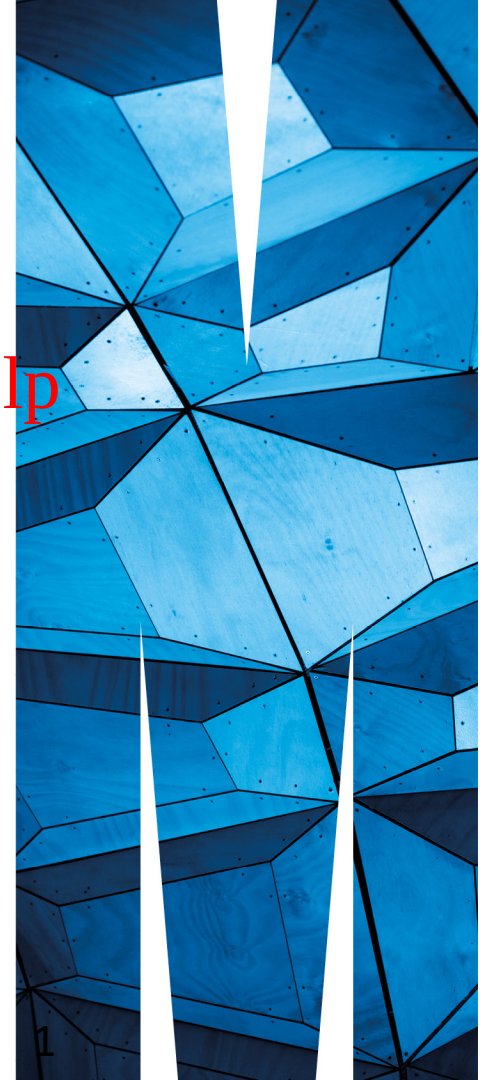
MONASH University

MONASH BUSINESS SCHOOL

Classification

Supervised Learning Technique

Presented by Klaus Ackermann



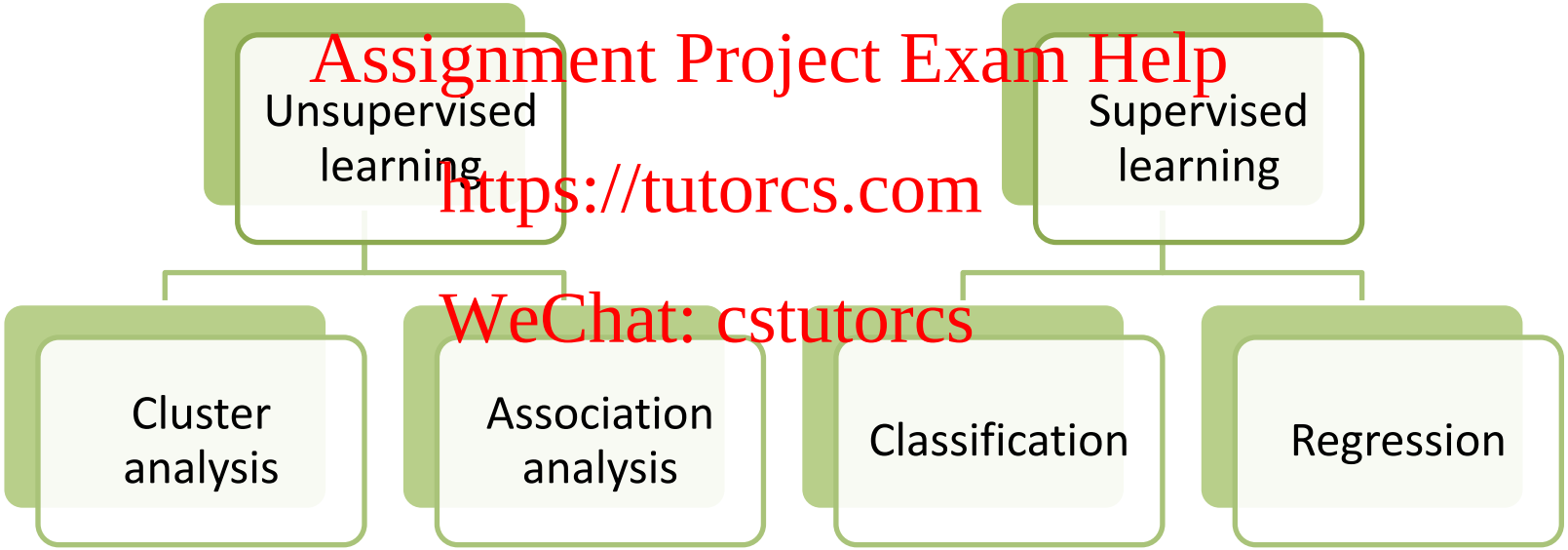
GROUP OF EIGHT AUSTRALIA

AACSB ACCREDITED

EFMD EQUIS ACCREDITED

ASSOCIATION AMBA ACCREDITED

Introduction



Supervised Learning

- Start with some data that provides information about a set of “cases”
- For each case, we have predictor variables and a target variable

- Divide the data up into **training data** and **test data**
  - When dividing, be careful not to “cheat”

  - The test split must have the same form as if you would use in in the real world:
    - Cross-section Prediction: Random Splits
    - Temporal Prediction: Use cases from the future for the test set, not the past

Supervised Learning

- Use the training data to develop a model predicting the value of the target variable for given values of the predictor variables

- The test data must not be viewed at all until the model is built

- The model is judged on how well it "predicts" the test data, in other words we care about the **out of sample performance**. *How well would the model do when new information is collected that it had not seen before?*

Supervised Learning

- Usually try more than one model and compare their performance

- To do this without data snooping, divide the data into three parts: Training data, validating data, and test data

- The test data **must not** be involved in building or choosing the model

Supervised Learning

- The data consists of a (usually large) number of cases
- For each *case*, the data provides inputs $x_1, \ldots, x_n$ and a corresponding value for the target variable

- For example, input variables may be
  Age, Sex, Income, Number of children, Size of loan

- and the target variable may be
  "*Default on loan*" with possible values *Yes* and *No*
- Such data may be divided into parts to be used for training and testing

Advantages of Trees

- There are many classification procedures

- We focus on trees because they
  - Can be built efficiently
  - Easy to interpret
  - Can be building blocks for more accurate

- We consider *classification* and *regression* trees

Advantages of Trees – KTC Data Example

- Looking at those who purchased time share in a holiday house
- Data available
  - Numeric: Age, Income, Number of children
  - Binary: female, married, car loan, mortgage

- Predict price paid

  OR

- Predict bought or not

Classification and Regression Trees CART

MONASH University

- If the target variable is categorical, e.g. with possible values *Yes* and *No* then for given values of the predictor variables, the decision tree predicts whether the target variable takes the value *Yes*
  - we have a *Classification Tree*

- If the target variable is numerical, then for given values of the predictor variables, the decision tree predicts the value of the target variable
  - we have a *Regression Tree*

- *The predictor variables can be of various types in each case*

Example: Ride-on Mowers

- Predictor variables: Lot size, Income
- Target variable: *Owner*
  Owner of ride-on mower: **Own** or **Not**
- Before introducing a tree classification method, let's investigate how ownership appears to depend on these two variables
- We have a very small data set and we use all the data to set up a model. We'll deal with validation and testing later

MONASH University

Ride-on Mower Data

HH_ID	Income	Lot_Size	Owner	HH_ID	Income	Lot_Size	Owner
1	60	18.4	Own	13	75	19.6	Not
2	85.5	16.8	Own	14	52.8	20.8	Not
3	64.8	21.6	Own	15	64.8	17.2	Not
4	61.5	20.8	Own	16	43.2	20.4	Not
5	87	23.6	Own	17	84	17.6	Not
6	110.1	19.2	Own	18	49.2	17.6	Not
7	108	17.6	Own	19	59.4	16	Not
8	82.8	22.4	Own	20	66	18.4	Not
9	69	20	Own	21	47.4	16.4	Not
10	93	20.8	Own	22	33	18.8	Not
11	51	22	Own	23	51	14	Not
12	81	20	Own	24	63	14.8	Not

MONASH University

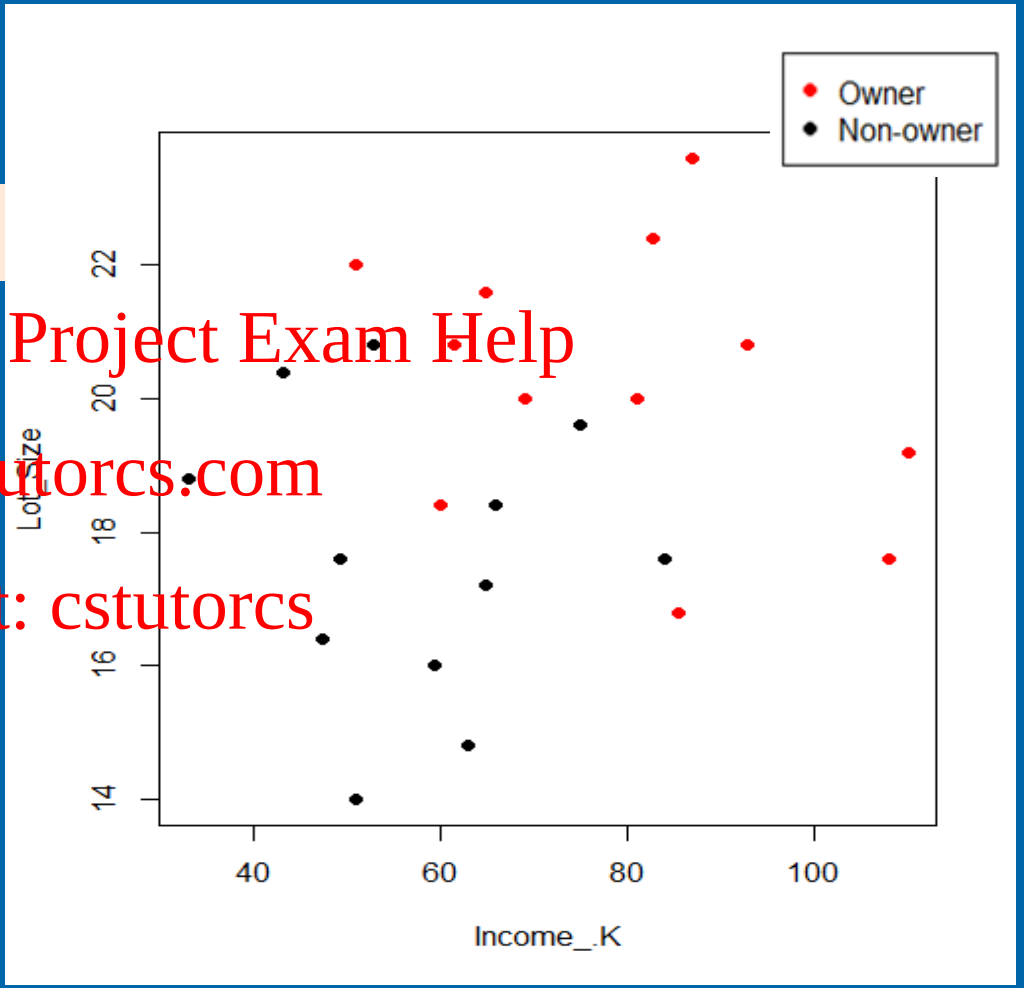
Ride-on Mower Data

We try to divide the data into a small number of rectangles, such that each rectangle has either

- mostly red, or
- mostly black dots.

We only use horizontal or vertical lines.

Here, we try doing this “by hand”



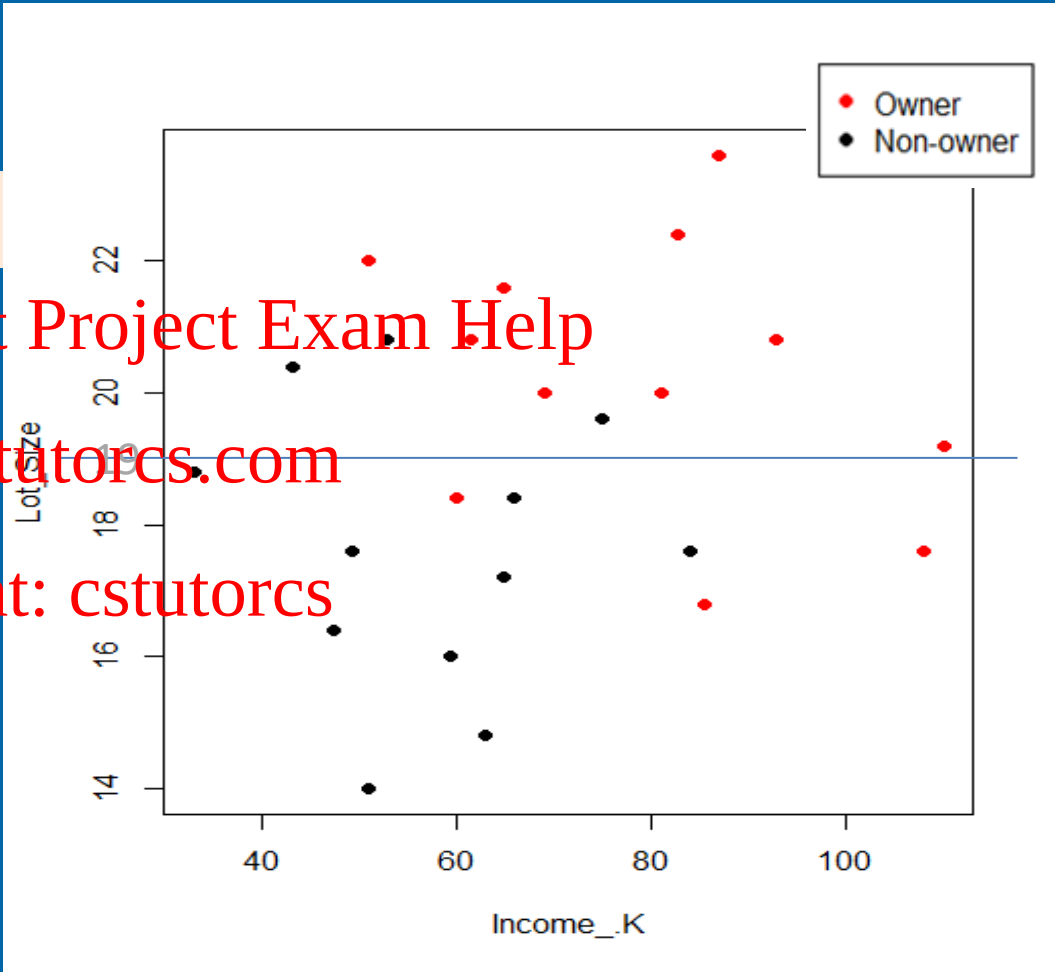
MONASH University

Partition 1

If lot size > 19, likely to be an owner

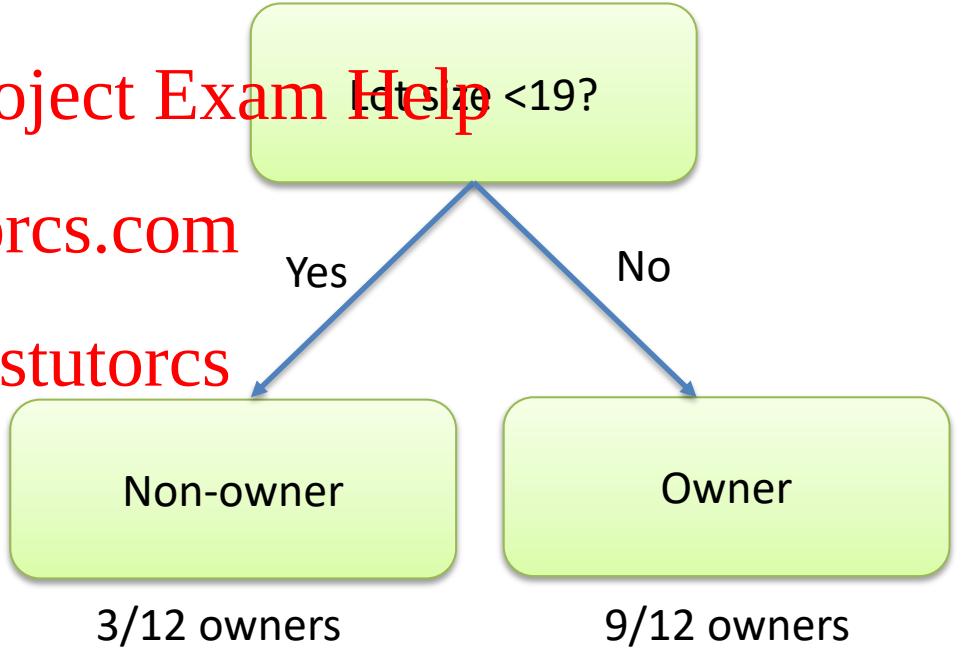
If lot size < 19, likely to be a non-owner

We are trying to find the best separating line



Tree 1

- If I only know about lot size, then based on this tree, I would predict that Lot_size < 19 is a non-owner
- Lot_size > 19 is an owner
- This is based on the largest classification group at the end of the tree (“leaf”)

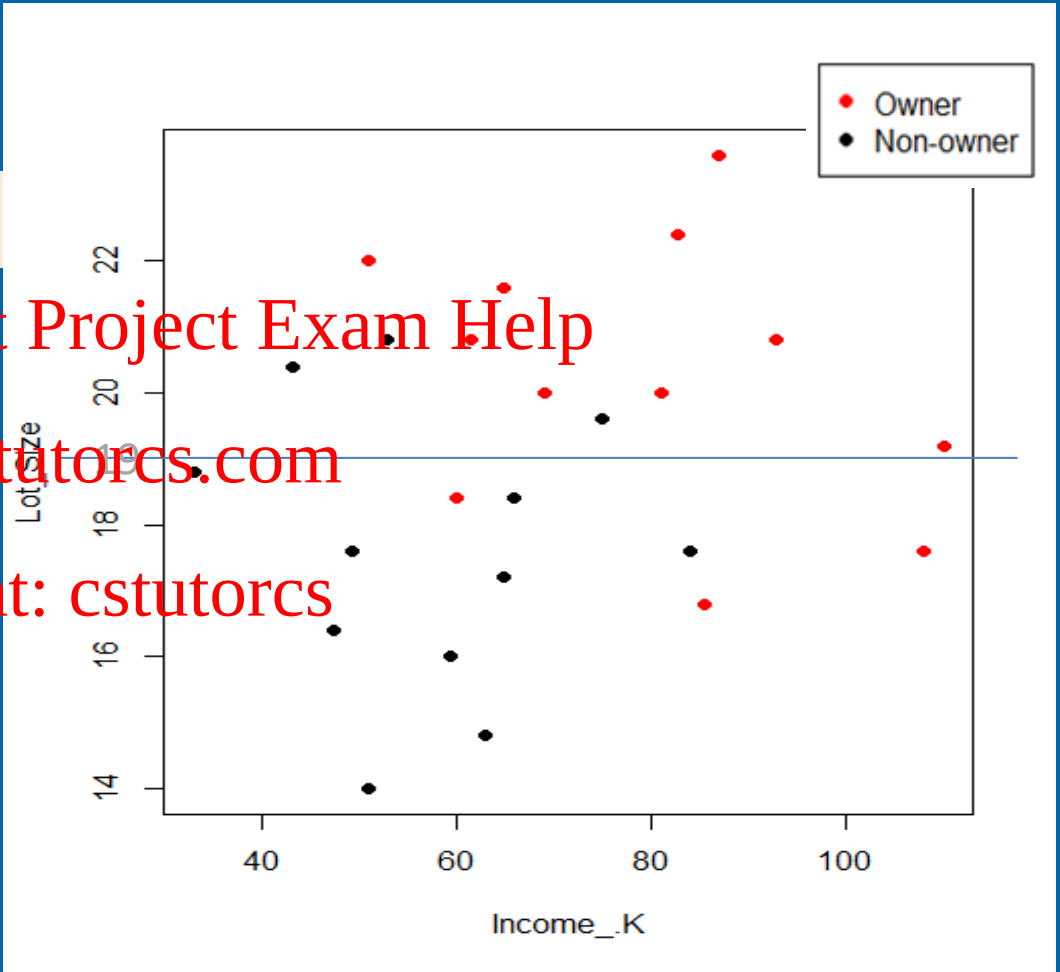


MONASH University

Partition 1

We are trying to choose the division into two parts of the data to minimise the impurity.
We choose which variable to split, and where to split it.
Choices: half way between pairs of consecutive values for the variable

Recursive partitioning



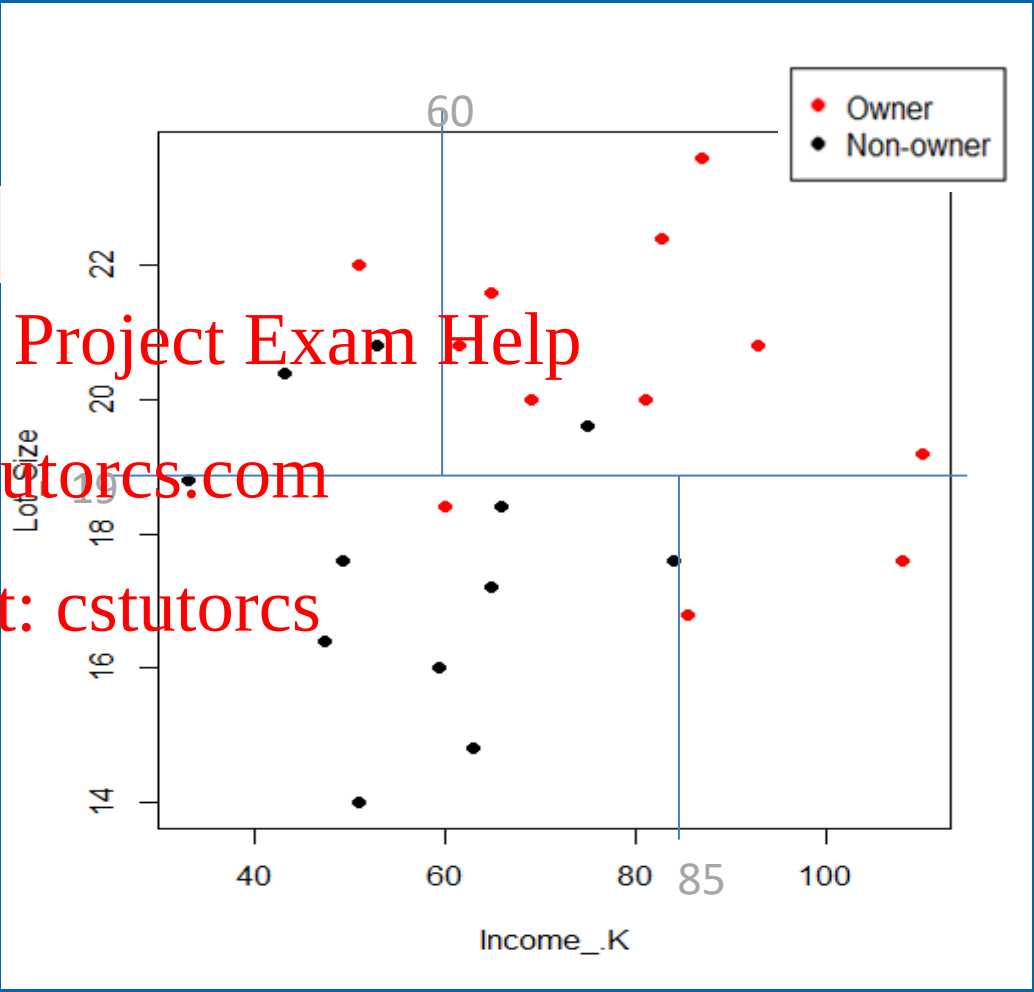
MONASH University

Partition 2

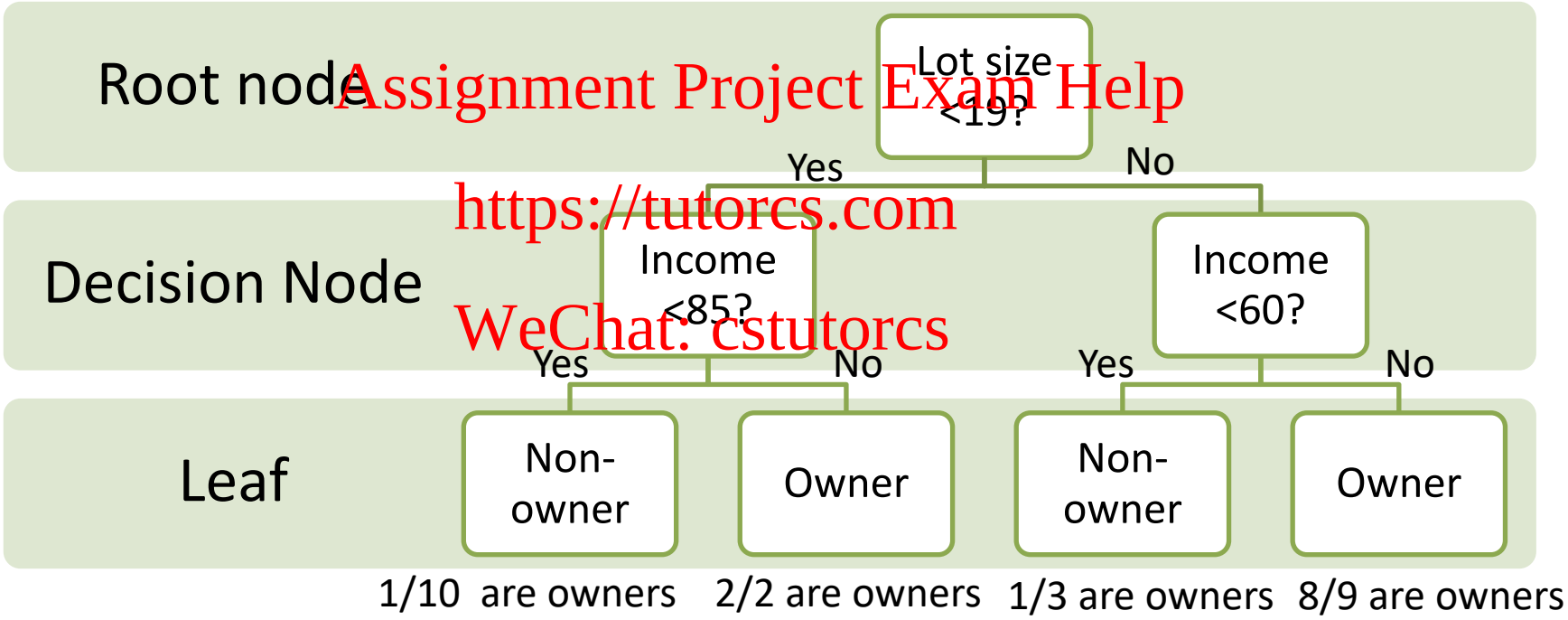
If lot size > 19, and income > 60,
likely to be an owner

If lot size > 19, and income < 60,
likely to be a non-owner

If lot size < 19 and income < 85,
likely to be a non-owner



Tree 2



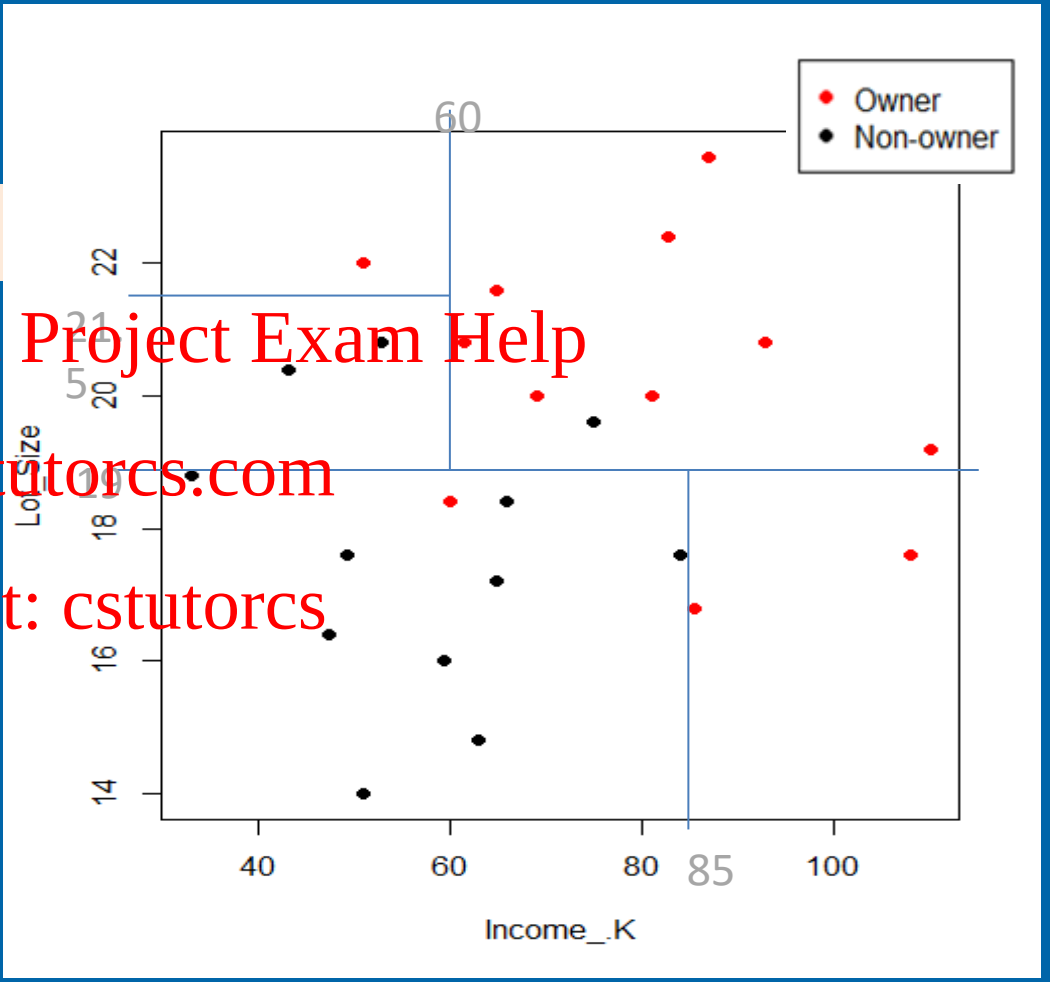
MONASH University

Partition 3

If lot size > 19, and income > 60,
likely to be an owner

If lot size > 19, and income < 60,
likely to be a non-owner

If lot size < 19 and income < 85,
likely to be a non-owner



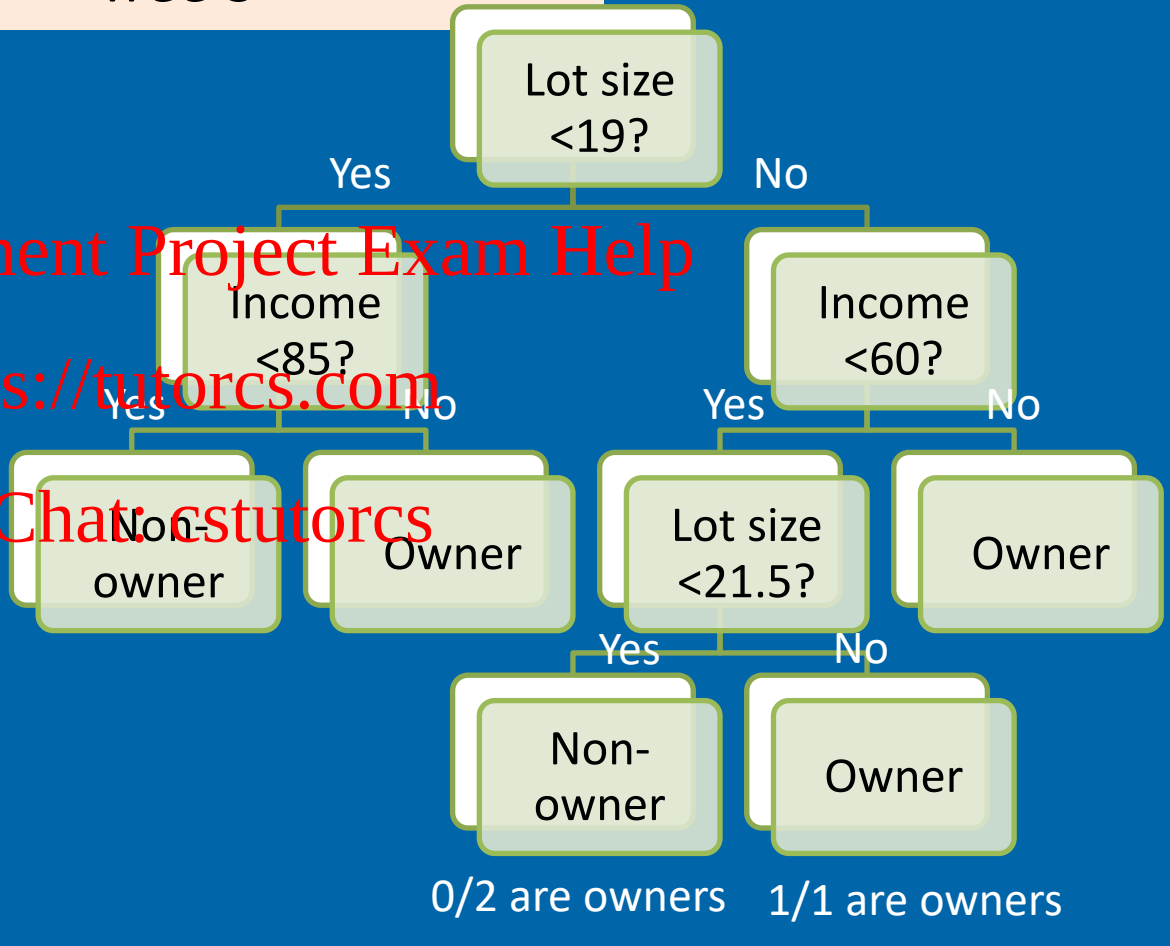
Tree 3

New case: Lot size 18.5, income $61,000
How classified?

Trace a path from the root node to a leaf.

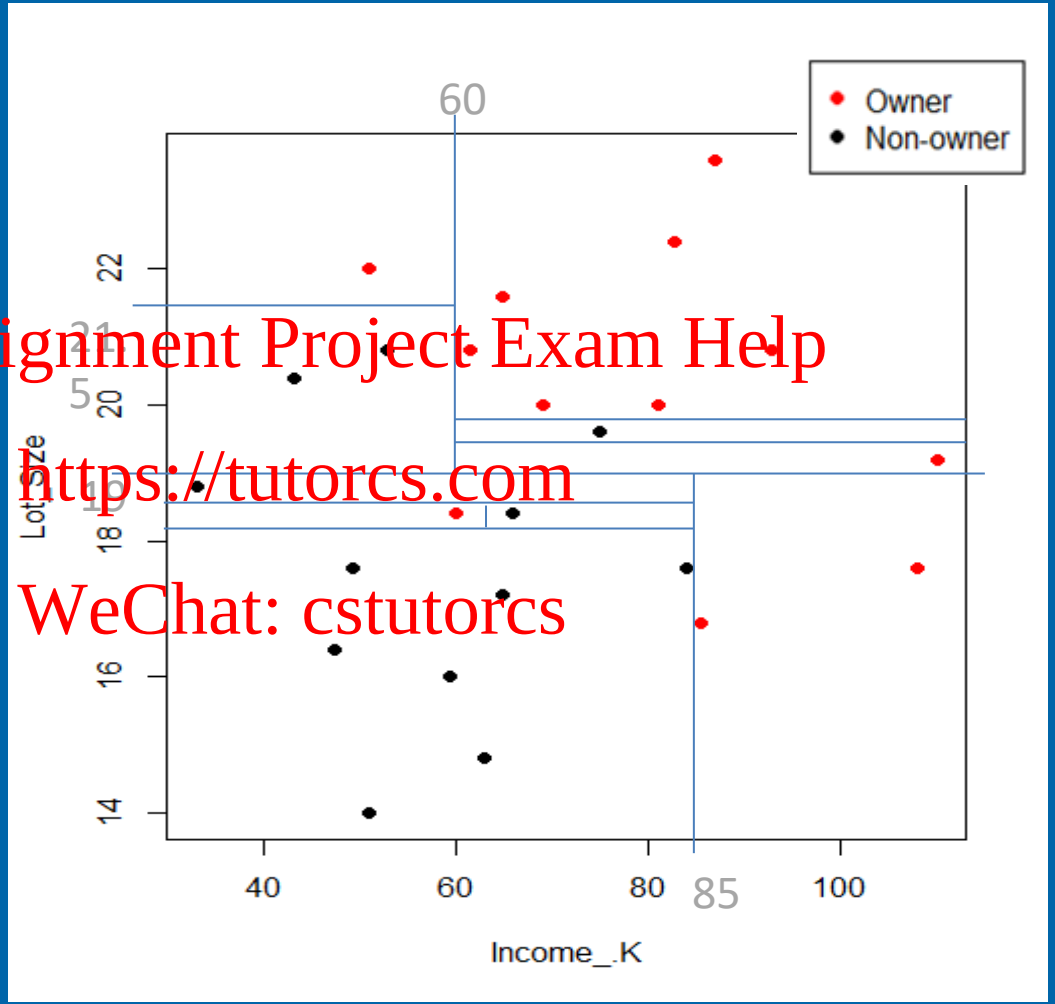
At each node, choose the option corresponding to the new case

At the leaf, if the proportion of non-owners is higher than owners, predict the new case is a non-owner. And vice versa.



MONASH University

Overfitting



Systematic Construction of Trees

- We have been aiming to minimise disorder – trying to divide the data up so that in each partition there is just one colour, or at least very few of the other colour.
- So far we have done it in an ad hoc way.

If we specify a *measure of disorder*, then we can use computing power to choose at which value of which variable to put the next dividing line.

Choices: half way between pairs of consecutive values for the variable

Measuring Disorder

A measure of disorder is referred to as *deviance*

In this example we have two possible values of the target variable. We look at measures of disorder in this case

- In a particular segment of the space (at a particular leaf):
  - Suppose the proportion of red dots is ***p***, and of black dots is **1 – *p***
  - If *p = 0.5,* we have *maximum disorder*
  - If *p* = 0 or 1, we have *minimum (zero) disorder*

Measuring Disorder

- When there are two values of the target variable, the *Gini index* for disorder is $2p(1-p)$

This is a parabola, maximum at $p = 0.5$ and 0 at $p = 0$ or p = 1

If there are *m* possible outcomes: $Gini = \sum_{k=1}^{m} p_k(1-p_k)$

Measuring Disorder

Other measures of disorder

Entropy = $-2[p\log(p)+(1-p)\log(1-p)]$

- (very similar shape to Gini)

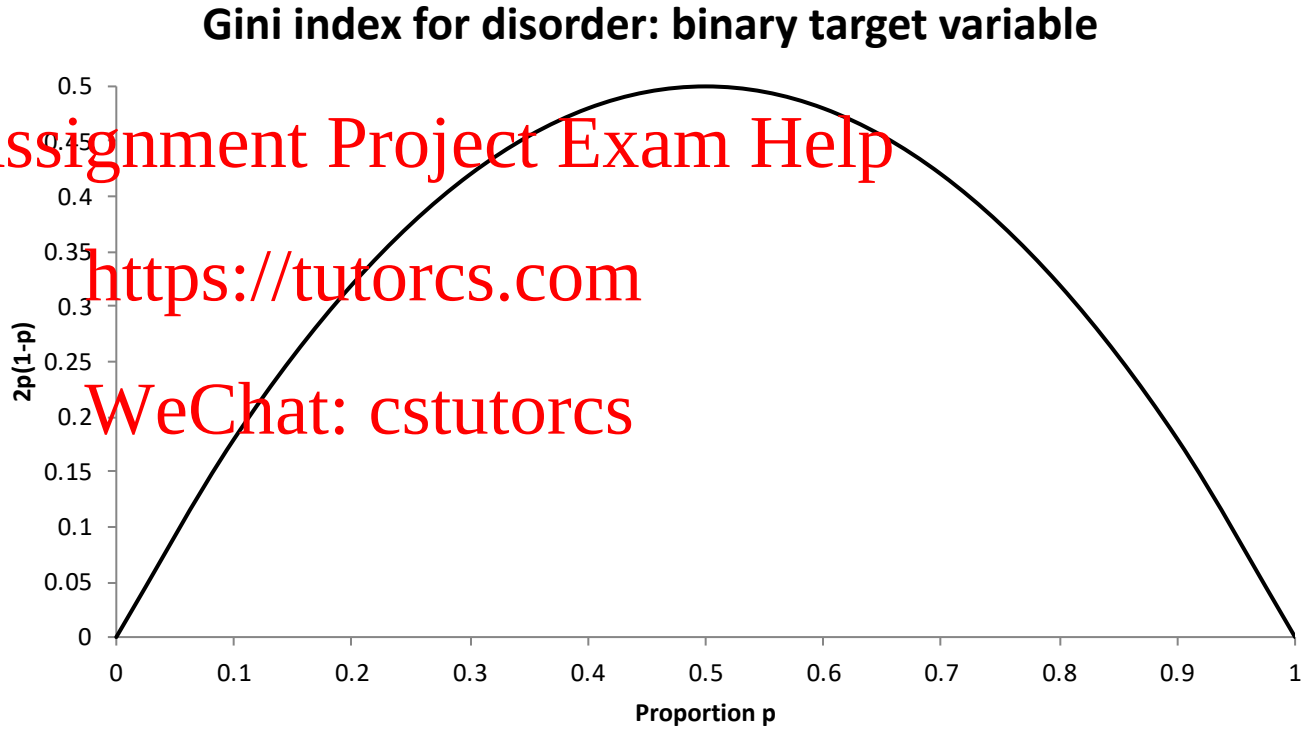
Misclassification index = $\min(p,1-p)$

- are other indexes of disorder (used in r)

Measuring Disorder

The *Gini index for disorder* is a parabola

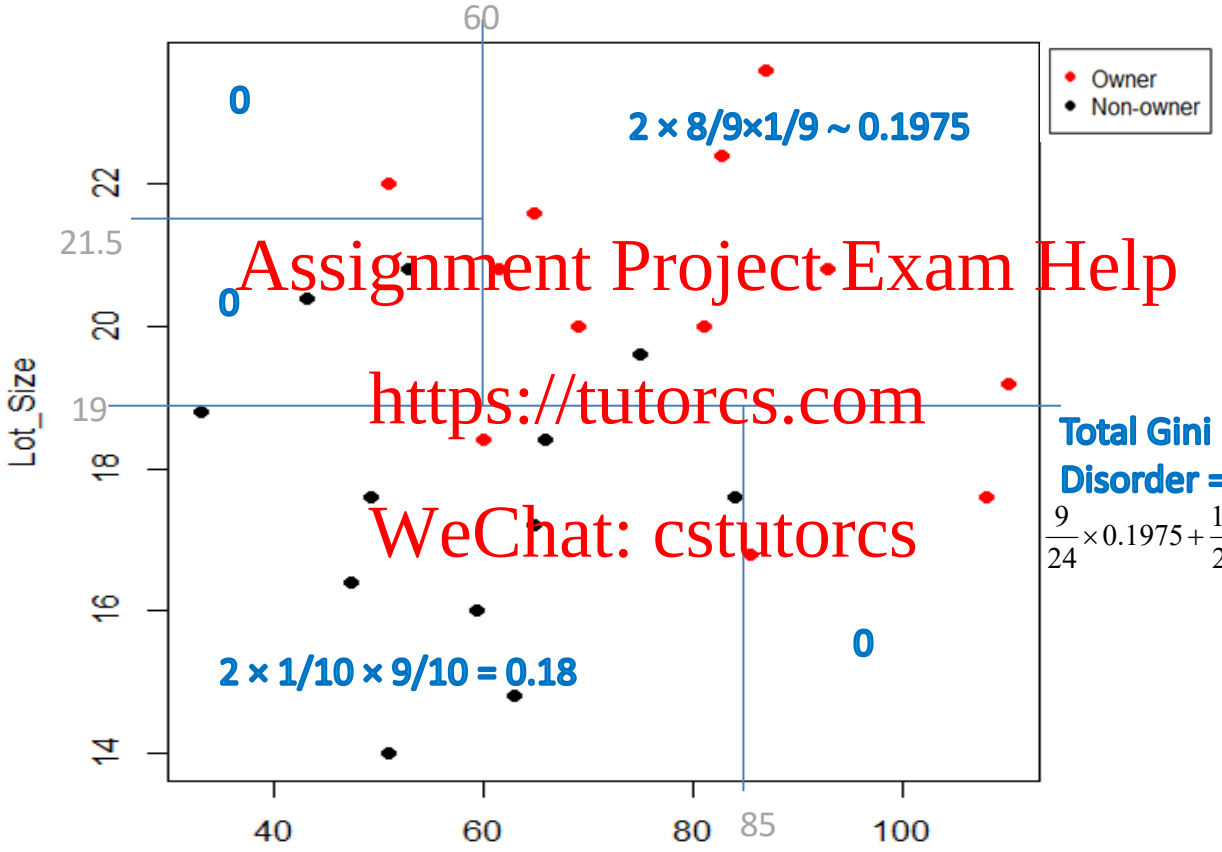
- maximum at $p = 0.5$ and 0 at $p = 0$ or p = 1



Gini Index of Disorder

- To get the Gini index of disorder for the whole tree
  - Calculate the contribution from each leaf as
  - Calculate the weighted sum of Gini indexes for each leaf, weighting by the number of observations in that leaf, as a proportion of all the observations in the sample.
- At each stage, we choose the node and the split at that node to minimise disorder.
- The algorithm is “greedy”: We never undo a partition
- Need to decide when to stop, or go all the way and then prune

Contributions to Disorder Measure



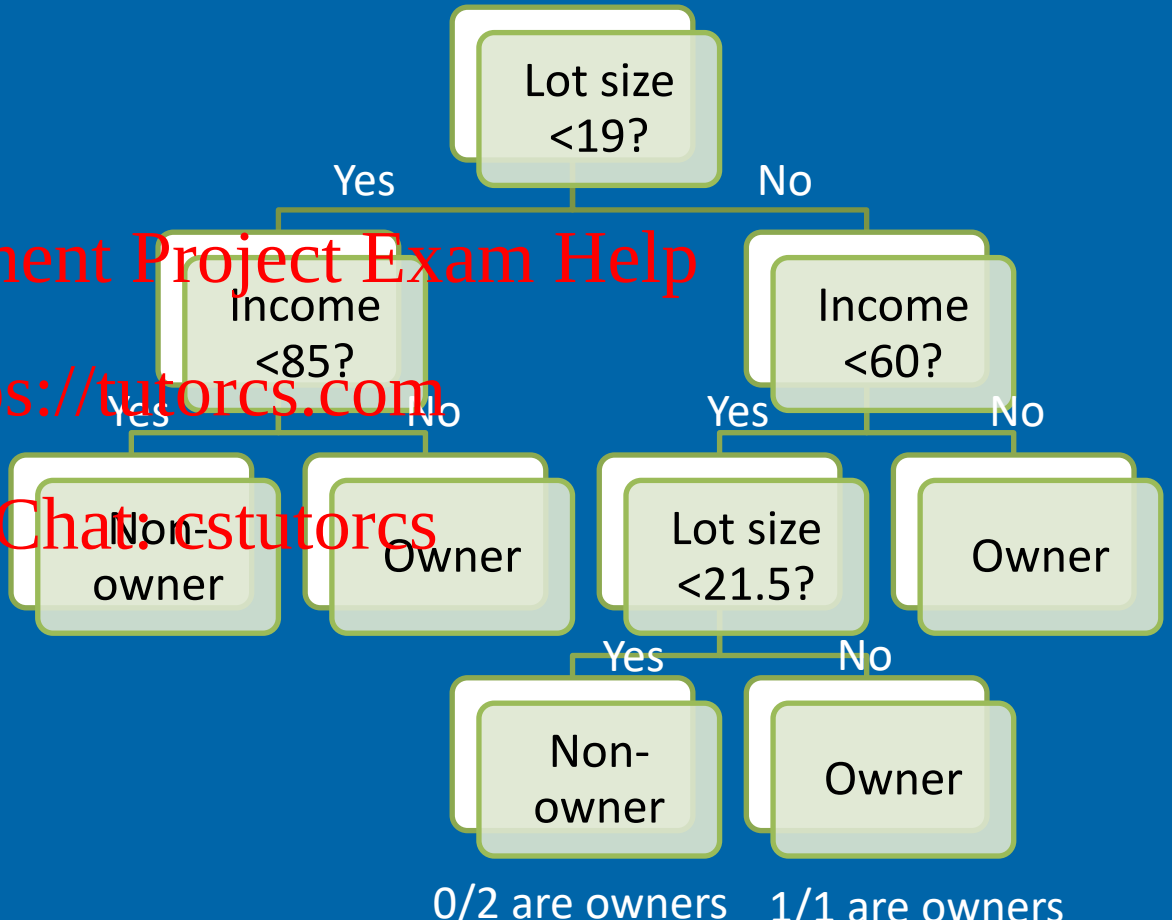
Total Gini measure of Disorder =

$$\frac{9}{24}\times 0.1975+\frac{10}{24}\times 0.18=0.149$$

$$Gini = 2p(1-p)$$

MONASH University

Tree 3



Systematic Classification by Tree

- Picking out by hand classifications that lead to low deviance worked fairly well for this small example
- The computer algorithm can very quickly find at each stage the split that minimises deviance (which variable to split, and where to split it)

Avoiding Overfitting

- We can restrict the size of the tree, or grow a large tree and then prune it back

- These options are investigated in *Workshop 10 Classification trees*

Cross Validation

- Small amount of data
- Wish to decide on optimum tree size
- Divide the data into, say, 4 groups of equal size (more commonly 10.) Call them A, B, C, D
- Leave out one of these four by turn, e.g. leave out A, and prune your tree using the remaining data: B, C, and D.
- Test how well the resulting tree predicts results for A.
- Automated in the command rpart

MONASH University

Assessing the Model: Confusion Matrix

- We can check how reliable the predictions are
  - For the training data used to create the model
  - For test data where we know the value of the target variable but did not use the data in developing the model
- We consider *confusion matrix* and *error rate*

True Positive
TP = number of correct predictions that target = Own

False Negative
FN = number of incorrect predictions that target = Not

False positive
FP = number of incorrect predictions that target = Own

True negative
TN = number of correct predictions that target = Not

Error rate = $$\frac{FP + FN}{TP + FP + TN + FN}$$

Confusion Matrix:

	Predicted	
Actual	Own	Not
Own	TP	FN
Not	FP	TN

Assessing the Model: Confusion Matrix

- How do you expect the error rate to compare for training and test data?
- How do you expect the error rate to compare for eight-leaf tree and four-leaf tree?

True Positive
TP = number of correct predictions that target = Own

False Negative
FN = number of incorrect predictions that target = Not

False positive
FP = number of incorrect predictions that target = Own

True negative
TN = number of correct predictions that target = Not

Error rate = $$\frac{FP + FN}{TP + FP + TN + FN}$$

Confusion Matrix:

	Predicted	
Actual	Own	Not
Own	TP	FN
Not	FP	TN

MONASH University

Assessing the Model: Confusion Matrix

- More generally, the *Error Rate* is the proportion of wrong classifications
- So for example in the following confusion matrix

	Predicted		
Actual	Renting	Mortgage	Own outright
Renting	26	**18**	**7**
Mortgage	**15**	56	**20**
Own outright	**2**	**13**	25

Error rate =

$$\frac{\text{Sum of off-diagonals}}{\text{Sum of all cells}} = \frac{18+7+20+15+2+13}{(18+7+20+15+2+13)+(26+56+25)}$$

$$= \frac{75}{182} = 0.412$$

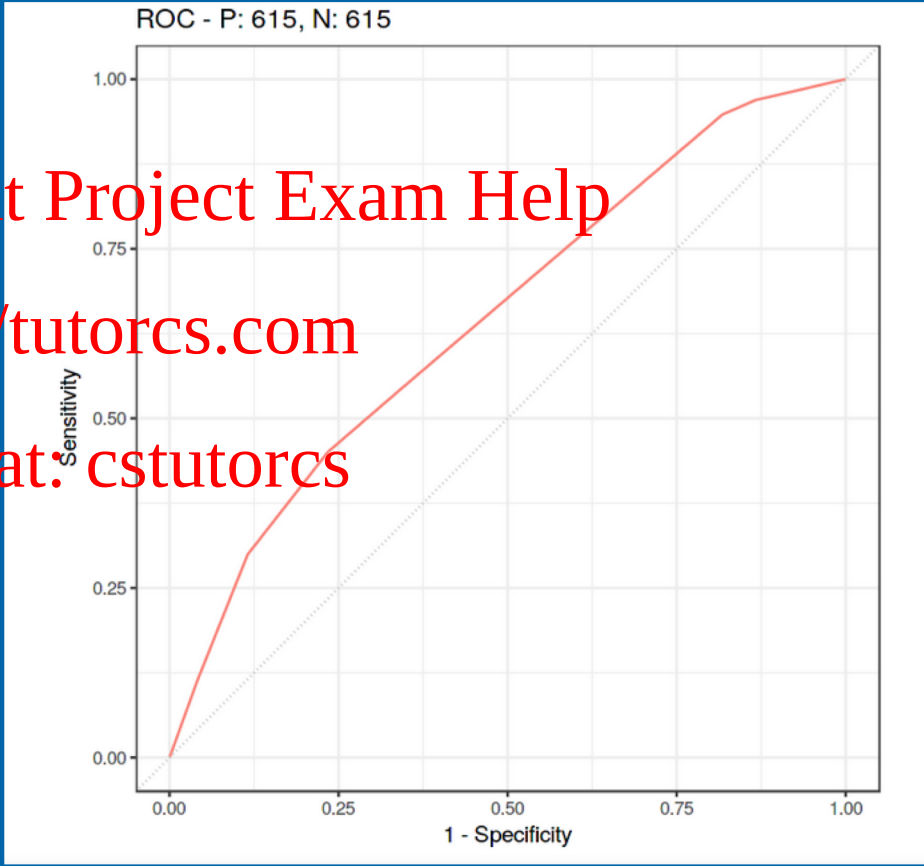
Receiver Operating Characteristic Curve - ROC Curve

- True Positive Rate = $\frac{True\ Positive\ (TP)}{Positive\ (P)}$=TPR
  - Other names: Sensitivity, Recall
- True Negative Rate = $\frac{True\ Negative\ (TN)}{Negative\ (N)}$=TNR
  - Other names: Specificity, Selectivity
- False Positive Rate = $\frac{False\ Positive\ (FP)}{Negative\ (N)}$=FPR = 1-TNR

MONASH University

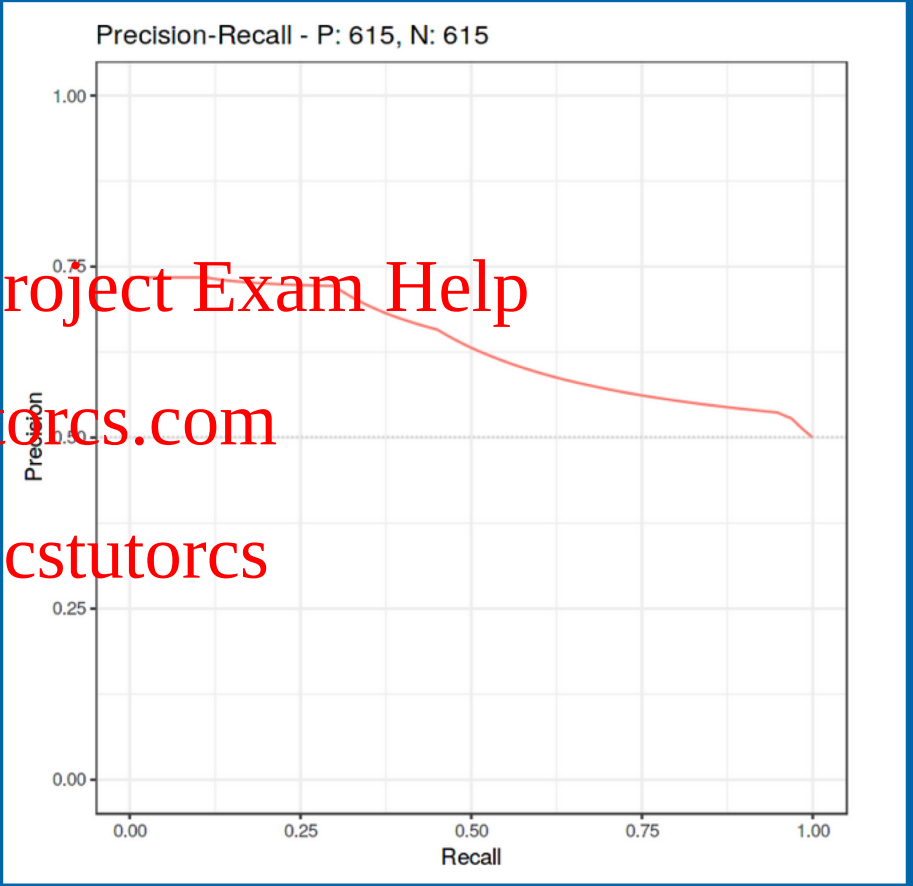
Receiver Operating Characteristic Curve - ROC Curve

AUC: 0.65



MONASH University

Precision-Recall Curve



MONASH University

MONASH BUSINESS SCHOOL

GROUP OF EIGHT AUSTRALIA

AACSB ACCREDITED

EFMD EQUIS ACCREDITED

ASSOCIATION OF MBAS ACCREDITED