भारतीय प्रौद्योगिकी संस्थान जम्मू
INDIAN INSTITUTE OF TECHNOLOGY JAMMU

विद्याधनं सर्वधन प्रधानम्

Decision Tree Learning

Dr. Shaifu Gupta
shaifu.gupta@iitjammu.ac.in

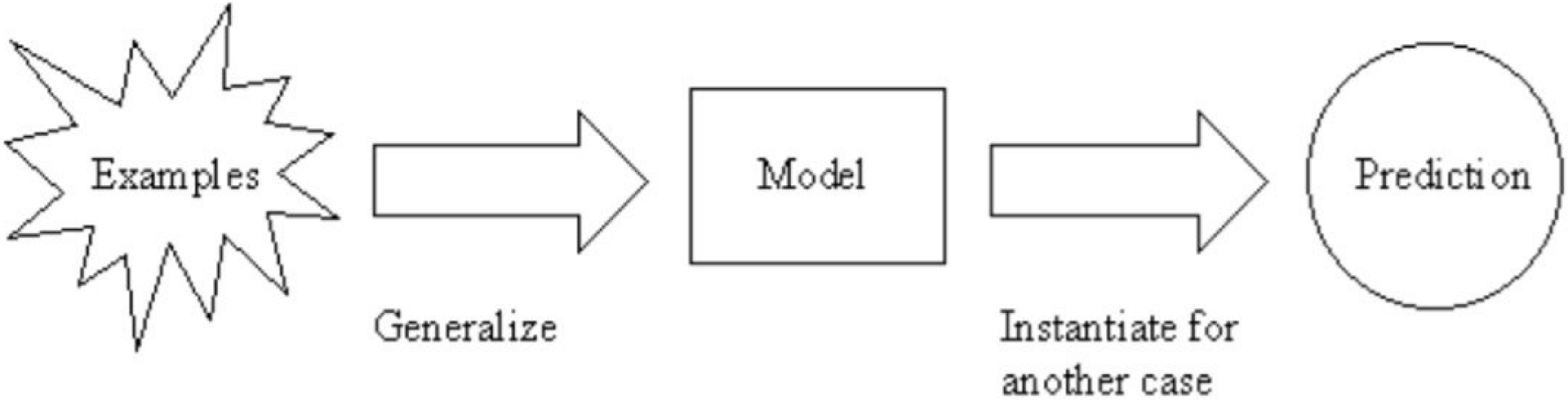
Contents

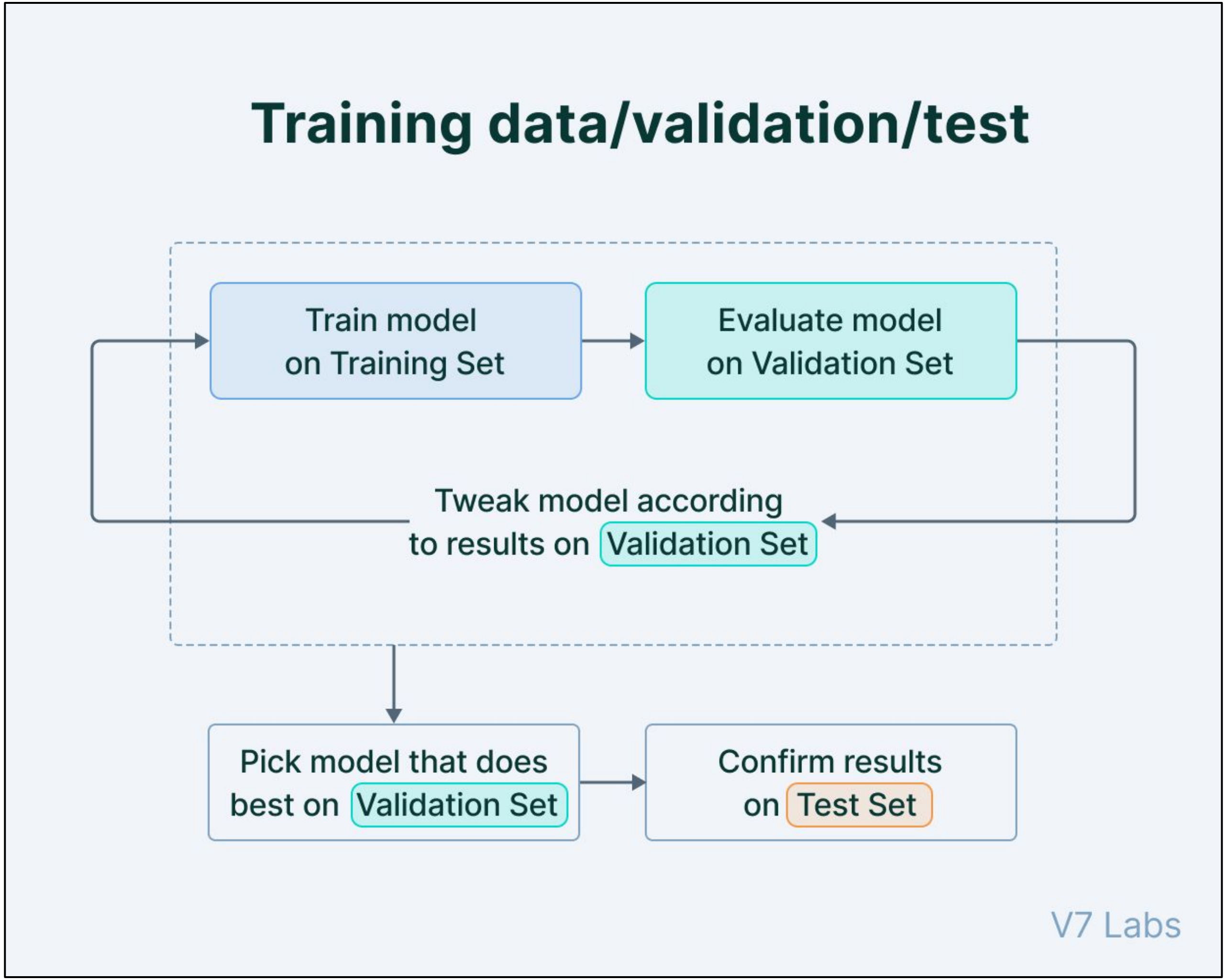
- Decision Trees concept
- Entropy, Information gain
- ID3 algorithm
- CART algorithm, Gini Impurity
- C4.5 algorithm
- Overfitting
- Methods to reduce overfitting
- Random Forest

Decision Trees

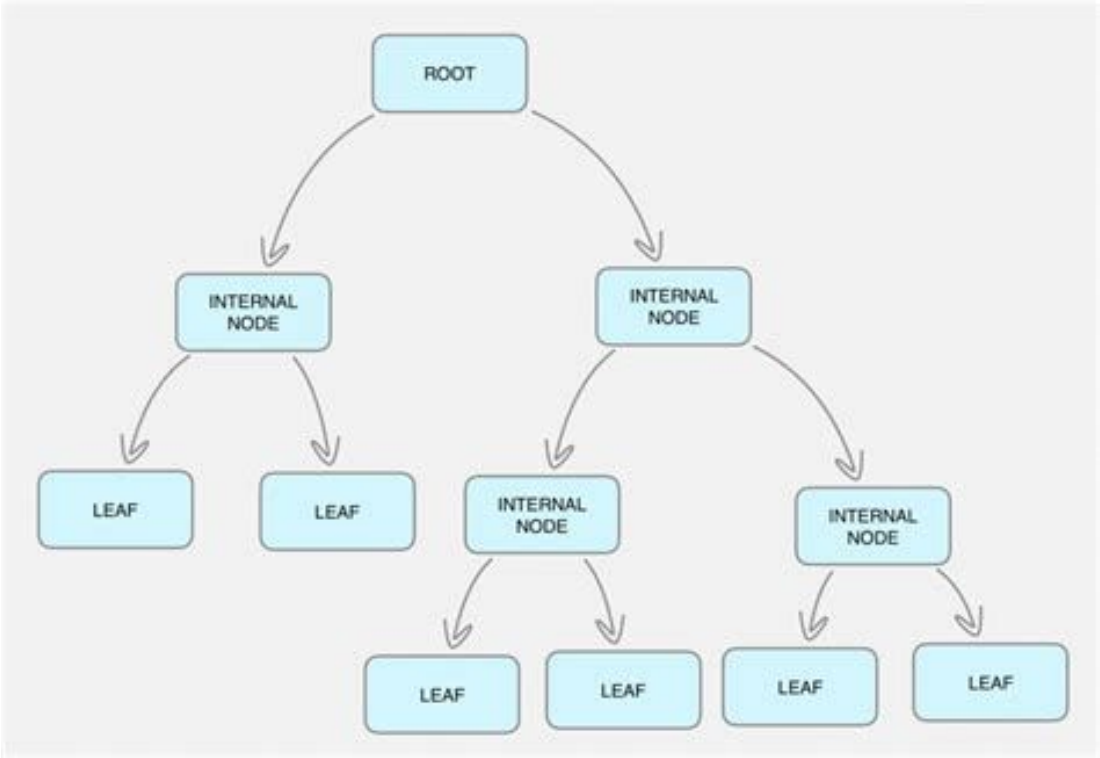
Tree in which each branch node represents a choice between a number of alternatives, and each leaf node represents a decision.

Supervised learning algorithm!

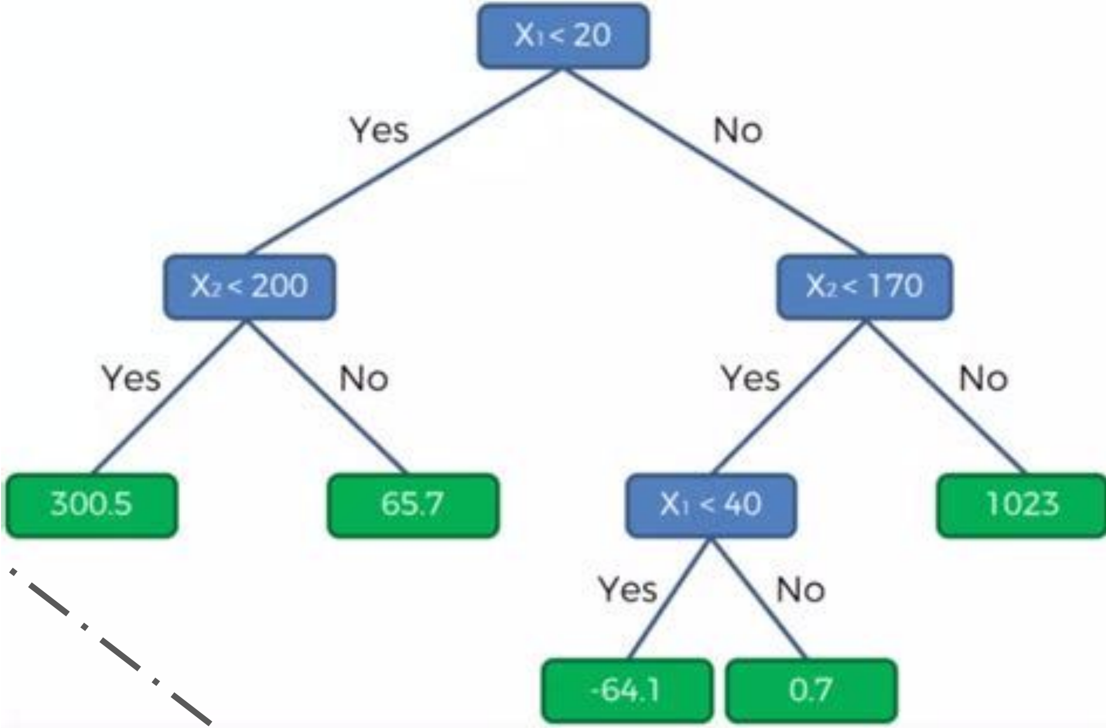


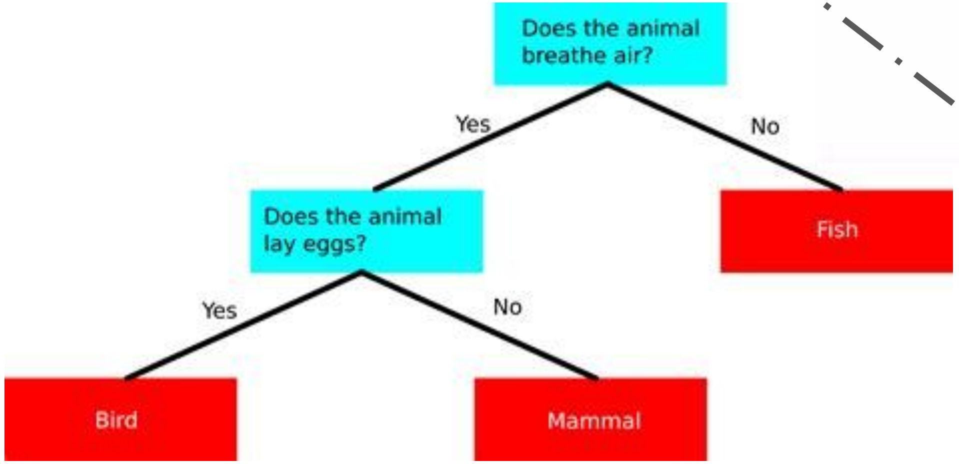

Training data/validation/test
Train model on Training Set
Evaluate model on Validation Set
Tweak model according to results on Validation Set
Pick model that does best on Validation Set
Confirm results on Test Set
V7 Labs

Decision Tree: structure

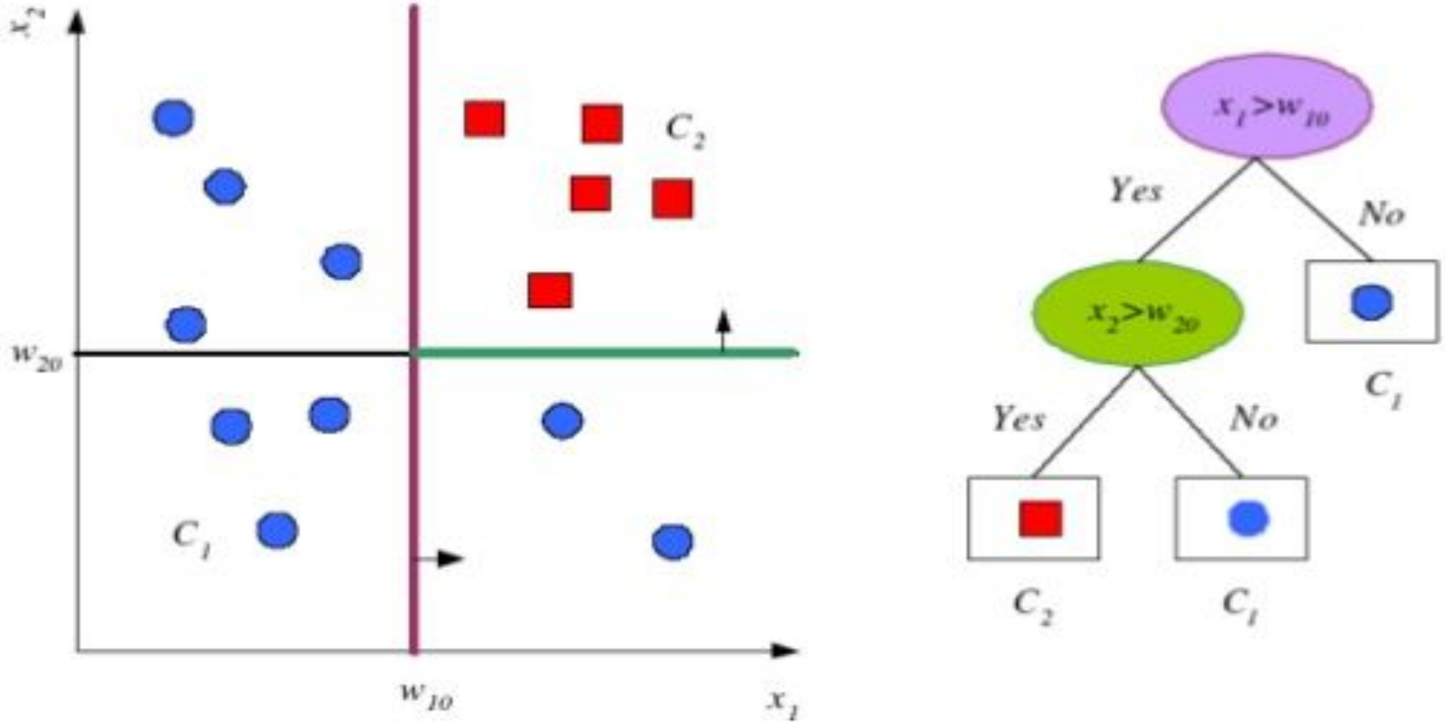


Examples

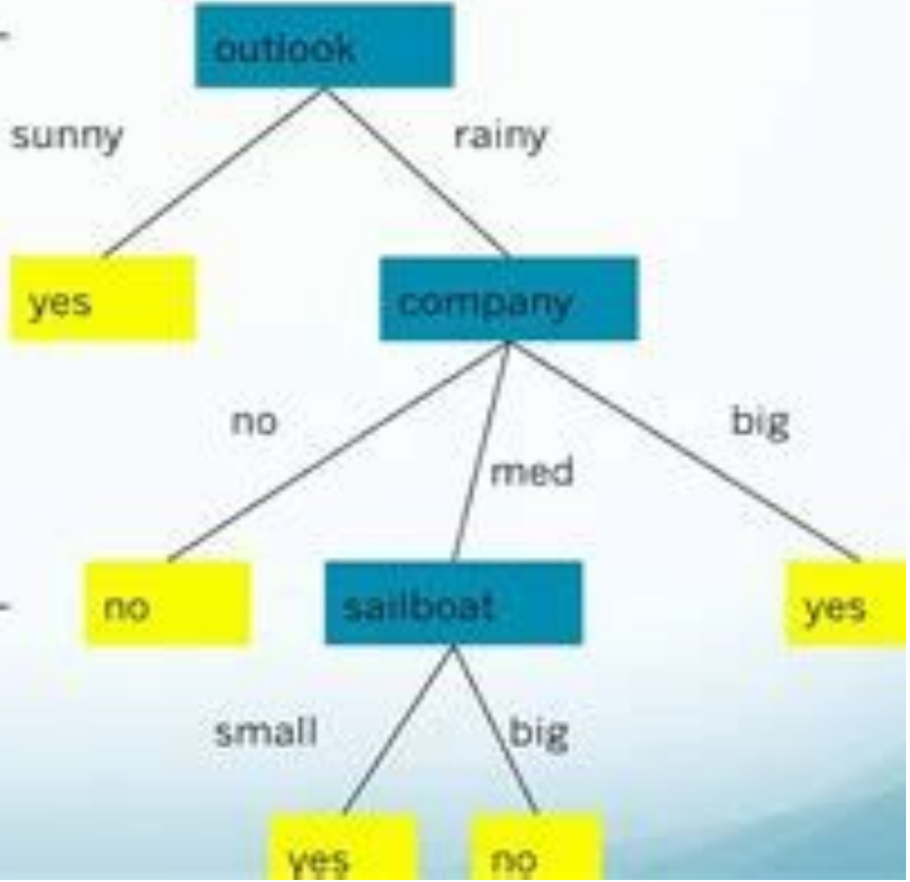

$X_1 < 20$
Yes
No
$X_2 < 200$
$X_2 < 170$
Yes
No
Yes
No
300.5
65.7
$X_1 < 40$
1023
Yes
No
-64.1
0.7


Does the animal breathe air?
Yes
No
Does the animal lay eggs?
Fish
Yes
No
Bird
Mammal

Learning classification tree



#	Attribute			Class
	Outlook	Company	Sailboat	Sail?
1	sunny	big	small	yes
2	sunny	med	small	yes
3	sunny	med	big	yes
4	sunny	no	small	yes
5	sunny	big	big	yes
6	rainy	no	small	no
7	rainy	med	small	yes
8	rainy	big	big	yes
9	rainy	no	big	no
10	rainy	med	big	no



What criteria should a decision tree algorithm use to split variables/columns?

Entropy

Used to measure uncertainty / disorder

Example:

Mixed structure

Positive (1): ⅔   [10/15]

Negative (0): ⅓ [5/15]

The more mixed (1)s and (0)s in column, higher the entropy

1
1
1
0
1
1
1
1
0
0
1
0
1
1
0

Entropy

$$-\sum_{i=1}^{c} P(x_i) log_b P(x_i)$$

$$-(10/15 \cdot \log_2(10/15) + 5/15 \cdot \log_2(5/15))$$
$$-(-.389975 + -.528308)$$
$$-(-.918278)$$
$$.918278$$

1
1
1
0
1
1
1
1
0
0
1
0
1
1
0

b=2 irrespective of number of classes

Entropy

What will the entropy for all positives or all negatives ?

- Entropy is 0 if all the members belong to the same class.
- Entropy is 1 when the collection contains an equal no. of +ve and -ve examples.
- Entropy is between 0 and 1 if collection contains unequal no. of +ve and -ve examples.

Goal: Find best attributes to split on when building a decision tree based on reduction in entropy.

Keep splitting the variables/columns until mixed target column is no longer mixed.

Information gain

- Use entropy to measure quality of split
- Compute entropies for branches, determine quality of the split by weighting entropy of each branch by how many elements it has.
- Subtract from previous entropy to measure reduction -> Information gain

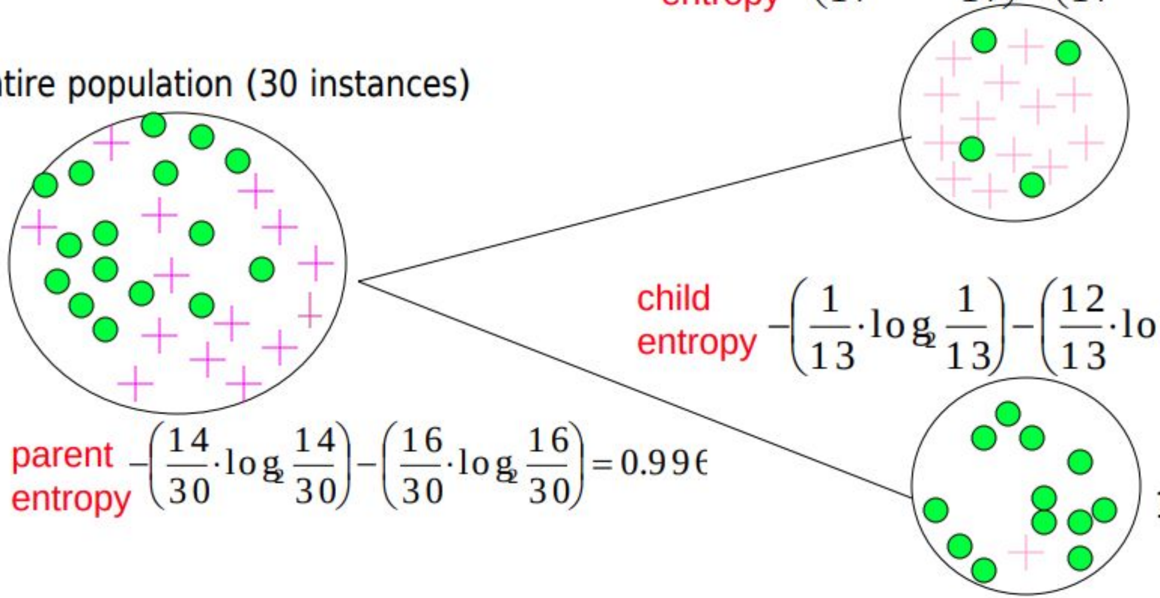
$$IG(T,A) = Entropy(T) - \sum_{v \epsilon A} \frac{|T_v|}{|T|} \cdot Entropy(T_v)$$

T = Target column, A = the variable (column) we are testing, v = each value in A

Information Gain = entropy(parent) – [average entropy(children)]

child entropy $-\left(\frac{13}{17}\cdot\log_2\frac{13}{17}\right)-\left(\frac{4}{17}\cdot\log_2\frac{4}{17}\right)=0.787$

Entire population (30 instances)



child entropy $-\left(\frac{1}{13}\cdot\log_2\frac{1}{13}\right)-\left(\frac{12}{13}\cdot\log_2\frac{12}{13}\right)=0.391$

parent entropy $-\left(\frac{14}{30}\cdot\log_2\frac{14}{30}\right)-\left(\frac{16}{30}\cdot\log_2\frac{16}{30}\right)=0.996$

(Weighted) Average Entropy of Children = $\left(\frac{17}{30}\cdot 0.787\right)+\left(\frac{13}{30}\cdot 0.391\right)=0.615$

Information Gain= 0.996 - 0.615 = 0.38 for this split

Putting it all together : ID3 algorithm

- ID3: Iterative Dichotomizer 3
- Follows greedy approach by selecting a best attribute that yields maximum Information Gain
- The steps in ID3 algorithm are as follows:
    - Calculate entropy for target (using all training examples).
    - For each attribute/feature:
        - Calculate entropy for all its categorical values.
        - Calculate information gain for the feature.
    - Find the feature with maximum information gain.
    - Repeat it until we get the desired tree.

Gini Impurity -> CART Algorithm

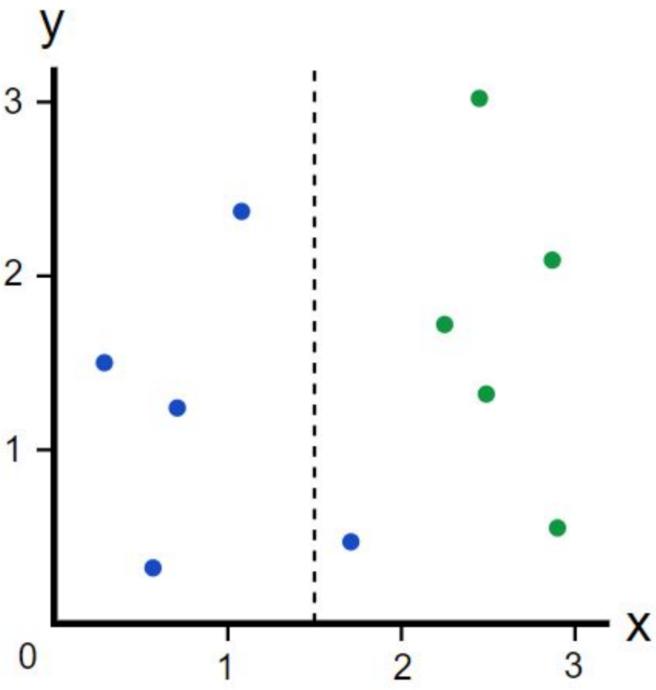
One of the other methods used in decision tree algorithms to decide optimal split from a root node, and subsequent splits.

The lower the Gini Impurity the better the split

$$G = \sum_{i=1}^{C} p(i) * (1 - p(i))$$

Where p(i) is the probability of class i.

Gini Impurity : Example



Left Branch has only blues, so G_left = 0

Right Branch has 1 blue and 5 greens, so G_right= 0.278

Quality of split obtained by weighting impurity of each branch by how many elements it has:
0.4*0 + 0.6*0.278 = 0.167

Amount of impurity “removed” with this split (Gini Gain)
0.5 - 0.167 = 0.333

Higher Gini Gain = Better Split

Gini Impurity on continuous data

Weight	Heart Disease
220	Yes
180	Yes
225	Yes
190	No
155	No

Lowest

Weight	Heart Disease
155	No
180	Yes
190	No
220	Yes
225	Yes

Highest

Gini Impurity on continuous data

Weight	Heart Disease
155	No
180	Yes
190	No
220	Yes
225	Yes

167.5

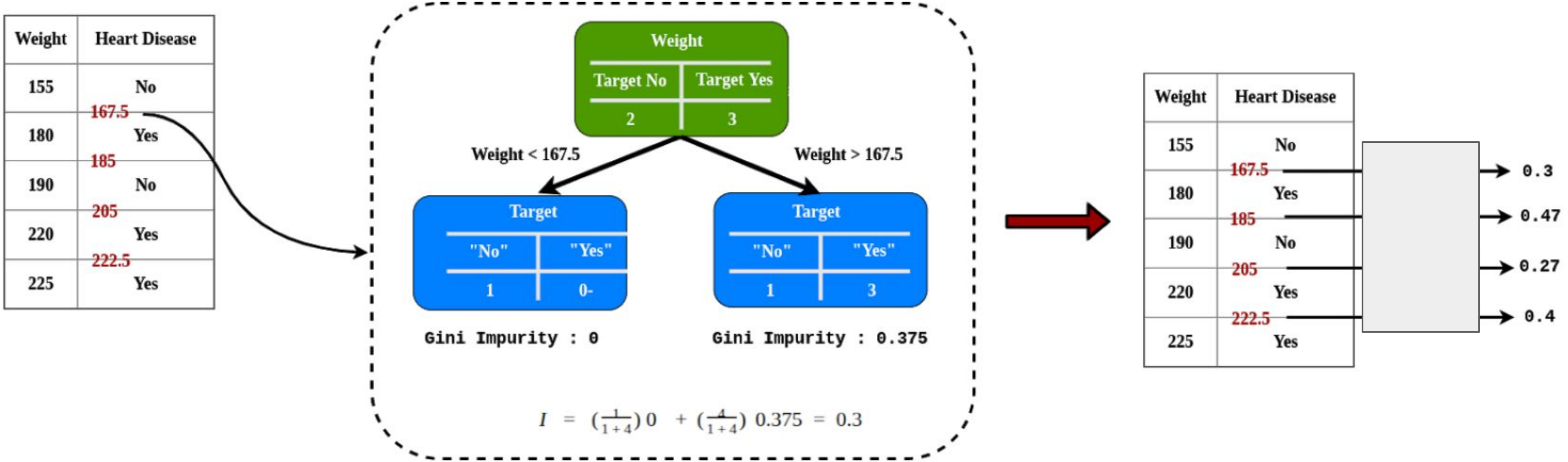
185

205

222.5

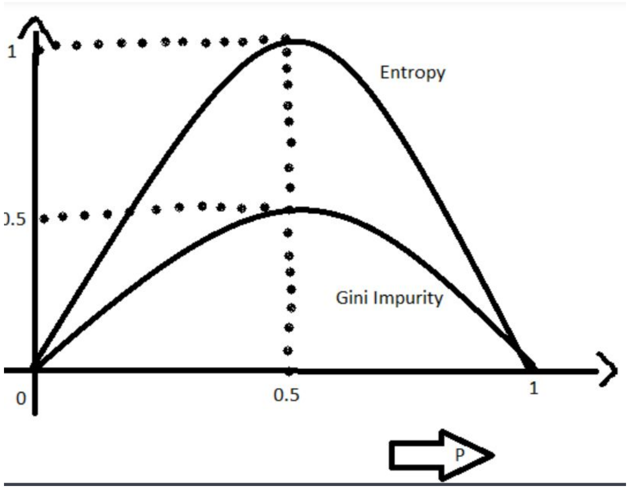
Calculate the average weight

Gini Impurity on continuous data



Gini Impurity vs Entropy

Gini Impurity is more efficient than entropy in terms of computing power
Computationally, entropy is more complex since it makes use of logarithms and consequently, the calculation of the Gini Index will be faster.



C4.5 Algorithm

- ID3 applicable for discrete datasets
- Extended to C4.5
  - Handling both continuous and discrete attributes
  - Pruning trees after creation - Reduce overfitting

- C4.5 uses Gain Ratio

$$Gain\,Ratio(S) = \frac{Gain(S)}{Split\,Info(S)}$$

Example

Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	85	85	Weak	No
2	Sunny	80	90	Strong	No
3	Overcast	83	78	Weak	Yes
4	Rain	70	96	Weak	Yes
5	Rain	68	80	Weak	Yes
6	Rain	65	70	Strong	No
7	Overcast	64	65	Strong	Yes
8	Sunny	72	95	Weak	No
9	Sunny	69	70	Weak	Yes
10	Rain	75	80	Weak	Yes
11	Sunny	75	70	Strong	Yes
12	Overcast	72	90	Strong	Yes
13	Overcast	81	75	Weak	Yes
14	Rain	71	80	Strong	No

Overfitting

Lose some generalization capability.

Overfitting happens when learning algorithm continues to develop hypotheses that reduces training set error at the cost of an increased test set error.

Causes

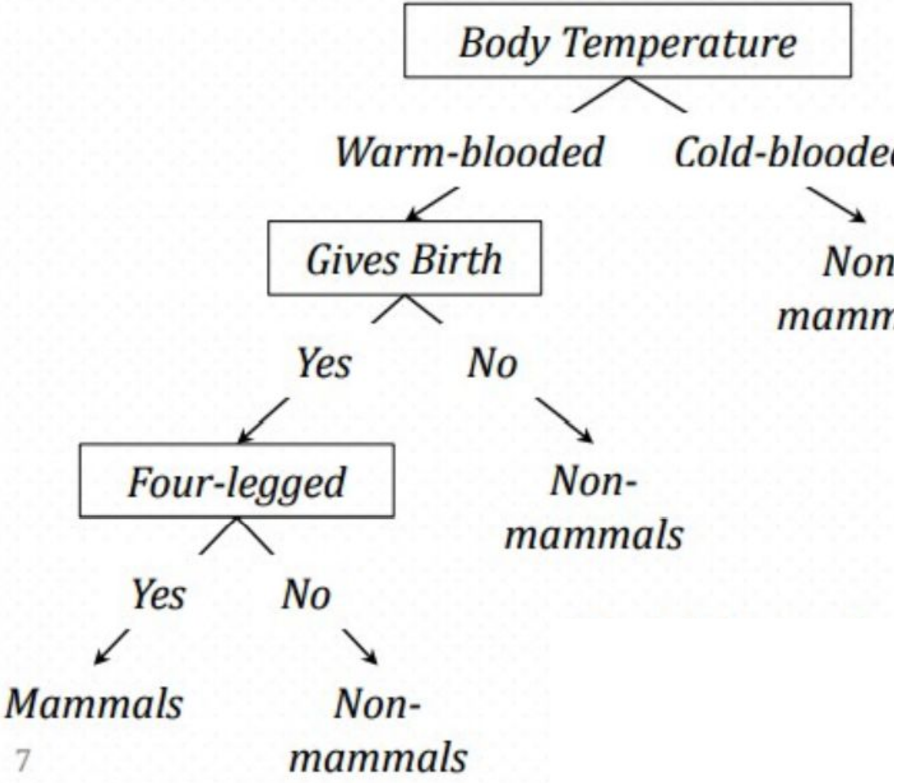
- Due to Presence of Noise
- Due to Lack of Representative Instances

Overfitting due to noise

An example training set for classifying mammals. Asterisks denote mislabelings.

Name	Body Temperature	Gives Birth	Four-legged	Hibernates	Class Label
Porcupine	Warm-blooded	Yes	Yes	Yes	***Yes***
Cat	Warm-blooded	Yes	Yes	No	***Yes***
Bat	Warm-blooded	Yes	No	Yes	***No****
Whale	Warm-blooded	Yes	No	No	***No****
Salamander	Cold-blooded	No	Yes	Yes	***No***
Komodo dragon	Cold-blooded	No	Yes	No	***No***
Python	Cold-blooded	No	No	Yes	***No***
Salmon	Cold-blooded	No	No	No	***No***
Eagle	Warm-blooded	No	No	No	***No***
Guppy	Cold-blooded	Yes	No	No	***No***

Model 1



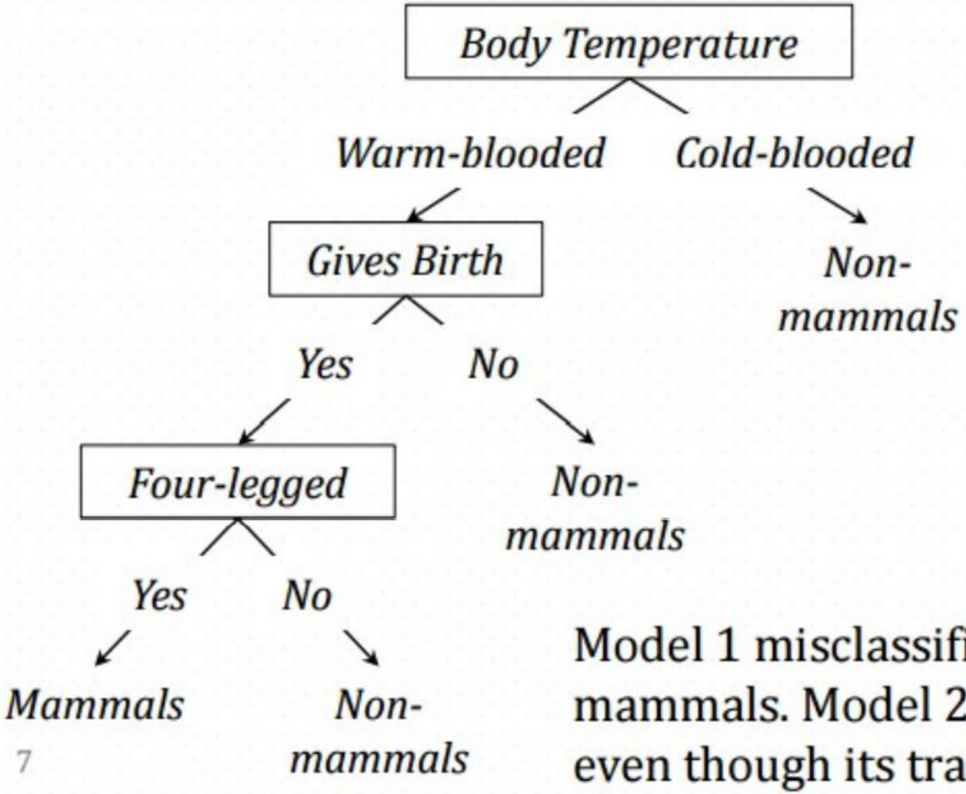
Body Temperature
Warm-blooded
Cold-bloode
Gives Birth
Non
mamm
Yes
No
Four-legged
Non-
mammals
Yes
No
Mammals
Non-
mammals

Overfitting due to noise

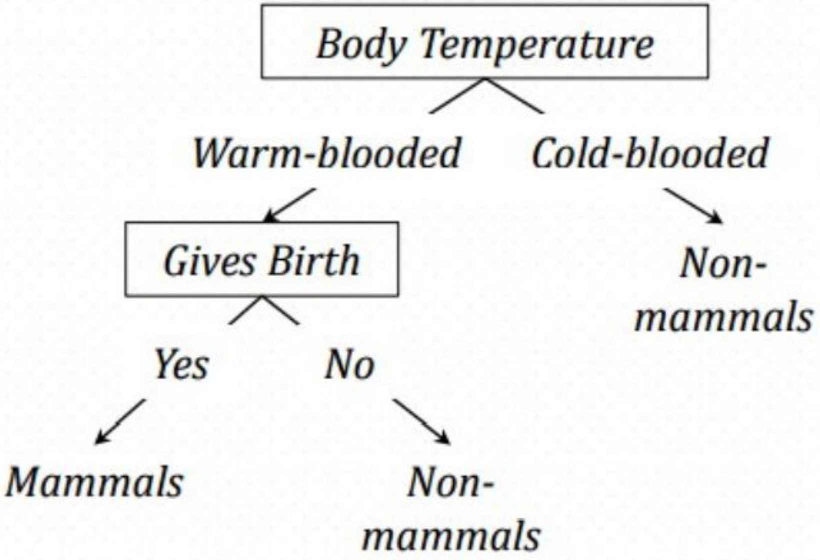
An example testing set for classifying mammals.

Name	Body Temperature	Gives Birth	Four-legged	Hibernates	Class Label
Human	Warm-blooded	Yes	No	No	***Yes***
Pigeon	Warm-blooded	No	No	No	***No***
Elephant	Warm-blooded	Yes	Yes	No	***Yes***
Leopard shark	Cold-blooded	Yes	No	No	***No***
Turtle	Cold-blooded	No	Yes	No	***No***
Penguin	Cold-blooded	No	No	No	***No***
Eel	Cold-blooded	No	No	No	***No***
Dolphin	Warm-blooded	Yes	No	No	***Yes***
Spiny anteater	Warm-blooded	No	Yes	Yes	***Yes***
Gila monster	Cold-blooded	No	Yes	Yes	***No***

Model 1



Model 2

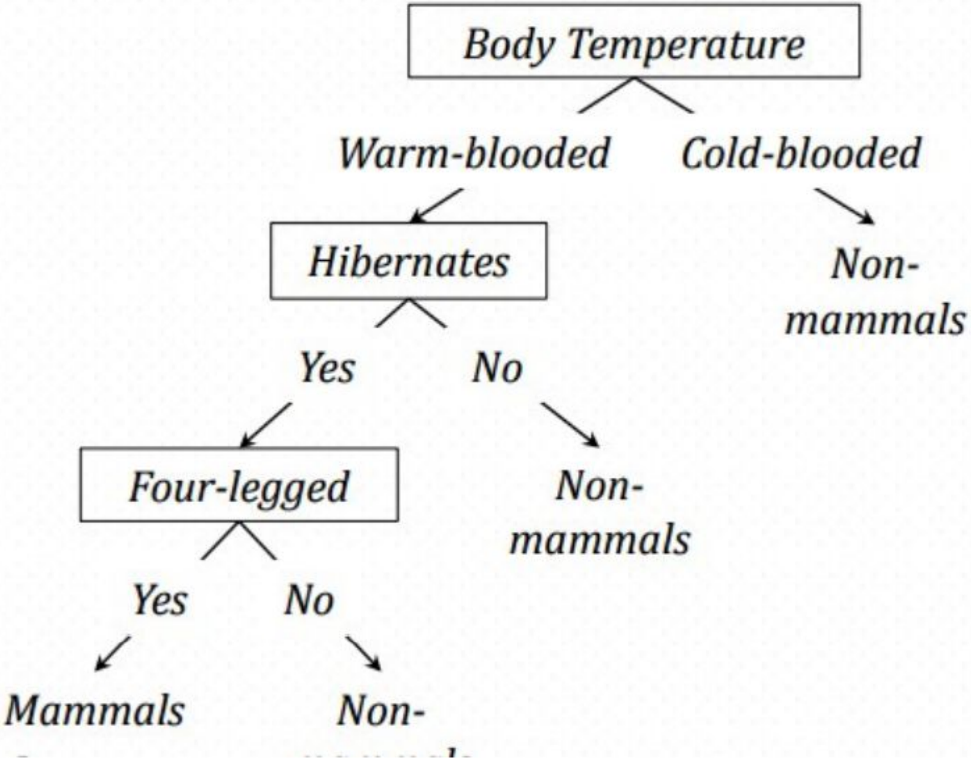


Model 1 misclassifies humans and dolphins as non-mammals. Model 2 has a lower test error rate (10%) even though its training error rate is higher (20%).

Overfitting due to lack of samples

An example training set for classifying mammals.

Name	Body Temperature	Gives Birth	Four-legged	Hibernates	Class Label
Salamander	Cold-blooded	No	Yes	Yes	***No***
Guppy	Cold-blooded	Yes	No	No	***No***
Eagle	Warm-blooded	No	No	No	***No***
Poorwill	Warm-blooded	No	No	Yes	***No***
Platypus	Warm-blooded	No	Yes	Yes	***Yes***

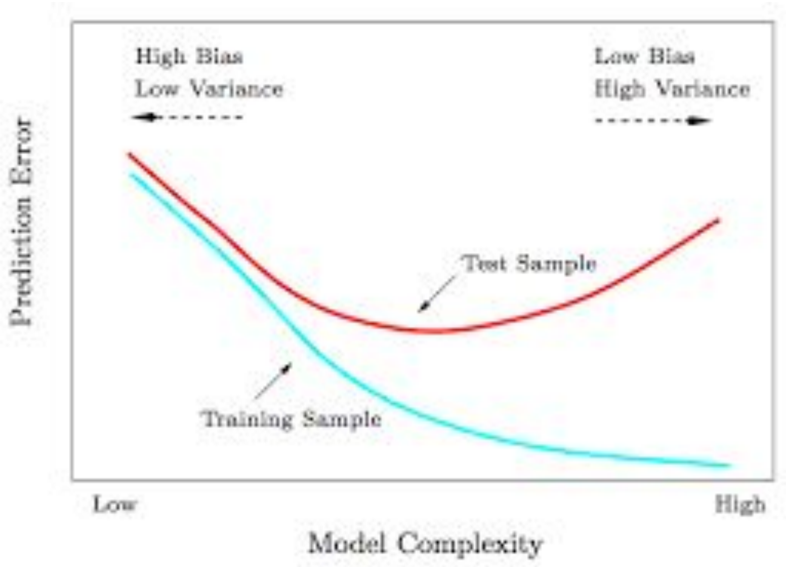


- Although the model's training error is zero, its error rate on the test set is 30%.
- Humans, elephants, and dolphins are misclassified because the decision tree classifies all warmblooded vertebrates that do not hibernate as non-mammals.

“A good model must not only fit the training data well
but also accurately classify records it has never seen.”

Identify overfitting

Relation between error and model complexity



Avoid overfitting in decision trees

Identify and removes subtrees that are likely to be due to noise

- Early stopping: stop growing tree earlier, before it reaches the point where it perfectly classifies the training data. (depth goes beyond limit, IG insufficient)
- Post-pruning: allow the tree to overfit the data, and then post-prune the tree.

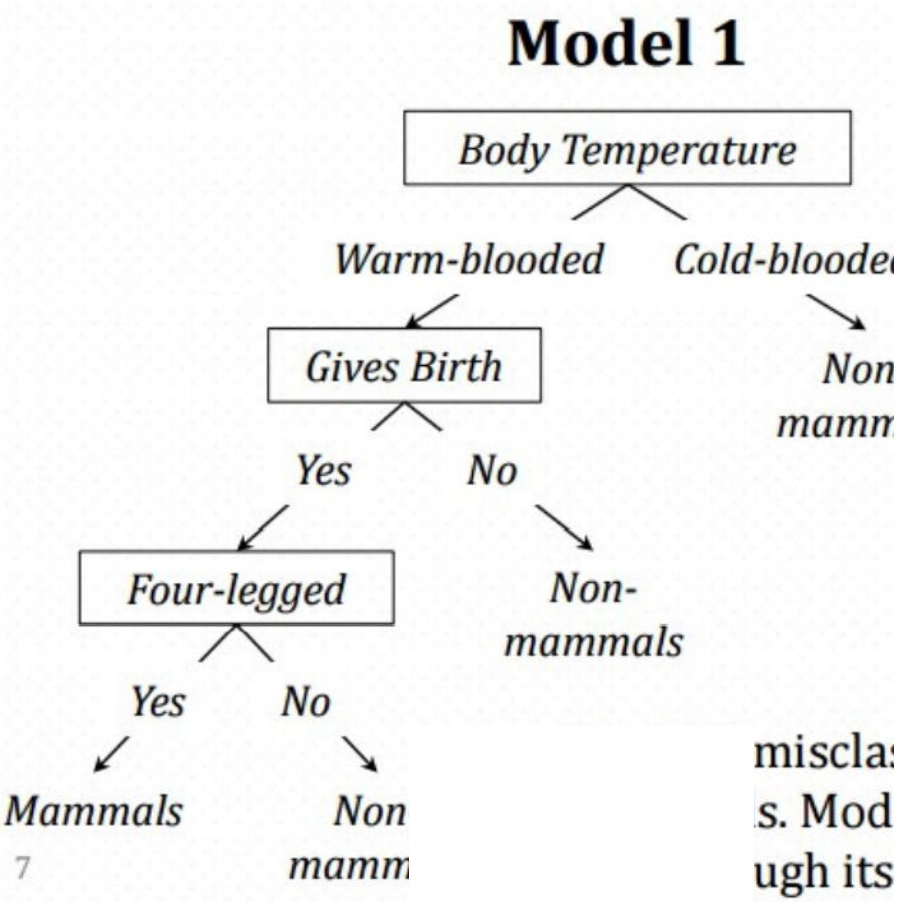
Select “best” tree:

- measure performance over training data
- measure performance over separate validation data set

Post-Pruning (Reduced error pruning)

- Consider each of the decision nodes in the tree to be candidates for pruning.
- Pruning decision node: remove subtree rooted at that node, making it a leaf node, and assign it most common classification of training examples affiliated with that node.
- Nodes are removed only if the resulting pruned tree performs no worse than the original over the validation set.
- Pruning of nodes continues until further pruning is harmful (i.e., decreases accuracy of the tree over the validation set).

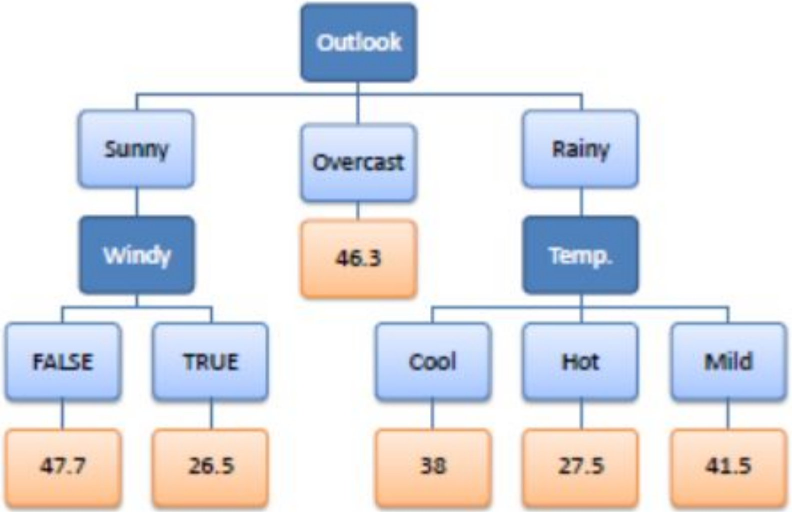
Example



Model 1
Body Temperature
Warm-blooded
Cold-bloode
Gives Birth
Non
mamm
Yes
No
Four-legged
Non-
mammals
Yes
No
Mammals
Non
mamm
7
miscla
s. Mod
ugh its

ID3 variation for regression

Outlook	Temp.	Humidity	Windy	Hours Played
Predictors				Target
Rainy	Hot	High	False	25
Rainy	Hot	High	True	30
Overcast	Hot	High	False	46
Sunny	Mild	High	False	45
Sunny	Cool	Normal	False	52
Sunny	Cool	Normal	True	23
Overcast	Cool	Normal	True	43
Rainy	Mild	High	False	35
Rainy	Cool	Normal	False	38
Sunny	Mild	Normal	False	46
Rainy	Mild	Normal	True	48
Overcast	Mild	High	True	52
Overcast	Hot	Normal	False	44
Sunny	Mild	High	True	30



Hours Played
25
30
46
45
52
23
43
35
38
46
48
52
44
30

$$Count = n = 14$$

$$Average = \bar{x} = \frac{\sum x}{n} = 39.8$$

$$Standard\ Deviation = S = \sqrt{\frac{\sum(x - \bar{x})^2}{n}} = 9.32$$

$$Coeffeicient\ of\ Variation = CV = \frac{S}{\bar{x}} * 100\% = 23\%$$

$$S(T,X)=\sum_{c\in X}P(c)S(c)$$

		Hours Played (StDev)	Count
Outlook	Overcast	3.49	4
	Rainy	7.78	5
	Sunny	10.87	5
			14

S(Hours, Outlook) = **P**(Sunny)***S**(Sunny) + **P**(Overcast)***S**(Overcast) + **P**(Rainy)***S**(Rainy)

= (4/14)*3.49 + (5/14)*7.78 + (5/14)*10.87

= 7.66

		Hours Played (StDev)
Outlook	Overcast	3.49
	Rainy	7.78
	Sunny	10.87
SDR=1.66		

		Hours Played (StDev)
Temp.	Cool	10.51
	Hot	8.95
	Mild	7.65
SDR= 0.48		

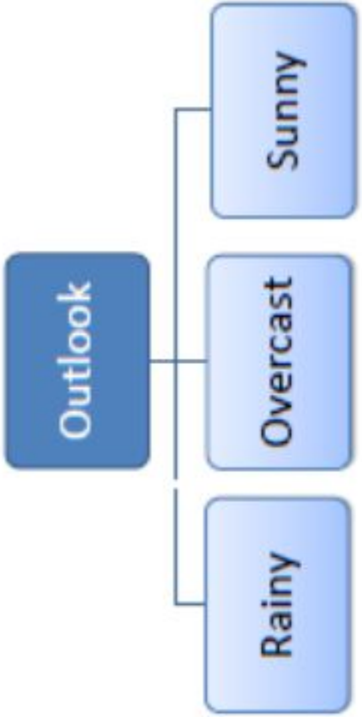
		Hours Played (StDev)
Humidity	High	9.36
	Normal	8.37
SDR=0.28		

		Hours Played (StDev)
Windy	False	7.87
	True	10.59
SDR=0.29		

$$SDR(T, X) = S(T) - S(T, X)$$

SDR(Hours , Outlook) = **S**(Hours) – **S**(Hours, Outlook)

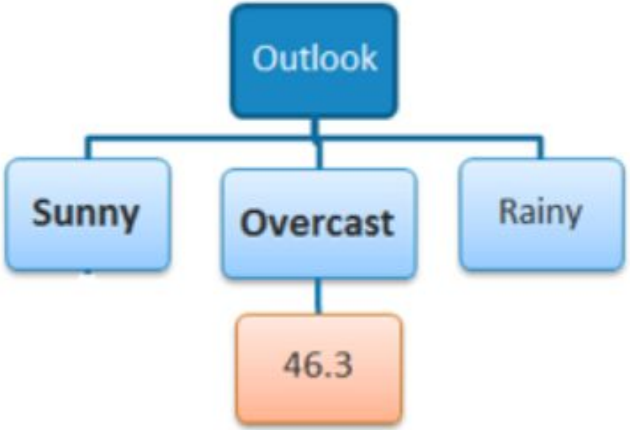
= 9.32 – 7.66 = 1.66



Outlook	Temp	Humidity	Windy	Hours Played
Sunny	Mild	High	FALSE	45
Sunny	Cool	Normal	FALSE	52
Sunny	Cool	Normal	TRUE	23
Sunny	Mild	Normal	FALSE	46
Sunny	Mild	High	TRUE	30
Overcast	Hot	High	FALSE	46
Overcast	Cool	Normal	TRUE	43
Overcast	Mild	High	TRUE	52
Overcast	Hot	Normal	FALSE	44
Rainy	Hot	High	FALSE	25
Rainy	Hot	High	TRUE	30
Rainy	Mild	High	FALSE	35
Rainy	Cool	Normal	FALSE	38
Rainy	Mild	Normal	TRUE	48

Outlook - Overcast

		Hours Played (StDev)	Hours Played (AVG)	Hours Played (CV)	Count
Outlook	Overcast	3.49	46.3	8%	4
	Rainy	7.78	35.2	22%	5
	Sunny	10.87	39.2	28%	5



Outlook - Sunny

Temp	Humidity	Windy	Hours Played
Mild	High	FALSE	45
Cool	Normal	FALSE	52
Cool	Normal	TRUE	23
Mild	Normal	FALSE	46
Mild	High	TRUE	30
			S = 10.87
			AVG = 39.2
			CV = 28%

		Hours Played (StDev)	Count
Temp	Cool	14.50	2
	Mild	7.32	3

SDR = 10.87-((2/5)*14.5 + (3/5)*7.32) = **0.678**

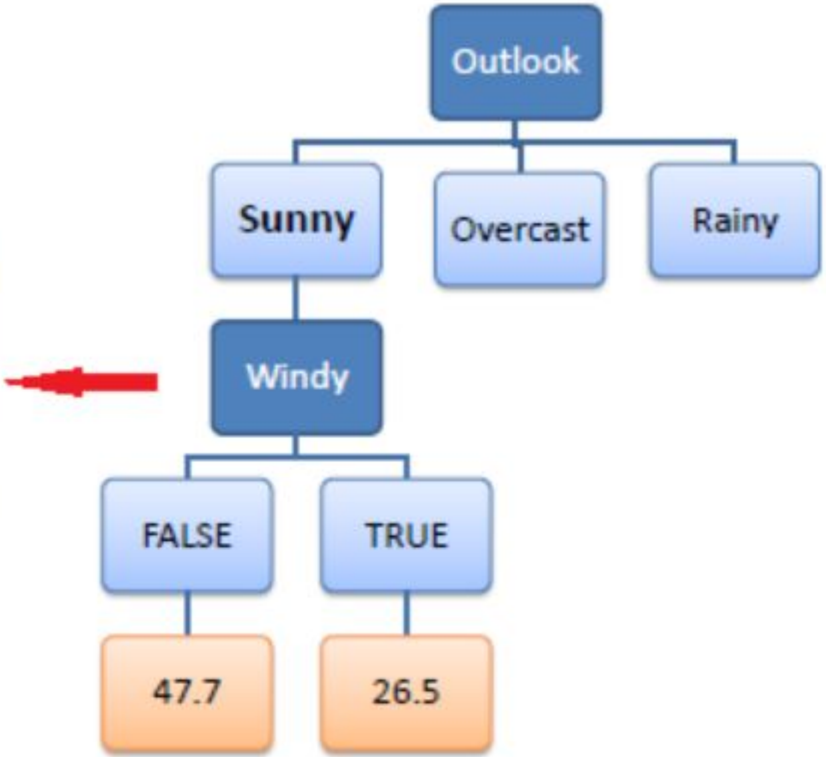
		Hours Played (StDev)	Count
Humidity	High	7.50	2
	Normal	12.50	3

SDR = 10.87-((2/5)*7.5 + (3/5)*12.5) = **0.370**

		Hours Played (StDev)	Count
Windy	False	3.09	3
	True	3.50	2

SDR = 10.87-((3/5)*3.09 + (2/5)*3.5) = **7.62**

Temp	Humidity	Windy	Hours Played
Mild	High	FALSE	45
Cool	Normal	FALSE	52
Mild	Normal	FALSE	46
Cool	Normal	TRUE	23
Mild	High	TRUE	30



Outlook - Rainy

Temp	Humidity	Windy	Hours Played
Hot	High	FALSE	25
Hot	High	TRUE	30
Mild	High	FALSE	35
Cool	Normal	FALSE	38
Mild	Normal	TRUE	48
			S = 7.78
			AVG = 35.2
			CV = 22%

		Hours Played (StDev)	Count
Temp	Cool	0	1
	Hot	2.5	2
	Mild	6.5	2

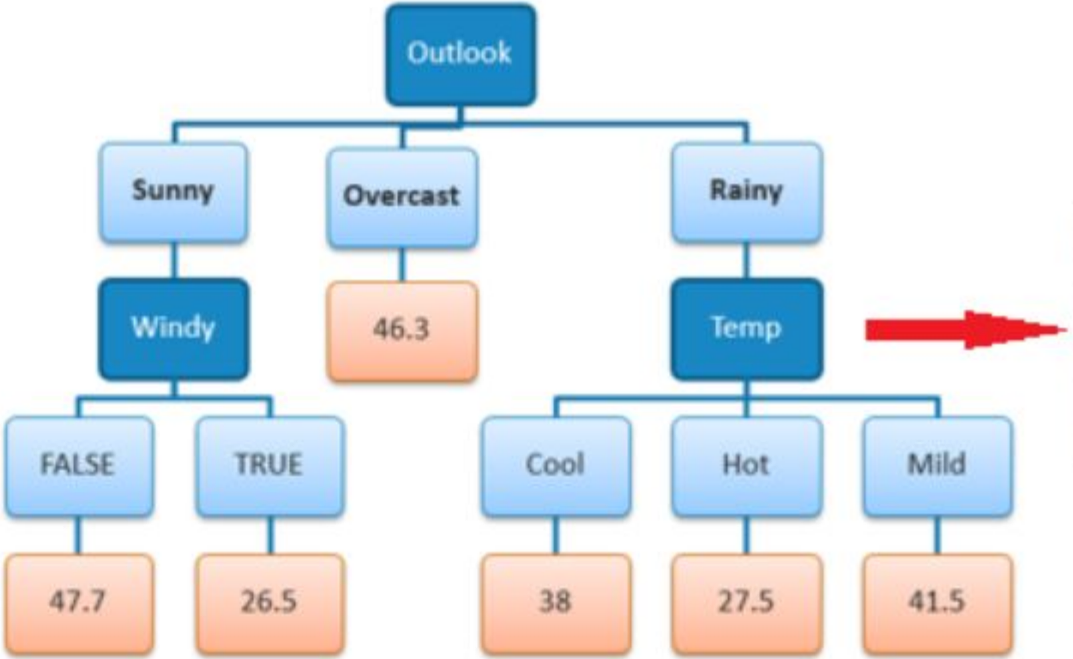
SDR = 7.78 - ((1/5)*0+(2/5)*2.5 + (2/5)*6.5) = **4.18**

		Hours Played (StDev)	Count
Humidity	High	4.1	3
	Normal	5.0	2

SDR = 7.78 - ((3/5)*4.1 + (2/5)*5.0) = **3.32**

		Hours Played (StDev)	Count
Windy	False	5.6	3
	True	9.0	2

SDR = 7.78 - ((3/5)*5.6 + (2/5)*9.0) = **0.82**



Temp	Hours Played
Cool	38
Hot	25
Hot	30
Mild	35
Mild	48

Real valued features/ attributes

Create a discrete attribute to test continuous

Temperature = 24.50C

(Temperature > 22.00C) = {true, false}

Where to set the threshold?

Temperature	15^{0}C	18^{0}C	19^{0}C	22^{0}C	24^{0}C	27^{0}C
PlayTennis	No	No	Yes	Yes	Yes	No

Random forest

- Utilizes ensemble learning (combines many classifiers) to provide solutions
- Consists of many decision trees
- Predicts by taking average (regression) or majority vote (classification) of output from various trees
- It reduces the overfitting of datasets
- Trained through bagging

Random forest vs decision tree

Main difference between decision tree algorithm and random forest algorithm is that establishing root nodes and segregating nodes is done randomly in the latter.

Random forest : Bagging

- Random forest classifier divides training dataset into subsets.
- These subsets are given to every decision tree in the random forest system.
- Each decision tree produces its specific output.

Note: not suited to classification problems with a skewed class distribution