EMORY
UNIVERSITY

GOIZUETA
BUSINESS
SCHOOL

Introduction to Classification

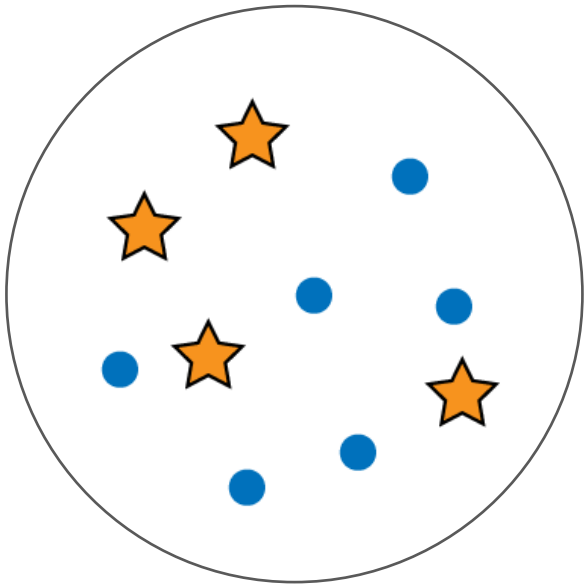
Introduction to Business Analytics

Vilma Todri
Assistant Professor
Goizueta Business School
Emory University
vtodri@emory.edu

Probability 101

★ = Customer made a purchase

● = Customer did not make a purchase



$P(★) =$

$P(●) =$

Data Set

Target Variable

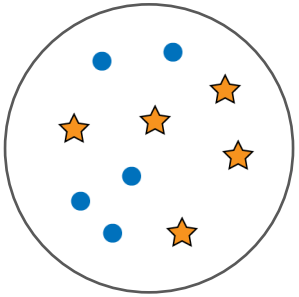
Attributes / Features

Conversion	Sex	Clicked on Ad
1	F	1
0	F	0
0	M	0
1	M	1
1	F	0
1	M	1
0	F	0
1	M	1
0	F	0
0	F	0

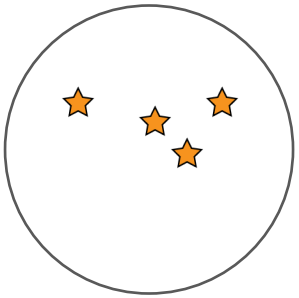
★ Conversion = 1

● Conversion = 0

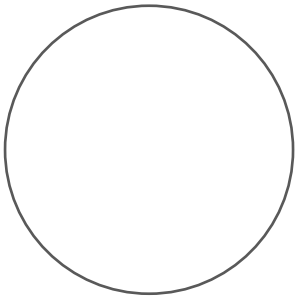
Which attribute is more informative?



- Whether the customer has clicked on an ad in the past (0/1)

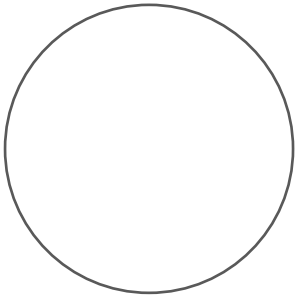


Value of Attribute = 1

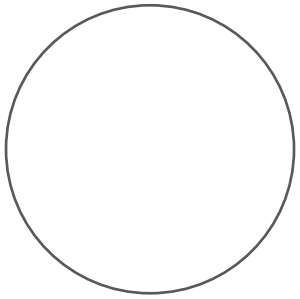


Value of Attribute = 0

- Sex of customer (F/M)



Value of Attribute = F



Value of Attribute = M

Data Set

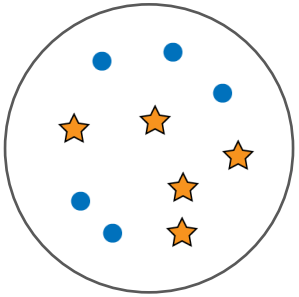
Target Variable

Attributes / Features

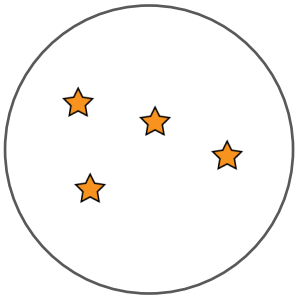
Conversion	Sex	Clicked on Ad
1	F	1
0	F	0
0	M	0
1	M	1
1	F	0
1	M	1
0	F	0
1	M	1
0	F	0
0	F	0

★ Conversion = 1
● Conversion = 0

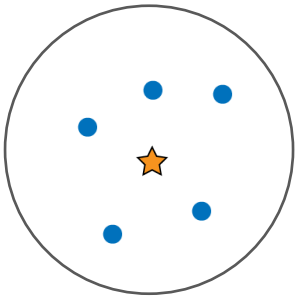
Which attribute is more informative?



- Whether the customer has clicked on an ad in the past (0/1)

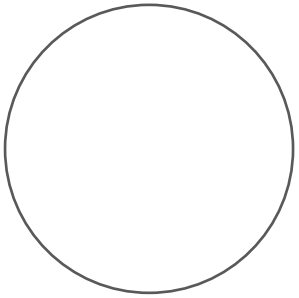


Value of Attribute = 1

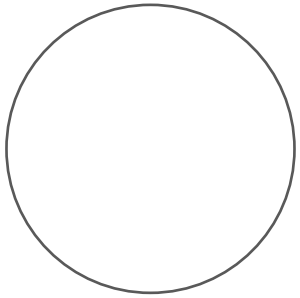


Value of Attribute = 0

- Sex of customer (F/M)



Value of Attribute = F

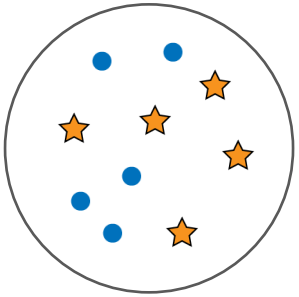


Value of Attribute = M

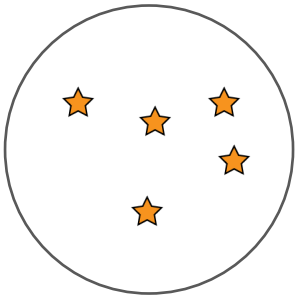
Data Set

Conversion	Sex	Clicked on Ad
1	F	1
0	F	0
0	M	0
1	M	1
1	F	0
1	M	1
0	F	0
1	M	1
0	F	0
0	F	0

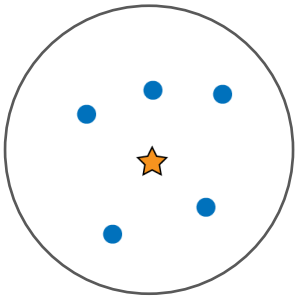
Which attribute is more informative?



- Whether the customer has clicked on an ad in the past (0/1)

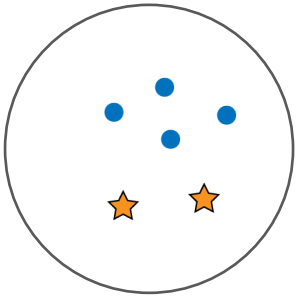


Value of Attribute = 1

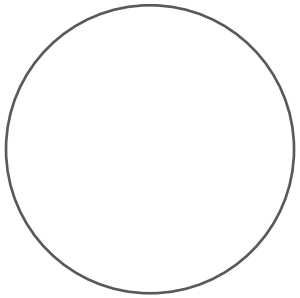


Value of Attribute = 0

- Sex of customer (F/M)



Value of Attribute = F

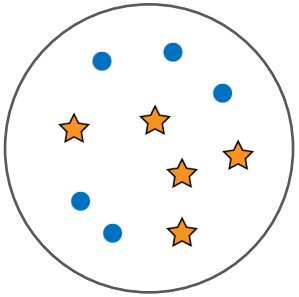


Value of Attribute = M

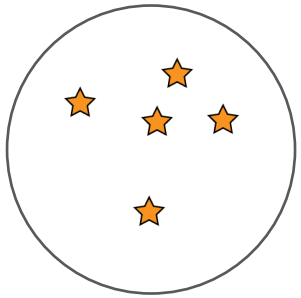
Data Set

Conversion	Sex	Clicked on Ad
1	F	1
0	F	0
0	M	0
1	M	1
1	F	0
1	M	1
0	F	0
1	M	1
0	F	0
0	F	0

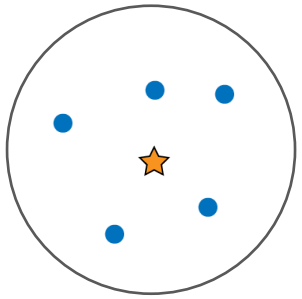
Which attribute is more “informative”?



- Whether the customer has clicked on an ad in the past (0/1)

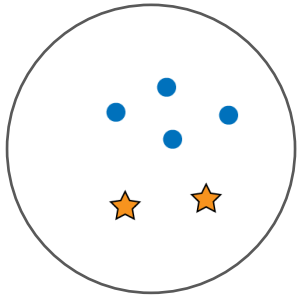


Value of Attribute = 1

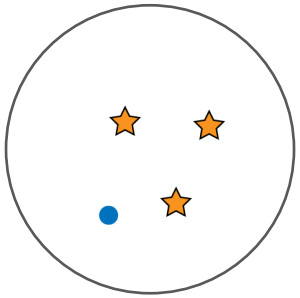


Value of Attribute = 0

- Sex of customer (F/M)



Value of Attribute = F



Value of Attribute = M

Supervised Segmentation

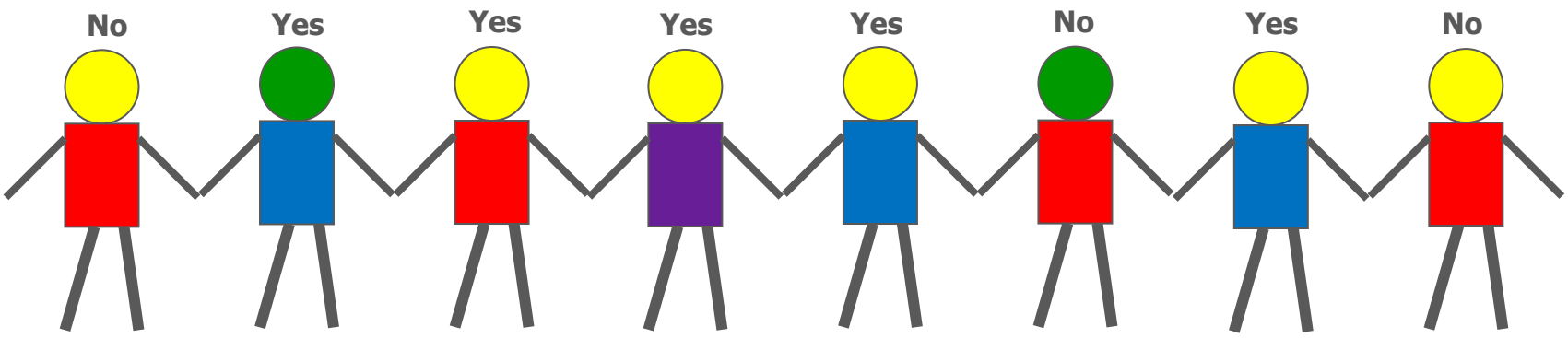
- How can we **segment** the population into groups that differ from each other *with respect to some* ***quantity of interest***?

- Informative attributes
  - Find **knowable** attributes that **correlate** with the **target of interest**
    - Increase accuracy
    - Alleviate computational problems
    - e.g. *tree induction*

- How can we judge whether a variable contains important information about the target variable?
  - How much?

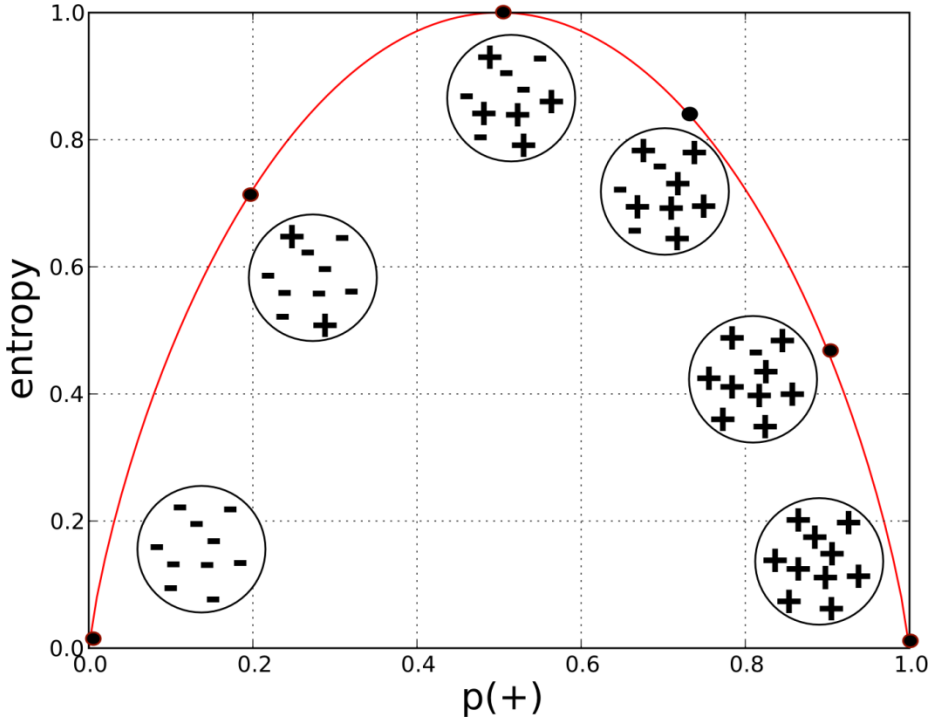
Selecting Informative Attributes

Objective: Based on customer attributes, partition the customers into **subgroups** that are **less impure** – with respect to the class (i.e., such that in each group as many instances as possible belong to the same class)



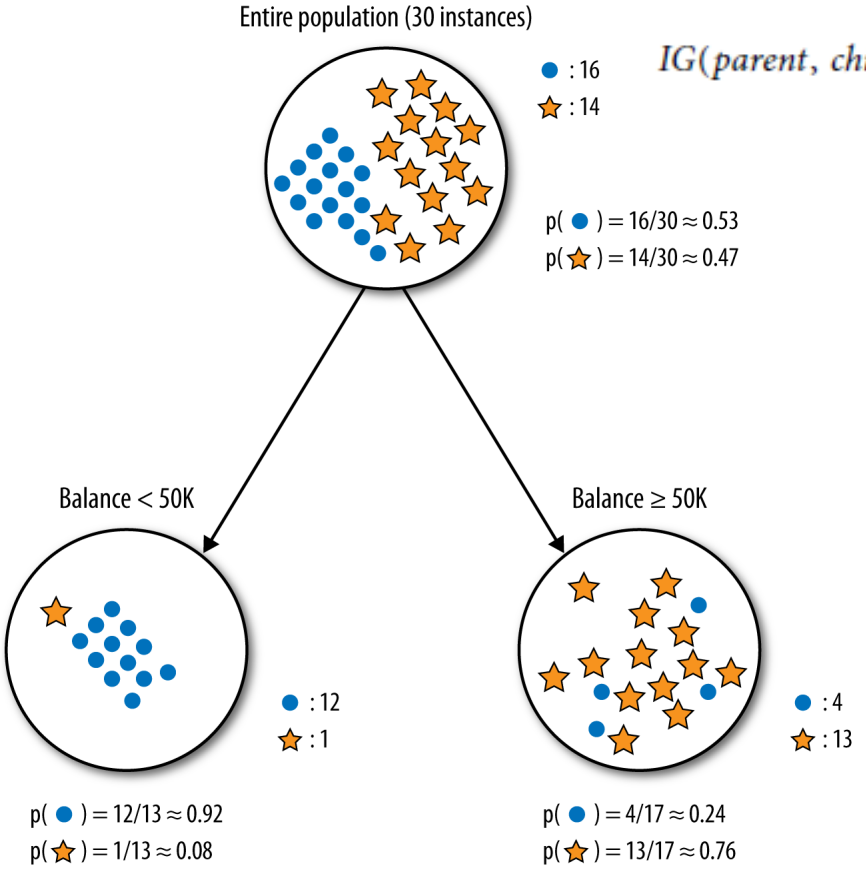
Selecting Informative Attributes

- One of the most common splitting criterion is called **information gain** (IG)
  - It is based on a **purity measure** called **entropy**
    - $entropy = -\mathrm{p}_1 \log_2(p_1) - p_2 \log_2(p_2) - \ldots = -[\mathrm{p}_1 \log_2(p_1) + p_2 \log_2(p_2) + \ldots]$
    - Measures the general disorder of a set



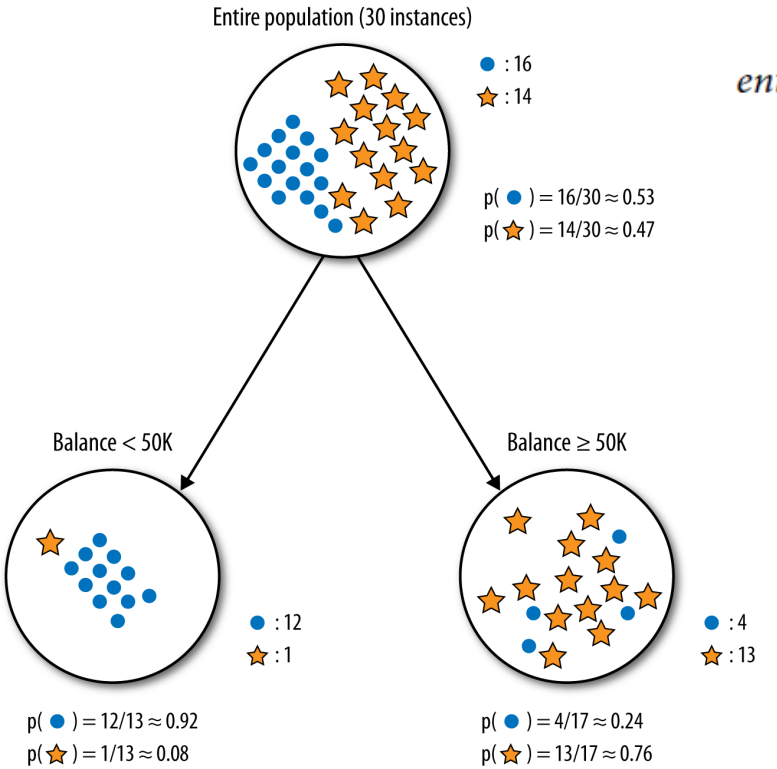
Information Gain

- Information gain measures the ***change* in entropy** due to any amount of new information being added



$$IG(parent, children) = entropy(parent) - [p(c_1) \times entropy(c_1) + p(c_2) \times entropy(c_2) + \cdots]$$

Information Gain



$$entropy(parent) = -[p(\bullet) \times \log_2 p(\bullet) + p(\star) \times \log_2 p(\star)]$$
$$\approx -[0.53 \times -0.9 + 0.47 \times -1.1]$$
$$\approx 0.99 \quad \text{(very impure)}$$

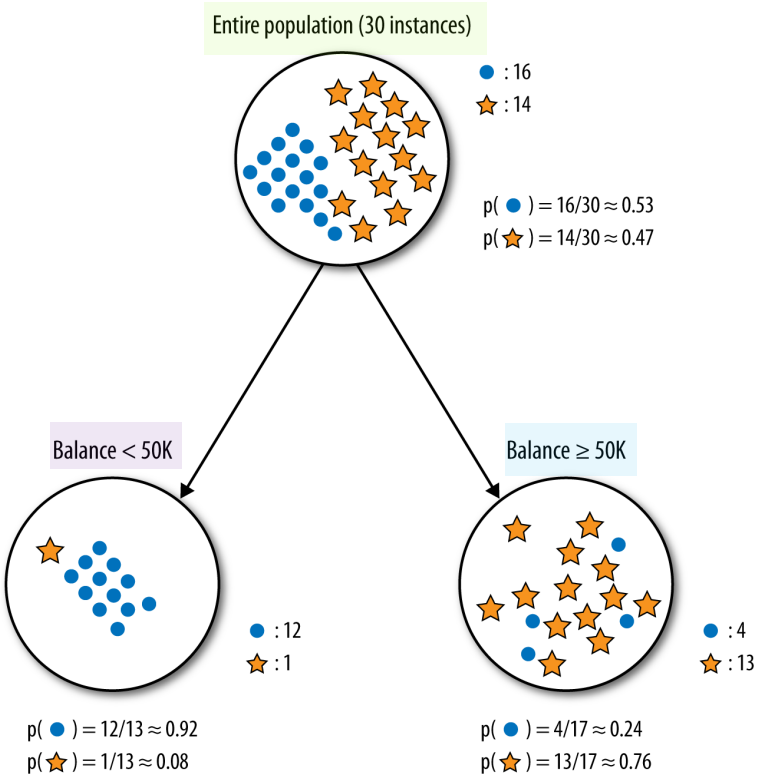
The entropy of the *left* child is:

$$entropy(Balance < 50K) = -[p(\bullet) \times \log_2 p(\bullet) + p(\star) \times \log_2 p(\star)]$$
$$\approx -[0.92 \times (-0.12) + 0.08 \times (-3.7)]$$
$$\approx 0.39$$

The entropy of the *right* child is:

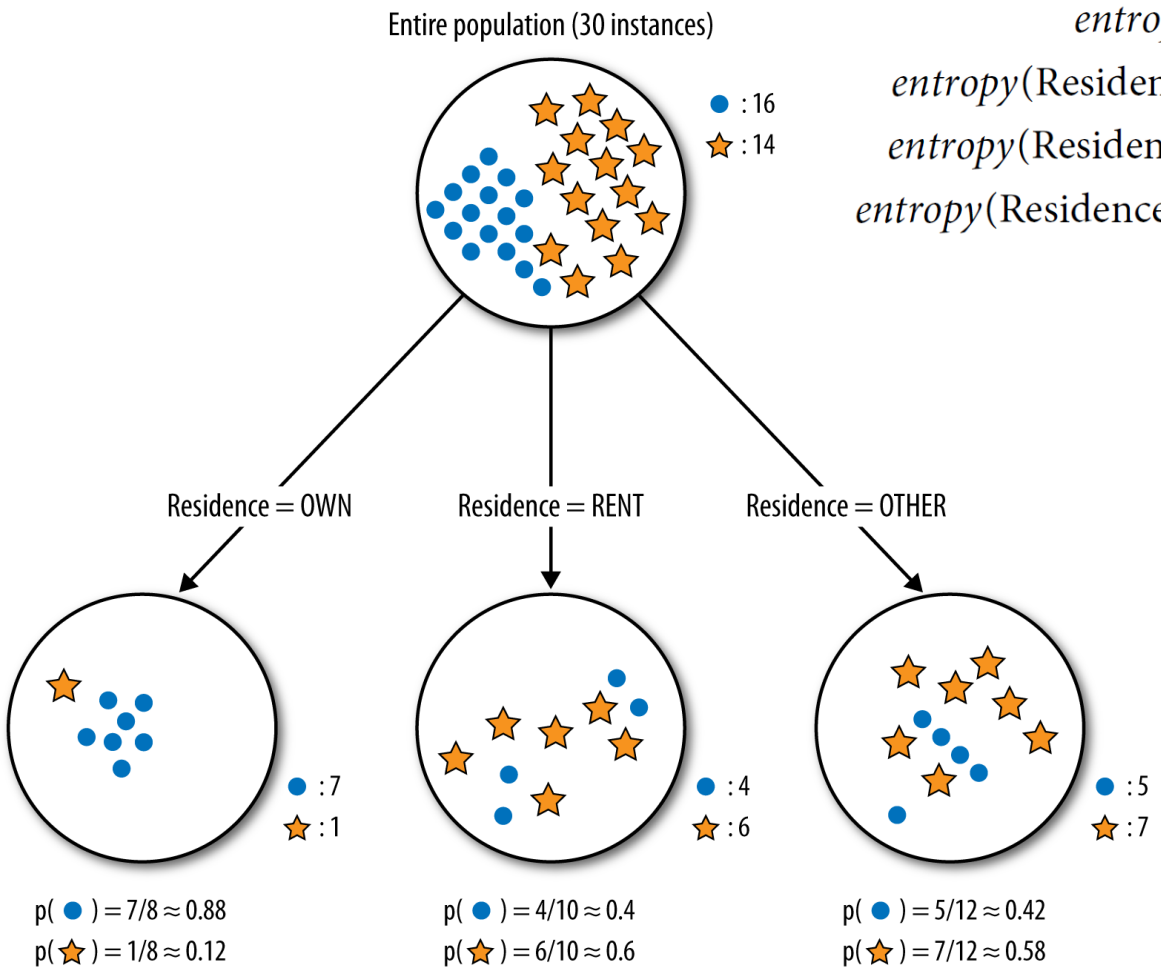
$$entropy(Balance \geq 50K) = -[p(\bullet) \times \log_2 p(\bullet) + p(\star) \times \log_2 p(\star)]$$
$$\approx -[0.24 \times (-2.1) + 0.76 \times (-0.39)]$$
$$\approx 0.79$$

Information Gain



$$
\begin{aligned}
IG &= entropy(parent) - [p(\text{Balance} < 50\text{K}) \times entropy(\text{Balance} < 50\text{K}) \\
&\quad + p(\text{Balance} \geq 50\text{K}) \times entropy(\text{Balance} \geq 50\text{K})] \\
&\approx 0.99 - [0.43 \times 0.39 + 0.57 \times 0.79] \\
&\approx 0.37
\end{aligned}
$$

Information Gain



$$entropy(parent) \approx 0.99$$
$$entropy(\text{Residence=OWN}) \approx 0.54$$
$$entropy(\text{Residence=RENT}) \approx 0.97$$
$$entropy(\text{Residence=OTHER}) \approx 0.98$$
$$IG \approx 0.13$$

Attribute Selection

Reasons for selecting only a subset of attributes:

- Better **insights** and business understanding
- Better **explanations** and more tractable models
- Reduced **cost**
- **Faster** predictions
- **Better** predictions!
  - Over-fitting (*to be continued*..)

(and also determining the most informative attributes..)

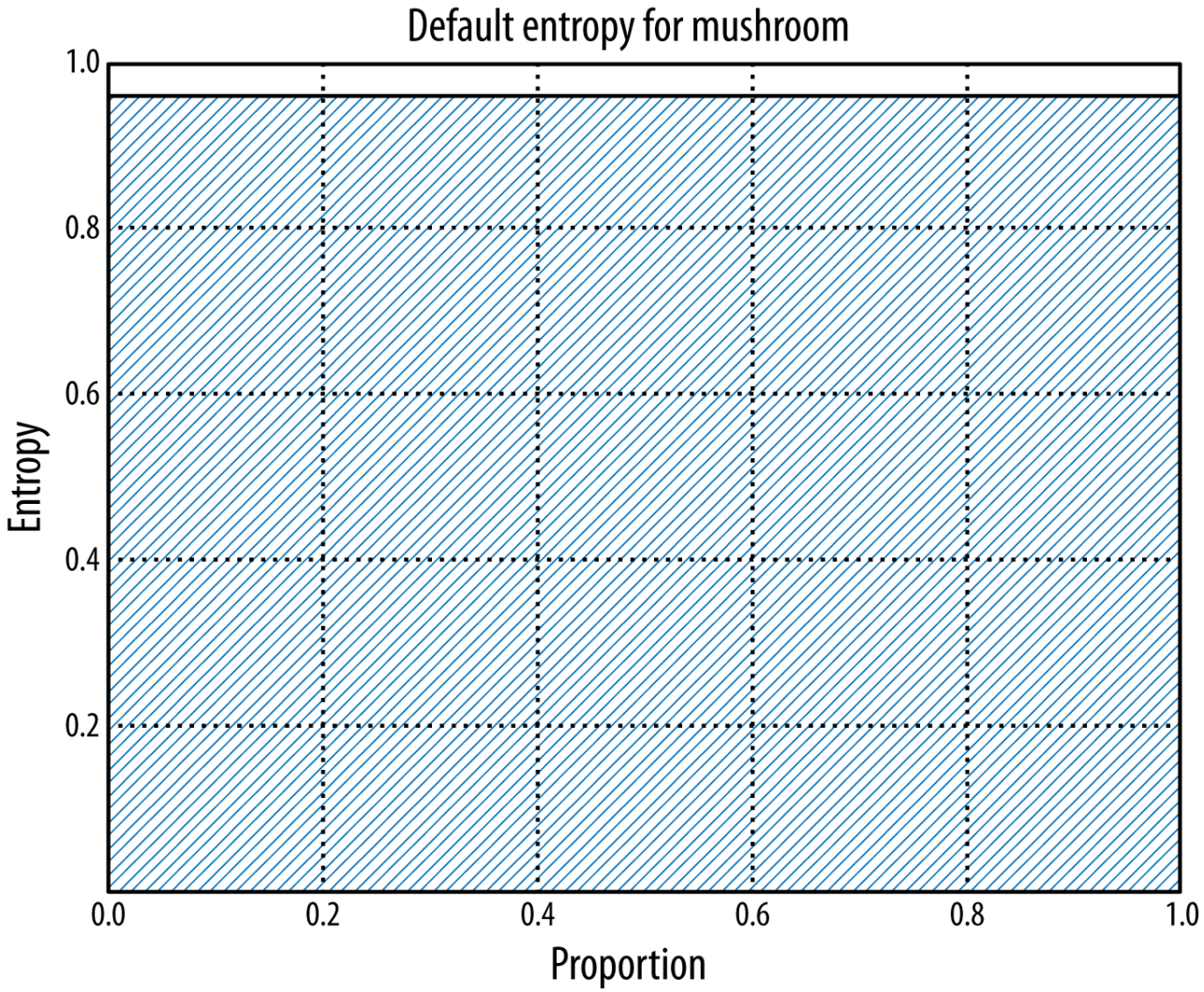
Example: Attribution Selection with Information Gain

- This dataset includes descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family.
- Each species is identified as definitely *edible*, definitely *poisonous*, or of *unknown edibility and not recommended*.
  - This latter class was combined with the poisonous one.
- The Guide clearly states that there is **no simple rule for determining the edibility of a mushroom**; no rule like "leaflets three, let it be" for Poisonous Oak and Ivy.

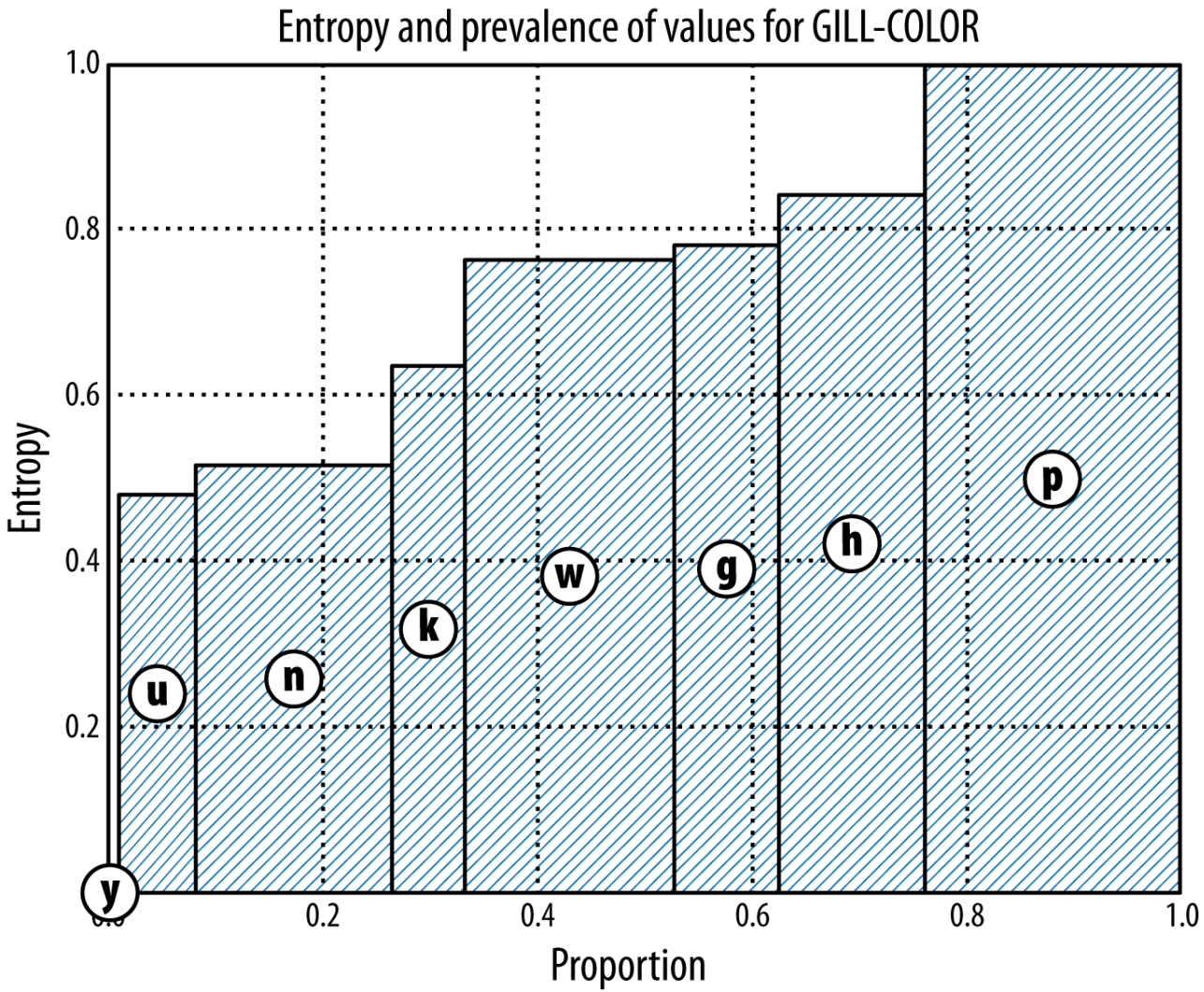
Example: Attribution Selection with Information Gain

Attribute name	Possible values
CAP-SHAPE	bell, conical, convex, flat, knobbed, sunken
CAP-SURFACE	fibrous, grooves, scaly, smooth
CAP-COLOR	brown, buff, cinnamon, gray, green, pink, purple, red, white, yellow
BRUISES?	yes, no
ODOR	almond, anise, creosote, fishy, foul, musty, none, pungent, spicy
GILL-ATTACHMENT	attached, descending, free, notched
GILL-SPACING	close, crowded, distant
GILL-SIZE	broad, narrow
GILL-COLOR	black, brown, buff, chocolate, gray, green, orange, pink, purple, red, white, yellow
STALK-SHAPE	enlarging, tapering
STALK-ROOT	bulbous, club, cup, equal, rhizomorphs, rooted, missing
STALK-SURFACE-ABOVE-RING	fibrous, scaly, silky, smooth
STALK-SURFACE-BELOW-RING	fibrous, scaly, silky, smooth

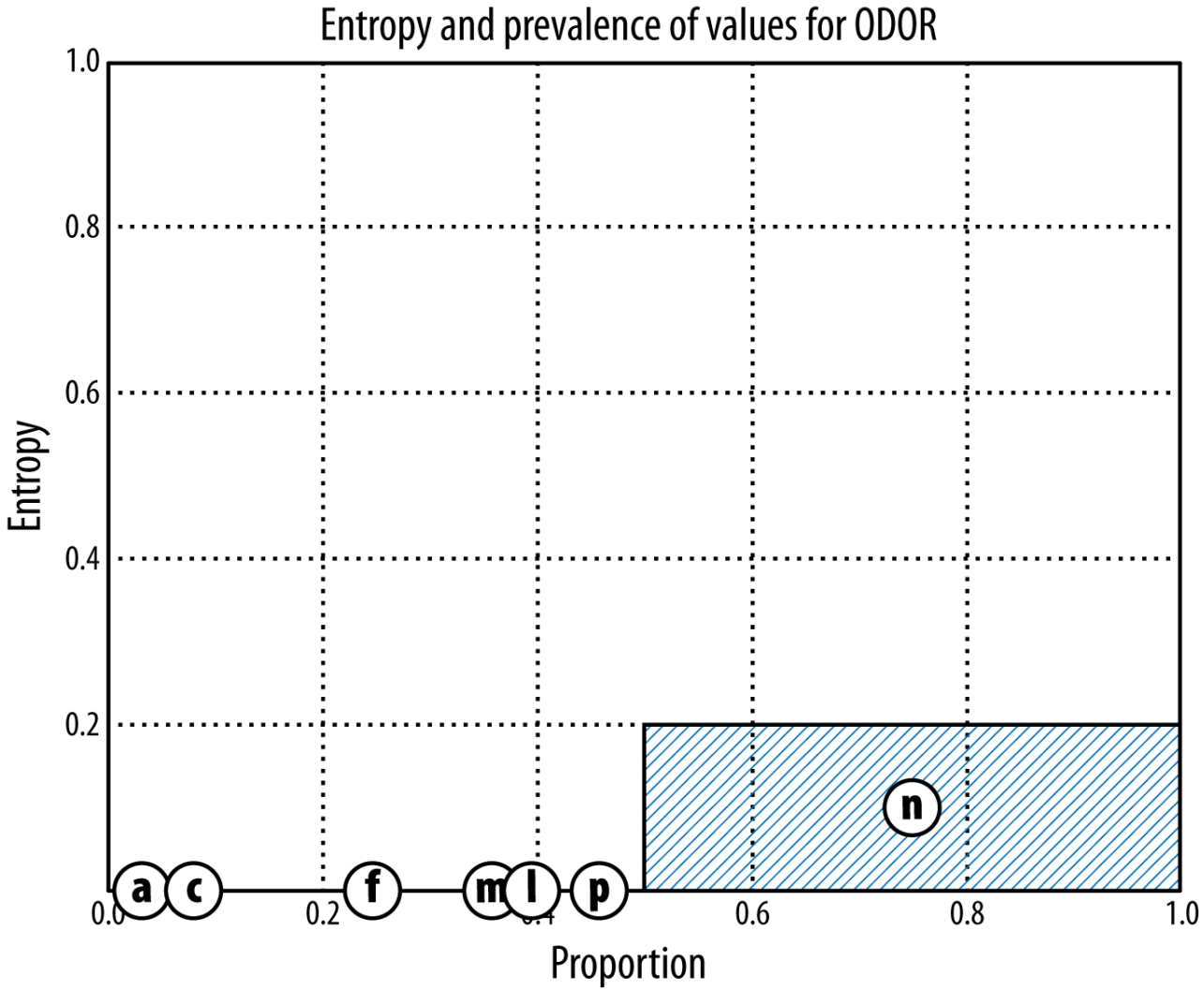
Example: Attribution Selection with Information Gain



Example: Attribution Selection with Information Gain



Example: Attribution Selection with Information Gain

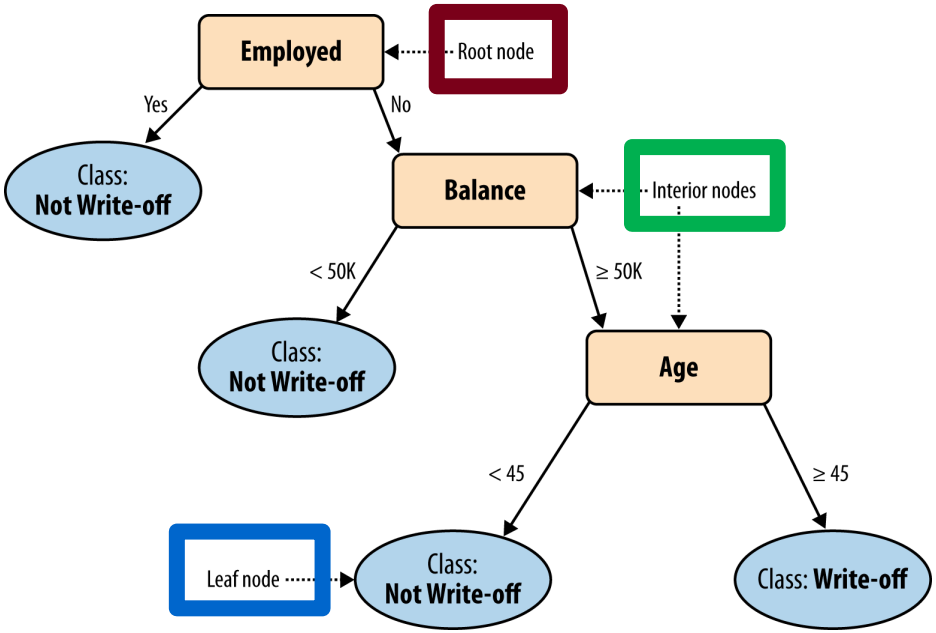


Multivariate Supervised Segmentation

- If we select the ***single* variable** that gives the most information gain, we create a very *simple* segmentation.

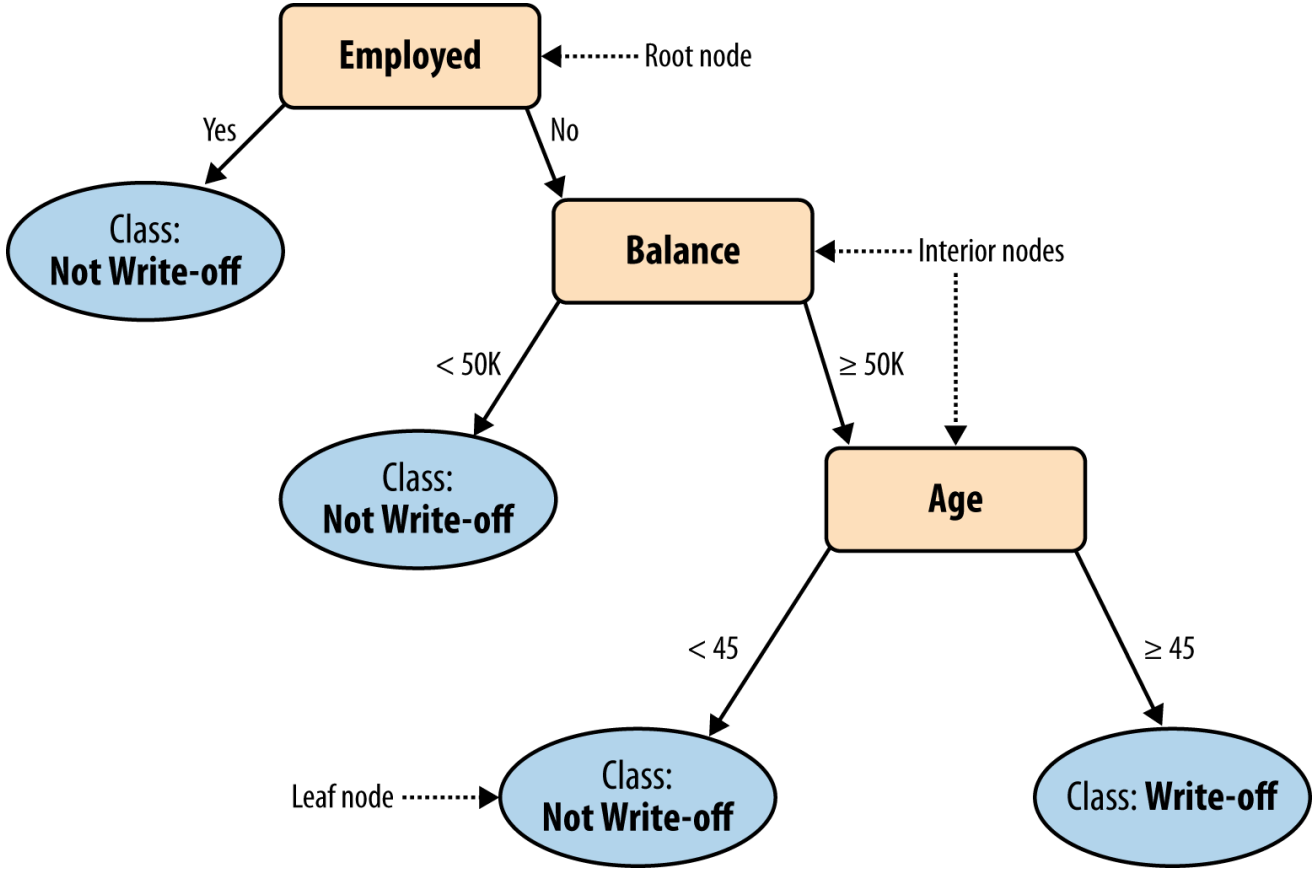
- If we **select multiple attributes** each giving some information gain, how do we put them together?

Tree-Structured Models



Tree-Structured Models

- Classify ‘John Doe’
  - Balance=115K, Employed=No, and Age=40



Tree-Structured Models: “Rules”

- No two parents share descendants
- There are no cycles
- The branches always “point downwards”
- Every example always ends up at a leaf node with some specific class determination
  - Probability estimation trees, regression trees (*to be continued*..)

Tree Induction

- How do we create a classification tree from data?
  - **divide-and-conquer** approach
  - take each data subset and ***recursively*** apply attribute selection to find the best attribute to partition it
- When do we stop?
  - The nodes are pure,
  - there are no more variables, or
  - even earlier (over-fitting – *to be continued*..)

Tree Induction: Algorithmic View

Input: labeled input data *D*

Create root node (represents entire *D*)

repeat

- Select eligible node for splitting
- Split selected node
  - Choose the “best” attribute for the node
  - Partition the data according to values of this attribute
  - Each partition becomes a new node

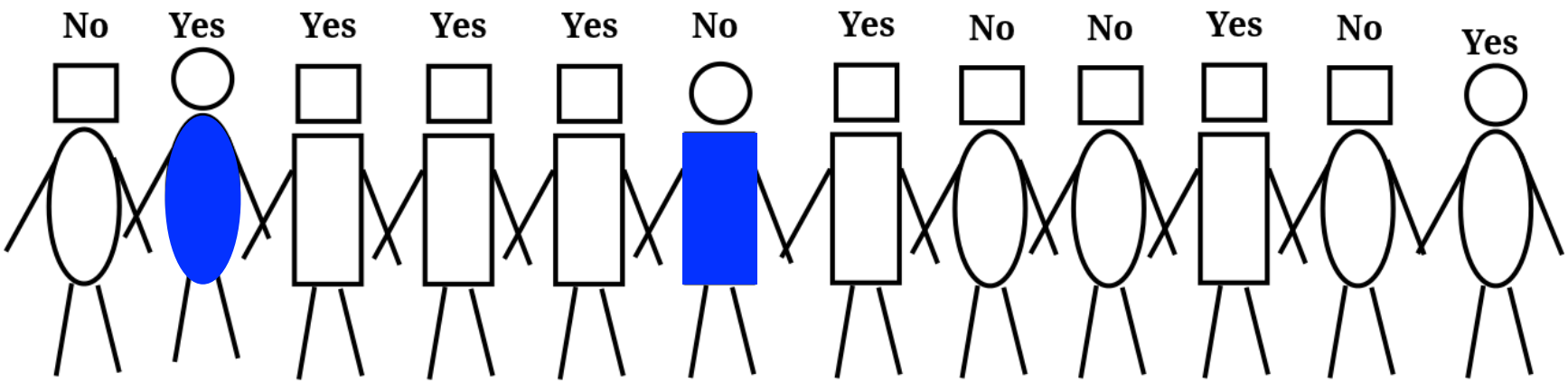
until no more eligible nodes to split

Prune overfitting nodes from the tree

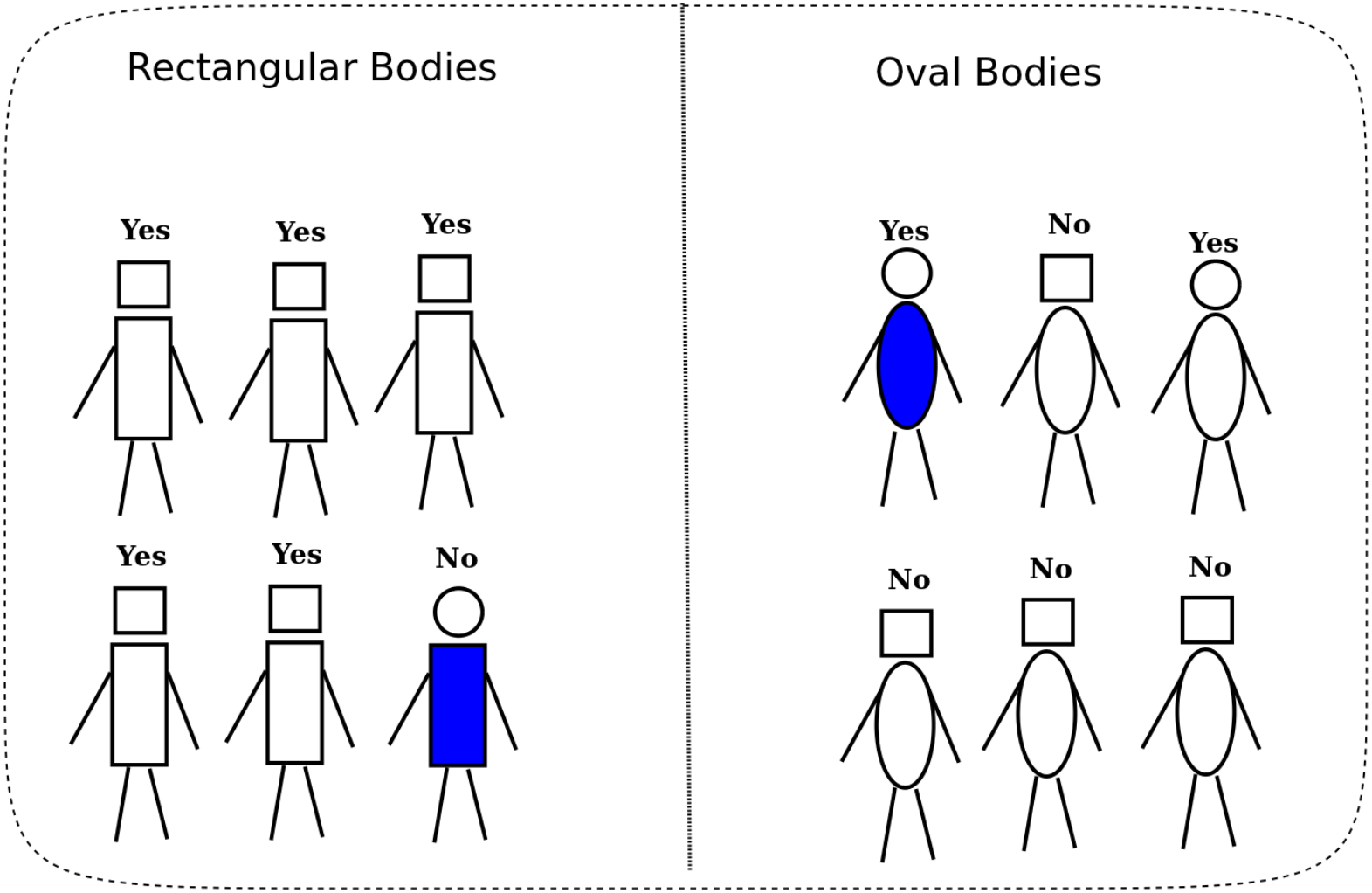
Label each leaf node with its dominant class

Output: decision tree

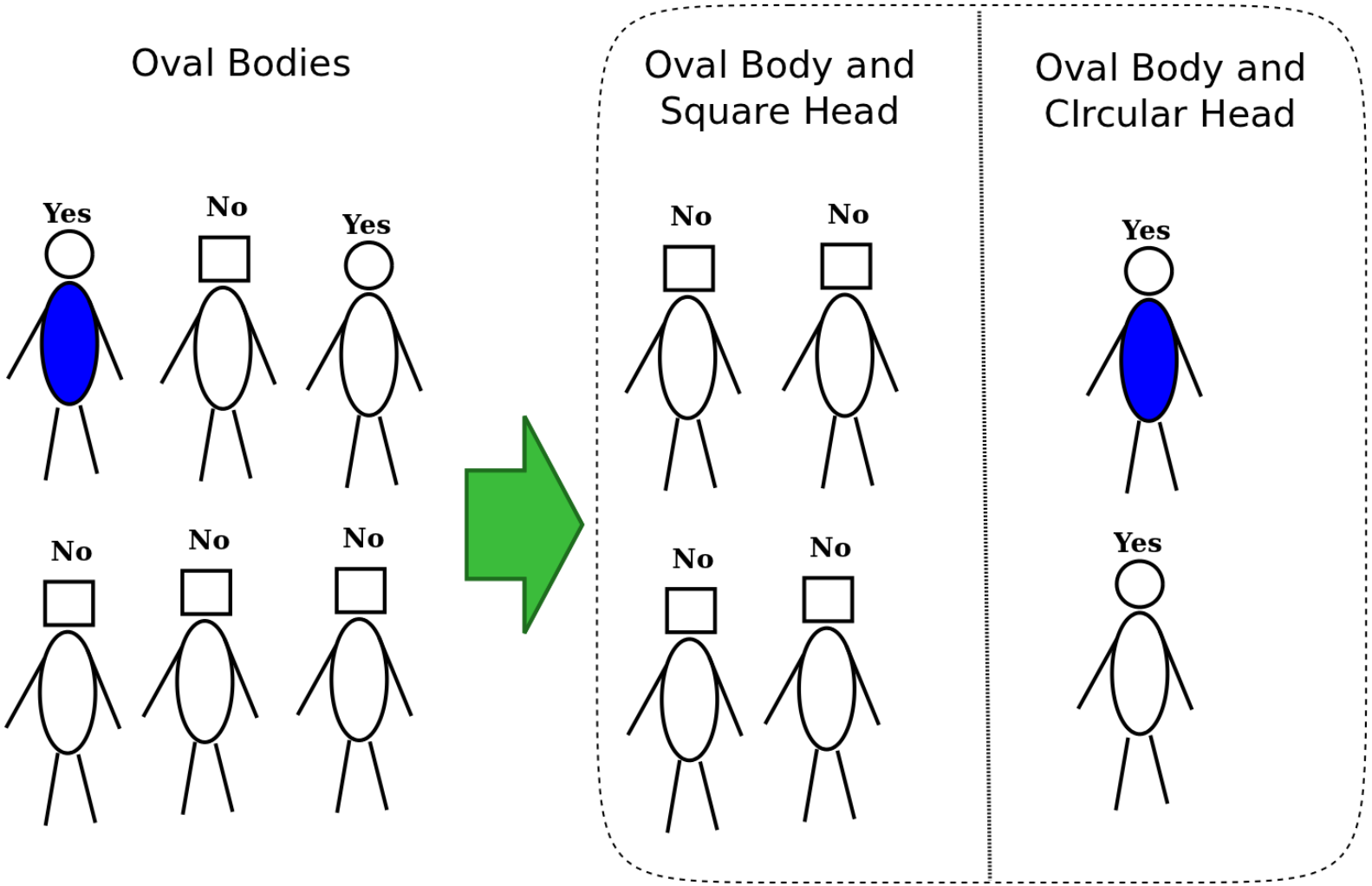
Supervised Segmentation



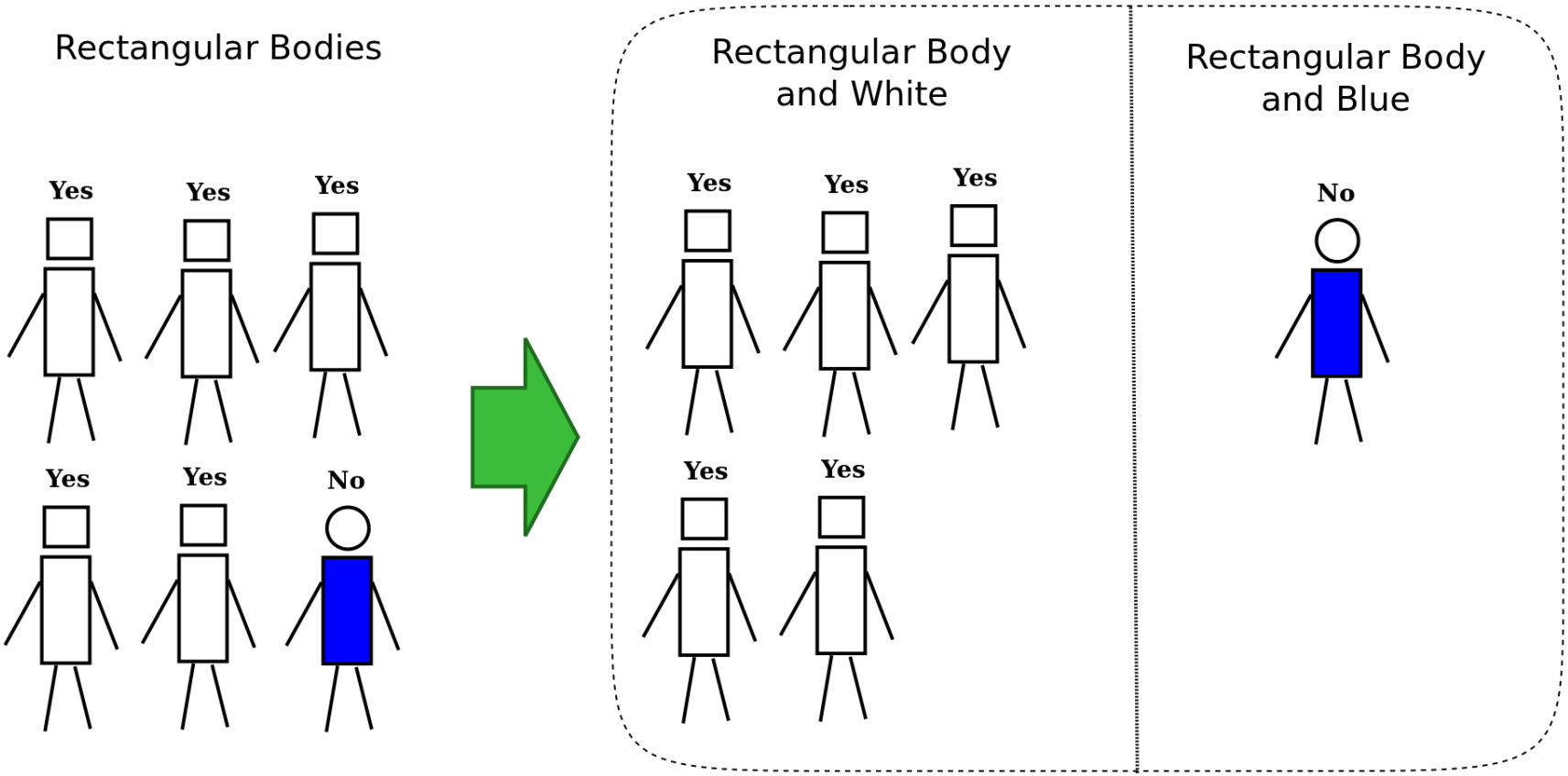
Supervised Segmentation



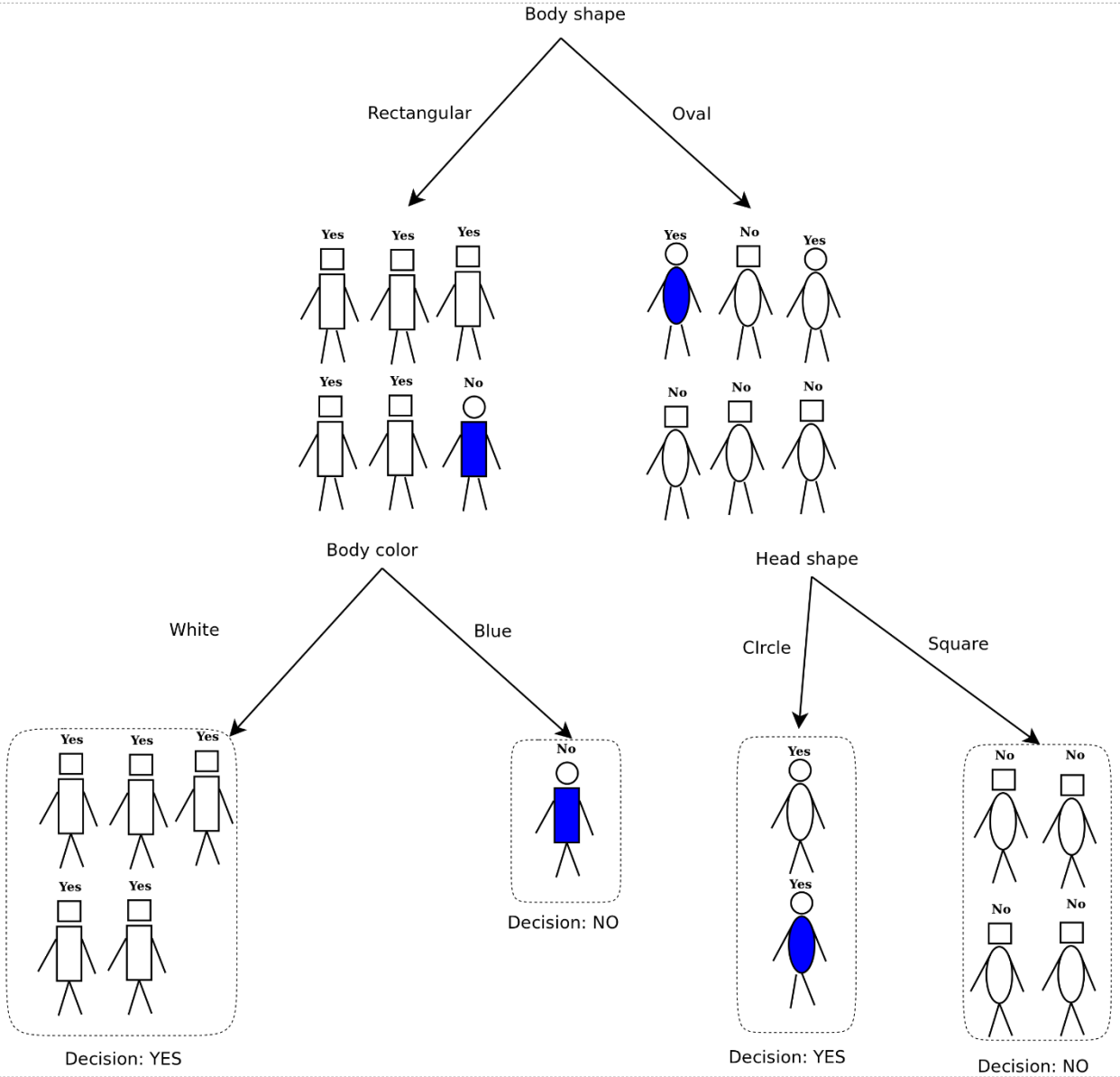
Supervised Segmentation



Supervised Segmentation



Supervised Segmentation



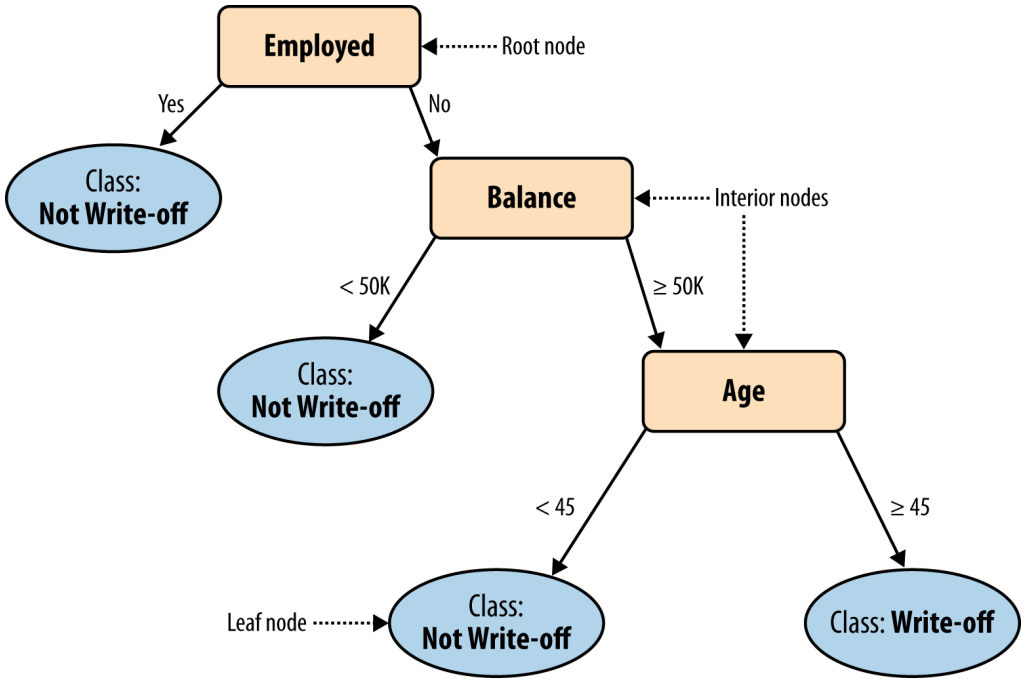
Why Decision Trees?

- Decision trees (DTs), or classification trees, are one of the most popular data mining tools
  - (along with linear/logistic regression)
- They're:
  - Easy to understand
  - Easy to implement
  - Easy to use
  - Computationally cheap
- Almost all data mining packages include DTs
- They have advantages for model comprehensibility, which is important for:
  - model evaluation
  - communication to non-DM-savvy stakeholders

Trees as Sets of Rules

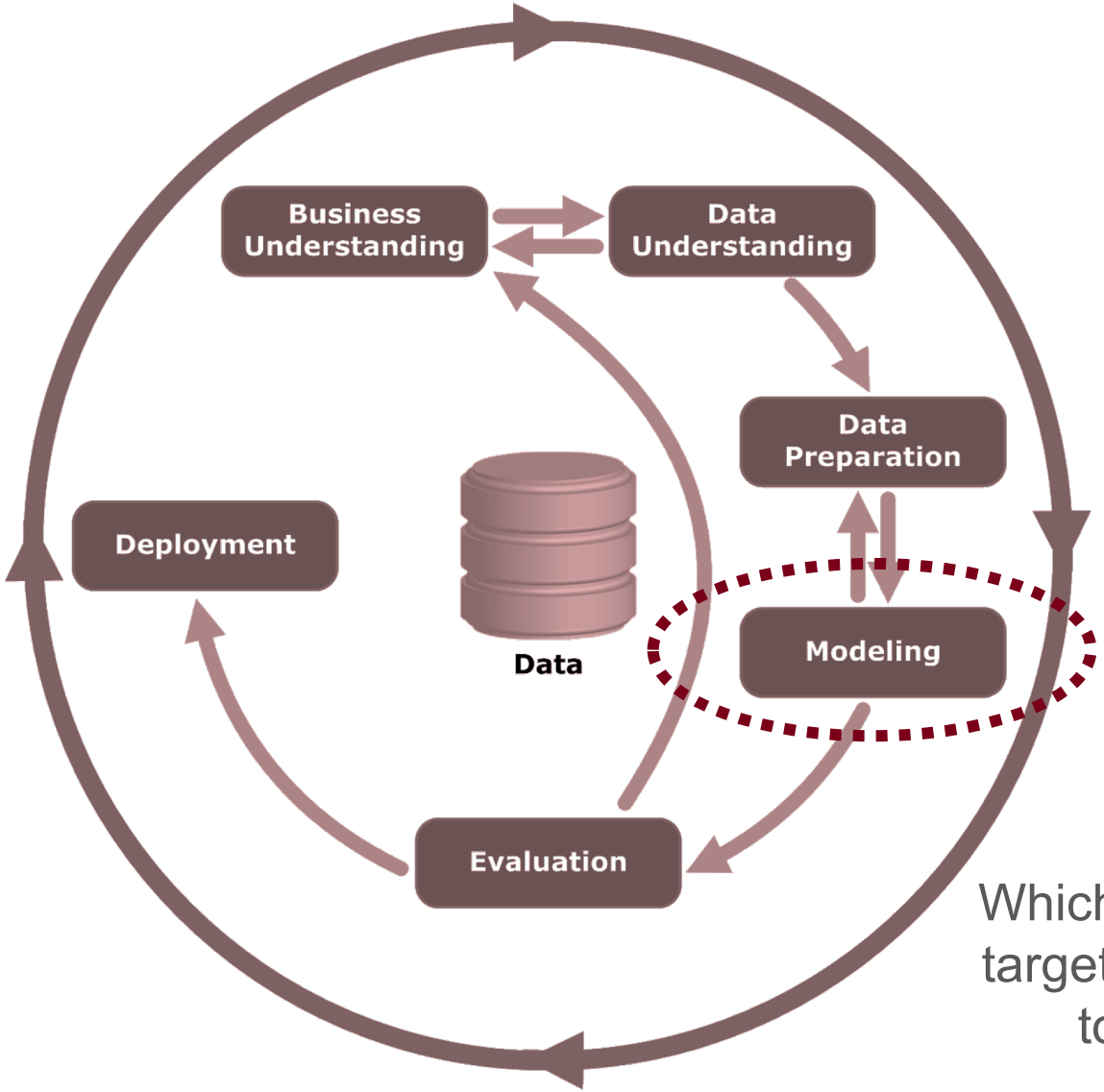
- The classification tree is equivalent to a rule set.
- Each rule consists of the attribute tests along the path connected with **AND**

Trees as Sets of Rules



- IF (Employed = Yes) THEN Class=No Write-off
- IF (Employed = No) AND (Balance < 50k) THEN Class=No Write-off
- IF (Employed = No) AND (Balance ≥ 50k) AND (Age < 45) THEN Class=No Write-off
- IF (Employed = No) AND (Balance ≥ 50k) AND (Age ≥ 45) THEN Class=Write-off

Let's focus back in on actually mining the data..

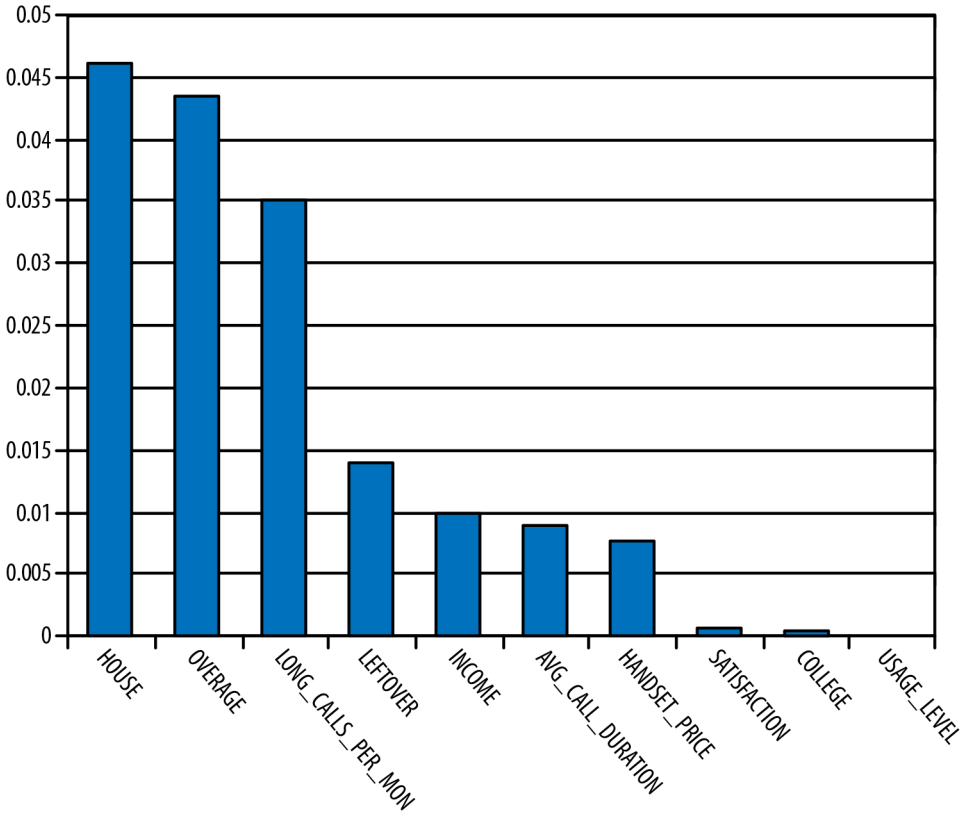


Which customers should TelCo target with a special offer, prior to contract expiration?

MegaTelCo: Predicting Churn with Tree Induction

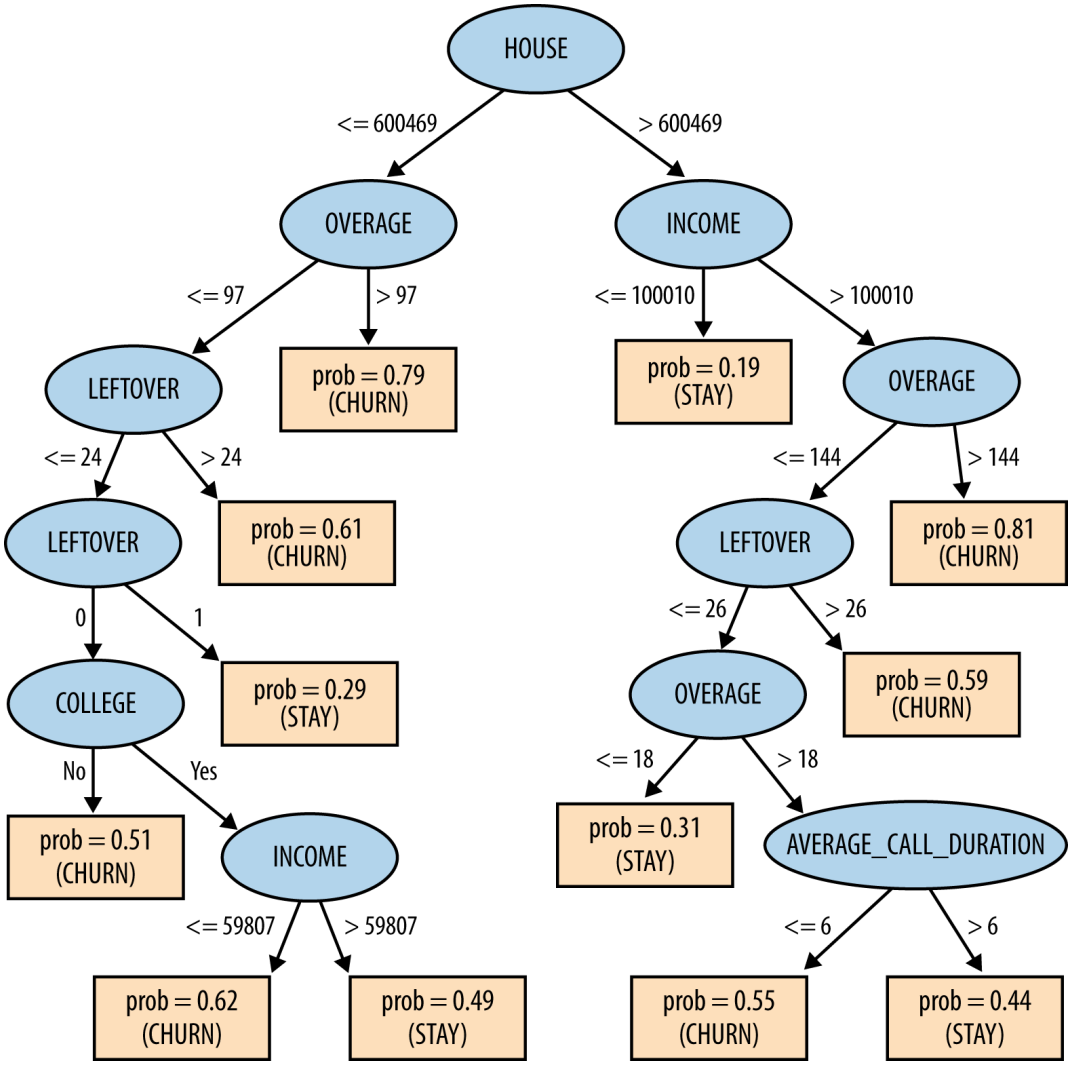
Variable	Explanation
COLLEGE	Is the customer college educated?
INCOME	Annual income
OVERAGE	Average overcharges per month
LEFTOVER	Average number of leftover minutes per month
HOUSE	Estimated value of dwelling (from census tract)
HANDSET_PRICE	Cost of phone
LONG_CALLS_PER_MONTH	Average number of long calls (15 mins or over) per month
AVERAGE_CALL_DURATION	Average duration of a call
REPORTED_SATISFACTION	Reported level of satisfaction
REPORTED_USAGE_LEVEL	Self-reported usage level
LEAVE *(Target variable)*	Did the customer stay or leave (churn)?

MegaTelCo: Predicting Churn with Tree Induction



Rank	Info. gain	Attribute name
1	0.0461	HOUSE
2	0.0436	OVERAGE
3	0.0350	LONG_CALLS_PER_MON
4	0.0136	LEFTOVER
5	0.0101	INCOME
6	0.0089	AVG_CALL_DURATION
7	0.0076	HANDSET_PRICE
8	0.0003	SATISFACTION
9	0.000	COLLEGE
10	0.000	USAGE_LEVEL

MegaTelCo: Predicting Churn with Tree Induction



EMORY
UNIVERSITY

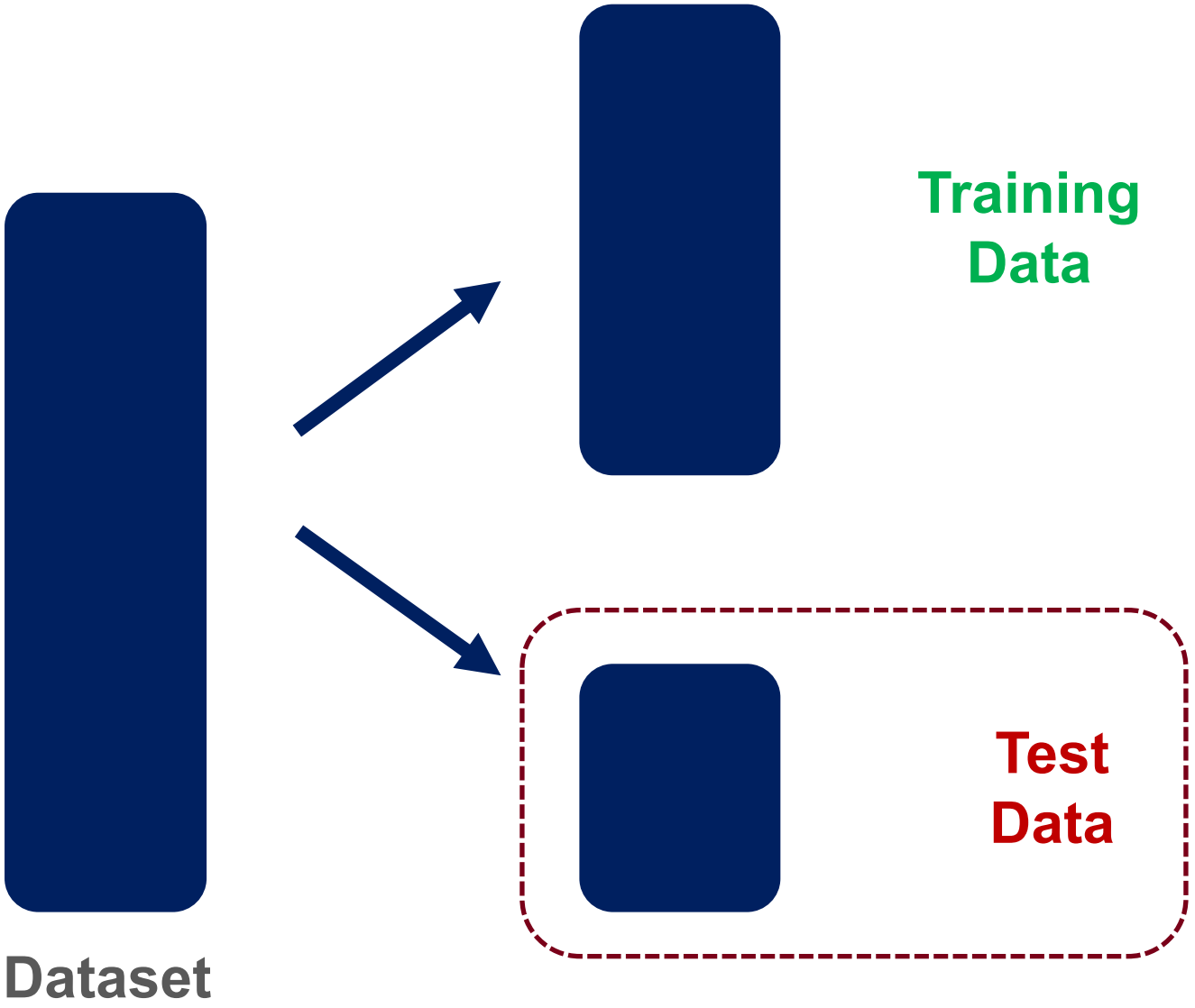
Evaluation

Model Evaluation

How do we measure **generalization performance**?

- Multiple classification techniques are available
  - Within the same technique, multiple parameterizations
- Which model is the best model?
  - Need to assess each model's performance
- Reminder:
  - Predictive performance measures must be calculated on **test** (i.e., NOT training!) data

Generalization Performance



Evaluating Classifiers: Plain Accuracy

$$\text{accuracy} = \frac{\text{Number of correct classification decisions made}}{\text{Total number of classification decisions made}}$$

$$= 1 - \text{error rate}$$

- *Too* simplistic..

Accuracy and Misclassification Error

- **Error** = classifying a record as belonging to one class when it belongs to another class
- **Error rate** = percent of misclassified records out of the total records in the validation data
- **Accuracy** = 1 – Error Rate

Misclassifications vs Correct Classifications

Actual Class	Prediction	
Positive	Positive	✓
Positive	Negative	✗
Negative	Positive	✗
Negative	Negative	✓

False Negative
vs
False Positive

Evaluating Classifiers: The Confusion Matrix

- A confusion matrix for a problem involving n classes is an $n \times n$ matrix
  - with the columns labeled with actual classes and the rows labeled with predicted classes
- It separates out the decisions made by the classifier
  - making explicit how one class is being confused for another

Predicted	Actual	
	Positive	**Negative**
Positive	True Positives (TP)	False Positives (FP)
Negative	False Negatives (FN)	True Negatives (TN)

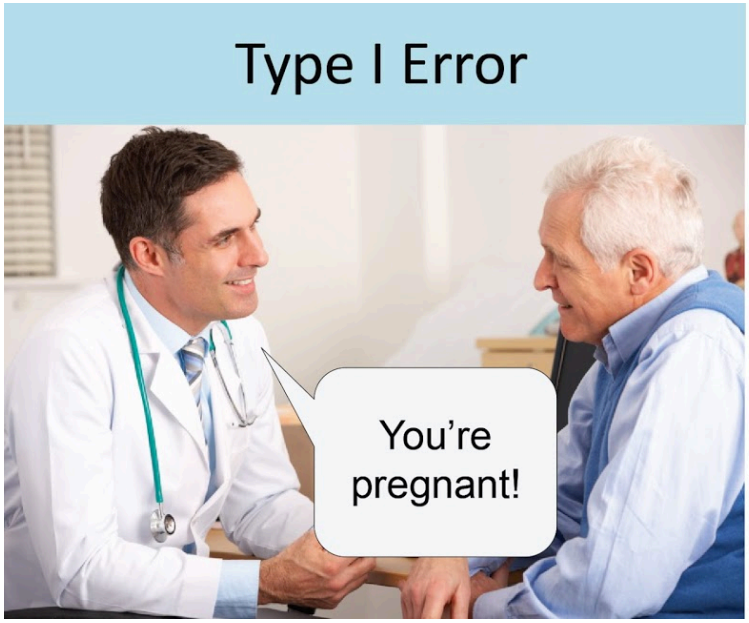
Evaluating Classifiers: The Confusion Matrix

Predicted	Actual	
	Positive	**Negative**
Positive	True Positives (TP)	False Positives (FP)
Negative	False Negatives (FN)	True Negatives (TN)

- The errors of the classifier are the false positives and false negatives
  - In statistics:
    - False Positives = Type I errors
    - False Negatives = Type II errors
- Similar type of matrix can be constructed when having more than 2 classes

Type I & Type II Errors

False Positive (FP)



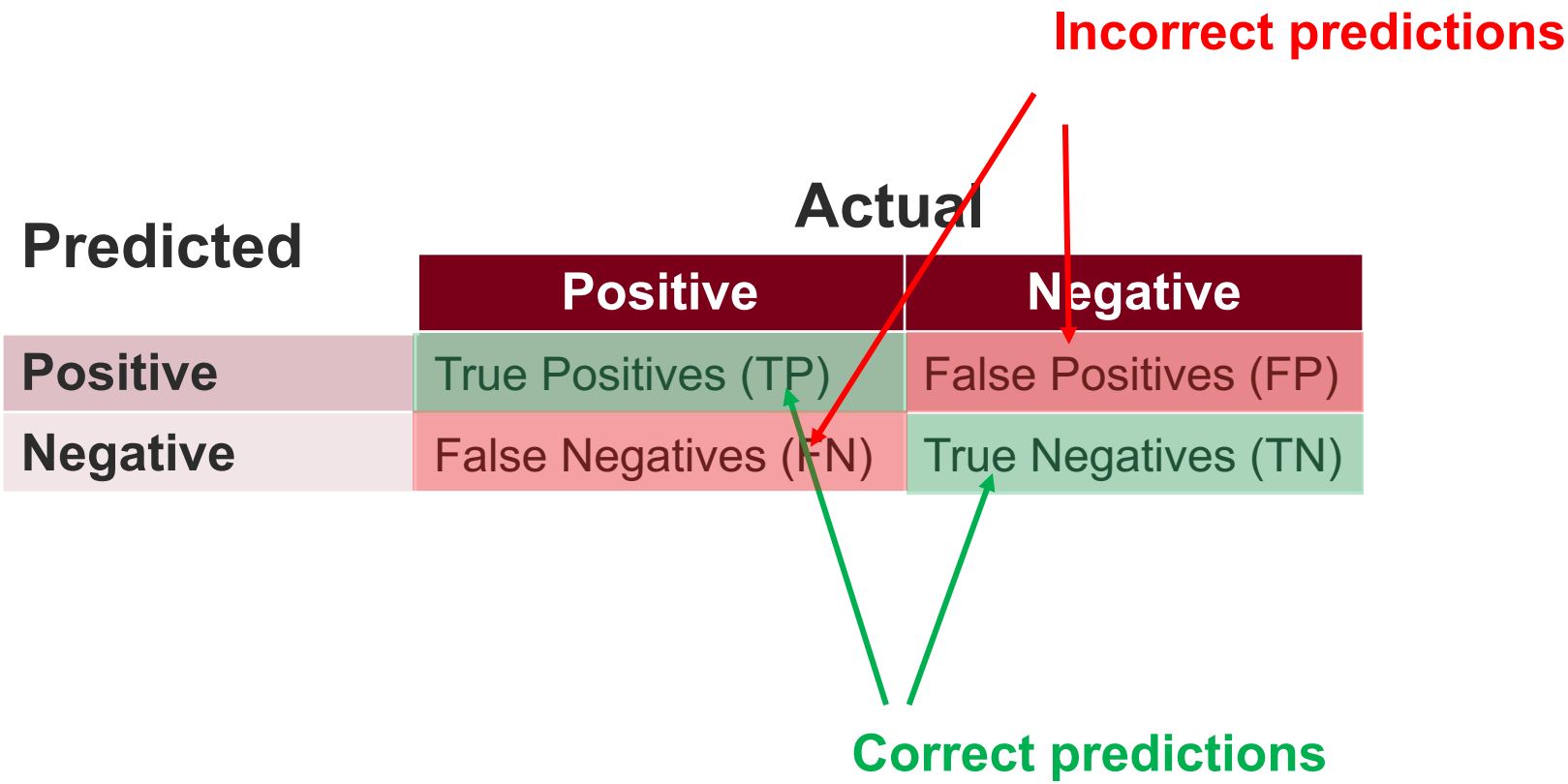
False Negative (FP)

Building a Confusion Matrix

Default Truth	Model Prediction
0	0
1	1
0	1
0	1
0	0
1	1
0	0
0	0
1	1
1	0

Actual class / Predicted class	Default	No Default	Total
Default	3	2	5
No Default	1	4	5
Total	4	6	10

The Confusion Matrix



Evaluation of Multi-Class Classification Tasks

- Assume that the target variable has more than two values (classes); e.g., C_1, C_2, ..., C_k
  - E.g., Risk = High, Medium, or Low
- Predicted(C_i)
  - Set of data points predicted to be of class C_i by the model
- Actual(C_i)
  - Set of data points that actually are of class C_i
- Then:

$$\text{Precision}_{C_i} = \frac{|\text{Predicted}(C_i) \cap \text{Actual}(C_i)|}{|\text{Predicted}(C_i)|}$$

The Confusion Matrix & Other Common Metrics

Predicted	Actual: Positive	Actual: Negative
Positive	True Positives (TP)	False Positives (FP)
Negative	False Negatives (FN)	True Negatives (TN)

- Note that all predictions: All = TP + FP + FN + TN
- Some common metrics:
  - **Accuracy** = (TP + TN) / (TP + FP + FN + TN)
  - **Precision**$_{Positive}$ = TP / (TP + FP) Precision$_{Negative}$ = TN / (TN + FN)
  - **Recall**$_{Positive}$ = TP / (TP + FN) Recall$_{Negative}$ = TN / (TN + FP)
    - Precision and Recall metrics can be calculated for any class
- Very similar calculations when having more than 2 classes

F-Measure

- The special-purpose combination of the precision and recall score (of an important class):

$$\text{F-measure}_{\text{Positive}} = 2 \times \frac{\text{Precision}_{\text{Positive}} \times \text{Recall}_{\text{Positive}}}{\text{Precision}_{\text{Positive}} + \text{Recall}_{\text{Positive}}}$$

- Represents a *harmonic* mean of precision and recall
- Always between 0 (worst performance) and 1 (best performance)

Naïve (or Majority) Rule

- **Naïve rule**: classify all records as belonging to the majority (most prevalent) class
  - Not a real method (typically does not make sense in many real-world applications)
  - Sometimes used as benchmark: we hope to outperform it
  - However, when goal is to identify high-value but rare outcomes, we may do well by doing worse than the naïve rule
    - to be continued..

Predicted	Actual: Positive	Actual: Negative
Positive	True Positives (TP)	False Positives (FP)
Negative	False Negatives (FN)	True Negatives (TN)

EMORY
UNIVERSITY

GOIZUETA
BUSINESS
SCHOOL

Introduction to Classification

Introduction to Business Analytics

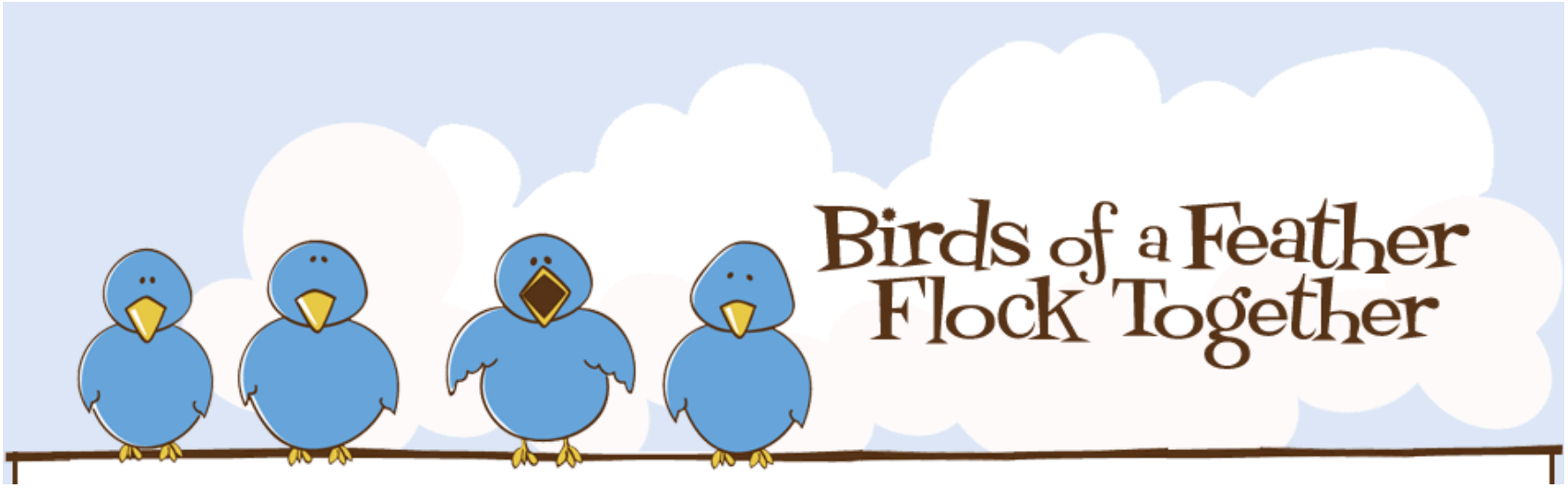
Vilma Todri
Assistant Professor
Goizueta Business School
Emory University
vtodri@emory.edu

EMORY
UNIVERSITY

k-Nearest Neighbors

kNN

k-Nearest Neighbors (kNN) algorithm

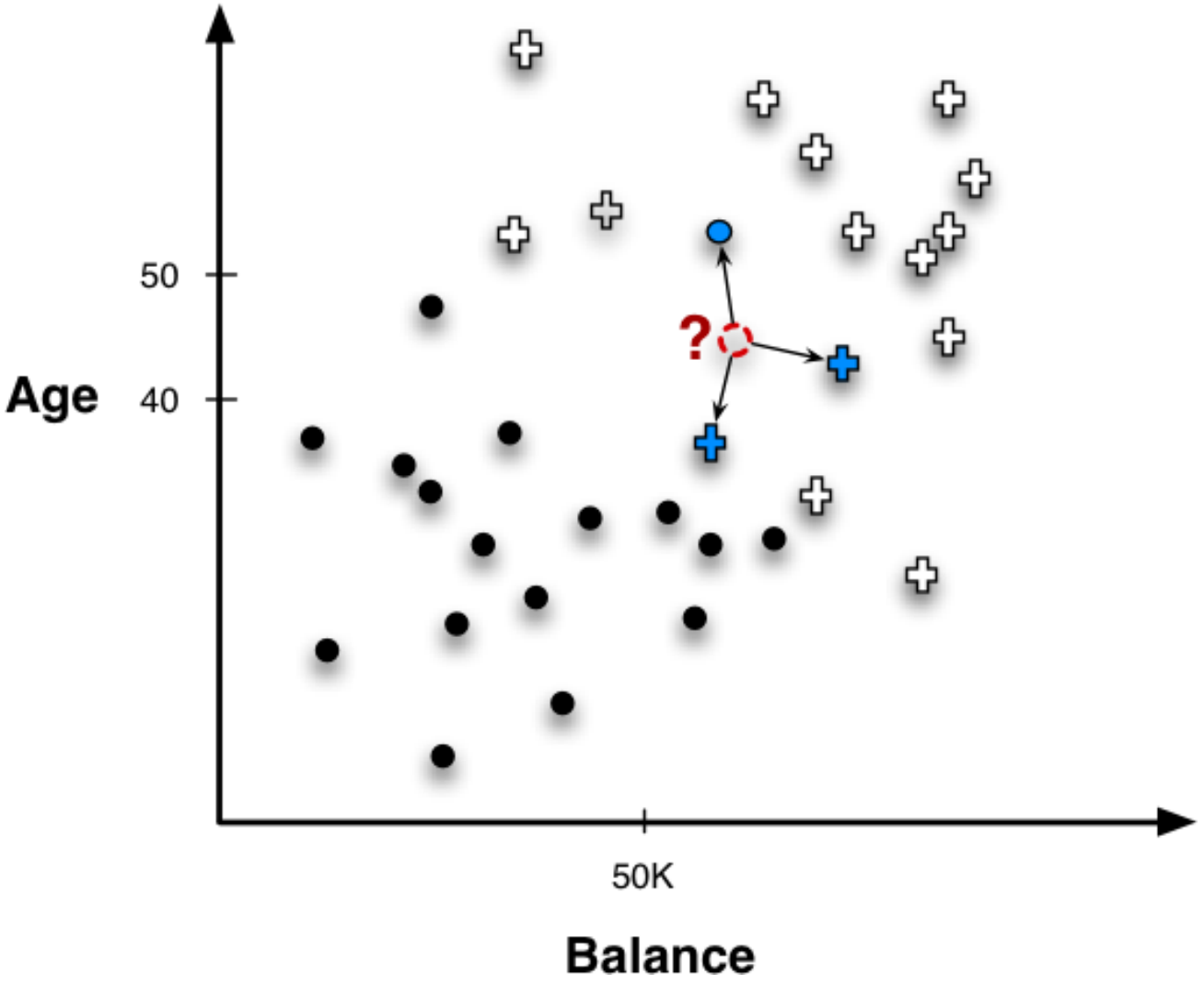


kNN Assumption: points that are close to one another are similar

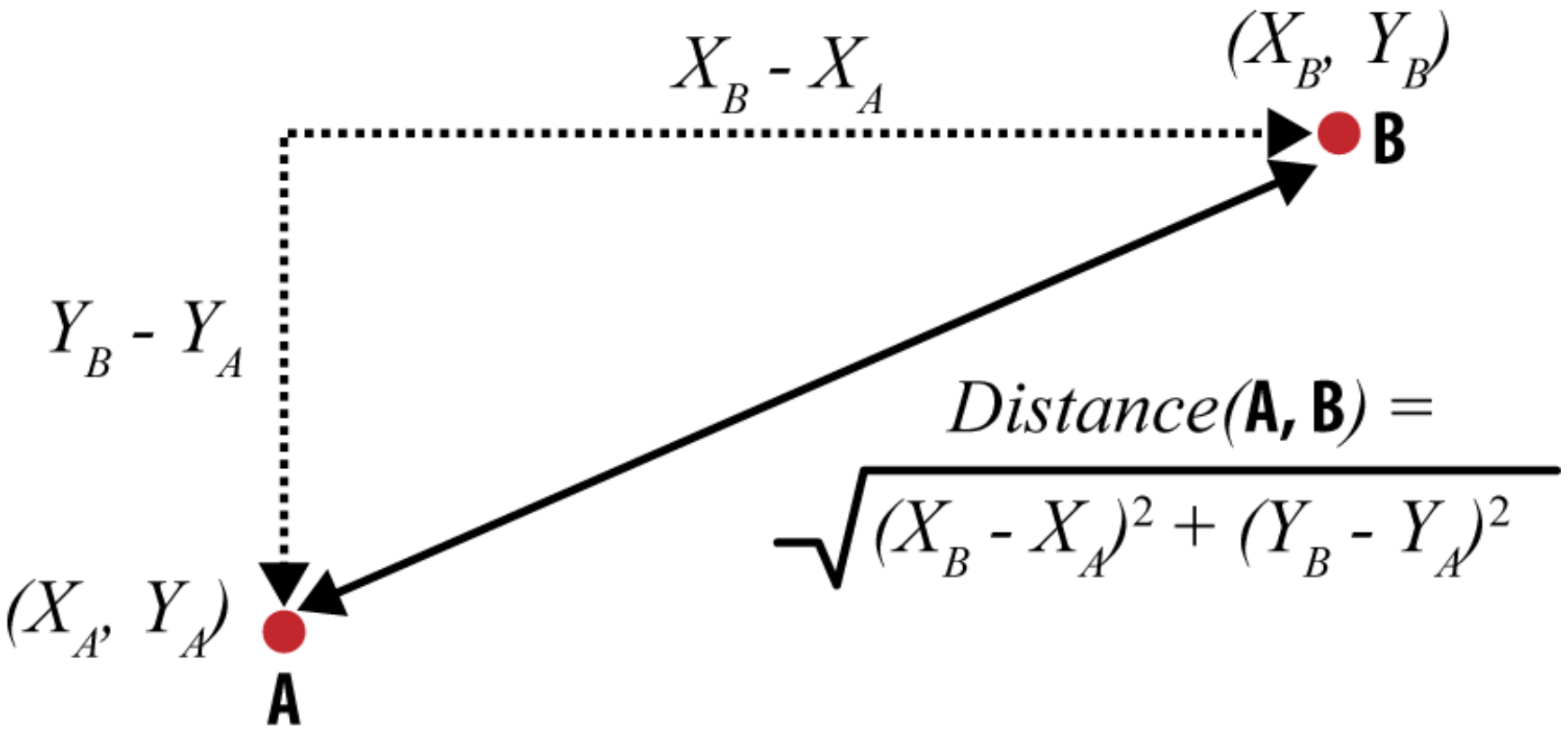
k-NN Classifiers: Basic Idea

- k-NN stands for "k Nearest Neighbors"
- For a given *p*-dimensional record to be classified, **identify nearby records**
  - "Near" means records with similar attribute values x_1, x_2, …, x_p
- Classify the record as whatever the predominant class is among the nearby records (the "neighbors")

Nearest Neighbors for Predictive Modeling



Euclidean Distance



Similarity and Distance

- If two *objects* can be represented as *feature vectors*, then we can compute the distance between them

- When retrieving neighbors, we don't use the target variable as an attribute

Attribute	Person A	Person B
Age	23	40
Years at current address	2	10
Residential status (1=Owner, 2=Renter, 3=Other)	2	1

Nearest Neighbors for Predictive Modeling

$$Distance(\mathbf{A}, \mathbf{B}) = \sqrt{(X_B - X_A)^2 + (Y_B - Y_A)^2}$$

Customer	Age	Income (1000s)	Cards	Response (target)	Distance from David
David	*37*	*50*	*2*	*?*	0
John	35	35	3	Yes	
Rachael	22	50	2	No	
Ruth	63	200	1	No	
Jefferson	59	170	1	No	
Norah	25	40	4	Yes	

Other Distance Functions

$$d_{Manhattan}(\boldsymbol{X}, \boldsymbol{Y}) = \|\boldsymbol{X} - \boldsymbol{Y}\|_1 = |x_1 - y_1| + |x_2 - y_2| + \cdots$$

$$d_{Jaccard}(X, Y) = 1 - \frac{|X \cap Y|}{|X \cup Y|}$$

$$d_{Cosine}(\boldsymbol{X}, \boldsymbol{Y}) = 1 - \frac{\boldsymbol{X} \cdot \boldsymbol{Y}}{\|\boldsymbol{X}\|_2 \cdot \|\boldsymbol{Y}\|_2}$$

Key Issue: Normalization

- **Problem**: Raw distance measures are highly influenced by scale of measurements
  - E.g., income=$100,000; height=1.50m
    - Income will highly dominate height in distance computations, because of highly differing scales of the two attributes
- **Solution**: normalize (standardize) the data first
  - Normalization transforms variables (attributes) of different scales (different orders of magnitude) into similar scale so that they can be compared and can contribute equally to the distance computations

Normalization Approaches

- Scaling/normalizing is used to standardize the intervals before measuring distances
  - min-max scaling numerical attribute to interval [0,1]

$$z = (x - \min) / (\max - \min)$$

  - x := original value of the attribute
  - min := smallest value of the attribute
  - max := largest value of the attribute
  - z is the resulting (scaled) value of the attribute in the range [0,1]
- z-score scaling to standardize the intervals

$$z = (x - m) / s$$

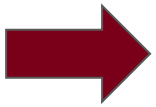
  - m := mean value of the attribute
  - s := standard deviation (or mean absolute deviation, which is more robust to outliers than standard deviation)
- When attributes have different importance
  - Weighted distances may be used
  - E.g., $d(A, B) = w_1|a_1 - b_1| + \cdots + w_k|a_k - b_k|$

Min-Max Approach: Example

- Setting: Consider the age and income data of several employees (as provided below)
- Task: Normalize the data using min-max approach

$$z = (x - \min) / (\max - \min)$$

Name	Age	Income
Alice	70	100,000
Bob	25	50,000
Cindy	30	60,000
David	20	70,000
Earl	60	80,000



Name	Age(Norm)	Income(Norm)
Alice	1.0	1.0
Bob	0.1	0.0
Cindy	0.2	0.2
David	0.0	0.4
Earl	0.8	0.6

$$z = (x - \min) / (\max - \min)$$
$$= (60\text{-}20) / (70\text{-}20) = 0.8$$

How many neighbors?

- No simple answer
- ***Odd numbers*** are convenient for breaking ties for majority vote classification with two-class problems
- In general, the greater k is, the more the estimates are smoothed out among neighbors
    - What happens if k=n where n is the size of the training data set?
- We will revisit this question again (To be Continued...)

How much influence?

- Even when we have chosen k, some "nearest neighbors" are nearer than others
- Should this influence how they're used?
  - i.e., Should we allow the most similar neighbors to have greater influence on our prediction?
- We can use *weighted voting* or *similarity moderated voting* such that each neighbor's contribution is scaled by its similarity.
- The inverse of the square of the distance is commonly used e.g., $w(x, y) = \frac{1}{dist^2(x,y)}$

Similarity-moderated kNN

Name	Distance	Similarity weight	Contribution	Class
Rachael	15.0	0.004444	0.344	No
John	15.2	0.004348	0.336	Yes
Norah	15.7	0.004032	0.312	Yes
Jefferson	122.0	0.000067	0.005	No
Ruth	152.2	0.000043	0.003	No
		0.012934	1.000	

0.65 Yes
0.35 No

Advantage: Weighted scoring reduces the importance of deciding how many neighbors to use.

Why k-NN?

- "Lazy" learning approach
  - As opposed to "eager" approaches (e.g., decision trees)
  - No model building (data as model) → Faster to train but slower to estimate
- Enhancements
  - Weighted distance (closer neighbors have more impact)
- Strengths
  - Easy to implement and use
  - Robust (handles noisy data well, except for very low k values)
  - No statistical / distributional assumptions required
  - Captures complex interactions between variables without building models
- Weaknesses
  - Takes more time to perform estimation; computational efficiency
  - Requires a lot of storage
  - Lack of interpretable model
  - Curse of dimensionality and domain knowledge

Thank you!

Questions?