CMP713 Data Mining 2021F Assignment 2

due 24 December 2021

Question

In this assignment you will be analyzing the **mushroom dataset** from UCI machine learning repositories. Google for the dataset and download it to your computer. In this assignment, you are given an analysis of this dataset. However, the codes are hidden and you can only see the outputs of the codes. Your job is to figure out what the codes are in each block. You must **exactly** reproduce the below section (1.1) and produce a pdf file of your analysis that shows both your code and the outputs. You will then submit this pdf file as your assignment. You can download a template Rmd file here Good luck!

Analysis

Load the file

We need to first load the file into R. Since the data file doesn't contain a header row, we need to load the file accordingly and later assign the column names by manually.

```
## # A tibble: 10 x 23
##    edibility cap.shape cap.surface cap.color bruises odor  gill.attachment
##    <fct>     <fct>     <fct>       <fct>     <fct>   <fct> <fct>
##  1 p         x         s           n         t       p     f
##  2 e         x         s           y         t       a     f
##  3 e         b         s           w         t       l     f
##  4 p         x         y           w         t       p     f
##  5 e         x         s           g         f       n     f
##  6 e         x         y           y         t       a     f
##  7 e         b         s           w         t       a     f
##  8 e         b         y           w         t       l     f
##  9 p         x         y           w         t       p     f
## 10 e         b         s           y         t       a     f
## # ... with 16 more variables: gill.spacing <fct>, gill.size <fct>,
## #   gill.color <fct>, stalk.shape <fct>, stalk.root <fct>,
## #   stalk.surface.above.ring <fct>, stalk.surface.below.ring <fct>,
## #   stalk.color.above.ring <fct>, stalk.color.below.ring <fct>,
## #   veil.type <fct>, veil.color <fct>, ring.number <fct>, ring.type <fct>,
## #   spore.print.color <fct>, population <fct>, habitat <fct>
```

```
## spec_tbl_df [8,124 x 23] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ edibility              : Factor w/ 2 levels "p","e": 1 2 2 1 2 2 2 2 1 2 ...
##  $ cap.shape              : Factor w/ 6 levels "x","b","s","f",..: 1 1 2 1 1 1 2 2 1 2 ...
##  $ cap.surface            : Factor w/ 4 levels "s","y","f","g": 1 1 1 2 1 2 1 2 2 1 ...
##  $ cap.color              : Factor w/ 10 levels "n","y","w","g",..: 1 2 3 3 4 2 3 3 3 2 ...
##  $ bruises                : Factor w/ 2 levels "t","f": 1 1 1 1 2 1 1 1 1 1 ...
##  $ odor                   : Factor w/ 9 levels "p","a","l","n",..: 1 2 3 1 4 2 2 3 1 2 ...
##  $ gill.attachment        : Factor w/ 2 levels "f","a": 1 1 1 1 1 1 1 1 1 1 ...
##  $ gill.spacing           : Factor w/ 2 levels "c","w": 1 1 1 1 2 1 1 1 1 1 ...
##  $ gill.size              : Factor w/ 2 levels "n","b": 1 2 2 1 2 2 2 2 1 2 ...
```

```
##  $ gill.color              : Factor w/ 12 levels "k","n","g","p",..: 1 1 2 2 1 2 3 2 4 3 ...
##  $ stalk.shape             : Factor w/ 2 levels "e","t": 1 1 1 1 2 1 1 1 1 1 ...
##  $ stalk.root              : Factor w/ 5 levels "e","c","b","r",..: 1 2 2 1 1 2 2 2 1 2 ...
##  $ stalk.surface.above.ring: Factor w/ 4 levels "s","f","k","y": 1 1 1 1 1 1 1 1 1 1 ...
##  $ stalk.surface.below.ring: Factor w/ 4 levels "s","f","y","k": 1 1 1 1 1 1 1 1 1 1 ...
##  $ stalk.color.above.ring  : Factor w/ 9 levels "w","g","p","n",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ stalk.color.below.ring  : Factor w/ 9 levels "w","p","g","b",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ veil.type               : Factor w/ 1 level "p": 1 1 1 1 1 1 1 1 1 1 ...
##  $ veil.color              : Factor w/ 4 levels "w","n","o","y": 1 1 1 1 1 1 1 1 1 1 ...
##  $ ring.number             : Factor w/ 3 levels "o","t","n": 1 1 1 1 1 1 1 1 1 1 ...
##  $ ring.type               : Factor w/ 5 levels "p","e","l","f",..: 1 1 1 1 2 1 1 1 1 1 ...
##  $ spore.print.color       : Factor w/ 9 levels "k","n","u","h",..: 1 2 2 1 2 1 1 2 1 1 ...
##  $ population              : Factor w/ 6 levels "s","n","a","v",..: 1 2 2 1 3 2 2 1 4 1 ...
##  $ habitat                 : Factor w/ 7 levels "u","g","m","d",..: 1 2 3 1 2 2 3 3 2 3 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   edibility = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),
##   ..   cap.shape = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),
##   ..   cap.surface = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),
##   ..   cap.color = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),
##   ..   bruises = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),
##   ..   odor = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),
##   ..   gill.attachment = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),
##   ..   gill.spacing = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),
##   ..   gill.size = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),
##   ..   gill.color = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),
##   ..   stalk.shape = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),
##   ..   stalk.root = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),
##   ..   stalk.surface.above.ring = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),
##   ..   stalk.surface.below.ring = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),
##   ..   stalk.color.above.ring = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),
##   ..   stalk.color.below.ring = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),
##   ..   veil.type = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),
##   ..   veil.color = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),
##   ..   ring.number = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),
##   ..   ring.type = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),
##   ..   spore.print.color = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),
##   ..   population = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE),
##   ..   habitat = col_factor(levels = NULL, ordered = FALSE, include_na = FALSE)
##   .. )
##  - attr(*, "problems")=<externalptr>
```

Exploratory Data Analysis

Let's check the NA counts.

```
## The dataset contains 0 NAs.
```

Great, there are no NA values. The target column is the first column (edibility). Let's check if any of the input columns have some weird distribution:

```
## $cap.shape
##
##    b    c    f    k    s    x
##  452    4 3152  828   32 3656
##
```

```
## $cap.surface
##
##    f    g    s    y
## 2320    4 2556 3244
##
## $cap.color
##
##    b    c    e    g    n    p    r    u    w    y
##  168   44 1500 1840 2284  144   16   16 1040 1072
##
## $bruises
##
##    f    t
## 4748 3376
##
## $odor
##
##    a    c    f    l    m    n    p    s    y
##  400  192 2160  400   36 3528  256  576  576
##
## $gill.attachment
##
##    a    f
##  210 7914
##
## $gill.spacing
##
##    c    w
## 6812 1312
##
## $gill.size
##
##    b    n
## 5612 2512
##
## $gill.color
##
##    b    e    g    h    k    n    o    p    r    u    w    y
## 1728   96  752  732  408 1048   64 1492   24  492 1202   86
##
## $stalk.shape
##
##    e    t
## 3516 4608
##
## $stalk.root
##
##    ?    b    c    e    r
## 2480 3776  556 1120  192
##
## $stalk.surface.above.ring
##
##    f    k    s    y
##  552 2372 5176   24
```

```
##
## $stalk.surface.below.ring
##
##    f    k    s    y
##  600 2304 4936  284
##
## $stalk.color.above.ring
##
##    b    c    e    g    n    o    p    w    y
##  432   36   96  576  448  192 1872 4464    8
##
## $stalk.color.below.ring
##
##    b    c    e    g    n    o    p    w    y
##  432   36   96  576  512  192 1872 4384   24
##
## $veil.type
##
##    p
## 8124
##
## $veil.color
##
##    n    o    w    y
##   96   96 7924    8
##
## $ring.number
##
##    n    o    t
##   36 7488  600
##
## $ring.type
##
##    e    f    l    n    p
## 2776   48 1296   36 3968
##
## $spore.print.color
##
##    b    h    k    n    o    r    u    w    y
##   48 1632 1872 1968   48   72   48 2388   48
##
## $population
##
##    a    c    n    s    v    y
##  384  340  400 1248 4040 1712
##
## $habitat
##
##    d    g    l    m    p    u    w
## 3148 2148  832  292 1144  368  192
```

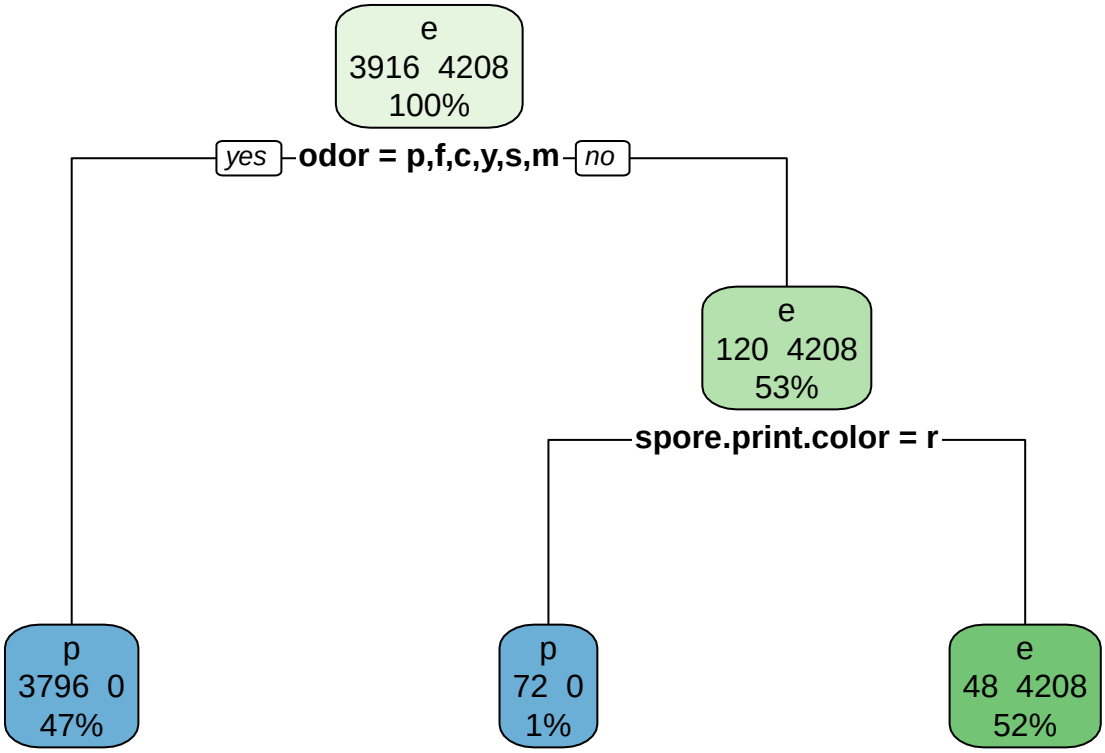
Looks like `veil.type` has only one level. We can remove it from the dataset. The remaining columns are as follows:

```
##  [1] "edibility"                "cap.shape"
```

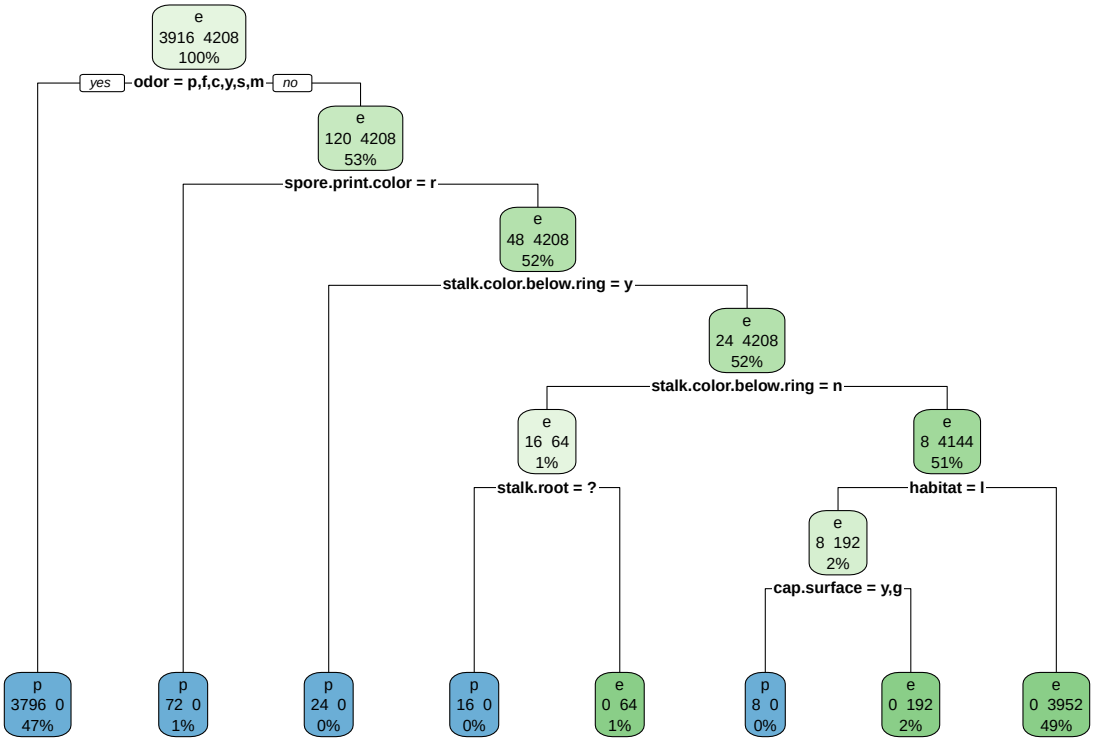
```
##  [3] "cap.surface"              "cap.color"
##  [5] "bruises"                  "odor"
##  [7] "gill.attachment"          "gill.spacing"
##  [9] "gill.size"                "gill.color"
## [11] "stalk.shape"              "stalk.root"
## [13] "stalk.surface.above.ring" "stalk.surface.below.ring"
## [15] "stalk.color.above.ring"   "stalk.color.below.ring"
## [17] "veil.color"               "ring.number"
## [19] "ring.type"                "spore.print.color"
## [21] "population"               "habitat"
```

Decision Tree

Let's build a decision tree with **default parameters** to see if we can detect some good rules. We will use `rpart.plot` with `extra` parameter set to `101` to draw the resulting model. This shows us the number of edible and poisonous mushrooms at each node:



We already got a very nice result, but let's dig deeper by decreasing the value of the complexity parameter (cp) to 0.001. This will let the algorithm go deeper finding more complicated rules:



Now we have a perfect classification. Although this is called "overfitting" in data mining and is not a preferred thing to have; we don't want to risk it with mushrooms, do we? Let's use `rpart.rules` with `extra` parameter set to 4, to have a readable version of the detected rules:

```
##  edibility     p    e
##          p [1.00  .00] when odor is p or f or c or y or s or m
##          p [1.00  .00] when odor is                a or l or n & spore.print.color is
##          p [1.00  .00] when odor is                a or l or n & spore.print.color is k or n or u or
##          p [1.00  .00] when odor is                a or l or n & spore.print.color is k or n or u or
##          p [1.00  .00] when odor is                a or l or n & spore.print.color is k or n or u or
##          e [ .00 1.00] when odor is                a or l or n & spore.print.color is k or n or u or
##          e [ .00 1.00] when odor is                a or l or n & spore.print.color is k or n or u or
##          e [ .00 1.00] when odor is                a or l or n & spore.print.color is k or n or u or
```

We have 5 rules for poisonous mushrooms and 3 for edible ones. Now you can print these rules the next time you go camping in the wilderness.

You should prepare the analysis report in an Rmd (R markdown) file, knit the results to PDF and submit the PDF using this form.