UCIrvine | The Paul Merage School of Business

2022.11.24

BANA 273 Session 6

Classification using Decision Trees

Prof. Vibs Abhishek

The Paul Merage School of Business

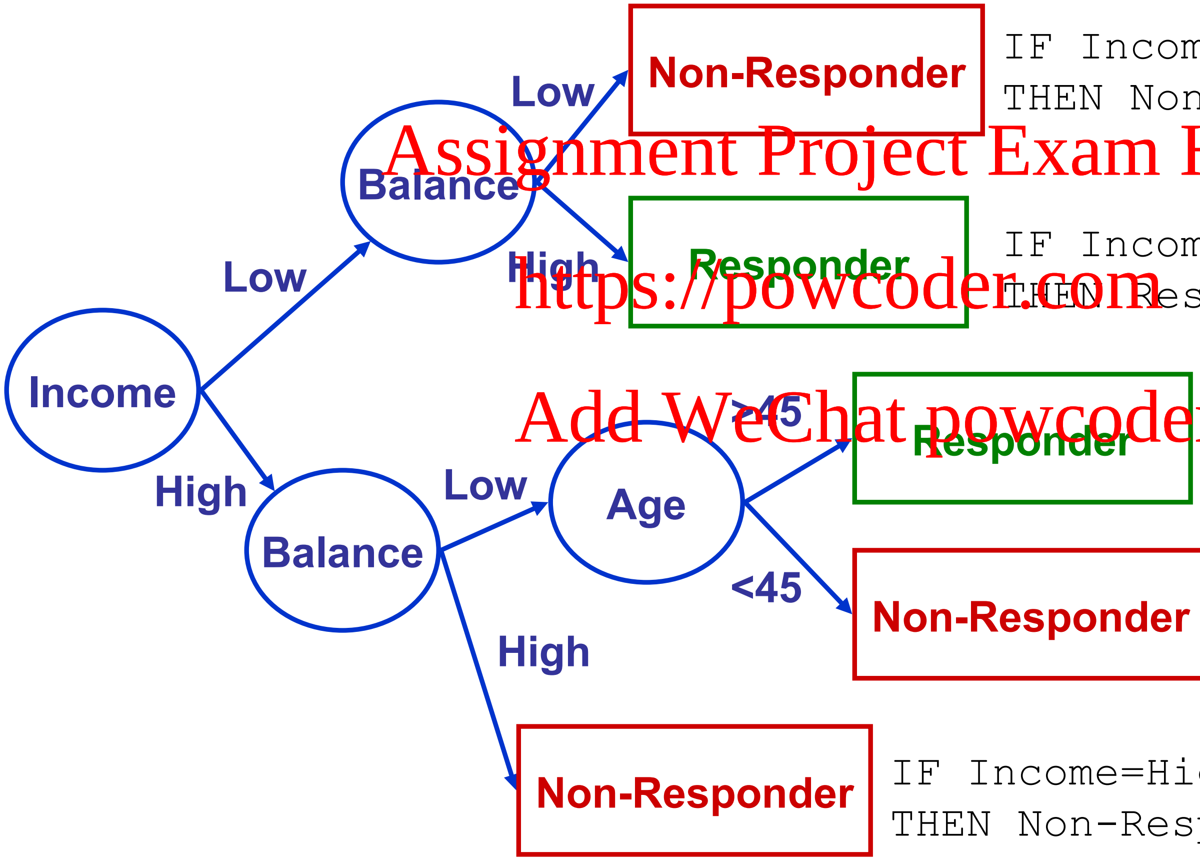
University of California, Irvine

Agenda

- Using Decision Tree for Classification
- Building Decision Trees
- Review Assignment 2

Reading Rules off the Decision Tree

For each leaf in the tree, read the rule from the root to that leaf.
You will arrive at a set of rules.



IF Income=Low AND Balance=Low
THEN Non-Responder

IF Income=Low AND Balance=High
THEN Responder

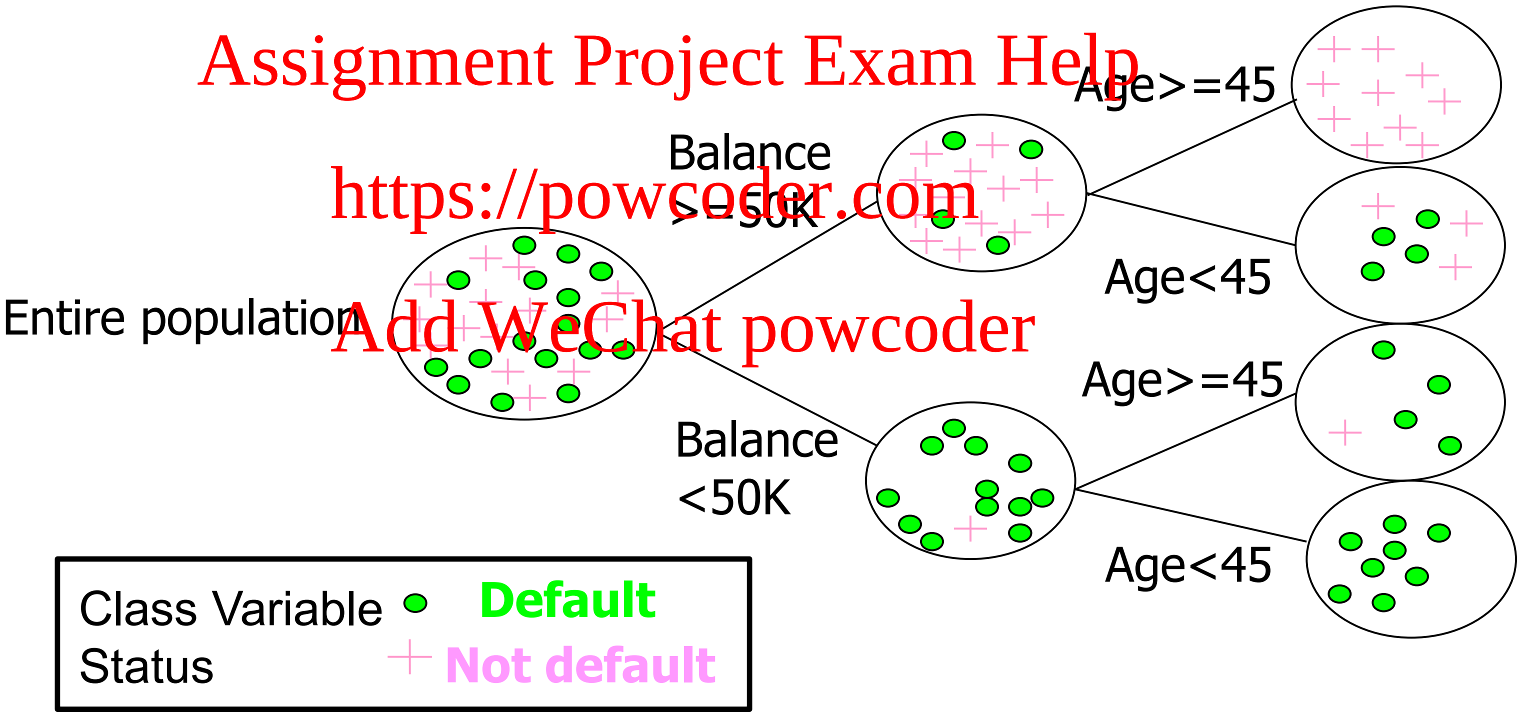
IF Income=High AND
Balance=Low AND
Age>45
THEN Responder

IF Income=High AND
Balance=Low AND
Age<45
THEN Non-Responder

IF Income=High AND Balance=High
THEN Non-Responder

Goal of Decision Tree Construction

- Partition the training instances into <u>purer</u> sub groups
  - pure: the instances in a sub-group mostly belong to the same class



- How to build a tree: How to split instances into purer sub-groups

Purity Measures

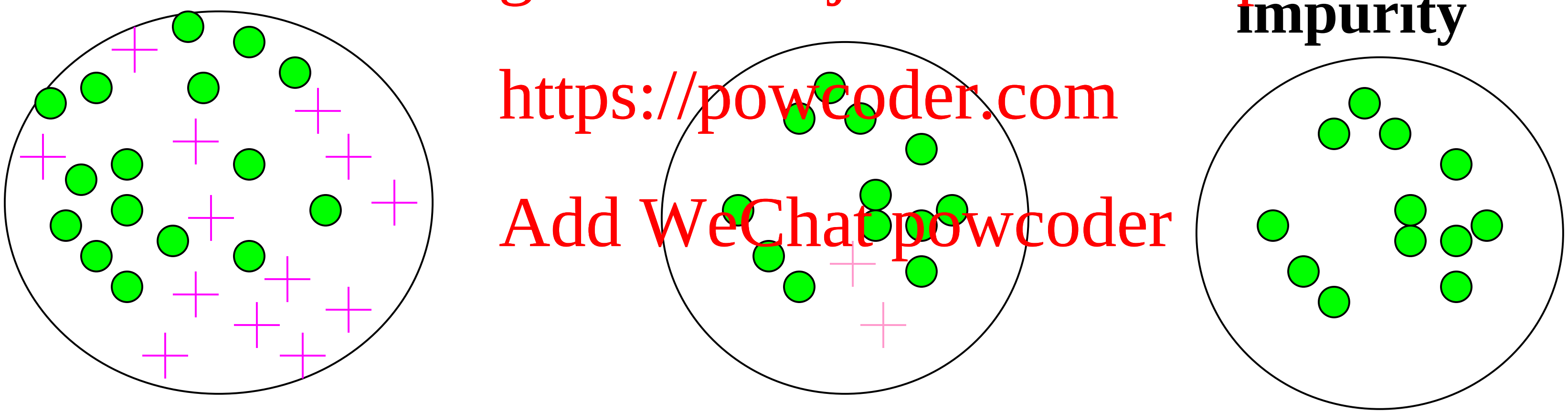
- Purity measures: Many available
  - Gini (population diversity)
  - Entropy (information gain)
  - Information Gain
  - Chi-square Test
- Most common one (from information theory) is: *Information Gain*

Why do we want to identify pure sub groups?

- To classify a new instance, we can determine the leaf that the instance belongs to based on its attributes.
- If the leaf is very pure (e.g. all have defaulted) we can determine with greater confidence that the new instance belongs to this class (i.e., the “Default” class.)
- If the leaf is not very pure (e.g. a 50%/50% mixture of the two classes, Default and Not Default), our prediction for the new instance is more like a random guess.

Impurity

Very impure group **Less impure** **Minimum impurity**



The figures above show distribution of the class variable

Class Variable Status — Default; Not default

Example Split

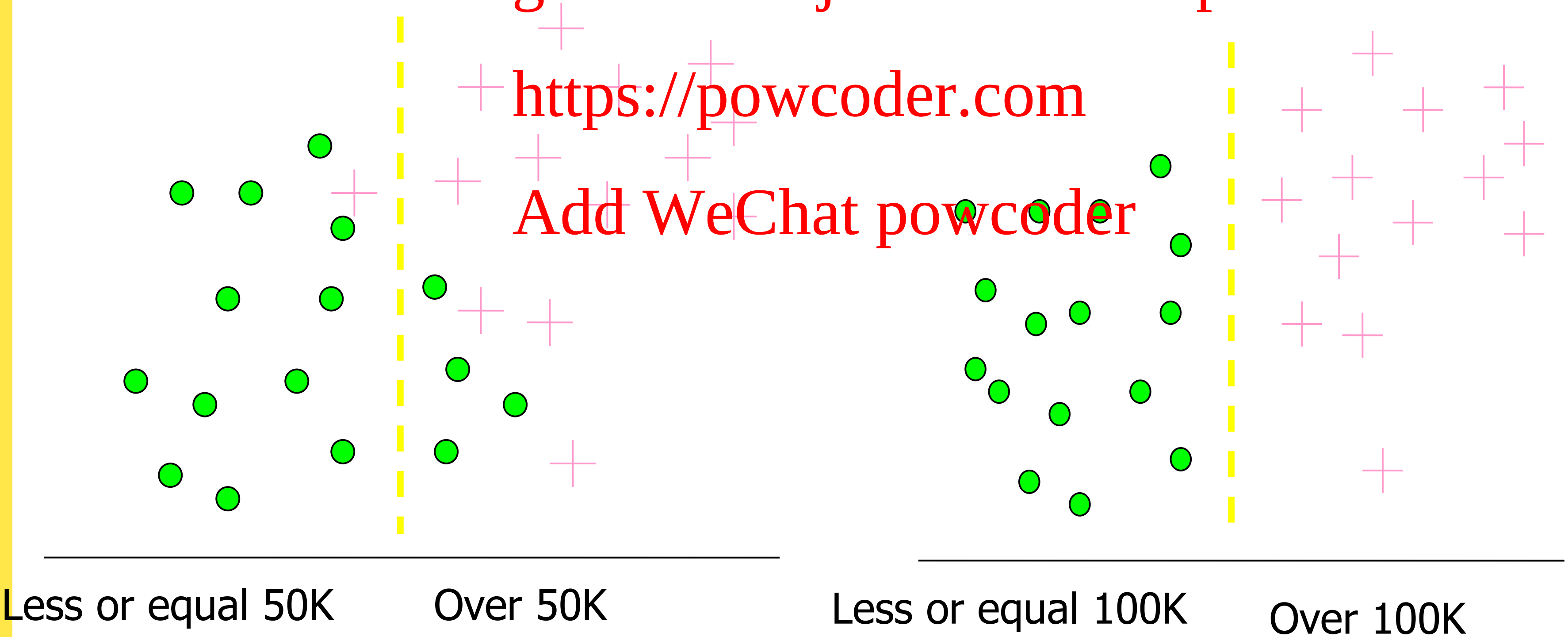
Consider the two following splits.

Which one is more informative?

Class Variable Status: Default / Not default

Split over whether Balance exceeds 50K

Split over whether Income exceeds 100K



Decision Tree Construction

- A tree is constructed by recursively partitioning the examples.

- With each partition the examples are split into increasingly purer sub groups.

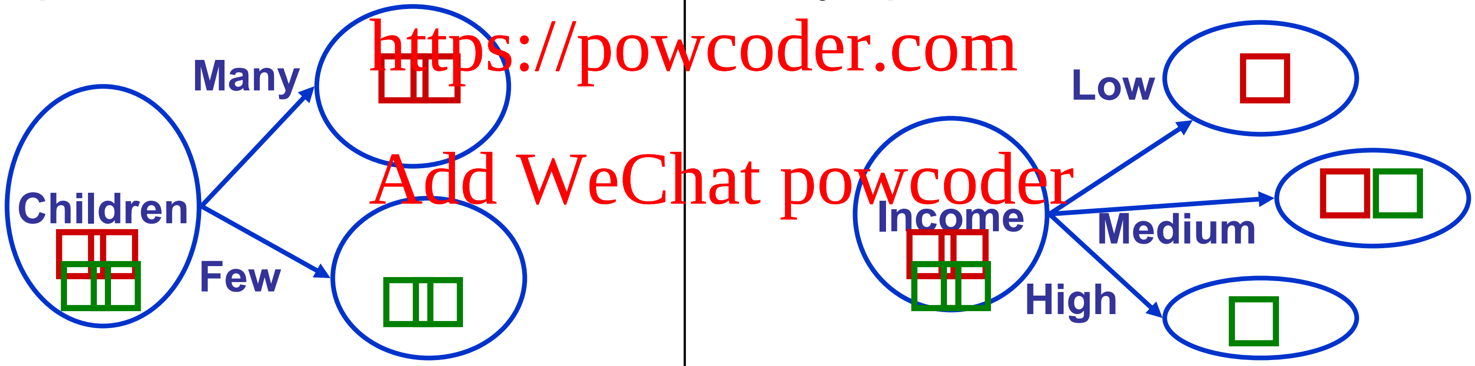
- The key in building a tree: How to split

Choosing a Split

ApplicantID	City	Children	Income	Status
1	Philly	Many	Medium	DEFAULTS
2	Philly	Many	Low	DEFAULTS
3	Philly	Few	Medium	PAYS
4	Philly	Few	High	PAYS

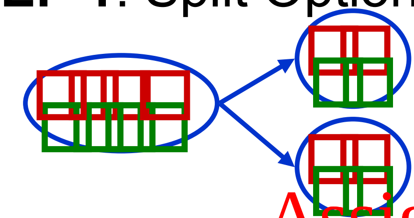
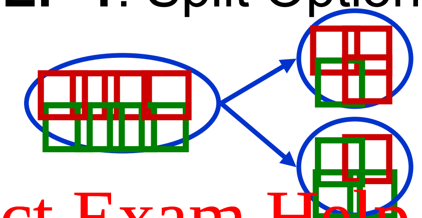
Try split on Children attribute:

Try split on Income attribute:



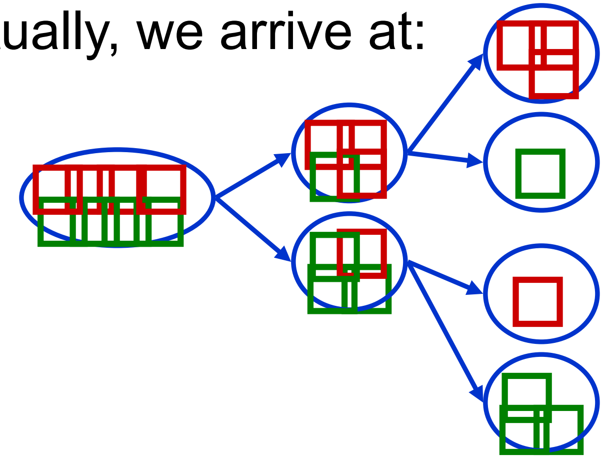
Notice how the split on the Children attribute gives purer partitions. It is therefore chosen as the first split (and in this case the only split – because the two sub-groups are 100% pure).

Recursive Steps in Building a Tree Example

STEP 1: Split Option A  Not good as sub-nodes are still very heterogenous!	**STEP 1**: Split Option B  Better, as purity of sub-nodes is improving.

STEP 2: Choose Split Option B as it is the better split.

STEP 3: Try out splits on each of the sub-nodes of Split Option B. Eventually, we arrive at:



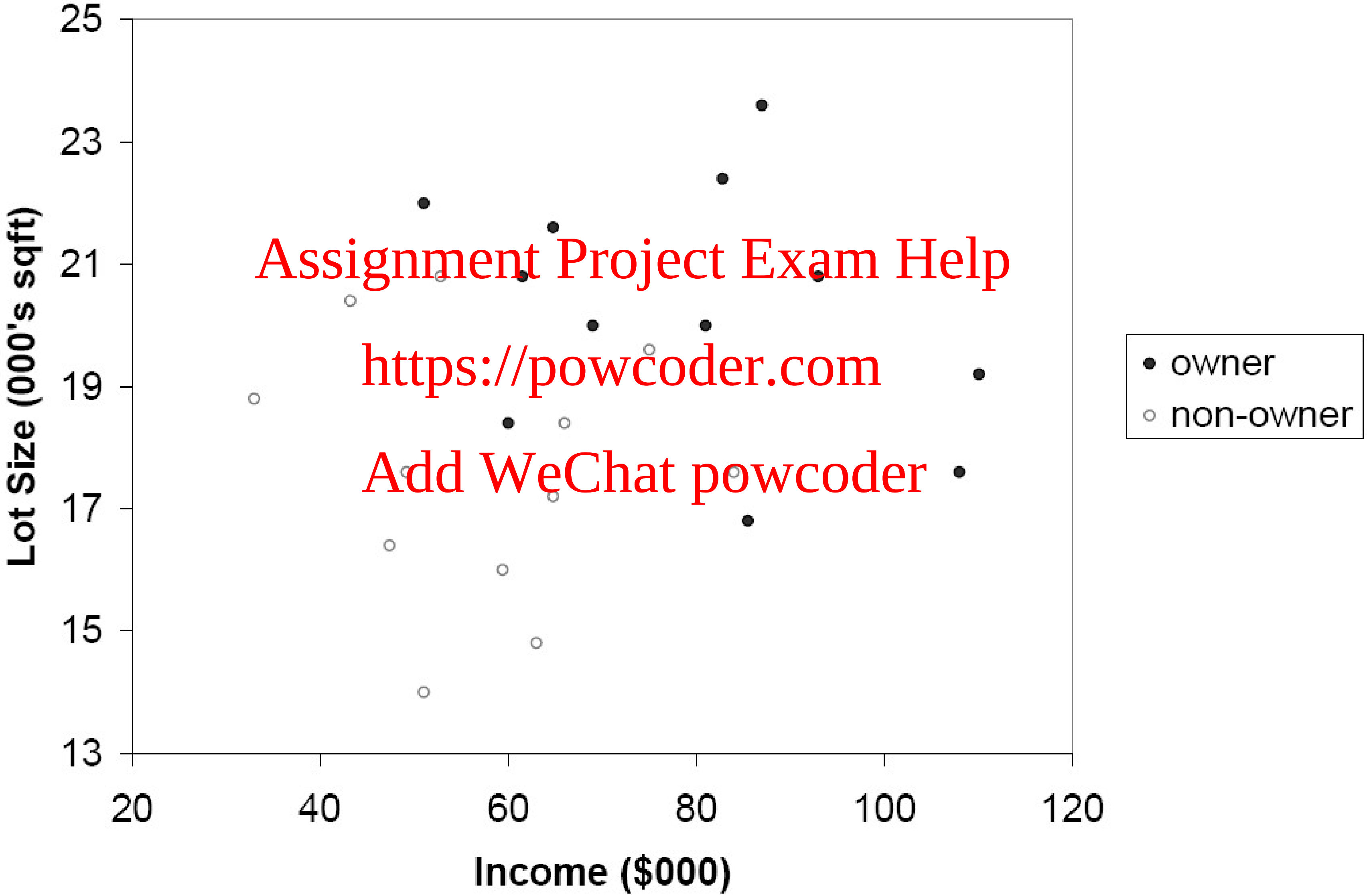
Notice how examples in a parent node are split between sub-nodes - i.e. notice how the training examples are partitioned into smaller and smaller subsets. Also, notice that sub-nodes are purer than parent nodes.

Example 1: Riding Mower

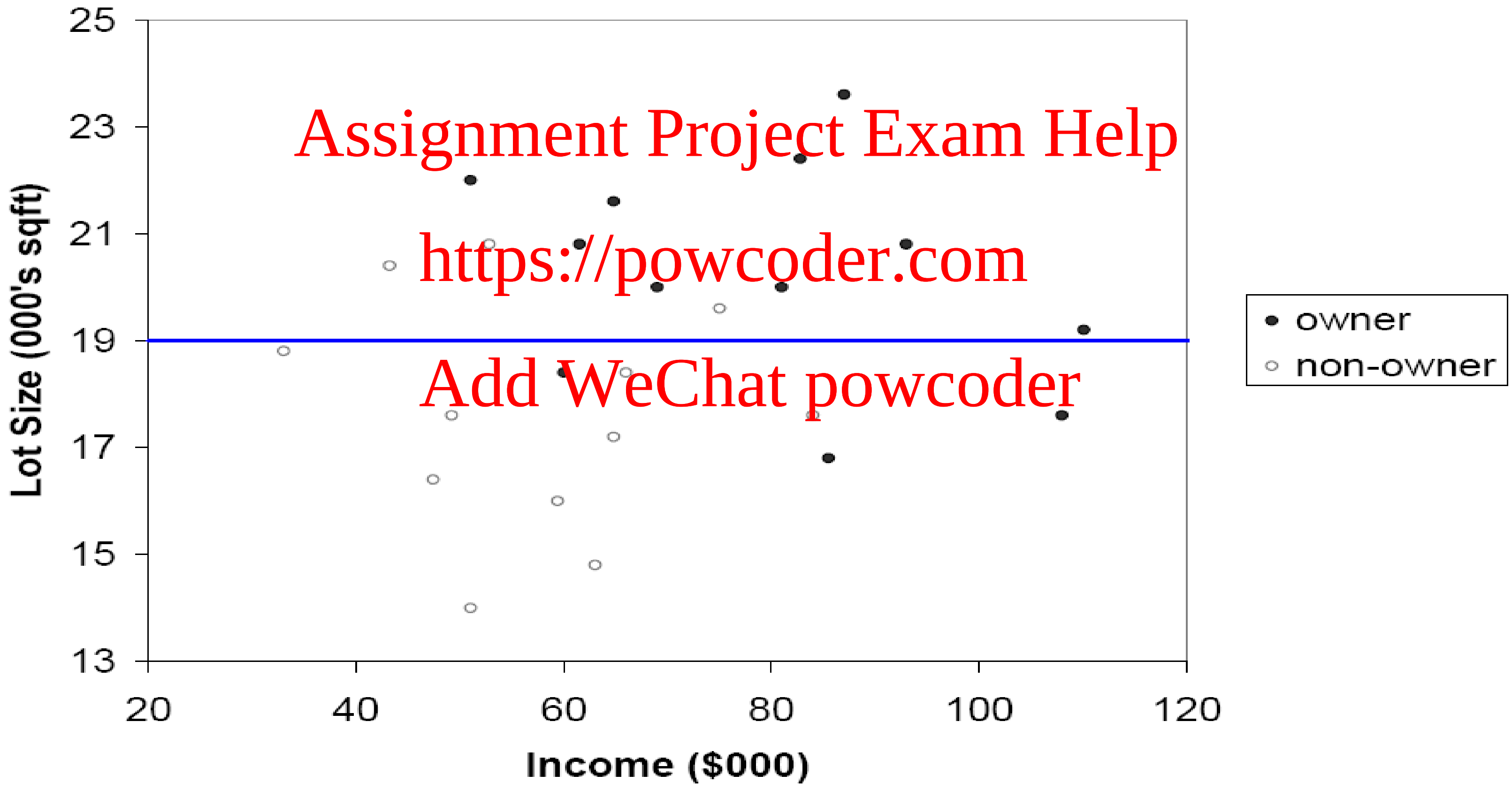
Lot Size, Income, and Ownership of a Riding Mower for 24 Households

Household number	Income ($ 000's)	Lot Size (000's ft^2)	Ownership of, riding mower
1	60	18.4	Owner
2	85.5	16.8	Owner
3	64.8	21.6	Owner
4	61.5	20.8	Owner
5	87	23.6	Owner
6	110.1	19.2	Owner
7	108	17.6	Owner
8	82.8	22.4	Owner
9	69	20	Owner
10	93	20.8	Owner
11	51	22	Owner
12	81	20	Owner
13	75	19.6	Non-Owner
14	52.8	20.8	Non-Owner
15	64.8	17.2	Non-Owner
16	43.2	20.4	Non-Owner
17	84	17.6	Non-Owner
18	49.2	17.6	Non-Owner
19	59.4	16	Non-Owner
20	66	18.4	Non-Owner
21	47.4	16.4	Non-Owner
22	33	18.8	Non-Owner
23	51	14	Non-Owner
24	63	14.8	Non-Owner

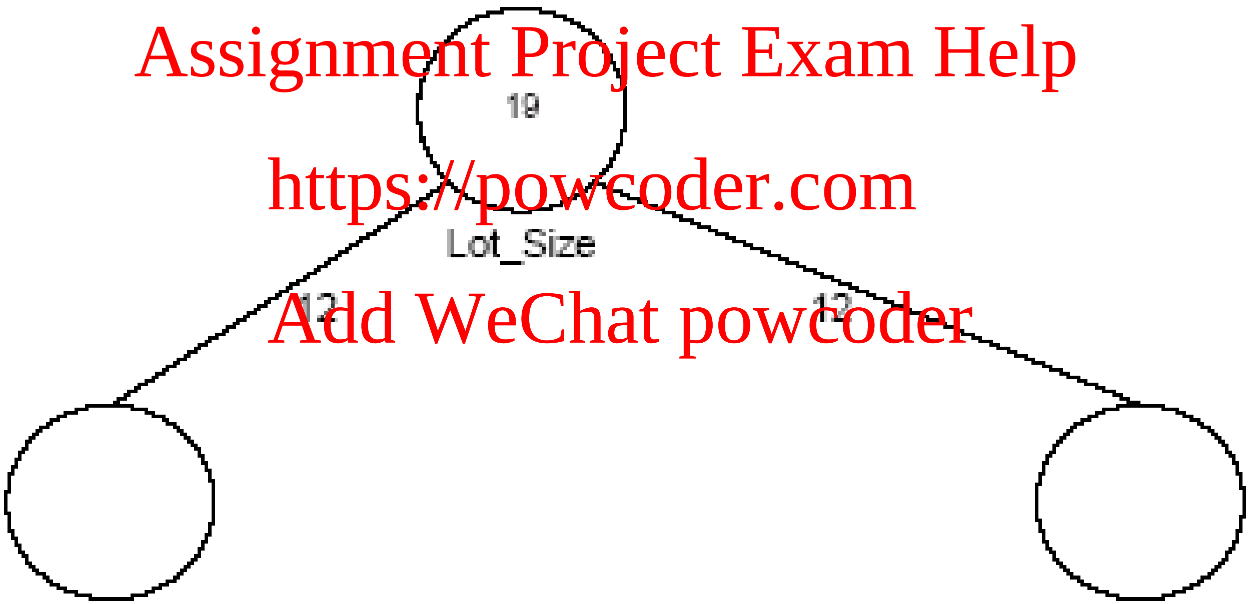
Scatterplot of Lot Size versus Income



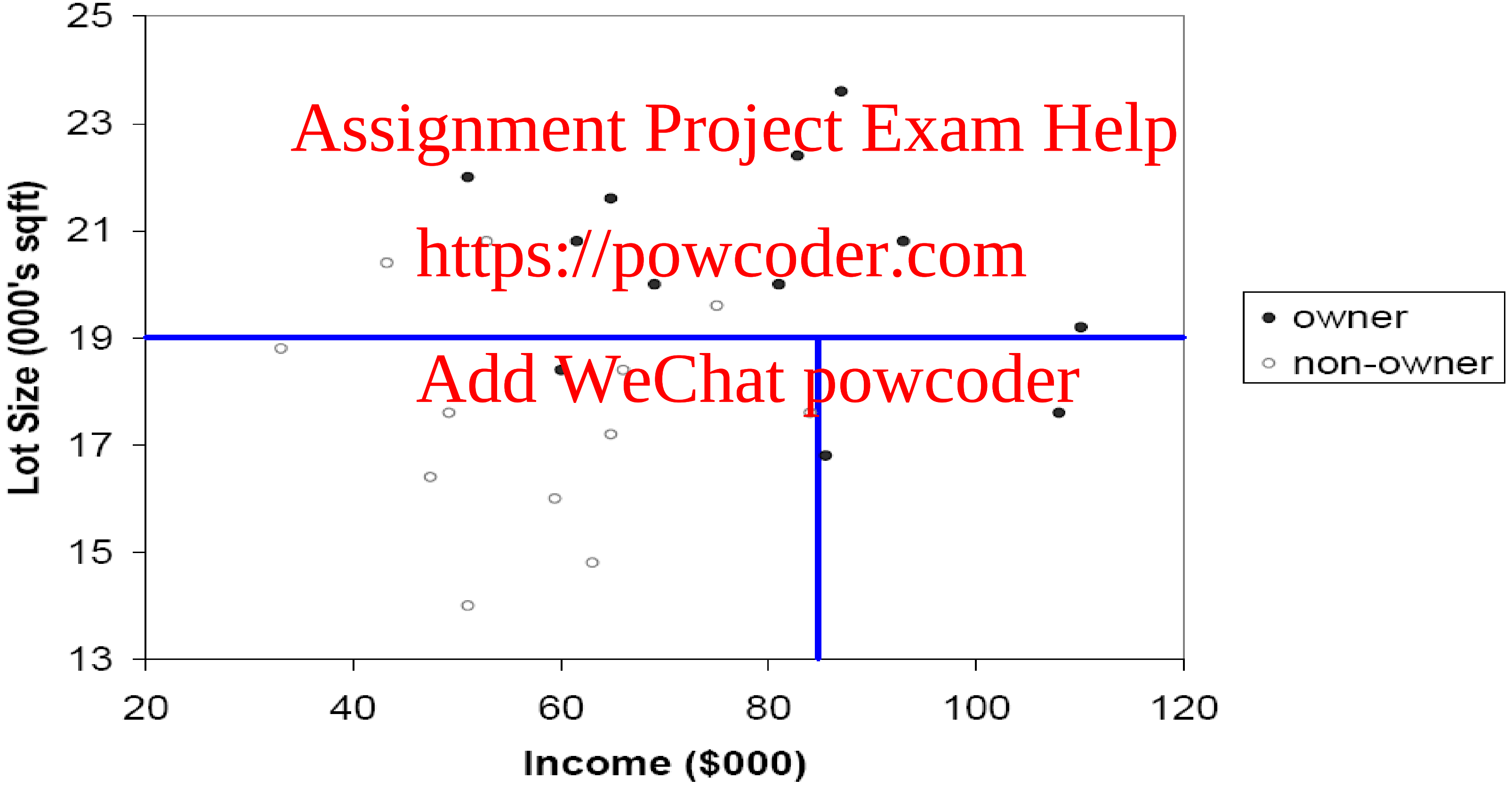
Splitting the Observations by Lot Size Value of 19



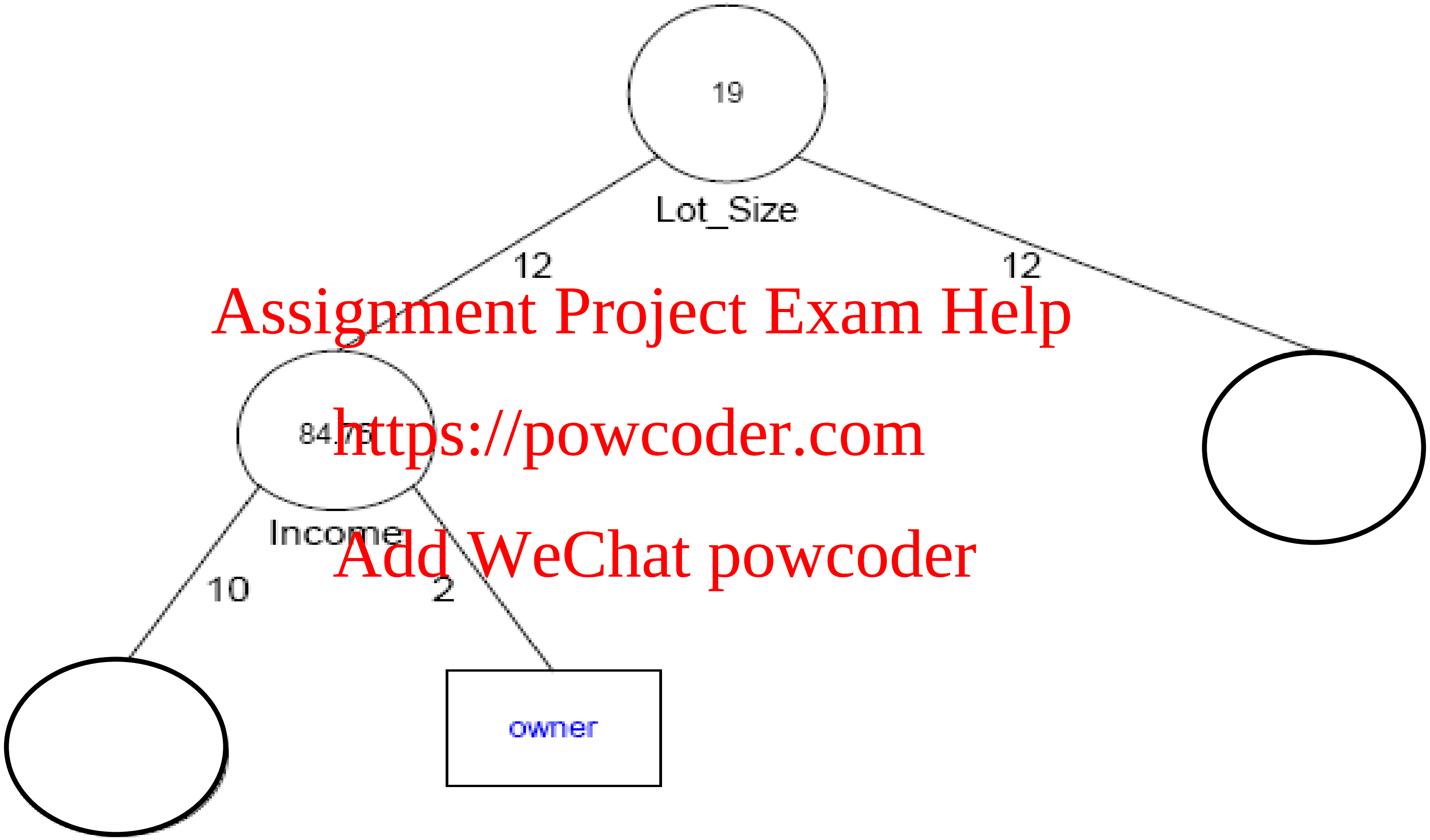
Tree Diagram: First Split

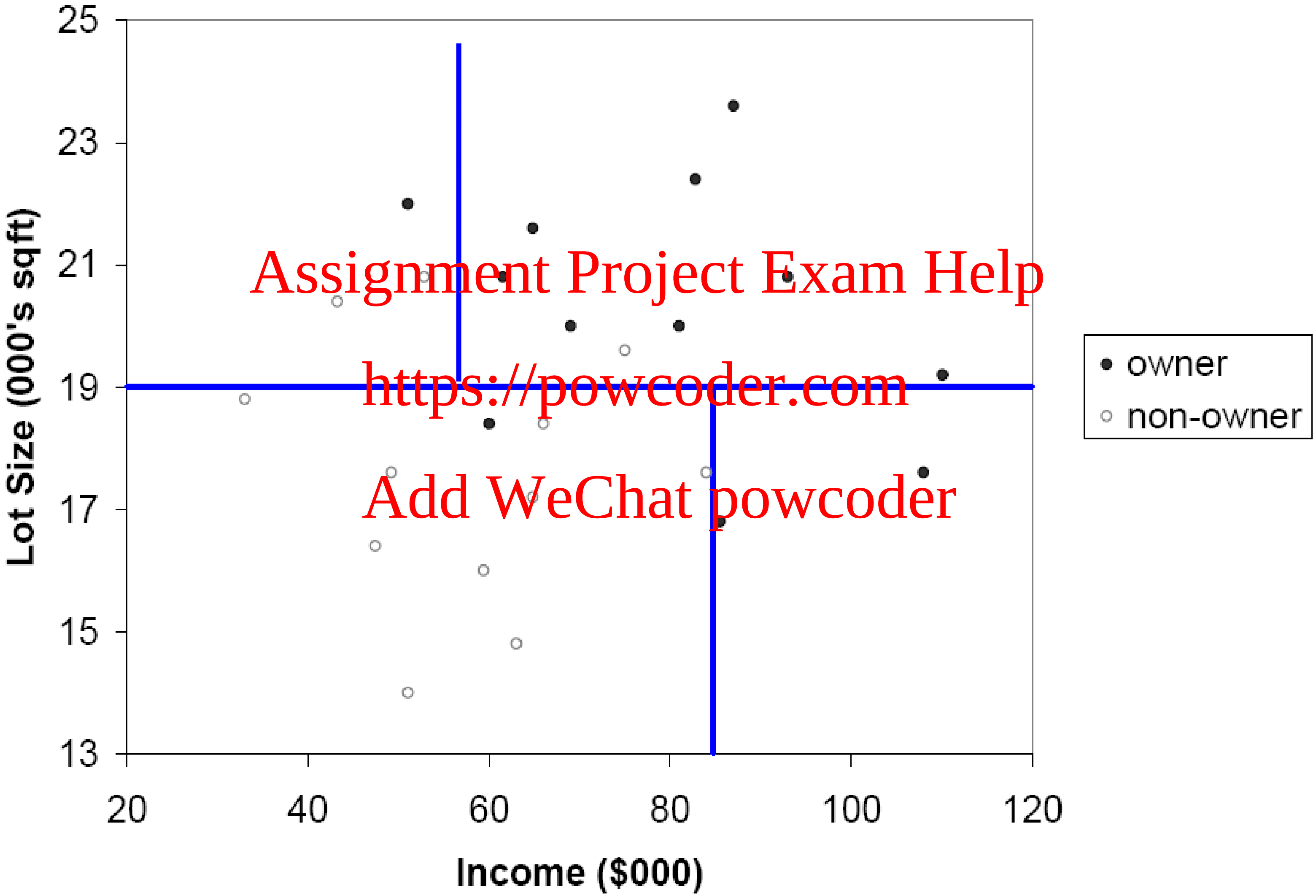


Second Split: Lot Size Value of 19K and then Income Value of 84.75K

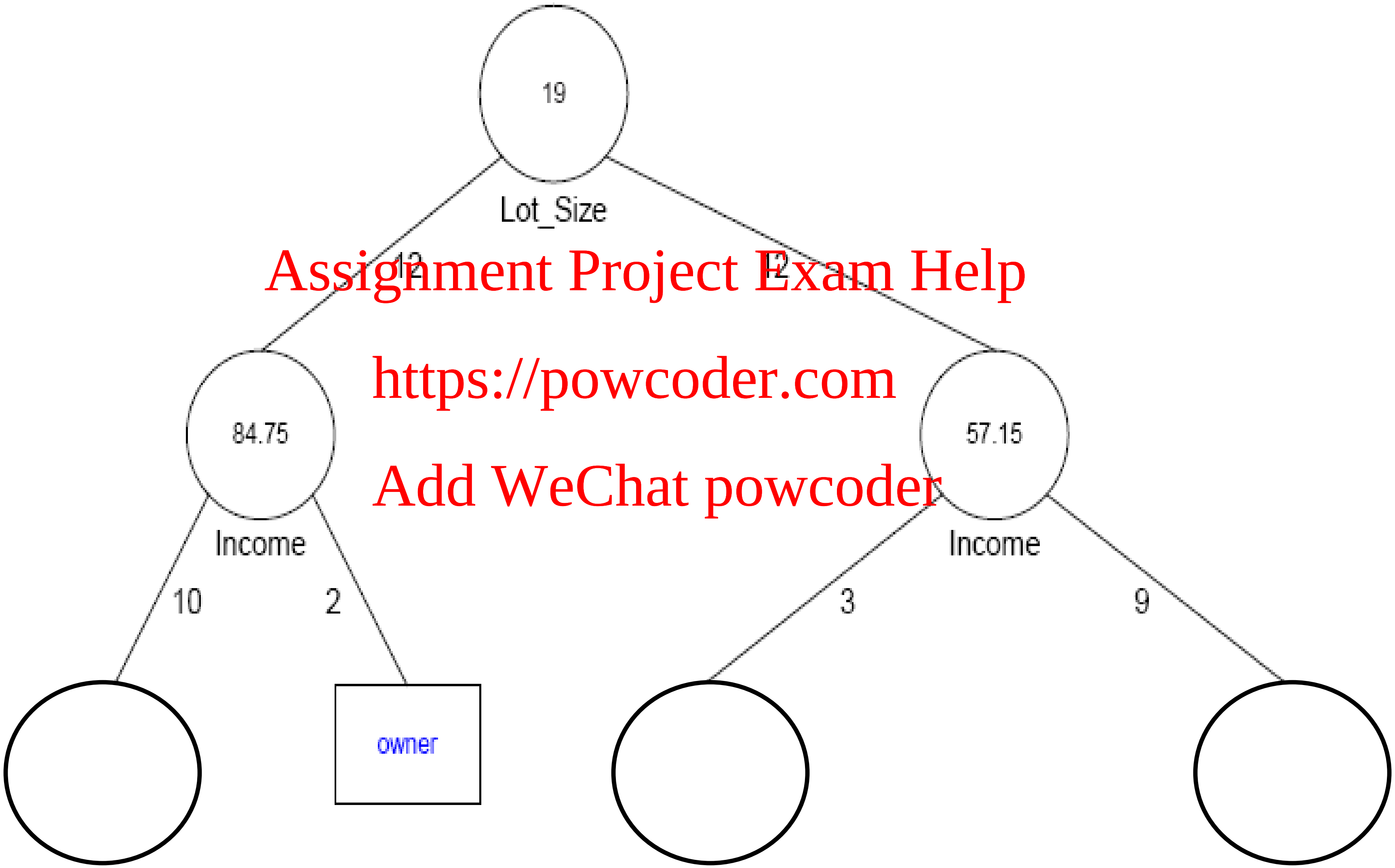


Tree Diagram: First Two Splits

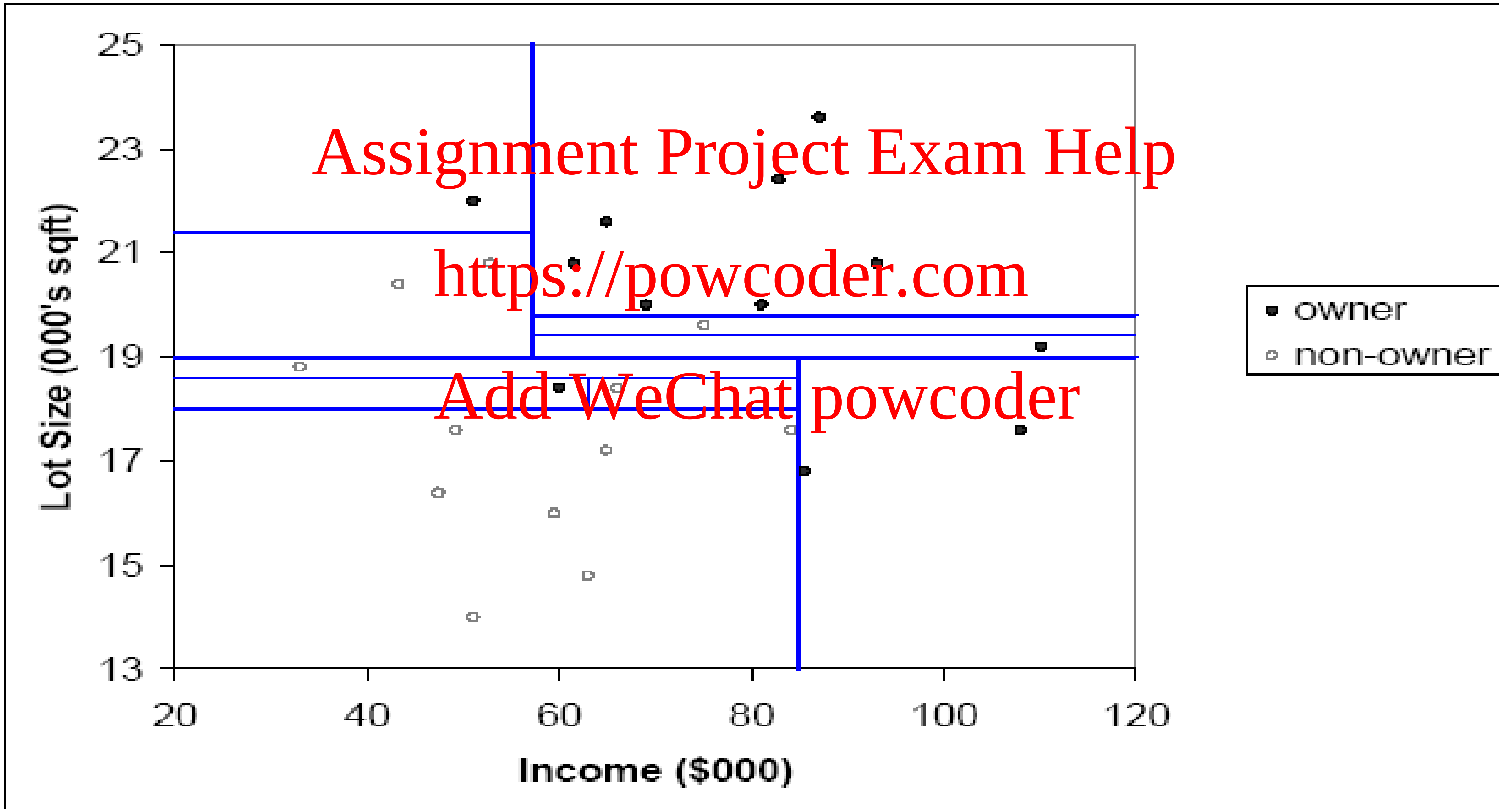




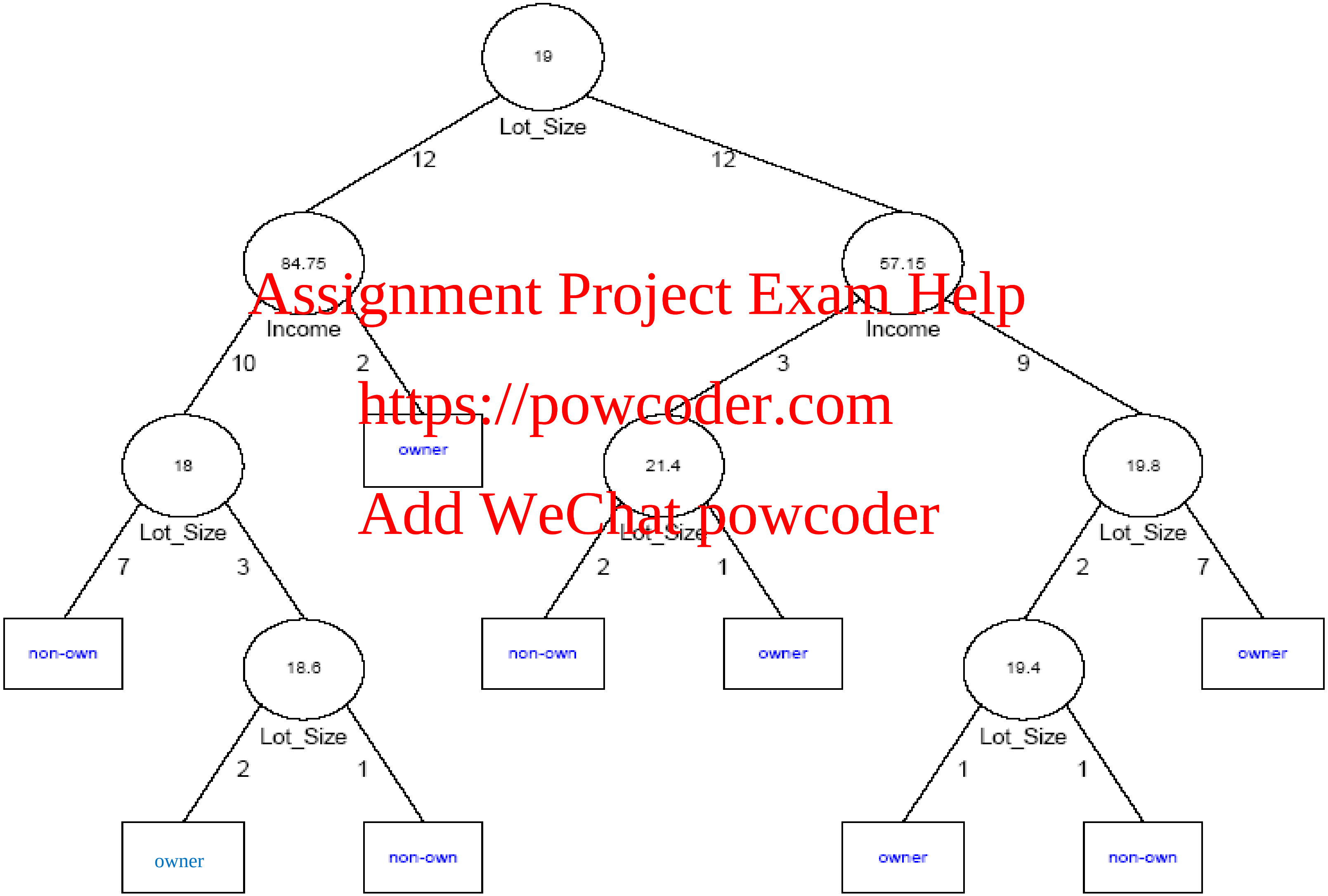
25
23
21
19
17
15
13
20
40
60
80
100
120
Lot Size (000's sqft)
Income ($000)
owner
non-owner


19
Lot_Size
12
12
84.75
57.15
Income
Income
10
2
3
9
owner

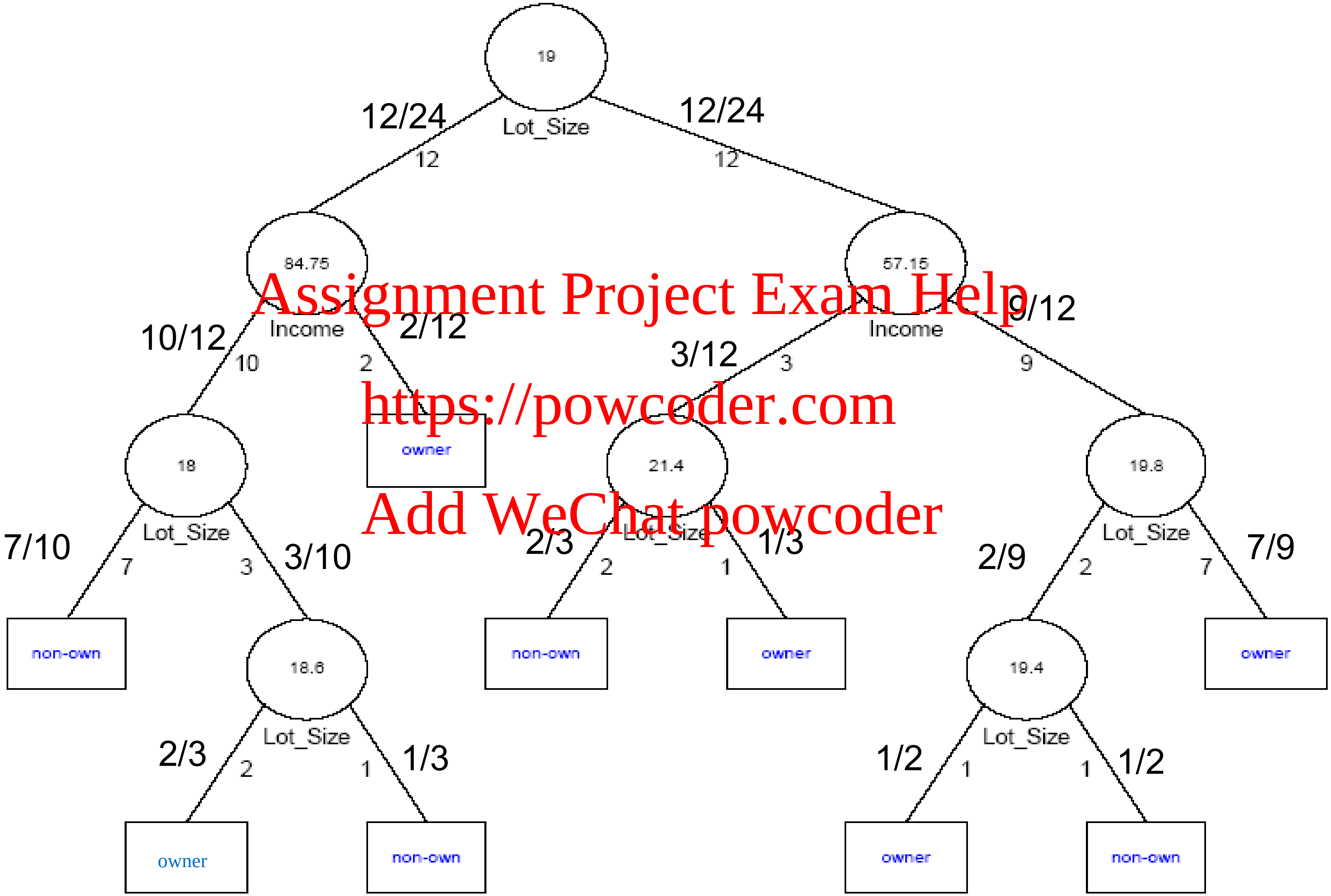
Final Partitioning



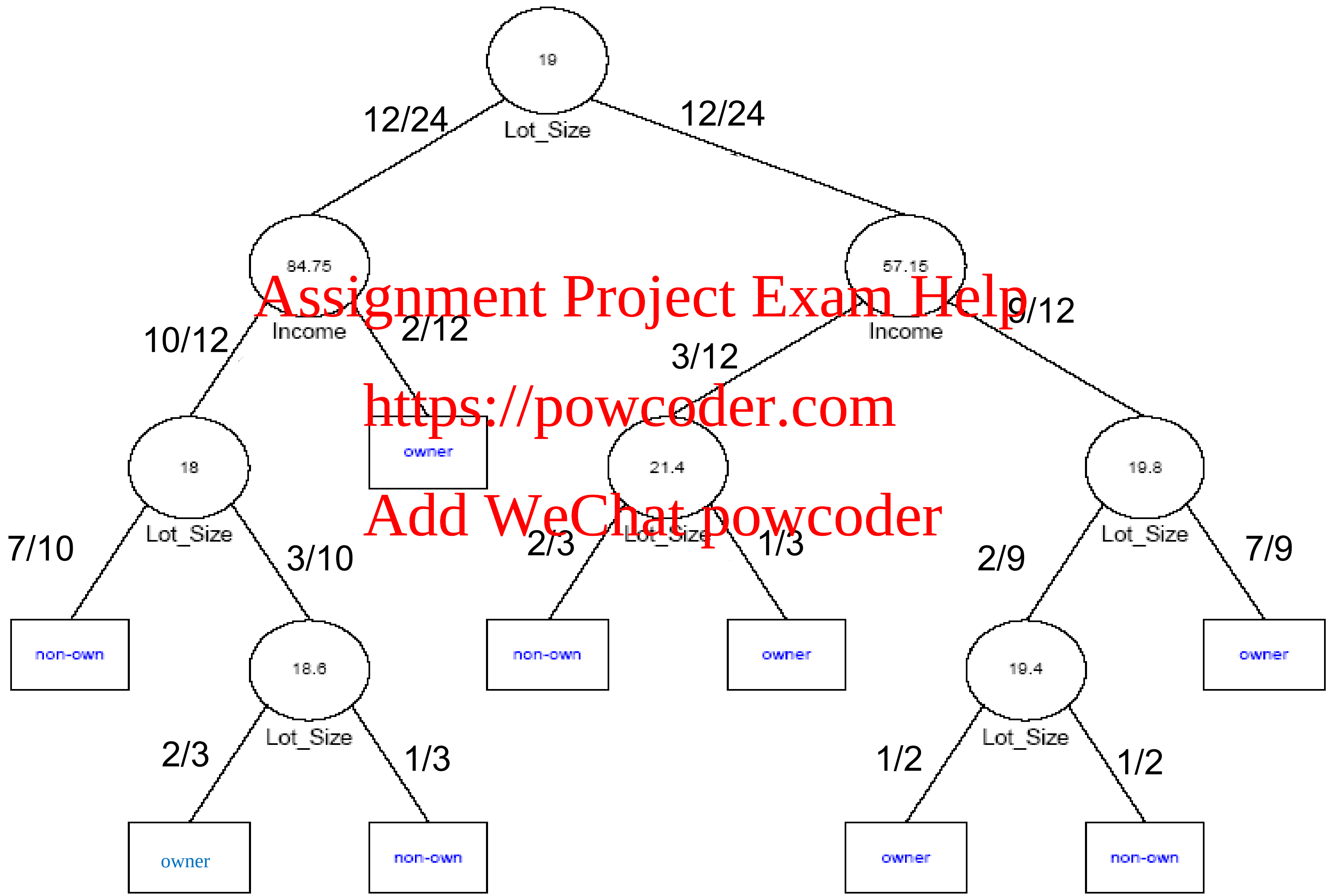
Full Tree


19
Lot_Size
12
12
84.75
Income
57.15
Income
10
2
3
9
18
Lot_Size
owner
21.4
Lot_Size
19.8
Lot_Size
7
3
2
1
2
7
non-own
18.6
Lot_Size
non-own
owner
19.4
Lot_Size
owner
2
1
1
1
owner
non-own
owner
non-own

Calculate the probability of each branch

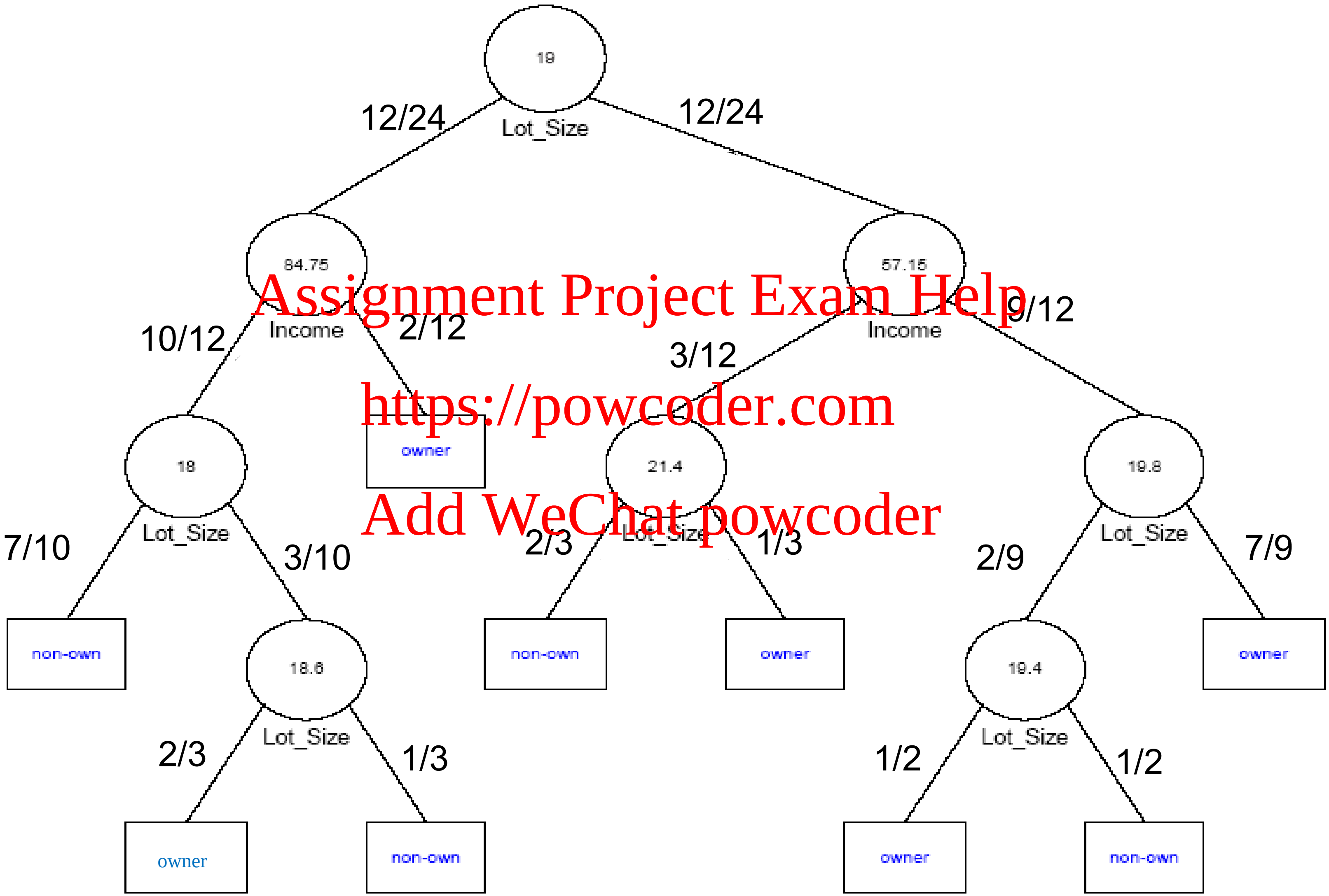


Given lot size = 20, what is the probability of owner?



P(Owner | Lot size = 20) = P(Owner & Lot Size=20)/ (P(Owner & Lot Size=20)+P(Non-Owner & Lot Size=20))

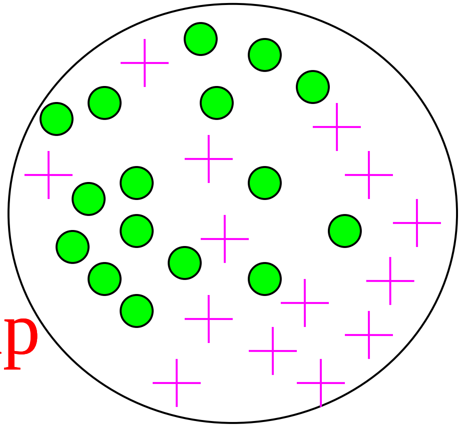
Given Income = 60, what is the probability of owner?



Calculating Impurity

- Impurity = Entropy = $\sum_i - p_i \log_2 p_i$

  p_i is proportion of class i



- For example: our initial population is composed of 16 cases of class “Default” and 14 cases of class “Not default”

Entropy(entire population of examples)=

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

Calculating the Information Gain of a Split

1. For each sub-group produced by the split, calculate the impurity/entropy of that subset.

2. Calculate the weighted entropy of the split by weighting each sub-group's entropy by the proportion of training examples (out of the training examples in the parent node) that are in that subset.

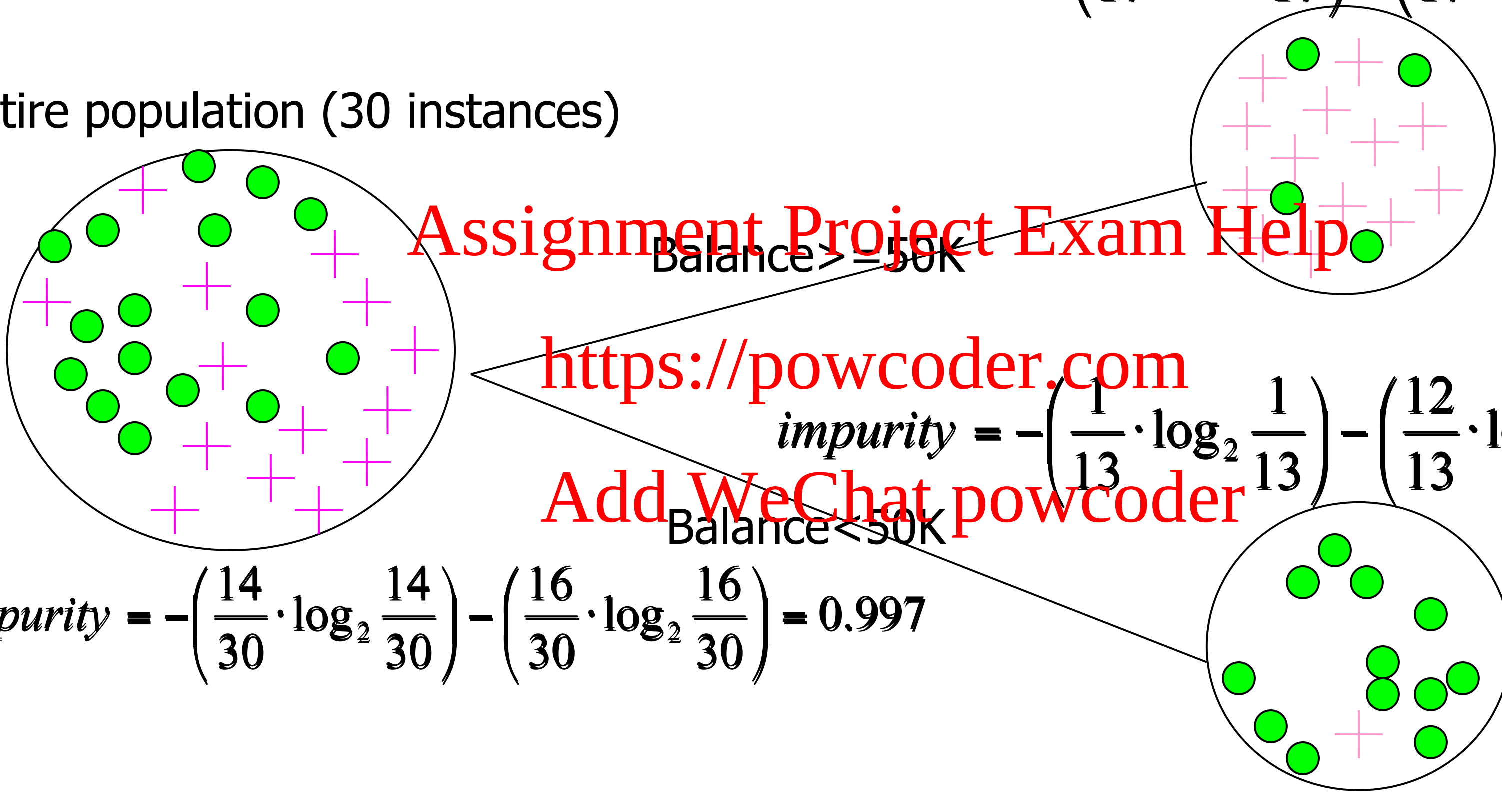
3. Calculate the entropy of the parent node, and subtract the weighted entropy of the child nodes to obtain the information gain for the split.

Calculating Information Gain

Information Gain = Entropy (parent) – Entropy (children)

$$impurity = -\left(\frac{13}{17}\cdot\log_2\frac{13}{17}\right)-\left(\frac{4}{17}\cdot\log_2\frac{4}{17}\right)=0.787$$

Entire population (30 instances)



$$impurity = -\left(\frac{1}{13}\cdot\log_2\frac{1}{13}\right)-\left(\frac{12}{13}\cdot\log_2\frac{12}{13}\right)=0.391$$

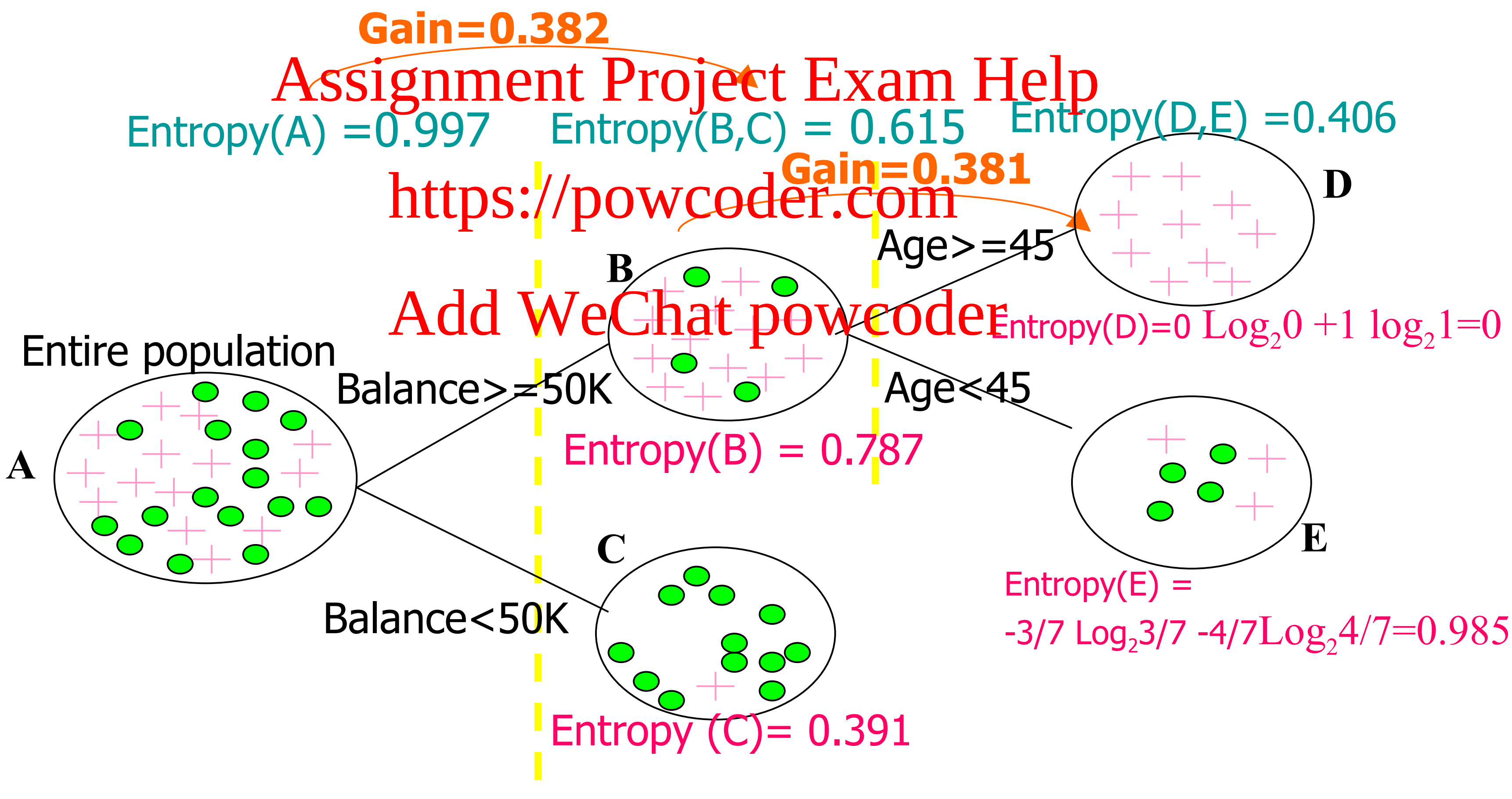
$$impurity = -\left(\frac{14}{30}\cdot\log_2\frac{14}{30}\right)-\left(\frac{16}{30}\cdot\log_2\frac{16}{30}\right)=0.997$$

(Weighted) Average Entropy of Children = $\left(\frac{17}{30}\cdot 0.787\right)+\left(\frac{13}{30}\cdot 0.391\right)=0.615$

Information Gain= 0.997 - 0.615 = 0.382

Information Gain

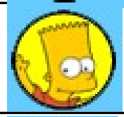
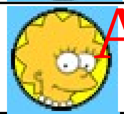
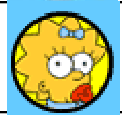
Information Gain = Entropy (parent) – Entropy (children)

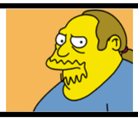


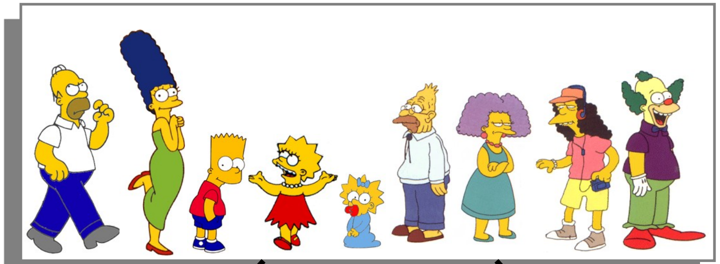
Which attribute to split over?

- At each node examine splits over each of the attributes
- Select the attribute for which the maximum information gain is obtained
  - For a continuous attribute, also need to consider different ways of splitting (>50 or <=50; >60 or <=60)
  - For a categorical attribute with lots of possible values, sometimes also need to consider how to group these values (branch 1 corresponds to {A,B,E} and branch 2 corresponds to {C,D,F,G})

Example 2

Person		Hair Length	Weight	Age	Class
	Homer	0”	250	36	M
	Marge	10”	150	34	F
	Bart	2”	90	10	M
	Lisa	6”	78	8	F
	Maggie	4”	20	1	F
	Abe	1”	170	70	M
	Selma	8”	160	41	F
	Otto	10”	180	38	M
	Krusty	6”	200	45	M

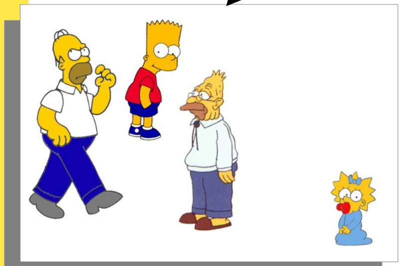
	Comic	8”	290	38	?



$$Entropy(4\textbf{F},5\textbf{M}) = -(4/9)\log_2(4/9) - (5/9)\log_2(5/9)$$
$$= \mathbf{0.9911}$$

yes | no

Hair Length <= 5?



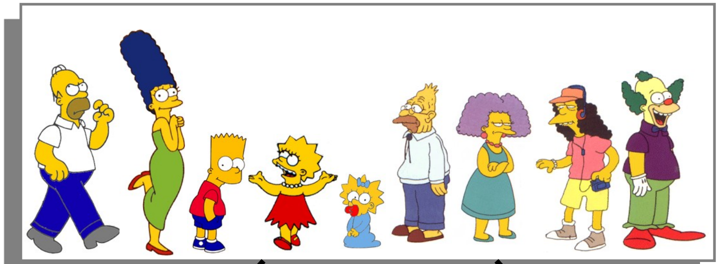
Let us try splitting on *Hair length*

$$Entropy(1\textbf{F},3\textbf{M}) = -(1/4)\log_2(1/4) - (3/4)\log_2(3/4)$$
$$= \mathbf{0.8113}$$

$$Entropy(3\textbf{F},2\textbf{M}) = -(3/5)\log_2(3/5) - (2/5)\log_2(2/5)$$
$$= \mathbf{0.9710}$$

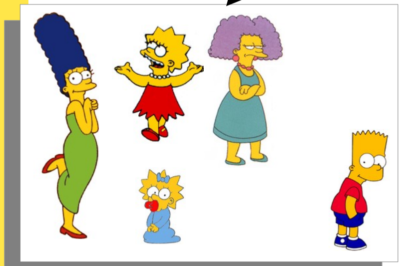
***Gain*= Entropy of parent – Weighted average of entropies of the children**

Gain(Hair Length <= 5) = **0.9911** – (4/9 * **0.8113** + 5/9 * **0.9710**) = **0.0911**



$$Entropy(4\textbf{F},5\textbf{M}) = -(4/9)\log_2(4/9) - (5/9)\log_2(5/9)$$
$$= \mathbf{0.9911}$$

yes — Weight <= 160? — no

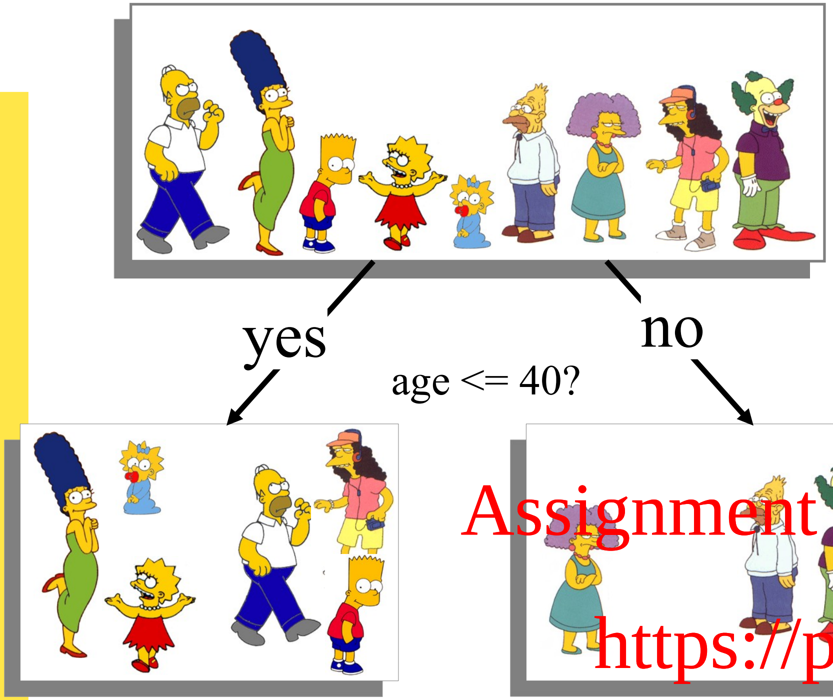


Let us try splitting on *Weight*

$$Entropy(4\textbf{F},1\textbf{M}) = -(4/5)\log_2(4/5) - (1/5)\log_2(1/5)$$
$$= \mathbf{0.7219}$$

$$Entropy(0\textbf{F},4\textbf{M}) = -(0/4)\log_2(0/4) - (4/4)\log_2(4/4)$$
$$= \mathbf{0}$$

Gain(Weight <= 160) = **0.9911** – (5/9 * **0.7219** + 4/9 * **0**) = **0.5900**



Entropy(4**F**,5**M**) = -(4/9)log$_2$(4/9) - (5/9)log$_2$(5/9)
= **0.9911**

Let us try splitting on *Age*

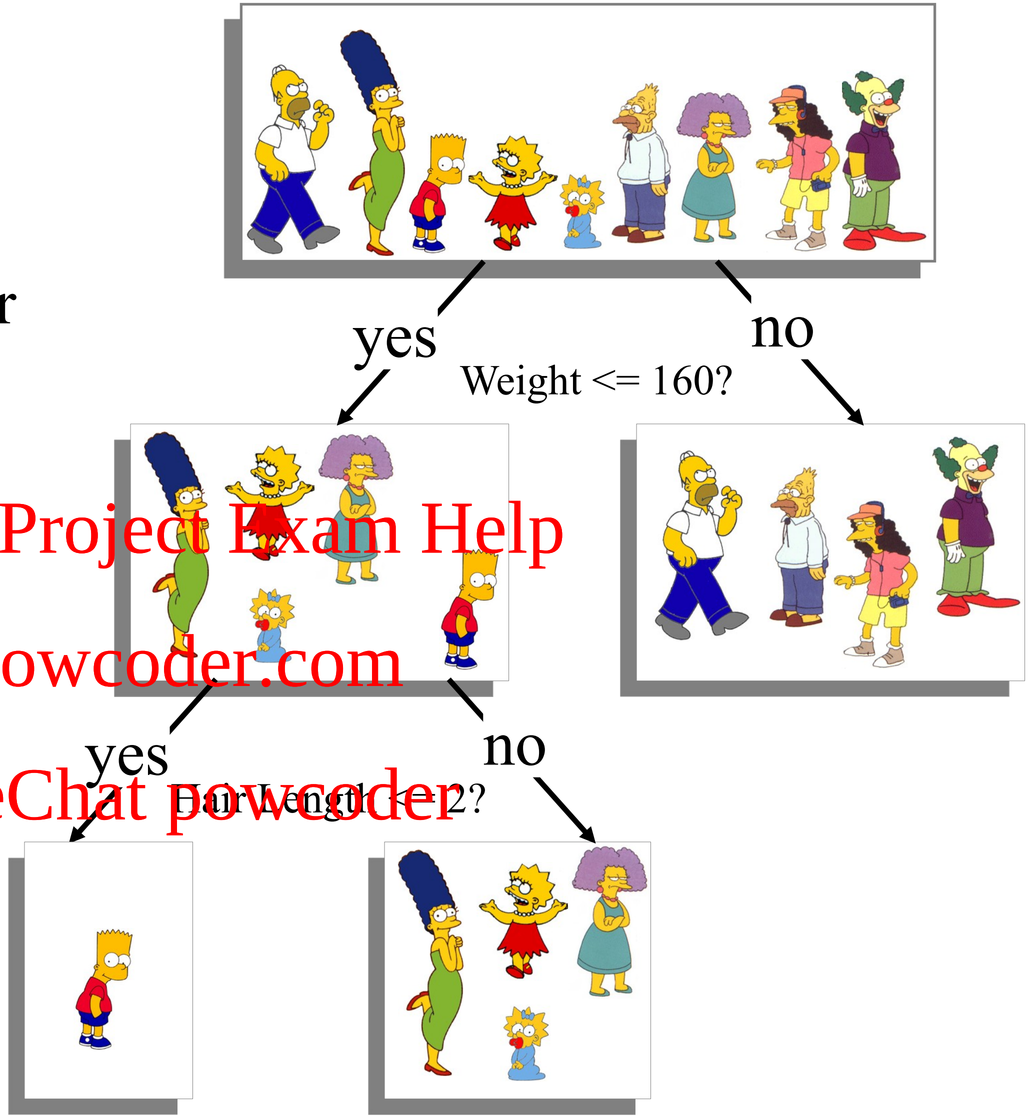
Entropy(3**F**,3**M**) = -(3/6)log$_2$(3/6) - (3/6)log$_2$(3/6)
= ***1***

Entropy(1**F**,2**M**) = -(1/3)log$_2$(1/3) - (2/3)log$_2$(2/3)
= ***0.9183***

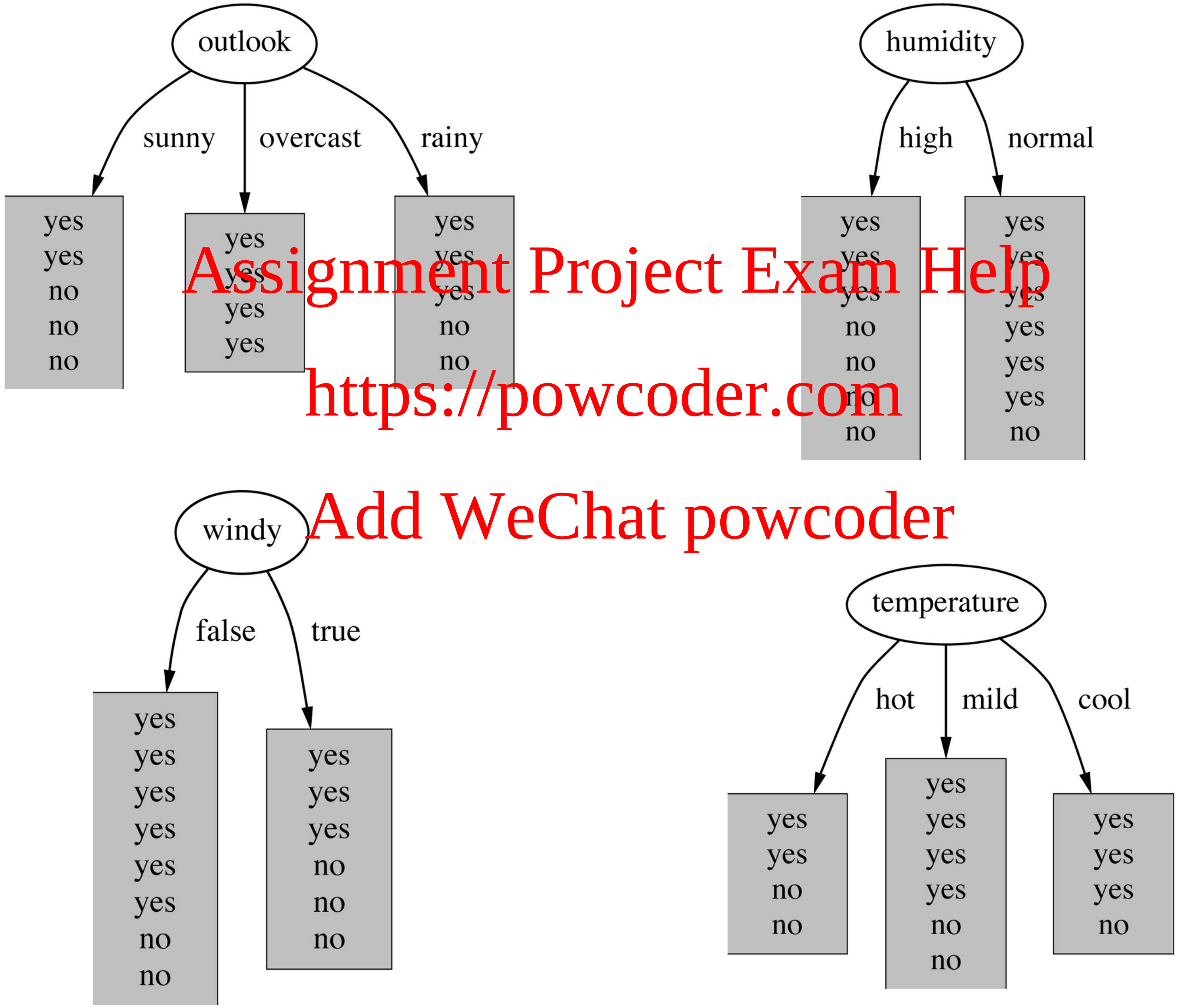
Gain(Age <= 40) = **0.9911** – (6/9 * **1** + 3/9 * **0.9183**) = **0.0183**

Of the 3 features we had, *Weight* was the best. But while people who weigh over 160 are perfectly classified (as males), the under 160 people are not perfectly classified… So we simply continue splitting…

This time we find that we can split on *Hair length,* and then we are done!



Example 3: Which attribute to split on?



Exercise – Decision Tree

Customer ID	Student	Credit Rating	Class: Buy PDA
1	No	Fair	No
2	No	Excellent	No
3	No	Fair	Yes
4	No	Fair	Yes
5	Yes	Fair	Yes
6	Yes	Excellent	No
7	Yes	Excellent	Yes
8	No	Excellent	No

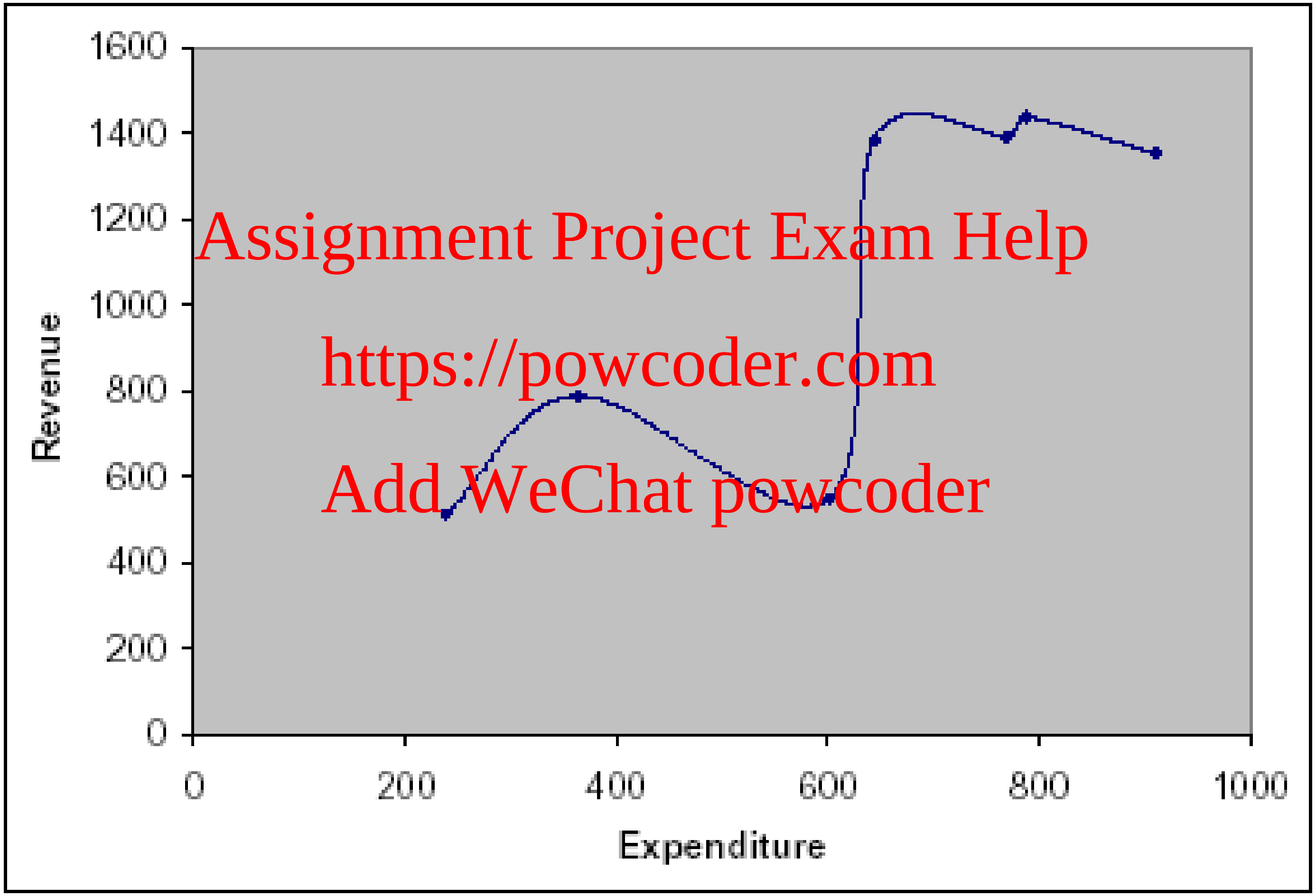
Which attribute to split on first?

$\log_2(2/3) = -0.585$, $\log_2(1/3) = -1.585$, $\log_2(1/2) = -1$, $\log_2(3/5) = -0.737$,
$\text{Log}_2(2/5) = -1.322$, $\log_2(1/4) = -2$, $\log_2(3/4) = -0.415$

Building a Tree - Stopping Criteria

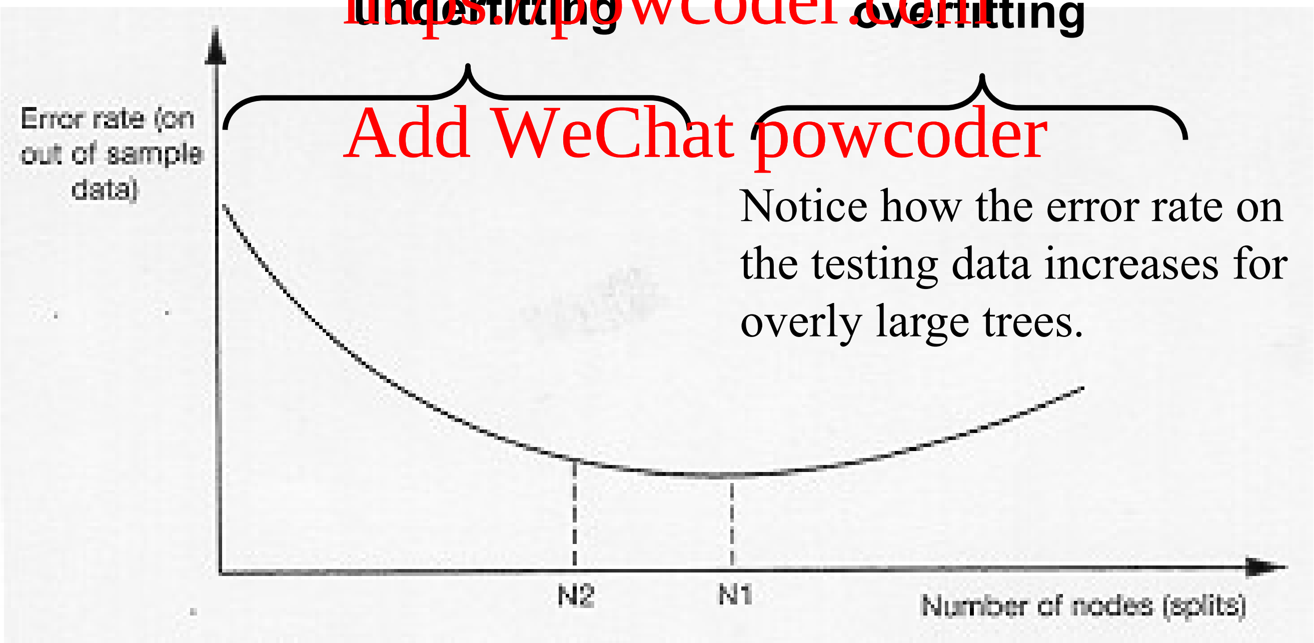
- You can stop building the tree when:
  - The impurity of all nodes is zero: Problem is that this tends to lead to bushy, highly-branching trees, often with one example at each node.
  - No split achieves a significant gain in purity (information gain not high enough).
  - Node size is too small: That is, there are less than a certain number of examples, or proportion of the training set, at each node.

Over-fitting



Overfitting & Underfitting

- **Overfitting:** the model performs poorly on new examples (e.g. testing examples) as it is too highly trained to the specific training examples (pick up patterns and noises).
- **Underfitting:** the model performs poorly on new examples as it is too simplistic to distinguish between them (i.e. has not picked up the important patterns from the training examples)

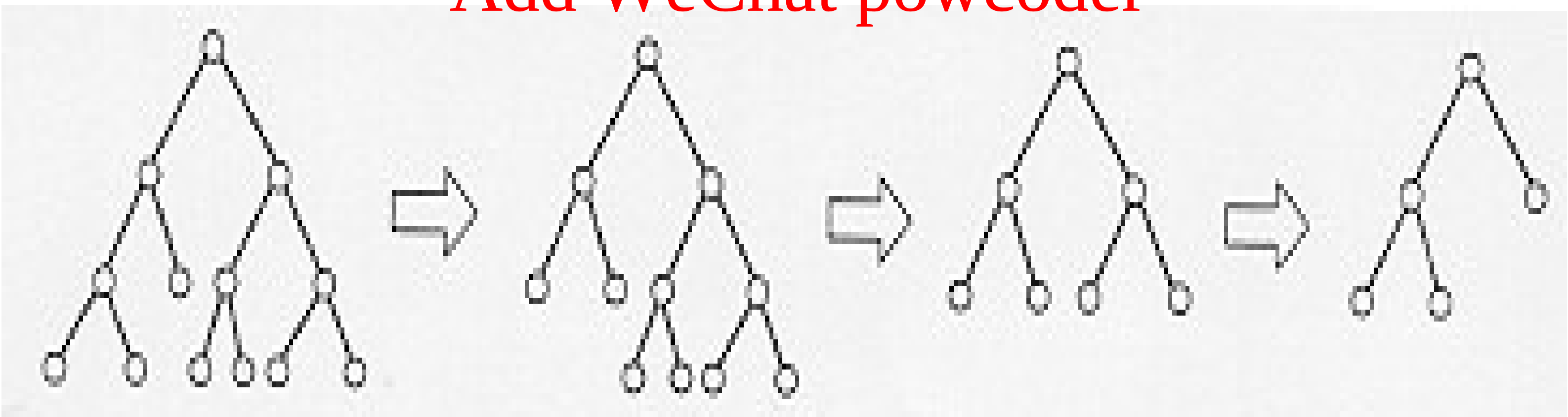


Pruning

A decision trees is typically more accurate on its *training* data than on its *test* data. Removing branches from a tree can often improve its accuracy on a test set.

Classification and Regression Tree (CART) : Use validation data to delete “weak” sub-trees

Assess whether splitting a node improves purity by a statistically significant amount

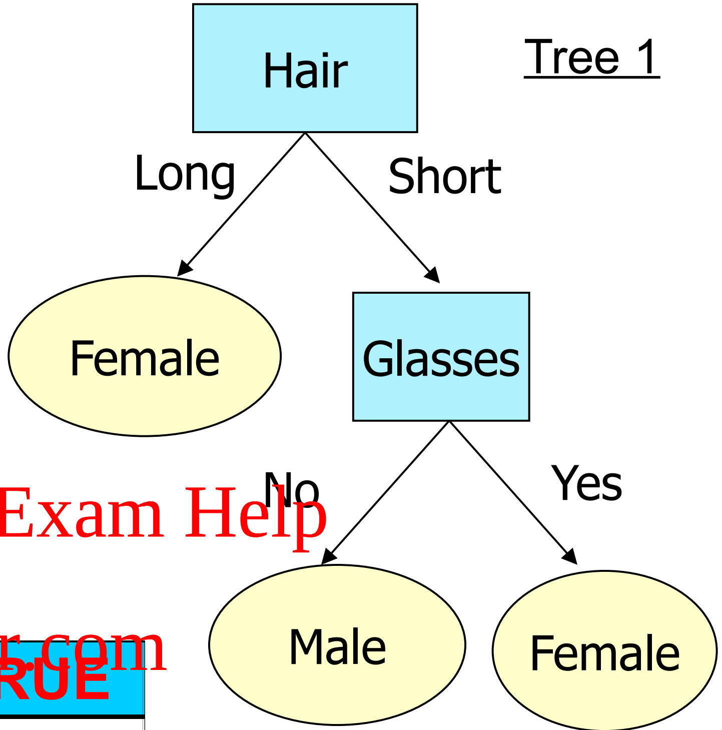


Training

Name	Hair	Glasses	Class
Mary	Long	No	Female
Mike	Short	No	Male
Bill	Short	No	Male
Jane	Long	No	Female
Ann	Short	Yes	Female

100% accurate on training data

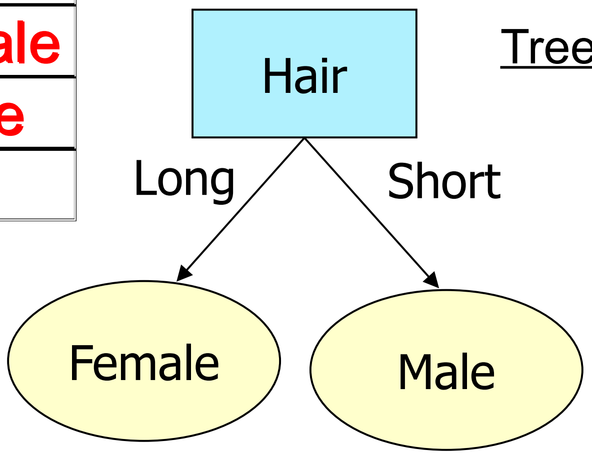
Tree 1



Testing

Hair	Glasses	Tree 1	Tree 2	TRUE
Short	Yes	Female	Male	Male
Short	No	Male	Male	Female
Long	No	Female	Female	Female
Short	Yes	Female	Male	Male
	Error:	75%	25%	

Tree 2



There are many possible splitting rules that perfectly classify the data, but will not generalize to future datasets.

Decision Tree Classification in a Nutshell

- Decision tree
  - A tree structure
  - Internal node denotes a test on an attribute
  - Branch represents an outcome of the test
  - Leaf nodes represent class labels
- Decision tree generation consists of two phases
  - Tree construction
    - At start, all the training examples are at the root
    - Partition examples recursively based on selected attributes
  - Tree pruning
    - Identify and remove branches that reflect noise or outliers
    - To avoid overfitting
- Use of decision tree: Classifying an unknown sample
  - Test the attribute values of the sample against the decision tree

Strengths

- In practice: One of the most popular methods
  - Very comprehensible – the tree structure specifies the entire decision structure
    - Easy for decision makers to understand model's rational
    - Map nicely to a set of business rules
  - Relatively easy to implement
- Very fast to run (to classify examples) with large data sets
- Good at handling missing values: just treat "missing" as a value – can become a good predictor
- Weakness
  - Bad at handling continuous data, good at categorical input and output.

Which attribute will you use as the root of the tree, given the following information:

$\text{gain}(\textit{Outlook}) = 0.247 \text{ bits}$

$\text{gain}(\textit{Temperature}) = 0.029 \text{ bits}$

$\text{gain}(\textit{Humidity}) = 0.152 \text{ bits}$

$\text{gain}(\textit{Windy}) = 0.048 \text{ bits}$

A: Outlook

B: Humidity

C: Windy

D: Temperature

E: None of the above

What is overfitting?

A: When the model fit is better on the top side

B: When the model fit is worse on the top side

C: When the model captures the correct trend and has best accuracy

D: When the model captures noise in the data, hurting accuracy

E: None of the above

Weka Example – Classification using Naïve Bayes

- Download file from Canvas:
  - 4bank-data-8.arff
- Switch tab to "classify"
- Select method: NaiveBayes
- Verify class variable set to "pep"
- Use 10 fold cross validation
- Run classifier
- Examine confusion matrix

Weka Exercise

- Follow instructions on
- http://facweb.cs.depaul.edu/mobasher/classes/ect584/WEKA/classify.html
- Data files posted on Canvas
- We will use J48 which is an implementation of the C4.5 algorithm

Next Session

- Association Rules