Institut Informatika & Bisnis DARMAJAYA
Yayasan Alfian Husin

SAINS DATA
Institut Informatika & Bisnis Darmajaya

Kampus Merdeka INDONESIA JAYA

MERDEKA BELAJAR

DATA SCIENCE DARMAJAYA
"YOUR BEST FUTURE IN DATA"

MEETING: [5]

INTRODUCTION TO PREDICTIVE MODELING : FROM CORRELATION TO SUPERVISED SEGMENTATION

BY: HENDRA KURNIAWAN

[12-09-2023] | [KODE MK: BDG23407 | MATAKULIAH: DATA SCIENCE AND BUSINESS ANALYTICS | SKS: 2/2 | VERSI: 01

- Fundamental concepts:
  - *Identifying informative attributes; Segmenting data by progressive attribute selection.*
- Exemplary techniques:
  - *Finding correlations; Attribute/variable selection; Tree induction.*

- The previous chapters discussed models and modeling at a high level. This chapter delves into one of the main topics of data mining: predictive modeling.
- Following our example of data mining for churn prediction from the first section, we will begin by thinking of predictive modeling as *supervised* segmentation.

- In the process of discussing supervised segmentation, we introduce one of the fundamental ideas of data mining: finding or selecting important, informative variables or "attributes" of the entities described by the data.
- What exactly it means to be "informative" varies among applications, but generally, *information is a quantity that reduces uncertainty about something.*
- Finding informative attributes also is the basis for a widely used predictive modeling technique called *tree induction*, which we will introduce toward the end of this chapter as an application of this fundamental concept.

- By the end of this chapter we will have achieved an understanding of:
  - the basic concepts of predictive modeling;
  - the fundamental notion of finding informative attributes, along with one particular, illustrative technique for doing so;
  - the notion of tree-structured models;
  - and a basic understanding of the process for extracting tree structured models from a dataset—performing supervised segmentation.

Outline

- **Models, Induction, and Prediction**
- Supervised Segmentation
  - Selecting Informative Attributes
  - Example: Attribute Selection with Information Gain
  - Supervised Segmentation with Tree-Structured Models

Models, Induction, and Prediction

- 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.
- For example, a map is a model of the physical world. It abstracts away a tremendous amount of information that the mapmaker deemed irrelevant for its purpose. It preserves, and sometimes further simplifies, the relevant information.

Models, Induction, and Prediction

- Various professions have well-known model types: an architectural blueprint, an engineering prototype, the Black-Scholes model of option pricing(選擇權定價模式), and so on.
- Each of these abstracts away details that are not relevant to their main purpose and keeps those that are.

Models, Induction, and Prediction

- In data science, a predictive model is a formula for estimating the unknown value of interest: the target.
- The formula could be mathematical, or it could be a logical statement such as a rule. Often it is a hybrid of the two.
- Given our division of supervised data mining into classification and regression, we will consider classification models (and class-probability estimation models) and regression models.

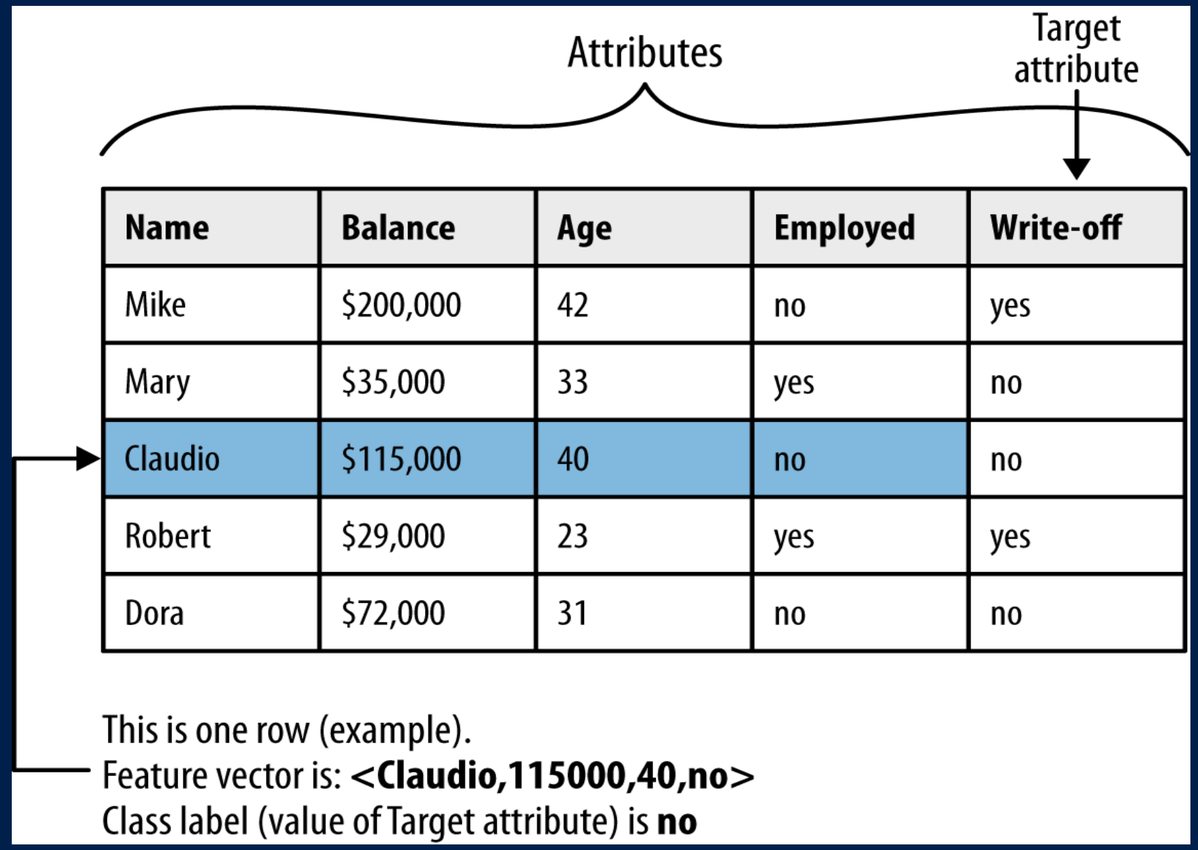
Terminology : Prediction

- In common usage, prediction means to forecast a future event.
- In data science, prediction more generally means *to estimate an unknown value*. This value could be something in the future (in common usage, true prediction), but it could also be something in the present or in the past.
- Indeed, since data mining usually deals with historical data, models very often are built and tested using events from the past.
- The key is that the model is intended to be used to estimate an unknown value.

Models, Induction, and Prediction

- Supervised learning is model creation where the model describes a relationship between a set of selected variables (*attributes* or *features*)
- Predefined variable called the *target* variable. The model estimates the value of the target variable as a function (possibly a probabilistic function) of the features.
- So, for our churn-prediction problem we would like to build a model of the propensity to churn as a function of customer account attributes, such as age, income, length with the company, number of calls to customer service, overage charges, customer demographics, data usage, and others.

Models, Induction, and Prediction



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

Many Names for the Same Things

- The principles and techniques of data science historically have been studied in several different fields, including machine learning, pattern recognition (模式識別), statistics, databases, and others.
- As a result there often are several different names for the same things. We
- typically will refer to a *dataset*, whose form usually is the same as a *table* of a database or a *worksheet* of a spreadsheet. A dataset contains a set of *examples* or *instances*. An instance also is referred to as a *row* of a database table or sometimes a *case* in statistics.

Many Names for the Same Things

- The features (table columns) have many different names as well. Statisticians speak of *independent variables* or *predictors* as the attributes supplied as input. In operations research you may also hear *explanatory variable*.
- The target variable, whose values are to be predicted, is commonly called the *dependent variable* in statistics.

Models, Induction, and Prediction

- The creation of models from data is known as model induction.
- The procedure that creates the model from the data is called the induction algorithm or learner. Most inductive procedures have variants that induce models both for classification and for regression.

Terminology : Induction and deduction

Induction can be contrasted with deduction. Deduction starts with general rules and specific facts, and creates other specific facts from them. The use of our models can be considered a procedure of (probabilistic) deduction. We will get to this shortly. The input data for the induction algorithm, used for inducing the model, are called the training data. they are called labeled data because the value for the target variable (the label) is known.

Outline

- Models, Induction, and Prediction
- **Supervised Segmentation**
  - Selecting Informative Attributes
  - Example: Attribute Selection with Information Gain
  - Supervised Segmentation with Tree-Structured Models

Supervised Segmentation

- A predictive model focuses on estimating the value of some particular target variable of interest.
- To try to segment the population into subgroups that have different values for the target variable .
- Segmentation may at the same time provide a human-understandable set of segmentation patterns.

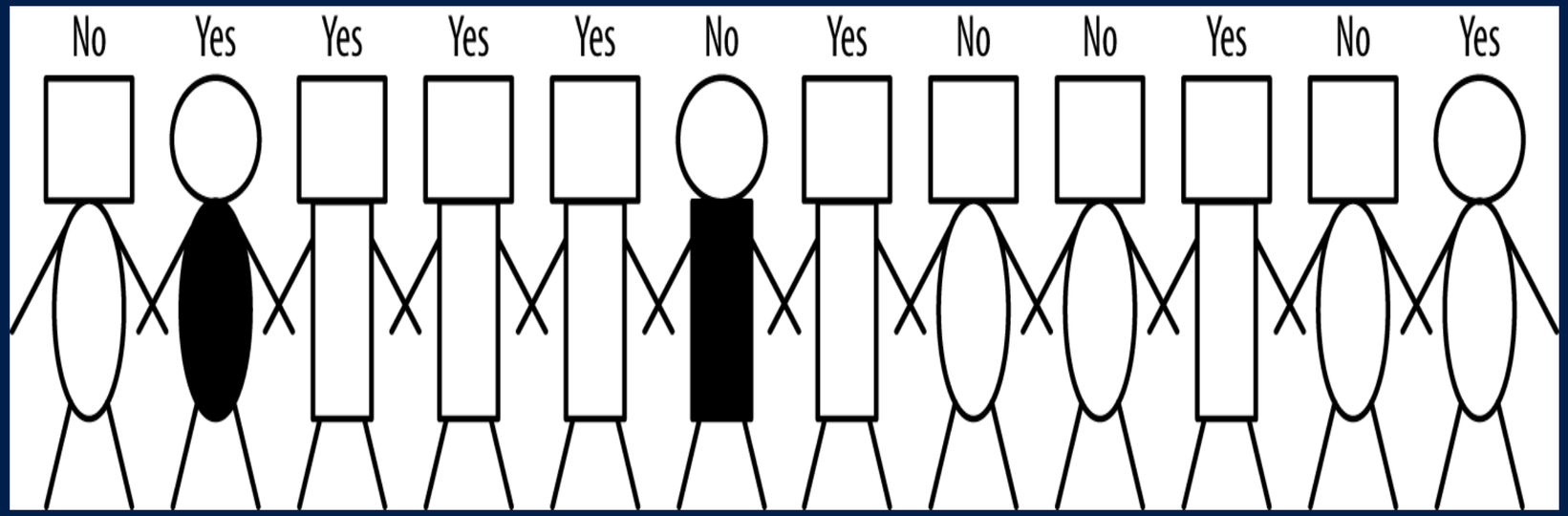
Supervised Segmentation

- We might like to rank the variables by how good they are at predicting the value of the target.
- In our example, what variable gives us the most information about the future churn rate of the population? Being a professional? Age? Place of residence? Income? Number of complaints to customer service? Amount of overage charges?
- We now will look carefully into one useful way to select informative variables, and then later will show how this technique can be used repeatedly to build a supervised segmentation.

Outline

- Models, Induction, and Prediction
- Supervised Segmentation
  - **Selecting Informative Attributes**
  - Example: Attribute Selection with Information Gain
  - Supervised Segmentation with Tree-Structured Models

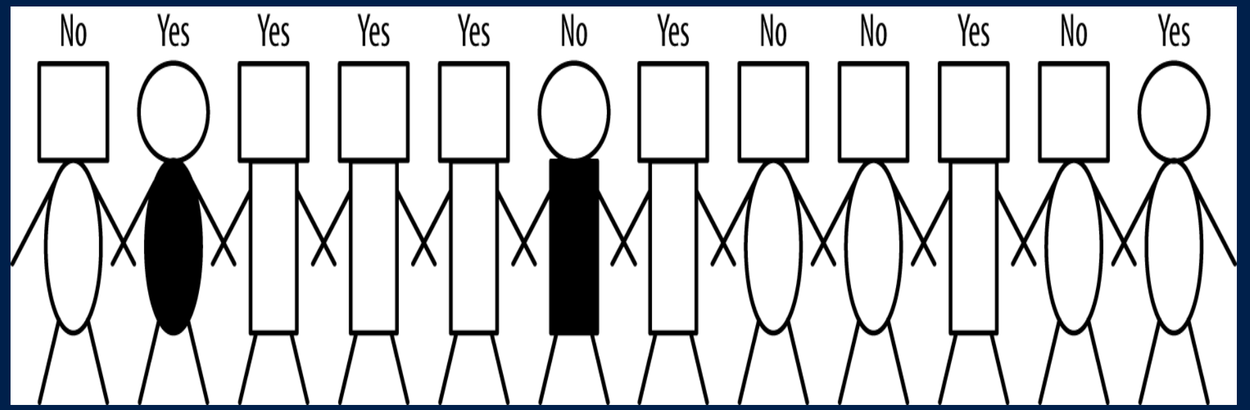
Selecting Informative Attributes



- The label over each head represents the value of the target variable (write-off or not).
- Colors and shapes represent different predictor attributes.

Selecting Informative Attributes

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



Selecting Informative Attributes

- So let's ask ourselves:
  - which of the attributes would be best to segment these people into groups, in a way that will distinguish write-offs from non-write-offs?
- Technically, we would like the resulting groups to be as *pure* as possible. By pure we mean *homogeneous with respect to the target variable*. If every member of a group has the same value for the target, then the group is pure. If there is at least one member of the group that has a different value for the target variable than the rest of the group, then the group is impure.
- Unfortunately, in real data we seldom expect to find a variable that will make the segments pure.

Selecting Informative Attributes

- *Purity measure*.
- The most common splitting criterion is called *information gain*, and it is based on a purity measure called *entropy*.
- Both concepts were invented by one of the pioneers of information theory, Claude Shannon, in his seminal work in the field (Shannon, 1948).

Selecting Informative Attributes

- 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 these properties of interest.

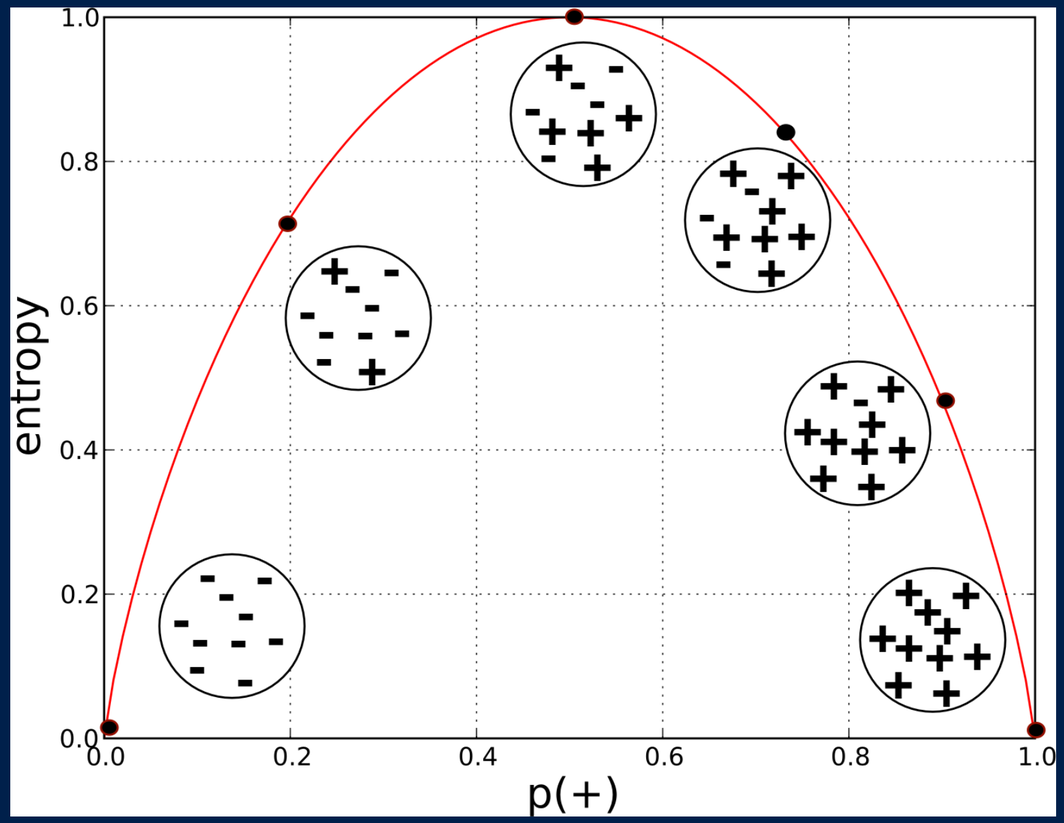
Selecting Informative Attributes

Equation 3-1. Entropy

$$entropy = - p1 \log(p1) - p2 \log(p2) - \ldots$$

Each pi is the probability (the relative percentage) of property i within the set, ranging from pi = 1 when all members of the set have property i, and pi = 0 when no members of the set have property i. The … simply indicates that there may be more than just two properties (and for the technically minded, the logarithm is generally taken as base 2).

Selecting Informative Attributes



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

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

entropy(S)

$= - 0.7 \times \log_2 (0.7) - 0.3 \times \log_2 (0.3)$

$\approx - 0.7 \times - 0.51 - 0.3 \times - 1.74$

≈ 0.88

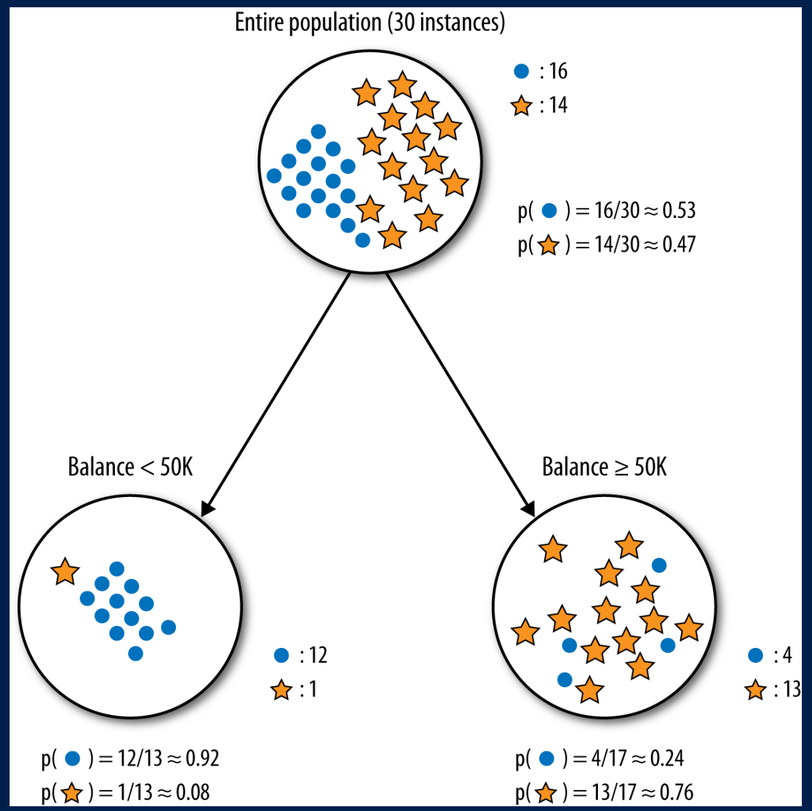
Selecting Informative Attributes

Equation Information gain :

IG(parent, children) = entropy(parent) – [p(c1) × entropy(c1) + p(c2) × entropy(c2) + . . .]

Notably, the entropy for each child (ci) is weighted by the proportion of instances belonging to that child, p(ci). This addresses directly our concern from above that splitting off a single example, and noticing that that set is pure, may not be as good as splitting the parent set into two nice large, relatively pure subsets, even if neither is pure.

Selecting Informative Attributes



$$entropy(parent)$$

$$= - p(\bullet) \times \log_2 p(\bullet) - p(\star) \times \log_2 p(\star)$$

$$\approx - 0.53 \times - 0.9 - 0.47 \times - 1.1$$

$$\approx 0.99 \text{ (very impure)}$$

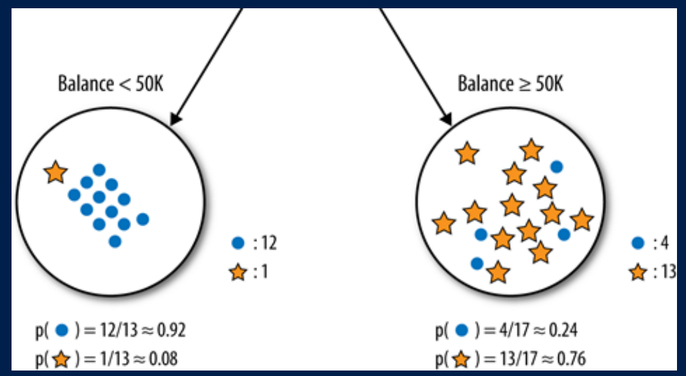
Selecting Informative Attributes

- The entropy of the *left* child is:

$$\text{entropy}(\text{Balance} < 50K) = -\,p(\bullet) \times \log_2 p(\bullet) - p(\star) \times \log_2 p(\star)$$
$$\approx -\,0.92 \times (-\,0.12) - 0.08 \times (-\,3.7)$$
$$\approx 0.39$$

- The entropy of the *right* child is:

$$\textit{entropy}(\textit{Balance} \geq 50K) = -\,p(\bullet) \times \log_2 p(\bullet) - p(\star) \times \log_2 p(\star)$$
$$\approx -\,0.24 \times (-\,2.1) - 0.76 \times (-\,0.39)$$
$$\approx 0.79$$



Selecting Informative Attributes

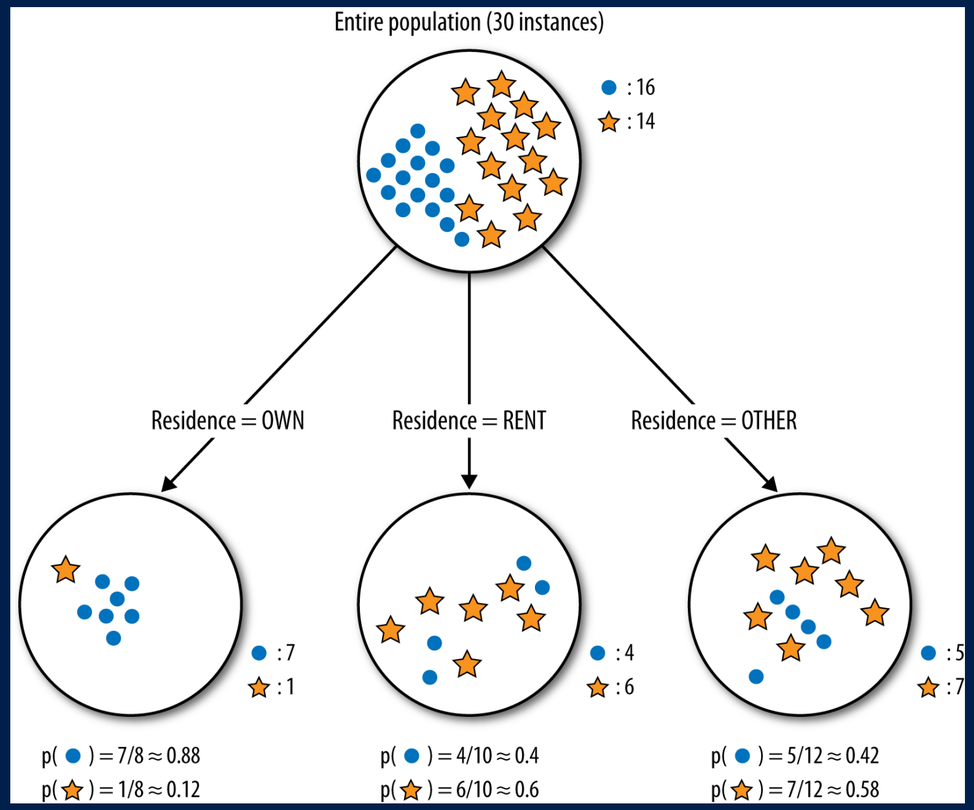
Information Gain

= *entropy*(*parent*) - (*p*(Balance < 50K) × *entropy*(Balance < 50K) + *p*(Balance ≥ 50K) × *entropy*(Balance ≥ 50K))

≈ 0.99 – (0.43 × 0.39 + 0.57 × 0.79)

≈ 0.37

Selecting Informative Attributes



entropy(*parent*) ≈ 0.99

entropy(Residence=OWN) ≈ 0.54

entropy(Residence=RENT) ≈ 0.97

entropy(Residence=OTHER) ≈ 0.98

Information Gain ≈ 0.13

Numeric variables

- We have not discussed what exactly to do if the attribute is numeric.
- Numeric variables can be “discretized” by choosing a split point(or many split points).
- For example, Income could be divided into two or more ranges. Information gain can be applied to evaluate the segmentation created by this discretization of the numeric attribute. We still are left with the question of how to choose the split point(s) for the numeric attribute.
- Conceptually, we can try all reasonable split points, and choose the one that gives the highest information gain.

Outline

- Models, Induction, and Prediction
- Supervised Segmentation
  - Selecting Informative Attributes
  - **Example: Attribute Selection with Information Gain**
  - Supervised Segmentation with Tree-Structured Models

Example: Attribute Selection with Information Gain

- For a dataset with instances described by attributes and a target variable.
- We can determine which attribute is the most informative with respect to estimating the value of the target variable.
- We also can rank a set of attributes by their informativeness, in particular by their information gain.

Example: Attribute Selection with Information Gain

- This is a classification problem because we have a target variable, called *edible?, with* two values *yes (edible) and no (poisonous), specifying our two classes.*
- *Each of the rows* in the training set has a value for this target variable. We will use information gain to answer the question: "Which single attribute is the most useful for distinguishing edible (edible?=Yes) mushrooms from poisonous (edible?=No) ones?"

Table 3-1. The attributes of the Mushroom dataset

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
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

GILL:蘑菇的菌褶

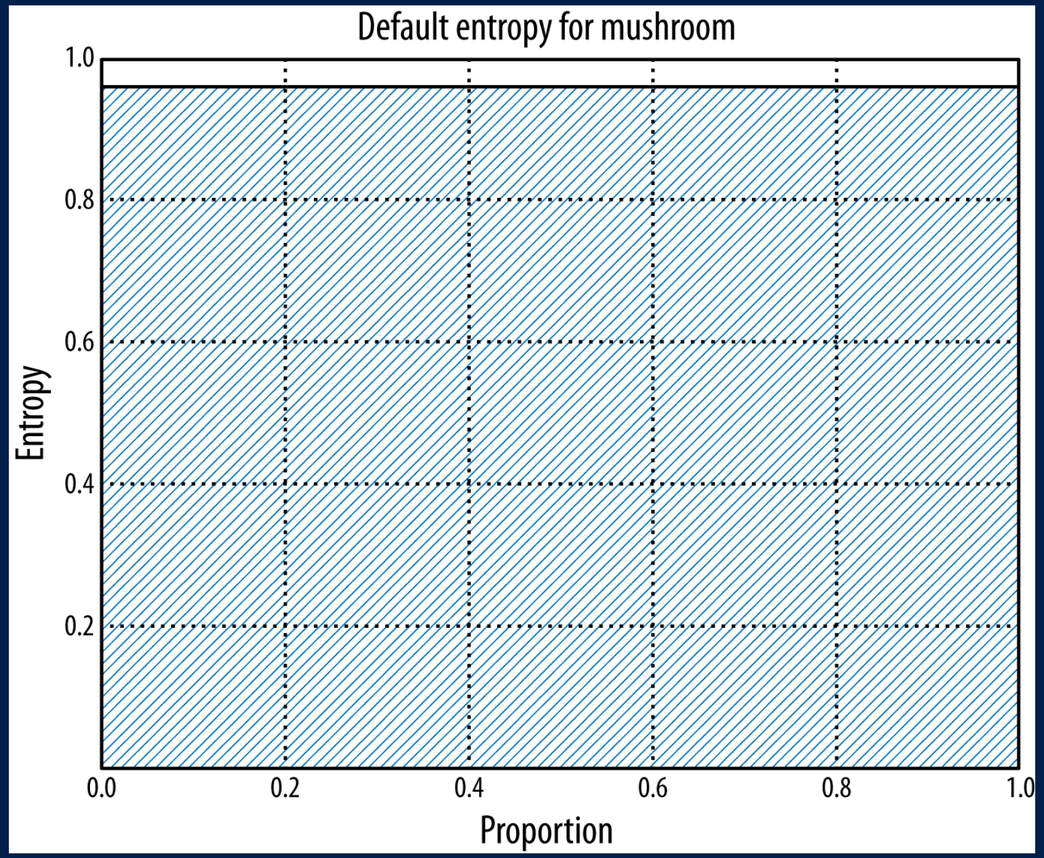
SPORE:孢子

UCI dataset

- We use 5,644 examples from the dataset,
- comprising 2,156 poisonous
- and 3,488 edible mushrooms
- 23 attributes

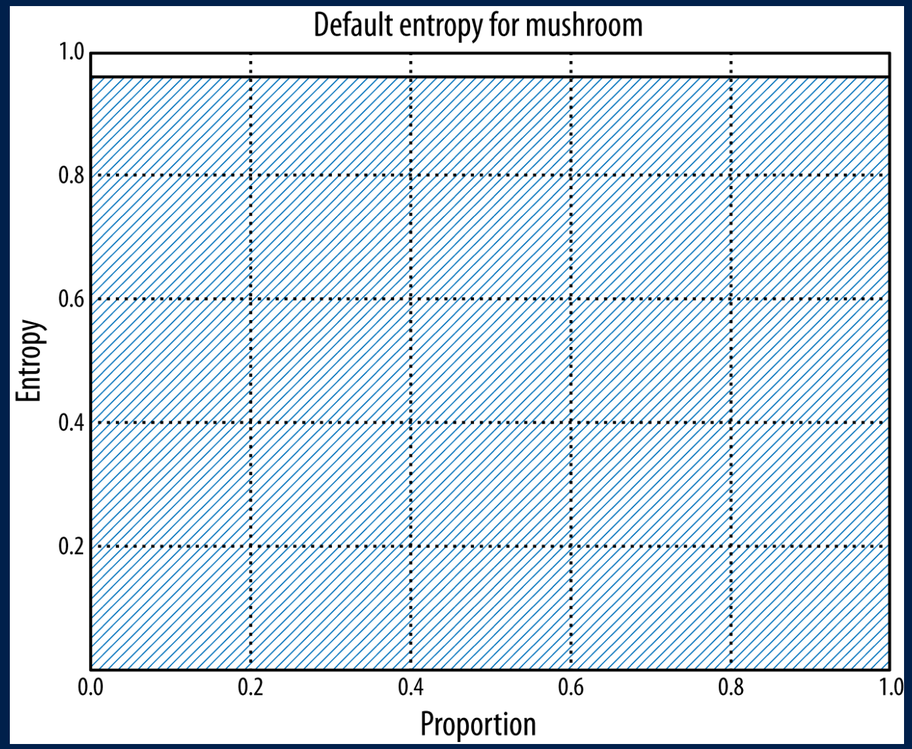
Example: Attribute Selection with Information Gain

- *Figure. Entropy chart for the entire Mushroom dataset. The entropy for the entire dataset is 0.96, so 96% of the area is shaded.*
- entropy(parent) ≈ 0.96



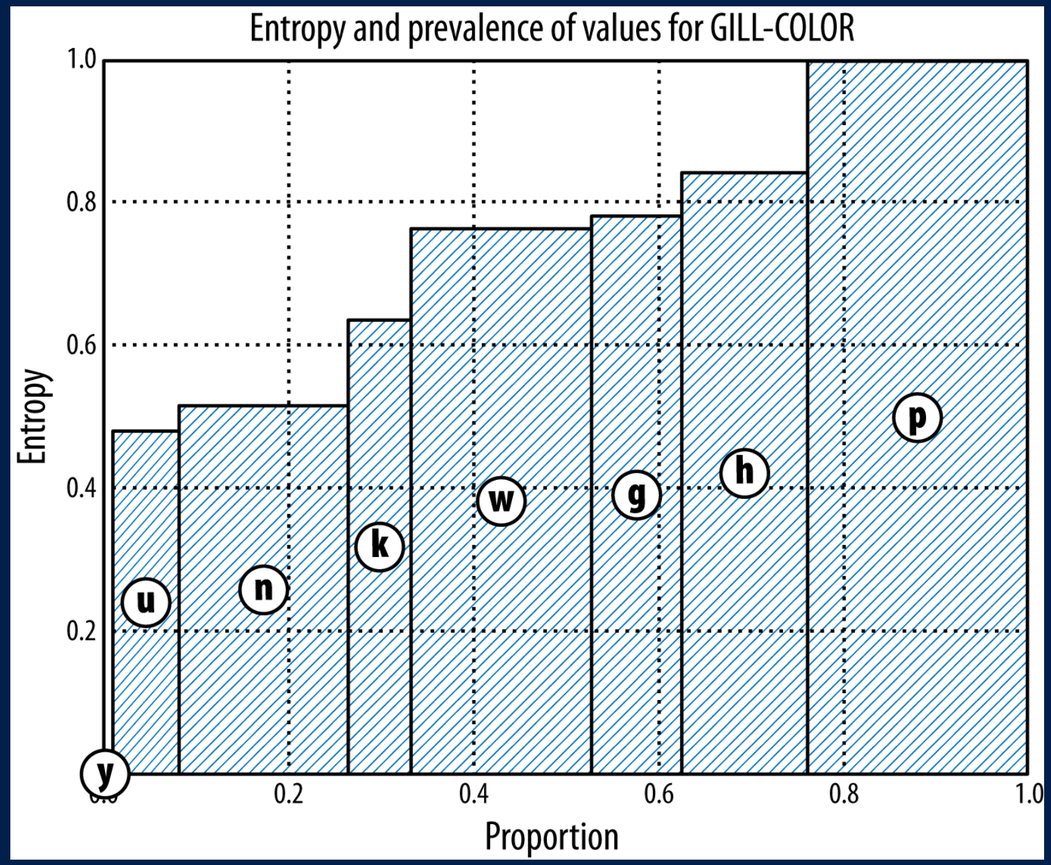
Example: Attribute Selection with Information Gain

- Figure. This is our starting entropy—any informative attribute should produce a new graph with less shaded area.



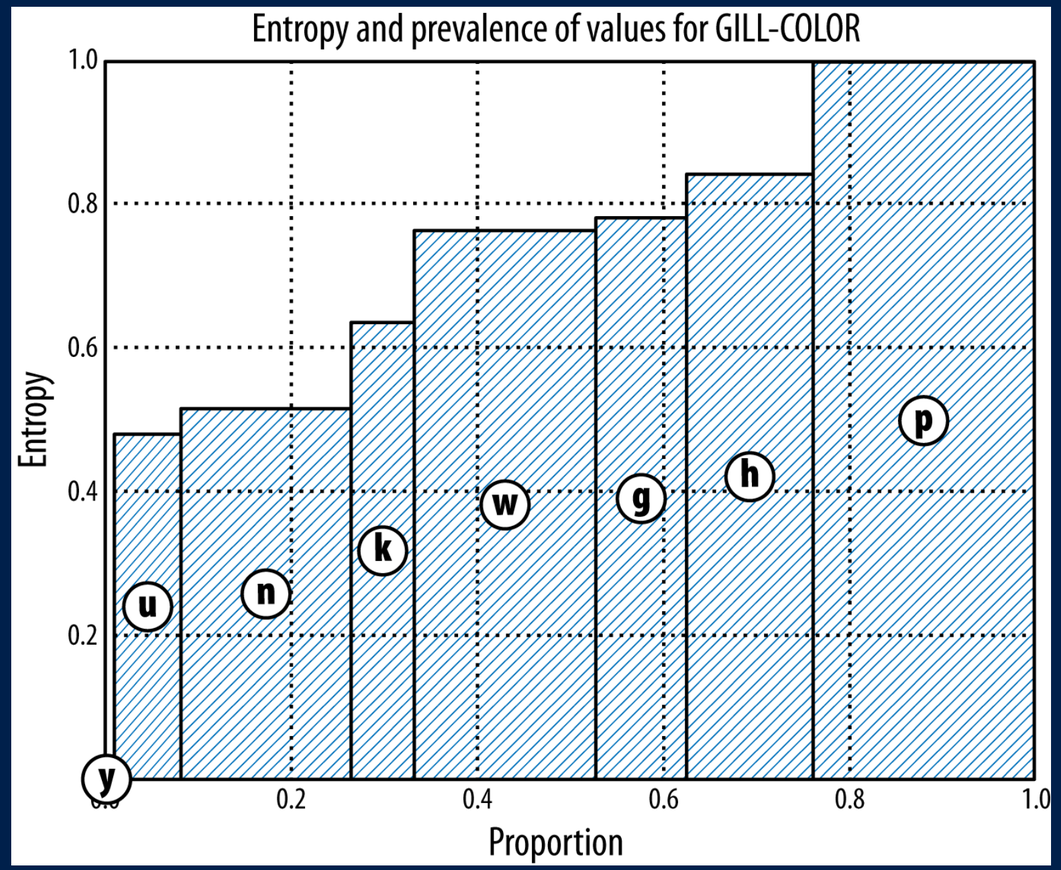
Example: Attribute Selection with Information Gain

- Figure. Entropy chart for the Mushroom dataset as split by GILL-COLOR, whose values are coded as y(yellow), u (purple), n (brown), and so on.



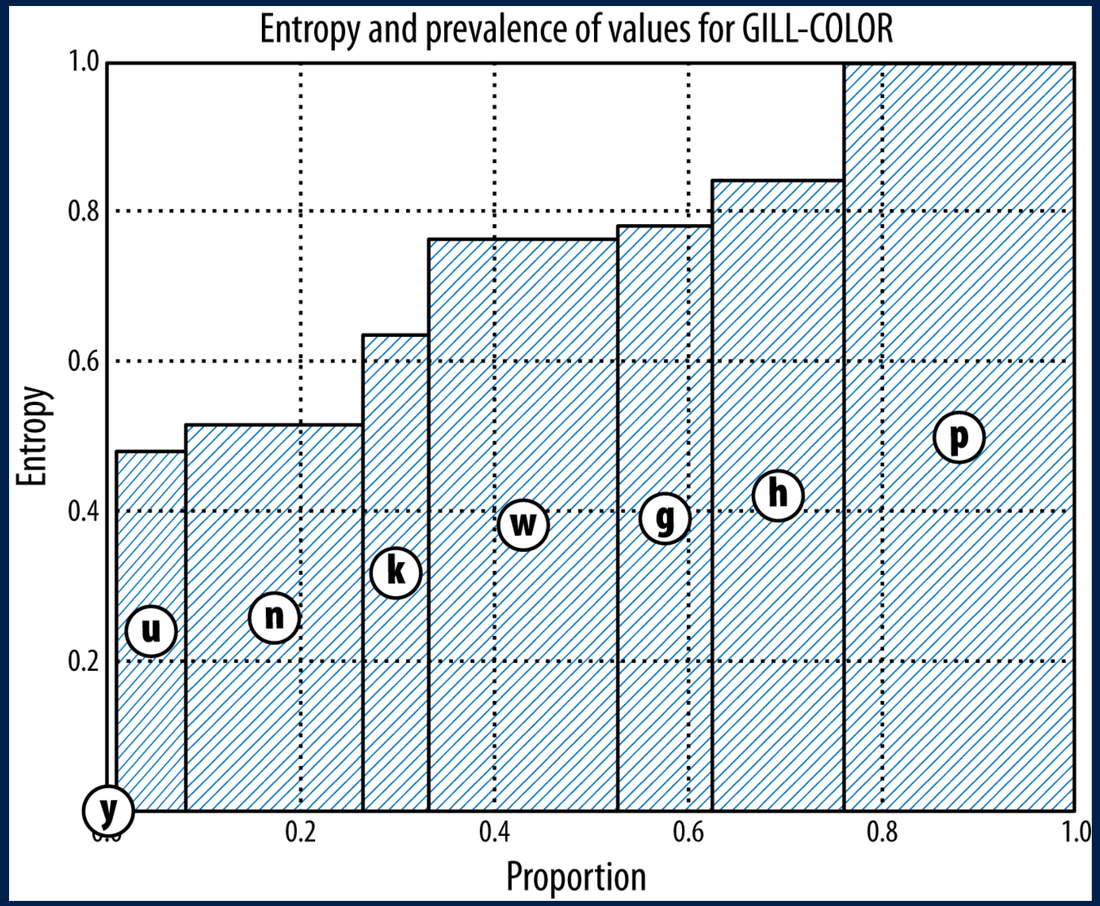
Example: Attribute Selection with Information Gain

- Figure 3-7. The width of each attribute represents what proportion of the dataset has that value, and the height is its entropy.



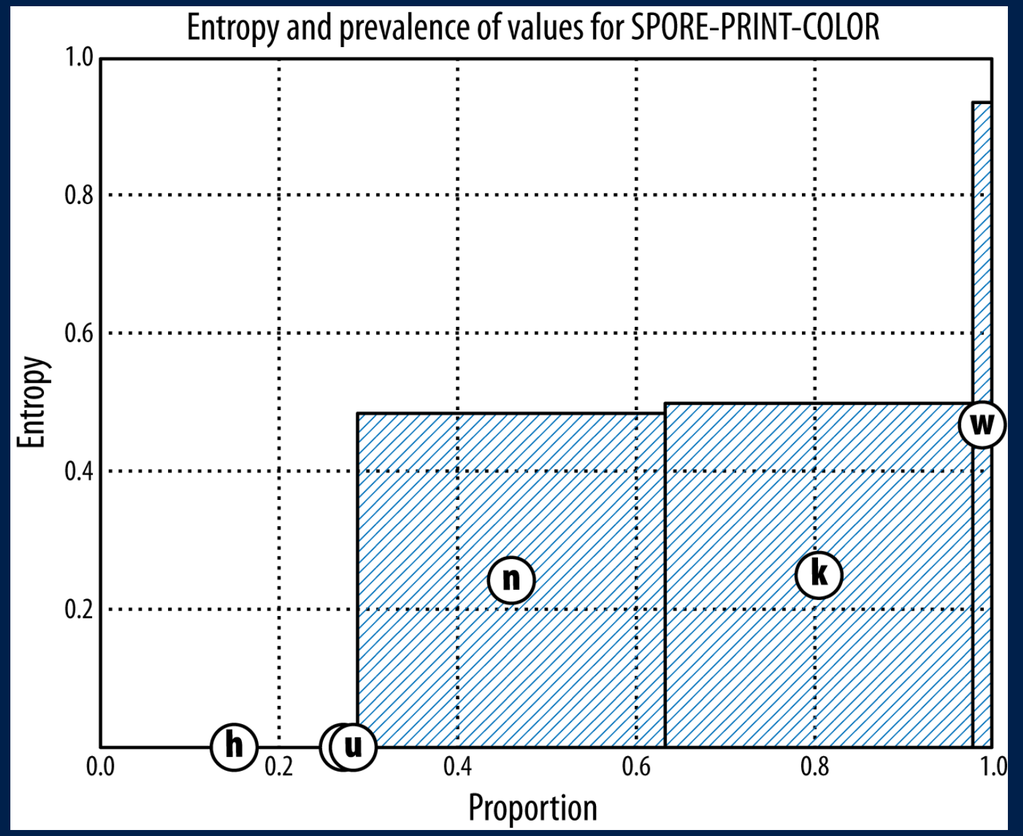
Example: Attribute Selection with Information Gain

Figure. Entropy chart for the Mushroom dataset as split by GILL-COLOR.



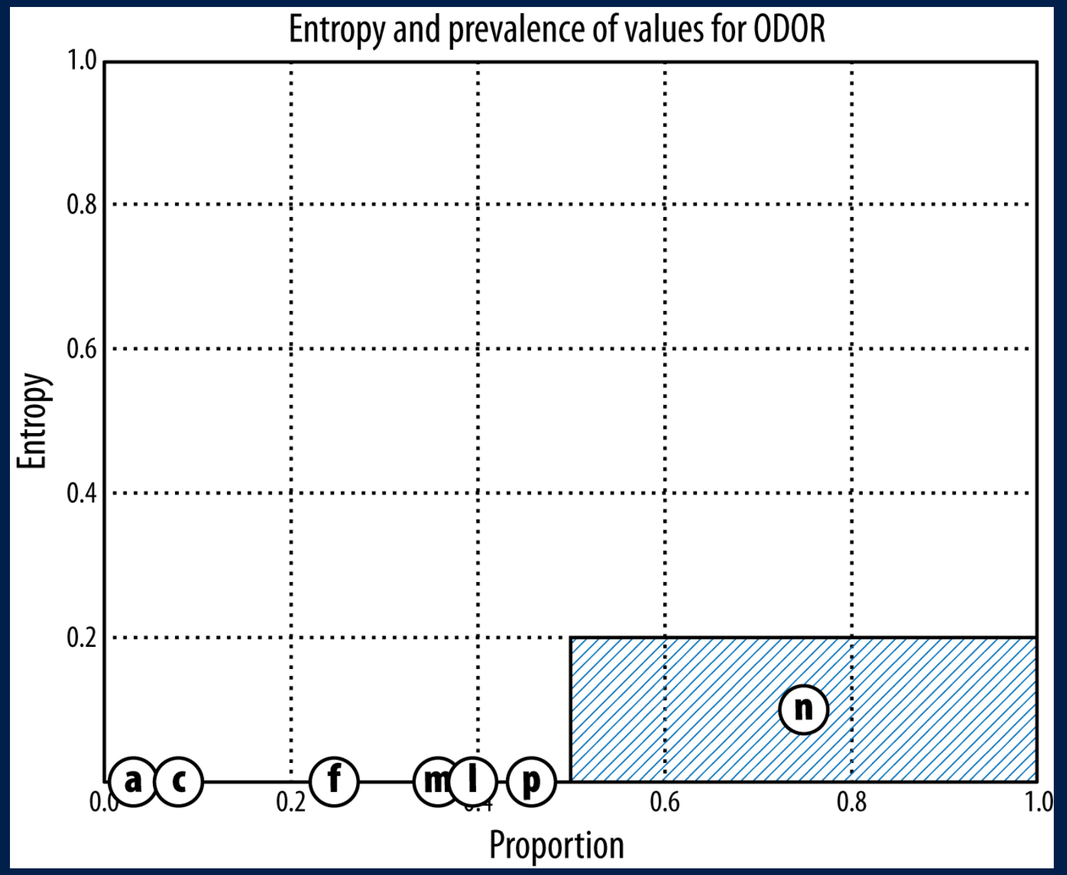
Example: Attribute Selection with Information Gain

- *Figure 3-8. Entropy chart for the Mushroom dataset as split by SPORE-PRINT-COLOR.*
- A few of the values, such as h (chocolate), specify the target value perfectly and thus produce zero-entropy bars. But notice that they don’t account for very much of the population, only about 30%.



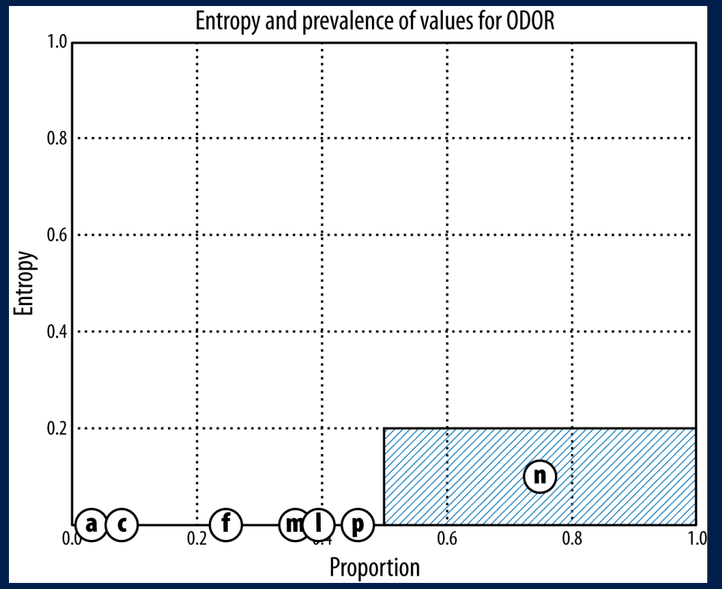
Example: Attribute Selection with Information Gain

- *Figure 3-9. Entropy chart for the Mushroom dataset as split by ODOR.*



Example: Attribute Selection with Information Gain

- In fact, ODOR has the highest information gain of any attribute in the Mushroom dataset. It can reduce the dataset's total entropy to about 0.1, which gives it an information gain of 0.96 – 0.1 = 0.86.
  - Many odors are completely characteristic of poisonous or edible mushrooms, so odor is a very informative attribute to check when considering mushroom edibility.
  - See footnote 5 on page 61.



Example: Attribute Selection with Information Gain

- If you’re going to build a model to determine the mushroom edibility using only a *single feature,* you should choose its odor.
- If you were going to build a more complex model you might start with the attribute ODOR before considering adding others.
- In fact, this is exactly the topic of the next section.

Outline

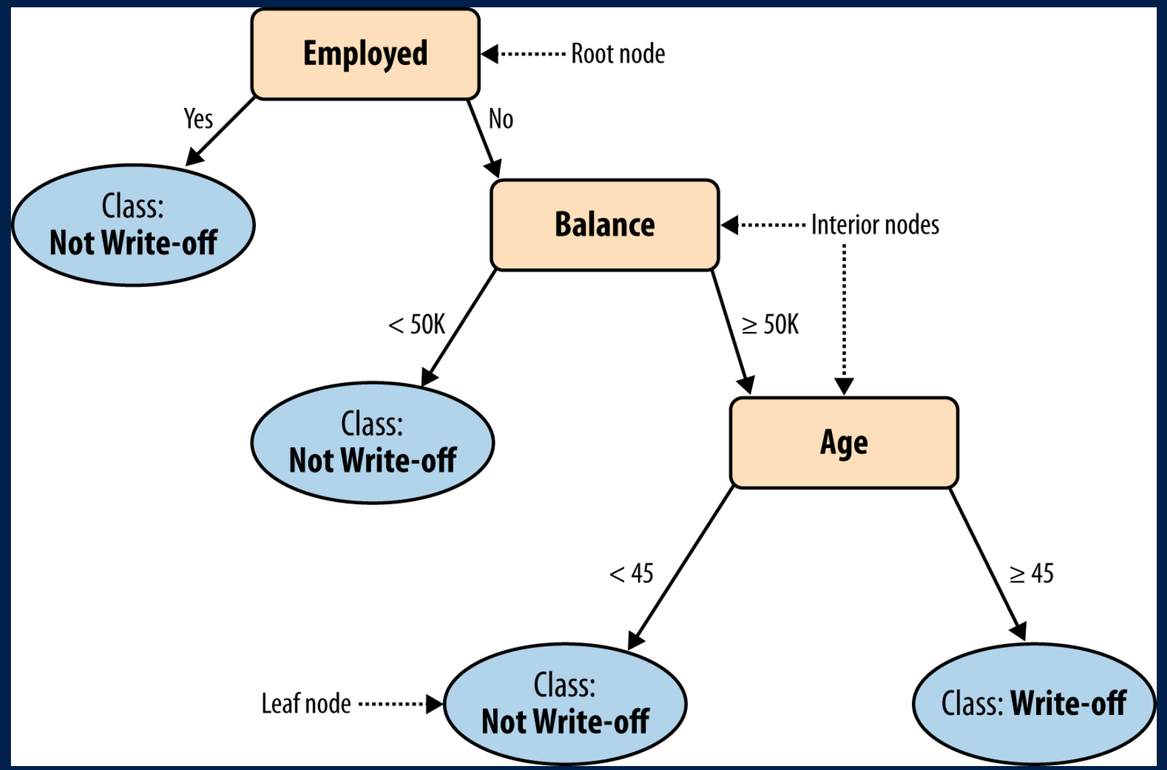
- Models, Induction, and Prediction
- Supervised Segmentation
  - Selecting Informative Attributes
  - Example: Attribute Selection with Information Gain
  - **Supervised Segmentation with Tree-Structured Models**

Supervised Segmentation with Tree-structured Models

- Let’s continue on the topic of creating a supervised segmentation, because as important as it is, attribute selection alone does not seem to be sufficient.
- 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, it’s not clear how to put them together.
- We now introduce an elegant application of the ideas we’ve developed for selecting important attributes, to produce a multivariate (multiple attribute) supervised segmentation.

Supervised Segmentation with Tree-structured Models

- Consider a segmentation of the data to take the form of a "tree," such as that shown in Figure 3-10.



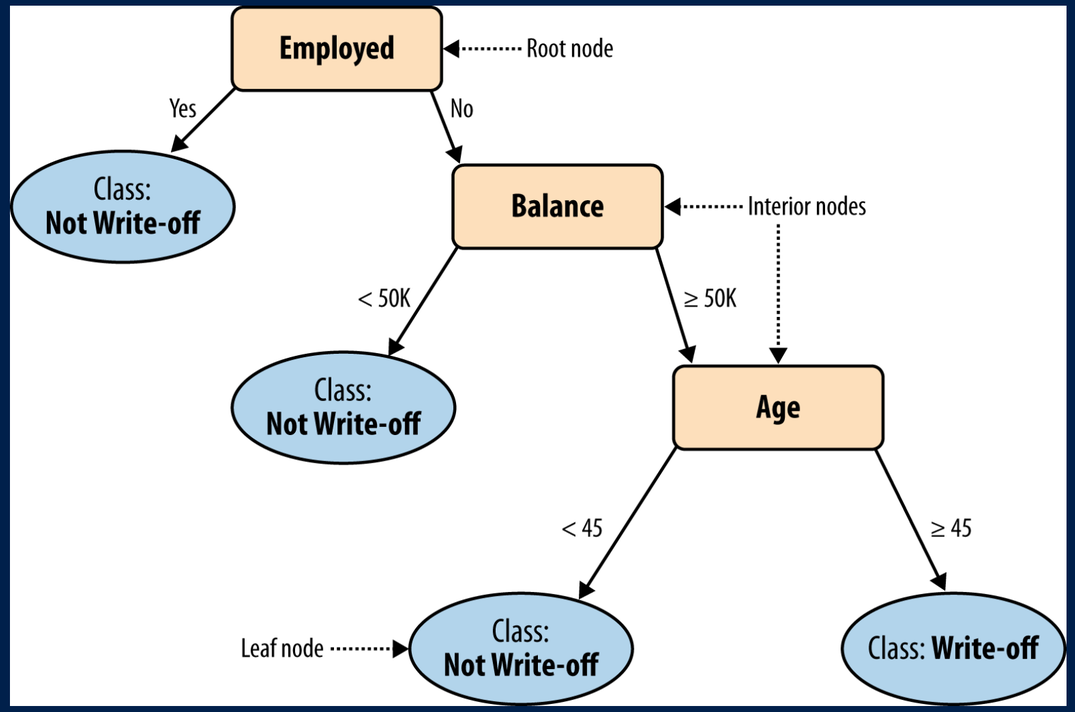
Supervised Segmentation with Tree-structured Models

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**



- The values of Claudio's attributes are *Balance=115K, Employed=No, and Age=40.*

Supervised Segmentation with Tree-Structured Models

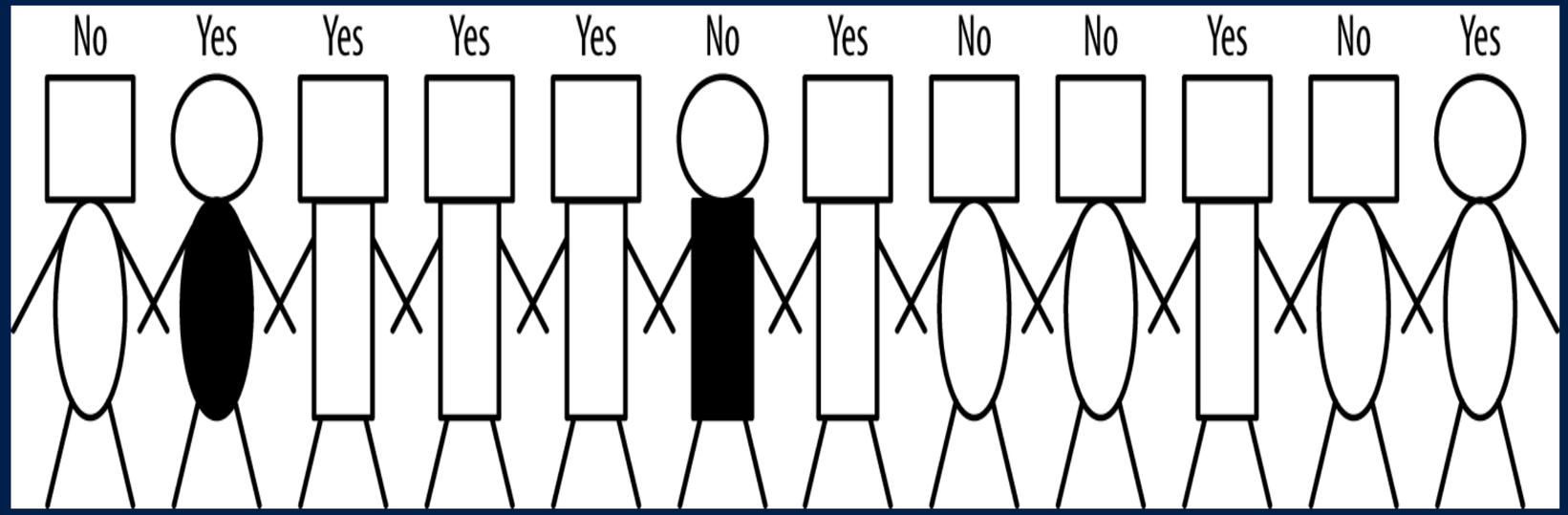
- There are many techniques to induce a supervised segmentation from a dataset. One of the most popular is to create a tree-structured model (*tree induction).*
- *These techniques* are popular because tree models are easy to understand, and because the induction procedures are elegant (simple to describe) and easy to use. They are robust to many common data problems and are relatively efficient.

Supervised Segmentation with Tree-Structured Models

- Combining the ideas introduced above, the goal of the tree is to provide a supervised segmentation— more specifically, to partition the instances, based on their attributes, into subgroups that have similar values for their target variables.
- We would like for each “leaf ” segment to contain instances that tend to belong to the same class.

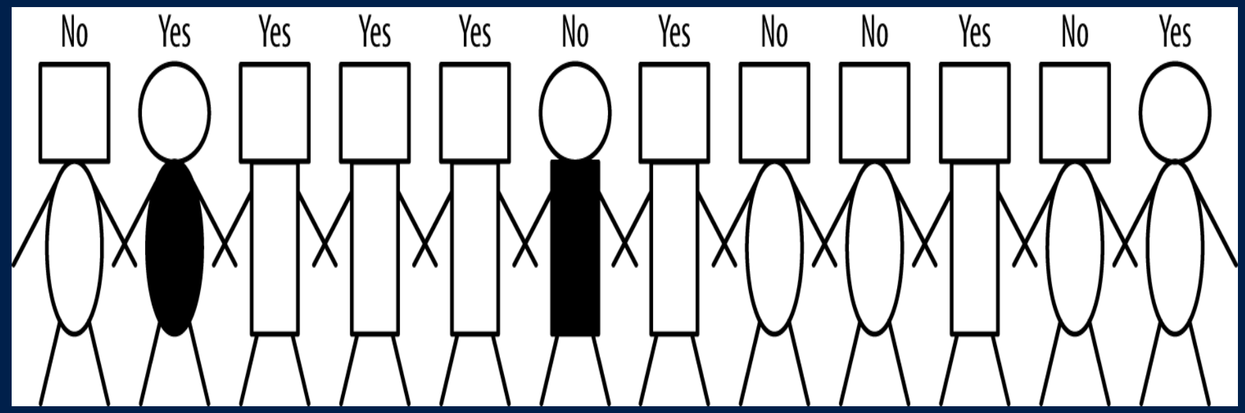
Supervised Segmentation with Tree-Structured Models

- To illustrate the process of classification tree induction, consider the very simple example set shown in figure below.



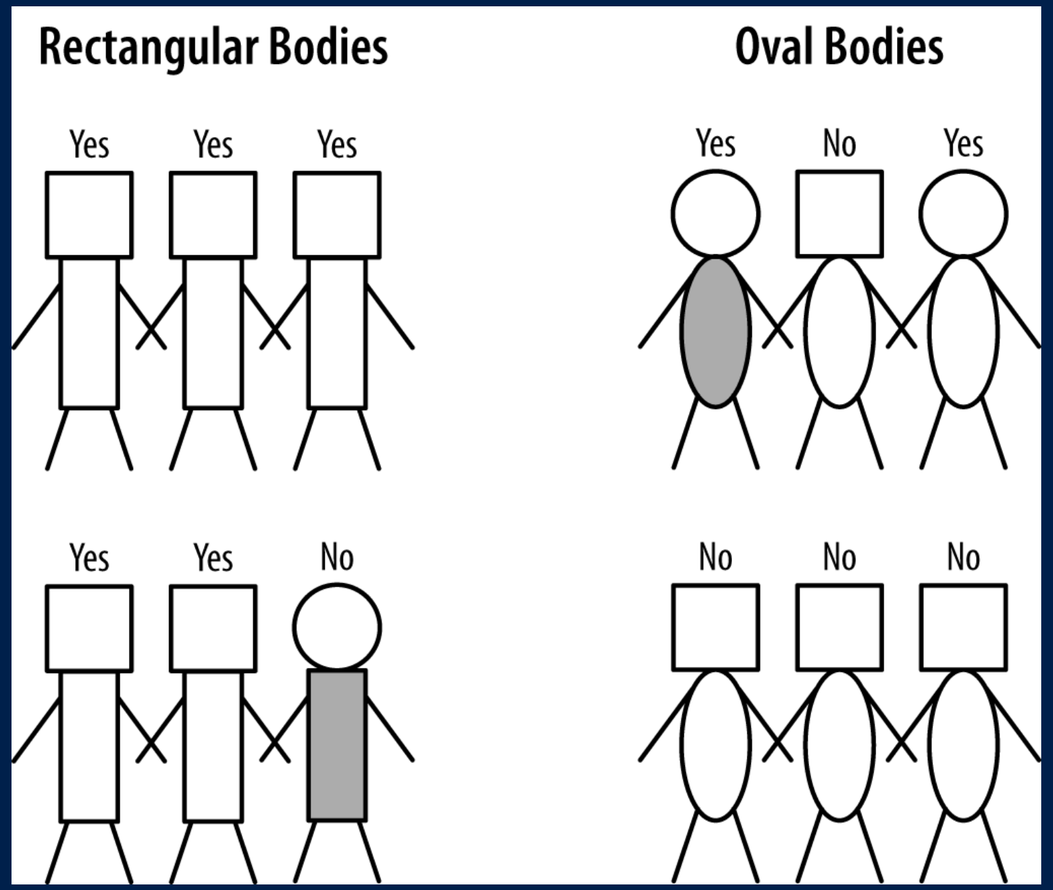
Selecting Informative Attributes

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



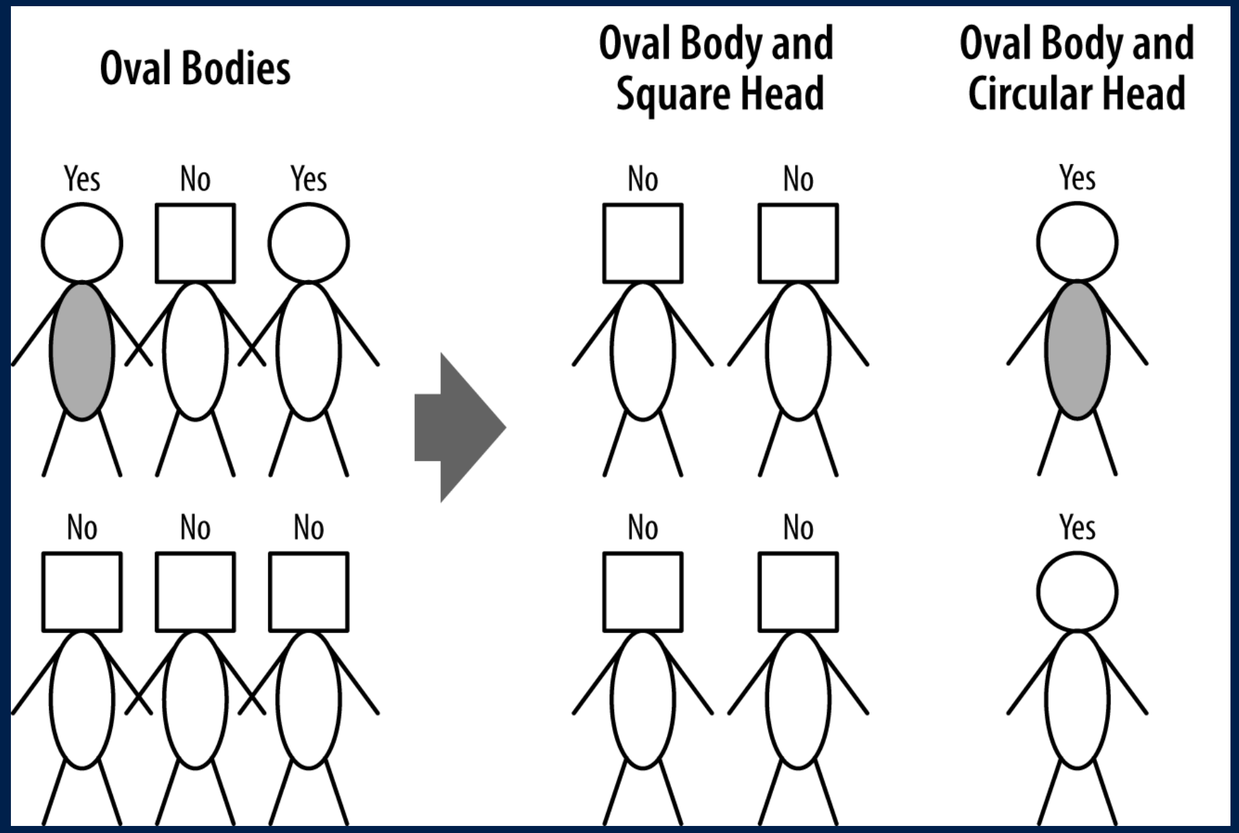
Supervised Segmentation with Tree-Structured Models

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



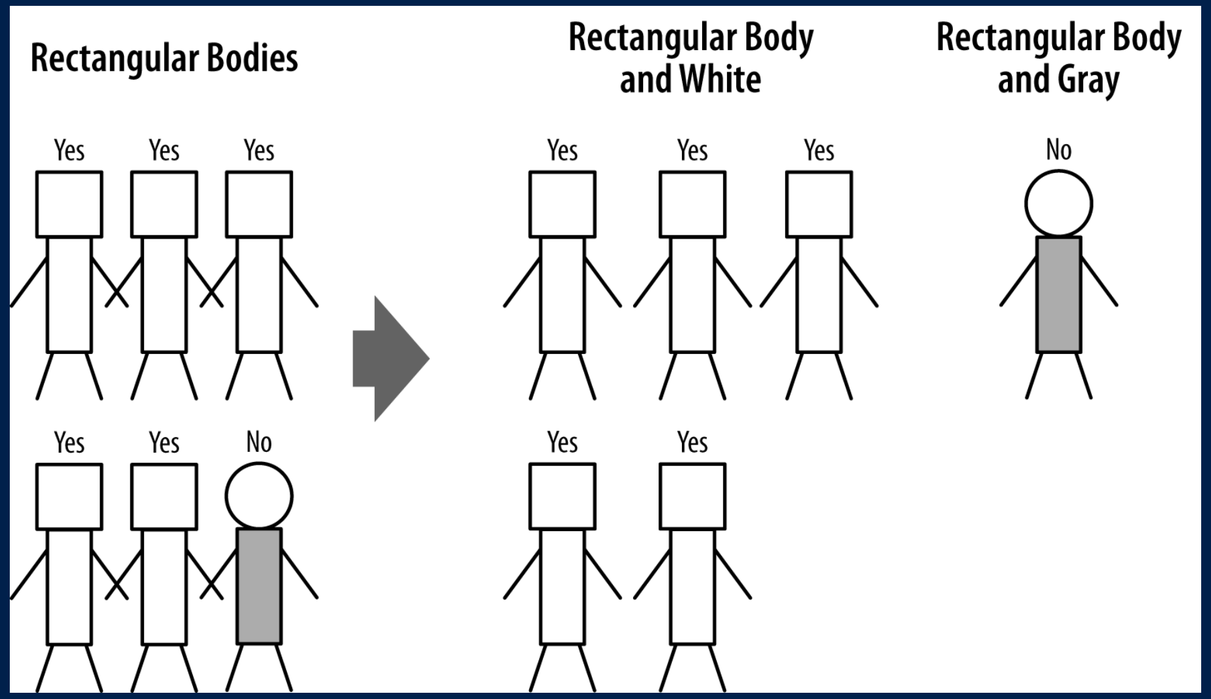
Supervised Segmentation with Tree-Structured Models

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

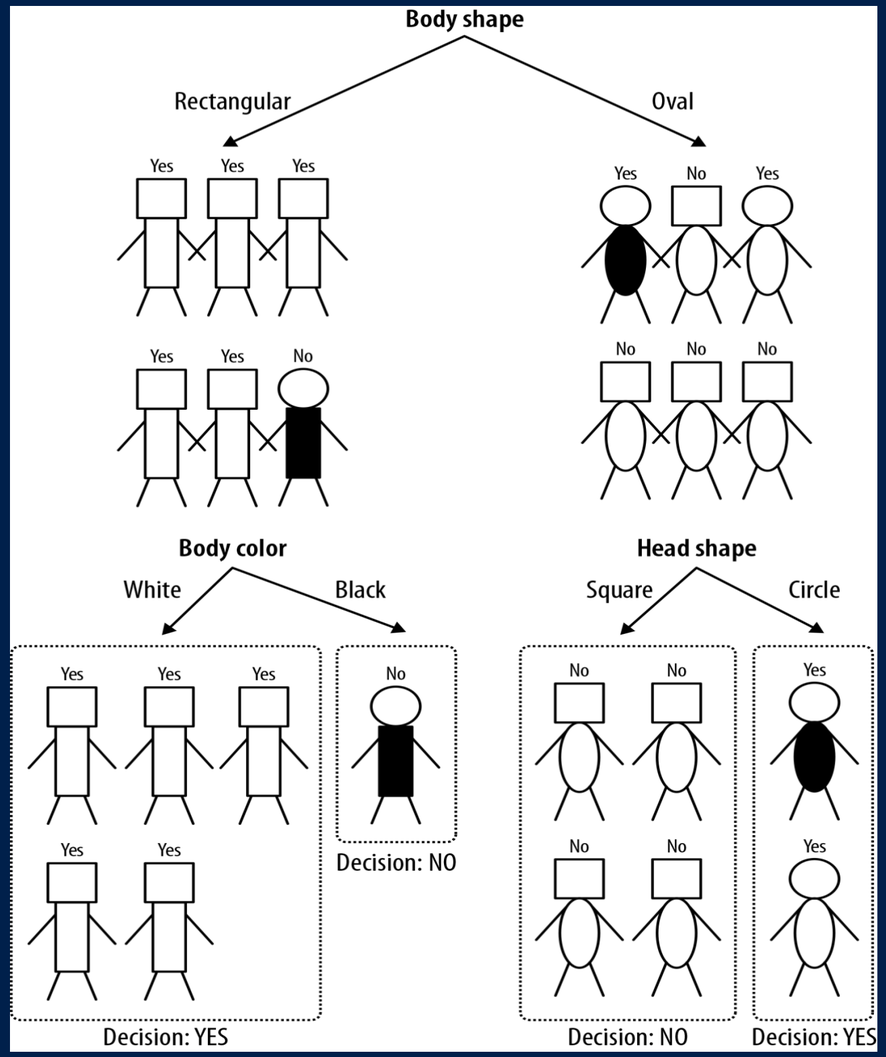


Supervised Segmentation with Tree-Structured Models

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



- Figure. The classification tree resulting from the splits done



Supervised Segmentation with Tree-Structured Models

- In summary, the procedure of classification tree induction is a recursive process of divide and conquer, where the goal at each step is to select an attribute to partition the current group into subgroups that are as pure as possible with respect to the target variable.
- We perform this partitioning recursively, splitting further and further until we are done. We choose the attributes to split upon by testing all of them and selecting whichever yields the purest subgroups.

Supervised Segmentation with Tree-Structured Models

- When are we done? (In other words, when do we stop recursing?)
  - It should be clear that we would stop when the nodes are pure, or when we run out of variables to split on.

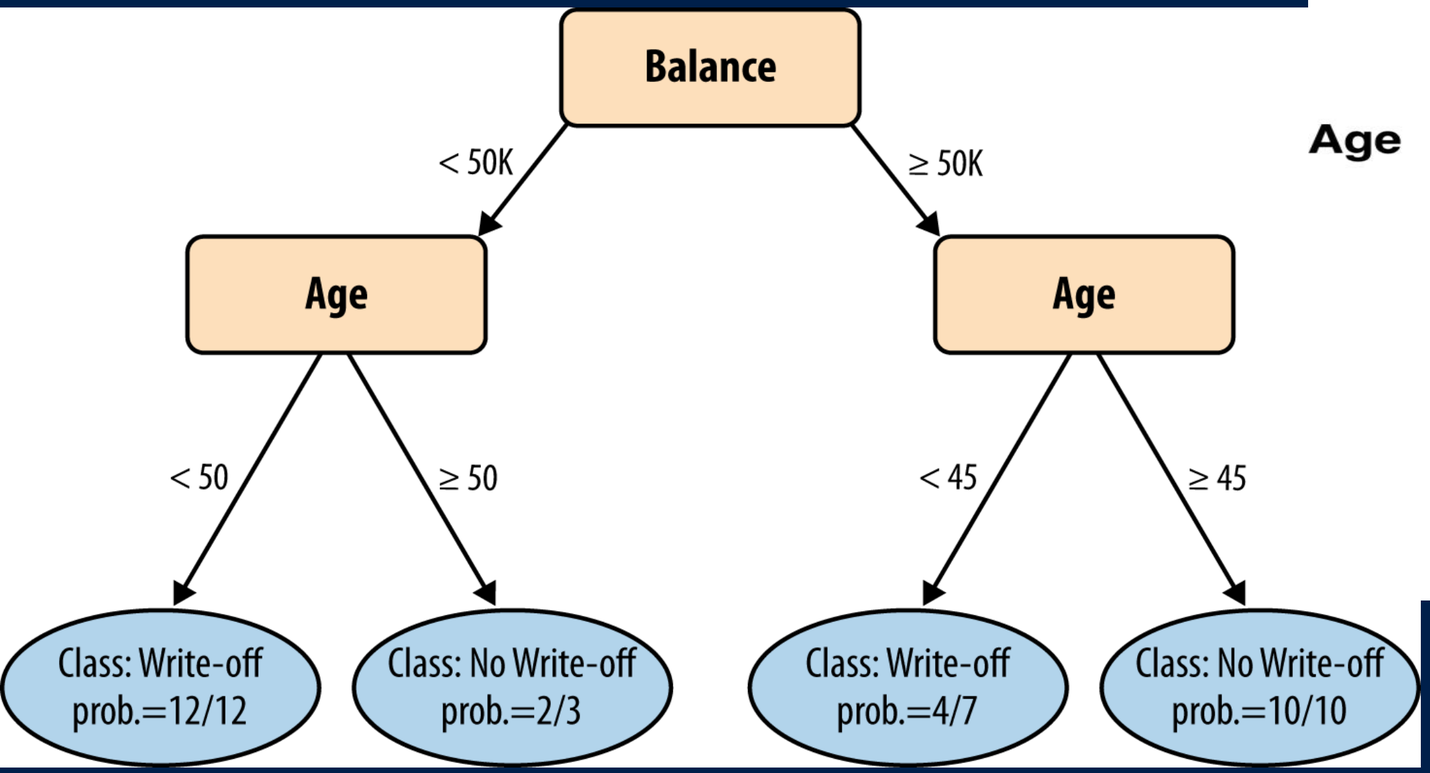
Visualizing Segmentations

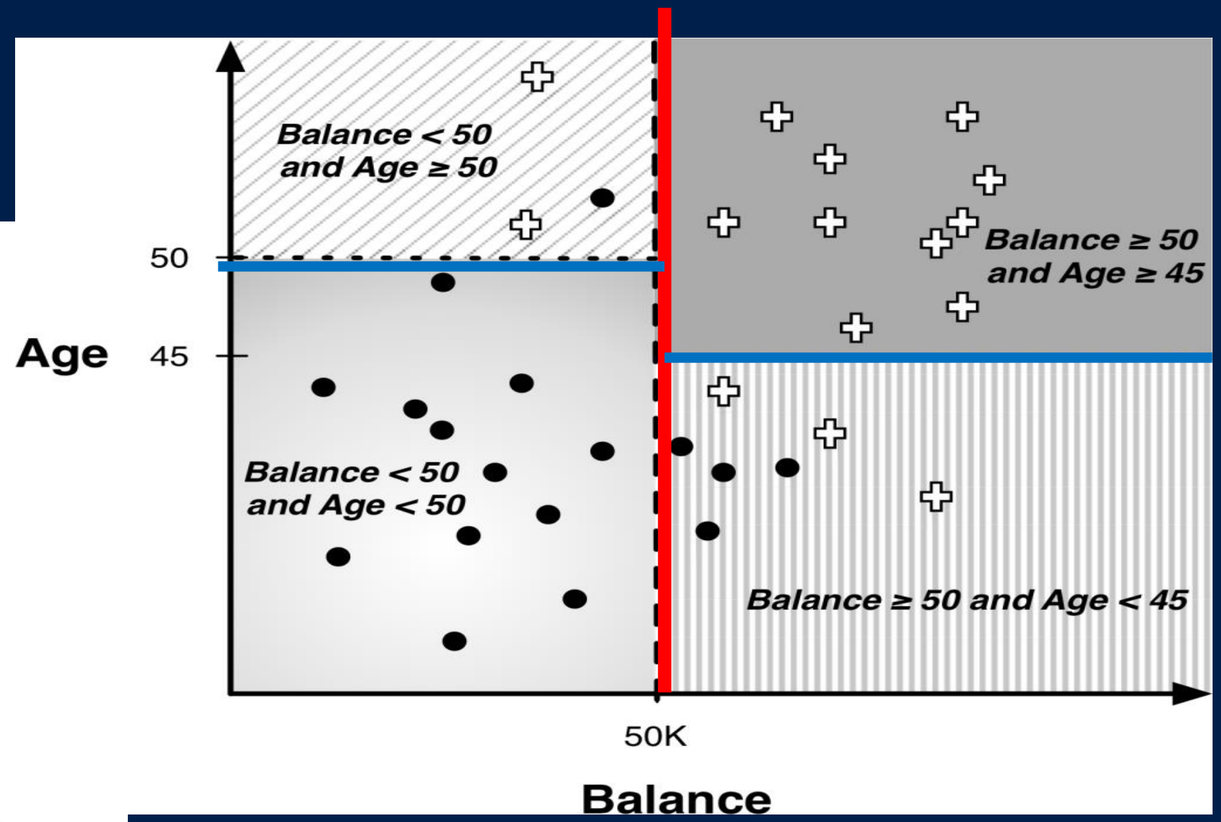
- It is instructive to visualize exactly how a classification tree partitions the instance space.
- The instance space is simply the space described by the data features.
- A common form of instance space visualization is a scatterplot on some pair of features, used to compare one variable against another to detect correlations and relationships.

Visualizing Segmentations

- Though data may contain dozens or hundreds of variables, it is only really possible to visualize segmentations in two or three dimensions at once
- Visualizing models in instance space in a few dimensions is useful for understanding the different types of models because it provides insights that apply to higher dimensional spaces as well

Visualizing Segmentations



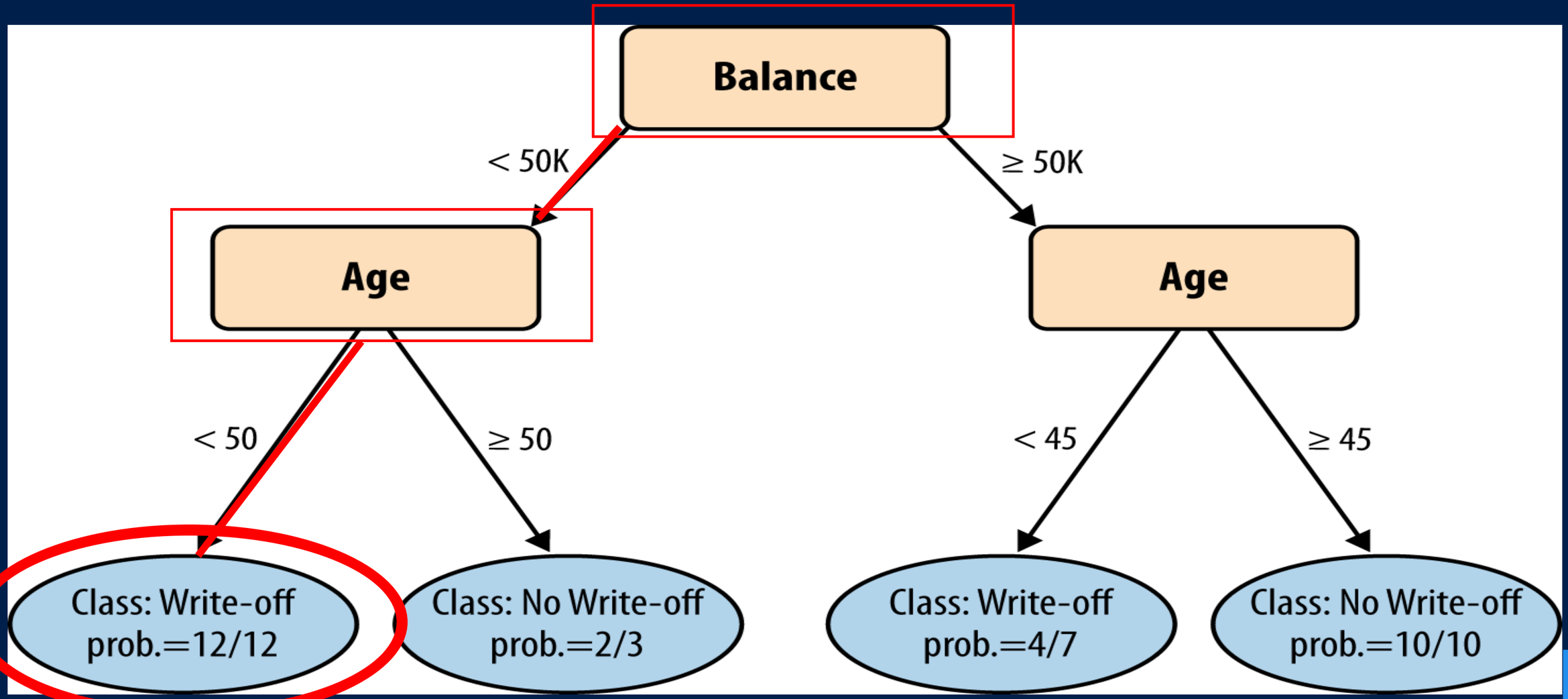


*The black dots correspond to instances of the class Write-off.
*The plus signs correspond to instances of class non-Write-off.

Trees as Sets of Rules

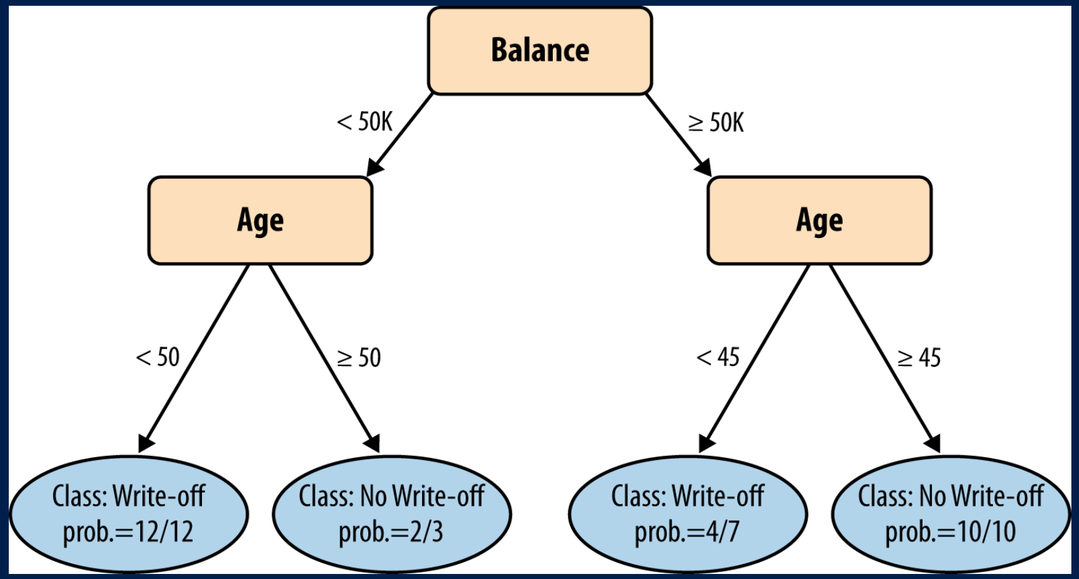
- You classify a new unseen instance by starting at the root node and following the attribute tests downward until you reach a leaf node, which specifies the instance's predicted class.
- If we trace down a single path from the root node to a leaf, collecting the conditions as we go, we generate a rule.
- Each rule consists of the attribute tests along the path connected with AND.

Trees as Sets of Rules



Trees as Sets of Rules

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



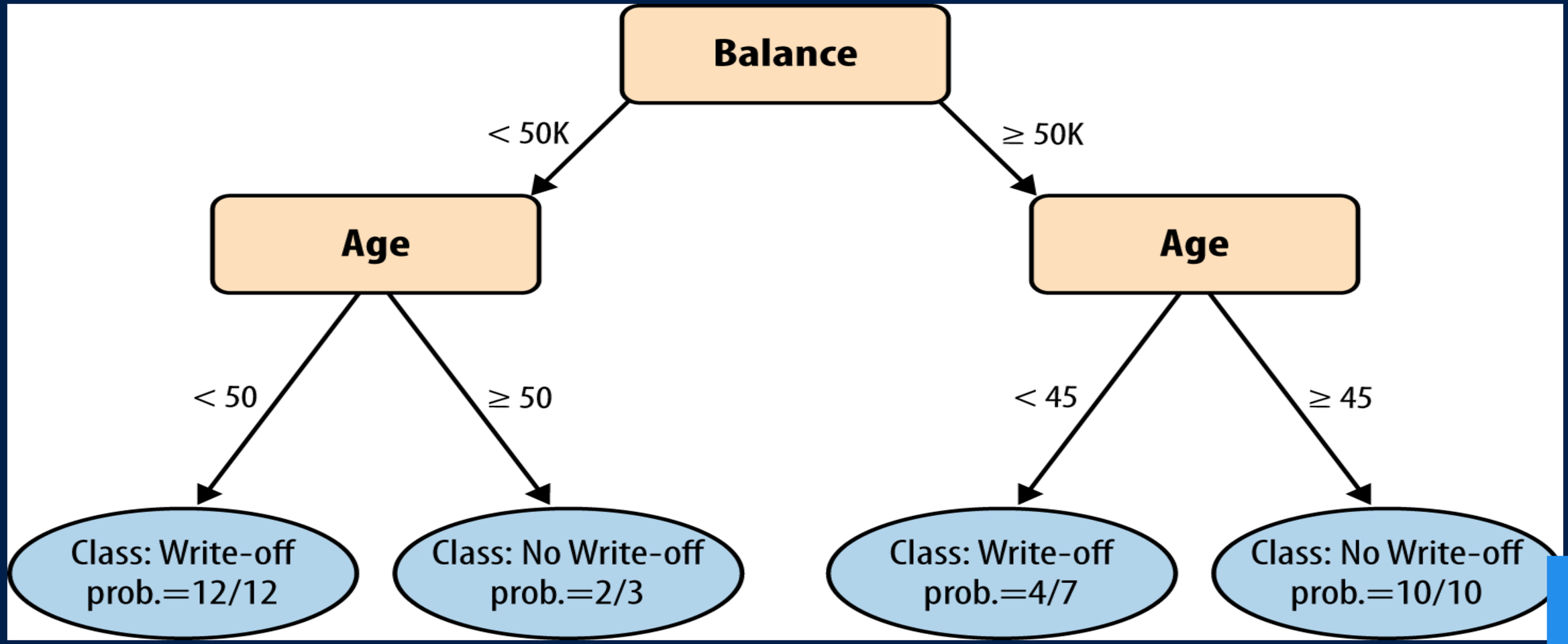
Trees as Sets of Rules

- The classification tree is equivalent to this rule set.
- Every classification tree can be expressed as a set of rules this way.

Probability Estimation

- In many decision-making problems, we would like a more informative **prediction** than just a classification.
- For example, in our churn prediction problem. If we have the customers' probability of leaving when their contracts are about to expire, we could rank them and use a limited incentive budget to the highest probability instances.
- Alternatively, we may want to allocate our incentive budget to the instances with the highest expected loss, for which you'll need the probability of churn.

Probability Estimation Tree



Probability Estimation

- If we are satisfied to assign the same class probability to every member of the segment corresponding to a tree leaf, we can use instance counts at each leaf to compute a class probability estimate.
- For example, if a leaf contains n positive instances and m negative instances, the probability of any new instance being positive may be estimated as n/(n+m). This is called a frequency-based estimate of class membership probability.

Probability Estimation

- A problem: we may be overly optimistic about 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?
- This phenomenon is one example of a fundamental issue in data science (“overfitting”).

Probability Estimation

- Instead of simply computing the frequency, we would often use a "smoothed" version of the frequency-based estimate, known as the Laplace correction, the purpose of which is to moderate the influence of leaves with only a few instances.
- The equation for binary class probability estimation becomes:

$$p(c) = \frac{n + 1}{n + m + 2}$$

where n is the number of examples in the leaf belonging to class c, and m is the number of examples not belonging to class c.

Example: Addressing the Churn Problem with Tree Induction

- We have a historical data set of 20,000 customers.
- At the point of collecting the data, each customer either had stayed with the company or had left (churned).

Example: Addressing the Churn Problem with Tree Induction

Table 3-2. Attributes for the cellular phone churn-prediction problem

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)?

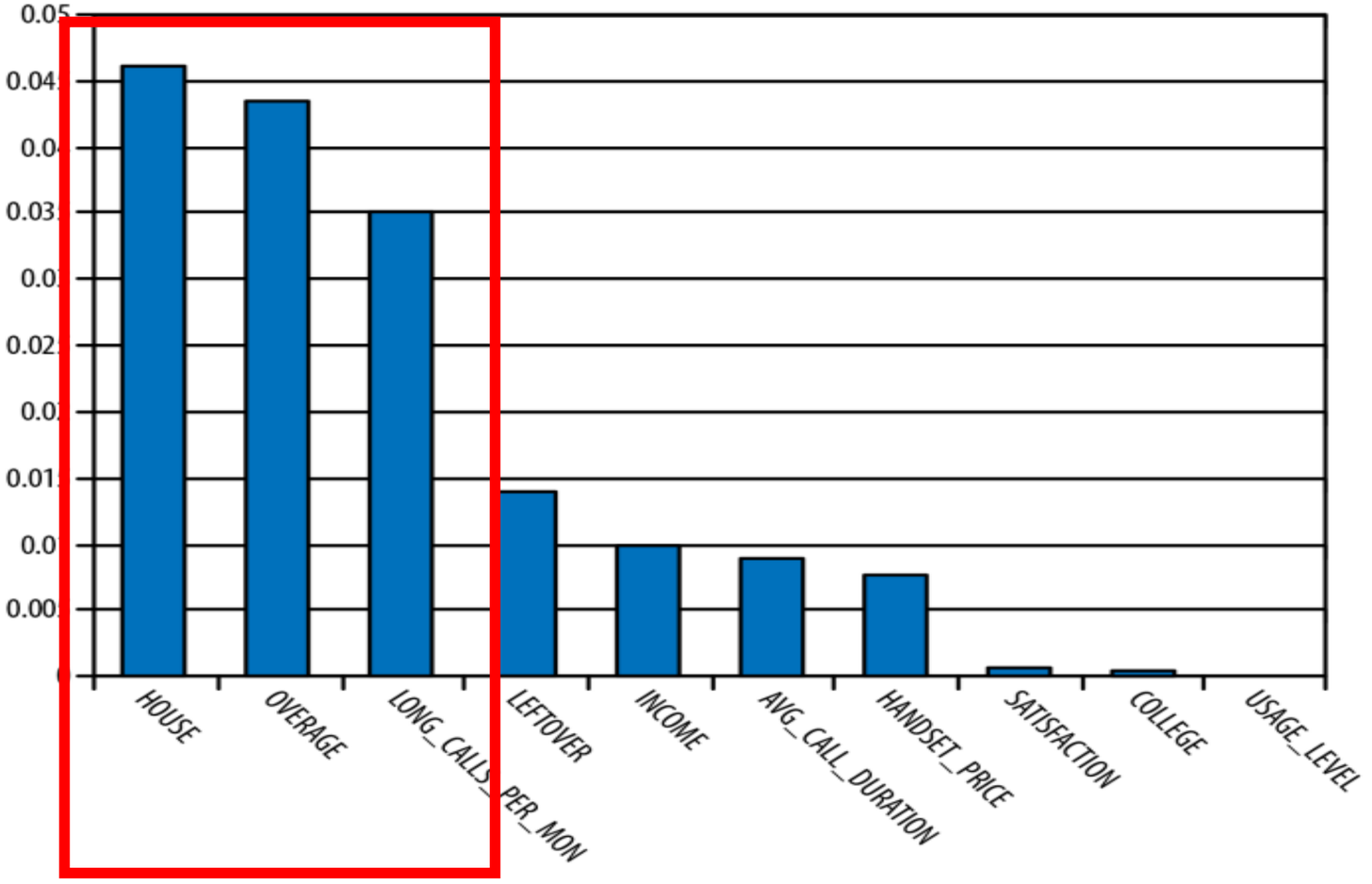
Example: Addressing the Churn Problem with Tree Induction

- How good are each of these variables individually?
- For this we measure the information gain

of each attribute, as discussed earlier. Specifically, we apply Equation 3-2 to each variable independently over the entire set of instances, to see what each gains us.

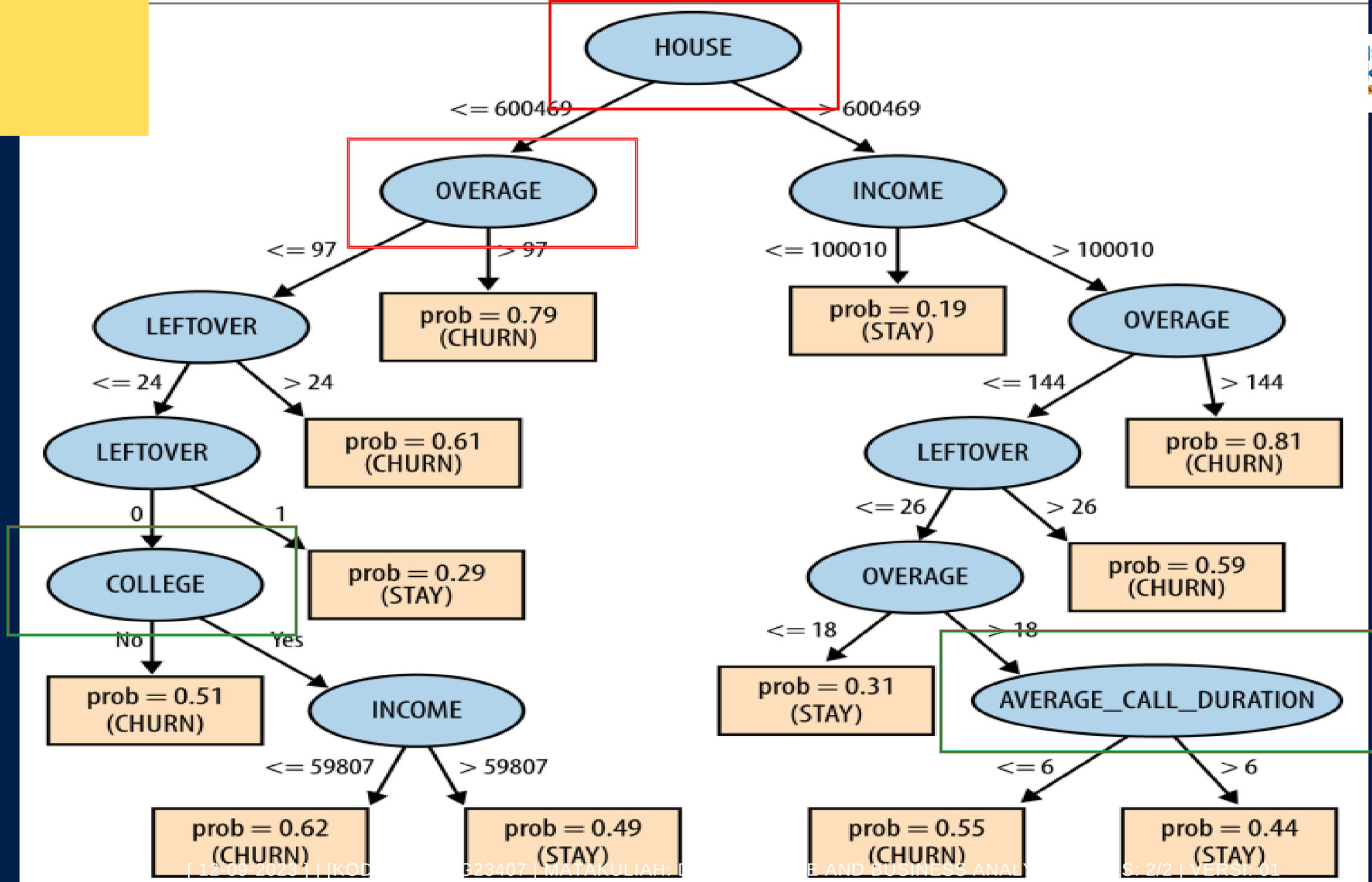
Equation 3-2. Information gain

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



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

Figure 3-17. Churn attributes from Table 3-2 ranked by information gain.


HOUSE
<= 600469
> 600469
OVERAGE
INCOME
<= 97
> 97
<= 100010
> 100010
LEFTOVER
prob = 0.79 (CHURN)
prob = 0.19 (STAY)
OVERAGE
<= 24
> 24
<= 144
> 144
LEFTOVER
prob = 0.61 (CHURN)
LEFTOVER
prob = 0.81 (CHURN)
0
1
<= 26
> 26
COLLEGE
prob = 0.29 (STAY)
OVERAGE
prob = 0.59 (CHURN)
No
Yes
<= 18
> 18
prob = 0.51 (CHURN)
INCOME
prob = 0.31 (STAY)
AVERAGE_CALL_DURATION
<= 59807
> 59807
<= 6
> 6
prob = 0.62 (CHURN)
prob = 0.49 (STAY)
prob = 0.55 (CHURN)
prob = 0.44 (STAY)

Example: Addressing the Churn Problem with Tree Induction

- The answer is that the table ranks each feature by how good it is independently, evaluated separately on the entire population of instances.
- Nodes in a classification tree depend on the instances above them in the tree.

Example: Addressing the Churn Problem with Tree Induction

- Therefore, except for the root node, features in a classification tree are not evaluated on the entire set of instances.
- The information gain of a feature depends on the set of instances against which it is evaluated, so the ranking of features for some internal node may not be the same as the global ranking.

Institut Informatika & Bisnis DARMAJAYA
Yayasan Alfian Husin

SAINS DATA
Institut Informatika & Bisnis Darmajaya

Kampus Merdeka INDONESIA JAYA

MERDEKA BELAJAR

THANK YOU!!

DATA SCIENCE DARMAJAYA “YOUR BEST FUTURE IN DATA”