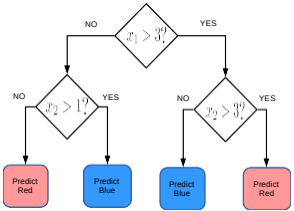
Learning by Asking Questions!

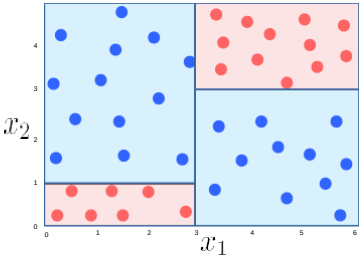
Given $x_1 < 3$, is feature 2 (x_2) greater than 1?



What We Learned

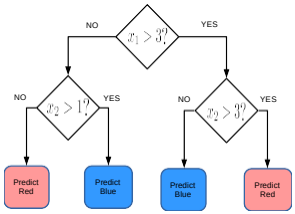
- A Decision Tree (DT) consisting of a set of rules **learned** from training data

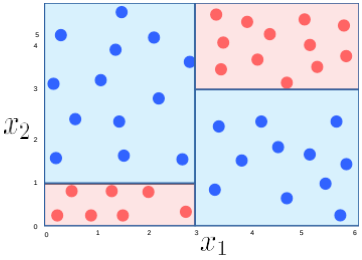




What We Learned

- A Decision Tree (DT) consisting of a set of rules **learned** from training data

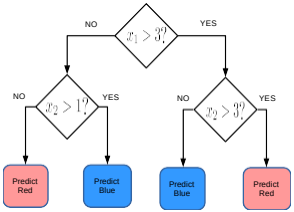


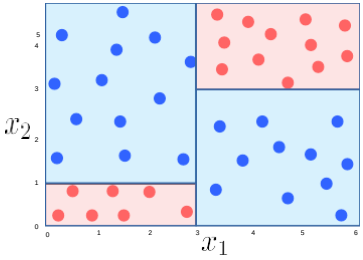


- These rules try to perform a recursive partitioning of the training data into "homogeneous" regions

What We Learned

- A Decision Tree (DT) consisting of a set of rules **learned** from training data

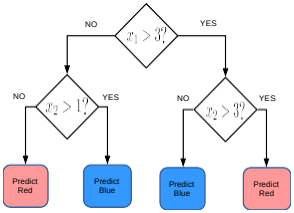


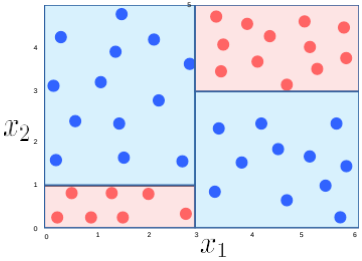


- These rules perform a recursive partitioning of the training data into "homogeneous" region. Homogeneous means that the outputs are same/similar for all inputs in that region

What We Learned

- A Decision Tree (DT) consisting of a set of rules **learned** from training data

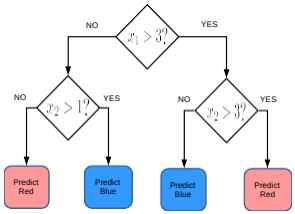


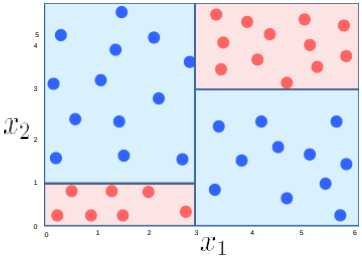


- These rules perform a recursive partitioning of the training data into"homogeneous" regions
- Homogeneous means that the outputs are same/similar for all inputs in that region
- Given a new test input, we can use the DT to predict its label

What We Learned

- A Decision Tree (DT) consisting of a set of rules **learned** from training data





- These rules perform a recursive partitioning of the training data into "homogeneous" regions
- Homogeneous means that the outputs are same/similar for all inputs in that region
- Given a new test input, we can use the DT to predict its label
- A key benefit of DT: Prediction at test time is very fast (just testing a few conditions)

Decision Tree for Classification: Another Example

- Deciding whether to play or not to play Tennis on a Saturday
  - Each input (a Saturday) has 4 categorical features: Outlook, Temp., Humidity, Wind
  - A binary classification problem (play vs no-play)

Example credit: TomMitchell

Decision Tree for Classification: Another Example

- Deciding whether to play or not to play Tennis on a Saturday
  - Each input (a Saturday) has 4 categorical features: Outlook, Temp., Humidity,
  - Wind A binary classification problem (play vs no-play)
  - Left: Training data, Right: A decision tree constructed using this data

Example credit: TomMitchell

Decision Tree for Classification: Another Example

- Deciding whether to play or not to play Tennis on a Saturday
  - Each input (a Saturday) has 4categoricalfeatures: Outlook, Temp., Humidity, Wind
  - A binary classification problem (play vs no-play)
  - Left: Training data, Right: A decision tree constructed using this data

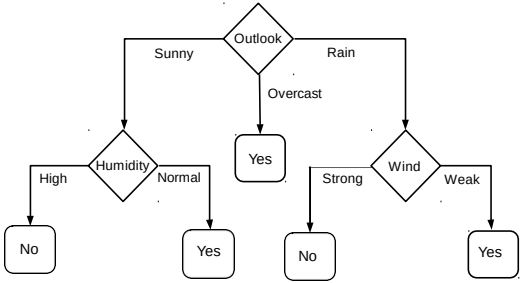
day	outlook	temperature	humidity	wind	play
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rain	mild	high	weak	yes
5	rain	cool	normal	weak	yes
6	rain	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rain	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rain	mild	high	strong	no

Example credit: TomMitchell

Decision Tree for Classification: Another Example

- Deciding whether to play or not to play Tennis on a Saturday
  - Each input (a Saturday) has 4 categorical features: Outlook, Temp., Humidity, Wind
  - A binary classification problem (play vs no-play)
  - Left: Training data, Right: A decision tree constructed using this data

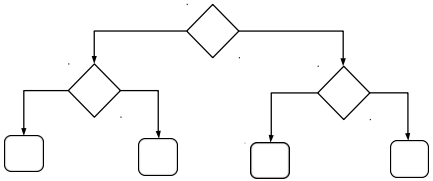
day	outlook	temperature	humidity	wind	play
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rain	mild	high	weak	yes
5	rain	cool	normal	weak	yes
6	rain	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rain	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rain	mild	high	strong	no



Example credit: TomMitchell

Some Considerations: Shape/Size of DT

- What should be the size/shape of the DT?
  - Number of internal and leaf nodes
  - Branching factor of internal nodes
  - Depth of the tree

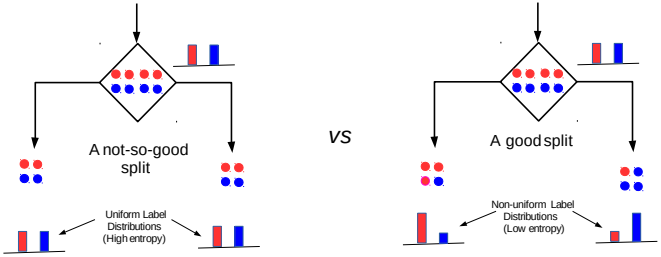


Some Considerations: Internal Nodes

- How to split at each internal node (what attribute we should choose for splitting the dataset)?

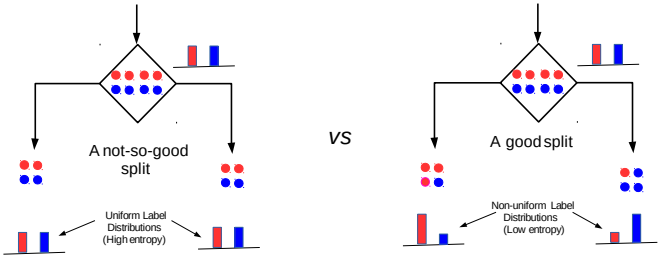
Some Considerations: Internal Nodes

- How to split at each internal node (what attribute we should choose for splitting the dataset)?
- No matter how we split, the goal should be to have splits that result in groups as "pure" as possible



Some Considerations: Internal Nodes

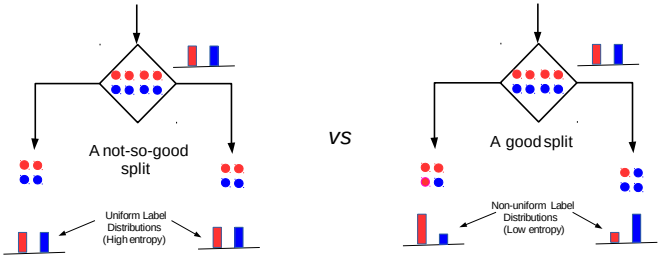
- How to split at each internal node (what attribute we should choose for splitting the dataset)?
- No matter how we split, the goal should be to have splits that result in groups as "pure" as possible



- For classification problems, entropy of the label distribution is a measure of purity

Some Considerations: Internal Nodes

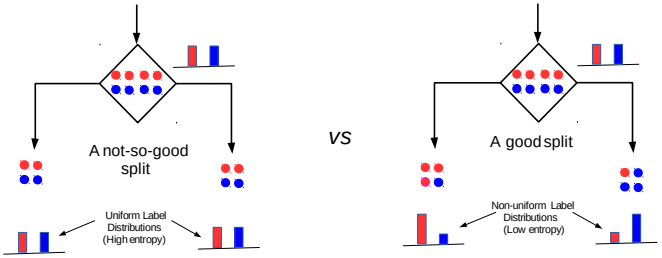
- How to split at each internal node(what attribute we should choose for splitting the dataset)?
- No matter how we split, the goal should be to have splits that result in groups as“pure”as possible



- For classification problems,entropyof the label distribution is a measure of purity
  - Low entropy ⇒ high purity (less uniform labeldistribution)

Some Considerations: Internal Nodes

- How to split at each internal node (what attribute we should choose for splitting the dataset)?
- No matter how we split, the goal should be to have splits that result in groups as "pure" as possible

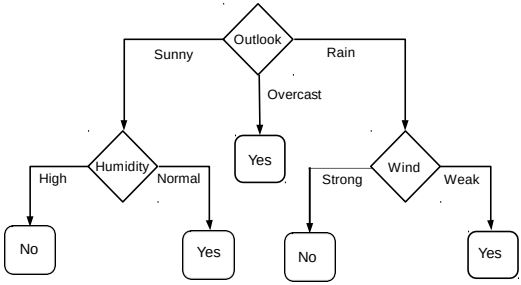


- For classification problems, entropy of the label distribution is a measure of purity
  - Low entropy ⇒ high purity (less uniform label distribution)
  - Splits that give the largest reduction (before split vs after split) in entropy are preferred (this reduction is also known as "information gain")

Decision Tree Construction

- As an illustration, let's look at one way of constructing a decision tree for some given data
- We will use the entropy/information-gain based splitting criterion for this illustration

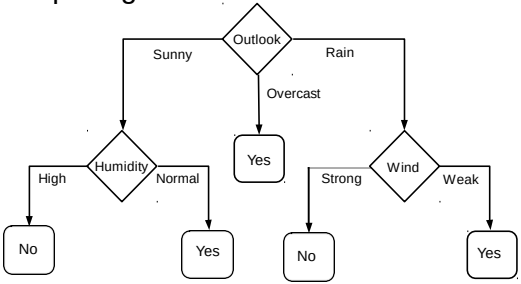
day	outlook	temperature	humidity	wind	play
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rain	mild	high	weak	yes
5	rain	cool	normal	weak	yes
6	rain	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rain	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rain	mild	high	strong	no



Decision Tree Construction

- As an illustration, let's look at one way of constructing a decision tree for some given data
- We will use the entropy/information-gain based splitting criterion for this illustration

day	outlook	temperature	humidity	wind	play
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rain	mild	high	weak	yes
5	rain	cool	normal	weak	yes
6	rain	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rain	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rain	mild	high	strong	no

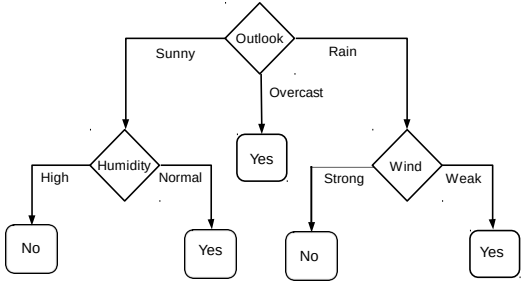


- Question: Why does it make more sense to test the feature "outlook" first?

Decision Tree Construction

- As an illustration, let's look at one way of constructing a decision tree for some given data
- We will use the entropy/information-gain based splitting criterion for this illustration

day	outlook	temperature	humidity	wind	play
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rain	mild	high	weak	yes
5	rain	cool	normal	weak	yes
6	rain	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rain	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rain	mild	high	strong	no

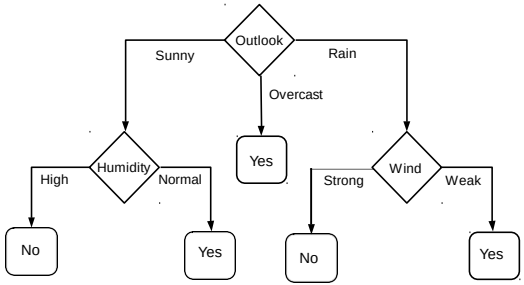


- Question:Why does it make more sense to test the feature "outlook" first?
- Answer:Of all the 4 features, it's most informative (highest information gain as the root node)

Decision Tree Construction

- As an illustration, let's look at one way of constructing a decision tree for some given data
- We will use the entropy/information-gain based splitting criterion for this illustration

day	outlook	temperature	humidity	wind	play
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rain	mild	high	weak	yes
5	rain	cool	normal	weak	yes
6	rain	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rain	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rain	mild	high	strong	no



- Question:Why does it make more sense to test the feature "outlook" first?
- Answer:Of all the 4 features, it's most informative (highestinformation gainas the root node)
- **Analogy:** Playing the game 20 Questions (the most useful questions first)

Entropy and Information Gain

- Consider a set S of inputs with a total C classes, p_c = fraction of inputs from class/label c

Entropy and Information Gain

- Consider a set S of inputs with a total C classes, p_c = fraction of inputs from class/label c
- Entropy of the set S: $H(S) - \sum_{c \in C} p_c \log_2 p_c$

Entropy and Information Gain

- Consider a set S of inputs with a total C classes, p_c = fraction of inputs from class/label c
- Entropy of the set S: $H(S) = -\sum_{c \in C} p_c \log_2 p_c$
- The difference in the entropy before and after the split is called information gain (IG)

Entropy and Information Gain

- Consider a set S of inputs with a total C classes, p_c = fraction of inputs from class/label c
- Entropy of the set S: $H(S) = -\sum_{c \in C} p_c \log_2 p_c$
- The difference in the entropy before and after the split is calledinformation gain (IG)
- For one group S being split into two smaller groups S_1 and S_2, we can calculate the IG as follows

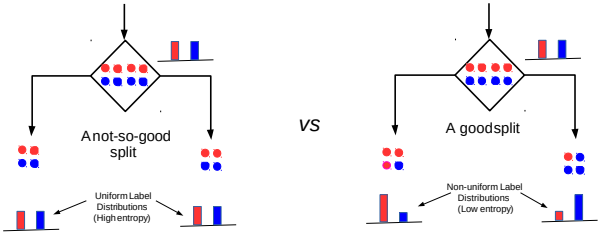
$$IG = H(S) - \frac{|S_1|}{|S|} H(S_1) - \frac{|S_2|}{|S|} H(S_2)$$

Entropy and Information Gain

- Consider a set S of inputs with a total C classes, p_c = fraction of inputs from class/label c
- Entropy of the set S: $H(S) = -\sum_{c \in C} p_c \log_2 p_c$
- The difference in the entropy before and after the split is called information gain (IG)
- For one group S being split into two smaller groups S_1 and S_2, we can calculate the IG as follows

$$IG = H(S) - \frac{|S_1|}{|S|} H(S_1) - \frac{|S_2|}{|S|} H(S_2)$$

- For DT construction, entropy/IG gives us a criterion to select the best split for an internal node



Play prediction training dataset

day	outlook	temperature	humidity	wind	play
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rain	mild	high	weak	yes
5	rain	cool	normal	weak	yes
6	rain	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rain	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rain	mild	high	strong	no

DT Construction using IG Criterion

- Let's look at IG based DT construction for the Tennis example
- Let's begin with the root node of the DT and compute *IG* of each feature
- Consider feature "wind" $\in$*{*weak,strong*}* and its *IG* at root

day	outlook	temperature	humidity	wind	play
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rain	mild	high	weak	yes
5	rain	cool	normal	weak	yes
6	rain	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rain	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rain	mild	high	strong	no

DT Construction using IG

Criterion

- Let's look at IG based DT construction for the Tennis example
- Let's begin with the root node of the DT and compute *IG* of each feature
- Consider feature "wind" $\in$ {weak,strong} and its *IG* at root

- Root node: S = [9+, 5−] (all training data: 9 play, 5 no-play)
- Entropy: $H(S) = -(9/14)\log_2(9/14) - (5/14)\log_2(5/14) = 0.94$

day	outlook	temperature	humidity	wind	play
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rain	mild	high	weak	yes
5	rain	cool	normal	weak	yes
6	rain	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rain	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rain	mild	high	strong	no

DT Construction using IG Criterion

- Let's look at IG based DT construction for the Tennis example
- Let's begin with the root node of the DT and compute *IG* of each feature
- Consider feature "wind" $\in$ {weak,strong} and its *IG* at root

day	outlook	temperature	humidity	wind	play
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rain	mild	high	weak	yes
5	rain	cool	normal	weak	yes
6	rain	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rain	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rain	mild	high	strong	no

- Root node: S = [9+, 5−] (all training data: 9 play, 5 no-play)
- Entropy: $H(S) = -(9/14)\log_2(9/14) - (5/14)\log_2(5/14) = 0.6428$

  * (-.6374) * 0.3571*(-1.485) = 0.4097 + 0.5305 = 0.94
- S_{weak} = [6+, 2−] $\Rightarrow H(S_{weak}) = 0.811$, S_{strong} = [3+, 3−] $\Rightarrow H(S_{strong}) = 1$

DT Construction using IG Criterion

day	outlook	temperature	humidity	wind	play
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rain	mild	high	weak	yes
5	rain	cool	normal	weak	yes
6	rain	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rain	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rain	mild	high	strong	no

- Let's look at IG based DT construction for the Tennis example
- Let's begin with the root node of the DT and compute *IG* of each feature
- Consider feature "wind" $\in$ {weak,strong} and its *IG* at root

- Root node: S = [9+, 5−] (all training data: 9 play, 5 no-play)
- Entropy: $H(S) = -(9/14)\log_2(9/14) - (5/14)\log_2(5/14) = 0.94$
- S_{weak} = [6+, 2−] $\Rightarrow H(S_{weak}) = 0.811$, S_{strong} = [3+, 3−] $\Rightarrow H(S_{strong}) = 1$

$$IG(S, \text{wind}) = H(S) - \frac{|S_{weak}|}{|S|} H(S_{weak}) - \frac{|S_{strong}|}{|S|} H(S_{strong})$$

$= 0.94 - 8/14 * 0.811 - 6/14 * 1 = 0.048$

DT Construction using IG Criterion

- Let's look at IG based DT construction for the Tennis example
- Let's begin with the root node of the DT and compute *IG* of each feature
- Consider feature "wind" $\in$ {weak,strong} and its *IG* at root

day	outlook	temperature	humidity	wind	play
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rain	mild	high	weak	yes
5	rain	cool	normal	weak	yes
6	rain	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rain	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rain	mild	high	strong	no

- Root node: S = [9+, 5−] (all training data: 9 play, 5 no-play)
- Entropy: $H(S) = -(9/14)\log_2(9/14) - (5/14)\log_2(5/14) = 0.94$
- S_{weak} = [6+, 2−] $\Rightarrow H(S_{weak}) = 0.811$, S_{strong} = [3+, 3−] $\Rightarrow H(S_{strong}) = 1$

$$IG(S, \text{wind}) = H(S) - \frac{|S_{weak}|}{|S|} H(S_{weak}) - \frac{|S_{strong}|}{|S|} H(S_{strong})$$

$$= 0.94 - 8/14 * 0.811 - 6/14 * 1 = 0.048$$

Likewise, $IG(S, \text{outlook}) = 0.246$, $IG(S, \text{humidity}) = 0.151$, $IG(S, \text{temperature}) = 0.029 \Rightarrow$ outlook chosen

DT Construction using IG Criterion: Growing the tree

- Having decided which feature to test at the root, let's grow the tree

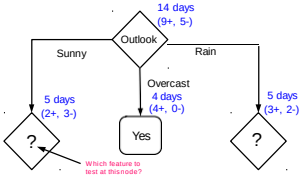
DT Construction using IG Criterion: Growing thetree

- Having decided which feature to test at the root, let's grow the tree
- How to decide which feature to test at the next level (level 2) ?

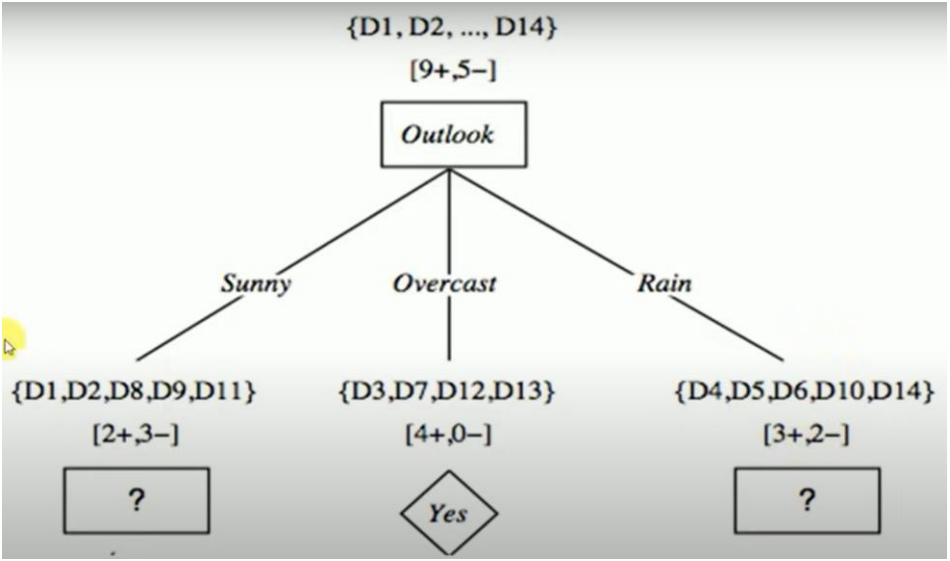
DT Construction using IG Criterion: Growing thetree

- Having decided which feature to test at the root, let's grow the tree How to
- decide which feature to test at the next level (level 2) ?
- **Rule:** Iterate - for each child node, select the feature with the highest IG

day	outlook	temperature	humidity	wind	play
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rain	mild	high	weak	yes
5	rain	cool	normal	weak	yes
6	rain	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rain	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rain	mild	high	strong	no



DT Construction using IG Criterion: Growing thetree



DT Construction using IG Criterion: Growing thetree

Full Dataset

Day	Outlook	Temp	Humidity	Wind	Play Tennis
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

Selected Part

Day	Temp	Humidity	Wind	Play Tennis
D1	Hot	High	Weak	No
D2	Hot	High	Strong	No
D8	Mild	High	Weak	No
D9	Cool	Normal	Weak	Yes
D11	Mild	Normal	Strong	Yes

DT Construction using IG Criterion: Growing thetree

Selected Part

Day	Temp	Humidity	Wind	Play Tennis
D1	Hot	High	Weak	No
D2	Hot	High	Strong	No
D8	Mild	High	Weak	No
D9	Cool	Normal	Weak	Yes
D11	Mild	Normal	Strong	Yes

Attribute: Temp

$Values\ (Temp) = Hot, Mild, Cool$

$S_{Sunny} = [2+, 3-]$ $\quad Entropy(S_{Sunny}) = -\frac{2}{5}log_2\frac{2}{5} - \frac{3}{5}log_2\frac{3}{5} = 0.97$

$S_{Hot} \leftarrow [0+, 2-]$ $\quad Entropy(S_{Hot}) = 0.0$

$S_{Mild} \leftarrow [1+, 1-]$ $\quad Entropy(S_{Mild}) = 1.0$

$S_{Cool} \leftarrow [1+, 0-]$ $\quad Entropy(S_{Cool}) = 0.0$

$$Gain\ (S_{Sunny}, Temp) = Entropy(S) - \sum_{v \in \{Hot, Mild, Cool\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S_{Sunny}, Temp)$$

$$= Entropy(S) - \frac{2}{5}Entropy(S_{Hot}) - \frac{2}{5}Entropy(S_{Mild})$$

$$-\frac{1}{5}Entropy(S_{Cool})$$

$$Gain(S_{sunny}, Temp) = 0.97 - \frac{2}{5}0.0 - \frac{2}{5}1 - \frac{1}{5}0.0 = 0.570$$

DT Construction using IG Criterion: Growing thetree

Selected Part (Sunny)

Day	Temp	Humidity	Wind	Play Tennis
D1	Hot	High	Weak	No
D2	Hot	High	Strong	No
D8	Mild	High	Weak	No
D9	Cool	Normal	Weak	Yes
D11	Mild	Normal	Strong	Yes

Attribute: Humidity

$$Values\,(Humidity) = High, Normal$$

$$S_{Sunny} = [2+, 3-] \qquad Entropy(S) = -\frac{2}{5}log_2\frac{2}{5} - \frac{3}{5}log_2\frac{3}{5} = 0.97$$

$$S_{high} \leftarrow [0+, 3-] \qquad Entropy(S_{High}) = 0.0$$

$$S_{Normal} \leftarrow [2+, 0-] \qquad Entropy(S_{Normal}) = 0.0$$

$$Gain\,(S_{Sunny}, Humidity) = Entropy(S) - \sum_{v \in \{High, Normal\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S_{Sunny}, Humidity) = Entropy(S) - \frac{3}{5}Entropy(S_{High}) - \frac{2}{5}Entropy(S_{Normal})$$

$$Gain(S_{sunny}, Humidity) = 0.97 - \frac{3}{5}0.0 - \frac{2}{5}0.0 = 0.97$$

DT Construction using IG Criterion: Growing thetree

Selected Part (Sunny)

Day	Temp	Humidity	Wind	Play Tennis
D1	Hot	High	Weak	No
D2	Hot	High	Strong	No
D8	Mild	High	Weak	No
D9	Cool	Normal	Weak	Yes
D11	Mild	Normal	Strong	Yes

Attribute: Wind

$Values\,(Wind) = Strong, Weak$

$S_{Sunny} = [2+, 3-]$ $\quad Entropy(S) = -\frac{2}{5}log_2\frac{2}{5} - \frac{3}{5}log_2\frac{3}{5} = 0.97$

$S_{Strong} \leftarrow [1+, 1-]$ $\quad Entropy(S_{Strong}) = 1.0$

$S_{Weak} \leftarrow [1+, 2-]$ $\quad Entropy(S_{Weak}) = -\frac{1}{3}log_2\frac{1}{3} - \frac{2}{3}log_2\frac{2}{3} = 0.9183$

$$Gain\,(S_{Sunny}, Wind) = Entropy(S) - \sum_{v \in \{Strong, Weak\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S_{Sunny}, Wind) = Entropy(S) - \frac{2}{5}Entropy(S_{Strong}) - \frac{3}{5}Entropy(S_{Weak})$$

$$Gain(S_{sunny}, Wind) = 0.97 - \frac{2}{5}1.0 - \frac{3}{5}0.918 = 0.0192$$

DT Construction using IG Criterion: Growing thetree

Selected Part (Sunny)

Day	Temp	Humidity	Wind	Play Tennis
D1	Hot	High	Weak	No
D2	Hot	High	Strong	No
D8	Mild	High	Weak	No
D9	Cool	Normal	Weak	Yes
D11	Mild	Normal	Strong	Yes

$$Gain(S_{sunny}, Temp) = 0.570$$

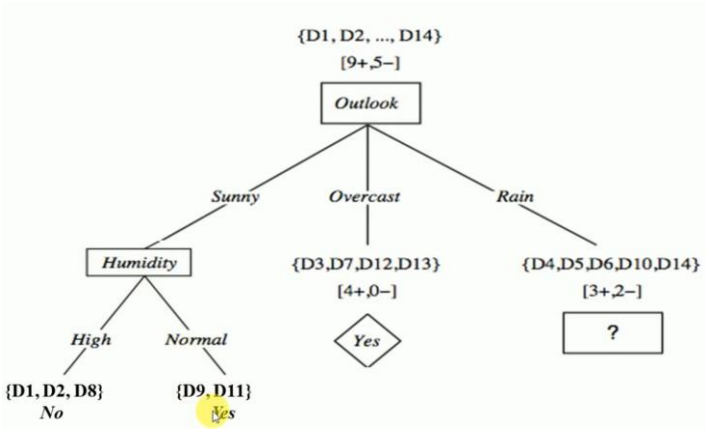
$$Gain(S_{sunny}, Humidity) = 0.97$$

$$Gain(S_{sunny}, Wind) = 0.0192$$

DT Construction using IG Criterion: Growing thetree

Selected Part (Sunny)

Day	Temp	Humidity	Wind	Play Tennis
D1	Hot	High	Weak	No
D2	Hot	High	Strong	No
D8	Mild	High	Weak	No
D9	Cool	Normal	Weak	Yes
D11	Mild	Normal	Strong	Yes



DT Construction using IG Criterion: Growing thetree

Selected Part (Rain)

Day	Temp	Humidity	Wind	Play Tennis
D4	Mild	High	Weak	Yes
D5	Cool	Normal	Weak	Yes
D6	Cool	Normal	Strong	No
D10	Mild	Normal	Weak	Yes
D14	Mild	High	Strong	No

Attribute: Temp

$Values\,(Temp) = Hot, Mild, Cool$

$S_{Rain} = [3+, 2-]$ $\quad Entropy(S_{Sunny}) = -\frac{3}{5}log_2\frac{3}{5} - \frac{2}{5}log_2\frac{2}{5} = 0.97$

$S_{Hot} \leftarrow [0+, 0-]$ $\quad Entropy(S_{Hot}) = 0.0$

$S_{Mild} \leftarrow [2+, 1-]$ $\quad Entropy(S_{Mild}) = -\frac{2}{3}log_2\frac{2}{3} - \frac{1}{3}log_2\frac{1}{3} = 0.9183$

$S_{Cool} \leftarrow [1+, 1-]$ $\quad Entropy(S_{Cool}) = 1.0$

$$Gain\,(S_{Rain}, Temp) = Entropy(S) - \sum_{v \in \{Hot, Mild, Cool\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S_{Rain}, Temp) = Entropy(S) - \frac{0}{5}Entropy(S_{Hot}) - \frac{3}{5}Entropy(S_{Mild}) - \frac{2}{5}Entropy(S_{Cool})$$

$$Gain(S_{Rain}, Temp) = 0.97 - \frac{0}{5}0.0 - \frac{3}{5}0.918 - \frac{2}{5}1.0 = 0.0192$$

DT Construction using IG Criterion: Growing thetree

Selected Part (Rain)

Day	Temp	Humidity	Wind	Play Tennis
D4	Mild	High	Weak	Yes
D5	Cool	Normal	Weak	Yes
D6	Cool	Normal	Strong	No
D10	Mild	Normal	Weak	Yes
D14	Mild	High	Strong	No

Attribute: Humidity

$Values\,(Humidity) = High, Normal$

$S_{Rain} = [3+, 2-]$ $\quad Entropy(S_{Sunny}) = -\frac{3}{5}log_2\frac{3}{5} - \frac{2}{5}log_2\frac{2}{5} = 0.97$

$S_{High} \leftarrow [1+, 1-]$ $\quad Entropy(S_{High}) = 1.0$

$S_{Normal} \leftarrow [2+, 1-]$ $\quad Entropy(S_{Normal}) = -\frac{2}{3}log_2\frac{2}{3} - \frac{1}{3}log_2\frac{1}{3} = 0.9183$

$$Gain\,(S_{Rain}, Humidity) = Entropy(S) - \sum_{v\in\{High,Normal\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S_{Rain}, Humidity) = Entropy(S) - \frac{2}{5}Entropy(S_{High}) - \frac{3}{5}Entropy(S_{Normal})$$

$$Gain(S_{Rain}, Humidity) = 0.97 - \frac{2}{5}\,1.0 - \frac{3}{5}0.918 = 0.0192$$

DT Construction using IG Criterion: Growing thetree

Selected Part (Rain)

Day	Temp	Humidity	Wind	Play Tennis
D4	Mild	High	Weak	Yes
D5	Cool	Normal	Weak	Yes
D6	Cool	Normal	Strong	No
D10	Mild	Normal	Weak	Yes
D14	Mild	High	Strong	No

Attribute: Wind

$Values\ (wind) = Strong, Weak$

$S_{Rain} = [3+, 2-]$ $\quad Entropy(S_{Sunny}) = -\frac{3}{5}log_2\frac{3}{5} - \frac{2}{5}log_2\frac{2}{5} = 0.97$

$S_{Strong} \leftarrow [0+, 2-]$ $\quad Entropy(S_{Strong}) = 0.0$

$S_{Weak} \leftarrow [3+, 0-]$ $\quad Entropy(S_{weak}) = 0.0$

$$Gain\ (S_{Rain}, Wind) = Entropy(S) - \sum_{v \in \{Strong, Weak\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S_{Rain}, Wind) = Entropy(S) - \frac{2}{5}Entropy(S_{Strong}) - \frac{3}{5}Entropy(S_{Weak})$$

$$Gain(S_{Rain}, Wind) = 0.97 - \frac{2}{5}\,0.0 - \frac{3}{5}0.0 = 0.97$$

DT Construction using IG Criterion: Growing thetree

Selected Part (Rain)

Day	Temp	Humidity	Wind	Play Tennis
D4	Mild	High	Weak	Yes
D5	Cool	Normal	Weak	Yes
D6	Cool	Normal	Strong	No
D10	Mild	Normal	Weak	Yes
D14	Mild	High	Strong	No

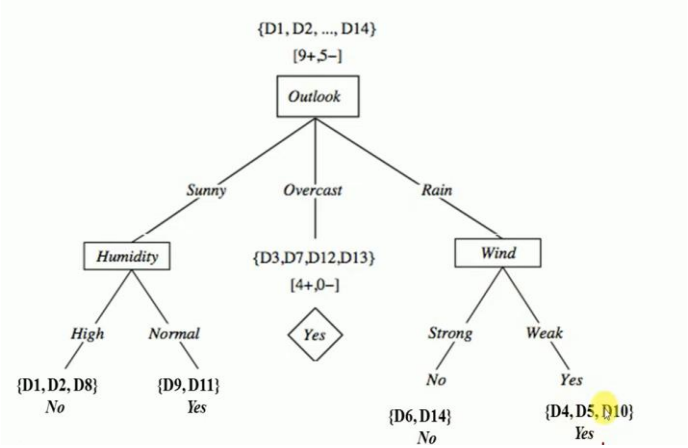
$$Gain(S_{Rain}, Temp) = 0.0192$$

$$Gain(S_{Rain}, Humidity) = 0.0192$$

$$Gain(S_{Rain}, Wind) = 0.97$$

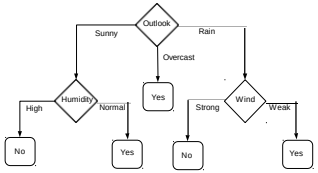
When to Stop Growing?

Day	Temp	Humidity	Wind	Play Tennis
D4	Mild	High	Weak	Yes
D5	Cool	Normal	Weak	Yes
D6	Cool	Normal	Strong	No
D10	Mild	Normal	Weak	Yes
D14	Mild	High	Strong	No



When to Stop Growing?

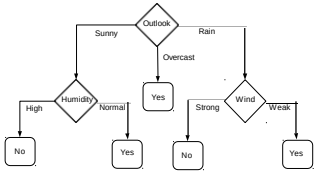
day	outlook	temperature	humidity	wind	play
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rain	mild	high	weak	yes
5	rain	cool	normal	weak	yes
6	rain	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rain	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rain	mild	high	strong	no



- Stop expanding a node further when

When to Stop Growing?

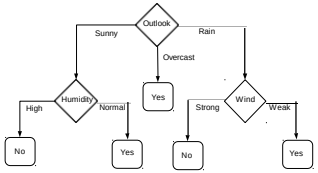
day	outlook	temperature	humidity	wind	play
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rain	mild	high	weak	yes
5	rain	cool	normal	weak	yes
6	rain	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rain	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rain	mild	high	strong	no



- Stop expanding a node further when
  - It consist of examples all having the same label (the node becomes “pure”)

When to Stop Growing?

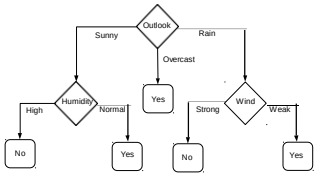
day	outlook	temperature	humidity	wind	play
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rain	mild	high	weak	yes
5	rain	cool	normal	weak	yes
6	rain	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rain	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rain	mild	high	strong	no



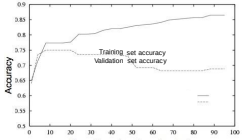
- Stop expanding a node further when
  - It consist of examples all having the same label (the node becomes “pure”)
  - We run out of features to test along the path to that node

When to Stop Growing?

day	outlook	temperature	humidity	wind	play
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rain	mild	high	weak	yes
5	rain	cool	normal	weak	yes
6	rain	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rain	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rain	mild	high	strong	no



- Stop expanding a node further when
  - It consist of examples all having the same label (the node becomes “pure”)
  - We run out of features to test along the path to that node
  - The DT starts to overfit (can be checked by monitoring the validation set accuracy)



Handling Numeric Features

Handling Numeric Features

- The examples we discussed contain categorical features. What about numerical features?
- How would you compute the class probabilities (for computing entropy) of numeric features.
    - Solution: quantize/threshold according to distinct numeric ranges.

Handling Numeric Features – example

Termperature	Play
37	No
33	No
44.5	Yes
30	Yes
15	Yes
12	No
17	Yes
22	No
20.5	Yes
24	Yes
31	Yes
27	Yes
45	Yes
34	No

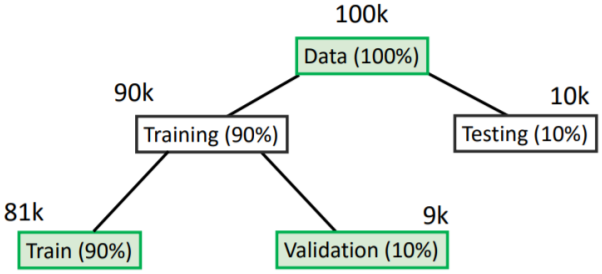
- Quantize into multiple bins according to bin size: (max-min)/N, where N is the desired number of bins; max and min are maximum and minimum features values in the training set.
- E.g. (45 – 12)/3 = 33/3 = 11
- So, the ranges will be:
  (12-23), [23-34), [34-45)

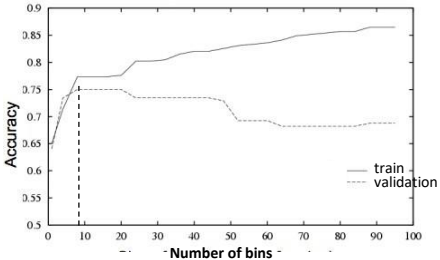
Handling Numeric Features – example

Termperature (Old)	Temperature (New)	Play
37	3	No
33	2	No
44	3	Yes
30	2	Yes
15	1	Yes
12	1	No
17	1	Yes
22	1	No
20	1	Yes
24	2	Yes
31	2	Yes
27	2	Yes
45	3	Yes
34	2	No

- Now, the converted temperature feature can be treated as a categorical feature rather than numerical.
- The test feature can also be converted in a similar way.
- The optimal number of bins can be selected by using the validation-set approach.

The validation set approach

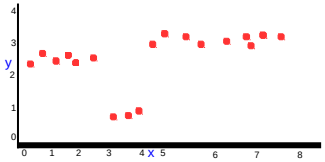




Decision trees for regression

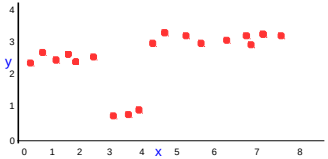
A Decision Tree for Regression

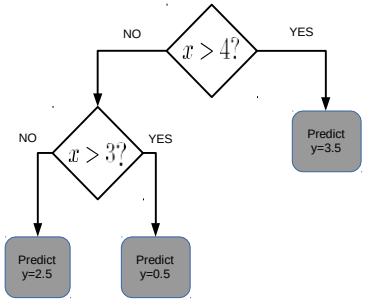
Decision Trees can also be used for regression problems



A Decision Tree for Regression

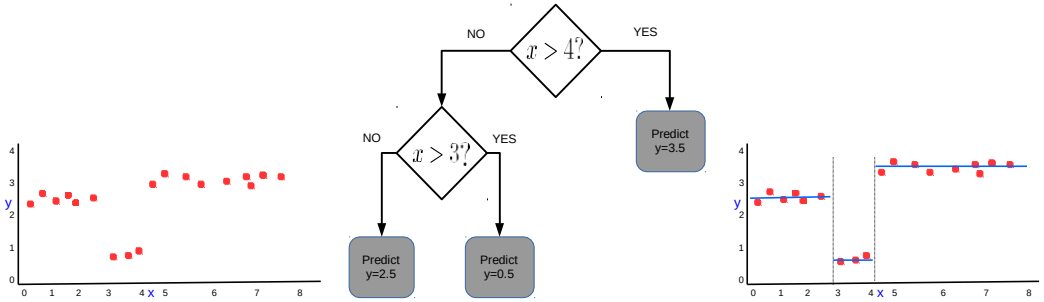
Decision Trees can also be used for regression problems





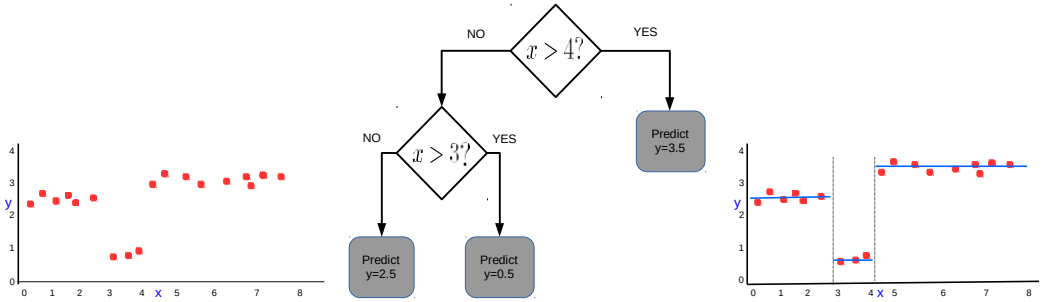
A Decision Tree for Regression

Decision Trees can also be used for regression problems



A Decision Tree for Regression

Decision Trees can also be used for regression problems



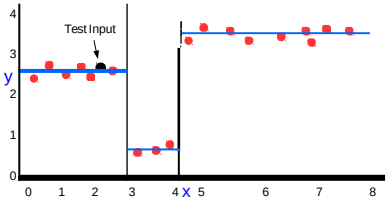
Here too, the DT partitions the training data into homogeneous regions (inputs with similar outputs)

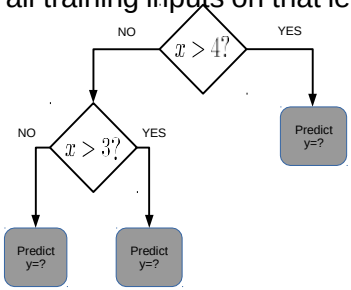
Some Considerations: Leaf Nodes

- What to do at each leaf node (the goal: make predictions)? Some options:
  - Make a constant prediction (majority/average) for every test input reaching that leaf node?
  - Use a nearest neighbors based prediction using all training inputs on that leaf node?

Some Considerations: Leaf Nodes

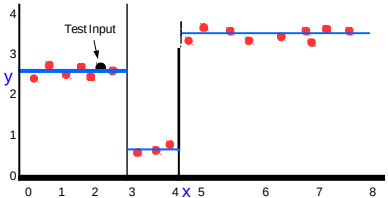
- What to do at each leaf node (the goal: make predictions)? Some options:
- Make a constant prediction (majority/average) for every test input reaching that leaf node?
- Use a nearest neighbors based prediction using all training inputs on that leaf node?

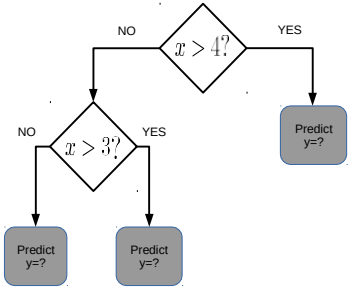




Some Considerations: Leaf Nodes

- What to do at each leaf node(the goal: make predictions)? Some options:
  - Make aconstant prediction(majority/average) for every test input reaching that leaf node?
  - Use a nearest neighbors based prediction using all training inputs on that leaf node?





- Constant prediction is the fastest at test time (and gives a piece-wise constant prediction rule)

Decision Tree: SomeComments

- Other alternatives to entropy for judging feature informativeness in DT classification?
  - Gini-index

$$\sum_{c=1}^{C} p_c(1 - p_c)$$

$p_c(1 - p_c)$ is is another popular choice

[1] Breiman, Leo; Friedman, J. H.; Olshen, R. A.; Stone, C. J. (1984). Classification and regression trees

Decision Tree: SomeComments

- Other alternatives to entropy for judging feature informativeness in DT classification?
- Gini-index $\sum_{c=1}^{C} p_c(1 - p_c)$ is another popular choice

- For DT regression (Regression Trees[1]), can split based on the variance in the outputs, instead of using entropy (which doesn't make sense for real-valued inputs)

[1] Breiman, Leo; Friedman, J. H.; Olshen, R. A.; Stone, C. J. (1984). Classification and regression trees

Decision Tree: SomeComments

- Other alternatives to entropy for judging feature informativeness in DT classification?
  - Gini-index $\sum_{c=1}^{C} p_c(1 - p_c)$ is another popular choice
- For DT regression (Regression Trees[1]), can split based on thevariancein the outputs, instead of using entropy (which doesn't make sense for real-valued inputs)
- RI-valued features (we already saw some examples) can be dealt with using thresholding
  - Need to be careful w.r.t. number of threshold points, how fine each range is,etc.

[1] Breiman, Leo; Friedman, J. H.; Olshen, R. A.; Stone, C. J. (1984). Classification and regression trees

Some Aspects about Decision Trees

Some key strengths:

- Simple and each to interpret
- Do not make any assumption about distribution of data
- Easily handle different types of features (real, categorical/nominal, etc.)
- Very fast at test time (just need to check the features, starting the root node and following the DT until you reach a leaf node)
- Multiple DTs can be combined via ensemble methods (e.g., Random Forest)
  - Each DT can be constructed using a (random) small subset of features

Some Aspects about Decision Trees

Some key strengths:

- Simple and each to interpret
- Do not make any assumption about distribution of data
- Easily handle different types of features (real, categorical/nominal, etc.)
- Very fast at test time (just need to check the features, starting the root node and following the DT until you reach a leaf node)
- Multiple DTs can be combined via ensemble methods (e.g., Random Forest)
  - Each DT can be constructed using a (random) small subset of features

Some key weaknesses:

- Overfitting is a general problem in decision trees. Can sometimes become very complex unless some *pruning* is applied

Suggested Readings

- Book: MLAA
  - Chapter 2: Topics 2.1-2.3
- Book: TM
  - Chapter 3: Topics 3.1-3.4