A Statisticians Guide to Mushrooms

Anthony Anderson

April 29, 2019

This project's aim is to identify the 5 most important variables in identifying edible or poisonous mushrooms. This dataset is taken from Kaggle, with 22 predictor variables and the binary response variable, "class", where p=poisonous, and e=edible. Through various dimension reduction methods I will attempt to identify the 5 most useful predictors for a camper, shroomer, or just curious laymen to use to more safely identify mushrooms to harvest and potentially eat. Note that classifying mushrooms through phenotypical measurements (color, size, etc.) is extremely difficult due to the wide variety in different species and overlap between phenotypic characteristics. The consensus of shroomers and mycologists is to use spore patterns and genetic samples when possible before risking eating or using a mushroom. Always follow the experts! This is no way a replacement for scientific and expert research, and is just an exploratory analysis for a well-documented problem in the shrooming world. Below is the context for each variable taken from the Kaggle dataset at https://www.kaggle.com/uciml/mushroom-classification:

cap-shape: bell=b,conical=c,convex=x,flat=f, knobbed=k,sunken=s

cap-surface: fibrous=f,grooves=g,scaly=y,smooth=s

cap-color: brown=n,buff=b,cinnamon=c,gray=g,green=r,pink=p,purple=u,red=e,white=w,yellow=y

bruises: bruises=t,no=f

odor: almond=a,anise=l,creosote=c,fishy=y,foul=f,musty=m,none=n,pungent=p,spicy=s

gill-attachment: attached=a,descending=d,free=f,notched=n

gill-spacing: close=c,crowded=w,distant=d

gill-size: broad=b,narrow=n

gill-color: black=k,brown=n,buff=b,chocolate=h,gray=g, green=r,orange=o,pink=p,purple=u,red=e,white=w,yellow=y

stalk-shape: enlarging=e,tapering=t

stalk-root: bulbous=b,club=c,cup=u,equal=e,rhizomorphs=z,rooted=r,missing=?

stalk-surface-above-ring: fibrous=f,scaly=y,silky=k,smooth=s

stalk-surface-below-ring: fibrous=f,scaly=y,silky=k,smooth=s

stalk-color-above-ring: brown=n,buff=b,cinnamon=c,gray=g,orange=o,pink=p,red=e,white=w,yellow=y

stalk-color-below-ring: brown=n,buff=b,cinnamon=c,gray=g,orange=o,pink=p,red=e,white=w,yellow=y

veil-type: partial=p,universal=u

veil-color: brown=n,orange=o,white=w,yellow=y

ring-number: none=n,one=o,two=t

ring-type: cobwebby=c,evanescent=e,flaring=f,large=l,none=n,pendant=p,sheathing=s,zone=z

spore-print-color: black=k,brown=n,buff=b,chocolate=h,green=r,orange=o,purple=u,white=w,yellow=y

population: abundant=a,clustered=c,numerous=n,scattered=s,several=v,solitary=y

habitat: grasses=g,leaves=l,meadows=m,paths=p,urban=u,waste=w,woods=d

Note all predictors are qualitative factored variables, which is realistic and useful for real world application since campers don't need to bring measurement tools with them.

Methodology:

To me, the most important operation is dimension reduction to reduce the predictors used to just 5 or less, something a shroomer can count on their hand.
Best subset selection is obviously the best idea in terms of model selection, so I will try that first. If the selection runs, I will move forward with the variables selected this way. If the best subset selection doesn't work (R finds it too big to run), I will run stepwise. The two best methods to use for maximum interpretability is logistic regression, and classification trees. I'll fit a long tree, a pruned tree, and then use randomForest() for bagging. Partial Least Squares or PCR can be helpful in accuracy, but since the point of this analysis is to make an easily interpretable set of rules for shrooming, they will be avoided.

Amazingly, even the most simple classification method, logistic, has 0 test error when the probability threshold is set to .7 when using only the 4 variables from the best subset selection. The classification tree also has a 0 error rate when using all predictors and Gini index for splits. The pruned tree, however, misclassified 1 observation, and the in the worst way: truly poisonous, but classified as edible. The bagged method, using default 500 trees and all predictors at each split, has a 0 test error rate.

I would choose the classification tree as my final method. Not only does it have a 0 test error rate, but because of how easy it is to follow, even the most amateur shroomer can follow the splits step by step. Although there are more splits so it isn't as simple, I would feel safer following a few extra splits on a model with 0 test error than one that identified poisonous mushrooms as edible as is in the pruned tree case. The splits are easily followable: smell the mushroom and identify its odor, then check the color of it's spore sprint, etc. Since all the variables are factors based on physical attributes, you can simply look at the mushroom to identify if you should keep it and harvest more, or throw it away.

Overall I'm very surprised with how well separated the data was. My very basic knowledge of mushrooms tells me that it shouldn't be possible, so it's likely that this dataset is not representative of all mushrooms in general. Specifically, the data is drawn from samples belonging to the Agaricus and Lepiota families. Current estimates of total types of mushrooms are around 10,000 but mycologists say that this could be only a tiny fraction, with many yet to be discovered and classified. There is a lot of overlap between physical characteristics between types as well, so again I would warn against using this classification tree seriously. At most, it could be used to decide what types of mushrooms to haul back home when space/weight is limited, with true genetic testing being the next step before eating or using them in anyway whatsoever.

Appendix

```
mushroom.data<-read.csv("mushrooms.csv")
mushroom.data<- mushroom.data[,-17] #veil type has only one level in this set (p),
#so we remove it to make analyzation possible
str(mushroom.data)
```

```
## 'data.frame':    8124 obs. of  22 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.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 ...
```

```
set.seed(42)
train<-sample(nrow(mushroom.data), .8*nrow(mushroom.data))
mushroom.train <- mushroom.data[train,]
mushroom.test <- mushroom.data[-train,]
```

```
#library(leaps)
#best.subset<- regsubsets(class~.-class,data=mushroom.data,nvmax=5, really.big=T)
#summary(best.subset)
#5(4) variables selected by subset: odor-n, stalk.root-c, stalk.root-r,
#stalk.surface.below.ringy, spore.print.color-r
```

```
best.sub.fit<- glm(class~odor+stalk.root+stalk.surface.below.ring+spore.print.color,
                   data=mushroom.train, family=binomial())
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(best.sub.fit)
```

```
##
## Call:
## glm(formula = class ~ odor + stalk.root + stalk.surface.below.ring +
##     spore.print.color, family = binomial(), data = mushroom.train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.29009  -0.00001   0.00000   0.00001   2.97601
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)               -6.454e+01  2.767e+04  -0.002    0.998
## odorc                      7.063e+01  2.020e+04   0.003    0.997
## odorf                      6.857e+01  1.926e+04   0.004    0.997
## odorl                     -5.795e-03  1.026e+04   0.000    1.000
## odorm                      2.414e+01  2.764e+04   0.001    0.999
## odorn                      2.146e+01  1.770e+04   0.001    0.999
## odorp                      6.986e+01  2.039e+04   0.003    0.997
## odors                      6.732e+01  1.886e+04   0.004    0.997
## odory                      6.728e+01  1.889e+04   0.004    0.997
## stalk.rootb               -8.204e-01  6.186e-01  -1.326    0.185
## stalk.rootc                2.081e+01  1.535e+04   0.001    0.999
## stalk.roote               -4.887e-02  5.827e+03   0.000    1.000
## stalk.rootr                1.600e+01  1.962e+04   0.001    0.999
```

```
## stalk.surface.below.ringk -2.470e-02  7.413e+03   0.000    1.000
## stalk.surface.below.rings  1.851e+01  7.298e+03   0.003    0.998
## stalk.surface.below.ringy  2.319e+01  7.298e+03   0.003    0.997
## spore.print.colorh         2.099e+01  2.145e+04   0.001    0.999
## spore.print.colork         7.775e-01  2.044e+04   0.000    1.000
## spore.print.colorn         8.000e-01  2.040e+04   0.000    1.000
## spore.print.coloro        -2.900e-07  2.897e+04   0.000    1.000
## spore.print.colorr         4.995e+01  2.678e+04   0.002    0.999
## spore.print.coloru         2.229e+01  3.322e+04   0.001    0.999
## spore.print.colorw         2.097e+01  1.998e+04   0.001    0.999
## spore.print.colory        -2.900e-07  2.938e+04   0.000    1.000
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 9003.68  on 6498  degrees of freedom
## Residual deviance:  125.44  on 6475  degrees of freedom
## AIC: 173.44
##
## Number of Fisher Scoring iterations: 23
```

```
best.sub.prob<-predict(best.sub.fit,mushroom.test,type="response")
best.sub.pred<-factor(best.sub.prob>.7,levels=c(FALSE,TRUE),labels=c("Edible","Poisonous"))
best.sub.error<-table(mushroom.test$class,best.sub.pred,dnn=c("Actual","Predicted"))
best.sub.error
```

```
##       Predicted
## Actual Edible Poisonous
##      e    861         0
##      p      0       764
```

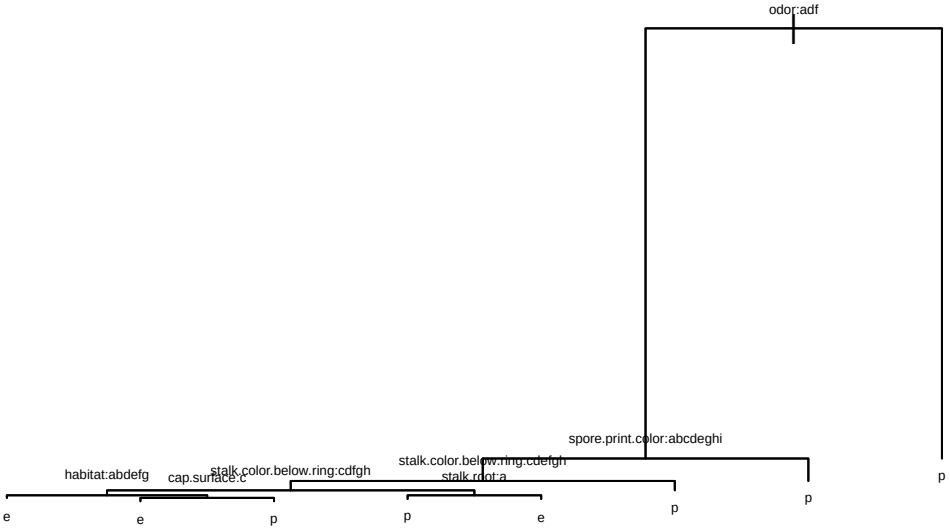
```
library(tree)
d.tree<- tree(class~.,data=mushroom.train,split="gini")
d.tree.pred<-predict(d.tree,mushroom.test,type="class")
d.tree.table<-table(mushroom.test$class,d.tree.pred,dnn=c("Actual","Predicted"))
d.tree.table
```

```
##       Predicted
## Actual   e   p
##      e 861   0
##      p   0 764
```

```
summary(d.tree)
```

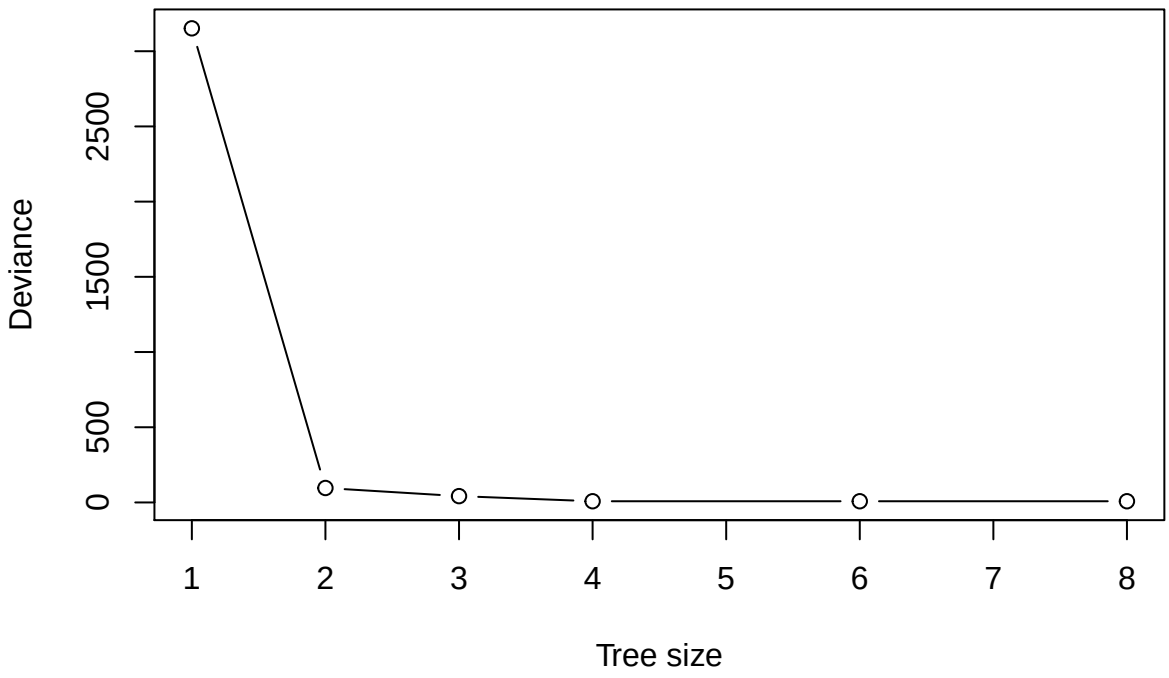
```
##
## Classification tree:
## tree(formula = class ~ ., data = mushroom.train, split = "gini")
## Variables actually used in tree construction:
## [1] "odor"                   "spore.print.color"
## [3] "stalk.color.below.ring" "habitat"
## [5] "cap.surface"            "stalk.root"
## Number of terminal nodes:  8
## Residual mean deviance:  0 = 0 / 6491
## Misclassification error rate: 0 = 0 / 6499
```

```
plot(d.tree)
text(d.tree, cex = 0.4)
```

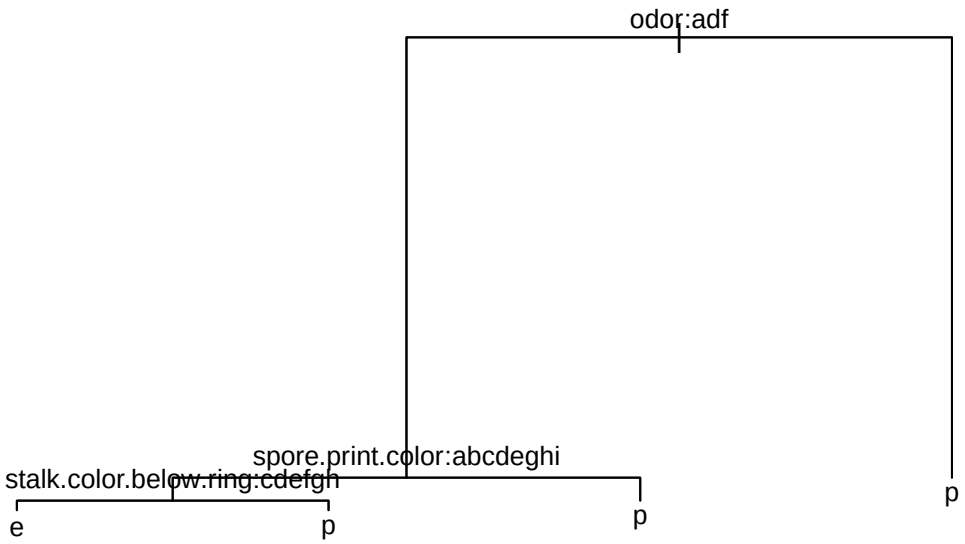



```
set.seed(42)
cross.tree <- cv.tree(d.tree, FUN = prune.misclass)

plot(cross.tree$size, cross.tree$dev, type = "b",
     xlab = "Tree size", ylab = "Deviance")
```




```
prune.tree <- prune.misclass(d.tree, best = 4)
plot(prune.tree)
text(prune.tree, cex = 0.8)
```




```
prune.tree.pred<-predict(prune.tree,mushroom.test,type="class")
prune.tree.table<-table(mushroom.test$class,prune.tree.pred,dnn=c("Actual","Predicted"))
prune.tree.table
```

```
##       Predicted
## Actual   e   p
##      e 861   0
##      p   1 763
```

```
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
set.seed(42)
bag.fit<-randomForest(class~.,data=mushroom.data, subset=train, mtry=21,importance=TRUE)
bag.fit
```

```
##
## Call:
##  randomForest(formula = class ~ ., data = mushroom.data, mtry = 21,      importance = TRUE, subset =
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 21
##
##         OOB estimate of  error rate: 0%
## Confusion matrix:
```

```
##      e    p class.error
## e 3347    0           0
## p    0 3152           0
```

```
bag.pred <- predict(bag.fit,mushroom.test,type="class")
bag.fit.table<- table(mushroom.test$class,bag.pred,dnn=c("Actual","Predicted"))
bag.fit.table
```

```
##       Predicted
## Actual   e   p
##      e 861   0
##      p   0 764
```