Decision Trees

COM 214: Introduction to Artificial Intelligence

Sasha Fedchin[1,2]

[1]Department of Software Engineering
American University of Central Asia

[2]Department of Computer Science
Tufts University

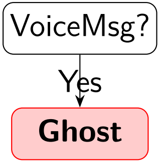
How Do You Decide Whether to Ghost a Text or Reply to It?

	VoiceMsg?	Sender	Busy?	Time	Num Messages	Reply?
1	No	Family	Yes	Day	1-2	Reply
2	Yes	Family	Yes	Night	>5	Ghost
3	No	Stranger	No	Day	1-2	Ghost
4	No	Friend	No	Day	>5	Ghost
5	No	Friend	No	Night	3-4	Ghost
6	No	Family	No	Night	>5	Reply
7	No	Friend	Yes	Day	3-4	Ghost
8	Yes	Friend	No	Day	1-2	Ghost
9	No	Friend	No	Day	1-2	Reply
10	No	Family	No	Night	3-4	Reply
11	No	Friend	No	Night	1-2	Reply
12	No	Family	Yes	Night	1-2	Ghost
13	No	Friend	No	Day	3-4	Ghost

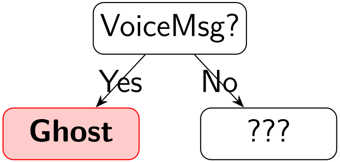
Decision Tree: an Example

VoiceMsg?

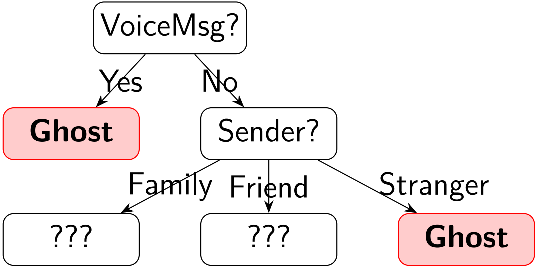
Decision Tree: an Example



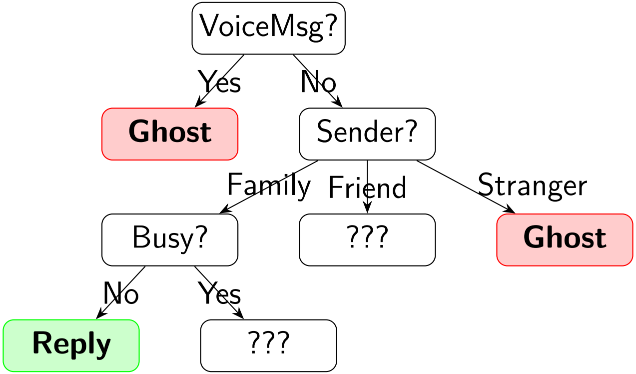
Decision Tree: an Example



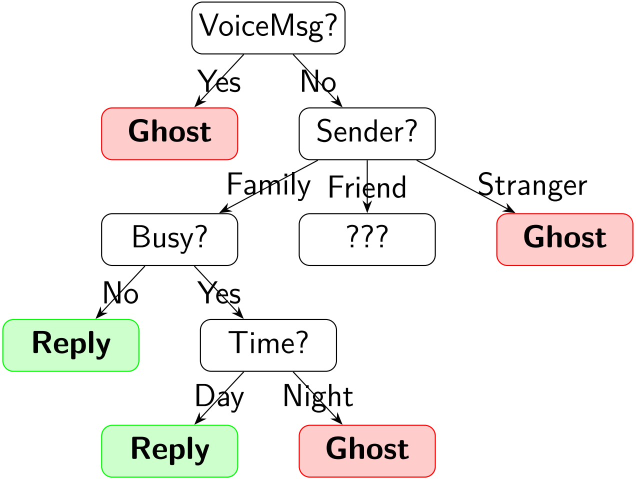
Decision Tree: an Example



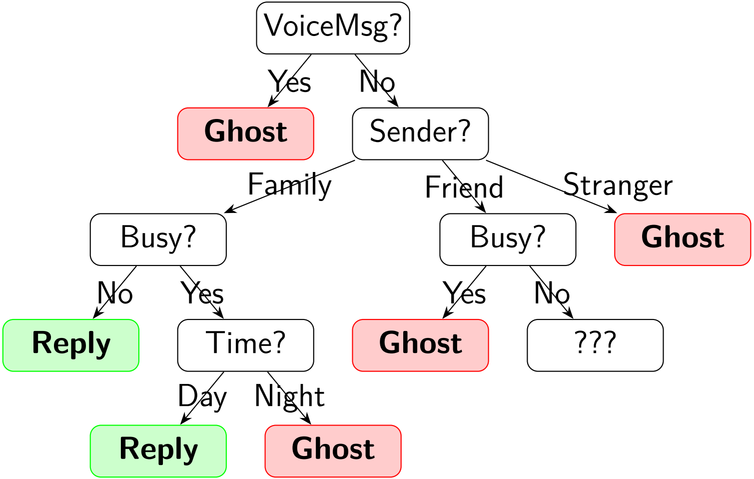
Decision Tree: an Example



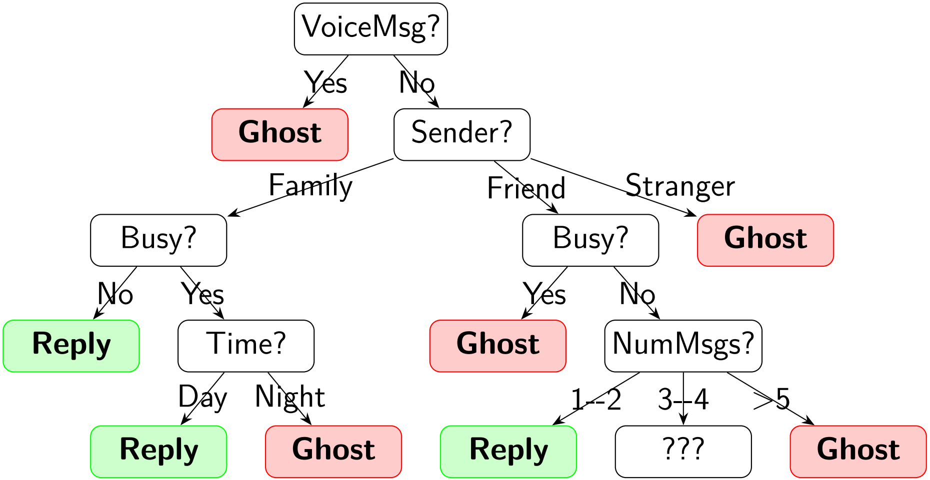
Decision Tree: an Example



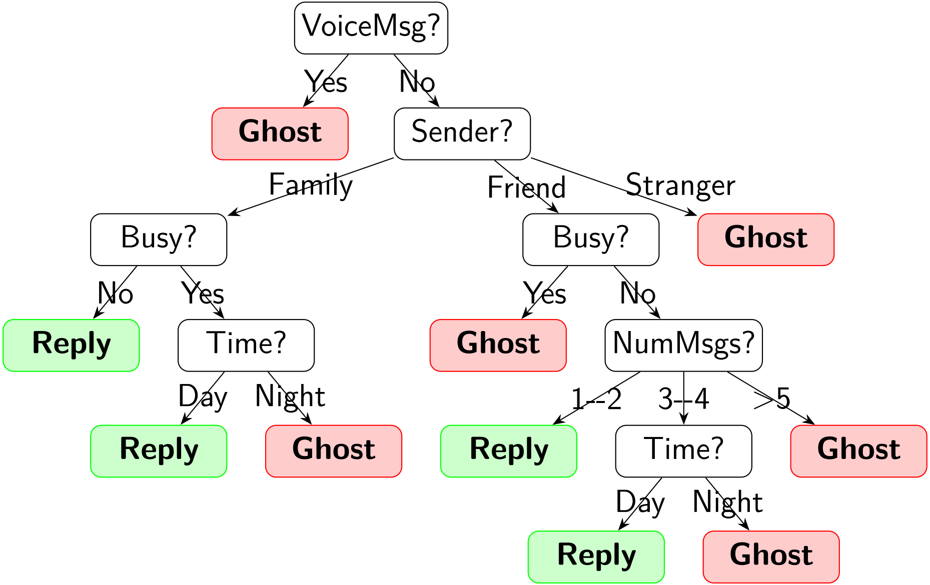
Decision Tree: an Example



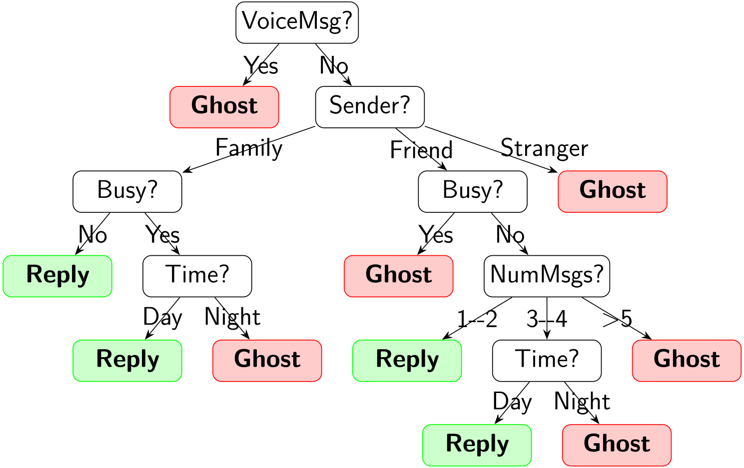
Decision Tree: an Example



Decision Tree: an Example



How Does Decision Tree Work?



- Classification happens from root to leaves.
- Each leaf nodes assign a classification (label).
- Each internal node tests an attribute.
- Edges are attribute values.

Example: [VoiceMsg?:No, Sender?:Friend, Busy?:No, NumMsgs?:3, Time?:13:04]

In-Class Exercise

- Construct a decision tree that describes how you deal with texts (very simple tree is OK, use the shape tool to draw boxes)
- How many different trees are possible? Are some trees better than others?
- Decision Trees are still very much being used in industry. Why do you think they have not been replaced with generative AI?

Expressiveness of Decision Trees

- Each decision describes a mapping from attribute values to labels (e.g., Ghost or Reply).

Expressiveness of Decision Trees

- Each decision describes a mapping from attribute values to labels (e.g., Ghost or Reply).
- The number of different trees depends on how many possible attribute value combinations there are.

Expressiveness of Decision Trees

- Each decision describes a mapping from attribute values to labels (e.g., Ghost or Reply).
- The number of different trees depends on how many possible attribute value combinations there are.
- In our example: 5 attributes.
  - VoiceMsg? (2 values)
  - Busy? (2 values)
  - Time? (2 values: day/night)
  - Sender? (3 values)
  - NumMsgs? (3 values)

Expressiveness of Decision Trees

- Each decision describes a mapping from attribute values to labels (e.g., Ghost or Reply).
- The number of different trees depends on how many possible attribute value combinations there are.
- In our example: 5 attributes.
  - VoiceMsg? (2 values)
  - Busy? (2 values)
  - Time? (2 values: day/night)
  - Sender? (3 values)
  - NumMsgs? (3 values)
- Total distinct input combinations $= 2 \times 2 \times 2 \times 3 \times 3 = 72$.

Expressiveness of Decision Trees

- Each decision describes a mapping from attribute values to labels (e.g., Ghost or Reply).
- The number of different trees depends on how many possible attribute value combinations there are.
- In our example: 5 attributes.
  - VoiceMsg? (2 values)
  - Busy? (2 values)
  - Time? (2 values: day/night)
  - Sender? (3 values)
  - NumMsgs? (3 values)
- Total distinct input combinations $= 2 \times 2 \times 2 \times 3 \times 3 = 72$.
- For each of these 72 cases we can decide ``Ghost" or ``Reply."

Expressiveness of Decision Trees

- Each decision describes a mapping from attribute values to labels (e.g., Ghost or Reply).
- The number of different trees depends on how many possible attribute value combinations there are.
- In our example: 5 attributes.
  - VoiceMsg? (2 values)
  - Busy? (2 values)
  - Time? (2 values: day/night)
  - Sender? (3 values)
  - NumMsgs? (3 values)
- Total distinct input combinations $= 2 \times 2 \times 2 \times 3 \times 3 = 72$.
- For each of these 72 cases we can decide ``Ghost" or ``Reply."
- That gives 2^{72} possible trees (more than a quadrillion).

2nd Law of Thermodynamics: Entropy Always Increases

Start of the semester:

End of the semester:

Entropy tells you how much information you need

Entropy of a Boolean variable

- S is a sample of training examples
- $p_{\oplus}$ is the proportion of positive examples in S
- $p_{\ominus}$ is the proportion of negative examples in S
- Entropy measures the uncertainty/messiness of S

$$Entropy(S) \equiv -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

Entropy Example

Suppose we have a sample $S = t, f, f, f$.

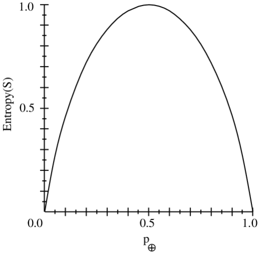
General definition of entropy

If a target value takes on many values,
$Entropy(S) \equiv \sum_{i=1}^{c} -p_i log_2(p_i)$

Entropy Example

Suppose we have a sample $S = t, f, f, f$.

$$
\begin{aligned}
Entropy(S) &= -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus} \\
&= -(1/4) \log_2 (1/4) - (3/4) \log_2 (3/4) \\
&= 0.811 bits
\end{aligned}
$$



General definition of entropy

If a target value takes on many values,
$Entropy(S) \equiv \sum_{i=1}^{c} -p_i log_2(p_i)$

Which attribute is the best classifier?

	VoiceMsg?	Sender	Busy?	Time	Num Messages	Reply?
1	No	Family	Yes	Day	1-2	Reply
2	Yes	Family	Yes	Night	>5	Ghost
3	No	Stranger	No	Day	1-2	Ghost
4	No	Friend	No	Day	>5	Ghost
5	No	Friend	No	Night	3-4	Ghost
6	No	Family	No	Night	>5	Reply
7	No	Friend	Yes	Day	3-4	Ghost
8	Yes	Friend	No	Day	1-2	Ghost
9	No	Friend	No	Day	1-2	Reply
10	No	Family	No	Night	3-4	Reply
11	No	Friend	No	Night	1-2	Reply
12	No	Family	Yes	Night	1-2	Ghost
13	No	Friend	No	Day	3-4	Ghost

Information Gain

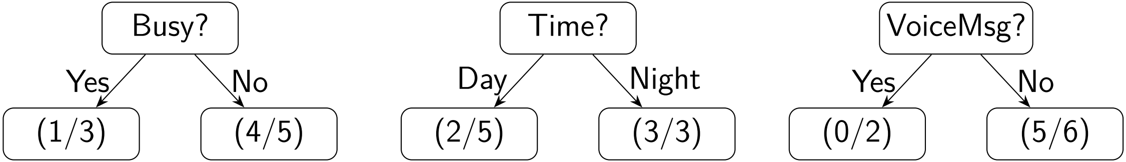
- Initially, we have 13 examples, of which 5 are classified as "Reply" and 8 classified as "Ghost" (5/8).

Information Gain

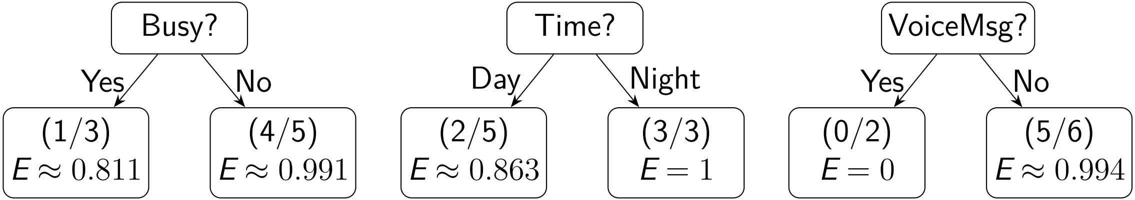
- Initially, we have 13 examples, of which 5 are classified as "Reply" and 8 classified as "Ghost" (5/8).
- So the entropy is $-\frac{5}{13}log_2(\frac{5}{13}) - \frac{8}{13}log_2(\frac{8}{13}) \approx 0.961$

Information Gain

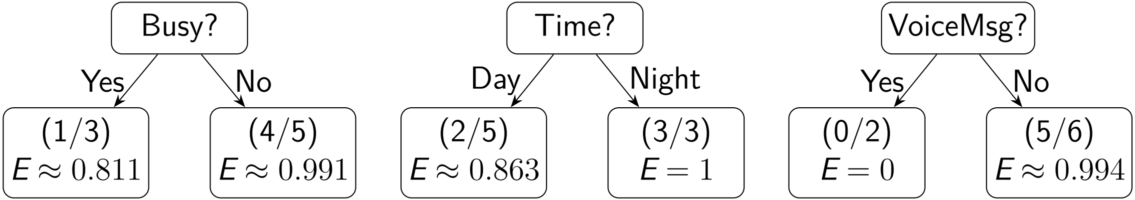
- Initially, we have 13 examples, of which 5 are classified as "Reply" and 8 classified as "Ghost" (5/8).
- So the entropy is $-\frac{5}{13}log_2(\frac{5}{13}) - \frac{8}{13}log_2(\frac{8}{13}) \approx 0.961$
-



Information Gain

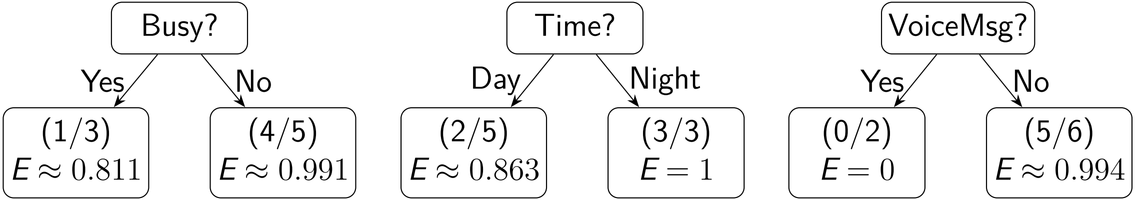
- Initially, we have 13 examples, of which 5 are classified as "Reply" and 8 classified as "Ghost" (5/8).
- So the entropy is $-\frac{5}{13}log_2(\frac{5}{13}) - \frac{8}{13}log_2(\frac{8}{13}) \approx 0.961$
- 

Information Gain

- Initially, we have 13 examples, of which 5 are classified as "Reply" and 8 classified as "Ghost" (5/8).
- So the entropy is $-\frac{5}{13}log_2(\frac{5}{13}) - \frac{8}{13}log_2(\frac{8}{13}) \approx 0.961$
- 

- When splitting on Busy, the final entropy is $\frac{4}{13} * 0.811 + \frac{9}{13} * 0.991 \approx 0.936$.

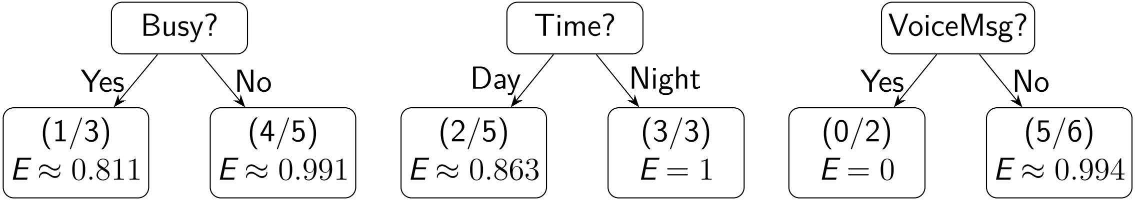
Information Gain

- Initially, we have 13 examples, of which 5 are classified as "Reply" and 8 classified as "Ghost" (5/8).
- So the entropy is $-\frac{5}{13}log_2(\frac{5}{13}) - \frac{8}{13}log_2(\frac{8}{13}) \approx 0.961$
- 

- When splitting on Busy, the final entropy is $\frac{4}{13} * 0.811 + \frac{9}{13} * 0.991 \approx 0.936$.
- When splitting on Time, the final entropy is $\frac{7}{13} * 0.863 + \frac{6}{13} * 1 \approx 0.926$.

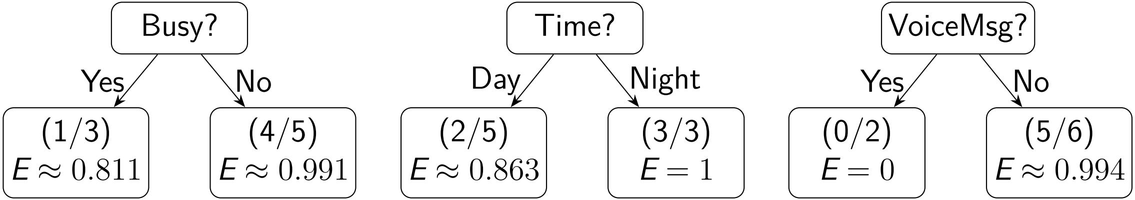
Information Gain

- Initially, we have 13 examples, of which 5 are classified as "Reply" and 8 classified as "Ghost" (5/8).
- So the entropy is $-\frac{5}{13}log_2(\frac{5}{13}) - \frac{8}{13}log_2(\frac{8}{13}) \approx 0.961$
-



- When splitting on Busy, the final entropy is $\frac{4}{13} * 0.811 + \frac{9}{13} * 0.991 \approx 0.936$.
- When splitting on Time, the final entropy is $\frac{7}{13} * 0.863 + \frac{6}{13} * 1 \approx 0.926$.
- When splitting on VoiceMsg, the final entropy is $\frac{2}{13} * 0 + \frac{11}{13} * 0.994 \approx 0.841$.

Information Gain

- Initially, we have 13 examples, of which 5 are classified as "Reply" and 8 classified as "Ghost" (5/8).
- So the entropy is $-\frac{5}{13} log_2(\frac{5}{13}) - \frac{8}{13} log_2(\frac{8}{13}) \approx 0.961$
- 

- When splitting on Busy, the final entropy is $\frac{4}{13} * 0.811 + \frac{9}{13} * 0.991 \approx 0.936$.
- When splitting on Time, the final entropy is $\frac{7}{13} * 0.863 + \frac{6}{13} * 1 \approx 0.926$.
- When splitting on VoiceMsg, the final entropy is $\frac{2}{13} * 0 + \frac{11}{13} * 0.994 \approx 0.841$.
- So, the highest info gain is $0.961 - 0.841 = 0.12$ (VoiceMsg).

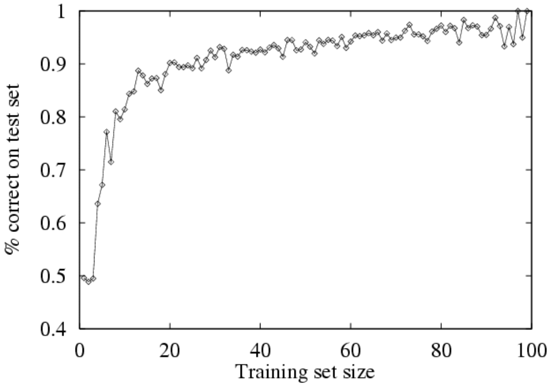
ID3 (Iterative Dichotomiser 3) Algorithm

ID3 Algorithm

- If all examples have same label:
  - Return leaf with that label.
- Else if there are no features left to test:
  - Return leaf with most common label.
- Else:
  - Choose feature $\hat{F}$ to maximizes the info-gain relative to the current node.
  - Add a branch from the node for each possible value of f in $\hat{F}$.
  - For each branch:
    - Remove $\hat{F}$ from the set of features.
    - Recursively call the algorithm to deepen the decision tree.

Learning Curve (this is for an example in your book)

- 100 examples, split into training/test data sets
- Each data point is average of 20 trials
- What do we infer?



Pruning

Pruning: Remove the subtree rooted at the node, make it a leaf, and assign it the most common classification.

- Apply a statistical test to estimate whether expanding/pruning a node is likely to improve the performance beyond the training set.

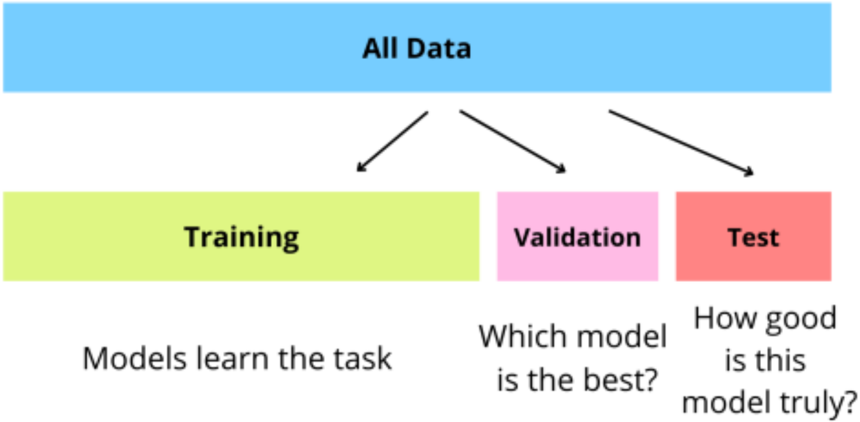
Pruning

Pruning: Remove the subtree rooted at the node, make it a leaf, and assign it the most common classification.

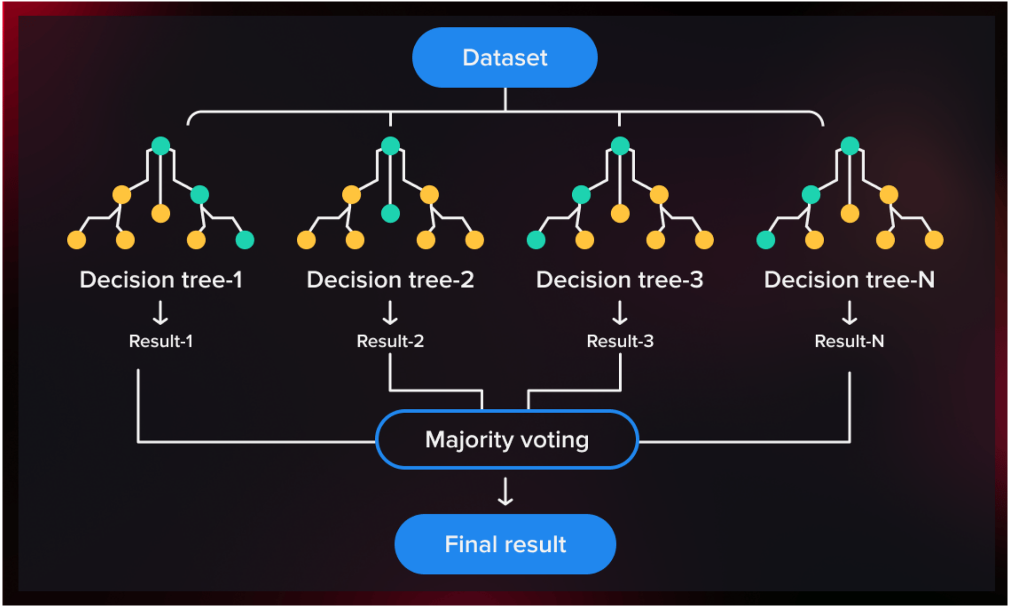
When to prune?

- Apply a statistical test to estimate whether expanding/pruning a node is likely to improve the performance beyond the training set.
- AND/OR use validation data: a separate set of examples, distinct from the training set, to evaluate utility of post-pruning nodes from the tree.

Never Test on you Training Data!



Random Forests



Random Forests

- Generate K (potentially smaller) datasets by randomly sampling **with** replacement.
- Select random sampling of attributes at each split point in constructing the tree.
- Reduce variance (more likely for one classifier to make a mistake than for half of all classifiers to make a mistake)

Decision Tree Recap

- Decision trees can be used to learn discrete-valued functions.
- The ID3 algorithm maximizes information gain to learn a decision tree.
- Pruning reduced overfitting.
- Random forests make decision trees viable on large datasets.