Hands-On Data Science with R
From Rattle to R

Graham.Williams@togaware.com

24th August 2014

Visit http://HandsOnDataScience.com/ for more Chapters.

Rattle (Williams, 2011), the R Analytic Tool To Learn Easily, is a simple to use graphical user interface for data mining using the statistical language R (R Core Team, 2014). Rattle is a freely available package that interfaces to over 100 other R packages that support the Data Scientist. It is one place to start in our journey to Data Science, but generally not the end point.

Rattle's user interface steps through the data mining tasks, recording the actual R code as it goes. The R code can be saved to file and used as an automatic script, loaded into R (outside of Rattle) to repeat the data mining exercise, and to learn how to interact with R itself. Repeatability is important both in science and in commerce and so the fact that Rattle records everything it does as a script was a design principle from the start.

In this chapter we begin with a brief overview of Rattle, with a focus on the underlying R code, and then discover how to extend the capabilities of Rattle. We explore extensions that allow us to enhance the significant capabilities of Rattle with the full power of R.

The required packages for this module include:

```
library(rattle)        # The Rattle GUI for Data Mining.
```

As we work through this chapter, new R commands will be introduced. Be sure to review the command's documentation and understand what the command does. You can ask for help using the ? command as in:

```
?read.csv
```

We can obtain documentation on a particular package using the *help=* option of `library()`:

```
library(help=rattle)
```

This chapter is intended to be hands on. To learn effectively, you are encouraged to have R running (e.g., RStudio) and to run all the commands as they appear here. Check that you get the same output, and you understand the output. Try some variations. Explore.

1 Installing R, RStudio, Rattle

R is the statistical software suite and programming language of choice for the Data Scientist. RStudio is recommended as a modern integrated development environment for working with R whilst ESS (Emacs Speaks Statistics) mode in Emacs has been a favourite for very many years. Finally, Rattle can then be initiated from within the basic R console, or from the Console within RStudio or ESS.

All of this software is free (as in libre) and open source software, often referred to as FOSS, or FLOSS by adding in the French word libre to emphasise the point that "Free software is a matter of liberty, not price. To understand the concept, you should think of free as in free speech, not as in free beer." (Richard Stallman of GNU.org). We recommend installing it on a free and open source operating system, such as Ubuntu. R, RStudio, and Rattle all run perfectly well on multiple GNU/Linux, Apple Macintosh OS/X, and Microsoft Windows operating systems.

The FLOSS stack offers the best experience enhanced by the most powerful suite of tools that come by default with the GNU suite that has been under development since the 1970's. It is also clearly the most cost effective Data Scientist's toolkit, installing R, RStudio and Rattle on the Ubuntu distribution of the GNU/Linux operating system. This provides a complete free and open source environment.

Ubuntu can easily be installed to take the place of, or side-by-side with (to dual boot), OS/X or Windows on virtually any computer. Alternatively, using Oracle's VirtualBox we can run Ubuntu inside of either OS/X or Windows simply as another application.

Then R itself needs to be installed on your system and instructions to do so are available from the R Project. Once R is installed we simply install other packages using `install.packages()`:

```
install.packages("rattle")
```

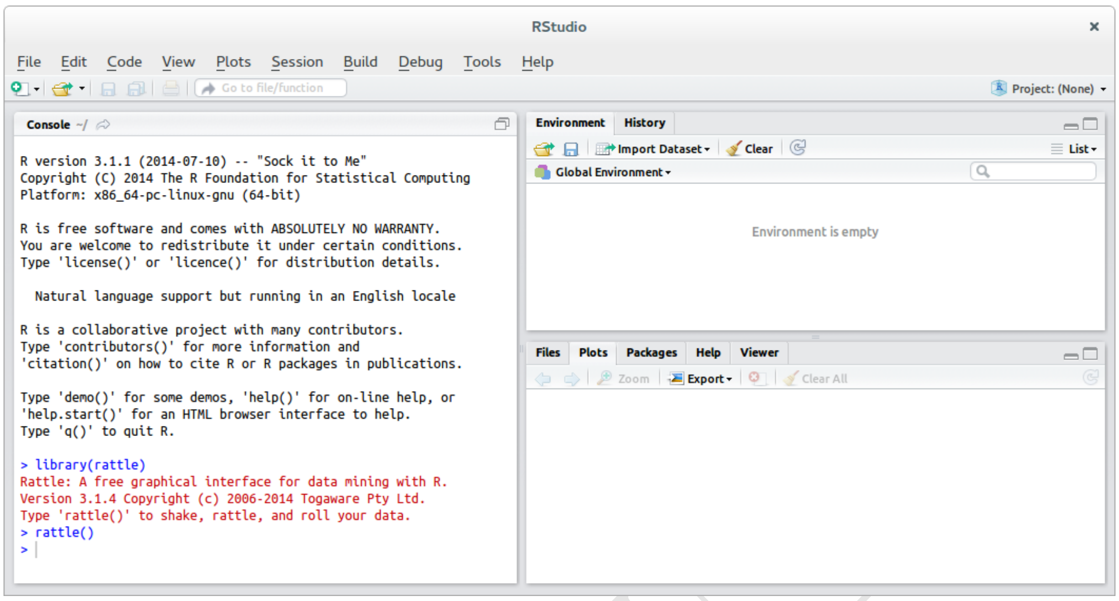
Then the `library()` command below will load the package and the start up Rattle:

```
library(rattle)
rattle()
```

Under Ubuntu, all that is required to install R and rattle is:

```
$ wajig install r-recommended r-cran-rattle
```

2 Starting Rattle



```
R version 3.1.1 (2014-07-10) -- "Sock it to Me"
Copyright (C) 2014 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> library(rattle)
Rattle: A free graphical interface for data mining with R.
Version 3.1.4 Copyright (c) 2006-2014 Togaware Pty Ltd.
Type 'rattle()' to shake, rattle, and roll your data.
> rattle()
> 
```

Rattle is started from R. There are several ways that Rattle might be configured for your particular computer. For example, some installations set up an icon on the desktop from which Rattle is automatically invoked. The most common way though is to start up Rattle from within R.

Even starting up R depends on your particular platform. Generally, it is started from a desktop icon or from the Application menu. Alternatively, on Linux it is often started up from a terminal window, like gnome-terminal or xterm. From the terminal we simply type the command R to invoke R itself.

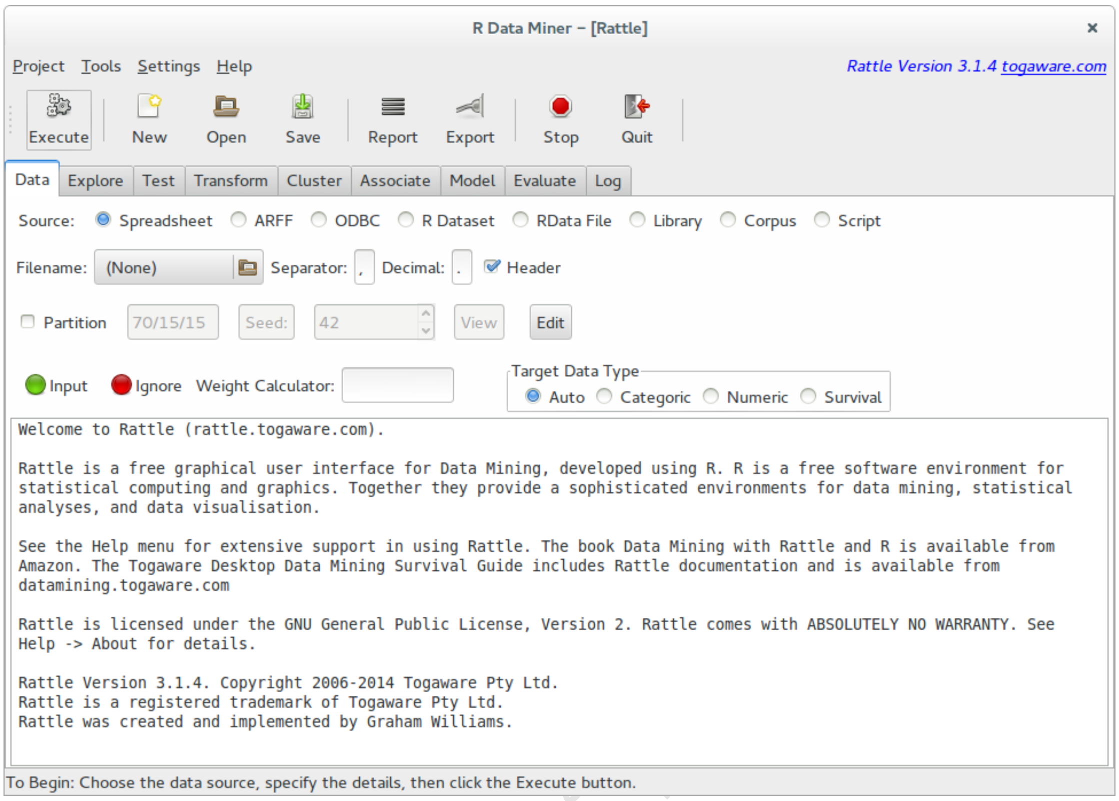
An increasingly popular approach is to use RStudio. RStudio includes an R console. We can see the RStudio application below, with the commands to start up Rattle. Do note that this only works with the Desktop version of RStudio and not the server version of RStudio. The server version runs the interface in a browser on your desktop and communicates to a remote server running R itself. RStudio handles all of the graphical interface. Because Rattle has its own graphical interface, RStudio is unable to capture that interface from the server and display it on your desktop.

We can access the desktop version of RStudio from a server by running an XWindows server, such as xming, on our desktop.

Whichever way we start R, we initiate Rattle with:

```
library(rattle)
rattle()
```

3 Getting Familiar With Rattle


R Data Miner – [Rattle]
Project Tools Settings Help
Rattle Version 3.1.4 togaware.com
Execute New Open Save Report Export Stop Quit
Data Explore Test Transform Cluster Associate Model Evaluate Log
Source: Spreadsheet ARFF ODBC R Dataset RData File Library Corpus Script
Filename: (None) Separator: , Decimal: . Header
Partition 70/15/15 Seed: 42 View Edit
Input Ignore Weight Calculator:
Target Data Type
Auto Categoric Numeric Survival

```
Welcome to Rattle (rattle.togaware.com).

Rattle is a free graphical user interface for Data Mining, developed using R. R is a free software environment for
statistical computing and graphics. Together they provide a sophisticated environments for data mining, statistical
analyses, and data visualisation.

See the Help menu for extensive support in using Rattle. The book Data Mining with Rattle and R is available from
Amazon. The Togaware Desktop Data Mining Survival Guide includes Rattle documentation and is available from
datamining.togaware.com

Rattle is licensed under the GNU General Public License, Version 2. Rattle comes with ABSOLUTELY NO WARRANTY. See
Help -> About for details.

Rattle Version 3.1.4. Copyright 2006-2014 Togaware Pty Ltd.
Rattle is a registered trademark of Togaware Pty Ltd.
Rattle was created and implemented by Graham Williams.
```

To Begin: Choose the data source, specify the details, then click the Execute button.

The Rattle interface is based on a set of tabs through which we proceed, left to right. For any tab, once we have set up the required information, we **must** click the Execute button (or F2) to perform the actions. Take a moment to explore the interface a little by clicking through the various tabs. Notice the Help menu and find that the help layout mimics the tab layout.

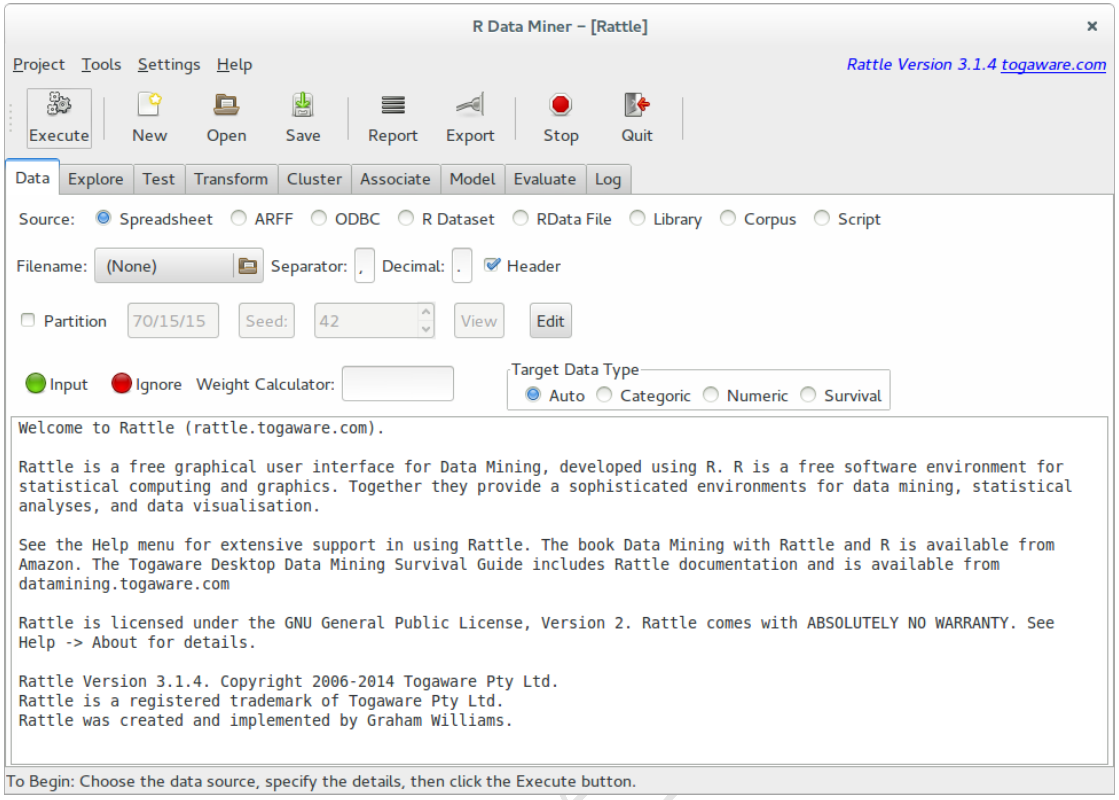
To Quit from Rattle we simply click on the Quit button in the main Rattle window.

To Quit from RStudio we choose Quit from the File menu.

If we are using a terminal to run R then we can press 'Ctrl-D' (i.e. press the 'Control' key and then the 'D' key together).

In most cases we are asked whether to save our workspace. For now (and indeed for most users) we **do not** save the workspace.

4 The Initial Interface


R Data Miner – [Rattle]
Project Tools Settings Help
Rattle Version 3.1.4 togaware.com
Execute New Open Save Report Export Stop Quit
Data Explore Test Transform Cluster Associate Model Evaluate Log
Input Ignore

Source: [✓ Spreadsheet] ARFF ODBC R Dataset RData File Library Corpus Script

Filename: (None) Separator: , Decimal: . [✓ Header]

Partition 70/15/15 Seed: 42 View Edit

Weight Calculator:

Target Data Type: [✓ Auto] Categoric Numeric Survival

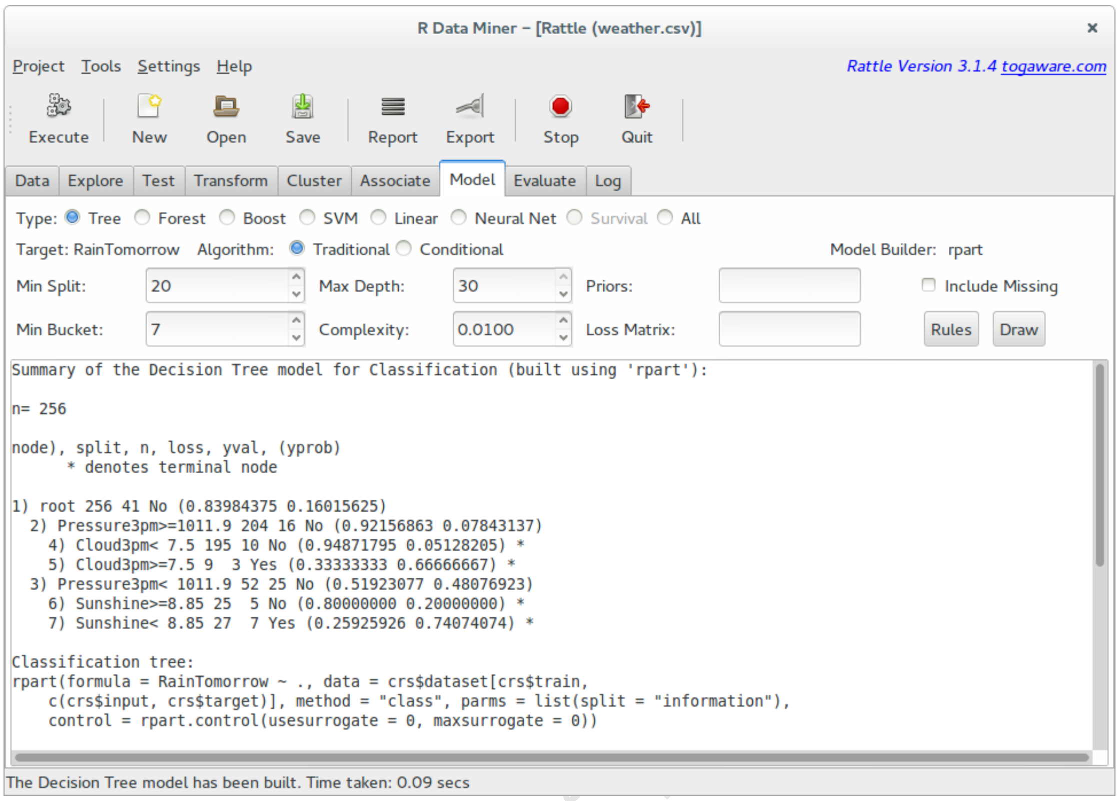
```
Welcome to Rattle (rattle.togaware.com).

Rattle is a free graphical user interface for Data Mining, developed using R. R is a free software environment for
statistical computing and graphics. Together they provide a sophisticated environments for data mining, statistical
analyses, and data visualisation.

See the Help menu for extensive support in using Rattle. The book Data Mining with Rattle and R is available from
Amazon. The Togaware Desktop Data Mining Survival Guide includes Rattle documentation and is available from
datamining.togaware.com

Rattle is licensed under the GNU General Public License, Version 2. Rattle comes with ABSOLUTELY NO WARRANTY. See
Help -> About for details.

Rattle Version 3.1.4. Copyright 2006-2014 Togaware Pty Ltd.
Rattle is a registered trademark of Togaware Pty Ltd.
Rattle was created and implemented by Graham Williams.
```

To Begin: Choose the data source, specify the details, then click the Execute button.

The process that we implement in Rattle and that is reflected in the tabs that we see in the Rattle interface is:

1. Load a **Data**set;
2. Select **Variables** for exploring and mining;
3. **Sample** the data into training and test datasets;
4. **Explore** the distributions of the data;
5. Perhaps **Test** some of the distributions;
6. Optionally **Transform** our data;
7. Build **Cluster**s or **Association Rules** from the data;
8. Build predictive **Model**s;
9. **Evaluate** the models;
10. Record the steps in building your model as listed in the **Log**.

5 Load Data, Build Model


R Data Miner – [Rattle (weather.csv)]
Project Tools Settings Help
Rattle Version 3.1.4 togaware.com
Execute New Open Save Report Export Stop Quit
Data Explore Test Transform Cluster Associate Model Evaluate Log
Rules Draw

Type: [✓ Tree] Forest Boost SVM Linear Neural Net Survival All
Target: RainTomorrow Algorithm: [✓ Traditional] Conditional Model Builder: rpart
Min Split: 20 Max Depth: 30 Priors: Include Missing
Min Bucket: 7 Complexity: 0.0100 Loss Matrix:

```
Summary of the Decision Tree model for Classification (built using 'rpart'):

n= 256

node), split, n, loss, yval, (yprob)
      * denotes terminal node

1) root 256 41 No (0.83984375 0.16015625)
  2) Pressure3pm>=1011.9 204 16 No (0.92156863 0.07843137)
    4) Cloud3pm< 7.5 195 10 No (0.94871795 0.05128205) *
    5) Cloud3pm>=7.5 9  3 Yes (0.33333333 0.66666667) *
  3) Pressure3pm< 1011.9 52 25 No (0.51923077 0.48076923)
    6) Sunshine>=8.85 25  5 No (0.80000000 0.20000000) *
    7) Sunshine< 8.85 27  7 Yes (0.25925926 0.74074074) *

Classification tree:
rpart(formula = RainTomorrow ~ ., data = crs$dataset[crs$train,
    c(crs$input, crs$target)], method = "class", parms = list(split = "information"),
    control = rpart.control(usesurrogate = 0, maxsurrogate = 0))
```

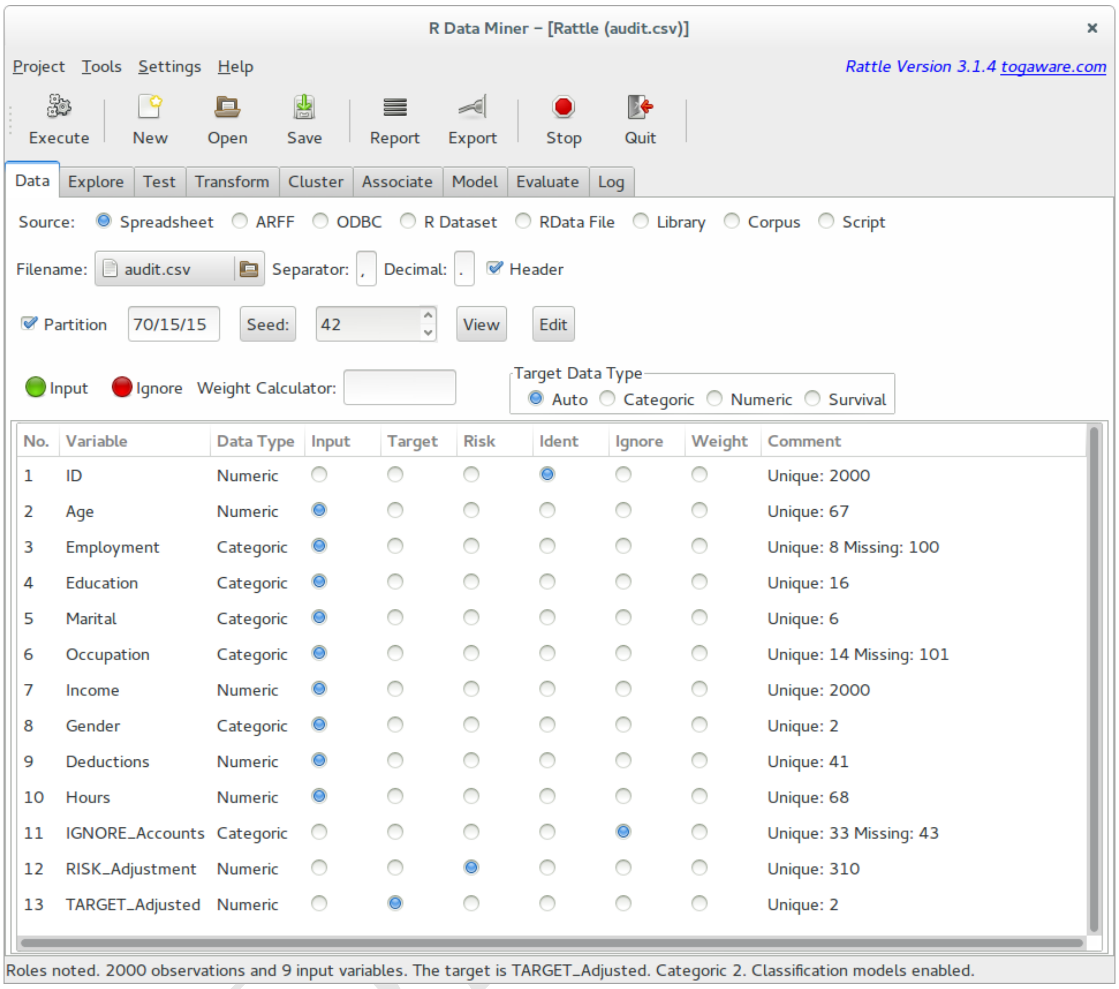
The Decision Tree model has been built. Time taken: 0.09 secs

Our first familiarisation task is to load the sample weather dataset supplied with Rattle and build a simple model.

1. Start up Rattle.
2. Click the Execute button.
3. Answer Yes to load the example weather dataset.
4. Click the Model tab.
5. Click the Execute button.
6. Click the Draw button.

This is our very first model. It is a decision tree model and can be used to predict the probability that it will rain in Canberra (Australia) tomorrow, given today's conditions in Canberra.

6 Audit: Load Dataset


R Data Miner – [Rattle (audit.csv)]
Project Tools Settings Help
Rattle Version 3.1.4 togaware.com
Execute New Open Save Report Export Stop Quit
Data Explore Test Transform Cluster Associate Model Evaluate Log

Source: [✓ Spreadsheet] ARFF ODBC R Dataset RData File Library Corpus Script

Filename: audit.csv Separator: , Decimal: . [✓ Header]

[✓ Partition] 70/15/15 Seed: 42 View Edit

🟢 Input 🔴 Ignore Weight Calculator:

Target Data Type: [✓ Auto] Categoric Numeric Survival

No.	Variable	Data Type	Input	Target	Risk	Ident	Ignore	Weight	Comment
1	ID	Numeric	○	○	○	●	○	○	Unique: 2000
2	Age	Numeric	●	○	○	○	○	○	Unique: 67
3	Employment	Categoric	●	○	○	○	○	○	Unique: 8 Missing: 100
4	Education	Categoric	●	○	○	○	○	○	Unique: 16
5	Marital	Categoric	●	○	○	○	○	○	Unique: 6
6	Occupation	Categoric	●	○	○	○	○	○	Unique: 14 Missing: 101
7	Income	Numeric	●	○	○	○	○	○	Unique: 2000
8	Gender	Categoric	●	○	○	○	○	○	Unique: 2
9	Deductions	Numeric	●	○	○	○	○	○	Unique: 41
10	Hours	Numeric	●	○	○	○	○	○	Unique: 68
11	IGNORE_Accounts	Categoric	○	○	○	○	●	○	Unique: 33 Missing: 43
12	RISK_Adjustment	Numeric	○	○	●	○	○	○	Unique: 310
13	TARGET_Adjusted	Numeric	○	●	○	○	○	○	Unique: 2

Roles noted. 2000 observations and 9 input variables. The target is TARGET_Adjusted. Categoric 2. Classification models enabled.

We now switch to the sample Audit dataset provided with rattle (Williams, 2014).

1. Click the Data tab.
2. Click the Filename: button where weather.csv is currently listed.
3. Choose the audit.csv file to load
4. Load the file into Rattle.

Be sure to investigate what the audit dataset is about, and the meaning of each of the variables. You should document this.

7 Audit: Explore

Switching to the Explore tab investigate for any interesting patterns in the data. In particular, consider at least the following options.

1. Various summaries, noting any skewness or high values of kurtosis.
2. Anything interesting about missing values?
3. What does the cross tabulation suggest, if anything?
4. Various distribution plots including Benford's Law.
5. Any correlation between variables?

8 Audit: Test

The Test tab provides the opportunity to test out statistical hypotheses.

9 Audit: Transform

10 Audit: Cluster

11 Audit: Associate

12 Audit: Predictive Model

Exercise: Draw a tree and plot the evaluation.

13 Model — Multiple Models of the Same Type

Rattle can build multiple models of different types in a single run, but not variations of a model type. Here we illustrate in R code how we can do this.

Suppose we want to build several variations of decision trees, with different options. On a small test dataset, run Rattle withing RStudio to build models with different parameter choices. Grab the R code from the Log tab, paste into RStudio, and turn it into a schedule to run on the full dataset overnight.

Here's an example. Run Rattle over the weather dataset, using a small training partition, to build a decision tree with different parameter options. The Rattle Log code for just two runs:

```
crs$rpart <- rpart(RainTomorrow ~ .,
    data=crs$dataset[crs$train, c(crs$input, crs$target)],
    method="class", parms=list(split="information"),
    control=rpart.control(usesurrogate=0, maxsurrogate=0))

crs$rpart <- rpart(RainTomorrow ~ .,
    data=crs$dataset[crs$train, c(crs$input, crs$target)],
    method="class", parms=list(split="information"),
      control=rpart.control(minsplit=5, maxdepth=10,
          cp=0.000010, usesurrogate=0, maxsurrogate=0))
```

We can see that each run overwrites the previous model. So tidying things up a little, we might end up with the following script:

```
ds   <- crs$dataset[crs$train, c(crs$input, crs$target)]
form <- formula("RainTomorrow ~ .")

m.rp01 <- rpart(form, data=ds, method="class", parms=list(split="information"),
                control=rpart.control(usesurrogate=0, maxsurrogate=0))

m.rp02 <- rpart(form, data=ds, method="class", parms=list(split="information"),
                control=rpart.control(minsplit=5, maxdepth=10,
                    cp=0.000010, usesurrogate=0, maxsurrogate=0))
```

Change `ds` to point to the full (i.e., larger) dataset and set the script running overnight:

```
ds <- crs$dataset[crs$train, c(crs$input, crs$target)]
```

With a bit of care we can then use Rattle to explore the different models! For example:

```
crs$rpart <- m.rp02
```

Then click on Draw within the Model Tab, Decision Tree option, to draw the tree. Of course we can copy the code to draw the tree and run it directly in RStudio:

```
fancyRpartPlot(m.rp02, main="My Very Own Decision Tree")
```

Concept suggested by Richard Calaba (140607) on Google Code.

14 Audit: Evaluate

15 Audit: Review the Log

16 Evaluate — Score — Class and Probability

Rattle (Evaluate Tab, Score Option) can score new data to report the predicted Class or Probability, but not both at the same time. Here we score and collect both and then save into the one file, together with the original dataset being scored.

We begin by saving the dataset to be scored into the data frame `ds`. Notice that because we have a random forest as one of the models, we use `na.omit()` to eliminate rows with missing data. The dataset we are scoring is the test dataset that is created by default as a partition of the dataset loaded into Rattle.

```
ds <- na.omit(crs$dataset[crs$test, c(crs$input, crs$target)])
```

We then simply call `predict()` for each of the models we which to use to score our dataset, appending the results as additional columns of the data frame `ds`.

```
ds$rp.cl <- predict(crs$rpart, newdata=ds, type="class")
ds$rp.pr <- predict(crs$rpart, newdata=ds)[,2]
ds$rf.cl <- predict(crs$rf,    newdata=ds)
ds$rf.pr <- predict(crs$rf,    newdata=ds, type="prob")[,2]
```

After checking the data frame looks okay using `head()` we save the results to file using `write.csv()`.

```
head(ds)
write.csv(ds, file="scores.csv", row.names=FALSE)
```

Other models can easily be added using the above template. For example, to add an AdaBoost model, build the model in Rattle and then score using the Evaluate tab. From the Rattle Log tab we will see the syntax used to score using the new model. The following lines come directly from the Log Tab:

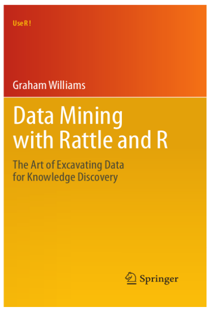
```
crs$pr <- predict(crs$rpart,
                  newdata=crs$dataset[crs$validate, c(crs$input)],
                  type="class")
crs$pr <- predict(crs$ada,
                  newdata=crs$dataset[crs$validate, c(crs$input)])
```

We simply paste these lines into the above script and change the lines to:

```
ds$ada.cl <- predict(crs$ada, newdata=ds)
ds$ada.pr <- predict(crs$ada, newdata=ds, type="prob")[,2]
```

Concept suggested by Richard Calaba (140607) on Google Code.

17 Further Reading and Acknowledgements



The Rattle Book, published by Springer, provides a comprehensive introduction to data mining and analytics using Rattle and R. It is available from Amazon. Other documentation on a broader selection of R topics of relevance to the data scientist is freely available from `http://datamining.togaware.com`, including the Datamining Desktop Survival Guide.

This chapter is one of many chapters available from `http://HandsOnDataScience.com`. In particular follow the links on the website with a * which indicates the generally more developed chapters.

18 References

R Core Team (2014). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

Williams GJ (2009). "Rattle: A Data Mining GUI for R." *The R Journal*, **1**(2), 45–55. URL http://journal.r-project.org/archive/2009-2/RJournal_2009-2_Williams.pdf.

Williams GJ (2011). *Data Mining with Rattle and R: The art of excavating data for knowledge discovery*. Use R! Springer, New York. URL http://www.amazon.com/gp/product/1441998896/ref=as_li_qf_sp_asin_tl?ie=UTF8&tag=togaware-20&linkCode=as2&camp=217145&creative=399373&creativeASIN=1441998896.

Williams GJ (2014). *rattle: Graphical user interface for data mining in R*. R package version 3.1.4, URL http://rattle.togaware.com/.

This document, sourced from RattleO.Rnw revision 508, was processed by KnitR version 1.6 of 2014-05-24 and took 1.4 seconds to process. It was generated by gjw on nyx running Ubuntu 14.04.1 LTS with Intel(R) Xeon(R) CPU W3520 @ 2.67GHz having 4 cores and 12.3GB of RAM. It completed the processing 2014-08-24 11:09:12.

Other resources include:

- http://rattle.togaware.com.