Data Analytics and Business Intelligence (8696/8697)

Decision Trees

Graham.Williams@togaware.com

Data Scientist
Australian Taxation Office

Adjunct Professor, University of Canberra

Graham.Williams@togaware.com
http://datamining.togaware.com
Overview

1 Introduction

2 Decision Trees
   - Basics
   - Example
   - Algorithm

3 Building Decision Trees
   - In Rattle
   - In R
OVERVIEW

1 INTRODUCTION

2 DECISION TREES
   • Basics
   • Example
   • Algorithm

3 BUILDING DECISION TREES
   • In Rattle
   • In R
Decision Trees

Overview

1. Introduction

2. Decision Trees
   - Basics
   - Example
   - Algorithm

3. Building Decision Trees
   - In Rattle
   - In R
**Predictive Modelling: Classification**

- Goal of classification is to build *models* (sentences) in a knowledge representation (language) from examples of past decisions.

- The model is to be used on unseen cases to make decisions.

- Often referred to as supervised learning.

- Common approaches: decision trees; neural networks; logistic regression; support vector machines.
**LANGUAGE: DECISION TREES**

- Knowledge representation: A flow-chart-like tree structure
- Internal nodes denotes a test on a variable
- Branch represents an outcome of the test
- Leaf nodes represent class labels or class distribution

![Decision Tree Diagram]

- **Gender**
  - Male
  - Female
- **Age**
  - $< 42$
  - $> 42$
- **Y**
- **N**
Decision tree induction is an example of a recursive partitioning algorithm: divide and conquer.

At start, all the training examples are at the root

Partition examples recursively based on selected variables
Training Dataset: Buys Computer?

What rule would you “learn” to identify who buys a computer?

<table>
<thead>
<tr>
<th>Age</th>
<th>Income</th>
<th>Student</th>
<th>Credit</th>
<th>Buys</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 30</td>
<td>High</td>
<td>No</td>
<td>Poor</td>
<td>No</td>
</tr>
<tr>
<td>&lt; 30</td>
<td>High</td>
<td>No</td>
<td>Good</td>
<td>Yes</td>
</tr>
<tr>
<td>30 – 40</td>
<td>High</td>
<td>No</td>
<td>Poor</td>
<td>Yes</td>
</tr>
<tr>
<td>&gt; 40</td>
<td>Medium</td>
<td>No</td>
<td>Poor</td>
<td>Yes</td>
</tr>
<tr>
<td>&gt; 40</td>
<td>Low</td>
<td>Yes</td>
<td>Poor</td>
<td>Yes</td>
</tr>
<tr>
<td>&gt; 40</td>
<td>Low</td>
<td>Yes</td>
<td>Good</td>
<td>No</td>
</tr>
<tr>
<td>30 – 40</td>
<td>Low</td>
<td>Yes</td>
<td>Good</td>
<td>Yes</td>
</tr>
<tr>
<td>&lt; 30</td>
<td>Medium</td>
<td>No</td>
<td>Poor</td>
<td>No</td>
</tr>
<tr>
<td>&lt; 30</td>
<td>Low</td>
<td>Yes</td>
<td>Poor</td>
<td>No</td>
</tr>
<tr>
<td>&gt; 40</td>
<td>Medium</td>
<td>Yes</td>
<td>Poor</td>
<td>Yes</td>
</tr>
<tr>
<td>&lt; 30</td>
<td>Medium</td>
<td>Yes</td>
<td>Good</td>
<td>Yes</td>
</tr>
<tr>
<td>30 – 40</td>
<td>Medium</td>
<td>No</td>
<td>Good</td>
<td>Yes</td>
</tr>
<tr>
<td>30 – 40</td>
<td>High</td>
<td>Yes</td>
<td>Poor</td>
<td>Yes</td>
</tr>
<tr>
<td>&gt; 40</td>
<td>Medium</td>
<td>No</td>
<td>Good</td>
<td>No</td>
</tr>
<tr>
<td>&gt; 40</td>
<td>Medium</td>
<td>No</td>
<td>Good</td>
<td>No</td>
</tr>
</tbody>
</table>
Output: Decision Tree for Buys Computer

One possible tree:

```
Student
   /   
  Yes  No
 [71/29] [29/71]
```
Output: Decision Tree for Buys Computer

One possible tree:

```
Student
   Yes
   Income
      Low [50/50]
      H/M [100/0]
   No [29/71]
```

Copyright © 2014, Graham.Williams@togaware.com
**Output: Decision Tree for Buys Computer**

One possible tree:

```
Student
  Yes
  Income
    Low
    Age
      < 30 [0/100]
      30-40, >40 [100/0]
  No [29/71]
    Income
      H/M
      Age
      < 30 [67/33] [100/0]
```
Output: Decision Tree for Buys Computer

One possible tree:
One possible tree:
Algorithm for Decision Tree Induction

- A greedy algorithm: takes the best immediate (local) decision while building the overall model
- Tree constructed top-down, recursive, divide-and-conquer
- Begin with all training examples at the root
- Data is partitioned recursively based on selected variables
- Select variables on basis of a measure
- Stop partitioning when:
  - All samples for a given node belong to the same class
  - There are no remaining variables for further partitioning – majority voting is employed for classifying the leaf
  - There are no samples left
Basic Motivation: Entropy

We are trying to predict output $Y$ (e.g., Yes/No) from input $X$.

- A random data set may have high entropy:
  - $Y$ is from a uniform distribution
  - a frequency distribution would be flat!
  - a sample will include uniformly random values of $Y$

- A data set with low entropy:
  - $Y$’s distribution will be very skewed
  - a frequency distribution will have a single peak
  - a sample will predominately contain just Yes or just No

- Work towards reducing the amount of entropy in the data!
Basic Motivation: Entropy

We are trying to predict output $Y$ (e.g., Yes/No) from input $X$.

- A random data set may have high entropy:
  - $Y$ is from a uniform distribution
  - a frequency distribution would be flat!
  - a sample will include uniformly random values of $Y$

- A data set with low entropy:
  - $Y$’s distribution will be very skewed
  - a frequency distribution will have a single peak
  - a sample will predominately contain just Yes or just No

- Work towards reducing the amount of entropy in the data!
Basic Motivation: Entropy

We are trying to predict output $Y$ (e.g., Yes/No) from input $X$.

- A random data set may have high entropy:
  - $Y$ is from a uniform distribution
  - a frequency distribution would be flat!
  - a sample will include uniformly random values of $Y$

- A data set with low entropy:
  - $Y$’s distribution will be very skewed
  - a frequency distribution will have a single peak
  - a sample will predominately contain just Yes or just No

- Work towards reducing the amount of entropy in the data!
Basic Motivation: Entropy

We are trying to predict output $Y$ (e.g., Yes/No) from input $X$.

- A random data set may have high entropy:
  - $Y$ is from a uniform distribution
  - a frequency distribution would be flat!
  - a sample will include uniformly random values of $Y$

- A data set with low entropy:
  - $Y$’s distribution will be very skewed
  - a frequency distribution will have a single peak
  - a sample will predominately contain just Yes or just No

- Work towards reducing the amount of entropy in the data!
**Entropy**

We are trying to predict output $Y$ from input $X$.

$X = \text{Course}$

$Y = \text{Purchase Neo1973}$

<table>
<thead>
<tr>
<th>$X$</th>
<th>$Y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Math</td>
<td>Yes</td>
</tr>
<tr>
<td>History</td>
<td>No</td>
</tr>
<tr>
<td>CS</td>
<td>Yes</td>
</tr>
<tr>
<td>Math</td>
<td>No</td>
</tr>
<tr>
<td>Math</td>
<td>No</td>
</tr>
<tr>
<td>CS</td>
<td>Yes</td>
</tr>
<tr>
<td>History</td>
<td>No</td>
</tr>
<tr>
<td>Math</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Assuming this represents true probabilities:

$P(Yes) = 0.5$

$P(Math) = 0.5$

$P(Math \& Yes) = 0.25$

$P(\text{History} \& Yes) = 0$

http://www.cs.cmu.edu/~awm/tutorials
We are trying to predict output $Y$ from input $X$.

$X = \text{Course}$  
$Y = \text{Purchase Neo1973}$

Assuming this represents true probabilities:

$P(\text{Yes}) = 0.5$  
$P(\text{Math}) = 0.5$  
$P(\text{Math} \& \text{Yes}) = 0.25$  
$P(\text{History} \& \text{Yes}) = 0$

<table>
<thead>
<tr>
<th>$X$</th>
<th>$Y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Math</td>
<td>Yes</td>
</tr>
<tr>
<td>History</td>
<td>No</td>
</tr>
<tr>
<td>CS</td>
<td>Yes</td>
</tr>
<tr>
<td>Math</td>
<td>No</td>
</tr>
<tr>
<td>Math</td>
<td>No</td>
</tr>
<tr>
<td>CS</td>
<td>Yes</td>
</tr>
<tr>
<td>History</td>
<td>No</td>
</tr>
<tr>
<td>Math</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Entropy

We are trying to predict output $Y$ from input $X$.

$X = \text{Course}$
$Y = \text{Purchase Neo1973}$

<table>
<thead>
<tr>
<th>$X$</th>
<th>$Y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Math</td>
<td>Yes</td>
</tr>
<tr>
<td>History</td>
<td>No</td>
</tr>
<tr>
<td>CS</td>
<td>Yes</td>
</tr>
<tr>
<td>Math</td>
<td>No</td>
</tr>
<tr>
<td>Math</td>
<td>No</td>
</tr>
<tr>
<td>CS</td>
<td>Yes</td>
</tr>
<tr>
<td>History</td>
<td>No</td>
</tr>
<tr>
<td>Math</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Focus on $Y$

$P(\text{Yes}) = 0.5$
$P(\text{No}) = 0.5$

Uniform distribution of $Y$

Entropy of $Y$ is 1

$$E(p, n) = - \frac{p}{p+n} \log_2 \frac{p}{p+n} \quad - \frac{n}{p+n} \log_2 \frac{n}{p+n}$$

$log_2(0.5) = -1$
**Entropy**

We are trying to predict output $Y$ from input $X$.

$X = \text{Course}$  
$Y = \text{Purchase Neo1973}$

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>Math</td>
<td>Yes</td>
</tr>
<tr>
<td>History</td>
<td>No</td>
</tr>
<tr>
<td>CS</td>
<td>Yes</td>
</tr>
<tr>
<td>Math</td>
<td>No</td>
</tr>
<tr>
<td>Math</td>
<td>No</td>
</tr>
<tr>
<td>CS</td>
<td>Yes</td>
</tr>
<tr>
<td>History</td>
<td>No</td>
</tr>
<tr>
<td>Math</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Focus on $Y$

$P(\text{Yes}) = 0.5$  
$P(\text{No}) = 0.5$

Uniform distribution of $Y$  
Entropy of $Y$ is 1

\[
E(p, n) = -\frac{p}{p+n} \log_2 \frac{p}{p+n} - \frac{n}{p+n} \log_2 \frac{n}{p+n}
\]

$log_2(0.5) = -1$

http://www.cs.cmu.edu/~awm/tutorials
## Entropy

We are trying to predict output $Y$ from input $X$.

$X = \text{Course}$  
$Y = \text{Purchase Neo1973}$

<table>
<thead>
<tr>
<th>$X$</th>
<th>$Y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Math</td>
<td>Yes</td>
</tr>
<tr>
<td>History</td>
<td>No</td>
</tr>
<tr>
<td>CS</td>
<td>Yes</td>
</tr>
<tr>
<td>Math</td>
<td>No</td>
</tr>
<tr>
<td>Math</td>
<td>No</td>
</tr>
<tr>
<td>CS</td>
<td>Yes</td>
</tr>
<tr>
<td>History</td>
<td>No</td>
</tr>
<tr>
<td>Math</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Focus on $Y$

$P(\text{Yes}) = 0.5$

$P(\text{No}) = 0.5$

Uniform distribution of $Y$

Entropy of $Y$ is 1

$$E(p, n) = -\frac{p}{p+n} \log_2 \frac{p}{p+n} - \frac{n}{p+n} \log_2 \frac{n}{p+n}$$

$log_2(0.5) = -1$

http://www.cs.cmu.edu/~awm/tutorials
**Entropy**

We are trying to predict output $Y$ from input $X$.

$X = \text{Course}$

$Y = \text{Purchase Neo1973}$

<table>
<thead>
<tr>
<th>$X$</th>
<th>$Y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Math</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>History</strong></td>
<td>No</td>
</tr>
<tr>
<td>CS</td>
<td>Yes</td>
</tr>
<tr>
<td>Math</td>
<td>No</td>
</tr>
<tr>
<td>Math</td>
<td>No</td>
</tr>
<tr>
<td>CS</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>History</strong></td>
<td>No</td>
</tr>
<tr>
<td>Math</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Focus on just students of History

$P(\text{Yes}) = 0$

$P(\text{No}) = 1$

Skewed distribution of $Y$

Entropy of $Y$ is 0

$$E(p, n) = -\frac{0}{0+2} \log_2 \frac{0}{0+2} - \frac{2}{0+2} \log_2 \frac{2}{0+2}$$

$$\log_2(0) = -\infty \quad \log_2(1) = 0$$

http://www.cs.cmu.edu/~awm/tutorials
Entrophy

We are trying to predict output $Y$ from input $X$.

$X = \text{Course}$

$Y = \text{Purchase Neo1973}$

<table>
<thead>
<tr>
<th>$X$</th>
<th>$Y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Math</td>
<td>Yes</td>
</tr>
<tr>
<td>History</td>
<td>No</td>
</tr>
<tr>
<td>CS</td>
<td>Yes</td>
</tr>
<tr>
<td>Math</td>
<td>No</td>
</tr>
<tr>
<td>Math</td>
<td>No</td>
</tr>
<tr>
<td>CS</td>
<td>Yes</td>
</tr>
<tr>
<td>History</td>
<td>No</td>
</tr>
<tr>
<td>Math</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Focus on just students of History

$P(\text{Yes}) = 0$

$P(\text{No}) = 1$

Skewed distribution of $Y$

Entropy of $Y$ is 0

$$E(p, n) = \frac{-0}{0+2} \log_2 \frac{0}{0+2} - \frac{2}{0+2} \log_2 \frac{2}{0+2}$$

$\log_2(0) = -\infty$, $\log_2(1) = 0$
Entropy

We are trying to predict output $Y$ from input $X$.

$X = \text{Course}$

$Y = \text{Purchase Neo1973}$

<table>
<thead>
<tr>
<th>$X$</th>
<th>$Y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Math</td>
<td>Yes</td>
</tr>
<tr>
<td>History</td>
<td>No</td>
</tr>
<tr>
<td>CS</td>
<td>Yes</td>
</tr>
<tr>
<td>Math</td>
<td>No</td>
</tr>
<tr>
<td>Math</td>
<td>No</td>
</tr>
<tr>
<td>CS</td>
<td>Yes</td>
</tr>
<tr>
<td>History</td>
<td>No</td>
</tr>
<tr>
<td>Math</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Focus on just students of History

$P(\text{Yes}) = 0$

$P(\text{No}) = 1$

Skewed distribution of $Y$

Entropy of $Y$ is 0

$$E(p, n) = -\frac{0}{0+2} \log_2 \frac{0}{0+2} - \frac{2}{0+2} \log_2 \frac{2}{0+2}$$

$log_2(0) = -\infty \, \log_2(1) = 0$

http://www.cs.cmu.edu/~awm/tutorials
**Variable Selection Measure: Entropy**

- Information gain (ID3/C4.5)
- Select the variable with the highest information gain
- Assume there are two classes: \( P \) and \( N \)
- Let the data \( S \) contain \( p \) elements of class \( P \) and \( n \) elements of class \( N \)
- The amount of information, needed to decide if an arbitrary example in \( S \) belongs to \( P \) or \( N \) is defined as

\[
I_E(p, n) = - \frac{p}{p+n} \log_2 \frac{p}{p+n} - \frac{n}{p+n} \log_2 \frac{n}{p+n}
\]
Variable Selection Measure: Gini

- Gini index of impurity – traditional statistical measure – CART
- Measure how often a randomly chosen observation is incorrectly classified if it were randomly classified in proportion to the actual classes.
- Calculated as the sum of the probability of each observation being chosen times the probability of incorrect classification, equivalently:

\[ I_G(p, n) = 1 - (p^2 + (1 - p)^2) \]

- As with Entropy, the Gini measure is maximal when the classes are equally distributed and minimal when all observations are in one class or the other.
**Variable Selection Measure**

The graph illustrates the variable importance measure as a function of the proportion of positives. The measure is highest when the proportion of positives is between 0.5 and 0.75, indicating that variables are most important in this range. The measure decreases as the proportion of positives deviates from this range, reaching its minimum at both 0.0 and 1.0. The curve for the Gini measure is shown in red, while the curve for the Info measure is shown in blue.

**Formula**
- **Info**
- **Gini**

- **Proportion of Positives**
- **Measure**

**Variable Importance Measure**
Information Gain

Now use variable $A$ to partition $S$ into $v$ cells: \( \{S_1, S_2, \ldots, S_v\} \)

If $S_i$ contains $p_i$ examples of $P$ and $n_i$ examples of $N$, the
information now needed to classify objects in all subtrees $S_i$ is:

\[
E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p + n} I(p_i, n_i)
\]

So, the information gained by branching on $A$ is:

\[
Gain(A) = I(p, n) - E(A)
\]

So choose the variable $A$ which results in the greatest gain in
information.
OVERVIEW

1. Introduction

2. Decision Trees
   - Basics
   - Example
   - Algorithm

3. Building Decision Trees
   - In Rattle
   - In R
library(rattle)
rattle()
Load Example Weather Dataset

- Click on the Execute button and an example dataset is offered.
- Click on Yes to load the weather dataset.
Summary of the Weather Dataset

- A summary of the weather dataset is displayed.
Model Tab — Decision Tree

- Click on the Model tab to display the modelling options.
**Build Tree to Predict RainTomorrow**

- Decision Tree is the default model type—simply click Execute.
Decision Tree Predicting RainTomorrow

- Click the Draw button to display a tree (Settings → Advanced Graphics).
**Evaluate Decision Tree**

- Click Evaluate tab—options to evaluate model performance.
Evaluate Decision Tree—Error Matrix

- Click Execute to display simple error matrix.
- Identify the True/False Positives/Negatives.

![Error Matrix](R Data Miner - [Rattle (weather.csv)])

Error matrix for the Decision Tree model on weather.csv [validate] (counts):

```
  Predicted
  Actual No Yes
     No 39  5
     Yes 5  5
```

Overall error: 0.1851852

Rattle timestamp: 2013-07-03 28:37:37 gjw
**Decision Tree Risk Chart**

- Click the Risk type and then Execute.
Decision Tree ROC Curve

- Click the ROC type and then Execute.
Click the Score type to score a new dataset using model.

A model can be deployed on a dataset to obtain scores or classifications for each observation in the dataset.

By default the testing dataset (if any) will be scored. Otherwise the training dataset is scored. As an alternative, a CSV file can be loaded and scored. This choice of what is scored is controlled by the radio button options.

For binary models a probability score can be recorded. For regression models a value is recorded for each observation. Otherwise a class will be recorded for each observation. This can be controlled by the Class and Probability radio buttons.

The resulting CSV file will include just those variables having a role as Identifier (plus the Target and the Score), or else all of the variables.

The name of a CSV file into which the results will be written will be prompted for.
Log of R Commands

- Click the Log tab for a history of all your interactions.
- Save the log contents as a script to repeat what we did.
Log of R Commands—rpart()

- Here we see the call to rpart() to build the model.
- Click on the Export button to save the script to file.
Rattle provides some basic help—click Yes for R help.

A decision tree is the prototypical data mining tool, used widely for its ease of interpretation. It consists of a root node split by a single variable into two partitions. In turn, these two new partitions become new nodes that may then each be further split on a single (and usually different) variable. This divide and conquering continues until no further splitting would improve the performance of the model.

While a choice of measures are available to select a variable to split the dataset on, the Gini measure is used, and generally is no different to the information measure for binary classification. To explore the alternatives, copy the relevant code from the Log and paste it into the R Console and change any of the options.

Common options that a user may change from their default values are available.

Priors: used to boost a particularly important class, by giving it a higher prior probability. Expects a list of numbers that sum up to 1, and of the same length as the number of classes in the training dataset: e.g., 0.5, 0.5.

Loss Matrix: used to weight the outcome classes differently. e.g., 0, 0.1, 1, 0.

Other options exist, but are not usually required. For example, 10-fold cross validation, used in deciding how to prune to the best decision tree, is generally regarded as the right number. Transferring the commands from the Log tab into the R Console does give you full access to all options.

Decision trees work with both numeric and categoric data.

The rpart package is used to build the decision tree.

Would you like to view the R help?
OVERVIEW

1 INTRODUCTION

2 DECISION TREES
   • Basics
   • Example
   • Algorithm

3 BUILDING DECISION TREES
   • In Rattle
   • In R
ds <- weather
head(ds, 4)

## Date Location MinTemp MaxTemp Rainfall Evaporation Sunshine
## 1 2007-11-01 Canberra 8.0 24.3 0.0 3.4 6.3
## 2 2007-11-02 Canberra 14.0 26.9 3.6 4.4 9.7
## 3 2007-11-03 Canberra 13.7 23.4 3.6 5.8 3.3
## 4 2007-11-04 Canberra 13.3 15.5 39.8 7.2 9.1

summary(ds[, c(3:5, 23)])

## MinTemp MaxTemp Rainfall RISK_MM
## Min. :-5.30 Min. : 7.6 Min. : 0.00 Min. : 0.00
## 1st Qu.: 2.30 1st Qu.:15.0 1st Qu.: 0.00 1st Qu.: 0.00
## Median : 7.45 Median :19.6 Median : 0.00 Median : 0.00
## Mean : 7.27 Mean :20.6 Mean : 1.43 Mean : 1.43

....

http://togaware.com
Building Decision Trees

### Weather Dataset - Target

target <- "RainTomorrow"

summary(ds[target])

```r
## RainTomorrow
## No : 300
## Yes: 66
```

(form <- formula(paste(target, "~ ")))

```r
## RainTomorrow ~ 
```

(vars <- names(ds)[-c(1, 2, 23)])

```r
## [1] "MinTemp"     "MaxTemp"     "Rainfall"     "Evaporation"
## [5] "Sunshine"   "WindGustDir" "WindGustSpeed" "WindDir9am"
## [9] "WindDir3pm" "WindSpeed9am" "WindSpeed3pm" "Humidity9am"
## [13] "Humidity3pm" "Pressure9am" "Pressure3pm" "Cloud9am"
## [17] "Cloud3pm"   "Temp9am"     "Temp3pm"     "RainToday"
## [21] "RainTomorrow"
```
**Simple Train/Test Paradigm**

```r
set.seed(1421)
train <- c(sample(1:nrow(ds), 0.70*nrow(ds)))  # Training dataset
head(train)

## [1] 288 298 363 107  70  232

length(train)

## [1] 256

test <- setdiff(1:nrow(ds), train)  # Testing dataset
length(test)

## [1] 110
```
Display the Model

```r
model <- rpart(form, ds[train, vars])
model

## n= 256
##
## node), split, n, loss, yval, (yprob)
## * denotes terminal node
##
## 1) root 256 44 No (0.82812 0.17188)
##   2) Humidity3pm< 59.5 214 21 No (0.90187 0.09813)
##     4) WindGustSpeed< 64 204 14 No (0.93137 0.06863)
##     8) Cloud3pm< 6.5 163 5 No (0.96933 0.03067) *
##     9) Cloud3pm>=6.5 41 9 No (0.78049 0.21951)
##   18) Temp3pm< 26.1 34 4 No (0.88235 0.11765) *
##   19) Temp3pm>=26.1 7 2 Yes (0.28571 0.71429) *
....
```

• Notice the legend to help interpret the tree.
Performance on Test Dataset

- The `predict()` function is used to score new data.

```r
head(predict(model, ds[test,], type="class"))
## 2 4 6 8 11 12
## No No No No No No
## Levels: No Yes

table(predict(model, ds[test,], type="class"), ds[test, target])
##       No Yes
## No    77  14
## Yes   11   8
```
Example DTREE Plot using Rattle

Building Decision Trees In R
An R Scripting Hint

- Notice the use of variables `ds, target, vars`.
- Change these variables, and the remaining script is unchanged.
- Simplifies script writing and reuse of scripts.

```r
ds <- iris
target <- "Species"
vars <- names(ds)
```

- Then repeat the rest of the script, without change.
An R Scripting Hint — Unchanged Code

- This code remains the same to build the decision tree.

```r
form <- formula(paste(target, "~ ."))
train <- c(sample(1:nrow(ds), 0.70*nrow(ds)))
test <- setdiff(1:nrow(ds), train)
model <- rpart(form, ds[train, vars])
model

## n= 105
##
## node), split, n, loss, yval, (yprob)
## * denotes terminal node
##
## 1) root 105 69 setosa (0.34286 0.32381 0.33333)
## 2) Petal.Length< 2.6 36 0 setosa (1.00000 0.00000 0.00000) *
## 3) Petal.Length>=2.6 69 34 virginica (0.00000 0.49275 0.50725)
## 6) Petal.Length< 4.95 35 2 versicolor (0.00000 0.94286 0.05714) *
## 7) Petal.Length>=4.95 34 1 virginica (0.00000 0.02941 0.97059) *
```
An R Scripting Hint — Unchanged Code

- Similarly for the predictions.

```r
head(predict(model, ds[test,], type="class"))
```

```r
## 3 8 9 10 11 12
## setosa setosa setosa setosa setosa setosa
## Levels: setosa versicolor virginica
```

```r
table(predict(model, ds[test,], type="class"), ds[test, target])
```

```r
## setosa versicolor virginica
## setosa 14 0 0
## versicolor 0 15 4
## virginica 0 1 11
```
Modelling Framework

Language  Tree with single variable tests

Measure  Entropy, Gini, ...

Search  Recursive partitioning
Summary

Decision Tree Induction.

Most widely deployed machine learning algorithm.

Simple idea, powerful learner.

Available in R through the rpart package.

Related packages include party, Cubist, C50, RWeka (J48).
Data Mining with Rattle and R
Graham Williams
2011, Springer, Use R!
Chapter 11.
Data Mining with Rattle and R

Graham Williams

2011, Springer, Use R!


Chapter 11.