Data Science

8 - Tree-Based and Ensemble Models

Asst. Prof. Dr. Santitham Prom-on

Department of Computer Engineering, Faculty of Engineering
King Mongkut's University of Technology Thonburi

Lecture 8 - Topics

- Classification models
- Decision tree concept
- Entropy, information gain and tree induction
- Evaluation of classification models

Decision tree classification and regression

https://colab.research.google.com/drive/1cvP80R1XhTRYG1Jx33wosLCk23UumyOJ

Predictive Modeling as Supervised Segmentation

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

Quantity of interest
=
Things we would like to predict or estimate

Target and Attribute

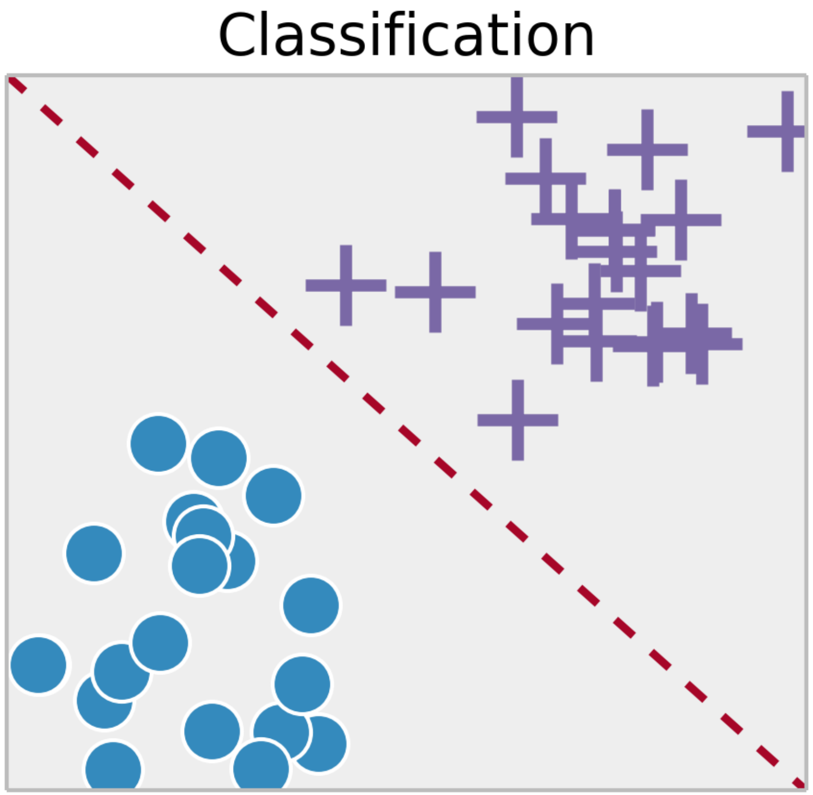
target	*Attribute 1*	*Attribute 2*	…	*Attribute k*
y_1	x_{11}	x_{12}	$\cdots$	x_{1k}
y_2	x_{21}	x_{22}	$\cdots$	x_{2k}
$\vdots$	$\vdots$	$\vdots$		$\vdots$
y_n	x_{n1}	x_{n2}	$\cdots$	x_{nk}

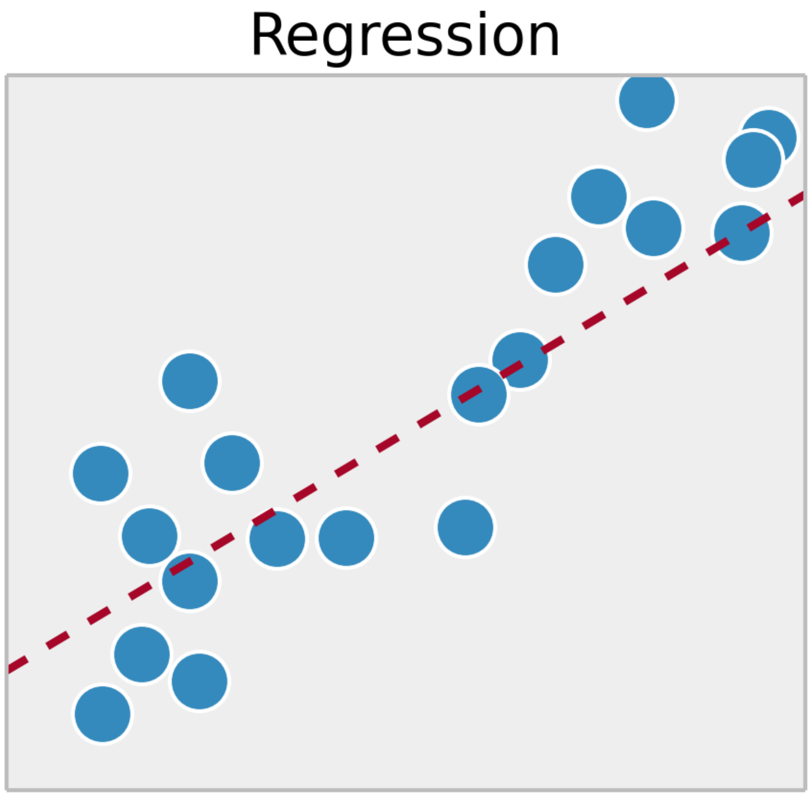
Data record

Target attribute / class to predict

Attributes

Classification vs Regression





Discrete Target (class)

Continuous Target (value)

Supervised classification Example

Attributes

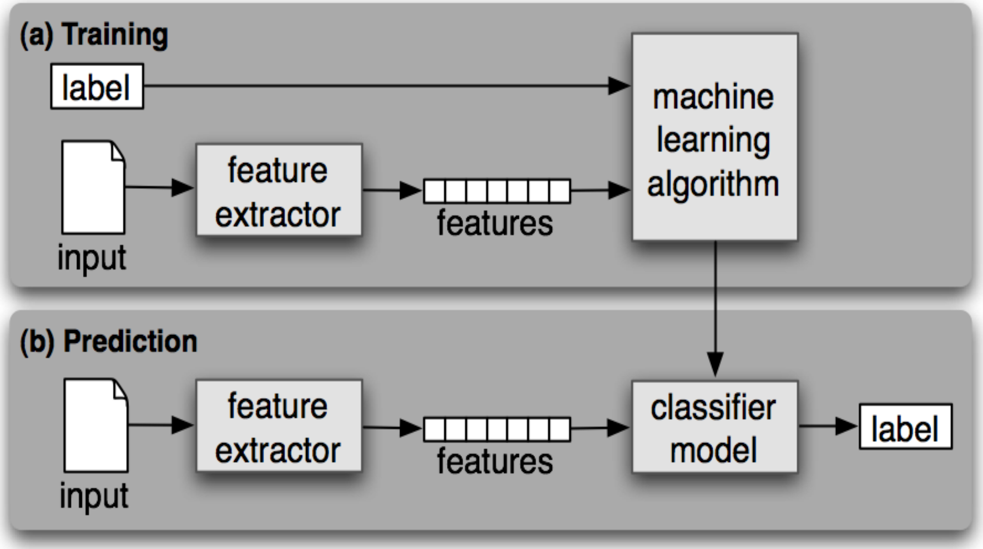
Target attribute

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**

Induction (Training)

- The creation of models from data is known as model induction (training).
- Induction is a term from philosophy that refers to generalizing from specific cases to general rules
- Our models are general rules in a statistical sense
- Most inductive procedures have variants that induce models both for classification and regression



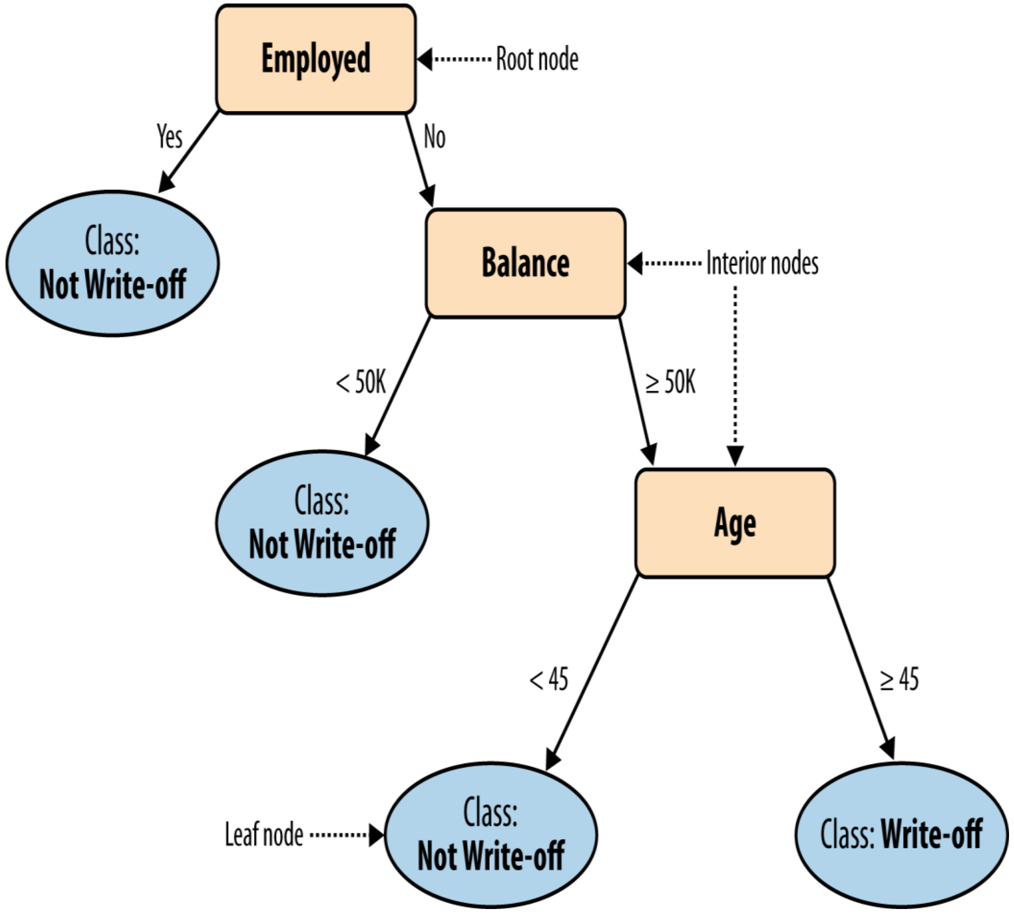
Which attribute should be used to segment?

- **Fundamental concept**: Which variable contains the most information?

- **Aims**: automatic selection, ranking

Decision Tree

- A tree consists of nodes: interior and terminal
- Interior node contains a test of an attribute
- Terminal node is a segment

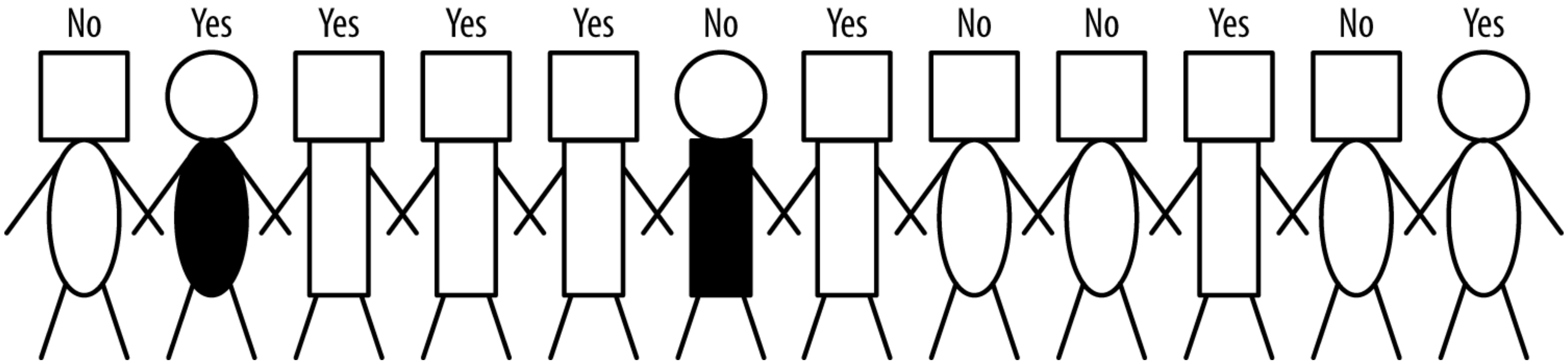


<Claudio, 115000, 40, No>?

Tree Induction

- How do we create a decision tree from data?
- Tree induction takes a divide-and-conquer approach,
    1. starting with the whole dataset
    2. applying variable selection to create subgroups
    3. Recursively repeating step 2 for each subgroup
- We will illustrate this using the write-off example

Example: Write-off



- Attributes
  - head-shape: square, circular
  - body-shape: rectangular, oval
  - body-color: black, white
- Target variable
  - write-off: Yes, No

Which attribute should be best to segment these people into groups, in a way to distinguish write-off from non-write-off?

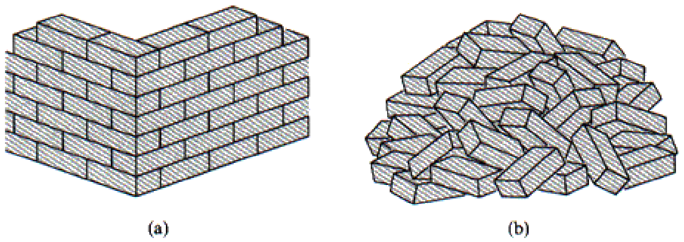
Purity

- Technically, we would like the group to be as pure as possible.
- Pure means homogeneous with respect to the target variable
- If some member in the group has a different target then the group is impure
- Comparing
  - G1 = {Y, Y, Y, Y}
  - G2 = {Y, N, N, Y}

In real data, however, we rarely find pure segments.

Entropy

- Entropy is a measure of disorder that can be applied to a set, such as one of our individual segments
- Disorder corresponds to how mixed (impure) the segment is with respect to the target
- For example, a mixed up segment with lots of write-offs and lots off non-write-offs would have high entropy

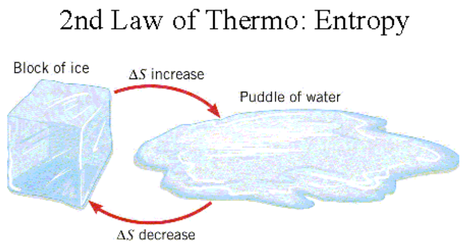


Entropy Formula

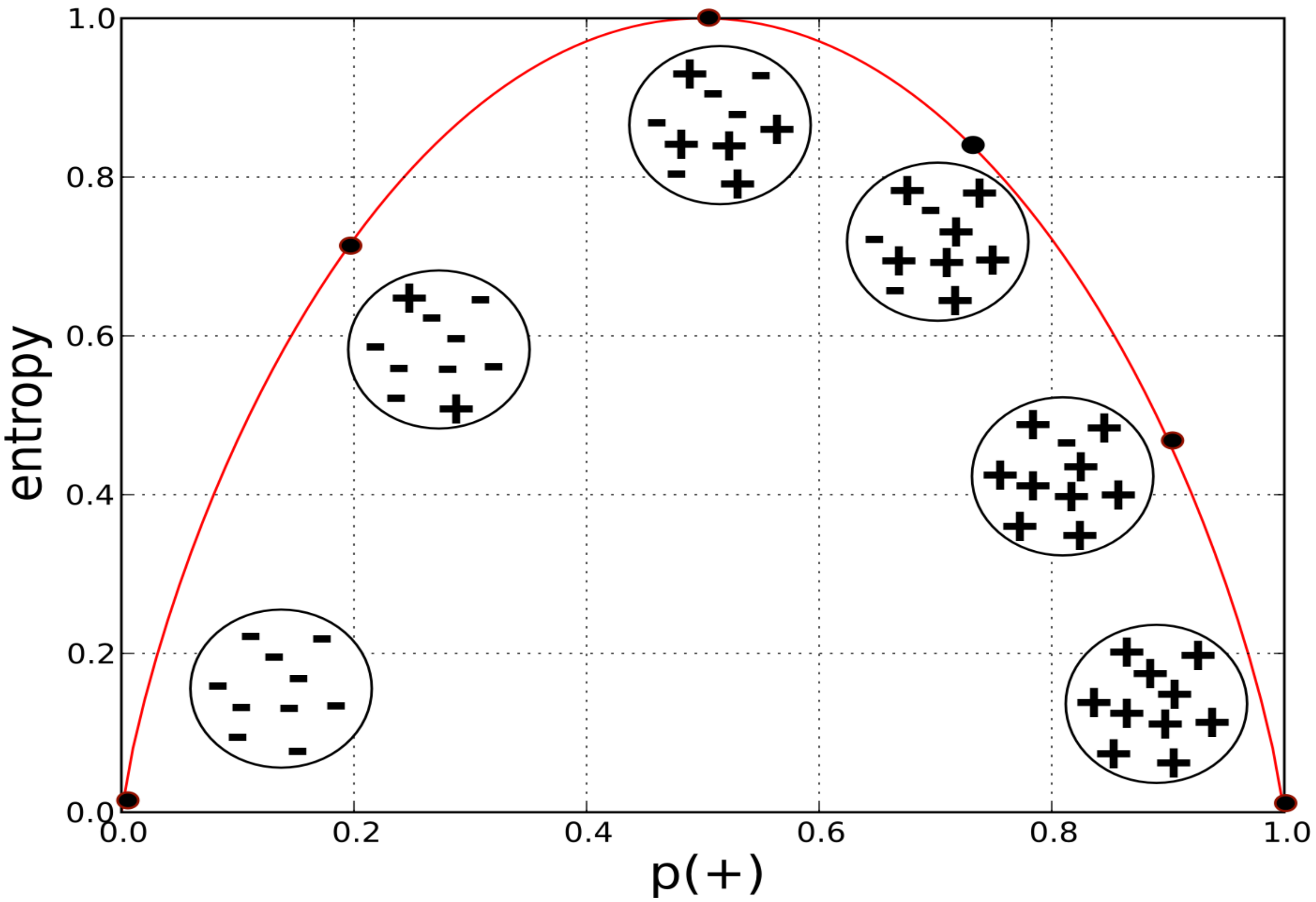
- More technically, entropy is defined as

$$entropy = -p_1\log(p_1) - p_2\log(p_2) - \ldots$$

- This is based on Gibbs entropy in thermodynamics
- Each p_i is the probability (the relative percentage) of property i (e.g. write-offs/non-write-offs) of the target.

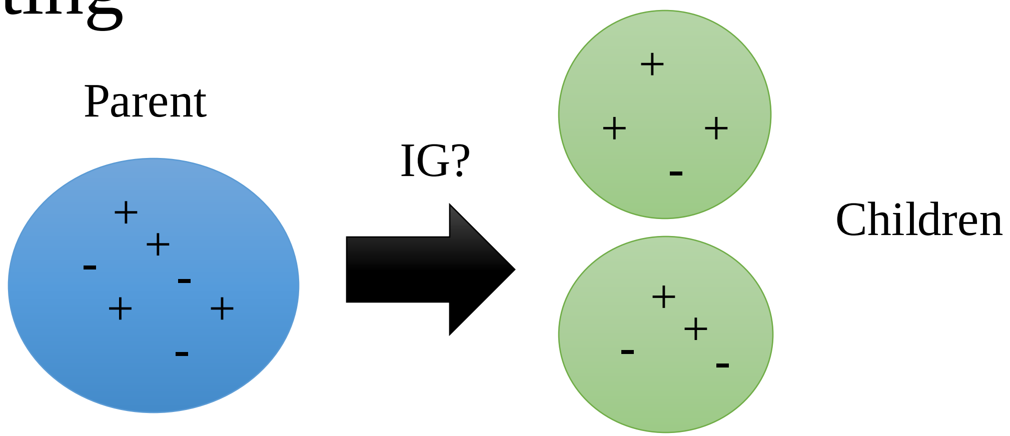


Entropy of a two-class set



Information Gain

- Entropy is only part of the story. We would like to measure how informative an attribute is with respect to target: how much gain in information it gives us about the target?
- Information gain (IG) measure how much attribute improves (decreases) entropy over the whole segmentation it creates.
- In our context, IG measures the change in entropy due to further splitting



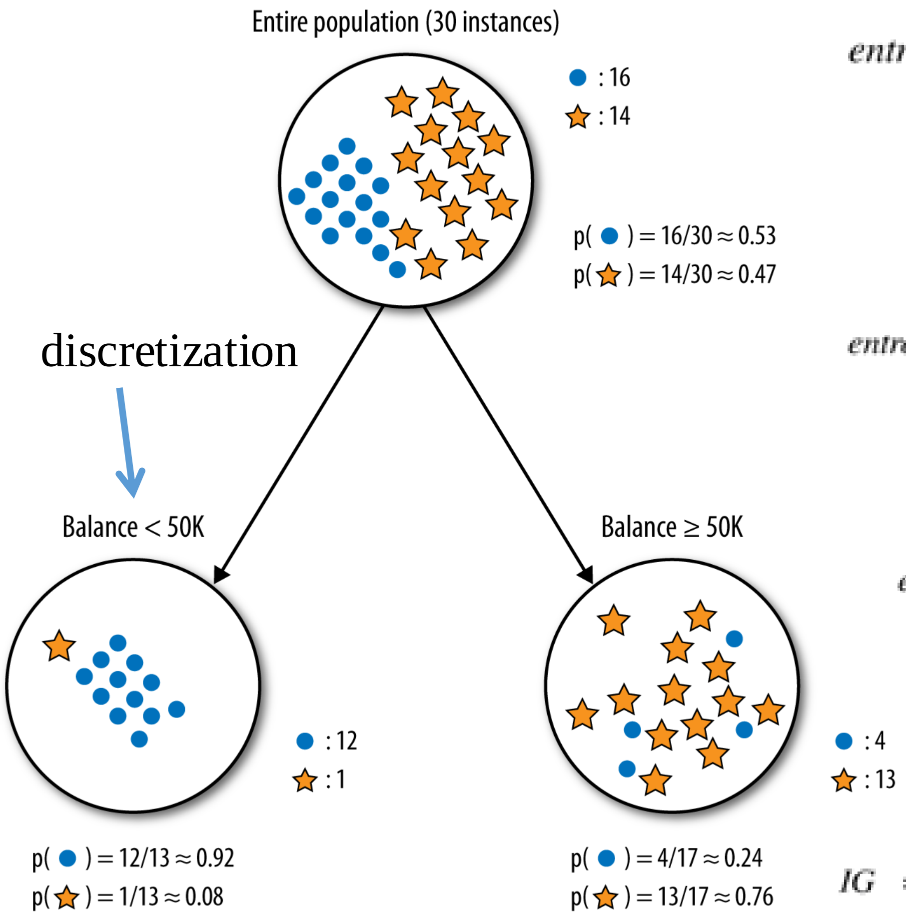
Information Gain Formula

IG(*parent*, *children*)

$$= \text{entropy}(parent) - [p(c_1)\times\text{entropy}(c_1) + p(c_2)\times\text{entropy}(c_2) + \ldots]$$

- The entropy for each child (c_i) is weighted by the proportion of instances belonging to that child, $p(c_i)$.

Example 1



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

Entropy of the left child is

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

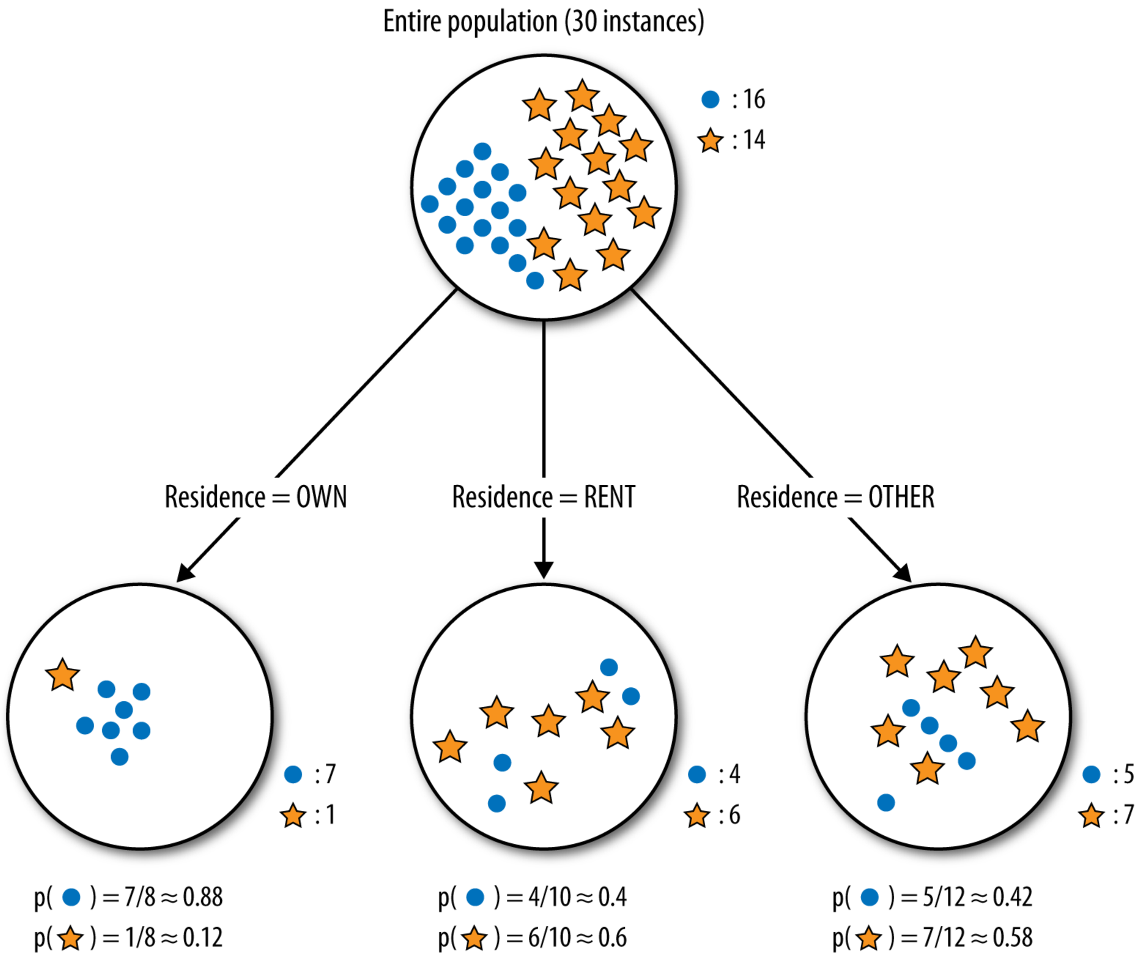
Entropy of the right child is

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

Information gain is

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

Example 2



Calculations are omitted

$entropy(parent) \approx 0.99$

$entropy(\text{Residence=OWN}) \approx 0.54$

$entropy(\text{Residence=RENT}) \approx 0.97$

$entropy(\text{Residence=OTHER}) \approx 0.98$

$\text{IG} \approx 0.13$

Residence variable is less informative than Balance.

Splitting criteria

Regression: residual sum of squares

$$RSS = \sum_{left} (y_i - y_L^*)^2 + \sum_{right} (y_i - y_R^*)^2$$

where y_L^* = mean y-value for left node

y_R^* = mean y-value for right node

Classification: Gini criterion (Similar to Information Gain)

$$Gini = N_L \sum_{k=1,\dots,K} p_{kL} (1- p_{kL}) + N_R \sum_{k=1,\dots,K} p_{kR} (1- p_{kR})$$

where p_{kL} = proportion of class k in left node

p_{kR} = proportion of class k in right node

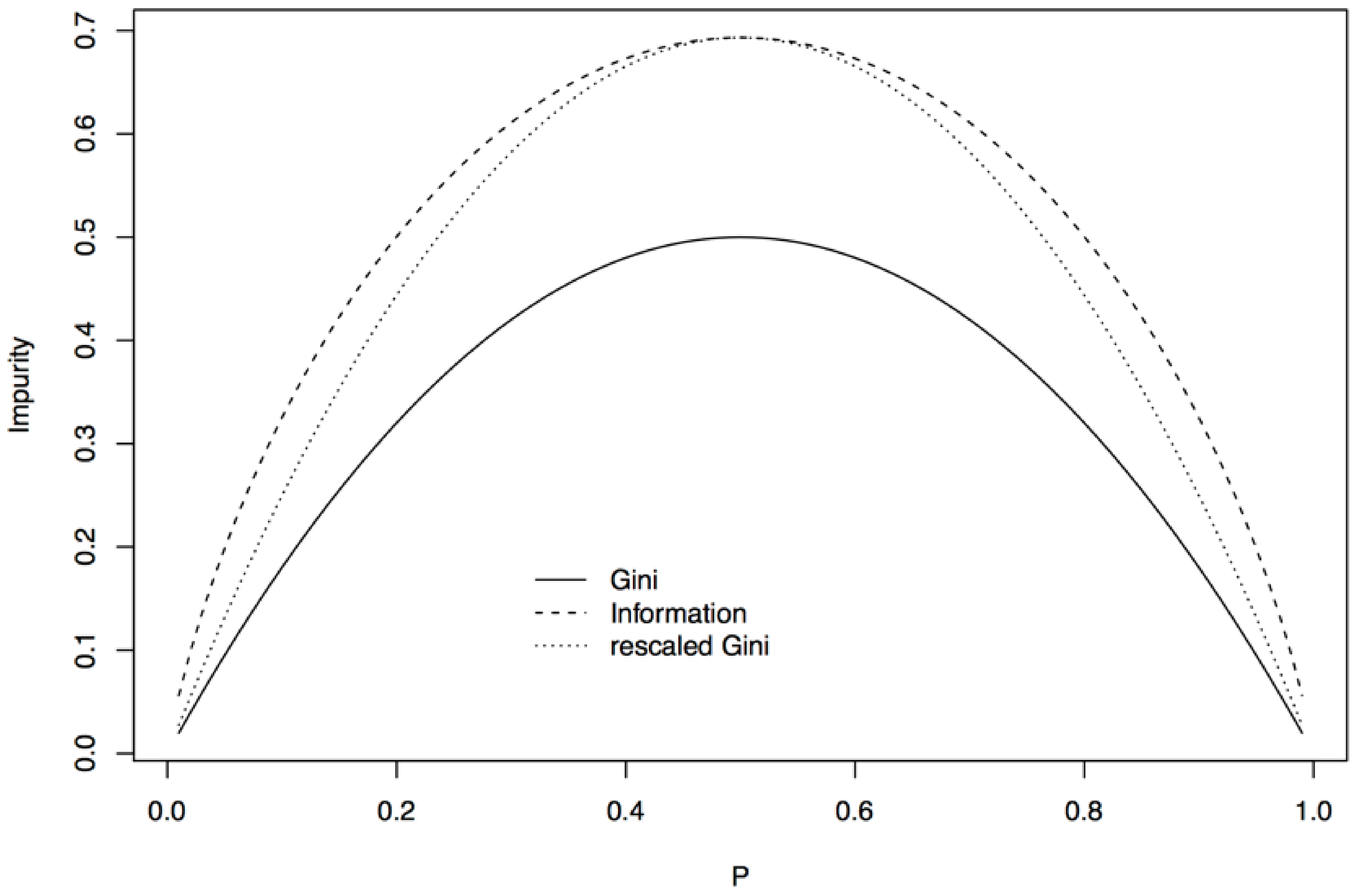
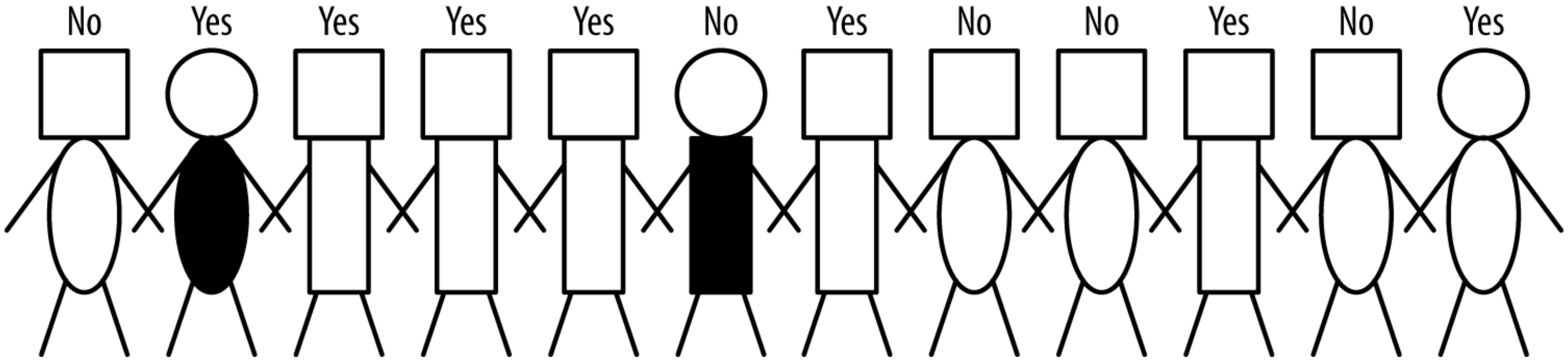


Figure 2: Comparison of Gini and Information impurity for two groups.

Example



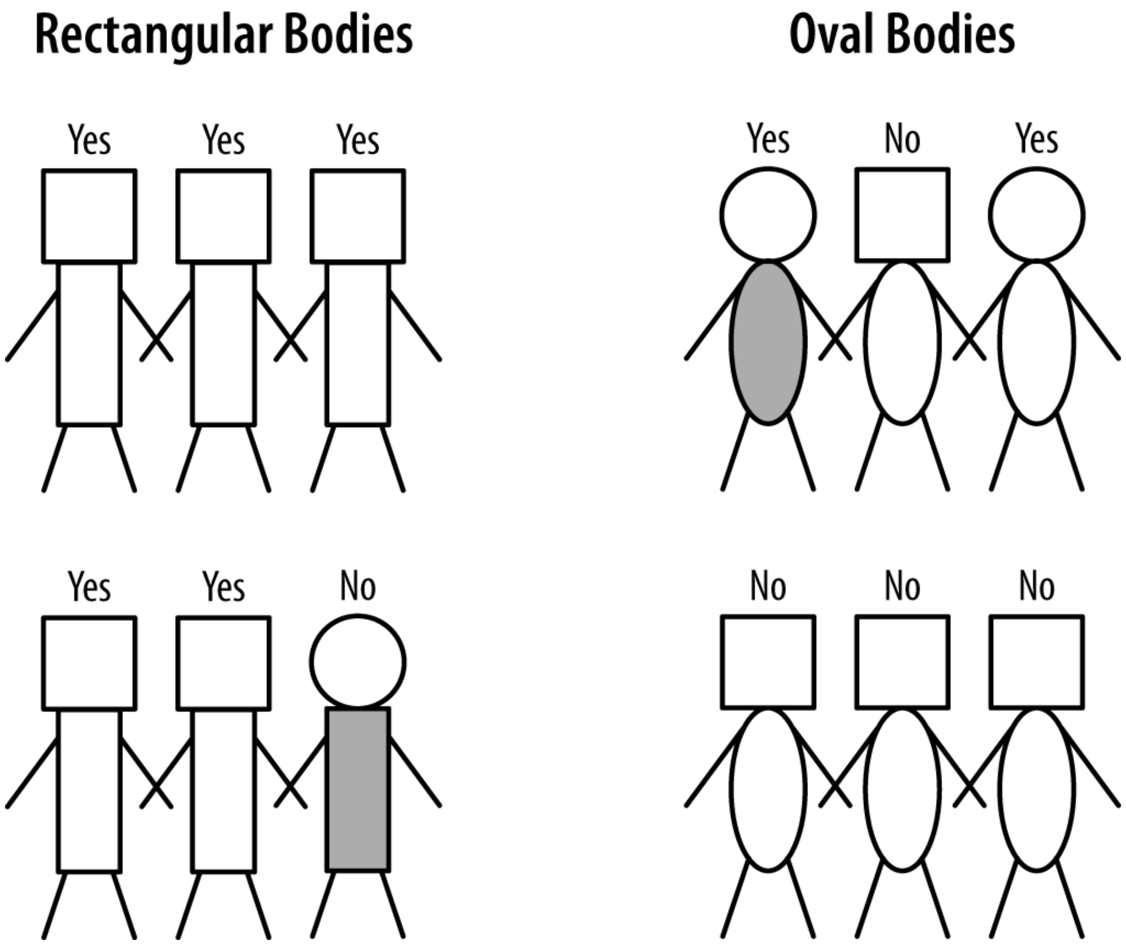
Attributes

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

Target variable

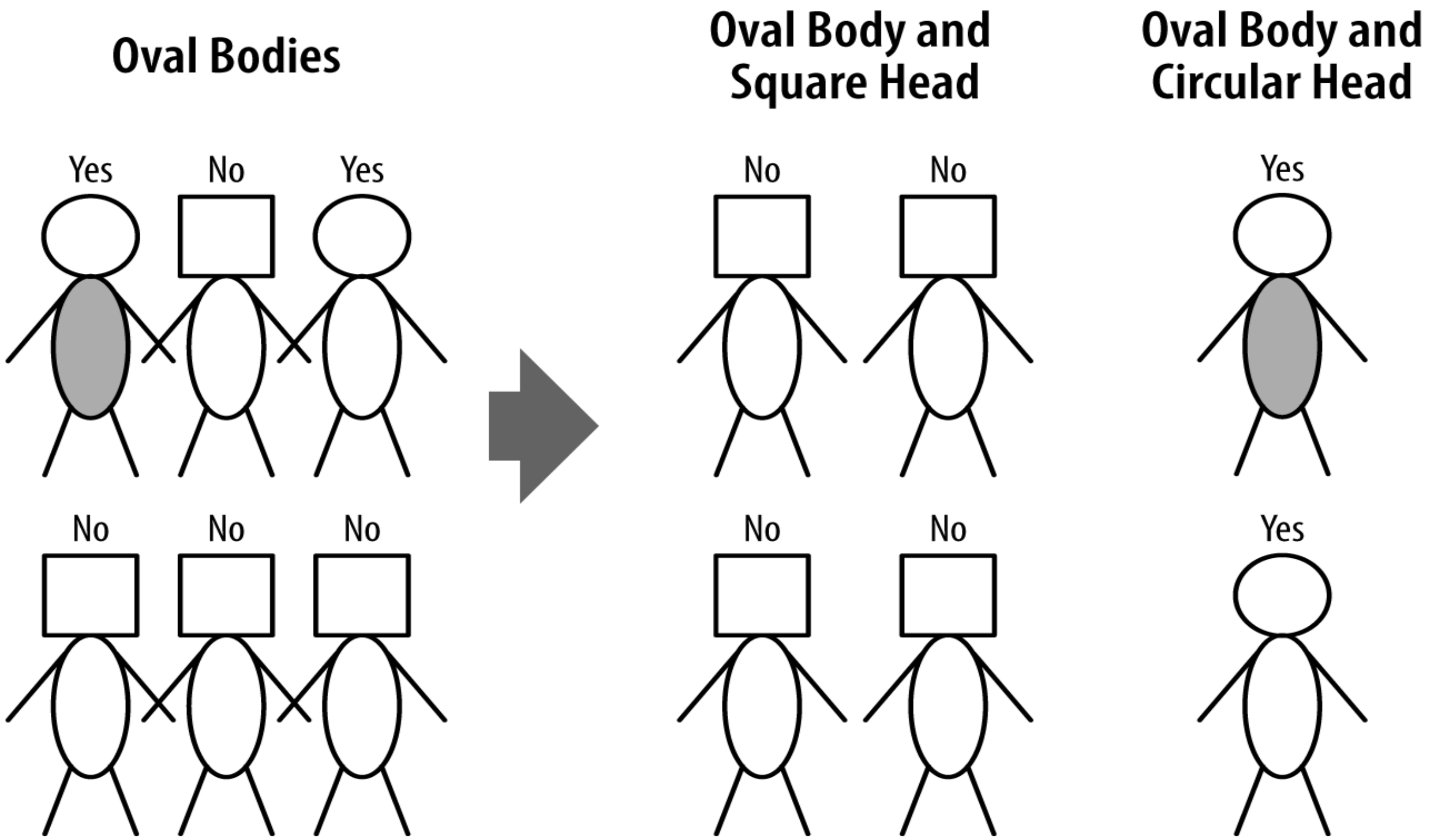
- write-off: Yes, No

First Partitioning: body-shape

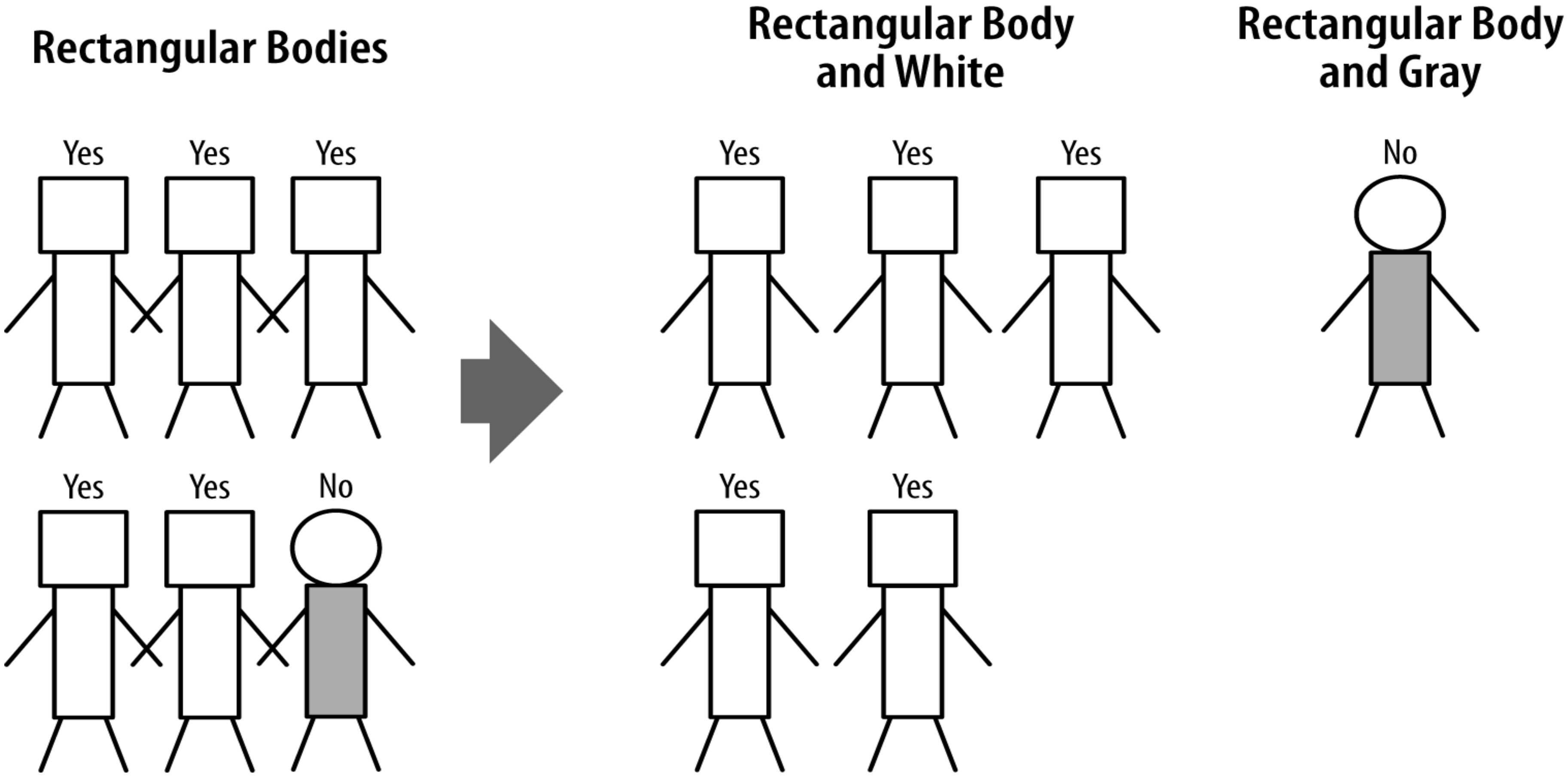


body-shape has the highest IG, so it is selected as the first attribute

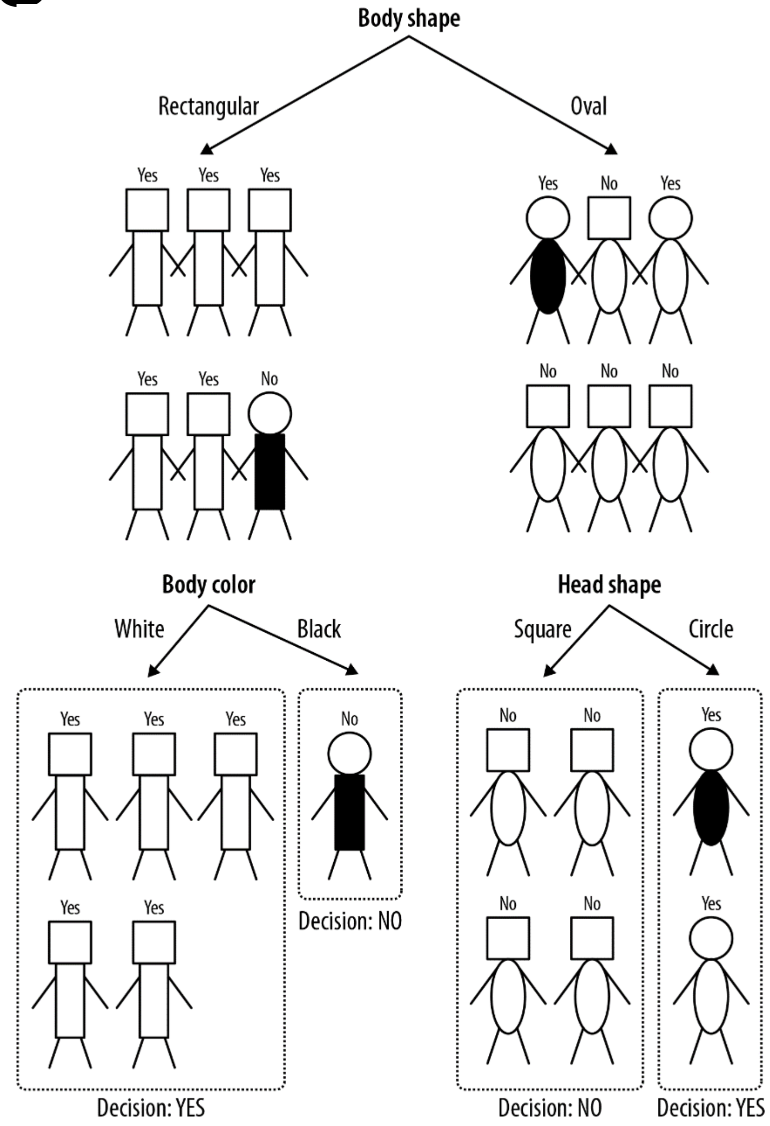
2^{nd} partitioning: oval-body, head-type



3rd partitioning: rectangular-body, body-color



Resulting Decision Tree



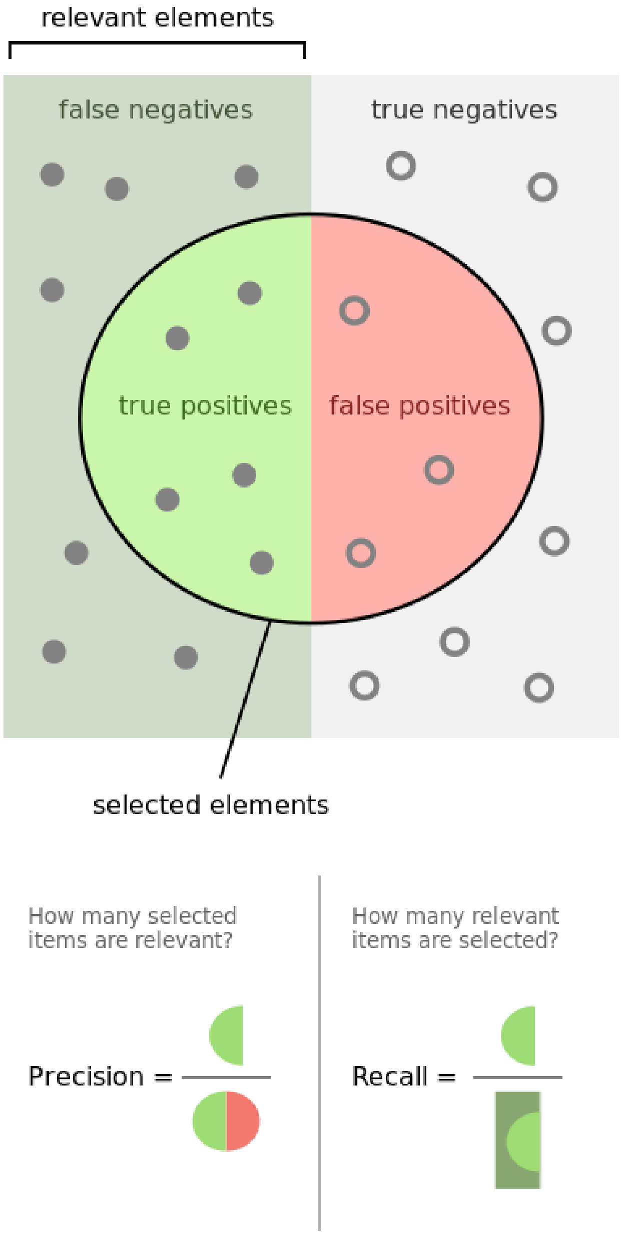
Accuracy, Precision and Recall

	Actual Positive (p)	Actual Negative (n)
The model says "Yes" = positive (y)	True positives	False positives
The model says "No" = not positive (n)	False negatives	True negatives

- Accuracy = (TP + TN)/(TP + FP + TN + FN)
- Recall (Completeness) = true positive rate = TP/(TP + FN)
- Precision (Exactness) = the accuracy over the cases predicted to be positive, TP/(TP + FP)
- F-measure = the harmonic mean of precision and recall

  = the balance between recall and precision

  $$= 2 \cdot \frac{precision * recall}{precision + recall}$$



Receiver operating characteristics Area under the ROC curve

True Positive Rate (**TPR**) is a synonym for recall and is therefore defined as follows:

$$TPR = \frac{TP}{TP + FN}$$

False Positive Rate (**FPR**) is defined as follows:

$$FPR = \frac{FP}{FP + TN}$$

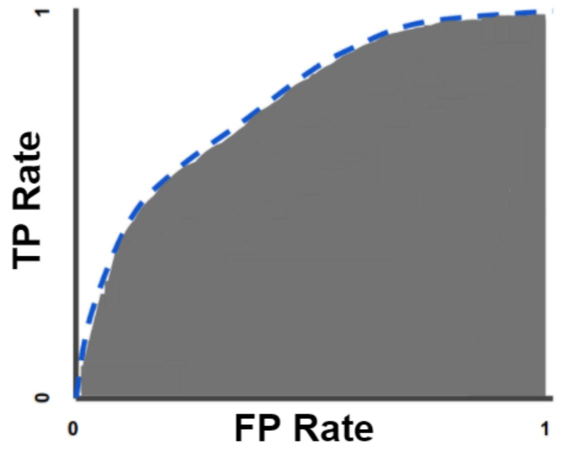


Figure 5. AUC (Area under the ROC Curve).

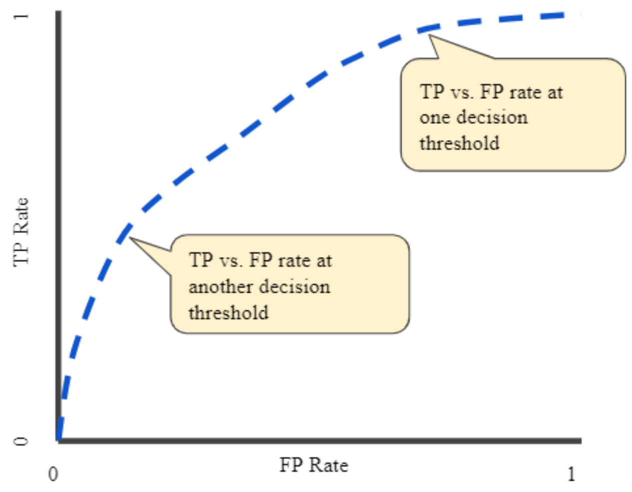
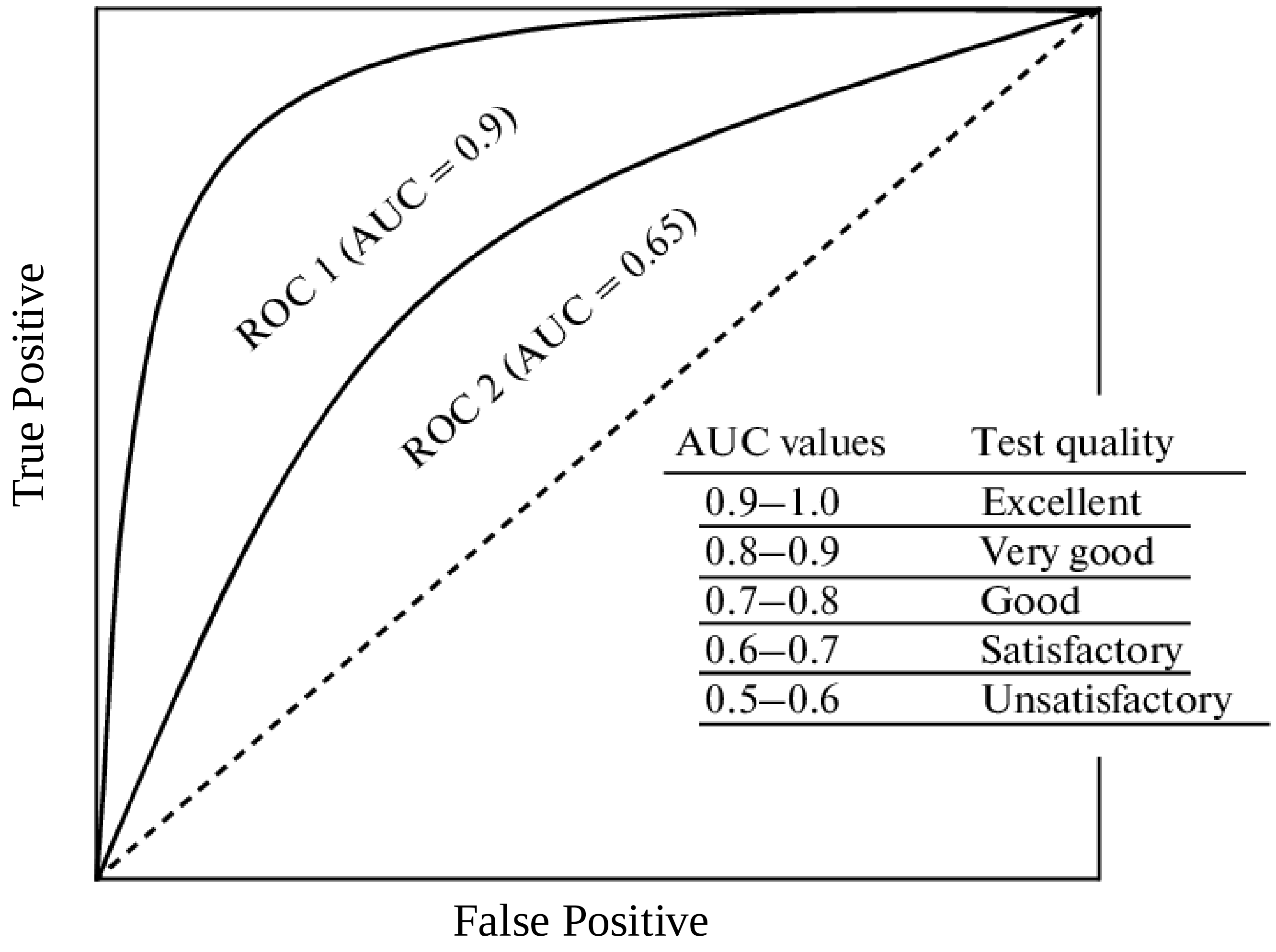


Figure 4. TP vs. FP rate at different classification thresholds.

AUC - ROC



AUC values	Test quality
0.9–1.0	Excellent
0.8–0.9	Very good
0.7–0.8	Good
0.6–0.7	Satisfactory
0.5–0.6	Unsatisfactory

End of Lecture 8