Lecture 5: Decision Trees and Their Representation

COMP 343, Spring 2022

Victoria Manfredi

WESLEYAN UNIVERSITY

Acknowledgements: These slides are based primarily on content from the book "Machine Learning" by Tom Mitchell, and on slides created by Vivek Srikumar (Utah), Dan Roth (Penn), and Jessica Wu (Harvey Mudd College)

Today's Topics

Homework 2 out

- Due Wed., February 16 by 5p

Decision trees (non-linear classifiers)

- Our first learning algorithm!
- Representation: what are decision trees?

Looking Forward

Using supervised learning

✓ What is our instance space?

- What are the inputs to the problem? What are the features?

✓ What is our label space?

- What kind of learning task are we dealing with?

✓ What is our hypothesis space?

- What functions should the learning algorithm search over?

4. What is our learning algorithm?
   - How do we learn the model from the labeled data?
5. What is our loss function or evaluation metric?
   - How do we measure success? What drives learning?

Much of the rest of the semester

Coming up …

Different hypothesis spaces and learning algorithms

Decision trees and ID3 algorithm

- (Non-parametric) regression and classification

Provides a "gentle" introduction to machine learning concepts

Rule-based decision algorithm

Linear regression

- Least mean squares regression

Linear classifiers

- Perceptron

Non-linear classifiers

- Multi-layer perceptron (neural networks)

Coming up …

1. Representation: What are decision trees?

2. Algorithm: Learning decision trees
    - The ID3 algorithm: a greedy heuristic

3. Some extensions

Decision Trees

REPRESENTATION

How to represent data?

Data can be represented as a big table, with columns denoting different attributes

Name	Label
Norman Danner	+
Karen Collins	-
Dan Licata	+
Danny Krizanc	+
Saray Shai	+
Wai Kiu Chan	-

How to represent data?

Data can be represented as a big table, with columns denoting different attributes

Name	Label
Norman Danner	+
Karen Collins	-
Dan Licata	+
Danny Krizanc	+
Saray Shai	+
Wai Kiu Chan	-

Problem: this representation does not generalize.

If name is not in table, we don't know what to do. So we need to extract attributes from the data

How to represent data?

Extract attributes from the data

Attributes/features x_i y_i

Each row is an instance

Name	Faculty department area	Name contains Dan	Second character of first name	Label
Norman Danner	CS	Yes	o	+
Karen Collins	Math	No	a	-
Dan Licata	CS	Yes	a	+
Danny Krizanc	CS	Yes	a	+
Saray Shai	CS	No	a	+
Wai Kiu Chan	Math	No	a	-

How to represent data?

Once we add other (informative) columns, maybe we don't need original data. Instead work with featurized version of data

Name	Faculty department area	Name contains Dan	Second character of first name	Label
Norman Danner	CS	Yes	o	+
Karen Collins	Math	No	a	-
Dan Licata	CS	Yes	a	+
Danny Krizanc	CS	Yes	a	+
Saray Shai	CS	No	a	+
Wai Kiu Chan	Math	No	a	-

How to represent data?

How big is this table?

Name	Faculty department area	Name contains Dan	Second character of first name	Label
Norman Danner	CS	Yes	o	+
Karen Collins	Math	No	a	-
Dan Licata	CS	Yes	a	+
Danny Krizanc	CS	Yes	a	+
Saray Shai	CS	No	a	+
Wai Kiu Chan	Math	No	a	-

How to represent data?

How big is this table?

Name	Faculty department area	Name contains Dan	Second character of first name	Label
Norm				
Karen				
Dan L				
Danny Krizanc	CS	Yes	a	+
Saray Shai	CS	No	a	+
Wai Kiu Chan	Math	No	a	-

With these three attributes, how many unique rows are possible?

$$2 \times 2 \times 26 = 104$$

There may be an infinite number of names, but we can have at most 104 possible buckets for names!

How to represent data?

Problem: how many columns could exist?

Name	Faculty department area	Name contains Dan	Second character of first name	Label
Norm				
Karen				
Dan L				
Danny				
Saray				
Wai K				

With these three attributes, how many unique rows are possible?

$$2 \times 2 \times 26 = 104$$

There may be an infinite number of names, but we can have at most 104 possible buckets for names!

If there are 100 attributes, all binary, how many unique rows are possible?

How to represent data?

Problem: how many columns could exist?

Name	Faculty department area	Name contains Dan	Second character of first name	Label
Norm				
Karen				
Dan L				
Danny				
Saray				
Wai K				

With these three attributes, how many unique rows are possible?

$$2 \times 2 \times 26 = 104$$

There may be an infinite number of names, but we can have at most 104 possible buckets for names!

If there are 100 attributes, all binary, how many unique rows are possible?

(100 times) $2 \times 2 \times 2 \times 2 \times \cdots \times 2 = 2^{100}$

How to represent data?

Problem: how many columns could exist?

Name	Faculty department area	Name contains Dan	Second character of first name	Label
Norm				
Karen				
Dan L				
Danny				
Saray				
Wai K				

With these three attributes, how many unique rows are possible?

$$2 \times 2 \times 26 = 104$$

There may be an infinite number of names, but we can have at most 104 possible buckets for names!

If there are 100 attributes, all binary, how many unique rows are possible?

(100 times) $2 \times 2 \times 2 \times 2 \times \cdots \times 2 = 2^{100}$

If we wanted to store all possible rows, this number is too large. How to represent data in better, more efficient way?

How to represent data?

Problem: how many columns could exist?

Name	Faculty department area	Name contains Dan	Second character of first name	Label
Norm				
Karen				
Dan L				
Danny Krizanc	CS	Yes	a	+
Saray Shai	CS	No	a	+
Wai Kiu Chan	Math	No	a	-

One way of thinking about decision trees is as a more efficient way of organizing data

What are decision trees?

A hierarchical data structure that represents data using a divide-and-conquer strategy

What are decision trees?

A hierarchical data structure that represents data using a divide-and-conquer strategy

Decision trees are also a hypothesis class. Can be used as a hypothesis class for non-parametric classification or regression

What are decision trees?

A hierarchical data structure that represents data using a divide-and-conquer strategy

Decision trees are also a hypothesis class. Can be used as a hypothesis class for non-parametric classification or regression

General idea: given a collection of examples, learn a decision tree that represents it. Use this representation to classify new examples

What are decision trees?

Decision trees are a family of classifiers (our focus) for instance that are represented by collections of attributes (i.e., feature vectors: color=; shape=; label=)

What are decision trees?

Decision trees are a family of classifiers (our focus) for instance that are represented by collections of attributes (i.e., feature vectors: color=; shape=; label=)

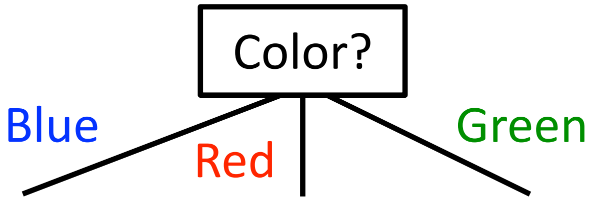
$\boxed{\text{Color?}}$

How do decision trees work?

- Nodes are tests for feature values

What are decision trees?

Decision trees are a family of classifiers (our focus) for instance that are represented by collections of attributes (i.e., feature vectors: color=; shape=; label=)

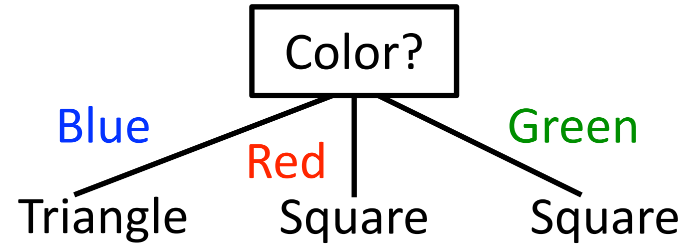


How do decision trees work?

- Nodes are tests for feature values
- There is one branch for every value that the feature can take

What are decision trees?

Decision trees are a family of classifiers (our focus) for instance that are represented by collections of attributes (i.e., feature vectors: color=; shape=; label=)



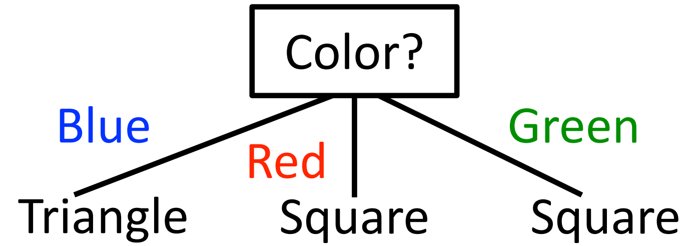
How do decision trees work?

- Nodes are tests for feature values
- There is one branch for every value that the feature can take
- Leaves of the tree specify the category (class labels)

What are decision trees?

Decision trees are a family of classifiers (our focus) for instance that are represented by collections of attributes (i.e., feature vectors: color=; shape=; label=)

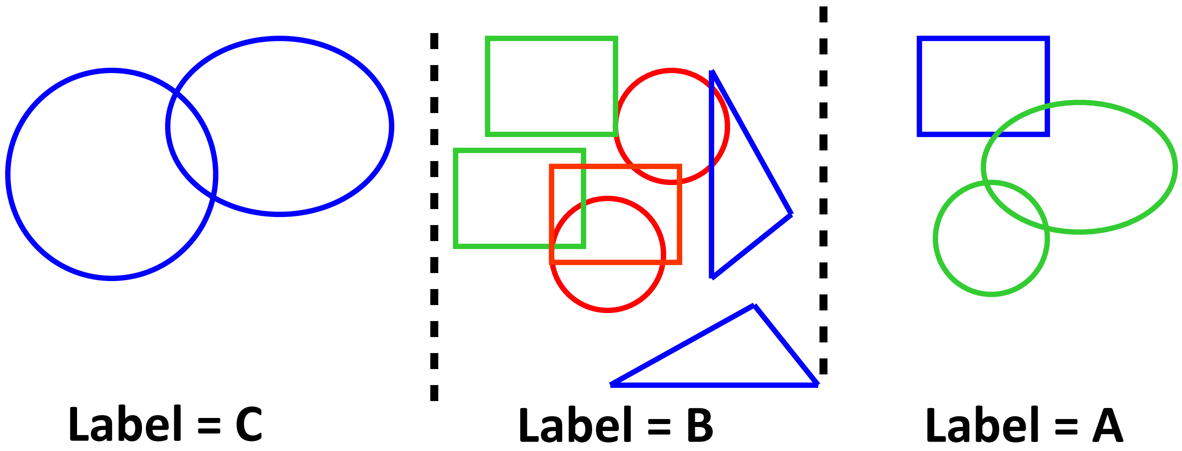
How do decision trees work?



- Nodes are tests for feature values
- There is one branch for every value that the feature can take
- Leaves of the tree specify the category (class labels)

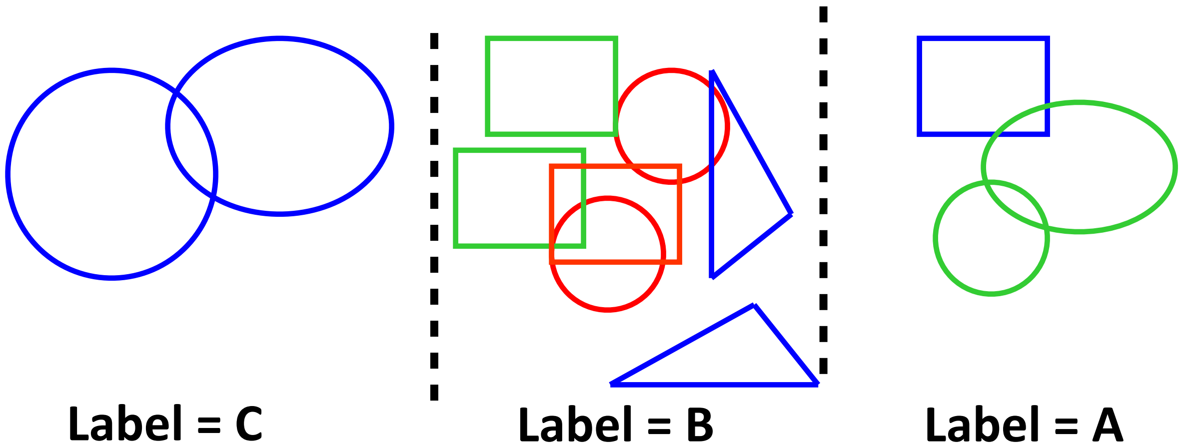
Like playing twenty questions game

Let’s build a decision tree for classifying shapes



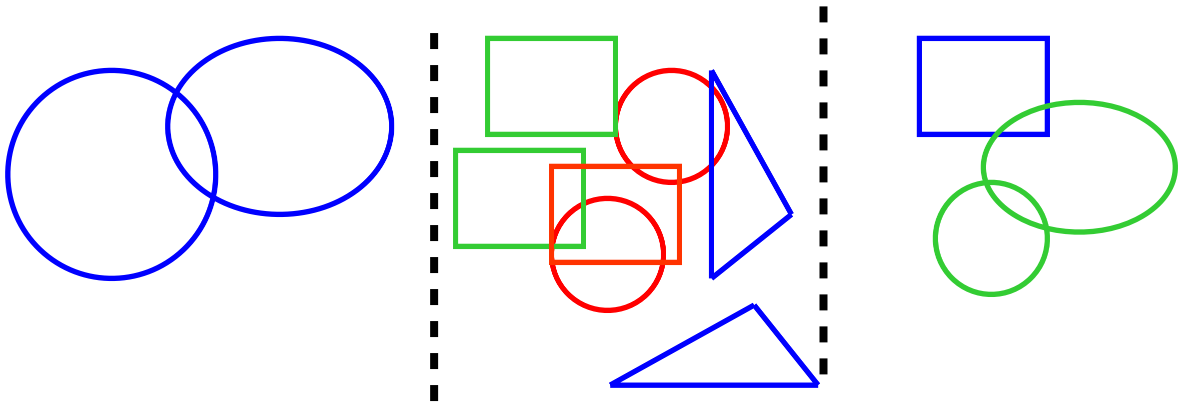
Our dataset comprises shapes

Let’s build a decision tree for classifying shapes



What are some attributes of the shapes?

Let's build a decision tree for classifying shapes

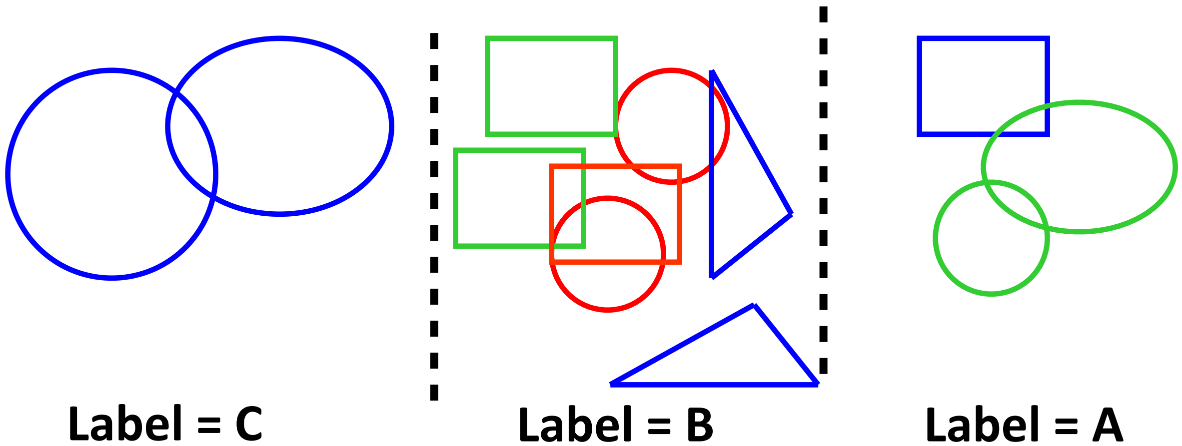


Label = C **Label = B** **Label = A**

What are some attributes of the shapes?

Color, shape, size

Let's build a decision tree for classifying shapes



What are some attributes of the shapes?

Color, shape, size

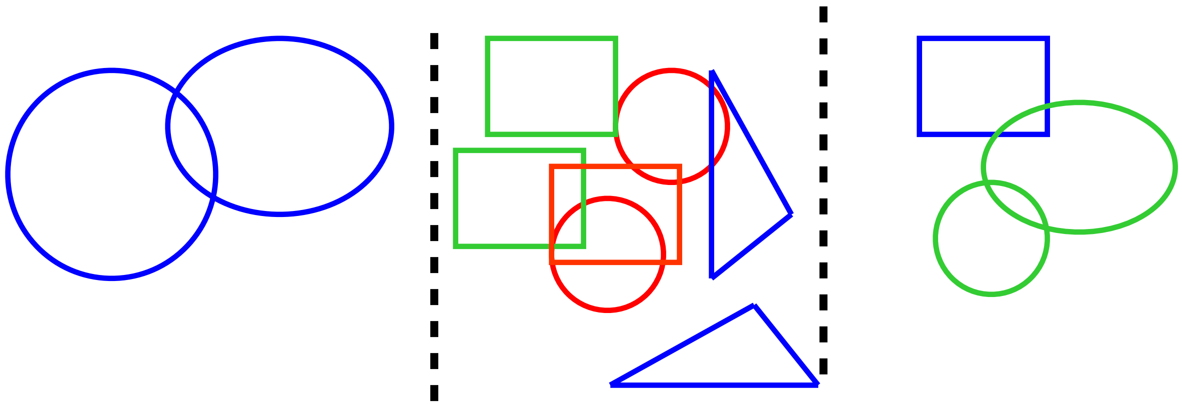
Let's build a decision tree

Internal node: test feature x_i

Branch from node: select one value for x_i

Each leaf node: predict label y

Let’s build a decision tree for classifying shapes



Label = C **Label = B** **Label = A**

What are some attributes of the shapes?

Color, shape, size

Color?

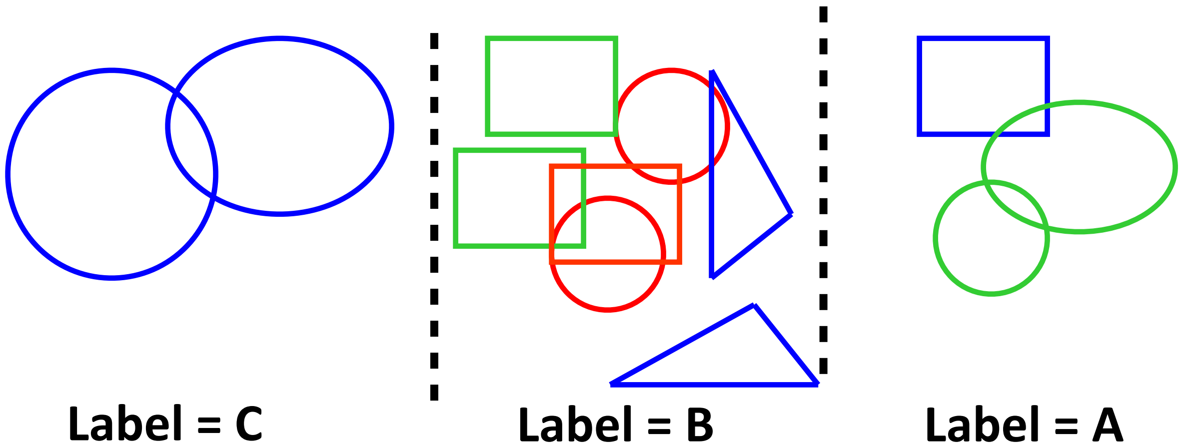
First, what is the color of the shape?

Internal node: test feature x_i

Branch from node: select one value for x_i

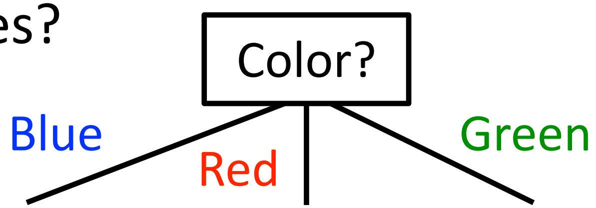
Each leaf node: predict label y

Let's build a decision tree for classifying shapes



What are some attributes of the shapes?

Color, shape, size



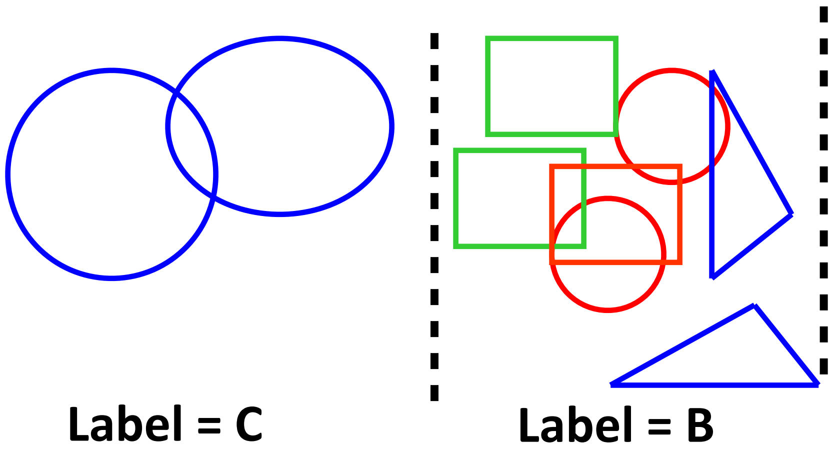
3 options for color, so 3 edges

Internal node: test feature x_i

Branch from node: select one value for x_i

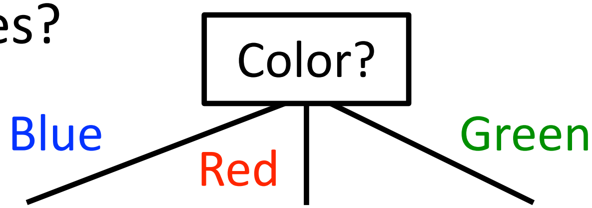
Each leaf node: predict label y

Let's build a decision tree for classifying shapes



What are some attributes of the shapes?

Color, shape, size



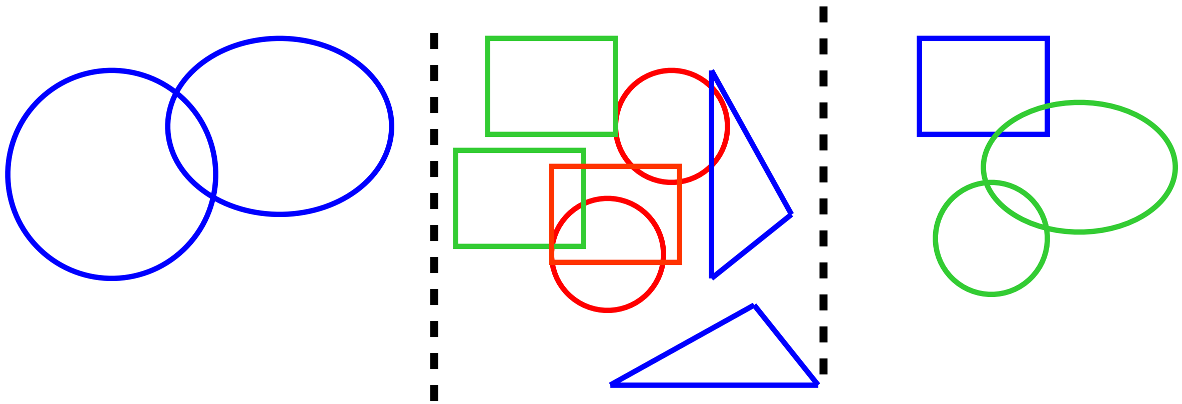
If color is red, you know label immediately. What is it?

Internal node: test feature x_i

Branch from node: select one value for x_i

Each leaf node: predict label y

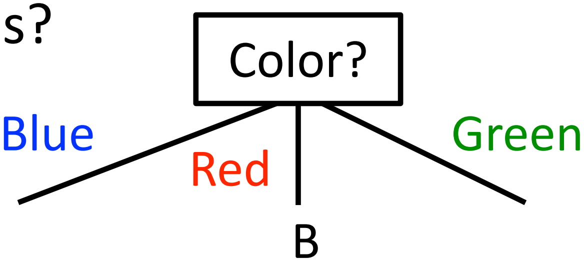
Let's build a decision tree for classifying shapes



Label = C **Label = B** **Label = A**

What are some attributes of the shapes?

Color, shape, size



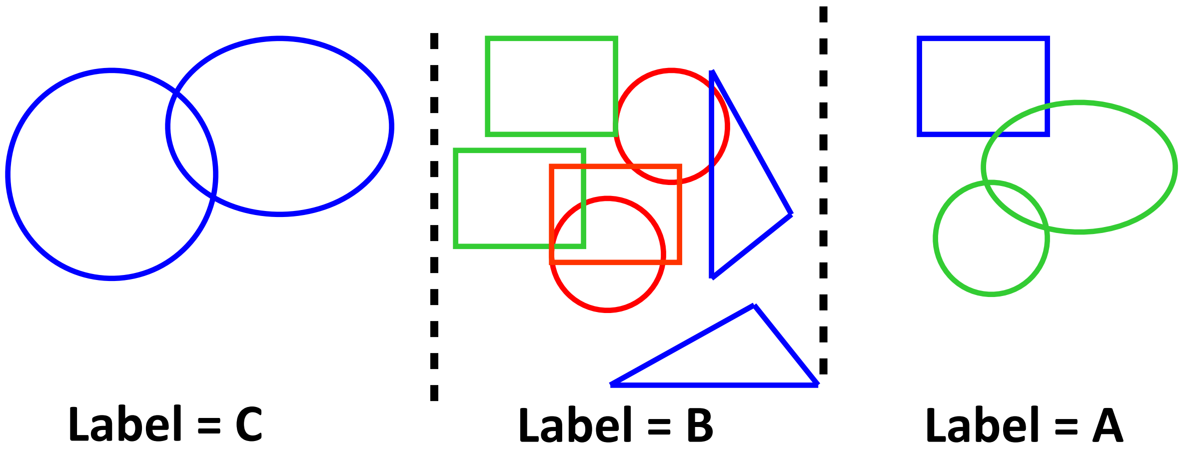
If color is red, you know label immediately. What is it?

Internal node: test feature x_i

Branch from node: select one value for x_i

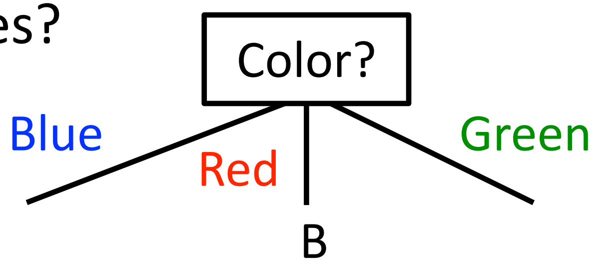
Each leaf node: predict label y

Let's build a decision tree for classifying shapes



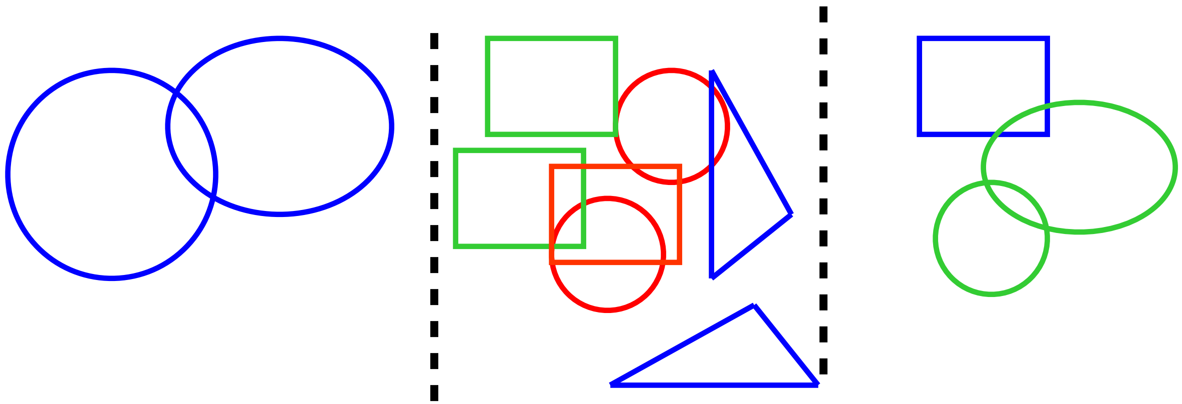
What are some attributes of the shapes?

Color, shape, size



If color is blue, what should we ask about next?

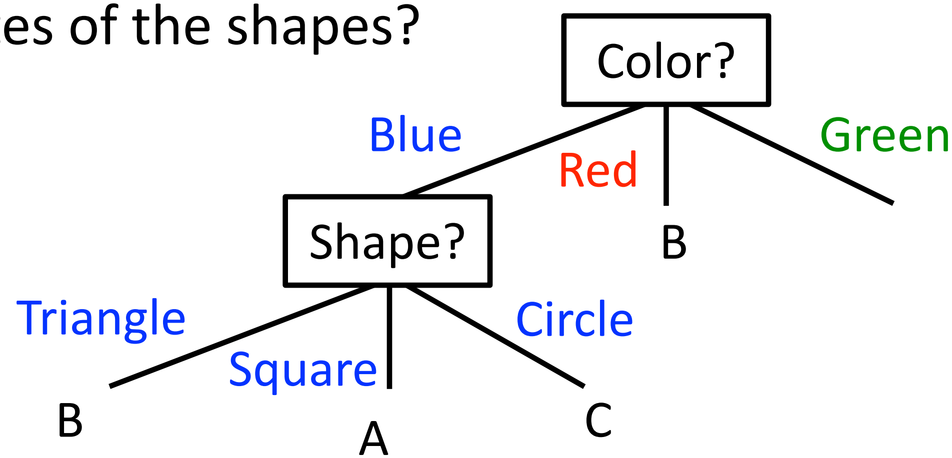
Let’s build a decision tree for classifying shapes



Label = C **Label = B** **Label = A**

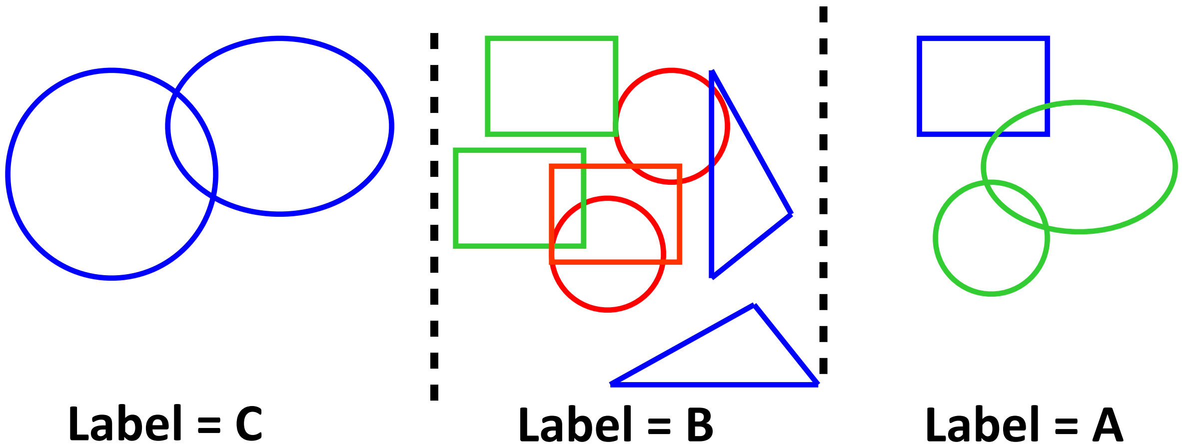
What are some attributes of the shapes?

Color, shape, size



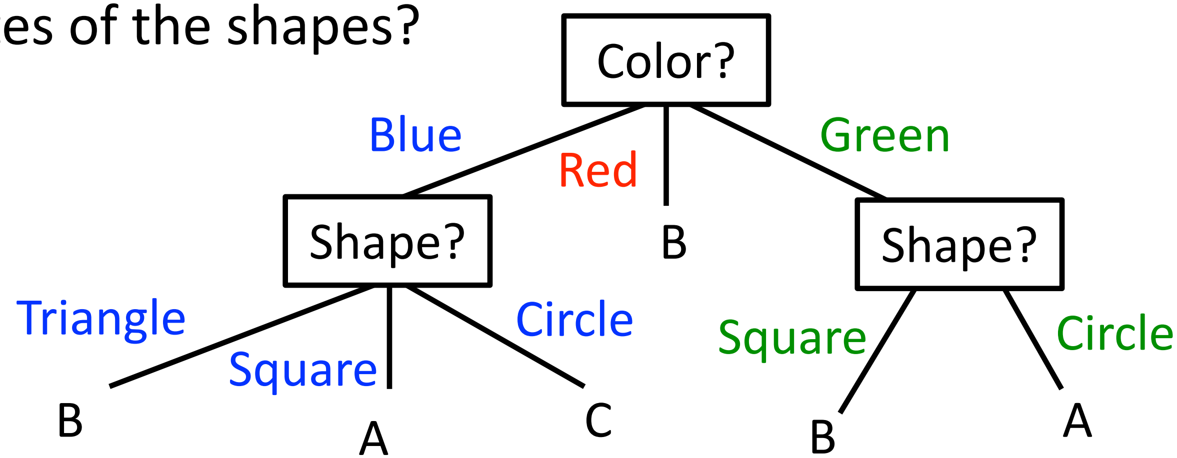
If color is blue, what should we ask about next?

Let’s build a decision tree for classifying shapes



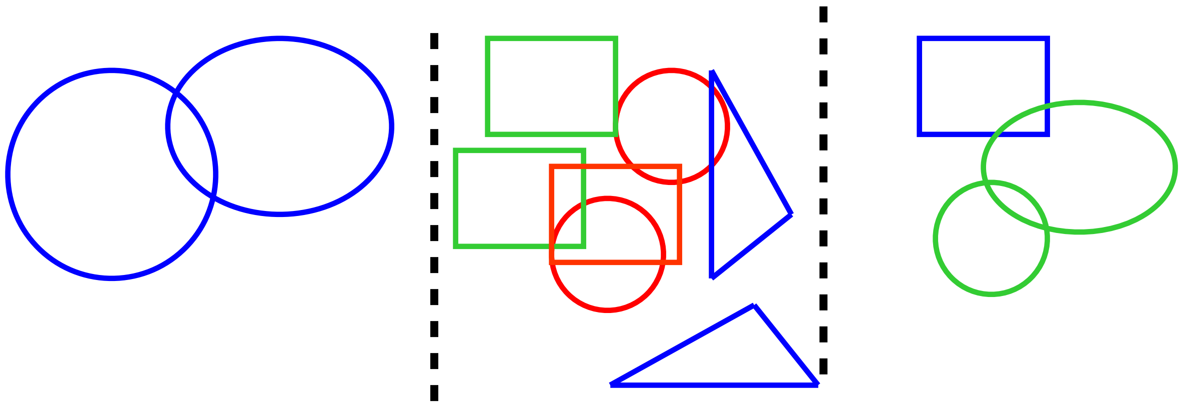
What are some attributes of the shapes?

Color, shape, size



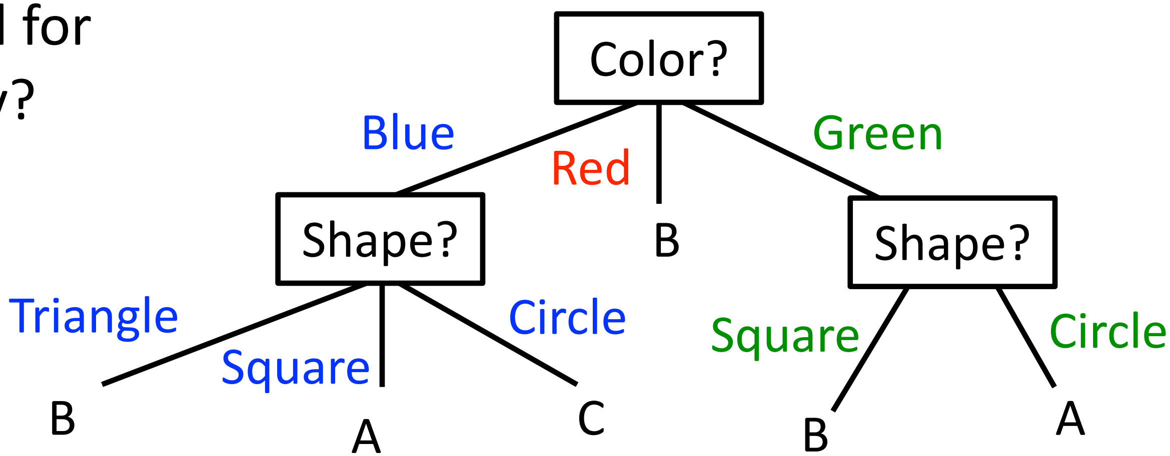
If color is green, we also ask about shape

Let’s build a decision tree for classifying shapes

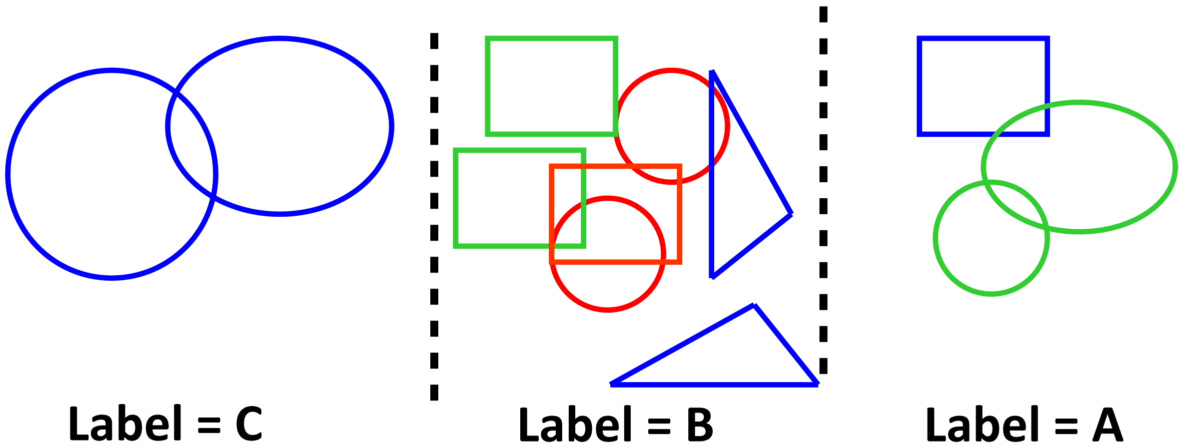


Label = C **Label = B** **Label = A**

3 min: What is the label for a red triangle? And why?

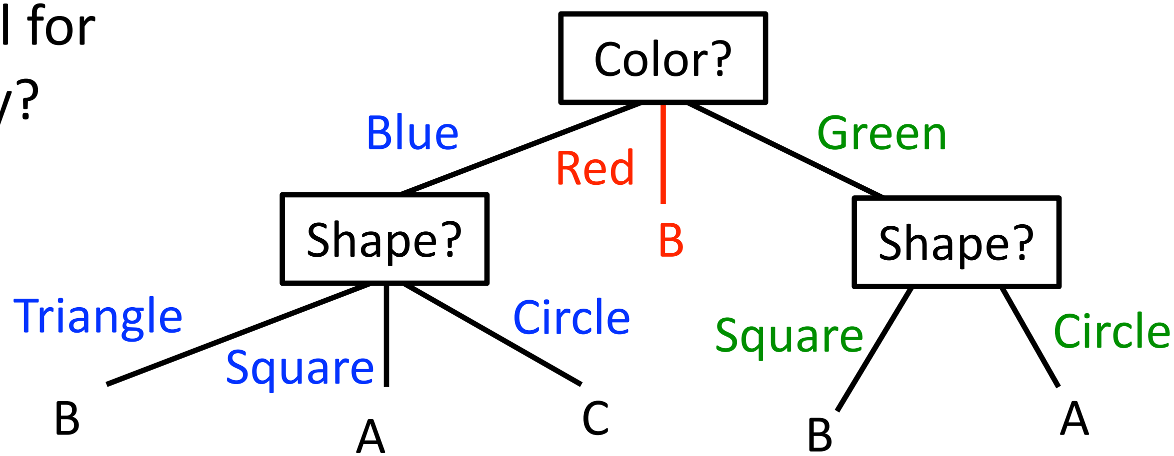


Let’s build a decision tree for classifying shapes

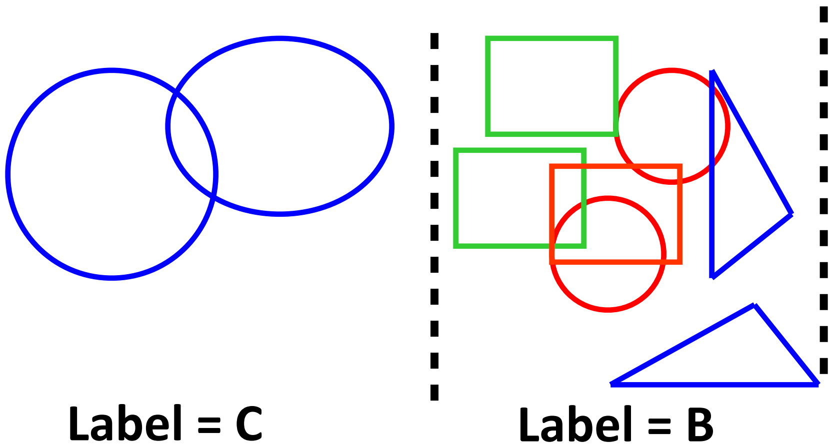


3 min: What is the label for a red triangle? And why?

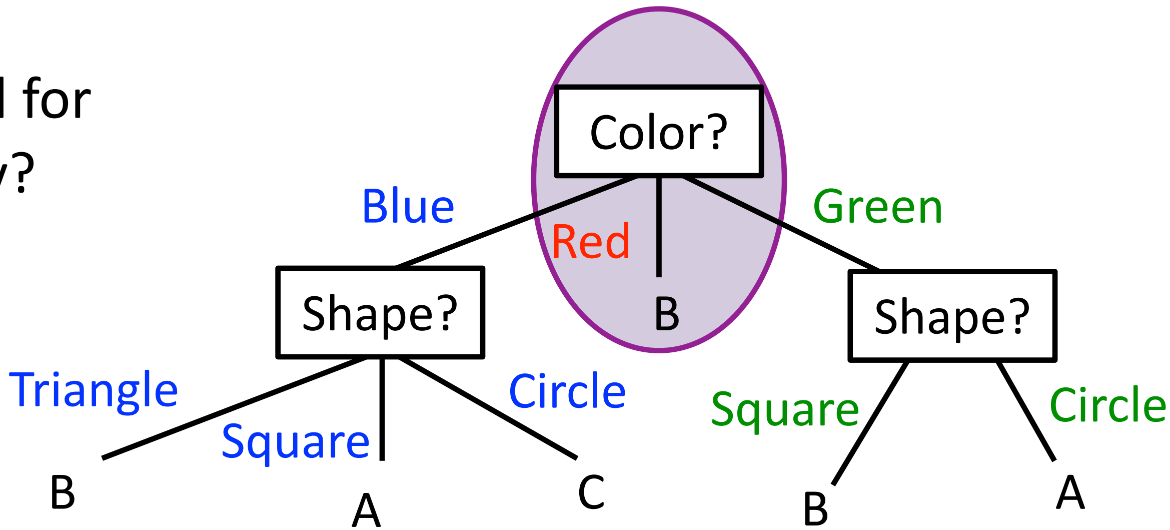
Even though no red triangle in data, can still label it. Start at root and go down tree. If color is red, go down that edge and label is B.



Let’s build a decision tree for classifying shapes

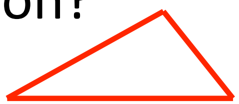


3 min: What is the label for a red triangle? And why?

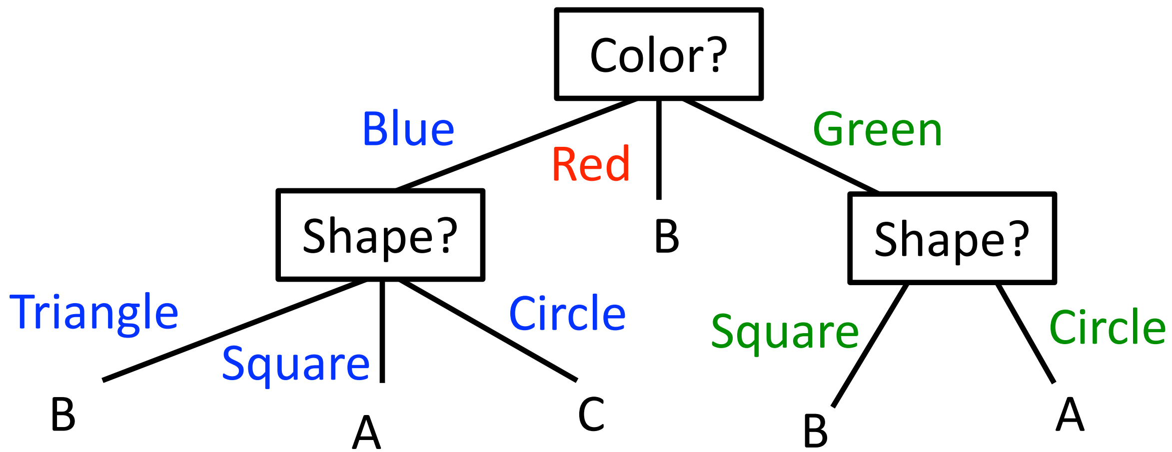


This is generalization! We predict labels for shapes we have never seen before

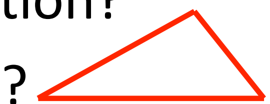
Let's build a decision tree for classifying shapes

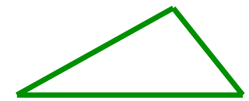
1. How do we learn a decision tree? Coming up soon
2. How to use a decision tree for prediction?
   - What is the label for a red triangle? 

     $\Longrightarrow$ Just follow a path from the root to a leaf

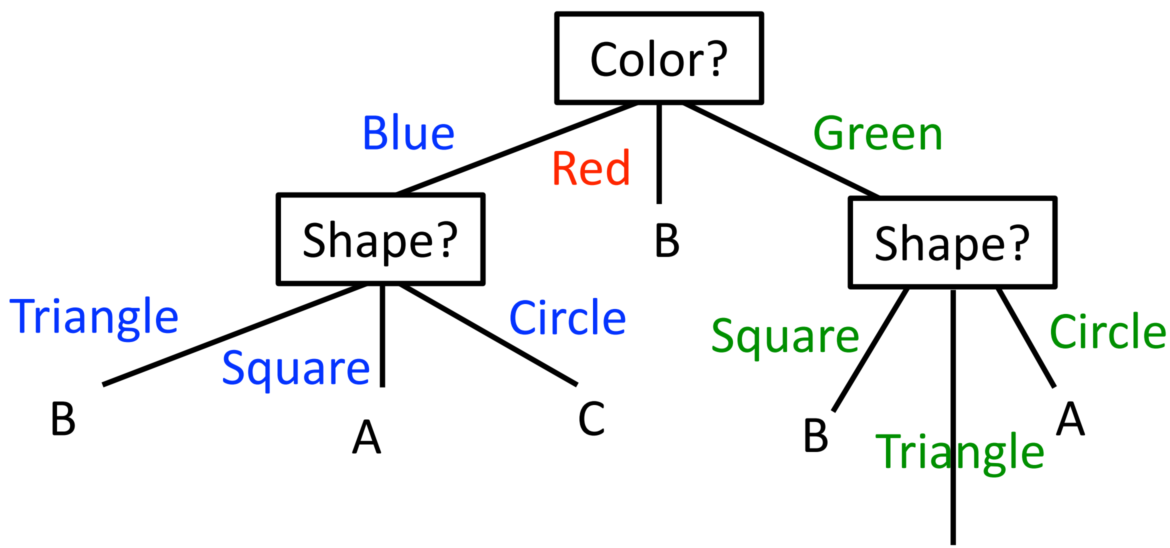


Let's build a decision tree for classifying shapes

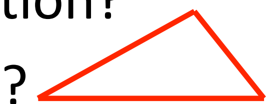
1. How do we learn a decision tree? Coming up soon
2. How to use a decision tree for prediction?
    - What is the label for a red triangle? 

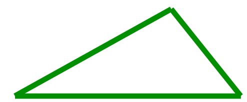
      $\Longrightarrow$ Just follow a path from the root to a leaf
    - What about a green triangle? 

If we knew shapes could be circles, squares or triangle, then add branch

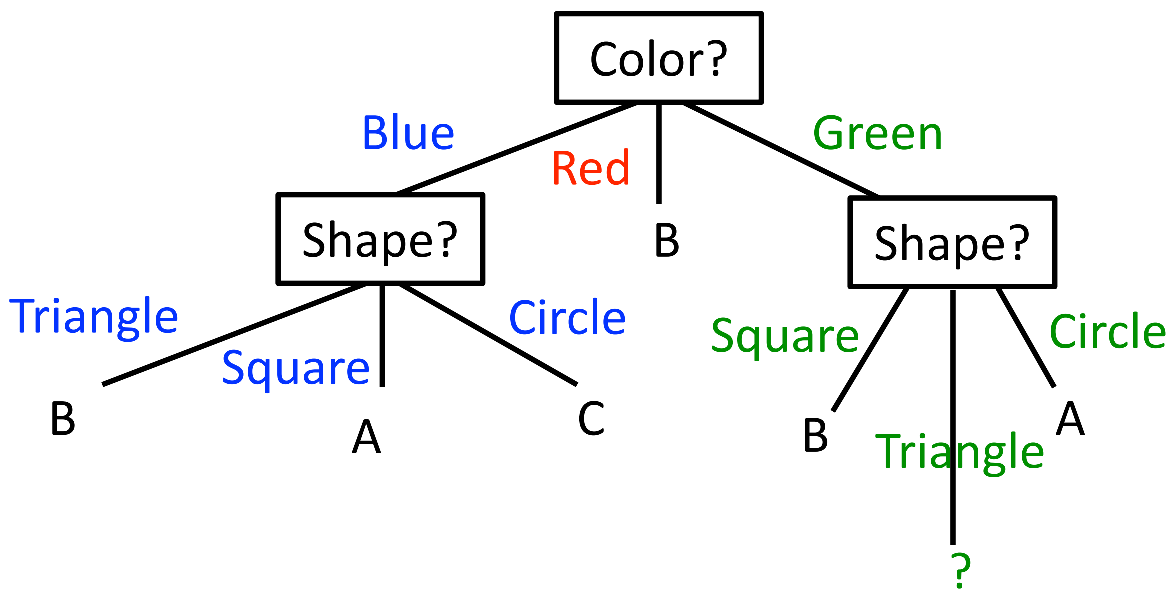


Let’s build a decision tree for classifying shapes

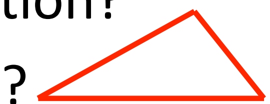
1. How do we learn a decision tree? Coming up soon
2. How to use a decision tree for prediction?
   - What is the label for a red triangle? 

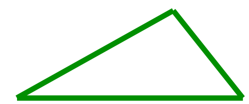
     $\Longrightarrow$ Just follow a path from the root to a leaf
   - What about a green triangle? 

3 min: Think about how to label branch?



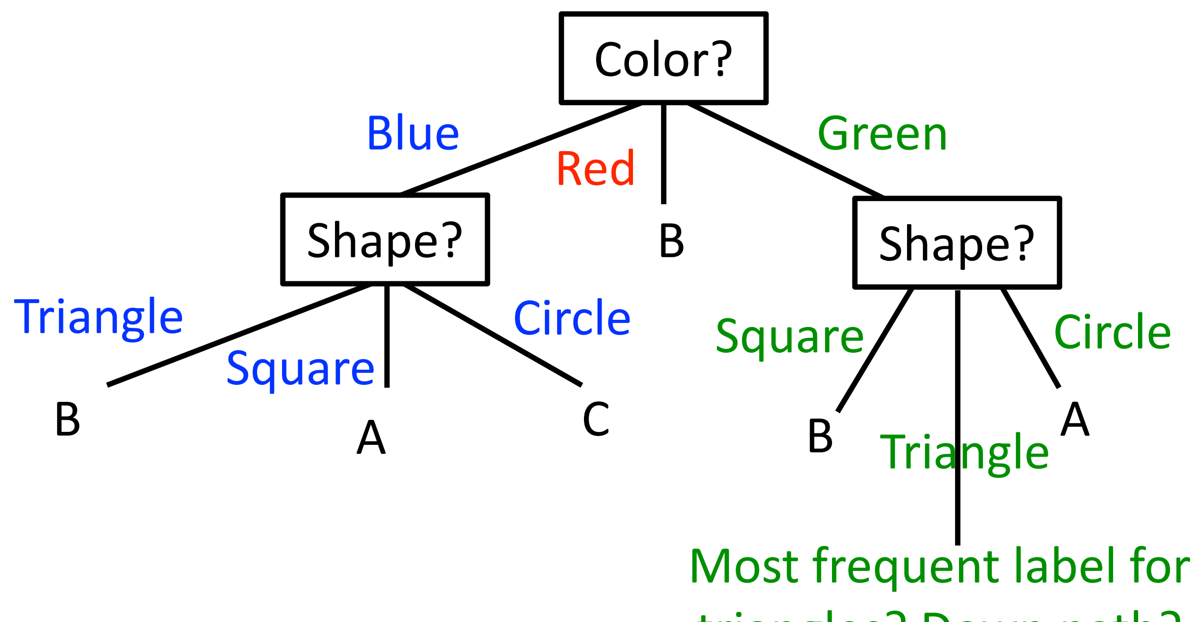
Let's build a decision tree for classifying shapes

1. How do we learn a decision tree? Coming up soon
2. How to use a decision tree for prediction?
   - What is the label for a red triangle? 

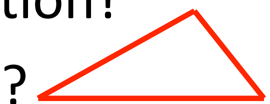
     $\Longrightarrow$ Just follow a path from the root to a leaf
   - What about a green triangle? 

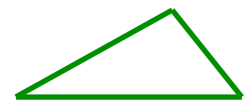
What about most frequent label?

Surprisingly good. Minimizes chance of mistake



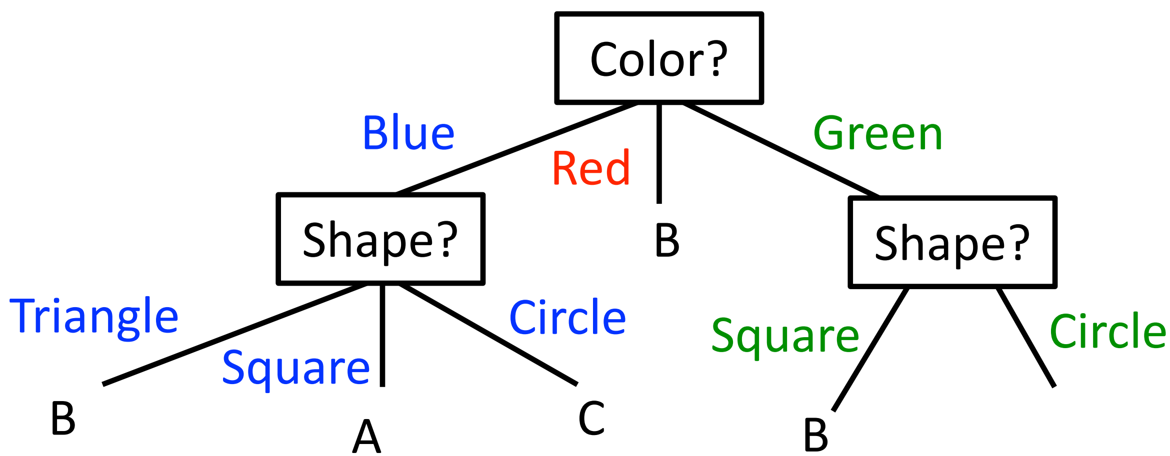
Let’s build a decision tree for classifying shapes

1. How do we learn a decision tree? Coming up soon
2. How to use a decision tree for prediction?
    - What is the label for a red triangle? 

        $\Longrightarrow$ Just follow a path from the root to a leaf
    - What about a green triangle? 

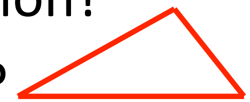
What if something does not fall into any of these categories?

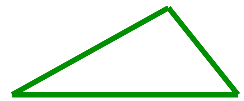
Maybe yellow diamond?



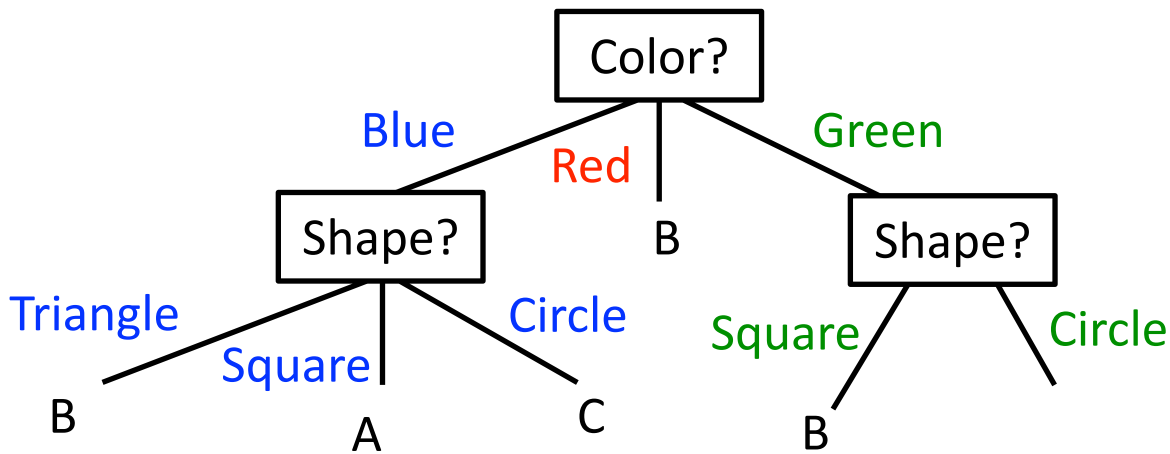
One option: add another edge in tree for “unseen”

Let's build a decision tree for classifying shapes

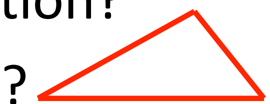
1. How do we learn a decision tree? Coming up soon
2. How to use a decision tree for prediction?
   - What is the label for a red triangle? 

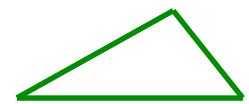
     $\Longrightarrow$ Just follow a path from the root to a leaf
   - What about a green triangle? 

Why might we not want to keep adding edges for every possibility?



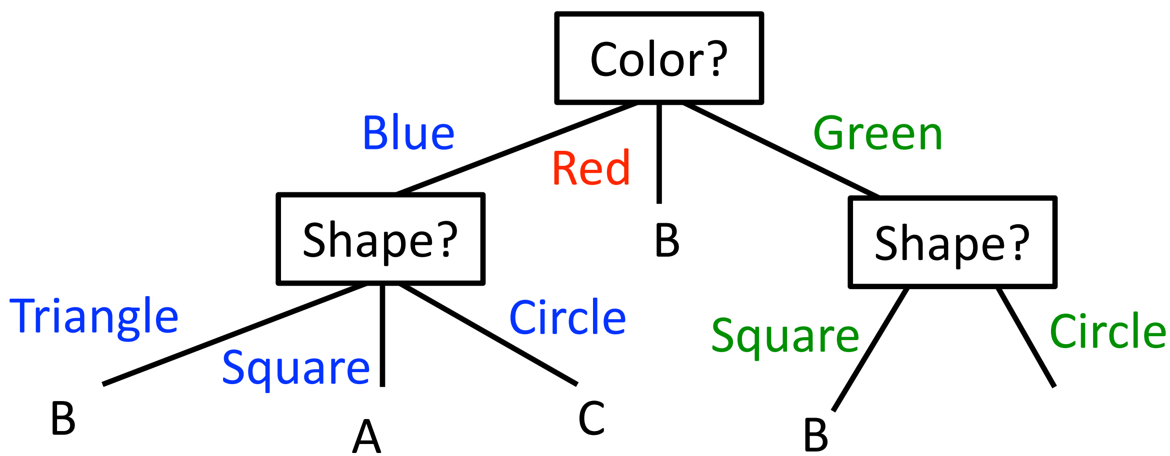
Let’s build a decision tree for classifying shapes

1. How do we learn a decision tree? Coming up soon
2. How to use a decision tree for prediction?
   - What is the label for a red triangle? 

     $\Longrightarrow$ Just follow a path from the root to a leaf
   - What about a green triangle? 

Why might we not want to keep adding edges for every possibility?

To get some level of compaction. Completing the full tree not efficient



Problem setting for decision trees

Set of possible instances, X

- Each instance $\mathbf{x} \in X$ is a feature vector
- E.g., <shape=square, color=red>

Problem setting for decision trees

Set of possible instances, X

- Each instance $\mathbf{x} \in X$ is a feature vector
- E.g., <shape=square, color=red>

Set of possible labels, Y

- Y is discrete-valued

Problem setting for decision trees

Set of possible instances, X

- Each instance $\mathbf{x} \in X$ is a feature vector
- E.g., <shape=square, color=red>

Set of possible labels, Y

- Y is discrete-valued

Unknown target function, $f : X \rightarrow Y$

Problem setting for decision trees

Set of possible instances, X

- Each instance $\mathbf{x} \in X$ is a feature vector
- E.g., <shape=square, color=red>

Set of possible labels, Y

- Y is discrete-valued

Unknown target function, $f : X \to Y$

Set of function hypotheses, $H = \{h \mid h : X \to Y\}$

- Each hypothesis h is a decision tree
- Trees sort $\mathbf{x}$ to leaf which assigns $y \in Y$

Expressivity of decision trees

What is the hypothesis class that decision trees represent?

Remember, a hypothesis is just a function that maps instances to labels

So what class of functions can decision trees represent?

3min: what do you think this class of functions is?

Expressivity of decision trees

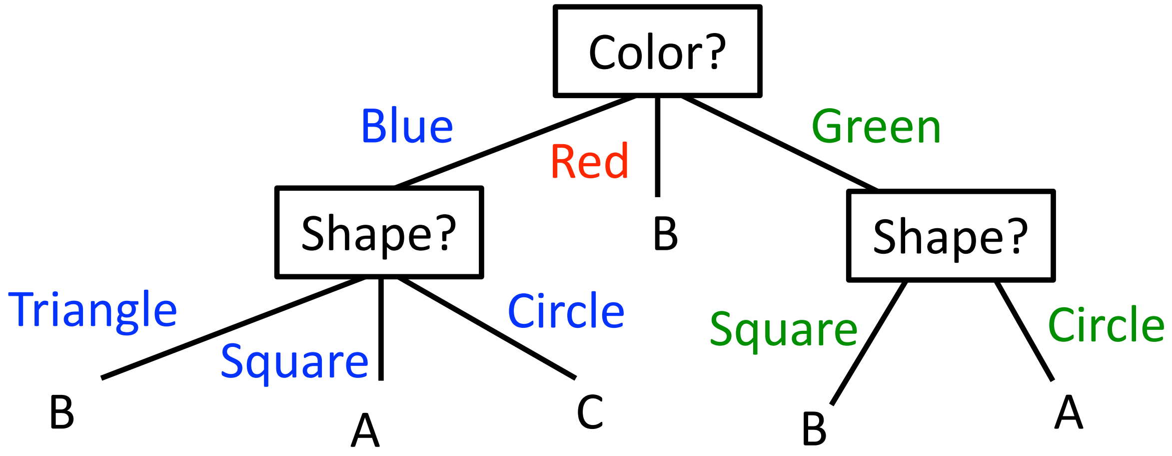
What functions can decision trees represent?

- *Any* Boolean function (assuming all features are Boolean)

Expressivity of decision trees

What functions can decision trees represent?

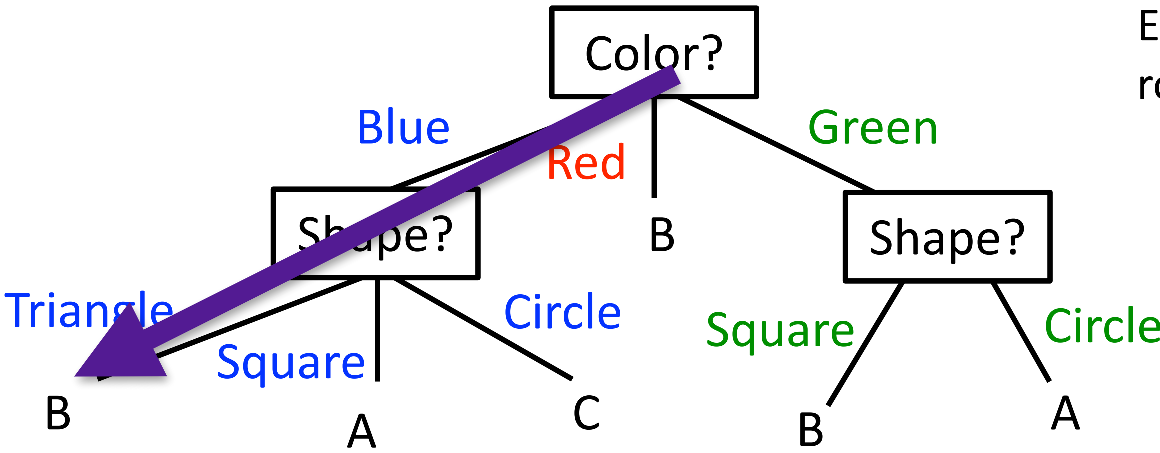
- *Any* Boolean function (assuming all features are Boolean)



Expressivity of decision trees

What functions can decision trees represent?

- *Any* Boolean function (assuming all features are Boolean)

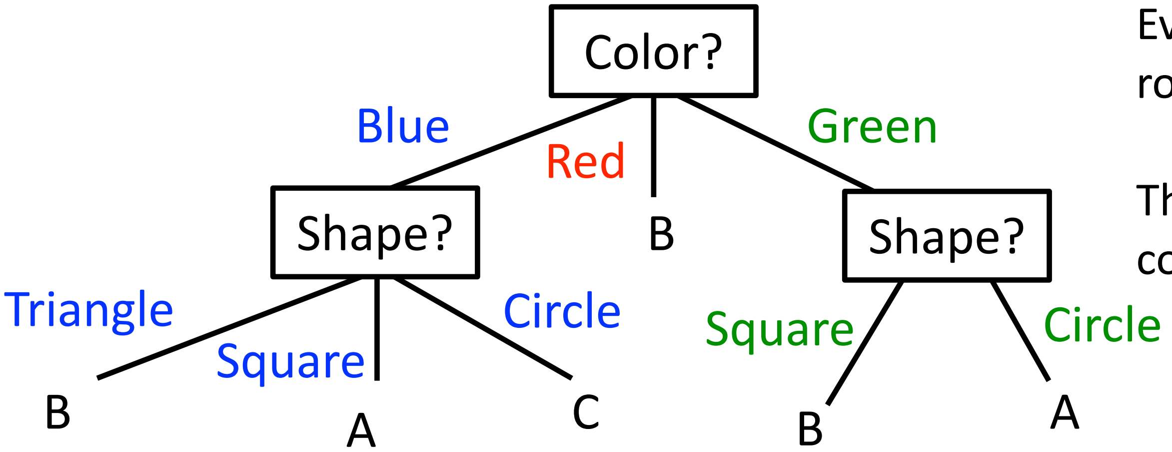


Every path from the tree to a root is a rule

Expressivity of decision trees

What functions can decision trees represent?

- *Any* Boolean function (assuming all features are Boolean)



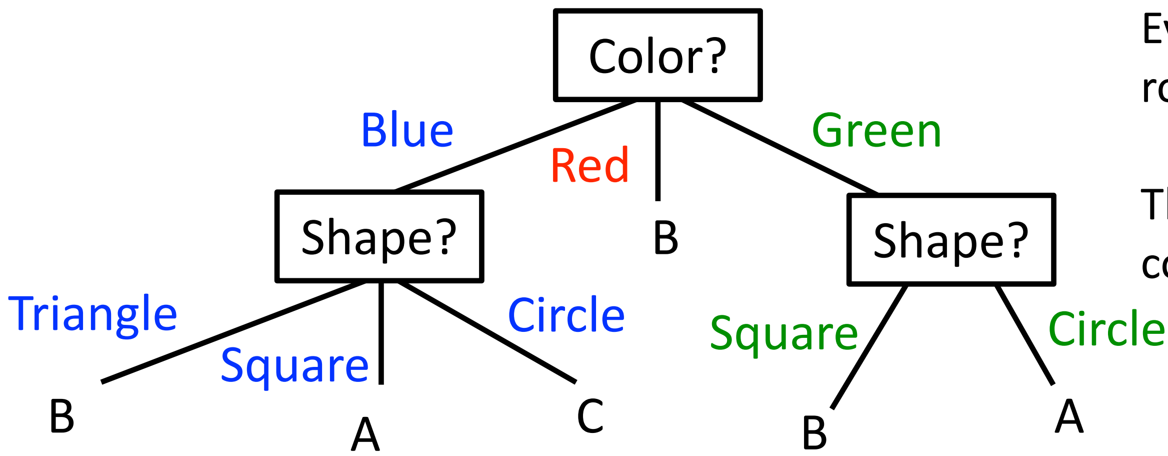
Every path from the tree to a root is a rule

The full tree is equivalent to the conjunction of all the rules

Expressivity of decision trees

What functions can decision trees represent?

- *Any* Boolean function (assuming all features are Boolean)



Every path from the tree to a root is a rule

The full tree is equivalent to the conjunction of all the rules

(Color=Blue AND Shape=Triangle => Label=B) AND

(Color=Blue AND Shape=Square => Label=A) AND

(Color=Blue AND Shape=Circle => Label=C) AND ....

Expressivity of decision trees

What functions can decision trees represent?

- *Any* Boolean function (assuming all features are Boolean)

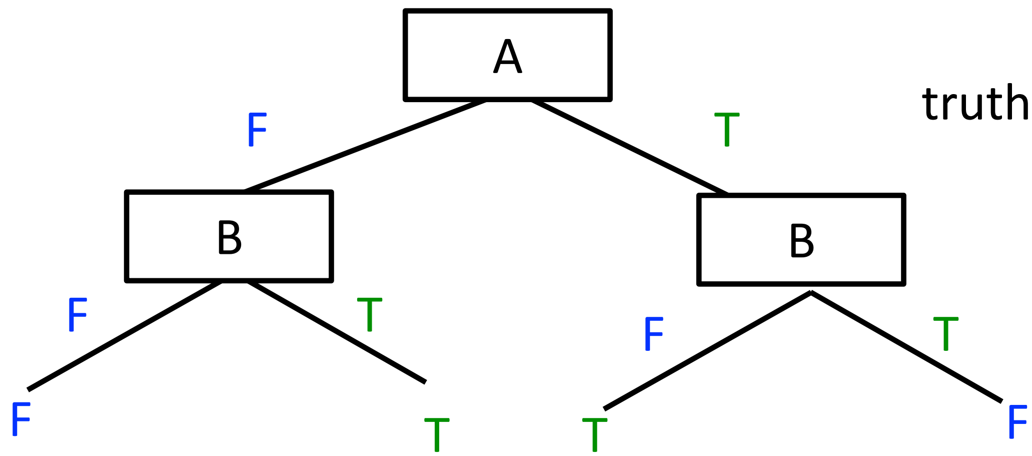
3 min: how would you represent this function as a decision tree?

A	**B**	**A xor B**
F	F	**F**
F	T	**T**
T	F	**T**
T	T	**F**

Expressivity of decision trees

What functions can decision trees represent?

- *Any* Boolean function (assuming all features are Boolean)



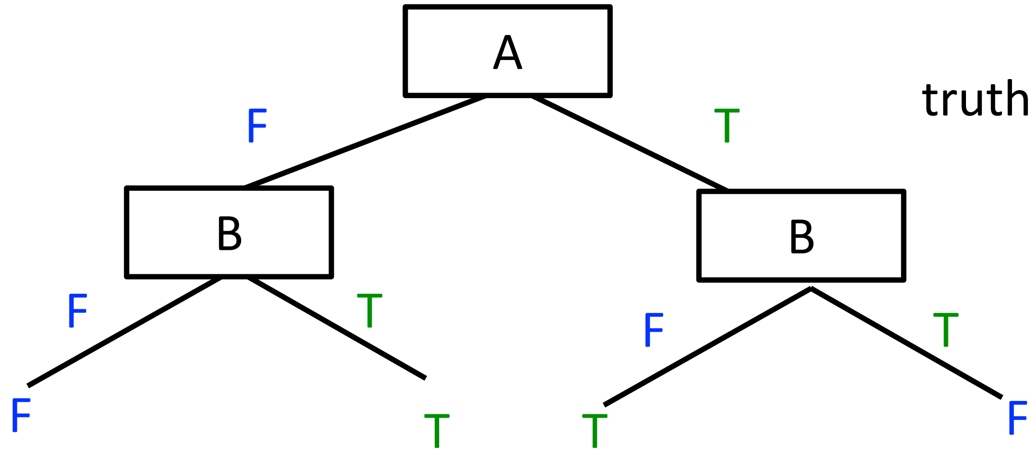
truth table row →path to leaf

A	B	A xor B
F	F	**F**
F	T	**T**
T	F	**T**
T	T	**F**

Expressivity of decision trees

What functions can decision trees represent?

- *Any* Boolean function (assuming all features are Boolean)



truth table row →path to leaf

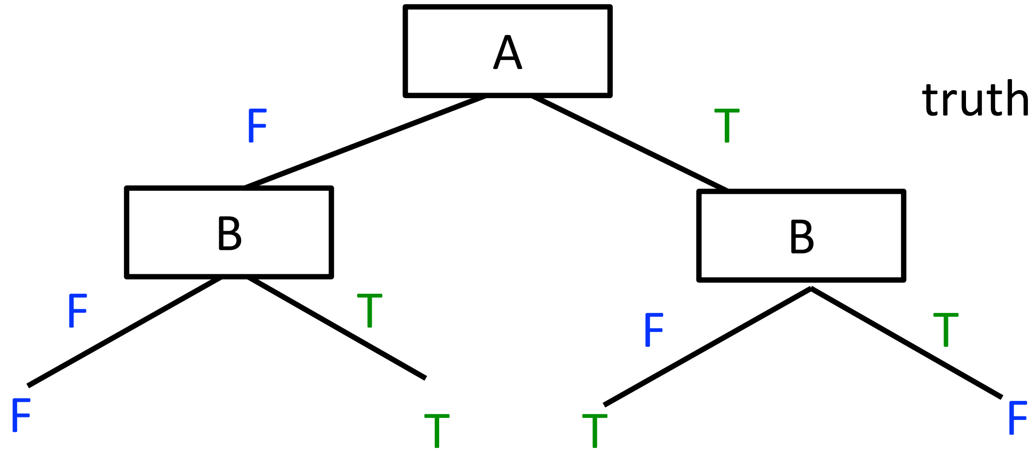
In worst case, how many nodes needed?

A	B	A xor B
F	F	F
F	T	T
T	F	T
T	T	F

Expressivity of decision trees

What functions can decision trees represent?

- *Any* Boolean function (assuming all features are Boolean)



truth table row →path to leaf

A	B	A xor B
F	F	**F**
F	T	**T**
T	F	**T**
T	T	**F**

In worst case, how many nodes needed?
Exponentially many!

Decision trees have variable-sized hypothesis space: as # of nodes (or depth) increases, hypothesis space grows (can express more complex functions)

Decision Trees

Outputs are discrete categories (classification)

Decision Trees

Outputs are discrete categories (classification)

But real valued outputs are possible (regression trees)

Decision Trees

Outputs are discrete categories (classification)

But real valued outputs are possible (regression trees)

There are methods for handling noisy data (noise in the label or in the features) and for handling missing attribute values. Pruning trees helps with noise. Smaller trees also tend to generalize better. More on this later …

Numeric attributes and decision boundaries

We have seen instances represented as attribute-value pairs (color=Blue, shape=Square, second letter=a)

- Values have been categorical

Numeric attributes and decision boundaries

We have seen instances represented as attribute-value pairs (color=Blue, shape=Square, second letter=a)

- Values have been categorical

How do we deal with numeric feature values (e.g., length = ?)

What can we do?

Numeric attributes and decision boundaries

We have seen instances represented as attribute-value pairs (color=Blue, shape=Square, second letter=a)

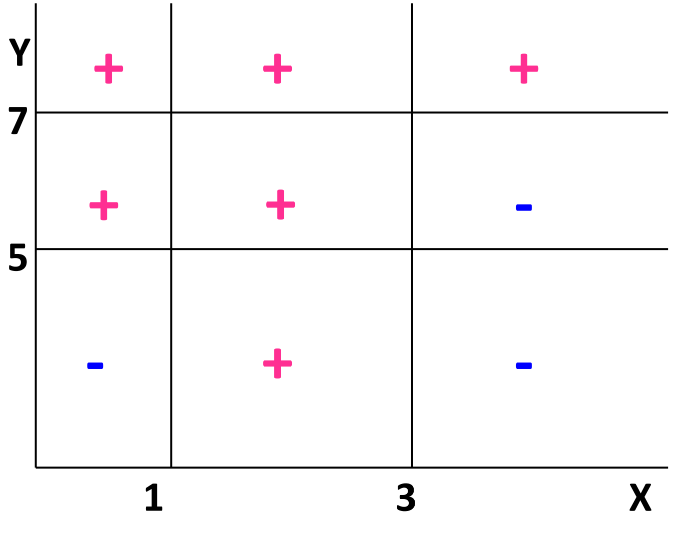
- Values have been categorical

How do we deal with numeric feature values (e.g., length = ?)

- Discretize values
- Use thresholds on the values for splitting nodes

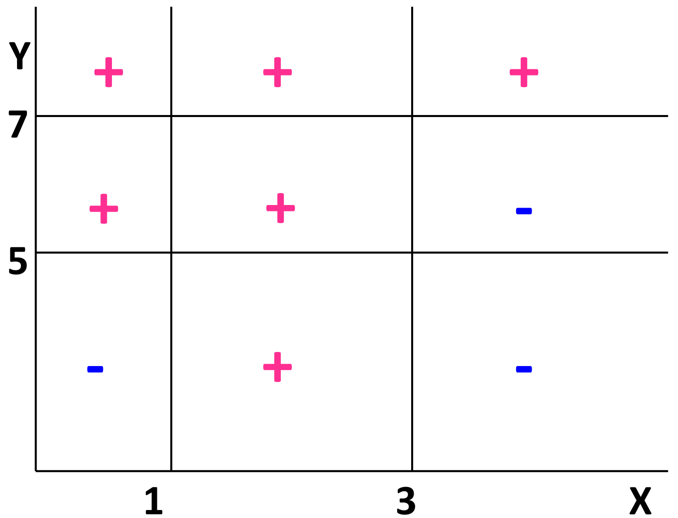
Numeric attributes and decision boundaries

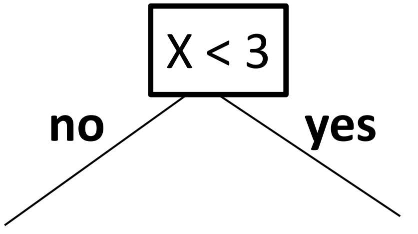
Dividing the features space into axis-parallel rectangles



Numeric attributes and decision boundaries

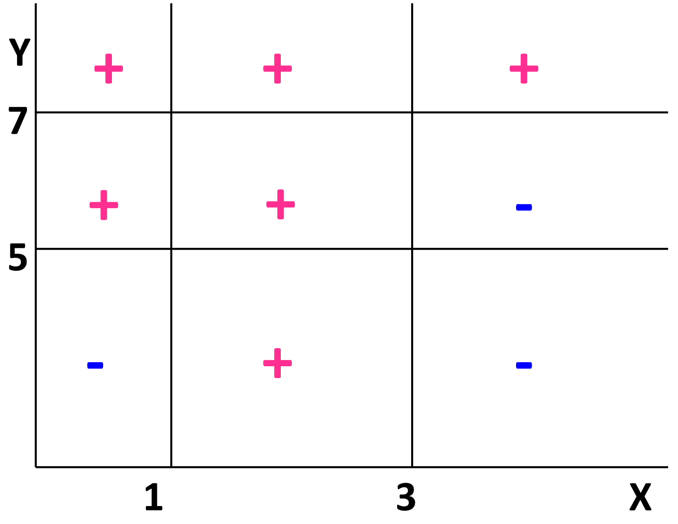
Dividing the features space into axis-parallel rectangles

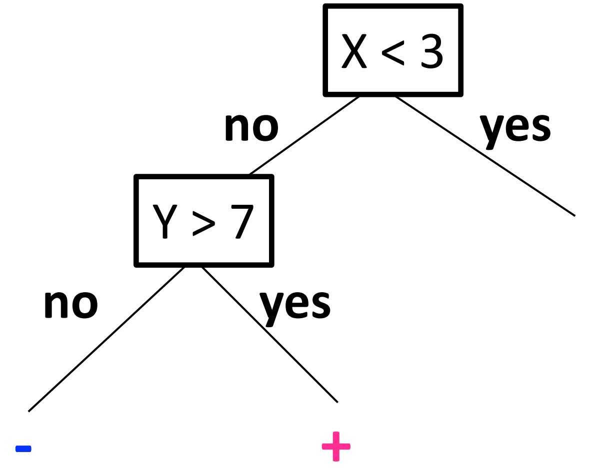




Numeric attributes and decision boundaries

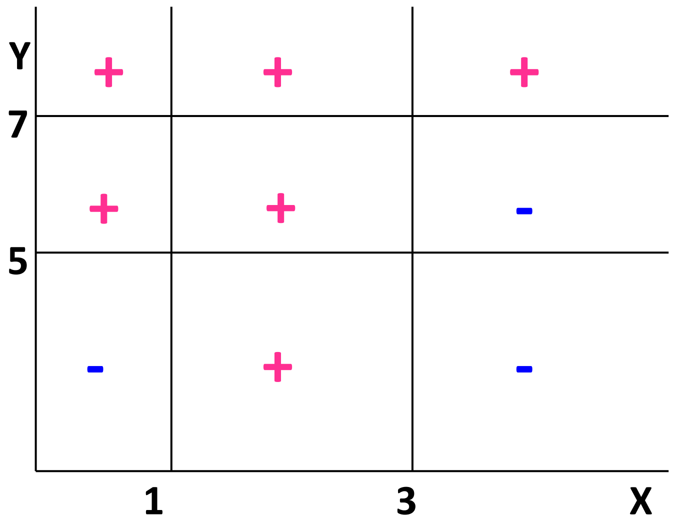
Dividing the features space into axis-parallel rectangles

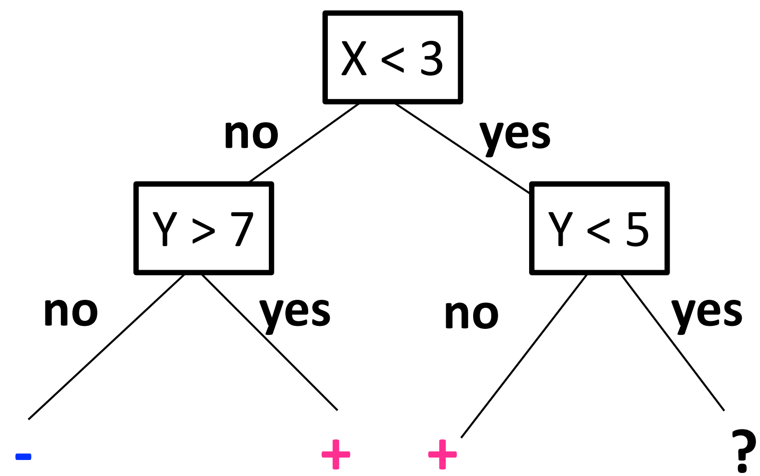




Numeric attributes and decision boundaries

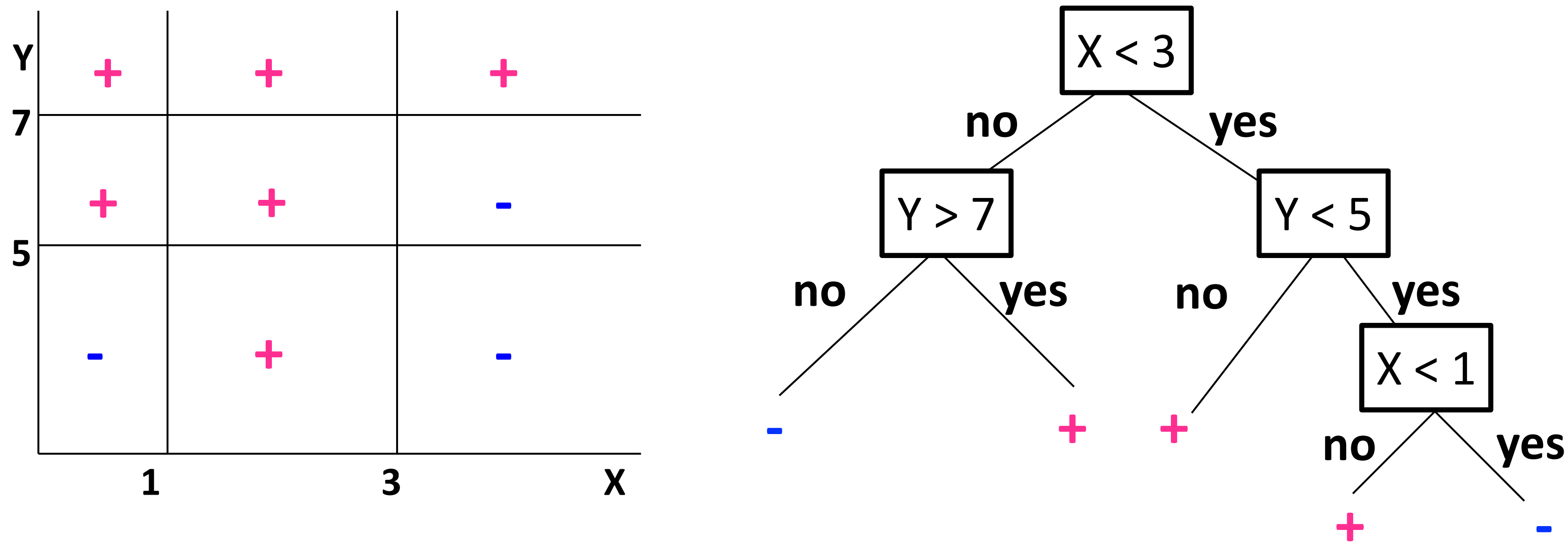
Dividing the features space into axis-parallel rectangles





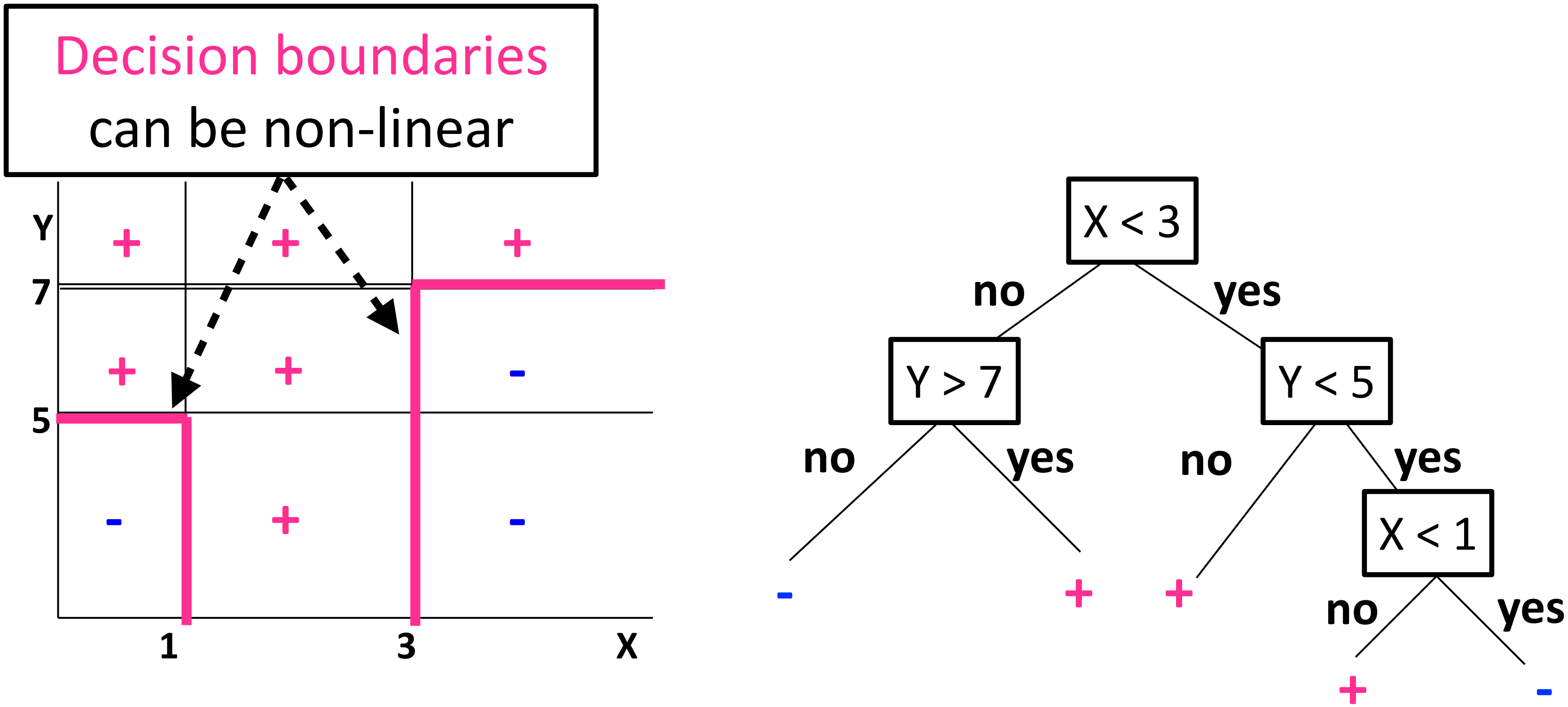
Numeric attributes and decision boundaries

Dividing the features space into axis-parallel rectangles



Numeric attributes and decision boundaries

Dividing the features space into axis-parallel rectangles



Decision Trees

TAKEAWAYS

Summary: decision trees

Decision trees can represent any Boolean function

A way to represent a lot of data

A natural representation (think playing 20 questions)

Predicting with a decision tree classifier is easy

Summary: decision trees

Decision trees can represent any Boolean function

A way to represent a lot of data

A natural representation (think playing 20 questions)

Predicting with a decision tree classifier is easy

Clearly, given a dataset, there are many decision trees that can represent it

Learning a good representation from data is the challenge

Problem setting for decision tree learning

Set of possible instances, X

- Each instance $\mathbf{x} \in X$ is a feature vector
- E.g., <shape=square, color=red>

Set of possible labels, Y

- Y is discrete-valued

Unknown target function, $f : X \rightarrow Y$

Set of function hypotheses, $H = \{h \mid h : X \rightarrow Y\}$

- Each hypothesis h is a decision tree
- Trees sort $\mathbf{x}$ to leaf which assigns $y \in Y$