Decision trees

DSTA

Decision trees

- A simple-yet-effective classification algorithm: CART
- it introduces us to predictive modeling

Study Plan

Ch. 3 of Provost-Fawcett's Data Science for Business

- introduces CART by example
- illustrates the use of Entropy
- it shows the training and testing of a *predictive model.*

Models

A simplified representation of reality

A predictive model is a formula for estimating the unknown value of interest, often called the *target* attribute.

. . .

Regression: numerical target

. . .

Classification: class membership, e.g., Class-probability estimation

---

A descriptive model is a formula for estimating the underlying phenomena and causal connections between values.

Descriptive modeling often is used to work towards a causal understanding of the data-generating process (e.g., why do users watch Sci-Fi sagas?)

Supervised segmentation

divide the dataset into *segments* (sets of rows) by the value of their output variable.

. . .

If the segmentation is done using values of variables that will be known when the target is not then these segments can be used to predict the value of the target.

Example: the golf days example seen when studying Gini.

Day	Outlook	Temp.	Hum.	Wind	Play?
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	?

Questions

What are the variables that contain important information about the target variable?

Can they be selected automatically?

Selecting informative attributes

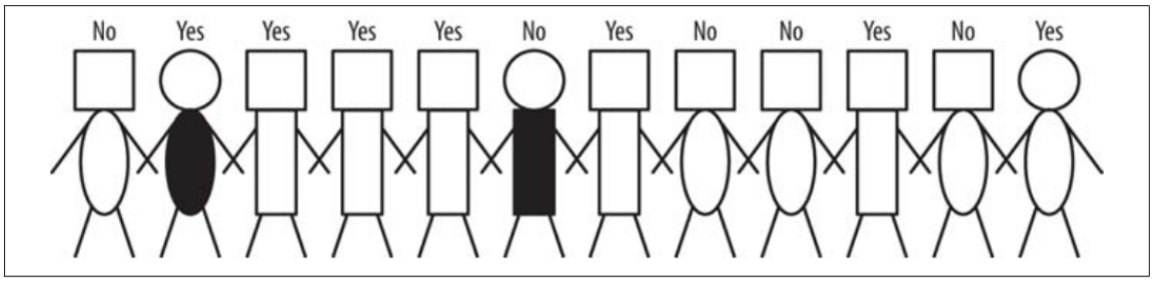


Figure 3-2. A set of people to be classified. The label over each head represents the value of the target variable (write-off or not). Colors and shapes represent different predictor attributes.

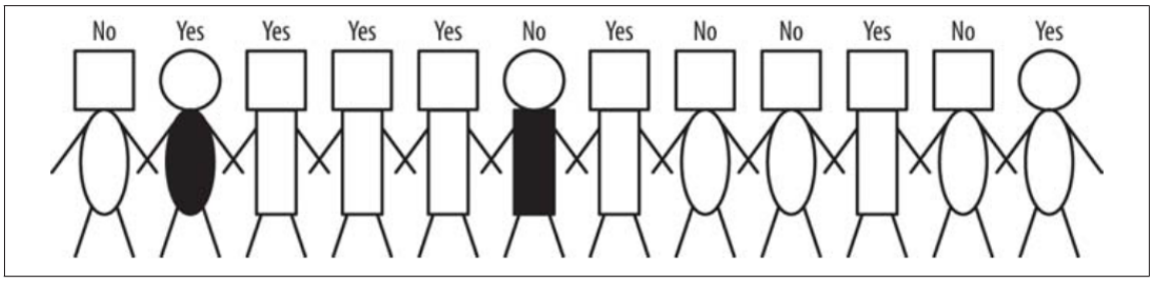


Figure 3-2. A set of people to be classified. The label over each head represents the value of the target variable (write-off or not). Colors and shapes represent different predictor attributes.

- Attributes
  - head-shape: {square, circular}
  - body-shape: {rectangular, oval}
  - body-color: {gray, white}
- Target
  - write-off: {yes, no}

Measure:

purity of segments: all datapoints have the same target variable value.

Complications

1. non-binary attributes in binary classification
2. non-discrete attributes
3. attributes seldom split a dataset perfectly.
4. attributes seldom split segments perfectly (see final slides)

Let's focus on 3.

The importance of purity

if an unlabelled datapoint is with a pure segment, we safely assigh the same target variable valus of its segment.

Day	Outlook	Temp.	Hum.	Wind	Play?
1	Sunny	Hot	High	Strong	No
2	Sunny	Hot	High	Strong	No
3	Sunny	Hot	High	Strong	No
6	Sunny	Hot	High	Strong	?

Predict 'No.'

Impurity

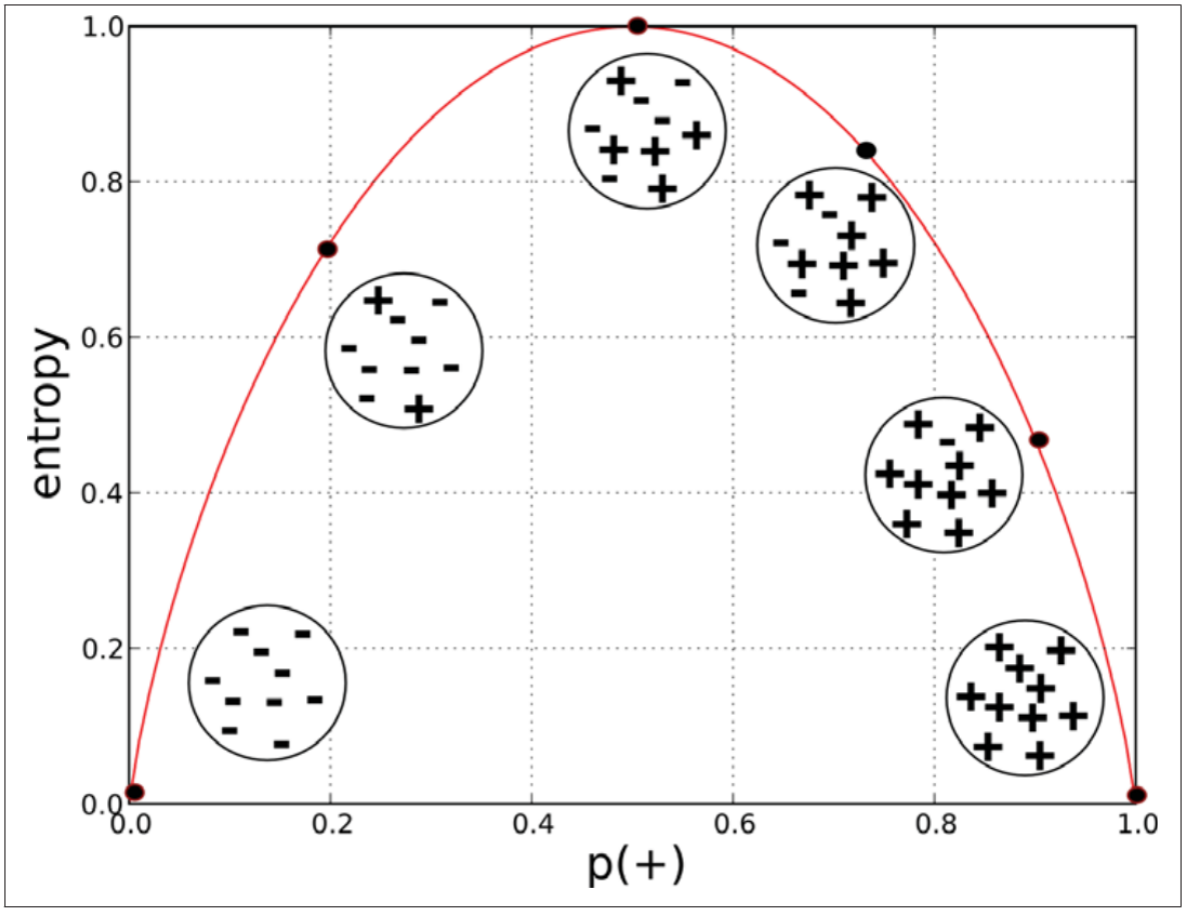
Day	Outlook	Temp.	Hum.	Wind	Play?
1	Sunny	Hot	High	Strong	No
2	Sunny	Hot	High	Strong	No
3	Sunny	Hot	High	Strong	No
4	Sunny	Hot	High	Strong	Yes
6	Sunny	Hot	High	Strong	?

assign the target value with the same probability as the frequency in the segment:

$Pr[\text{Play}_6 = \text{'No'}] = \frac{3}{4}$, $Pr[\text{Play}_6 = \text{'Yes'}] = \frac{1}{4}$.

(alternative: majority voting?)

Entropy and impurity



We would improve prediction by decreasing segment impurity.

Entropy (or Gini impurity) *drives* segmentation.

Information gain

Definition

IG measures how much much an attribute improves entropy over the whole segmentation it creates.

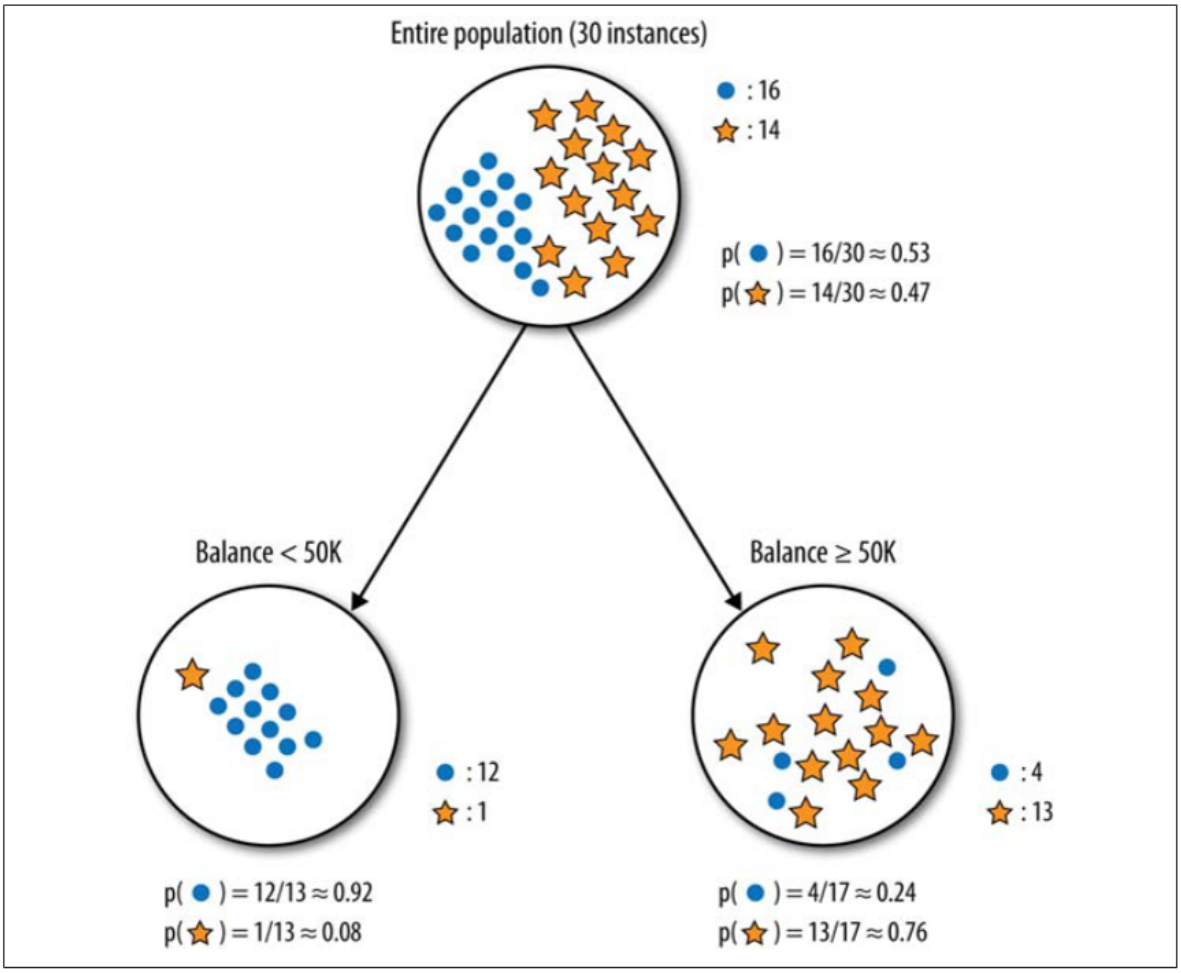


Figure 3-4. Splitting the "write-off" sample into two segments, based on splitting the Balance attribute (account balance) at 50K.

How much purer are the children wrt. their parent segment?

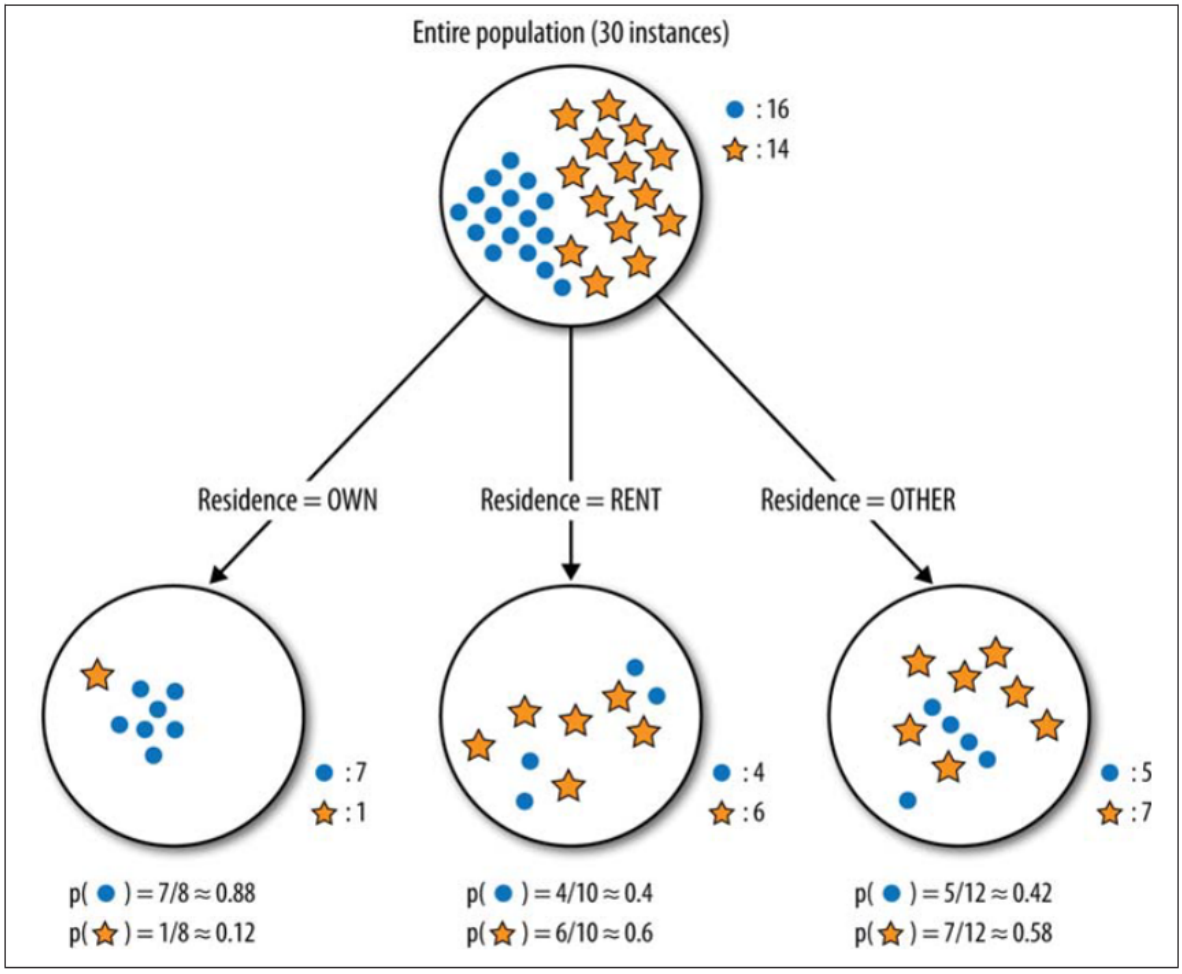


Figure 3-5. A classification tree split on the three-valued Residence attribute.

IG(parent, children) = H(parent) - $[prop(c_a) \cdot H(c_a) + prop(c_b) \cdot H(c_b) + ...]$

where $prop(c_x)$ is the proportion of el. assigned to c_i : $\frac{|c_i|}{n}$

. . .

[$IG(P, C)$ is connected with Kullback-Leibler divergence]

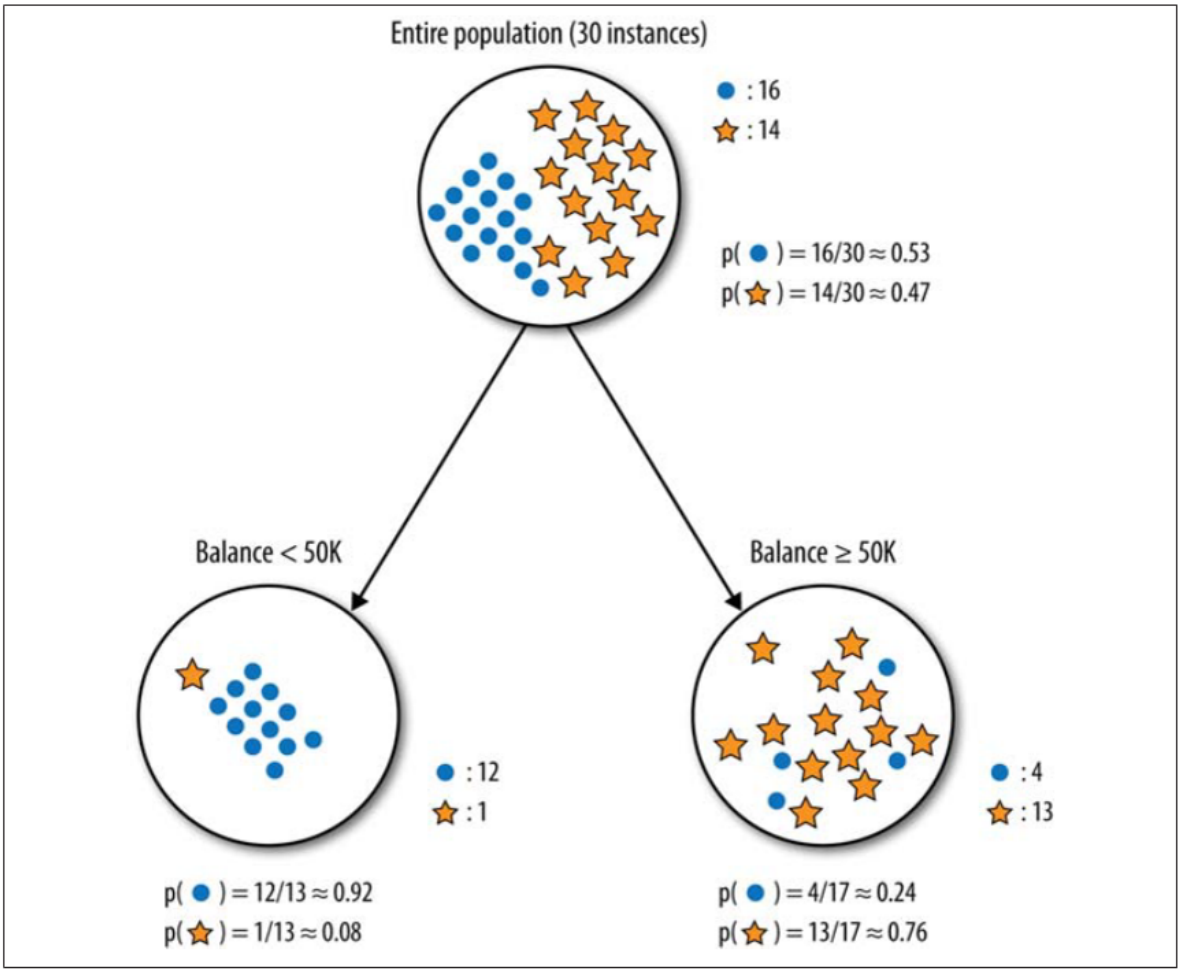


Figure 3-4. Splitting the "write-off" sample into two segments, based on splitting the Balance attribute (account balance) at 50K.

H(parent) = 0.99

H(left) = 0.39

H(right) = 0.79

$IG(p, C) = 0.99 - [13/30 \cdot 0.39 + 17/30 \cdot 0.79] = 0.37$

Notice the *discretization* of the numerical var. we *split on*

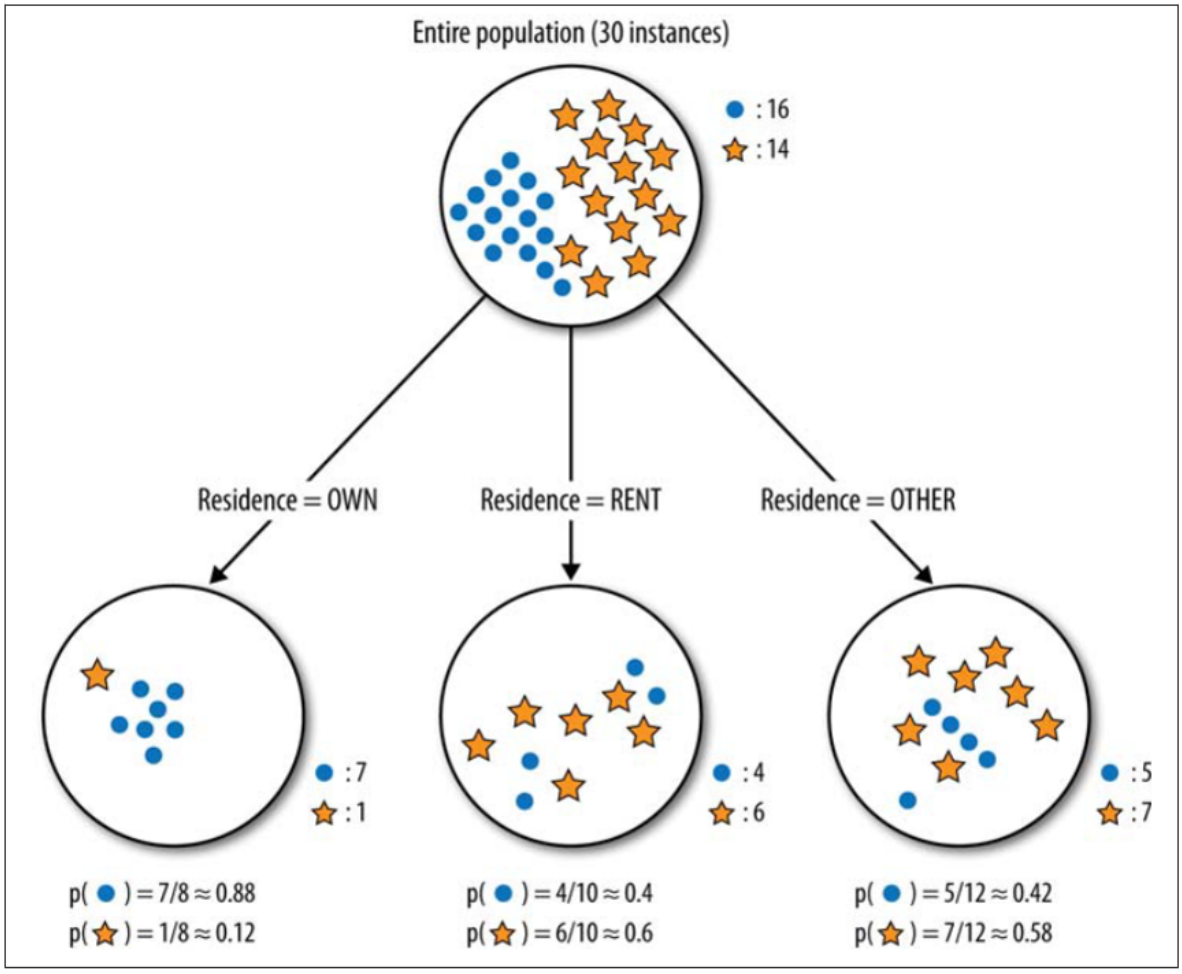


Figure 3-5. A classification tree split on the three-valued Residence attribute.

H(parent) = 0.99

H(left) = 0.54

H(center) = 0.97

H(right) = 0.98

IG(p, C) = 0.13

Numeric targets

Discretization may reduce numerical dimensions to discrete ones

In regression, variance works as the analogous of information entropy.

Attribute selection with IG

By example

A graphical method deployed to visualize Information gain:

The shaded area represents Entropy.

the white area 'reclaimed' from the shade is the Information gain at each attempt.

Mushroom Data Set

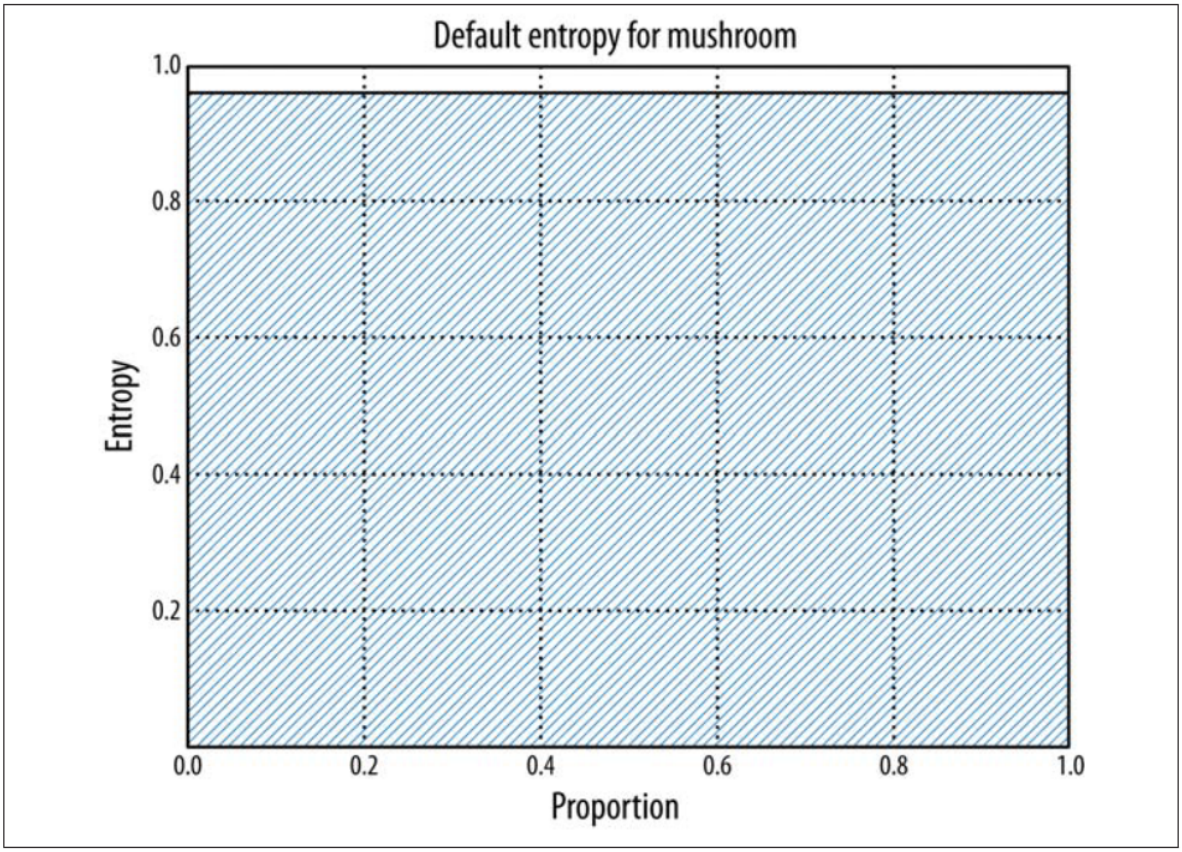
Download: Data Folder, Data Set Description

Abstract: From Audobon Society Field Guide; mushrooms described in terms of physical characteristics; classification: poisonous or edible

Data Set Characteristics:	Multivariate	**Number of Instances:**	8124	**Area:**	Life
Attribute Characteristics:	Categorical	**Number of Attributes:**	22	**Date Donated**	1987-04-27
Associated Tasks:	Classification	**Missing Values?**	Yes	**Number of Web Hits:**	500803

edible: 4208 (51.8%), poisonous: the rest.

$\mathrm{H} = -[0.518 \log 0.518 + 0.482 \log 0.482] = -[.518 \cdot -.949 + .482 \cdot -1.053] = .999$


Default entropy for mushroom
1.0
0.8
0.6
0.4
0.2
Entropy
0.0
0.2
0.4
0.6
0.8
1.0
Proportion

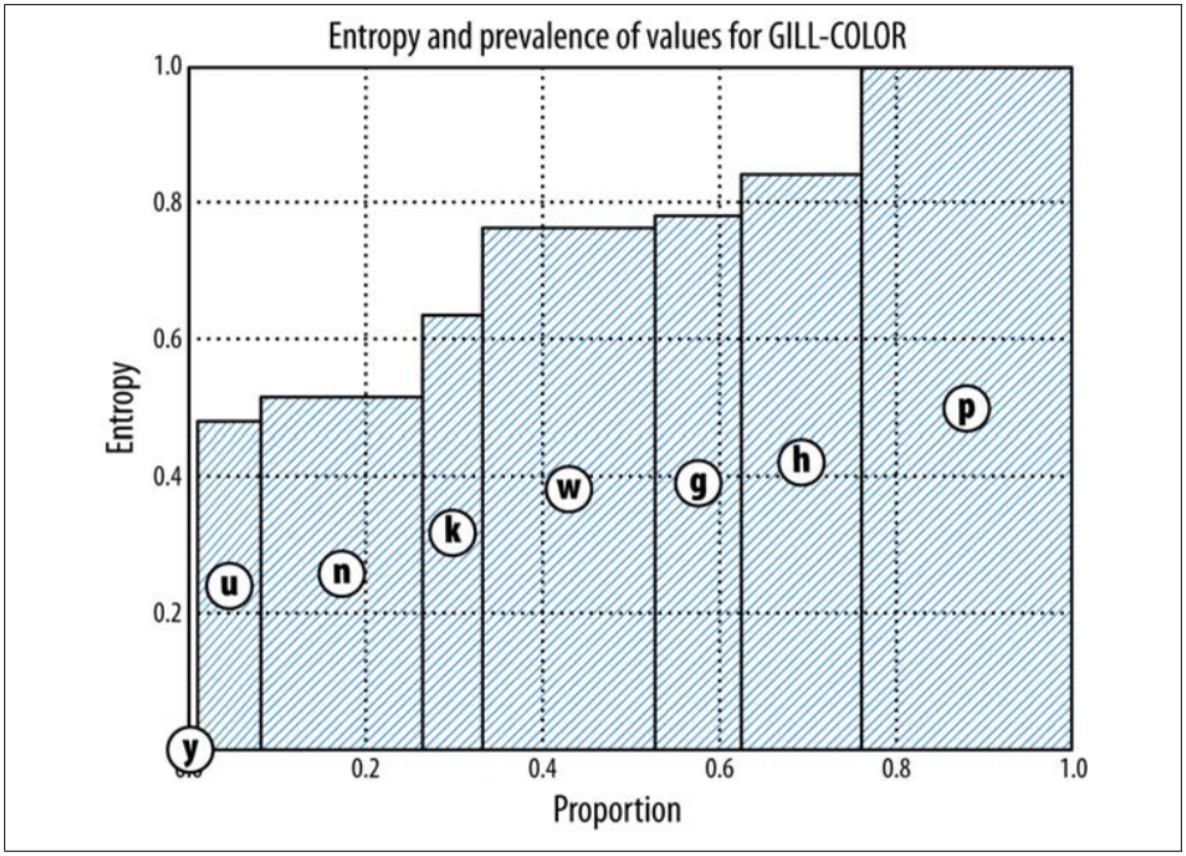


Figure 3-7. Entropy chart for the Mushroom dataset as split by GILL-COLOR. The amount of shading corresponds to the total (weighted sum) entropy, with each bar corresponding to the entropy of one of the attribute's values, and the width of the bar corresponding to the prevalence of that value in the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

Decision trees (DTs)

DT: iterated supervised segmentation

node → leaves represents a segmentation that increases *purity,* i.e., decreases information Entropy or Gini impurity.

. . .

Iterate until the set of all root → leaf trajectories gives a complete classification.

. . .

Measure: total entropy of the set of leaf segments.

Decision tree: a set of if-then rules over attribute (or discretized) values

Each observation falls into 1! leaf, and each leaf is uniquely defined by a set of rules.

CART, visually

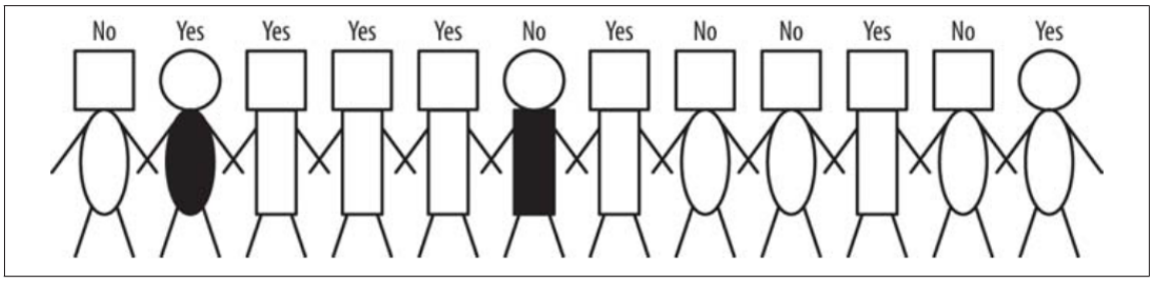


Figure 3-2. A set of people to be classified. The label over each head represents the value of the target variable (write-off or not). Colors and shapes represent different predictor attributes.

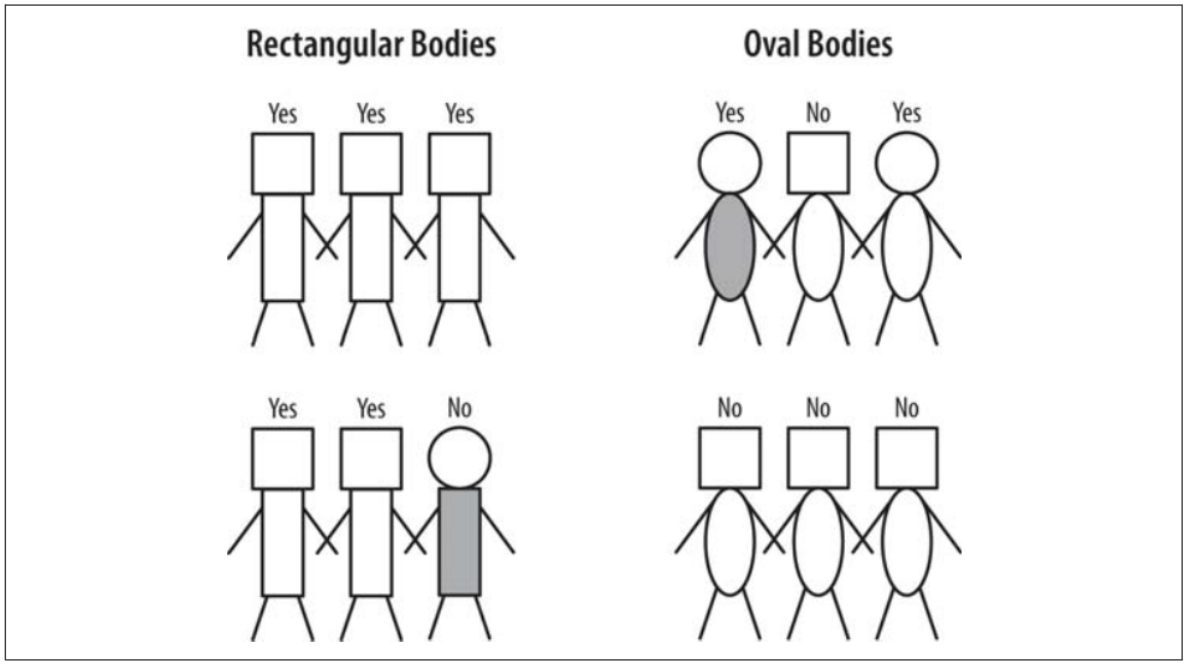


Figure 3-11. First partitioning: splitting on body shape (rectangular versus oval).

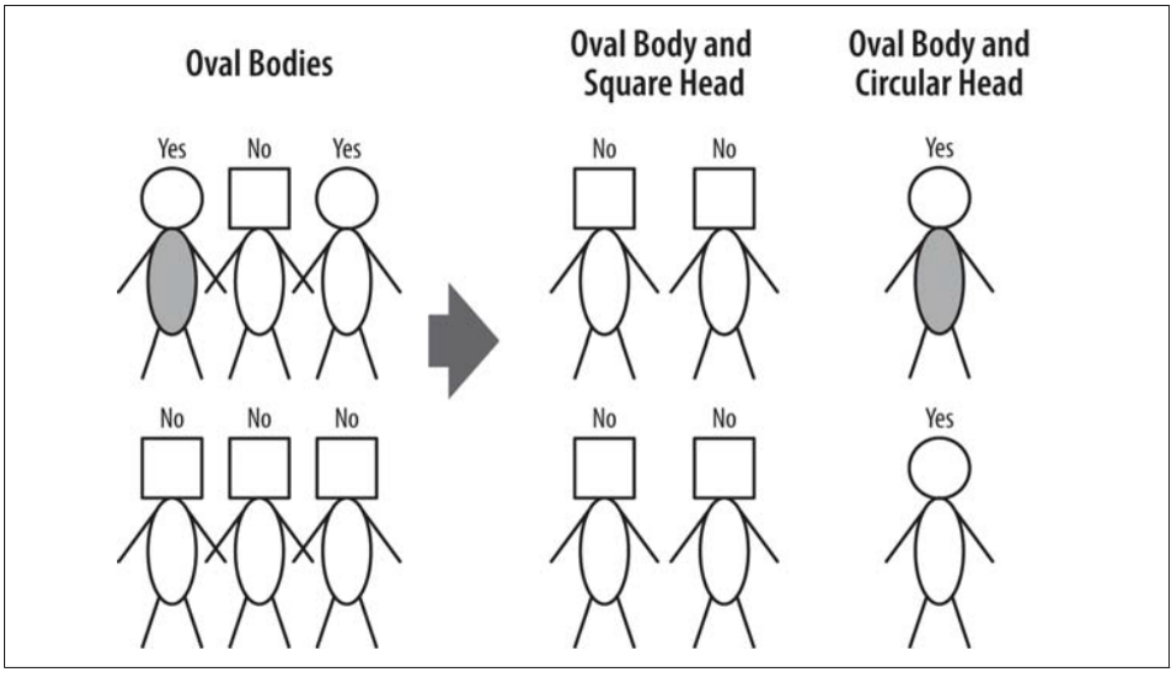


Figure 3-12. Second partitioning: the oval body people sub-grouped by head type.

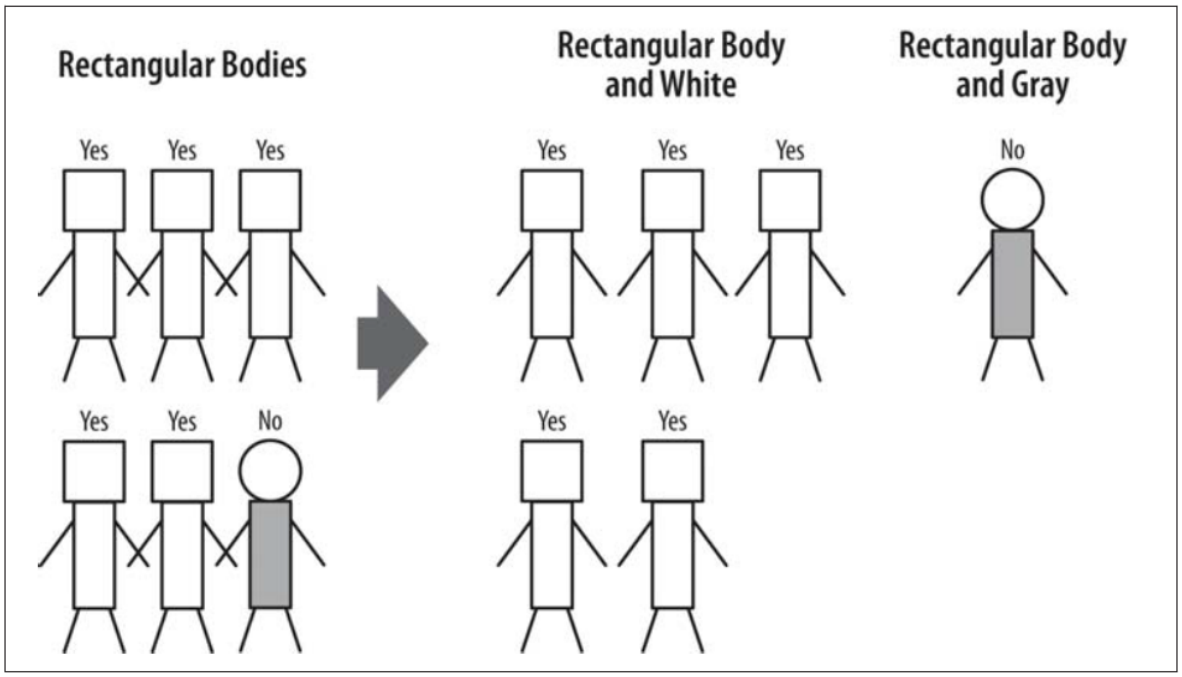
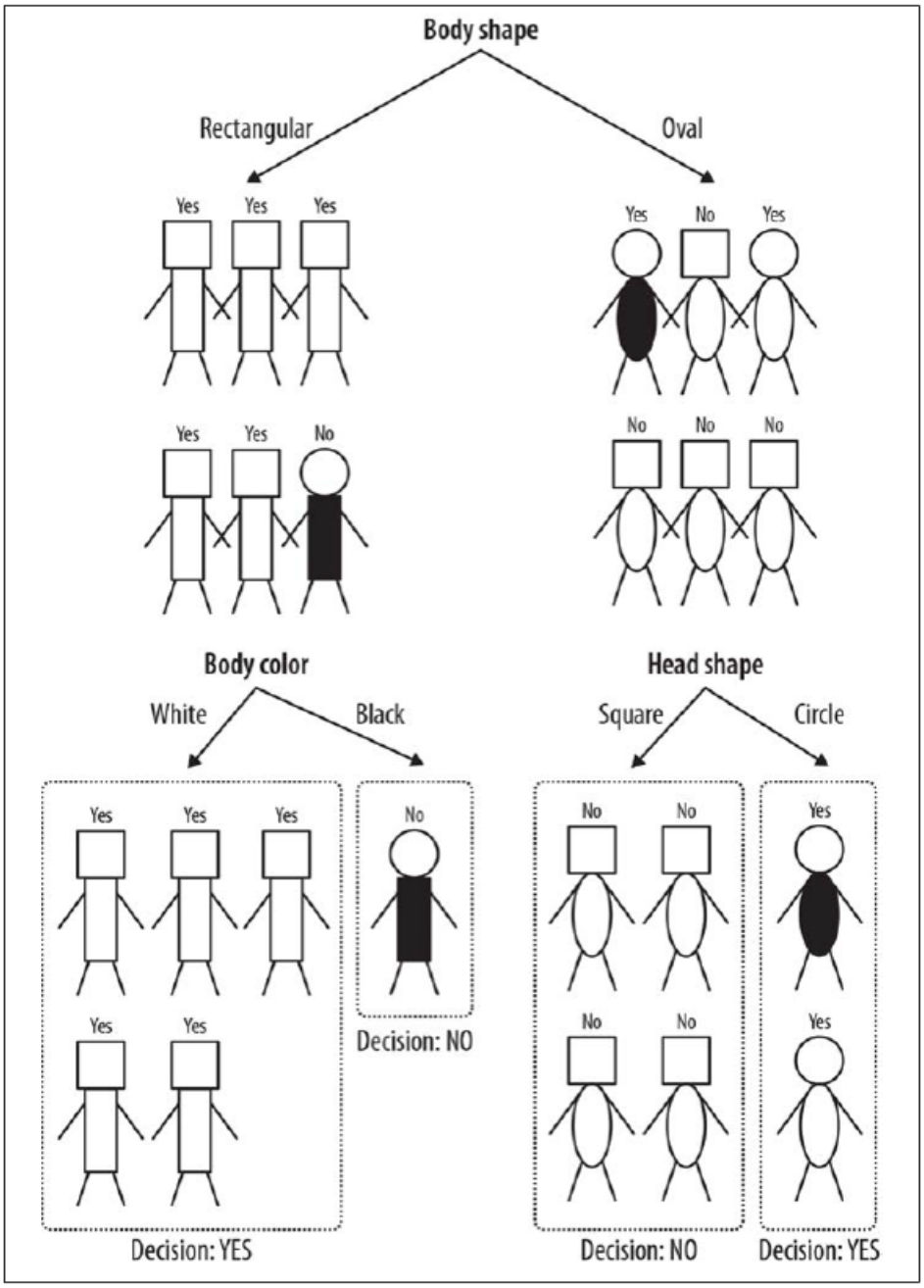


Figure 3-13. Third partitioning: the rectangular body people subgrouped by body color.


Body shape
Rectangular
Oval
Yes Yes Yes
Yes No Yes
Yes Yes No
No No No
Body color
White
Black
Head shape
Square
Circle
Yes Yes Yes
Yes Yes
No
Decision: NO
No No
No No
Yes
Yes
Decision: YES
Decision: NO
Decision: YES

Attributes seldom split groups perfectly

Even if one subgroup happens to be pure, the other may not!

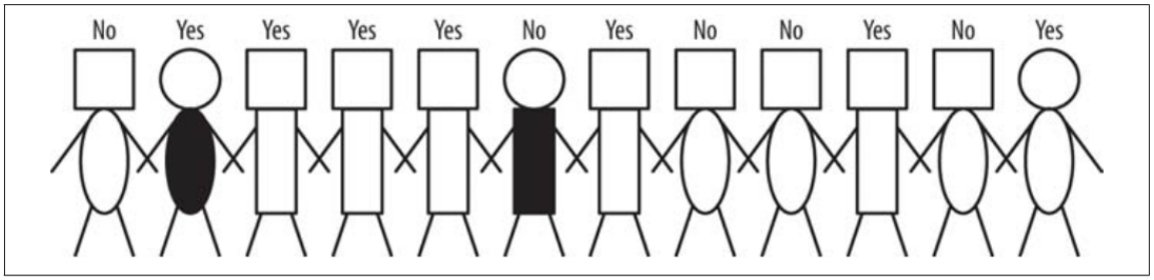


Figure 3-2. A set of people to be classified. The label over each head represents the value of the target variable (write-off or not). Colors and shapes represent different predictor attributes.

If the second person were not there, then `body-color=gray` would create a pure segment (write-off=no).

However, `body-color=white`, still is not pure.

`body-color=gray` only splits off one single data point into the pure subset.

Is this better than a split that does not produce any pure subset, but reduces impurity more broadly?