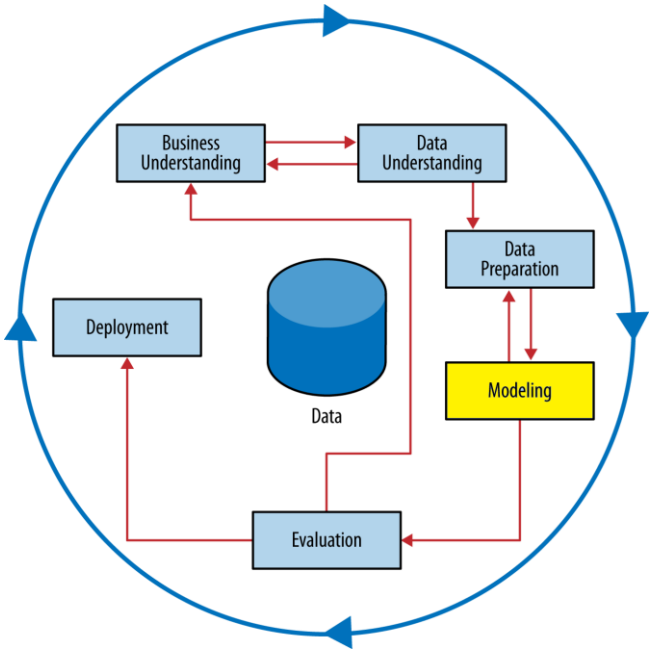
Predictive Modeling: Decision Tree

PF3

Decision Tree and Predictive Modeling

- Predictive Modeling: a Classification Example
  - Model training: tree induction
    - Supervised learning
  - Predict with a classification tree
    - Class prediction
    - Class probability estimation

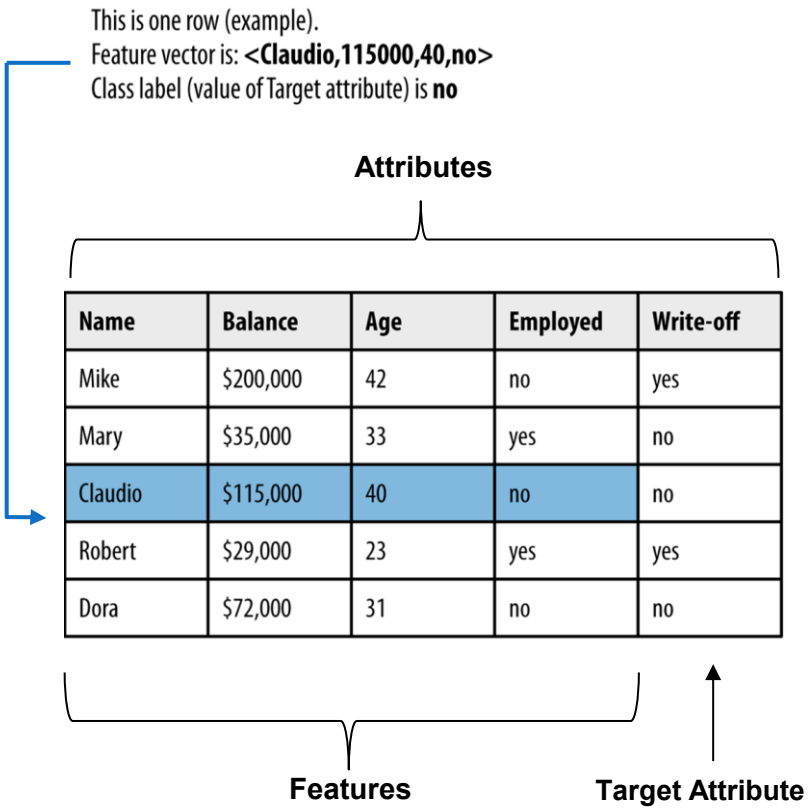


- Decision Tree
  - Identify **informative attributes** to predict the **target** value
  - Segment data by **recursive attribute selection**

Supervised Segmentation with Tree

- The **target** attribute supervises the model training process.
  - Instances (e.g., customers) are segmented into subgroups according to their feature values.
  - The goal is to have members in each subgroup with similar target values.

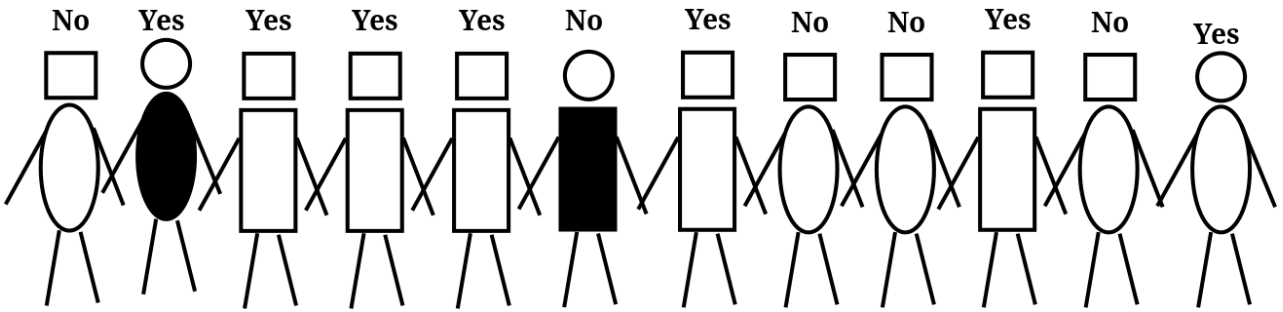
- **Informative attributes**
  - A feature is informative if it can help us learn more about the target attribute.



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

Select Informative Attributes

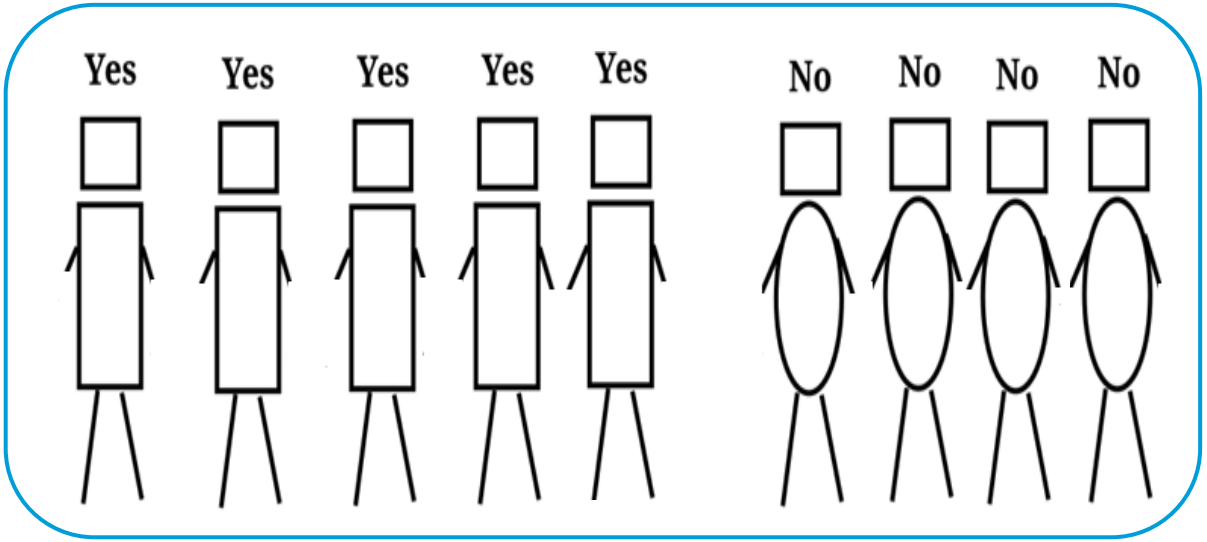
- **Objective**: based on customer **features**, segment them into subgroups so that we can best distinguish write-offs (reject) from no write-offs (approve).
  - In each subgroup, have as many instances with the same target value (i.e., class label) as possible.



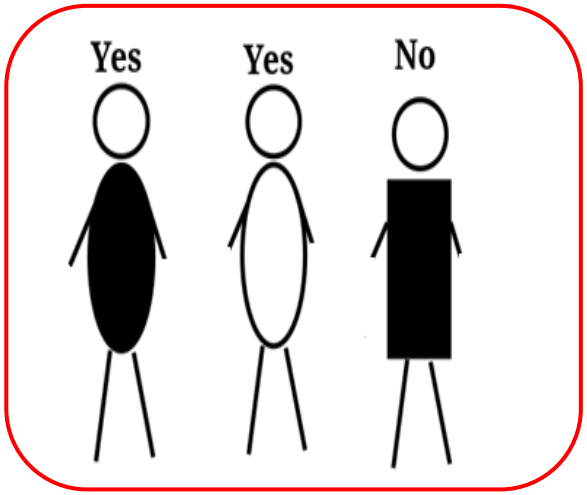
Features
Head-shape: Square, Circle
Body-shape: Rectangular, Oval
Body-Color: Black, White

Target
Loan Write-off: Yes, No

Segment by Head-Shape?



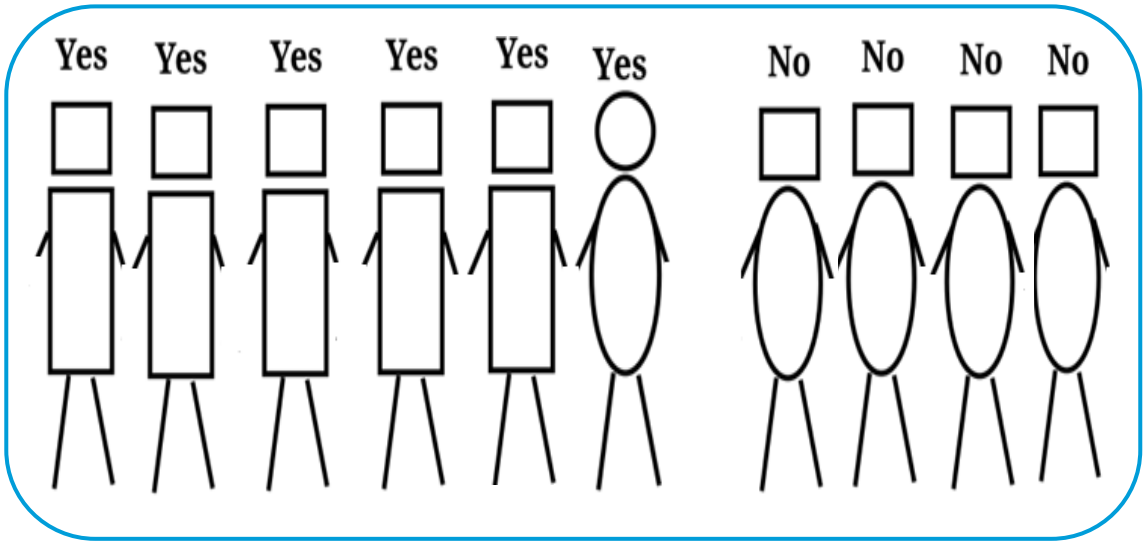
Group 1 (**Square Head**)
Result: 5 Yes 4 No



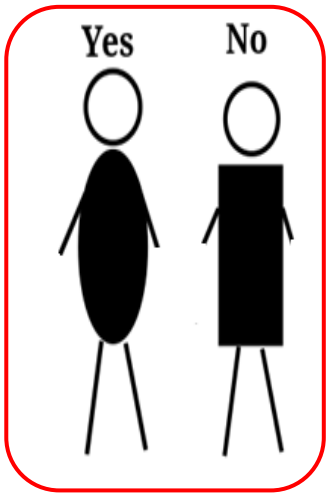
Group 2 (**Circle Head**)
Result: 2 Yes 1 No

- **Very impure in both groups**
- End the segmentation? Still room for improvement as the feature values are not homogenous.
  - Group 1 (Rectangular vs Oval body)
  - Group 2 (Oval vs Rectangular body)

Segment by Body-Color?



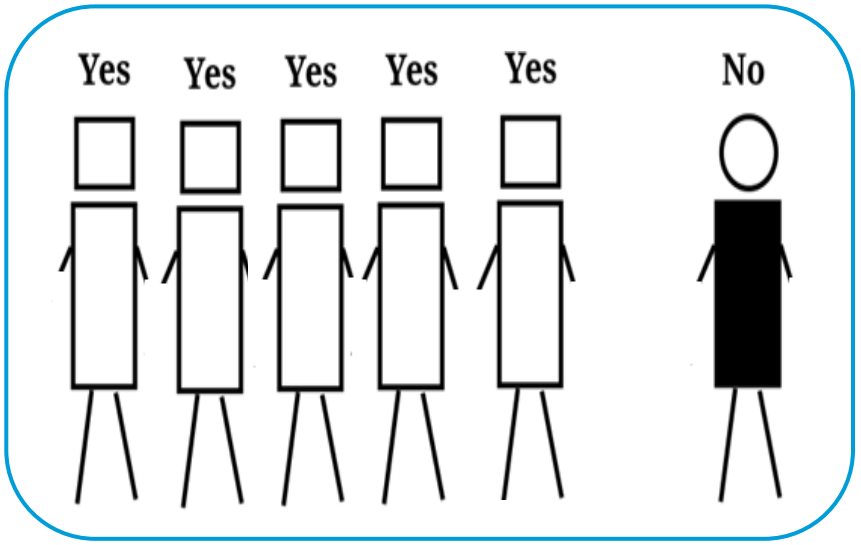
Group 1 (White Body)
Result: 6 Yes 4 No



Group 2 (Black Body)
Result: 1 Yes 1 No

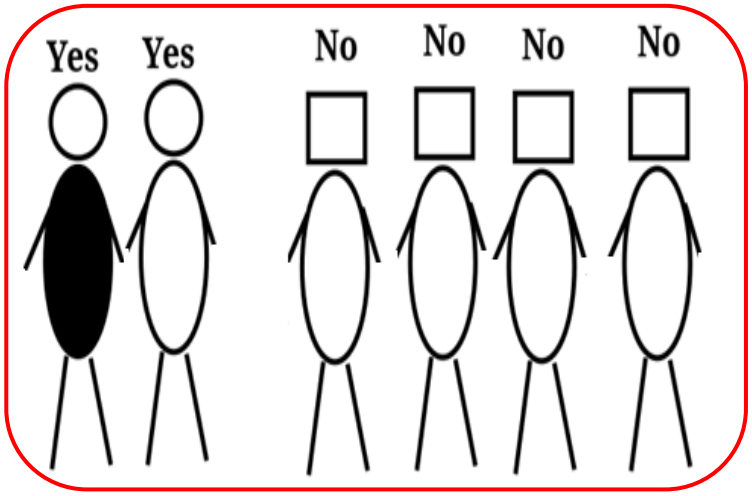
- **Very impure in both groups.**
- End segmentation? Still room for improvement.
  - Group 1 (Rectangular vs Oval body)
  - Group 2 (Rectangular vs Oval body)

Segment by Body-Shape?



Group 1 (**Rectangular Body**)
Result: 5 Yes 1 No

- **Purer than the earlier two.**
- End segmentation? Still room for improvement.
  - Group 1 (White vs Black body color)
  - Group 2 (Circle vs Square head)



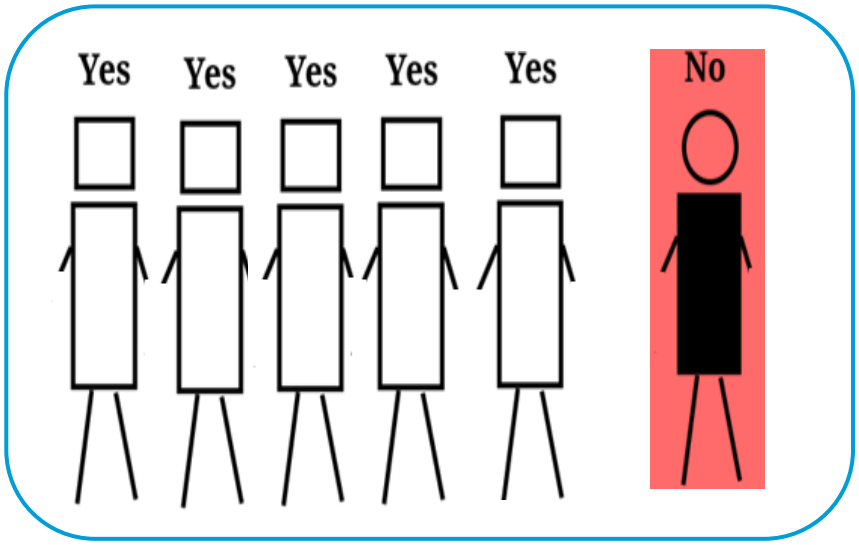
Group 2 (**Oval Body**)
Result: 2 Yes 4 No

What's Next?

- Body-shape seems to be the best feature in segmenting customers in the first split.
  - How to quantify "the usefulness" of each feature?

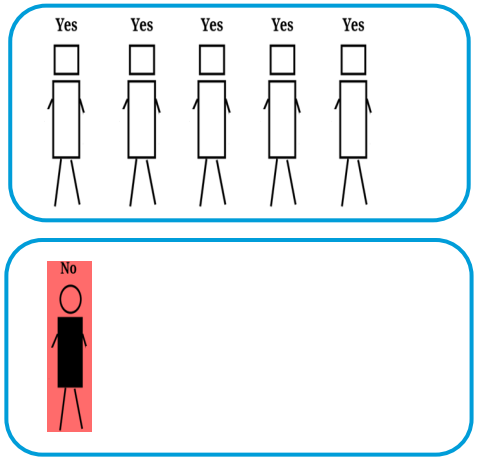
- After the first split with body-shape, each subgroup is still not pure in terms of the target. So further segmentation for each subgroup?
  - Which feature should be used in the second split?

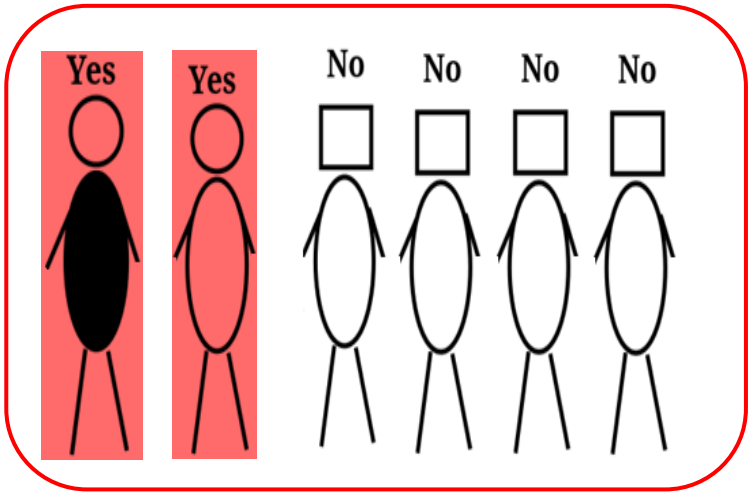
Further Segmentation (Second Split)



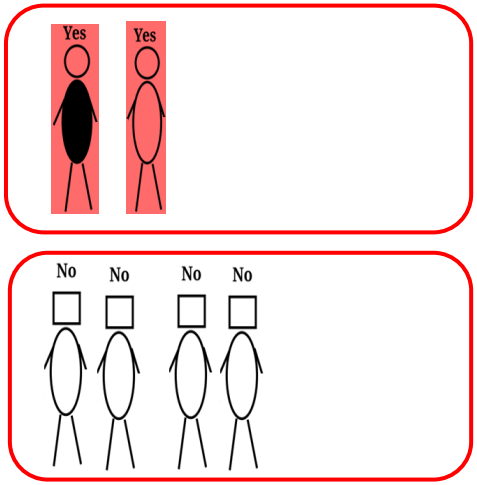
Group 1
(Rectangular Body)
Result: 5 Yes 1 No

Further segment Group 1 by **body color**



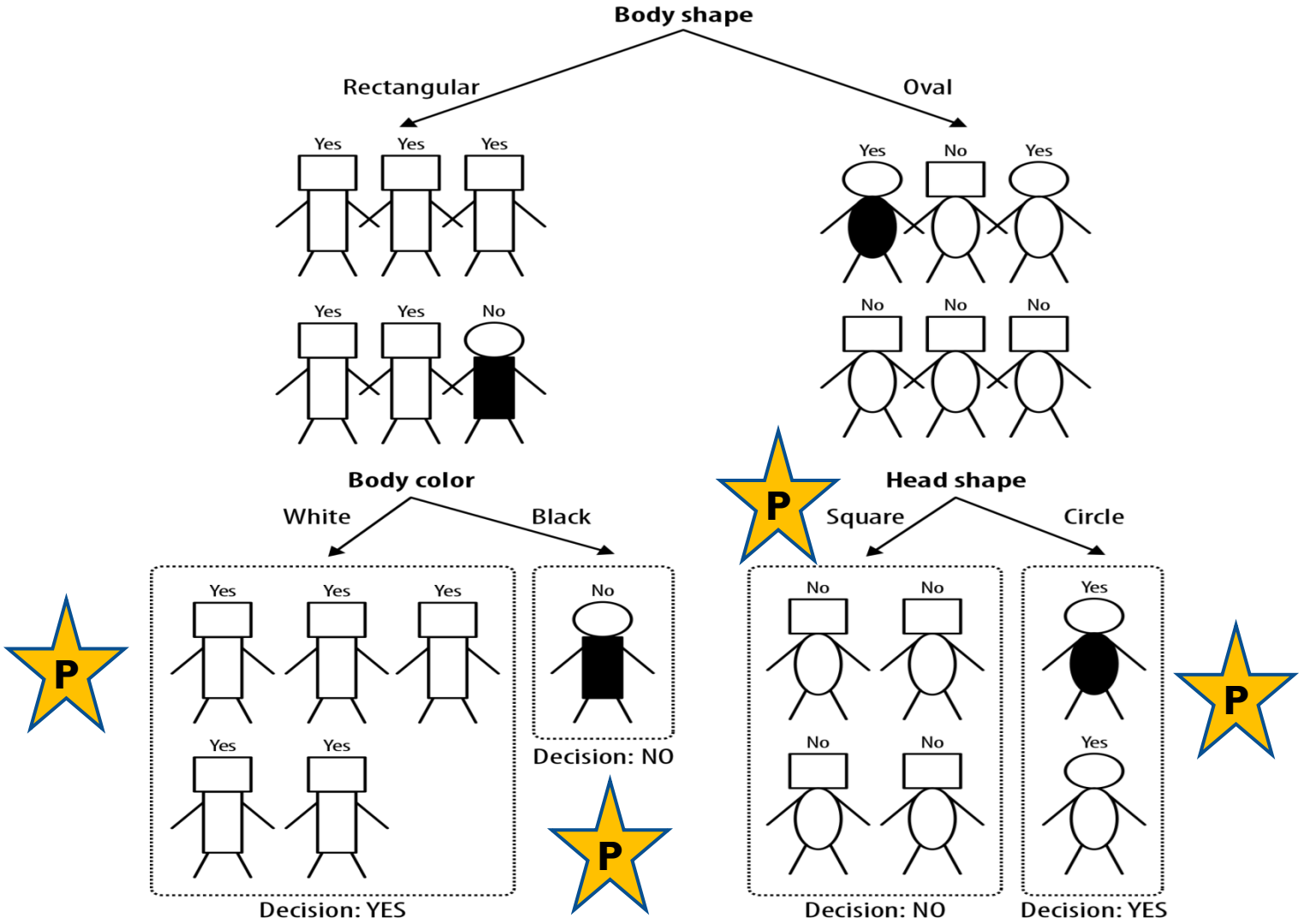


Further segment Group 2 by **head shape**



Group 2 (Oval Body)
Result: 2 Yes 4 No

Final Model


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

Informative Attributes

- Subgroups are expected to be as pure as possible.
  - A pure group: all members in a group have the same target value.
  - An impure group: at least one member has a different target value.

- **Entropy** is a measure of disorder (i.e., impurity) in a group.

$$\boldsymbol{Entropy} \downarrow \Rightarrow \boldsymbol{Purity} \uparrow$$

- Which feature is the best in splitting a node? We compare their **information gain** in a split.
  - Information gain (IG) captures the change of entropy (impurity) in the data, due to the split with a feature.
  - If a feature can reduce the impurity (entropy) substantively, then it is informative in predicting the target.

Entropy

- **Entropy** is computed with:

$$\boldsymbol{entropy} = -p_1 \log_2(p_1) - p_2 \log_2(p_2) - \cdots - p_n \log_2(p_n)$$

  - n is the number of classes for the target
    - e.g., n = 2 for a binary target attribute
  - p_i is the proportion of instances of the i^{th} class in the group.
  - $\log_2$ is logarithm with base 2

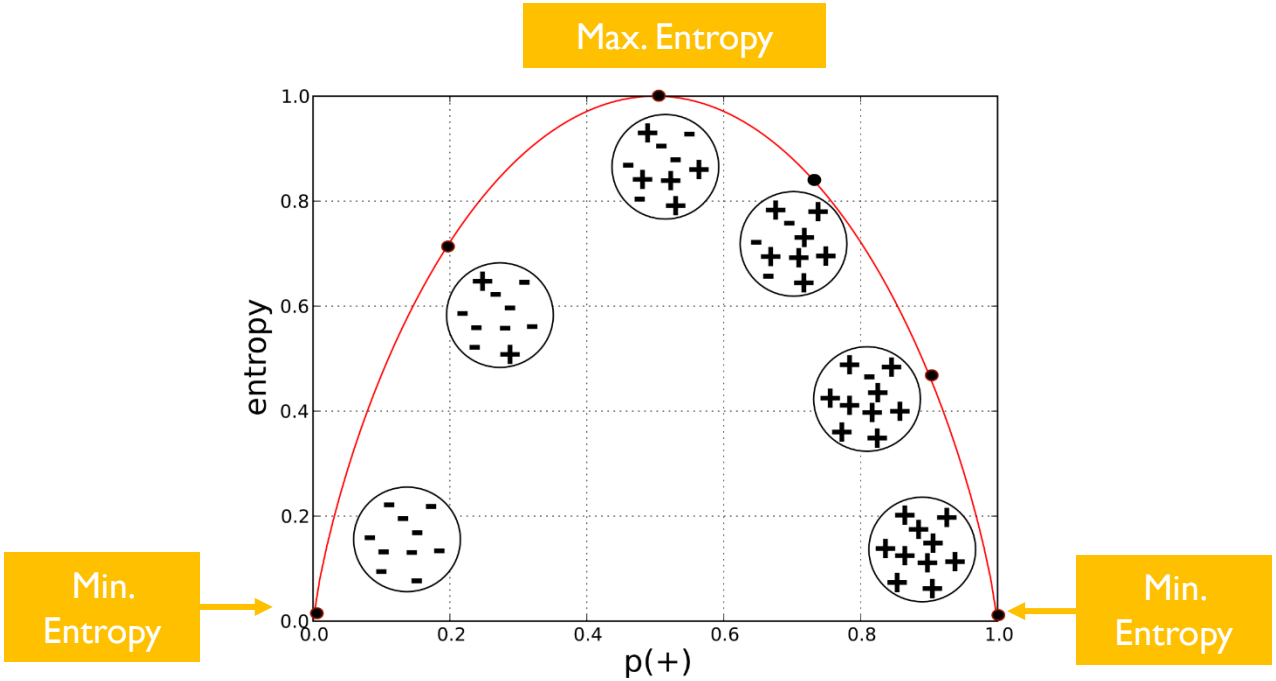
- Entropy is ranging from
  - 0 at minimum disorder (all members the same), to
  - 1 at maximum disorder (classes are equally mixed)

Entropy for a Two-class Group

- In a binary classification task where the target attribute has only two classes (n = 2): + and –.

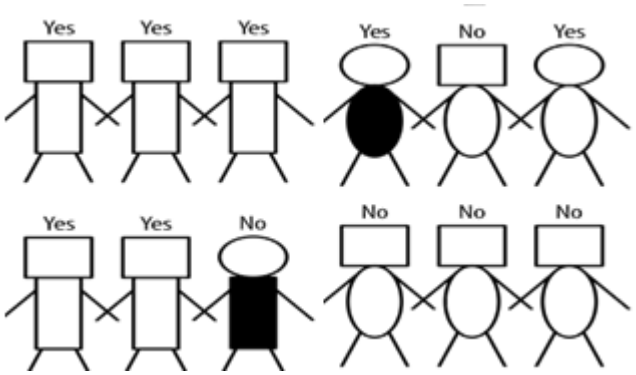
$$\boldsymbol{entropy} = -p_1 \log_2(p_1) - p_2 \log_2(p_2)$$

  - p_1 and p_2 are the proportions of + and – class in a group

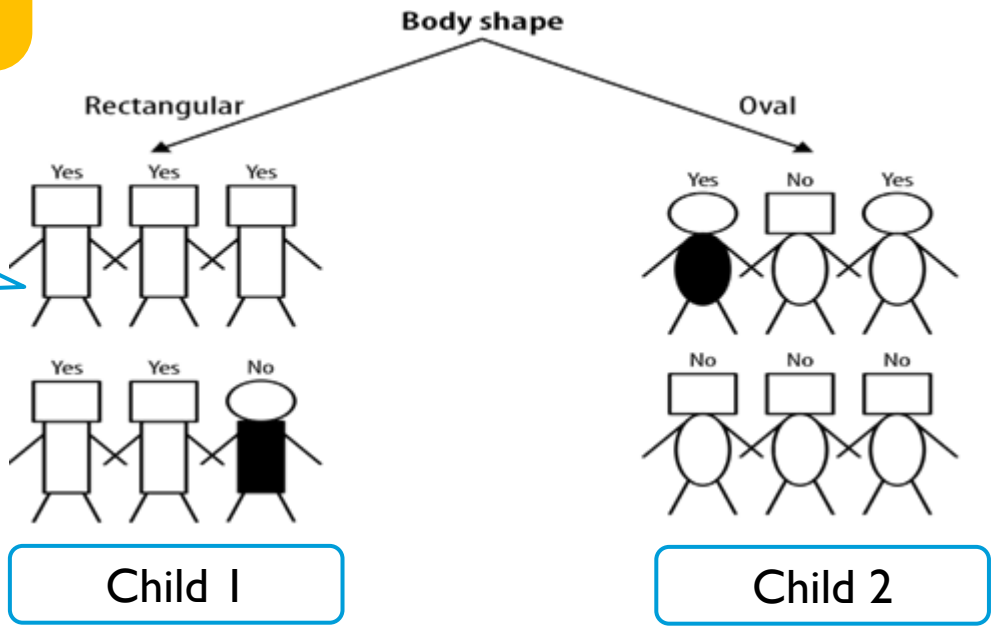


Entropy for Parent and Child Nodes

Parent
No segmentation yet



Entropy(p)
$(-7/12)*\log2(7/12) - (5/12)*\log2(5/12) = 0.98$



2 Children
One feature is used for splitting.

Entropy(c)
$\text{Entropy}(c_1) = 0.65$
$\text{Entropy}(c_2) = 0.92$

Information Gain

- **Information gain** measures the change of entropy due to the use of any feature in a split.

$$Information\ Gain\ =\ entropy(parent) - entropy(childeren)$$

- Each child node (c_i) contains $p(c_i)$ of the total instances.
  - If there are m children after splitting

$$p(c_1) + p(c_2) + \cdots + p(c_m) = 1$$

  - **Entropy for all children**: sum of all child nodes' entropy weighted by the proportion of instances in each node.

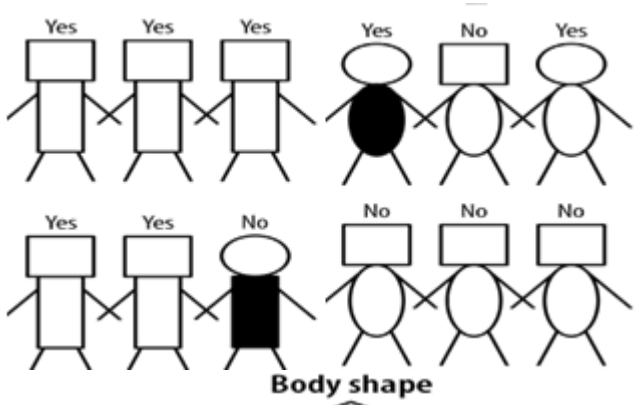
$$\boldsymbol{entropy}(childeren) = \sum_{i=1}^{m} p(c_i)\ \times\ entropy(c_i)$$

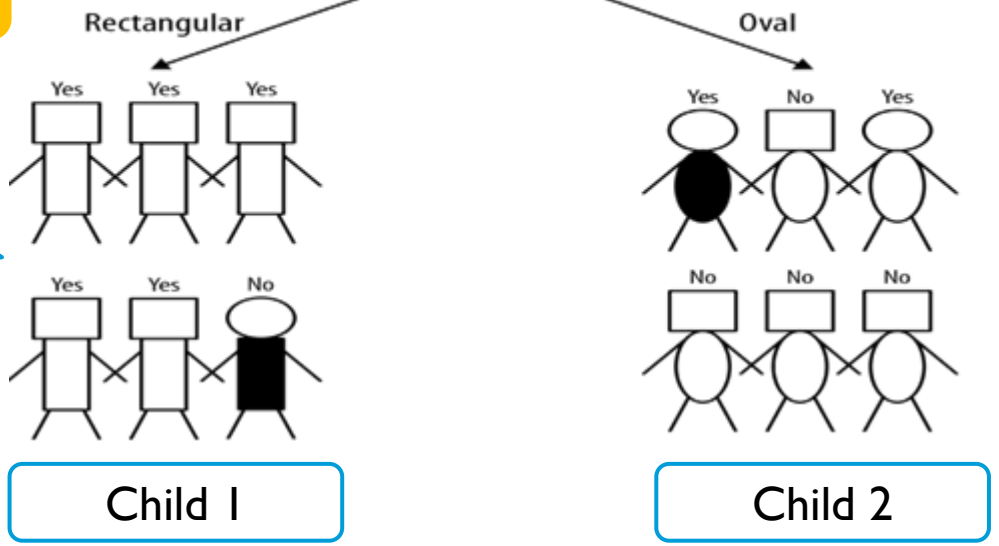
Information Gain of *Body Shape*

Entropy(p)

(-7/12)*log2(7/12) – (5/12)*log2(5/12) = 0.98

$IG\ (body\ shape)\ = 0.2$





Entropy(c)

Entropy(c_1) = 0.65

Entropy(c_2) = 0.92

Entropy(c) = (1/2) * 0.65 + (1/2) * 0.92 = 0.78

Recursive Feature Selection

- The child node in the first split (e.g., rectangular body shape) now become the parent node in second splitting.
  - Need to calculate and compare the information gain for features recursively in splitting each subgroup.

Parent

Entropy(p)

(-5/6)*log2(5/6) – (1/6)*log2(1/6) = 0.65

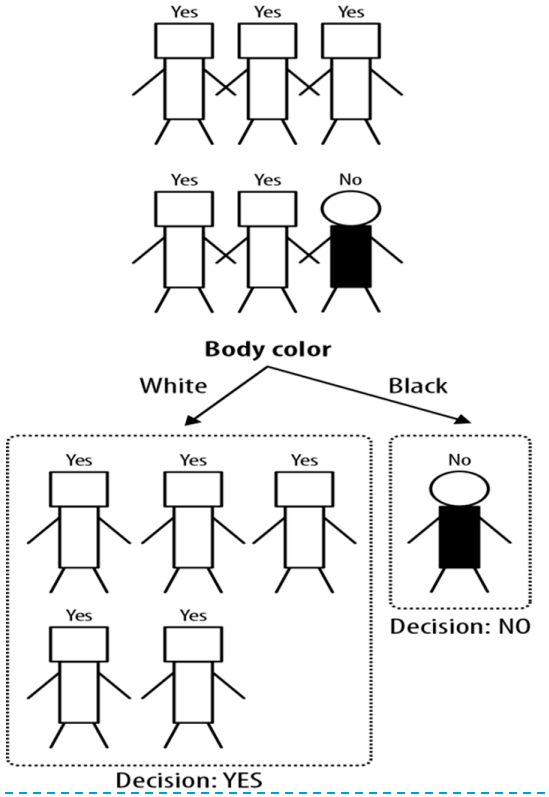
2 Children

Segmented by Body Color

Entropy(c) = 0

Entropy(c_1) = 0 & Entropy(c_2) = 0

$$IG\ (Body\ Color)\ = 0.65$$



Compute Information Gain for *Balance*

- **Target**:
  - write-off (★) and not write-off (●)
- **Feature**:
  - Outstanding balance (<50K or otherwise)
    - $c_1 = Balance < 50K$
    - $c_2 = Balance \geq 50K$

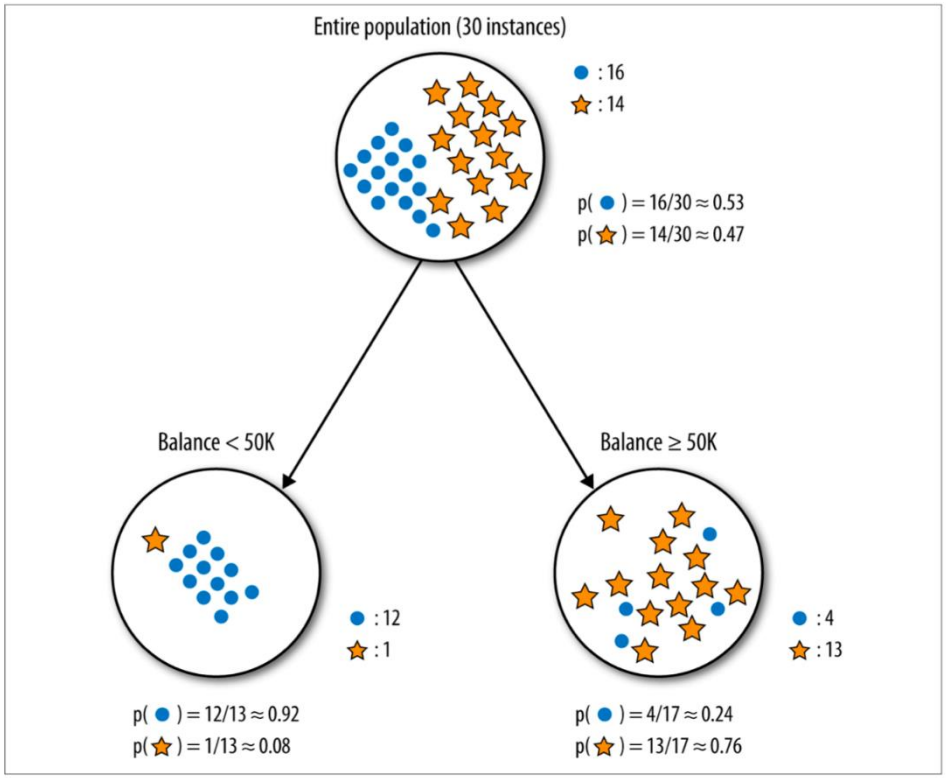
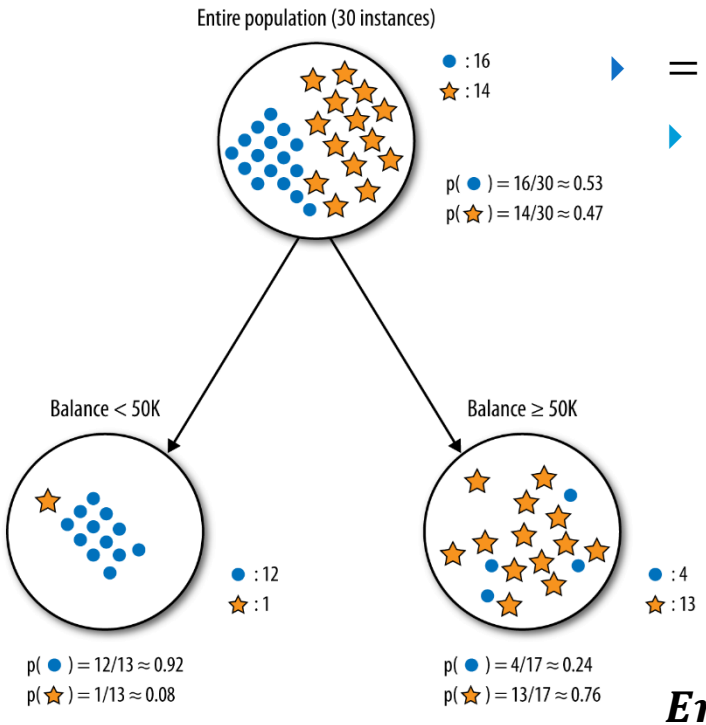


Figure 3-4. Splitting the "write-off" sample into two segments, based on splitting the Balance attribute (account balance) at 50K.

Step 1 – Entropy of Parent & Child Nodes



$$\boldsymbol{Entropy(parent)}$$

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

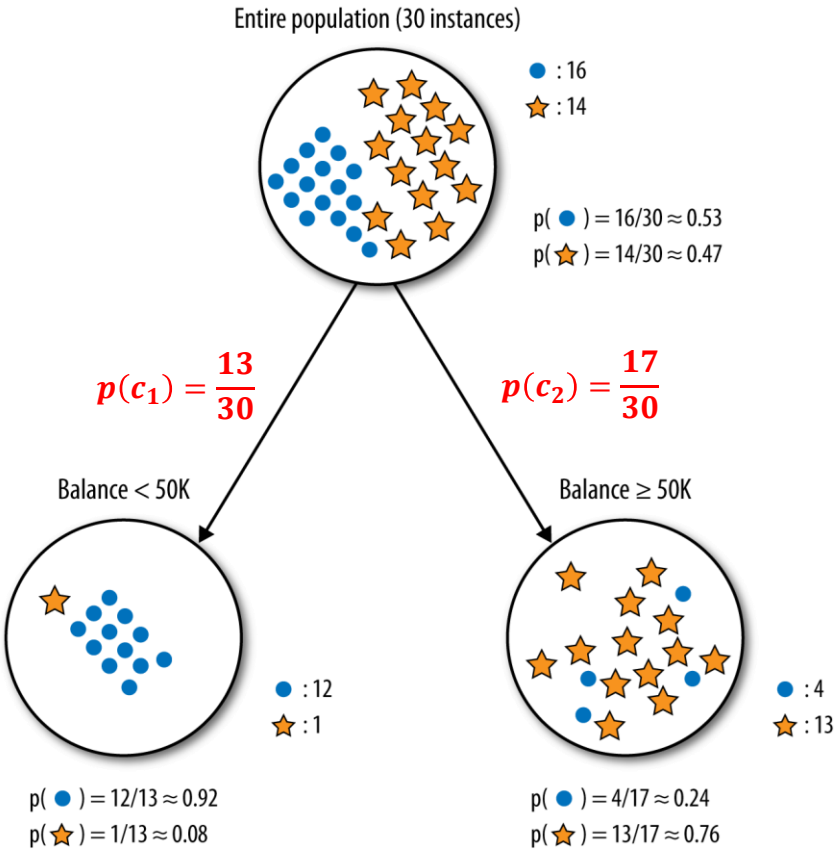
$$\boldsymbol{Entropy(c_1)}$$

- $= -\frac{12}{13} \times log_2\left(\frac{12}{13}\right) - \frac{1}{13} \times log_2\left(\frac{1}{13}\right)$
- $= 0.39$

$$\boldsymbol{Entropy(c_2)}$$

- $= -\frac{4}{17} \times log_2\left(\frac{4}{17}\right) - \frac{13}{17} \times log_2\left(\frac{13}{17}\right)$
- $= 0.79$

Step 2 – Weighted Entropy of the Children



Entropy of each child

- $Entropy(c_1) = 0.39$
- $Entropy(c_2) = 0.79$

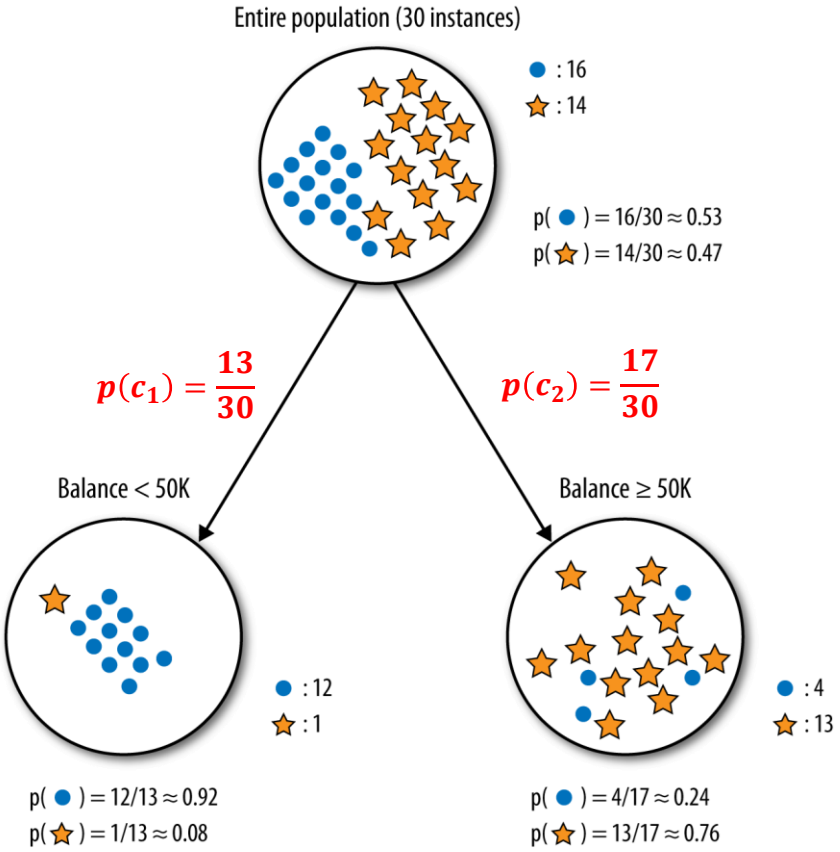
Proportion of instances in each child

- $p(c_1) = \frac{13}{30}$
- $p(c_2) = \frac{17}{30}$

Entropy (children)

- $= \frac{13}{30} \times 0.39 + \frac{17}{30} \times 0.79$
- $= 0.62$

Step 3 – Information Gain of *Balance*



$$Parent\ entropy = 0.99$$

$$Children\ entropy = 0.62$$

$$\boldsymbol{IG(Balance \geq 50K)}$$

- $\boldsymbol{= 0.99 - 0.62}$
- $\boldsymbol{= 0.37}$

Compute Information Gain of *Residence*

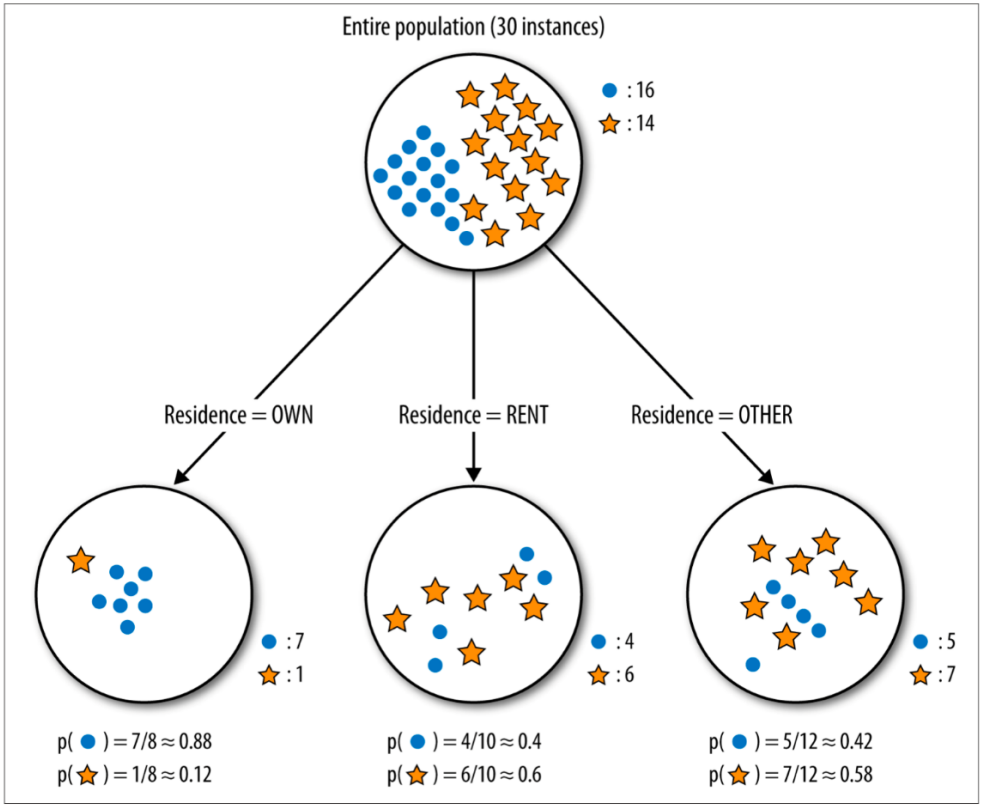


Figure 3-5. A classification tree split on the three-valued Residence attribute.

- *Entropy* (*parent*) ≈ 0.99
- *Entropy* (Residence=OWN) ≈ 0.54
- *Entropy* (Residence=RENT) ≈ 0.97
- *Entropy* (Residence=OTHER) ≈ 0.98
- Entropy(*children*) ≈ 0.86

…

***IG (Residence)* ≈ 0.13**

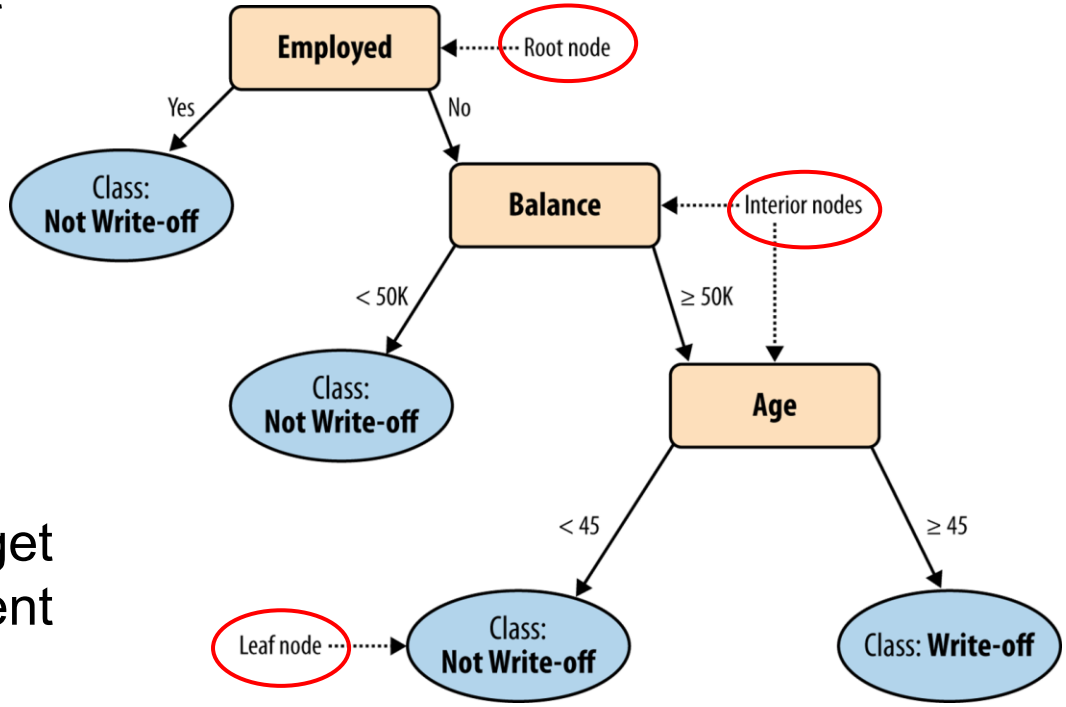
Conclusion:
Balance is more informative than *Residence* in predicting loan write-off.

Train a Tree Model

- How do we build a classification tree?
  - **Divide-and-conquer** approach
    - Take each subgroup and recursively apply feature/attribute selection to find the best one for segmentation.
  - When shall we stop?
    - The leaf nodes are pure.
    - There is no more features for splitting.
    - Avoid over-fitting by controlling the size of the tree.

Tree Model: a Set of Rules

- **Root node**: the starting point
- **Interior node**: middle parts connecting nodes above and below
- **Terminal (leaf) node**: the classification outcome for target variable, each leaf is a segment
- **Branches** are defined by distinct values of the features

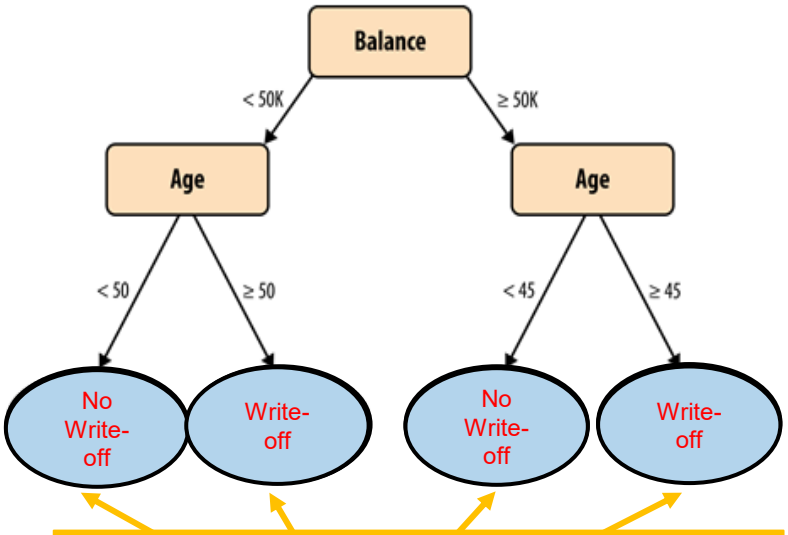


Unemployed clients with high outstanding balance and age above 45 tend to have their loans written off.

Class Prediction

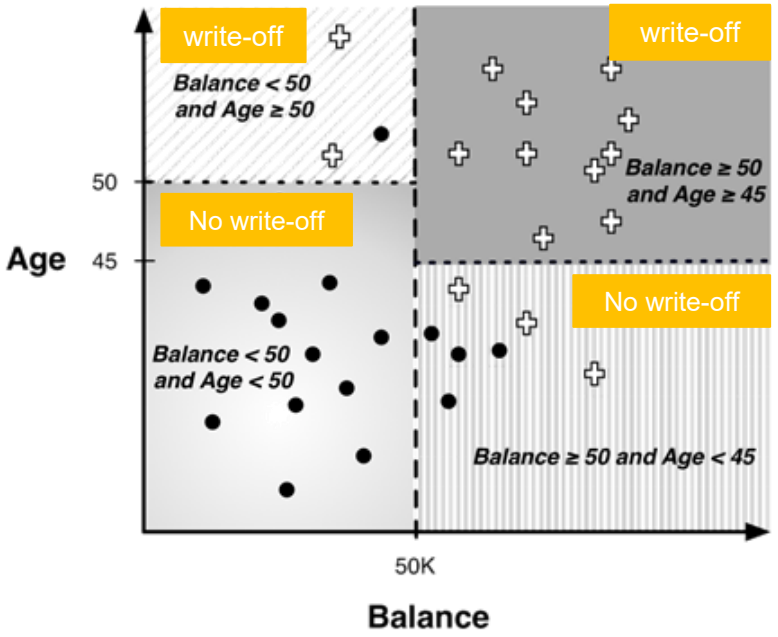
- With a tree model trained, the segmentation result can be visualized in an **instance space**:
  - ✚ write-off • no write-off
  - Lines separating the regions are decision boundaries

A classification tree



Predicted class (target value) is the majority class in each leaf node

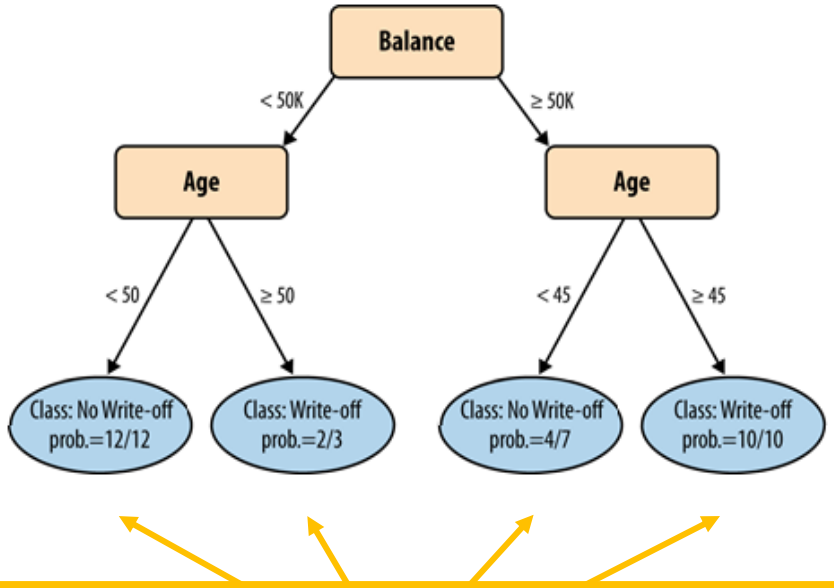
Instance space segmented by the classification tree



Class Probability Estimation

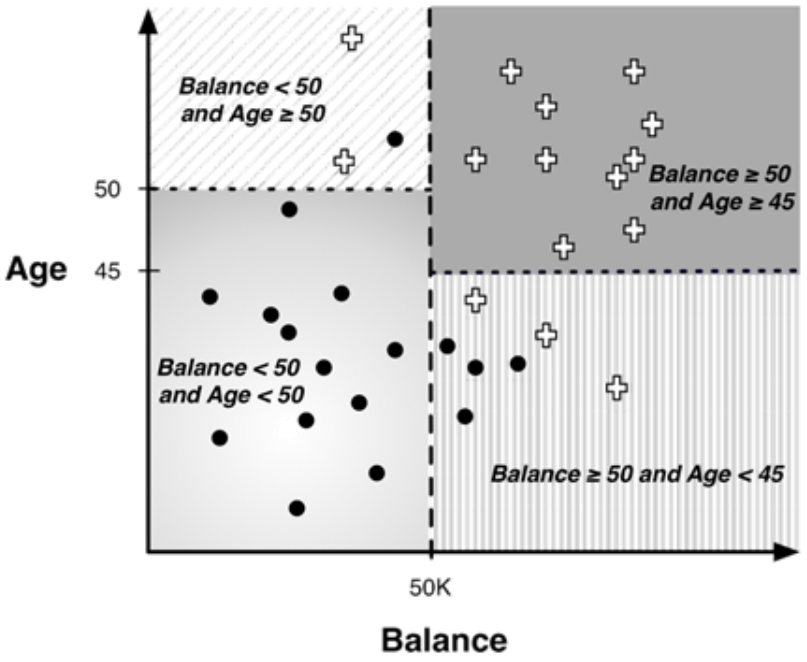
- How do we estimate the class probability?
  - The segmenting result in some leaf nodes are impure.

A classification tree



Estimated class probability is the probability of the majority class in each leaf node

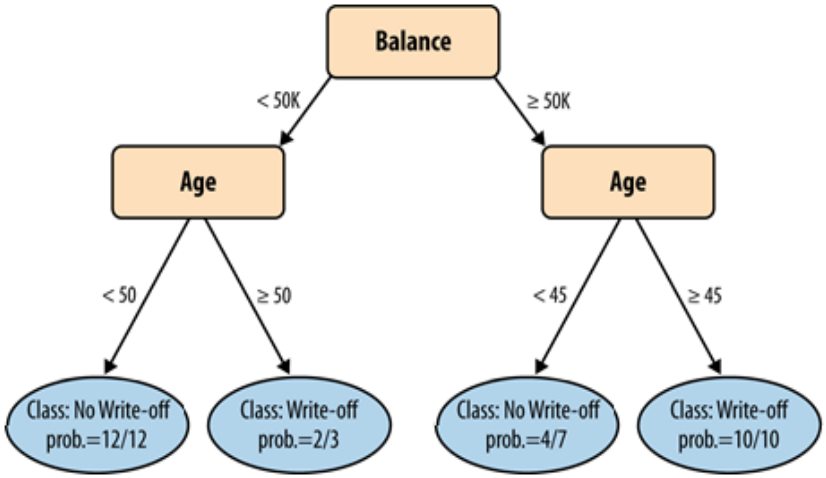
Instance space segmented by the classification tree



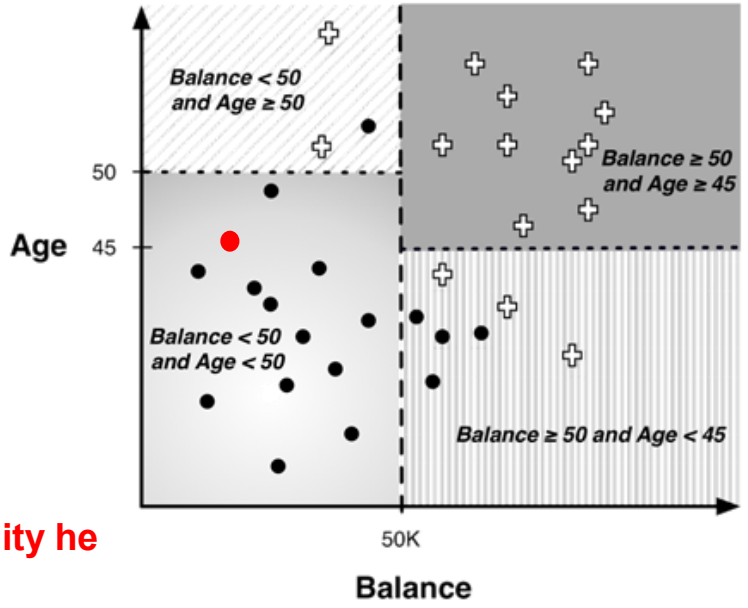
Predict a New Customer's Loan Status

- For the loan write-off case, if a new customer:
  - Balance < 50K and age < 50
- What is the predicted class for this customer?
  - How likely will he default on the loan (thus be written off)?

A classification tree



Instance space segmented by the classification tree



Predicted Class: No Write-off

Can we say that the probability he will NOT default is 100%?

Two Probability Estimation Methods

- Assume a leaf node with n positive, m negative instances:

Frequency-based estimate

- Probability that a new instance is positive is

$$\frac{n}{n+m}$$

Laplace correction

- Smoothed version of the frequency-based estimate

$$\frac{n+1}{n+m+2}$$

- Why a Laplace corrected probability is necessary?
  - With a bigger data size, we are more confident with the result.

(n, m)	$(2, 0)$	$(20, 0)$
Frequency	$\frac{2}{2} = 1$	$\frac{20}{20} = 1$
Laplace	$\frac{3}{4} = 0.75$	$\frac{21}{22} = 0.95$

Effect of Laplace Correction

- Consider two groups where 2/3 instances being positive:
  - A: $(n, m) = (4, 2)$, total 6
  - B: $(n, m) = (18, 9)$, total 27
- Frequency-based probability of positive class
  - Solid constant line $= 2/3$
- Laplace corrected probability
  - Dashed curve line
    - A: $p_{(c)} = \frac{4+1}{6+2} = 0.6250$
    - B: $p_{(c)} = \frac{18+1}{27+2} = 0.6552$

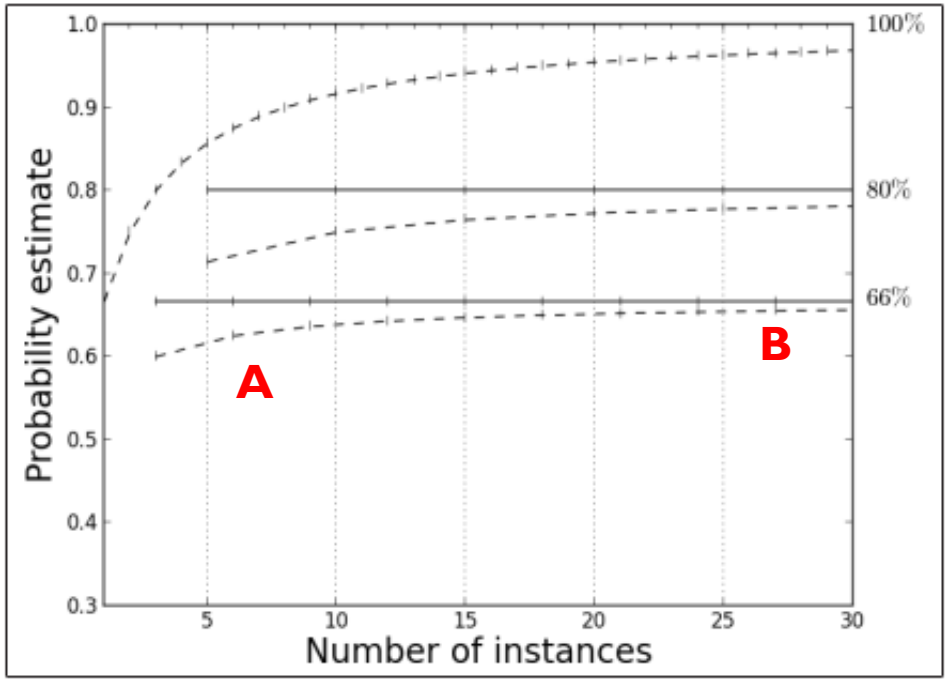


Figure 3-16. The effect of Laplace smoothing on probability estimation for several instance ratios.

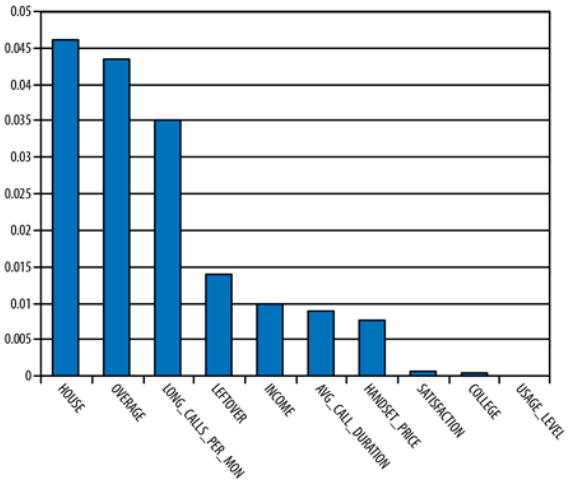
As the number of total instances increases, Laplace-corrected probability converges to the frequency-based estimate.

Compare Importance of Features

- Predicting **customer churn** based on their features.

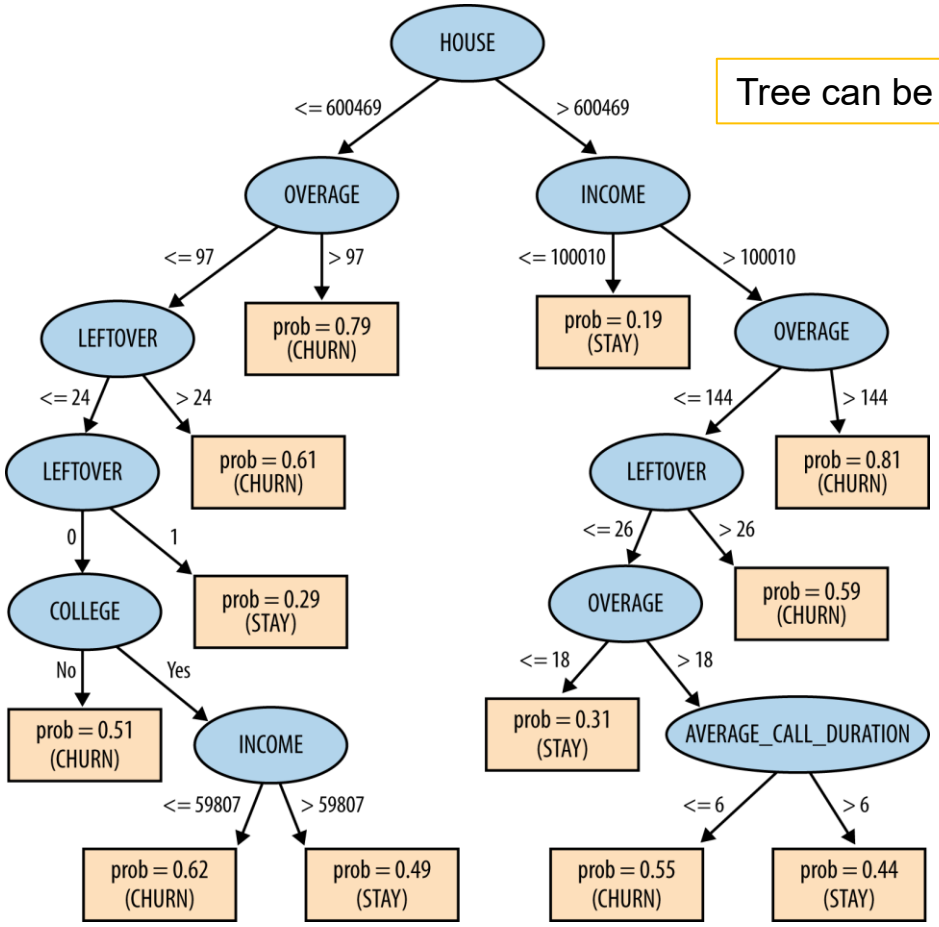
Variable	Explanation
COLLEGE	Is the customer college educated?
INCOME	Annual income
OVERAGE	Average overcharges per month
LEFTOVER	Average number of leftover minutes per month
HOUSE	Estimated value of dwelling (from census tract)
HANDSET_PRICE	Cost of phone
LONG_CALLS_PER_MONTH	Average number of long 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)?

- Here is a list of features ranked by their Information Gain, evaluated on entire dataset.
  - Can we use these features one by one according the order?



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

Decision Tree



Tree can be asymmetric.

Same continuous features can be used more than once in the same branch.

Can categorical feature be used more than once in the same branch? No

There can be misclassification in the leaf nodes.

Leaf nodes from the same feature aims to have different class values.

Decision Trees

Pros:

- Easily visualized and interpreted.
- No feature normalization or scaling needed.
- Work well with datasets with a mixture of variable types (continuous, categorical).

Cons:

- Even after tuning, decision trees can often still overfit.
- Usually need an ensemble of trees for better generalization performance.