Project 8: Applied Theory & Practice I, Mushrooms

CSCI 568 - Data Mining

Chong Ding

25 November, 2011

1 Starting and Preprocessing

By starting to read the given mushroom cvs and metadata files, I realized the dataset can be read by either Knime or Weka unless correspoding attributes' descriptions are added. The first attribute followed by a bunch of other mushroom's attributes is whether a mushromm is poisonous or edible. Some of the attributes can be unknow which show a question mark in the dataset file, but no one is missing edibility status.

I found some ARFF file of mushroom dataset on-line can compared with the given metadata file. The attributes are in different order in the one I found on-line and given one, so I just change the order of the attributes description to match the given mushroom dataset. By simply running the script called preproc.sh in *./preprocessing*, it'll combine the attributes description and mushroom dataset imto a ARFF mushroom dataset.

Weka are used in this project since it's more straightforward.

2 Can you generate summary statistics that help describe the data?

Since all the twenty-two attributes are nominally valued. it makes no sense to generate summary statistics, such as max, min, median or averages, etc. So only the distribution can be explored. Figure 1 is the visualization of the distribution of edible vs poisonous for all of the attributes. It can be found that, for the attributes gill-attachment, gill-spacing, vell-color and ring-number, only one type are very dominant, the other types in the attributes are rare.

3 Can the edible and poisonous data objects be distilled into groups?

Yes, using association analysis or clustering analysis can distill mushroom into groups.

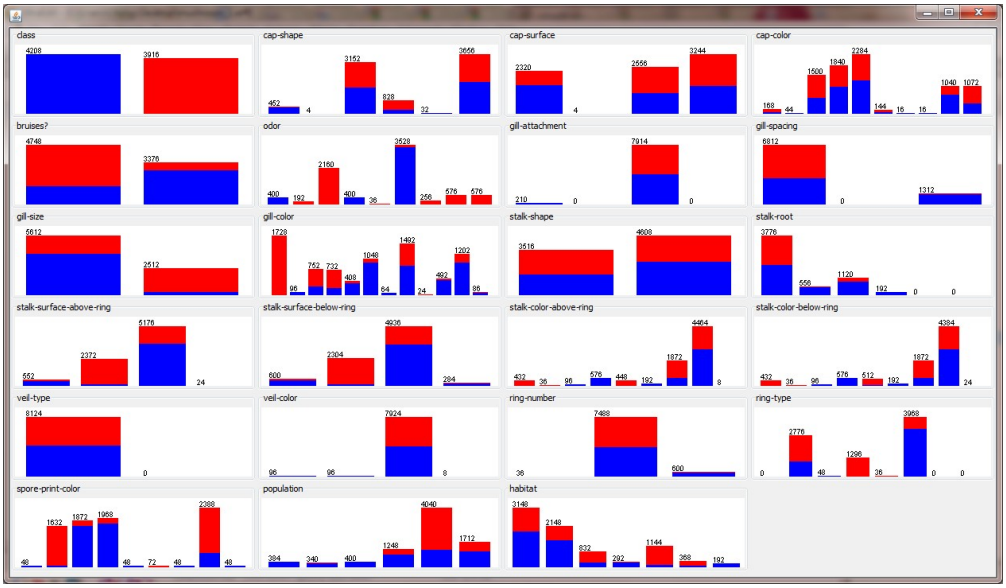

Figure 1: The distribution of edibility among all attributes.

4 Can a classification model be created that can predict whether a mushroom is edible or poisonous?

Yes. Both Tree-based classifier and Rule-based classifier are tried. Two different algorithms are used for each type of classifier. All the results are generated by using Weka.

4.1 Tree-based classifier

4.1.1 NBTree algorithm

Please refer to the file *trees.NBTree* in *./exploring*.

4.1.2 SimpleCart algorithm

```
CART Decision Tree

odor=(c)|(f)|(m)|(p)|(s)|(y): p(3796.0/0.0)
odor!=(c)|(f)|(m)|(p)|(s)|(y)
|  spore-print-color=(r): p(72.0/0.0)
|  spore-print-color!=(r)
|  |  stalk-color-below-ring=(y): p(24.0/0.0)
|  |  stalk-color-below-ring!=(y)
|  |  |  cap-surface=(g): p(4.0/0.0)
|  |  |  cap-surface!=(g)
|  |  |  |  stalk-color-below-ring=(n)
```

```
|  |  |  |  |  stalk-surface-above-ring=(k): p(16.0/0.0)
|  |  |  |  |  stalk-surface-above-ring!=(k): e(64.0/0.0)
|  |  |  |  stalk-color-below-ring!=(n): e(4144.0/4.0)

Number of Leaf Nodes: 7

Size of the Tree: 13
```

4.2 Rule-based classifier

4.2.1 OneR

```
odor:
a -> e
c -> p
f -> p
l -> e
m -> p
n -> e
p -> p
s -> p
y -> p
(8004/8124 instances correct)
```

4.2.2 PART

```
PART decision list
------------------

odor = f: p (2160.0)

gill-size = b AND
ring-number = o: e (3392.0)

ring-number = t AND
spore-print-color = w: e (528.0)

odor = s: p (576.0)

odor = y: p (576.0)

stalk-shape = e AND
stalk-surface-below-ring = s AND
odor = p: p (256.0)

stalk-shape = e AND
```

```
odor = c: p (192.0)

gill-size = n AND
stalk-surface-above-ring = s AND
population = v: e (192.0)

gill-size = b: p (108.0)

stalk-surface-below-ring = s AND
bruises? = f: e (60.0)

stalk-surface-below-ring = y: p (40.0)

bruises? = f: e (36.0)

: p (8.0)

Number of Rules  :  13
```

5 Do any anomalies exist in the dataset?

Unkonwn, but it could exist in a certain possibility. There are some mushrooms that do not match rules which cover most of the dataset. These mushrooms can be considered to be outliers, or they are grouped into other kinds of mushroom, because the given dataset doesn't include too many these kinds. If this dataset can represent all kinds of mushroom in the world, than we can say these are anomalies exist.

6 Can any association rules be generated from this dataset?

I used Weka and use apriori association algorithm with default minimum support and minimum confidence (0.95 and 0.9) to generate top 10 rules. Belowing are the rules generated when all attributes are considered.

```
 1. veil-color=w 7924 ==> veil-type=p 7924    conf:(1)
 2. gill-attachment=f 7914 ==> veil-type=p 7914    conf:(1)
 3. gill-attachment=f veil-color=w 7906 ==> veil-type=p 7906    conf:(1)
 4. gill-attachment=f 7914 ==> veil-color=w 7906    conf:(1)
 5. gill-attachment=f veil-type=p 7914 ==> veil-color=w 7906    conf:(1)
 6. gill-attachment=f 7914 ==> veil-type=p veil-color=w 7906    conf:(1)
 7. veil-color=w 7924 ==> gill-attachment=f 7906    conf:(1)
 8. veil-type=p veil-color=w 7924 ==> gill-attachment=f 7906    conf:(1)
 9. veil-color=w 7924 ==> gill-attachment=f veil-type=p 7906    conf:(1)
10. veil-type=p 8124 ==> veil-color=w 7924    conf:(0.98)
```

All the rules are related to veil-type. They are useless. I run the algorithm again without considering veil-type, then the followin rules are generated:

```
 1. veil-color=w ring-number=o 7288 ==> gill-attachment=f 7288    conf:(1)
 2. gill-attachment=f gill-spacing=c 6602 ==> veil-color=w 6602    conf:(1)
 3. gill-spacing=c veil-color=w ring-number=o 6272 ==> gill-attachment=f 6272    conf:(1)
 4. gill-attachment=f gill-spacing=c ring-number=o 6272 ==> veil-color=w 6272    conf:(1)
 5. gill-attachment=f gill-size=b 5402 ==> veil-color=w 5402    conf:(1)
 6. stalk-surface-above-ring=s veil-color=w 4984 ==> gill-attachment=f 4984    conf:(1)
 7. gill-attachment=f stalk-surface-above-ring=s 4984 ==> veil-color=w 4984    conf:(1)
 8. gill-size=b veil-color=w ring-number=o 4784 ==> gill-attachment=f 4784    conf:(1)
 9. gill-attachment=f gill-size=b ring-number=o 4784 ==> veil-color=w 4784    conf:(1)
10. stalk-surface-below-ring=s veil-color=w 4744 ==> gill-attachment=f 4744    conf:(1)
```

Again, these rules are meaningless. In order to try and find more useful rules, I remove gill-attachment, gill-spacing, veil-color. The following rules are generated.

```
 1. stalk-shape=t 4608 ==> ring-number=o 4608    conf:(1)
 2. population=v 4040 ==> ring-number=o 3952    conf:(0.98)
 3. class=p 3916 ==> ring-number=o 3808    conf:(0.97)
 4. stalk-root=b 3776 ==> ring-number=o 3656    conf:(0.97)
 5. class=e 4208 ==> gill-size=b 3920    conf:(0.93)
 6. bruises?=f 4748 ==> ring-number=o 4408    conf:(0.93)
 7. ring-type=p 3968 ==> stalk-surface-above-ring=s 3664    conf:(0.92)
 8. stalk-surface-above-ring=s 5176 ==> ring-number=o 4736    conf:(0.91)
 9. stalk-surface-above-ring=s stalk-surface-below-ring=s 4156
 ==> ring-number=o 3788    conf:(0.91)
10. stalk-surface-below-ring=s 4936 ==> ring-number=o 4496    conf:(0.91)
```

Keep remoing some attributes which are not that useful and dominant in the 10 rules. I remove ring-number. The following rules are generated.

```
 1. odor=n gill-size=b 3288 ==> class=e 3216    conf:(0.98)
 2. bruises?=t stalk-surface-below-ring=s 3040 ==> stalk-surface-above-ring=s 2968
 conf:(0.98)
 3. odor=n 3528 ==> class=e 3408    conf:(0.97)
 4. stalk-surface-below-ring=s ring-type=p 3472 ==> stalk-surface-above-ring=s 3328
 conf:(0.96)
 5. bruises?=t 3376 ==> stalk-surface-above-ring=s 3232    conf:(0.96)
 6. bruises?=t ring-type=p 3184 ==> stalk-surface-above-ring=s 3040    conf:(0.95)
 7. gill-size=b stalk-surface-above-ring=s stalk-surface-below-ring=s 3064
 ==> class=e 2920    conf:(0.95)
 8. bruises?=t gill-size=b 3016 ==> stalk-surface-above-ring=s 2872    conf:(0.95)
 9. class=e ring-type=p 3152 ==> stalk-surface-above-ring=s 2992    conf:(0.95)
10. class=e odor=n 3408 ==> gill-size=b 3216    conf:(0.94)
```

I remove stalk-surface-above-ring. The following rules are generated.

```
 1. odor=n gill-size=b 3288 ==> class=e 3216    conf:(0.98)
 2. odor=n 3528 ==> class=e 3408    conf:(0.97)
 3. class=e odor=n 3408 ==> gill-size=b 3216    conf:(0.94)
 4. bruises?=t 3376 ==> ring-type=p 3184    conf:(0.94)
 5. class=e ring-type=p 3152 ==> gill-size=b 2960    conf:(0.94)
 6. bruises?=t stalk-surface-below-ring=s 3040 ==> ring-type=p 2848    conf:(0.94)
 7. gill-size=b stalk-surface-below-ring=s 3400 ==> class=e 3184    conf:(0.94)
 8. class=e stalk-surface-below-ring=s 3400 ==> gill-size=b 3184    conf:(0.94)
 9. odor=n 3528 ==> gill-size=b 3288    conf:(0.93)
10. class=e 4208 ==> gill-size=b 3920    conf:(0.93)
```

I remove gill-size and ring-type to generate the rules again.

```
 1. odor=n stalk-shape=t 2496 ==> class=e 2496    conf:(1)
 2. stalk-surface-below-ring=k 2304 ==> bruises?=f 2304    conf:(1)
 3. odor=f 2160 ==> class=p 2160    conf:(1)
 4. class=p stalk-surface-below-ring=k 2160 ==> bruises?=f 2160    conf:(1)
 5. stalk-shape=t stalk-root=b 2112 ==> bruises?=t 2112    conf:(1)
 6. bruises?=t stalk-shape=t 2112 ==> stalk-root=b 2112    conf:(1)
 7. odor=n stalk-shape=t stalk-surface-below-ring=s 2112 ==> class=e 2112    conf:(1)
 8. bruises?=t odor=n 2032 ==> stalk-surface-below-ring=s 2032    conf:(1)
 9. odor=n stalk-surface-below-ring=s 2872 ==> class=e 2792    conf:(0.97)
10. odor=n 3528 ==> class=e 3408    conf:(0.97)
```

These are the best top 10 rules found 5 of which are related with the conclusion about edible or poisonous. Continuing to remove attribute bruises? doesn't improve the rules. Results are shown below.

```
 1. odor=n stalk-shape=t 2496 ==> class=e 2496    conf:(1)
 2. odor=f 2160 ==> class=p 2160    conf:(1)
 3. odor=n stalk-shape=t stalk-surface-below-ring=s 2112 ==> class=e 2112    conf:(1)
 4. class=e stalk-shape=t stalk-root=b 1824 ==> stalk-surface-below-ring=s 1824    conf:(1)
 5. stalk-shape=t stalk-root=b habitat=d 1824 ==> class=e 1824    conf:(1)
 6. class=e stalk-shape=t habitat=d 1824 ==> stalk-root=b 1824    conf:(1)
 7. class=e stalk-shape=t stalk-root=b 1824 ==> habitat=d 1824    conf:(1)
 8. class=e stalk-surface-below-ring=s habitat=d 1824 ==> stalk-shape=t 1824    conf:(1)
 9. class=e stalk-shape=t habitat=d 1824 ==> stalk-surface-below-ring=s 1824    conf:(1)
10. class=e stalk-surface-below-ring=s habitat=d 1824 ==> stalk-root=b 1824    conf:(1)
```