DEPARTMENT *of*
INFORMATION SYSTEMS &
COMPUTER SCIENCE

Decision Trees

CSCI 111

Restaurant Waiting

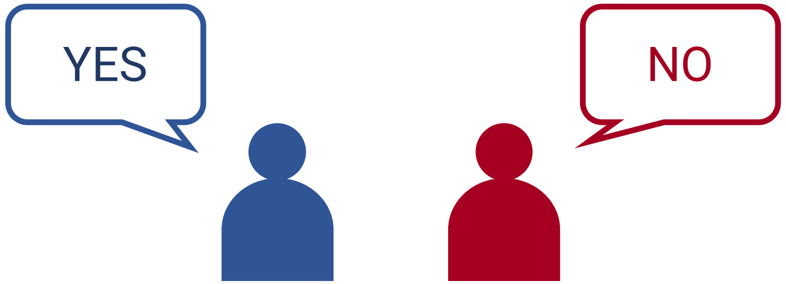
Example	Input Attributes										Output
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
$\mathbf{x}_1$	*Yes*	*No*	*No*	*Yes*	*Some*	*$$$*	*No*	*Yes*	*French*	*0–10*	$y_1 = Yes$
$\mathbf{x}_2$	*Yes*	*No*	*No*	*Yes*	*Full*	*$*	*No*	*No*	*Thai*	*30–60*	$y_2 = No$
$\mathbf{x}_3$	*No*	*Yes*	*No*	*No*	*Some*	*$*	*No*	*No*	*Burger*	*0–10*	$y_3 = Yes$
$\mathbf{x}_4$	*Yes*	*No*	*Yes*	*Yes*	*Full*	*$*	*Yes*	*No*	*Thai*	*10–30*	$y_4 = Yes$
$\mathbf{x}_5$	*Yes*	*No*	*Yes*	*No*	*Full*	*$$$*	*No*	*Yes*	*French*	*>60*	$y_5 = No$
$\mathbf{x}_6$	*No*	*Yes*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Italian*	*0–10*	$y_6 = Yes$
$\mathbf{x}_7$	*No*	*Yes*	*No*	*No*	*None*	*$*	*Yes*	*No*	*Burger*	*0–10*	$y_7 = No$
$\mathbf{x}_8$	*No*	*No*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Thai*	*0–10*	$y_8 = Yes$
$\mathbf{x}_9$	*No*	*Yes*	*Yes*	*No*	*Full*	*$*	*Yes*	*No*	*Burger*	*>60*	$y_9 = No$
$\mathbf{x}_{10}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$$$*	*No*	*Yes*	*Italian*	*10–30*	$y_{10} = No$
$\mathbf{x}_{11}$	*No*	*No*	*No*	*No*	*None*	*$*	*No*	*No*	*Thai*	*0–10*	$y_{11} = No$
$\mathbf{x}_{12}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$*	*No*	*No*	*Burger*	*30–60*	$y_{12} = Yes$

Examples for the restaurant domain.

Restaurant Waiting

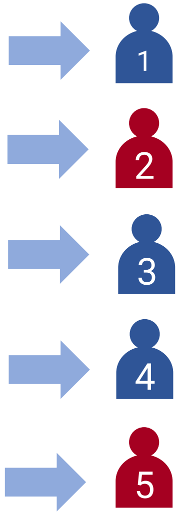
Given a set of factors (features), will we wait for a table?

- Blue: Yes (they waited)
- Red: No (they did not wait)



FEATURES | LABELS

Patrons	Hungry	Type	…	Friday	Will Wait
Some	Yes	French		No	Yes
Full	Yes	Thai		No	No
Some	No	Burger		No	Yes
Full	Yes	Thai		Yes	Yes
Full	No	French		Yes	No
• • •					• • •



Examples of scenarios and corresponding decisions

Restaurant Waiting

Goal:

Given a new scenario (set of features), predict whether they'll wait or not using a classification model

NEW DATA

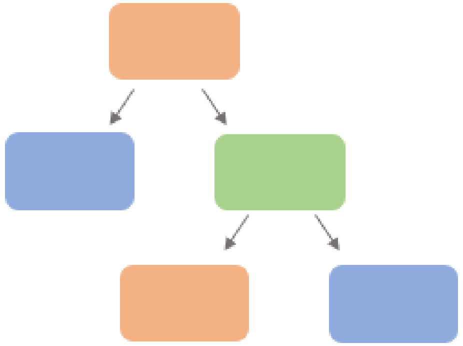
Hungry	Patron	…	Type
Yes	Some		Italian

LABEL: WILL WAIT?

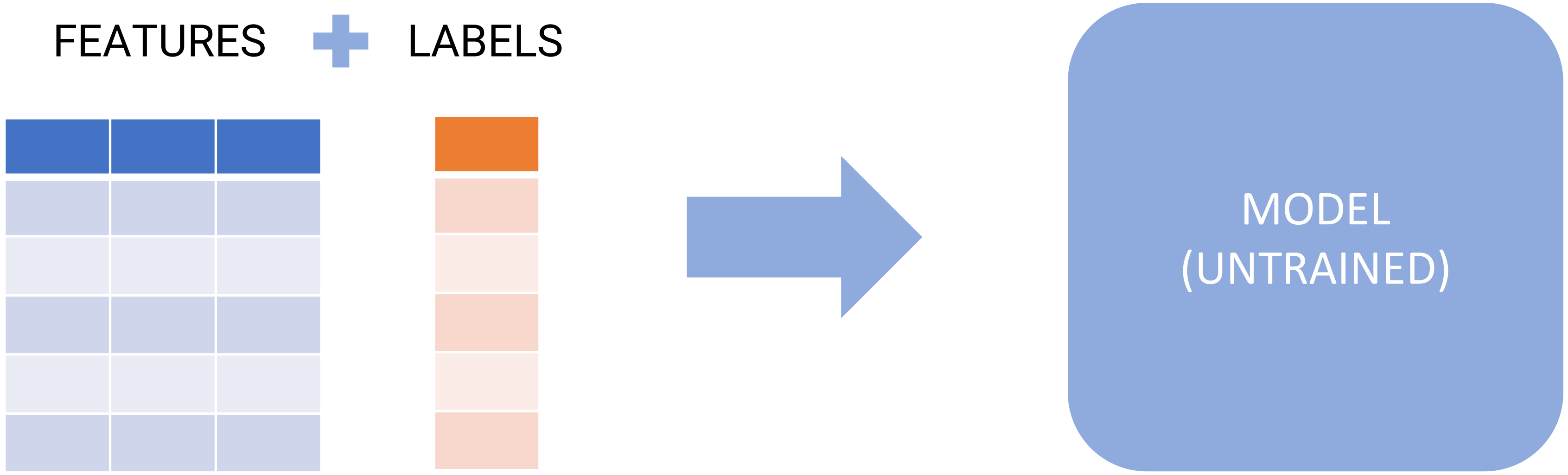
Classification: Decision Tree

- Review: a classifier learns a function from a labeled dataset
- A decision tree classifier encodes this function as a sequence of decisions or tests

TRAINED CLASSIFICATION MODEL
$y = f(X_1, X_2, X_3, \ldots X_m)$



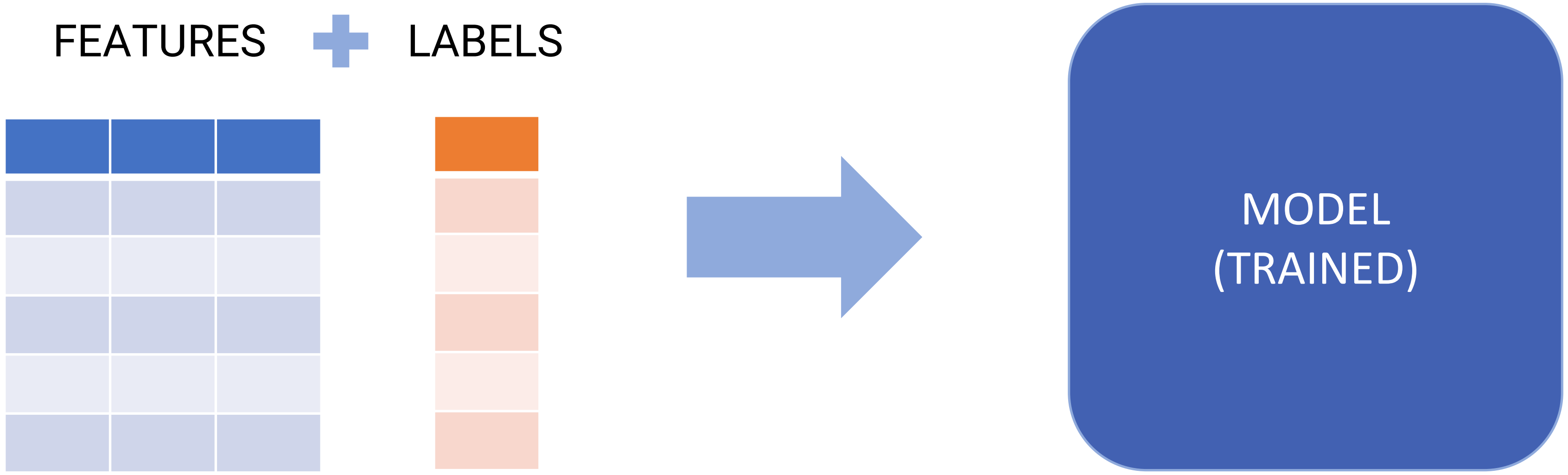
Classification: Decision Tree



TRAINING SET:

Scenarios and corresponding
decisions to wait or not

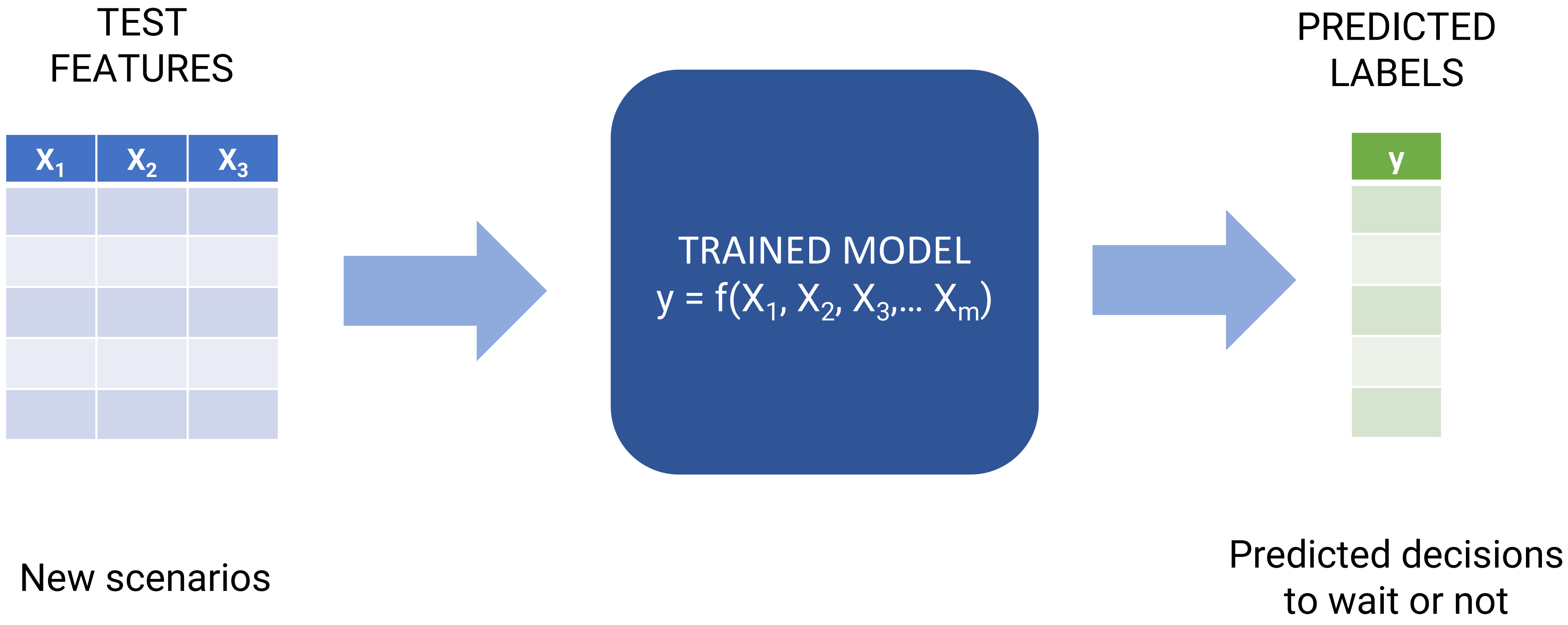
Classification: Decision Tree



TRAINING SET:

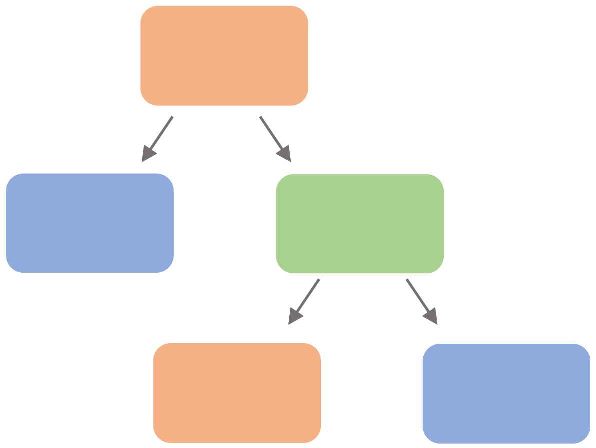
Scenarios and corresponding decisions to wait or not

Classification: Decision Tree



Classification: Decision Tree

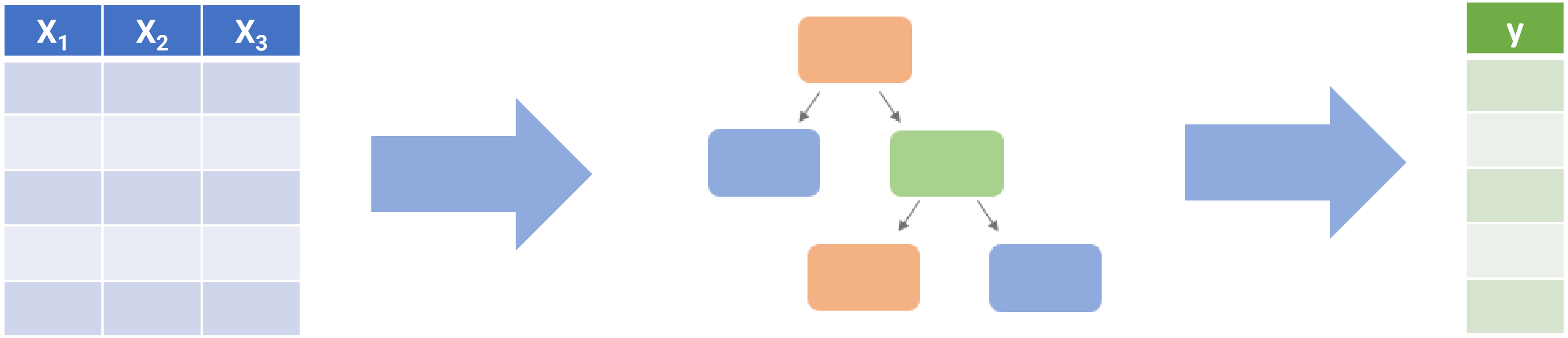
TRAINED MODEL
$y = f(X_1, X_2, X_3, \ldots X_m)$



Decision tree based on scenarios and corresponding decisions (training set)

Classification: Decision Tree

TEST
FEATURES

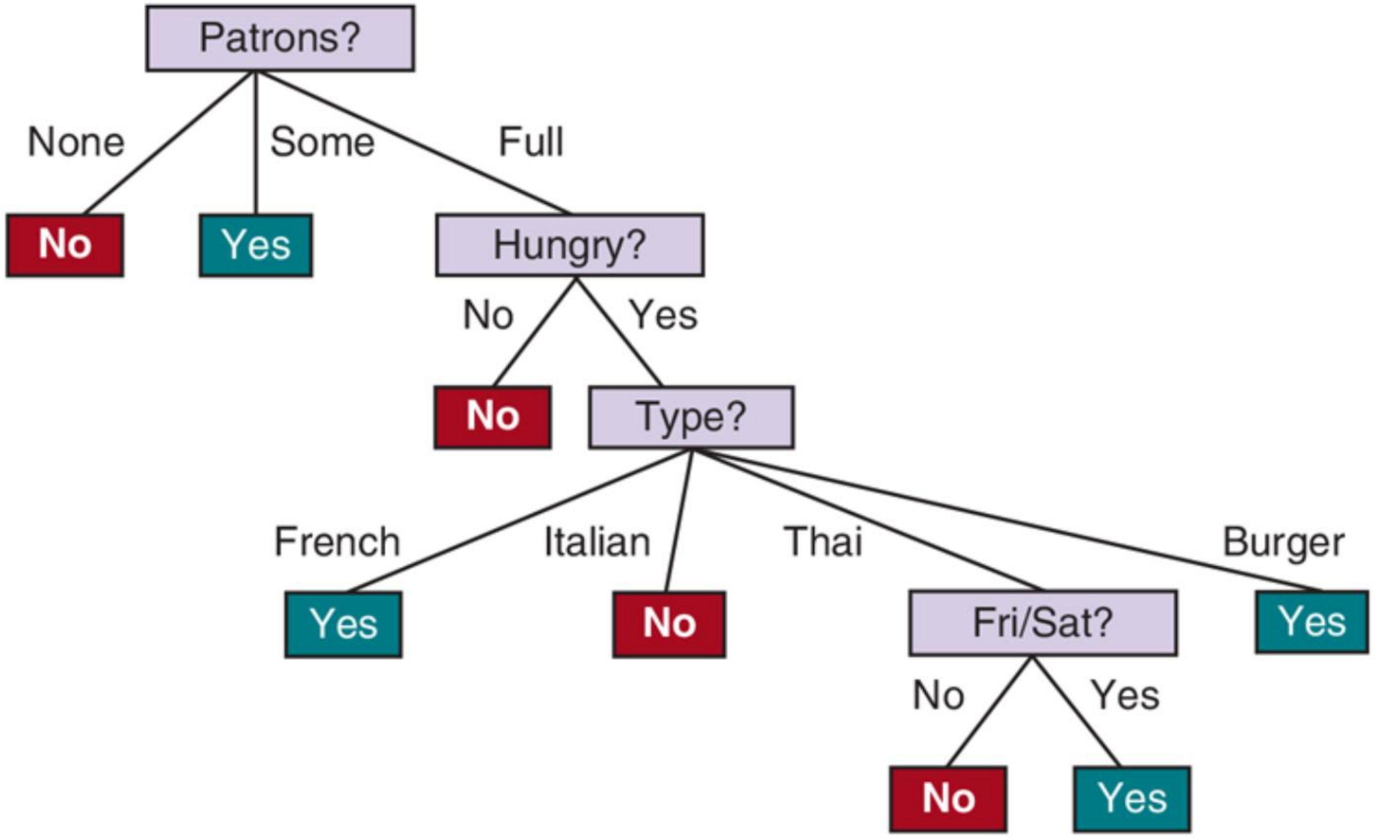


PREDICTED
LABELS

New scenarios

Predicted decisions
to wait or not

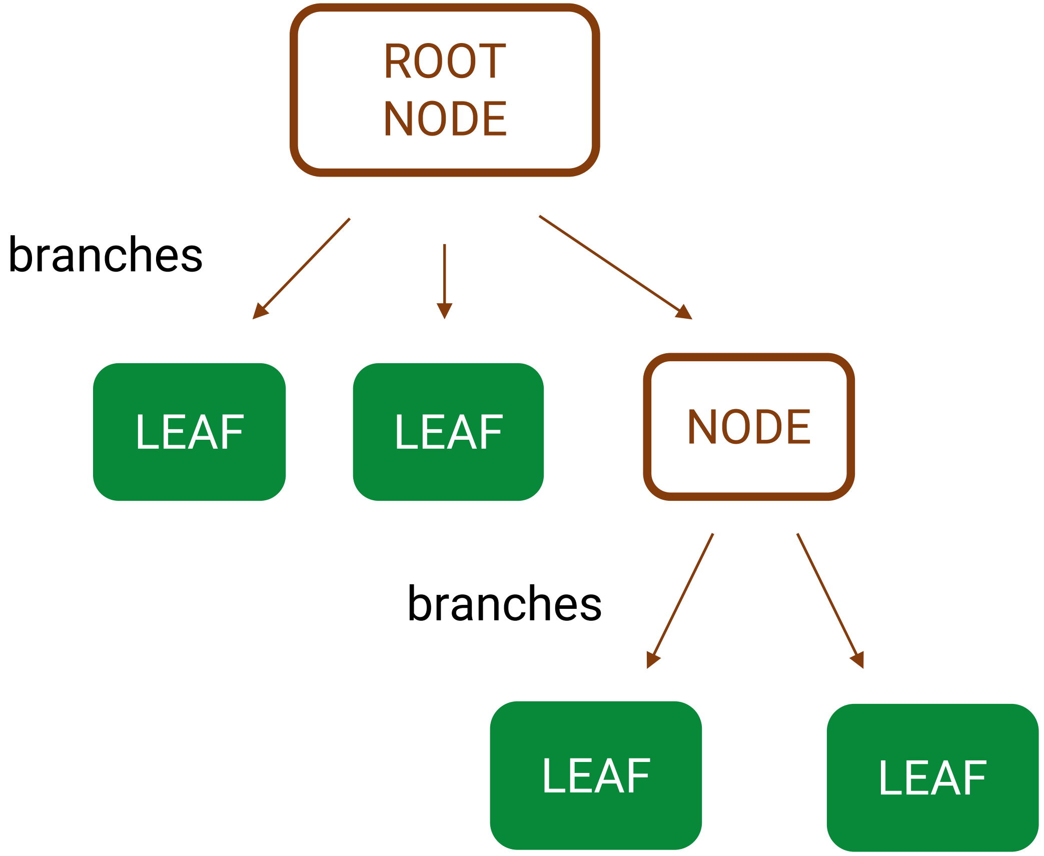
Classification: Decision Tree



The decision tree induced from the 12-example training set.

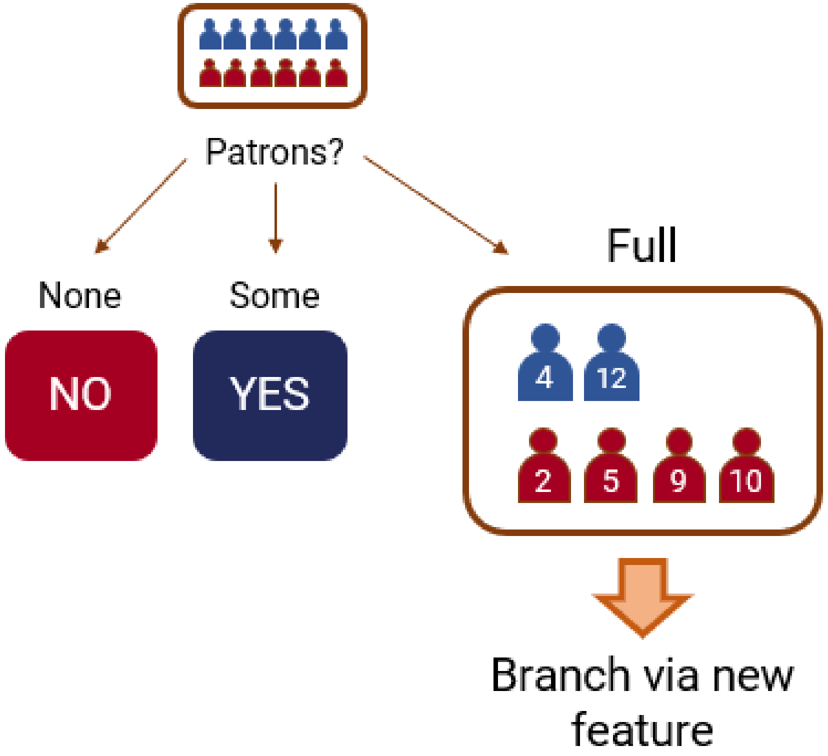
What's a tree (in graph theory)

- Hierarchical structure
  - made up of nodes
  - linked by parent-child (branch) relationships
- Terms:
  - Root: first node
  - Branch
  - Leaf: terminal node



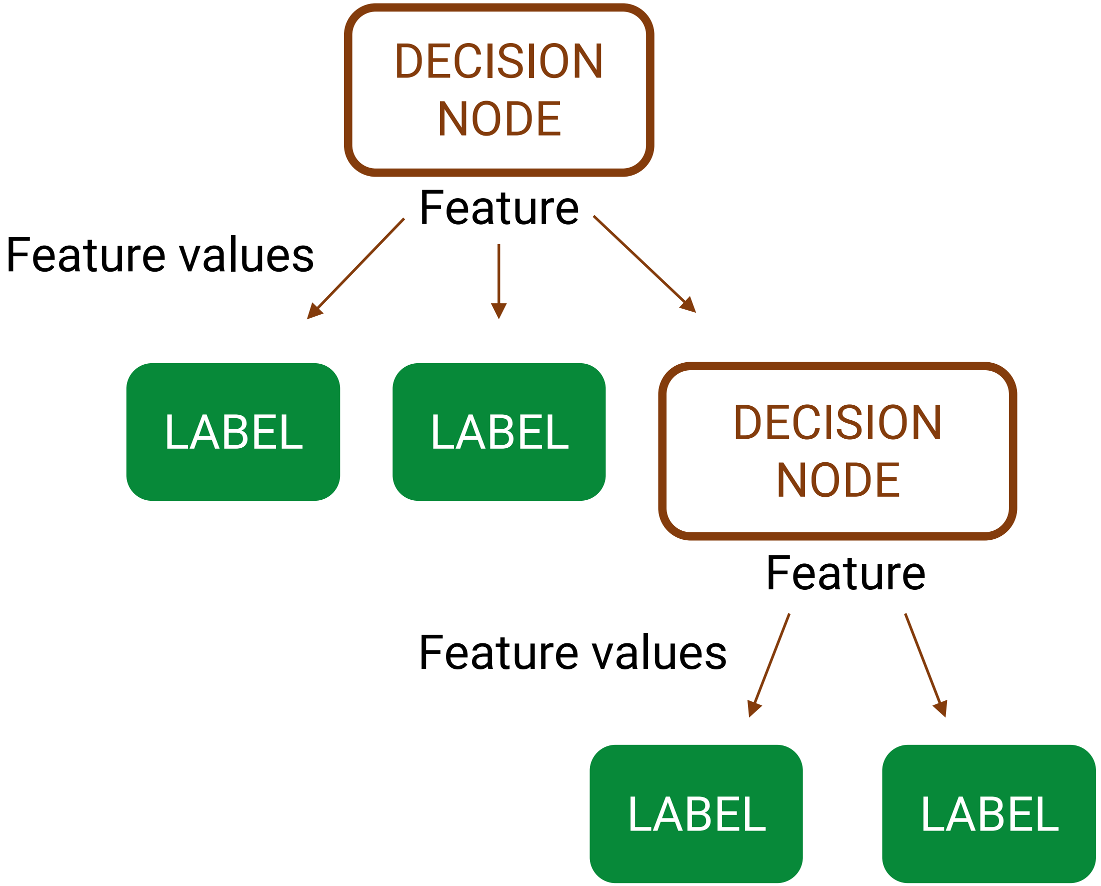
What's a decision tree

- A sequence of tests (decisions) induced from a dataset
- Each test is based on a single feature
- Eventually leads to a predicted label

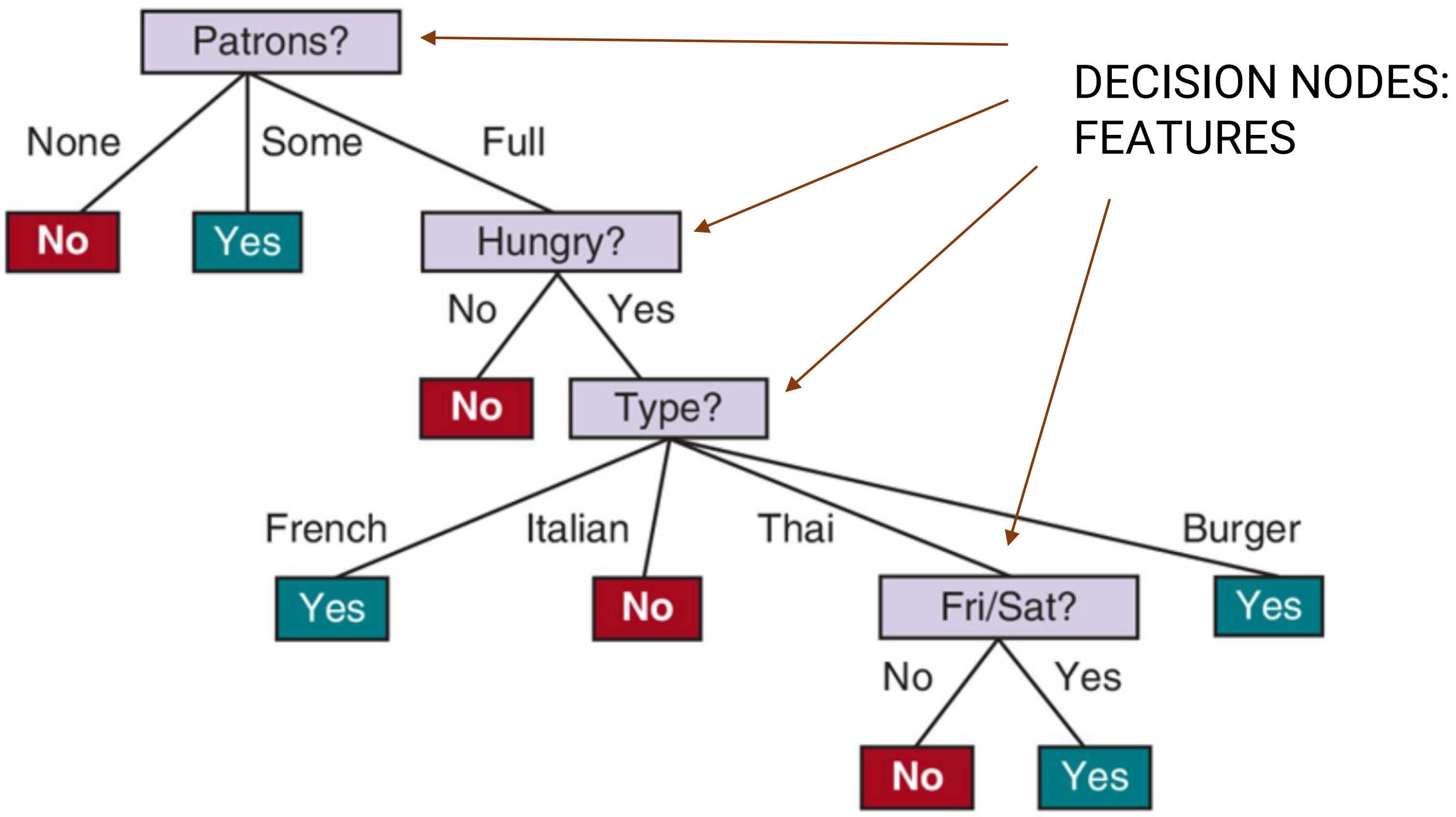


What’s a decision tree

- Features as decision nodes
- Feature values as branches
  - a split based on result of decision
- Leaf/terminal nodes as labels or classes

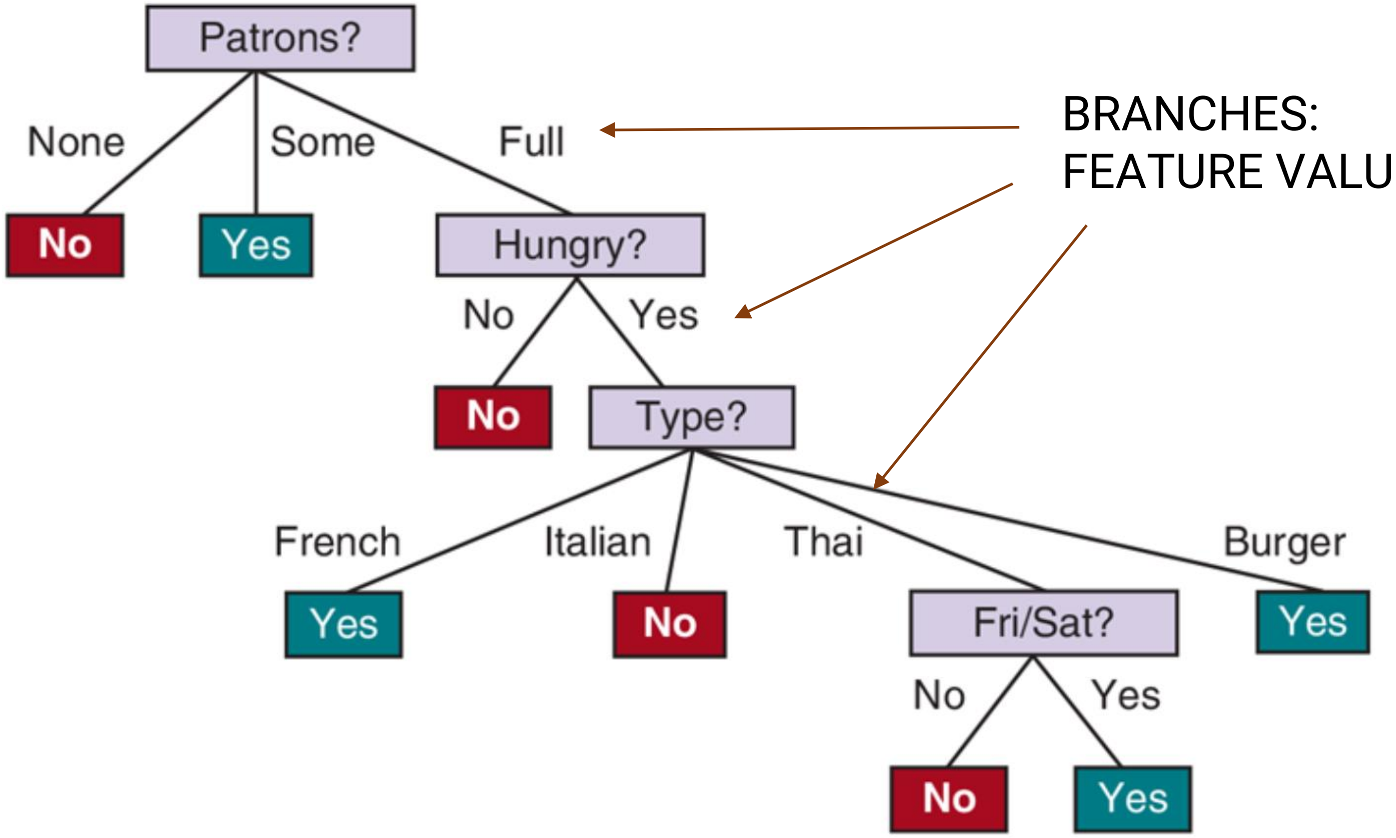


Classification: Decision Tree



The decision tree induced from the 12-example training set.

Classification: Decision Tree

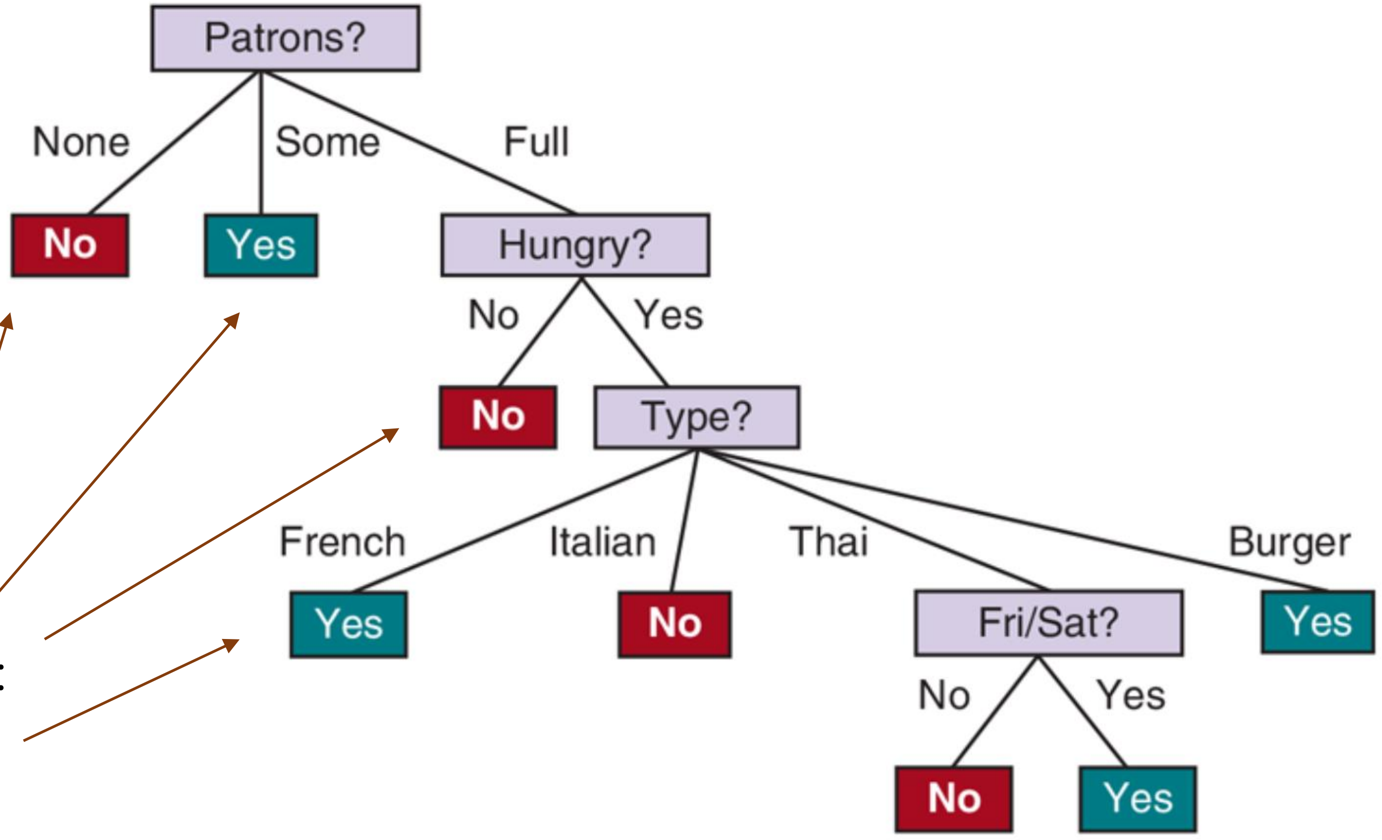


BRANCHES:
FEATURE VALUES

The decision tree induced from the 12-example training set.

Classification: Decision Tree

TERMINAL NODES:
LABELS/CLASSES



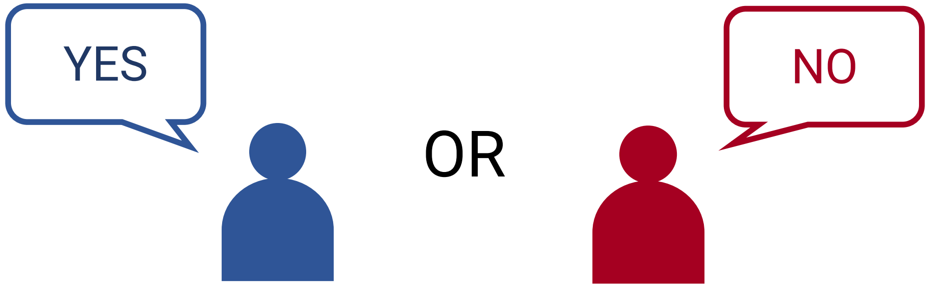
The decision tree induced from the 12-example training set.

Decision Tree Application

NEW DATA

Hungry	Patron	…	Type
Yes	Full		Italian

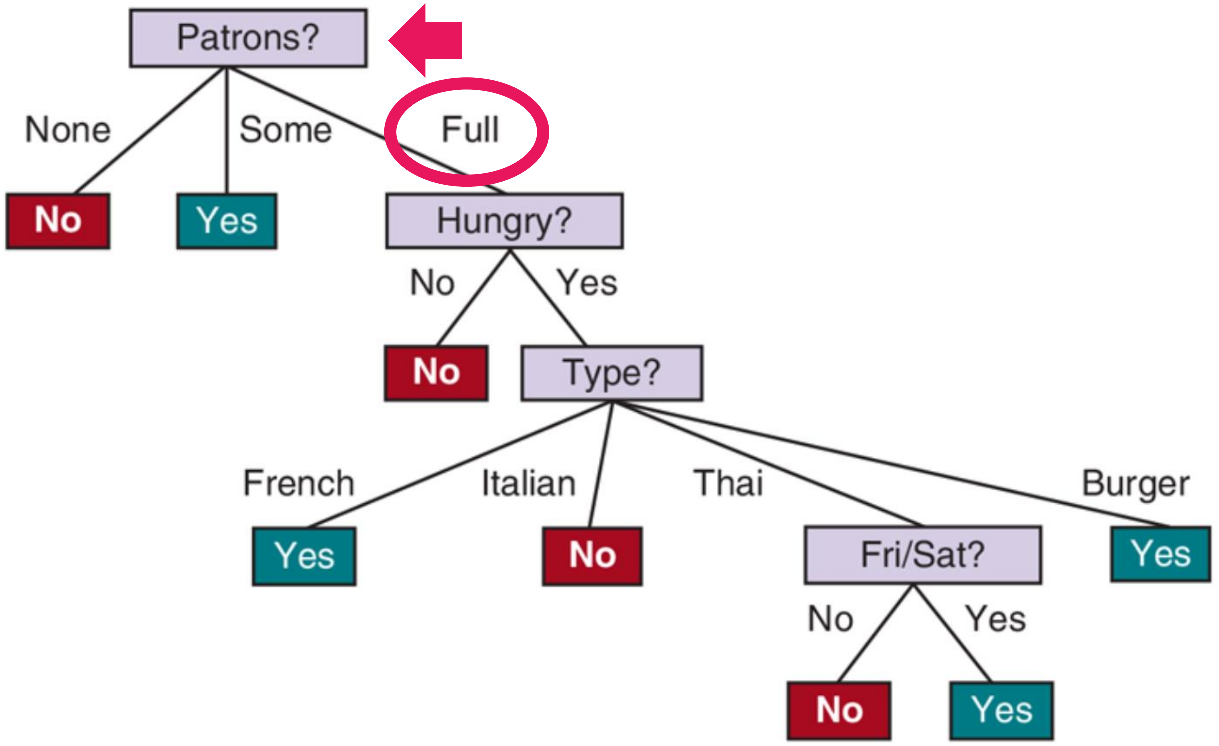
LABEL: WILL WAIT?



Decision Tree Application

NEW DATA

Hungry	Patron	...	Type
Yes	Full		Italian

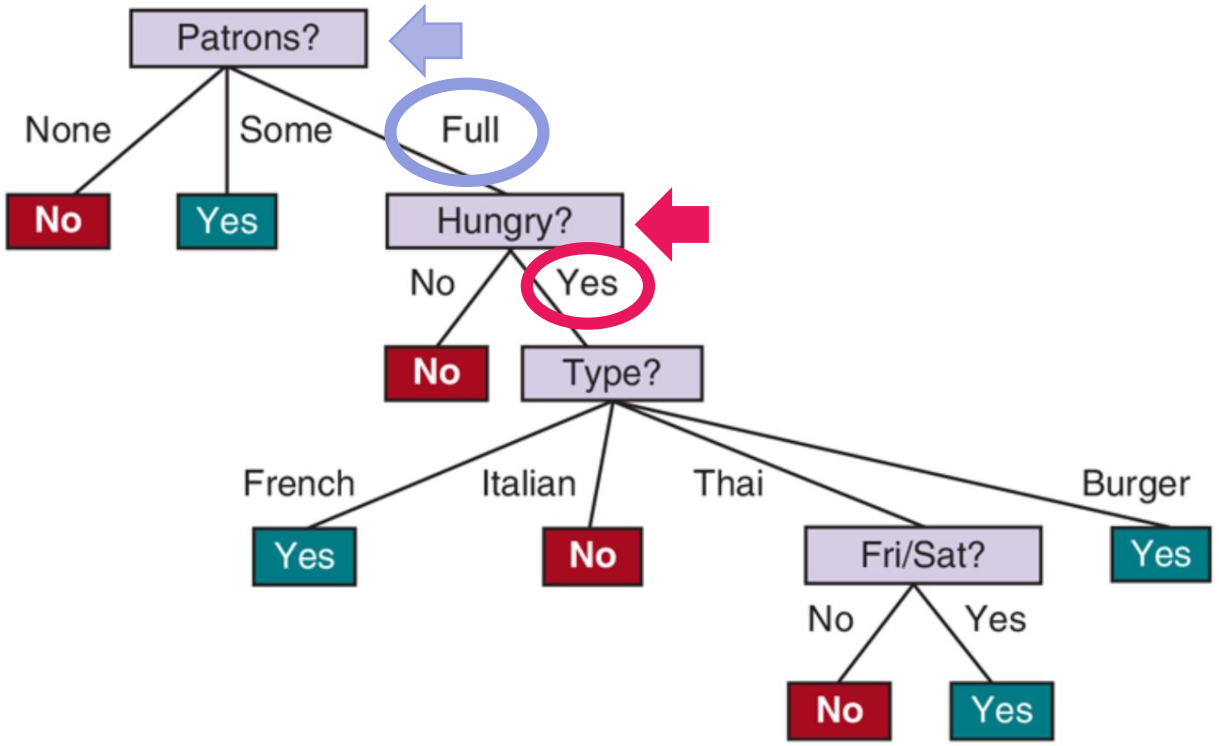


The decision tree induced from the 12-example training set.

Decision Tree Application

NEW DATA

Hungry	Patron	...	Type
Yes	Full		Italian

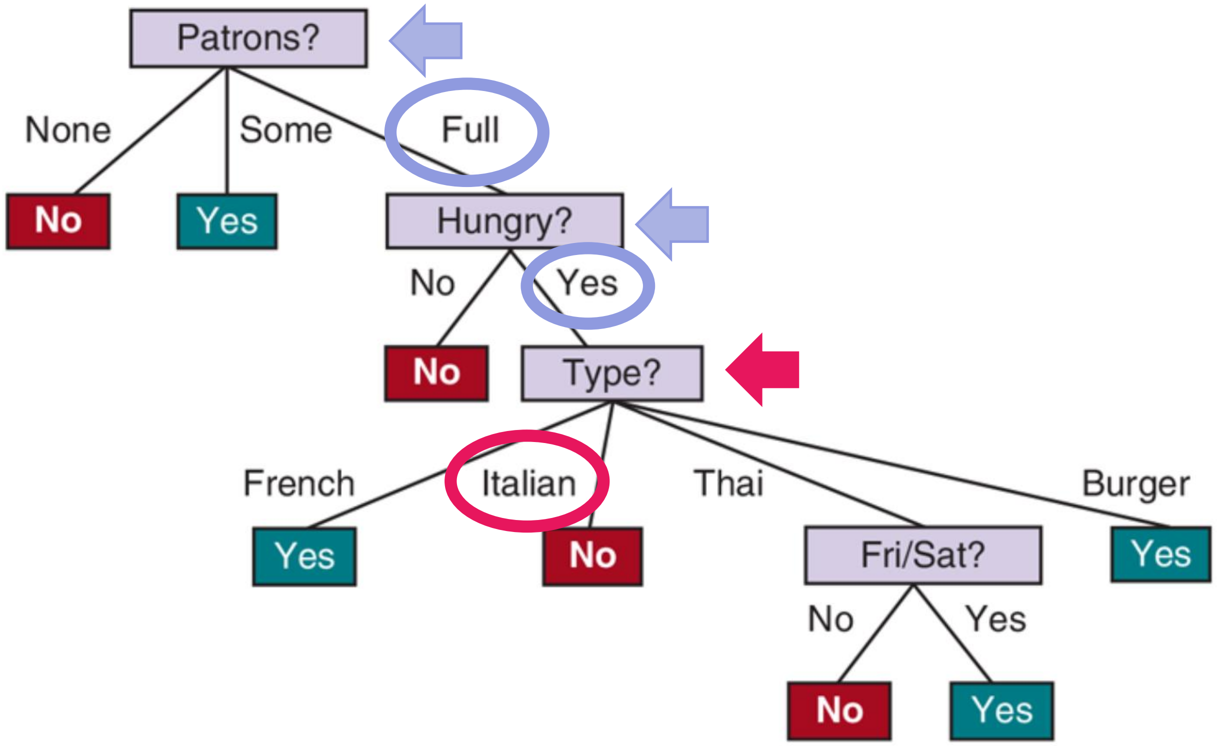


The decision tree induced from the 12-example training set.

Decision Tree Application

NEW DATA

Hungry	Patron	...	Type
Yes	Full		Italian



The decision tree induced from the 12-example training set.

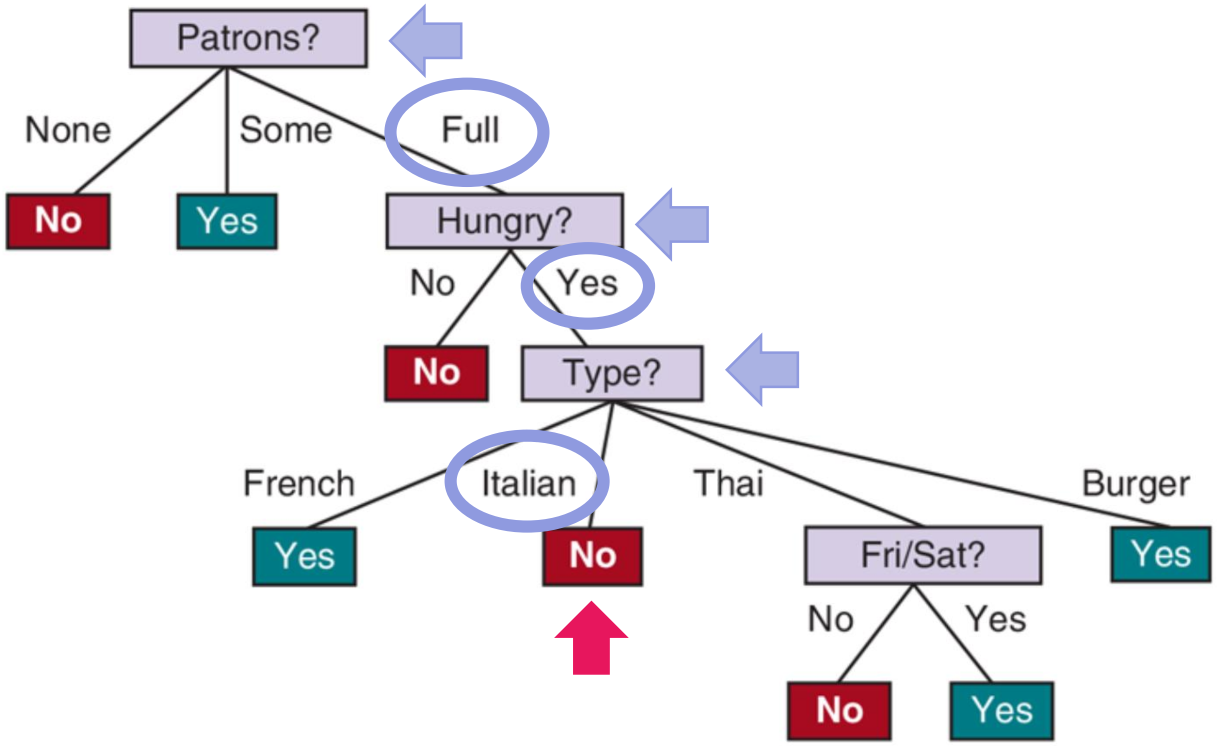
Decision Tree Application

NEW DATA

Hungry	Patron	…	Type
Yes	Full		Italian

REACHED TERMINAL NODE

LABEL: NO

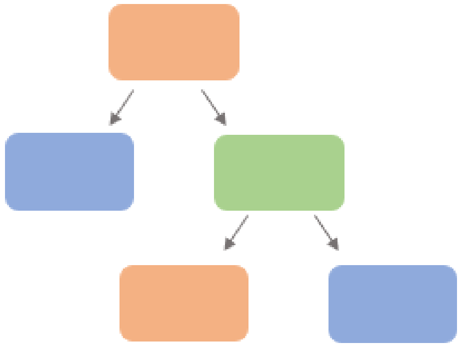


The decision tree induced from the 12-example training set.

Decision Tree Application

NEW DATA

Hungry	Patron	...	Type
Yes	Full		Italian



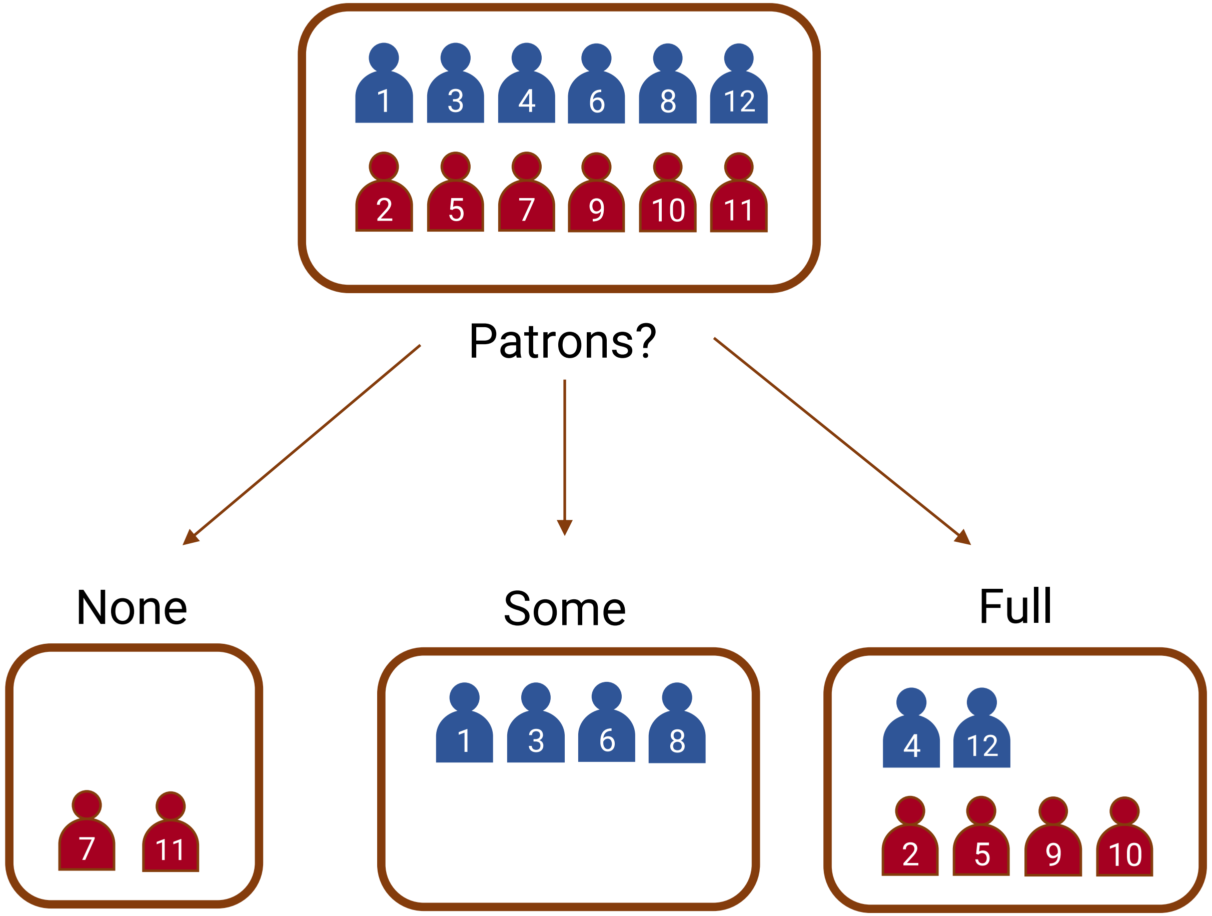
PREDICTED LABEL:
WILL WAIT?

DEPARTMENT of INFORMATION SYSTEMS & COMPUTER SCIENCE

Decision Tree Induction

Decision Tree Induction

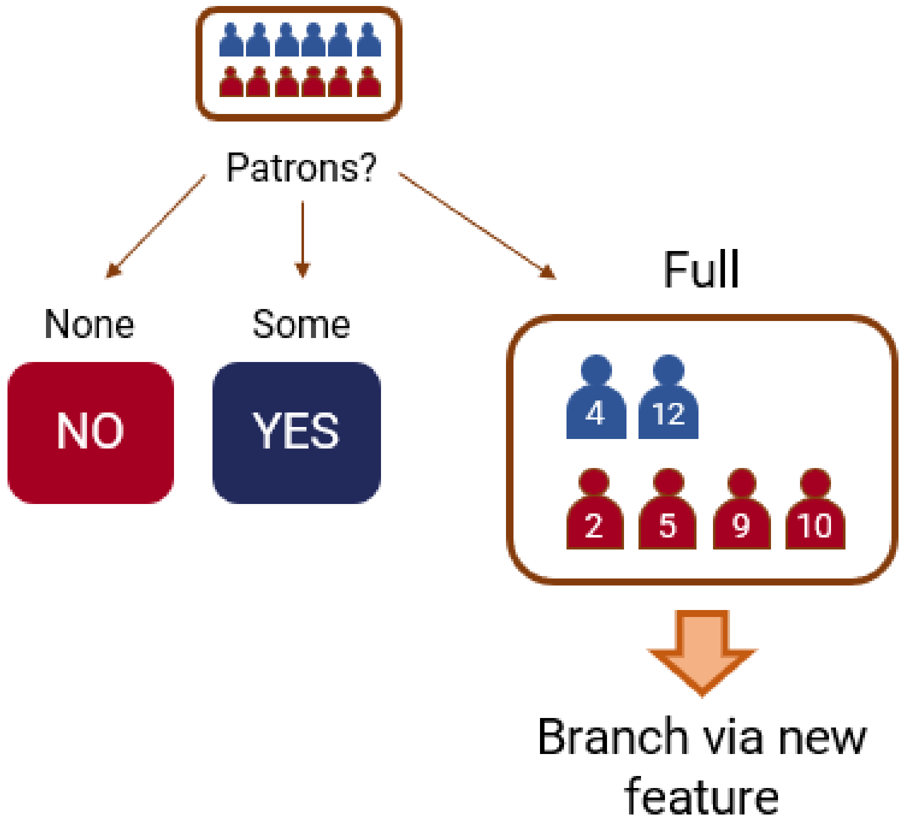
- Starts with the entire dataset
- Instances are split into nodes according to feature values
- A split represents a decision node in the tree



Decision Tree Induction

Instances "trickle down" until:

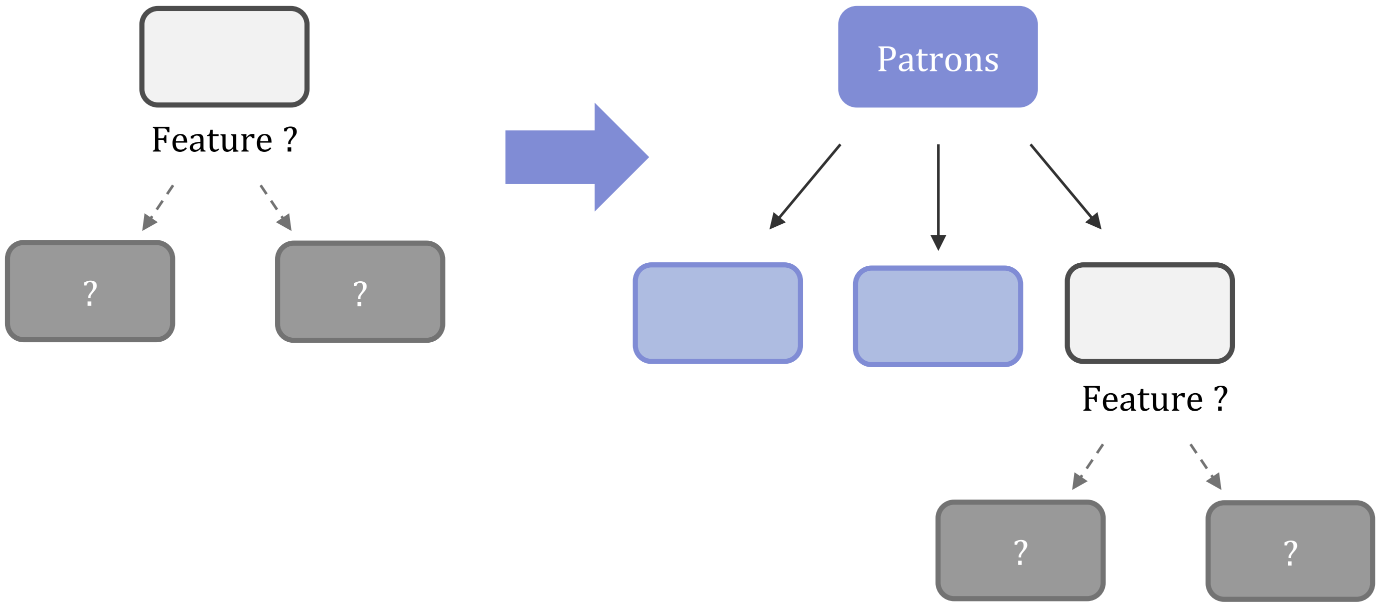
- homogeneity is achieved,
- all features have been used, or
- instances in a node are lower than a threshold



Decision Tree Induction

Greedy algorithm:

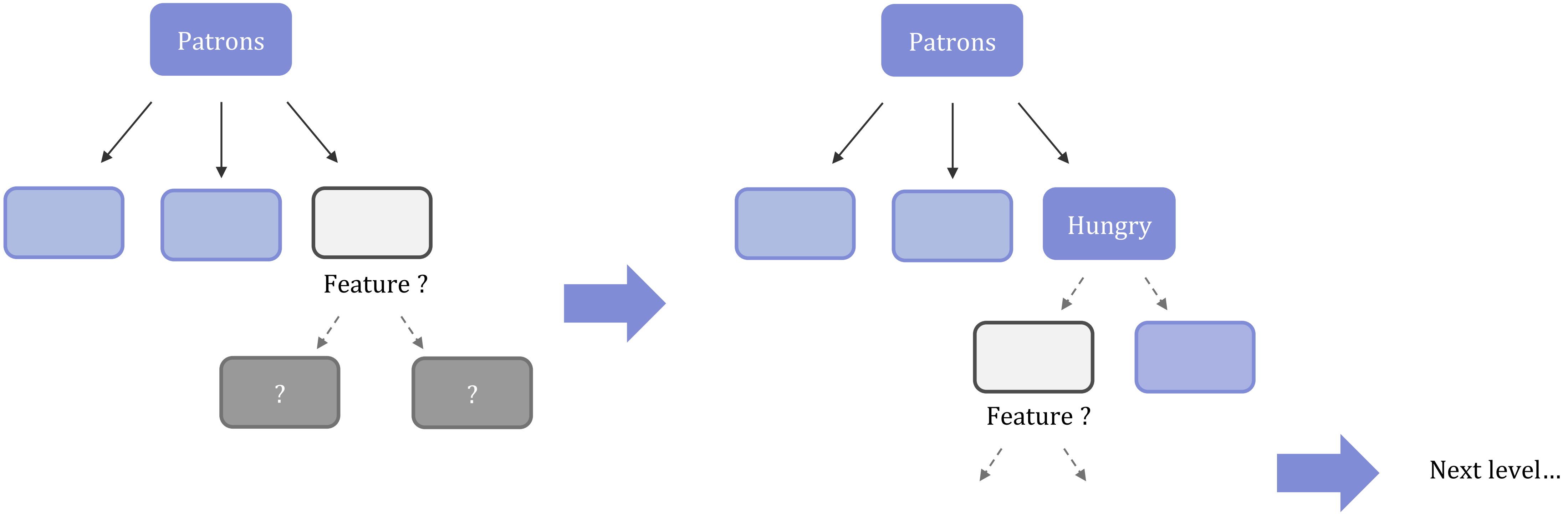
- To make a tree: determine the best attribute to split at every level



Decision Tree Induction

Greedy algorithm:

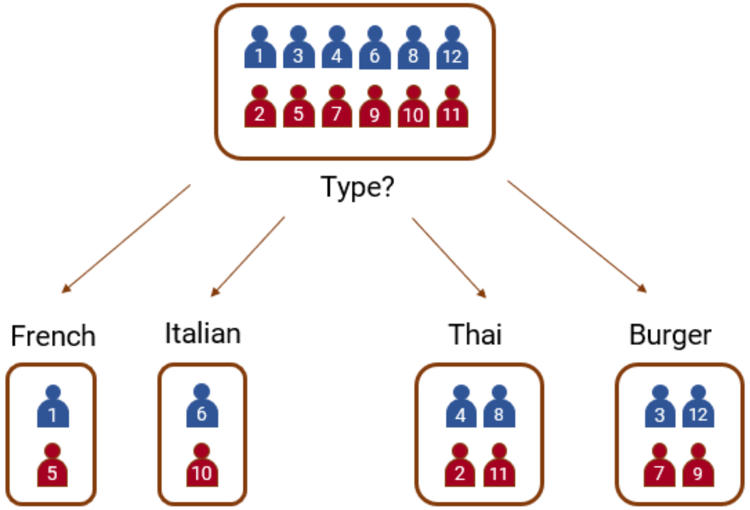
- To make a tree: determine the best attribute to split at every level



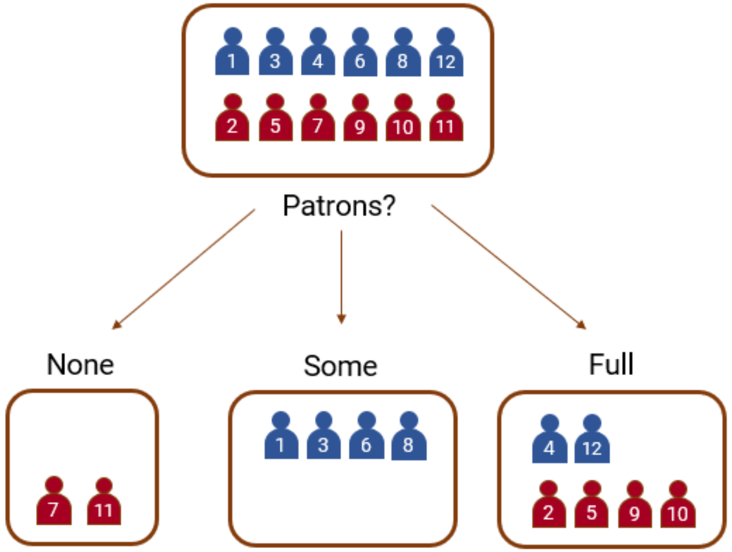
Decision Tree Induction

"Best Split"

- Attributes that <u>separate the data best</u> are more important
- Homogenous examples are preferred



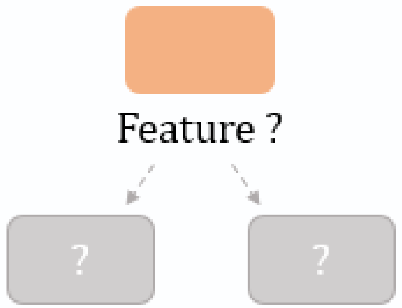
VS



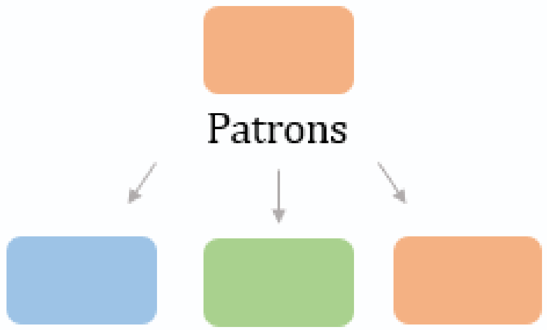
Decision Tree Induction

"Best Split"

- Score how well an attribute splits examples, then select attribute with the best score



Feature	GINI Score
Patrons	0.28
Hungry	0.37
Type	0.50
Friday	0.49



Decision Tree Induction

When inducing/making a decision tree, we consider:

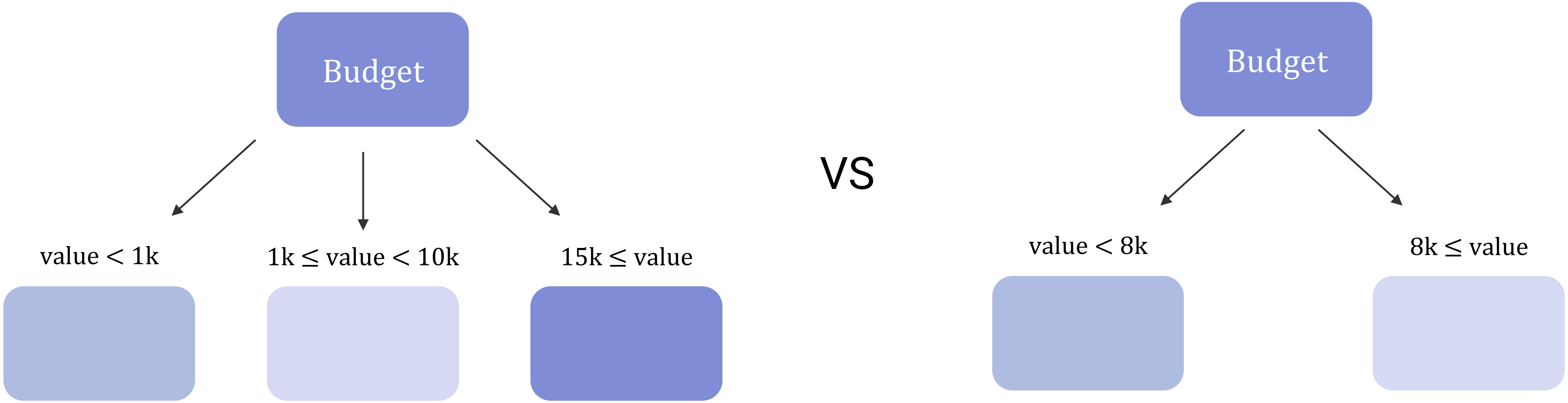
- Splitting: How to branch out an attribute
  - Nominal
  - Ordinal
  - Numeric
- Scoring splits: Which feature/attribute to split

DEPARTMENT *of*
INFORMATION SYSTEMS &
COMPUTER SCIENCE

Splitting by Attribute

Splitting

- There are multiple ways to split a feature/attribute
  - Especially for numeric values



Splitting Based on Attribute Type

- Nominal
  - no specific order or numerical value (e.g. gender, color)
- Ordinal
  - with a natural order or ranking (e.g. education levels)
- Continuous/numeric
  - numerical data that can take any value (e.g. temperature)

Splitting Nominal Attributes

- Nominal
  - Categories with no specific order or numerical value (e.g. gender, color)

- Possible Splits:
  - Multi-way split
  - Binary split

DEPARTMENT of INFORMATION SYSTEMS & COMPUTER SCIENCE

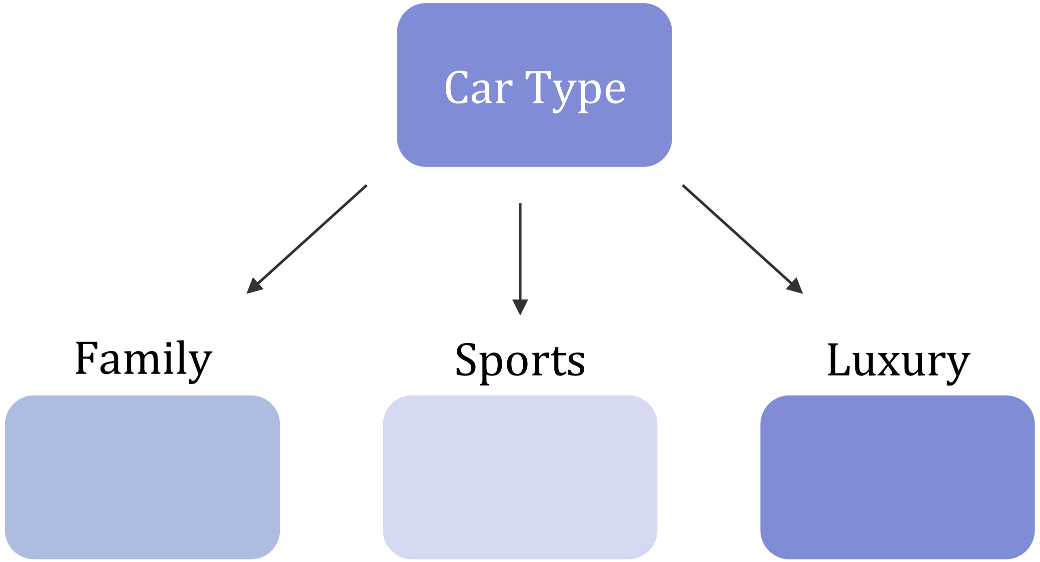
Splitting Nominal Attributes

Multi-way split

- Use as many partitions as distinct values

Example: Car Type

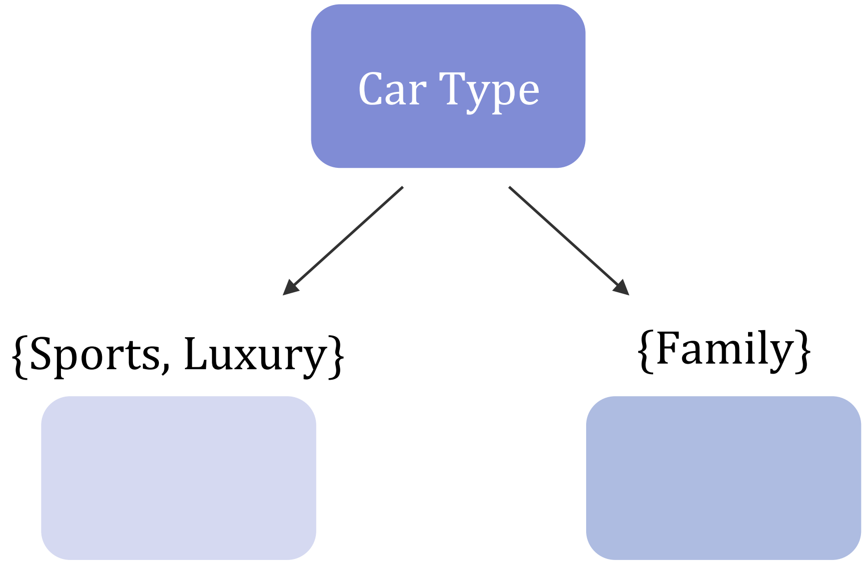
- Family
- Sport
- Luxury



Splitting Nominal Attributes

Binary split

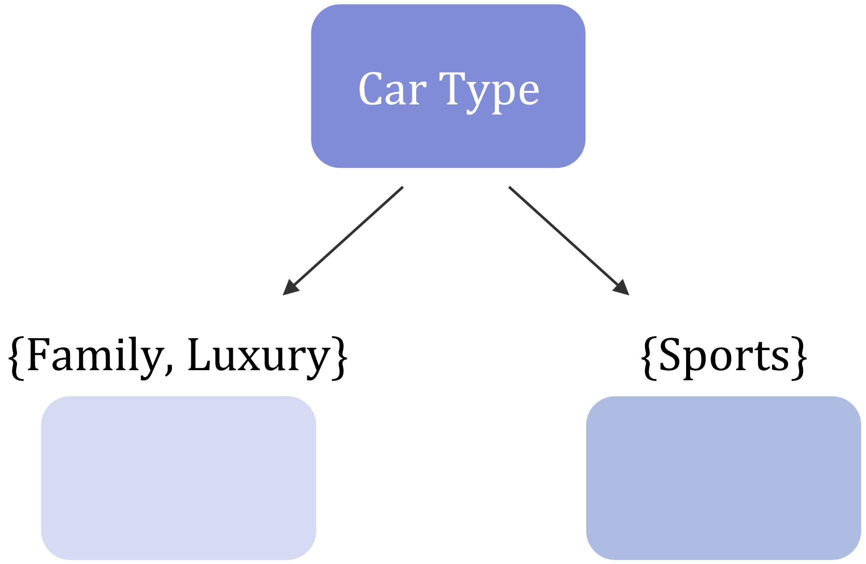
- Divides values into two subsets
- Requires finding the optimal partitioning



Splitting Nominal Attributes

Binary split

- Divides values into two subsets
- Requires finding the optimal partitioning



Splitting Ordinal Attributes

- Ordinal
  - Categories with a natural order or ranking (e.g. education level, patrons)

- Possible Splits:
  - Multi-way split
  - Binary split

DEPARTMENT *of*
INFORMATION SYSTEMS &
COMPUTER SCIENCE

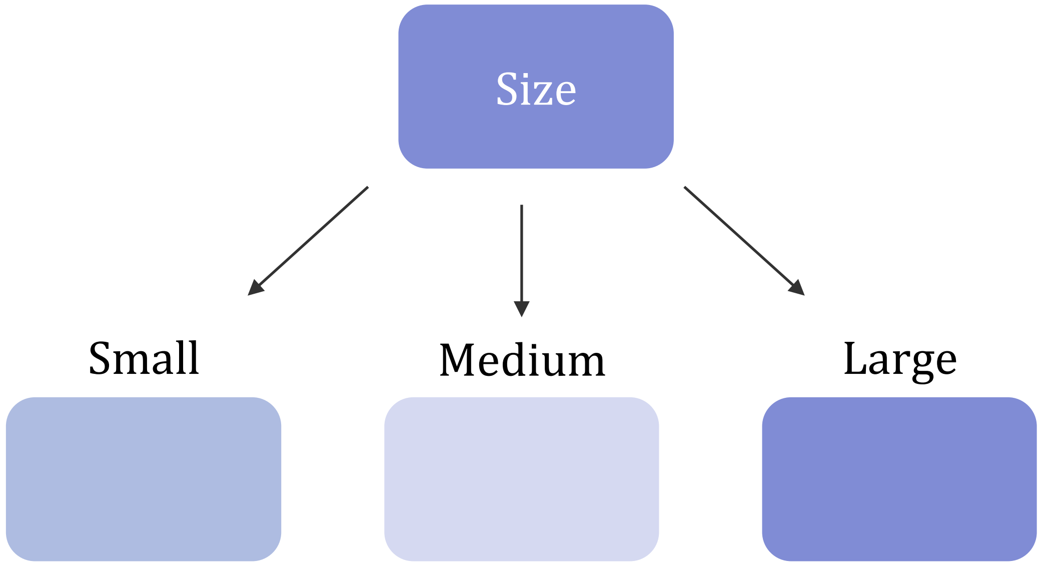
Splitting Ordinal Attributes

Multi-way split

- Use as many partitions as distinct values

Example: Size

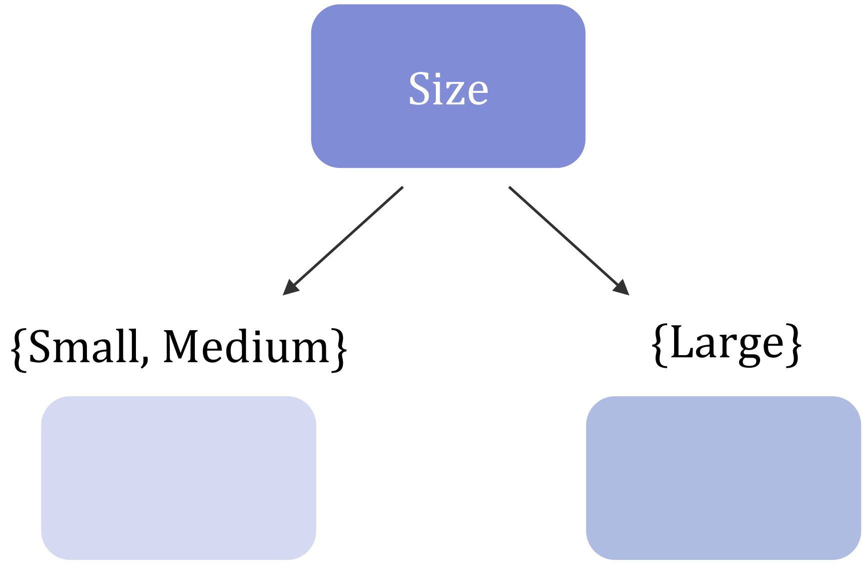
- Small
- Medium
- Large



Splitting Ordinal Attributes

Binary split

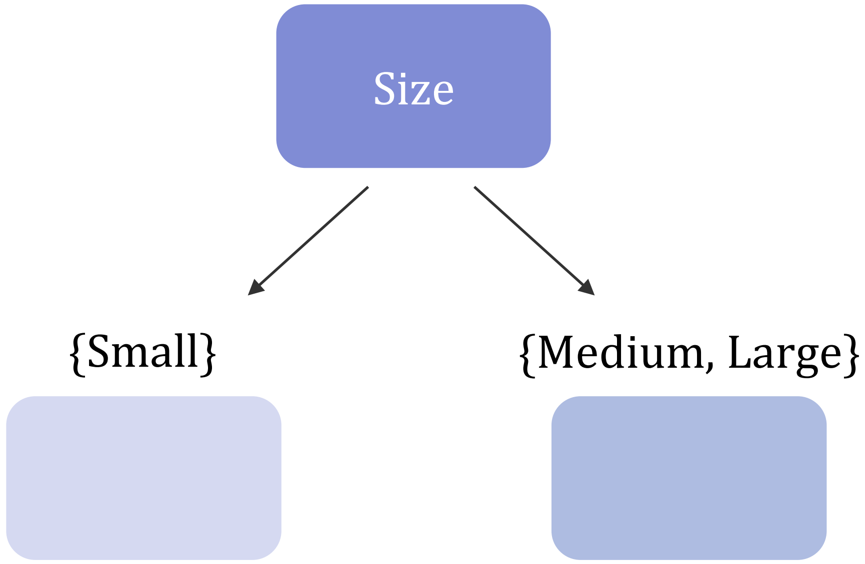
- Divides values into two subsets
- Requires finding the optimal partitioning



Splitting Ordinal Attributes

Binary split

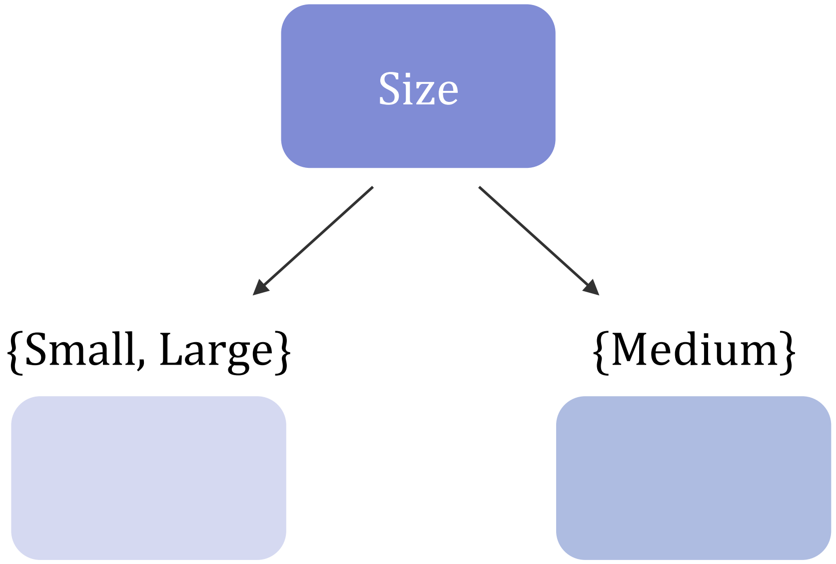
- Divides values into two subsets
- Requires finding the optimal partitioning



Splitting Ordinal Attributes

Binary split

- Divides values into two subsets
- Requires finding the optimal partitioning



Does this make sense?

Splitting Continuous Attributes

- Continuous/Numeric
  - Numerical data that can take any value (e.g. height, temperature)

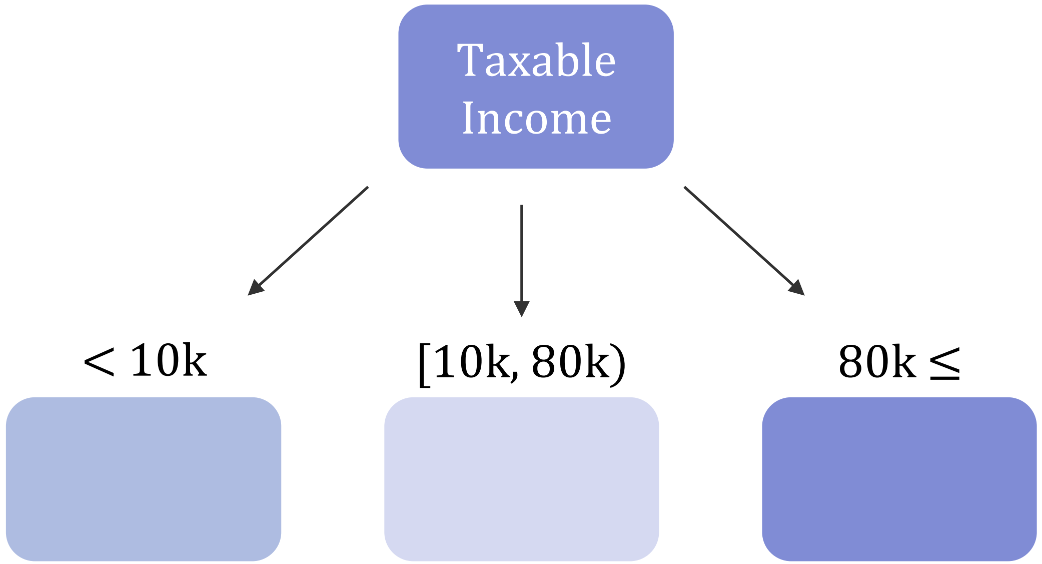
- Possible Splits:
  - Discretization
  - Binary Decision

DEPARTMENT of INFORMATION SYSTEMS & COMPUTER SCIENCE

Splitting Continuous Attributes

Discretization (Multi-way)

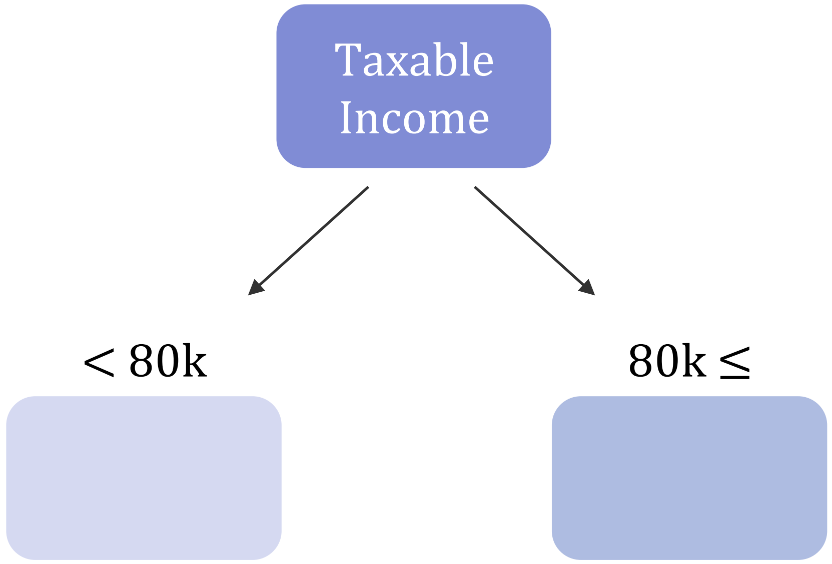
- Form an ordinal categorical attribute
- Bucketing, percentiles, clustering



Splitting Continuous Attributes

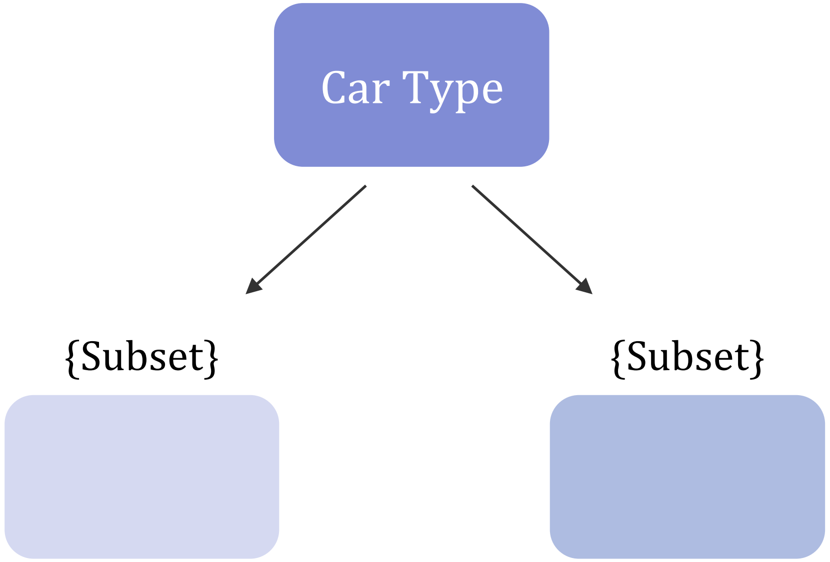
Binary Decision

- Consider all possible splits and find best cut
- Can be more compute intensive



Best Split via Scoring

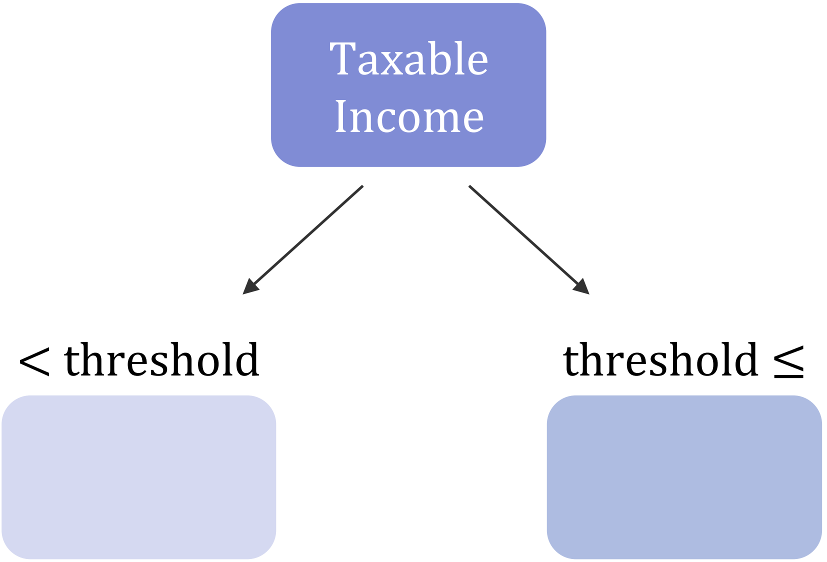
Sometimes use a scoring method to determine best binary split or thresholds



Binary Split	Score
{Sports} and {Luxury, Family}	
{Luxury} and {Sports, Family}	
{Family} and {Sports, Luxury}	

Best Split via Scoring

Sometimes use a scoring method to determine best binary split or thresholds

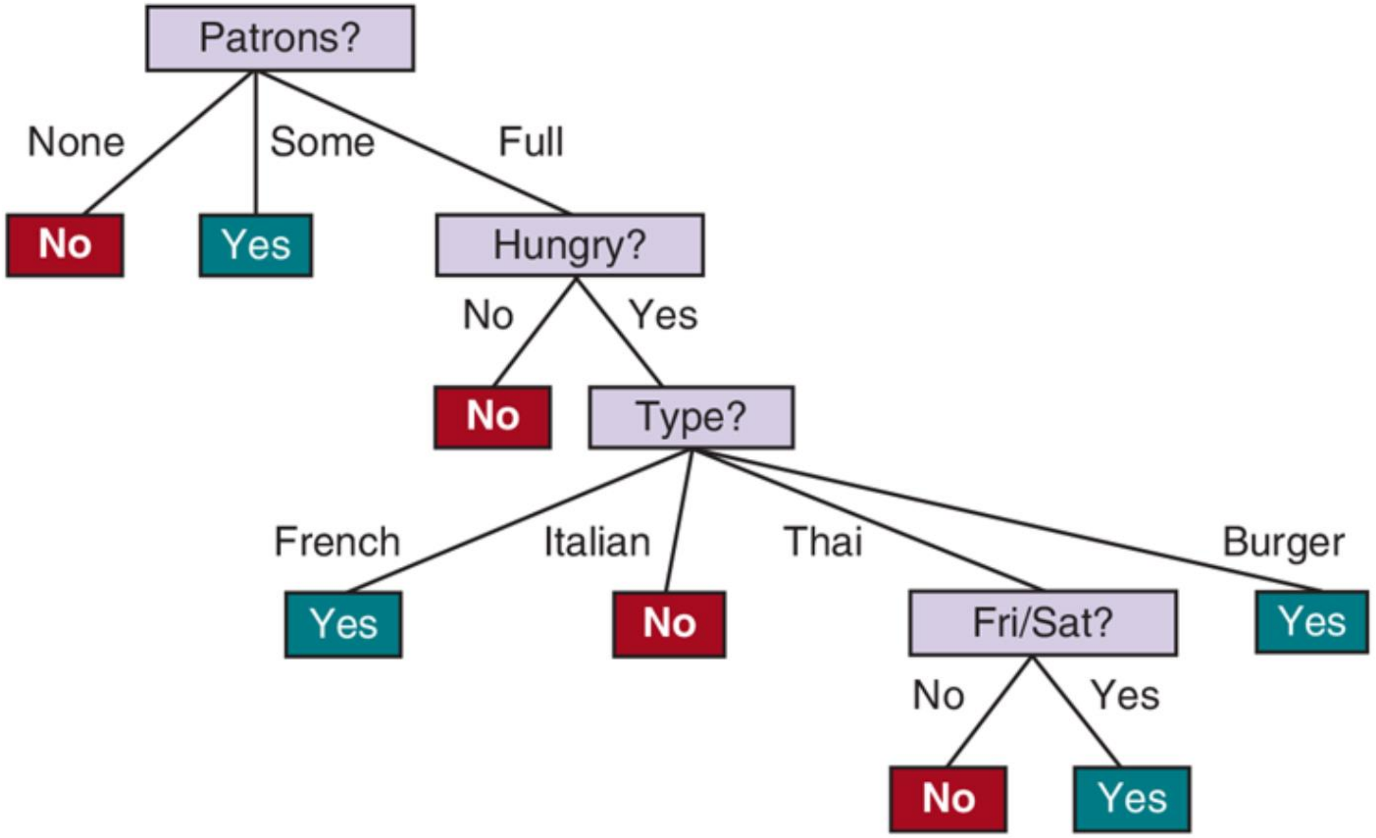


Threshold	Score
10k	
20k	
30k	
…	

DEPARTMENT of INFORMATION SYSTEMS & COMPUTER SCIENCE

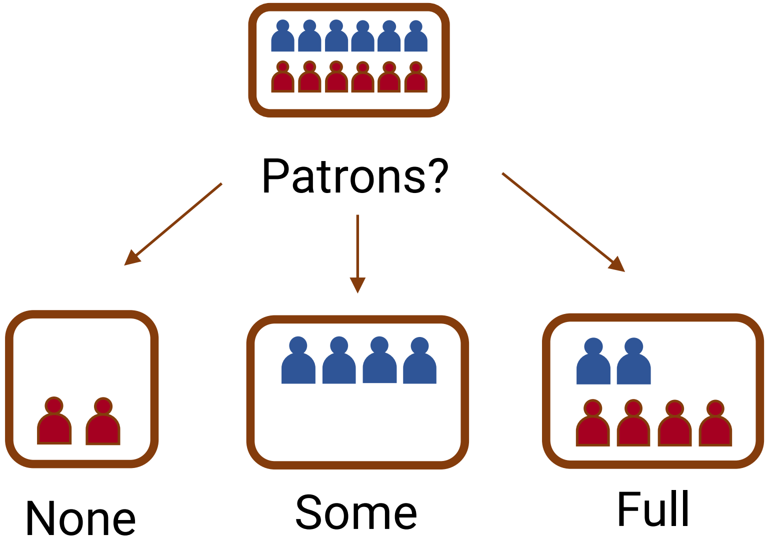
Scoring Splits

Decision Tree



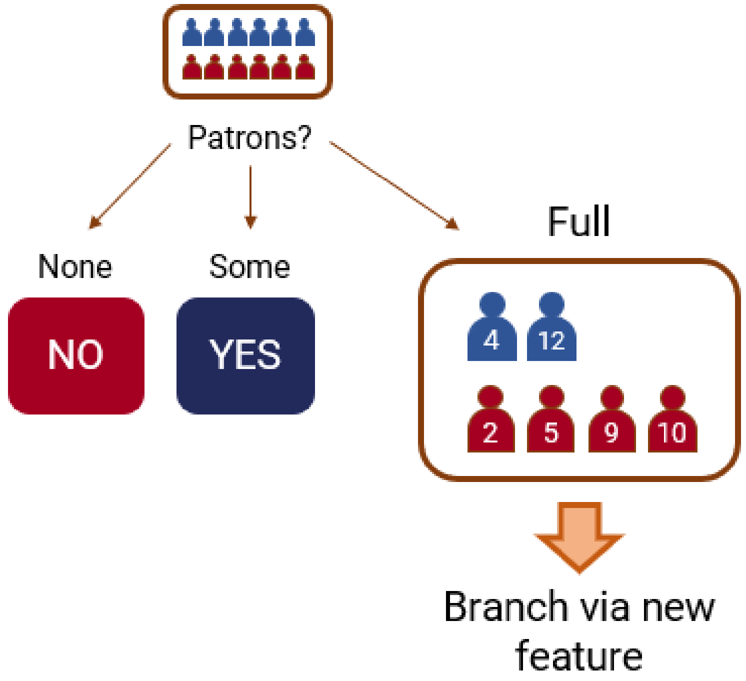
The decision tree induced from the 12-example training set.

Choosing the Feature



Why did we start with the Patrons feature for branching instead of other features?

Choosing the Feature

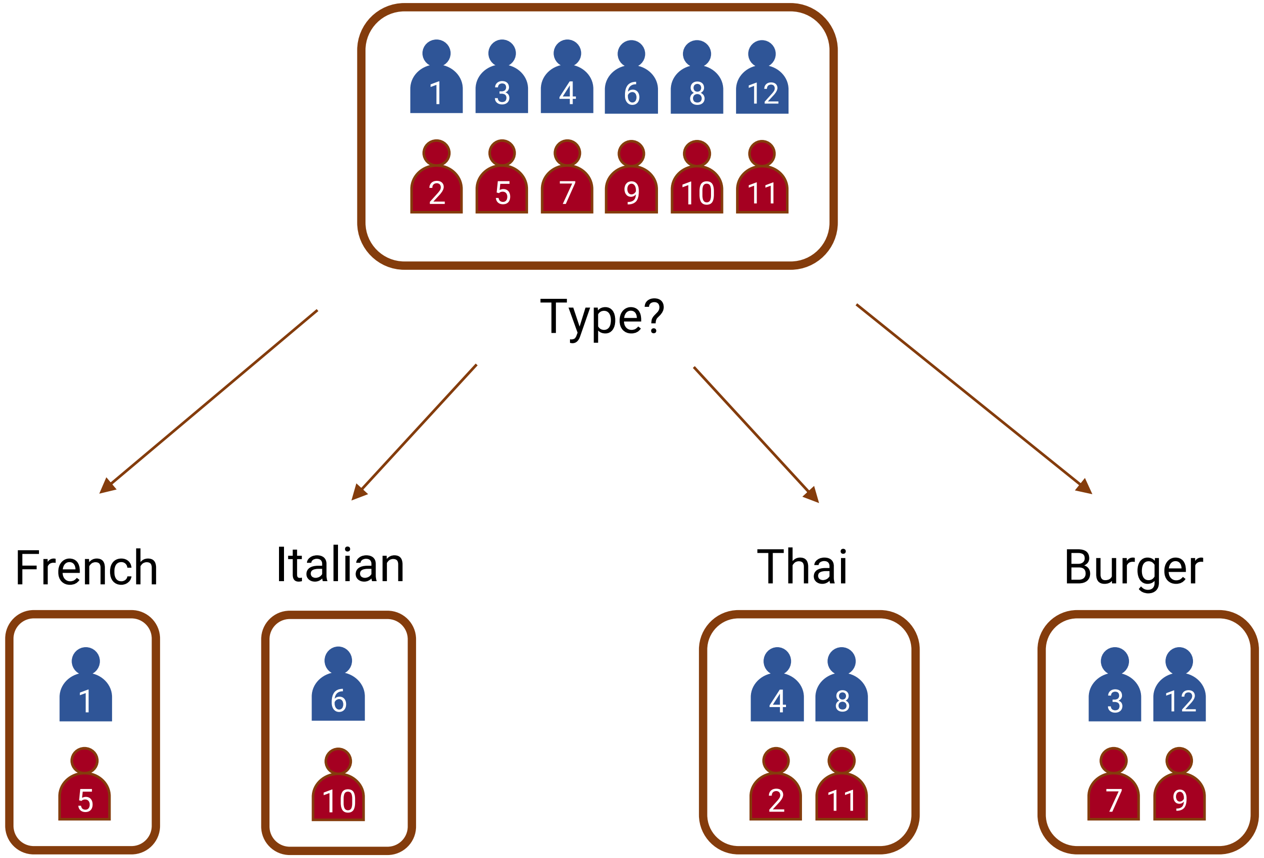


And how will we choose which feature to branch/split next?

Decision Tree

Which feature do we split?

The feature that can best distinguish examples by their labels



Same number of “Yes” and “No” per group: bad

Scoring splits

<u>Best split:</u>

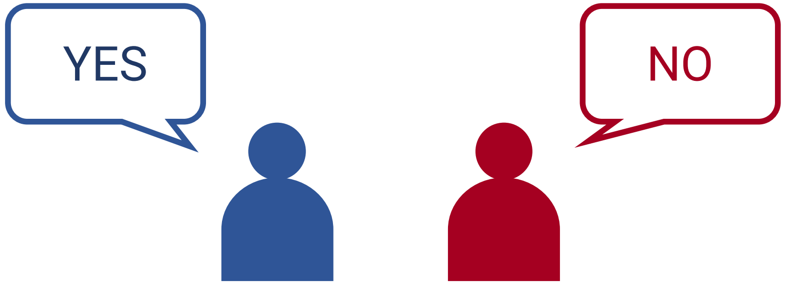
- Nodes with homogeneous class distributions are preferred
  - Homogeneous: when examples in a node tend to be in one class/label

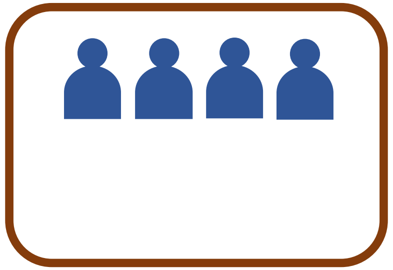
- Finding best split by <u>measuring node impurity</u>

DEPARTMENT of INFORMATION SYSTEMS & COMPUTER SCIENCE

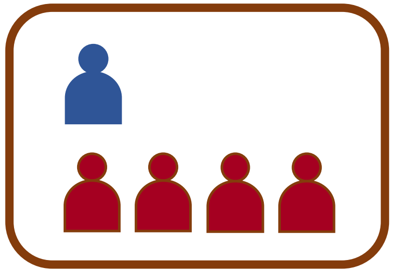
Node Impurity

Labels/Classes:

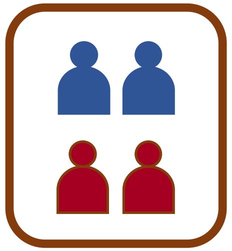




(Yes: 4, No: 0)
Homogeneous:
Pure



(Yes: 1, No: 4)
Non-homogeneous:
Low impurity



(Yes: 2, No: 2)
Non-homogeneous:
High impurity

Node Impurity

Different measures for node impurity

- Gini index
- Entropy
- Misclassification error

All these measures help determine most important attributes that:

- Separate examples best
- Provide the most homogeneity for the tree

DEPARTMENT *of* INFORMATION SYSTEMS & COMPUTER SCIENCE

GINI Index

Measure of Impurity: GINI

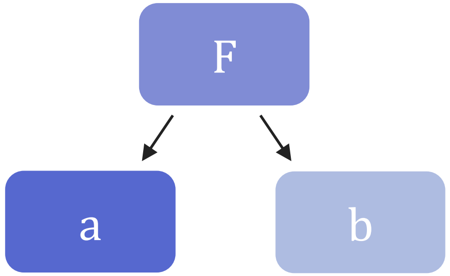
$$GINI(t) = 1 - \sum_{j} [p(j|t)]^2$$

t : node (e.g. the category like none/some/full for Patrons)
j : class (e.g. the label like Yes/No for Will Wait)
$p(j|t)$: relative frequency of the class in the group

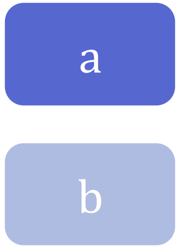
DEPARTMENT *of* INFORMATION SYSTEMS & COMPUTER SCIENCE

DECISION TREE

GINI INDEX FOR A FEATURE



PICK A FEATURE AND SPLIT



GINI(a)

GINI(b)

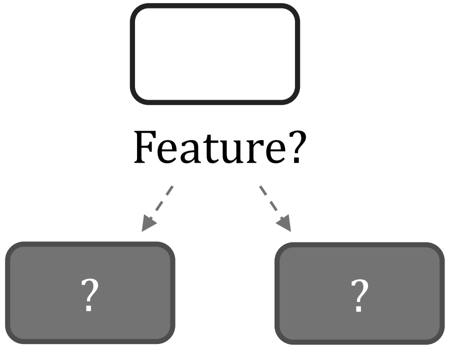
COMPUTE GINI INDEX FOR EACH NODE

$$GINI(a)\left(\frac{\#\ in\ a}{total\ \#}\right) + GINI(b)\left(\frac{\#\ in\ b}{total\ \#}\right)$$

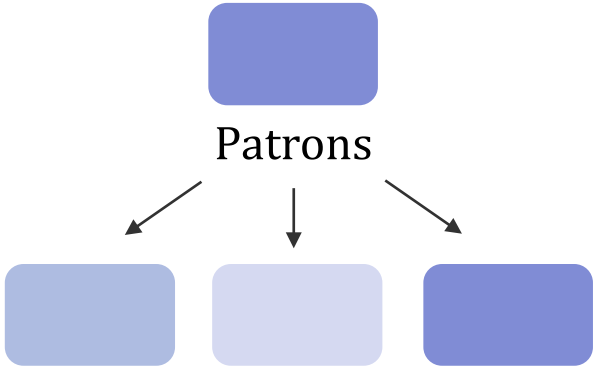
GET THE WEIGHTED AVERAGE

DECISION TREE

BRANCHING



FEATURE	GINI SCORE
Hungry	0.37
Patrons	0.28
Type	0.50
Friday	0.49



FIND FEATURE WITH THE SMALLEST GINI SCORE

SPLIT/BRANCH ACCORDING TO THAT FEATURE

DEPARTMENT of INFORMATION SYSTEMS & COMPUTER SCIENCE

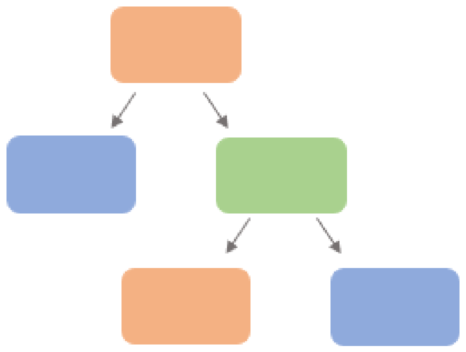
Summary

Classification: Decision Tree

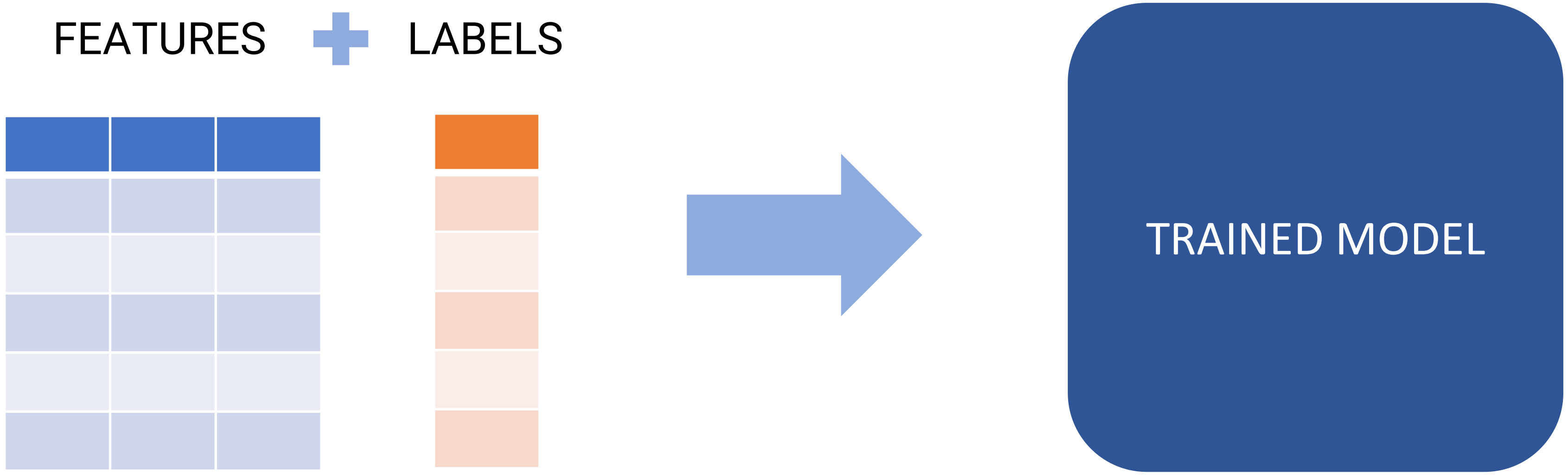
- A decision tree classifier encodes a function for the dataset as a <u>sequence of decisions</u> or tests
- Each test is based on a single feature
- Eventually leads to a predicted label

TRAINED CLASSIFICATION MODEL

$y = f(X_1, X_2, X_3, \ldots X_m)$



Classification: Decision Tree

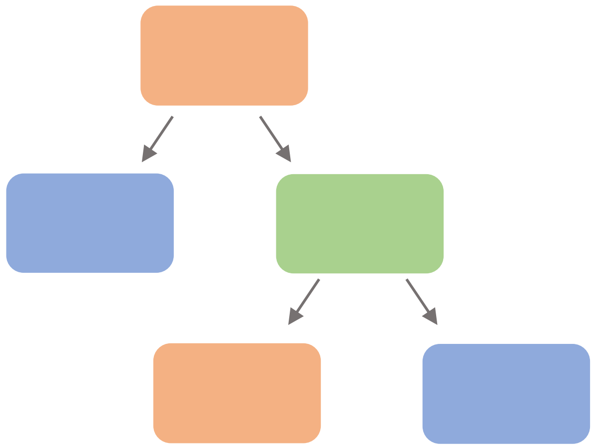


TRAINING SET:

Scenarios and corresponding decisions to wait or not

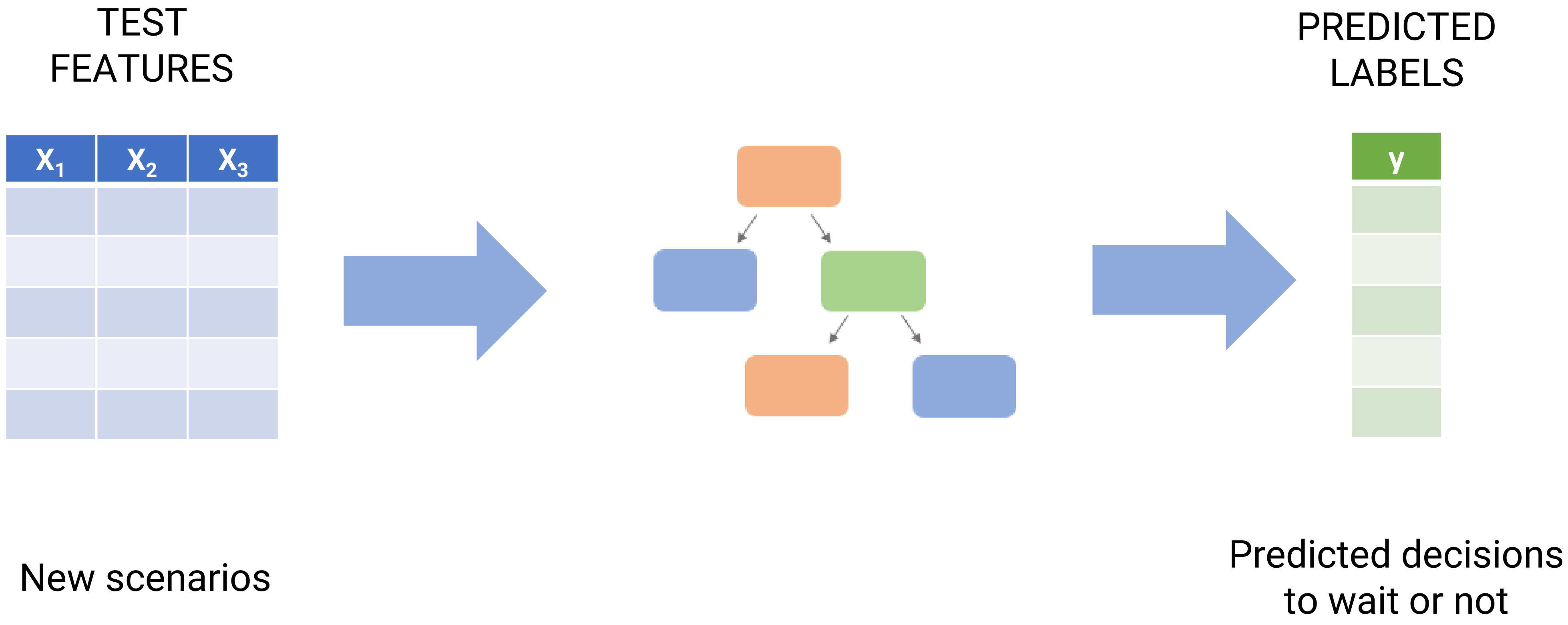
Classification: Decision Tree

TRAINED MODEL
$y = f(X_1, X_2, X_3, \ldots X_m)$



Decision tree based on scenarios and corresponding decisions (training set)

Classification: Decision Tree

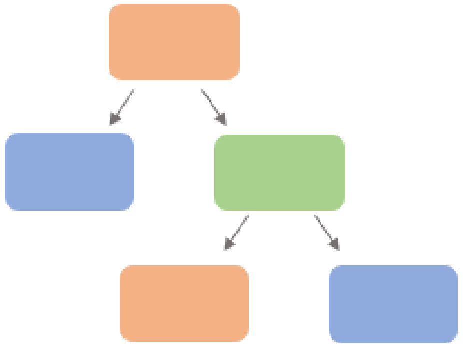


Restaurant Waiting

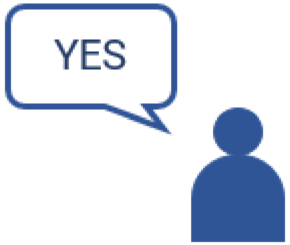
Given a new scenario (set of features), predict whether they'll wait or not using a decision tree

NEW DATA

Hungry	Patron	...	Type
Yes	Some		Italian

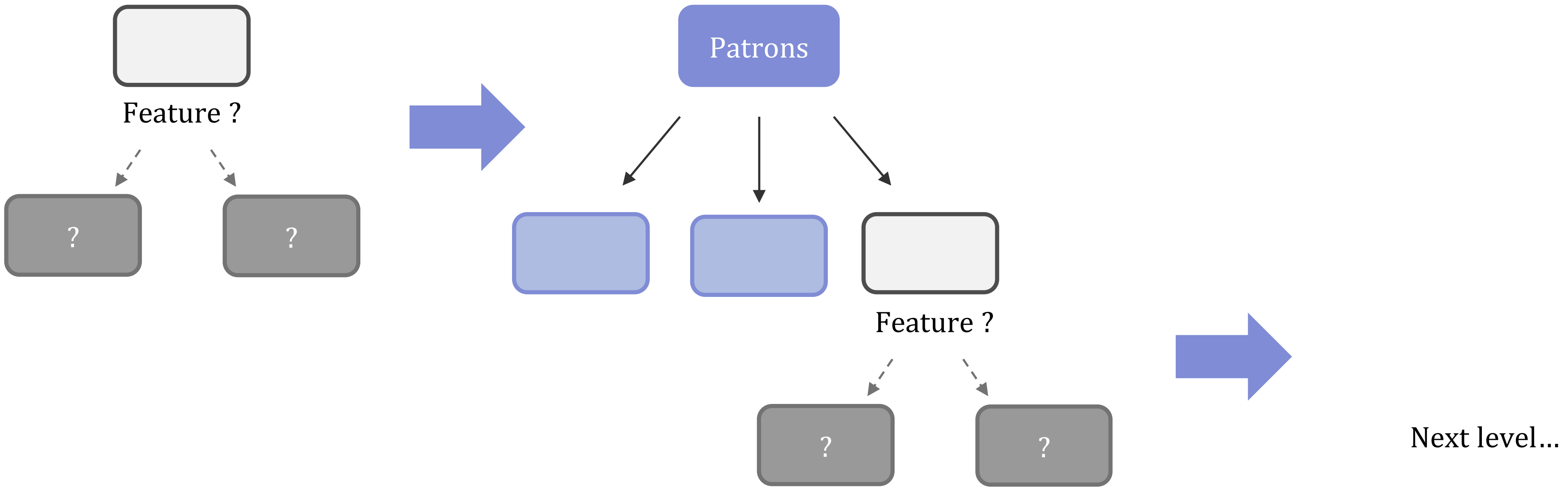


LABEL: WILL WAIT?



Decision Tree Induction

To make a tree: determine the best feature to split at every level based on node homogeneity/purity



Decision Tree: Advantages

- Easy to understand/interpret
- Minimal data preparation
  - Normalization not needed
  - Can handle numeric and categorical data
  - Easier to handle missing data

DEPARTMENT of INFORMATION SYSTEMS & COMPUTER SCIENCE

Decision Tree: Advantages

- Relatively fast for both induction and application
  - Induction: making a decision tree from a dataset
  - Application: using a decision tree to arrive at a predicted label

Decision Tree

- Hierarchical structure: Tree
  - Nodes → features
  - Branches → feature values
  - Leaf → labels/classes
- Inducing Tree:
  - Selecting attributes of highest importance (homogeneity/purity)
  - Splitting attribute on type
- Flexible, fast, and interpretable