CLASSIFICATION

Classification

- **Decision tree and CART**

Decision Tree...introduction

- Decision trees or recursive partitioning models are a decision support tool which uses a tree like consequences
- A decision tree creates a type of flowchart which consists of nodes (or leafs) and a set of decisions to be made based on node (or branches)
- The leaf and branch structure forms a hierarchical representation that mimic the form of a tree
- Decision tree learning is one of the most widely used and practical method for inductive inference and is an important tool in machine learning and predictive analysis.

Decision Tree...advantages

- They are simple to understand and interpret. People are able to understand decision tree models after a brief explanation
- Important insights can be generated based on experts describing a situation (like alternatives, probabilities and costs) and their preferences for outcomes.
- The algorithms are robust to noisy data and capable of learning disjunctive expressions.
- Help determine worst, best and expected values for difference scenarios.

Decision Tree...disadvantages

- For data including categorical variables with different variables with different number of levels, information gain in decision tree are biased in favor of those attributes with more levels.
- Calculations can get very complex particularly if many values are uncertain and/or if many outcomes are linked

Decision Tree...types

- Classification tree – for qualitative response.
- Regression tree – for quantitative response.

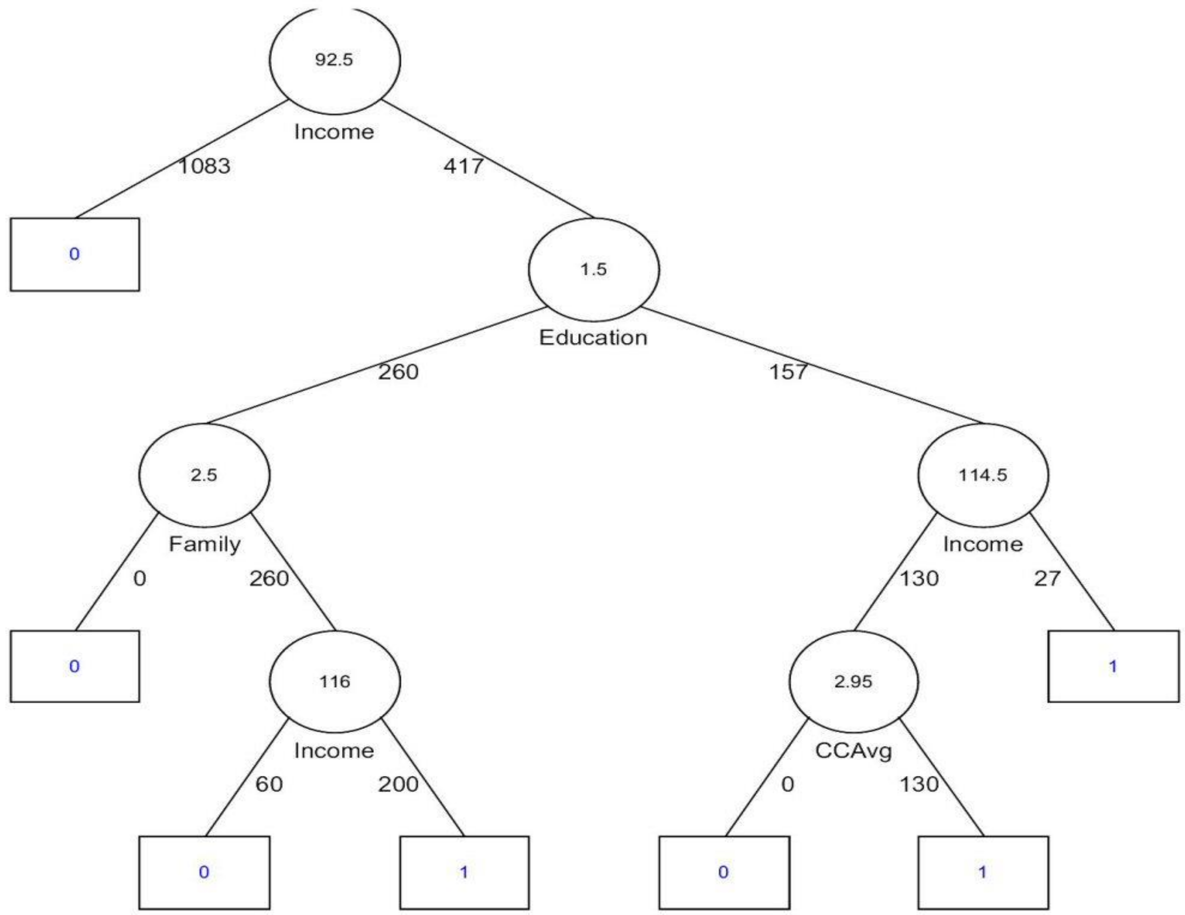
Decision Tree... Trees and Rules

Goal: Classify or predict an outcome based on a set of predictors. The output is a set of **rules**

Example:

- Goal: classify a record as "will accept credit card offer" or "will not accept"
- Rule might be "IF (Income > 92.5) AND (Education < 1.5) AND (Family <= 2.5) THEN Class = 0 (non-acceptor)
- Also called CART, Decision Trees, or just Trees
- Rules are represented by tree diagrams

Decision Tree...



Decision Tree...key idea

Recursive partitioning: Repeatedly split the records into two parts so as to achieve maximum homogeneity within the new parts

Pruning the tree: Simplify the tree by pruning peripheral branches to avoid over-fitting

Decision Tree... Recursive Partitioning Steps

- Pick one of the predictor variables, x_i
- Pick a value of x_i, say s_i, that divides the training data into two (not necessarily equal) portions
- Measure how "pure" or homogeneous each of the resulting portions are

  "Pure" = containing records of mostly one class
- Algorithm tries different values of x_i, and s_i to maximize purity in initial split
- After you get a "maximum purity" split, repeat the process for a second split, and so on

Decision Tree...Example: Riding Mowers

- Goal: Classify 24 households as owning or not owning
- Predictors = Income, Lot Size

Decision Tree

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

Decision Tree...How to split

- Order records according to one variable, say lot size
- Find midpoints between successive values
  E.g. first midpoint is 14.4 (halfway between 14.0 and 14.8)
- Divide records into those with lot size > 14.4 and those < 14.4
- After evaluating that split, try the next one, which is 15.4 (halfway between 14.8 and 16.0)

Decision Tree... splitting

Note: Categorical Variables

Examine all possible ways in which the categories can be split.

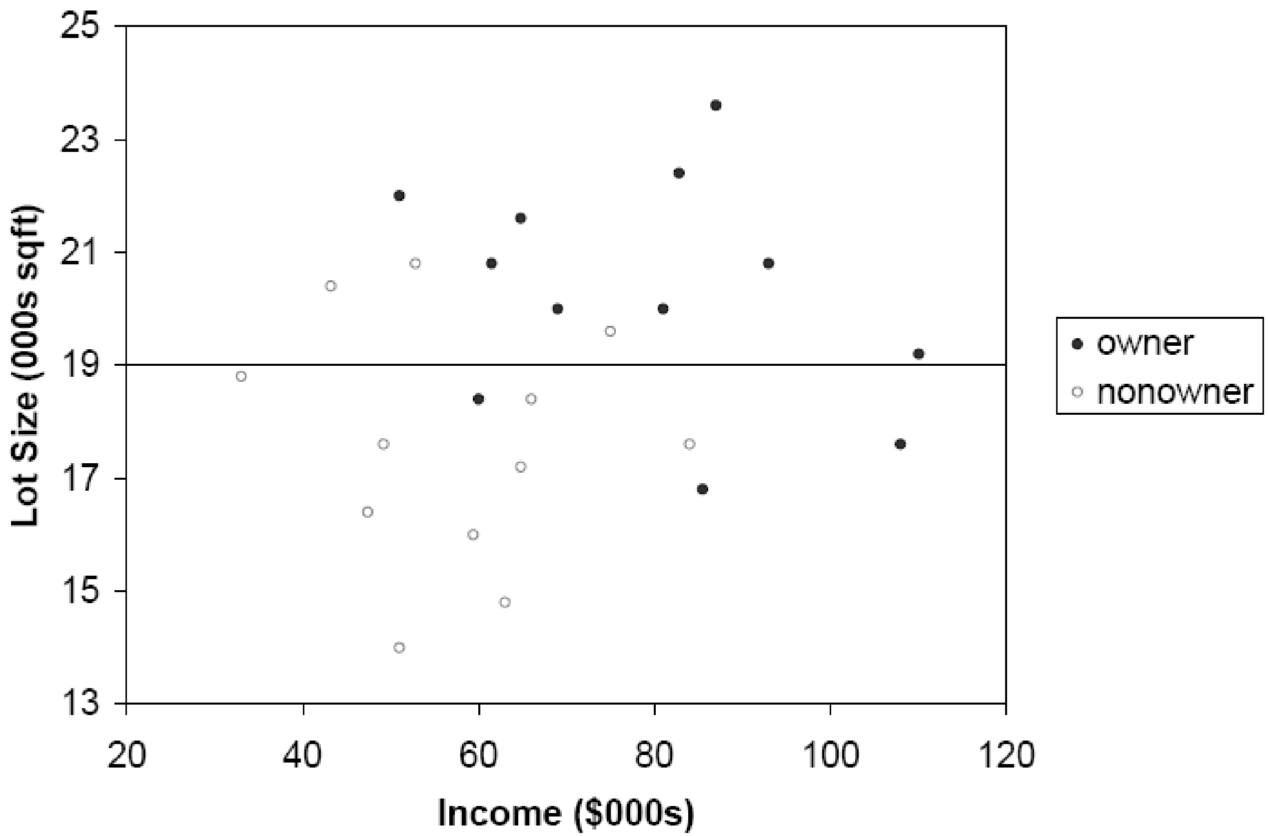
E.g., categories A, B, C can be split 3 ways

{A} and {B, C}
{B} and {A, C}
{C} and {A, B}

With many categories, # of splits becomes huge

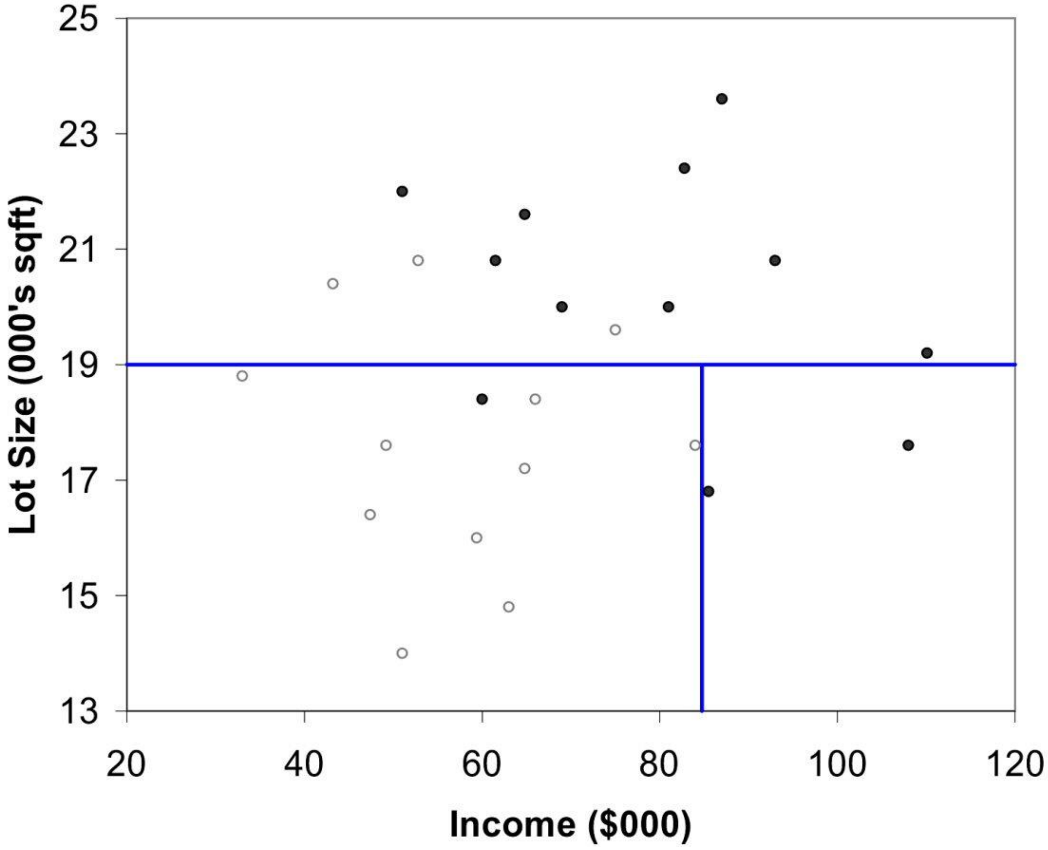
Decision Tree... splitting

The first split: Lot Size = 19,000



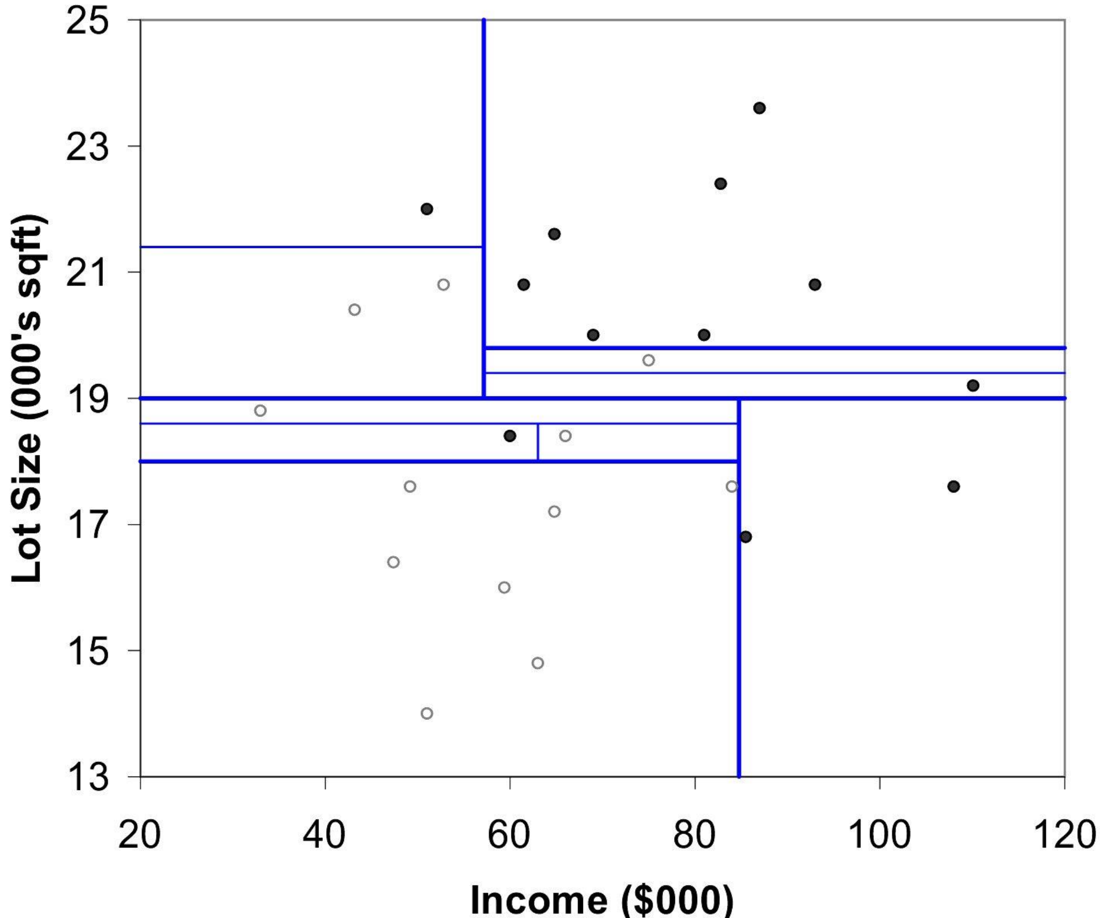
Decision Tree... splitting

Second Split: Income = $84,000



Decision Tree... splitting

After All Splits



Decision Tree... Measuring Impurity

Gini Index

Gini Index for rectangle *A* containing *m* records

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

p = proportion of cases in rectangle *A* that belong to class *k*

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

Decision Tree... Measuring Impurity

Entropy

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

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

Entropy ranges between 0 (most pure) and $\log_2(m)$ (equal representation of classes)

Decision Tree... Impurity and Recursive Partitioning

Obtain overall impurity measure (weighted avg. of individual rectangles)

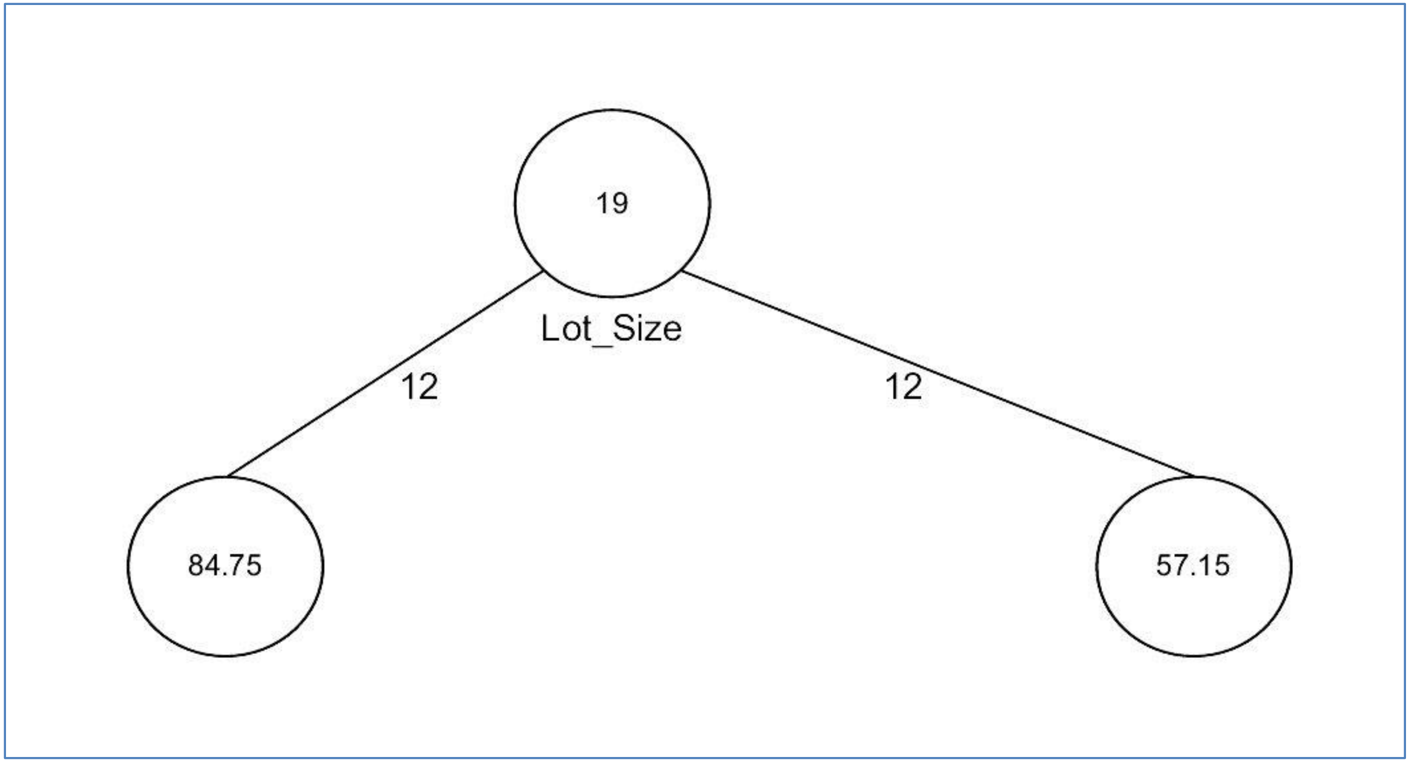
At each successive stage, compare this measure across all possible splits in all variables

Choose the split that reduces impurity the most

Chosen split points become nodes on the tree

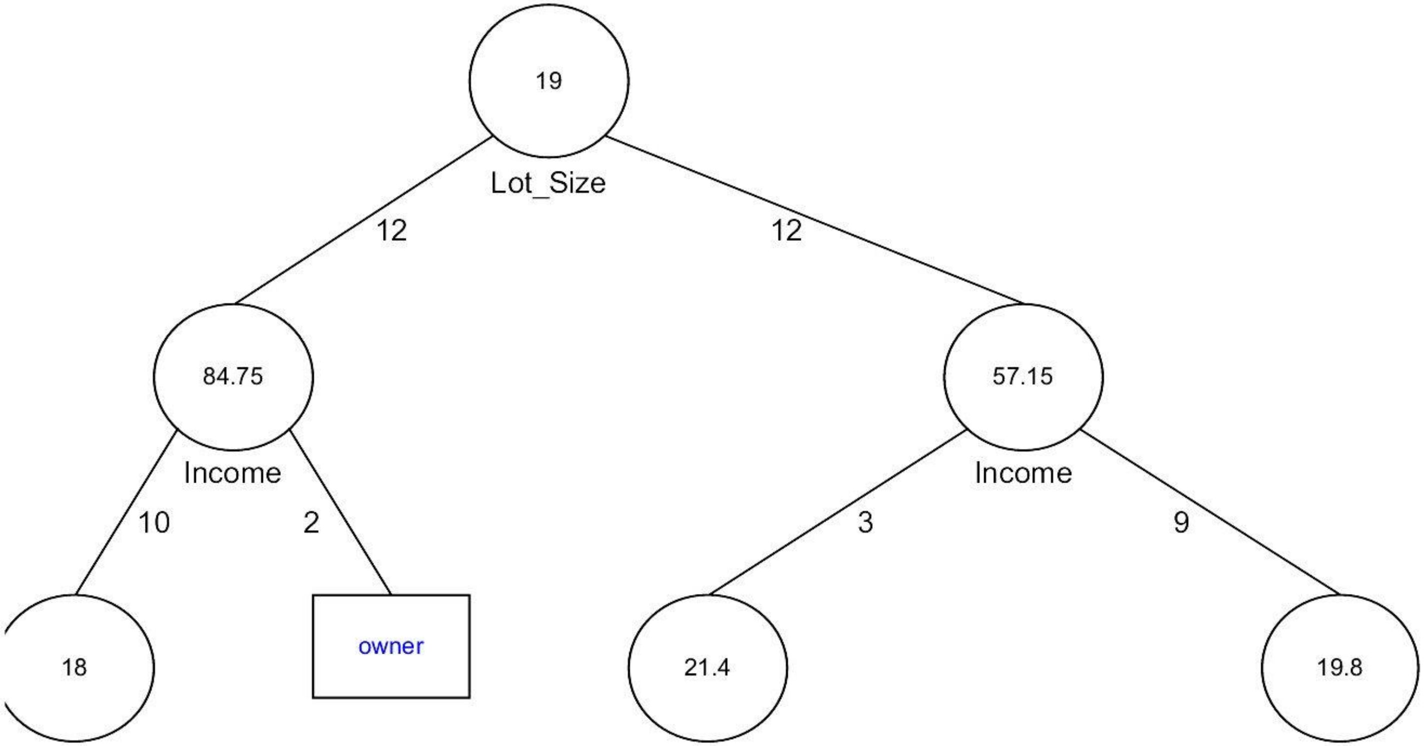
Decision Tree

First Split – The Tree



Decision Tree

Tree after three splits



Decision Tree

Tree Structure

Split points become nodes on tree (circles with split value in center)

Rectangles represent "leaves" (terminal points, no further splits, classification value noted)

Numbers on lines between nodes indicate # cases

Read down tree to derive rule

E.g., If lot size < 19, and if income > 84.75, then class = "owner"

Decision Tree

Determining Leaf Node Label

Each leaf node label is determined by “voting” of the records within it, and by the cutoff value

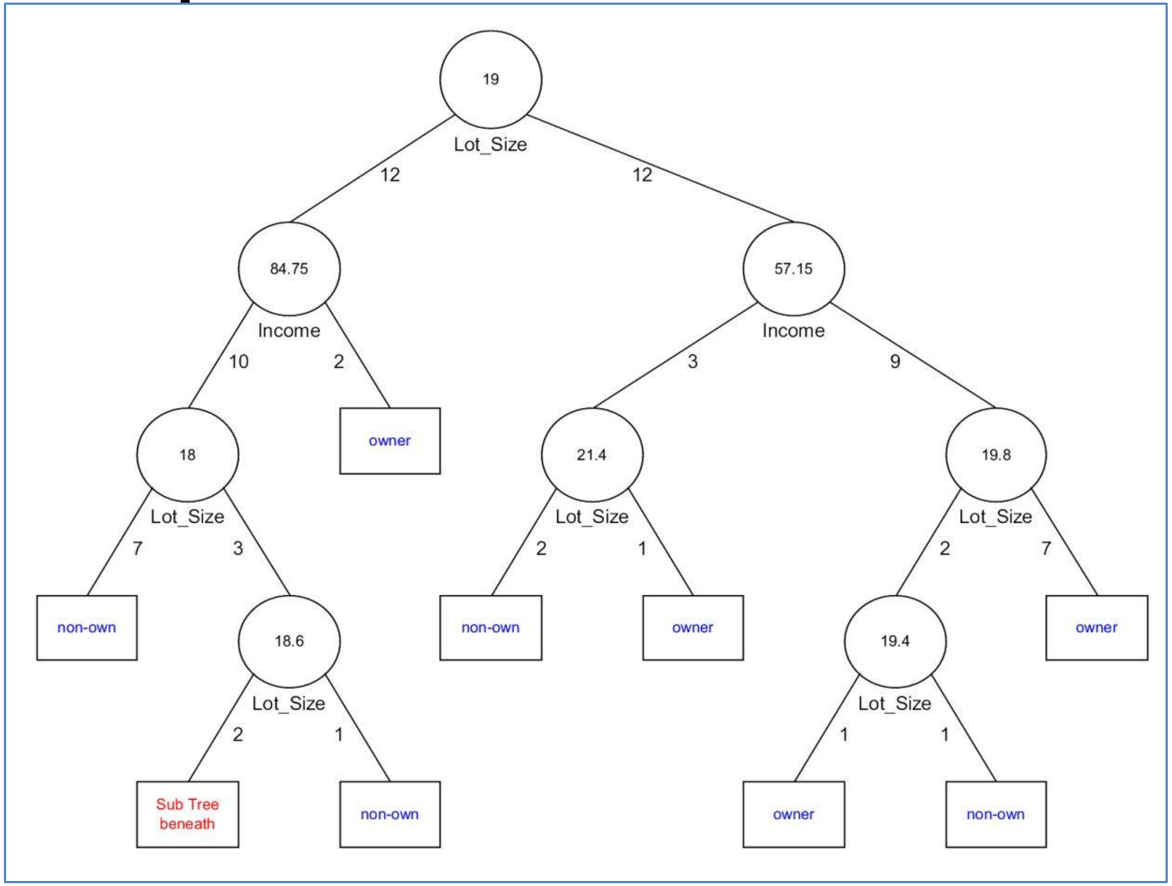
Records within each leaf node are from the training data

Default cutoff=0.5 means that the leaf node’s label is the majority class.

Cutoff = 0.75: requires majority of 75% or more “1” records in the leaf to label it a “1” node

Decision Tree

Tree after all splits



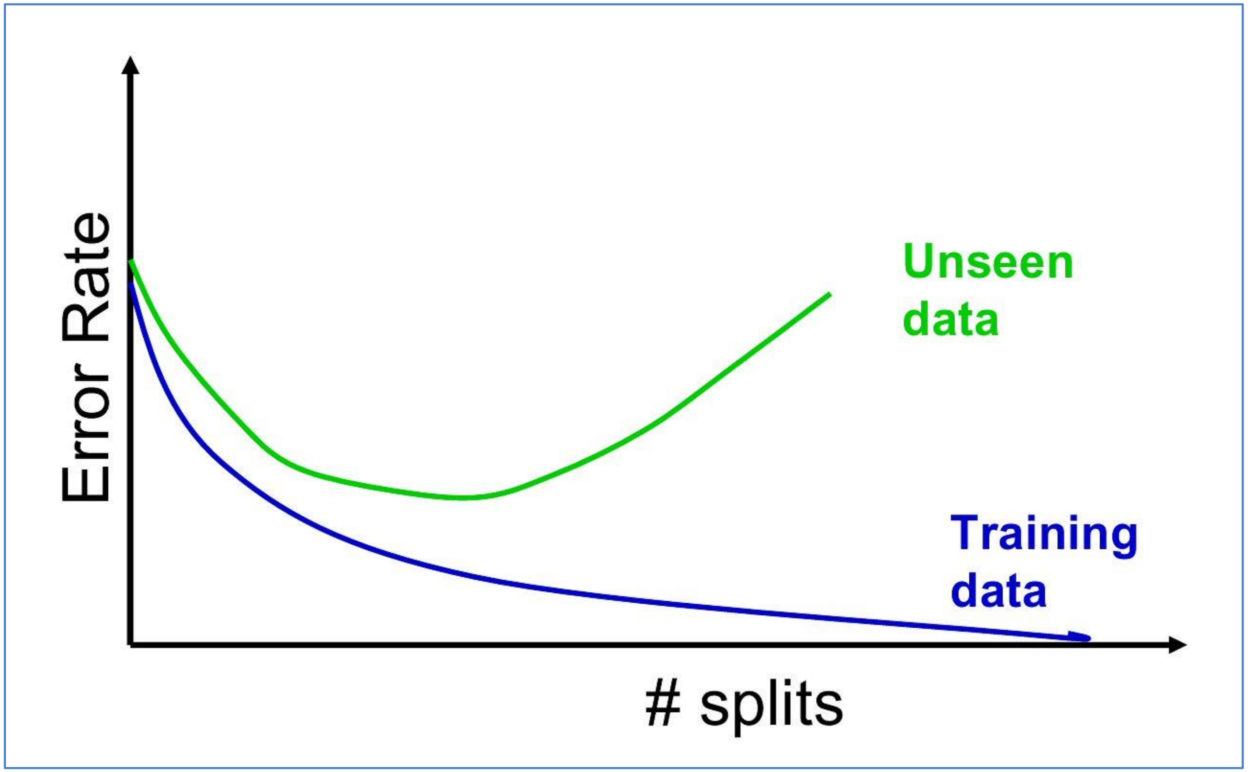
Decision Tree... The Overfitting Problem

Stopping Tree Growth

- Natural end of process is 100% purity in each leaf
- This **overfits** the data, which end up fitting noise in the data
- Overfitting leads to low predictive accuracy of new data
- Past a certain point, the error rate for the validation data starts to increase

Decision Tree

Full Tree Error Rate



Decision Tree

CHAID (*Chi*-squared Automatic Interaction Detector)

CHAID, older than CART, uses chi-square statistical test to limit tree growth

Splitting stops when purity improvement is not statistically significant

Decision Tree

Pruning

- CART lets tree grow to full extent, then prunes it back
- Idea is to find that point at which the validation error begins to rise
- Generate successively smaller trees by pruning leaves
- At each pruning stage, multiple trees are possible
- Use *cost complexity* to choose the best tree at that stage

Decision Tree

Cost Complexity

$$CC(T) = Err(T) + a\,L(T)$$

$CC(T)$ = cost complexity of a tree
$Err(T)$ = proportion of misclassified records
a = penalty factor attached to tree size (set by user)

Among trees of given size, choose the one with lowest CC
Do this for each size of tree

Decision Tree

Using Validation Error to Prune

Pruning process yields a set of trees of different sizes and associated error rates

Two trees of interest:
Minimum error tree
  Has lowest error rate on validation data
Best pruned tree
  Smallest tree within one std. error of min. error
  This adds a bonus for simplicity/parsimony

Decision Tree...regression

Regression Trees for Prediction

- Used with continuous outcome variable
- Procedure similar to classification tree
- Many splits attempted, choose the one that minimizes impurity

Decision Tree...regression

Differences from CT

Prediction is computed as the **average** of numerical target variable in the rectangle (in CT it is majority vote)

Impurity measured by **sum of squared deviations** from leaf mean

Performance measured by RMSE (root mean squared error)

Decision Tree...regression

Advantages of trees

- Easy to use, understand
- Produce rules that are easy to interpret & implement
- Variable selection & reduction is automatic
- Do not require the assumptions of statistical models
- Can work without extensive handling of missing data

Decision Tree...regression

Disadvantages

- May not perform well where there is structure in the data that is not well captured by horizontal or vertical splits
- Since the process deals with one variable at a time, no way to capture interactions between variables

Decision Tree

Summary

- Classification and Regression Trees are an easily understandable and transparent method for predicting or classifying new records
- A tree is a graphical representation of a set of rules
- Trees must be pruned to avoid over-fitting of the training data
- As trees do not make any assumptions about the data structure, they usually require large samples

Decision Tree

R/Python Session for Decision tree

Thank You