Keosotra Veng
Stat 6620

Homework 4 Part 2

Rule Learners Algorithm

Step 1: Step 1 - collecting data

- We got the data from: http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29

Step 2: Exploring and preparing the data ----

- Import mushrooms.csv into R

```
mushrooms <- read.csv("mushrooms.csv", stringsAsFactors = TRUE)
```

- Examine the structure of the data frame

```
str(mushrooms)
```

```
## 'data.frame':    8124 obs. of  23 variables:
##  $ type                    : Factor w/ 2 levels "edible","poisonous": 2 1
1 2 1 1 1 1 2 1 ...
##  $ cap_shape               : Factor w/ 6 levels "bell","conical",..: 3 3 1
3 3 3 1 1 3 1 ...
##  $ cap_surface             : Factor w/ 4 levels "fibrous","grooves",..: 4
4 4 3 4 3 4 3 3 4 ...
##  $ cap_color               : Factor w/ 10 levels "brown","buff",..: 1 10 9
9 4 10 9 9 9 10 ...
##  $ bruises                 : Factor w/ 2 levels "no","yes": 2 2 2 2 1 2 2
2 2 2 ...
##  $ odor                    : Factor w/ 9 levels "almond","anise",..: 8 1 2
8 7 1 1 2 8 1 ...
##  $ gill_attachment         : Factor w/ 2 levels "attached","free": 2 2 2 2
2 2 2 2 2 2 ...
##  $ gill_spacing            : Factor w/ 2 levels "close","crowded": 1 1 1 1
2 1 1 1 1 1 ...
##  $ gill_size               : Factor w/ 2 levels "broad","narrow": 2 1 1 2
1 1 1 1 2 1 ...
##  $ gill_color              : Factor w/ 12 levels "black","brown",..: 1 1 2
2 1 2 5 2 8 5 ...
##  $ stalk_shape             : Factor w/ 2 levels "enlarging","tapering": 1
1 1 1 2 1 1 1 1 1 ...
##  $ stalk_root              : Factor w/ 5 levels "bulbous","club",..: 3 2 2
3 3 2 2 2 3 2 ...
##  $ stalk_surface_above_ring: Factor w/ 4 levels "fibrous","scaly",..: 4 4
4 4 4 4 4 4 4 4 ...
##  $ stalk_surface_below_ring: Factor w/ 4 levels "fibrous","scaly",..: 4 4
4 4 4 4 4 4 4 4 ...
```

```
##  $ stalk_color_above_ring  : Factor w/ 9 levels "brown","buff",..: 8 8 8 8
8 8 8 8 8 8 ...
##  $ stalk_color_below_ring  : Factor w/ 9 levels "brown","buff",..: 8 8 8 8
8 8 8 8 8 8 ...
##  $ veil_type               : Factor w/ 1 level "partial": 1 1 1 1 1 1 1 1
1 1 ...
##  $ veil_color              : Factor w/ 4 levels "brown","orange",..: 3 3 3
3 3 3 3 3 3 3 ...
##  $ ring_number             : Factor w/ 3 levels "none","one","two": 2 2 2
2 2 2 2 2 2 2 ...
##  $ ring_type               : Factor w/ 5 levels "evanescent","flaring",..:
5 5 5 5 1 5 5 5 5 5 ...
##  $ spore_print_color       : Factor w/ 9 levels "black","brown",..: 1 2 2
1 2 1 1 2 1 1 ...
##  $ population              : Factor w/ 6 levels "abundant","clustered",..:
4 3 3 4 1 3 3 4 5 4 ...
##  $ habitat                 : Factor w/ 7 levels "grasses","leaves",..: 5 1
3 5 1 1 3 3 1 3 ...
```

- We drop the veil_type feature from the data

```
mushrooms$veil_type <- NULL
```

- Examine the class distribution. We have 4208 edible mushrooms and 3916 poisonous in our data set.

```
table(mushrooms$type)

##
##    edible poisonous
##      4208      3916
```

- We randomize our data set.

```
set.seed(123)
train_sample <- sample(8124, 7000)

str(train_sample)

##  int [1:7000] 2337 6404 3322 7171 7637 370 4288 7244 4476 3706 ...
```

- We set 7000 observations into training and the rest into testing.

```
mushrooms_train <- mushrooms[train_sample, ]
mushrooms_test  <- mushrooms[-train_sample, ]
```

Step 3: Training a model on the data ----

- Import RWeka into R

```
library(RWeka)
```

- we use OneR from RWeka to train our data

```
mushroom_1R <- OneR(type ~ ., data = mushrooms_train)
```

Keosotra Veng
Stat 6620

Step 4: Evaluating model performance ----

- We correctly predict 6895 out of 7000 in the training data set.

```
mushroom_1R

## odor:
##  almond  -> edible
##  anise   -> edible
##  creosote    -> poisonous
##  fishy   -> poisonous
##  foul    -> poisonous
##  musty   -> poisonous
##  none    -> edible
##  pungent -> poisonous
##  spicy   -> poisonous
## (6895/7000 instances correct)

summary(mushroom_1R)

##
## === Summary ===
##
## Correctly Classified Instances        6895               98.5    %
## Incorrectly Classified Instances       105                1.5    %
## Kappa statistic                          0.9699
## Mean absolute error                      0.015
## Root mean squared error                  0.1225
## Relative absolute error                  3.0039 %
## Root relative squared error             24.5108 %
## Total Number of Instances             7000
##
## === Confusion Matrix ===
##
##     a    b   <-- classified as
##  3626    0 |    a = edible
##   105 3269 |    b = poisonous

mushroom_pred <- predict(mushroom_1R, mushrooms_test)
```

cross tabulation of predicted versus actual classes

- We correctly predict 1109 observations out of 1124 in the test data set.
- We incrrectly predict 15 obseration where our preidction is edible but it's actually poinsonous

```
library(gmodels)
CrossTable(mushrooms_test$type, mushroom_pred,
           prop.chisq = FALSE, prop.c = FALSE, prop.r = FALSE,
           dnn = c('actual default', 'predicted default'))
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  1124
##
##
##                 | predicted default
## actual default |    edible | poisonous | Row Total |
## ---------------|-----------|-----------|-----------|
##         edible |       582 |         0 |       582 |
##                |     0.518 |     0.000 |           |
## ---------------|-----------|-----------|-----------|
##      poisonous |        15 |       527 |       542 |
##                |     0.013 |     0.469 |           |
## ---------------|-----------|-----------|-----------|
##   Column Total |       597 |       527 |      1124 |
## ---------------|-----------|-----------|-----------|
##
##
```

Step 5: Improving model performance ----

- We will be using JRip to try to improve our prediction.
- We correctly predict 7000 out of 7000 in our training data set with JRip

```
mushroom_JRip <- JRip(type ~ ., data = mushrooms_train)
mushroom_JRip
```

```
## JRIP rules:
## ===========
##
## (odor = foul) => type=poisonous (1860.0/0.0)
## (gill_size = narrow) and (gill_color = buff) => type=poisonous (986.0/0.0)
## (gill_size = narrow) and (odor = pungent) => type=poisonous (222.0/0.0)
## (odor = creosote) => type=poisonous (171.0/0.0)
## (spore_print_color = green) => type=poisonous (65.0/0.0)
## (stalk_surface_below_ring = scaly) and (stalk_surface_above_ring = silky)
=> type=poisonous (58.0/0.0)
## (habitat = leaves) and (cap_surface = scaly) and (population = clustered)
=> type=poisonous (10.0/0.0)
## (cap_surface = grooves) => type=poisonous (2.0/0.0)
##  => type=edible (3626.0/0.0)
##
## Number of Rules : 9
```

```
summary(mushroom_JRip)

##
## === Summary ===
##
## Correctly Classified Instances        7000               100      %
## Incorrectly Classified Instances         0                 0      %
## Kappa statistic                          1
## Mean absolute error                      0
## Root mean squared error                  0
## Relative absolute error                  0      %
## Root relative squared error              0      %
## Total Number of Instances             7000
##
## === Confusion Matrix ===
##
##     a    b   <-- classified as
##  3626    0 |    a = edible
##     0 3374 |    b = poisonous

mushroom_pred <- predict(mushroom_JRip, mushrooms_test)
```

Cross tabulation of predicted versus actual classes

- We perfectly predict whether the mushroom is poisonous or edible with JRip.
- We predict 582 edicble and 542 poisonous

```
library(gmodels)
CrossTable(mushrooms_test$type, mushroom_pred,
           prop.chisq = FALSE, prop.c = FALSE, prop.r = FALSE,
           dnn = c('actual default', 'predicted default'))

##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  1124
##
##
##                | predicted default
## actual default |    edible | poisonous | Row Total |
## ---------------|-----------|-----------|-----------|
##         edible |       582 |         0 |       582 |
##                |     0.518 |     0.000 |           |
## ---------------|-----------|-----------|-----------|
```

```
##      poisonous |         0 |       542 |       542 |
##                |     0.000 |     0.482 |           |
## ---------------|-----------|-----------|-----------|
##   Column Total |       582 |       542 |      1124 |
## ---------------|-----------|-----------|-----------|
##
##
```

Rule Learner Using C5.0 Decision Trees (not in text)

- Now let's try using C50 to predict whether the mushroom is edible or poisonous
- The result is the same as OneR algorithm in this traning data set

```
library(C50)
mushroom_c5rules <- C5.0(type ~ odor + gill_size, data = mushrooms_train, rul
es = TRUE)
mushroom_c5rules
```

```
##
## Call:
## C5.0.formula(formula = type ~ odor + gill_size, data =
##  mushrooms_train, rules = TRUE)
##
## Rule-Based Model
## Number of samples: 7000
## Number of predictors: 2
##
## Number of Rules: 2
##
## Non-standard options: attempt to group attributes
```

```
summary(mushroom_c5rules)
```

```
##
## Call:
## C5.0.formula(formula = type ~ odor + gill_size, data =
##  mushrooms_train, rules = TRUE)
##
##
## C5.0 [Release 2.07 GPL Edition]      Wed May 03 02:34:12 2017
## -------------------------------
##
## Class specified by attribute `outcome'
##
## Read 7000 cases (3 attributes) from undefined.data
##
## Rules:
##
## Rule 1: (3731/105, lift 1.9)
##  odor in {almond, anise, none}
```

```
##  ->  class edible  [0.972]
##
## Rule 2: (3269, lift 2.1)
##  odor in {creosote, fishy, foul, musty, pungent, spicy}
##  ->  class poisonous  [1.000]
##
## Default class: edible
##
##
## Evaluation on training data (7000 cases):
##
##          Rules
##    ----------------
##      No      Errors
##
##       2  105( 1.5%)   <<
##
##
##     (a)   (b)    <-classified as
##    ----  ----
##    3626          (a): class edible
##     105  3269    (b): class poisonous
##
##
##  Attribute usage:
##
##  100.00% odor
##
##
## Time: 0.0 secs

mushroom_pred <- predict(mushroom_c5rules, mushrooms_test)
```

Cross tabulation of predicted versus actual classes

- On our training data set the result of C50 algorithm is the same as OneR.
- We correctly predict 1109 out of 1124 observations
- There are 15 incorrect predictions where we predict edible but it's poisonous

```
library(gmodels)
CrossTable(mushrooms_test$type, mushroom_pred,
           prop.chisq = FALSE, prop.c = FALSE, prop.r = FALSE,
           dnn = c('actual default', 'predicted default'))

##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |         N / Table Total |
```

```
## |-------------------------|
##
##
## Total Observations in Table:  1124
##
##
##                | predicted default
## actual default |    edible | poisonous | Row Total |
## ---------------|-----------|-----------|-----------|
##         edible |       582 |         0 |       582 |
##                |     0.518 |     0.000 |           |
## ---------------|-----------|-----------|-----------|
##      poisonous |        15 |       527 |       542 |
##                |     0.013 |     0.469 |           |
## ---------------|-----------|-----------|-----------|
##   Column Total |       597 |       527 |      1124 |
## ---------------|-----------|-----------|-----------|
##
##
```