IST 565 - Final Project

Kelly Hwang

September 16, 2018

Introduction

Over the last 5 years, mushroom hunting has been gaining popularity with nature enthusiasts. Without any expensive gear or membership fees, hunters can enjoy the thrill of finding edible fungi relatively easily, while partaking in the great outdoors. However, it is extremely important for mushroom hunters to understand where to search, and what to search for. There are various books and guides available to educate readers on what's edible and what's not. It is of inevitable importance that hunters understand the difference between a poisonous and edible mushroom.

I will be using the Mushroom Classification dataset found on Kaggle. This dataset was originally contributed to the UCI Machine Learning repository 30 years ago, and contains descriptions of hypothetical samples belonging to over 23 species of gilled mushrooms, specifically in the Agaricus and Leiota family (as cited from the Kaggle page).

Each species is labeled as edible, poisonous, or unknown edibility. There is no simple rule that determines the edibility of a mushroom, and so I hope to build models that will be able to predict whether or not mushrooms are poisonous based off their features.

First importing data:

```
mushrooms <- read.csv("~/Syracuse - Grad School/IST 565 - Data Mining/mushroo
ms-data.csv")
```

A quick look at the data shows 23 different variables:

```
str(mushrooms)
```

```
## 'data.frame':    8124 obs. of  23 variables:
##  $ class                   : Factor w/ 2 levels "e","p": 2 1 1 2 1 1 1 1 2
1 ...
##  $ cap.shape               : Factor w/ 6 levels "b","c","f","k",..: 6 6 1
6 6 6 1 1 6 1 ...
##  $ cap.surface             : Factor w/ 4 levels "f","g","s","y": 3 3 3 4 3
4 3 4 4 3 ...
##  $ cap.color               : Factor w/ 10 levels "b","c","e","g",..: 5 10
9 9 4 10 9 9 9 10 ...
##  $ bruises                 : Factor w/ 2 levels "f","t": 2 2 2 2 1 2 2 2 2
2 ...
##  $ odor                    : Factor w/ 9 levels "a","c","f","l",..: 7 1 4
7 6 1 1 4 7 1 ...
##  $ gill.attachment         : Factor w/ 2 levels "a","f": 2 2 2 2 2 2 2 2 2
```

```
2 ...
##  $ gill.spacing            : Factor w/ 2 levels "c","w": 1 1 1 1 2 1 1 1 1
1 ...
##  $ gill.size               : Factor w/ 2 levels "b","n": 2 1 1 2 1 1 1 1 2
1 ...
##  $ gill.color              : Factor w/ 12 levels "b","e","g","h",..: 5 5 6
6 5 6 3 6 8 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 "?","b","c","e",..: 4 3 3
4 4 3 3 3 4 3 ...
##  $ stalk.surface.above.ring: Factor w/ 4 levels "f","k","s","y": 3 3 3 3 3
3 3 3 3 3 ...
##  $ stalk.surface.below.ring: Factor w/ 4 levels "f","k","s","y": 3 3 3 3 3
3 3 3 3 3 ...
##  $ stalk.color.above.ring  : Factor w/ 9 levels "b","c","e","g",..: 8 8 8
8 8 8 8 8 8 8 ...
##  $ stalk.color.below.ring  : Factor w/ 9 levels "b","c","e","g",..: 8 8 8
8 8 8 8 8 8 8 ...
##  $ veil.type               : Factor w/ 1 level "p": 1 1 1 1 1 1 1 1 1 1 ..
.
##  $ veil.color              : Factor w/ 4 levels "n","o","w","y": 3 3 3 3 3
3 3 3 3 3 ...
##  $ ring.number             : Factor w/ 3 levels "n","o","t": 2 2 2 2 2 2 2
2 2 2 ...
##  $ ring.type               : Factor w/ 5 levels "e","f","l","n",..: 5 5 5
5 1 5 5 5 5 5 ...
##  $ spore.print.color       : Factor w/ 9 levels "b","h","k","n",..: 3 4 4
3 4 3 3 4 3 3 ...
##  $ population              : Factor w/ 6 levels "a","c","n","s",..: 4 3 3
4 1 3 3 4 5 4 ...
##  $ habitat                 : Factor w/ 7 levels "d","g","l","m",..: 6 2 4
6 2 2 4 4 2 4 ...
```

```
summary(mushrooms)
```

```
##  class    cap.shape cap.surface   cap.color    bruises       odor
##  e:4208   b: 452    f:2320      n      :2284   f:4748   n      :3528
##  p:3916   c:   4    g:   4      g      :1840   t:3376   f      :2160
##           f:3152    s:2556      e      :1500            s      : 576
##           k: 828    y:3244      y      :1072            y      : 576
##           s:  32                w      :1040            a      : 400
##           x:3656                b      : 168            l      : 400
##                                 (Other): 220            (Other): 484
##  gill.attachment gill.spacing gill.size   gill.color   stalk.shape
##  a: 210          c:6812       b:5612    b      :1728   e:3516
##  f:7914          w:1312       n:2512    p      :1492   t:4608
##                                         w      :1202
##                                         n      :1048
##                                         g      : 752
```

```
##                                         h      : 732
##                                         (Other):1170
##  stalk.root stalk.surface.above.ring stalk.surface.below.ring
##  ?:2480     f: 552                   f: 600
##  b:3776     k:2372                   k:2304
##  c: 556     s:5176                   s:4936
##  e:1120     y:  24                   y: 284
##  r: 192
##
##
##  stalk.color.above.ring stalk.color.below.ring veil.type veil.color
##  w      :4464           w      :4384           p:8124    n:  96
##  p      :1872           p      :1872                     o:  96
##  g      : 576           g      : 576                     w:7924
##  n      : 448           n      : 512                     y:   8
##  b      : 432           b      : 432
##  o      : 192           o      : 192
##  (Other): 140           (Other): 156
##  ring.number ring.type spore.print.color population habitat
##  n:  36      e:2776    w      :2388      a: 384     d:3148
##  o:7488      f:  48    n      :1968      c: 340     g:2148
##  t: 600      l:1296    k      :1872      n: 400     l: 832
##              n:  36    h      :1632      s:1248     m: 292
##              p:3968    r      :  72      v:4040     p:1144
##                        b      :  48      y:1712     u: 368
##                        (Other): 144                 w: 192
```

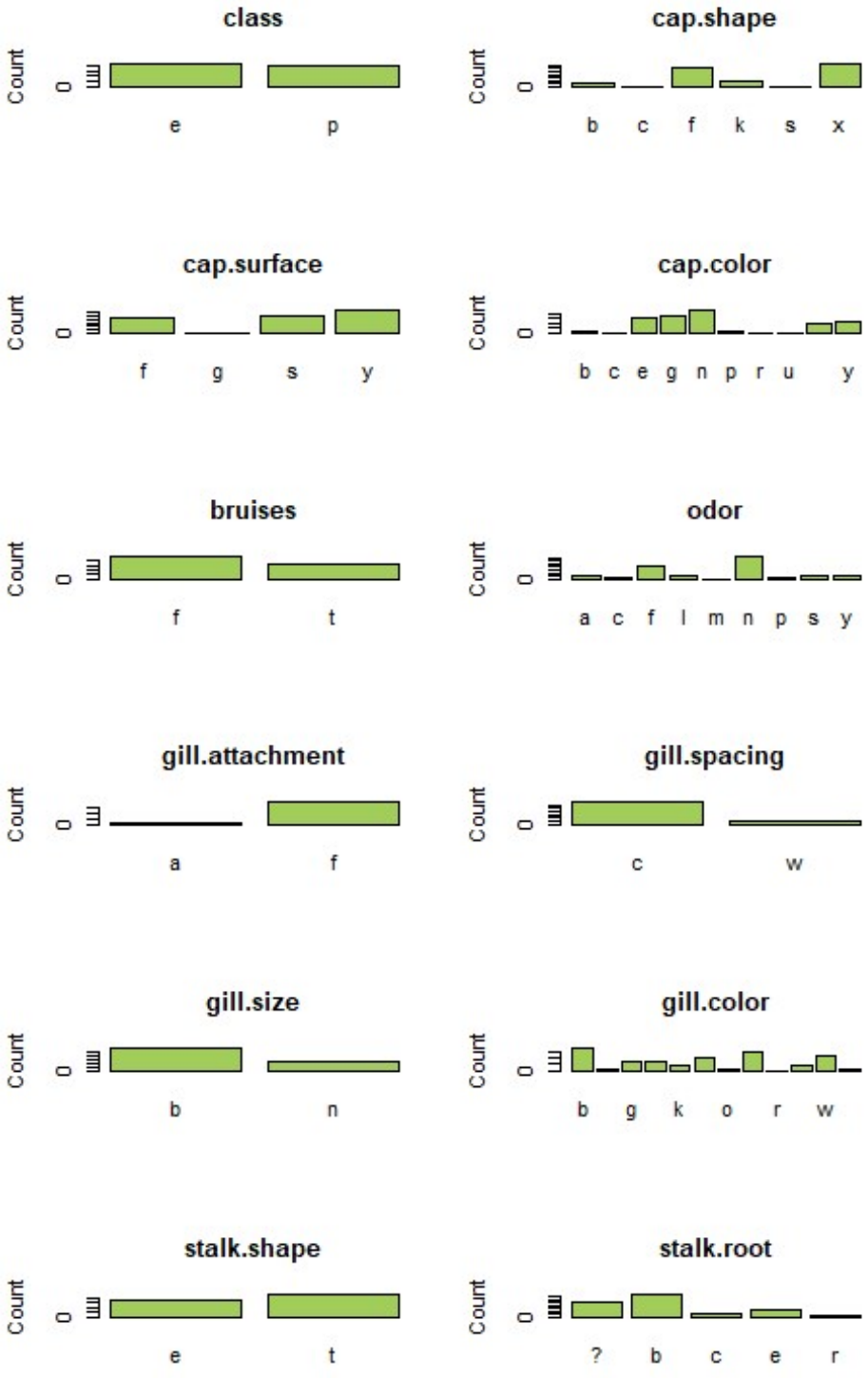
```
View(head(mushrooms))
```

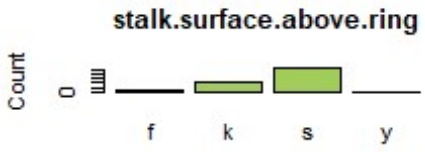
Check for any null values in the data set:

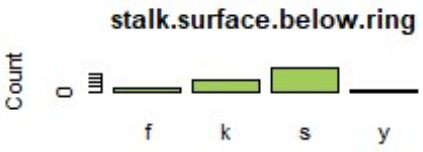
```
table(is.na(mushrooms))
```

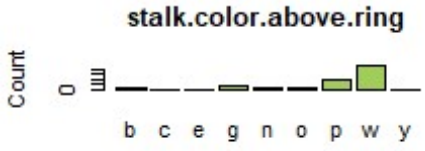
```
##
##  FALSE
## 186852
```

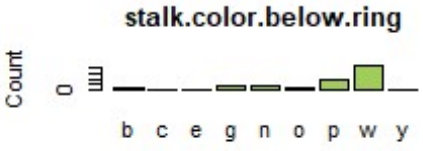
Plotting all variables to see distributions:

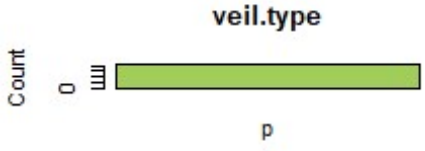


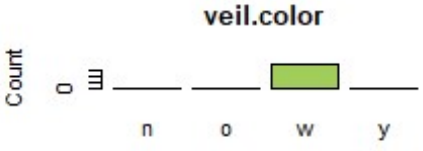


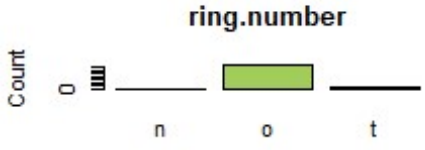


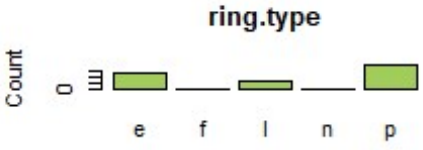


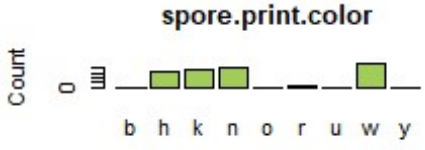


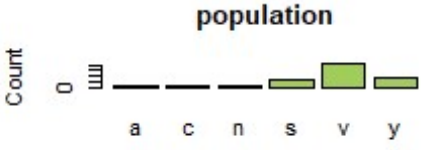












All mushrooms are classified as either edible (e) or poisonous (p). The goal is to determine if we can classify mushrooms as e/p based on their given attributes.

How many mushrooms in the data are edible vs poisonous?

```r
table(mushrooms$class)
```

```
##
##    e    p
## 4208 3916
```

```
## Percent of edible class: 51.79714 %
```

```
## Percent of poisonous class: 48.20286 %
```

Converting data from factor to numeric for models:

```r
mushrooms_data <- mushrooms[,2:23]
mushrooms_class <- mushrooms[,1]
mushrooms_data <- sapply(mushrooms_data, function(x) as.numeric(as.factor(x))
)
mushrooms_new <- data.frame(class = mushrooms_class, mushrooms_data)
```

Remove variable veil.type due to zero variance as shown by the distribution plot above:

```r
mushrooms_new$veil.type <- NULL
```

Split data into 70% training and 30% for testing:

```r
set.seed(0)
sample <- sample(2, nrow(mushrooms_new), replace = TRUE, prob = c(0.7, 0.3))
training <- mushrooms_new[sample == 1,]
testing <- mushrooms_new[sample == 2,]
```

Checking on the dimensions of training and testing:

```
## [1] 5658   22
```

```
## [1] 2466   22
```

Extract original class from testing set:

```r
testing_class <- testing[,1]
```

Remove class from testing set for modeling:

```r
testing_noclass <-testing[,2:22]
```

Building Naive Bayes model:

```r
nb_model <- naiveBayes(class ~ ., data = training, laplace = 1)
nb_class <- predict(nb_model, newdata = testing_noclass)
```

Accuracy against original labels:

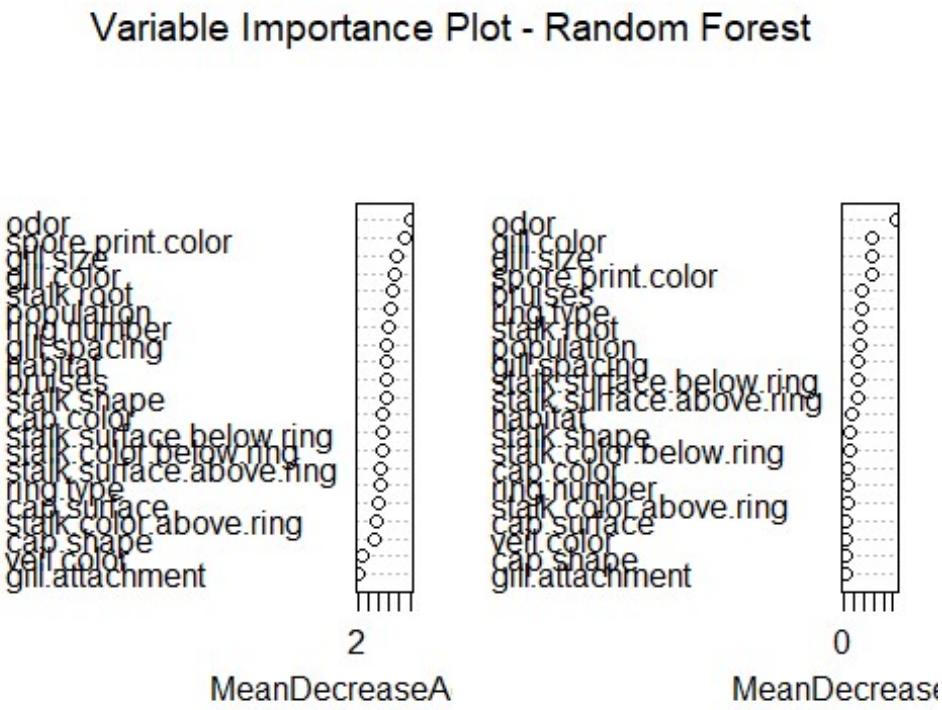
```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction    e    p
```

```
##          e 1159   93
##          p  106 1108
##
##                Accuracy : 0.9193
##                  95% CI : (0.9078, 0.9298)
##     No Information Rate : 0.513
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.8385
##  Mcnemar's Test P-Value : 0.395
##
##             Sensitivity : 0.9162
##             Specificity : 0.9226
##          Pos Pred Value : 0.9257
##          Neg Pred Value : 0.9127
##              Prevalence : 0.5130
##          Detection Rate : 0.4700
##    Detection Prevalence : 0.5077
##       Balanced Accuracy : 0.9194
##
##        'Positive' Class : e
##
```

Random Forest model with 100 trees:

```
rndf_model <- randomForest(class ~ ., data = training, ntree = 100, importanc
e = TRUE)
rndf_class <- predict(rndf_model, newdata = testing_noclass)
```

Variable Importance Plot for random forest:



Accuracy against original labels:

```
confusionMatrix(testing_class, rndf_class)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    e    p
##          e 1252    0
##          p    0 1214
##
##                Accuracy : 1
##                  95% CI : (0.9985, 1)
##     No Information Rate : 0.5077
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0000
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 1.0000
##              Prevalence : 0.5077
##          Detection Rate : 0.5077
##    Detection Prevalence : 0.5077
```

```
##       Balanced Accuracy : 1.0000
##
##        'Positive' Class : e
##
```

KSVM model:

```
svm_model <- ksvm(class ~ ., data = training, kernel = "polydot", kpar = list
(degree = 3), cross = 3)
svm_class <- predict(svm_model, newdata = testing_noclass)
```

Accuracy against original labels:

```
confusionMatrix(testing_class, rndf_class)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    e    p
##          e 1252    0
##          p    0 1214
##
##                Accuracy : 1
##                  95% CI : (0.9985, 1)
##     No Information Rate : 0.5077
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0000
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 1.0000
##              Prevalence : 0.5077
##          Detection Rate : 0.5077
##    Detection Prevalence : 0.5077
##       Balanced Accuracy : 1.0000
##
##        'Positive' Class : e
##
```

End of Analysis Findings

Veil type was the same among the entire mushroom data population. Odor, Stalk Root, Gill Color are the most important variables. Naive Bayes classification model produced an outcome of 91% accuracy. Random Forest model and SVM produced an outcome of 100% accuracy.