

Decision Trees

By Susan Miertschin

An Algorithm for Building Decision Trees

- C4.5 is a computer program for inducing classification rules in the form of decision trees from a set of given instances
- C4.5 is a software extension of the basic ID3 algorithm designed by Quinlan

Algorithm Description

- Select one attribute from a set of training instances
- Select an initial subset of the training instances
- Use the attribute and the subset of instances to build a decision tree
- Use the rest of the training instances (those not in the subset used for construction) to test the accuracy of the constructed tree
- If all instances are correctly classified – stop
- If an instances is incorrectly classified, add it to the initial subset and construct a new tree
- Iterate until
 - A tree is built that classifies all instance correctly
 - OR
 - A tree is built from the entire training set

Simplified Algorithm

- Let T be the set of training instances
- Choose an attribute that best differentiates the instances contained in T (C4.5 uses the Gain Ratio to determine)
- Create a tree node whose value is the chosen attribute
 - Create child links from this node where each link represents a unique value for the chosen attribute
 - Use the child link values to further subdivide the instances into subclasses

Example

Credit Card Promotion Data from Chapter 2

Example – Credit Card Promotion Data Descriptions

Attribute Name	Value Description	Numeric Values	Definition
Income Range	20-30K, 30-40K, 40-50K, 50-60K	20000, 30000, 40000, 50000	Salary range for an individual credit card holder
Magazine Promotion	Yes, No	1, 0	Did card holder participate in magazine promotion offered before?
Watch Promotion	Yes, No	1, 0	Did card holder participate in watch promotion offered before?
Life Ins Promotion	Yes, No	1, 0	Did card holder participate in life insurance promotion offered before?
Credit Card Insurance	Yes, No	1, 0	Does card holder have credit card insurance?
Sex	Male, Female	1, 0	Card holder's gender
Age	Numeric	Numeric	Card holder's age in whole years

Problem to be Solved from Data

- Acme Credit Card Company is going to do a life insurance promotion – sending the promo materials with billing statements. They have done a similar promotion in the past, with results as represented by the data set. They want to target the new promo materials to credit card holders similar to those who took advantage of the prior life insurance promotion.
- Use supervised learning with output attribute = life insurance promotion to develop a profile for credit card holders likely to accept the new promotion.

Sample of Credit Card Promotion Data (from Table 2.3)

Income Range	Magazine Promo	Watch Promo	Life Ins Promo	CC Ins	Sex	Age
40-50K	Yes	No	No	No	Male	45
30-40K	Yes	Yes	Yes	No	Female	40
40-50K	No	No	No	No	Male	42
30-40K	Yes	Yes	Yes	Yes	Male	43
50-60K	Yes	No	Yes	No	Female	38
20-30K	No	No	No	No	Female	55
30-40K	Yes	No	Yes	Yes	Male	35
20-30K	No	Yes	No	No	Male	27
30-40K	Yes	No	No	No	Male	43
30-40K	Yes	Yes	Yes	No	Female	41

Problem Characteristics

- Life insurance promotion is the output attribute
- Input attributes are income range, credit card insurance, sex, and age
 - Attributes related to the instance's response to other promotions is not useful for prediction because new credit card holders will not have had a chance to take advantage of these prior offers (except for credit card insurance which is always offered immediately to new card holders)
 - Therefore, magazine promo and watch promo are not relevant for solving the problem at hand – disregard – do not include this data in data mining

Apply the Simplified C4.5 Algorithm to the Credit Card Promotion Data

Income Range	Magazine Promo	Watch Promo	Life Ins Promo	CC Ins	Sex	Age
40-50K	Yes	No	No	No	Male	45
30-40K	Yes	Yes	Yes	No	Female	40
40-50K	No	No	No	No	Male	42
30-40K	Yes	Yes	Yes	Yes	Male	43
50-60K	Yes	No	Yes	No	Female	38
20-30K	No	No	No	No	Female	55
30-40K	Yes	No	Yes	Yes	Male	35
20-30K	No	Yes	No	No	Male	27
30-40K	Yes	No	No	No	Male	43
30-40K	Yes	Yes	Yes	No	Female	41

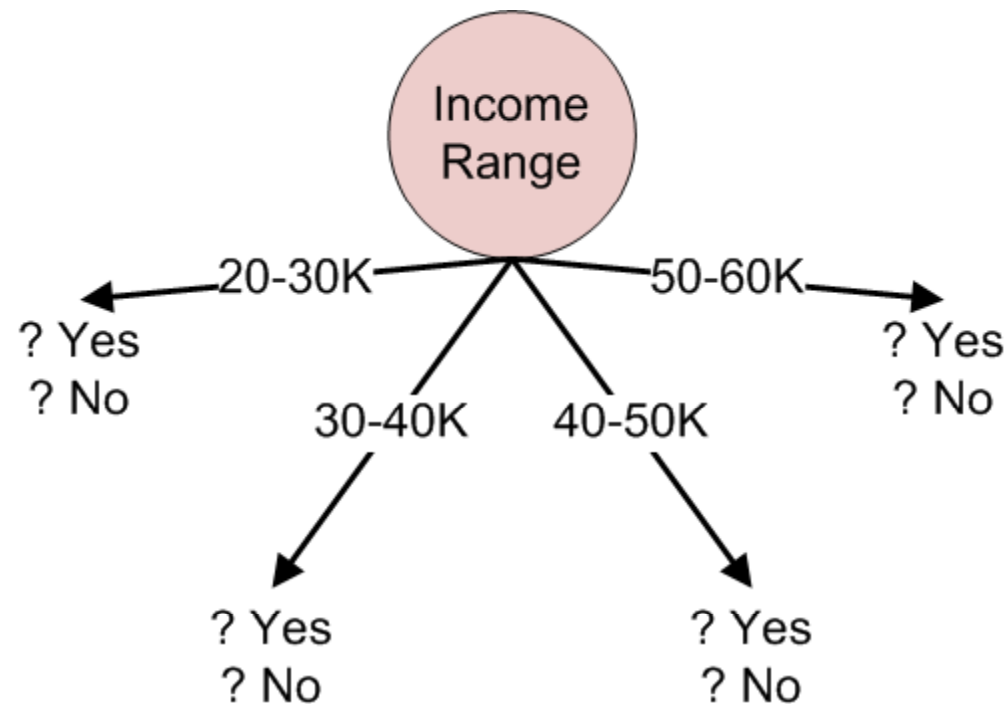
Training set = 15 instances (see handout)

Apply the Simplified C4.5 Algorithm to the Credit Card Promotion Data

Income Range	Magazine Promo	Watch Promo	Life Ins Promo	CC Ins	Sex	Age
40-50K	Yes	No	No	No	Male	45
30-40K	Yes	Yes	Yes	No	Female	40
40-50K	No	No	No	No	Male	42
30-40K	Yes	Yes	Yes	Yes	Male	43
50-60K	Yes	No	Yes	No	Female	38
20-30K	No	No	No	No	Female	55
30-40K	Yes	No	Yes	Yes	Male	35
20-30K	No	Yes	No	No	Male	27
30-40K	Yes	No	No	No	Male	43
30-40K	Yes	Yes	Yes	No	Female	41

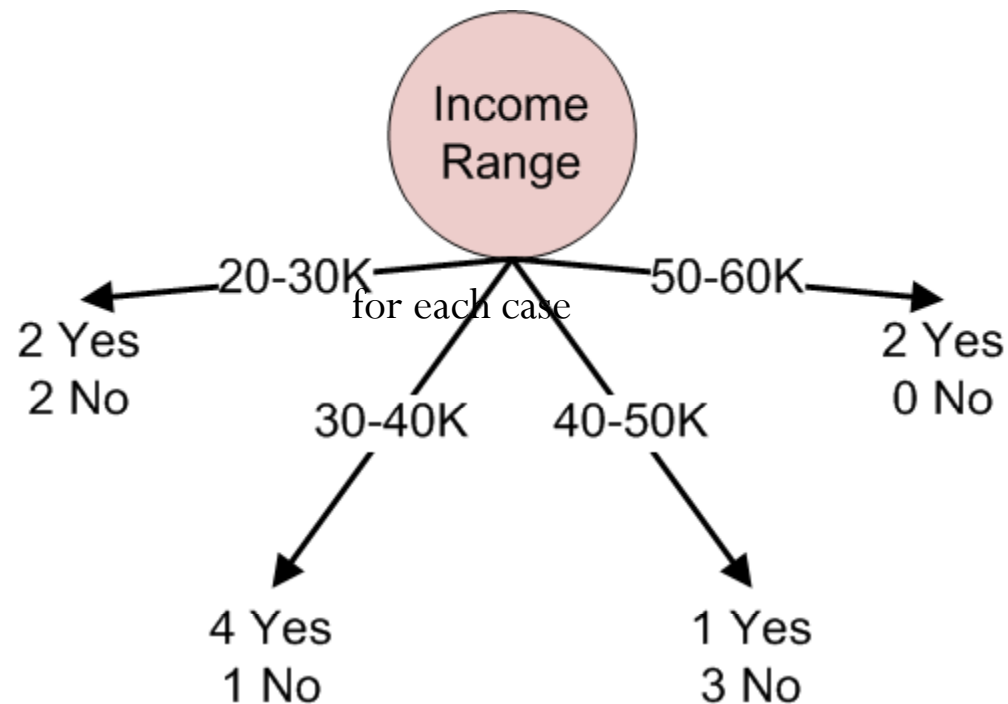
Step 2: Which input attribute best differentiates the instances?

Apply Simplified C4.5



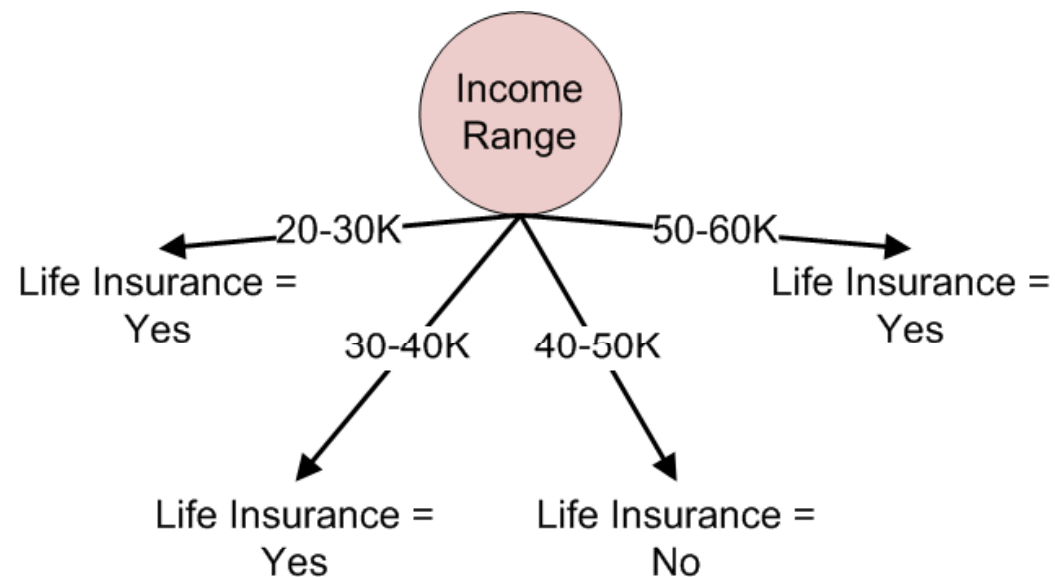
For each case (attribute value), how many instances of Life Insurance Promo = Yes and Life Insurance Promo = No?

Apply Simplified C4.5



For each branch, choose the most frequently occurring decision. If there is a tie, then choose Yes, since there are more overall Yes instances (9) than No instances (6) with respect to Life Insurance Promo

Apply Simplified C4.5



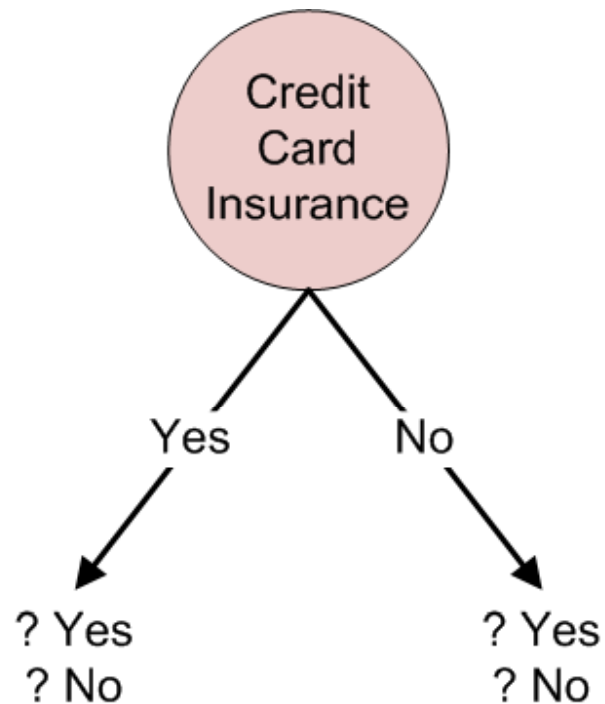
Evaluate the classification model (the tree) on the basis of accuracy. How many of the 15 training instances are classified correctly by this tree?

Apply Simplified C4.5

- Tree accuracy = $11/15 = 73.3\%$
- Tree cost = 4 branches for the computer program to use
- Goodness score for Income Range attribute is $11/15/4 = 0.183$
- Including Tree “cost” to assess goodness lets us compare trees

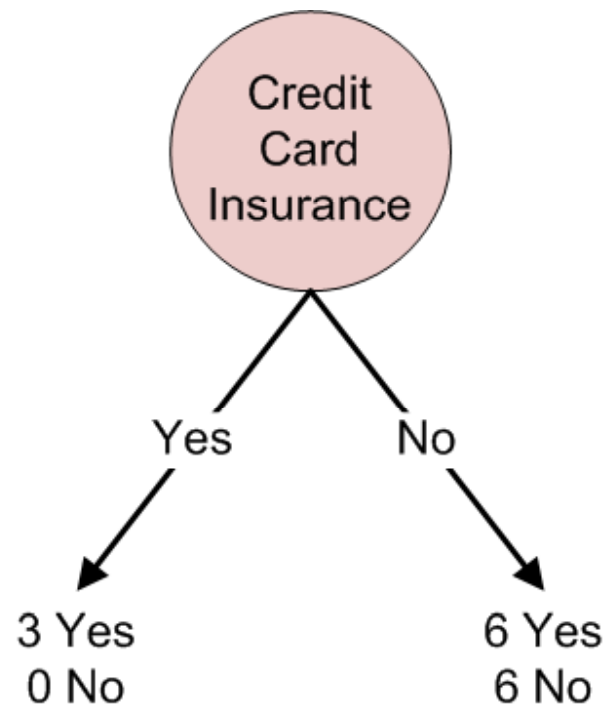
Apply Simplified C4.5

Consider a Different Top-Level Node



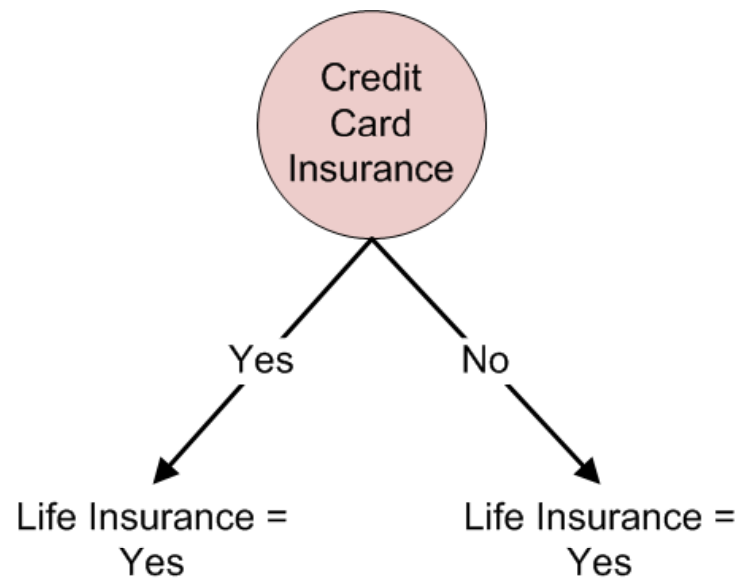
For each case (attribute value), how many instances of Life Insurance Promo = Yes and Life Insurance Promo = No?

Apply Simplified C4.5



For each branch, choose the most frequently occurring decision. If there is a tie, then choose Yes, since there are more total Yes instances (9) than No instances (6).

Apply Simplified C4.5

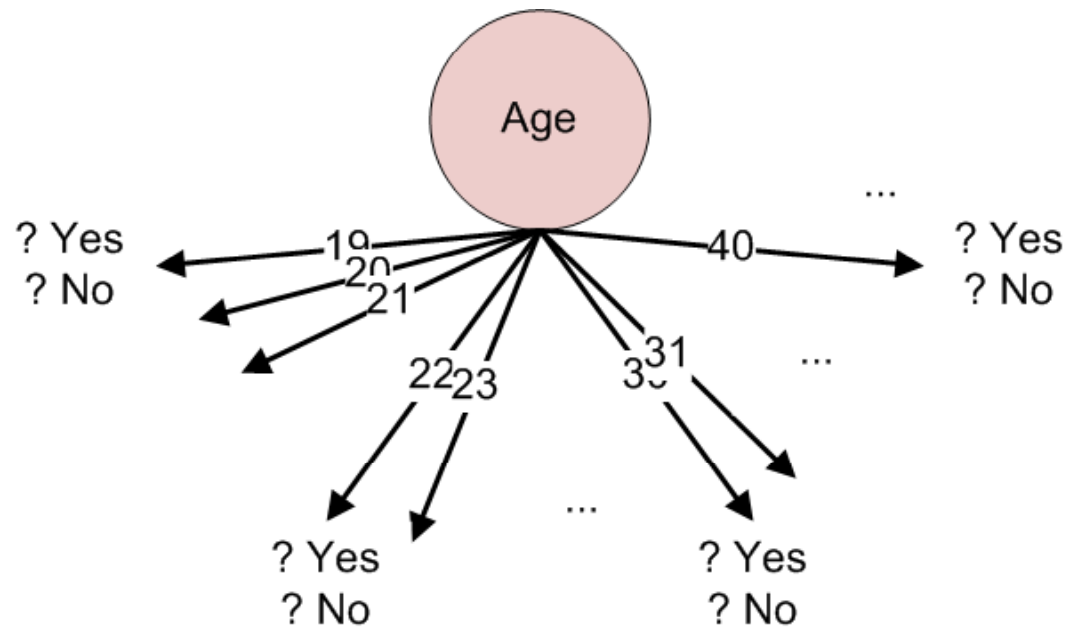


Evaluate the classification model (the tree). How many of the 15 training instances are classified correctly by this tree?

Apply Simplified C4.5

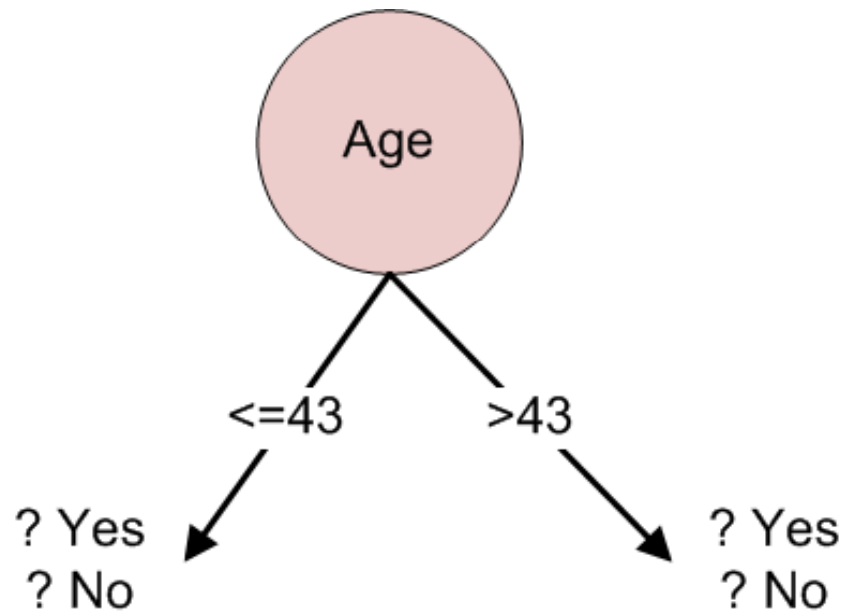
- Tree accuracy = $9/15 = 60.0\%$
- Tree cost = 2 branches for the computer program to use
- Goodness score for Income Range attribute is $9/15/2 = 0.300$
- Including Tree “cost” to assess goodness lets us compare trees

Apply Simplified C4.5



What's problematic about this?

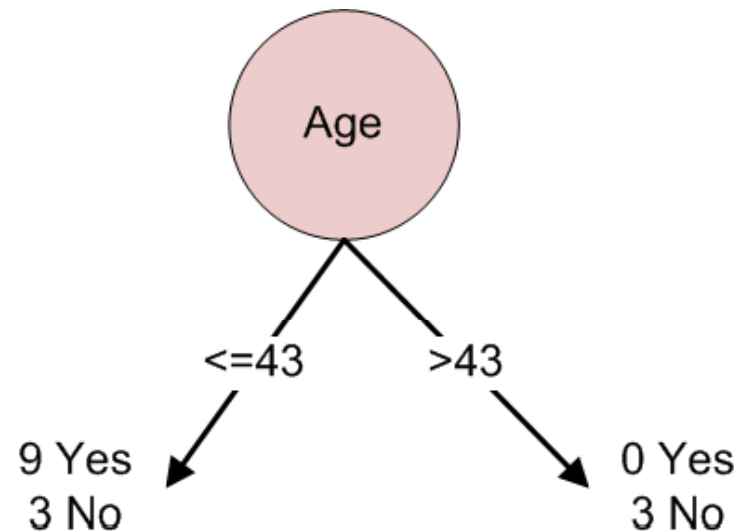
Apply Simplified C4.5



How many instances for each case?

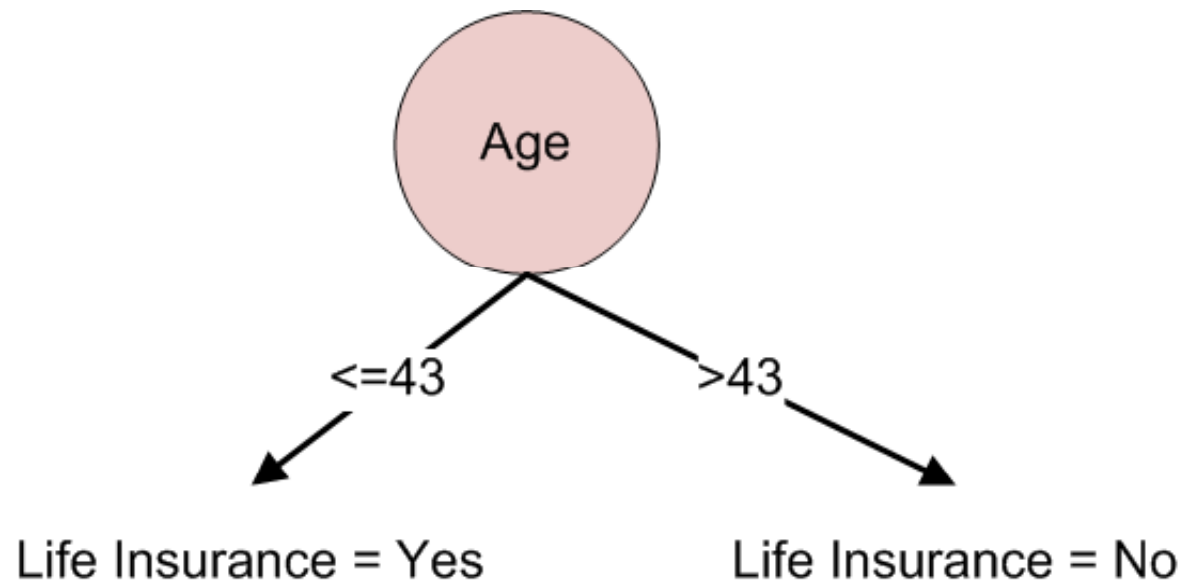
A binary split requires the addition of only two branches. Why 43?

Apply Simplified C4.5



For each branch, choose the most frequently occurring decision. If there is a tie, then choose Yes, since there are more total Yes instances (9) than No instances (6).

Apply Simplified C4.5

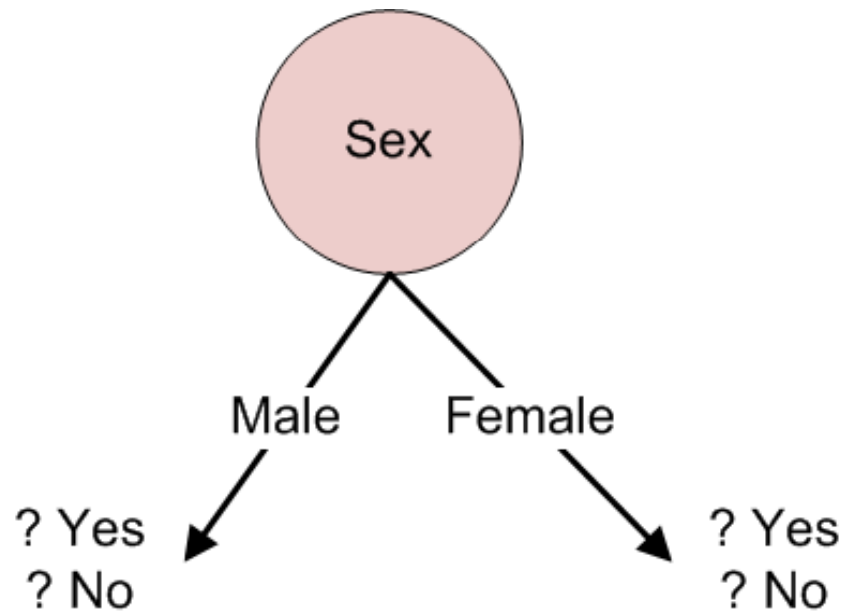


For this data, a binary split at 43 results in the best “score”.

Apply Simplified C4.5

- Tree accuracy = $12/15 = 80.0\%$
- Tree cost = 2 branches for the computer program to use
- Goodness score for Income Range attribute is $12/15/2 = 0.400$
- Including Tree “cost” to assess goodness lets us compare trees

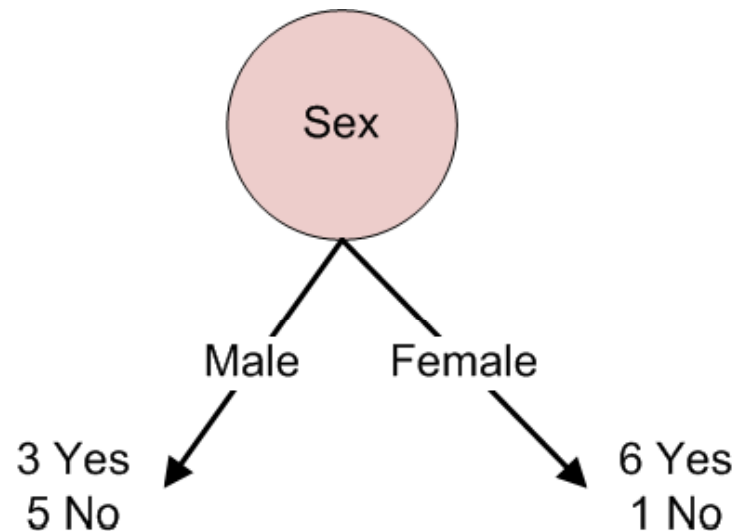
Apply Simplified C4.5



How many instances for each case?

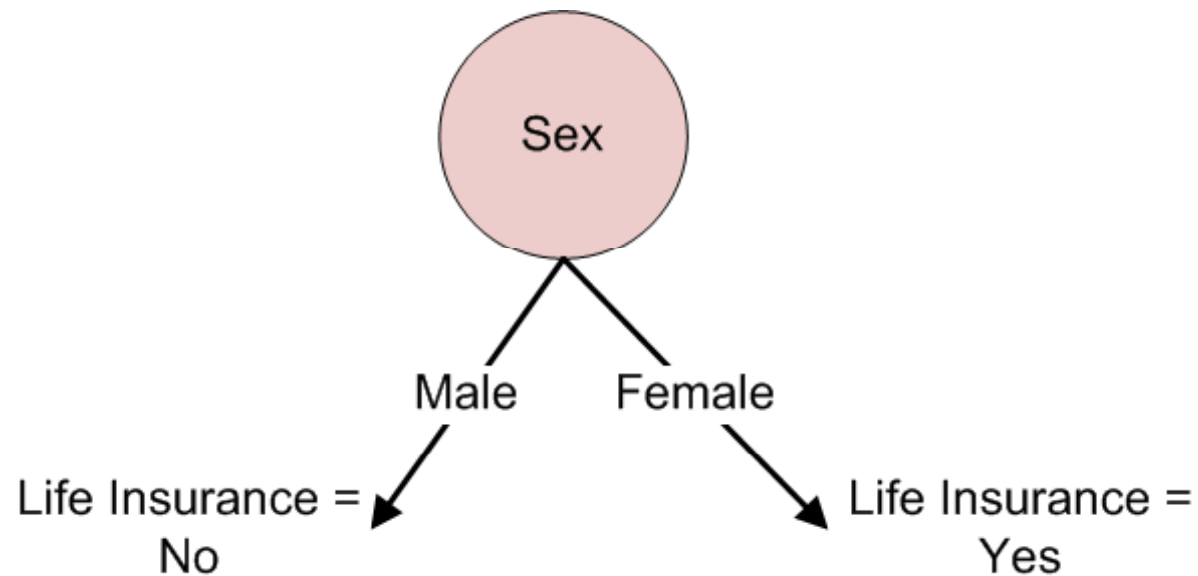
A binary split requires the addition of only two branches. Why 43?

Apply Simplified C4.5



For each branch, choose the most frequently occurring decision. If there is a tie, then choose Yes, since there are more total Yes instances (9) than No instances (6).

Apply Simplified C4.5



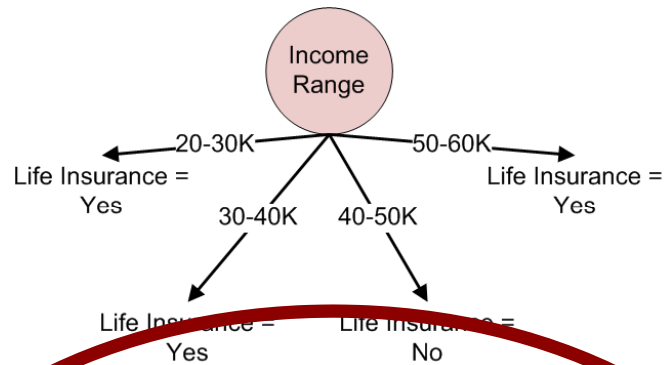
Evaluate the classification model (the tree). How many of the 15 training instances are classified correctly by this tree?

Apply Simplified C4.5

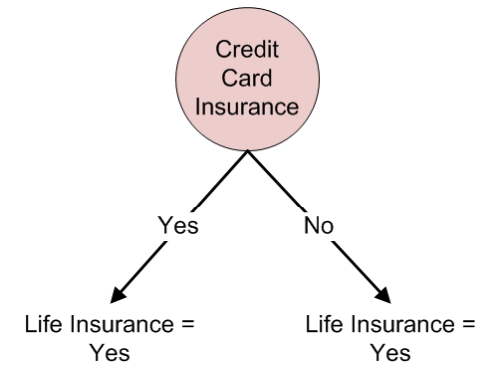
- Tree accuracy = $11/15 = 73.3\%$
- Tree cost = 2 branches for the computer program to use
- Goodness score for Income Range attribute is $11/15/2 = 0.367$
- Including Tree “cost” to assess goodness lets us compare trees

Apply Simplified C4.5

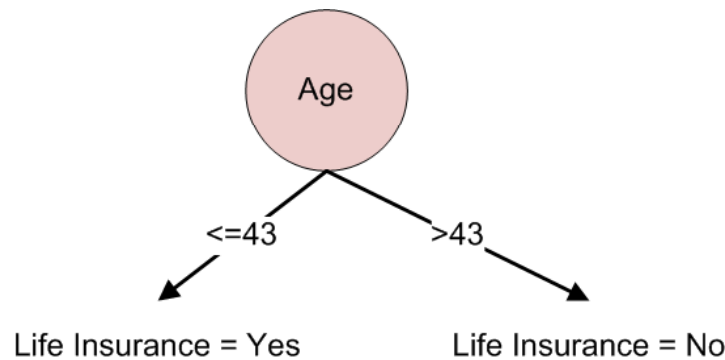
Model "goodness" = 0.183



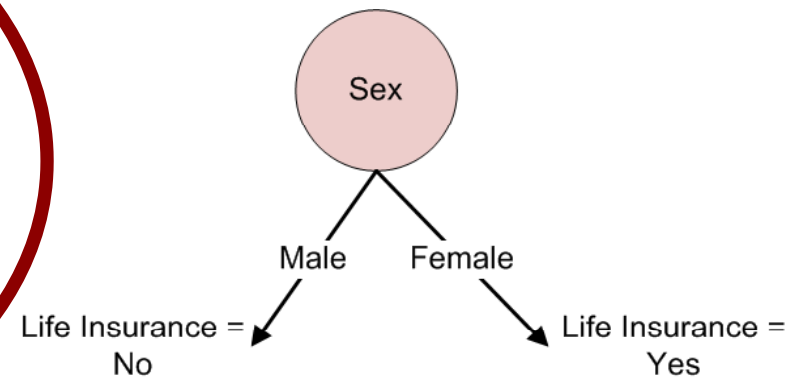
Model "goodness" = 0.30



Model "goodness" = 0.40



Model "goodness" = 0.367

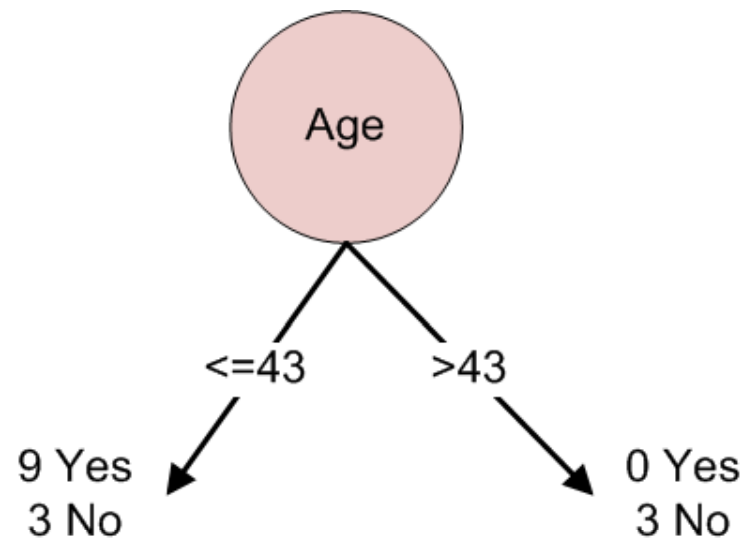


Apply Simplified C4.5

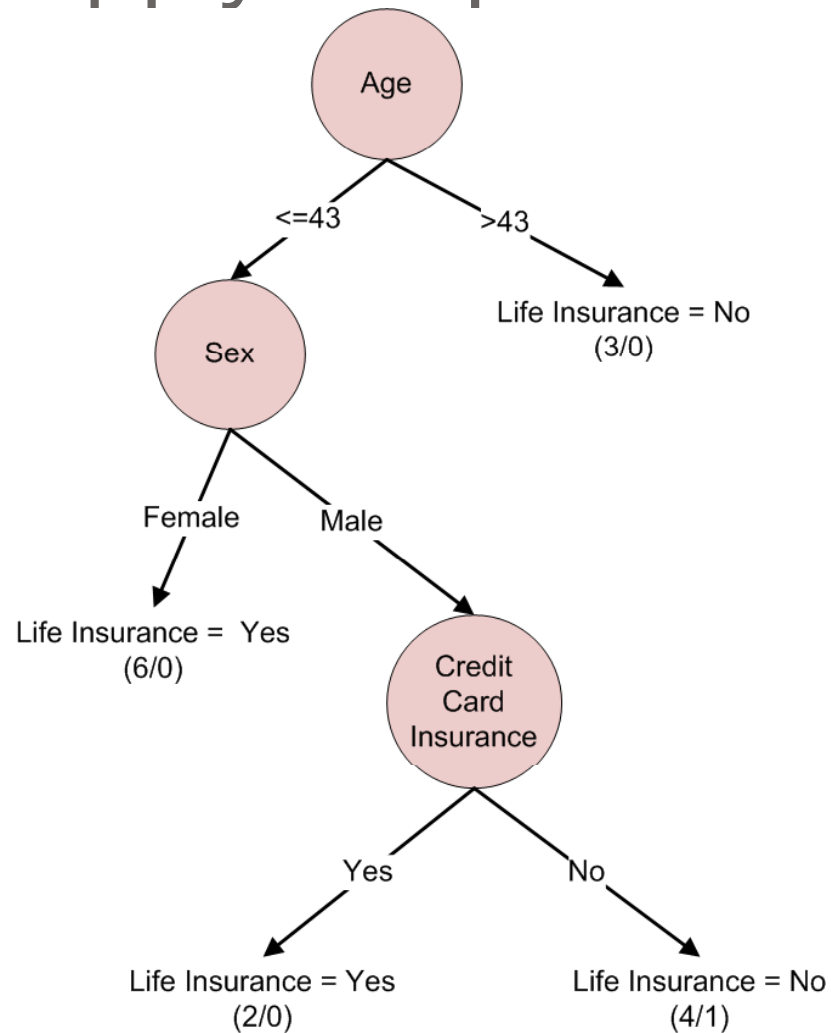
- Consider each branch and decide whether to terminate or add an attribute for further classification
- Different termination criteria make sense
 - If the instances following a branch satisfy a predetermined criterion, such as a certain level of accuracy, then the branch becomes a terminal path
 - No other attribute adds information

Apply Simplified C4.5

- 100% accuracy for >43 branch

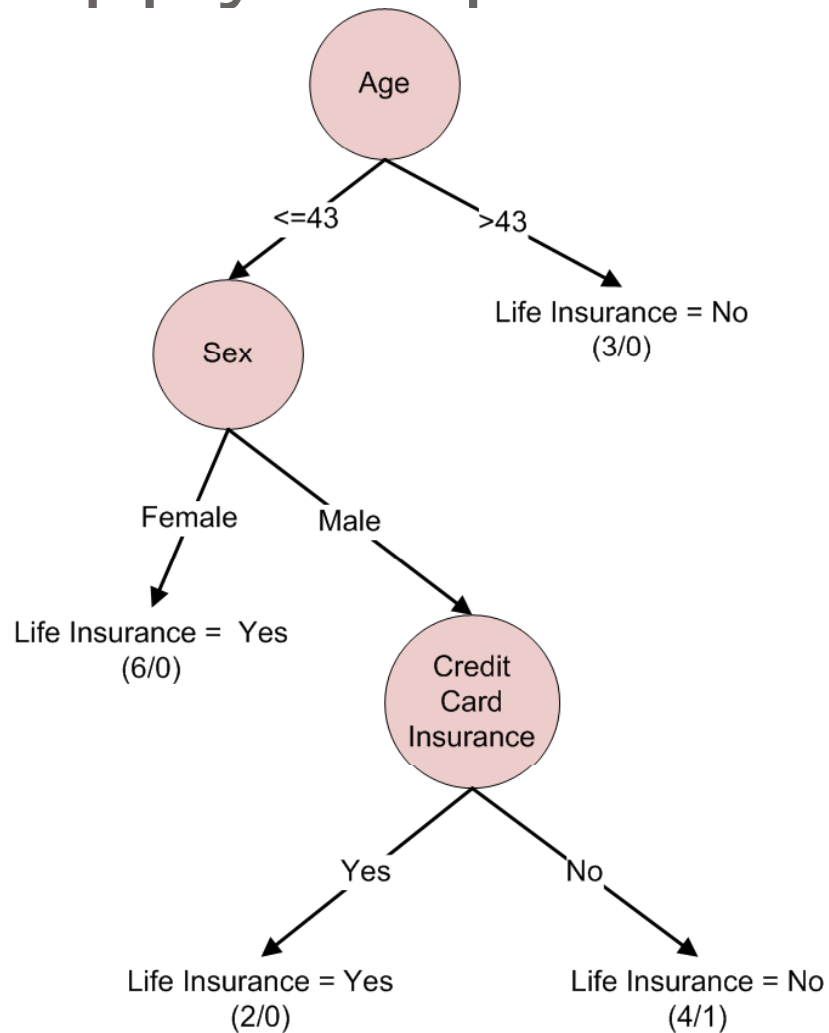


Apply Simplified C4.5



- Production rules are generated by following to each terminal branch

Apply Simplified C4.5



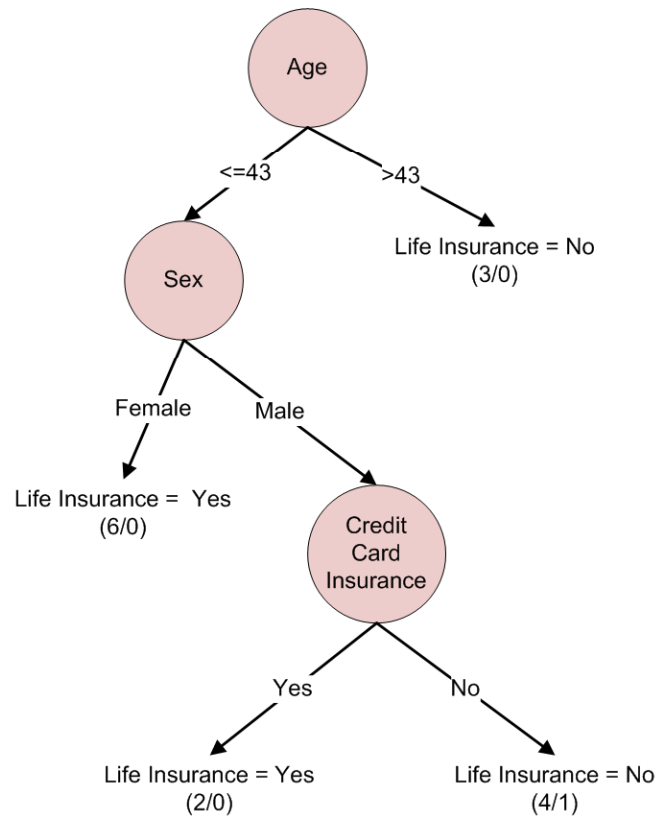
If Age ≤ 43 AND Sex =
Male AND CCI = No

Then Life Insurance
Promo = No

Accuracy = 75%

Coverage = 26.7%

Apply Simplified C4.5



Simplify the Rule

If Sex = Male AND CCIIns = No

Then Life Insurance Promo = No

Accuracy = 83.3%

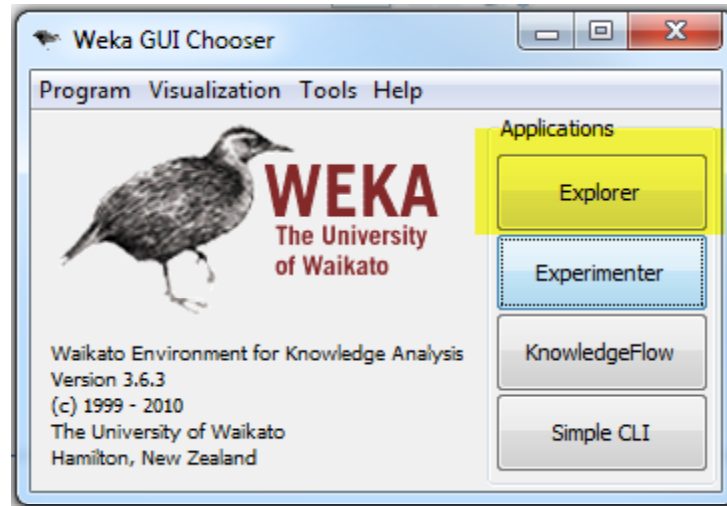
Coverage = 40.0%

This rule is more general, more accurate

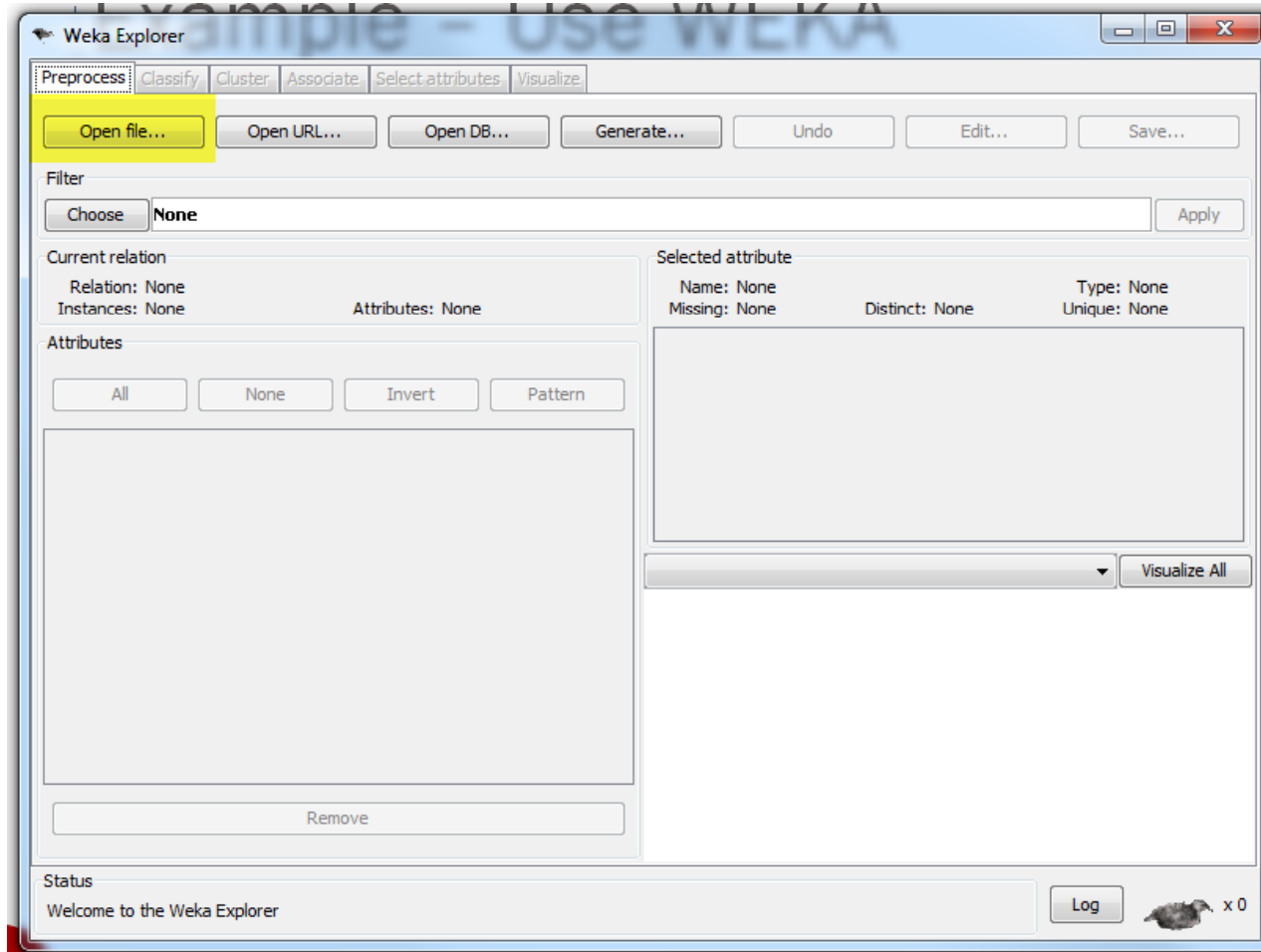
Decision Tree Algorithm Implementations

- Automate the process of rule creation
- Automate the process of rule simplification
- Choose a default rule – the one that states the classification of an instance that does not meet the preconditions of any listed rule

Example – Use WEKA

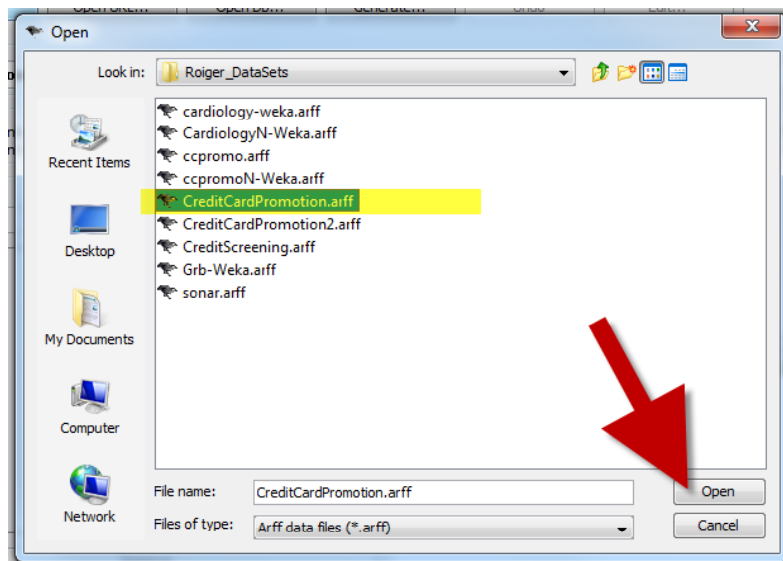


Example – Use WEKA



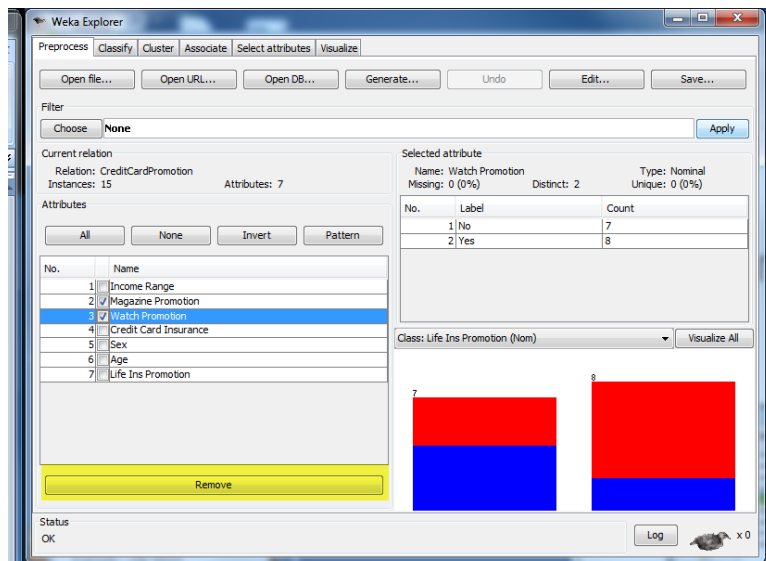
Example – Use WEKA

- Download CreditCardPromotion.zip from Blackboard and extract CreditCardPromotion.arff



Example – Use WEKA

- Why remove magazine promotion and watch promotion from the analysis?



Example – Use WEKA

The screenshot shows the Weka Explorer application window. The 'Classify' tab is active. The 'Current relation' is 'CreditCardPromotion-weka.filters.unsupervised.attribute.R...' with 15 instances and 5 attributes. The 'Attributes' list includes 'Income Range', 'Credit Card Insurance', 'Sex', 'Age', and 'Life Ins Promotion'. The 'Income Range' attribute is selected, showing a distribution of 4 instances for '40-50K', 5 for '30-40K', 2 for '50-60K', and 4 for '20-30K'. A bar chart visualizes the distribution of the 'Life Ins Promotion' class (Nominal) across these four income categories. The chart shows four bars with red and blue segments. The counts for the red segments are 4, 5, 2, and 4, corresponding to the four income categories.

No.	Label	Count
1	40-50K	4
2	30-40K	5
3	50-60K	2
4	20-30K	4

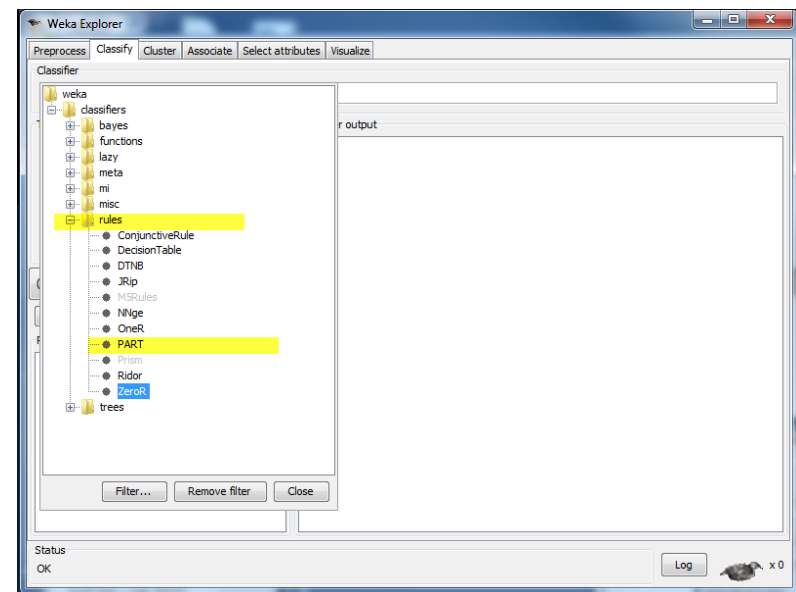
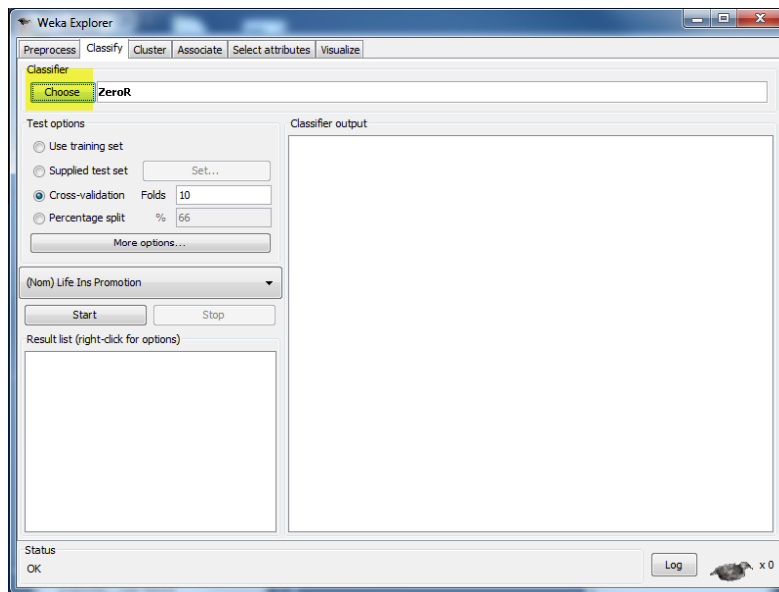
Class: Life Ins Promotion (Nom)

Status: OK

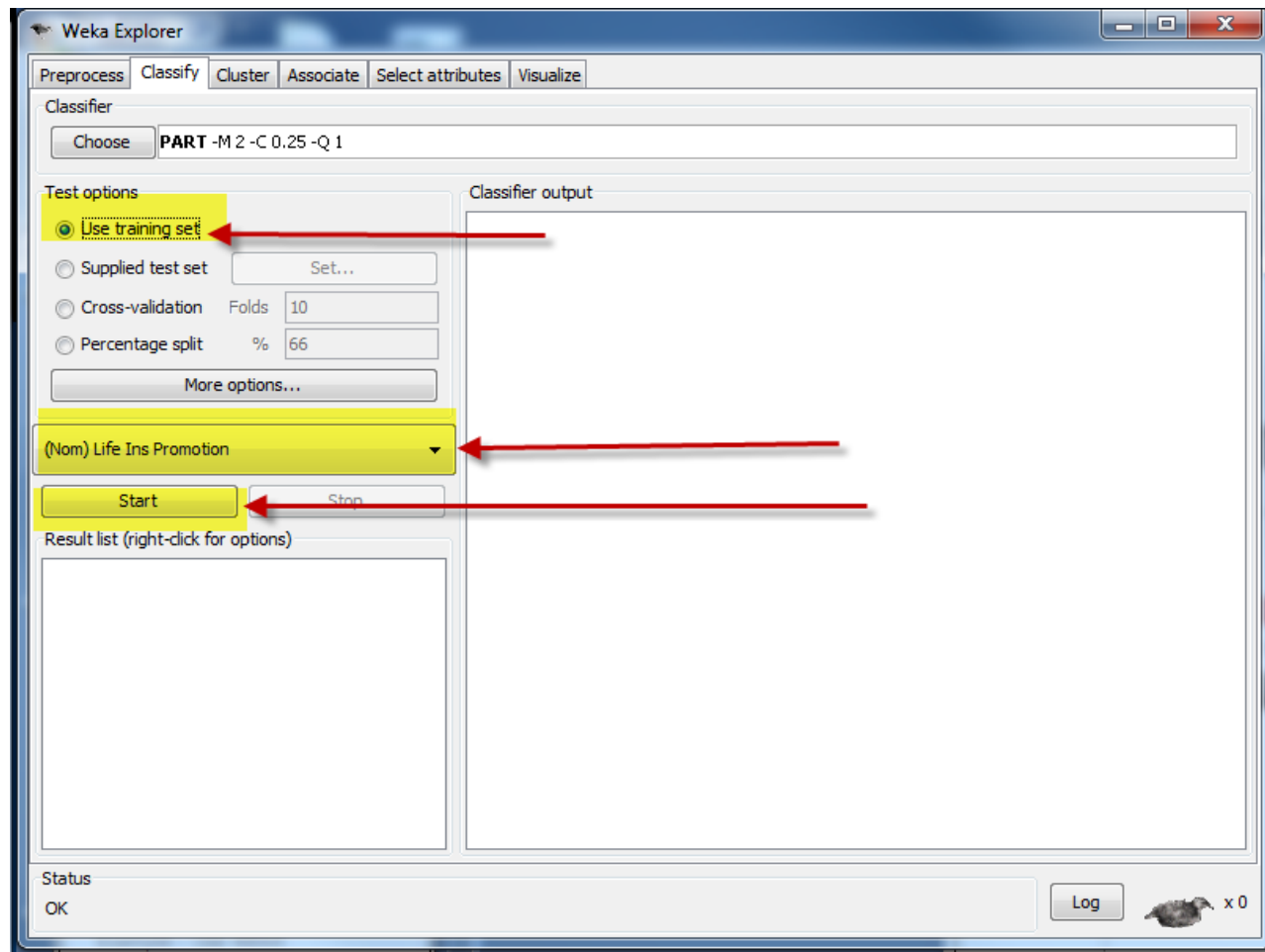
Example – Use WEKA

See algorithm options
through Choose

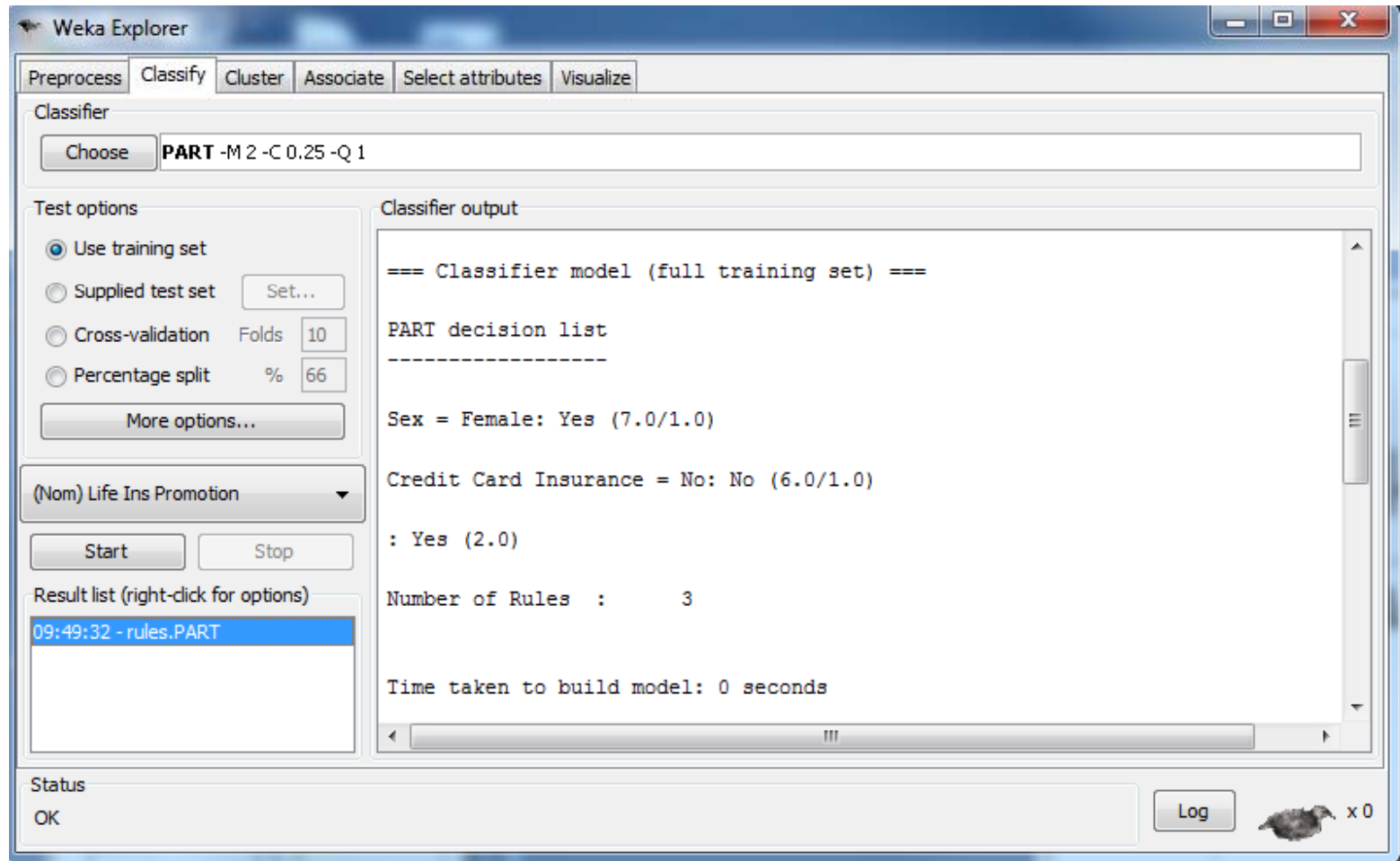
Choose PART under rules



Example – Use WEKA

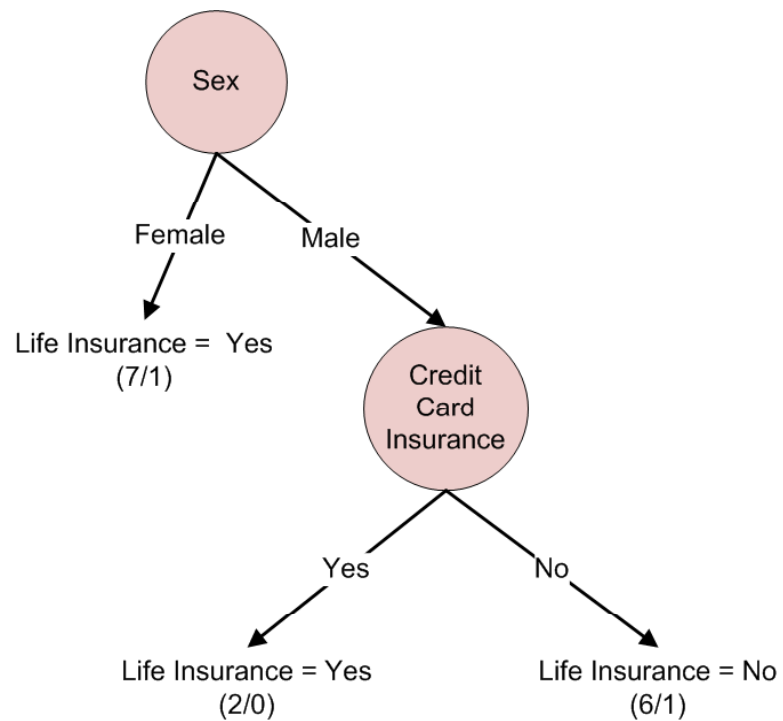


Example – Use WEKA



Example – Use WEKA

- Decision tree equivalent of rules generated by PART



Example – Use WEKA

The screenshot shows the Weka Explorer application window. The 'Classifier' tab is active, and the 'PART -M 2 -C 0.25 -Q 1' classifier is selected. The 'Test options' section shows 'Use training set' selected. The 'Classifier output' pane displays the following results:

```
Time taken to build model: 0 seconds
=== Evaluation on training set ===
=== Summary ===
Correctly Classified Instances      13          86.6667 %
Incorrectly Classified Instances    2           13.3333 %
Kappa statistic                    0.7222
Mean absolute error                 0.2254
Root mean squared error             0.3357
Relative absolute error             46.7286 %
Root relative squared error         68.5059 %
Total Number of Instances          15

=== Detailed Accuracy By Class ===
               TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area
0.833         0.111    0.833      0.833   0.833      0.88
0.889         0.167    0.889      0.889   0.889      0.88
Weighted Avg.  0.867         0.144    0.867      0.867   0.867      0.88

=== Confusion Matrix ===
a b <-- classified as
5 1 | a = No
1 8 | b = Yes
```

The 'Result list' shows a single entry: '09:49:32 -rules.PART'. The status bar at the bottom indicates 'OK'.

Decision Trees – Advantages

Pluses

- Easy to understand
- Map readily to production rules
- No prior assumptions about the nature of the data needed
 - e.g., no assumption of normally distributed data needed
- Apply to categorical data, but numerical data can be binned for application

Issues

- Output attribute must be categorical
- Only one output attribute
- Sufficiently robust?
 - Change in one training set data item can change outcome
- Numerical attributes can create complex decision trees (due to split algorithms)

Decision Trees

By Susan Miertschin