Data Science with R
Naive Bayes Clasification

Graham.Williams@togaware.com

9th June 2014

Visit http://onepager.togaware.com/ for more OnePageR's.

The required packages for this chapter include:

```
library(rattle)       # weather and normVarNames()
library(randomForest) # na.roughfix()
library(e1071)        # naiveBayes()
library(ROCR)         # prediction()
```

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 Prepare Weather Data for Modelling

See Chapters on Data and Model for the template for preparing data and building models. We repeat the setup here with little comment, except to note that we use the **weather** dataset from rattle (Williams, 2014).

```
library(rattle)         # Normalise names normVarNames() and weather dataset.
library(randomForest)   # Impute missing using na.roughfix().

dsname     <- "weather"
ds         <- get(dsname)
names(ds)  <- normVarNames(names(ds))
vars       <- names(ds)
target     <- "rain_tomorrow"
risk       <- "risk_mm"
id         <- c("date", "location")

ignore     <- union(id, if (exists("risk")) risk)
vars       <- setdiff(vars, ignore)

inputs     <- setdiff(vars, target)
numi       <- which(sapply(ds[inputs], is.numeric))
numc       <- names(numi)
cati       <- which(sapply(ds[inputs], is.factor))
catc       <- names(cati)

ds[numc]   <- na.roughfix(ds[numc]) # Impute missing values, roughly.
ds[target] <- as.factor(ds[[target]])   # Ensure the target is categoric.

nobs       <- nrow(ds)

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

set.seed(42)

train      <- sample(nobs, 0.7*nobs)
test       <- setdiff(seq_len(nobs), train)
actual     <- ds[test, target]
risks      <- ds[test, risk]
```

2 Review the Dataset

It is always a good idea to review the data.

```
dim(ds)

## [1] 366  24

names(ds)

##  [1] "date"            "location"        "min_temp"
##  [4] "max_temp"        "rainfall"        "evaporation"
##  [7] "sunshine"        "wind_gust_dir"   "wind_gust_speed"
## [10] "wind_dir_9am"    "wind_dir_3pm"    "wind_speed_9am"
## [13] "wind_speed_3pm"  "humidity_9am"    "humidity_3pm"
....

head(ds)

##         date location min_temp max_temp 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
....

tail(ds)

##           date location min_temp max_temp rainfall evaporation sunshine
## 361 2008-10-26 Canberra      7.9     26.1        0         6.8      3.5
## 362 2008-10-27 Canberra      9.0     30.7        0         7.6     12.1
## 363 2008-10-28 Canberra      7.1     28.4        0        11.6     12.7
## 364 2008-10-29 Canberra     12.5     19.9        0         8.4      5.3
....

str(ds)

## 'data.frame': 366 obs. of  24 variables:
##  $ date           : Date, format: "2007-11-01" "2007-11-02" ...
##  $ location       : Factor w/ 49 levels "Adelaide","Albany",..: 10 10 10 1...
##  $ min_temp       : num  8 14 13.7 13.3 7.6 6.2 6.1 8.3 8.8 8.4 ...
##  $ max_temp       : num  24.3 26.9 23.4 15.5 16.1 16.9 18.2 17 19.5 22.8 ...
....

summary(ds)

##       date                       location      min_temp        max_temp
##   Min.   :2007-11-01   Canberra     :366   Min.   :-5.30   Min.   : 7.6
##   1st Qu.:2008-01-31   Adelaide     :  0   1st Qu.: 2.30   1st Qu.:15.0
##   Median :2008-05-01   Albany       :  0   Median : 7.45   Median :19.6
##   Mean   :2008-05-01   Albury       :  0   Mean   : 7.27   Mean   :20.6
....
```

3 Naive Bayes Model

Here we use naiveBayes() from e1071 (Meyer *et al.*, 2014).

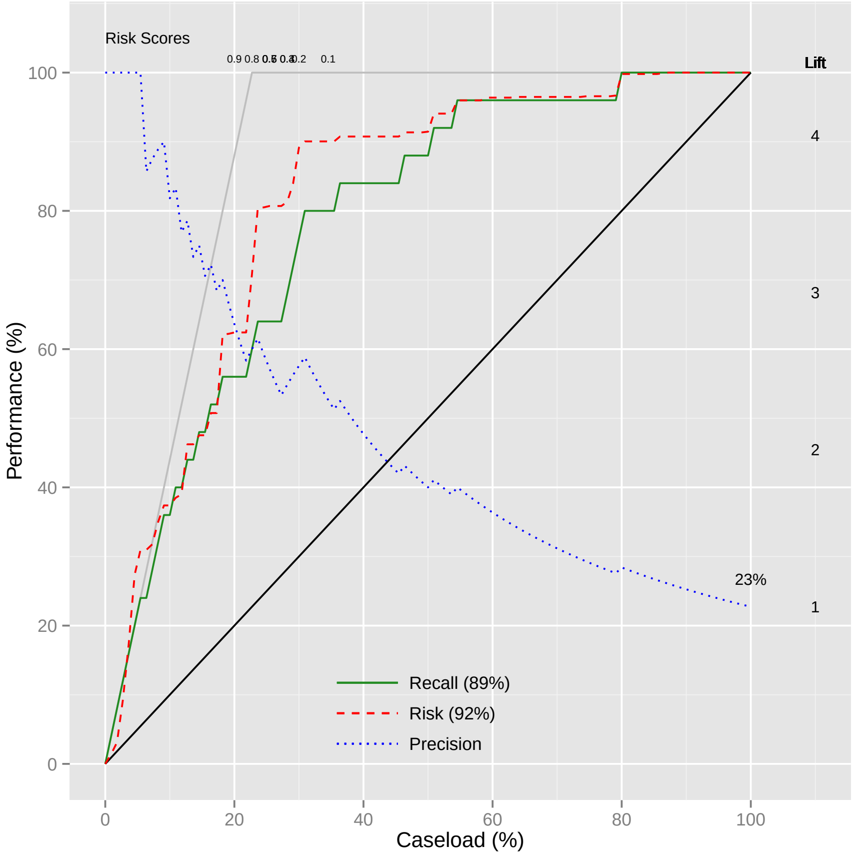
```
library(e1071)
model      <- naiveBayes(form, data=ds[train, vars])
model

##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x=X, y=Y, laplace=laplace)
##
## A-priori probabilities:
## Y
##     No    Yes
## 0.8398 0.1602
##
## Conditional probabilities:
##      min_temp
## Y       [,1]  [,2]
##   No   6.34 5.958
##   Yes 10.53 6.239
##
##      max_temp
## Y       [,1]  [,2]
##   No  19.94 6.730
##   Yes 22.15 5.977
##
##      rainfall
## Y       [,1]  [,2]
##   No  1.233 3.788
##   Yes 2.190 4.377
....
```

4 Naive Bayes Model Evaluation

Next we evaluate the model.

```
classes    <- predict(model, ds[test, vars], type="class")
acc        <- sum(classes == actual, na.rm=TRUE)/length(actual)
err        <- sum(classes != actual, na.rm=TRUE)/length(actual)
predicted  <- predict(model, ds[test, vars], type="raw")[,2]
pred       <- prediction(predicted, ds[test, target])
ate        <- attr(performance(pred, "auc"), "y.values")[[1]]
riskchart(predicted, actual, risks)
```

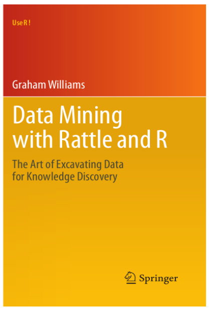


```
round(table(actual, classes, dnn=c("Actual", "Predicted"))/length(actual), 2)

##        Predicted
## Actual   No  Yes
##    No  0.67 0.10
##    Yes 0.08 0.15
```

5 Further Reading

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.


Use R!
Graham Williams
Data Mining with Rattle and R
The Art of Excavating Data for Knowledge Discovery
Springer

This module is one of many OnePageR modules available from `http://onepager.togaware.com`. In particular follow the links on the website with a * which indicates the generally more developed OnePageR modules.

6 References

Meyer D, Dimitriadou E, Hornik K, Weingessel A, Leisch F (2014). *e1071: Misc Functions of the Department of Statistics (e1071), TU Wien.* R package version 1.6-3, URL http://CRAN.R-project.org/package=e1071.

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 3.0.4, URL http://rattle.togaware.com/.

This document, sourced from BayesO.Rnw revision 419, was processed by KnitR version 1.6 of 2014-05-24 and took 2.8 seconds to process. It was generated by gjw on nyx running Ubuntu 14.04 LTS with Intel(R) Xeon(R) CPU W3520 @ 2.67GHz having 4 cores and 12.3GB of RAM. It completed the processing 2014-06-09 10:33:29.