Mushroom Classification

Vivek Pruthi, Rajesh Grandhi and Jyothi Pulimamidi

July 12, 2017

Figure 1:

```
library(ggplot2)
library(caret)
library(ggthemes)
```

Importing the data

```
mushrooms_data<-read.csv("C:\\vik\\2017\\personal\\DSLA\\course material\\project 1 files\\mushrooms.csv
```

Exploring the data

Dimensions of the mushroom datasets are :

```
dim(mushrooms_data)
```

```
## [1] 8124   23
```

Fields in the dataset are:

```
names(mushrooms_data)
```

```
##  [1] "class"                     "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.type"                "veil.color"
## [19] "ring.number"              "ring.type"
## [21] "spore.print.color"        "population"
## [23] "habitat"
```

Following are the definitions of these fields:

- Fields/Attributes/features of the dataframe are
  - classes: edible=e, poisonous=p
  - 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

Let's have a look at the structure of the dataset :

```
str(mushrooms_data)
```

```
## '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 ...
```

It is good to have a little peek at a slice of data.

```
head(mushrooms_data)
```

```
##   class cap.shape cap.surface cap.color bruises odor gill.attachment
## 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
##   gill.spacing gill.size gill.color stalk.shape stalk.root
## 1            c         n          k           e          e
## 2            c         b          k           e          c
## 3            c         b          n           e          c
## 4            c         n          n           e          e
## 5            w         b          k           t          e
## 6            c         b          n           e          c
##   stalk.surface.above.ring stalk.surface.below.ring stalk.color.above.ring
## 1                        s                        s                      w
## 2                        s                        s                      w
## 3                        s                        s                      w
## 4                        s                        s                      w
## 5                        s                        s                      w
## 6                        s                        s                      w
##   stalk.color.below.ring veil.type veil.color ring.number ring.type
## 1                      w         p          w           o         p
## 2                      w         p          w           o         p
## 3                      w         p          w           o         p
## 4                      w         p          w           o         p
```

```
## 5                    w         p          w           o         e
## 6                    w         p          w           o         p
##   spore.print.color population habitat
## 1                 k          s       u
## 2                 n          n       g
## 3                 n          n       m
## 4                 k          s       u
## 5                 n          a       g
## 6                 k          n       g
```

```
tail(mushrooms_data)
```

```
##      class cap.shape cap.surface cap.color bruises odor gill.attachment
## 8119     p         k           y         n       f    f               f
## 8120     e         k           s         n       f    n               a
## 8121     e         x           s         n       f    n               a
## 8122     e         f           s         n       f    n               a
## 8123     p         k           y         n       f    y               f
## 8124     e         x           s         n       f    n               a
##      gill.spacing gill.size gill.color stalk.shape stalk.root
## 8119            c         n          b           t          ?
## 8120            c         b          y           e          ?
## 8121            c         b          y           e          ?
## 8122            c         b          n           e          ?
## 8123            c         n          b           t          ?
## 8124            c         b          y           e          ?
##      stalk.surface.above.ring stalk.surface.below.ring
## 8119                        k                        s
## 8120                        s                        s
## 8121                        s                        s
## 8122                        s                        s
## 8123                        s                        k
## 8124                        s                        s
##      stalk.color.above.ring stalk.color.below.ring veil.type veil.color
## 8119                      p                      w         p          w
## 8120                      o                      o         p          o
## 8121                      o                      o         p          n
## 8122                      o                      o         p          o
## 8123                      w                      w         p          w
## 8124                      o                      o         p          o
##      ring.number ring.type spore.print.color population habitat
## 8119           o         e                 w          v       d
## 8120           o         p                 b          c       l
## 8121           o         p                 b          v       l
## 8122           o         p                 b          c       l
## 8123           o         e                 w          v       l
## 8124           o         p                 o          c       l
```

It is pertinent from the data that the fields in the dataset are of type factor i.e. these are catgorical variables with different levels. it is better to visualize this data . We will first check the summary and then explore the data visually :

```
summary(mushrooms_data)
```

```
##  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
```

As few of the levels are shown in summary as (others), let's check what the complete levels are of all the categorical variables in this dataset :

```
for(i in 1:23){
  print(names(mushrooms_data[i]))
  print(levels(mushrooms_data[,i]))
}
```

```
## [1] "class"
## [1] "e" "p"
## [1] "cap.shape"
## [1] "b" "c" "f" "k" "s" "x"
## [1] "cap.surface"
## [1] "f" "g" "s" "y"
## [1] "cap.color"
##  [1] "b" "c" "e" "g" "n" "p" "r" "u" "w" "y"
```

```
## [1] "bruises"
## [1] "f" "t"
## [1] "odor"
## [1] "a" "c" "f" "l" "m" "n" "p" "s" "y"
## [1] "gill.attachment"
## [1] "a" "f"
## [1] "gill.spacing"
## [1] "c" "w"
## [1] "gill.size"
## [1] "b" "n"
## [1] "gill.color"
##  [1] "b" "e" "g" "h" "k" "n" "o" "p" "r" "u" "w" "y"
## [1] "stalk.shape"
## [1] "e" "t"
## [1] "stalk.root"
## [1] "?" "b" "c" "e" "r"
## [1] "stalk.surface.above.ring"
## [1] "f" "k" "s" "y"
## [1] "stalk.surface.below.ring"
## [1] "f" "k" "s" "y"
## [1] "stalk.color.above.ring"
## [1] "b" "c" "e" "g" "n" "o" "p" "w" "y"
## [1] "stalk.color.below.ring"
## [1] "b" "c" "e" "g" "n" "o" "p" "w" "y"
## [1] "veil.type"
## [1] "p"
## [1] "veil.color"
## [1] "n" "o" "w" "y"
## [1] "ring.number"
## [1] "n" "o" "t"
## [1] "ring.type"
## [1] "e" "f" "l" "n" "p"
## [1] "spore.print.color"
## [1] "b" "h" "k" "n" "o" "r" "u" "w" "y"
## [1] "population"
## [1] "a" "c" "n" "s" "v" "y"
## [1] "habitat"
## [1] "d" "g" "l" "m" "p" "u" "w"
```

We can check their proportionate distribution too:

```
for(i in 1:23){
  print(names(mushrooms_data[i]))
 print(prop.table((table(mushrooms_data[,i])))*100)
}
```

```
## [1] "class"
##
##        e        p
## 51.79714 48.20286
## [1] "cap.shape"
##
##           b           c           f           k           s           x
##  5.56376169  0.04923683 38.79862137 10.19202363  0.39389463 45.00246184
## [1] "cap.surface"
```

```
##
##          f          g          s          y
## 28.55736091  0.04923683 31.46233383 39.93106844
## [1] "cap.color"
##
##         b         c         e         g         n         p
##  2.0679468  0.5416051 18.4638109 22.6489414 28.1142294  1.7725258
##         r         u         w         y
##  0.1969473  0.1969473 12.8015756 13.1954702
## [1] "bruises"
##
##       f       t
## 58.44412 41.55588
## [1] "odor"
##
##         a         c         f         l         m         n
##  4.9236829  2.3633678 26.5878877  4.9236829  0.4431315 43.4268833
##         p         s         y
##  3.1511571  7.0901034  7.0901034
## [1] "gill.attachment"
##
##        a        f
##  2.584934 97.415066
## [1] "gill.spacing"
##
##       c       w
## 83.85032 16.14968
## [1] "gill.size"
##
##       b       n
## 69.07927 30.92073
## [1] "gill.color"
##
##         b         e         g         h         k         n
## 21.2703102  1.1816839  9.2565239  9.0103397  5.0221566 12.9000492
##         o         p         r         u         w         y
##  0.7877893 18.3653373  0.2954210  6.0561300 14.7956672  1.0585918
## [1] "stalk.shape"
##
##       e       t
## 43.27917 56.72083
## [1] "stalk.root"
##
##        ?        b        c        e        r
## 30.526834 46.479567  6.843919 13.786312  2.363368
## [1] "stalk.surface.above.ring"
##
##        f        k        s        y
##  6.794682 29.197440 63.712457  0.295421
## [1] "stalk.surface.below.ring"
##
##        f        k        s        y
##  7.385524 28.360414 60.758247  3.495815
## [1] "stalk.color.above.ring"
```

```
##
##           b           c           e           g           n           o
##  5.31757755  0.44313146  1.18168390  7.09010340  5.51452486  2.36336780
##           p           w           y
## 23.04283604 54.94830133  0.09847366
## [1] "stalk.color.below.ring"
##
##          b          c          e          g          n          o
##  5.3175775  0.4431315  1.1816839  7.0901034  6.3023141  2.3633678
##          p          w          y
## 23.0428360 53.9635647  0.2954210
## [1] "veil.type"
##
##   p
## 100
## [1] "veil.color"
##
##           n           o           w           y
##  1.18168390  1.18168390 97.53815854  0.09847366
## [1] "ring.number"
##
##          n          o          t
##  0.4431315 92.1713442  7.3855244
## [1] "ring.type"
##
##          e          f          l          n          p
## 34.1703594  0.5908419 15.9527326  0.4431315 48.8429345
## [1] "spore.print.color"
##
##          b          h          k          n          o          r
##  0.5908419 20.0886263 23.0428360 24.2245199  0.5908419  0.8862629
##          u          w          y
##  0.5908419 29.3943870  0.5908419
## [1] "population"
##
##         a         c         n         s         v         y
##  4.726736  4.185130  4.923683 15.361891 49.729197 21.073363
## [1] "habitat"
##
##         d         g         l         m         p         u         w
## 38.749385 26.440177 10.241260  3.594289 14.081733  4.529788  2.363368
```

```
library(ggplot2)
library(gridExtra)
 p1<-ggplot(mushrooms_data,aes(x=class))+geom_histogram(stat="count",fill="blue")+ggtitle(label="Poison
 p2<-ggplot(mushrooms_data,aes(x=cap.shape))+geom_histogram(stat="count",aes(fill=class))+ggtitle(label=
 p3<-ggplot(mushrooms_data,aes(x=cap.surface))+geom_histogram(stat="count",aes(fill=class))+ggtitle(lab
 p4<-ggplot(mushrooms_data,aes(x=cap.color))+geom_histogram(stat="count",aes(fill=class))+ggtitle(label=
 p5<-ggplot(mushrooms_data,aes(x=bruises))+geom_histogram(stat="count",aes(fill=class))+ggtitle(label="
 p6<-ggplot(mushrooms_data,aes(x=odor))+geom_histogram(stat="count",aes(fill=class))+ggtitle(label="odo
 p7<-ggplot(mushrooms_data,aes(x=gill.attachment))+geom_histogram(stat="count",aes(fill=class))+ggtitle
 p8<-ggplot(mushrooms_data,aes(x=gill.spacing))+geom_histogram(stat="count",aes(fill=class))+ggtitle(la
 p9<-ggplot(mushrooms_data,aes(x=gill.size))+geom_histogram(stat="count",aes(fill=class))+ggtitle(label=
 p10<-ggplot(mushrooms_data,aes(x=gill.color))+geom_histogram(stat="count",aes(fill=class))+ggtitle(lab
```

```
p11<-ggplot(mushrooms_data,aes(x=stalk.shape))+geom_histogram(stat="count",aes(fill=class))+ggtitle(la
p12<-ggplot(mushrooms_data,aes(x=stalk.root))+geom_histogram(stat="count",aes(fill=class))+ggtitle(labe
p13<-ggplot(mushrooms_data,aes(x=stalk.surface.above.ring))+geom_histogram(stat="count",aes(fill=class
p14<-ggplot(mushrooms_data,aes(x=stalk.surface.below.ring))+geom_histogram(stat="count",aes(fill=class
p15<-ggplot(mushrooms_data,aes(x=stalk.color.above.ring))+geom_histogram(stat="count",aes(fill=class))+
p16<-ggplot(mushrooms_data,aes(x=stalk.color.below.ring))+geom_histogram(stat="count",aes(fill=class))+
p17<-ggplot(mushrooms_data,aes(x=veil.type))+geom_histogram(stat="count",aes(fill=class))+ggtitle(label
p18<-ggplot(mushrooms_data,aes(x=veil.color))+geom_histogram(stat="count",aes(fill=class))+ggtitle(labe
p19<-ggplot(mushrooms_data,aes(x=ring.number))+geom_histogram(stat="count",aes(fill=class))+ggtitle(la
p20<-ggplot(mushrooms_data,aes(x=ring.type))+geom_histogram(stat="count",aes(fill=class))+ggtitle(label
p21<-ggplot(mushrooms_data,aes(x=spore.print.color))+geom_histogram(stat="count",aes(fill=class))+ggti
p22<-ggplot(mushrooms_data,aes(x=population))+geom_histogram(stat="count",aes(fill=class))+ggtitle(labe
p23<-ggplot(mushrooms_data,aes(x=habitat))+geom_histogram(stat="count",aes(fill=class))+ggtitle(label=
grid.arrange(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16,p17,p18,p19,p20,p21,p22,p23,ncol=2
```


Poisonous or Edible
class
count
cap shape
cap.shape
cap surface
cap.surface
cap.color
bruises
odor
gill.attachment
gill.spacing
gill.size
gill.color
stalk.shape
stalk.root
stalk.surface.above.ring
stalk.surface.below.ring
stalk.color.above.ring
stalk.color.below.ring
veil.type
veil.color
ring.number
ring.type
spore.print.color
population
habitat

We can make the hunches based on the exploratory analysis , but will confirm the huncheas based on the model that we select for machine learning.

Machine Learning

We will follow following steps to decide about the classification model:

1. split the data in train set and test set
2. train the model on the train set
3. check the efficiency of the model on the train set
4. predict the classification of the test set data
5. check the efficiency of the model on the test set

We will iterate these steps for different models and then compare the efficiencies of different models to choose the best model.

Defining split factor

First of all we will define a splitting factor which will be used to split data between train and test set . As it is better to train the model on bigger data set and test on small dataset, we will use a variable to accomodate that thought. Thought behind defining the split factor is to check the effect of the size of training set on the efficiency of the model.

```
mushroom_split_factor<-0.8
```

We will now define the train and test sets:

```
set.seed(1)
mushrooms_split_index<-createDataPartition(mushrooms_data$class,p = mushroom_split_factor,list = FALSE)
mushrooms_trainset<-mushrooms_data[mushrooms_split_index,]
mushrooms_testset<-mushrooms_data[-mushrooms_split_index,]
```

We will now check the dimensions of mashromm dataset, mushroom_trainset and mushroom_testset to make sure that split is fine.

```
dim(mushrooms_data)
```

```
## [1] 8124   23
```

```
dim(mushrooms_testset)
```

```
## [1] 1624   23
```

```
dim(mushrooms_trainset)
```

```
## [1] 6500   23
```

As this is a classification problem. I intend to use rpart,Classification decision trees, bagging , Random Forest and boosting models and then compare the results.We will load the requisite packages here:

```
library(rpart)
library(rpart.plot)
library(caret)
```

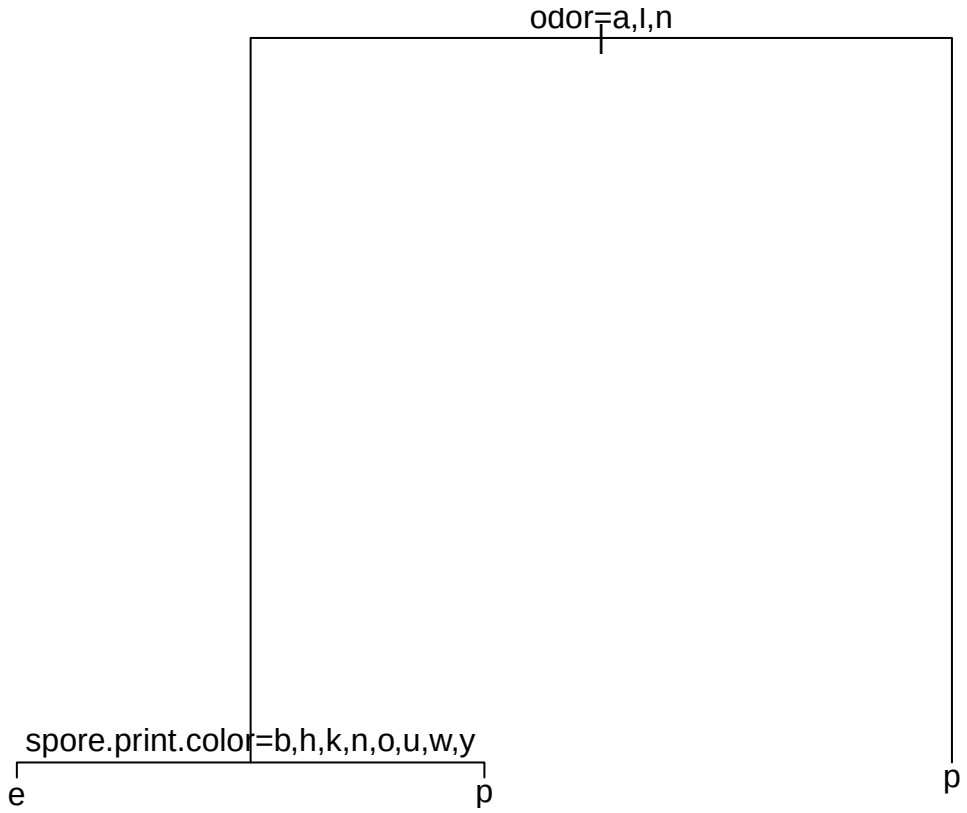
1. Model I : rpart

We will use the same trainset and testset defined earlier for different models . we will train the model on trainset,plot the model, predict for trainset ,calculate the efficiency of model on trainset ,predict for testset , calculate the efficiency for testset and then compare the change in efficiency from train to testset , which will give us an idea about underfitting or overfitting .

```
mushrooms_mdl_rpart<-rpart(class~.,mushrooms_trainset,method = "class")
```

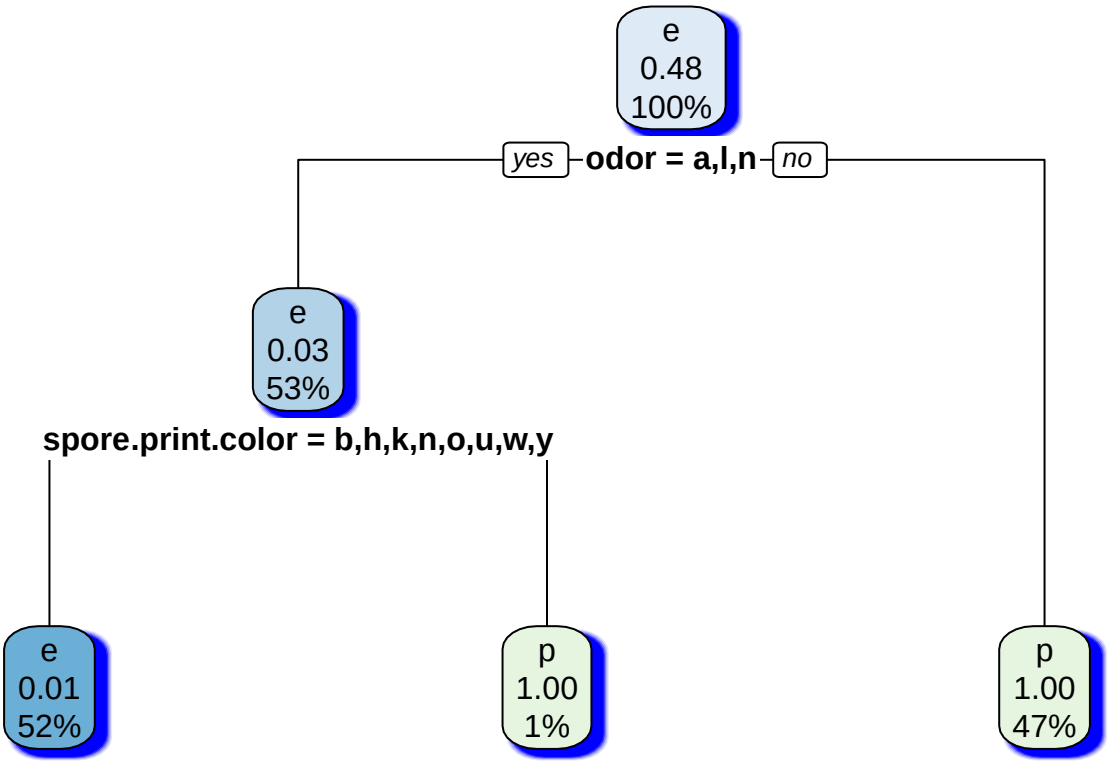
We will plot this model to get an insight now:

```
plot(mushrooms_mdl_rpart)
text(mushrooms_mdl_rpart,pretty = 0)
```



Let's look into a little better version of it :

```
rpart.plot(mushrooms_mdl_rpart,shadow.col = "blue")
```



Predictions for trainset :

```
mushroom_pred_rpart_train<-predict(mushrooms_mdl_rpart,mushrooms_trainset,type = "class")
```

let's look at the consolidated predictions:

```
table(mushroom_pred_rpart_train)
```

```
## mushroom_pred_rpart_train
##    e    p
## 3407 3093
```

To check for the accuracy for trainset :

```
confusionMatrix(mushroom_pred_rpart_train,mushrooms_trainset$class)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    e    p
##          e 3367   40
##          p    0 3093
##
##                Accuracy : 0.9938
##                  95% CI : (0.9916, 0.9956)
##     No Information Rate : 0.518
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9877
```

```
##  Mcnemar's Test P-Value : 6.984e-10
##
##             Sensitivity : 1.0000
##             Specificity : 0.9872
##          Pos Pred Value : 0.9883
##          Neg Pred Value : 1.0000
##              Prevalence : 0.5180
##          Detection Rate : 0.5180
##    Detection Prevalence : 0.5242
##       Balanced Accuracy : 0.9936
##
##        'Positive' Class : e
##
```

Let's look at the predictions on the test set and check the accuracy there :

```
mushroom_pred_rpart_test<-predict(mushrooms_mdl_rpart,mushrooms_testset,type="class")
table(mushroom_pred_rpart_test)
```

```
## mushroom_pred_rpart_test
##   e   p
## 849 775
```

```
confusionMatrix(mushroom_pred_rpart_test,mushrooms_testset$class)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   e   p
##          e 841   8
##          p   0 775
##
##                Accuracy : 0.9951
##                  95% CI : (0.9903, 0.9979)
##     No Information Rate : 0.5179
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.9901
##  Mcnemar's Test P-Value : 0.01333
##
##             Sensitivity : 1.0000
##             Specificity : 0.9898
##          Pos Pred Value : 0.9906
##          Neg Pred Value : 1.0000
##              Prevalence : 0.5179
##          Detection Rate : 0.5179
##    Detection Prevalence : 0.5228
##       Balanced Accuracy : 0.9949
##
##        'Positive' Class : e
##
```

As we see that the accuracy has increased from 99.38% to 99.51% from trainset to testset, which means our model has performed better for unseen data , but still the acceptance of the model depends upon what is the threshold above which, you will accept.

---

2. Model II : Decision Trees and Pruning :

```
library(tree)
```

Model:

```
mushroom_mdl_tree<-tree(class~.,mushrooms_testset)
```

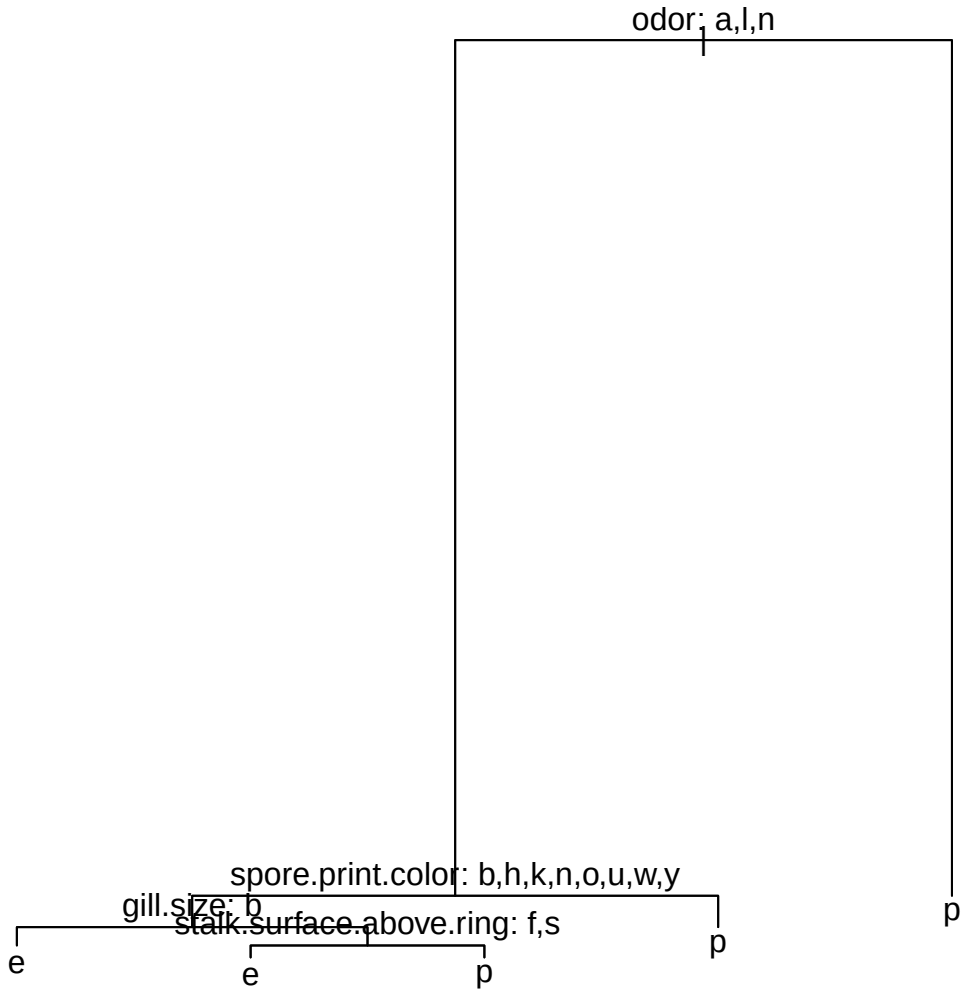
summary of the model :

```
summary(mushroom_mdl_tree)
```

```
##
## Classification tree:
## tree(formula = class ~ ., data = mushrooms_testset)
## Variables actually used in tree construction:
## [1] "odor"                      "spore.print.color"
## [3] "gill.size"                 "stalk.surface.above.ring"
## Number of terminal nodes:  5
## Residual mean deviance:  0.01037 = 16.79 / 1619
## Misclassification error rate: 0.001232 = 2 / 1624
```

Plotting the decision Tree:

```
plot(mushroom_mdl_tree)
text(mushroom_mdl_tree,pretty=0)
```



A look at the tree in text :

```
mushroom_mdl_tree
```

```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
##  1) root 1624 2249.00 e ( 0.517857 0.482143 )
##    2) odor: a,l,n 858  167.00 e ( 0.980186 0.019814 )
##      4) spore.print.color: b,h,k,n,o,u,w,y 849   90.56 e ( 0.990577 0.009423 )
##        8) gill.size: b 793    0.00 e ( 1.000000 0.000000 ) *
##        9) gill.size: n 56   45.93 e ( 0.857143 0.142857 )
```

```
##         18) stalk.surface.above.ring: f,s 50   16.79 e ( 0.960000 0.040000 ) *
##         19) stalk.surface.above.ring: k 6    0.00 p ( 0.000000 1.000000 ) *
##      5) spore.print.color: r 9    0.00 p ( 0.000000 1.000000 ) *
##    3) odor: c,f,m,p,s,y 766    0.00 p ( 0.000000 1.000000 ) *
```

Prediction for training set and the evaluation of efficiency of model on training set :

```
mushroom_pred_tree_train<-predict(mushroom_mdl_tree,mushrooms_trainset,type="class")
mushroom_tree_train_perf<-table(mushroom_pred_tree_train,mushrooms_trainset$class)
mushroom_tree_train_perf
```

```
##
## mushroom_pred_tree_train    e    p
##                        e 3367   14
##                        p    0 3119
```

```
sum(diag(mushroom_tree_train_perf))/sum(mushroom_tree_train_perf)
```

```
## [1] 0.9978462
```

Prediction for test set and the evaluation of efficiency of model on test set :

```
mushroom_pred_tree_test<-predict(mushroom_mdl_tree,mushrooms_testset,type="class")
mushroom_tree_test_perf<-table(mushroom_pred_tree_test,mushrooms_testset$class)
mushroom_tree_test_perf
```

```
##
## mushroom_pred_tree_test   e   p
##                       e 841   2
##                       p   0 781
```

```
sum(diag(mushroom_tree_test_perf))/sum(mushroom_tree_test_perf)
```

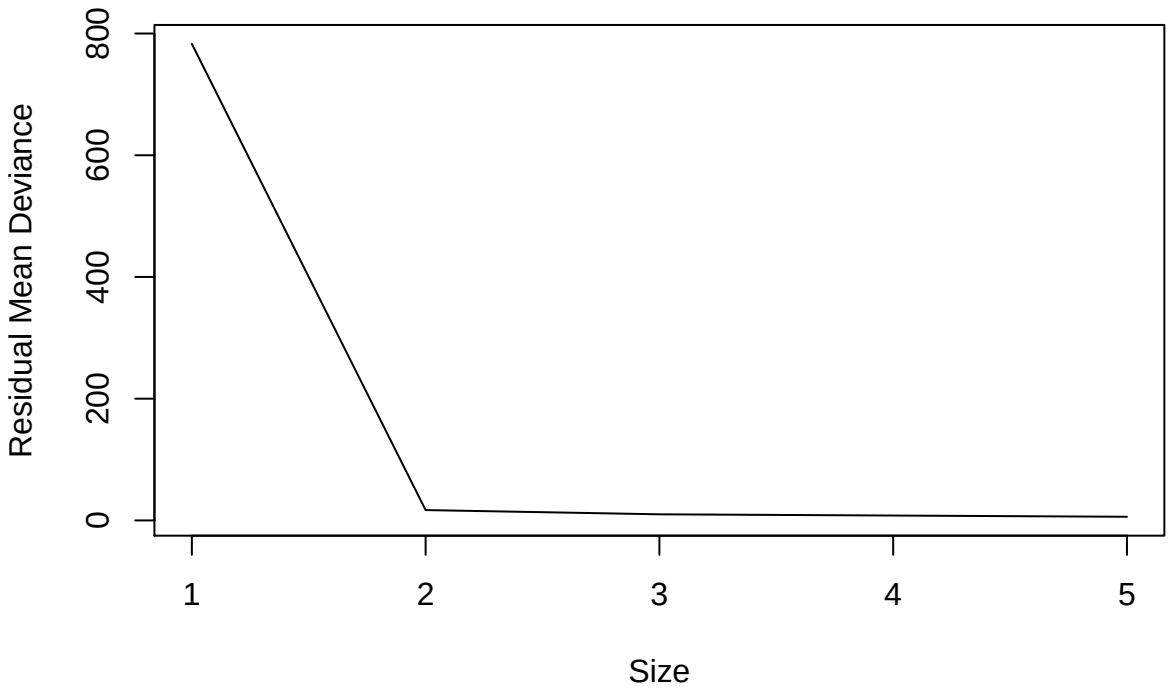
```
## [1] 0.9987685
```

In this model also , model performed better with test data than the training data.To find the optimal level of tree complexity, we can use cost complexity pruning in order to select sequence of trees. We do this by using cross validation. It will help us identify the size of tree that will have minimum residual mean davience.

```
set.seed(1)
mushroom_mdl_tree_cv<-cv.tree(mushroom_mdl_tree,FUN = prune.misclass)
mushroom_mdl_tree_cv
```

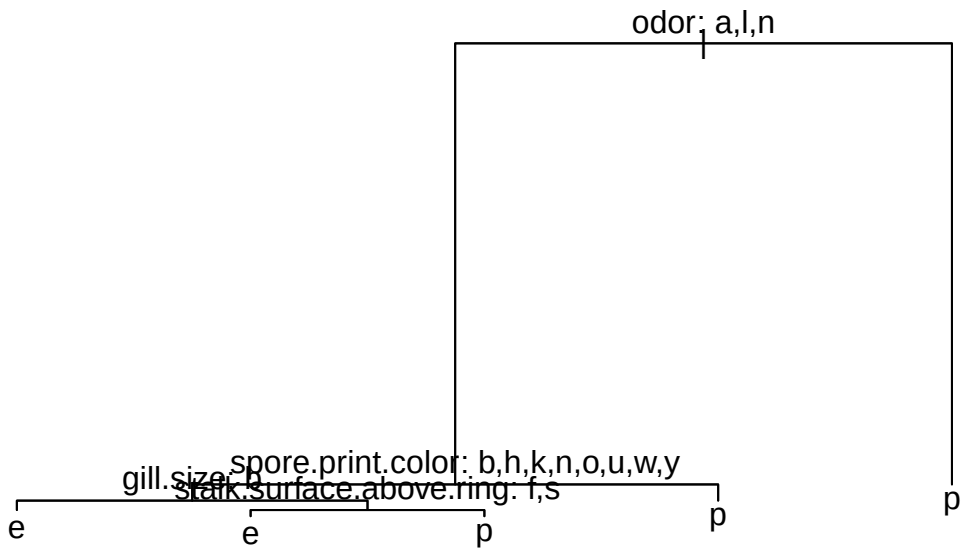
```
## $size
## [1] 5 3 2 1
##
## $dev
## [1]   6  10  17 783
##
## $k
## [1] -Inf    3    9  766
##
## $method
## [1] "misclass"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

```
plot(mushroom_mdl_tree_cv$size,mushroom_mdl_tree_cv$dev,type = "l",xlab = "Size",ylab="Residual Mean Dev
```



we can create a pruned tree for the optimum size 5 as:

```
mushroom_mdl_tree_prune<-prune.misclass(mushroom_mdl_tree,best=5)
plot(mushroom_mdl_tree_prune)
text(mushroom_mdl_tree_prune,pretty=0)
```




```
mushroom_pred_tree_train_prune<-predict(mushroom_mdl_tree_prune,mushrooms_trainset,type="class")
mushroom_pred_tree_test_prune<- predict(mushroom_mdl_tree_prune,mushrooms_testset,type="class")
mush_prn_train_perftab<-table(mushroom_pred_tree_train_prune,mushrooms_trainset$class)
mush_prn_test_perftab<-table(mushroom_pred_tree_test_prune,mushrooms_testset$class)
```

Performance of the pruned tree on trainset :

```
sum(diag(mush_prn_train_perftab))/sum(mush_prn_train_perftab)
```

```
## [1] 0.9978462
```

Performance of the pruned tree on testset :

```
sum(diag(mush_prn_test_perftab))/sum(mush_prn_test_perftab)
```

```
## [1] 0.9987685
```

In fact the tree that we created before pruning was optimum already as it had the 5 terminal nodes as were concluded from cross validation.

---

3. Model III : Bagging

Next Model that we will consider .Here we would try to create trees taking all variables into account while creating multiple trees and then using their average as the final result.first we will load the requisite package :

we will now create the model bagging the trees taking into account all the variables i.e. all the predictors should be considered for each split of the tree(minus the dependent variable):

```
set.seed(1)
mushroom_mdl_bagging<-randomForest(class~.,data=mushrooms_trainset,mtry=22,importance=TRUE)
```

Let's take a look at the bagged tree model:

```
mushroom_mdl_bagging
```

```
##
## Call:
##  randomForest(formula = class ~ ., data = mushrooms_trainset,      mtry = 22, importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 22
##
##         OOB estimate of  error rate: 0.02%
## Confusion matrix:
##      e    p  class.error
## e 3367    0 0.0000000000
## p    1 3132 0.0003191829
```

We would do the predictions for train and test dataset now and check the performance accuracy of the model.We could have used MSE , if the data would have been numeric to test the accuracy of model , but in this case we will use the confusionMatrix to check the efficiency of the model:

```
mushroom_pred_bag_train<-predict(mushroom_mdl_bagging,mushrooms_trainset)
mushroom_pred_bag_train_tbl<-table(mushroom_pred_bag_train,mushrooms_trainset$class)
mushroom_pred_bag_train_tbl
```

```
##
## mushroom_pred_bag_train    e    p
##                       e 3367    0
##                       p    0 3133
```

so the accuracy of the model for the training set is :

```
(sum(diag(mushroom_pred_bag_train_tbl))/sum(mushroom_pred_bag_train_tbl))*100
```

```
## [1] 100
```

As the model on the trainset may be overfitted to give 100% accuract, let's try this on testset:

```
mushroom_pred_bag_test<-predict(mushroom_mdl_bagging,mushrooms_testset)
mushroom_pred_bag_test_tbl<-table(mushroom_pred_bag_test,mushrooms_testset$class)
mushroom_pred_bag_test_tbl
```

```
##
## mushroom_pred_bag_test   e   p
##                       e 841   0
##                       p   0 783
```

so the accuracy of the model for the test set is :

```
(sum(diag(mushroom_pred_bag_test_tbl))/sum(mushroom_pred_bag_test_tbl))*100
```

```
## [1] 100
```

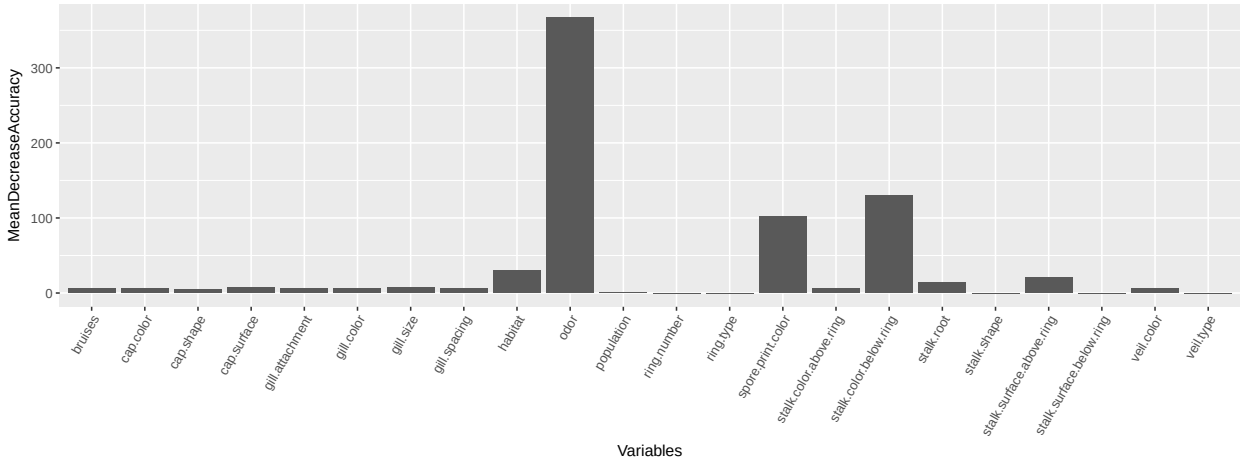
so , we can clearly see that bagging has improved the accuracy of the model.

Let us check at the importance of these variables in this model

```
mushroom_imp_bagging<-importance(mushroom_mdl_bagging)
mushroom_imp_bagging
```

```
##                                e           p MeanDecreaseAccuracy
## cap.shape                4.896537  -0.5887669             4.909002
## cap.surface              6.664476   7.9949604             6.901242
## cap.color                5.906325   3.1514921             5.904314
## bruises                  6.468706   3.6529900             6.475842
## odor                   817.423374 164.0282639           366.850539
## gill.attachment          5.919623   0.0000000             5.917669
## gill.spacing             5.578414   3.7414657             5.636164
## gill.size                7.171052   4.9527073             7.209767
## gill.color               5.907561   2.0077569             5.908320
## stalk.shape              0.000000   0.0000000             0.000000
## stalk.root              11.655025  18.9036291            13.857407
## stalk.surface.above.ring 19.566633 19.5185920            21.089021
## stalk.surface.below.ring  0.000000  0.0000000             0.000000
## stalk.color.above.ring    6.106820  1.4169494             6.105548
## stalk.color.below.ring  118.554154 77.5745890           129.701302
## veil.type                0.000000   0.0000000             0.000000
## veil.color               5.521455   0.0000000             5.519290
## ring.number              0.000000   0.0000000             0.000000
## ring.type                0.000000   0.0000000             0.000000
## spore.print.color       51.222850 140.7635297           101.475489
## population               1.001002   1.0010015             1.001002
## habitat                 29.438197  19.0325768            29.593294
##                         MeanDecreaseGini
## cap.shape                   2.677097e+00
## cap.surface                 5.811078e+00
## cap.color                   3.103987e-01
## bruises                     3.691480e-01
## odor                        3.044693e+03
## gill.attachment             3.061402e-01
## gill.spacing                2.600468e-01
## gill.size                   3.597446e-01
## gill.color                  2.943350e-01
## stalk.shape                 0.000000e+00
## stalk.root                  1.158332e+01
## stalk.surface.above.ring    1.085451e+01
## stalk.surface.below.ring    0.000000e+00
## stalk.color.above.ring      2.758827e-01
## stalk.color.below.ring      4.505889e+01
## veil.type                   0.000000e+00
## veil.color                  2.593538e-01
## ring.number                 0.000000e+00
## ring.type                   0.000000e+00
## spore.print.color           1.220579e+02
## population                  1.566138e-02
## habitat                     1.116141e-01
```

```
mushroom_imp_baggingdf<-as.data.frame(unlist(mushroom_imp_bagging))
ggplot(mushroom_imp_baggingdf,aes(x=row.names(mushroom_imp_baggingdf),y=MeanDecreaseAccuracy))+geom_bar
```



We can clearly see that the odor,stalk.colorbelow.ring and sport.printcolor are the top 3 variables in the bagged model.

---

4. Model IV : randomForest

This model allows random number of variables to be considered at each split unlike the bagging. By default in classification , number of variables considered are sqrt(total no. of variables) i.e for us , it is roundup(sqrt(23))=5

```
mushroom_mdl_ranforest<-randomForest(class~.,data=mushrooms_trainset,mtry=5,importance=TRUE,ntree=500)

mushroom_mdl_ranforest
```

```
##
## Call:
##  randomForest(formula = class ~ ., data = mushrooms_trainset,      mtry = 5, importance = TRUE, ntree
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 5
##
##         OOB estimate of  error rate: 0%
## Confusion matrix:
##      e    p class.error
## e 3367    0           0
## p    0 3133           0
```

let's check the predictions and accuracy on testset:

```
mushroom_pred_ranforest_train<-predict(mushroom_mdl_ranforest,mushrooms_trainset)
mushroom_pred_ranforest_traintbl<-table(mushroom_pred_ranforest_train,mushrooms_trainset$class)
```

Accuracy of the model is :

```
(sum(diag(mushroom_pred_ranforest_traintbl))/sum(mushroom_pred_ranforest_traintbl))*100
```

```
## [1] 100
```

Let's do the predictions for the testset and find the accuracy:

```
mushroom_pred_ranforest_test<-predict(mushroom_mdl_ranforest,mushrooms_testset)
mushroom_pred_ranforest_testtbl<-table(mushroom_pred_ranforest_test,mushrooms_testset$class)
mushroom_pred_ranforest_testtbl
```

```
##
## mushroom_pred_ranforest_test   e   p
##                            e 841   0
##                            p   0 783
```

Accuracy of the testset is :

```
(sum(diag(mushroom_pred_ranforest_testtbl))/sum(mushroom_pred_ranforest_testtbl))*100
```

```
## [1] 100
```

Let's check this model give what importance to which variable:

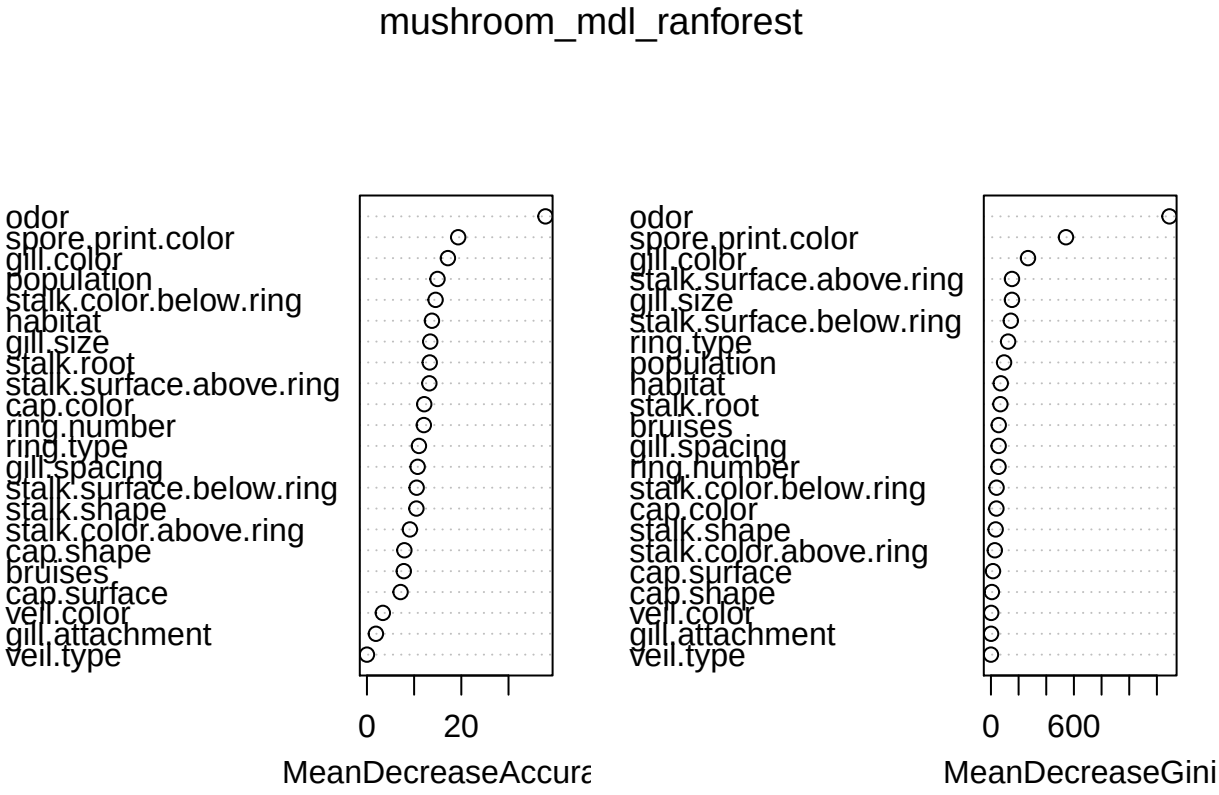
```
importance(mushroom_mdl_ranforest)
```

```
##                                e         p MeanDecreaseAccuracy
## cap.shape                 5.080529  6.619204             7.929082
## cap.surface               5.642044  6.372138             7.126618
## cap.color                12.275185  8.072518            12.139288
## bruises                   6.828824  6.736306             7.809696
## odor                     33.943566 30.613914            37.829428
## gill.attachment           1.951394  1.430975             1.900784
## gill.spacing              8.597167  9.644807            10.742491
## gill.size                14.472528 10.666829            13.408822
## gill.color               16.767960 10.086676            17.156519
## stalk.shape               7.604760 10.540555            10.481279
## stalk.root               12.480795  9.880242            13.304965
## stalk.surface.above.ring 12.699791  8.347161            13.223349
## stalk.surface.below.ring  9.468360  8.260831            10.531920
## stalk.color.above.ring    9.277861  6.081152             9.102598
## stalk.color.below.ring   14.895532  6.342392            14.545454
## veil.type                 0.000000  0.000000             0.000000
## veil.color                2.785361  3.733575             3.371209
## ring.number              11.137914 10.555566            12.054208
## ring.type                 8.686938  9.563414            11.015712
## spore.print.color        18.569862 15.223876            19.336402
## population               12.455042 12.080278            14.955008
## habitat                  13.178816  9.052500            13.770274
##                          MeanDecreaseGini
## cap.shape                       6.8611205
## cap.surface                    14.7355960
## cap.color                      39.5676751
## bruises                        56.3030363
## odor                         1290.9265078
## gill.attachment                 0.9371213
## gill.spacing                   55.8342115
## gill.size                     151.9342793
## gill.color                    267.9314166
## stalk.shape                    34.0853235
## stalk.root                     68.3581452
## stalk.surface.above.ring      152.1710823
## stalk.surface.below.ring      143.6505514
```

```
## stalk.color.above.ring         27.9892543
## stalk.color.below.ring         40.6361788
## veil.type                       0.0000000
## veil.color                      2.1918414
## ring.number                    54.9147640
## ring.type                     124.2530803
## spore.print.color             542.6689806
## population                     94.9234245
## habitat                        70.9116371
```

Graphically we can see the importance as:

```
varImpPlot(mushroom_mdl_ranforest)
```



We see that odor, spore.print.color and gill.color are top three variables to affect the accuracy of this model.

---

5. Model V : Boosting

In boosting, trees are grown sequentially. each tree is grown using the information from previously grown trees.
We first load the requisite package-gbm:

```
library(gbm)
```

```
## Loading required package: survival
```

```
##
## Attaching package: 'survival'
```

```
## The following object is masked from 'package:caret':
##
##     cluster
```

```
## Loading required package: splines
```

```
## Loading required package: parallel
```

```
## Loaded gbm 2.1.3
```

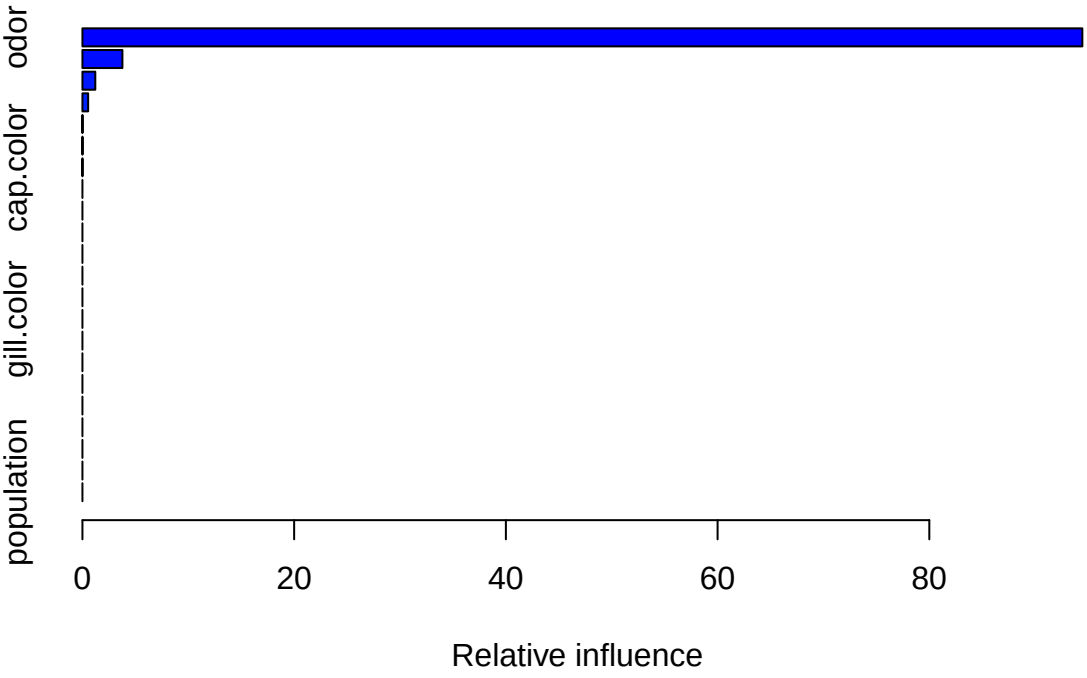
As this is a classification problem, we will use distribution="bernoulli" as one of the options of gbm() to cretae model. we have kept no. of trees as 500 just to keep it same with the bagging model to facilitate easy comparison. In this model the expectation from dependent variable is be in the form of 0 or 1 , so we change the data for class=p as 0 and class=e as 1. we will call these newsets as testset1 and trainset1

```
set.seed(1)
mushrooms_trainset1<-mushrooms_trainset
mushrooms_testset1<-mushrooms_testset
mushrooms_trainset1$class<-ifelse(mushrooms_trainset1$class=="e",1,0)
mushrooms_testset1$class<-ifelse(mushrooms_testset1$class=="e",1,0)
mushroom_mdl_boost<-gbm(class~.-class,mushrooms_trainset1,distribution="bernoulli",n.trees = 500,intera
```

```
## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 16: veil.type has no variation.
```

here is the summary of Model:

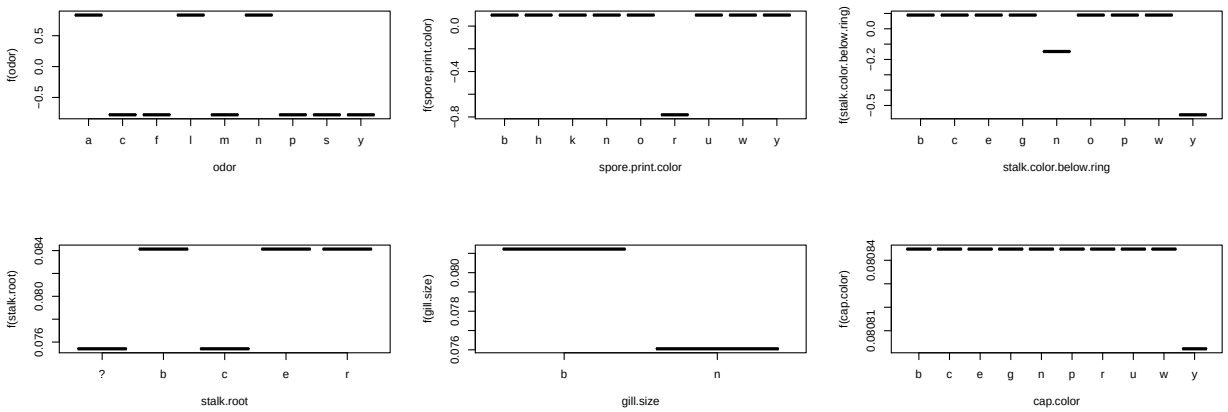
```
summary(mushroom_mdl_boost)
```




```
##                                           var      rel.inf
## odor                                     odor 9.445683e+01
## spore.print.color           spore.print.color 3.775892e+00
## stalk.color.below.ring stalk.color.below.ring 1.214827e+00
## stalk.root                         stalk.root 5.396358e-01
## gill.size                           gill.size 6.078012e-03
## stalk.surface.above.ring stalk.surface.above.ring 4.532729e-03
## cap.color                           cap.color 2.173386e-03
## habitat                               habitat 3.154492e-05
## cap.shape                           cap.shape 0.000000e+00
## cap.surface                       cap.surface 0.000000e+00
## bruises                             bruises 0.000000e+00
## gill.attachment               gill.attachment 0.000000e+00
## gill.spacing                     gill.spacing 0.000000e+00
## gill.color                         gill.color 0.000000e+00
## stalk.shape                       stalk.shape 0.000000e+00
## stalk.surface.below.ring stalk.surface.below.ring 0.000000e+00
## stalk.color.above.ring stalk.color.above.ring 0.000000e+00
## veil.type                           veil.type 0.000000e+00
## veil.color                         veil.color 0.000000e+00
## ring.number                       ring.number 0.000000e+00
## ring.type                           ring.type 0.000000e+00
## population                         population 0.000000e+00
```

let's check the partial dependence plots of top 6 variables:

```
par(mfrow=c(2,3))
plot(mushroom_mdl_boost,i="odor")
plot(mushroom_mdl_boost,i="spore.print.color")
plot(mushroom_mdl_boost,i="stalk.color.below.ring")
plot(mushroom_mdl_boost,i="stalk.root")
plot(mushroom_mdl_boost,i="gill.size")
plot(mushroom_mdl_boost,i="cap.color")
```



we will now use this model to do the prediction for testset:

```
mushroom_pred_boost_test<-predict(mushroom_mdl_boost,mushrooms_testset1,n.trees=500)
head(mushroom_pred_boost_test)
```

```
## [1] -0.7808930  0.8812237  0.8812237  0.8812237  0.8812237  0.8712815
```

we can use the confusionmatrix for accuracy after using ifelse as the predictions don't come in 0 or 1 format.

Summary of Models used:

Based on the analysis and checking the accuracy of the above models with test data, we will go for Bagging or Random Tree as they have already reached perfect accuracy. Increasing the complexity further with boosting and decreasing the explainabilty would not be appropriate.