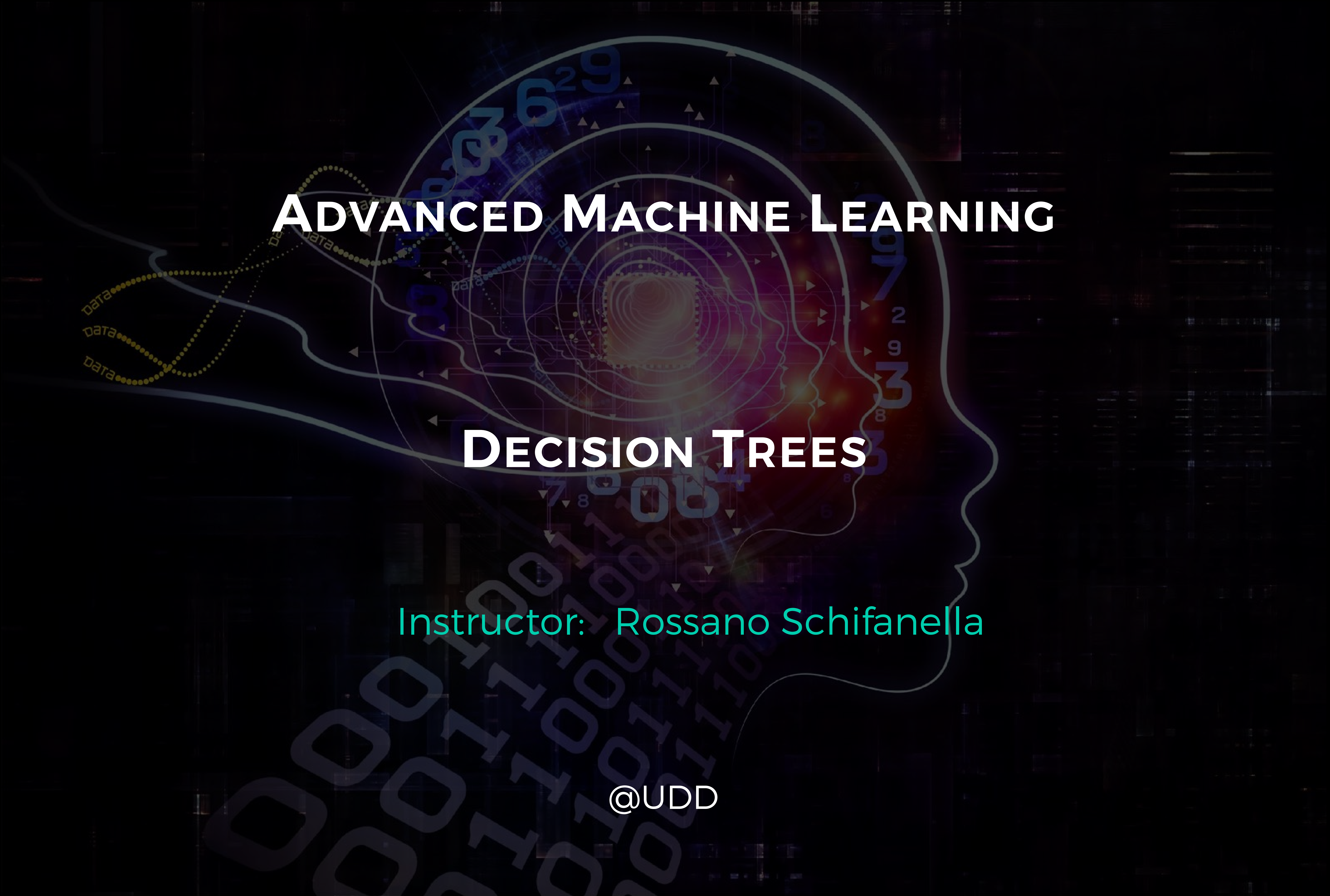


Advanced Machine Learning

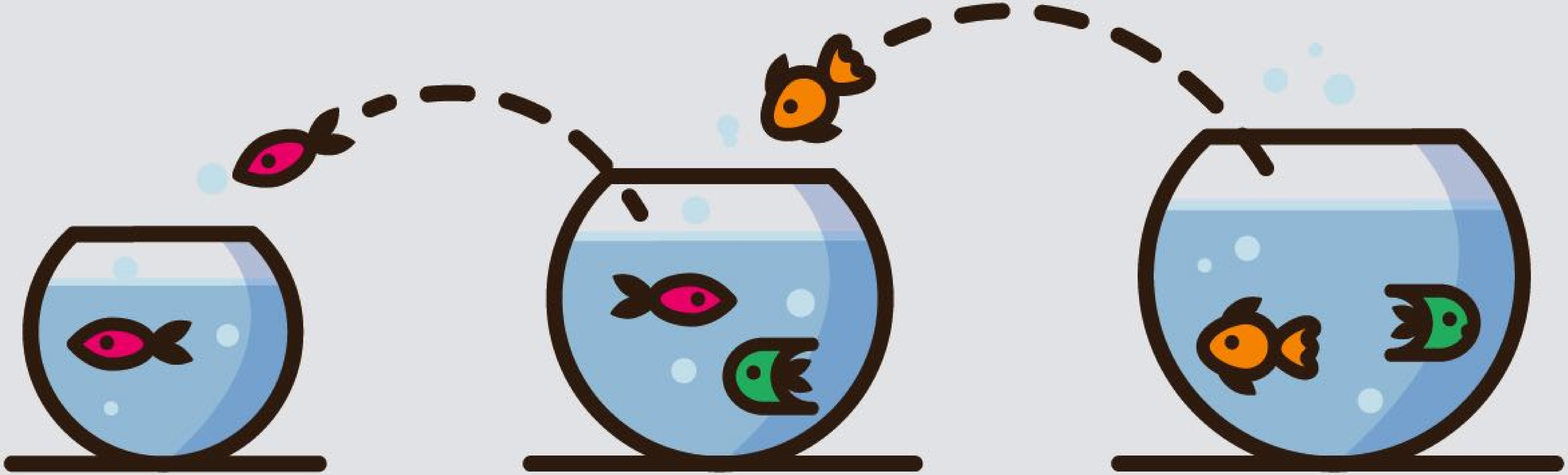
Decision Trees

Instructor: Rossano Schifanella

@UDD

Customer Churn Problem

Objective is retaining customers



Tasks

Do our customers fall into different groups?

no specific target has been specified for the grouping.

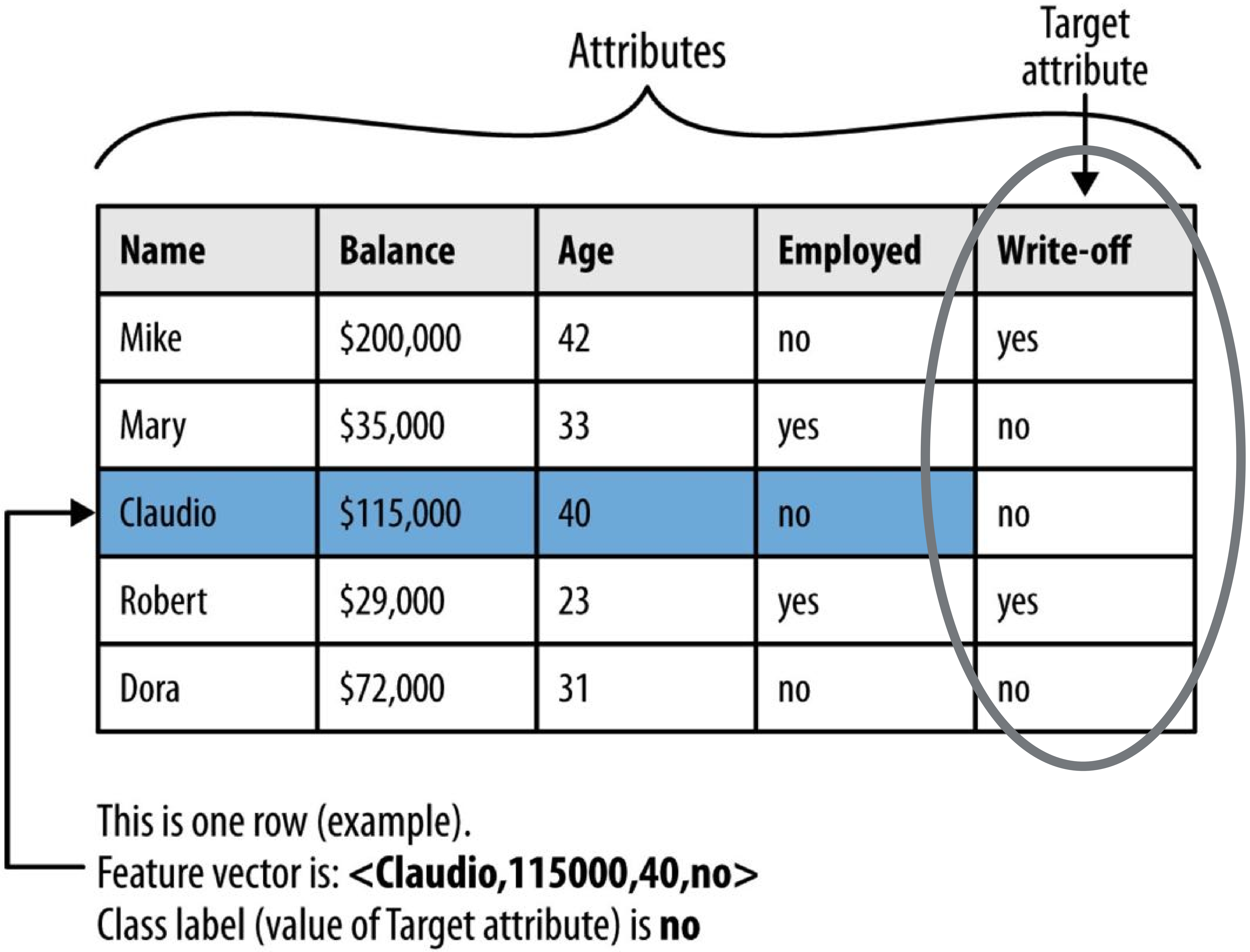
UNSUPERVISED
(descriptive)

Can we find groups of customers who have particularly high likelihoods of canceling their service soon after their contracts expire?

specific target defined: take action based on likelihood of churn

SUPERVISED
(predictive)

Data



Attributes
Target attribute

Name	Balance	Age	Employed	Write-off
Mike	$200,000	42	no	yes
Mary	$35,000	33	yes	no
Claudio	$115,000	40	no	no
Robert	$29,000	23	yes	yes
Dora	$72,000	31	no	no

This is one row (example).
Feature vector is: **<Claudio,115000,40,no>**
Class label (value of Target attribute) is **no**

If present, this **labeled** data. Not used in the learning phase!

Models

A model is a simplified representation of reality created to serve a purpose. It is simplified based on some assumptions about what is and is not important for the specific purpose, or sometimes based on constraints on information or tractability.

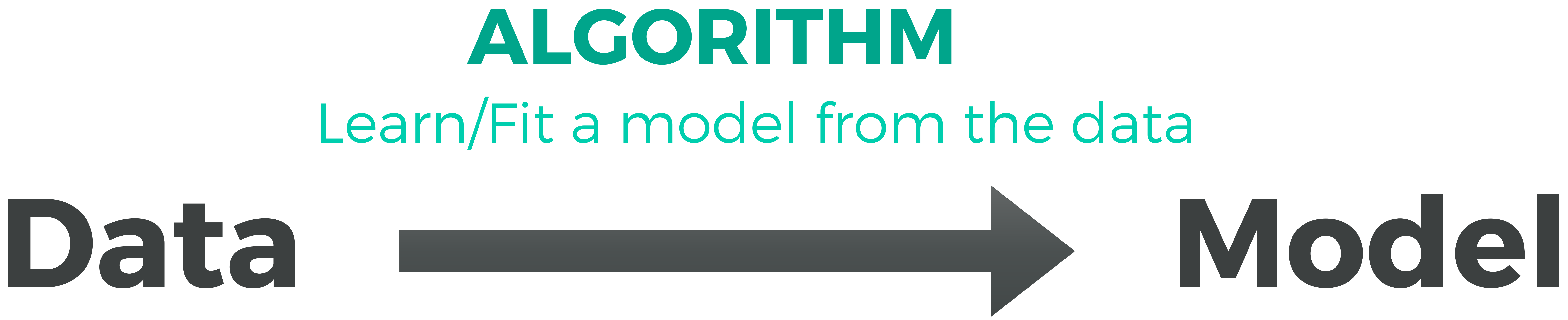
Models for Classification

Decision Trees
Random Forest
K-NN
Naive Bayes
SVM
Logistic Regression

Models for Regression

Linear Regression
Lasso
Ridge

Machine Learning Algorithm



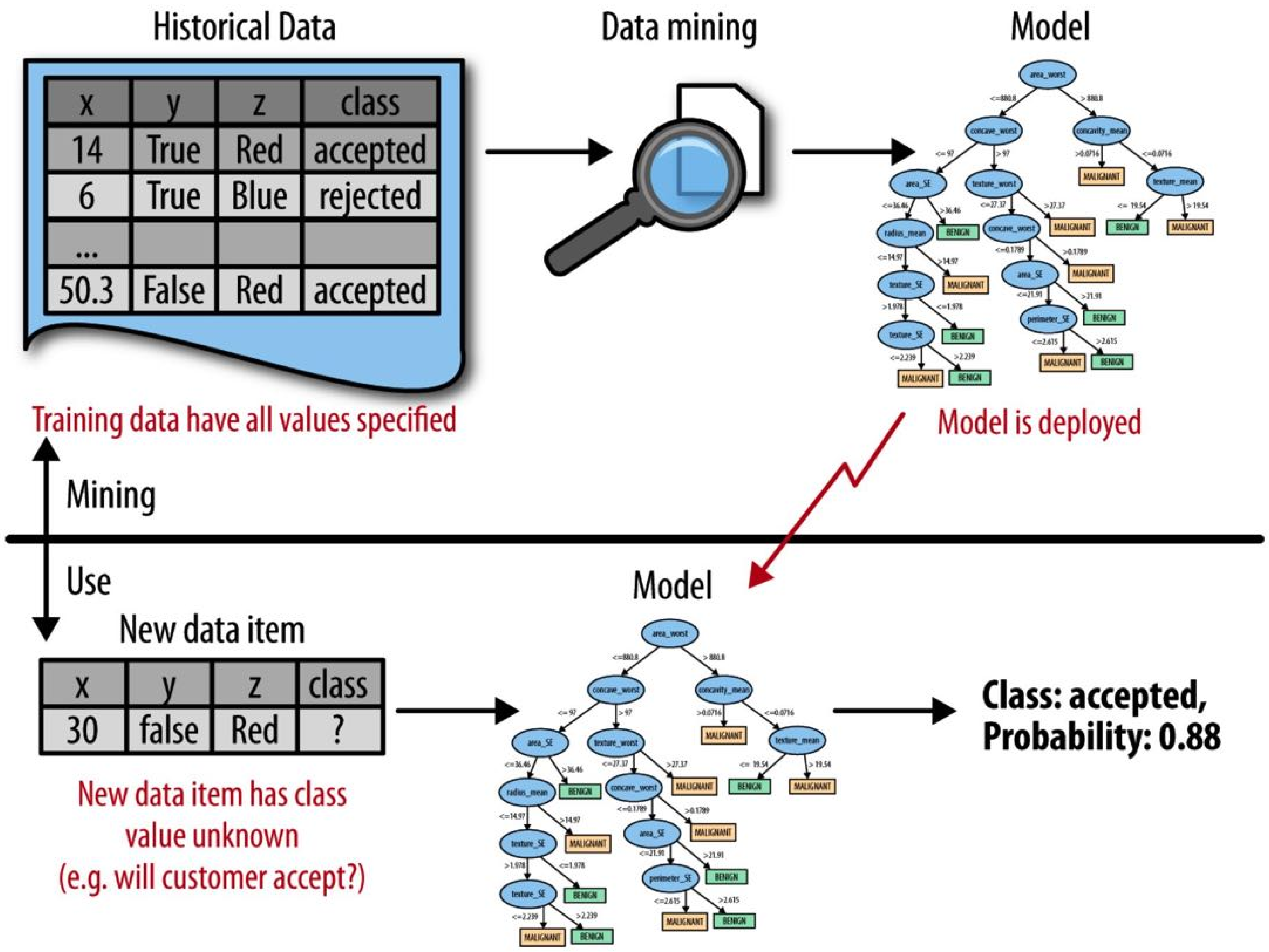
ALGORITHM

Learn/Fit a model from the data

Data

Model

Example of a Classifier



Historical Data

x	y	z	class
14	True	Red	accepted
6	True	Blue	rejected
...			
50.3	False	Red	accepted

Training data have all values specified

Data mining

Model

Model is deployed

Mining

Use

Model

New data item

x	y	z	class
30	false	Red	?

New data item has class value unknown (e.g. will customer accept?)

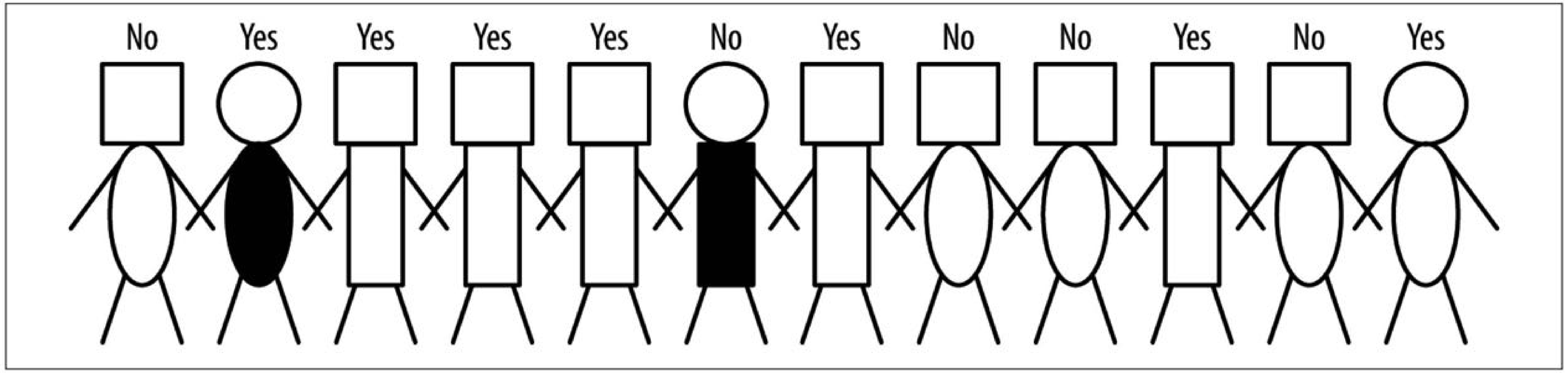
Class: accepted, Probability: 0.88

Decision Trees

Supervised Segmentation

- How can we **select one or more attributes/features/variables that will best divide the samples with respect to our target variable** of interest?
  - Middle-aged professionals who reside in New York City on average have a churn rate of 5%
    - What is the segment?
    - What is the target variable?

Supervised Segmentation



Attributes:

- **head-shape**: square, circular
- **body-shape**: rectangular, oval
- **body-color**: gray, white

Target:

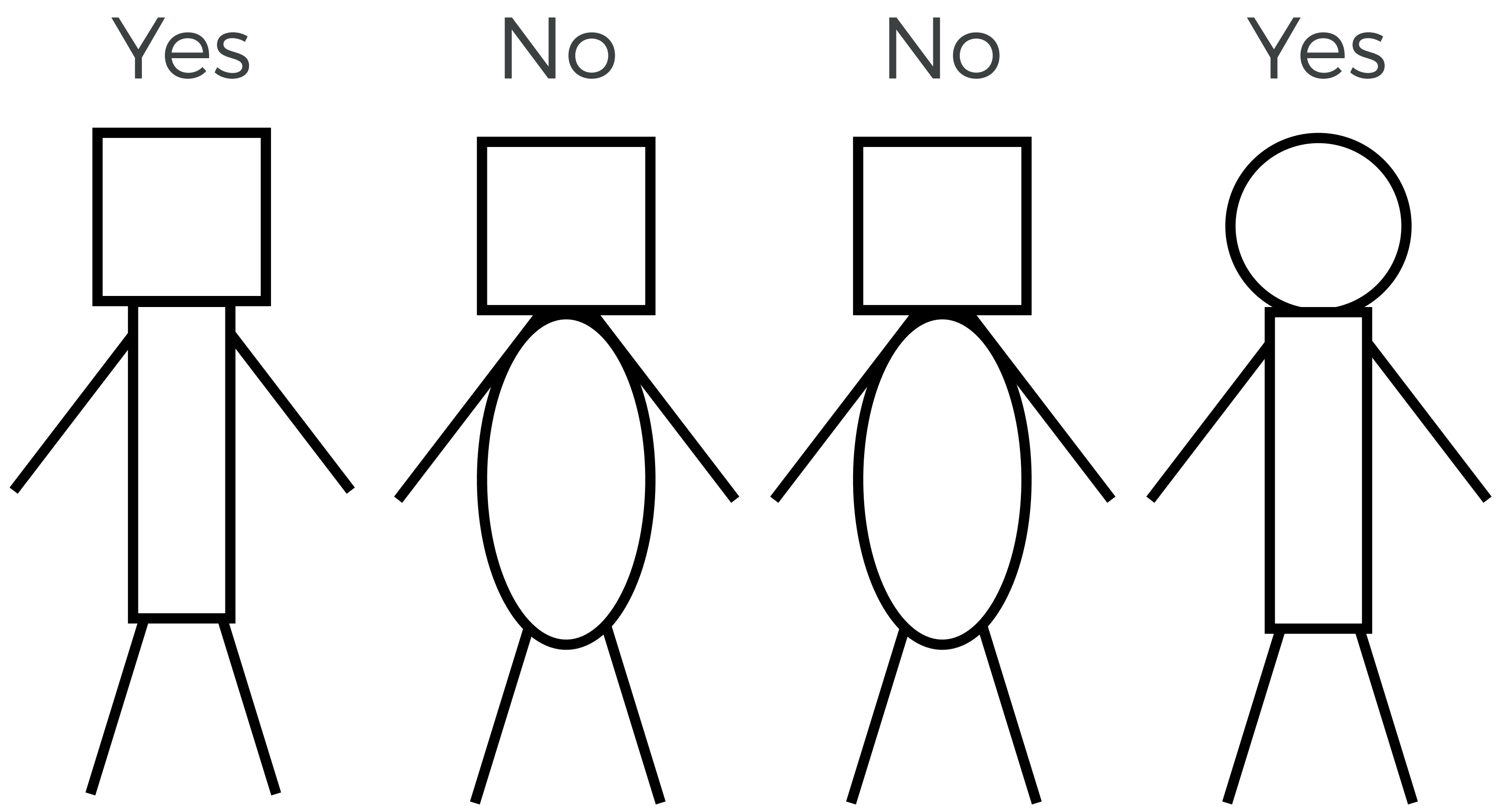
- **write-off**: yes/no

Which of the attributes would be the best to segment these people in groups such that write-offs will be distinguished from non-write-offs?

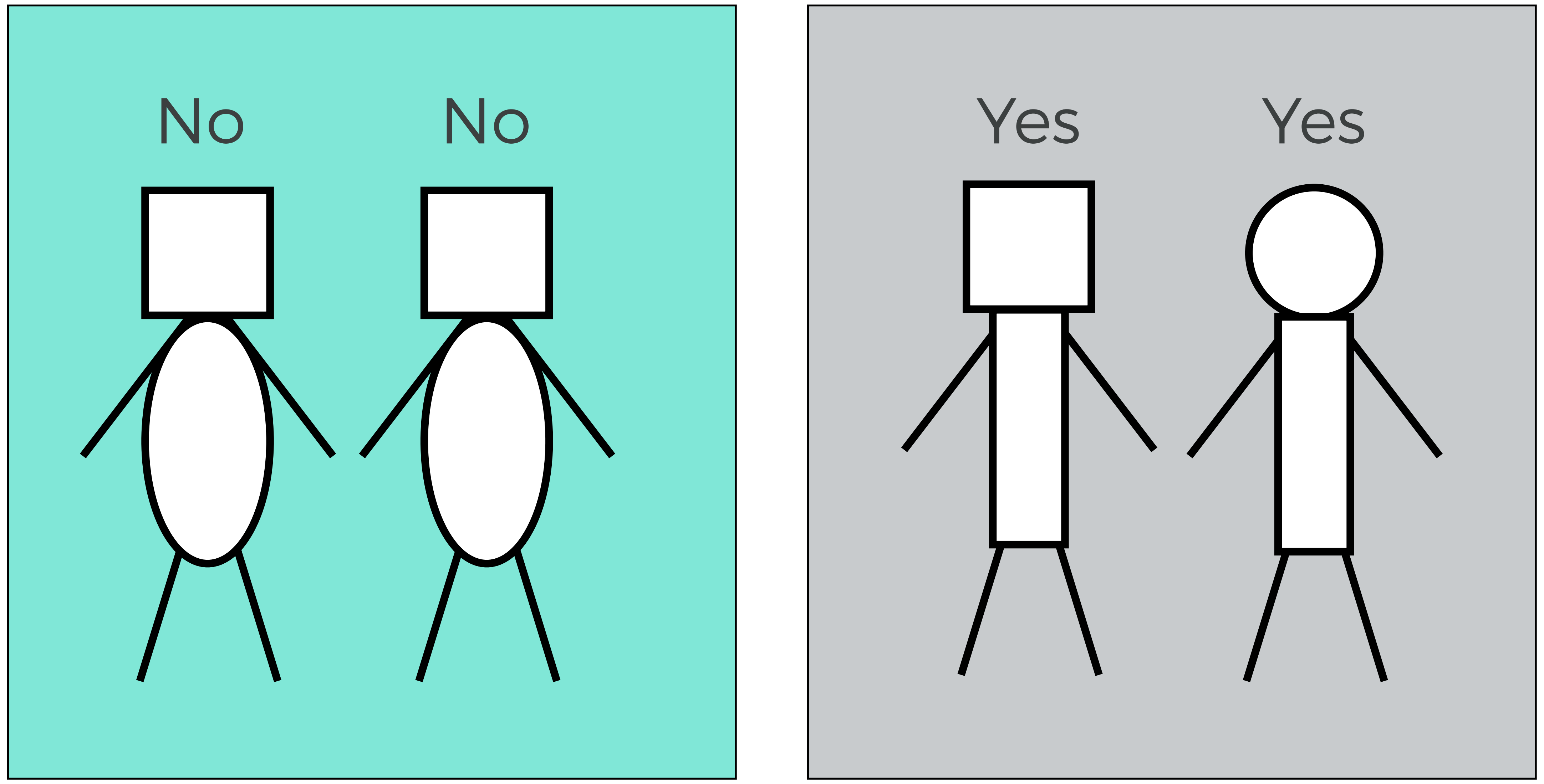
Resulting groups should be as pure as possible!

Group Purity

Assume this was our entire data



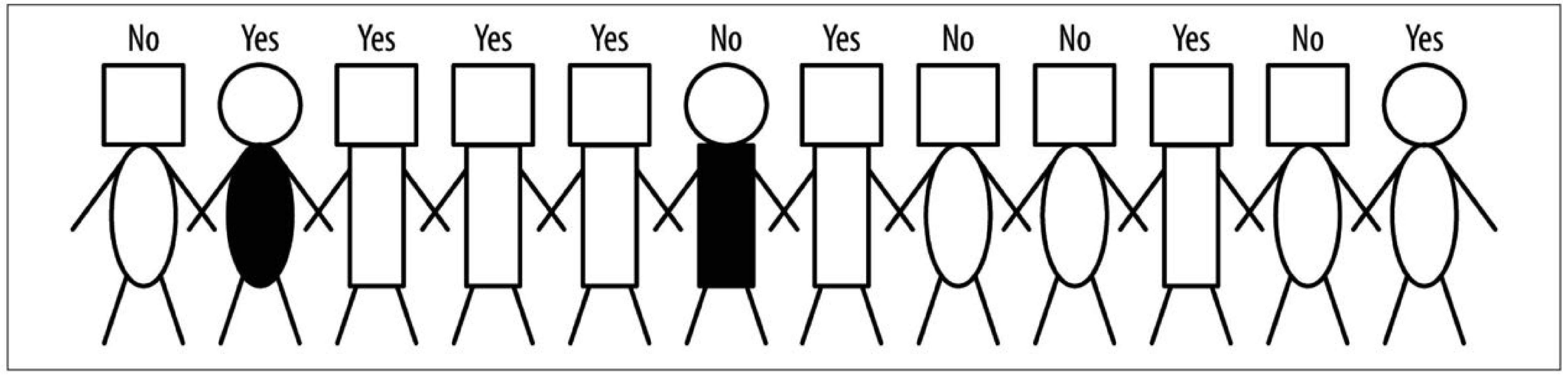
We can obtain **pure** groups splitting by **body-type**



A measure of purity: entropy

entropy = -p1 log(p1) - p2log(p2) - ...

where **p_i** is the relative percentage of property **i** within the set



Entropy measures the **general disorder** of the set, ranging from

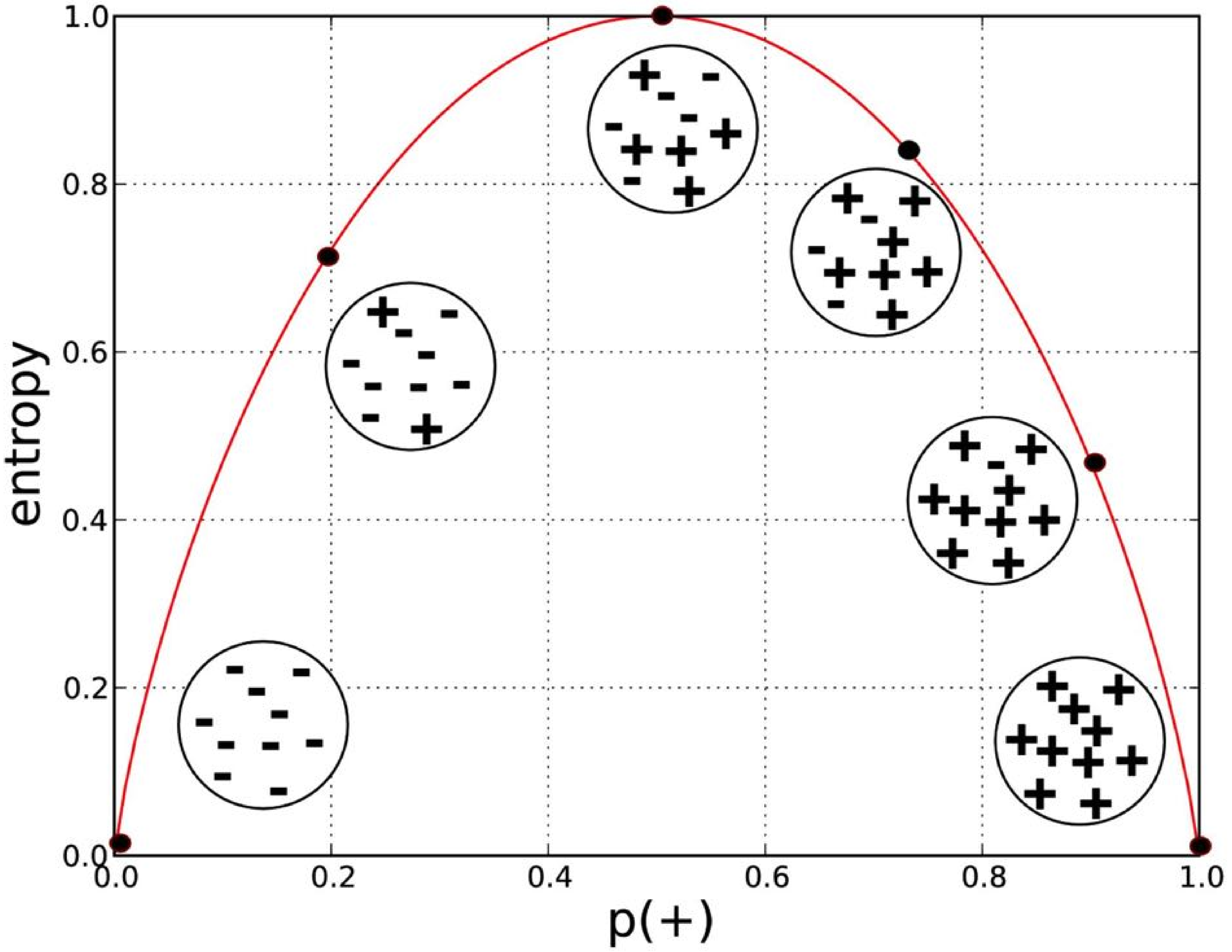
- **p_i = 0** at **minimum disorder** (the set has members all with the same property) to
- **p_i = 1** at **maximal disorder** (the properties are equally mixed)

p(non-write-off) = 7 / 10 = 0.7

p(write-off) = 3 / 10 = 0.3

$$\text{entropy}(S) = -0.7 \times \log_2(0.7) - 0.3 \times \log_2(0.3)$$
$$\approx -0.7 \times -0.51 - 0.3 \times -1.74$$
$$\approx \mathbf{0.88}$$

A measure of purity: entropy



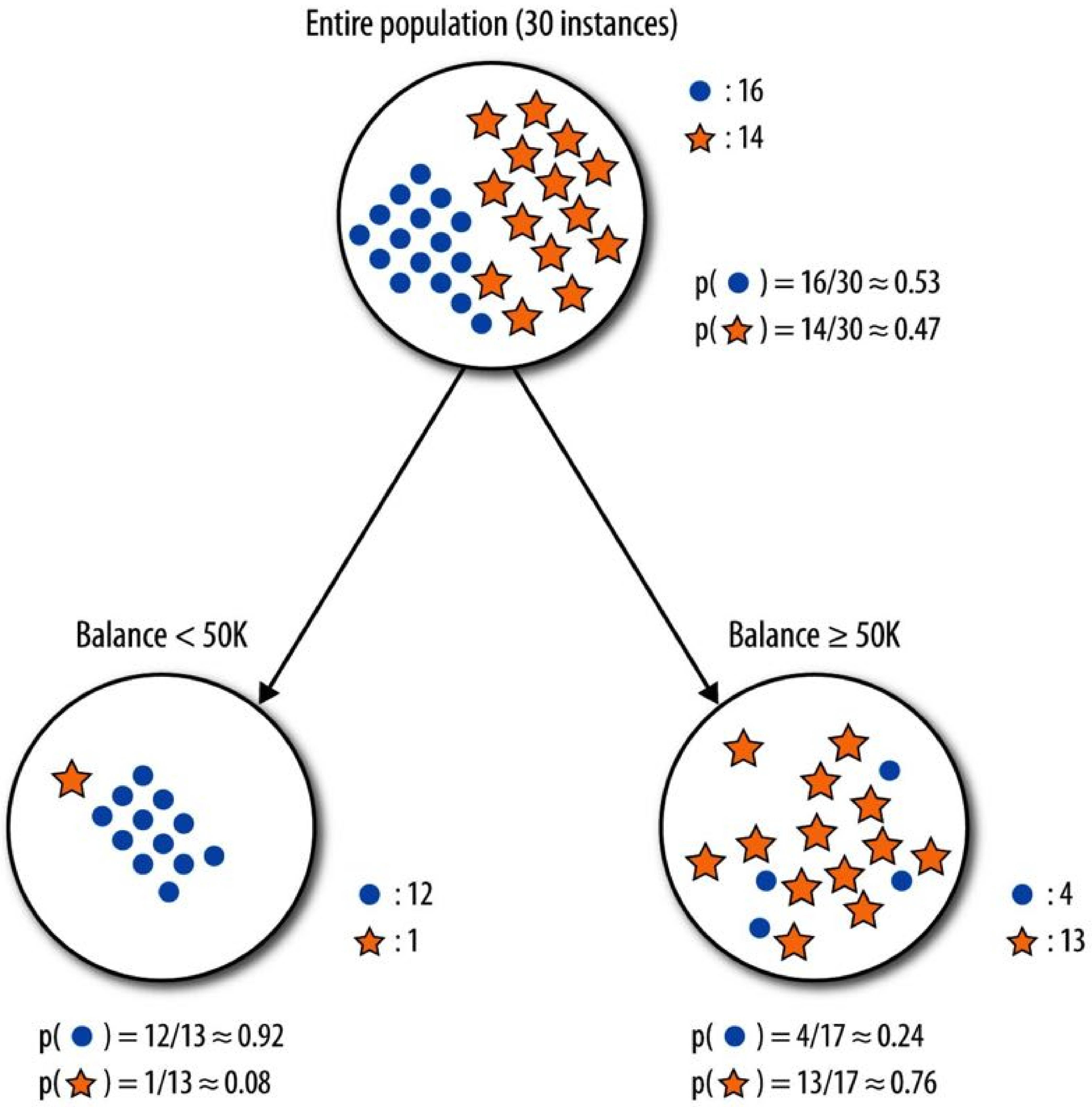
Information gain

- We would like to measure **how informative an attribute is with respect to our target**
    - how much gain in information it gives us about the value of the target variable.
- **Measures the change in entropy due to any amount of new information being added**
- **attribute k values:**
    - original set of examples the **parent set**
    - result of splitting the **k children sets**

Information gain

IG(parent, children) = entropy(parent) -
[p(c_1)*entropy(c_1)+p(c_2)+entropy(c_2)+ …]

where **c_i** is a child derived from the splitting



entropy(parent) ≈ 0.99

entropy(Balance < 50K) ≈ 0.54

entropy(Balance ≥ 50K) ≈ 0.97

IG = entropy(parent)-

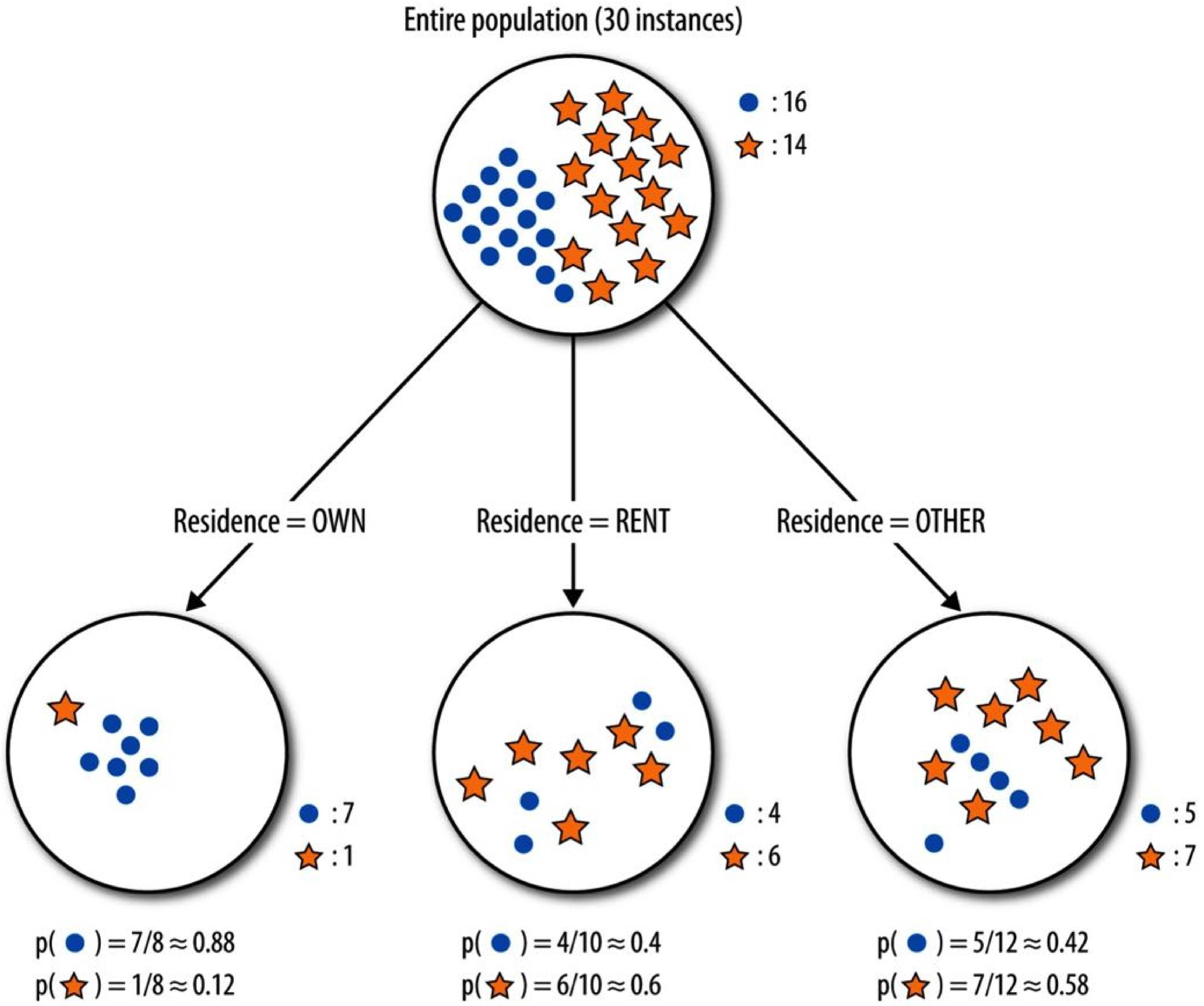
[p(Balance<50K)*entropy(Balance<50K) +

p(Balance≥50)* entropy(Balance ≥ 50K)]

≈ 0.99 - [0.43 × 0.39 + 0.57 × 0.79]

≈ 0.37

the split reduces entropy substantially



entropy(parent) ≈ 0.99

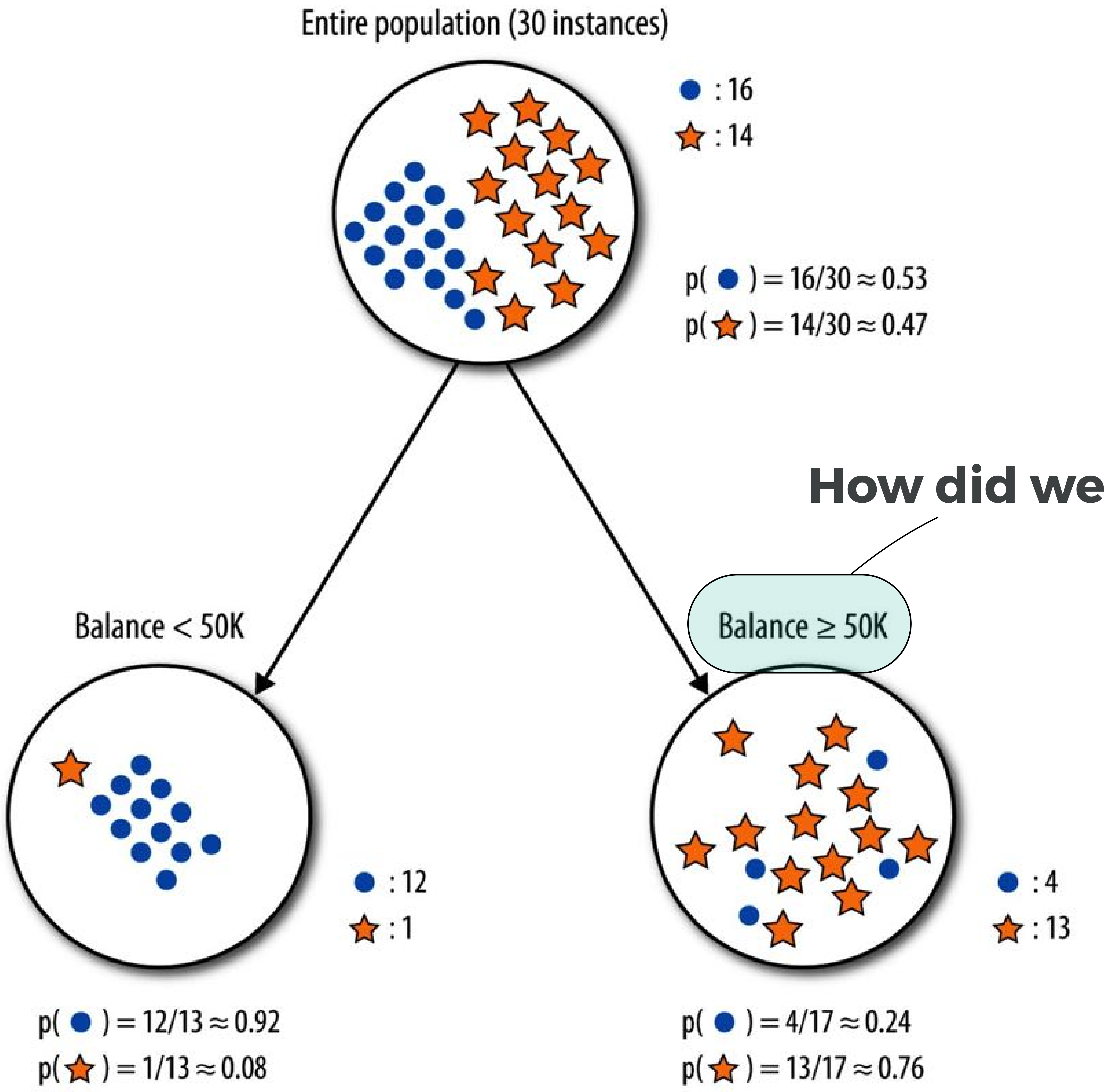
entropy(Residence=OWN) ≈ 0.54

entropy(Residence=RENT) ≈ 0.97

entropy(Residence=OTHER) ≈ 0.98

IG ≈ 0.13

this split (Residence) reduces entropy less than the previous case (Balance)



How did we come up with this?

Discretization

- Equal interval
- Equal frequency
- K-means

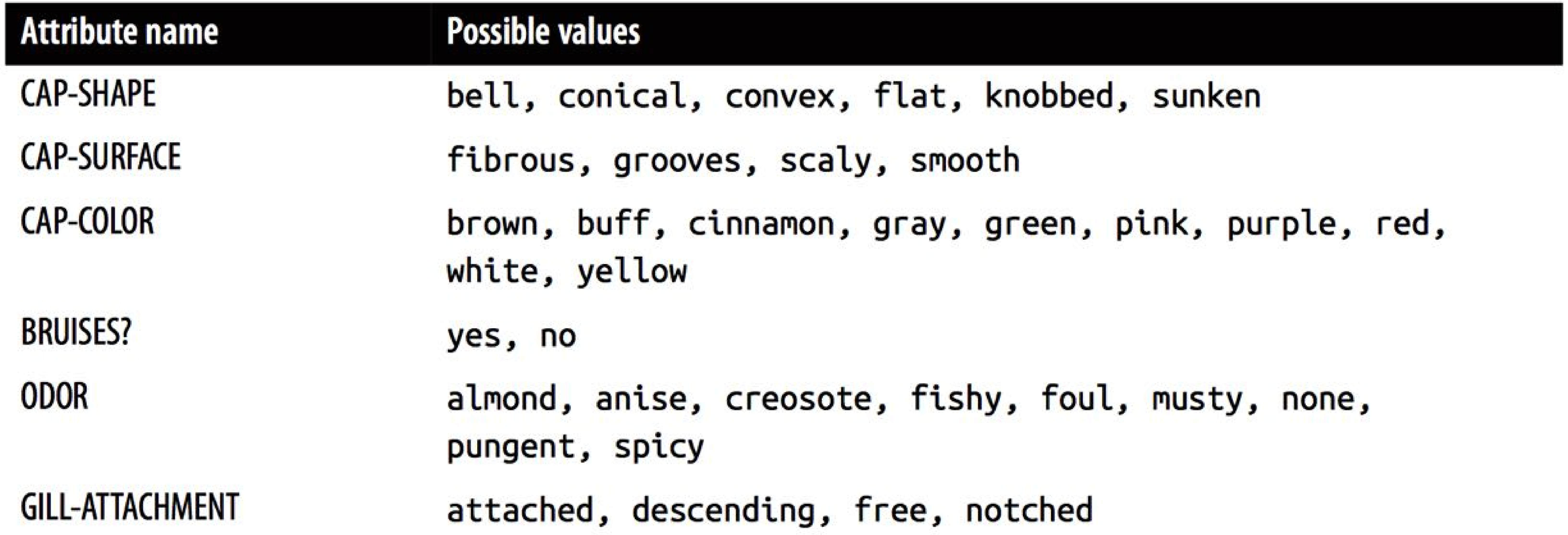
Binary Split that maximizes gain

More info at this link

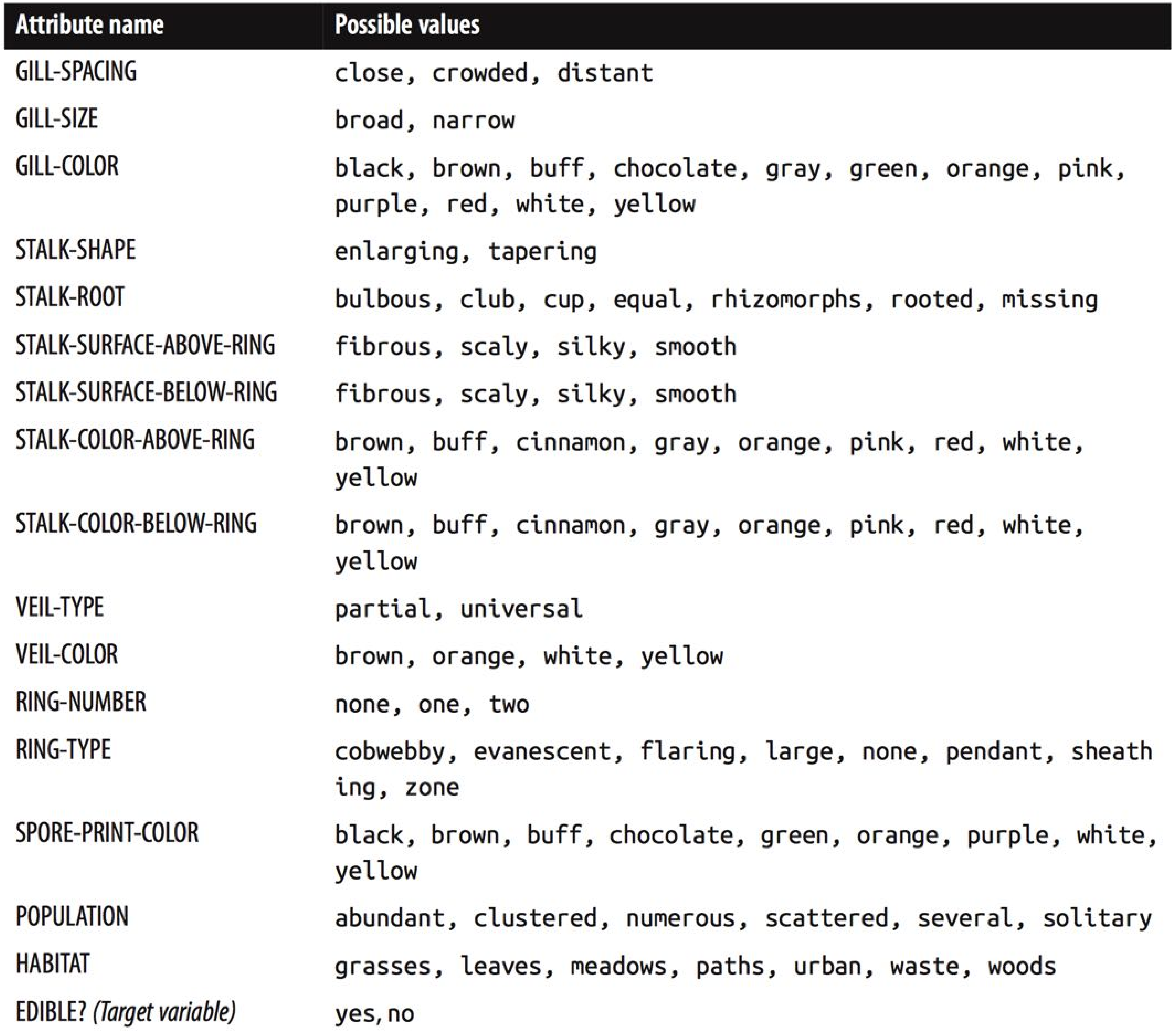
(Data Mining, Prof. Dr. J. Fürnkranz)

Example: Attribute Selection

- The Audubon Society Field Guide to North American Mushrooms
- 5,644 samples (2,156 poisonous, 3,488 edible mushrooms)
- 23 features



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



Attribute name	Possible values
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
STALK-COLOR-ABOVE-RING	brown, buff, cinnamon, gray, orange, pink, red, white, yellow
STALK-COLOR-BELOW-RING	brown, buff, cinnamon, gray, orange, pink, red, white, yellow
VEIL-TYPE	partial, universal
VEIL-COLOR	brown, orange, white, yellow
RING-NUMBER	none, one, two
RING-TYPE	cobwebby, evanescent, flaring, large, none, pendant, sheathing, zone
SPORE-PRINT-COLOR	black, brown, buff, chocolate, green, orange, purple, white, yellow
POPULATION	abundant, clustered, numerous, scattered, several, solitary
HABITAT	grasses, leaves, meadows, paths, urban, waste, woods
EDIBLE? *(Target variable)*	yes, no

You have to advice Christopher McCandless whether to eat a mushroom or not, before he starves.

He has time only for a brief call on the phone, and you can ask about only of the characteristics of the mushroom: **gill-color**, **spore-print-color**, **odor**.

Which one you choose?

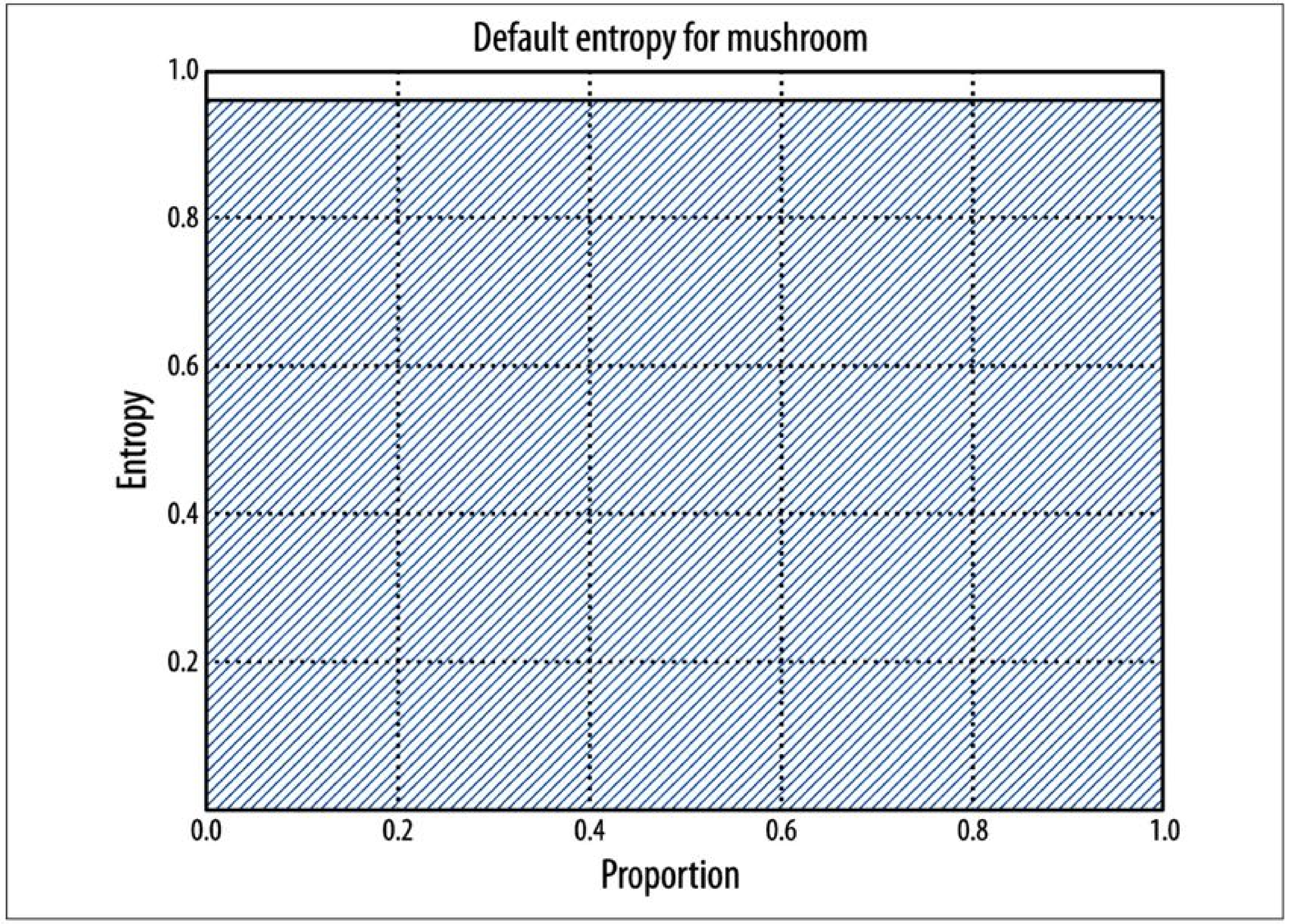


Figure 3-6. Entropy chart for the entire Mushroom dataset. The entropy for the entire dataset is 0.96, so 96% of the area is shaded.

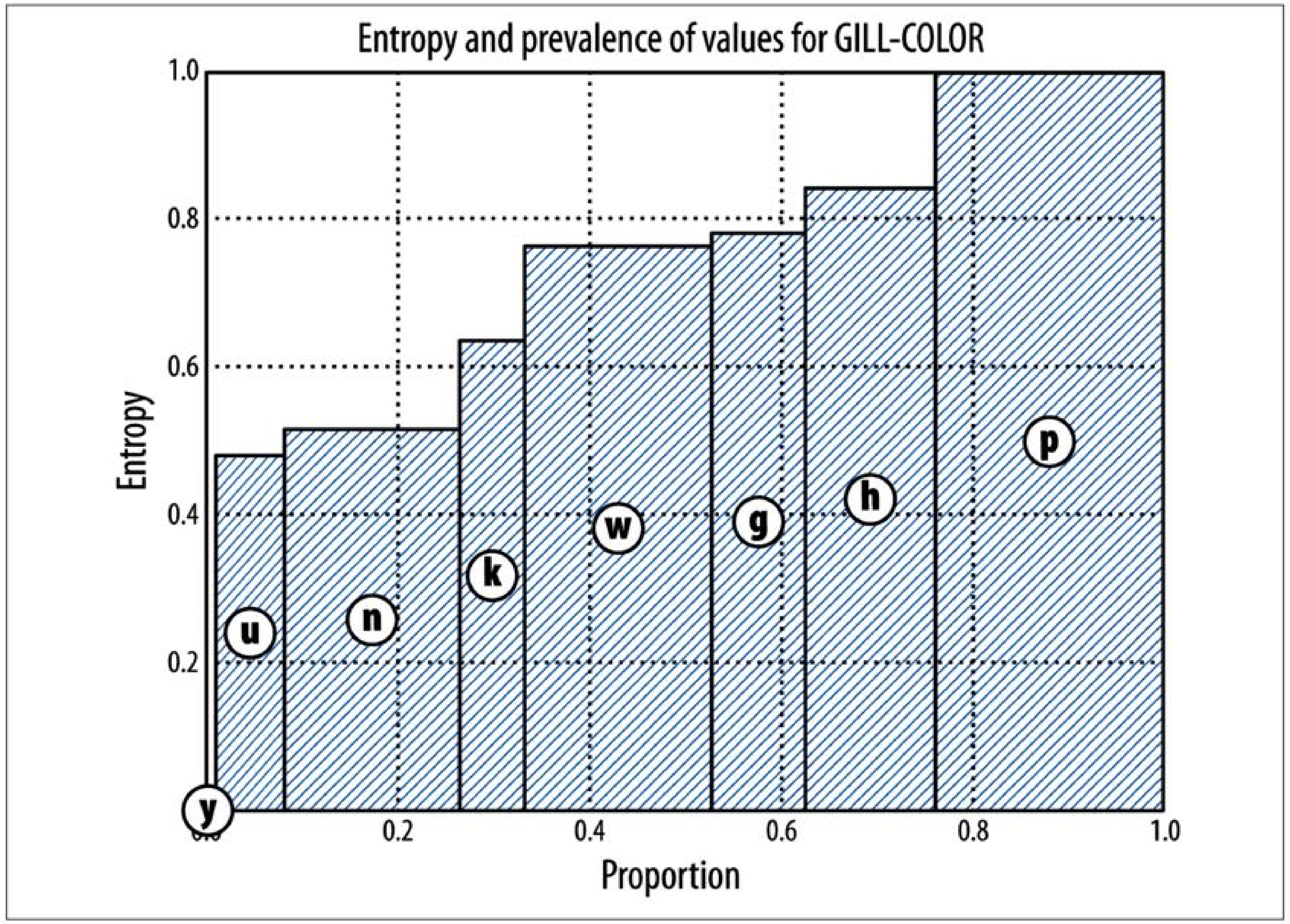


Figure 3-7. Entropy chart for the Mushroom dataset as split by GILL-COLOR. The amount of shading corresponds to the total (weighted sum) entropy, with each bar corresponding to the entropy of one of the attribute's values, and the width of the bar corresponding to the prevalence of that value in the data.

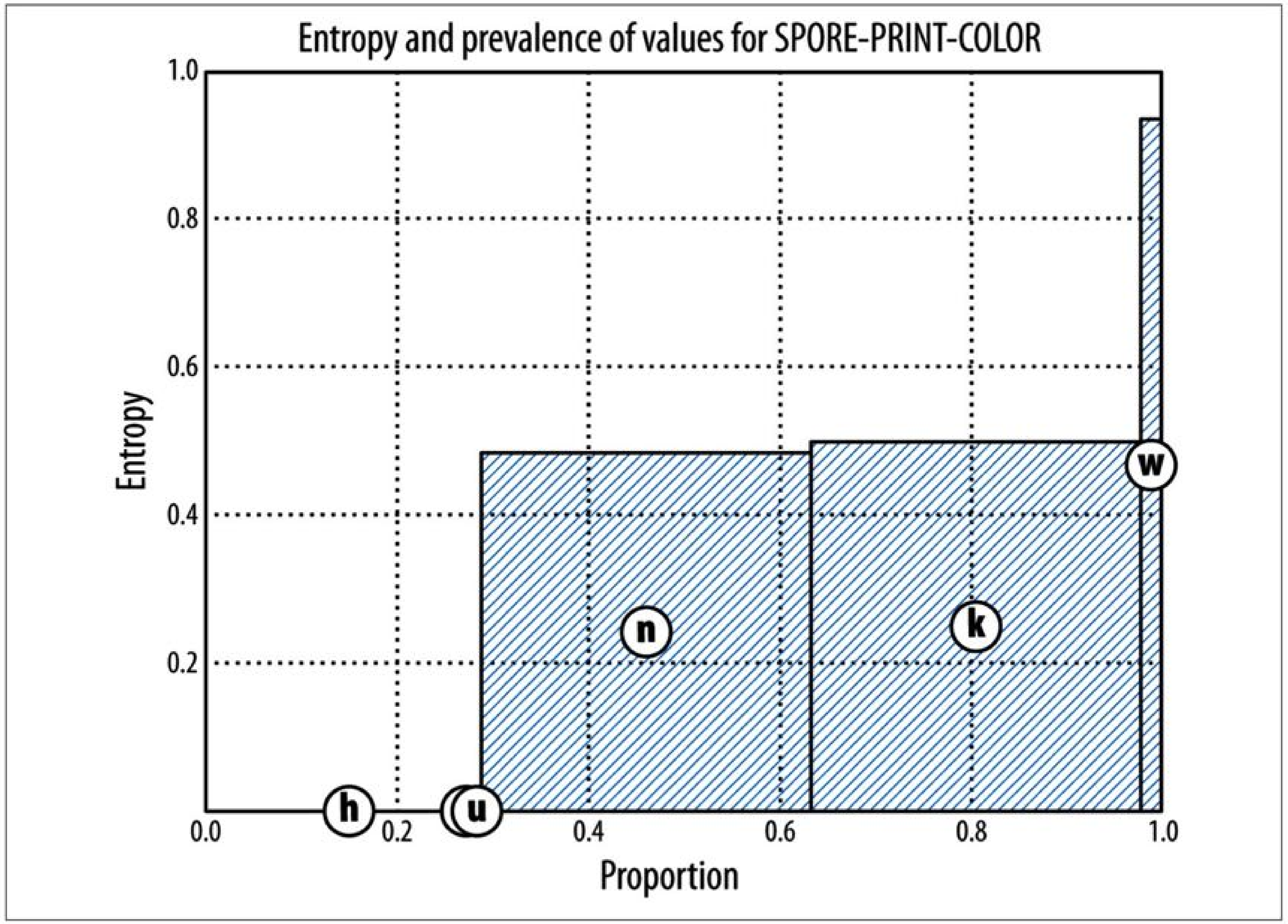


Figure 3-8. Entropy chart for the Mushroom dataset as split by SPORE-PRINT-COLOR. The amount of shading corresponds to the total (weighted sum) entropy, with each bar corresponding to the entropy of one of the attribute's values, and the width of the bar corresponding to the prevalence of that value in the data.

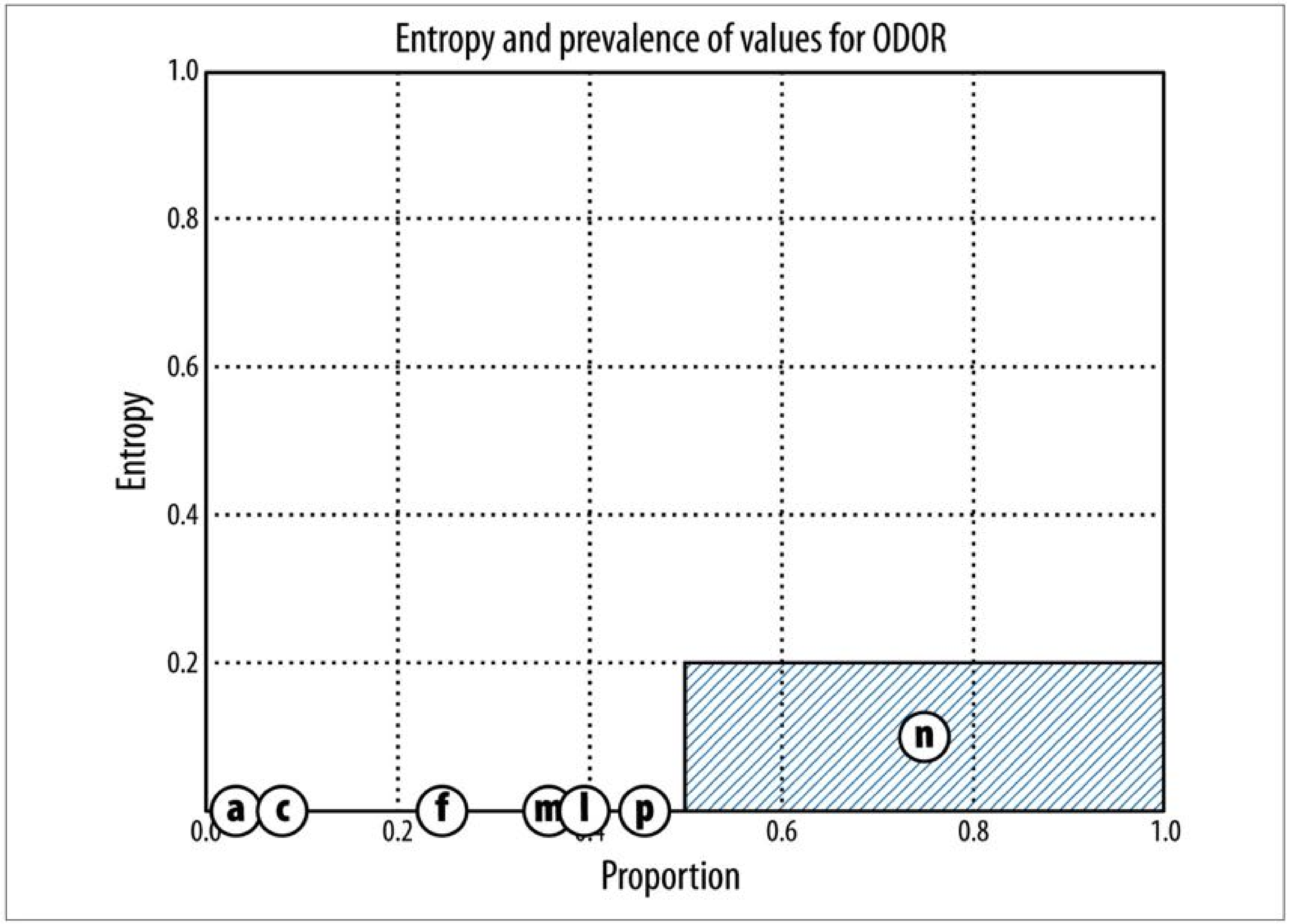


Figure 3-9. Entropy chart for the Mushroom dataset as split by ODOR. The amount of shading corresponds to the total (weighted sum) entropy, with each bar corresponding to the entropy of one of the attribute's values, and the width of the bar corresponding to the prevalence of that value in the data.

Another Measure of Impurity: Gini index

$$1 - \sum_{j} p(j \mid t)^2$$

- where **p(j | t)** is the relative frequency of **class j at node t**
- **Maximum (1 - 1/n$_c$)** when records are equally distributed among all classes, implying least interesting information
- **Minimum (0.0)** when all records belong to one class, implying most interesting information

Gini Index

When a node p is split into **k partitions** (children), the quality of split is computed as:

$$\sum_{i=1}^{k} \frac{n_i}{n} \times GINI(i)$$

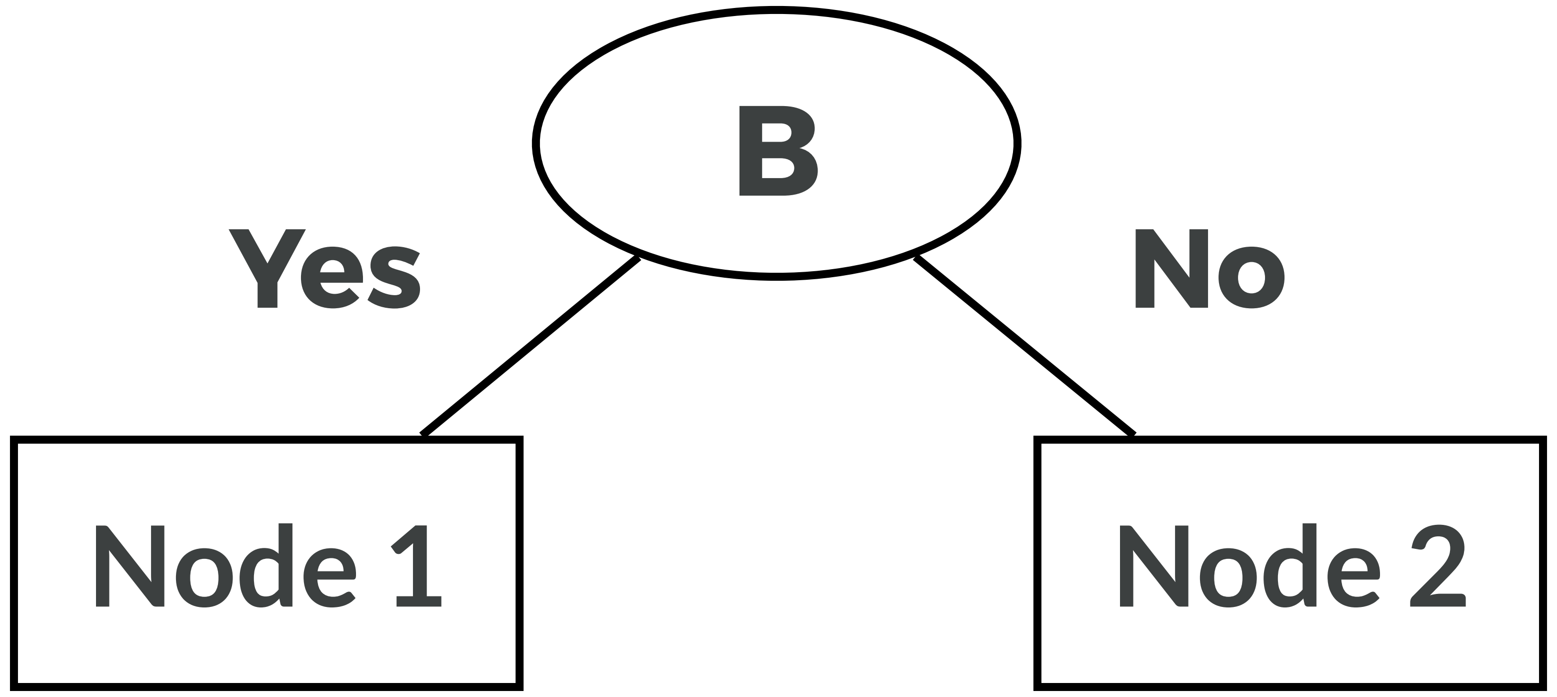
where **n_i = number of records at child i**

n = number of records at node p

Gini Index: example

- Splits into two partitions
  - Effect of weighting partitions:
  - Larger and purer partitions are sought for.

	Parent
C1	6
C2	6
Gini=0.5	



B

Yes

No

Node 1

Node 2

Gini(N1) = 1 – (5/6)2 – (2/6)2
= 0.194
Gini(N2) = 1 – (1/6)2 – (4/6)2
= 0.528

	N1	N2
C1	5	1
C2	2	4
Gini=0.333		

Gini(Children)
= 7/12 * 0.194 + 5/12 * 0.528
= 0.333

Splitting criteria based on classification error

- **Classification error at a node t :**

$$\mathrm{Error}(t) = 1 - \max P(i|t)$$

- **Measures misclassification error made by a node**.
  - **Maximum ($1-1/n_c$)** when records are equally distributed among all classes, implying least interesting information
  - **Minimum (0.0)** when all records belong to one class, implying most interesting information

Splitting criteria based on classification error

$\text{Error}(t) = 1 - \max P(i|t)$

C1	**0**
C2	**6**

P(C1) = 0/6 = 0   P(C2) = 6/6 = 1
Error = 1 – max (0, 1) = 1 – 1 = 0

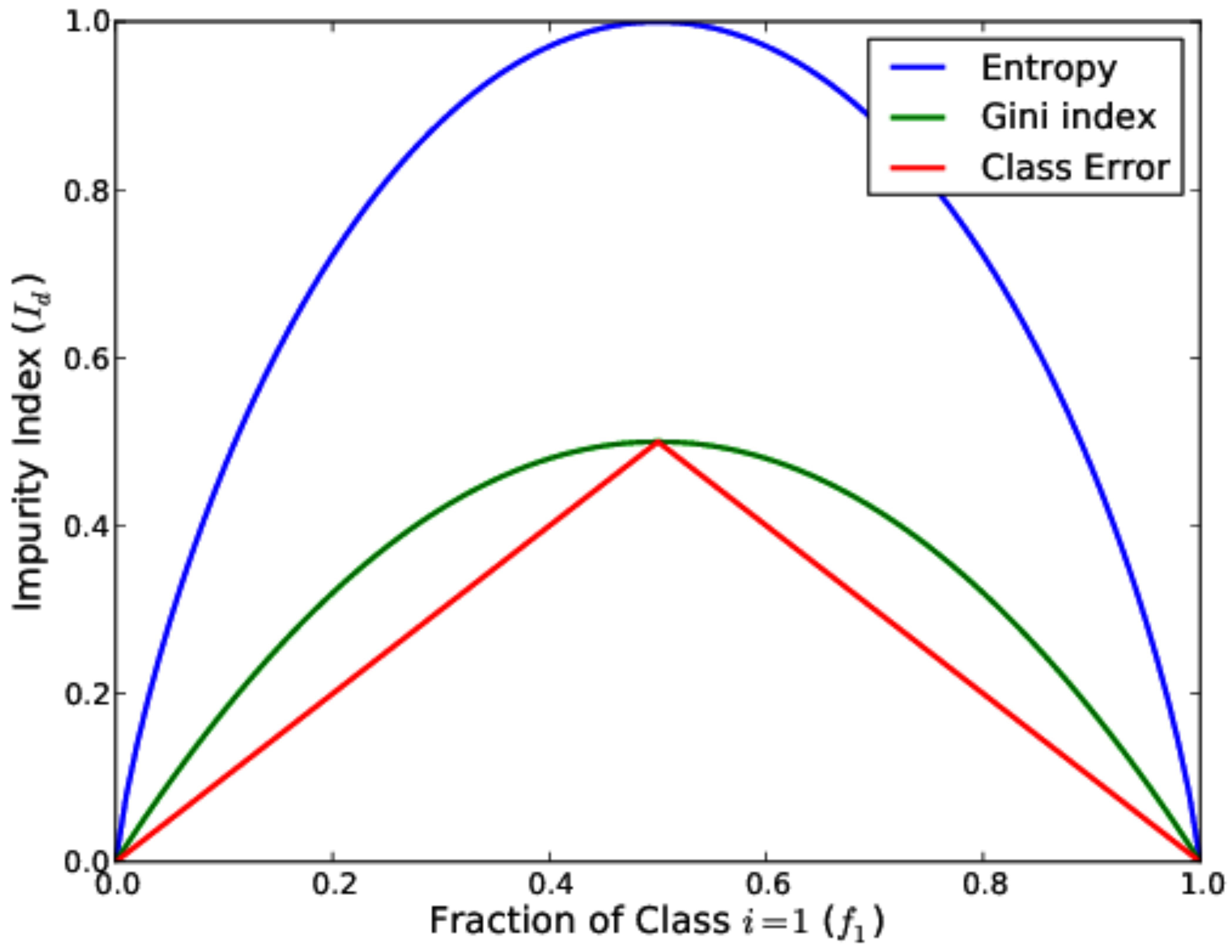
C1	**1**
C2	**5**

P(C1) = 1/6   P(C2) = 5/6
Error = 1 – max (1/6, 5/6) = 1 – 5/6 = 1/6

C1	**2**
C2	**4**

P(C1) = 2/6   P(C2) = 4/6
Error = 1 – max (2/6, 4/6) = 1 – 4/6 = 1/3

For a 2-class problem

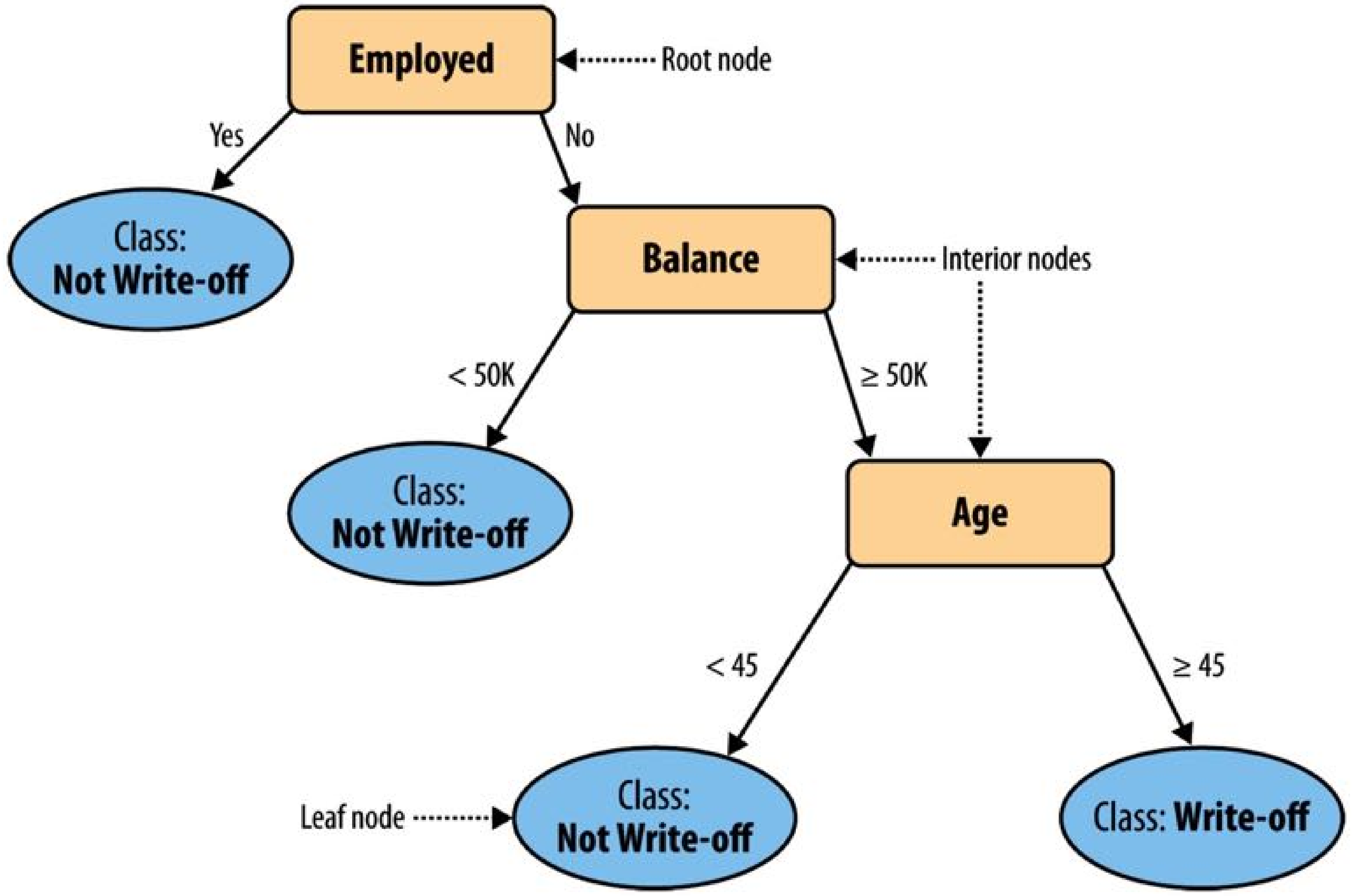


Decision Tree

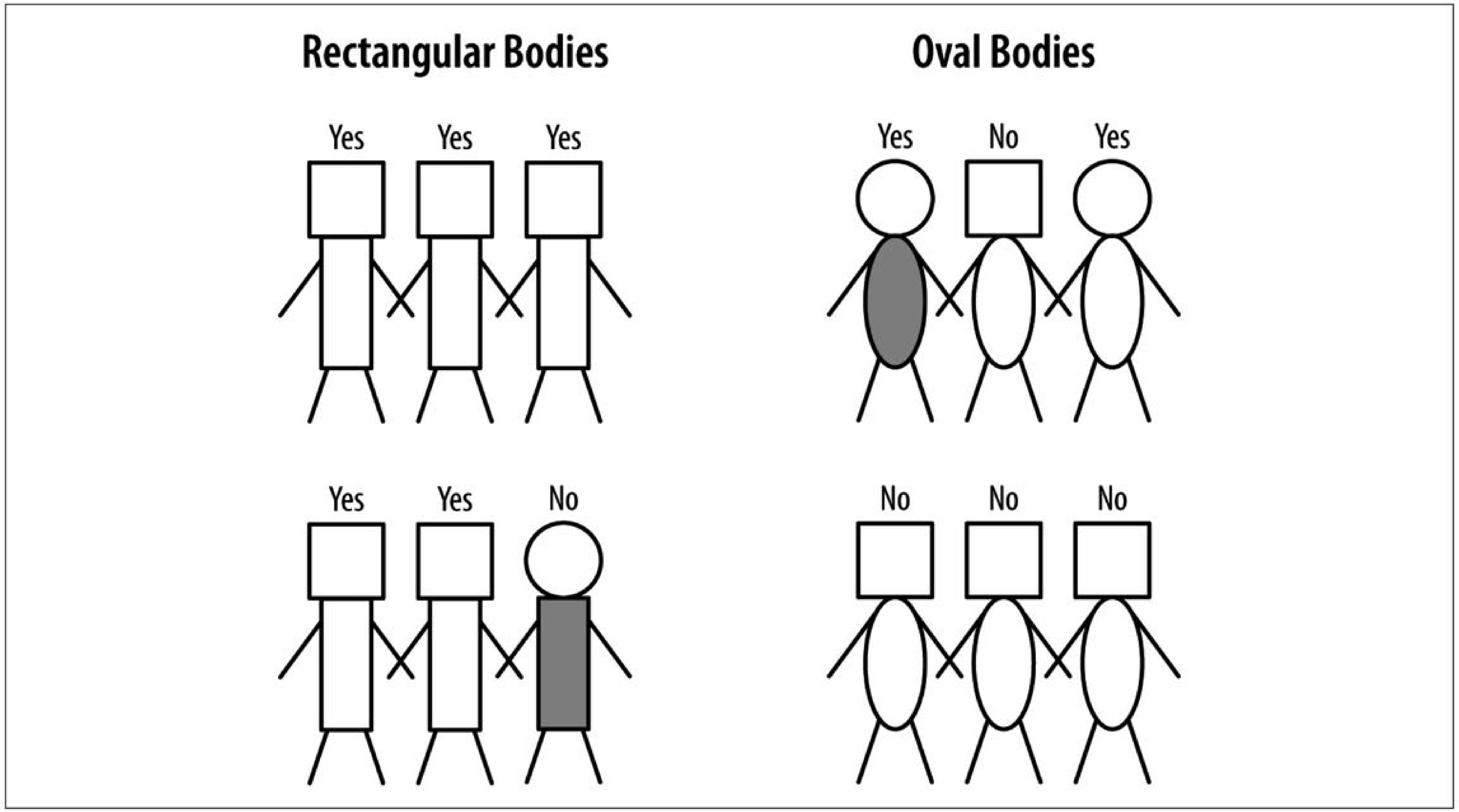
  - A flow-chart-like **tree structure**
  - **Internal node denotes a test on an attribute**
  - **Branch represents an outcome of the test**
  - **Leaf nodes represent class labels or class distribution**
- Decision tree generation consists of **two phases**
  - **Tree construction**
    - At start, 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

Decision Tree

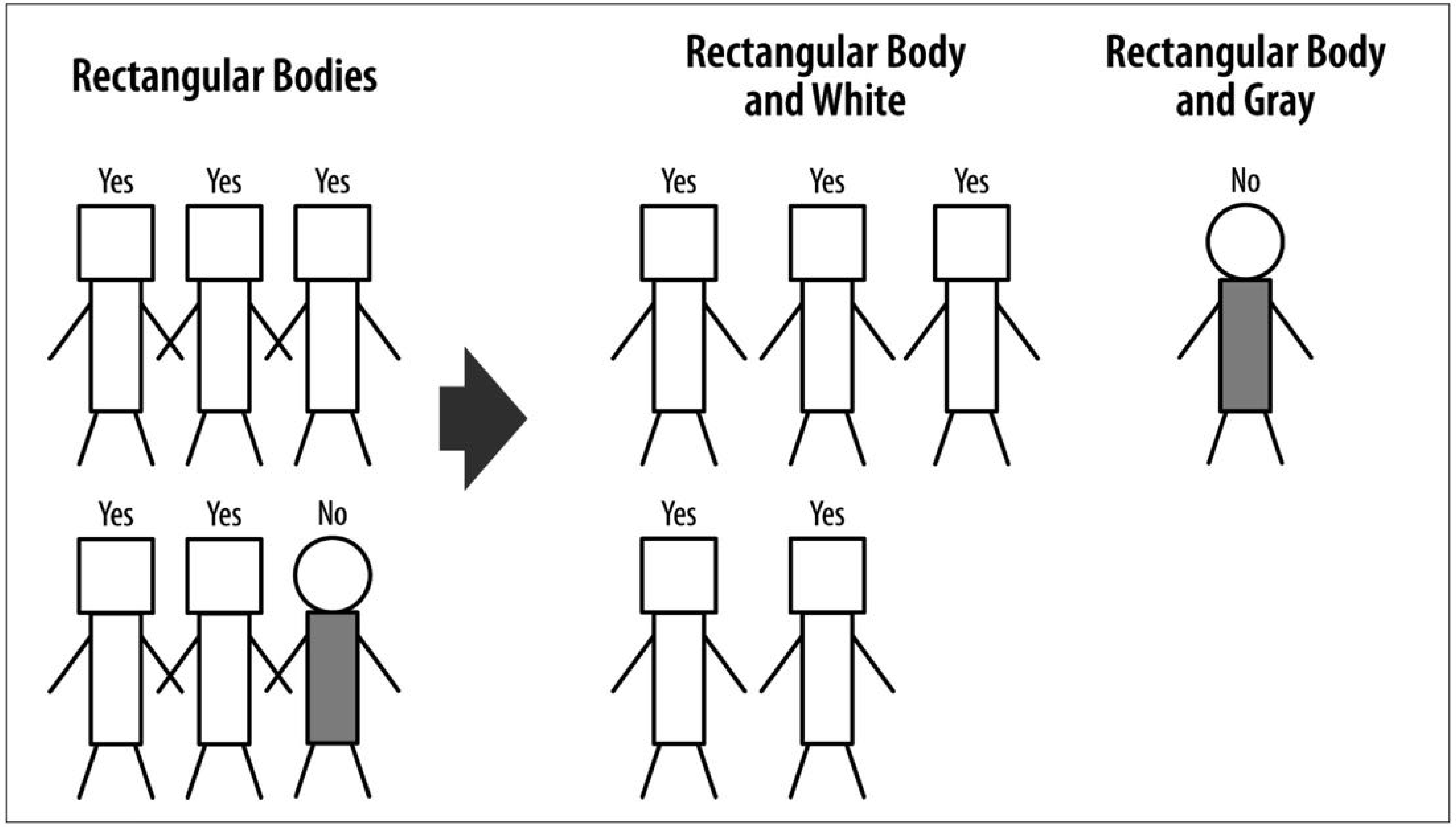
- **Classifying an unknown sample**
  - Test the attribute values of the sample against the decision tree



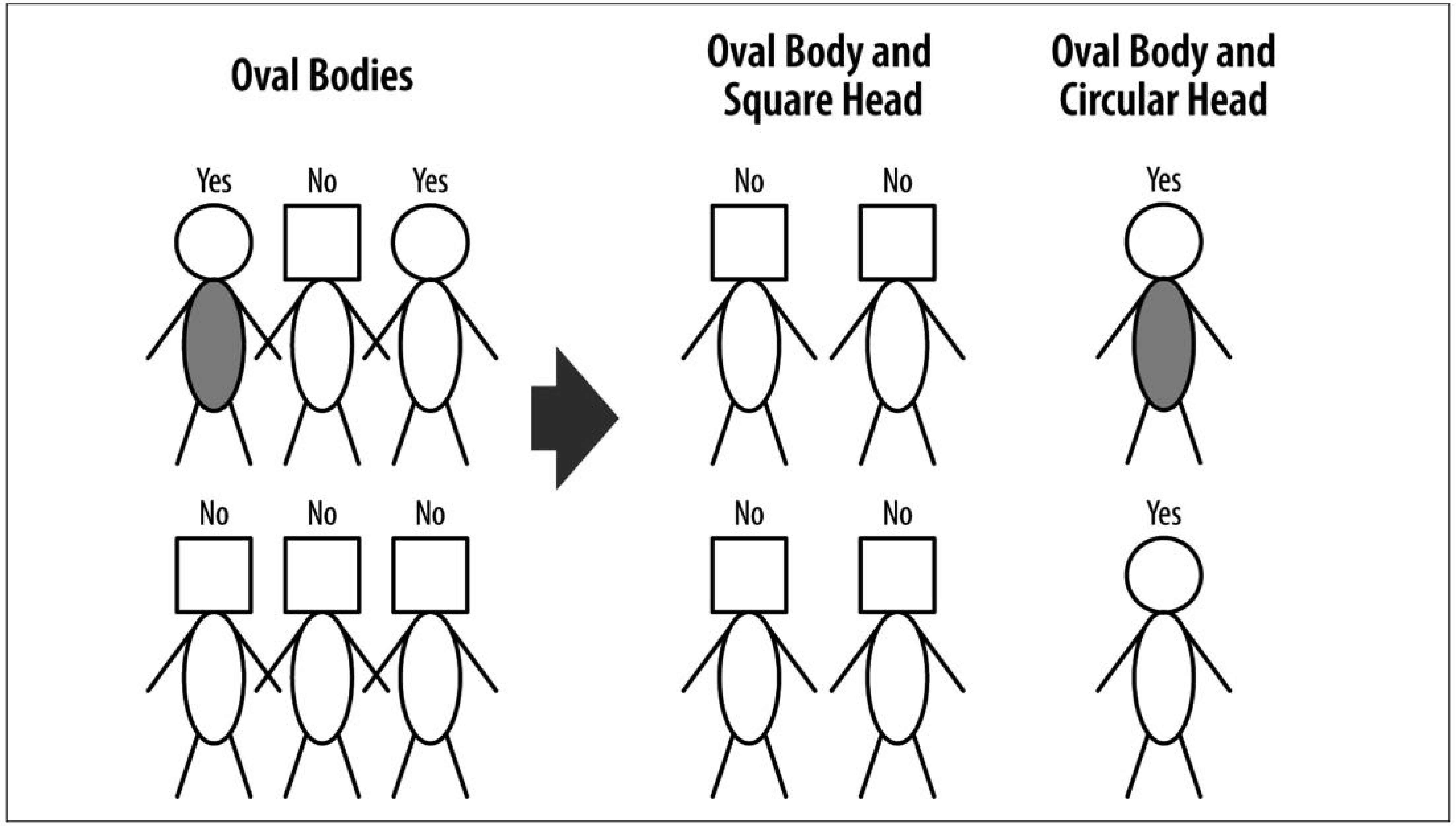
Tree Induction

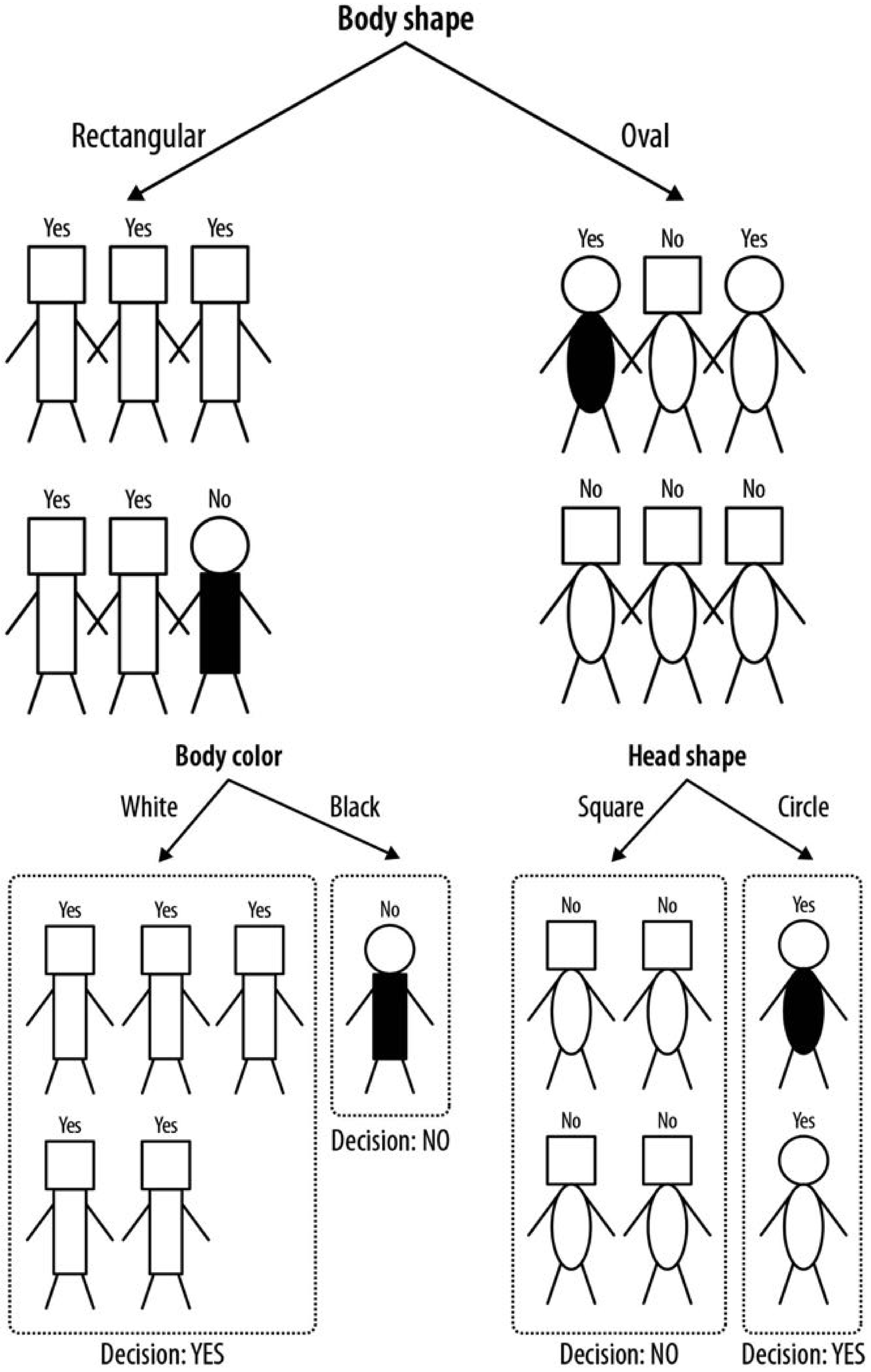


Tree Induction



Tree Induction





Very rare situation in which all the leaves are pure!

Tree Induction

- **Exponentially** many decision trees can be constructed from a given set of attributes
- Finding the most accurate tree is **NP-hard**
- In practice: **greedy algorithms**
    - Grow a decision tree by making a series of **locally optimum decisions** on which attributes to use for partitioning the data

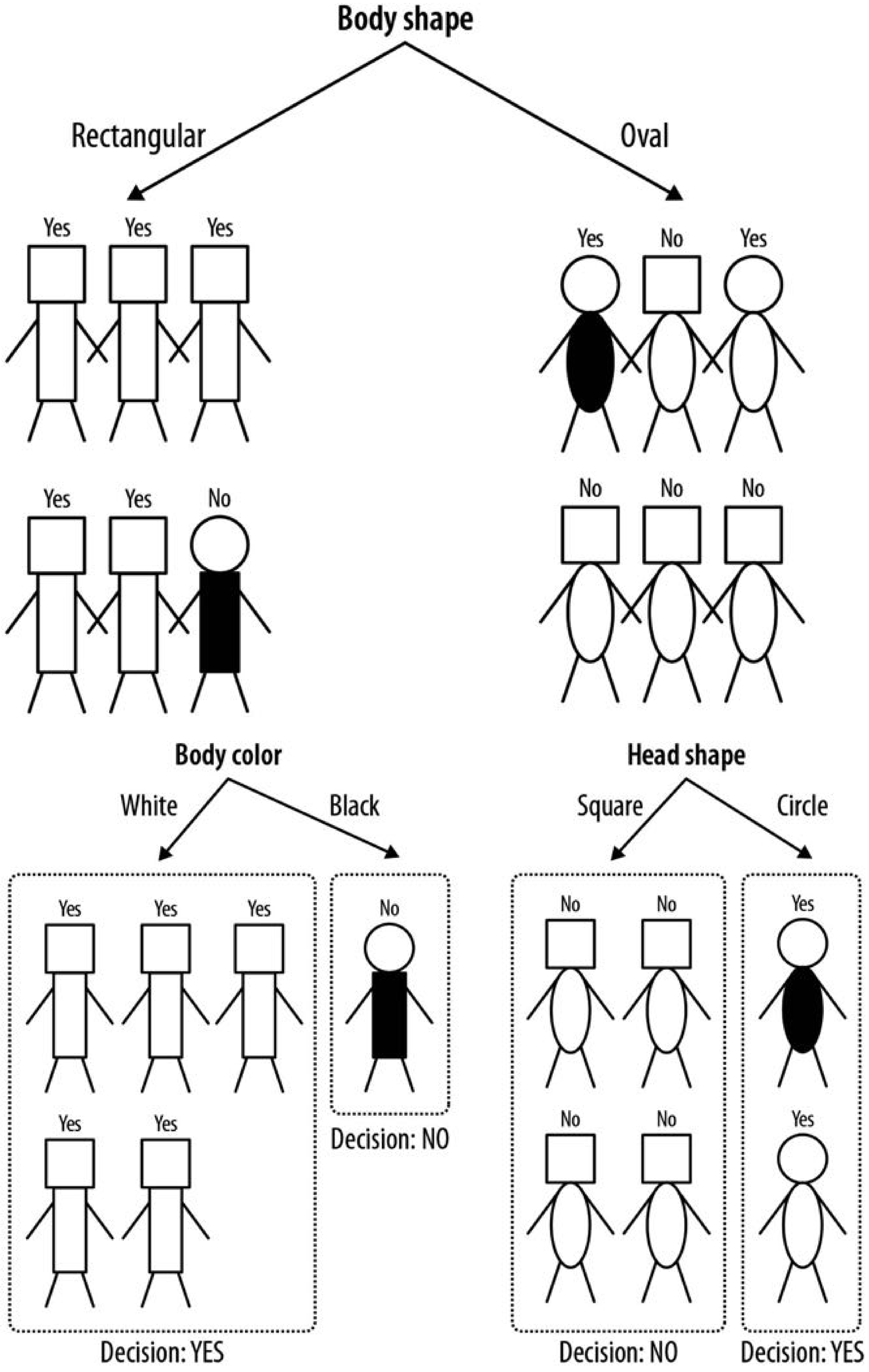
Tree induction as a set of rules

IF (Body shape=Rectangular) AND (Body Color=White) THEN **Class=YES**

IF (Body shape=Rectangular) AND (Body Color=Black) THEN **Class=NO**

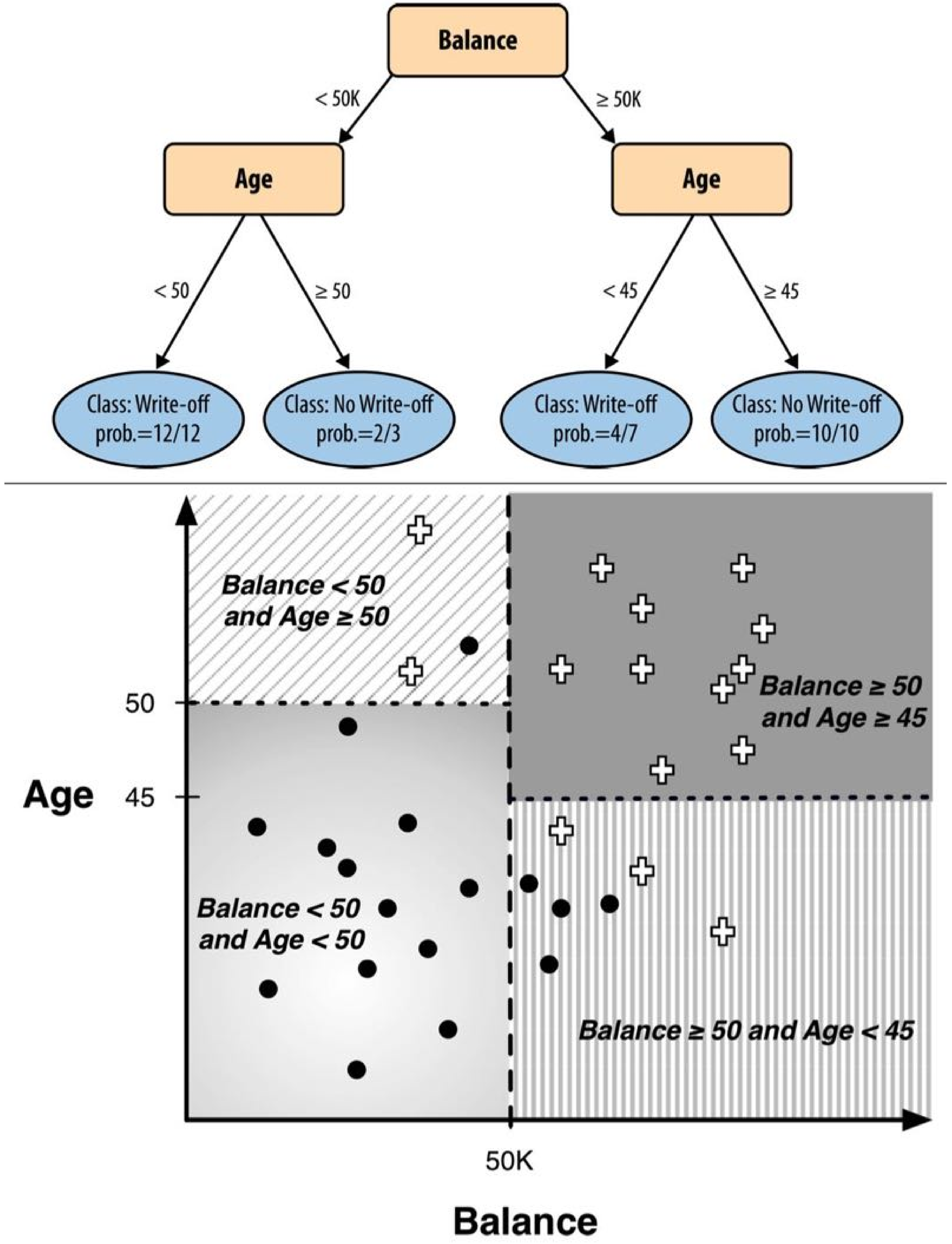
IF (Body shape=Oval) AND (Head Shape=Square) THEN **Class=NO**

IF (Body shape=Ovale) AND (Head Shape=Circle) THEN **Class=YES**



Probability Estimation

- Assign to each leaf **an estimate of the probability of membership** in the different classes
- Tree induction can easily produce probability estimation trees instead of simple classification trees
- **Frequency-based** estimate of class membership: if a leaf contains **n** positive and **m** negative instances, the probability of any new instance being positive may be estimated as **n/(n+m)**.

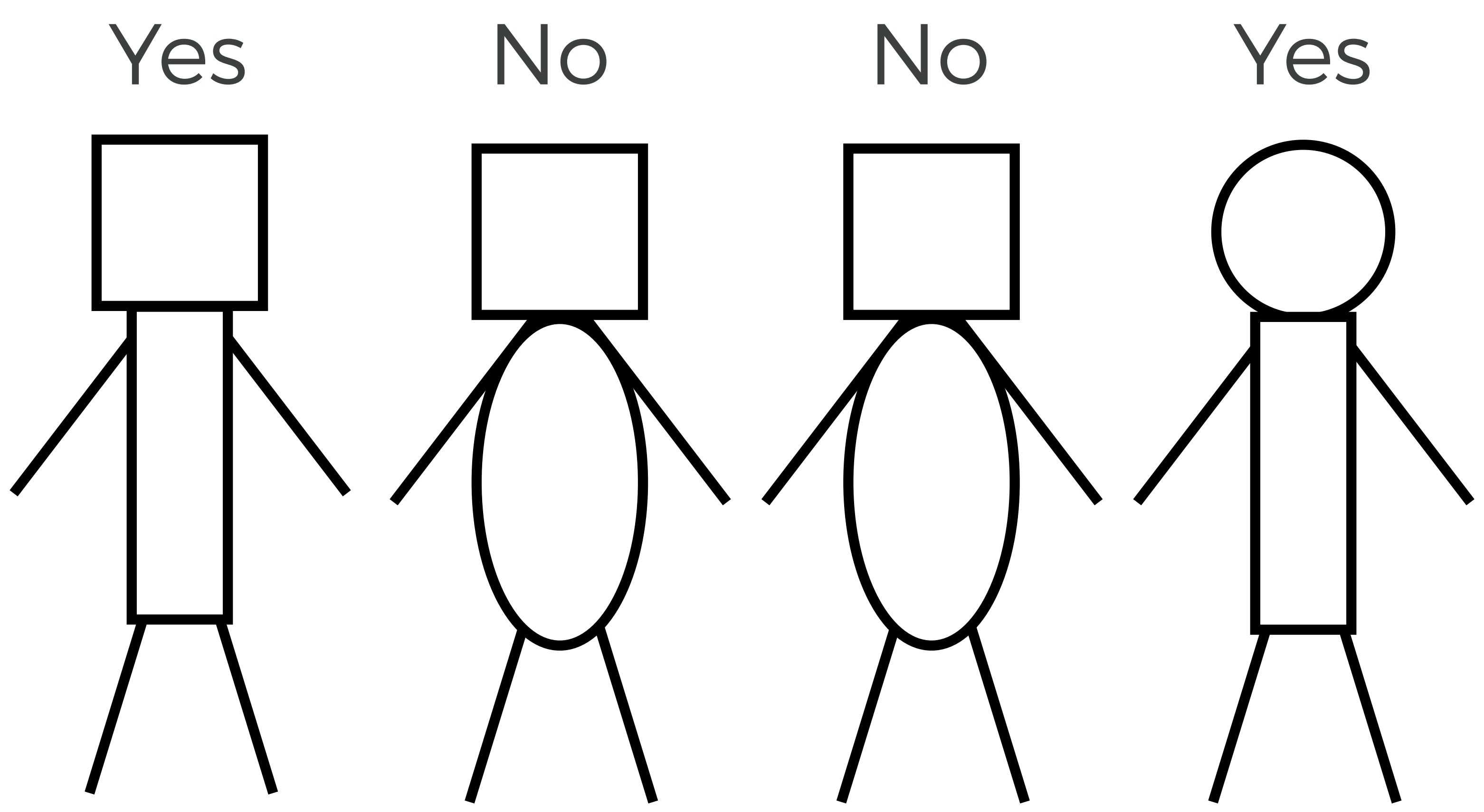

Balance
< 50K
≥ 50K
Age
Age
< 50
≥ 50
< 45
≥ 45
Class: Write-off prob.=12/12
Class: No Write-off prob.=2/3
Class: Write-off prob.=4/7
Class: No Write-off prob.=10/10
Balance < 50 and Age ≥ 50
Balance ≥ 50 and Age ≥ 45
Balance < 50 and Age < 50
Balance ≥ 50 and Age < 45
50
45
Age
50K
Balance

+ **write-off**

o **No write-off**

Probability Estimation: Issues

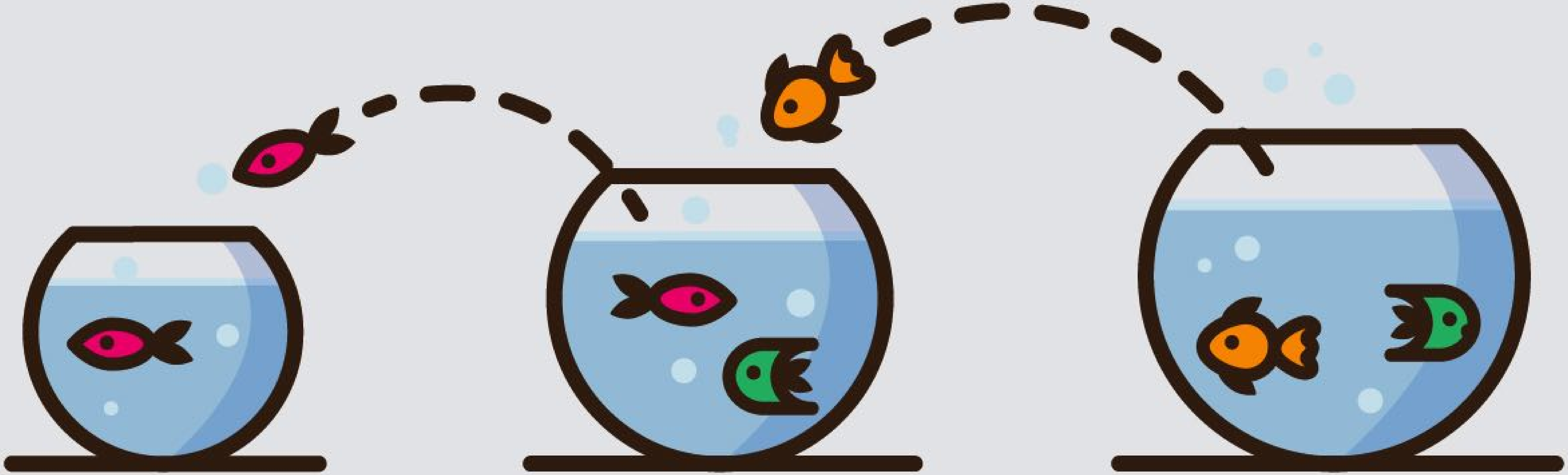
- Attention to the probability of class membership for segments **with very small numbers of instances**.



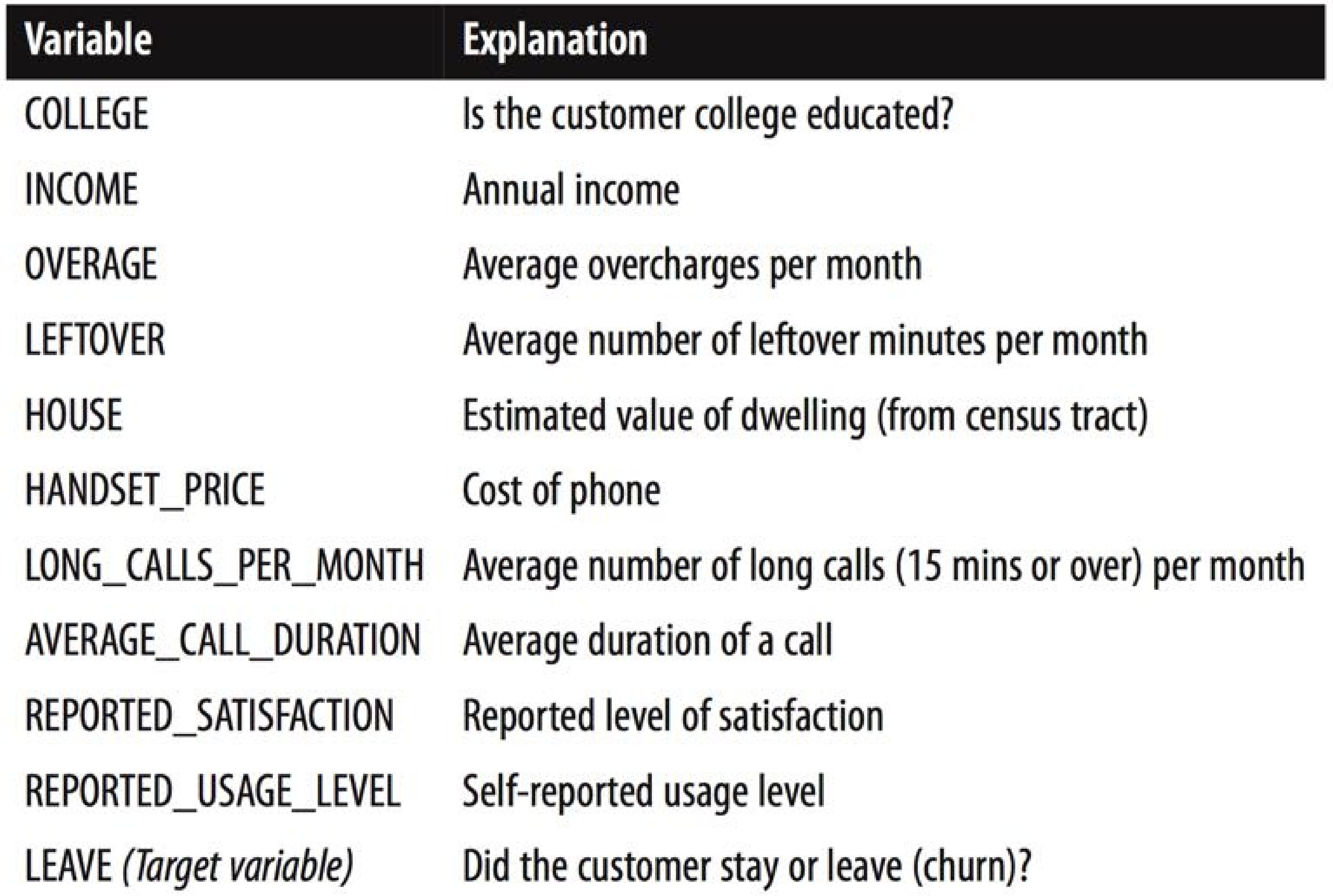
- At the extreme, if a leaf happens to have only a single instance, should we be willing to say that there is a 100% probability that members of that segment will have the class that this one instance happens to have

Churn Problem in Practice

With Decision Trees



Churn Problem: Features



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)?

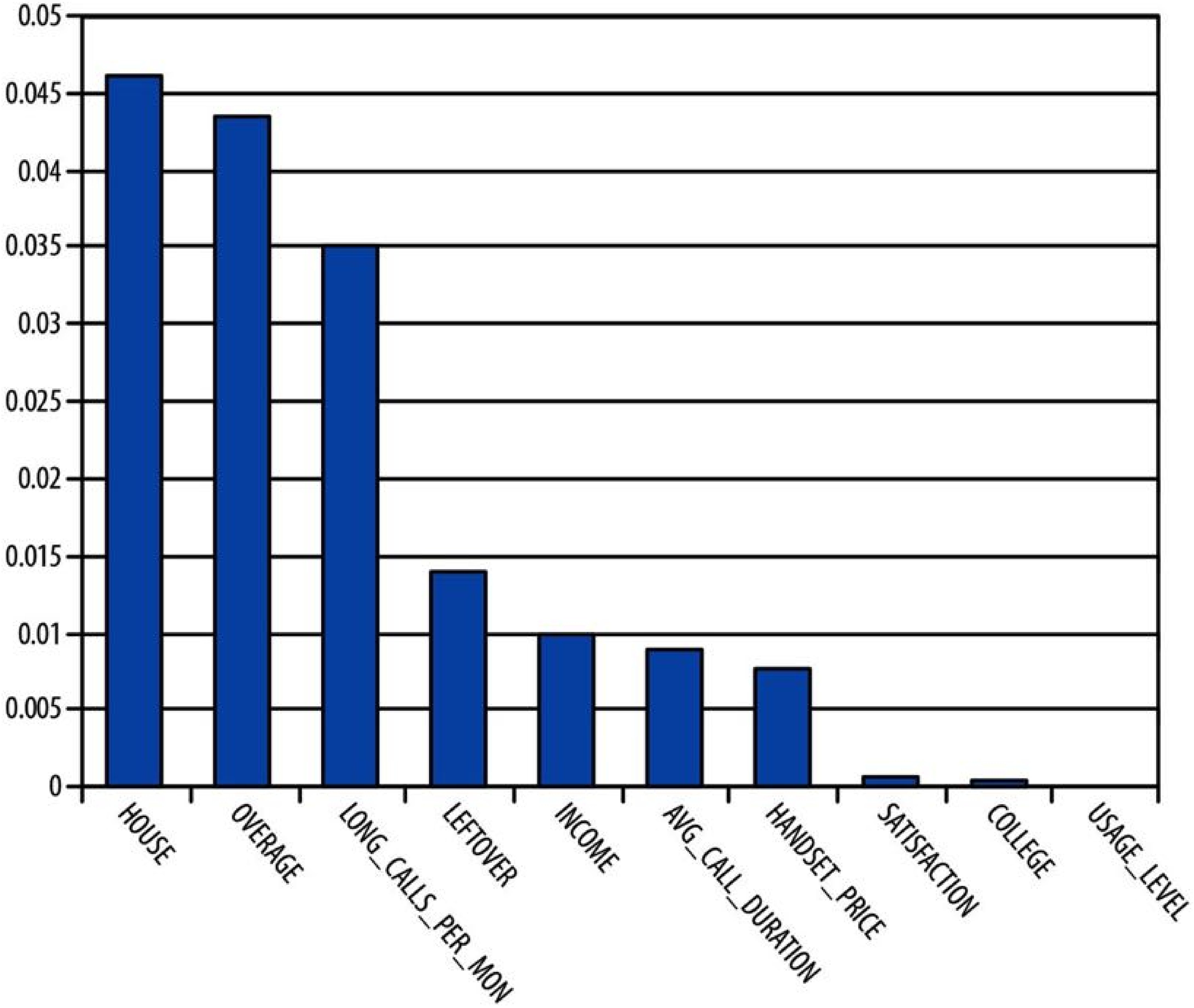
Dataset: 20.000 samples

Information Gain

How good are each of these feature individually?

Which is the root node?

House



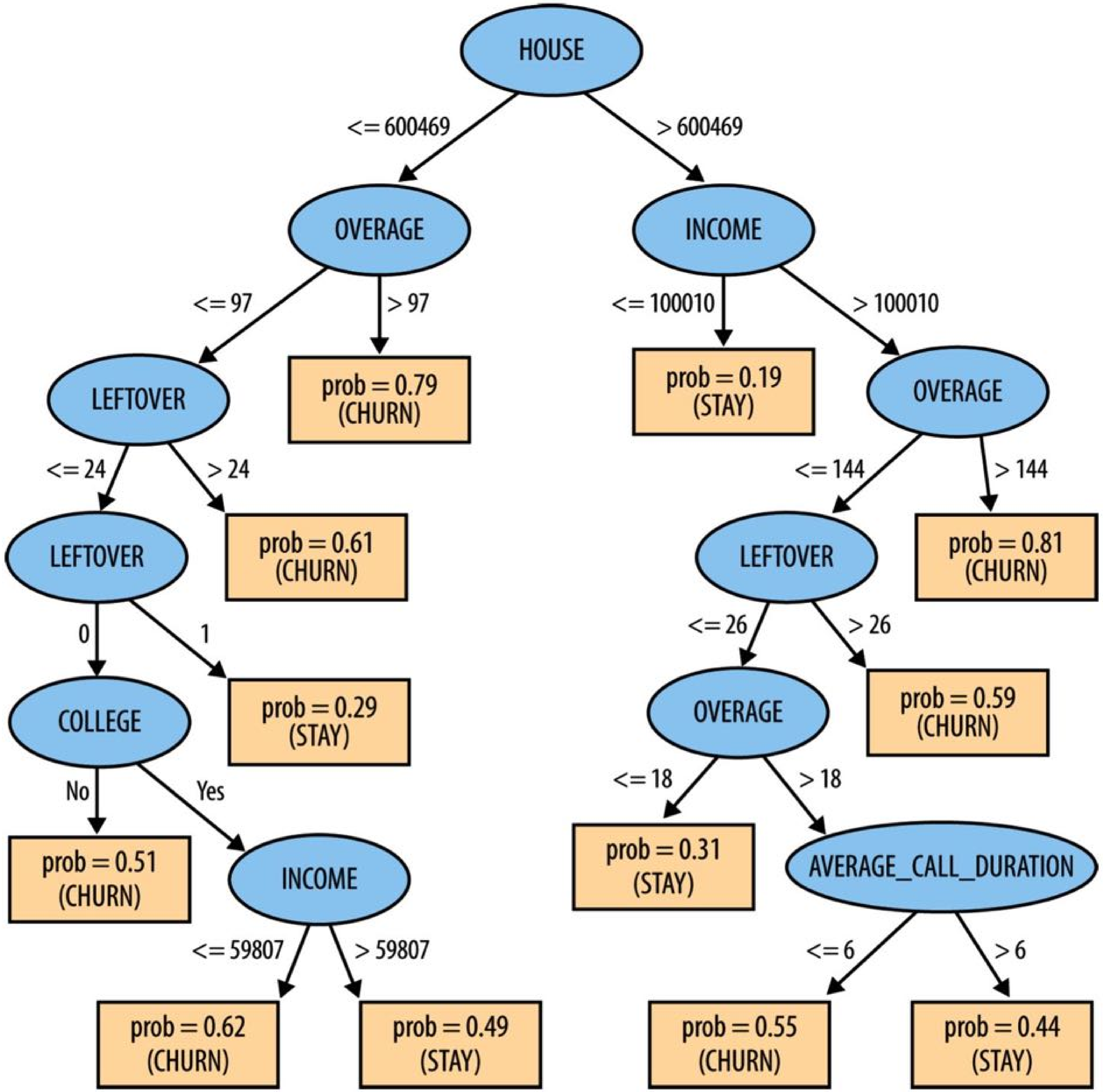
Rank	Info. gain	Attribute name
1	0.0461	HOUSE
2	0.0436	OVERAGE
3	0.0350	LONG_CALLS_PER_MON
4	0.0136	LEFTOVER
5	0.0101	INCOME
6	0.0089	AVG_CALL_DURATION
7	0.0076	HANDSET_PRICE
8	0.0003	SATISFACTION
9	0.000	COLLEGE
10	0.000	USAGE_LEVEL

Tree Induction

The **order** in which features are chosen for the tree **doesn’t exactly correspond to their ranking**.

Why is this?

The ranking is **global**, at each step of creation of the tree, the information gain is estimated locally.



Advantages of Decision Trees

- **Inexpensive** to construct
- Requires **no prior assumptions**
- **Extremely fast at classifying** unknown records
- **Easy to interpret** for small-sized trees
- Tree is **easy to visualize**
- **Accuracy is comparable** to other classification techniques for many **simple** data sets

Disadvantages of Decision Trees

- **Many simplifications** introduced to make computation feasible (stopping conditions, pruning)
- Can be very **non-robust**.
  - A small change in the training data can result in a big change in the tree, and thus a big change in final predictions.
- High risk of **overfitting**
- Subject to statistical errors

Questions?

@rschifan

schifane@di.unito.it

http://www.di.unito.it/~schifane