



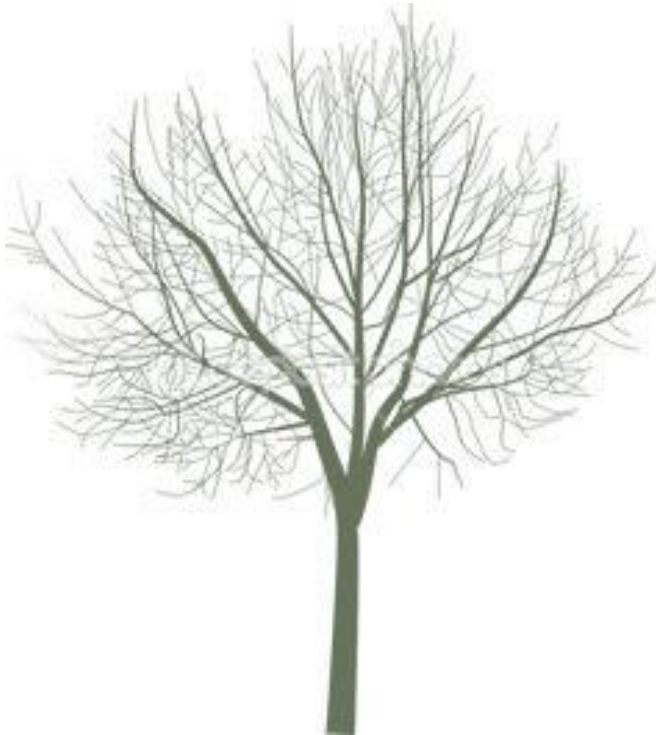
DECISION TREES AND RANDOM FORESTS

MSIS 4263

Obi Ogbanufe, PhD

DECISION TREE

Decision Tree is a **supervised learning** algorithm used mainly for solving classification problems



CLASSIFICATION

**Classification
And
Regression
Tree**

CART

REGRESSION

CLASSIFICATION AND REGRESSION TREES

Classification Trees

Classification Trees are used when the Dependent Variable is **categorical**

Categorical variables represent types of data which may be divided into groups or buckets: Race, Gender, Educational Level, Major etc.

Regression Trees

Regression Trees are used when the Dependent Variable is **continuous**

Continuous variables represent data with intervals between measurements: weight, height, price, speed etc.

DECISION TREE

A **decision tree** involves a set of nested tests which we use to “divide and conquer” a prediction problem

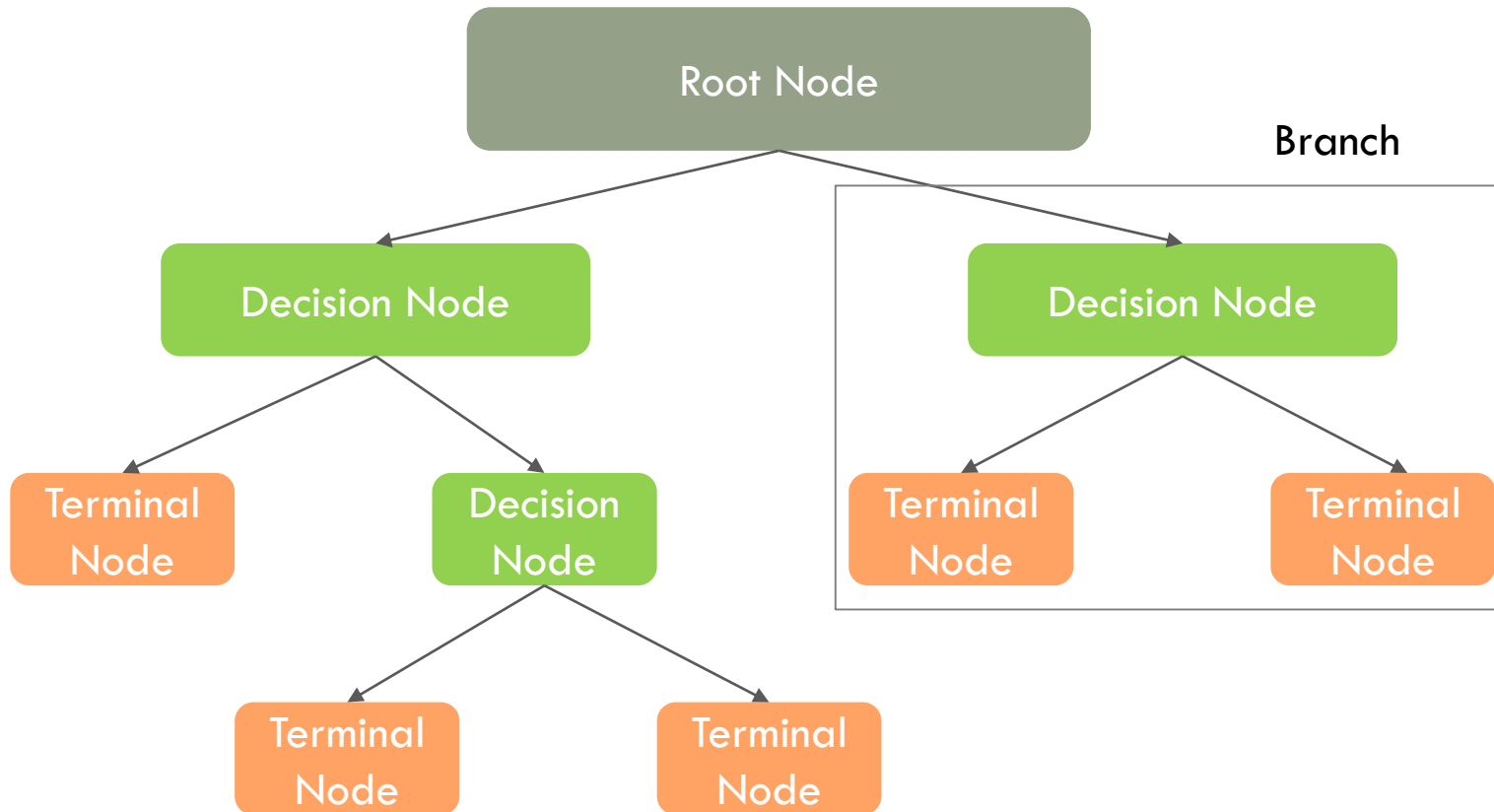
- Each **branch** represents a test (a decision)
- Each **node** represents the result of a test
- Each **leaf** (terminal node) represents a class assignment

Once developed, a decision tree can be used to classify new instances



DECISION TREE

A decision tree is a flow chart like structure

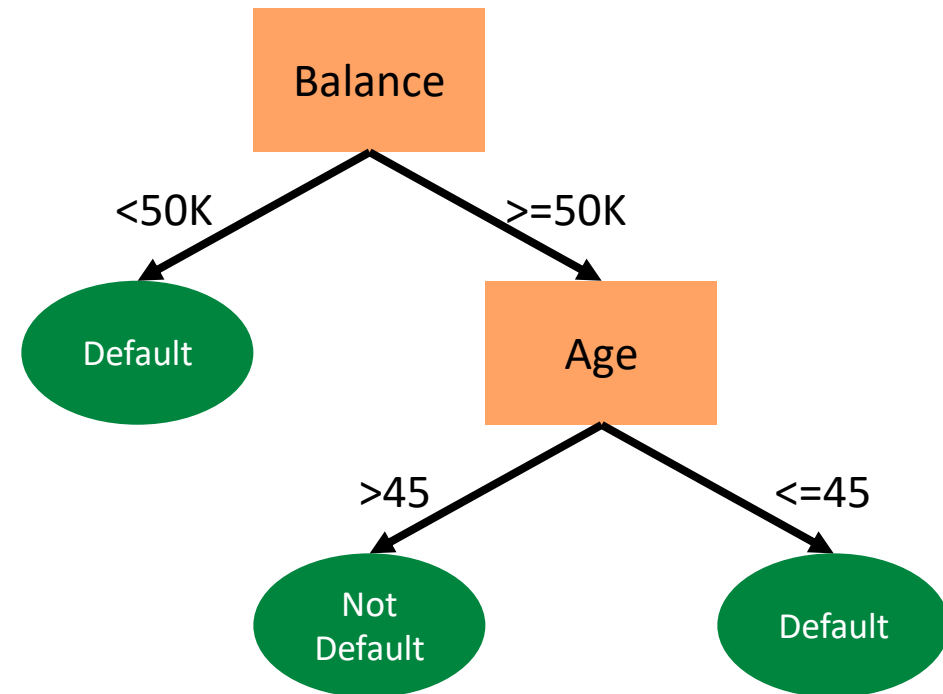


A SIMPLE EXAMPLE

We want a model that tells us if a prospective borrower is likely to default on a loan

We have historical data about past borrowers that can be used to train a model

Name	Balance	Age	Default
Mike	\$23,000	30	Yes
Mary	\$51,000	40	Yes
Bill	\$68,000	55	No
Jim	\$74,000	46	No
Dave	\$23,000	47	Yes
Anne	\$100,000	49	No



DECISION TREES VS. LOGISTIC REGRESSION

Since, **classification trees** use class (categorical variables), what's the difference between this and logistic regression?

Decision Trees

1. Classification Trees uses **categorical data (including binary data)** as **dependent variable**
2. Does not use equations and coefficients, instead has **predicted probabilities**

Logistic Regression

1. Logistic regression uses **only binary data** as **dependent variable**
2. Uses **equations and coefficients**



THE CULTIVATION OF TREES

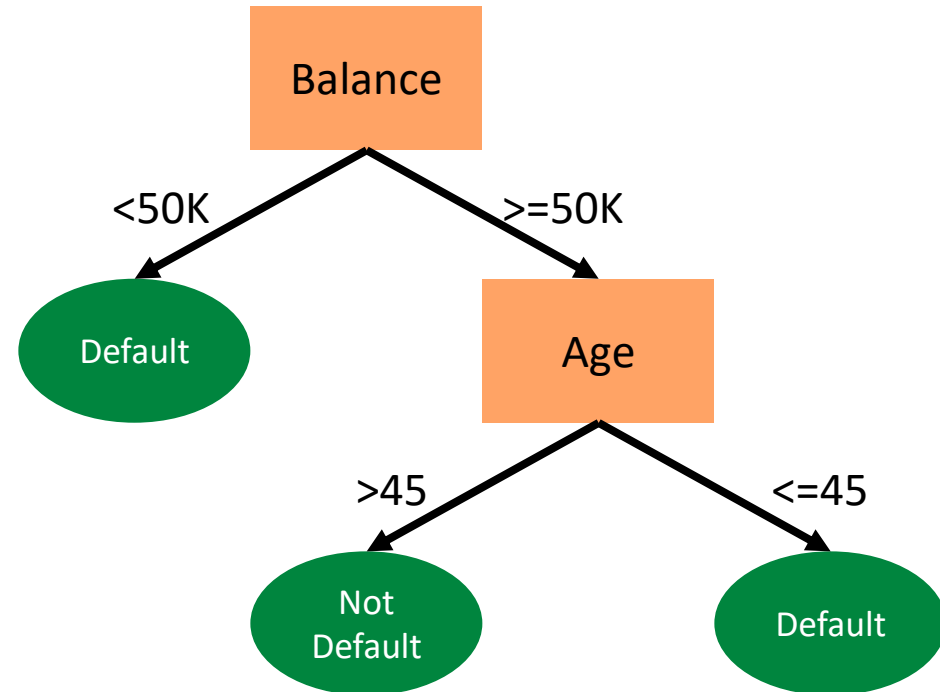
- **Split Search**
 - Which splits are to be considered?
- **Splitting Criterion**
 - Which split is best?
- **Stopping Rule**
 - When should the splitting stop?
- **Pruning Rule**
 - Should some branches be lopped off?

A SIMPLE EXAMPLE

Now we are presented with new data and asked to make a prediction

Mark, age 40, has an account balance of \$88,000

Would you predict Mark will default or not default?



DECISION TREE CONSTRUCTION

- Tree construction is performed in a top-down, recursive, divide-and-conquer manner
 - All training examples begin in the root node
 - Attributes are assumed to be categorical (or nominal) variables
 - *Attributes could be continuous (interval) variables*
 - Training examples are partitioned recursively based on selected attributes
- Decision tree algorithms are considered **greedy** because they make locally optimal decisions

GREEDY ALGORITHM

An algorithm that always takes the best immediate, or **local**, solution while finding an answer at each step.

A greedy algorithm chooses the best possible answer in each step and then moves on to the next step until it reaches the end. It does so without regard for the overall solution or consequences.

A greedy algorithm does not consider the overall **global** picture, hence the term greedy.

DECISION TREE ALGORITHM

The most commonly used algorithm was discovered independently by two separate researchers

- **CaRT** – Breimanetal 1984
- **ID3** – Quinlan 1986

The algorithm consists of the following:

1. Using your training data, select the best attribute to split on
2. Identify all possible values for that attribute
3. For each value, create a new child node
4. Allocate the observations to the appropriate child node
5. For each child node
 - If the node is **pure**, STOP
 - Else, recursively call the algorithm to split again

AN EXAMPLE OF DECISION TREE CLASSIFICATION

Outlook	Temp	Humidity	Windy	Play?
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

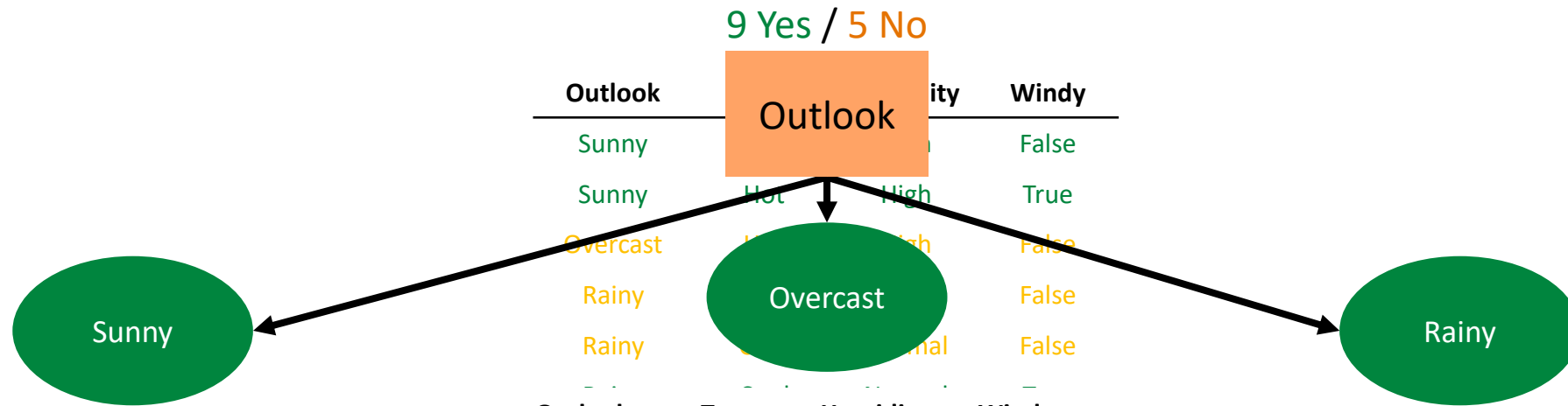
Task: Develop a decision to
predict if John will play tennis

9 Yes

5 No

14 Total

BUILDING OUR DECISION TREE



Outlook	Temp	Humidity	Windy
Sunny	Hot	High	False
Sunny	Hot	High	False
Sunny	Mild	High	False
Sunny	Cool	Normal	False
Sunny	Mild	Normal	True

2 Yes / 3 No

Split Further

Outlook	Temp	Humidity	Windy
Overcast	Hot	High	False
Overcast	Cool	Normal	True
Overcast	Mild	High	True
Overcast	Hot	Normal	False
Sunny	Mild	Normal	True
Overcast	Mild	High	True
Overcast	Hot	Normal	False
Rainy	Mild	High	True

4 Yes / 0 No

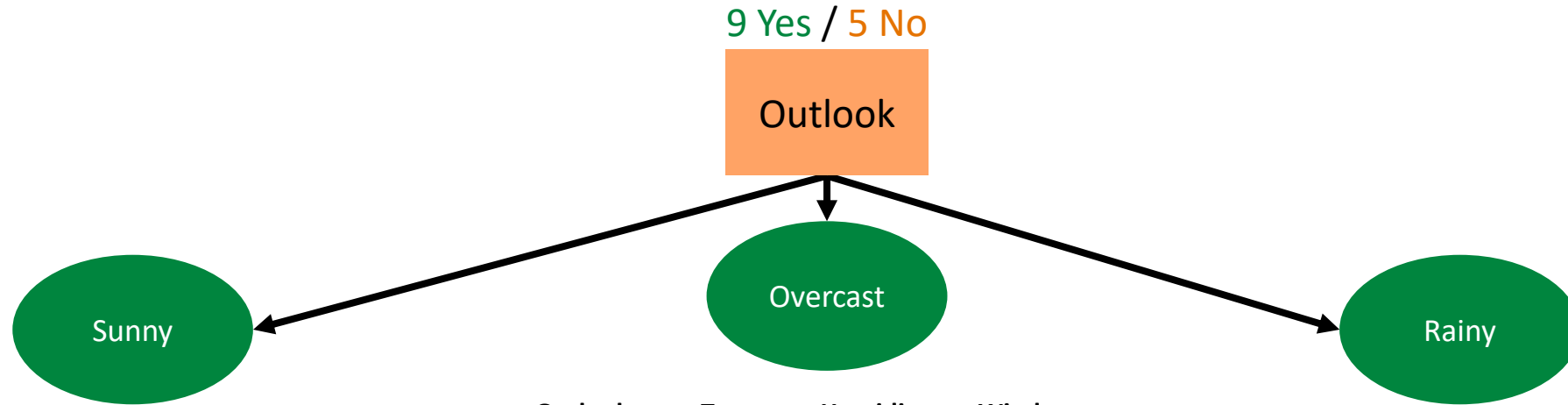
Pure Subset

Outlook	Temp	Humidity	Windy
Rainy	Mild	High	False
Rainy	Cool	Normal	False
Rainy	Cool	Normal	True
Rainy	Mild	Normal	False
Rainy	Mild	High	True

2 Yes / 3 No

Split Further

BUILDING OUR DECISION TREE



Outlook	Temp	Humidity	Windy
Sunny	Hot	High	False
Sunny	Hot	High	False
Sunny	Mild	High	False
Sunny	Cool	Normal	False
Sunny	Mild	Normal	True

2 Yes / 3 No

Split Further

Outlook	Temp	Humidity	Windy
Overcast	Hot	High	False
Overcast	Cool	Normal	True
Overcast	Mild	High	True
Overcast	Hot	Normal	False

4 Yes / 0 No

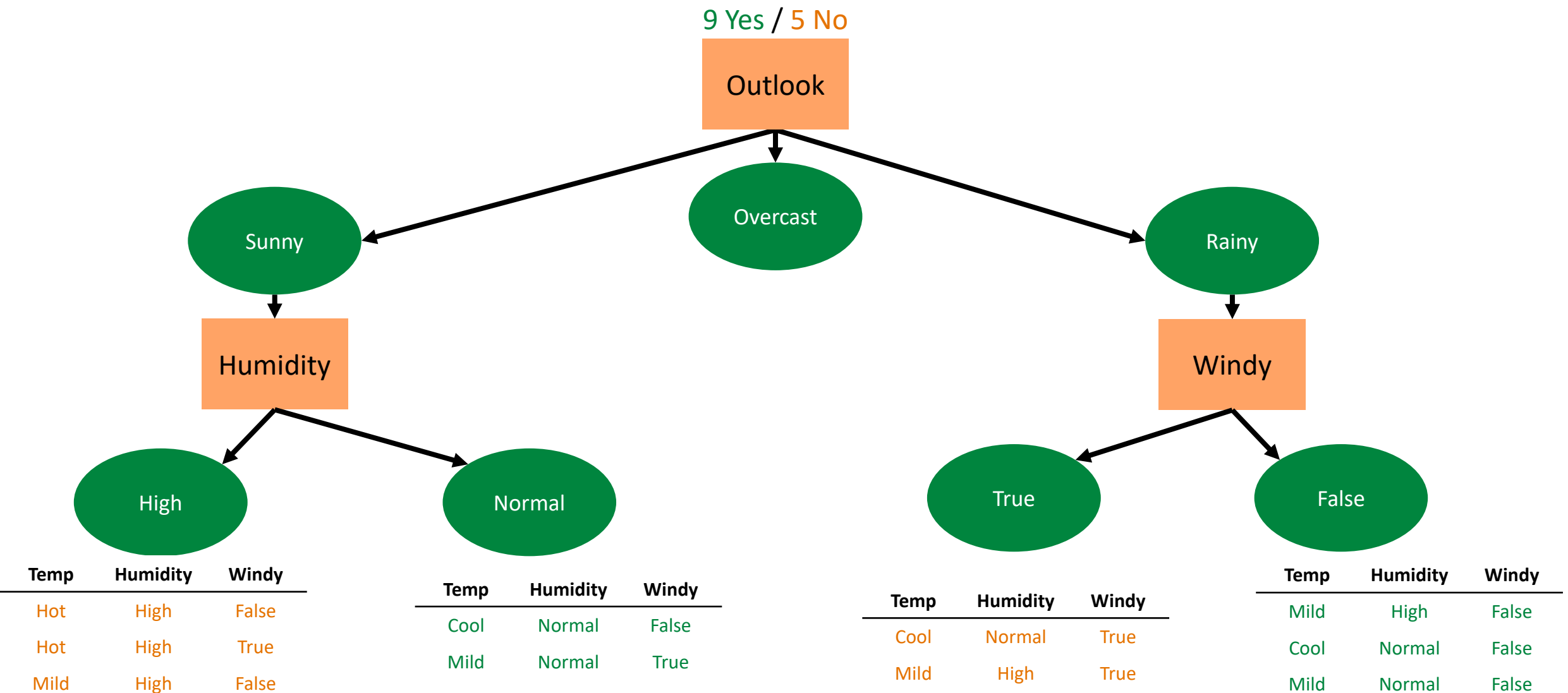
Pure Subset

Outlook	Temp	Humidity	Windy
Rainy	Mild	High	False
Rainy	Cool	Normal	False
Rainy	Cool	Normal	True
Rainy	Mild	Normal	False
Rainy	Mild	High	True

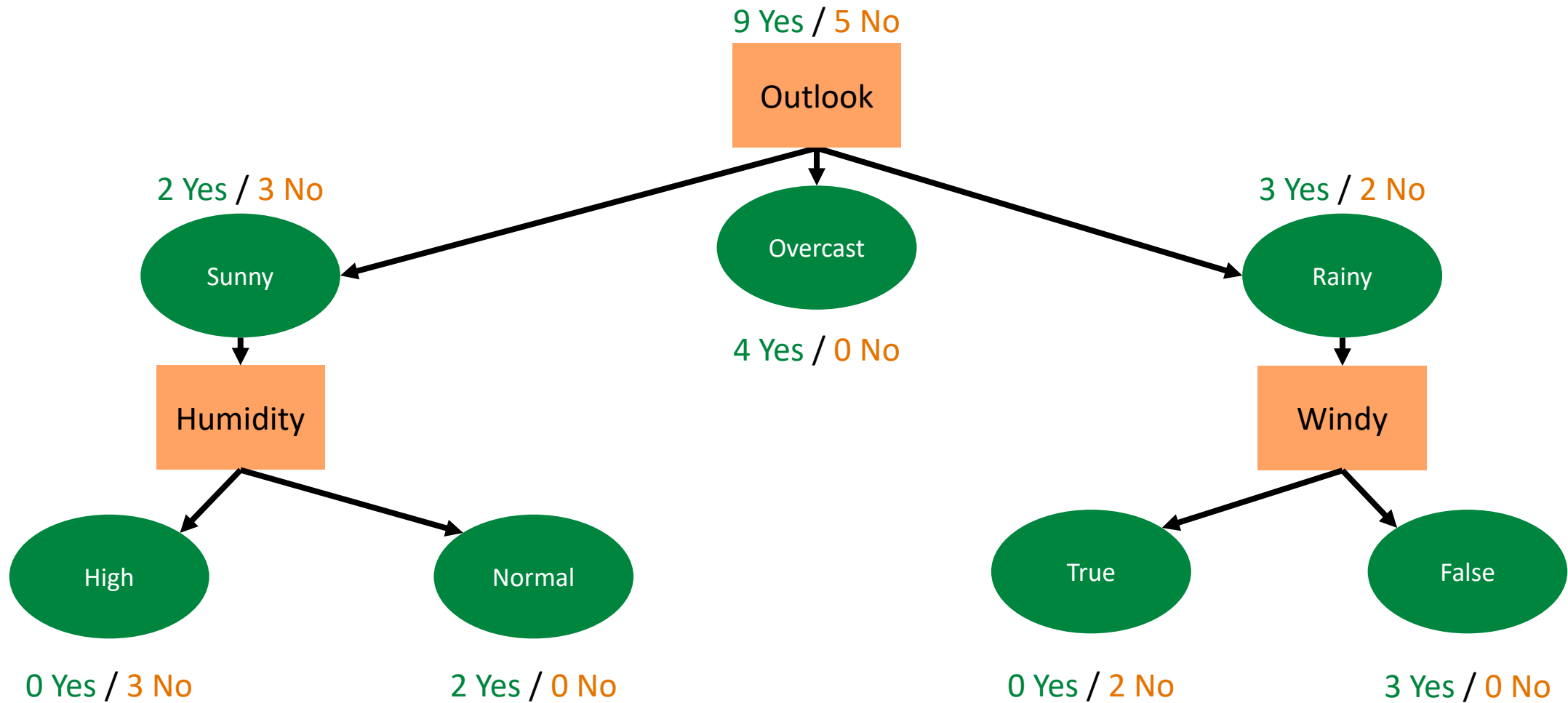
2 Yes / 3 No

Split Further

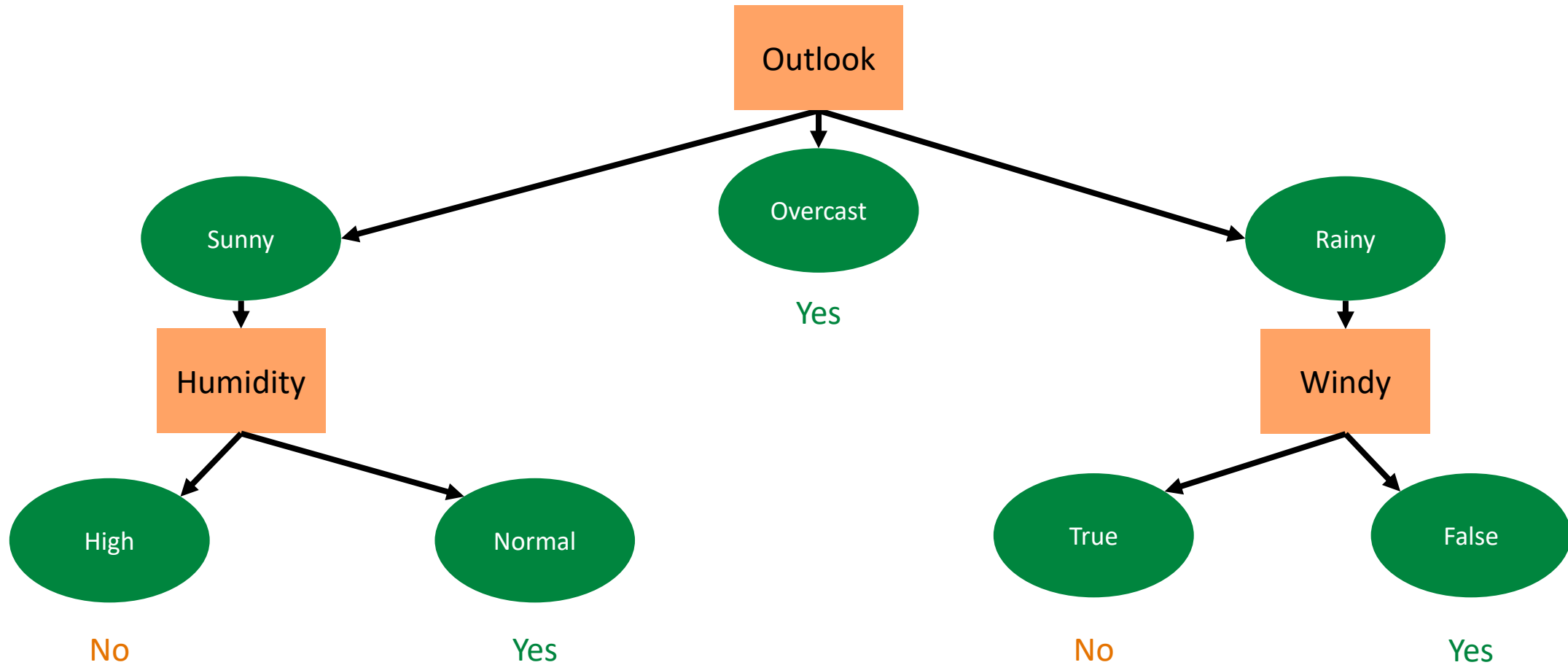
BUILDING OUR DECISION TREE



BUILDING OUR DECISION TREE

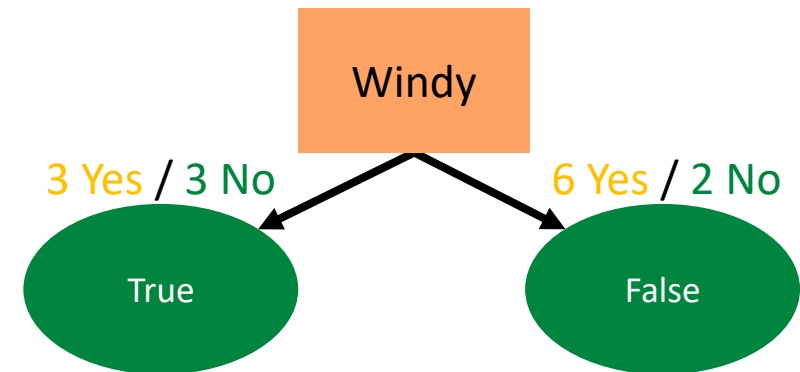
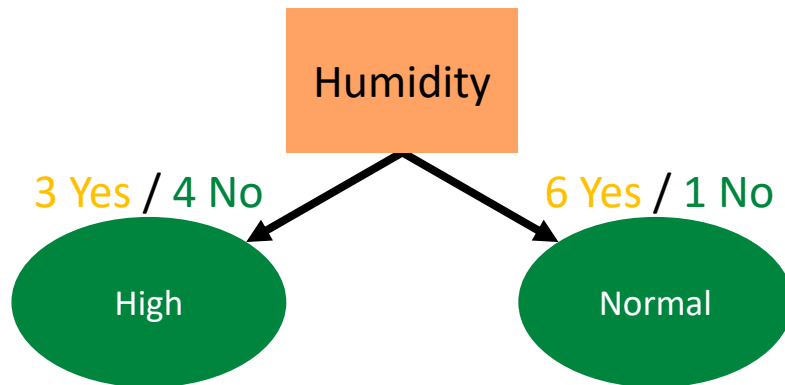
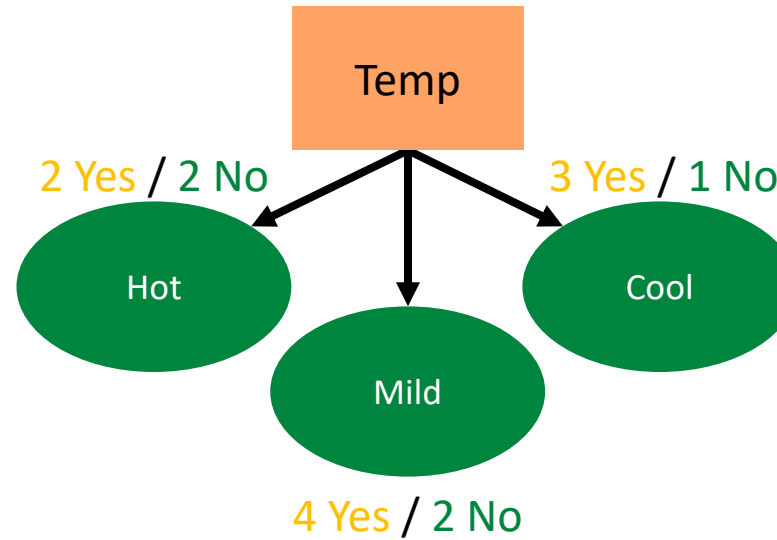
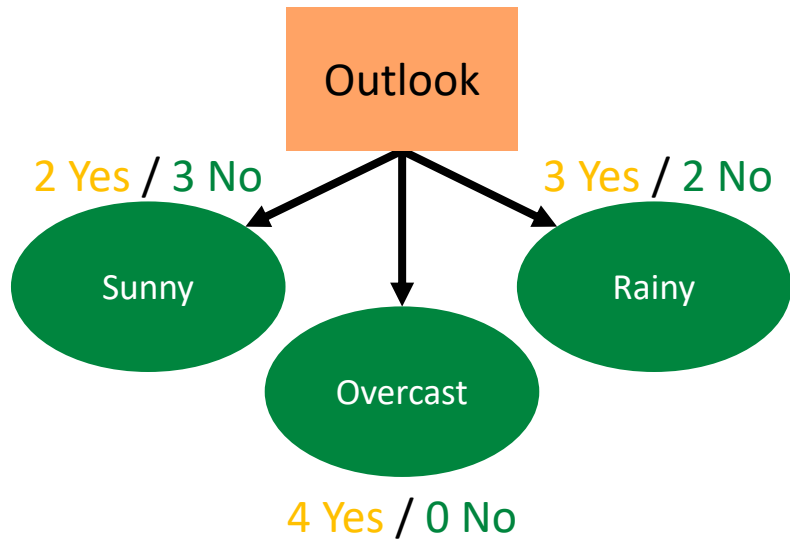


BUILDING OUR DECISION TREE



Outlook	Temp	Humidity	Windy	Play?
Sunny	Cool	High	True	?

SELECTING THE BEST ATTRIBUTES FOR SPLITS



A CRITERION FOR ATTRIBUTE SELECTION

A heuristic for attribute selection is simply to choose the attribute that minimizes the number of splits in the tree

However, it would be nice to have a measure to identify the best attribute

- Selecting the best attribute largely involves assessing **certainty**
 - For a node with 4 Yes/0 No or 0 Yes/4 No, we are relatively certain
 - For a node with 2 Yes/2 No, we are not certain at all

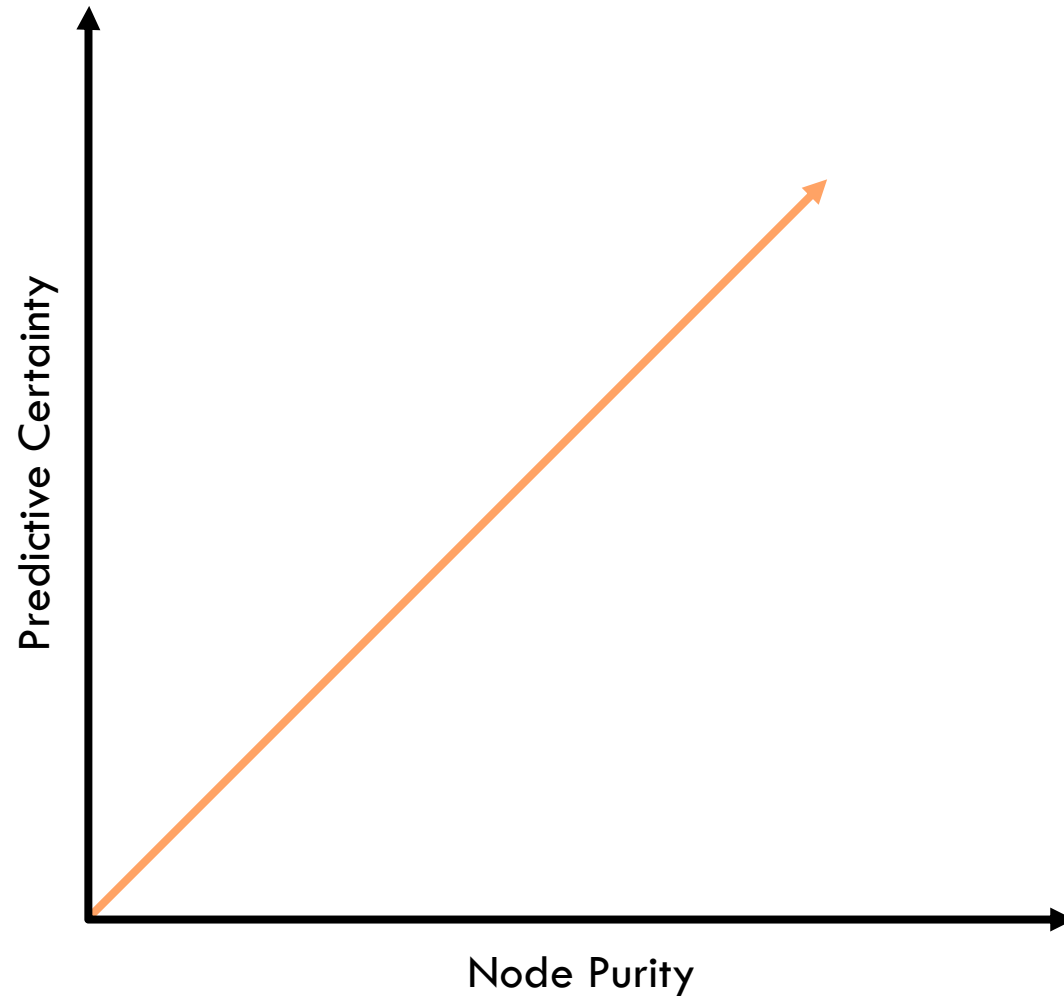


A CRITERION FOR ATTRIBUTE SELECTION

- For **pure nodes**, we are very certain
- For **impure nodes**, we are not certain
- A popular (im)purity criterion is **information**
 - Specifically, how much additional information is needed to be sure about a classification
 - It takes a low value of information for pure nodes and a high value for impure nodes
- Strategy: Choose the attribute that results in the highest degree of certainty



RELATIONSHIP BETWEEN PURITY AND CERTAINTY



ENTROPY

A measurement of uncertainty
of a class in a data subset



COMPUTING ENTROPY

- **Entropy** provides the information required to predict an event with certainty
- Information is measured in **bits**

$$Entropy = - \sum_{k=1}^m q_k \log_2 q_k = -q_1 \log_2 q_1 - q_2 \log_2 q_2 \dots - q_m \log_2 q_m$$

Where

- m is the number of classes
- q_k is the proportion of records belonging to class k

COMPUTING ENTROPY

$$Entropy = -q_1 \log_2 q_1 - q_2 \log_2 q_2 \dots - q_m \log_2 q_m$$

$$Entropy_{Sunny} = -\frac{2}{5} \log_2 \frac{2}{5} - \frac{3}{5} \log_2 \frac{3}{5}$$

$$Entropy_{Sunny} = 0.53 + 0.44 = 0.97 \text{ bits}$$

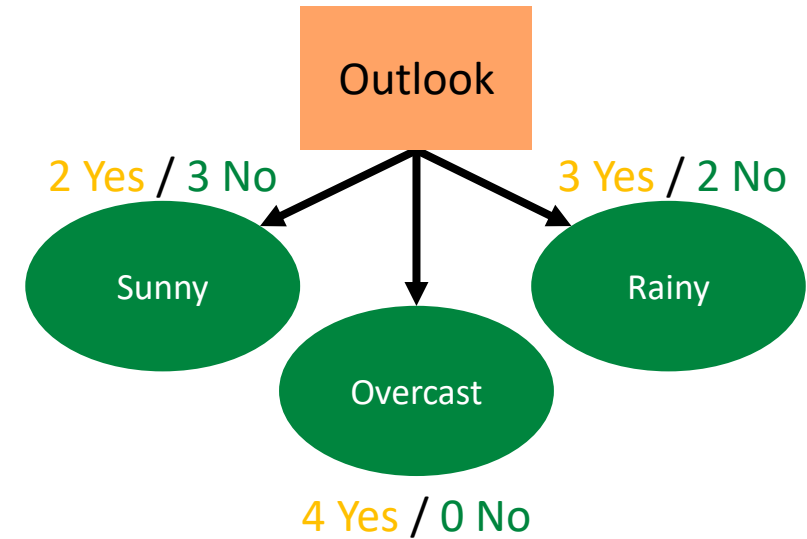
$$Entropy_{Overcast} = -\frac{4}{4} \log_2 \frac{4}{4} - \frac{0}{4} \log_2 \frac{0}{4}$$

$$Entropy_{Overcast} = 0.00 + \text{undefined} = 0.00 \text{ bits}$$

$$Entropy_{Rainy} = -\frac{3}{5} \log_2 \frac{3}{5} - \frac{2}{5} \log_2 \frac{2}{5}$$

$$Entropy_{Rainy} = 0.44 + 0.53 = 0.97 \text{ bits}$$

$$TWE_{Outlook} = \frac{5}{14} \cdot 0.97 + \frac{4}{14} \cdot 0 + \frac{5}{14} \cdot 0.97 = 0.69 \text{ bits}$$



q	-q log ₂ q
1	0.00
1/2	0.50
1/3	0.53
2/3	0.39
1/4	0.50
3/4	0.31
1/5	0.46
2/5	0.53
3/5	0.44
4/5	0.26
1/6	0.43
5/6	0.22 ²⁷

COMPARING TOTAL WEIGHTED ENTROPIES

The total weighted entropy is a measure of uncertainty if we select the associated node for splitting

Given the total weighted entropies for the tennis attributes, which attribute should we split on?

$$TWE_{Outlook} = 0.69 \text{ bits}$$

$$TWE_{Temperature} = 0.91 \text{ bits}$$











$$TWE_{Humidity} = 0.79 \text{ bits}$$

$$TWE_{Windy} = 0.89 \text{ bits}$$



Outlook has the smallest total weighted entropy (i.e. it requires the least amount of information to be certain about prediction)

ANOTHER EXAMPLE

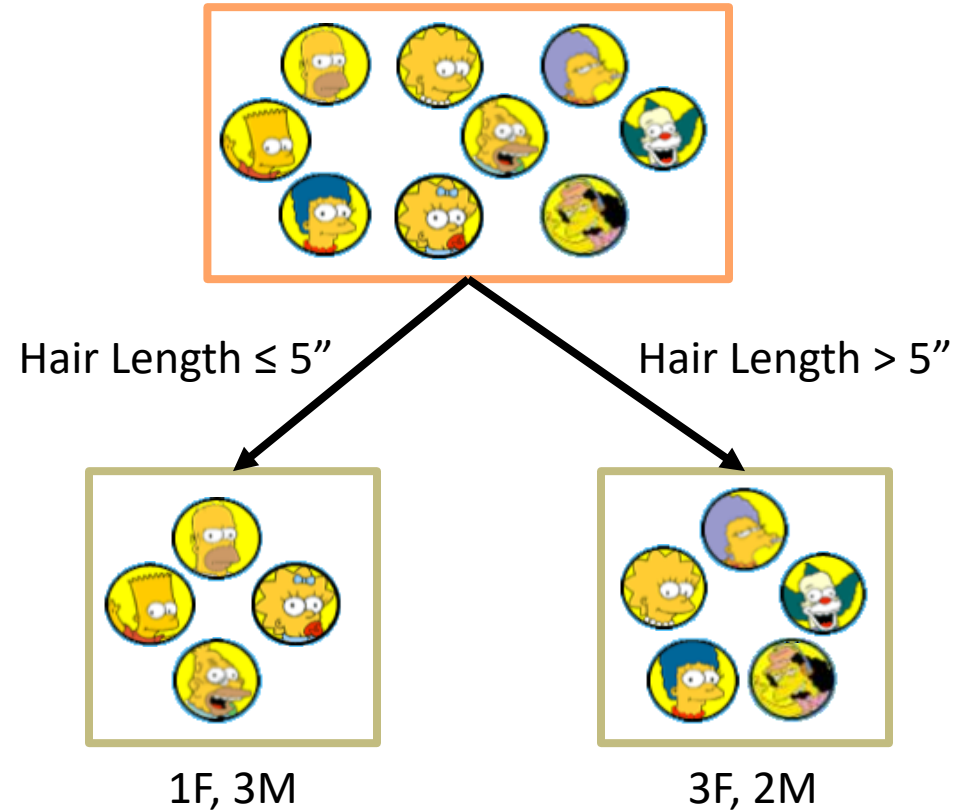
	<i>Person</i>	<i>Hair Length</i>	<i>Weight</i>	<i>Age</i>	<i>Class</i>
	Homer	0"	250	36	M
	Marge	10"	150	34	F
	Bart	2"	90	10	M
	Lisa	6"	78	8	F
	Maggie	4"	20	1	F
	Abe	1"	170	70	M
	Selma	8"	160	41	F
	Otto	10"	180	38	M
	Krusty	6"	200	45	M
	Comicbook Guy	8"	290	38	???

SPLITTING ON HAIR LENGTH

$$Entropy_{\leq 5} = -\frac{1}{4}\log_2\frac{1}{4} - \frac{3}{4}\log_2\frac{3}{4} = 0.81 \text{ bits}$$

$$Entropy_{>5} = -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5} = 0.97 \text{ bits}$$

$$TWE_{Hair} = \frac{4}{9} \cdot 0.81 + \frac{5}{9} \cdot 0.97 = 0.90 \text{ bits}$$

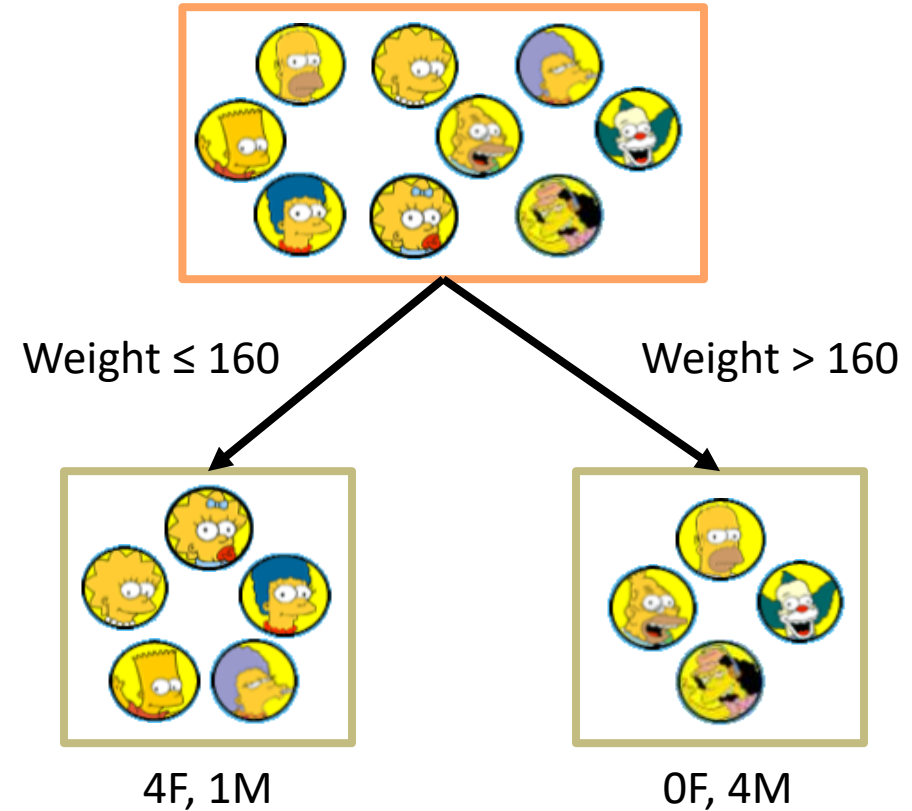


SPLITTING ON WEIGHT

$$Entropy_{\leq 160} = -\frac{4}{5} \log_2 \frac{4}{5} - \frac{1}{5} \log_2 \frac{1}{5} = 0.72 \text{ bits}$$

$$Entropy_{> 160} = -\frac{0}{4} \log_2 \frac{0}{4} - \frac{4}{4} \log_2 \frac{4}{4} = 0.00 \text{ bits}$$

$$TWE_{weight} = \frac{5}{9} \cdot 0.72 + \frac{4}{9} \cdot 0.00 = 0.40 \text{ bits}$$

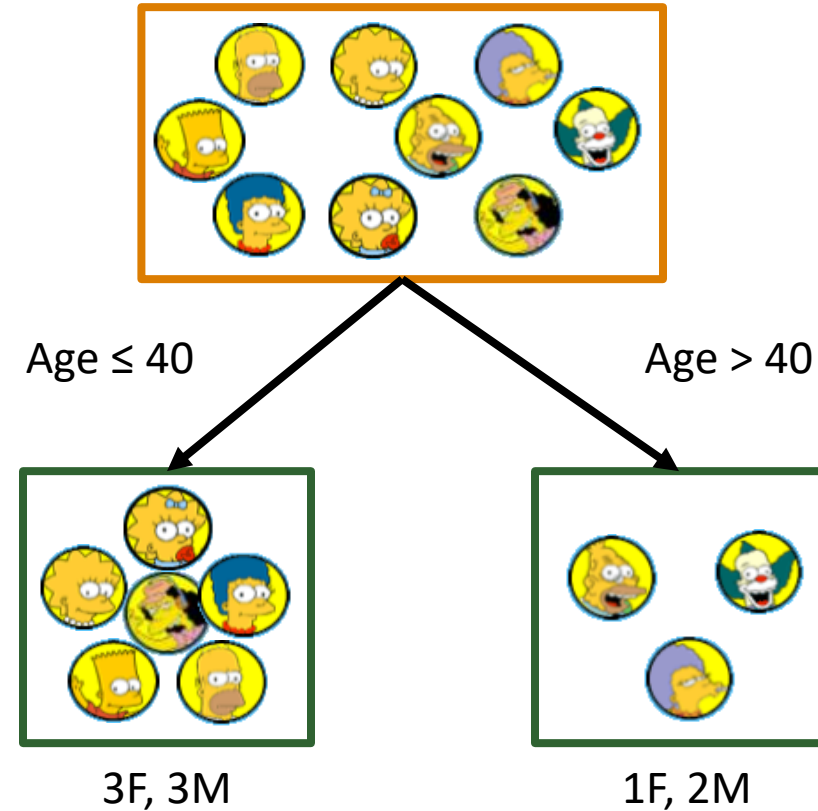


SPLITTING ON AGE

$$Entropy_{\leq 40} = -\frac{3}{6}\log_2\frac{3}{6} - \frac{3}{6}\log_2\frac{3}{6} = 1.00 \text{ bits}$$

$$Entropy_{>40} = -\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3} = 0.92 \text{ bits}$$

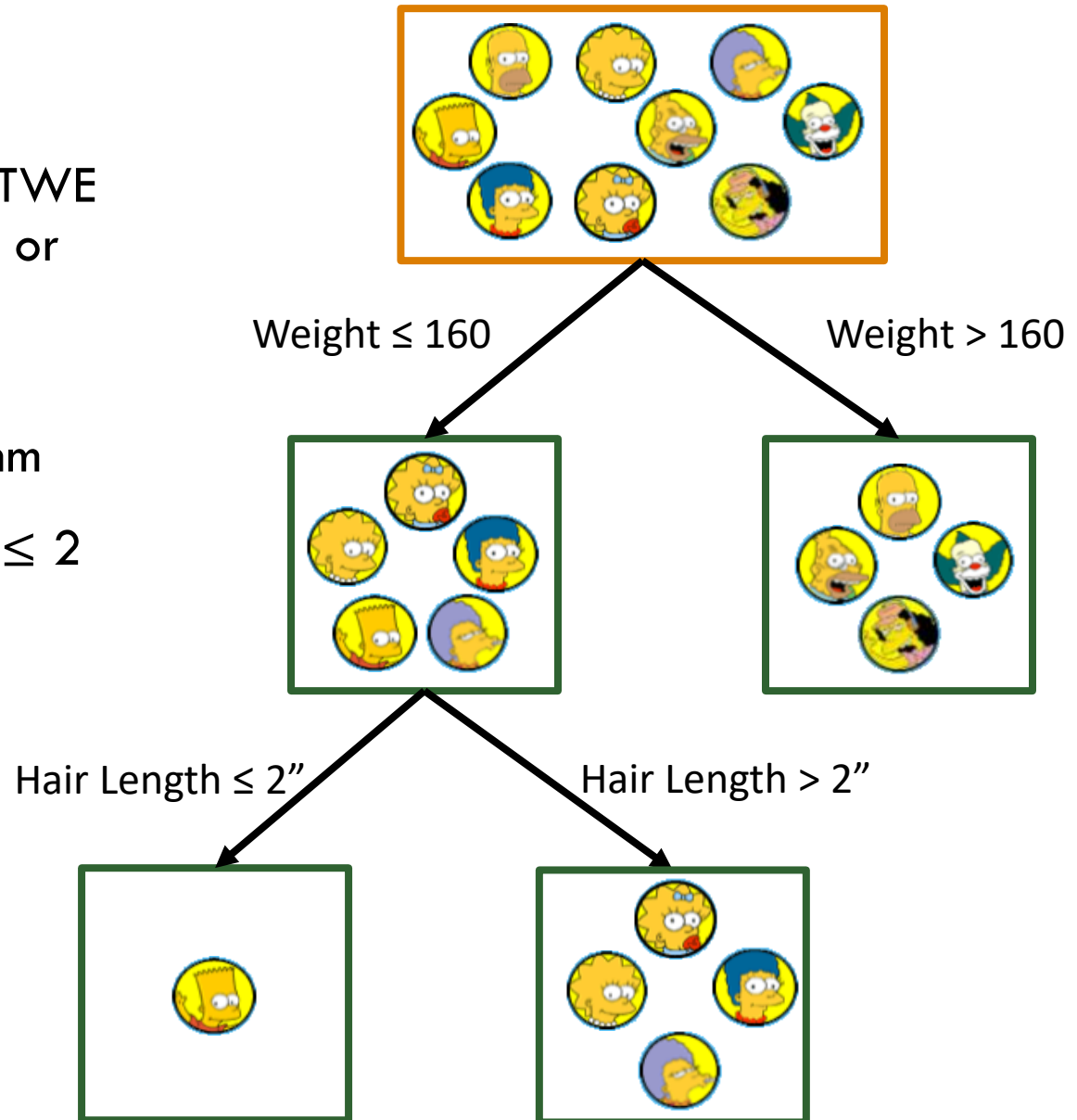
$$TWE_{Age} = \frac{6}{9} \cdot 1.00 + \frac{3}{9} \cdot 0.92 = 0.97 \text{ bits}$$



Splitting on weight yields the lowest TWE
(0.40 bits versus 0.90 for hair length or
0.97 for age)

The weight ≤ 160 node is not pure,
therefore recursively call the algorithm

At the next level, split on hair length ≤ 2
is best



TREES ARE EASILY CONVERTED TO RULES

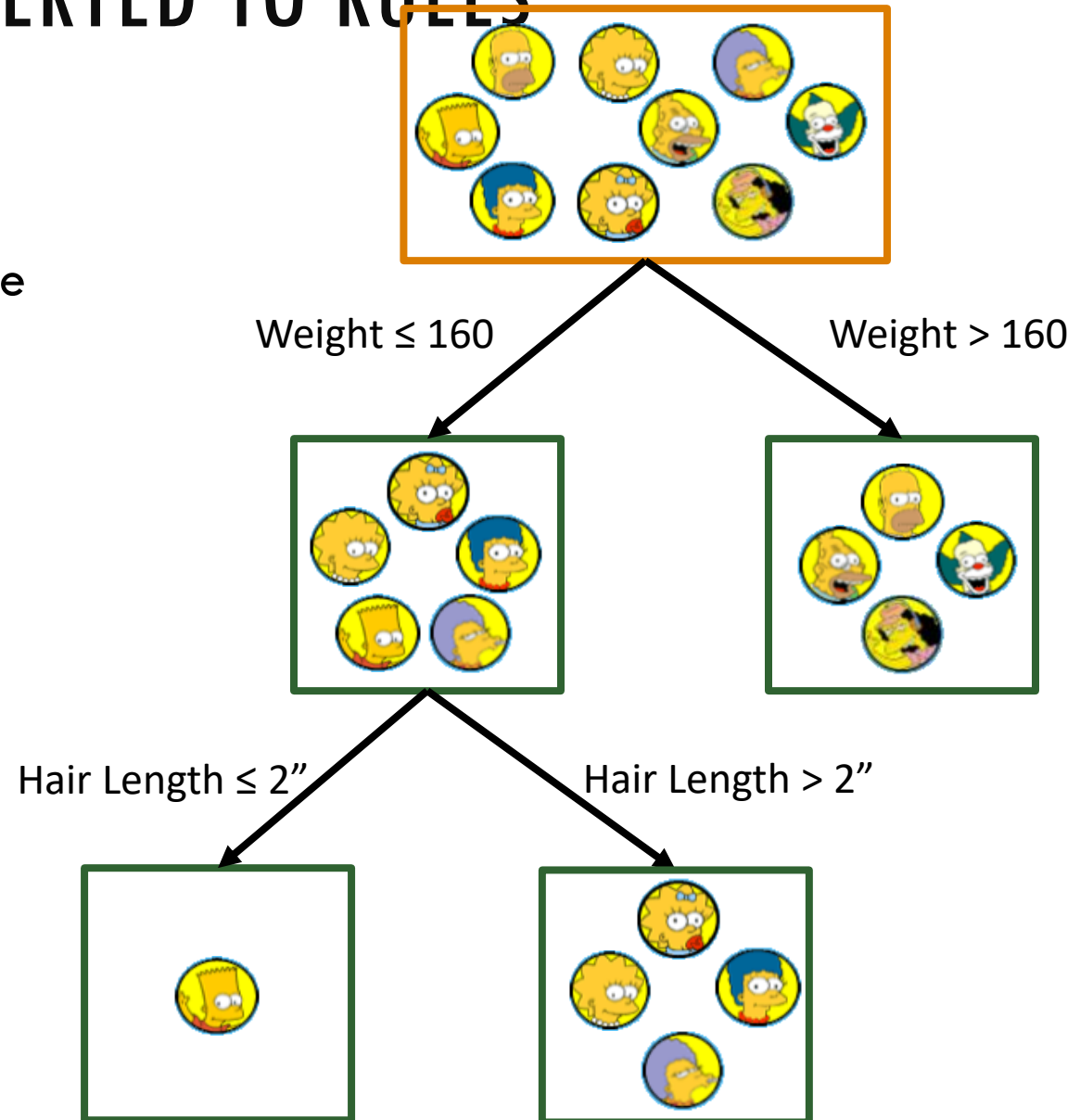
IF (Weight > 160) THEN Male

ELSE IF (Hair Length ≤ 2") THEN Male

ELSE Female

So what about Comicbook Guy?

- Hair Length: 8"
- Weight: 290
- Age: 38
- Class: **Male**

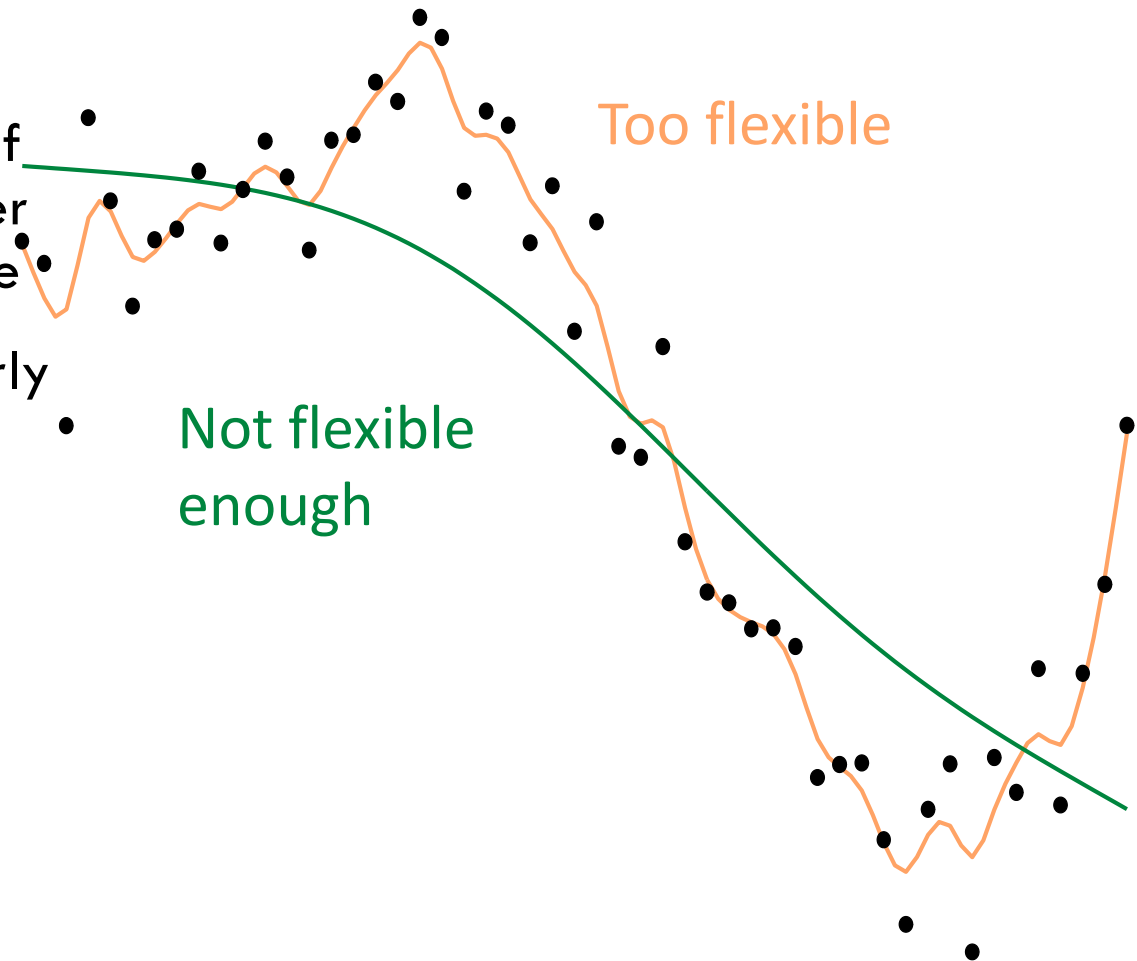


OVERFITTING

Remember that for supervised techniques, we typically use one part of our data to **train** the model and another part of our data to **test** its performance

Overfitting occurs when we use an overly flexible model that accommodates the nuances of the random noise in the **training data**

As a result, the model performs poorly when presented with the **testing data**



OVERFITTING TREES

In the context of decision trees, overfitting occurs when the tree has too many branches

- You end up fitting noise
- Great fit for training data, poor accuracy for unseen samples

Avoiding Overfitting

- Pre-pruning (Stunting)
 - Stop splitting if the number of cases in a node falls below a specified limit
 - Stop splitting if the split is not statistically significant at a specified level
 - It is difficult to choose a criterion
- Post-pruning
 - Remove branches from a “fully grown” tree
 - Select from a sequence of progressively pruned trees



SOME COMMON ISSUE

- Learning things that aren't true - Patterns may not represent any underlying rule or the rule may be obscure event where a relationship is known to exist (vanilla ice cream → car trouble)
- Learning things that are true, but not useful – Most rules learned are normal business rules (already known)
- Data integrity issues

STRENGTHS AND WEAKNESSES

Strengths

- Easy to interpret and visualize
- Easy to implement
- Relatively efficient
- Can handle mixed measurement scales
- Can handle missing values
- Relatively robust
- Extremely popular
- Little effort in data pre-processing

Weaknesses

- Volatile
- Sensitive to outliers
- Can result in large errors

WHAT IS BETTER THAN ONE DECISION TREE?

A Forest



RANDOM FOREST

Random Forests are an **ensemble** approach to classification

- Rather than using a single decision tree, multiple trees are constructed
- Each tree performs the classification and the results are aggregated

Called random, because of variations in the construction of the trees

- Each tree predicts the target variable using a random subset of the available predictors
- Each tree uses a random sample of the available data

Results of each tree are examined collectively and majority rules

A SIMPLE RANDOM FOREST EXAMPLE

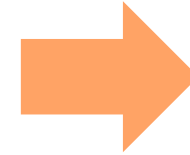
	A	B	C	D	E	F	G	H
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								



Tree 1

Predictors: A, C, D, F

Observations: 2, 3, 6, 8, 12, 16, 17, ...



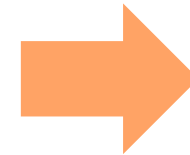
1



Tree 2

Predictors: B, D, E, G

Observations: 1, 2, 5, 7, 10, 14, 18, ...



0



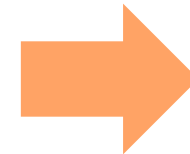
1



Tree 3

Predictors: B, C, E, G

Observations: 3, 4, 8, 9, 11, 13, 15, ...



1

