Python for Data Science

Machine Learning in Python: Classification

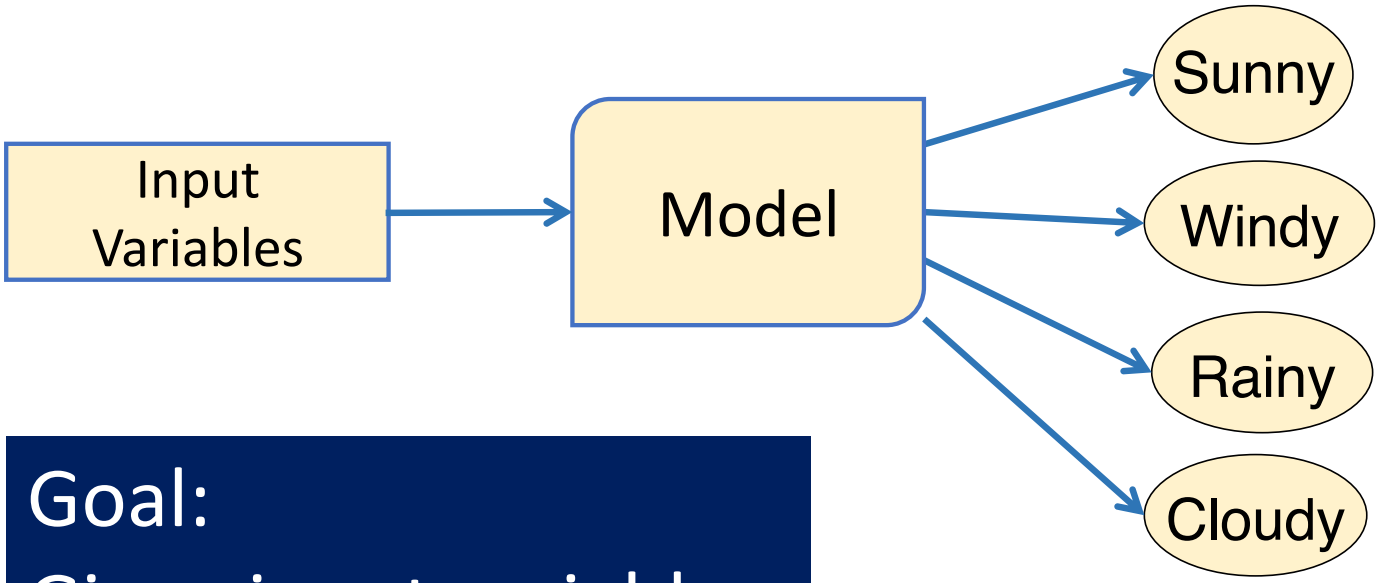
Dr. Ilkay Altintas and Dr. Leo Porter

Twitter: #UCSDpython4DS

By the end of this video, you should be able to:

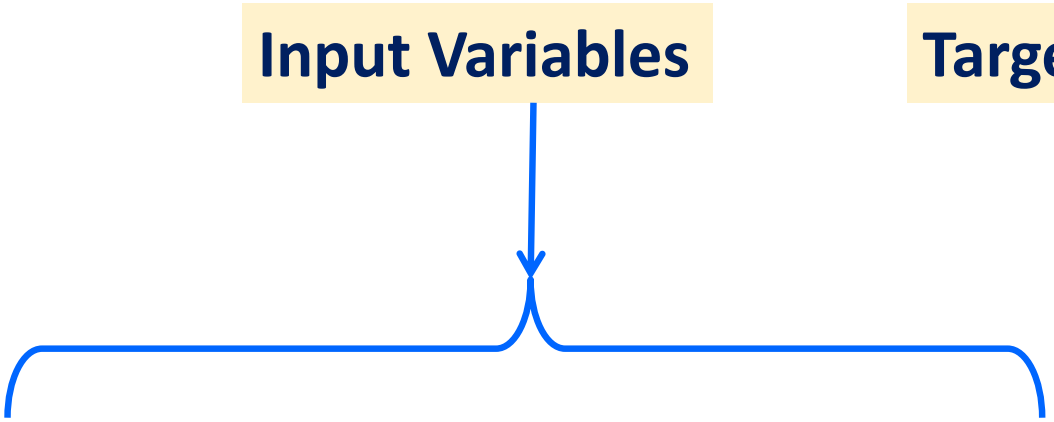
- Define what classification is
- Discuss whether classification is supervised or unsupervised
- Describe how binomial classification differs from multinomial classification

Target variable is categorical



Goal:
Given input variables, predict category

Data for Classification



Temperature	Humidity	Wind Speed	Weather
79	48	2.7	Sunny
60	80	3.8	Rainy
68	45	17.9	Windy
57	77	4.2	Cloudy

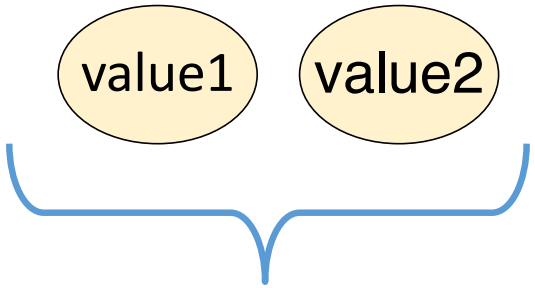
Classification is Supervised

Target **Label** **Output**

Class Variable **Class** **Category**

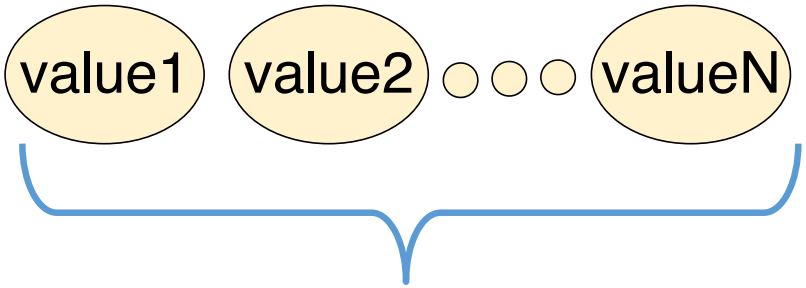
Temperature	Humidity	Wind Speed	Weather
79	48	2.7	Sunny
60	80	3.8	Rainy
68	45	17.9	Windy
57	77	4.2	Cloudy

Binary Classification



Target has two values

Multi-class Classification



Target has > 2 values

Classification Examples

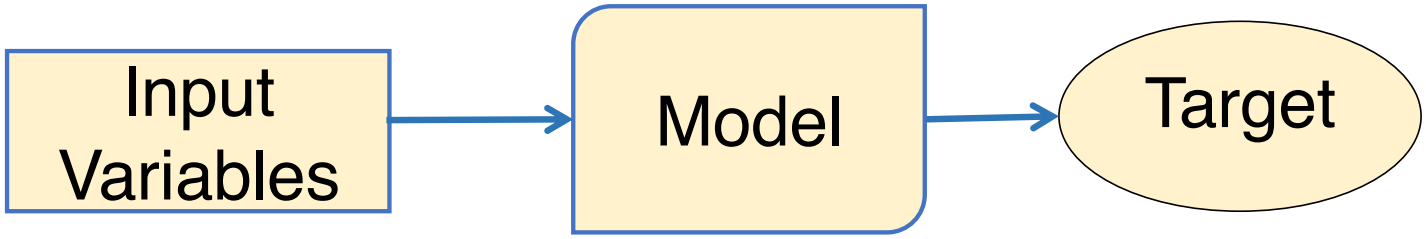
Binary Classification

- Will it rain tomorrow or not?
- Is this transaction legitimate or fraudulent

Multi-Class Classification

- What type of product will this customer buy?
- Is this tweet positive, negative, or neutral

- Predict category from input variables
- Classification is a supervised task
- Target variable is categorical



Python for Data Science

Machine Learning in Python: Building and Applying a Classification Model

Dr. Ilkay Altintas and Dr. Leo Porter

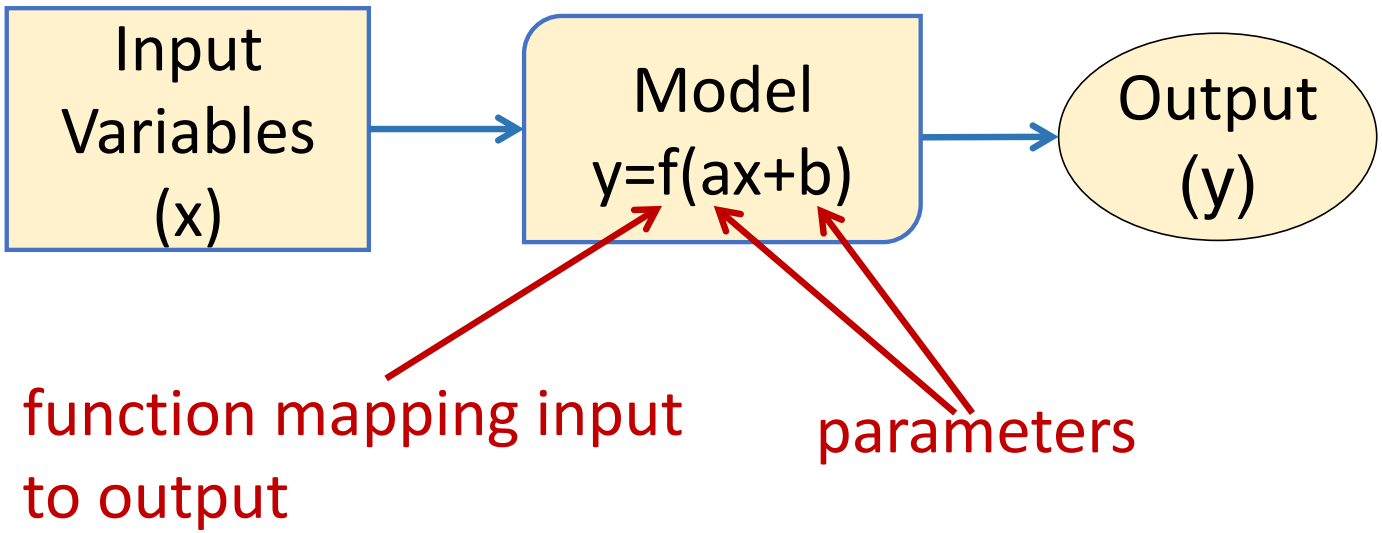
Twitter: #UCSDpython4DS

By the end of this video, you should be able to:

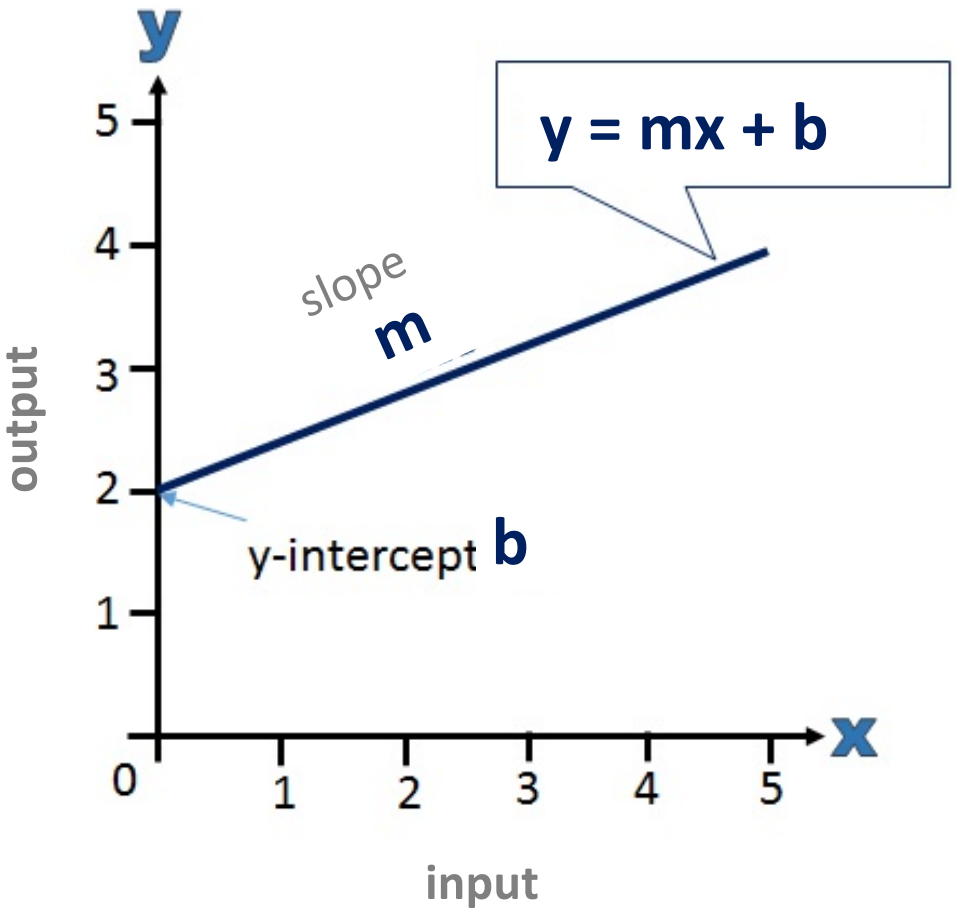
- Discuss what building a classification model means
- Explain the difference between building and applying a model
- Summarize why the parameters of a model need to be adjusted
- Describe the goal of a classification algorithm
- Name some common algorithms for classification

What is a Machine Learning Model?

- A mathematical model with parameters that map input to output

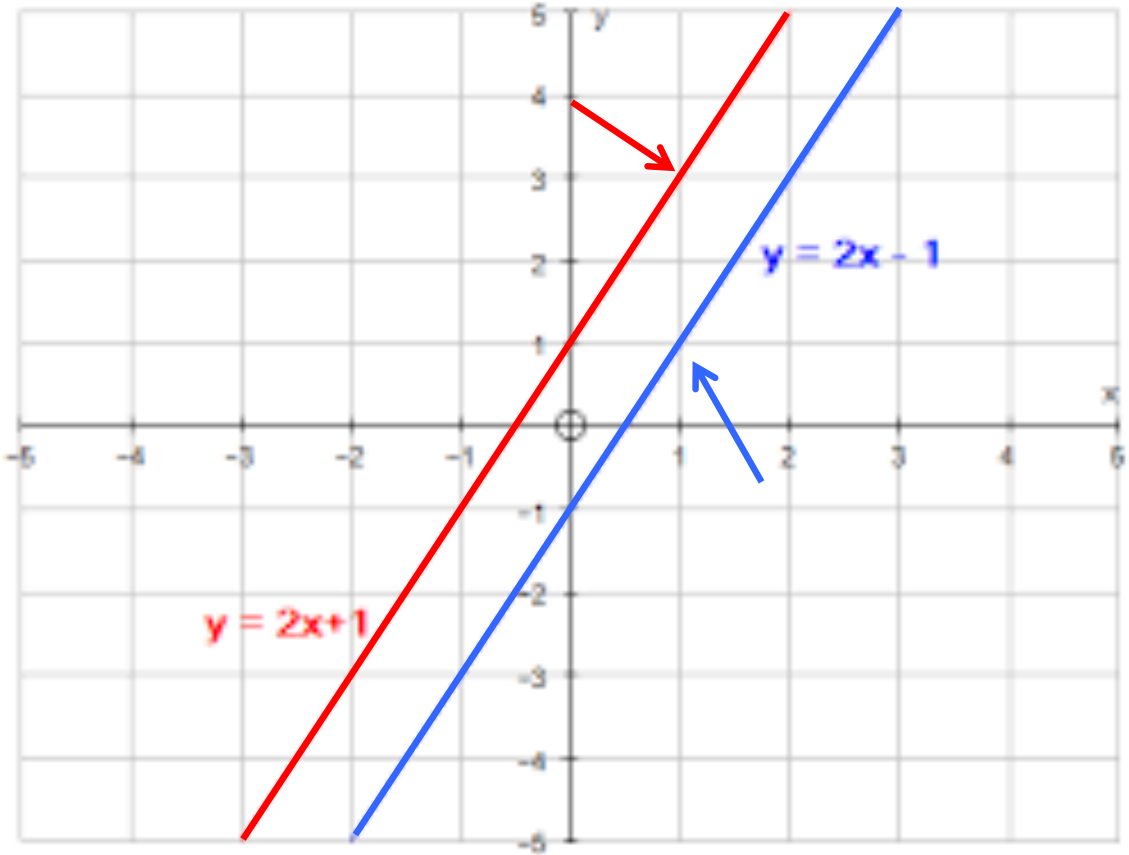


Example of Model



Adjusting Model Parameters

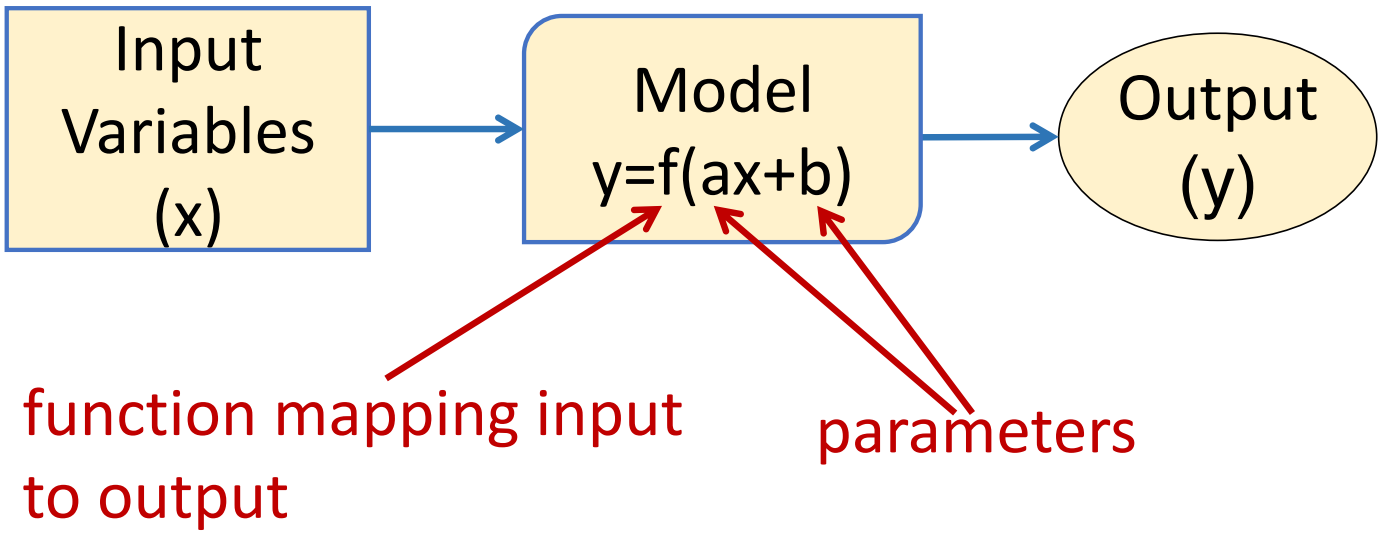
slope m = 2
y-intercept b = +1
x=1 => y=2*1+1= 3



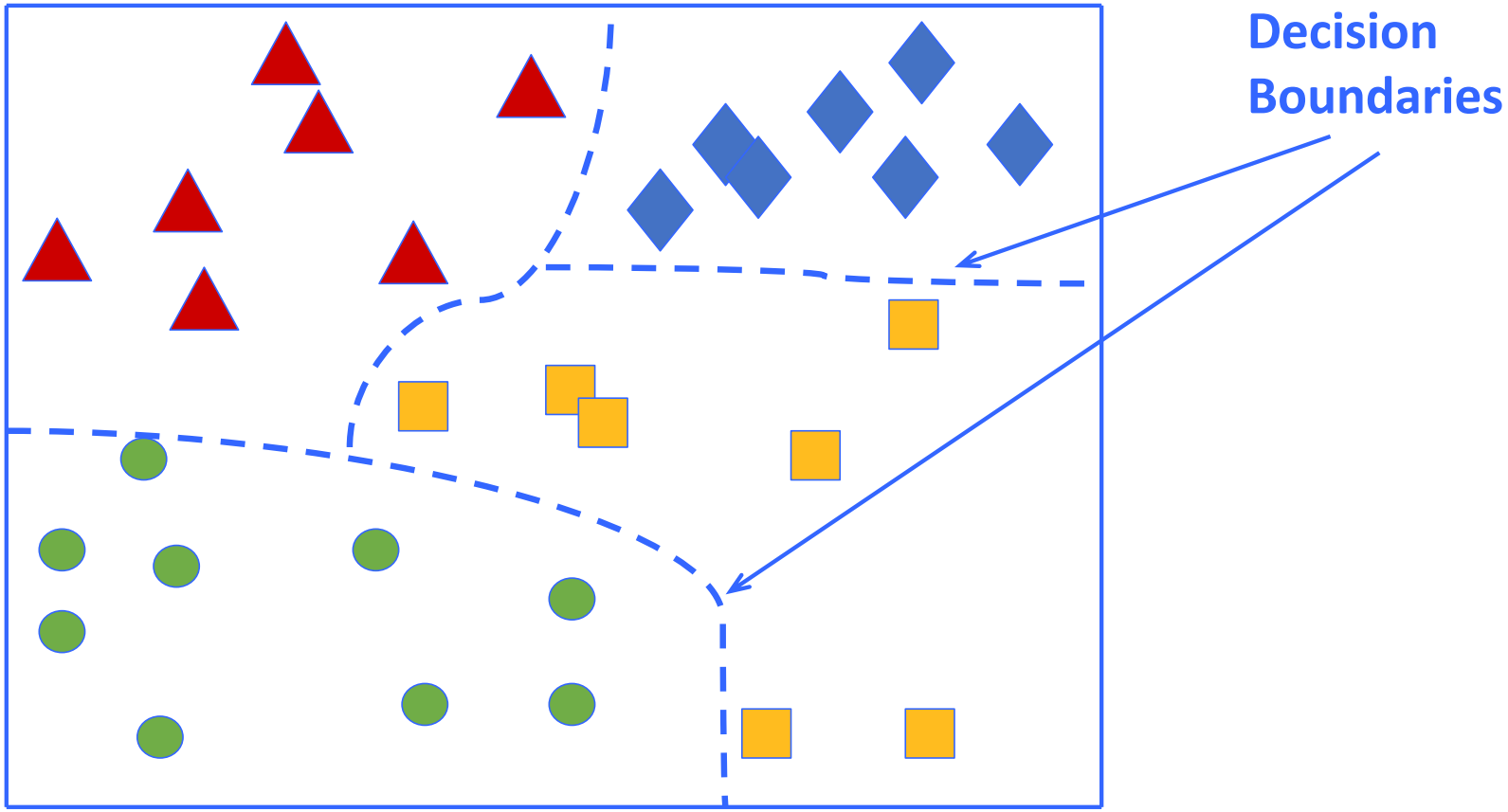
slope m = 2
y-intercept b = -1
x=1 => y=2*1-1=1

Building Machine Learning Model

Model parameters are adjusted during model training to change input-output mapping.



Building Classification Model

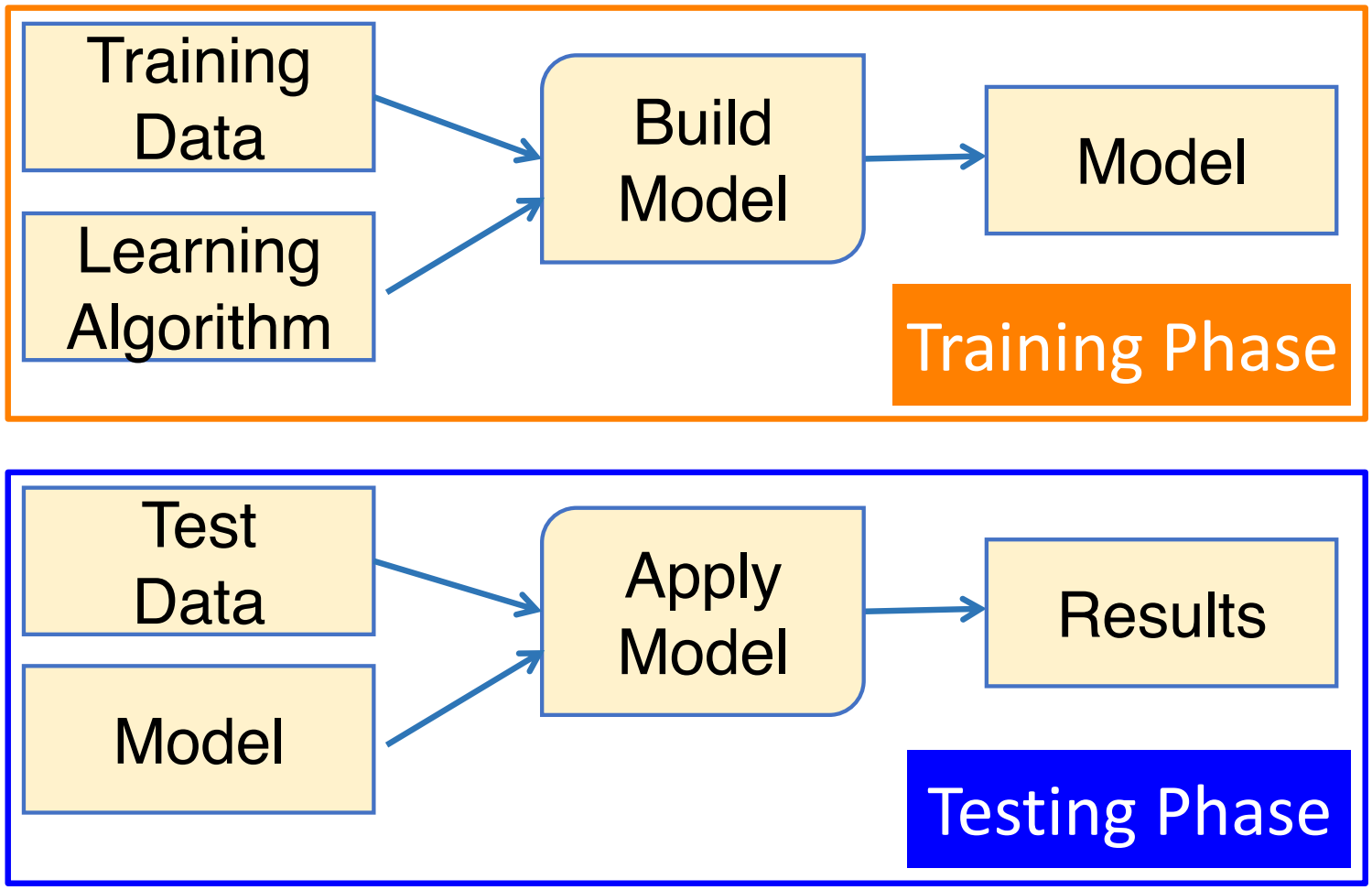


Building vs. Applying Model

- Training Phase
  - Adjust model parameters
  - Use training data

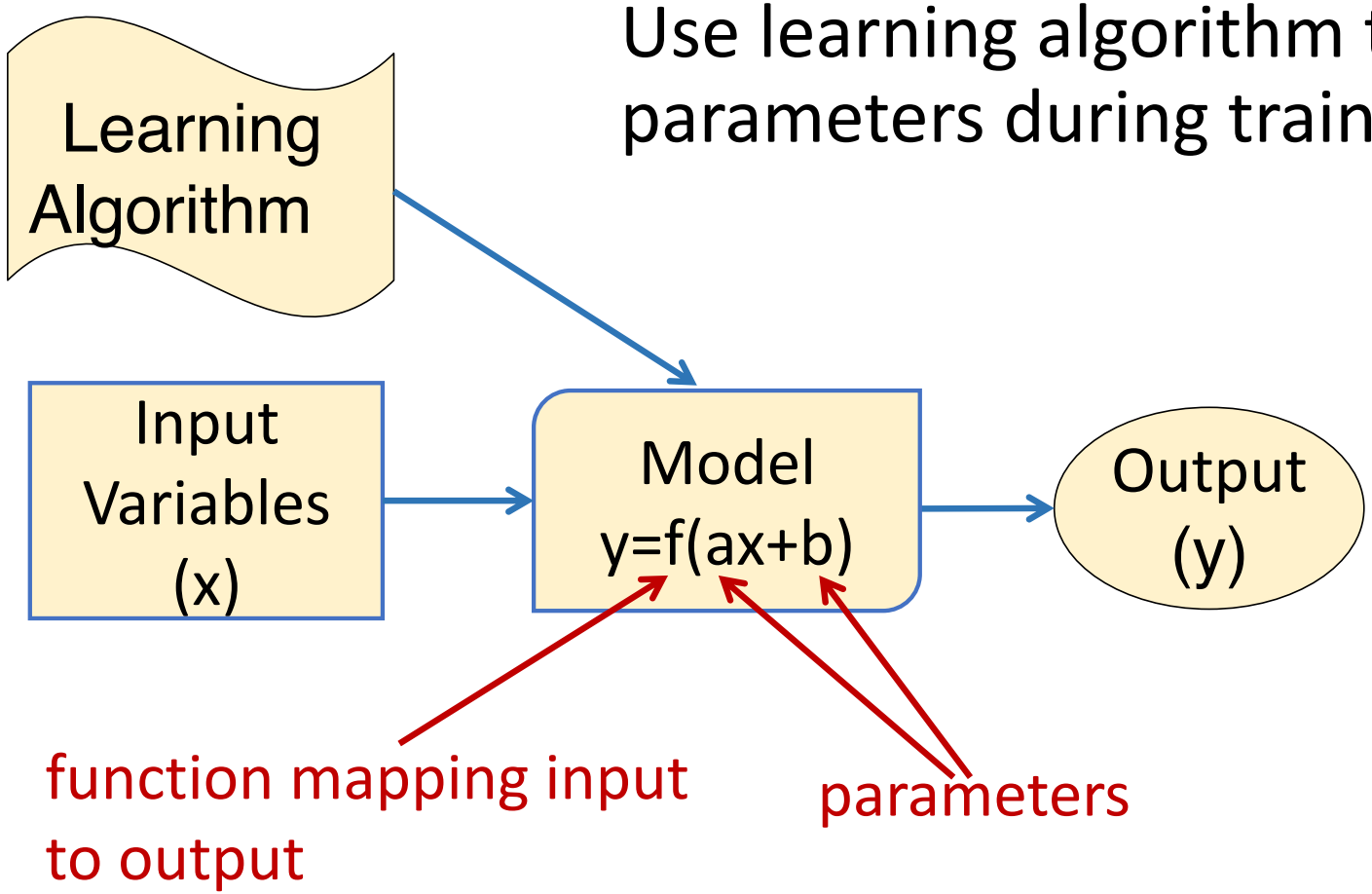
- Testing Phase
  - Apply learned model
  - Use new data

Building vs. Applying Model



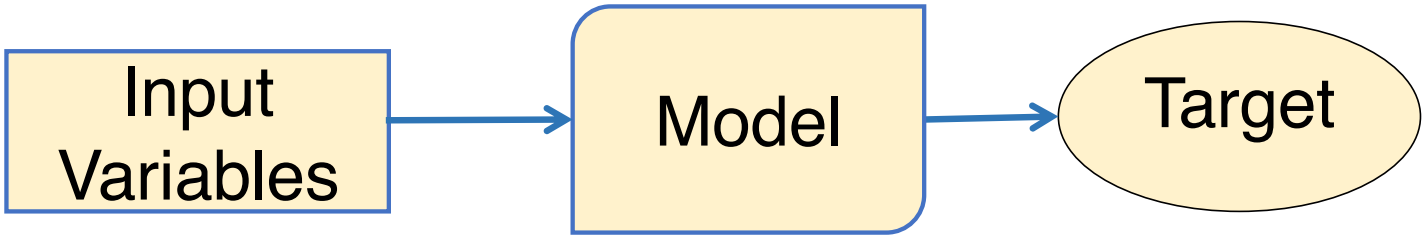
Building a Classification Model

Use learning algorithm to model parameters during training.

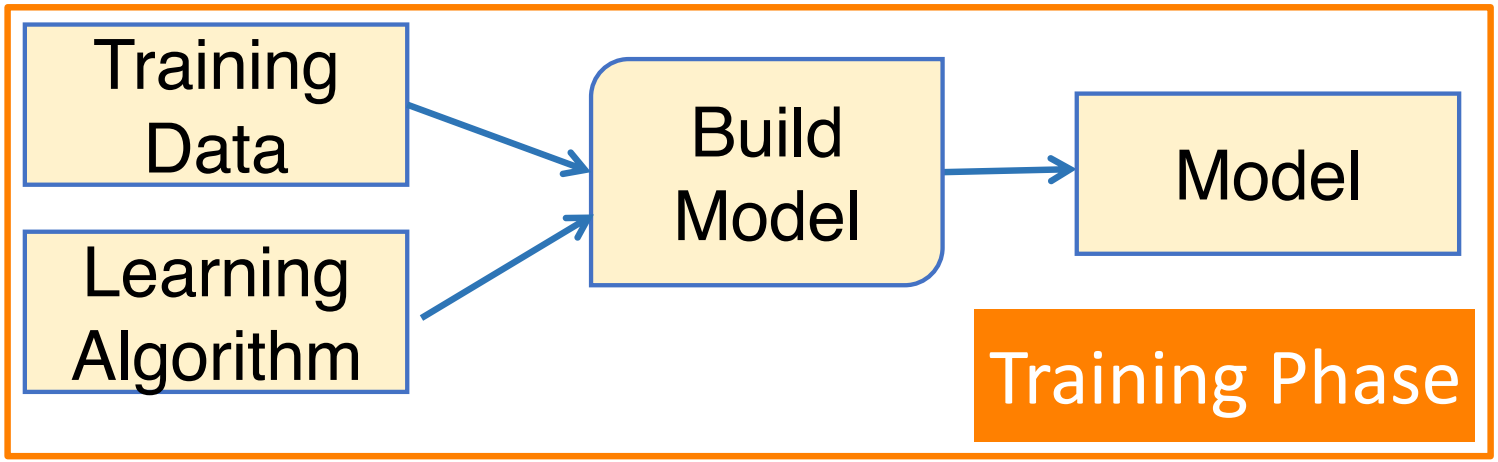


Classification

- Task: Predict category from input variables
- Goal: Match model outputs to targets (desired outputs)



Learning Algorithm



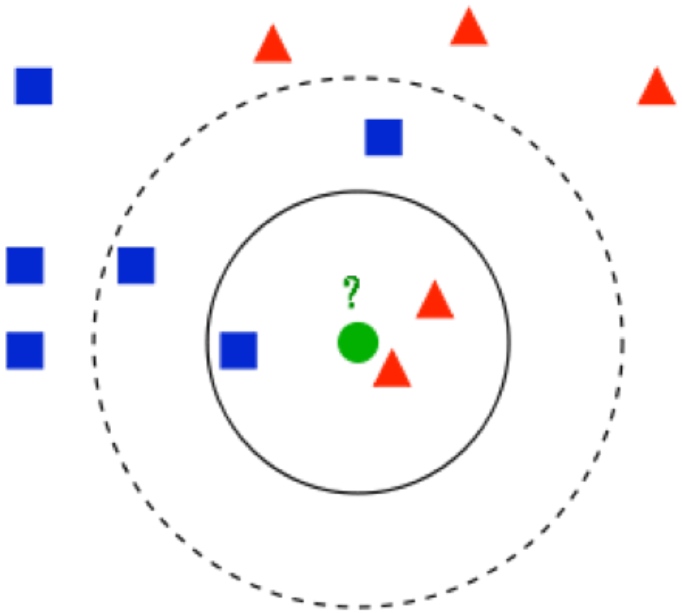
- Learning algorithm used to adjust model's parameters

Common Classification Algorithms

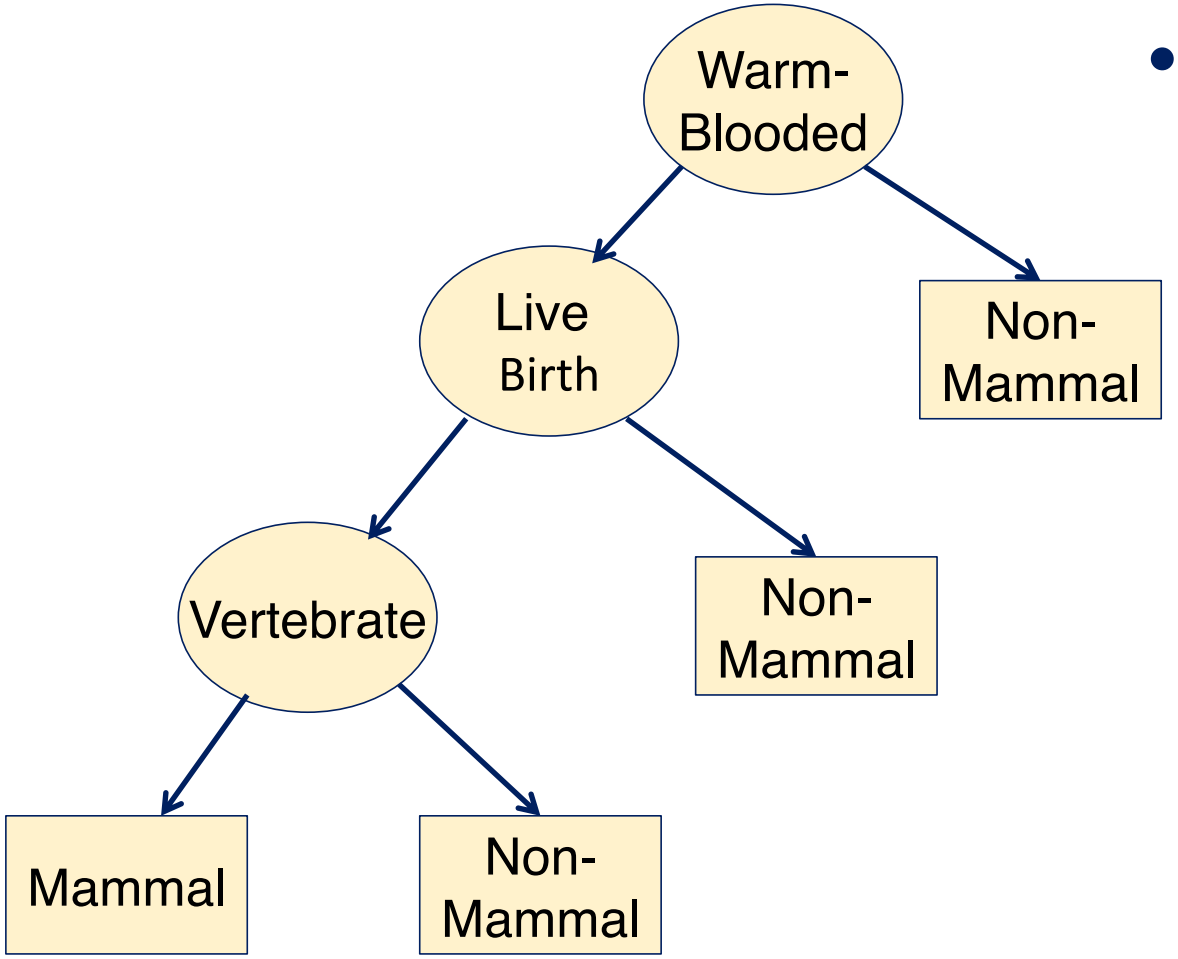
- kNN
- Decision Tree
- Naïve Bayes

kNN Overview

- Classify sample by looking at its closest neighbors



Decision Tree Overview



- Tree captures multiple classification decision paths

Naïve Bayes Overview

- Probabilistic approach to classification

Bayes' Theorem:

$$P(A|B) = \frac{P(B|A)\ P(A)}{P(B)}$$

Common Classification Algorithms

- kNN
- Decision Tree
- Naïve Bayes

Python for Data Science

Machine Learning in Python: Decision Trees

Dr. Ilkay Altintas and Dr. Leo Porter

Twitter: #UCSDpython4DS

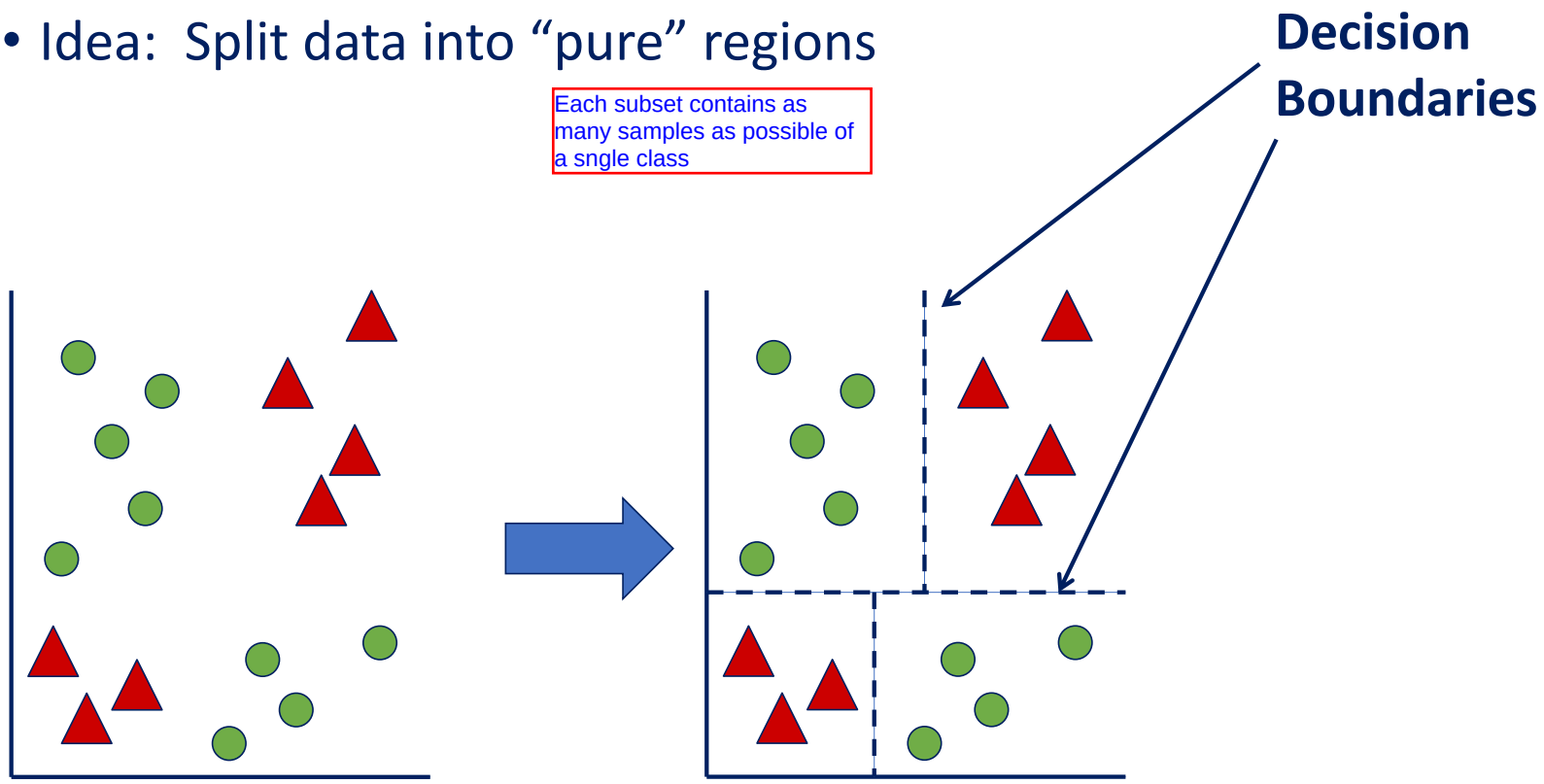
By the end of this video, you should be able to:

- Explain how a decision tree is used for classification
- Describe the process of constructing a decision tree for classification
- Interpret how a decision tree comes up with a classification decision

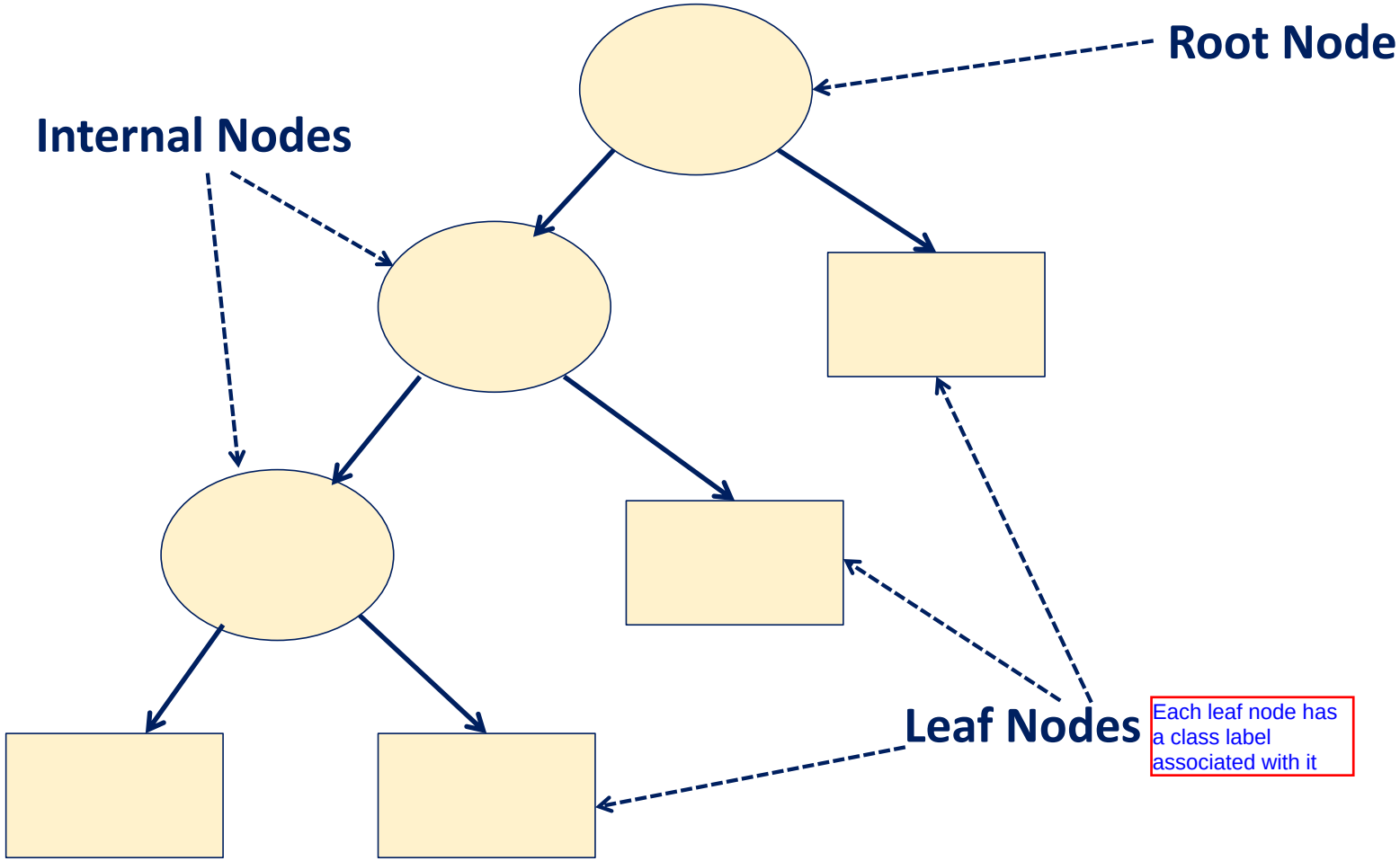
Decision Tree Overview

- Idea: Split data into “pure” regions

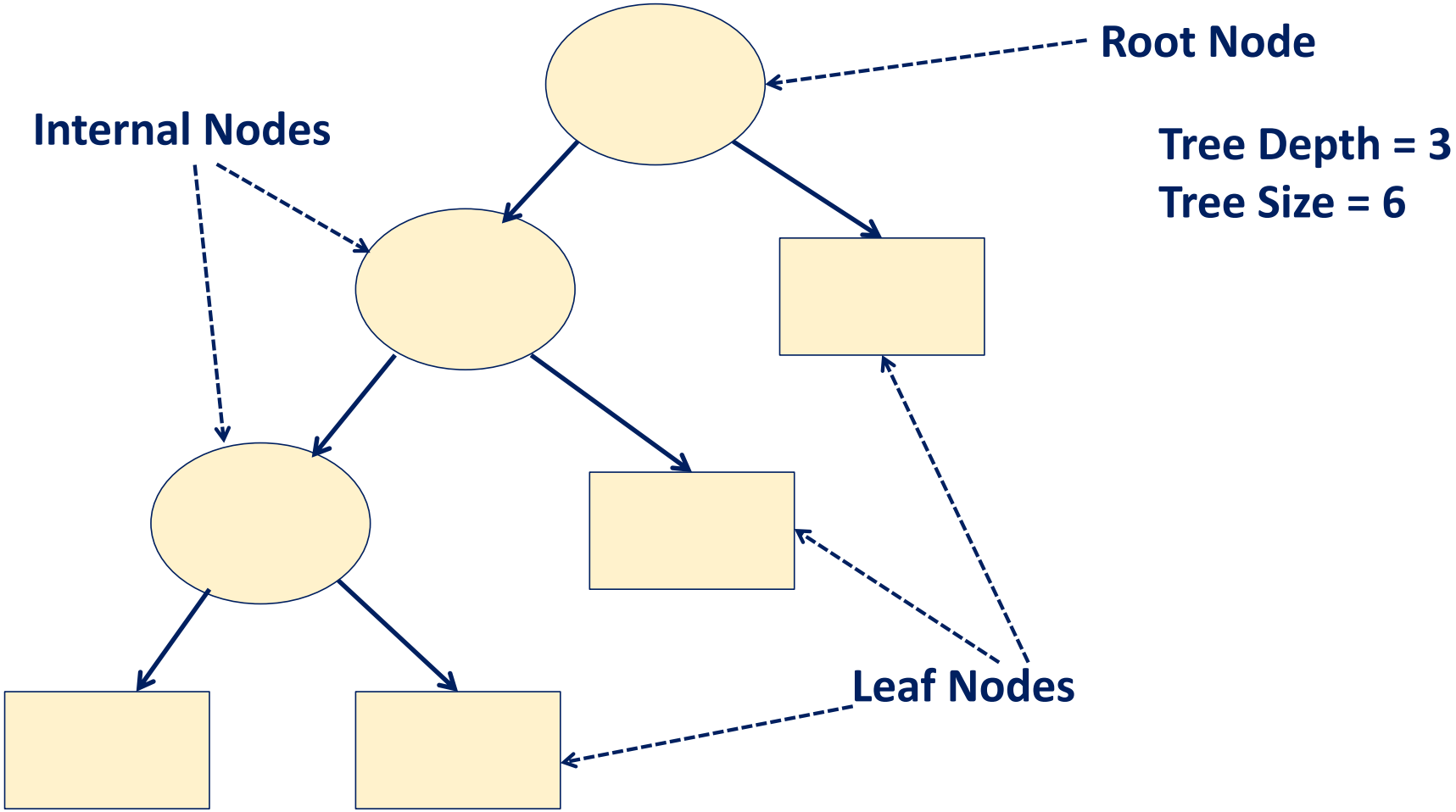
Each subset contains as many samples as possible of a sngle class



Classification Using Decision Tree

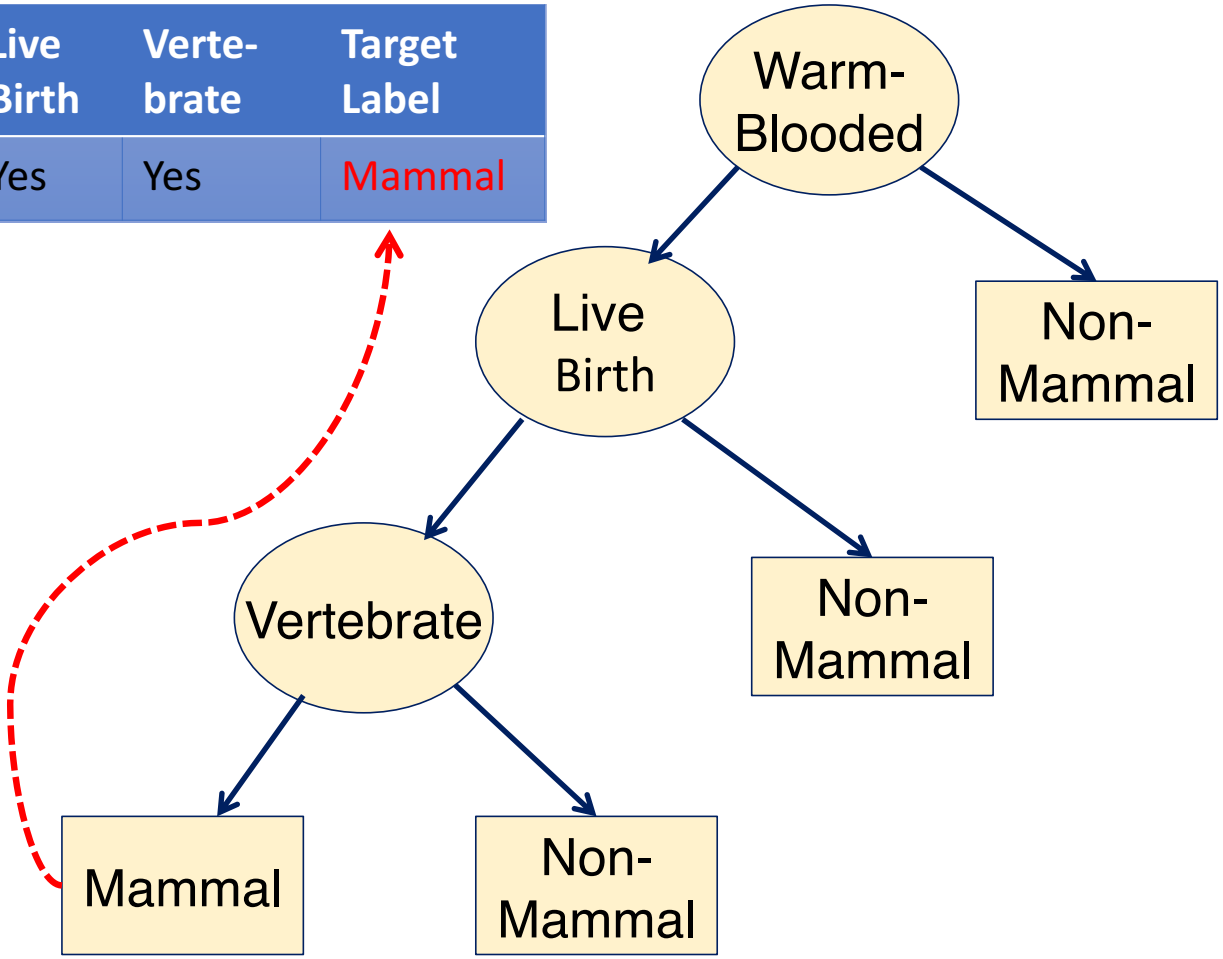


Classification Using Decision Tree



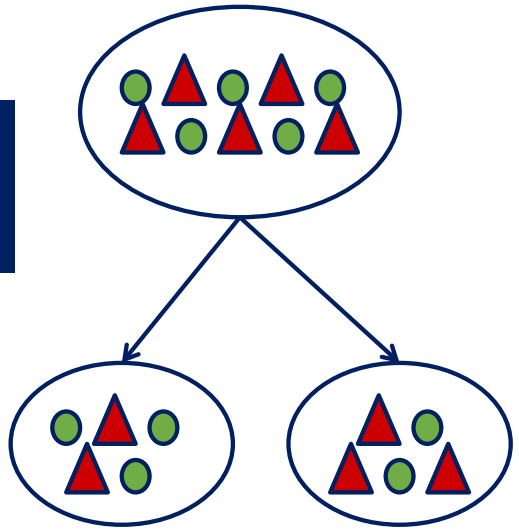
Example Decision Tree

Warm-Blooded	Live Birth	Verte-brate	Target Label
Yes	Yes	Yes	Mammal



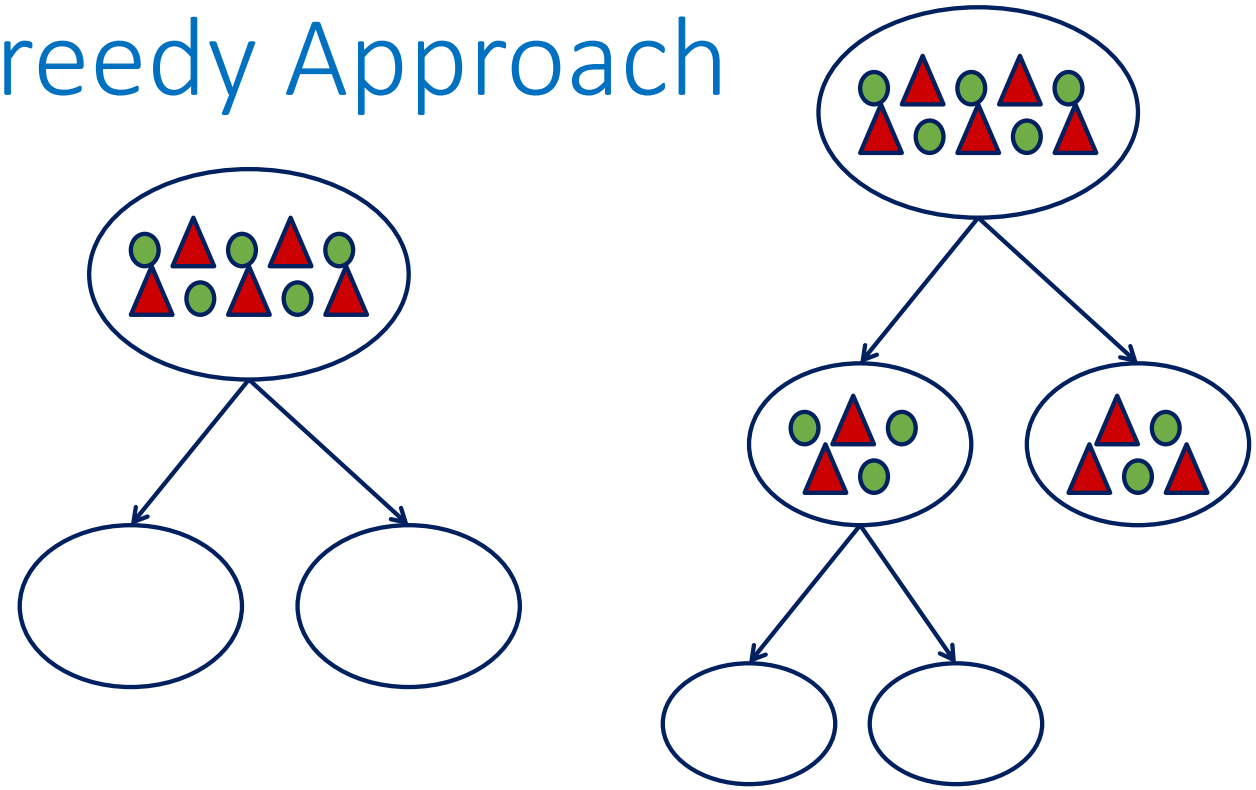
Constructing Decision Tree

Tree Induction



- Start with all samples at a node.
- Partition samples based on input to create purest subsets.
- Repeat to partition data into successively purer subsets.

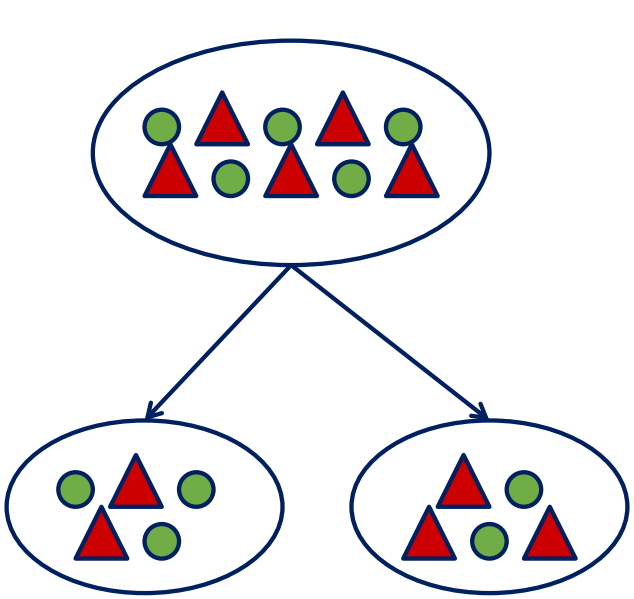
Greedy Approach



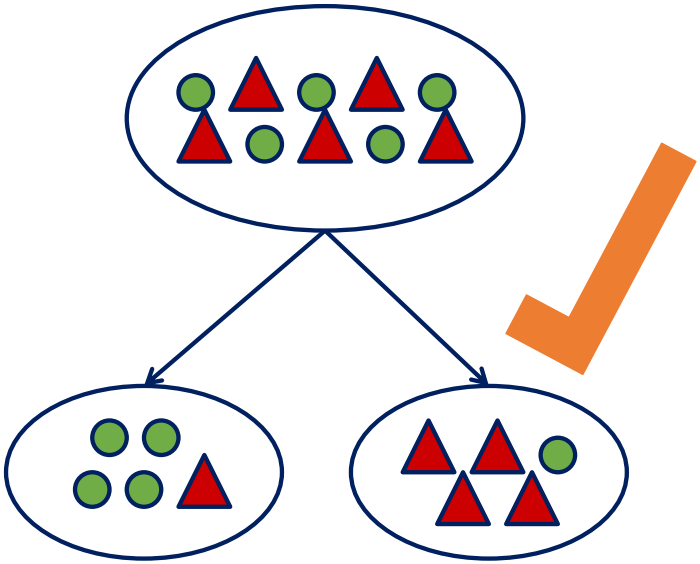
What's the best way to split the current node?

How to Determine Best Split?

Want subsets to be as homogeneous as possible



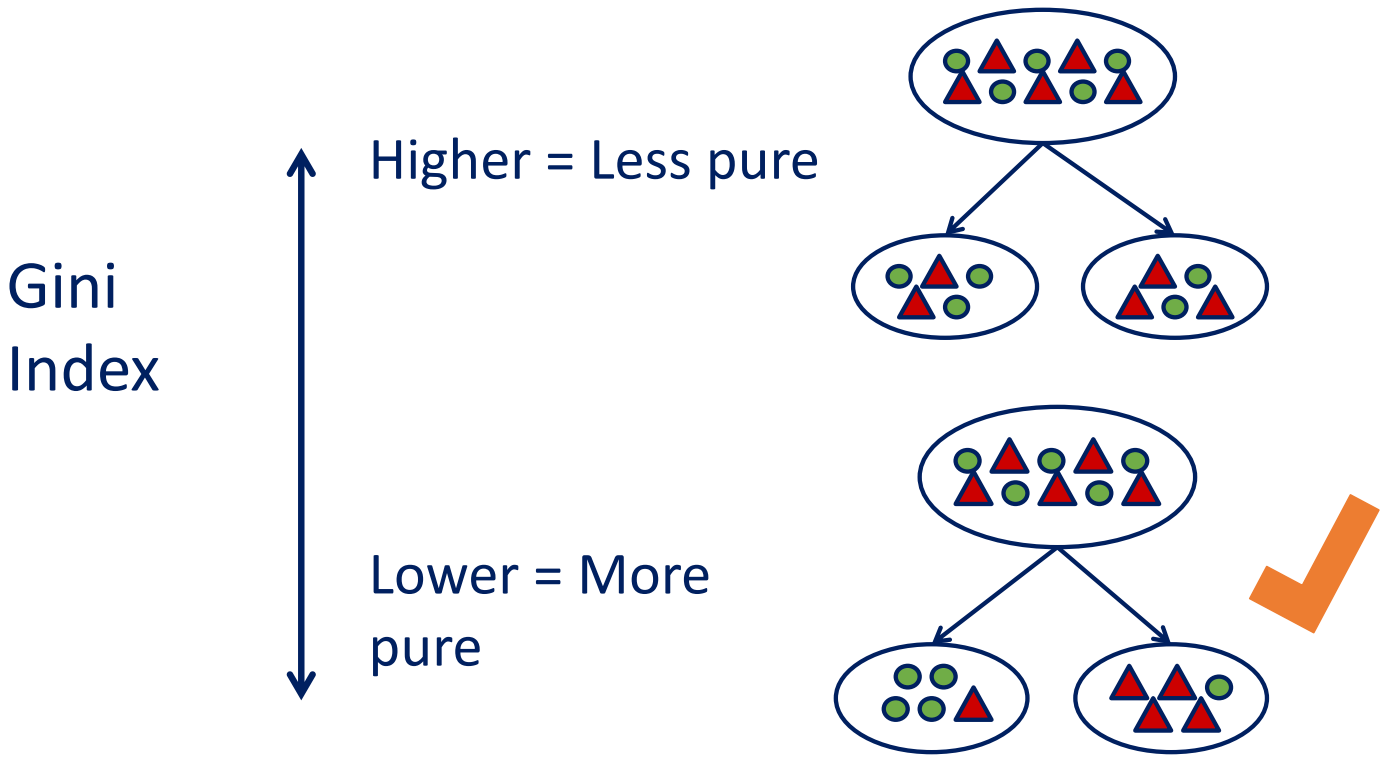
Less homogeneous = More pure



More homogeneous = More pure

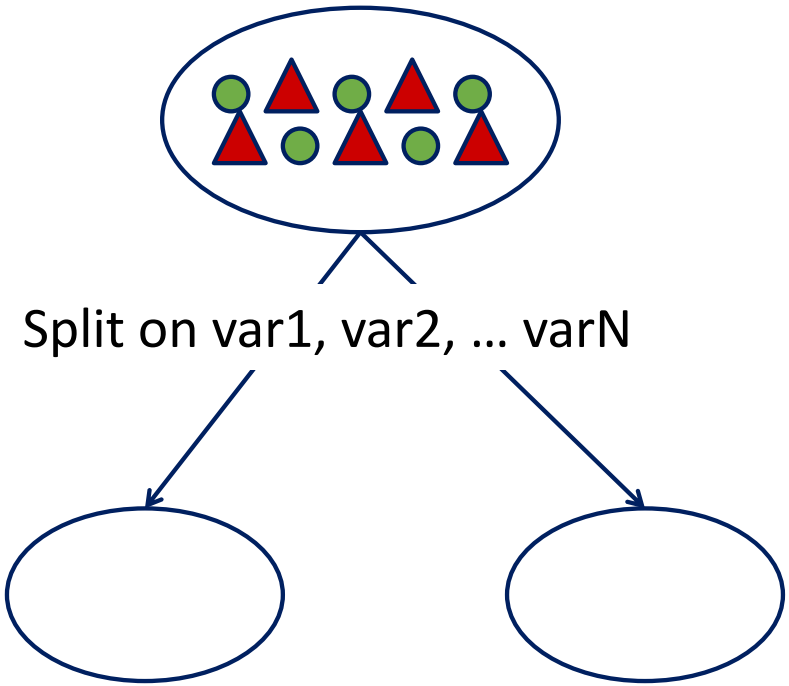
Impurity Measure

- To compare different ways to split data in a node



What Variable to Split On?

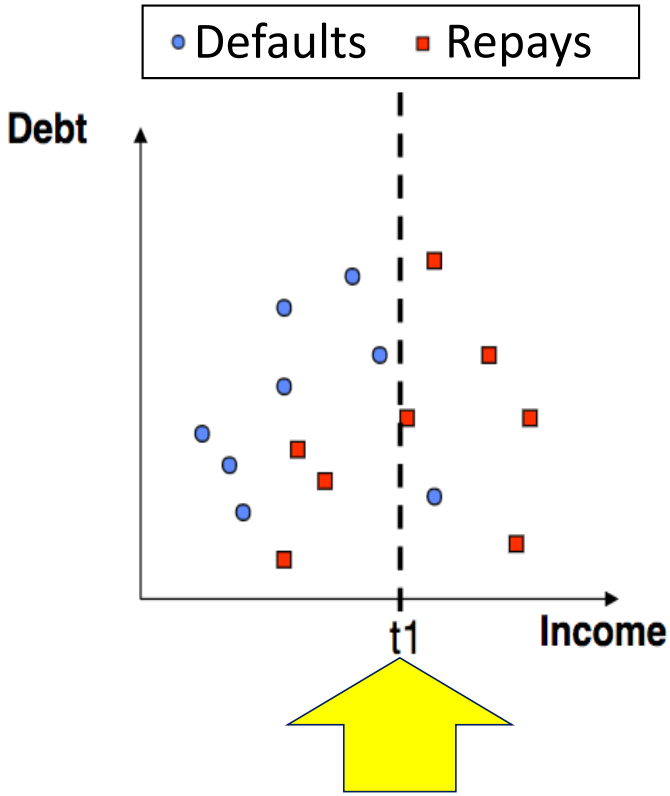
- Splits on all variables are tested

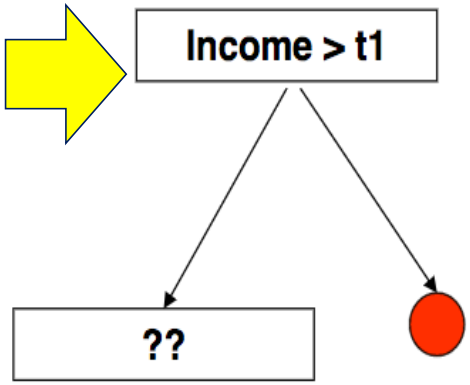


When to Stop Splitting a Node?

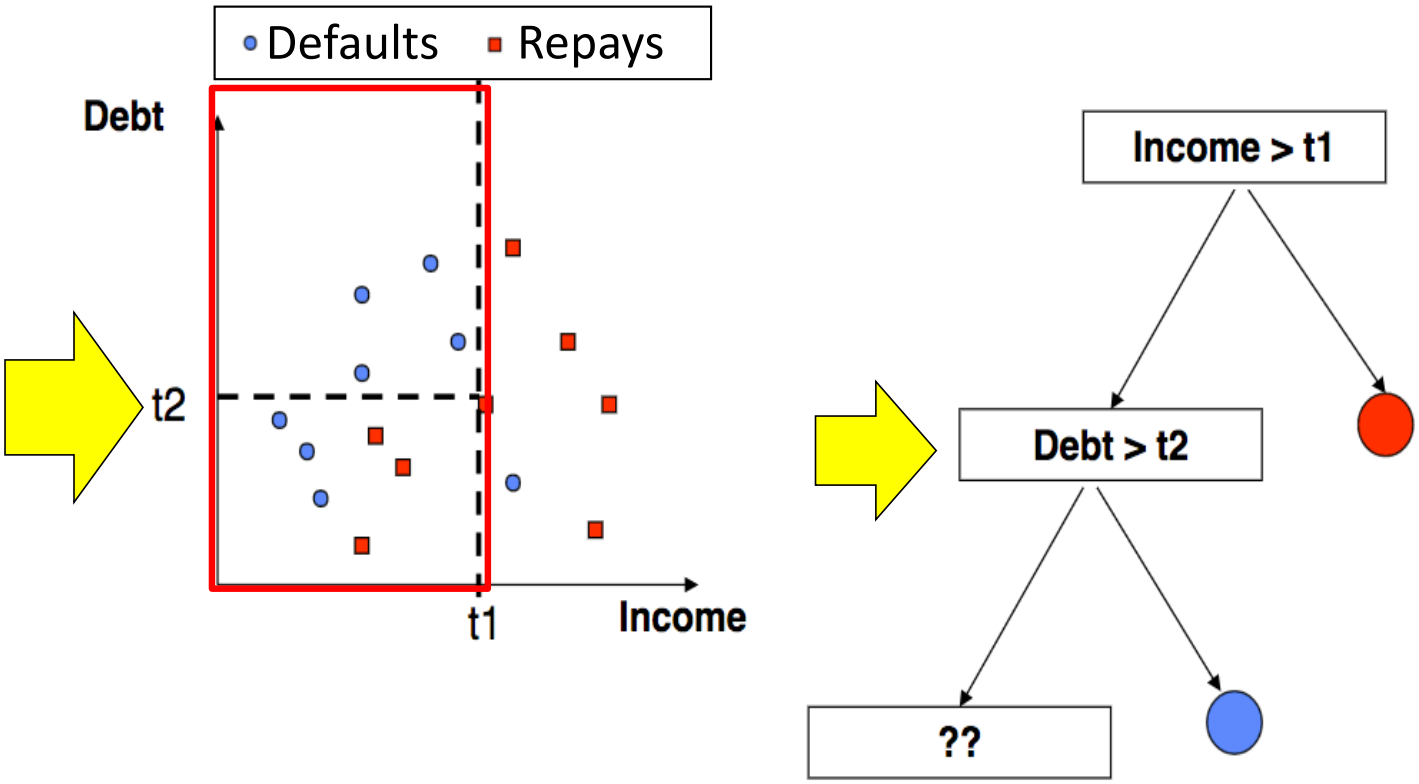
- All (or X% of) samples have same class label
- Number of samples in node reaches minimum
- Change in impurity measure is smaller than threshold
- Max tree depth is reached
- Others…

Tree Induction Example: Split 1

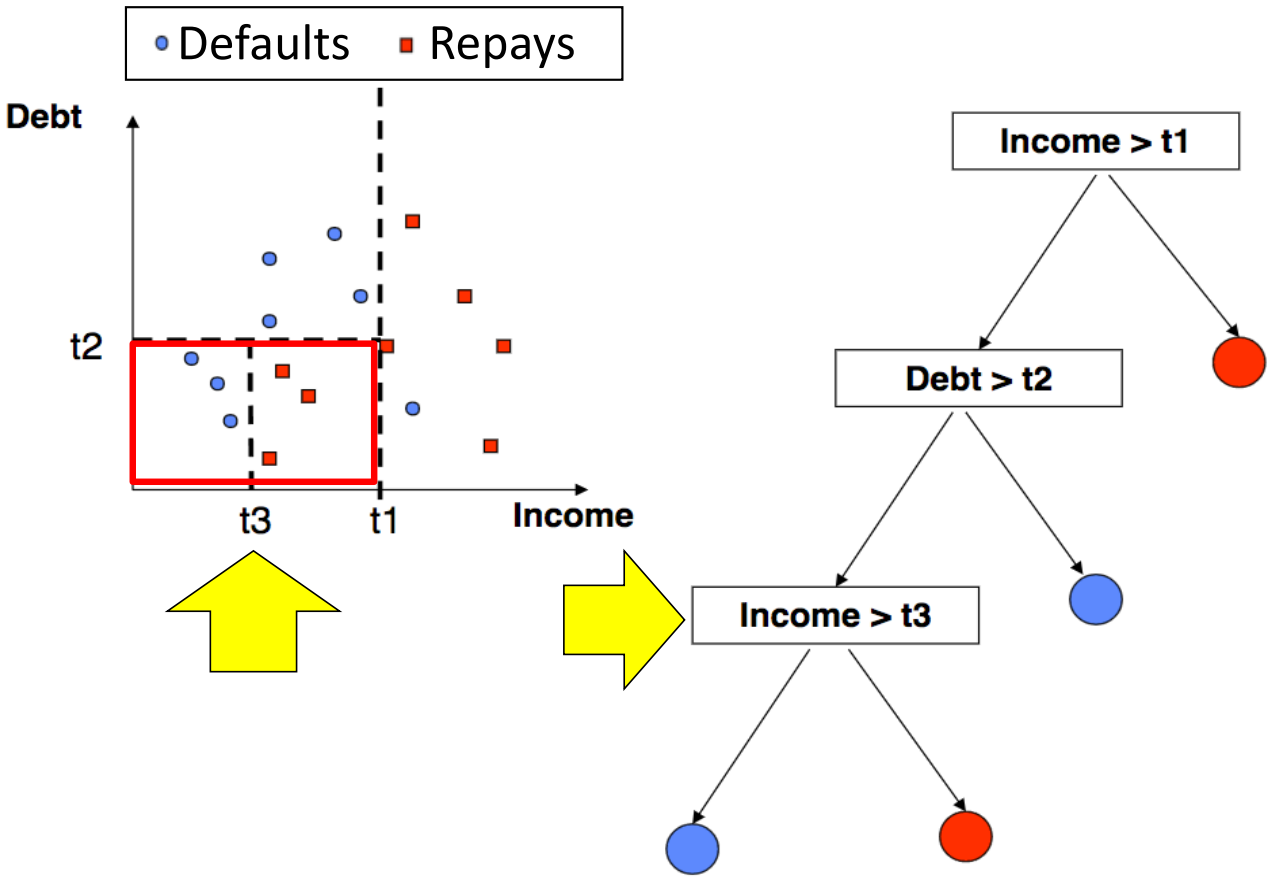




Tree Induction Example: Split 2

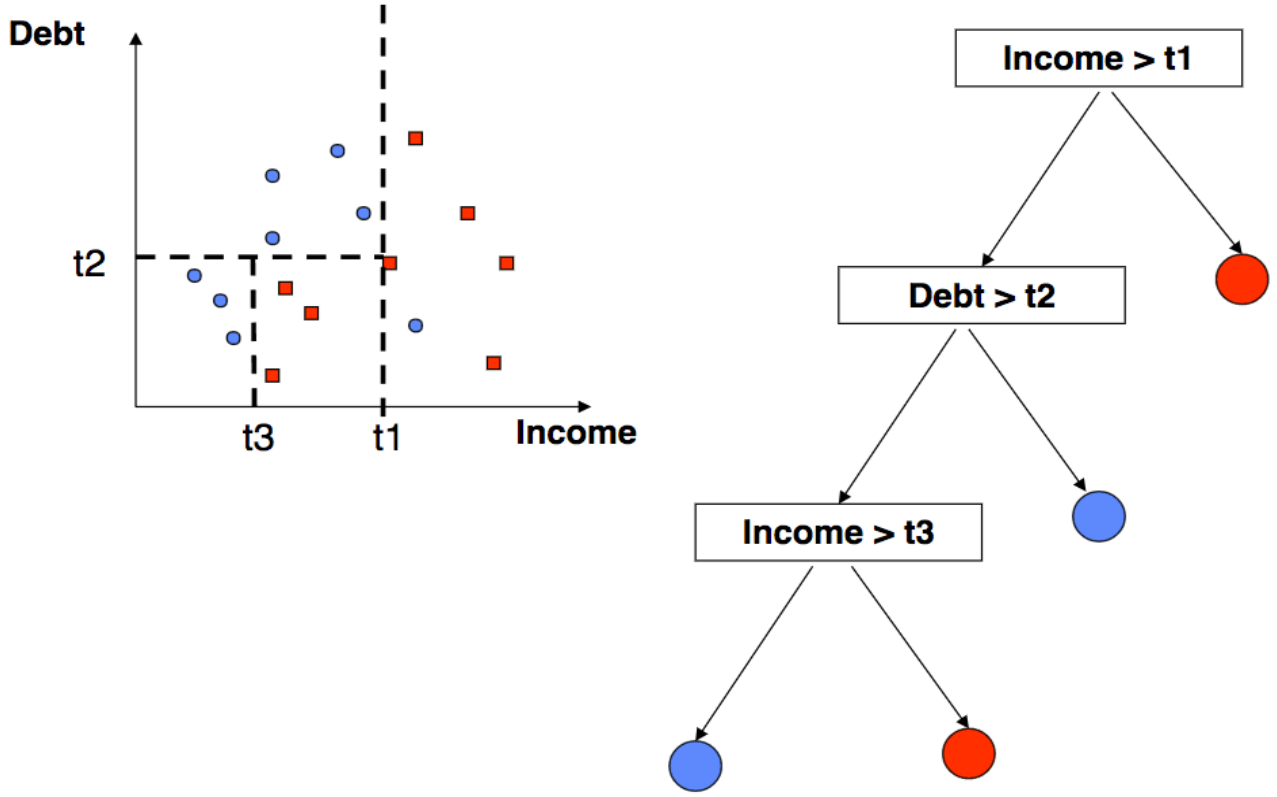


Tree Induction Example: Split 3



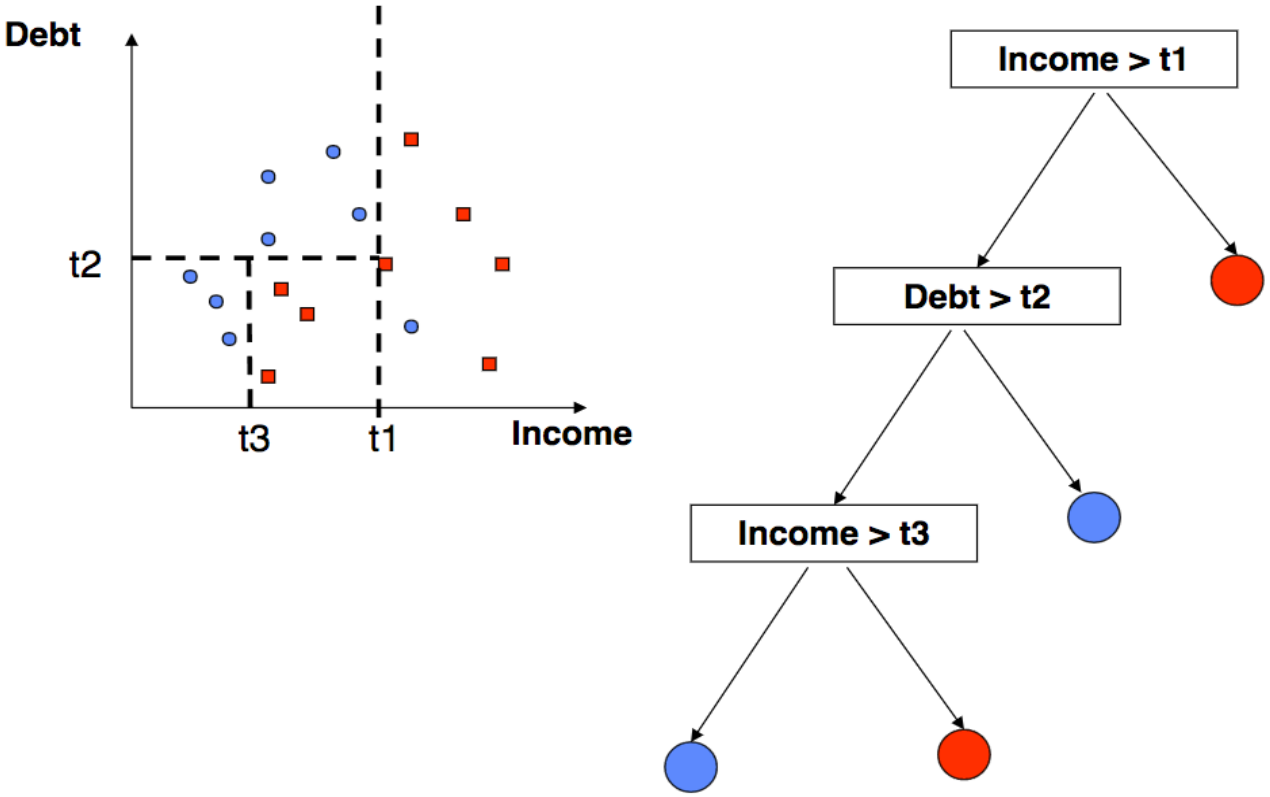
Tree Induction Example

- Resulting model



Decision Boundaries

- Rectilinear = Parallel to axes



Decision Tree for Classification

- Resulting tree is often simple and easy to interpret
- Induction is computationally inexpensive
- Greedy approach does not guarantee best solution
- Rectilinear decision boundaries

Decision Tree

the decision tree classifier uses a tree-like structure to specify a series of conditions that are tested to determine the class label for a sample.The decision tree is constructed by repeatedly splitting the data, and partitioning the data into successively more homogenous subsets.The resulting tree can often be easy to interpret.