Decision Trees

- Algorithms

Overview

- Overview of Decision Tree Algorithms
- Information Gain and Entropy
- Recursive Partitioning
- Gini Index and Entropy
- Decision Tree Algorithms – A Comparison

Decision Tree Algorithms

- ID3, C4.5, CART, CHAID, MARS, C5.0

- Algorithms differ in
  - Splitting criterion: information gain (Shannon Entropy, Gini impurity, misclassification error), use of statistical tests, objective function, etc.
  - Binary split vs. multi-way splits
  - Discrete vs. continuous variables
  - Pre vs. post-pruning

- The process of growing a decision tree can be expressed as a recursive algorithm as follows:
  1. Pick a feature such that when parent node is split, it results in the largest information gain.
  2. Stop if child nodes are pure or no improvement in class purity can be made.
  3. Go back to step 1 for each of the two child nodes.

Information Gain and Entropy

We define the criterion at a node such that it maximizes information gain

$$GAIN(\mathcal{D}, xj) = H(\mathcal{D}) - \sum_{v \in Values(x_j)} \frac{|\mathcal{D}_v|}{|\mathcal{D}|} H(\mathcal{D}_v)$$

Where $\mathcal{D}$ is the training set at the parent node, and $\mathcal{D}_v$ is a dataset at a child node upon splitting.

H(X) is the Shannon entropy or the 'average information' and given by

$$-\sum_{k=1}^{m} p(X = i)\, log_2(X = i)$$

And refers to the expected number of bits needed to encode a randomly drawn value of *X,* under most efficient coding scheme.

Decrease in Average entropy of child nodes which are increasingly homogenous

Thus **Information Gain** = Entropy(parent) – [Average of Entropy(children)]

Riding Mowers Dataset – Ownership of Lawn Mowers

Income	Lot_Size	Ownership
60.0	18.4	owner
85.5	16.8	owner
64.8	21.6	owner
61.5	20.8	owner
87.0	23.6	owner
110.1	19.2	owner
108.0	17.6	owner
82.8	22.4	owner
69.0	20.0	owner
93.0	20.8	owner
51.0	22.0	owner
81.0	20.0	owner
75.0	19.6	non-owner
52.8	20.8	non-owner
64.8	17.2	non-owner
43.2	20.4	non-owner
84.0	17.6	non-owner
49.2	17.6	non-owner
59.4	16.0	non-owner
66.0	18.4	non-owner
47.4	16.4	non-owner
33.0	18.8	non-owner
51.0	14.0	non-owner
63.0	14.8	non-owner

- Goal: Classify 24 households as owning or not owning lawn mowers
- Predictors : Income, Lot Size
- Outcome/Response : Owner/Non-Owner

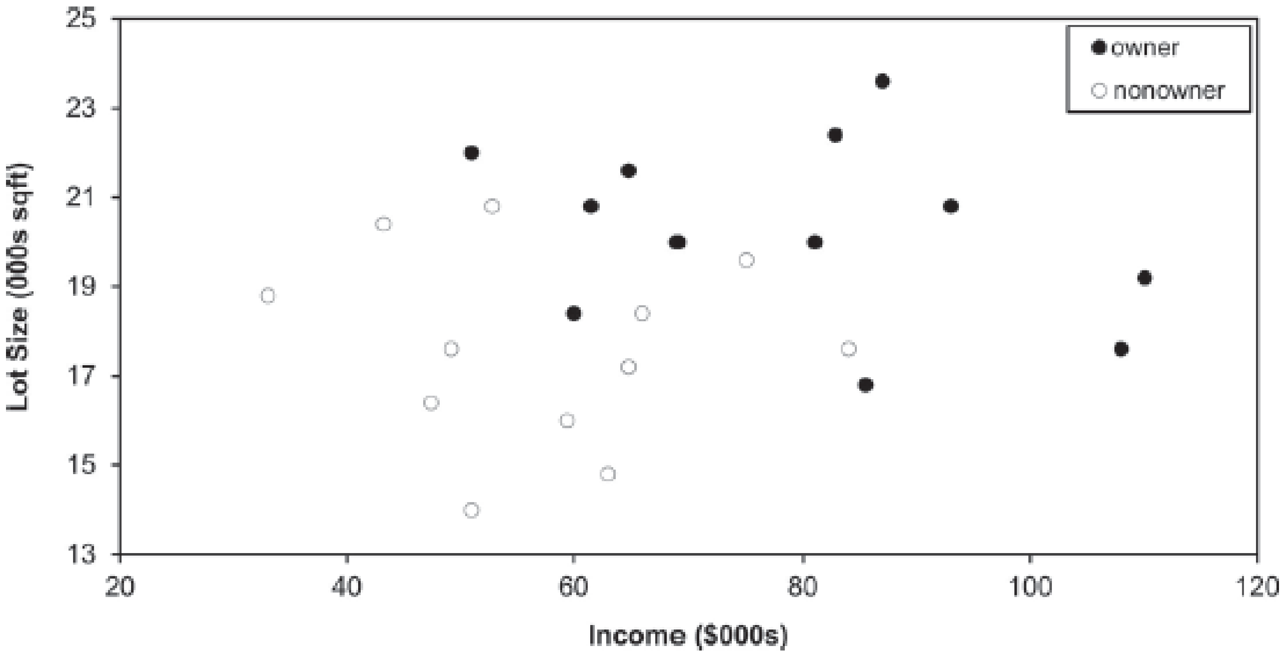


FIGURE 9.2 SCATTER PLOT OF LOT SIZE VS. INCOME FOR 24 OWNERS AND NONOWNERS OF RIDING MOWERS

Ref: Data Mining for Business Analytics: Concepts, Techniques and Applications in R, by Galit Shmueli et al., Wiley India, 2018.

Income	Lot_Size	Ownership
60.0	18.4	owner
85.5	16.8	owner
64.8	21.6	owner
61.5	20.8	owner

How to split using CART

- Order records according to one variable, say income
- Take a predictor value, say 60 (the first record) and divide records into those with income >= 60 and those < 60
- Measure resulting purity (homogeneity) of class in each resulting portion
- Try all other split values
- Repeat for other variable(s)
- Select the one variable & split that yields the most purity

The first split: Income = 60

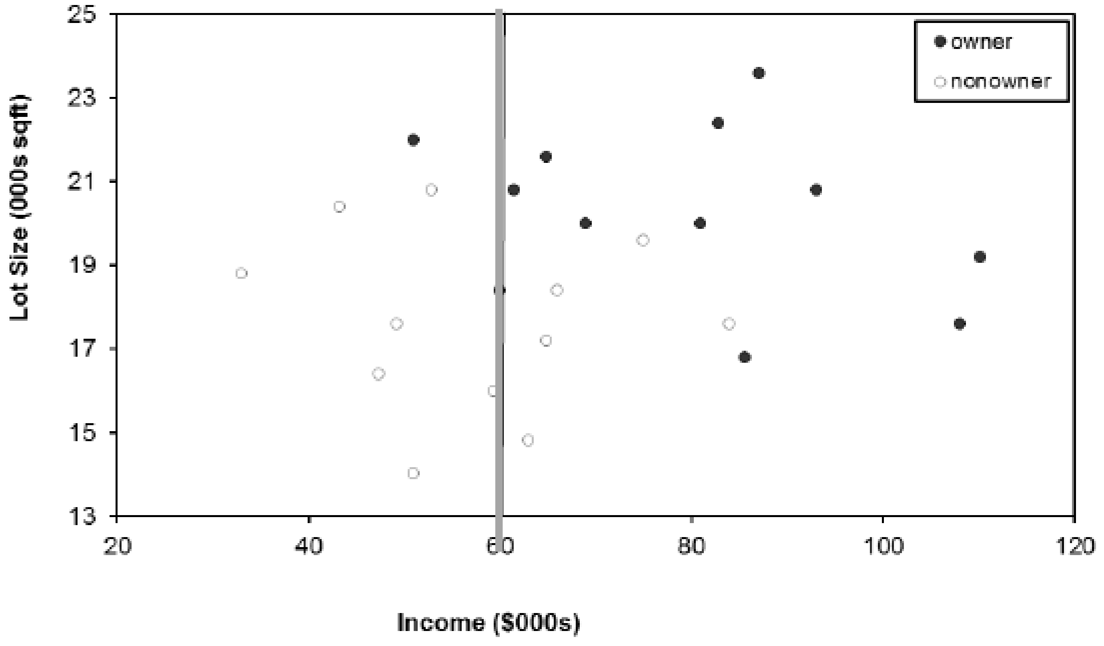


FIGURE 9.3 SPLITTING THE 24 RECORDS BY INCOME VALUE OF 60

Ref: Data Mining for Business Analytics: Concepts, Techniques and Applications in R, by Galit Shmueli et al., Wiley India, 2018.

The first split: Income = 60

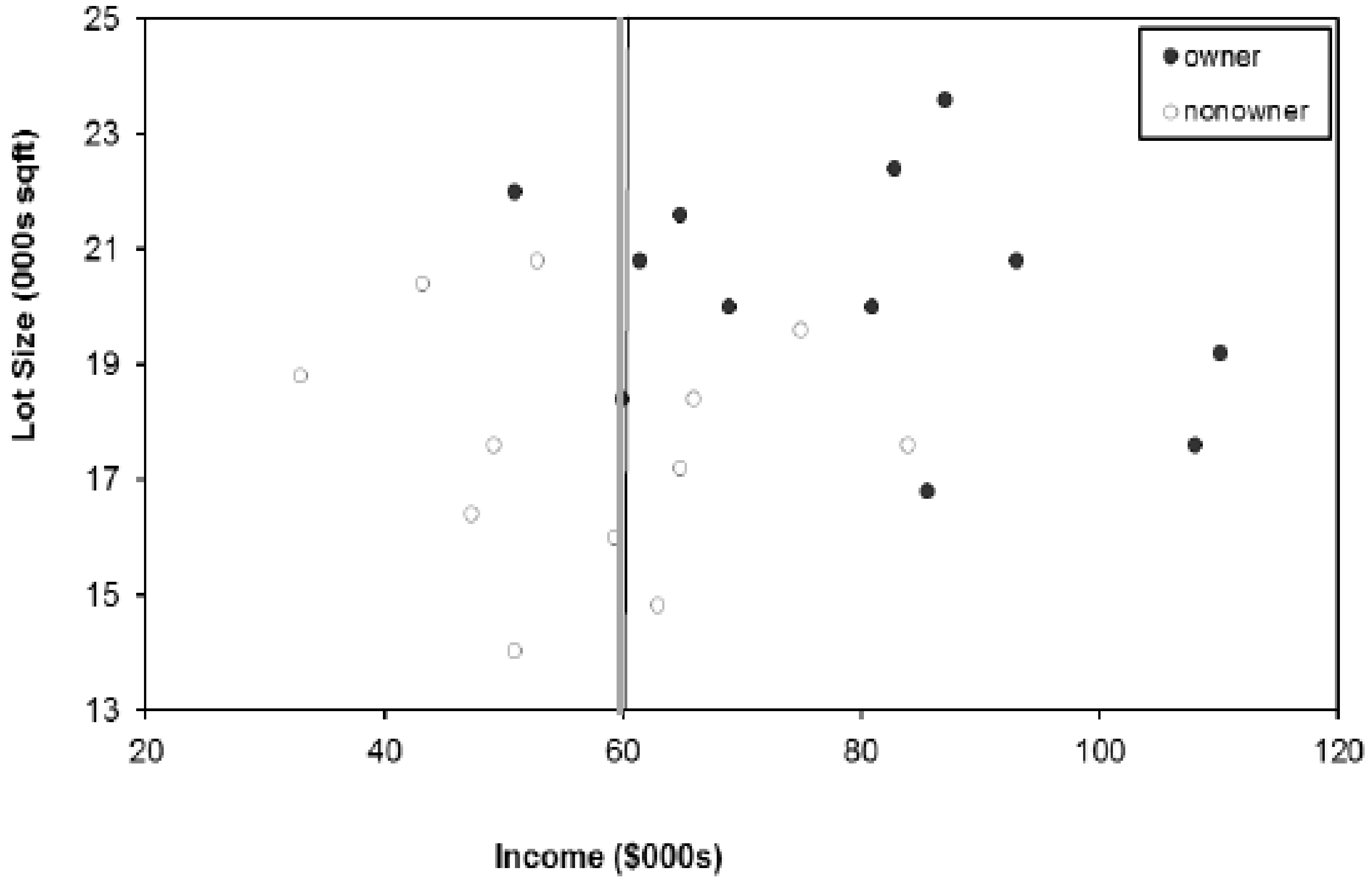
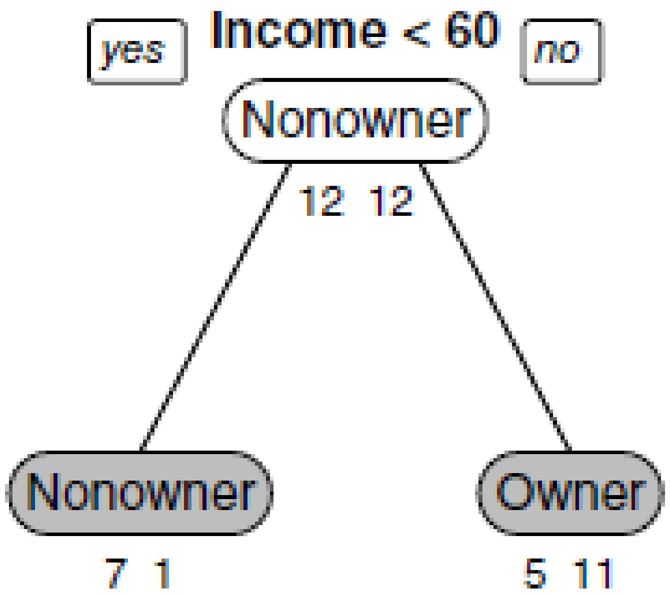


FIGURE 9.3 SPLITTING THE 24 RECORDS BY INCOME VALUE OF 60



Ref: Data Mining for Business Analytics: Concepts, Techniques and Applications in R, by Galit Shmueli et al., Wiley India, 2018.

Second Split: Lot size = 21

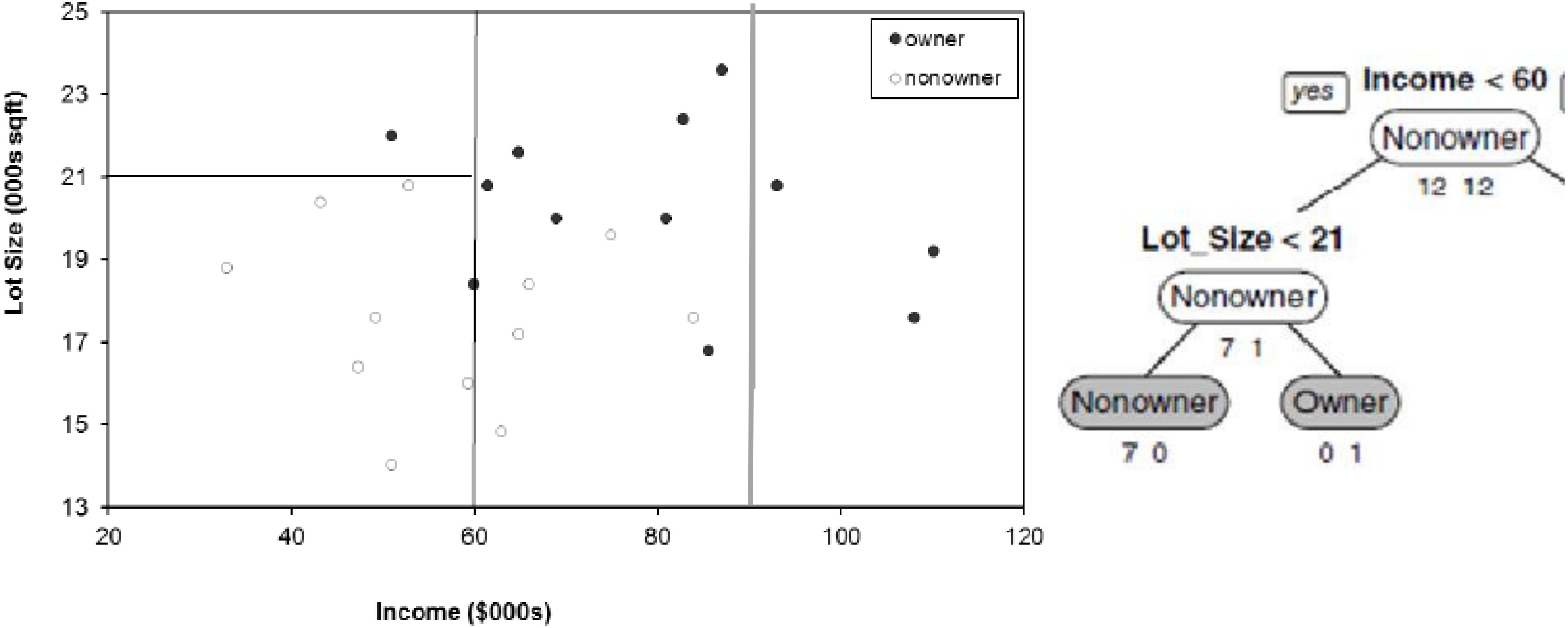


FIGURE 9.5 SPLITTING THE 24 RECORDS FIRST BY INCOME VALUE OF 60 AND THEN LOT SIZE VALUE OF 21

Ref: Data Mining for Business Analytics: Concepts, Techniques and Applications in R, by Galit Shmueli et al., Wiley India, 2018.

After All Splits

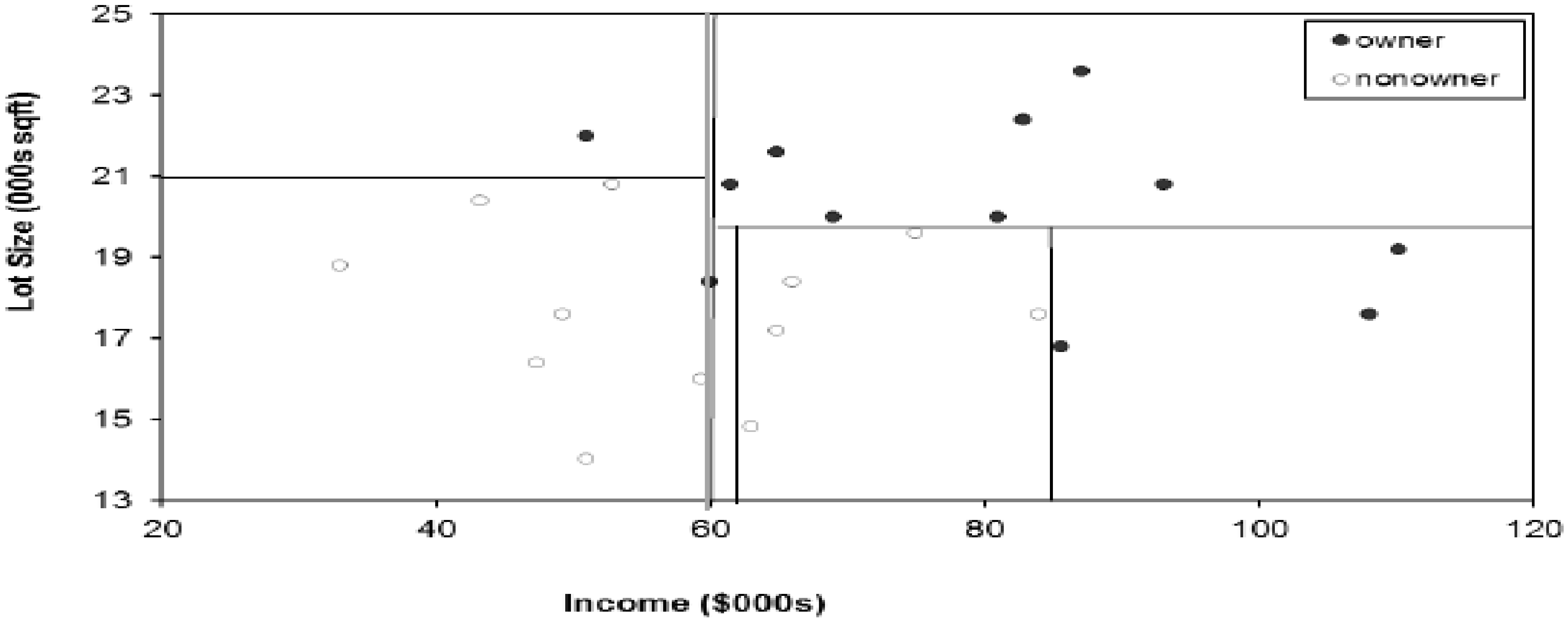


FIGURE 9.6 FINAL STAGE OF RECURSIVE PARTITIONING; EACH RECTANGLE CONSISTING OF A SINGLE CLASS (OWNERS OR NONOWNERS)

Ref: Data Mining for Business Analytics: Concepts, Techniques and Applications in R, by Galit Shmueli et al., Wiley India, 2018.

Axis-Parallel Decision Boundaries

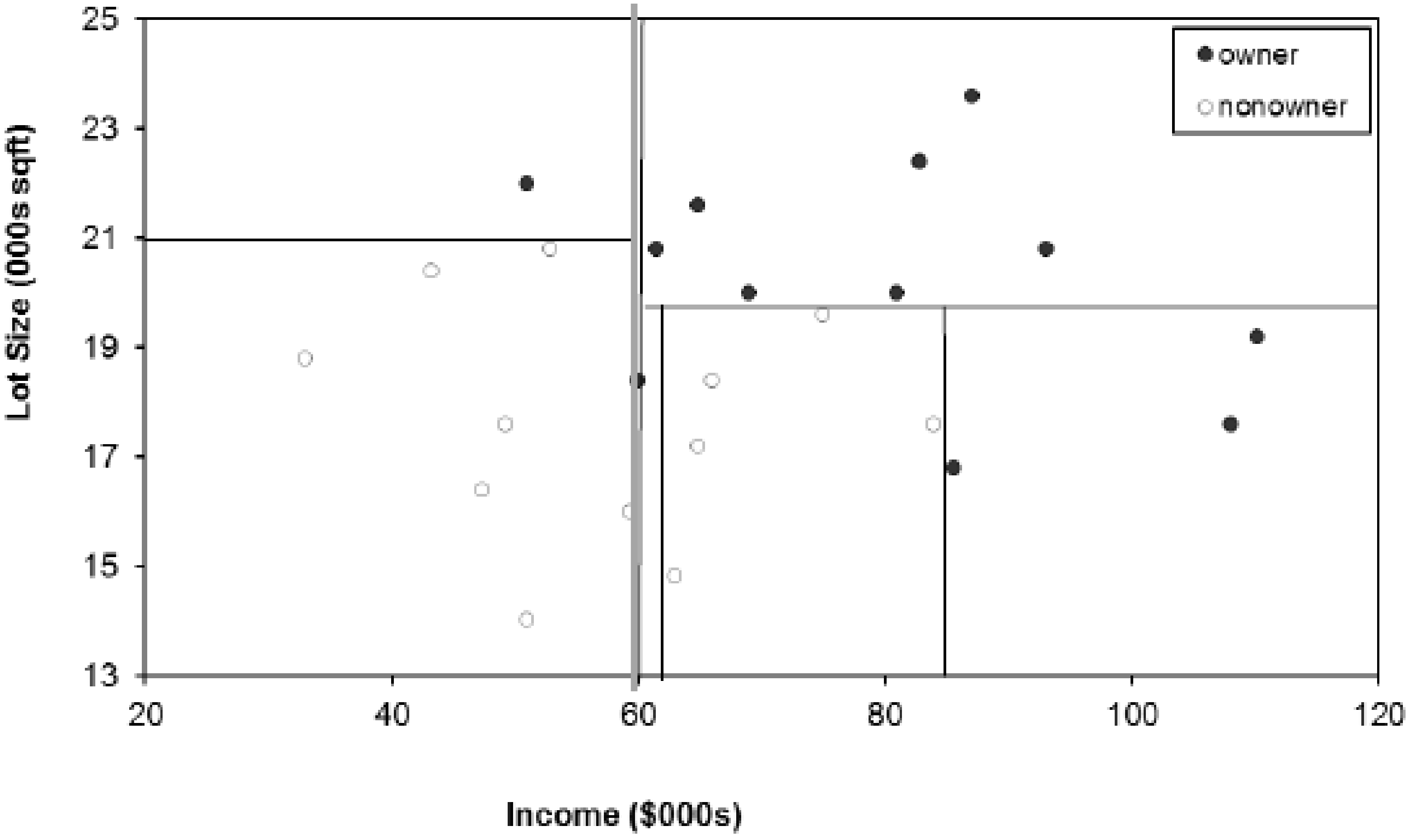
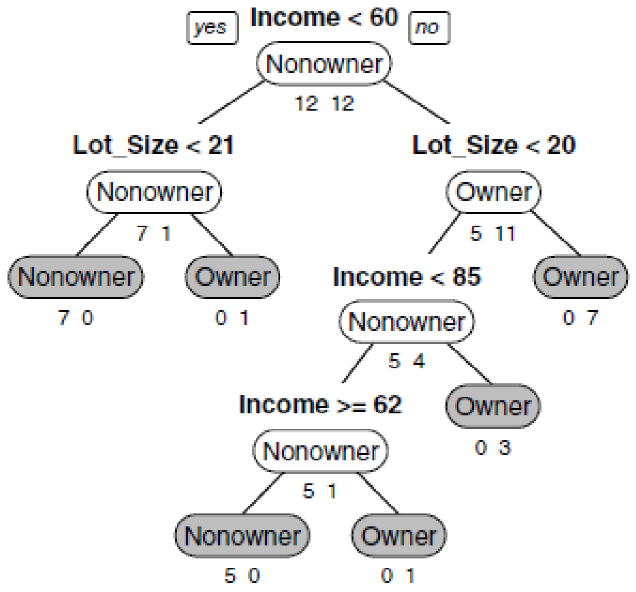


FIGURE 9.6 FINAL STAGE OF RECURSIVE PARTITIONING; EACH RECTANGLE CONSISTING OF A SINGLE CLASS (OWNERS OR NONOWNERS)



- Axis-parallel decision boundaries
- Hyper-rectangular decision regions corresponding to each leaf node

Ref: Data Mining for Business Analytics: Concepts, Techniques and Applications in R, by Galit Shmueli et al., Wiley India, 2018.

Splitting Criteria and Recursive Partitioning

- Obtain overall impurity measure (weighted avg. of individual rectangles)
  - Entropy
  - Gini Index
  - Misclassification Error

- At each successive stage, compare this measure across all possible splits in all variables

- Choose the split that reduces impurity the most

- Chosen split points become nodes on the tree

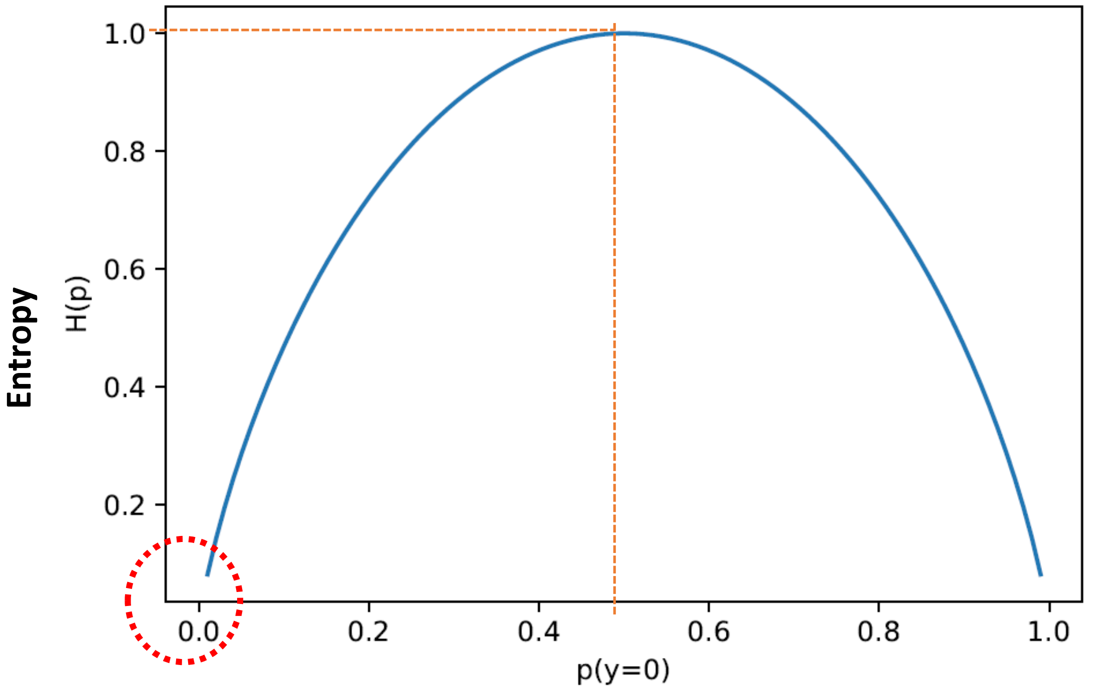
Ref: Data Mining for Business Analytics: Concepts, Techniques and Applications in R, by Galit Shmueli et al., Wiley India, 2018.

Entropy

$$entropy\,(A) = -\sum_{k=1}^{m} p_k\, log_2(p_k)$$

p = proportion of cases in rectangle *A* that belong to class *k* (out of *m* classes)

- Entropy ranges between 0 (most pure) and $\log_2(m)$ (equal representation of classes) implying for a 2 class scenario (m=2) with equal representation, entropy would be $\log_2(2) = 1$



Proportion of Records of one of the Classes in Binary Classification

Ref: Data Mining for Business Analytics: Concepts, Techniques and Applications in R, by Galit Shmueli et al., Wiley India, 2018.

Entropy

$$entropy\,(A) = -\sum_{k=1}^{m} p_k \log_2(p_k)$$

Owner = 12, Non-Owner = 12

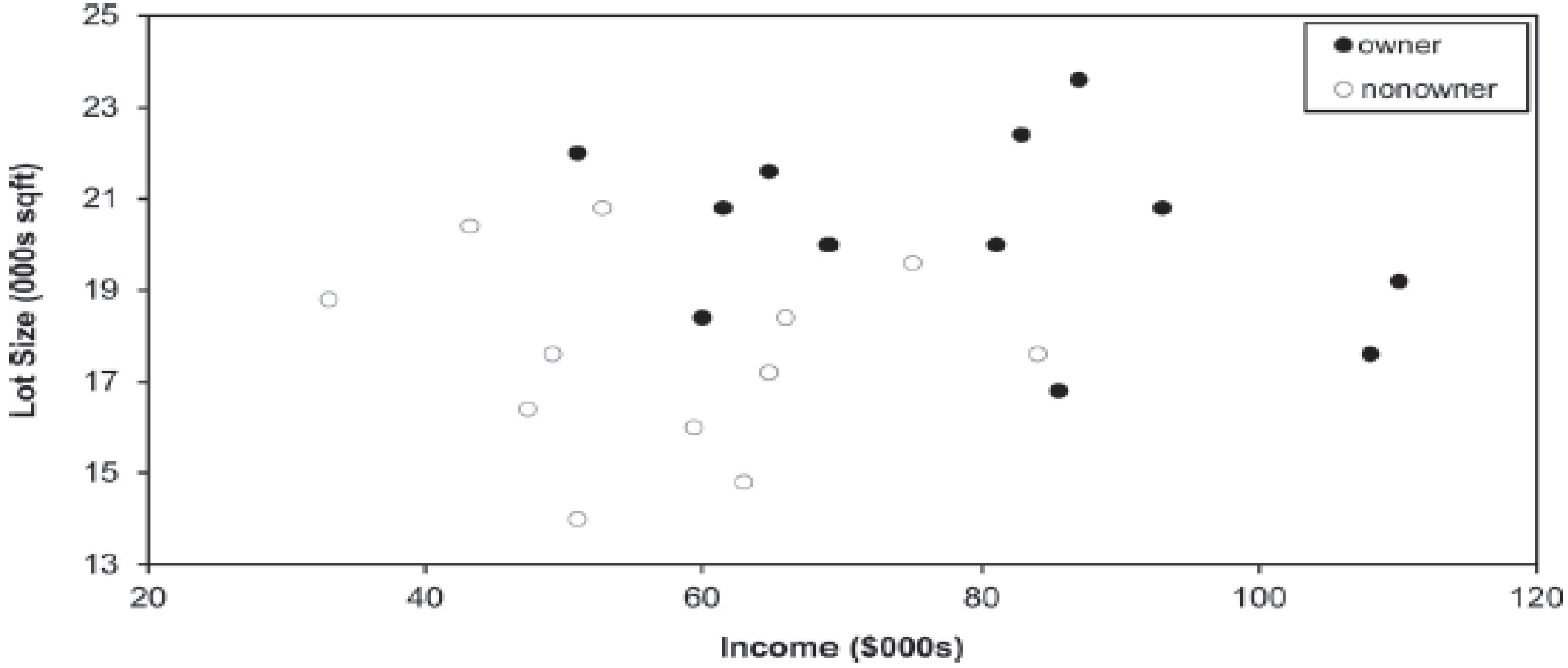


FIGURE 9.2 SCATTER PLOT OF LOT SIZE VS. INCOME FOR 24 OWNERS AND NONOWNERS OF RIDING MOWERS

Entropy = – [(12/24) $\log_2$ (12/24) + (12/24) $\log_2$ (12/24)]

= – [0.5 x -1 + 0.5 x -1]

Entropy = 1 (alternatively also calculated as $\log_2$ 2 in this case)

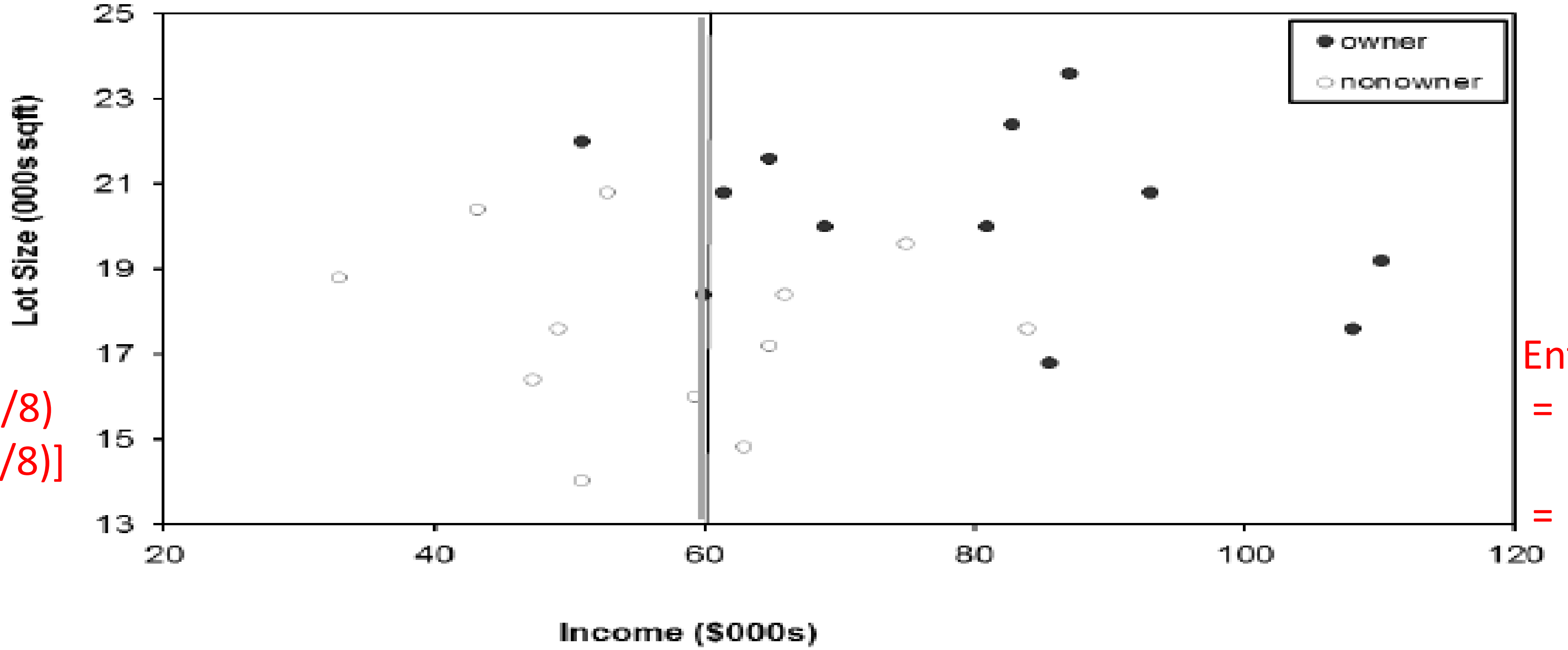
Ref: Data Mining for Business Analytics: Concepts, Techniques and Applications in R, by Galit Shmueli et al., Wiley India, 2018.

Entropy

$$entropy\,(A) = -\sum_{k=1}^{m} p_k \, log_2(p_k)$$

Owner = 1, Non-Owner = 7

Owner = 11, Non-Owner = 5



Entropy(L)
$= -[(7/8)\log_2(7/8) + (1/8)\log_2(1/8)]$
$= 0.544$

Entropy(R)
$= -[(11/16)\log_2(11/16) + (5/16)\log_2(5/16)]$
$= 0.896$

FIGURE 9.3 SPLITTING THE 24 RECORDS BY INCOME VALUE OF 60

Entropy = $(8/24)(0.544) + (16/24)(0.896) = 0.779$

Information Gain = Entropy(parent) – [Average of Entropy(children)]

Thus Entropy measure has dropped from 1 to 0.779 due to better homegeneity after this split, resulting in a +ve information gain

Ref: Data Mining for Business Analytics: Concepts, Techniques and Applications in R, by Galit Shmueli et al., Wiley India, 2018.

Gini Index

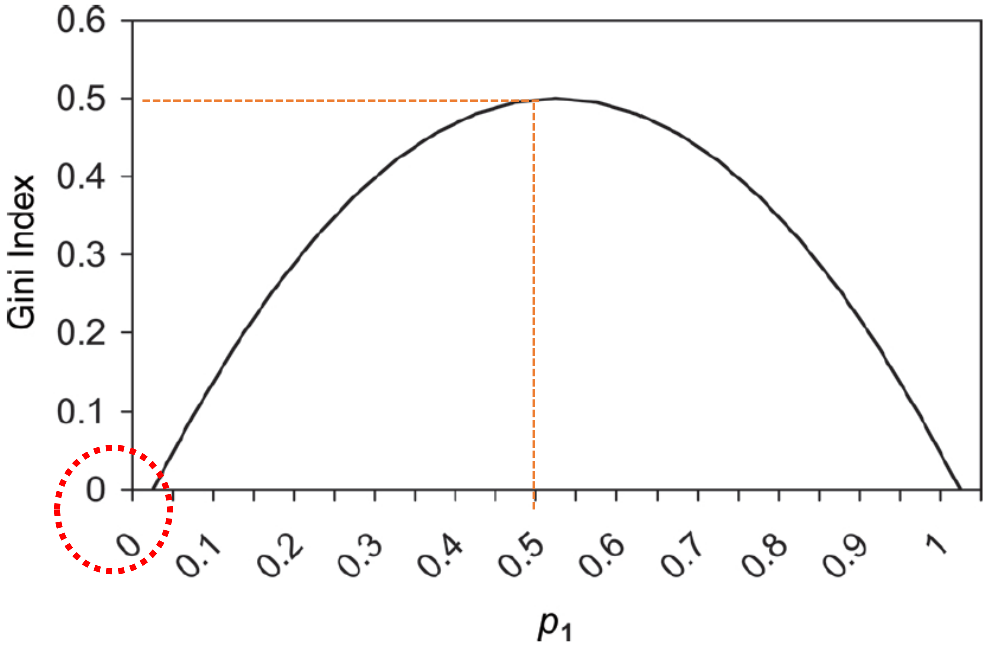
Gini Impurity Index for rectangle *A*

$$I(A) = 1 - \sum_{k=1}^{m} p_k^2$$

p = proportion of cases in rectangle *A* belonging to class *k* (out of *m* classes)

- I(A) = 0 when all cases belong to same class
- Max value when all classes are equally represented (= 0.50 in binary case)

- Gini is computationally more efficient to compute than entropy (due to the lack of the log), which could make code negligibly more efficient in terms of computational performance.



VALUES OF THE GINI INDEX FOR A TWO-CLASS CASE AS A FUNCTION OF THE PROPORTION OF RECORDS IN CLASS 1 (p_1)

Ref: Data Mining for Business Analytics: Concepts, Techniques and Applications in R, by Galit Shmueli et al., Wiley India, 2018.

Gini Score $I(A) = 1 - \sum_{k=1}^{m} p_k^2$

Owner = 12, Non-Owner = 12

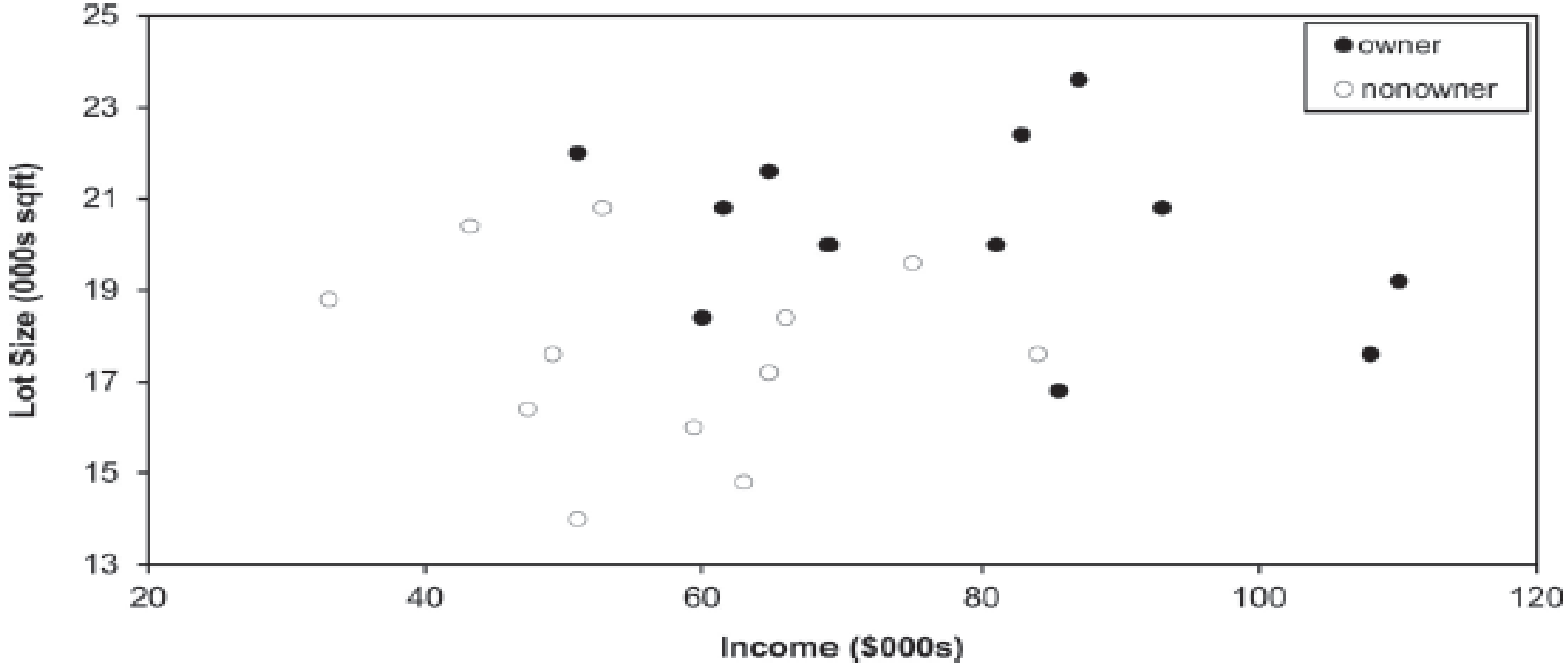


FIGURE 9.2 SCATTER PLOT OF LOT SIZE VS. INCOME FOR 24 OWNERS AND NONOWNERS OF RIDING MOWERS

Gini = $1 - (12/24)^2 - (12/24)^2 = 1 - (1/2)^2 - (1/2)^2 = 1 - (1/4) - (1/4)$

Gini = 0.5

Ref: Data Mining for Business Analytics: Concepts, Techniques and Applications in R, by Galit Shmueli et al., Wiley India, 2018.

Gini Score $I(A) = 1 - \sum_{k=1}^{m} p_k^2$

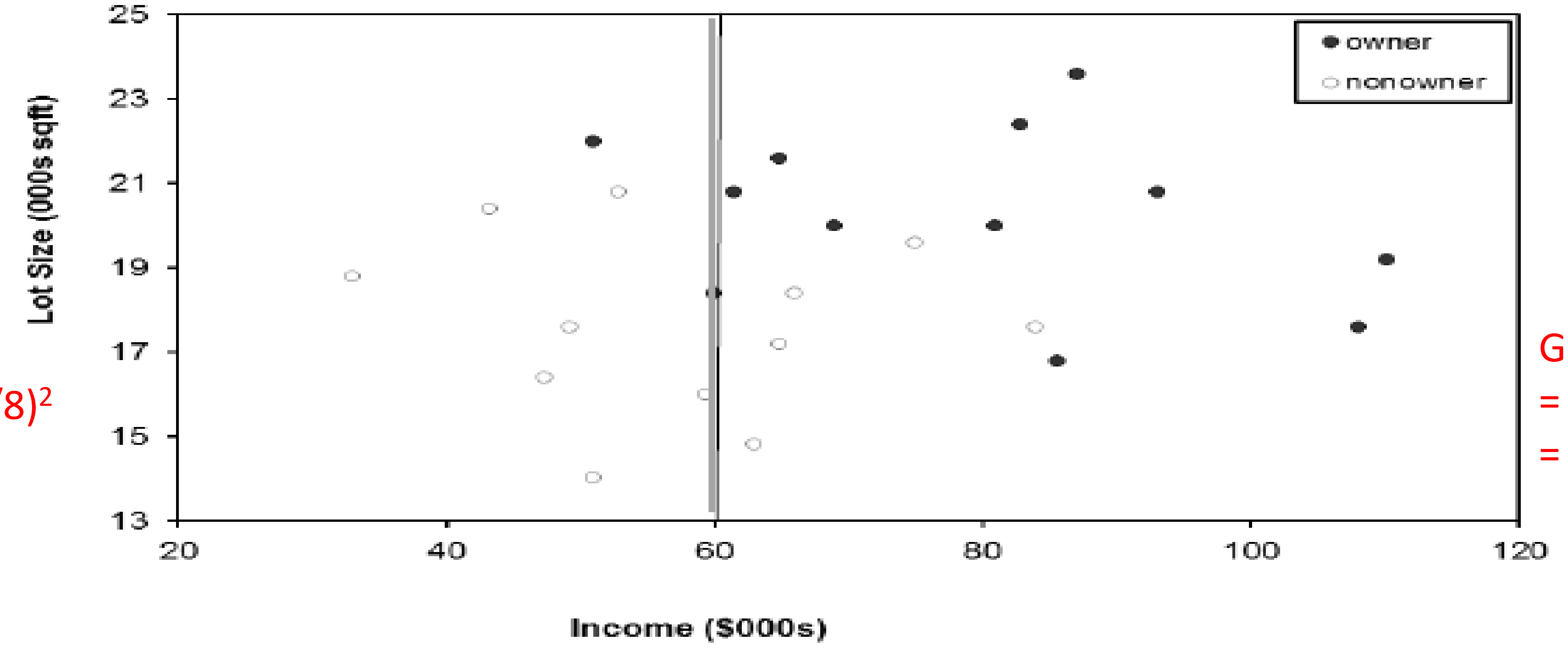
Owner = 1, Non-Owner = 7

Owner = 11, Non-Owner = 5

Gini_L

$= 1 – (7/8)^2 - (1/8)^2$

$= 0.219$



Gini_R

$= 1 – (11/16)^2 - (5/16)^2$

$= 0.430$

FIGURE 9.3 SPLITTING THE 24 RECORDS BY INCOME VALUE OF 60

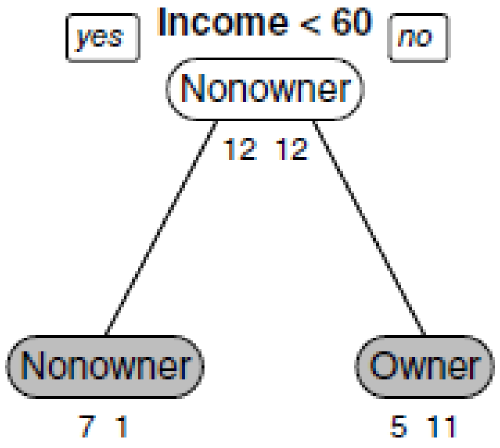
Gini = (8/24)(0.219) + (16/24)(0.430) = 0.359

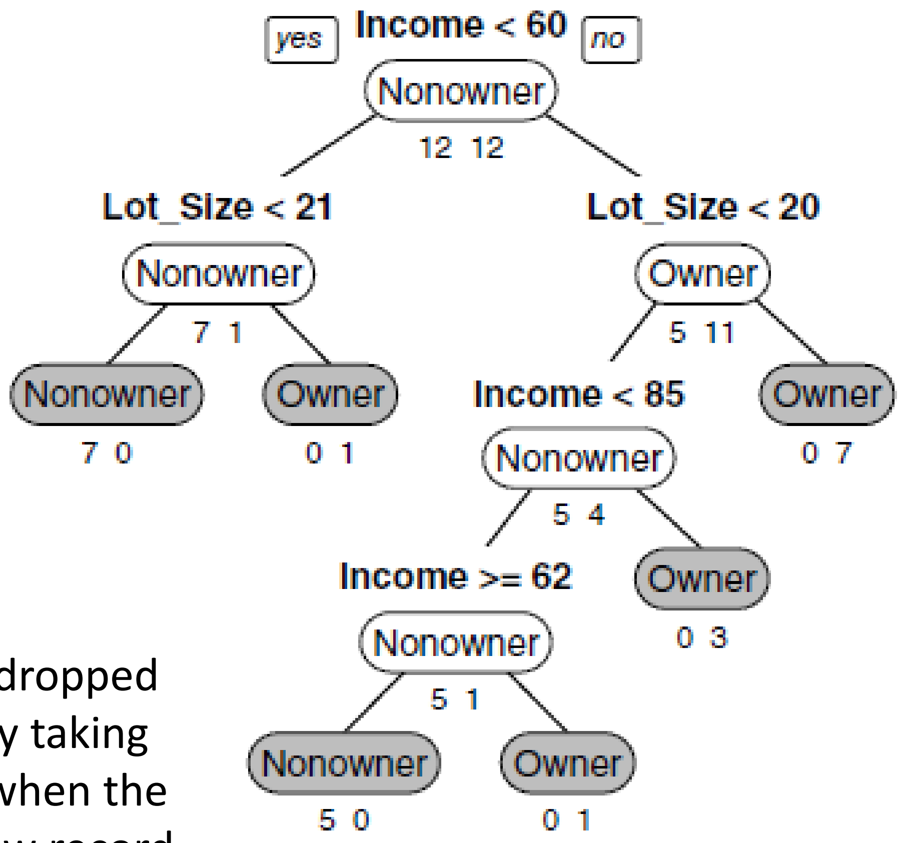
Thus Gini score measuring Impurity has dropped from 0.5 to 0.359 after this split indicating better homegeneity

Ref: Data Mining for Business Analytics: Concepts, Techniques and Applications in R, by Galit Shmueli et al., Wiley India, 2018.

First Split of The Tree

Tree after all splits





To classify a new record, it is "dropped" down the tree. When it has dropped all the way down to a terminal node, we can assign its class simply by taking a "vote" of all the training data that belonged to the terminal node when the tree was grown. The class with the highest vote is assigned to the new record. For instance, a new record reaching the rightmost terminal node which has a majority of records that belong to the owner class, would be classified as "owner.

Ref: Data Mining for Business Analytics: Concepts, Techniques and Applications in R, by Galit Shmueli et al., Wiley India, 2018.

Misclassification Error – Another Splitting Criteria

- Measures the impurity error
- Instead of using Entropy as an impurity measure, the misclassification error ERR is used,

$$GAIN(\mathcal{D}, xj) = ERR(\mathcal{D}) - \sum_{v \in Values(x_j)} \frac{|\mathcal{D}_v|}{|\mathcal{D}|} ERR(\mathcal{D}_v)$$

$$ERR(\mathcal{D}) = \frac{1}{n} \sum_{i=1}^{n} L(\hat{y}^{[i]}, y^{[i]})$$

with the 0-1 Loss,

$$L(\hat{y}^{[i]}, y^{[i]}) = \begin{cases} 0 \; if \; \hat{y} = y \\ 1 \; otherwise \end{cases}$$

This is case of the training set is equal to

$$ERR(p) = 1 - \max((p(i / x_j))$$

for a given node if **we use majority voting at this node**

Reference: Sebastian Raschka, STAT 451 Machine Learning Notes, http://stat.wisc.edu/~sraschka/teaching/stat451-fs2020/

Misclassification Error based Splitting

Owner = 1, Non-Owner = 7   Owner = 11, Non-Owner = 5

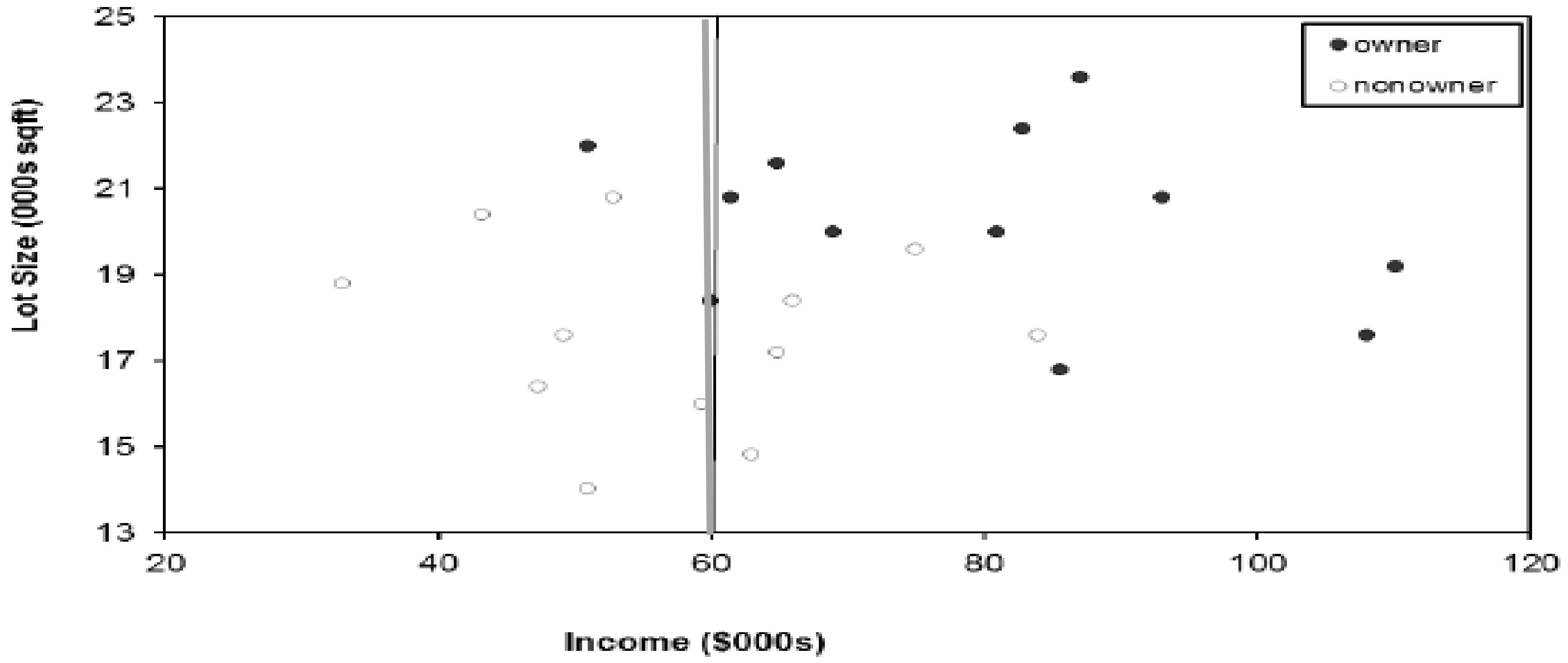
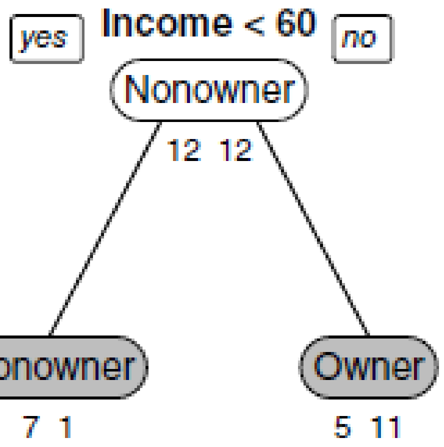


FIGURE 9.3 SPLITTING THE 24 RECORDS BY INCOME VALUE OF 60

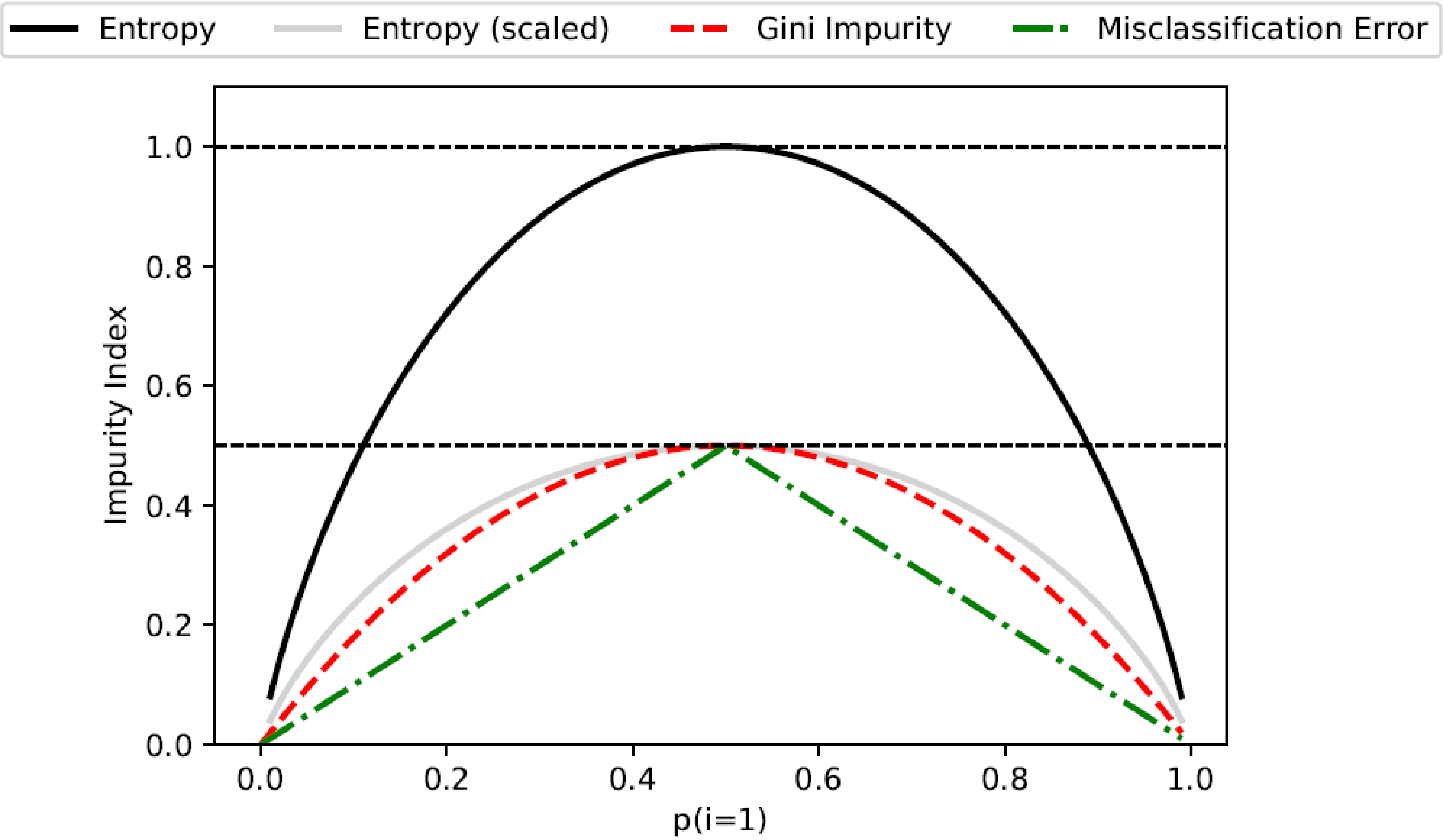


= 12/24 – 8/24*1/8 – 16/24*5/16
= 1/2 – 1/24 – 5/24
= 1/2 – 6/24
= 1/2 – 1/4
= 1/4

$$GAIN(\mathcal{D}, xj) = \mathrm{ERR}(\mathcal{D}) - \sum_{v \in Values(x_j)} \frac{|\mathcal{D}_v|}{|\mathcal{D}|} ERR(\mathcal{D}_v)$$

Sometimes, the information gain upon splitting the root node using the misclassification error as impurity metric turns out to be 0, thus misleading the algorithm to stop growing, when it should have ideally proceeded with more splits

Comparison of different impurity measures.



Reference: Sebastian Raschka, STAT 451 Machine Learning Notes, http://stat.wisc.edu/~sraschka/teaching/stat451-fs2020/

Decision Tree Algorithms – A Comparison

CART	ID3: Iterative Dichotomizer 3	C4.5
• L. Breiman (1984).	• J. R. Quinlan (1986)	• J.R. Quinlan (1993)
• discrete and continuous (numeric) features	• discrete only, cannot handle numeric features	• discrete and continuous (though latter is expensive)
• strictly binary splits (taller trees than ID3, C4.5)	• short and wide trees (compared to CART)	• Handles missing attributes
• Gini Impurity (CT) / Variance reduction (RT)	• Maximize Information gain / Minimize Entropy	• Gain Ratio (penalizes splitting categorical attributes with many values)
• cost complexity pruning	• no pruning, prone to overfitting	• post-pruning (bottom-up pruning)

Other algorithms include CHAID (CHi-squared Automatic Interaction Detector) - G. V. Kass, (1980)/ MARS (Multivariate adaptive regression splines) - J. H. Friedman (1991) / C5.0 (PATENTED), etc.

Reference: Sebastian Raschka, STAT 451 Machine Learning Notes, http://stat.wisc.edu/~sraschka/teaching/stat451-fs2020/

sklearn.tree.DecisionTreeClassifier

- Scikit-learn uses an optimised version of the CART algorithm

  Ref: https://scikit-learn.org/stable/modules/tree.html

class sklearn.tree.DecisionTreeClassifier(*, **criterion='gini'**, splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, class_weight=None, presort='deprecated', ccp_alpha=0.0)

Ref: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

Summary

- Decision Trees are among the simplest algorithm to understand,
- Powerful "White box" technique that can fit complex datasets
- Non-parametric approach
- Can operate without much pre-processing
- Splitting criteria varies according to algorithm is usually based on information gain (Entropy, Gini impurity, misclassification error)
- Key algorithms are ID3, C4.5, CART
- Algorithms vary in terms of binary split vs. multi-way splits, capability of handling of discrete vs. continuous variables, pre vs. post-pruning

References

- https://scikit-learn.org/stable/modules/tree.html
- https://towardsdatascience.com/decision-tree-overview-with-no-maths-66b256281e2b
- https://towardsdatascience.com/decision-tree-part-2-34b31b1dc328
- https://towardsdatascience.com/understanding-decision-tree-classification-with-scikit-learn-2ddf272731bd