Person 2 – Group 1

Waheed Anwar

2023-03-25

Link to data set: Mushrooms.csv https://archive.ics.uci.edu/ml/datasets/Mushroom

The following is a classification data set consisting of qualitative data in relation to edible and poisonous 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 ...
##  $ stalk.color.below.ring  : Factor w/ 9 levels "b","c","e","g",..: 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 ...
##  $ veil.color              : Factor w/ 4 levels "n","o","w","y": 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 ...
##  $ 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 ...
```

Split data into training and testing sets:

```
set.seed(1)

#use 70% of data set as training set and 30% as test set
sample <- sample(c(TRUE, FALSE), nrow(df), replace=TRUE, prob=c(0.7,0.3))
train  <- df[sample, ]
test   <- df[!sample, ]
```

Exploration

First off we see from the data frame that there doesn't seem to be an imbalance of data nor any missing data.

Statistical analysis: Finding the 'mode' of the mushroom class category

When it comes to a classification data set we observe qualitative data that does not include many integers nor measurements by which we can mathematically manipulate. For example, let's take a look at the following example:

```
# Which mushroom class is more common in the data: edible ('e') or poisonous ('p')?
find_most_common <- function(i) {
   variable <- unique(i)
   variable[which.max(tabulate(match(i, variable)))]
}
most_common_value <- find_most_common(train$class)

print(most_common_value)
```

```
## [1] e
## Levels: e p
```

It's important to note that this is specifically for the training data set that is separated. We can see that there are more edible mushrooms in the training data set than there are poisonous ones. What about the entire set? Let's see if we expand the scope to the entire set:

```
most_common_value <- find_most_common(df$class)

print(most_common_value)
```

```
## [1] e
## Levels: e p
```

We see that the result stays the same and that there are still more edible mushrooms than poisonous ones. ##

Graphically, here is proof that there is more edible than poisonous.

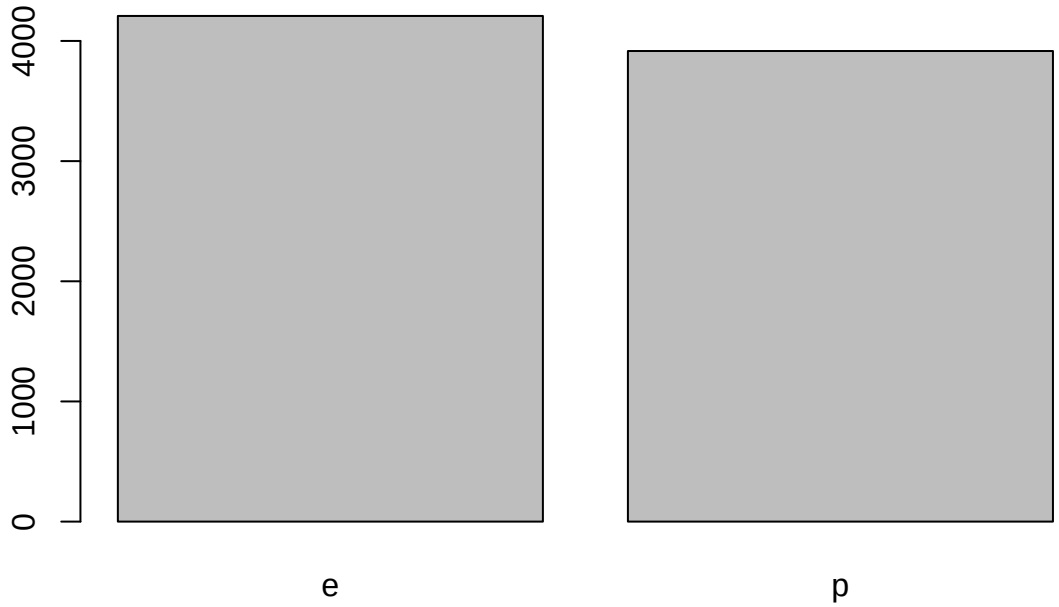
```
plot(df$class, type="h")
```

```
## Warning in plot.window(xlim, ylim, log = log, ...): graphical parameter "type"
## is obsolete
```

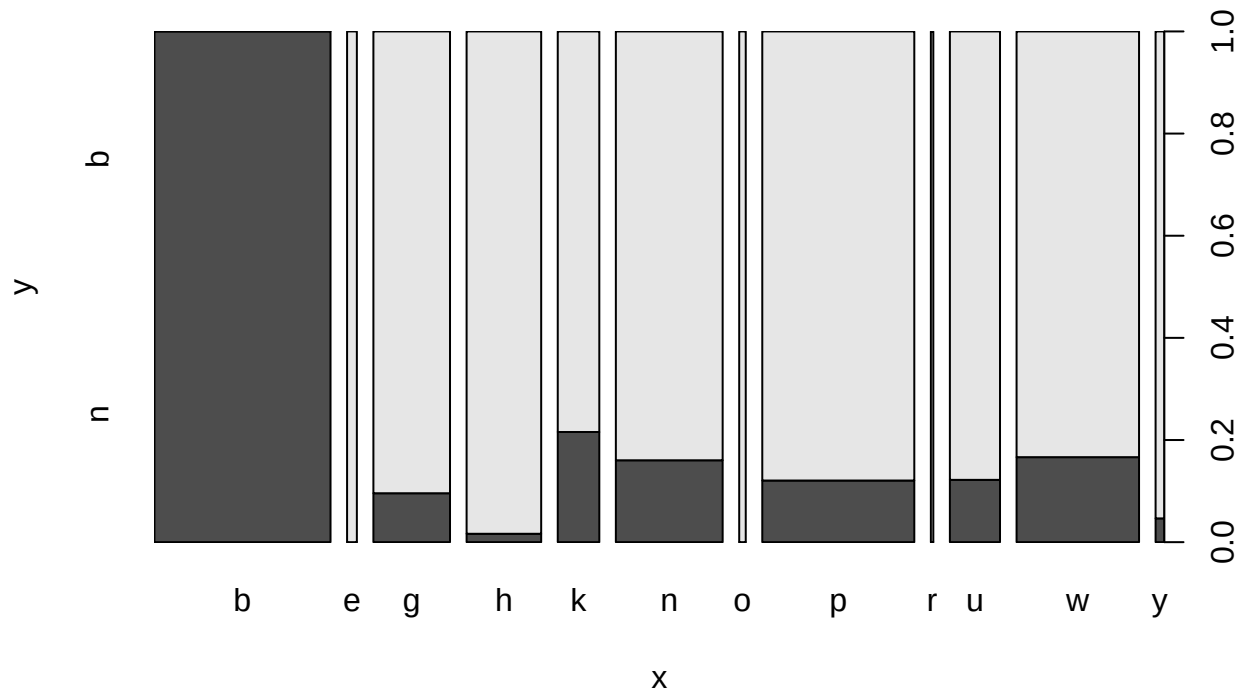
```
## Warning in axis(if (horiz) 2 else 1, at = at.l, labels = names.arg, lty =
## axis.lty, : graphical parameter "type" is obsolete
```

```
## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
## graphical parameter "type" is obsolete
```

```
## Warning in axis(if (horiz) 1 else 2, cex.axis = cex.axis, ...): graphical
## parameter "type" is obsolete
```




```
#exploring the relationship between gill color and gill size
plot(df$gill.color, df$gill.size)
```



Logistic Regression

```
# since all the variables are characters lets choose gill size to predict class
logistic <- glm(class~gill.size, data=train, family="binomial")
summary(logistic)
```

```
##
## Call:
## glm(formula = class ~ gill.size, family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0812  -0.8370  -0.8370   0.4935   1.5615
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.86879    0.03492  -24.88   <2e-16 ***
## gill.sizen   2.91271    0.08338   34.93   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 7827.7  on 5656  degrees of freedom
## Residual deviance: 6005.8  on 5655  degrees of freedom
## AIC: 6009.8
##
## Number of Fisher Scoring iterations: 4
```

```
probs <- predict(logistic, newdata=test)
pred <- ifelse(probs>0.5,1,0)
pred <- as.factor(pred)
levels(pred) <- list("e"="0","p"="1")
acc <- mean(as.integer(pred)==as.integer(test$class))
print(paste("accuracy = ", acc))
```

```
## [1] "accuracy =  0.74908796108634"
```

```
table(pred, test$class)
```

```
##
## pred    e    p
##    e 1145  528
##    p   91  703
```

kNN

We wish to see the similarity in data observations, meaning that for this portion we are looking to read/process data rather than aim to train it.

```
#importing the necessary packages
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

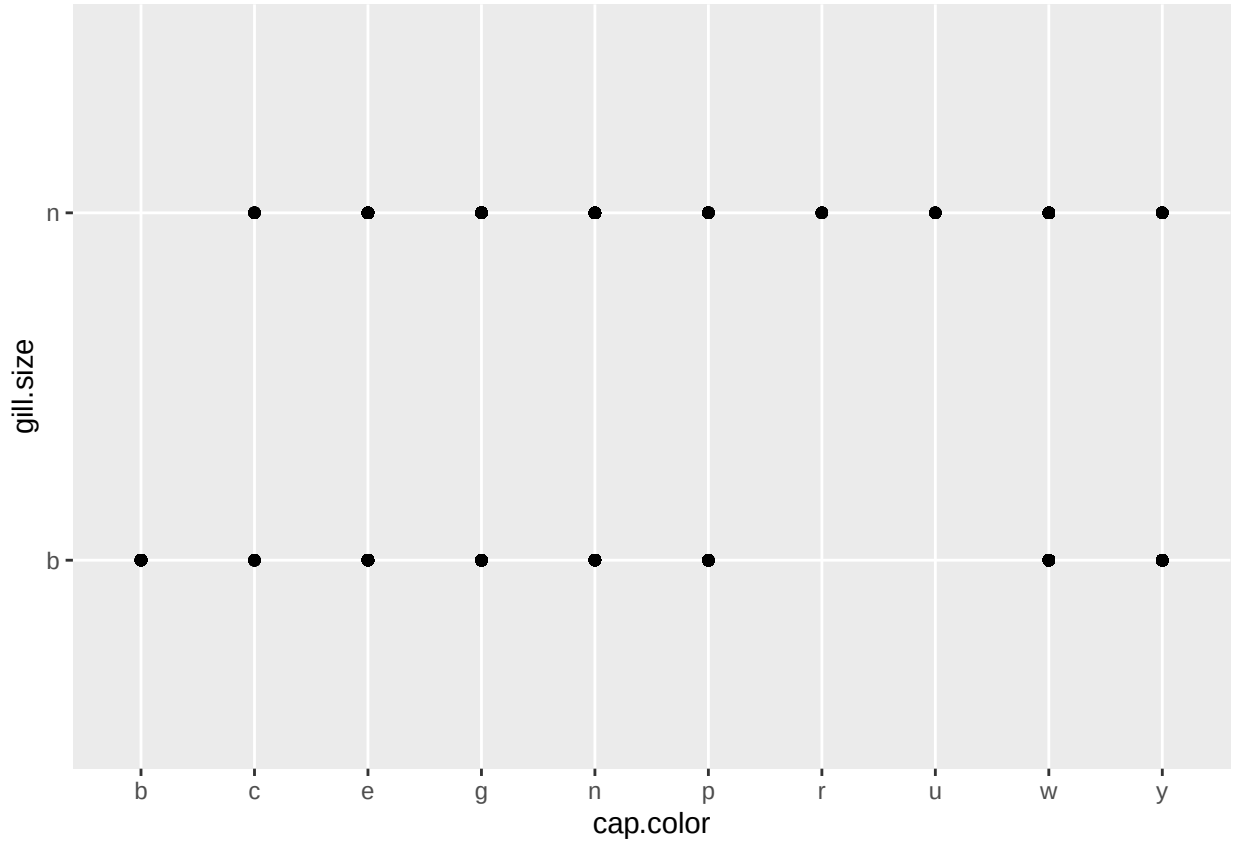
```
library(class)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

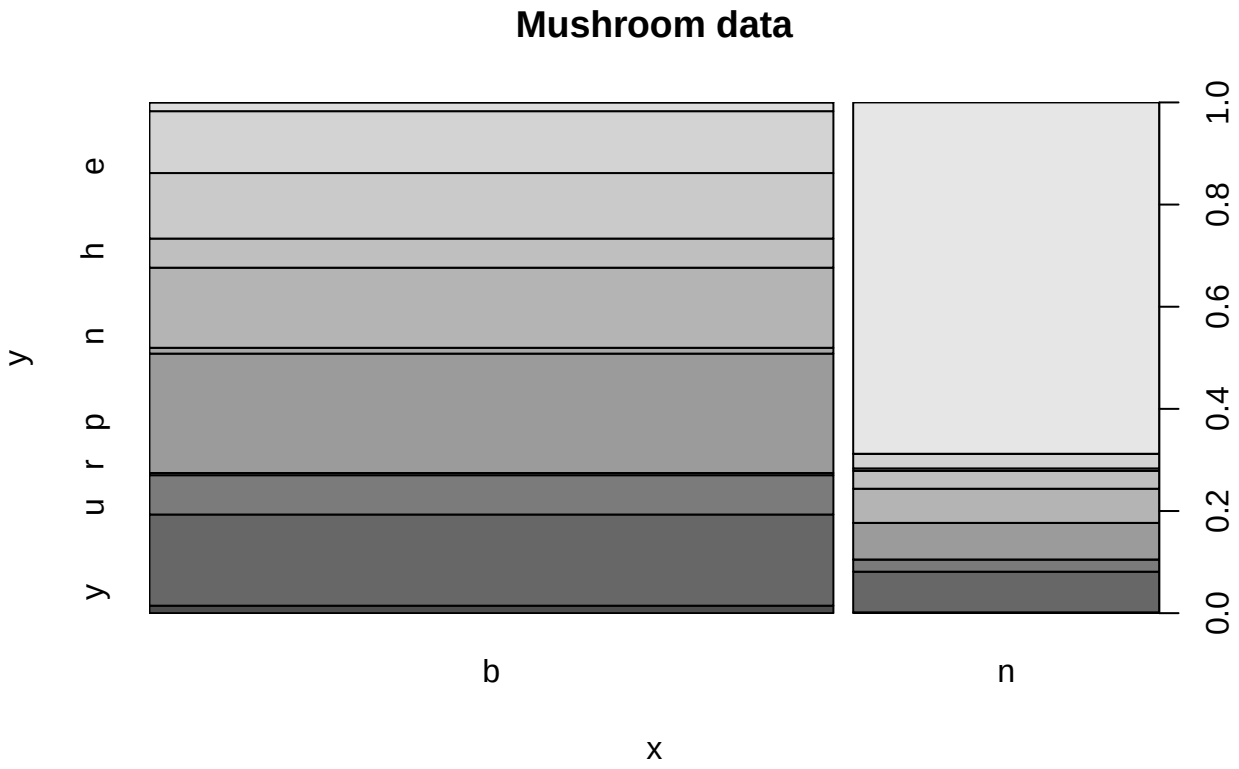
```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
ggplot(data = df, mapping = aes(x = cap.color, y = gill.size)) + geom_point()
```




```
# green, e
# red, p
plot(df$gill.size, df$gill.color, pch=10, bg = c("green","red")[unclass(df$class)], main = "Mushroom da
```




```
mushroom.train  <- df[sample, ]
mushroom.test   <- df[!sample, ]
#mushroom.trainLabels <- as.integer(df$class)
#mushroom_pred <- knn(train=mushroom.train, test=mushroom.test, cl=mushroom.trainLabels, k=3)

## had some trouble with knn
```

Decision Trees

First let's utilize the rpart() package:

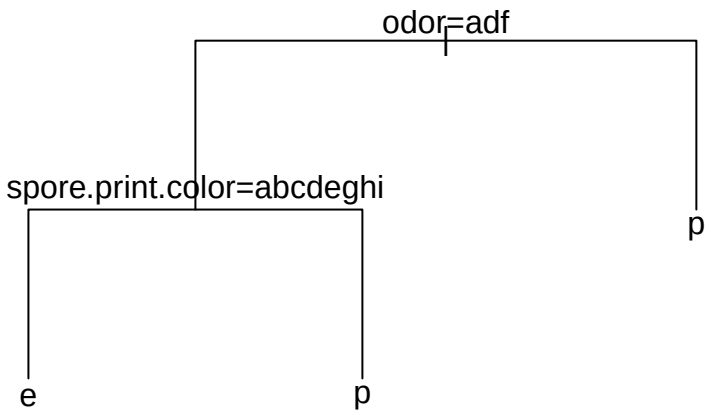
```
library(rpart)
tree_mushroom <- rpart(class~., data=df, method="class")
tree_mushroom
```

```
## n= 8124
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
## 1) root 8124 3916 e (0.51797144 0.48202856)
##   2) odor=a,l,n 4328  120 e (0.97227357 0.02772643)
##     4) spore.print.color=b,h,k,n,o,u,w,y 4256   48 e (0.98872180 0.01127820) *
##     5) spore.print.color=r 72    0 p (0.00000000 1.00000000) *
```

```
##   3) odor=c,f,m,p,s,y 3796    0 p (0.00000000 1.00000000) *
```

The decision tree will look like the following:

```
plot(tree_mushroom, uniform=TRUE, margin=0.2)
text(tree_mushroom)
```



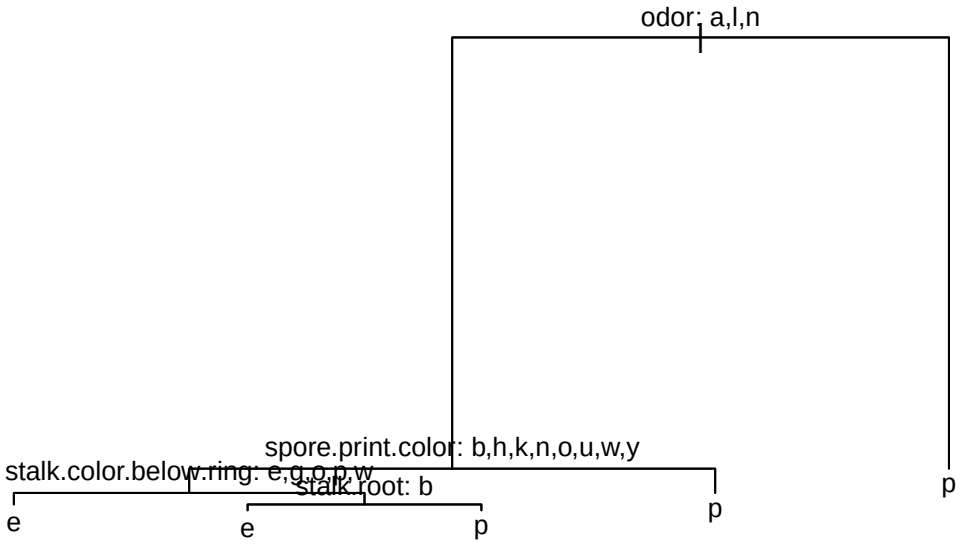
Utilizing the tree() package (which should give more splits due to the different metric):

```
library(tree)
tree_mushroom2 <- tree(class~., data=df)
tree_mushroom2
```

```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
##  1) root 8124 11250.0 e ( 0.517971 0.482029 )
##    2) odor: a,l,n 4328  1097.0 e ( 0.972274 0.027726 )
##      4) spore.print.color: b,h,k,n,o,u,w,y 4256   526.0 e ( 0.988722 0.011278 )
##        8) stalk.color.below.ring: e,g,o,p,w 4152   116.0 e ( 0.998073 0.001927 ) *
##        9) stalk.color.below.ring: n,y 104   138.6 e ( 0.615385 0.384615 )
##         18) stalk.root: b 64     0.0 e ( 1.000000 0.000000 ) *
```

```
##         19) stalk.root: ?,c 40     0.0 p ( 0.000000 1.000000 ) *
##       5) spore.print.color: r 72     0.0 p ( 0.000000 1.000000 ) *
##     3) odor: c,f,m,p,s,y 3796     0.0 p ( 0.000000 1.000000 ) *
```

```
plot(tree_mushroom2)
text(tree_mushroom2, cex = 0.8, pretty=1)
```



Notice that unlike the tree implemented using the rpart() (where the class of poisonous/edible branched from spore.print.color the tree implemented using the tree() package shows that there is an additional branch out after spore.print.color.

This can be explained since I mentioned that there will be more splits so lets take a look at an overview of this tree.

```
summary(tree_mushroom2)
```

```
##
## Classification tree:
## tree(formula = class ~ ., data = df)
```

```
## Variables actually used in tree construction:
## [1] "odor"                   "spore.print.color"      "stalk.color.below.ring"
## [4] "stalk.root"
## Number of terminal nodes:  5
## Residual mean deviance:  0.01429 = 116 / 8119
## Misclassification error rate: 0.0009847 = 8 / 8124
```

Notice the deviance and error rate.

Now what if we utilize the data we separated instead of the entire data frame?

```
tree_mushroom3 <- tree(class~., data=train)
pred <- predict(tree_mushroom3, newdata=test, type="class")
table(pred, test$class)
```

```
##
## pred    e    p
##    e 1236    2
##    p    0 1229
```

```
# the accuracy
mean(pred==test$class)
```

```
## [1] 0.9991893
```

Comparing Results / Analysis

The decision tree had the highest accuracy with an accuracy of almost 100% (exactly 99%), while the logistic regression had an accuracy of around 74% for the parameters given. The reason I think decision trees had the highest accuracy was because I altered my data for logistic regression slightly by giving the training data a higher percentage. Also because of issues with kNN and classification there is a slight advantage for the decision tree. The results actually surprised me because I thought that the logistic regression would have a higher accuracy but this may also be due to utilizing two methods of setting up decision trees.