

Decision Tree

Wednesday, November 30, 2016 5:54 AM



Class 14

★ Decision Trees ★

© Jigsaw Academy Education Pvt Ltd



DECISION TREES

➔ What is a Decision Tree?

Binary Target Variable

Continuous Target Variable

Terminology

Data Preparation

© Jigsaw Academy Education Pvt Ltd



What is a Decision Tree?

Example: Credit Card Company

1. Unprofitable customers –
do not use credit cards frequently / use card but pay on time
2. Profitable customers –
do not make payment in full (carry balances on card) or on time



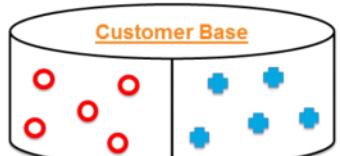
© Jigsaw Academy Education Pvt Ltd



What is a Decision Tree?

Example: Credit Card Company

1. Unprofitable customers –
do not use credit cards frequently / use card but pay on time
2. Profitable customers –
do not make payment in full (carry balances on card) or on time



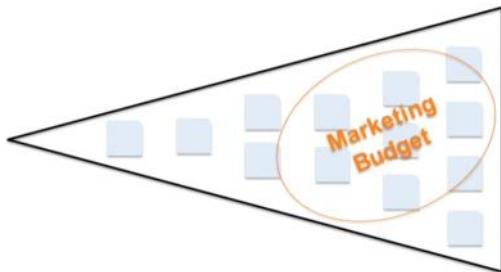
© Jigsaw Academy Education Pvt Ltd



Make the most of it's marketing budget:

What is a Decision Tree?

Potential Customers



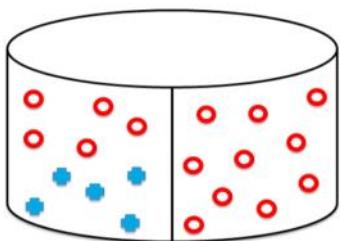
© Jigsaw Academy Education Pvt Ltd



What is a Decision Tree?

Existing Customers

New Customers



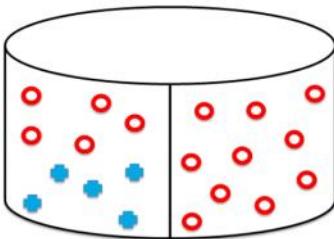
© Jigsaw Academy Education Pvt Ltd



Focus of attracting profitable customers.

What is a Decision Tree?

Existing Customers



New Customers

1. Age
2. Gender
3. Marital Status
4. # credit cards owned

© Jigsaw Academy Education Pvt Ltd



How can we use these variables be used to predict the profitability of these customers.

What is a Decision Tree?

Existing Customers



Customer	Age	Gender	Marital Status	# Cr. Cards	Profitability
1	36	M	M	1	P
2	32	M	S	3	U
3	38	M	M	2	P
4	40	M	S	1	U
5	44	M	M	0	P
6	56	F	M	0	P
7	58	F	S	1	U
8	30	F	S	2	P
9	28	F	M	1	U
10	26	F	M	0	U

© Jigsaw Academy Education Pvt Ltd



Tag the customers as profitable and unprofitable.

What is a Decision Tree?

Existing Customers

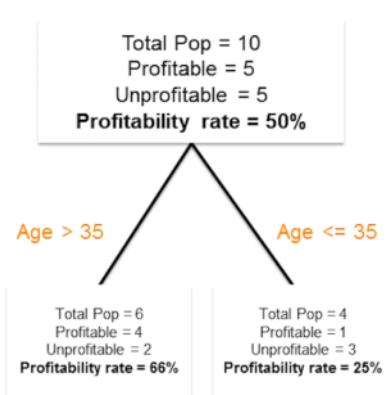
Customer	Age	Gender	Marital Status	# Cr. Cards	Profitability
1	36	M	M	1	P
2	32	M	S	3	U
3	38	M	M	2	P
4	40	M	S	1	U
5	44	M	M	0	P
6	56	F	M	0	P
7	58	F	S	1	U
8	30	F	S	2	P
9	28	F	M	1	U
10	26	F	M	0	U

Total Pop = 10
Profitable = 5
Unprofitable = 5
Profitability rate = 50%

© Jigsaw Academy Education Pvt Ltd



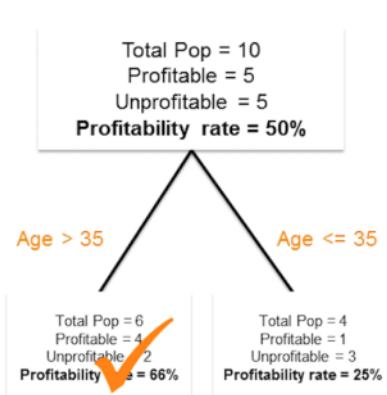
User	Marital Status	# Cr. Cards	Profitability
M	1	P	
S	3	U	
M	2	P	
S	1	U	
M	0	P	
M	0	P	
S	1	U	
S	2	P	
M	1	U	
M	0	U	



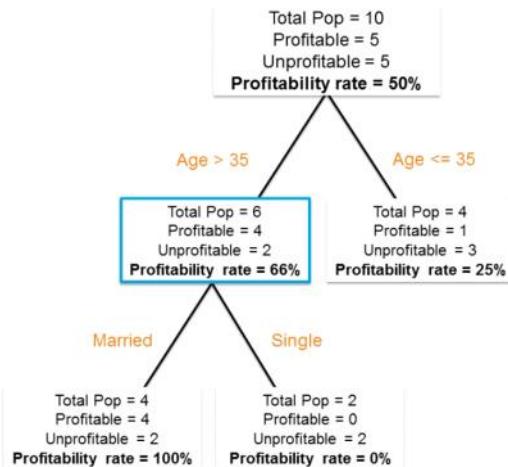
© Jigsaw Academy Education Pvt Ltd



User	Marital Status	# Cr. Cards	Profitability
M	1	P	
S	3	U	
M	2	P	
S	1	U	
M	0	P	
M	0	P	
S	1	U	
S	2	P	
M	1	U	
M	0	U	



© Jigsaw Academy Education Pvt Ltd



© Jigsaw Academy Education Pvt Ltd



What is a Decision Tree?

4 variables, 10 records

1. Age
2. Gender
3. Marital Status
4. # Credit Cards

- Why Age?
- Why split at 35?
- What if there are hundreds of variables and millions of records?

© Jigsaw Academy Education Pvt Ltd



Decision Trees

- Gini
- Entropy
- Chi Square
- Reduction of Variance

© Jigsaw Academy Education Pvt Ltd



DECISION TREES

DATA
SCIENCE
WITH R

What is a Decision Tree?

➔ **Binary Target Variable**

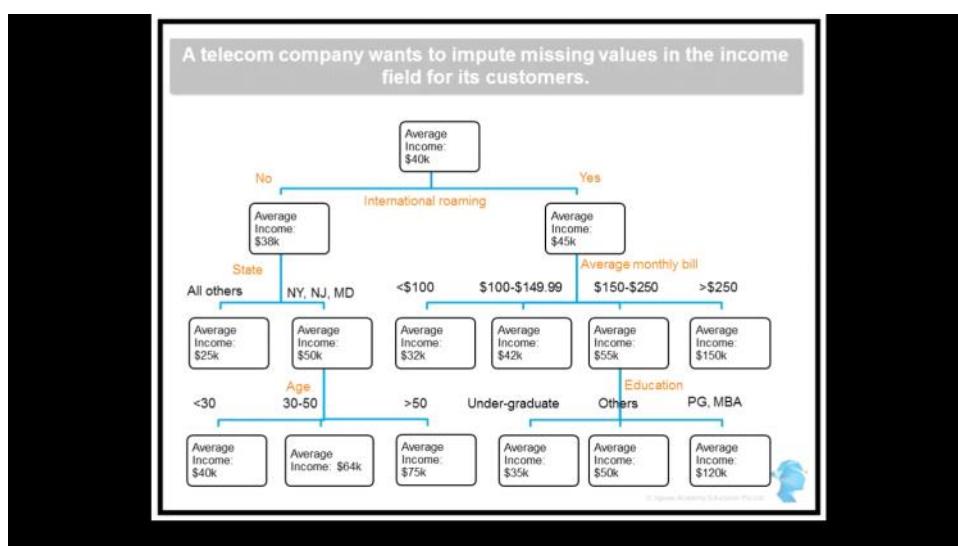
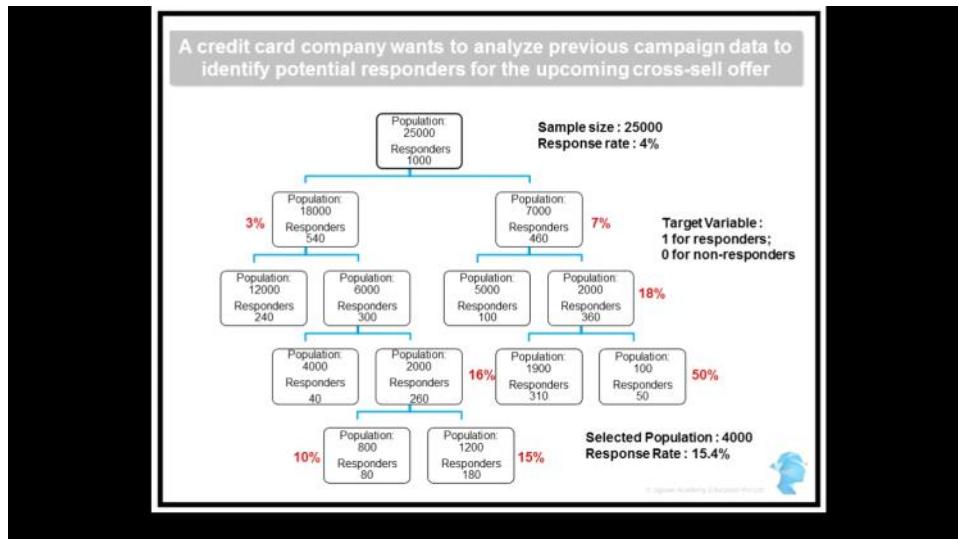
Continuous Target Variable

Terminology

Data Preparation

© Jigsaw Academy Education Pvt Ltd





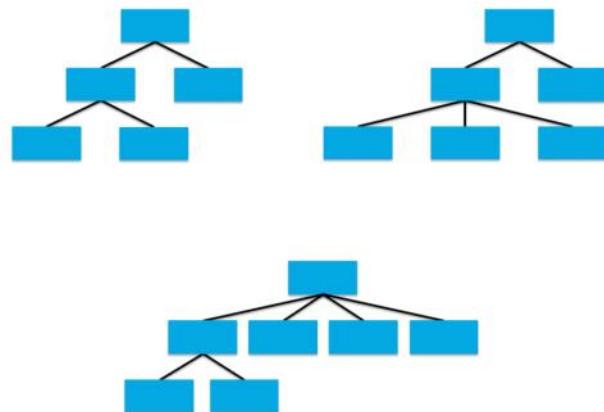
Each split in the tree is chosen to decrease the variance in the values of the target variable within each child node

© Jigsaw Academy Education Pvt Ltd

Continuous Target Variable

1. A decision tree is more suited to the prediction of discrete or categorical variables
2. To estimate a continuous variable, use a continuous function
3. Regression and neural nets are usually better for estimation of continuous variables

© Jigsaw Academy Education Pvt Ltd



© Jigsaw Academy Education Pvt Ltd

DECISION TREES

What is a Decision Tree?

Binary Target Variable

Continuous Target Variable



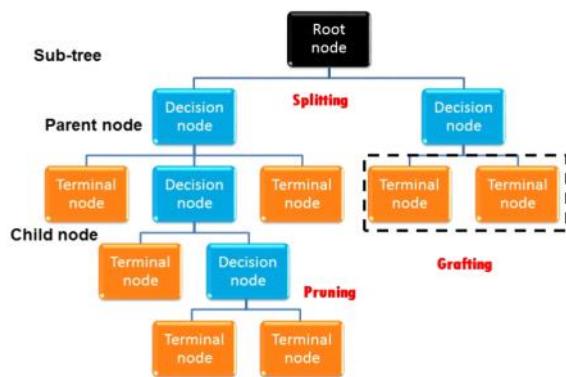
Terminology

Data Preparation



© Jigsaw Academy Education Pvt Ltd

Decision Tree Terminology



DECISION TREES

What is a Decision Tree?

Binary Target Variable

Continuous Target Variable

Terminology

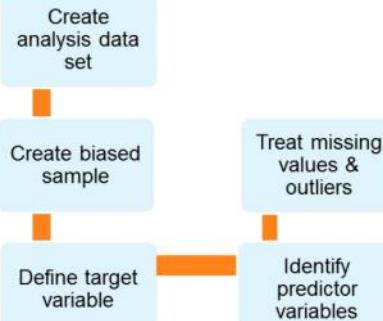


Data Preparation



© Jigsaw Academy Education Pvt Ltd

Data Preparation



© Jigsaw Academy & Education Pvt Ltd



DATA
SCIENCE
WITH R

Decision Trees

★ R Code Demo Part 1 ★

© Jigsaw Academy & Education Pvt Ltd



- Find the split**
- Identify all possible split options
 - Choose the best split

- Grow the tree**
- Continue growing tree till it meets a criterion
 - Grow the tree large and prune it

- Prune the tree**
- Create candidate sub-trees
 - Identify best candidate
 - Stop tree using a size based constraint

- Extract rules**
- Generate rule set corresponding to the tree
 - Simplify

© Jigsaw Academy & Education Pvt Ltd



Finding the Split



Training Data

3 variables

Male or Female

Low Income / Medium Income / High Income

Karnataka or UP or Tamil Nadu. etc

$$2 \times 3 \times 25 = 150 \text{ options}$$

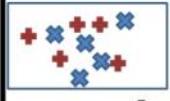
© Jigsaw Academy Education Pvt Ltd



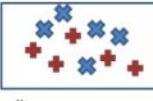
Finding the Split



Training Data



Poor split



Poor split

Good split



Perfect split?

© Jigsaw Academy Education Pvt Ltd



Effectiveness of Split

1. Categorical target variable

- Gini
- Chi-square
- Information gain

2. Continuous target variable

- Reduction in variance

3. Other methods

- Information gain ratio
- F test

© Jigsaw Academy Education Pvt Ltd



Gini

Gini coefficient = $(p(\text{red}))^2 + (p(\text{blue}))^2$
Where $p(\text{red})$ is proportion of reds in the data

of reds = 10
of blues = 10
Proportion of reds = .5
Proportion of blues = .5
Gini = $.5^2 + .5^2 = .5$

# of reds = 2 # of blues = 0 Proportion of reds = 1 Proportion of blues = 0 Gini = $1^2 + 0^2 = 1$	# of reds = 8 # of blues = 10 Proportion of reds = .45 Proportion of blues = .55 Gini = $.45^2 + .55^2 = .505$	# of reds = 8 # of blues = 2 Proportion of reds = .8 Proportion of blues = .2 Gini = $.8^2 + .2^2 = .68$	# of reds = 2 # of blues = 8 Proportion of reds = .2 Proportion of blues = .8 Gini = $.2^2 + .8^2 = .68$
--	--	--	--

Gini score for the split
 $(1*2/20) + (.505*18/20) = .55$

Gini score for the split
 $(.68*10/20) + (.68*10/20) = .68$

Gini

Gini coefficient = $(p(\text{red}))^2 + (p(\text{blue}))^2$

of reds = 10
of blues = 10
Proportion of reds = .5
Proportion of blues = .5
Gini = $.5^2 + .5^2 = .5$

Where $p(\text{red})$ is proportion of reds in the data

of reds = 2
of blues = 0
Proportion of reds = 1
Proportion of blues = 0
Gini = $1^2 + 0^2 = 1$

of reds = 8
of blues = 10
Proportion of reds = .45
Proportion of blues = .55
Gini = $.45^2 + .55^2 = .505$



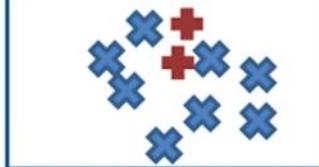
of reds = 2
of blues = 0
Proportion of reds = 1
Proportion of blues = 0
Gini = $1^2 + 0^2 = 1$

of reds = 8
of blues = 10
Proportion of reds = .45
Proportion of blues = .55
Gini = $.45^2 + .55^2 = .505$

Gini score for the split
 $(1*2/20) + (.505*18/20) = .55$



Proportion of blues = .5
Gini = $.5^2 + .5^2 = .5$



of reds = 8
of blues = 2
Proportion of reds = .8
Proportion of blues = .2
Gini = $.8^2 + .2^2 = .68$

of reds = 2
of blues = 8
Proportion of reds = .2
Proportion of blues = .8
Gini = $.2^2 + .8^2 = .68$



of reds = 8
of blues = 2
Proportion of reds = .8
Proportion of blues = .2
Gini = $.8^2 + .2^2 = .68$

of reds = 2
of blues = 8
Proportion of reds = .2
Proportion of blues = .8
Gini = $.2^2 + .8^2 = .68$

Gini score for the split
 $(.68*10/20) + (.68*10/20) = .68$

Gini

Press Esc to exit full screen mode.

Gini coefficient = $(p(\text{red}))^2 + (p(\text{blue}))^2$
Where $p(\text{red})$ is proportion of reds in the data

of reds = 10
of blues = 10
Proportion of reds = .5
Proportion of blues = .5
 $\text{Gini} = .5^2 + .5^2 = .5$

# of reds = 2 # of blues = 0 Proportion of reds = 1 Proportion of blues = 0 $\text{Gini} = 1^2 + 0^2 = 1$	# of reds = 8 # of blues = 10 Proportion of reds = .45 Proportion of blues = .55 $\text{Gini} = .45^2 + .55^2 = .505$	# of reds = 8 # of blues = 2 Proportion of reds = .8 Proportion of blues = .2 $\text{Gini} = .8^2 + .2^2 = .68$	# of reds = 2 # of blues = 8 Proportion of reds = .2 Proportion of blues = .8 $\text{Gini} = .2^2 + .8^2 = .68$
---	---	---	---

Gini score for the split
 $(1*2/20) + (.505*18/20) = .55$

Gini score for the split
 $(.68*10/20) + (.68*10/20) = .68$ ✓

© Jigsaw Academy & Education Pvt Ltd

DATA SCIENCE WITH R

DECISION TREES

Creating Decision Trees

Gini

Chi Square

Information Gain

Reduction in Variance

© Jigsaw Academy & Education Pvt Ltd

of reds = 10
of blues = 10
Proportion of reds = .5
Proportion of blues = .5

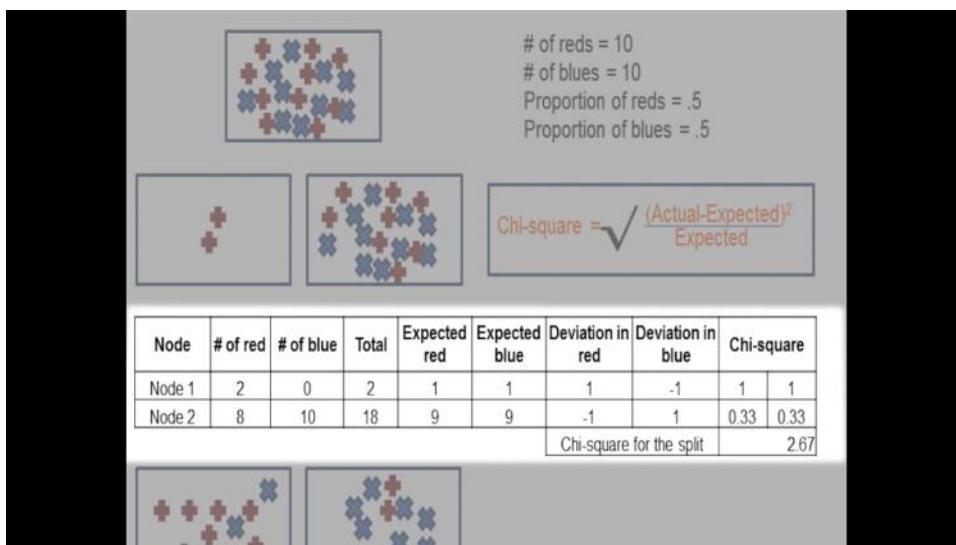
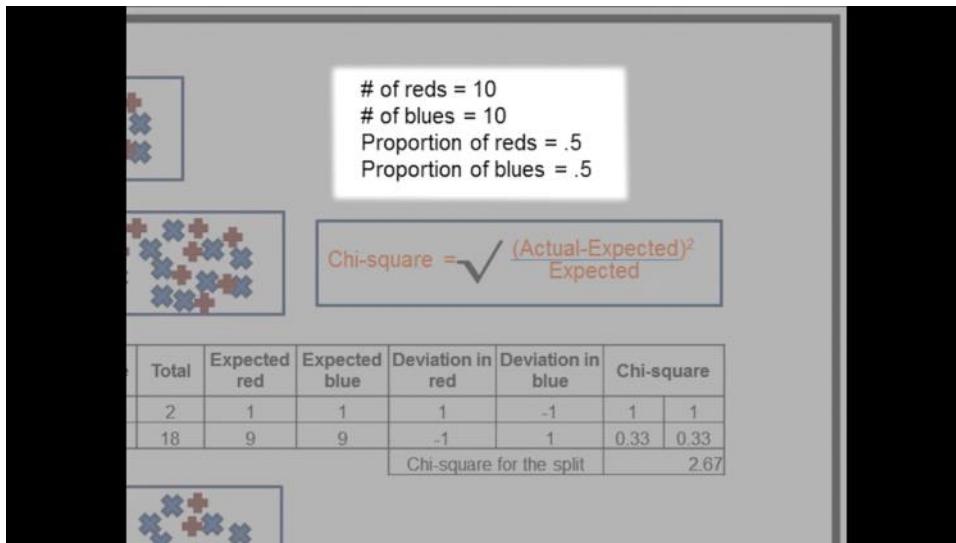
Chi-square = $\sqrt{\frac{(\text{Actual}-\text{Expected})^2}{\text{Expected}}}$

Node	# of red	# of blue	Total	Expected red	Expected blue	Deviation in red	Deviation in blue	Chi-square
Node 1	2	0	2	1	1	1	-1	1
Node 2	8	10	18	9	9	-1	1	0.33
								0.33
								Chi-square for the split
								2.67

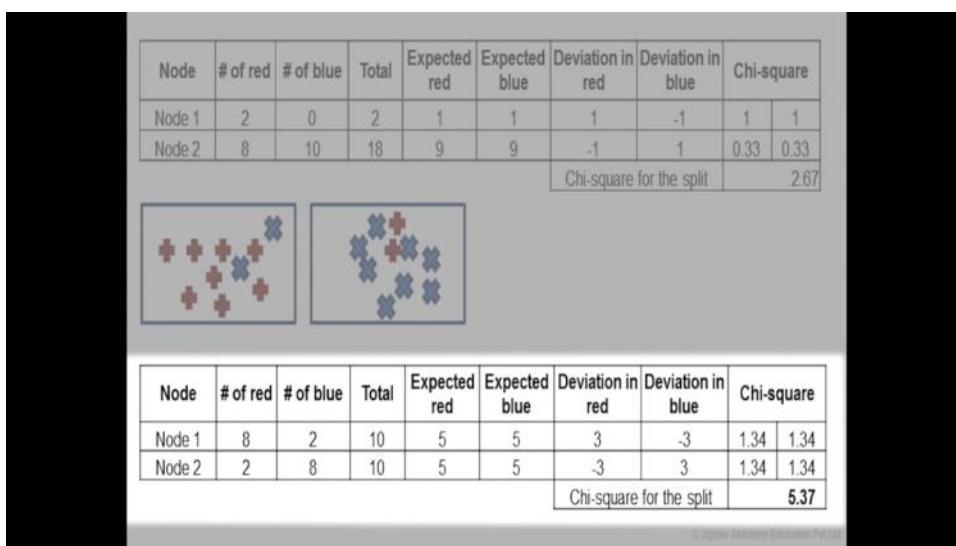
Node	# of red	# of blue	Total	Expected red	Expected blue	Deviation in red	Deviation in blue	Chi-square
Node 1	8	2	10	5	5	3	-3	1.34
Node 2	2	8	10	5	5	-3	3	1.34
								Chi-square for the split
								5.37

© Jigsaw Academy & Education Pvt Ltd

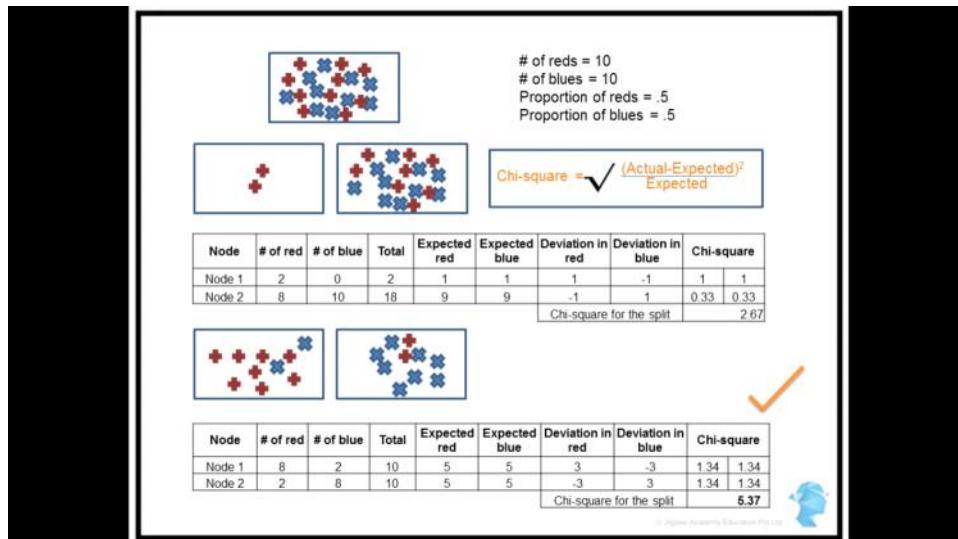
Higher values of chi square means that the variance is more significant and not by mere chance



The chi square for the split is the sum of all the chi squares calculated.



High score means a successful split with significantly population distribution.



DECISION TREES

Creating Decision Trees

Gini

Chi Square

Information Gain

Reduction in Variance

© Jigsaw Academy Education Pvt Ltd

DATA SCIENCE WITH R

Information Gain

Basic Logarithmic Functions

Logarithm or log of a number is the exponent or the power by which a fixed number, the base, has to be raised to produce that number.

Examples:

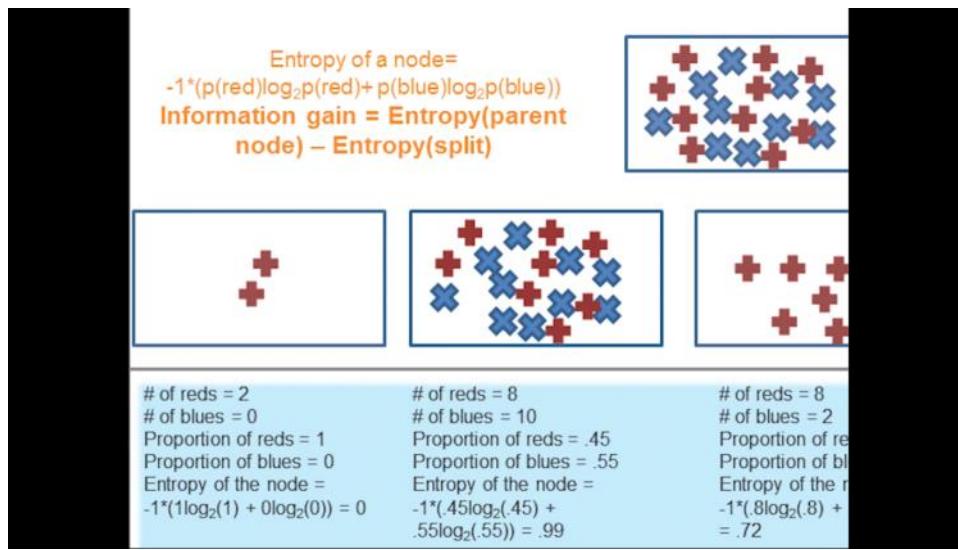
What is $\log 1000$ to the base 10?

$1000 = 10 * 10 * 10 = 10^3$

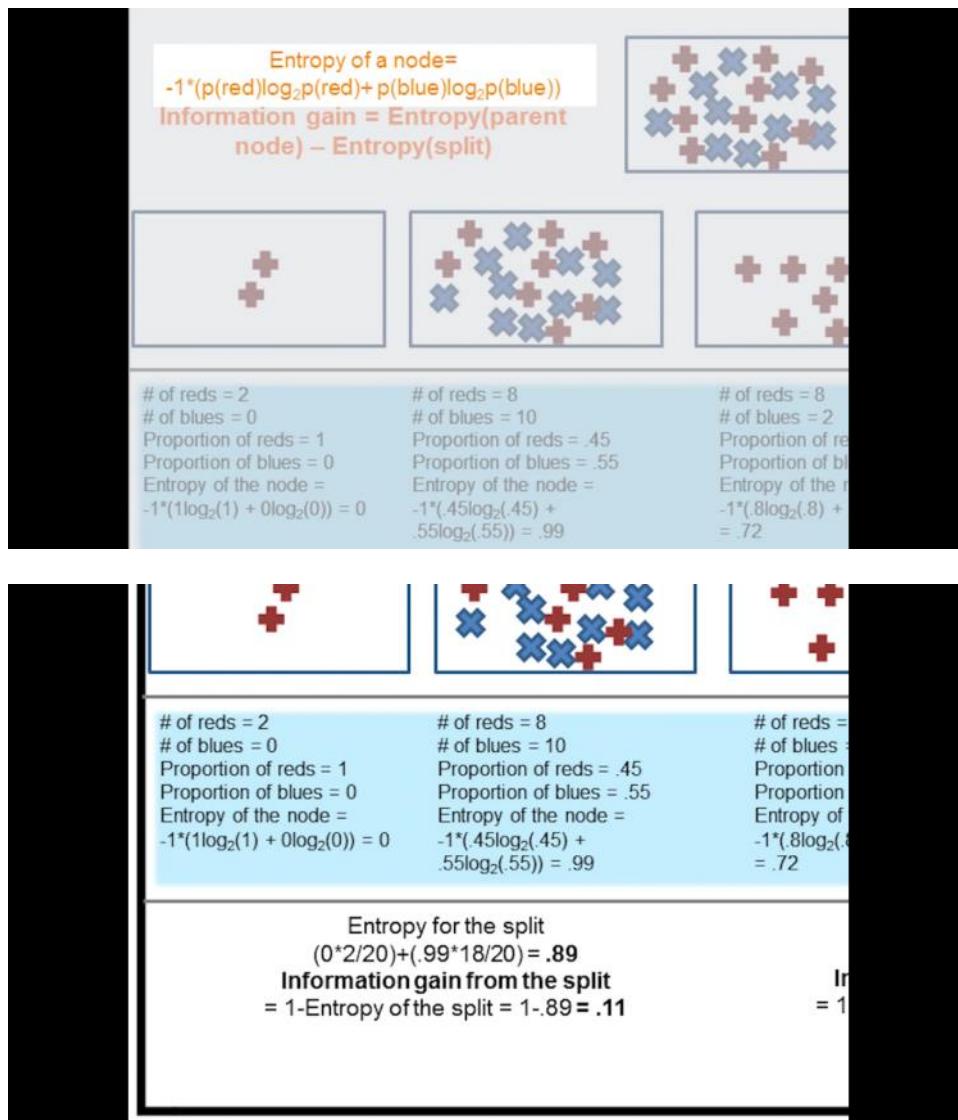
What is $\log 16$ to the base 2?

$16 = 2 * 2 * 2 * 2 = 2^4$

© Jigsaw Academy Education Pvt Ltd



Information gain is the same as entropy reduction. Both are used for decision tree algorithm



+ + + +

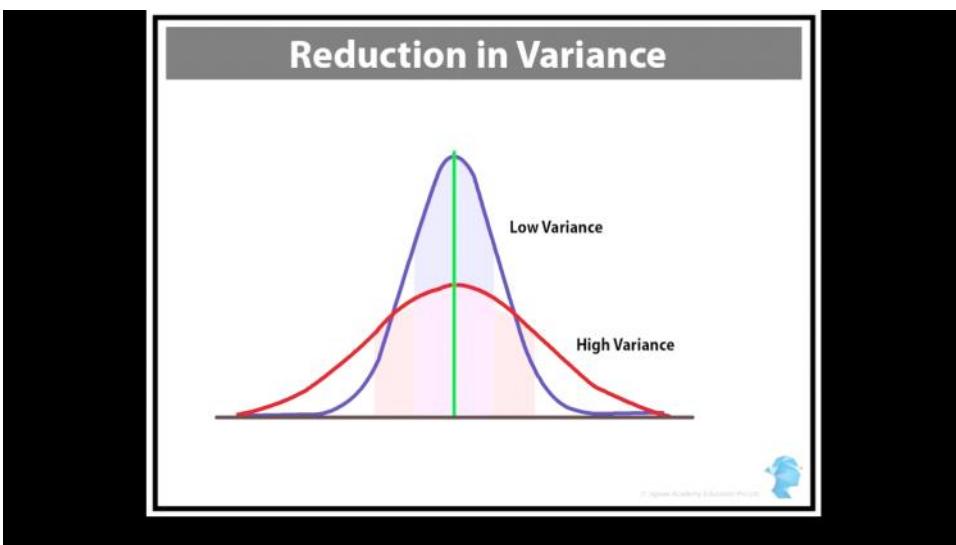
X X X X

# of reds = 8 # of blues = 2 Proportion of reds = .8 Proportion of blues = .2 Entropy of the node = $-1*(.8\log_2(.8) + .2\log_2(.2))$ = .72	# of reds = 2 # of blues = 8 Proportion of reds = .2 Proportion of blues = .8 Entropy of the node = $-1*(.2\log_2(.2) + .8\log_2(.8))$ = .72
--	--

Entropy for the split
 $(.72*10/20) + (.72*10/20) = .72$

Information gain from the split
 $= 1 - \text{Entropy of the split} = 1 - .72 = .28$

© Jigsaw Academy Education Pvt Ltd



Sigma square is variance:

DATA SCIENCE WITH R

$$\sigma^2 = \frac{\sum (X - \mu)^2}{N}$$

© Jigsaw Academy Education Pvt Ltd

Reduction in Variance

Consider
red = 1; blue = 0

of reds = 10
of blues = 10
Expected value = .5
Variance = $(20 * (.5)^2)/20 = .25$

of reds = 2
of blues = 0
Expected value = 1
Variance = $2 * 0 = 0$

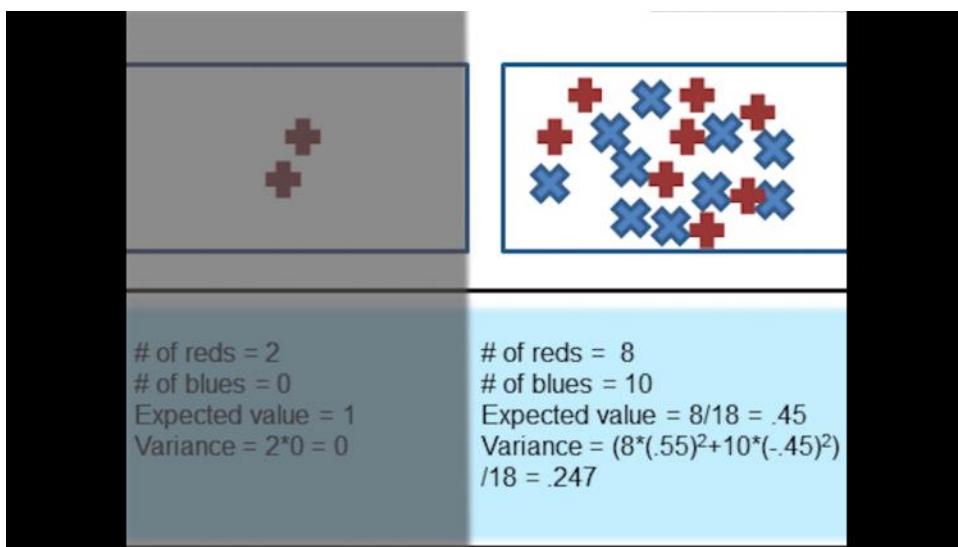
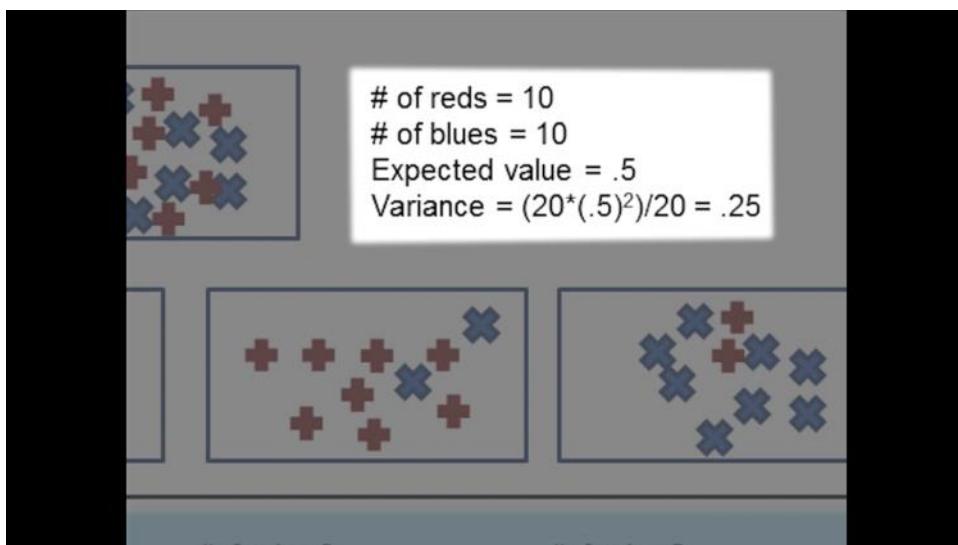
of reds = 8
of blues = 10
Expected value = $8/18 = .45$
Variance = $(8 * (.55)^2 + 10 * (-.45)^2) / 18 = .247$

of reds = 8
of blues = 2
Expected value = $8/10 = .8$
Variance = $(8 * (.2)^2 + 2 * (-.8)^2) / 10 = .16$

Variance for the split
 $(0^2/20) + (.247 * 18/20) = .22$
Reduction in variance = $.25 - .22 = .03$

Variance for the split
 $(.16 * 10/20) + (.16 * 10/20) = .16$
Reduction in variance = $.25 - .16 = .09$

© Jigsaw Academy & Education Project



of reds = 2
of blues = 0
Expected value = 1
Variance = $2 \cdot 0 = 0$

of reds = 8
of blues = 10
Expected value = $8/18 = .45$
Variance = $(8 \cdot (.55)^2 + 10 \cdot (-.45)^2) / 18 = .247$

Variance for the split
 $(0 \cdot 2/20) + (.247 \cdot 18/20) = .22$
Reduction in variance = $.25 - .22 = .03$

of reds = 8
of blues = 2
Expected value = $8/10 = .8$
Variance = $(8 \cdot (.2)^2 + 2 \cdot (-.8)^2) / 10 = .16$

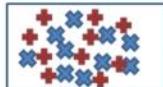
of reds = 2
of blues = 8
Expected value = $2/10 = .2$
Variance = $(2 \cdot (.2)^2 + 8 \cdot (-.2)^2) / 10 = .16$

Variance for the split
 $(.16 \cdot 10/20) + (.16 \cdot 10/20) = .16$
Reduction in variance = $.25 - .16 = .09$

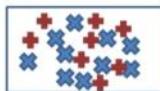
© Jigsaw Academy Education Pvt Ltd

Reduction in Variance

Consider
red = 1; blue = 0



of reds = 10
of blues = 10
Expected value = .5
Variance = $(20 \cdot (.5)^2) / 20 = .25$



of reds = 2
of blues = 0
Expected value = 1
Variance = $2 \cdot 0 = 0$

of reds = 8
of blues = 10
Expected value = $8/18 = .45$
Variance = $(8 \cdot (.55)^2 + 10 \cdot (-.45)^2) / 18 = .247$

of reds = 2
of blues = 2
Expected value = $2/10 = .2$
Variance = $(2 \cdot (.2)^2 + 2 \cdot (-.2)^2) / 10 = .16$

of reds = 2
of blues = 8
Expected value = $2/10 = .2$
Variance = $(2 \cdot (.2)^2 + 8 \cdot (-.2)^2) / 10 = .16$

Variance for the split
 $(0 \cdot 2/20) + (.247 \cdot 18/20) = .22$
Reduction in variance = $.25 - .22 = .03$

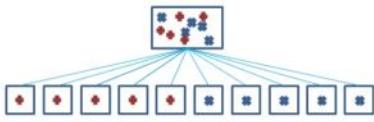
Variance for the split
 $(.16 \cdot 10/20) + (.16 \cdot 10/20) = .16$
Reduction in variance = $.25 - .16 = .09$

© Jigsaw Academy Education Pvt Ltd

Overfitting and Tree Pruning

Overfitting:

Greed for accuracy leading to over-complex trees



Pruning algorithms:

- **CART** – Prunes the tree by imposing a complexity penalty based on number of leaves in the tree.
- **C5** – Assumes a higher rate of error than what is seen on the training data. The smaller the node, the more the increase over observed. When the child node estimate is higher than the parent node, tree is pruned.

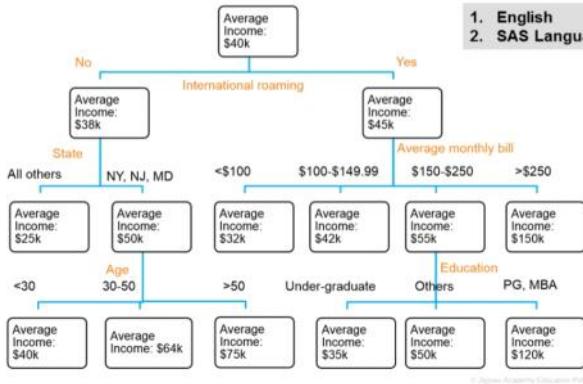
Manual pruning:

- Analyst needs to study the tree in detail. Any node that looks unstable should be pruned.

© Jigsaw Academy Education Pvt Ltd

Decision Tree Rules

A telecom company wants to impute missing values in the income field for its customers.



© Jigsaw Academy Education Pvt Ltd

DATA SCIENCE
WITH R

DECISION TREES

Application of Techniques

Uses of Decision Trees

Advantages and Disadvantages

What makes a Good Tree?

Widely used Software

© Jigsaw Academy Education Pvt Ltd

Application of Techniques

1. **Classification and Regression trees** (CART) algorithm uses the Gini method to create binary splits
2. **CHAID**, another popular algorithm uses the Chi-square test to produce multi-splits
3. **Gini** method is used in sociology and other 'noisy' domains
4. **Reduction in variance** algorithm used in regression trees

© Jigsaw Academy Education Pvt Ltd



Thumbrules:

- Binary target --> cart
- Multiple splits at each level(tree) --> CHAID
- Lot of noise --> CART
- F test --> for numeric target variables i.e. reduction in variance test

DATA
SCIENCE
WITH R

DECISION TREES

➔ Application of Techniques

Uses of Decision Trees

Advantages and Disadvantages

What Makes a Good Tree?

Widely used Software



Case Study: Store Card Fraud

JC Penney, a US based retailer, received 90000 credit applications in 2007. Of the 75000 approved, 940 were later found to be fraudulent.

Objective:

- To create a set of rules for the credit decisioning engine which could flag potentially fraud applications for further checks.



Business Constraint:

- The retailer wanted to ensure a false positive rate of less than .5 to minimize customer discomfort

© Jigsaw Academy Education Pvt Ltd



Case Study: Store Card Fraud

Methodology:

- A random set of 8460 non-fraudulent credit applications was combined with the 940 valid fraudulent applications
- Biased sample set created with 10% fraud rate
- Data partitioned into test (70%) and validation (30%)
- Multiple decision trees created using different algorithms
- Focus on maximizing fraud detection and minimize false positives

© Jigsaw Academy Education Pvt Ltd



Case Study: Store Card Fraud

Methodology:

- Final tree selected based on the performance on validation data set
- Rules extracted and simplified. Converted to SQL
- Rules implemented on the client engine

Results:

Model performance was tracked over the next 3 months. Out of 17340 applications coming in, the model flagged 585 applications of which 320 were later denied credit. Fraud rate for JC Penney came down from 1.25% to .7%.

© Jigsaw Academy Education Pvt Ltd

DATA
SCIENCE
WITH R

DECISION TREES

Application of Techniques

➔ **Uses of Decision Trees**

Advantages and Disadvantages

What Makes a Good Tree?

Widely used Software

© Jigsaw Academy Education Pvt Ltd



Uses of Decision Trees

- Variable selection
- Interaction detection
- Missing value imputation
- Model interpretation
- Predictive modeling

© Jigsaw Academy Education Pvt Ltd



DATA
SCIENCE
WITH R

DECISION TREES

Application of Techniques

Uses of Decision Trees



Advantages and Disadvantages

What makes a Good Tree?

Widely used Software

© Jigsaw Academy Education Pvt Ltd



Advantages

- ✓ Easy to use and understand
- ✓ Graphical representation aids visual understanding
- ✓ Require less data preparation than other methods
- ✓ Non parametric method

© Jigsaw Academy Education Pvt Ltd



Advantages

- ✓ Catch interactions and non-additive behavior between variables
- ✓ Result in form of rules. Easy to interpret.
- ✓ Deal well with outliers and missing values
- ✓ Combine well with other modeling techniques

© Jigsaw Academy Education Pvt Ltd



Disadvantages

1. Tendency to over-fit
2. Do not make use of information in data as well as neural networks or regression
3. May result in too many splits on variables with many values
4. Time series data requires extensive preparation

© Jigsaw Academy Education Pvt Ltd

DATA
SCIENCE
WITH R

DECISION TREES

Application of Techniques

Uses of Decision Trees

Advantages and Disadvantages



What Makes a Good Tree?

Widely used Software

© Jigsaw Academy Education Pvt Ltd



What Makes a Good Tree?

Most desirable trees are not necessarily the ones that do the best on the training data

Features of a good tree:

- Accurate
- Use few variables
- Easy to visualize and interpret
- Make sense intuitively

© Jigsaw Academy Education Pvt Ltd



When To Use?

- Very popular classification technique especially when problem is binary
- Not the first choice for estimating continuous values
- Best option when the interpretation of results is more important than the accuracy
- Use when time is of the essence
- Whenever available, use for initial data exploration in any project

© Jigsaw Academy Education Pvt Ltd

DATA
SCIENCE
WITH R

DECISION TREES

Application of Techniques

Uses of Decision Trees

Advantages and Disadvantages

What Makes a Good Tree?



Widely used Software

© Jigsaw Academy Education Pvt Ltd



Widely used Software

Features

- All popular algorithms including CHAID, CART, Gini, Variance, Information gain.
- Handle both categorical and numerical variables.
- Interactive with custom splits, pruning and growing of trees.
- Trees can be translated into base-set of English rules or programming statements such as SAS, SQL, Java



© Jigsaw Academy Education Pvt Ltd

Recap

- Helps you choose from multiple courses of action
- Provide highly effective structure
- Assist in forming a balanced picture of risks / rewards of each action
- Represent rules
- Used to explore data to gain insights
- Powerful first step in modeling process

© Jigsaw Academy Education Pvt Ltd

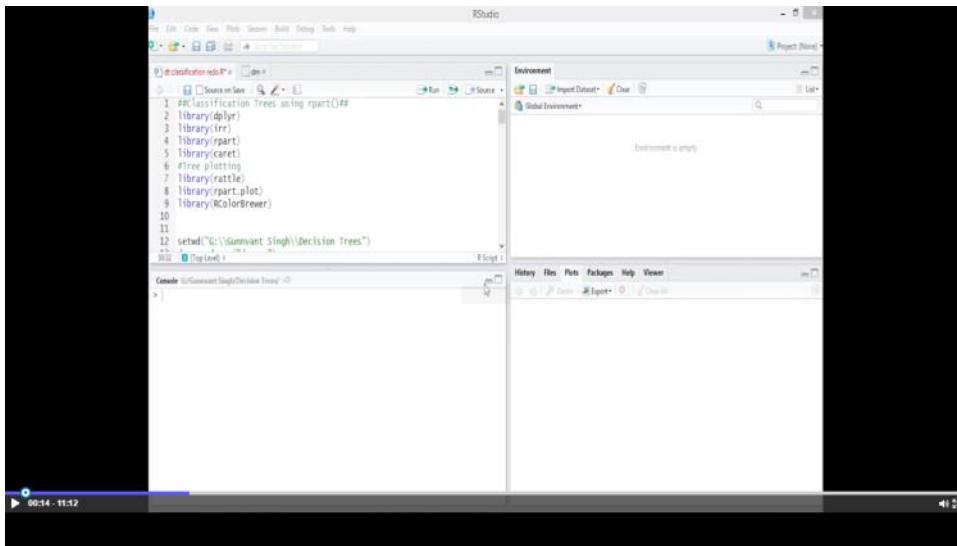
- Data mining techniques by Berry and Linoff

DATA
SCIENCE
WITH R

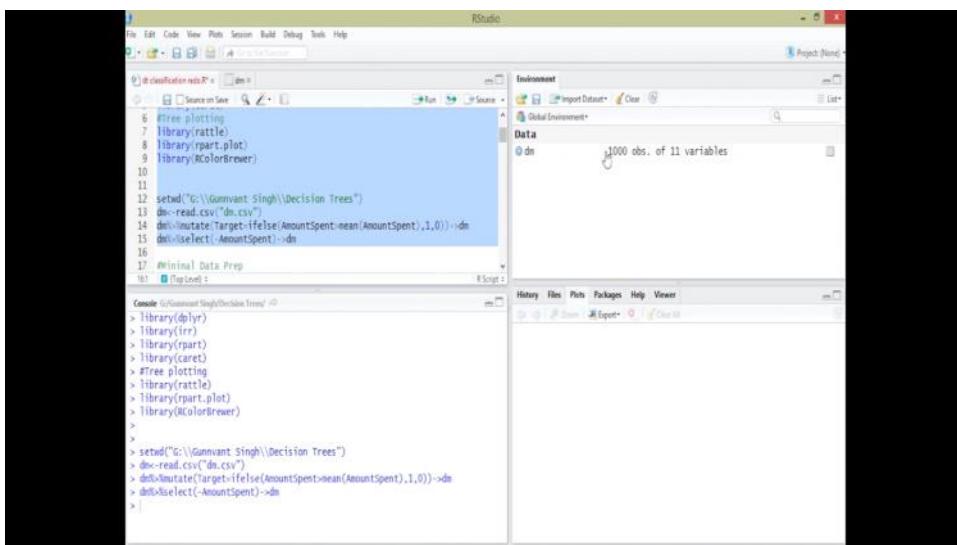
Decision Trees

★ R Code Demo Part 1 ★

© Jigsaw Academy Education Pvt Ltd



We will be using library (rpart) for classification and decision trees



The screenshot shows the RStudio interface. The top menu bar includes File, Edit, View, Plots, Session, build, Debug, tools, Help, and Project. The left sidebar has tabs for Source, Run, and Source. The main area shows a script named 'dt classification redo.R' with the following code:

```
1 #Classification Trees using rpart()#
2 library(dplyr)
3 library(irr)
4 library(rpart)
5 library(caret)
6 #tree plotting
7 library(rattle)
8 library(rpart.plot)
9 library(RColorBrewer)
10
11
12 setwd("C:\\Gumant.Singh\\Decision Trees")
13 dn<-read.csv("dn.csv")
14 dn$Institute<-ifelse(AmountSpent>mean(AmountSpent),1,0)~dn
15 dn$select(-AmountSpent)~dn
16
```

The right side of the interface shows the 'Environment' and 'Data' panes. The 'Data' pane displays the dataset 'dm' with the note "1000 observations of 11 variables".

ge	Gender	OwnHome	Married	Location	Salary	Children	History	Catalogs	Cust_Id	Target
idle	Male	Rent	Single	Close	63600	0	High	6	127	1
idle	Female	Rent	Single	Close	13500	0	Low	18	479	0
idle	Male	Own	Married	Close	85600	1	High	18	475	1
idle	Female	Own	Single	Close	68400	0	High	12	151	1
idle	Male	Own	Married	Close	30400	0	Low	6	320	0
idle	Female	Rent	Single	Close	48100	0	Medium	12	804	0
idle	Male	Own	Single	Close	68400	0	High	18	790	0
idle	Female	Own	Married	Close	51900	3	Low	6	43	0

Here is the data prep stage. It is minimal data prep. Substituting NA and converting to factors

```

RStudio
File Edit Code View Plots Session Build Debug Tools Help
Project (None)
Environment
Data dn 1000 obs. of 11 variables
Global Environment*
Data
  dn 1000 obs. of 11 variables
  13: dm<-read.csv("dm.csv")
  14: dm$Institute<-ifelse(AmountSpent>mean(AmountSpent),1,0)~dm
  15: dm$select(~AmountSpent)~dm
  16: #minimal Data Prep
  17: 
  18: dm$History1<-ifelse(is.na(dm$History),"Missing",as.character(dm$History))
  19: dm$History1<-as.factor(dm$History1)
  20: dm$History2<-as.factor(dm$History2)
  21: 
  22: summary(dm$History)
  23: 
  24: 
  25: 
  26: 
  27: 
  28: 
  29: 
  30: 
  31: 
  32: 
  33: 
  34: 
  35: 
  36: 
  37: 
  38: 
  39: 
  40: 
  41: 
  42: 
  43: 
  44: 
  45: 
  46: 
  47: 
  48: 
  49: 
  50: 
  51: 
  52: 
  53: 
  54: 
  55: 
  56: 
  57: 
  58: 
  59: 
  60: 
  61: 
  62: 
  63: 
  64: 
  65: 
  66: 
  67: 
  68: 
  69: 
  70: 
  71: 
  72: 
  73: 
  74: 
  75: 
  76: 
  77: 
  78: 
  79: 
  80: 
  81: 
  82: 
  83: 
  84: 
  85: 
  86: 
  87: 
  88: 
  89: 
  90: 
  91: 
  92: 
  93: 
  94: 
  95: 
  96: 
  97: 
  98: 
  99: 
  100: 
  101: 
  102: 
  103: 
  104: 
  105: 
  106: 
  107: 
  108: 
  109: 
  110: 
  111: 
  112: 
  113: 
  114: 
  115: 
  116: 
  117: 
  118: 
  119: 
  120: 
  121: 
  122: 
  123: 
  124: 
  125: 
  126: 
  127: 
  128: 
  129: 
  130: 
  131: 
  132: 
  133: 
  134: 
  135: 
  136: 
  137: 
  138: 
  139: 
  140: 
  141: 
  142: 
  143: 
  144: 
  145: 
  146: 
  147: 
  148: 
  149: 
  150: 
  151: 
  152: 
  153: 
  154: 
  155: 
  156: 
  157: 
  158: 
  159: 
  160: 
  161: 
  162: 
  163: 
  164: 
  165: 
  166: 
  167: 
  168: 
  169: 
  170: 
  171: 
  172: 
  173: 
  174: 
  175: 
  176: 
  177: 
  178: 
  179: 
  180: 
  181: 
  182: 
  183: 
  184: 
  185: 
  186: 
  187: 
  188: 
  189: 
  190: 
  191: 
  192: 
  193: 
  194: 
  195: 
  196: 
  197: 
  198: 
  199: 
  200: 
  201: 
  202: 
  203: 
  204: 
  205: 
  206: 
  207: 
  208: 
  209: 
  210: 
  211: 
  212: 
  213: 
  214: 
  215: 
  216: 
  217: 
  218: 
  219: 
  220: 
  221: 
  222: 
  223: 
  224: 
  225: 
  226: 
  227: 
  228: 
  229: 
  230: 
  231: 
  232: 
  233: 
  234: 
  235: 
  236: 
  237: 
  238: 
  239: 
  240: 
  241: 
  242: 
  243: 
  244: 
  245: 
  246: 
  247: 
  248: 
  249: 
  250: 
  251: 
  252: 
  253: 
  254: 
  255: 
  256: 
  257: 
  258: 
  259: 
  260: 
  261: 
  262: 
  263: 
  264: 
  265: 
  266: 
  267: 
  268: 
  269: 
  270: 
  271: 
  272: 
  273: 
  274: 
  275: 
  276: 
  277: 
  278: 
  279: 
  280: 
  281: 
  282: 
  283: 
  284: 
  285: 
  286: 
  287: 
  288: 
  289: 
  290: 
  291: 
  292: 
  293: 
  294: 
  295: 
  296: 
  297: 
  298: 
  299: 
  300: 
  301: 
  302: 
  303: 
  304: 
  305: 
  306: 
  307: 
  308: 
  309: 
  310: 
  311: 
  312: 
  313: 
  314: 
  315: 
  316: 
  317: 
  318: 
  319: 
  320: 
  321: 
  322: 
  323: 
  324: 
  325: 
  326: 
  327: 
  328: 
  329: 
  330: 
  331: 
  332: 
  333: 
  334: 
  335: 
  336: 
  337: 
  338: 
  339: 
  340: 
  341: 
  342: 
  343: 
  344: 
  345: 
  346: 
  347: 
  348: 
  349: 
  350: 
  351: 
  352: 
  353: 
  354: 
  355: 
  356: 
  357: 
  358: 
  359: 
  360: 
  361: 
  362: 
  363: 
  364: 
  365: 
  366: 
  367: 
  368: 
  369: 
  370: 
  371: 
  372: 
  373: 
  374: 
  375: 
  376: 
  377: 
  378: 
  379: 
  380: 
  381: 
  382: 
  383: 
  384: 
  385: 
  386: 
  387: 
  388: 
  389: 
  390: 
  391: 
  392: 
  393: 
  394: 
  395: 
  396: 
  397: 
  398: 
  399: 
  400: 
  401: 
  402: 
  403: 
  404: 
  405: 
  406: 
  407: 
  408: 
  409: 
  410: 
  411: 
  412: 
  413: 
  414: 
  415: 
  416: 
  417: 
  418: 
  419: 
  420: 
  421: 
  422: 
  423: 
  424: 
  425: 
  426: 
  427: 
  428: 
  429: 
  430: 
  431: 
  432: 
  433: 
  434: 
  435: 
  436: 
  437: 
  438: 
  439: 
  440: 
  441: 
  442: 
  443: 
  444: 
  445: 
  446: 
  447: 
  448: 
  449: 
  450: 
  451: 
  452: 
  453: 
  454: 
  455: 
  456: 
  457: 
  458: 
  459: 
  460: 
  461: 
  462: 
  463: 
  464: 
  465: 
  466: 
  467: 
  468: 
  469: 
  470: 
  471: 
  472: 
  473: 
  474: 
  475: 
  476: 
  477: 
  478: 
  479: 
  480: 
  481: 
  482: 
  483: 
  484: 
  485: 
  486: 
  487: 
  488: 
  489: 
  490: 
  491: 
  492: 
  493: 
  494: 
  495: 
  496: 
  497: 
  498: 
  499: 
  500: 
  501: 
  502: 
  503: 
  504: 
  505: 
  506: 
  507: 
  508: 
  509: 
  510: 
  511: 
  512: 
  513: 
  514: 
  515: 
  516: 
  517: 
  518: 
  519: 
  520: 
  521: 
  522: 
  523: 
  524: 
  525: 
  526: 
  527: 
  528: 
  529: 
  530: 
  531: 
  532: 
  533: 
  534: 
  535: 
  536: 
  537: 
  538: 
  539: 
  540: 
  541: 
  542: 
  543: 
  544: 
  545: 
  546: 
  547: 
  548: 
  549: 
  550: 
  551: 
  552: 
  553: 
  554: 
  555: 
  556: 
  557: 
  558: 
  559: 
  560: 
  561: 
  562: 
  563: 
  564: 
  565: 
  566: 
  567: 
  568: 
  569: 
  570: 
  571: 
  572: 
  573: 
  574: 
  575: 
  576: 
  577: 
  578: 
  579: 
  580: 
  581: 
  582: 
  583: 
  584: 
  585: 
  586: 
  587: 
  588: 
  589: 
  590: 
  591: 
  592: 
  593: 
  594: 
  595: 
  596: 
  597: 
  598: 
  599: 
  600: 
  601: 
  602: 
  603: 
  604: 
  605: 
  606: 
  607: 
  608: 
  609: 
  610: 
  611: 
  612: 
  613: 
  614: 
  615: 
  616: 
  617: 
  618: 
  619: 
  620: 
  621: 
  622: 
  623: 
  624: 
  625: 
  626: 
  627: 
  628: 
  629: 
  630: 
  631: 
  632: 
  633: 
  634: 
  635: 
  636: 
  637: 
  638: 
  639: 
  640: 
  641: 
  642: 
  643: 
  644: 
  645: 
  646: 
  647: 
  648: 
  649: 
  650: 
  651: 
  652: 
  653: 
  654: 
  655: 
  656: 
  657: 
  658: 
  659: 
  660: 
  661: 
  662: 
  663: 
  664: 
  665: 
  666: 
  667: 
  668: 
  669: 
  670: 
  671: 
  672: 
  673: 
  674: 
  675: 
  676: 
  677: 
  678: 
  679: 
  680: 
  681: 
  682: 
  683: 
  684: 
  685: 
  686: 
  687: 
  688: 
  689: 
  690: 
  691: 
  692: 
  693: 
  694: 
  695: 
  696: 
  697: 
  698: 
  699: 
  700: 
  701: 
  702: 
  703: 
  704: 
  705: 
  706: 
  707: 
  708: 
  709: 
  710: 
  711: 
  712: 
  713: 
  714: 
  715: 
  716: 
  717: 
  718: 
  719: 
  720: 
  721: 
  722: 
  723: 
  724: 
  725: 
  726: 
  727: 
  728: 
  729: 
  730: 
  731: 
  732: 
  733: 
  734: 
  735: 
  736: 
  737: 
  738: 
  739: 
  740: 
  741: 
  742: 
  743: 
  744: 
  745: 
  746: 
  747: 
  748: 
  749: 
  750: 
  751: 
  752: 
  753: 
  754: 
  755: 
  756: 
  757: 
  758: 
  759: 
  760: 
  761: 
  762: 
  763: 
  764: 
  765: 
  766: 
  767: 
  768: 
  769: 
  770: 
  771: 
  772: 
  773: 
  774: 
  775: 
  776: 
  777: 
  778: 
  779: 
  780: 
  781: 
  782: 
  783: 
  784: 
  785: 
  786: 
  787: 
  788: 
  789: 
  790: 
  791: 
  792: 
  793: 
  794: 
  795: 
  796: 
  797: 
  798: 
  799: 
  800: 
  801: 
  802: 
  803: 
  804: 
  805: 
  806: 
  807: 
  808: 
  809: 
  810: 
  811: 
  812: 
  813: 
  814: 
  815: 
  816: 
  817: 
  818: 
  819: 
  820: 
  821: 
  822: 
  823: 
  824: 
  825: 
  826: 
  827: 
  828: 
  829: 
  830: 
  831: 
  832: 
  833: 
  834: 
  835: 
  836: 
  837: 
  838: 
  839: 
  840: 
  841: 
  842: 
  843: 
  844: 
  845: 
  846: 
  847: 
  848: 
  849: 
  850: 
  851: 
  852: 
  853: 
  854: 
  855: 
  856: 
  857: 
  858: 
  859: 
  860: 
  861: 
  862: 
  863: 
  864: 
  865: 
  866: 
  867: 
  868: 
  869: 
  870: 
  871: 
  872: 
  873: 
  874: 
  875: 
  876: 
  877: 
  878: 
  879: 
  880: 
  881: 
  882: 
  883: 
  884: 
  885: 
  886: 
  887: 
  888: 
  889: 
  890: 
  891: 
  892: 
  893: 
  894: 
  895: 
  896: 
  897: 
  898: 
  899: 
  900: 
  901: 
  902: 
  903: 
  904: 
  905: 
  906: 
  907: 
  908: 
  909: 
  910: 
  911: 
  912: 
  913: 
  914: 
  915: 
  916: 
  917: 
  918: 
  919: 
  920: 
  921: 
  922: 
  923: 
  924: 
  925: 
  926: 
  927: 
  928: 
  929: 
  930: 
  931: 
  932: 
  933: 
  934: 
  935: 
  936: 
  937: 
  938: 
  939: 
  940: 
  941: 
  942: 
  943: 
  944: 
  945: 
  946: 
  947: 
  948: 
  949: 
  950: 
  951: 
  952: 
  953: 
  954: 
  955: 
  956: 
  957: 
  958: 
  959: 
  960: 
  961: 
  962: 
  963: 
  964: 
  965: 
  966: 
  967: 
  968: 
  969: 
  970: 
  971: 
  972: 
  973: 
  974: 
  975: 
  976: 
  977: 
  978: 
  979: 
  980: 
  981: 
  982: 
  983: 
  984: 
  985: 
  986: 
  987: 
  988: 
  989: 
  990: 
  991: 
  992: 
  993: 
  994: 
  995: 
  996: 
  997: 
  998: 
  999: 
  1000: 
  1001: 
  1002: 
  1003: 
  1004: 
  1005: 
  1006: 
  1007: 
  1008: 
  1009: 
  1010: 
  1011: 
  1012: 
  1013: 
  1014: 
  1015: 
  1016: 
  1017: 
  1018: 
  1019: 
  1020: 
  1021: 
  1022: 
  1023: 
  1024: 
  1025: 
  1026: 
  1027: 
  1028: 
  1029: 
  1030: 
  1031: 
  1032: 
  1033: 
  1034: 
  1035: 
  1036: 
  1037: 
  1038: 
  1039: 
  1040: 
  1041: 
  1042: 
  1043: 
  1044: 
  1045: 
  1046: 
  1047: 
  1048: 
  1049: 
  1050: 
  1051: 
  1052: 
  1053: 
  1054: 
  1055: 
  1056: 
  1057: 
  1058: 
  1059: 
  1060: 
  1061: 
  1062: 
  1063: 
  1064: 
  1065: 
  1066: 
  1067: 
  1068: 
  1069: 
  1070: 
  1071: 
  1072: 
  1073: 
  1074: 
  1075: 
  1076: 
  1077: 
  1078: 
  1079: 
  1080: 
  1081: 
  1082: 
  1083: 
  1084: 
  1085: 
  1086: 
  1087: 
  1088: 
  1089: 
  1090: 
  1091: 
  1092: 
  1093: 
  1094: 
  1095: 
  1096: 
  1097: 
  1098: 
  1099: 
  1100: 
  1101: 
  1102: 
  1103: 
  1104: 
  1105: 
  1106: 
  1107: 
  1108: 
  1109: 
  1110: 
  1111: 
  1112: 
  1113: 
  1114: 
  1115: 
  1116: 
  1117: 
  1118: 
  1119: 
  1120: 
  1121: 
  1122: 
  1123: 
  1124: 
  1125: 
  1126: 
  1127: 
  1128: 
  1129: 
  1130: 
  1131: 
  1132: 
  1133: 
  1134: 
  1135: 
  1136: 
  1137: 
  1138: 
  1139: 
  1140: 
  1141: 
  1142: 
  1143: 
  1144: 
  1145: 
  1146: 
  1147: 
  1148: 
  1149: 
  1150: 
  1151: 
  1152: 
  1153: 
  1154: 
  1155: 
  1156: 
  1157: 
  1158: 
  1159: 
  1160: 
  1161: 
  1162: 
  1163: 
  1164: 
  1165: 
  1166: 
  1167: 
  1168: 
  1169: 
  1170: 
  1171: 
  1172: 
  1173: 
  1174: 
  1175: 
  1176: 
  1177: 
  1178: 
  1179: 
  1180: 
  1181: 
  1182: 
  1183: 
  1184: 
  1185: 
  1186: 
  1187: 
  1188: 
  1189: 
  1190: 
  1191: 
  1192: 
  1193: 
  1194: 
  1195: 
  1196: 
  1197: 
  1198: 
  1199: 
  1200: 
  1201: 
  1202: 
  1203: 
  1204: 
  1205: 
  1206: 
  1207: 
  1208: 
  1209: 
  1210: 
  1211: 
  1212: 
  1213: 
  1214: 
  1215: 
  1216: 
  1217: 
  1218: 
  1219: 
  1220: 
  1221: 
  1222: 
  1223: 
  1224: 
  1225: 
  1226: 
  1227: 
  1228: 
  1229: 
  1230: 
  1231: 
  1232: 
  1233: 
  1234: 
  1235: 
  1236: 
  1237: 
  1238: 
  1239: 
  1240: 
  1241: 
  1242: 
  1243: 
  1244: 
  1245: 
  1246: 
  1247: 
  1248: 
  1249: 
  1250: 
  1251: 
  1252: 
  1253: 
  1254: 
  1255: 
  1256: 
  1257: 
  1258: 
  1259: 
  1260: 
  1261: 
  1262: 
  1263: 
  1264: 
  1265: 
  1266: 
  1267: 
  1268: 
  1269: 
  1270: 
  1271: 
  1272: 
  1273: 
  1274: 
  1275: 
  1276: 
  1277: 
  1278: 
  1279: 
  1280: 
  1281: 
  1282: 
  1283: 
  1284: 
  1285: 
  1286: 
  1287: 
  1288: 
  1289: 
  1290: 
  1291: 
  1292: 
  1293: 
  1294: 
  1295: 
  1296: 
  1297: 
  1298: 
  1299: 
  1300: 
  1301: 
  1302: 
  1303: 
  1304: 
  1305: 
  1306: 
  1307: 
  1308: 
  1309: 
  1310: 
  1311: 
  1312: 
  1313: 
  1314: 
  1315: 
  1316: 
  1317: 
  1318: 
  1319: 
  1320: 
  1321: 
  1322: 
  1323: 
  1324: 
  1325: 
  1326: 
  1327: 
  1328: 
  1329: 
  1330: 
  1331: 
  1332: 
  1333: 
  1334: 
  1335: 
  1336: 
  1337: 
  1338: 
  1339: 
  1340: 
  1341: 
  1342: 
  1343: 
  1344: 
  1345: 
  1346: 
  1347: 
  1348: 
  1349: 
  1350: 
  1351: 
  1352: 
  1353: 
  1354: 
  1355: 
  1356: 
  1357: 
  1358: 
  1359: 
  1360: 
  1361: 
  1362: 
  1363: 
  1364: 
  1365: 
  1366: 
  1367: 
  1368: 
  1369: 
  1370: 
  1371: 
  1372: 
  1373: 
  1374: 
  1375: 
  1376: 
  1377: 
  1378: 
  1379: 
  1380: 
  1381: 
  1382: 
  1383: 
  1384: 
  1385: 
  1386: 
  1387: 
  1388: 
  1389: 
  1390: 
  1391: 
  1392: 
  1393: 
  1394: 
  1395: 
  1396: 
  1397: 
  1398: 
  1399: 
  1400: 
  1401: 
  1402: 
  1403: 
  1404: 
  1405: 
  1406: 
  1407: 
  1408: 
  1409: 
  1410: 
  1411: 
  1412: 
  1413: 
  1414: 
  1415: 
  1416: 
  1417: 
  1418: 
  1419: 
  1420: 
  1421: 
  1422: 
  1423: 
  1424: 
  1425: 
  1426: 
  1427: 
  1428: 
  1429: 
  1430: 
  1431: 
  1432: 
  1433: 
  1434: 
  1435: 
  1436: 
  1437: 
  1438: 
  1439: 
  1440: 
  1441: 
  1442: 
  1443: 
  1444: 
  1445: 
  1446: 
  1447: 
  1448: 
  1449: 
  1450: 
  1451: 
  1452: 
  1453: 
  1454: 
  1455: 
  1456: 
  1457: 
  1458: 
  1459: 
  1460: 
  1461: 
  1462: 
  1463: 
  1464: 
  1465: 
  1466: 
  1467: 
  1468: 
  1469: 
  1470: 
  1471: 
  1472: 
  1473: 
  1474: 
  1475: 
  1476: 
  1477: 
  1478: 
  1479: 
  1480: 
  1481: 
  1482: 
  1483: 
  1484: 
  1485: 
  1486: 
  1487: 
  1488: 
  1489: 
  1490: 
  1491: 
  1492: 
  1493: 
  1494: 
  1495: 
  1496: 
  1497: 
  1498: 
  1499: 
  1500: 
  1501: 
  1502: 
  1503: 
  1504: 
  1505: 
  1506: 
  1507: 
  1508: 
  1509: 
  1510: 
  1511: 
  1512: 
  1513: 
  1514: 
  1515: 
  1516: 
  1517: 
  1518: 
  1519: 
  1520: 
  1521: 
  1522: 
  1523: 
  1524: 
  1525: 
  1526: 
  1527: 
  1528: 
  1529: 
  1530: 
  1531: 
  1532: 
  1533: 
  1534: 
  1535: 
  1536: 
  1537: 
  1538: 
  1539: 
  1540: 
  1541: 
  1542: 
  1543: 
  154
```

RStudio

```
[1]: dclassification.rda.R
2: 
3: 
4: 
5: 
6: 
7: 
8: 
9: 
10: 
11: 
12: 
13: 
14: 
15: 
16: 
17: 
18: 
19: 
20: 
21: 
22: 
23: 
24: 
25: 
26: 
27: 
28: 
29: 
30: 
31: 
32: 
33: 
34: 
35: 
36: 
```

Console (dclassification.R) >

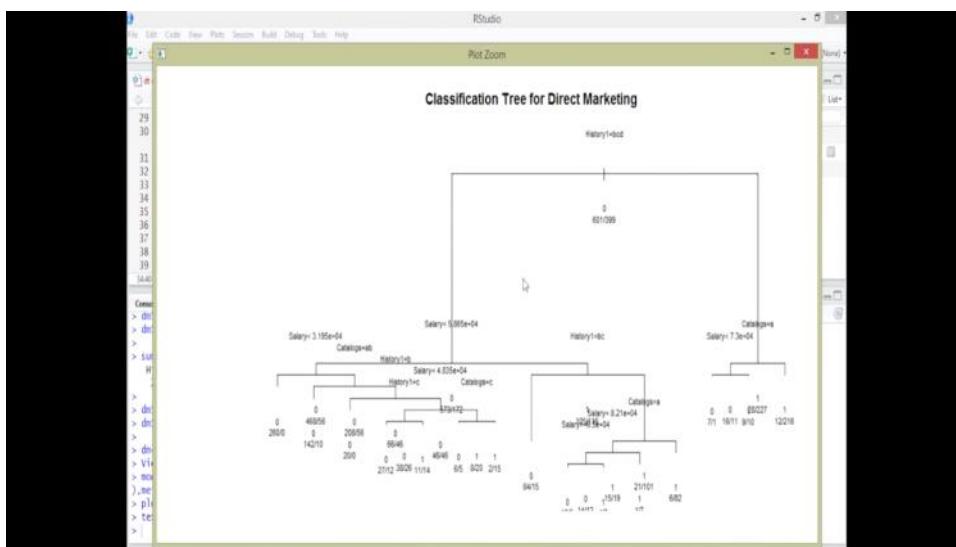
```
> dm<-read.csv("dm.csv")
> dm$Mutate([target=felse(AmountSpent>mean(AmountSpent),1,0))>dm
> dm$Select(-AmountSpent)>dm
> View(dm)
> dmHistory1<-ifelse(is.na(dm$History),"Missing",as.character(dm$History))
> dmHistory1<-as.factor(dm$History1)
> 
> summary(dm$History1)
   High    Low   Medium Missing 
  255    230    222    303 
> 
> dm$Children<-as.factor(dm$Children)
> dm$Catalogs<-as.factor(dm$Catalogs)
> 
> dm<-dm[,-8]
> View(dm)
> 
```

RStudio

```
[1]: dclassification.rda.R
2: 
3: 
4: 
5: 
6: 
7: 
8: 
9: 
10: 
11: 
12: 
13: 
14: 
15: 
16: 
17: 
18: 
19: 
20: 
21: 
22: 
23: 
24: 
25: 
26: 
27: 
28: 
29: 
30: 
31: 
32: 
33: 
34: 
35: 
36: 
```

Console (dclassification.R) >

```
> dm$Select(-AmountSpent)>dm
> View(dm)
> dmHistory1<-ifelse(is.na(dm$History),"Missing",as.character(dm$History))
> dmHistory1<-as.factor(dm$History1)
> 
> summary(dm$History1)
   High    Low   Medium Missing 
  255    230    222    303 
> 
> dm$Children<-as.factor(dm$Children)
> dm$Catalogs<-as.factor(dm$Catalogs)
> 
> dm<-dm[,-8]
> View(dm)
> mod<-rpart(Target~.,data=dm[, -9],control=rpart.control(cp=0.002 ,maxdepth=7),method="class",parms=list(split="gini"))
> 
> plot(mod, margin=0.1, main="Classification Tree for Direct Marketing")
> text(mod, use.n=TRUE, all=TRUE, cex=.7)
> 
> fancyRpartPlot(mod)
> 
> printcp(mod)
> plotcp(mod, minline = TRUE)
> 
```



We can make it prettier by using the "fancyrpartplot(model)
This uses the following libraries --> rattle, rpart.plot, RColorBrewer

The screenshot shows the RStudio interface with the following details:

- File Edit Code View Plots Session Build Debug Tools Help**
- Project (None)**
- Environment** pane showing:
 - Global Environment
 - Data: dn (1000 obs. of 11 variables), mod (List of 15)
- Code Editor** pane containing R code for building a classification tree:

```
library(rpart)
library(rpart.plot)
library(caret)
library(rattle)
library(rpart)
library(Kolmogorov)
dn = read.csv("dn.csv")
setwd("G:\\Gaurav Singh\\Decision Trees")
dnHistory1<-file("15.naidsHistory"),'Missing',as.character(dn$History))
dnHistory1<-as.factor(dn$History1)
summary(dn$History1)
High Low Medium Missing
255 230 212 303
dn$children<-as.factor(dn$children)
dn$catalogs<-as.factor(dn$catalogs)
dn<-dn[-8]
mod=rpart(target~.,data=dn[, -9],control=rpart.control(cp=0.002,noundp=2),
method="class", parms=list(split="gini"))
plot(mod, margin=1, main="Classification Tree for Direct Marketing")
text(mod, use.na=TRUE, all=TRUE, cex=.7)
>
```
- Plots** pane showing a "Classification Tree for Direct Marketing". The tree has the following structure:

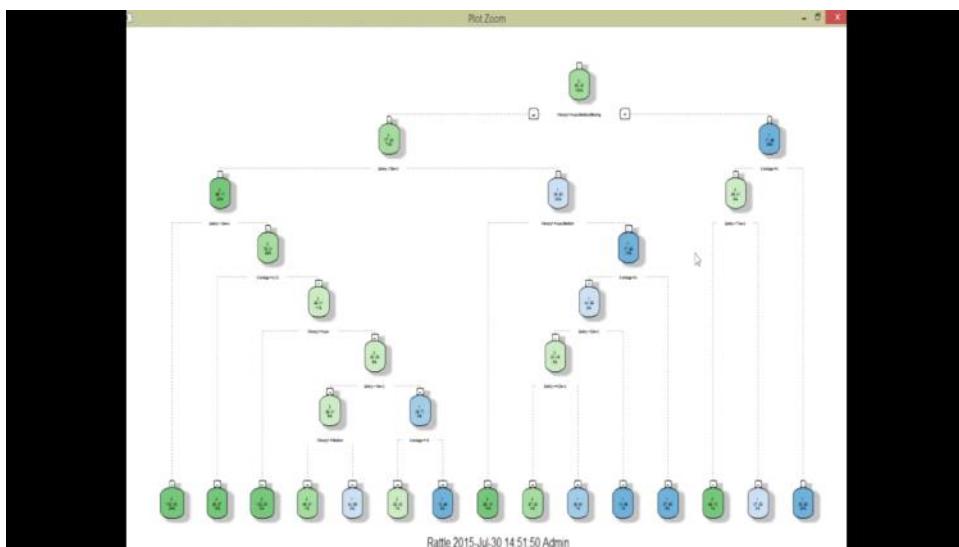
```
root[1] > High[2] > Low[3] > Medium[4] > Missing[5]
```

Leaf nodes (Terminal Nodes):

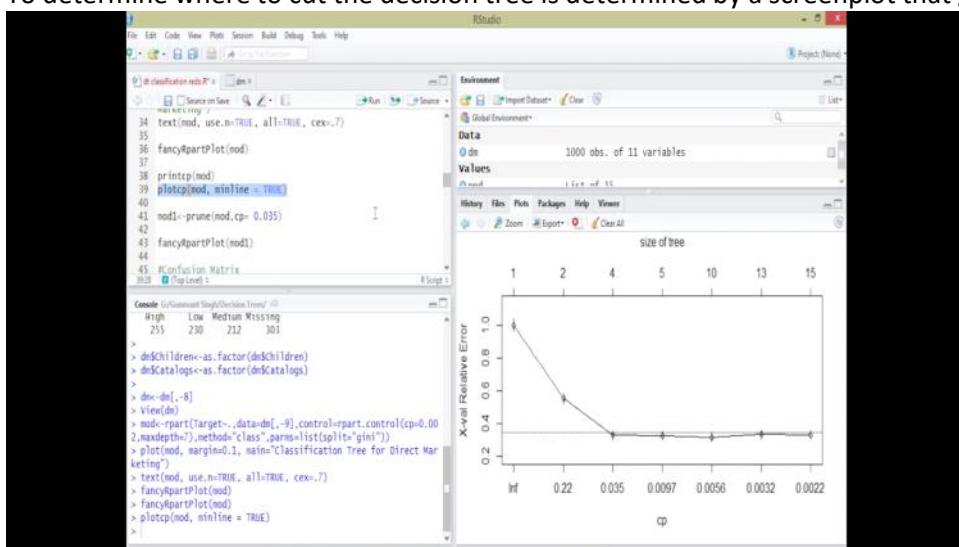
 - Node 1: 601 obs. (0.601)
 - Node 2: 230 obs. (0.230)
 - Node 3: 212 obs. (0.212)
 - Node 4: 303 obs. (0.303)
 - Node 5: 1000 obs. (1.000)

Variables used in splits:

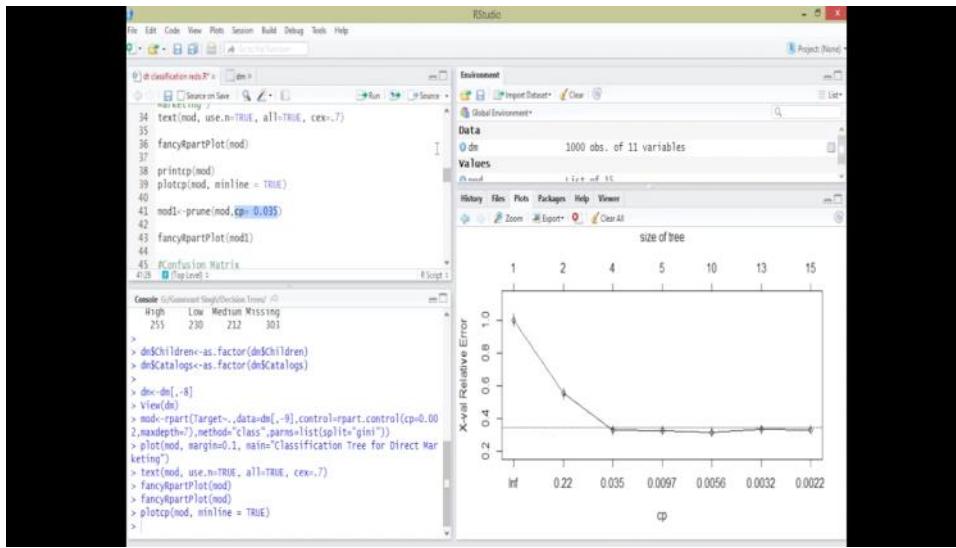
 - Node 1: History
 - Node 2: Income
 - Node 3: Income
 - Node 4: Income
 - Node 5: Income



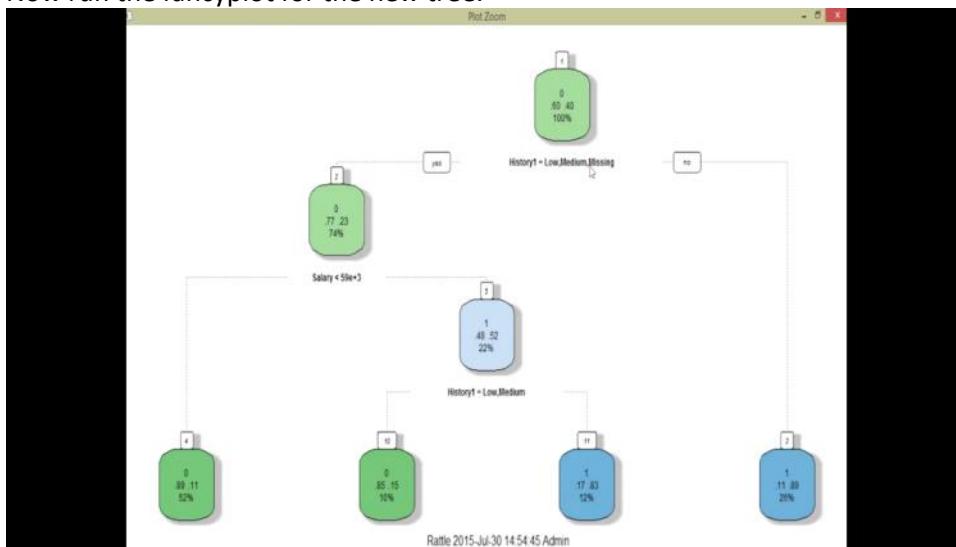
To determine where to cut the decision tree is determined by a screenplot that gives a breakdown:



Use the CP value where the curve drops below the line to determine where to prune.



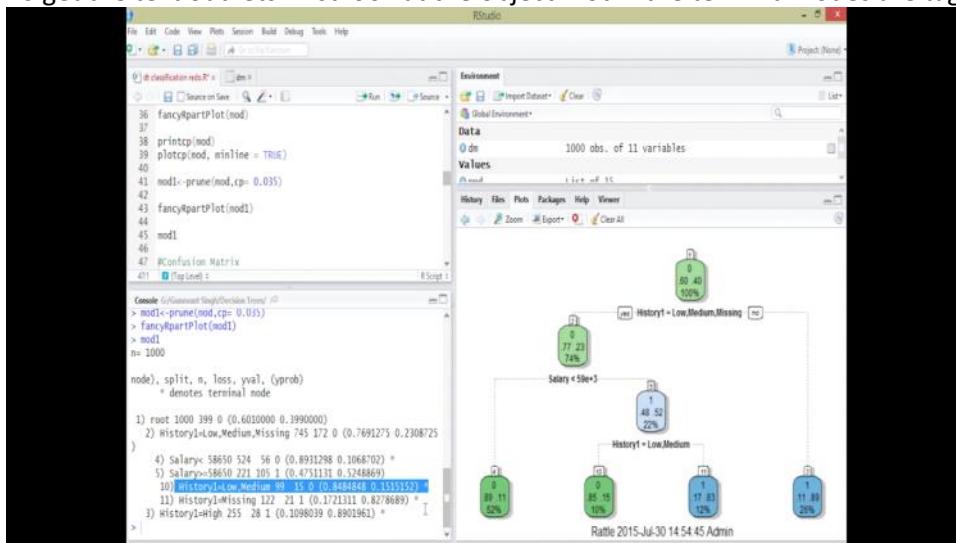
Now run the fancyplot for the new tree.

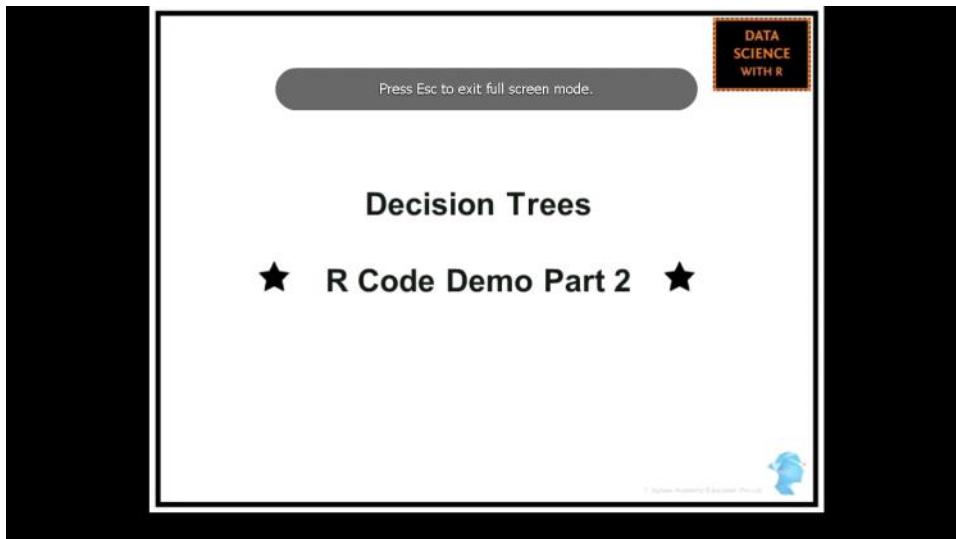


The next step is to derive the rules after pruning .

The rules can either be derived by interpreting the plot or using the text output of decision tree.

To get the text out let's first look at the object mod1. the terminal nodes are tagged with *:





Classification - Excel								
Rule	# obs	Node	# goods [1]	# bads [0]	bad rate	%age of total obs	Cumulative Total	Jobs at Node
1. If history1=[low,Medium,Missing] and salary < 58650	524	4	56	46	89.3130%	52.400%	77.870%	77.870%
2. if salary > 58650 and history = [low,medium]	99	10	15	84	84.8485%	9.900%	62.300%	15.977%
3. if salary >= 58650 and history = [missing]	122	11	101	21	17.2131%	12.200%	74.500%	3.494%
4. if history1 = [high]	255	3	227	28	10.9804%	25.500%	100.000%	4.059%

Classification - Excel								
Rule	# obs	Node	# goods [1]	# bads [0]	bad rate	%age of total obs	Cumulative Total	Jobs at Node
1. If history1=[low,Medium,Missing] and salary < 58650	524	4	56	46	89.3130%	52.400%	52.400%	52.400%
2. if salary > 58650 and history = [low,medium]	99	10	15	84	84.8485%	9.900%	62.300%	62.300%
3. if salary >= 58650 and history = [missing]	122	11	101	21	17.2131%	12.200%	74.500%	74.500%
4. if history1 = [high]	255	3	227	28	10.9804%	25.500%	100.000%	100.000%

RStudio Environment pane showing:

- Data: dn (1000 obs. of 11 variables)
- Values: mod (List of 15), mod1 (List of 15)

RStudio Environment pane showing:

- Data: dn (1000 obs. of 11 variables)
- Values: actual (num [1:1000] 0 1 0 1 1 0 0 0 1 ...), mod (List of 15), mod1 (List of 15), predicted (num [1:1000] 1 1 0 1 1 0 0 1 0 1 ...)

Use confusion matrix from package caret to generate a confusion matrix.

RStudio Console output:

```
Confusion Matrix and Statistics
Reference
Prediction 0 1
0 552 71
1 49 328

Accuracy : 0.88
95% CI : (0.8582, 0.8995)
No Information Rate : 0.601
P-Value [Acc > NIR] : < 2e-16

Kappa : 0.747
McNemar's Test P-Value : 0.05523

Sensitivity : 0.8221
Specificity : 0.9185
```

Load the library ROCR to create the ROC(receiver operating) curve:

The screenshot shows the RStudio interface. In the top-left pane, there is a script editor with the following R code:

```

51 #ROC curve analysis
52 #library(ROCR)
53 pred<-prediction(actual,predicted)
54 perf<-performance(pred,"tpr","fpr")
55 plot(perf,col="red")
56 abline(0,1, lty = 8, col = "grey")
57 auc<-performance(pred,"auc")
58 unlist(auc@y.values)
59

```

In the top-right pane, the 'Environment' tab is selected, showing objects like 'dn', 'mod', 'mod1', and 'predicted'. The bottom-right pane shows the 'Packaging' window with a list of packages in the User Library:

Name	Description	Version
assertthat	Easy pre and post assertions.	0.1
BH	Boost C++ Header Files	1.55.0-1
bitops	Bitwise Operations	1.0-6
BradleyTerry2	Bradley-Terry Models	1.0-6
brglm	Bias reduction in binomial-response generalized linear models.	0.5-9
carr	Companion to Applied Regression	2.0-25
caret	Classification and Regression Training	6.0-47
caTools	Tools: moving window statistics, GF, Base64, RDC	1.17.1
chron	Chronological Objects which can Handle Dates	2.3-47

We will use the prediction function as part of ROCR.

Once we have that we will run a performance function with the earlier output.

After getting the output, we plot it in a graph and draw a 45 degree line using abline

Then we compute the area under the curve which tells about the lift/improvement:

Auc<-performance(pred,"auc")

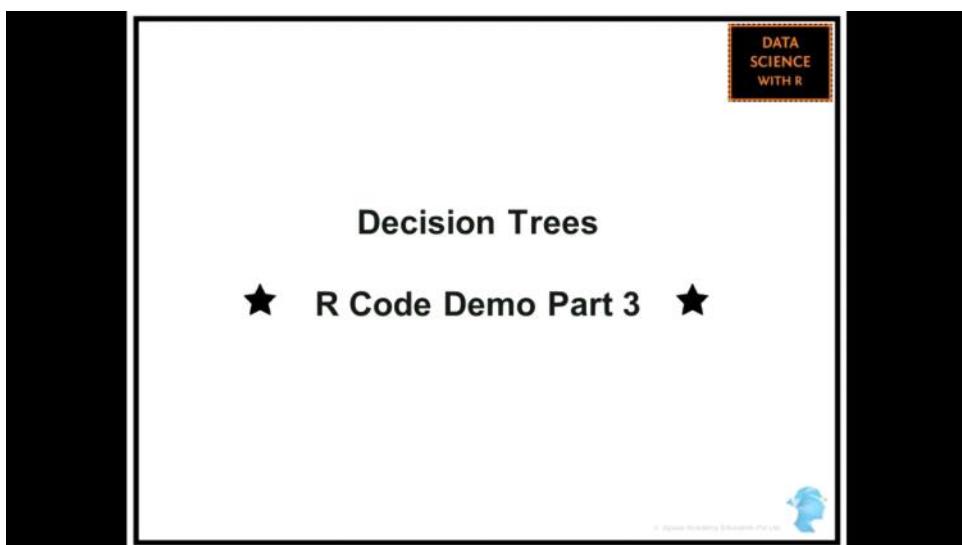
It is a S4 object.

Since we need data from Y values, we extract it.

Since it is a list object, we will unlist it.

Unlist(auc@y.values)

For a decent classifier our AUC needs to be better than 0.6



How to build regression models using decision trees. We will use rpart to build both the classification tree and regression tree:

The screenshot shows the RStudio interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Tools, Help. The left pane contains a script editor with the following R code:

```
#Regression Trees using rpart()#
library(dplyr)
library(rattle)
#tree plotting
library(rpart.plot)
library(RColorBrewer)
setwd("C:\\Gunnvant Singh\\Decision Trees")
dn<-read.csv("dn.csv")

```

The right pane shows the Environment tab with "Global Environment" and "Environment is empty". Below it is the Data tab.

We will look at predicting amount spent

The screenshot shows the RStudio interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Tools, Help. The left pane contains a script editor with the same R code as before. The right pane shows the Environment tab with a data preview:

Gender	OwnHome	Married	Location	Salary	Children	History	Catalogs	AmountSpent	Cont_M
Female	Own	Single	Far	47500	0	High	6	765	247
Male	Rent	Single	Close	63000	0	High	8	1318	327
Female	Rent	Single	Close	23500	0	Low	18	296	479
Male	Own	Married	Close	85000	2	High	18	2436	475
Female	Own	Single	Close	68400	0	High	12	1384	251
Male	Own	Married	Close	30400	0	Low	6	405	328
Female	Rent	Single	Close	48200	0	Medium	12	781	384
Male	Own	Single	Close	63000	0	High	18	1155	798
Female	Own	Married	Close	52000	2	Low	6	156	43

The Data tab below shows "1000 obs. of 11 variables" and "dn".

This requires minimal data prep.

The screenshot shows the RStudio interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Tools, Help. The left pane contains a script editor with the following R code:

```
#Minimal Data Prep
dn$History<-ifelse(is.na(dn$History),"Missing",as.character(dn$History))
dn$History1<-as.factor(dn$History1)
summary(dn$History1)
```

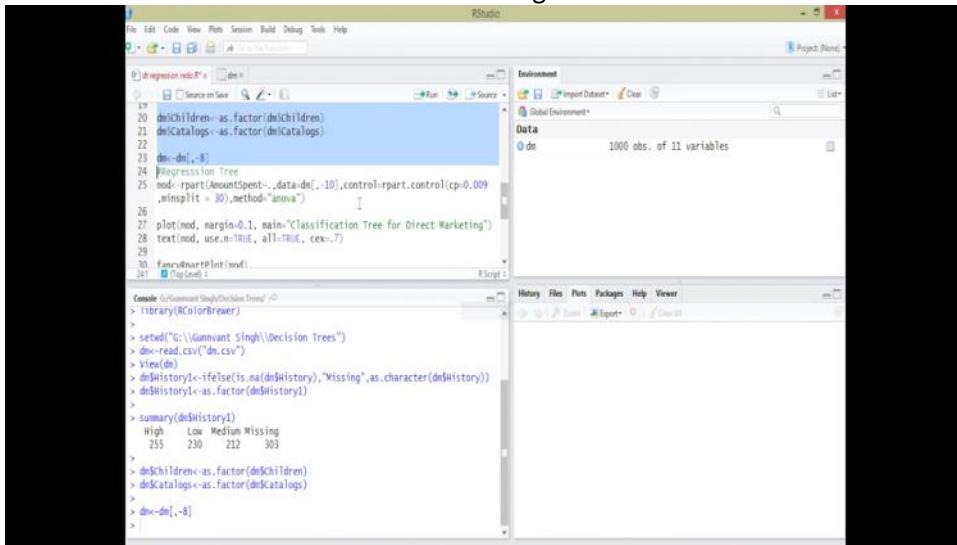
The right pane shows the Environment tab with a data preview:

Gender	OwnHome	Married	Location	Salary	Children	History	Catalogs	AmountSpent	Cont_M
Female	Own	Single	Far	47500	0	High	6	765	247
Male	Rent	Single	Close	63000	0	High	8	1318	327
Female	Rent	Single	Close	23500	0	Low	18	296	479
Male	Own	Married	Close	85000	2	High	18	2436	475
Female	Own	Single	Close	68400	0	High	12	1384	251
Male	Own	Married	Close	30400	0	Low	6	405	328
Female	Rent	Single	Close	48200	0	Medium	12	781	384
Male	Own	Single	Close	63000	0	High	18	1155	798
Female	Own	Married	Close	52000	2	Low	6	156	43

The Data tab below shows "1000 obs. of 11 variables" and "dn".

First step is to replace NA's . We do it for history, this is basically an indicator of historic spending behavior

We also want to convert children and catalogs to factors as well.

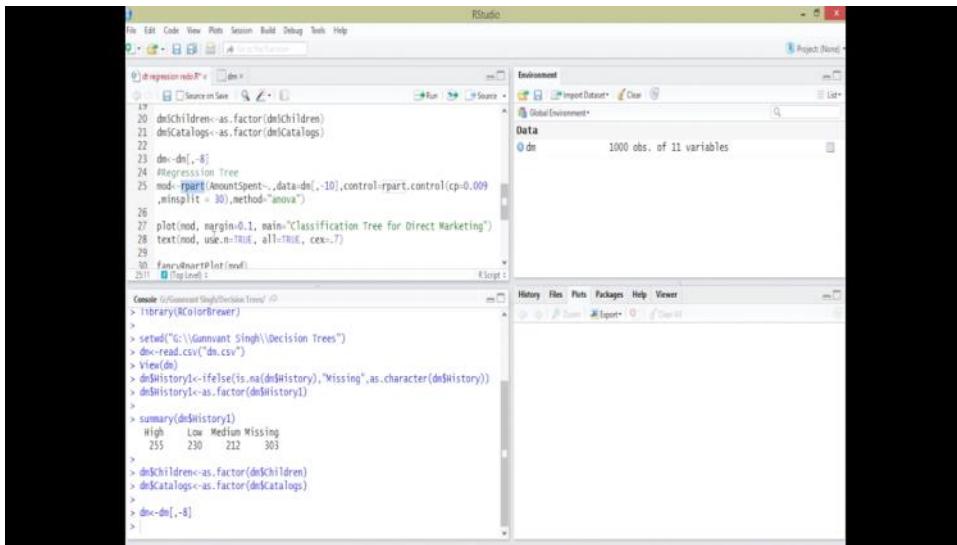


The screenshot shows the RStudio interface with the following code in the script pane:

```
## Regression Tree
dn$children<-as.factor(dn$children)
dn$catalogs<-as.factor(dn$catalogs)
dn<-dn[-8]
#Regression Tree
mod<-rpart(AmountSpent~.,data=dn[,-10],control=rpart.control(cp=0.009
,minsplits = 30),method="anova")
plot(mod, margin=0.1, main="Classification Tree for Direct Marketing")
text(mod, use.n=TRUE, all=TRUE, cex=.7)
fancyRpartPlot(mod)
```

The environment pane shows the dataset 'dn' with 1000 observations and 11 variables. The global environment pane shows 'mod' as a list of 15 elements.

We will use all our variable and run the rpart command to find the Amount spent,excluding the customer id

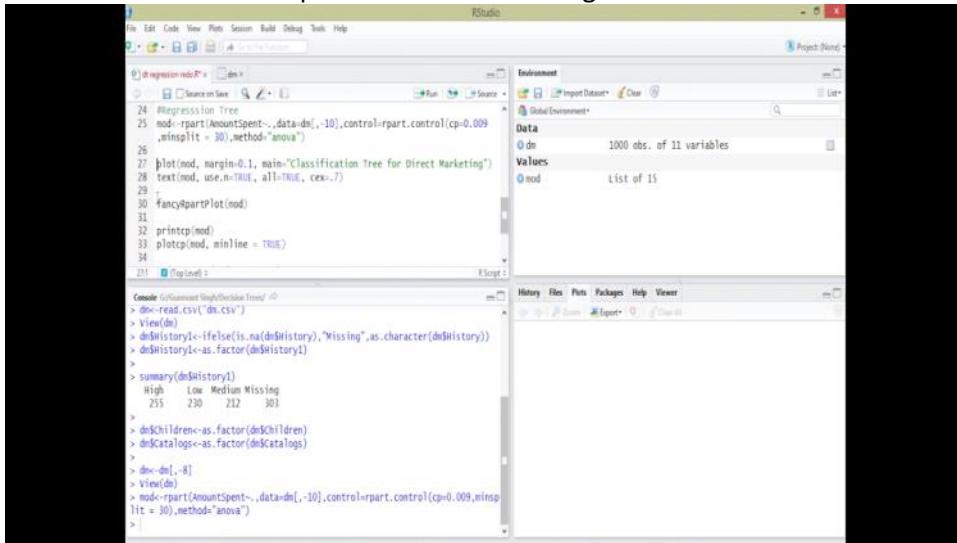


The screenshot shows the RStudio interface with the same R code as the previous one, but with a different output in the console pane:

```
## Regression Tree
dn$children<-as.factor(dn$children)
dn$catalogs<-as.factor(dn$catalogs)
dn<-dn[-8]
#Regression Tree
mod<-rpart(AmountSpent~.,data=dn[,-10],control=rpart.control(cp=0.009
,minsplits = 30),method="anova")
plot(mod, margin=0.1, main="Classification Tree for Direct Marketing")
text(mod, use.n=TRUE, all=TRUE, cex=.7)
fancyRpartPlot(mod)
```

The environment pane shows the dataset 'dn' with 1000 observations and 11 variables. The global environment pane shows 'mod' as a list of 15 elements.

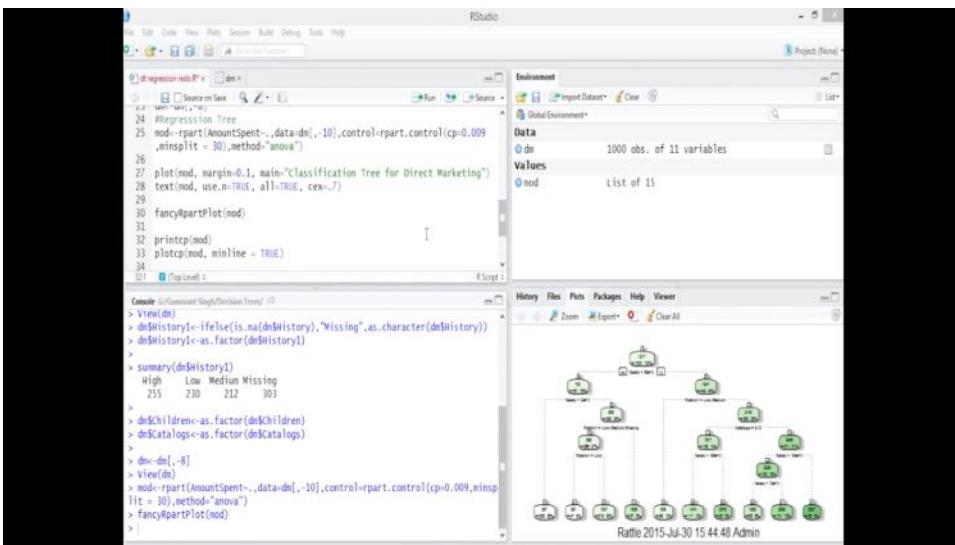
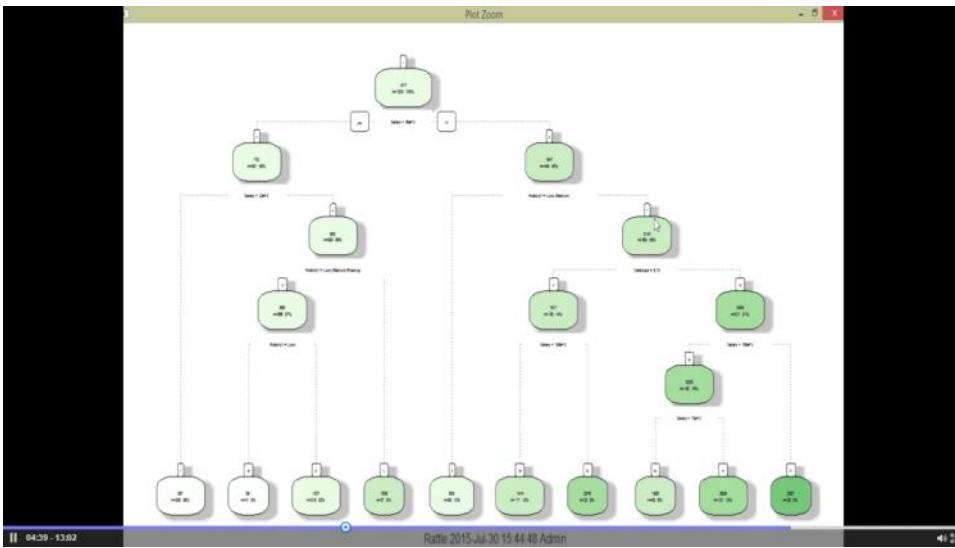
Method anova tells the rpart model to build a regression model.



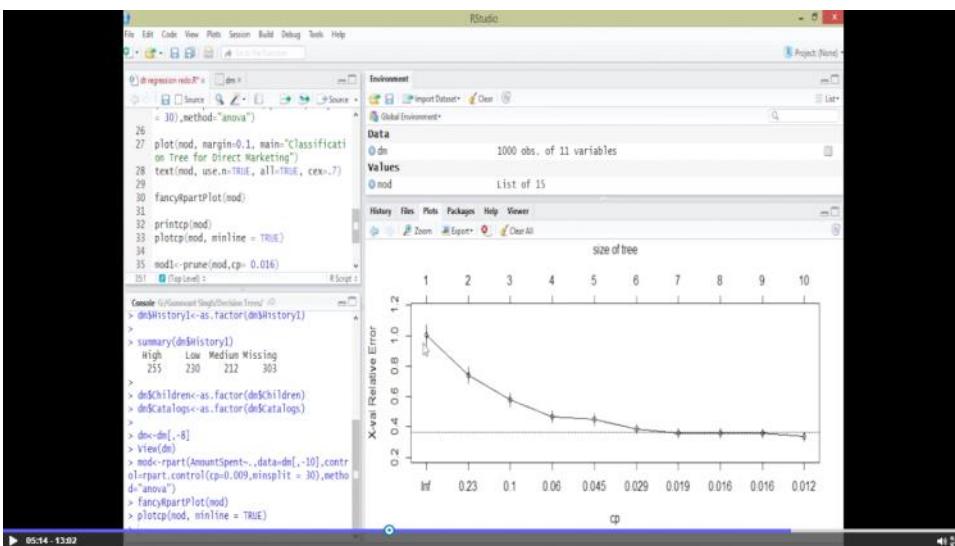
The screenshot shows the RStudio interface with the same R code as the previous ones, but with a different output in the console pane:

```
## Regression Tree
mod<-rpart(AmountSpent~.,data=dn[,-10],control=rpart.control(cp=0.009
,minsplits = 30),method="anova")
plot(mod, margin=0.1, main="Classification Tree for Direct Marketing")
text(mod, use.n=TRUE, all=TRUE, cex=.7)
fancyRpartPlot(mod)
print(mod)
plot(mod, minline = TRUE)
summary(dn$History1)
High Low Medium Missing
255 230 212 303
dn$children<-as.factor(dn$children)
dn$catalogs<-as.factor(dn$catalogs)
dn<-dn[-8]
View(dn)
mod<-rpart(AmountSpent~.,data=dn[,-10],control=rpart.control(cp=0.009,minsplits = 30),method="anova")
```

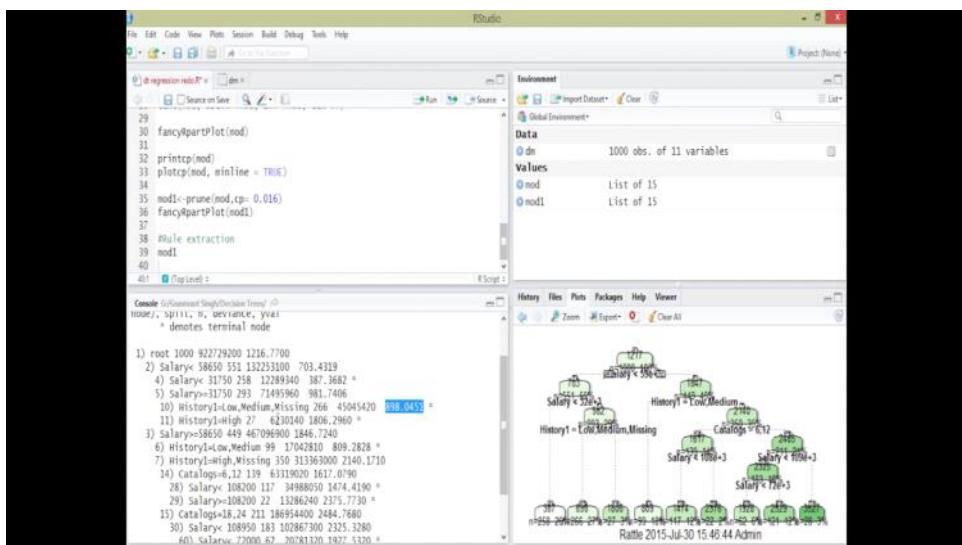
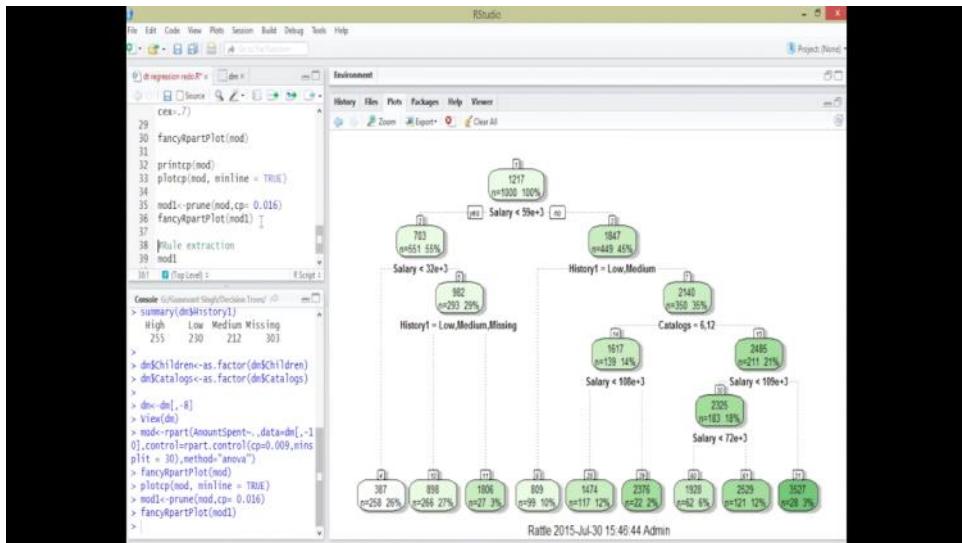
The environment pane shows the dataset 'dn' with 1000 observations and 11 variables. The global environment pane shows 'mod' as a list of 15 elements.



Since this is a big tree we will need to prune it. To do that we will need to have a plot of cost complexity and minimum error. To do that we will use plotCP command



Using this plot we will prune the tree for a value of 0.016. that's where the line intersect



Organize the summary of this output in an excel sheet.

The screenshot shows an Excel spreadsheet titled "Regression Trees - Excel" with the following data:

Node	Rule	Mean value	Population Mean	Population std z value	# obs	%age of total
1	4 Salary < 31750	387.3862	1216.77	961.0688 -0.80299562	258	30.3177%
2	10 Salary < 3157 -> Salary < 31500 and History1=Low, Medium, Missing	390.041		0.39441925	266	31.2574%
3	11 Salary < 31750 -> Salary < 31500 and History1 = [High]	1306.296		1.879460252	27	3.1723%
4	4 Salary < 31500 and History1 = [Low and Medium]	389.2628		0.84206583	98	11.6333%
5	28 Salary < 310200 and History1=[High and Missing] and Catalogs > [6,12]	1474.419		1.54014549	117	13.7485%
29	29 208200->Salary < [High and Missing] and Catalogs > [6,12]	2375.772		2.47011881	22	2.5812%
30	30 Salary < 208200 and Catalogs > [6,12]	1937.352		2.08961335	62	7.20553%
31	31 200000->Salary < 208200 and Catalogs > [6,12]	2529.157		2.611690912	121	14.2161%
32	32 Salary < 108950 and Catalogs > [6,12]	3326.821		3.089607055	26	3.29025%

Calculate the means, population mean, std dev , z values

Regression Trees - Excel

A	B	C	D	E	F	G	H
Node	Rule	Mean value	Population Mean	Population std. value	# obs	%age of total	
4	Salary > 31750	387.3862	1216.77	961.0608	-0.302949342	258	30.3177%
10	Salary > 3157 <=58550 and History1 = [Low, Medium, Missing]	890.041		0.93441925		266	31.2573%
11	Salary > 11750 <=58000 and History1 = [High]	1806.296		1.87946252		27	3.1723%
6	Salary >=58550 and History1 = [Low and Medium]	809.2620		0.84206552		98	12.6333%
20	58550 <=Salary <=80200 and History1=[High and Missing] and Catalogue = [6,12]	1474.415		1.53414529		117	13.74853%
29	103200 <=Salary and History1=[High and Missing] and Catalogue = [6,12]	2375.775		2.470511881		22	2.58519%
60	58550 <=Salary <=72000 and Catalogue = [18,24]	1927.532		2.086013315		82	7.28535%
61	72000 <=Salary <=100750 and Catalogue = [18,24]	2529.357		2.616900212		121	14.21057%
31	Salary >=100750 and Catalogue = [18,24]	3526.422		1.069607051		26	3.29025%

The excel formula should be $(C2-D\$2)/E\2 for the computation of Z values.

[Previous](#)[Next](#)

From <https://iigsawacademy.net/courses/119/pages/video/t11-dot-2-19-r-code-demo-part-3-00-13-02?module_item_id=9670>

RECAP

- Explore and prepare data for Decision Tree models
- Create a decision tree classifier
- Extract Rules from a Decision Tree Classifier
- Performance evaluation: Kappa, ROC, AUC and Confusion Matrix
- Create a decision tree regression model
- Extract rules and segment the population