Decision Tree

Classification

Age	Income	City	Gender	**Response**
30	50K	New York	M	No
50	125K	Tampa	F	Yes
…	…	…	…	Yes
…	…	…	…	No
28	35K	Orlando	M	???
…	..	…	…	???
…	..	…	…	???

- Mapping instances onto a predefined set of classes.
- Examples
  - Classify each customers as “Responder” versus “Non-Responder”
  - Classify cellular calls as “legitimate” versus “fraudulent”

Prediction

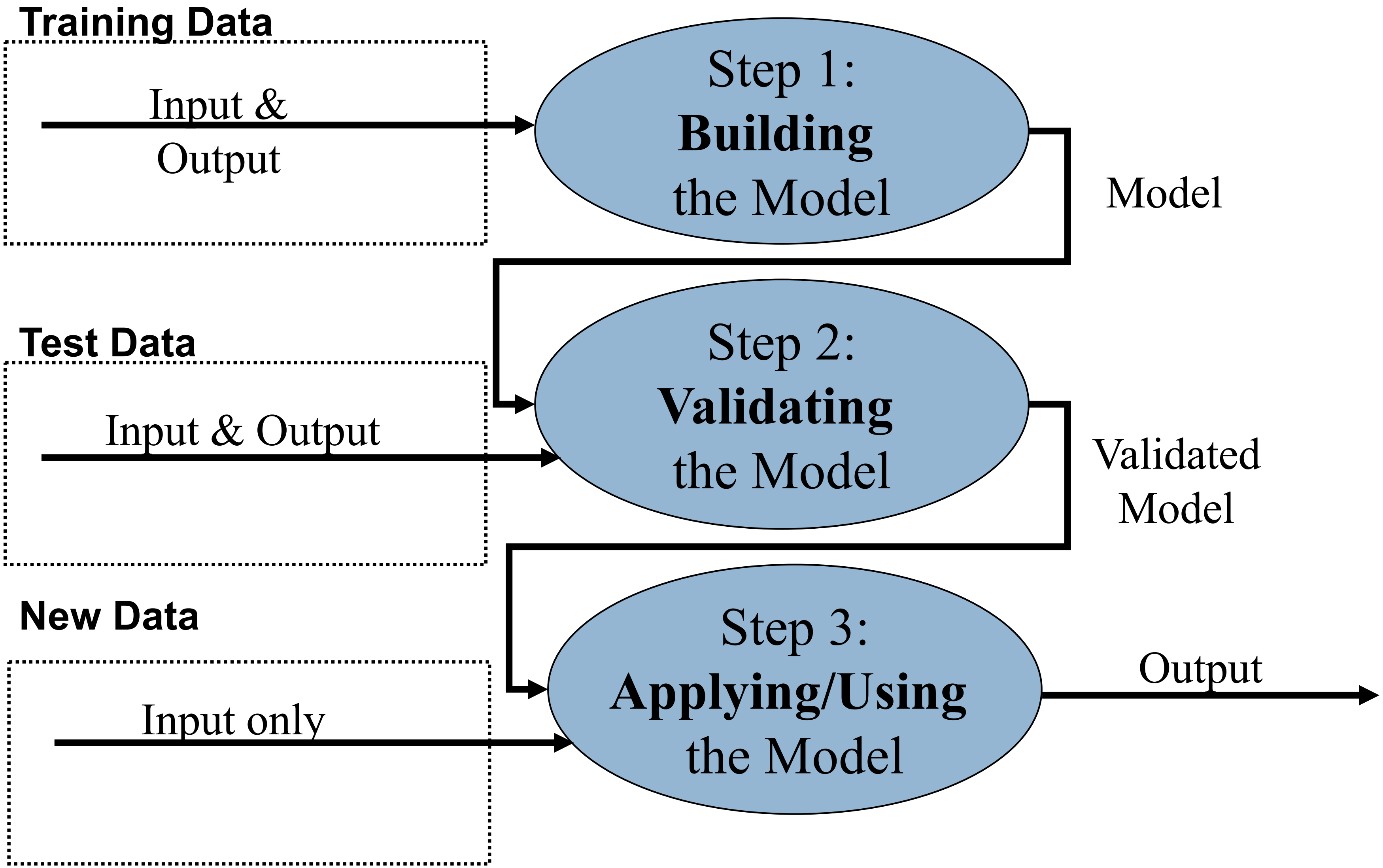
- Broad definition: build model to estimate any type of values (predictive data mining)
- Narrow definition: estimate continuous values
  - Examples
    - Predict how much money a customer will spend
    - Predict the value of a stock

Age	Income	City	Gender	**Dollar Spent**
30	50K	New York	M	$150
50	125K	Tampa	F	$400
…	…	…	…	$0
…	…	…	…	$230
28	35K	Orlando	M	???
…	..	…	…	???
…	..	…	…	???

Classification: Terminology

- Inputs = Predictors = Independent Variables
- Outputs = Responses = Dependent Variables
- Models = Classifiers
- Data points: examples, instances
- With classification, we want to build a (classification) model to predict the outputs given the inputs.

Steps in Classification

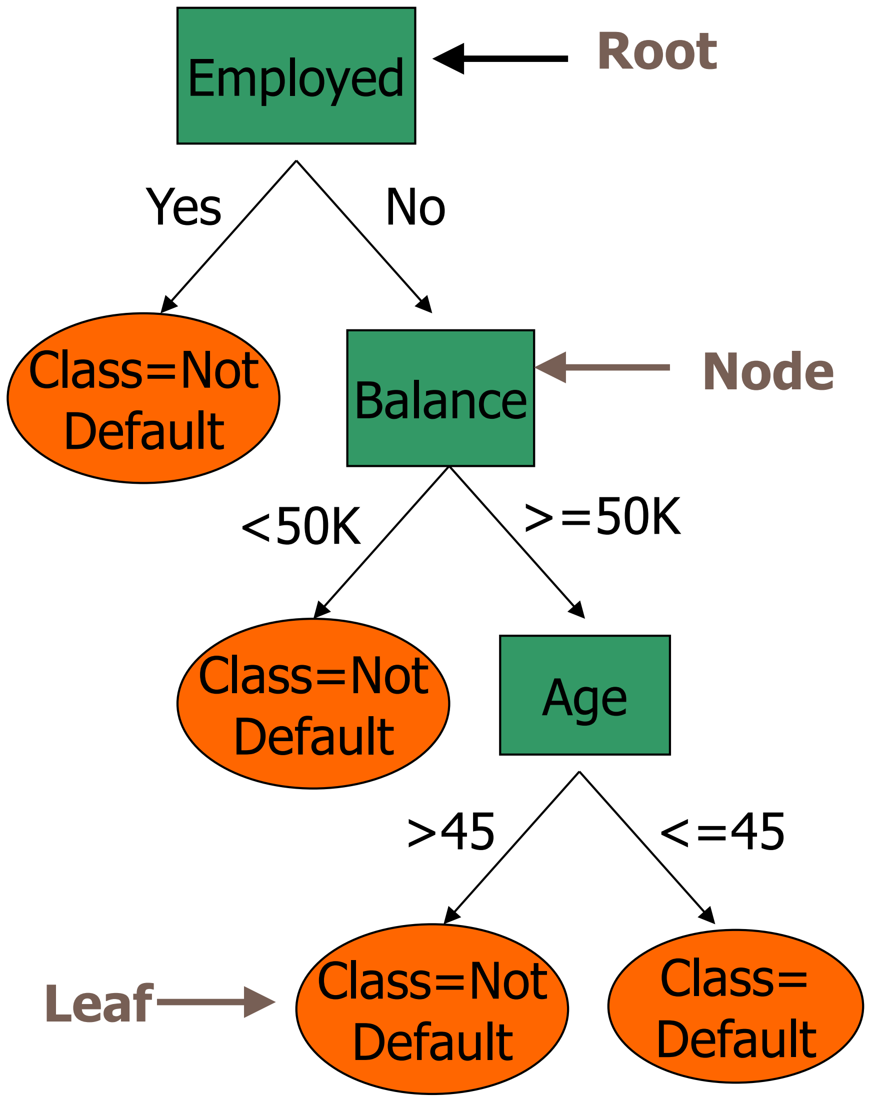


Common Classification Techniques

- Decision tree
- Logistics regression
- K-nearest neighbors
- Neural Network
- Naïve Bayes

Decision Tree --- An Example

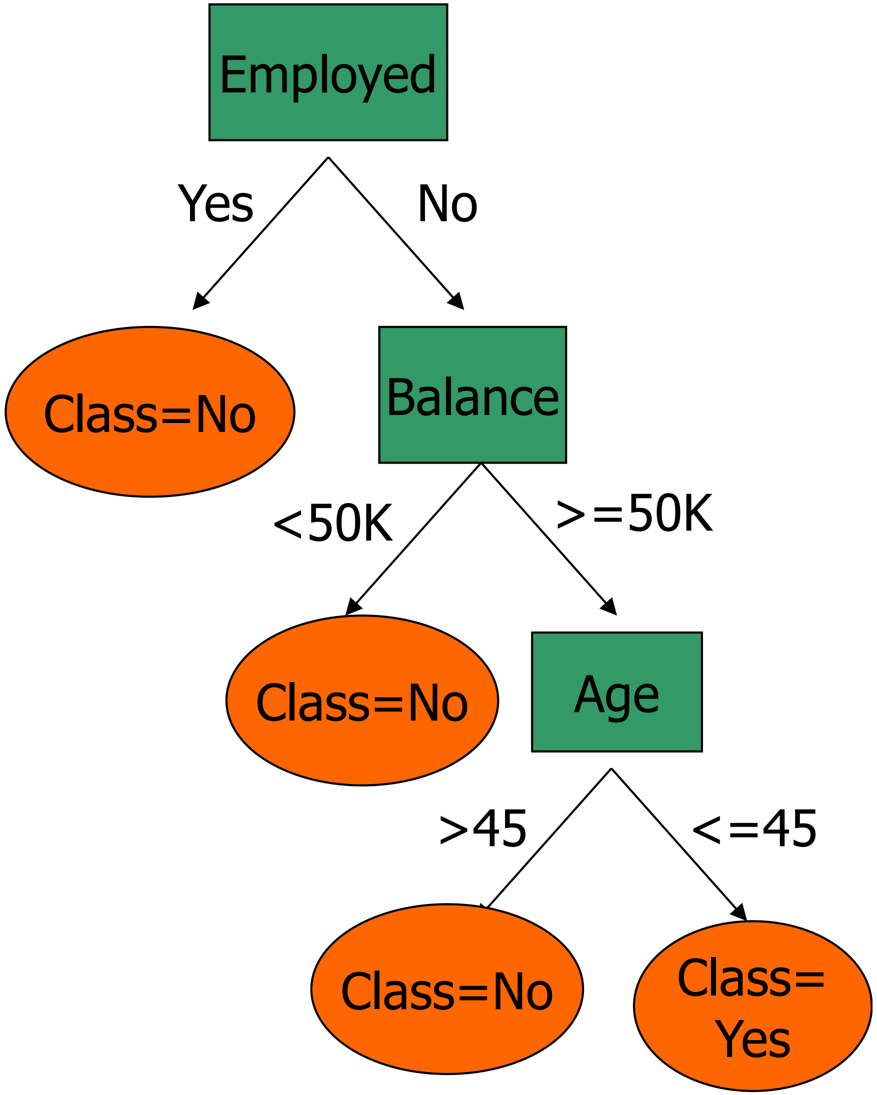
Name	Balance	Age	Emp.	Default
Mike	23,000	30	yes	no
Mary	51,100	40	yes	no
Bill	48,000	40	no	no
Jim	53,000	45	no	yes
Dave	65,000	60	no	no
Anne	30,000	35	no	no



Decision Tree Representation

A series of nested tests:

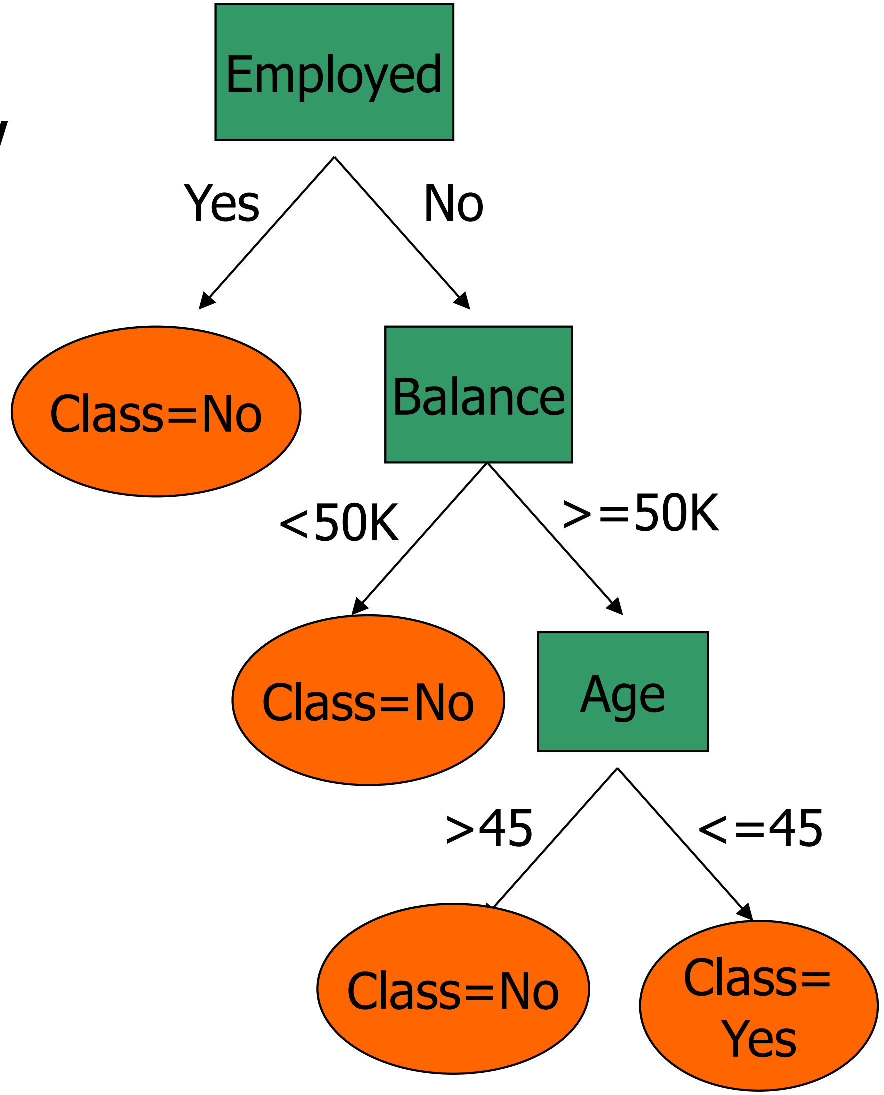
- Each **node** represents a test on one attribute
  - Nominal attributes: each branch could represent one or more values
  - Numeric attributes are split into ranges, normally binary split
- **Leaves**
  - A class assignment (E.g, Default /Not default)



The Use of a Decision Tree: Classifying New Instances

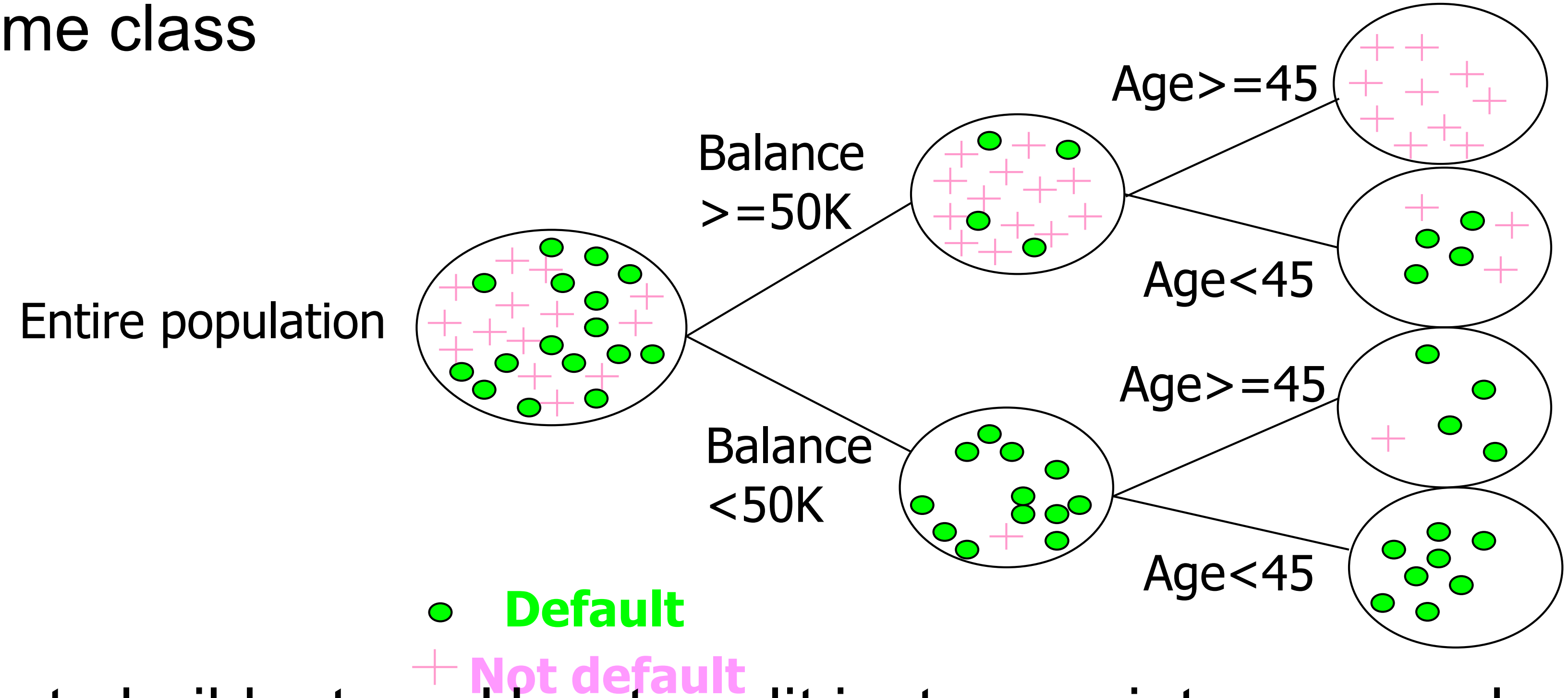
To determine the class of a new instance: e.g., Mark, age 40, unemployed, balance 88K.

- The instance is routed down the tree according to values of attributes.
- At each node a test is applied to one attribute.
- When a leaf is reached the instance is assigned to a class.
- Mark: Yes



Goal of Decision Tree Construction

- Partition the training instances into <u>purer</u> sub groups
  - pure: the instances in a sub-group mostly belong to the same class



- How to build a tree: How to split instances into purer sub-groups

Why do we want to identify pure sub groups?

- To classify a new instance, we can determine the leaf that the instance belongs to based on its attributes.
- If the leaf is very pure (e.g. all have defaulted) we can determine with greater confidence that the new instance belongs to this class (i.e., the “Default” class.)
- If the leaf is not very pure (e.g. a 50%/50% mixture of the two classes, Default and Not Default), our prediction for the new instance is more like a random guessing.

Decision Tree Construction

- A tree is constructed by recursively partitioning the examples.
- With each partition the examples are split into increasingly purer sub groups.
- The key in building a tree: How to split

Recursive Steps in Building a Tree

- STEP 1:
- Try using different attributes to split the training examples into different subsets.

- STEP 2:
- Rank the splits. Choose the best split.

- STEP 3:
- For each node obtained by splitting, repeat from STEP 1, until no more good splits are possible.

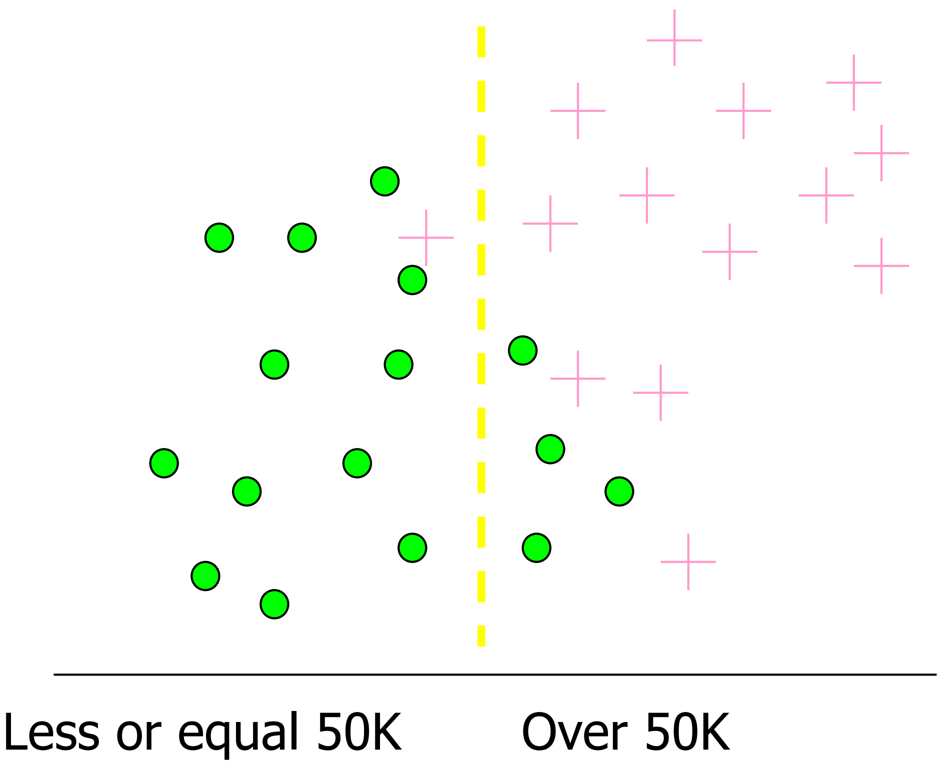
- Note: Usually it is not possible to create leaves that are completely pure - i.e. contain one class only - as that would result in a very bushy tree which is not sufficiently general. However, it is possible to create leaves that are purer – i.e. contain predominantly one class - and we can settle for that.

Purity Measures

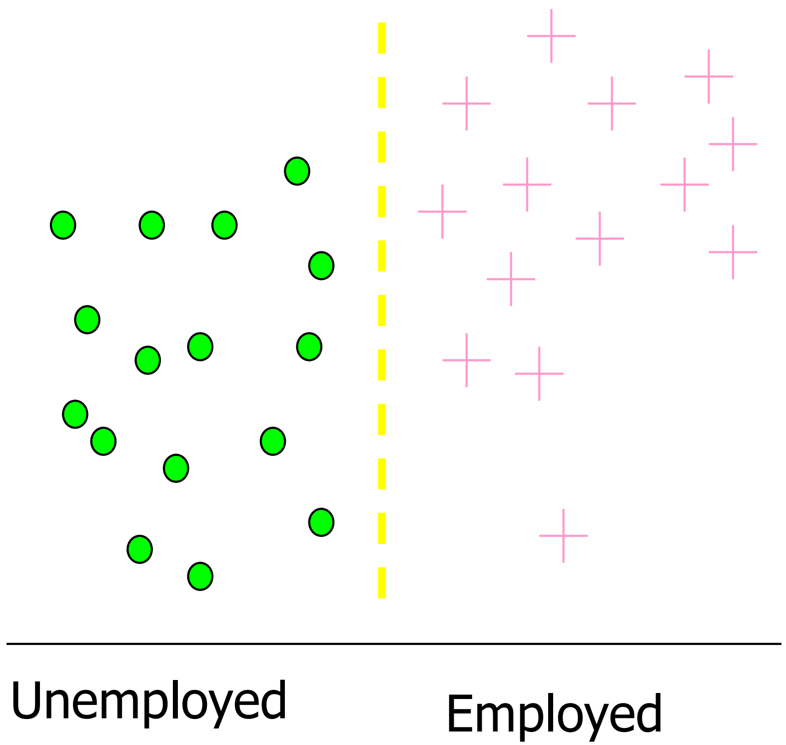
- Purity measures: Many available
  - Gini (population diversity)
  - Entropy (information gain)
  - Information Gain Ratio
  - Chi-square Test
- Most common one (from information theory) is: *Information Gain*
  - Informally: How informative is the attribute in distinguishing among instances (e.g., credit applicants) from different classes (Yes/No default)

Information Gain

Split over whether Balance exceeds 50K

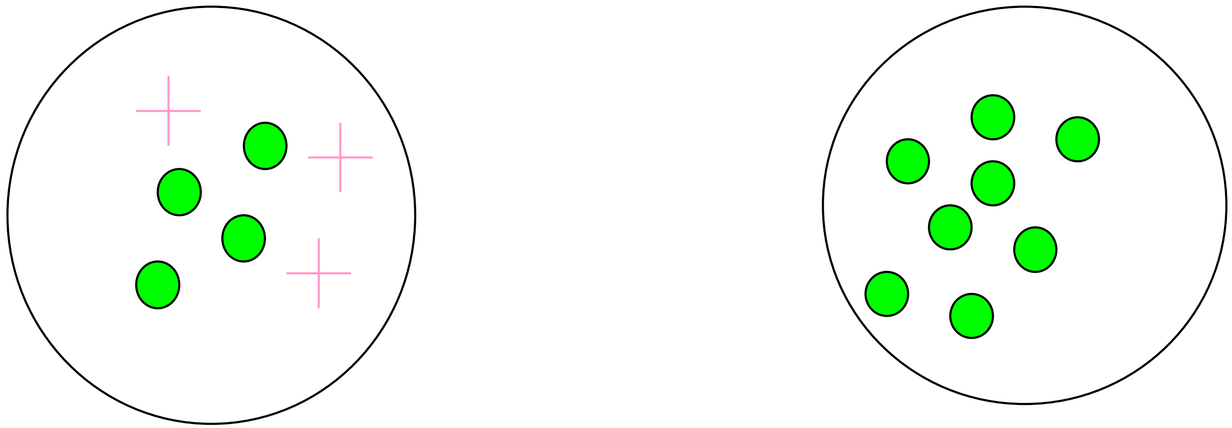


Split over whether applicant is employed



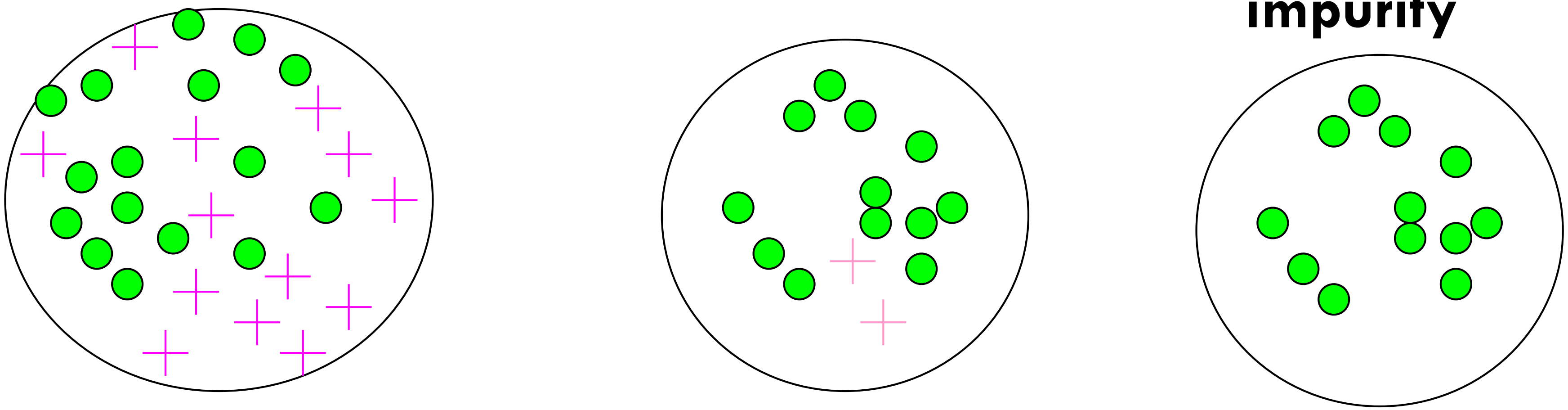
Information Gain

- **Impurity/Entropy:**
  - Measures the level of **impurity/chaos** in a group of examples
  - Information gain is defined as the decrease in impurity with the split generating more pure sub-groups



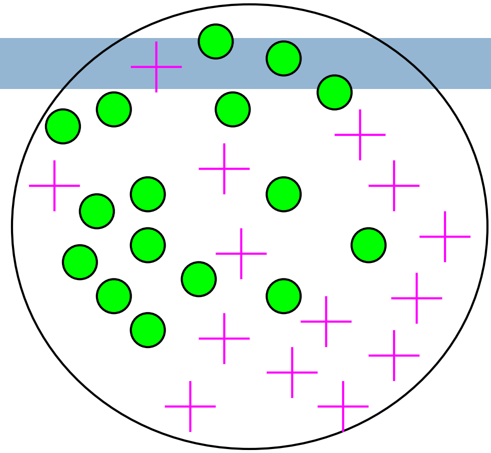
Impurity

Very impure group **Less impure** **Minimum impurity**



When examples can belong to one of two classes:
What is the worst case of impurity?

Calculating Impurity



- Impurity = $\sum_i - p_i \log_2 p_i$

  p_i is proportion of class *i*

- For example: our initial population is composed of 16 cases of class “Default” and 14 cases of class “Not default”

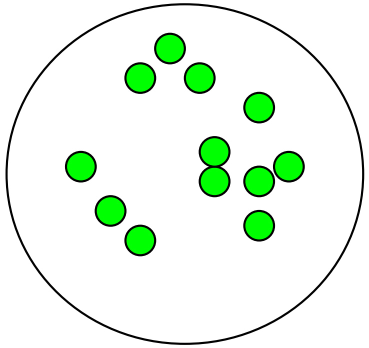
Impurity (entire population of examples)=

$$-\left(\frac{14}{30}\cdot\log_2\frac{14}{30}\right)-\left(\frac{16}{30}\cdot\log_2\frac{16}{30}\right)=0.997$$

2-class Cases:

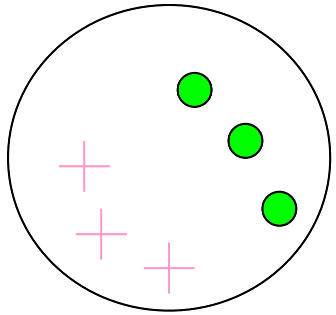
- What is the impurity of a group in which all examples belong to the same class?
  - Impurity= $- 1 \log_2 1 - 0 \log_2 0 = 0$ (lowest possible value)

Minimum impurity



- What is the impurity of a group with 50% in either class?
  - Impurity= $-0.5 \log_2 0.5 – 0.5 \log_2 0.5 =1$ (highest possible value)

Maximum impurity

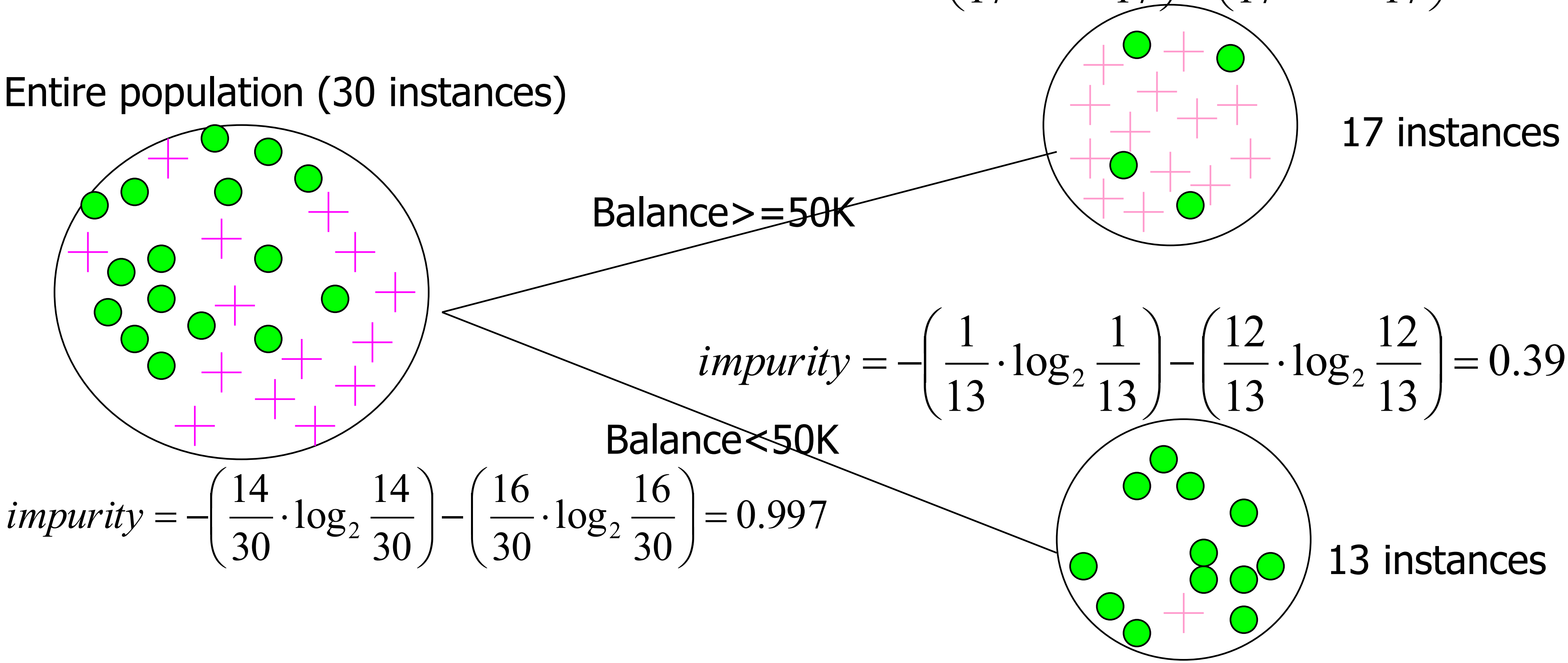


0 log(0) is defined as 0

Calculating Information Gain

Information Gain = Impurity (parent) – Impurity (children)

$$impurity = -\left(\frac{13}{17}\cdot\log_2\frac{13}{17}\right) - \left(\frac{4}{17}\cdot\log_2\frac{4}{17}\right) = 0.787$$



$$impurity = -\left(\frac{1}{13}\cdot\log_2\frac{1}{13}\right) - \left(\frac{12}{13}\cdot\log_2\frac{12}{13}\right) = 0.391$$

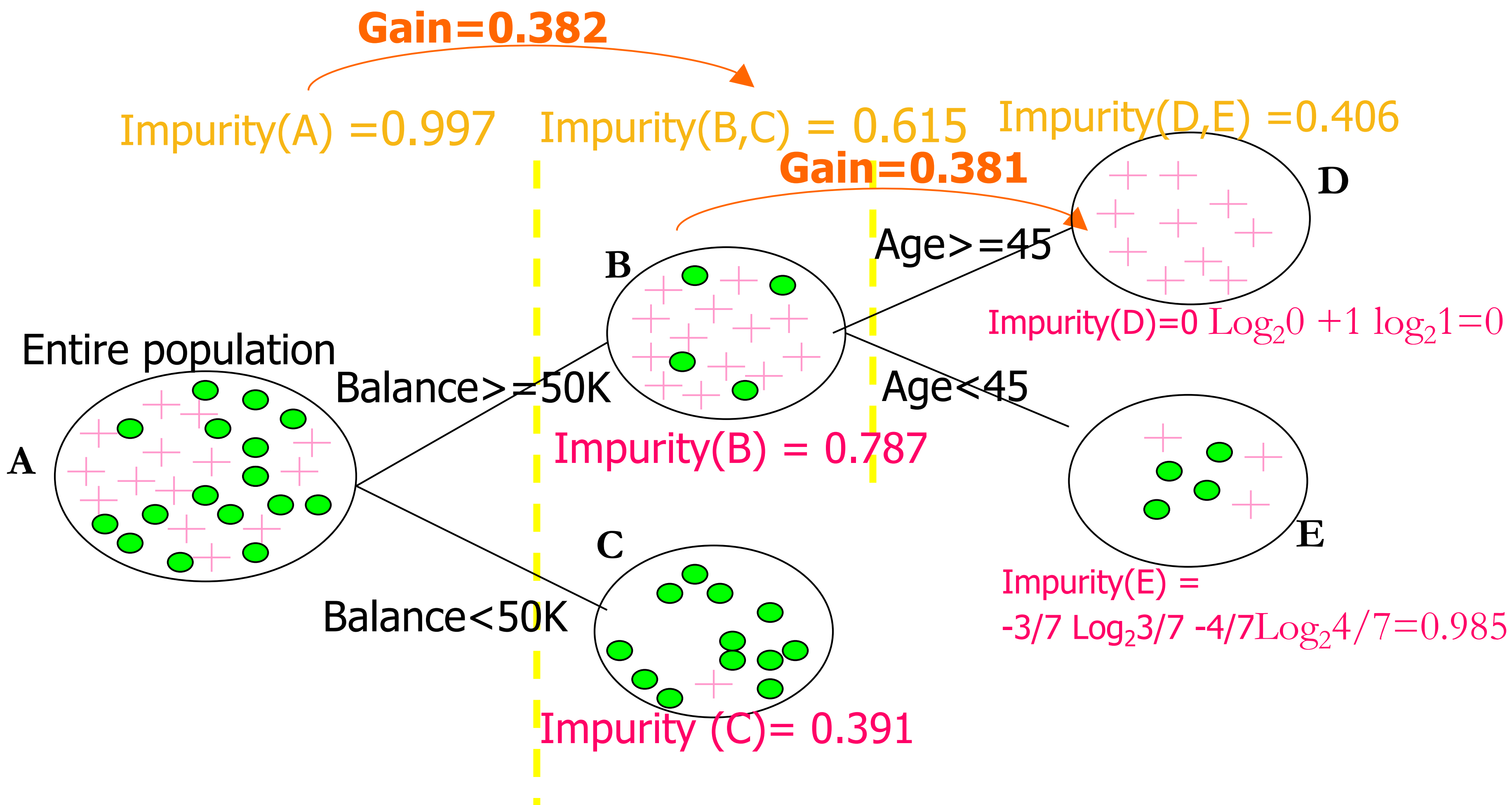
$$impurity = -\left(\frac{14}{30}\cdot\log_2\frac{14}{30}\right) - \left(\frac{16}{30}\cdot\log_2\frac{16}{30}\right) = 0.997$$

(Weighted) Average Impurity of Children = $\left(\frac{17}{30}\cdot 0.787\right) + \left(\frac{13}{30}\cdot 0.391\right) = 0.615$

Information Gain= 0.997 - 0.615 = 0.382

Information Gain

Information Gain = Impurity (parent) – Impurity (children)

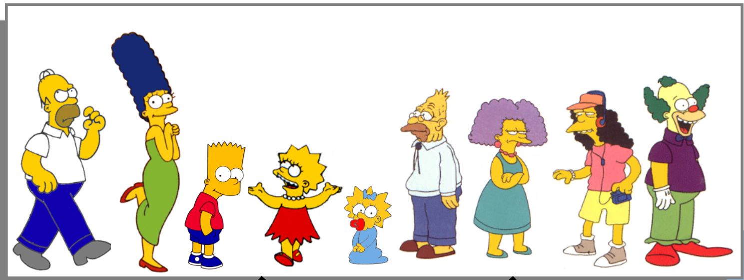


Which attribute to split over?

- At each node examine splits over each of the attributes
- Select the attribute for which the maximum information gain is obtained
  - For a continuous attribute, also need to consider different ways of splitting (>50 or <=50; >60 or <=60)
  - For a categorical attribute with lots of possible values, sometimes also need to consider how to group these values (branch 1 corresponds to {A,B,E} and branch 2 corresponds to {C,D,F,G})

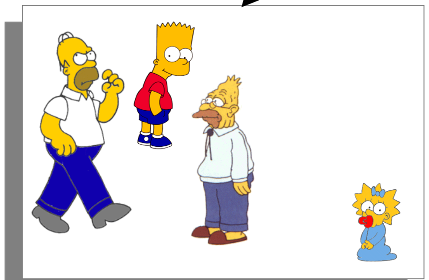
Person		Hair Length	Weight	Age	Class
	Homer	0”	250	36	M
	Marge	10”	150	34	F
	Bart	2”	90	10	M
	Lisa	6”	78	8	F
	Maggie	4”	20	1	F
	Abe	1”	170	70	M
	Selma	8”	160	41	F
	Otto	10”	180	38	M
	Krusty	6”	200	45	M

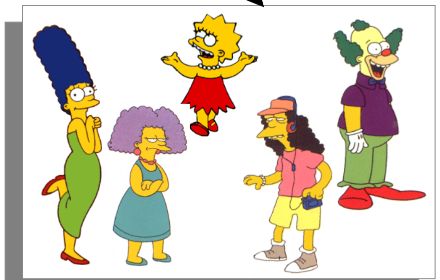
	Comic	8”	290	38	?



$Entropy(4F,5M) = -(4/9)\log_2(4/9) - (5/9)\log_2(5/9)$
$= 0.9911$

yes | Hair Length <= 5? | no





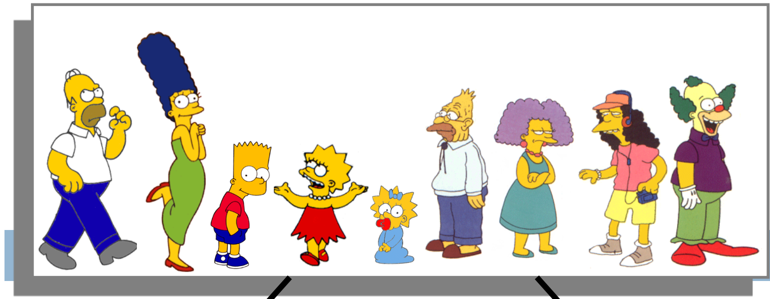
Let us try splitting on *Hair length*

$Entropy(1F,3M) = -(1/4)\log_2(1/4) - (3/4)\log_2(3/4)$
$= 0.8113$

$Entropy(3F,2M) = -(3/5)\log_2(3/5) - (2/5)\log_2(2/5)$
$= 0.9710$

***Gain*= Entropy of parent – Weighted average of entropies of the children**

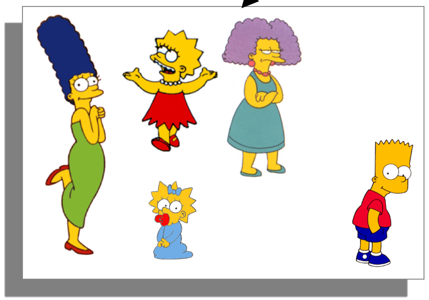
Gain(Hair Length <= 5) = **0.9911** – (4/9 * **0.8113** + 5/9 * **0.9710**) = **0.0911**



$$Entropy(4F,5M) = -(4/9)\log_2(4/9) - (5/9)\log_2(5/9) = 0.9911$$

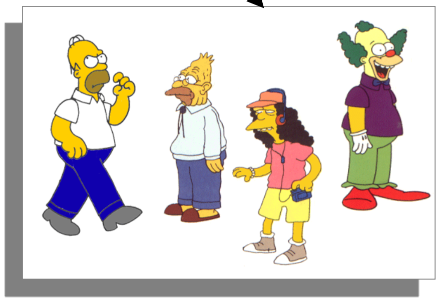
Weight <= 160?

yes



$$Entropy(4F,1M) = -(4/5)\log_2(4/5) - (1/5)\log_2(1/5) = 0.7219$$

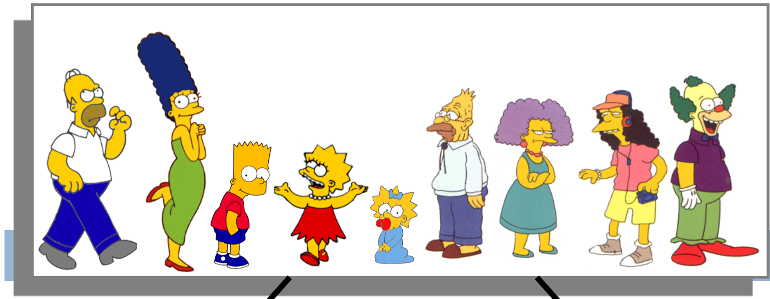
no



$$Entropy(0F,4M) = -(0/4)\log_2(0/4) - (4/4)\log_2(4/4) = 0$$

Let us try splitting on *Weight*

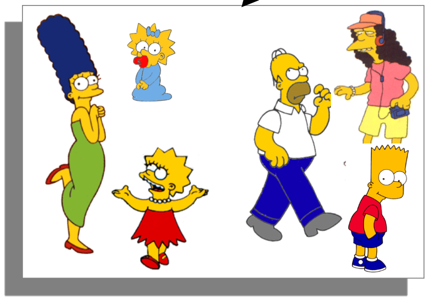
Gain(Weight <= 160) = **0.9911** – (5/9 * **0.7219** + 4/9 * **0**) = **0.5900**



$Entropy(4\textcolor{red}{F},5\textcolor{blue}{M}) = -(4/9)\log_2(4/9) - (5/9)\log_2(5/9)$
$= \textcolor{magenta}{0.9911}$

yes　　　no

age <= 40?



Let us try splitting on *Age*

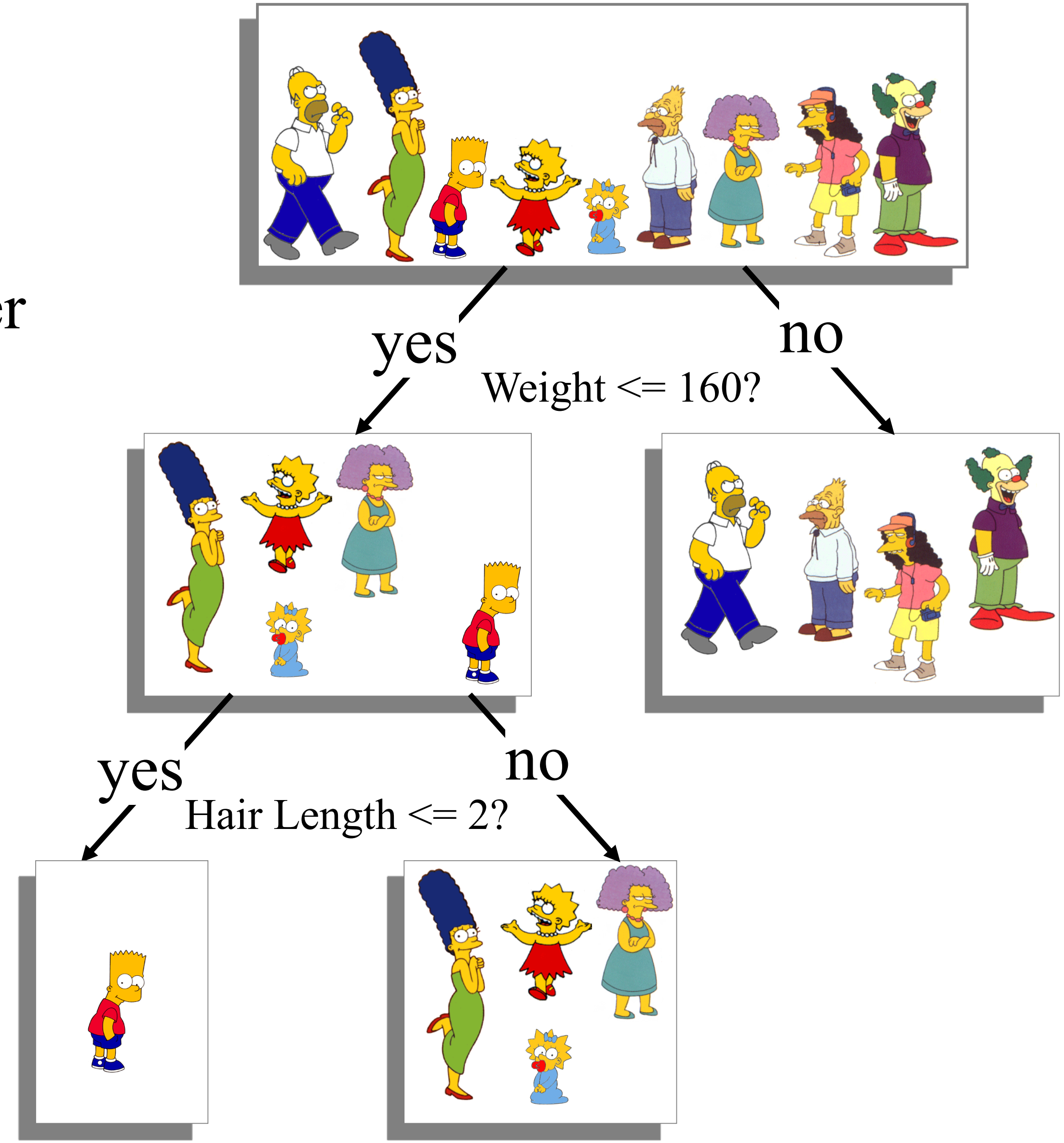
$Entropy(3\textcolor{red}{F},3\textcolor{blue}{M}) = -(3/6)\log_2(3/6) - (3/6)\log_2(3/6)$
$= \textcolor{green}{1}$

$Entropy(1\textcolor{red}{F},2\textcolor{blue}{M}) = -(1/3)\log_2(1/3) - (2/3)\log_2(2/3)$
$= \textcolor{purple}{0.9183}$

Gain(Age <= 40) = 0.9911 – (6/9 * 1 + 3/9 * 0.9183) = **0.0183**

Of the 3 features we had, *Weight* was the best. But while people who weigh over 160 are perfectly classified (as males), the under 160 people are not perfectly classified… So we simply continue splitting!

This time we find that we can split on *Hair length,* and then we are done!



Aha, I got a tree!

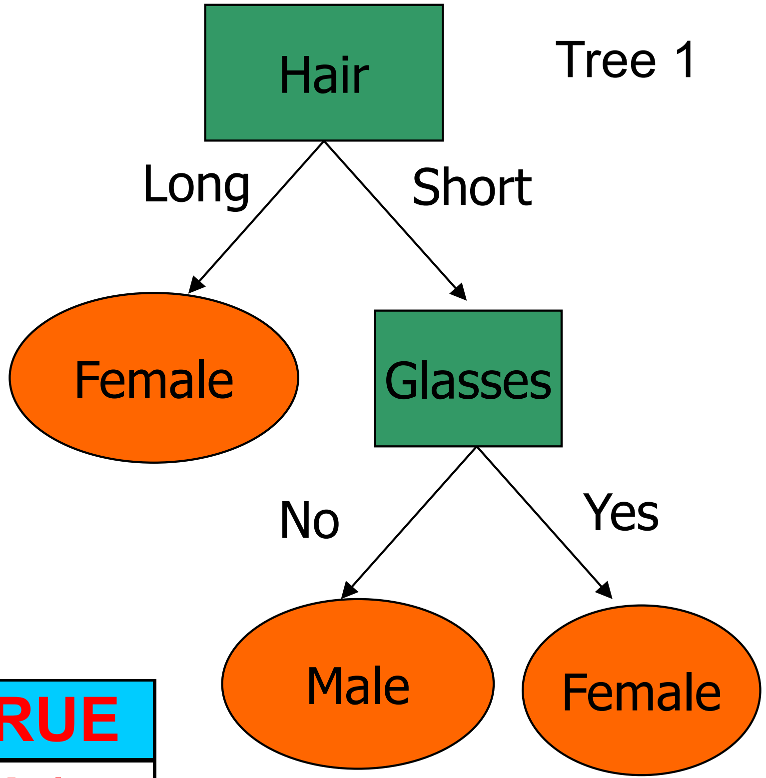
Note: the splitting decision is not only to choose attribute, but to choose the value to split for a continuous attribute (e.g. Hair <=5 or Hair <=2)

Building a Tree - Stopping Criteria

- You can stop building the tree when:
  - The impurity of all nodes is zero: Problem is that this tends to lead to bushy, highly-branching trees, often with one example at each node.
  - No split achieves a significant gain in purity (information gain not high enough)
  - Node size is too small: That is, there are less than a certain number of examples, or proportion of the training set, at each node.

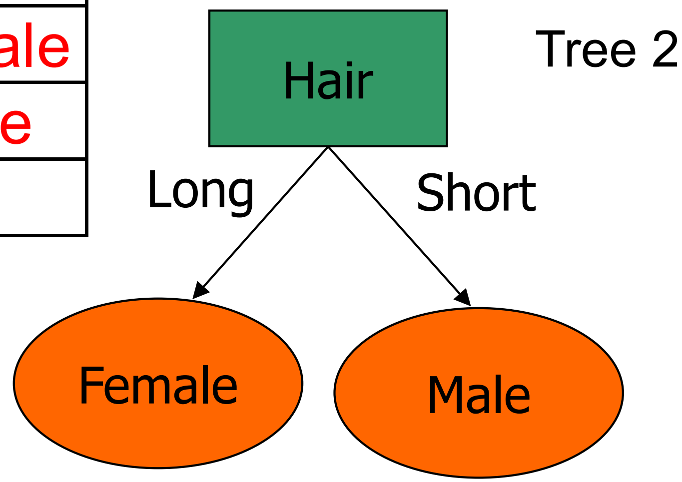
Training

Name	Hair	Glasses	Class
Mary	Long	No	Female
Mike	Short	No	Male
Bill	Short	No	Male
Jane	Long	No	Female
Ann	Short	Yes	Female



Testing

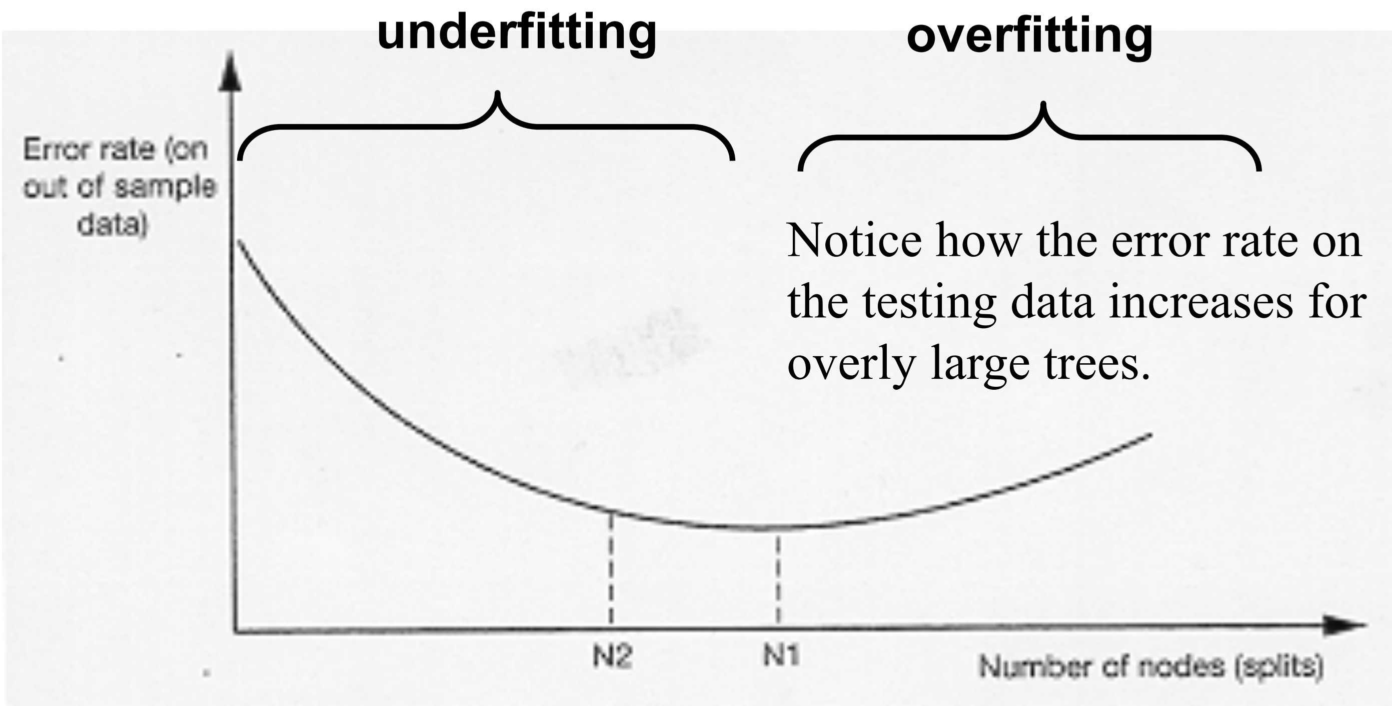
Hair	Glasses	Tree 1	Tree 2	TRUE
Short	Yes	Female	Male	Male
Short	No	Male	Male	Female
Long	No	Female	Female	Female
Short	Yes	Female	Male	Male
	Error:	75%	25%	



There are many possible splitting rules that perfectly classify the data, but will not generalize to future datasets.

Overfitting & Underfitting

- **Overfitting**: the model performs poorly on new examples (e.g. testing examples) as it is too highly trained to the specific training examples (pick up patterns and noises).
- **Underfitting**: the model performs poorly on new examples as it is too simplistic to distinguish between them (i.e. has not picked up the important patterns from the training examples)



Pruning

A decision trees is typically more accurate on its *training* data than on its *test* data. Removing branches from a tree can often improve its accuracy on a test set - so-called '**reduced error pruning**'. The intention of this pruning is to cut off branches from the tree when this improves performance on test data - this reduces overfitting and makes the tree more general.
Small is beautiful.

Decision Tree Classification in a Nutshell

- Decision tree
  - A tree structure
  - Internal node denotes a test on an attribute
  - Branch represents an outcome of the test
  - Leaf nodes represent class labels or class distribution
- Decision tree generation consists of two phases
  - Tree construction
    - At start, all the training examples are at the root
    - Partition examples recursively based on selected attributes
  - Tree pruning
    - Identify and remove branches that reflect noise or outliers
    - To avoid overfitting
- Use of decision tree: Classifying an unknown sample
  - Test the attribute values of the sample against the decision tree

Strengths & Weaknesses

- In practice: One of the most popular method. Why?
  - Very comprehensible – the tree structure specifies the entire decision structure
    - Easy for decision makers to understand model's rational
    - Map nicely to a set of business rules
  - Relatively easy to implement
- Very fast to run (to classify examples) with large data sets
- Good at handling missing values: just treat "missing" as a value – can become a good predictor
- Weakness
  - Bad at handling continuous data, good at categorical input and output.
    - Continuous output: high error rate
    - Continuous input: ranges may introduce bias

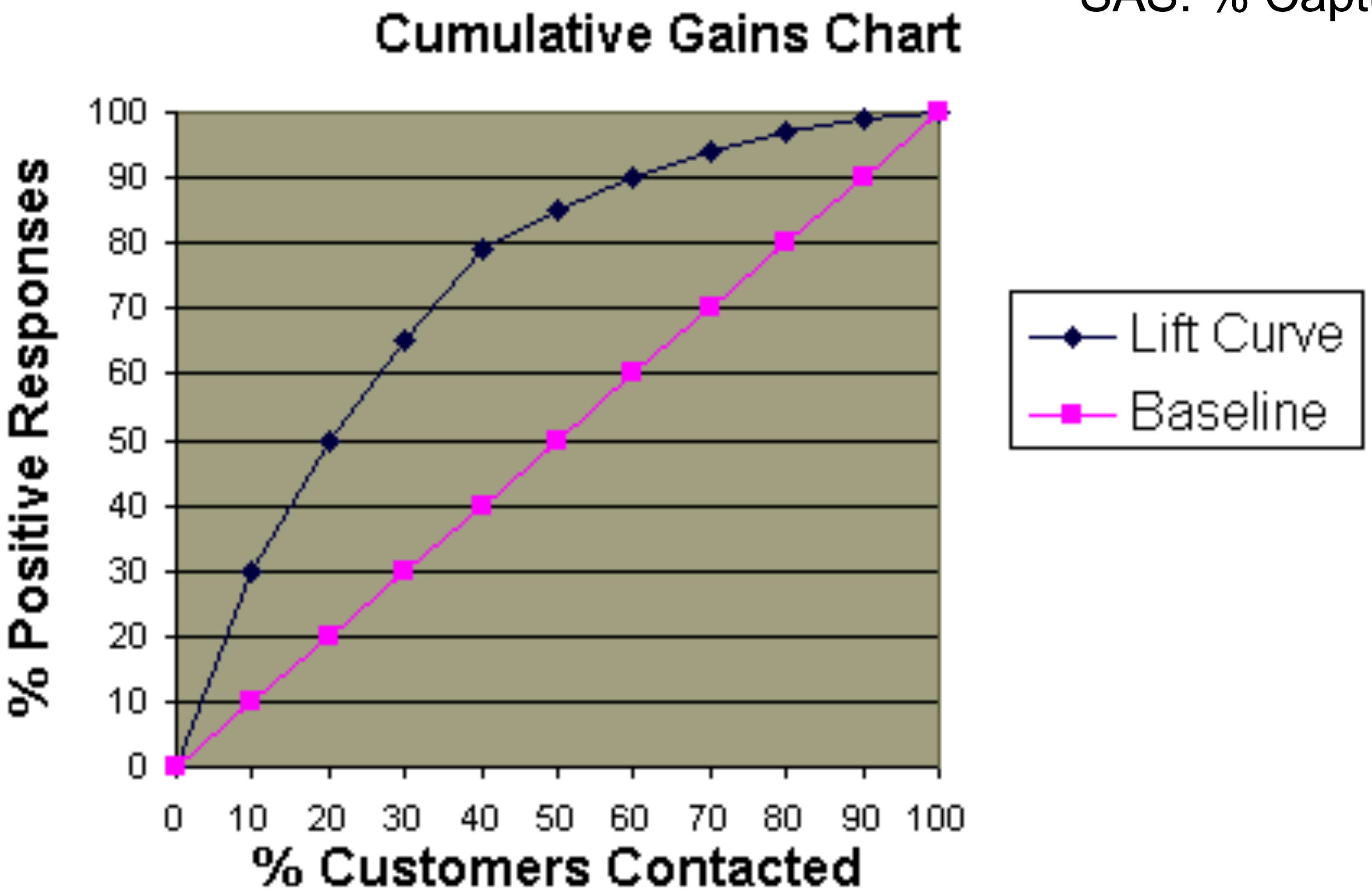
Different Decision Tree Algorithms

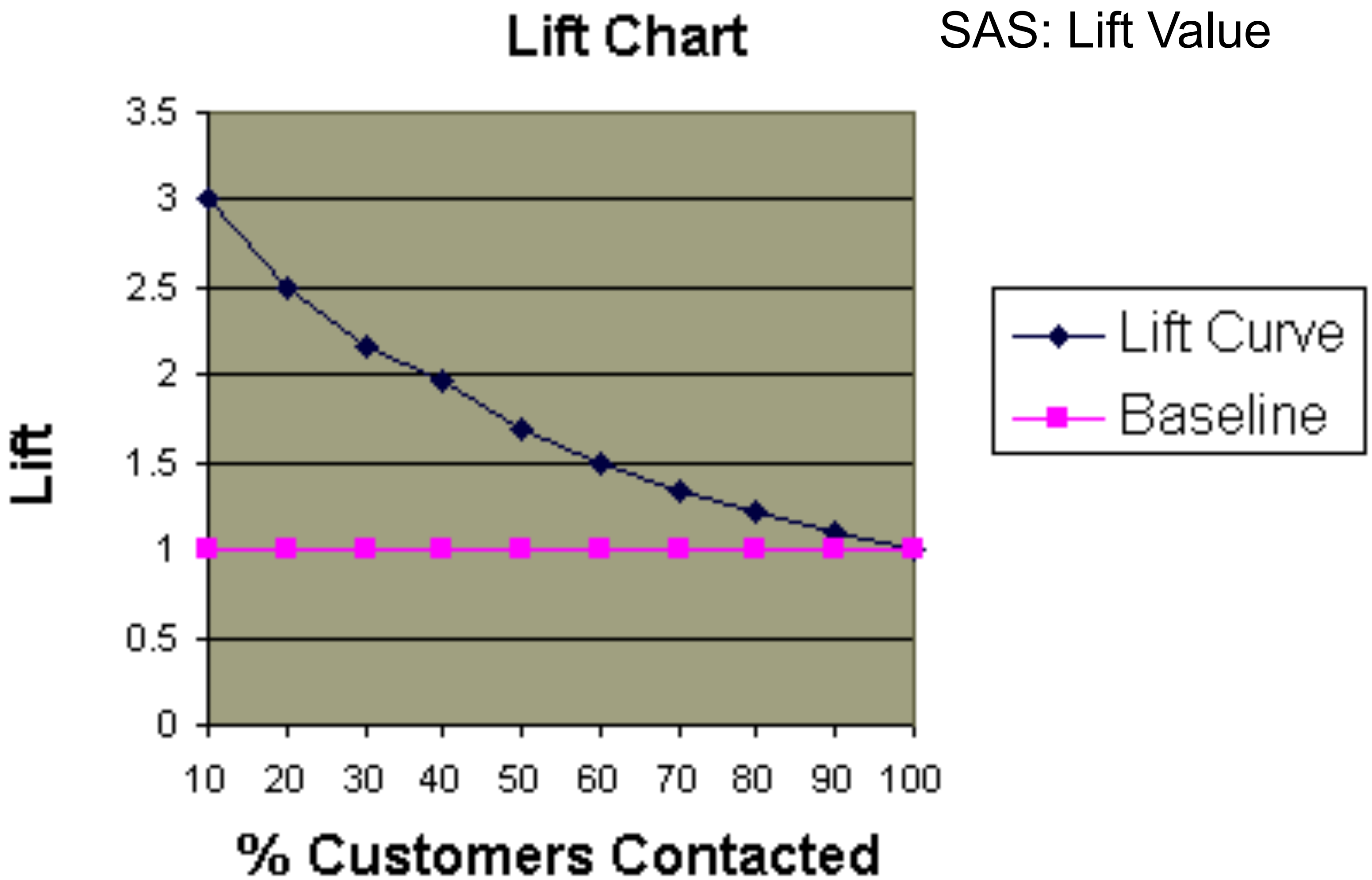
- ID3, ID4, ID5, C4.0, C4.5, C5.0, ACLS, and ASSISTANT:
  - Use information gain as splitting criterion
- CART (Classification And Regression Trees):
  - Uses Gini diversity index as measure of impurity when deciding splitting.
- CHAID:
  - A statistical approach that uses the Chi-squared test when deciding on the best split.
- Hunt's Concept Learning System (CLS), and MINIMAX:
  - Minimizes the cost of classifying examples correctly or incorrectly.

10-fold Cross Validation

- Break data into 10 sets of size n/10.
- Train on 9 datasets and test on 1.
- Repeat 10 times and take a mean accuracy.

SAS: % Captured Response





SAS: Lift Value

In SAS, %Response =
Percentage of Responses in the top n% ranked individuals. It should be relatively high in the top deciles. And a decreasing plotted curve indicates a good model. The lift chart captures the same information on a different scale.

Case Discussion

- Fleet
  1. How many input variables are used to build the tree? How many show up in the tree built? Why?
  2. How can the tree built be used for segmentation?
  3. How can the new campaign results help enhance the tree?

Exercise – Decision Tree

Customer ID	Student	Credit Rating	Class: Buy PDA
1	No	Fair	No
2	No	Excellent	No
3	No	Fair	Yes
4	No	Fair	Yes
5	Yes	Fair	Yes
6	Yes	Excellent	No
7	Yes	Excellent	Yes
8	No	Excellent	No

Which attribute to split on first?

$\log_2(2/3) = -0.585$, $\log_2(1/3) = -1.585$, $\log_2(1/2) = -1$, $\log_2(3/5) = -0.737$,
$\text{Log}_2(2/5) = -1.322$, $\log_2(1/4) = -2$, $\log_2(3/4) = -0.415$