Patient3

Anuhya B S

2022-06-08

```
#Importing libraries
library('R.matlab')
```

```
## R.matlab v3.6.2 (2018-09-26) successfully loaded. See ?R.matlab for help.
```

```
##
## Attaching package: 'R.matlab'
```

```
## The following objects are masked from 'package:base':
##
##     getOption, isOpen
```

```
library(caTools)
library(e1071)
library(class)
library(tree)
library(randomForest)
```

```
## randomForest 4.7-1.1
```

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

```
#Loading Data
p1 <- readMat("data-science-P3.mat")

info <- as.data.frame(p1[2])
info <- t(info)
info <- as.data.frame(info)
lab.grp <-as.data.frame(matrix(nrow=0,ncol=1))
lab.wrd <-as.data.frame(matrix(nrow=0,ncol=1))
for (i in 1:360){
  lab.grp <-  rbind(lab.grp,info$cond[[i]])
  lab.wrd <-  rbind(lab.wrd,info$word[[i]])
}

p1.data <- p1$data
voxels <-as.data.frame(matrix(nrow=0,ncol=21764))
for (i in 1:360){
  voxels <-  rbind(voxels,p1.data[[i]][[1]])
}
```

```
# Principal Component Analysis for Feature Reduction
pr.out <- prcomp(voxels)
cumsum((pr.out$sdev^2)/sum(pr.out$sdev^2))
```

```
##   [1] 0.3757427 0.4392668 0.4768138 0.5093430 0.5287687 0.5477745 0.5643294
```

```
##   [8] 0.5783444 0.5898335 0.6004230 0.6099440 0.6185135 0.6265621 0.6340233
##  [15] 0.6407812 0.6471868 0.6531277 0.6587143 0.6641753 0.6693652 0.6743700
##  [22] 0.6792443 0.6836777 0.6879053 0.6921097 0.6960308 0.6998422 0.7034581
##  [29] 0.7069645 0.7102128 0.7133764 0.7163604 0.7192686 0.7221335 0.7249169
##  [36] 0.7275472 0.7301507 0.7326295 0.7350073 0.7373544 0.7395926 0.7417911
##  [43] 0.7439470 0.7460368 0.7481077 0.7501247 0.7521187 0.7540778 0.7560033
##  [50] 0.7578860 0.7597500 0.7615435 0.7633327 0.7650660 0.7667728 0.7684572
##  [57] 0.7701115 0.7717360 0.7733182 0.7748959 0.7764537 0.7779938 0.7795102
##  [64] 0.7810149 0.7825040 0.7839726 0.7854231 0.7868516 0.7882727 0.7896898
##  [71] 0.7910699 0.7924390 0.7937885 0.7951167 0.7964307 0.7977389 0.7990301
##  [78] 0.8003114 0.8015779 0.8028337 0.8040792 0.8053206 0.8065531 0.8077718
##  [85] 0.8089731 0.8101687 0.8113575 0.8125381 0.8137183 0.8148812 0.8160370
##  [92] 0.8171853 0.8183238 0.8194495 0.8205638 0.8216720 0.8227761 0.8238767
##  [99] 0.8249662 0.8260481 0.8271262 0.8281992 0.8292681 0.8303265 0.8313742
## [106] 0.8324182 0.8334565 0.8344905 0.8355188 0.8365377 0.8375503 0.8385613
## [113] 0.8395686 0.8405674 0.8415631 0.8425522 0.8435356 0.8445155 0.8454862
## [120] 0.8464504 0.8474109 0.8483693 0.8493223 0.8502695 0.8512134 0.8521527
## [127] 0.8530886 0.8540187 0.8549465 0.8558699 0.8567879 0.8577044 0.8586168
## [134] 0.8595273 0.8604367 0.8613357 0.8622299 0.8631237 0.8640131 0.8649006
## [141] 0.8657848 0.8666656 0.8675430 0.8684140 0.8692824 0.8701483 0.8710106
## [148] 0.8718670 0.8727211 0.8735720 0.8744222 0.8752674 0.8761083 0.8769460
## [155] 0.8777808 0.8786115 0.8794405 0.8802674 0.8810915 0.8819115 0.8827310
## [162] 0.8835446 0.8843546 0.8851635 0.8859713 0.8867756 0.8875752 0.8883706
## [169] 0.8891649 0.8899546 0.8907415 0.8915256 0.8923076 0.8930880 0.8938667
## [176] 0.8946425 0.8954163 0.8961872 0.8969548 0.8977207 0.8984831 0.8992440
## [183] 0.9000043 0.9007622 0.9015183 0.9022719 0.9030231 0.9037702 0.9045148
## [190] 0.9052583 0.9059997 0.9067403 0.9074768 0.9082089 0.9089398 0.9096685
## [197] 0.9103961 0.9111233 0.9118470 0.9125685 0.9132863 0.9140030 0.9147162
## [204] 0.9154269 0.9161357 0.9168424 0.9175474 0.9182501 0.9189500 0.9196498
## [211] 0.9203464 0.9210398 0.9217303 0.9224189 0.9231057 0.9237898 0.9244722
## [218] 0.9251521 0.9258292 0.9265055 0.9271806 0.9278533 0.9285235 0.9291915
## [225] 0.9298577 0.9305231 0.9311859 0.9318461 0.9325044 0.9331614 0.9338154
## [232] 0.9344674 0.9351184 0.9357668 0.9364133 0.9370594 0.9377013 0.9383419
## [239] 0.9389786 0.9396147 0.9402502 0.9408824 0.9415141 0.9421435 0.9427690
## [246] 0.9433940 0.9440160 0.9446356 0.9452546 0.9458717 0.9464847 0.9470965
## [253] 0.9477061 0.9483147 0.9489199 0.9495245 0.9501282 0.9507290 0.9513283
## [260] 0.9519263 0.9525225 0.9531171 0.9537091 0.9542984 0.9548866 0.9554728
## [267] 0.9560572 0.9566397 0.9572210 0.9578017 0.9583787 0.9589550 0.9595280
## [274] 0.9600996 0.9606698 0.9612374 0.9618039 0.9623682 0.9629318 0.9634942
## [281] 0.9640525 0.9646096 0.9651639 0.9657165 0.9662686 0.9668193 0.9673690
## [288] 0.9679151 0.9684592 0.9690023 0.9695433 0.9700818 0.9706175 0.9711519
## [295] 0.9716826 0.9722115 0.9727385 0.9732635 0.9737857 0.9743072 0.9748273
## [302] 0.9753440 0.9758589 0.9763725 0.9768844 0.9773958 0.9779065 0.9784137
## [309] 0.9789188 0.9794230 0.9799258 0.9804251 0.9809212 0.9814160 0.9819092
## [316] 0.9823981 0.9828850 0.9833684 0.9838510 0.9843317 0.9848098 0.9852845
## [323] 0.9857567 0.9862236 0.9866885 0.9871484 0.9876065 0.9880591 0.9885069
## [330] 0.9889494 0.9893856 0.9898208 0.9902457 0.9906688 0.9910878 0.9915013
## [337] 0.9919088 0.9923130 0.9927169 0.9931163 0.9935144 0.9939085 0.9942967
## [344] 0.9946818 0.9950626 0.9954415 0.9958154 0.9961852 0.9965534 0.9969187
## [351] 0.9972789 0.9976345 0.9979883 0.9983332 0.9986774 0.9990175 0.9993545
## [358] 0.9996855 1.0000000 1.0000000
```

```
pcs <- as.data.frame(pr.out$x[,1:300])
pcs$grp <- lab.grp$V1
```

```
#pcs$wrd <- lab.wrd$V1
```

```
# Splitting data into training and test data
set.seed(100)
#sample <- sample(1:nrow(pcs), 300)
pcs.train <- pcs[1:300,]
pcs.test <- pcs[301:360,]
#pcs.train <- subset(pcs, sample == TRUE)
#pcs.test  <- subset(pcs, sample == FALSE)
pcs.train.x <- subset(pcs.train, select = -c(grp))
pcs.train.labs <- pcs.train$grp
pcs.test.x <- subset(pcs.test, select = -c(grp))
pcs.test.labs <- pcs.test$grp
```

```
# Classification Algorithms

# Naive Bayes Classifier

nb.fit <- naiveBayes(grp ~ . , data = pcs.train)
nb.class <- predict(nb.fit,pcs.test.x)
nb.class
```

```
##  [1] manmade   furniture furniture clothing  manmade   clothing  vegetable
##  [8] tool      manmade   manmade   tool      furniture manmade   tool
## [15] furniture manmade   animal    bodypart  insect    vegetable vehicle
## [22] building  buildpart manmade   vegetable insect    bodypart  bodypart
## [29] insect    manmade   animal    insect    buildpart vegetable building
## [36] furniture manmade   bodypart  animal    tool      manmade   manmade
## [43] building  kitchen   building  kitchen   bodypart  tool      clothing
## [50] manmade   clothing  manmade   kitchen   bodypart  bodypart  animal
## [57] manmade   kitchen   insect    manmade
## 12 Levels: animal bodypart building buildpart clothing furniture ... vehicle
```

```
confusion_mat.nb = as.matrix(table(Actual_Values = pcs.test.labs, Predicted_Values = nb.class))
print(confusion_mat.nb)
```

```
##              Predicted_Values
## Actual_Values animal bodypart building buildpart clothing furniture insect
##     animal         0        2        0         0        0         1      0
##     bodypart       0        0        0         0        0         0      0
##     building       0        0        0         0        1         0      0
##     buildpart      2        1        0         0        1         0      1
##     clothing       1        0        0         0        0         1      1
##     furniture      1        0        1         0        0         2      0
##     insect         0        1        0         0        0         0      0
##     kitchen        0        0        0         0        1         1      1
##     manmade        0        2        2         0        1         0      0
##     tool           0        0        1         0        0         0      1
##     vegetable      0        1        0         1        0         0      0
##     vehicle        0        0        0         1        0         0      1
##              Predicted_Values
## Actual_Values kitchen manmade tool vegetable vehicle
##     animal          1       0    1         0       0
##     bodypart        2       2    1         0       0
##     building        0       3    1         0       0
```

```
##     buildpart       0       0    0         0       0
##     clothing        0       2    0         0       0
##     furniture       0       0    0         1       0
##     insect          0       3    0         0       1
##     kitchen         0       1    1         0       0
##     manmade         0       0    0         0       0
##     tool            0       2    1         0       0
##     vegetable       0       1    0         2       0
##     vehicle         1       1    0         1       0
```

```
print(mean(nb.class == pcs.test$grp))
```

```
## [1] 0.08333333
```

```
# KNN

knn.pred <- knn(pcs.train.x, pcs.test.x, pcs.train.labs, k=5)

confusion_mat.knn = as.matrix(table(pcs.test.labs, knn.pred))
print(confusion_mat.knn)
```

```
##              knn.pred
## pcs.test.labs animal bodypart building buildpart clothing furniture insect
##     animal         0        1        0         0        0         0      0
##     bodypart       0        3        0         0        0         0      0
##     building       0        2        0         0        0         0      1
##     buildpart      0        3        0         0        0         2      0
##     clothing       0        0        0         0        0         3      0
##     furniture      0        0        1         0        0         1      0
##     insect         0        0        1         0        1         0      1
##     kitchen        0        0        1         0        0         0      1
##     manmade        2        0        1         0        0         0      0
##     tool           0        1        0         1        1         0      0
##     vegetable      0        2        1         0        0         0      0
##     vehicle        0        1        0         0        0         1      0
##              knn.pred
## pcs.test.labs kitchen manmade tool vegetable vehicle
##     animal          0       0    2         2       0
##     bodypart        0       0    1         1       0
##     building        0       1    0         0       1
##     buildpart       0       0    0         0       0
##     clothing        0       1    0         0       1
##     furniture       0       0    2         0       1
##     insect          0       0    0         2       0
##     kitchen         0       0    3         0       0
##     manmade         0       0    1         0       1
##     tool            0       2    0         0       0
##     vegetable       1       0    0         0       1
##     vehicle         1       2    0         0       0
```

```
print(mean(knn.pred == pcs.test$grp))
```

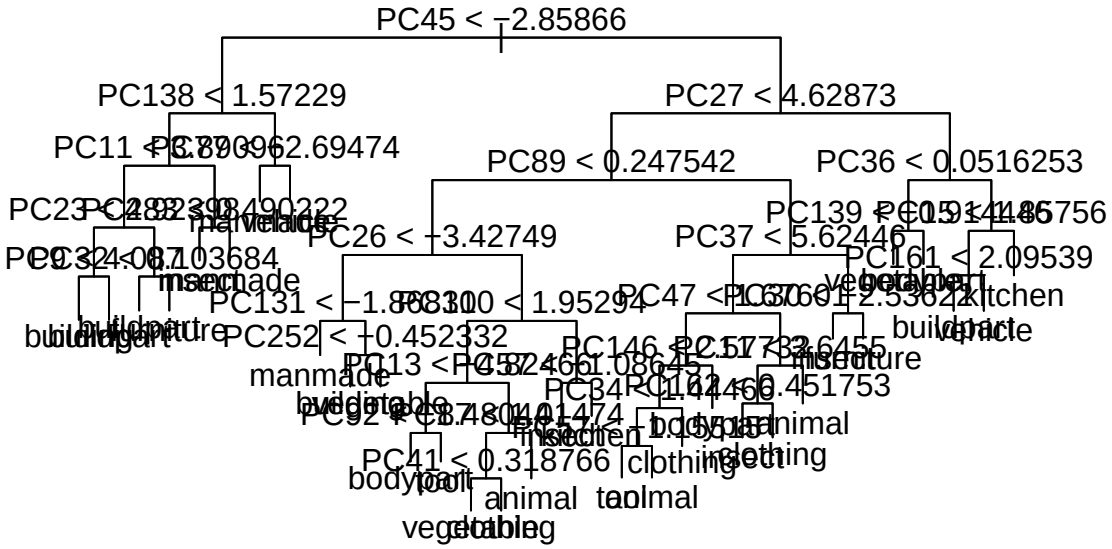
```
## [1] 0.08333333
```

```
# Decision Trees
set.seed(100)
tree.fit <- tree(as.factor(grp) ~ ., data = pcs.train)
```

```
summary(tree.fit)
```

```
##
## Classification tree:
## tree(formula = as.factor(grp) ~ ., data = pcs.train)
## Variables actually used in tree construction:
##  [1] "PC45"  "PC138" "PC11"  "PC23"  "PC9"   "PC32"  "PC283" "PC77"  "PC27"
## [10] "PC89"  "PC26"  "PC131" "PC252" "PC100" "PC13"  "PC92"  "PC87"  "PC41"
## [19] "PC57"  "PC37"  "PC47"  "PC146" "PC34"  "PC162" "PC30"  "PC36"  "PC139"
## [28] "PC15"  "PC161"
## Number of terminal nodes:  32
## Residual mean deviance:  2.209 = 592.1 / 268
## Misclassification error rate: 0.4233 = 127 / 300
```

```
plot(tree.fit)
text(tree.fit, pretty = 0)
```




```
tree.pred <- predict(tree.fit, newdata = pcs.test, type = "class")
tree.pred
```

```
##  [1] buildpart insect    clothing  building  clothing  buildpart vegetable
##  [8] tool      vegetable vegetable animal    manmade   vegetable building
## [15] vegetable insect    vegetable buildpart vehicle   building  vehicle
## [22] clothing  vehicle   vehicle   vegetable animal    building  kitchen
## [29] insect    tool      tool      buildpart kitchen   insect    buildpart
## [36] tool      clothing  clothing  clothing  bodypart  buildpart insect
## [43] buildpart vehicle   bodypart  kitchen   kitchen   insect    tool
```

```
## [50] vegetable vehicle   bodypart  buildpart vehicle   kitchen   vegetable
## [57] furniture building  vegetable furniture
## 12 Levels: animal bodypart building buildpart clothing furniture ... vehicle
```

```
confusion_mat.dt = as.matrix(table(pcs.test.labs, tree.pred))
print(confusion_mat.dt)
```

```
##              tree.pred
## pcs.test.labs animal bodypart building buildpart clothing furniture insect
##     animal         1        0        1         0        1         0      0
##     bodypart       0        0        0         0        0         0      3
##     building       0        0        1         1        0         1      0
##     buildpart      0        0        1         0        1         0      0
##     clothing       1        0        0         0        1         0      0
##     furniture      0        0        0         1        0         0      1
##     insect         0        0        1         1        1         0      0
##     kitchen        0        0        0         2        0         0      1
##     manmade        0        0        0         2        1         0      0
##     tool           0        3        0         0        0         0      0
##     vegetable      0        0        0         0        1         0      0
##     vehicle        0        0        1         1        0         1      1
##              tree.pred
## pcs.test.labs kitchen manmade tool vegetable vehicle
##     animal          2       0    0         0       0
##     bodypart        1       0    0         0       1
##     building        0       0    1         1       0
##     buildpart       1       0    0         1       1
##     clothing        0       0    2         1       0
##     furniture       0       1    0         2       0
##     insect          0       0    0         1       1
##     kitchen         0       0    1         1       0
##     manmade         0       0    0         0       2
##     tool            0       0    0         1       1
##     vegetable       1       0    1         2       0
##     vehicle         0       0    0         0       1
```

```
print(mean(tree.pred == pcs.test$grp))
```

```
## [1] 0.1
```

```
# Random Forest

rf.fit <- randomForest(as.factor(grp) ~ ., data = pcs.train,, mtry = 80, importance = TRUE)
summary(rf.fit)
```

```
##                Length Class  Mode
## call              6   -none- call
## type              1   -none- character
## predicted       300   factor numeric
## err.rate       6500   -none- numeric
## confusion       156   -none- numeric
## votes          3600   matrix numeric
## oob.times       300   -none- numeric
## classes          12   -none- character
## importance     4200   -none- numeric
## importanceSD   3900   -none- numeric
```

```
## localImportance    0   -none- NULL
## proximity          0   -none- NULL
## ntree              1   -none- numeric
## mtry               1   -none- numeric
## forest            14   -none- list
## y                300   factor numeric
## test               0   -none- NULL
## inbag              0   -none- NULL
## terms              3   terms  call
```

```
rf.pred <- predict(rf.fit, newdata = pcs.test, type = "class")
confusion_mat.rf = as.matrix(table(pcs.test.labs, rf.pred))
print(confusion_mat.rf)
```

```
##              rf.pred
## pcs.test.labs animal bodypart building buildpart clothing furniture insect
##     animal         0        1        1         0        0         1      0
##     bodypart       0        0        0         0        0         0      0
##     building       0        0        0         0        1         1      0
##     buildpart      0        1        1         2        0         0      0
##     clothing       1        0        0         0        2         0      0
##     furniture      0        2        1         1        0         0      0
##     insect         1        0        0         0        0         0      1
##     kitchen        0        0        0         1        0         1      1
##     manmade        1        1        1         0        0         0      0
##     tool           0        0        0         0        0         0      0
##     vegetable      2        0        2         0        0         0      0
##     vehicle        1        0        1         0        0         1      0
##              rf.pred
## pcs.test.labs kitchen manmade tool vegetable vehicle
##     animal          0       0    1         0       1
##     bodypart        1       1    2         0       1
##     building        0       0    0         2       1
##     buildpart       0       1    0         0       0
##     clothing        2       0    0         0       0
##     furniture       0       0    1         0       0
##     insect          0       1    0         0       2
##     kitchen         0       0    2         0       0
##     manmade         0       0    0         1       1
##     tool            1       1    0         2       1
##     vegetable       0       0    0         1       0
##     vehicle         0       0    1         0       1
```

```
print(mean(rf.pred == pcs.test$grp))
```

```
## [1] 0.1166667
```

```
manmade <- c("furniture", "clothing", "manmade", "tool", "kitchen", "vehicle", "building", "buildpart")
natural <- c("insect", "animal", "vegetable", "bodypart")
df_new <- within(pcs, {
 cls <- "manmade"
 cls[grp %in% manmade] <- "manmade"
 cls[grp %in% natural] <- "natural"
})
pcs$cls <- df_new$cls
```

```r
# Splitting data into training and test data
set.seed(100)
#sample <- sample(1:nrow(pcs), 300)
pcs.train <- pcs[1:300,]
pcs.test <- pcs[301:360,]
#pcs.train <- subset(pcs, sample == TRUE)
#pcs.test  <- subset(pcs, sample == FALSE)
pcs.train.x <- subset(pcs.train, select = -c(grp,cls))
pcs.train.labs <- pcs.train$cls
pcs.test.x <- subset(pcs.test, select = -c(grp,cls))
pcs.test.labs <- pcs.test$cls
```

```r
# Classification Algorithms

# Naive Bayes Classifier

nb.fit <- naiveBayes(cls ~ . , data = pcs.train)
nb.class <- predict(nb.fit,pcs.test.x)
nb.class
```

```
##  [1] manmade manmade manmade manmade manmade manmade natural manmade manmade
## [10] manmade manmade manmade manmade manmade manmade natural manmade natural
## [19] manmade manmade manmade manmade manmade manmade manmade natural manmade
## [28] manmade natural manmade manmade manmade manmade manmade manmade manmade
## [37] manmade manmade natural manmade manmade manmade manmade manmade manmade
## [46] manmade manmade natural manmade manmade manmade manmade manmade manmade
## [55] natural manmade manmade manmade manmade manmade
## Levels: manmade natural
```

```r
confusion_mat.nb = as.matrix(table(Actual_Values = pcs.test.labs, Predicted_Values = nb.class))
print(confusion_mat.nb)
```

```
##              Predicted_Values
## Actual_Values manmade natural
##       manmade      35       5
##       natural      17       3
```

```r
print(mean(nb.class == pcs.test$cls))
```

```
## [1] 0.6333333
```

```r
# KNN

knn.pred <- knn(pcs.train.x, pcs.test.x, pcs.train.labs, k=3)

confusion_mat.knn = as.matrix(table(pcs.test.labs, knn.pred))
print(confusion_mat.knn)
```

```
##              knn.pred
## pcs.test.labs manmade natural
##       manmade      32       8
##       natural       9      11
```

```r
print(mean(knn.pred== pcs.test$cls))
```

```
## [1] 0.7166667
```

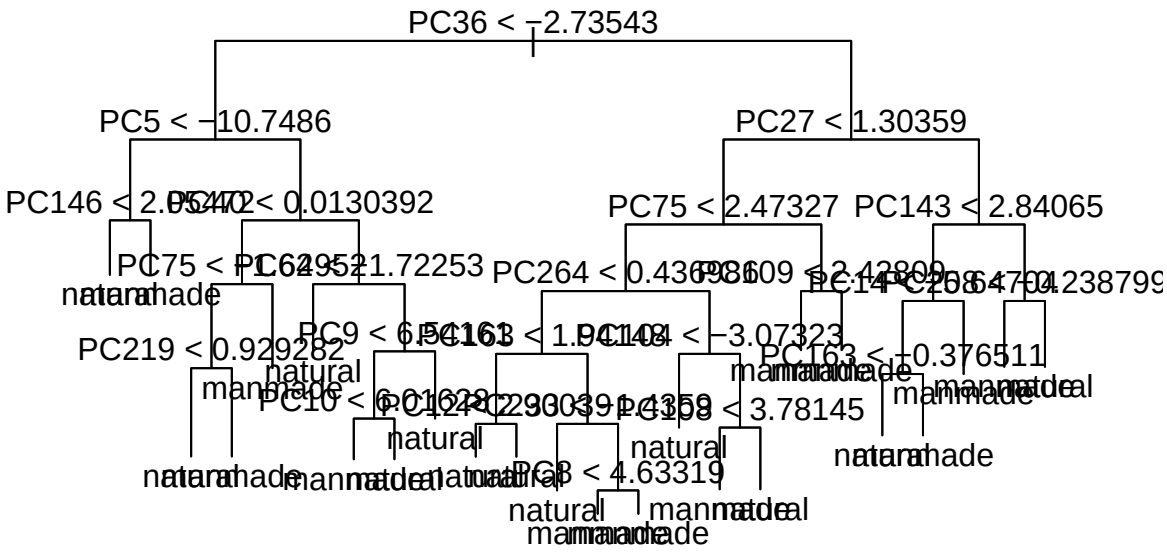
```r
# Decision Trees
set.seed(100)
tree.fit <- tree(as.factor(cls) ~ ., data = pcs.train)
```

```
## Warning in tree(as.factor(cls) ~ ., data = pcs.train): NAs introduced by
## coercion
```

```r
summary(tree.fit)
```

```
##
## Classification tree:
## tree(formula = as.factor(cls) ~ ., data = pcs.train)
## Variables actually used in tree construction:
##  [1] "PC36"  "PC5"   "PC146" "PC40"  "PC75"  "PC219" "PC62"  "PC9"   "PC10"
## [10] "PC27"  "PC264" "PC163" "PC12"  "PC233" "PC8"   "PC104" "PC108" "PC109"
## [19] "PC143" "PC14"  "PC208"
## Number of terminal nodes:  24
## Residual mean deviance:  0.2615 = 72.16 / 276
## Misclassification error rate: 0.06667 = 20 / 300
```

```r
plot(tree.fit)
text(tree.fit, pretty = 0)
```




```r
tree.pred <- predict(tree.fit, newdata = pcs.test, type = "class")
```

```
## Warning in pred1.tree(object, tree.matrix(newdata)): NAs introduced by coercion
```

```
tree.pred
```

```
##  [1] natural natural natural manmade natural natural natural manmade manmade
## [10] manmade manmade manmade manmade manmade manmade natural manmade manmade
## [19] natural natural natural manmade manmade manmade natural natural manmade
## [28] manmade natural natural manmade manmade manmade manmade manmade natural
## [37] natural manmade natural manmade manmade natural natural natural manmade
## [46] manmade manmade natural natural natural manmade natural manmade natural
## [55] manmade manmade natural manmade manmade natural
## Levels: manmade natural
```

```
confusion_mat.dt = as.matrix(table(pcs.test.labs, tree.pred))
print(confusion_mat.dt)
```

```
##              tree.pred
## pcs.test.labs manmade natural
##       manmade      24      16
##       natural       9      11
```

```
print(mean(tree.pred== pcs.test$cls))
```

```
## [1] 0.5833333
```

```
# Random Forest

rf.fit <- randomForest(as.factor(cls) ~ ., data = pcs.train,, mtry = 80, importance = TRUE)
summary(rf.fit)
```

```
##                 Length Class  Mode
## call               6   -none- call
## type               1   -none- character
## predicted        300   factor numeric
## err.rate        1500   -none- numeric
## confusion          6   -none- numeric
## votes            600   matrix numeric
## oob.times        300   -none- numeric
## classes            2   -none- character
## importance      1204   -none- numeric
## importanceSD     903   -none- numeric
## localImportance    0   -none- NULL
## proximity          0   -none- NULL
## ntree              1   -none- numeric
## mtry               1   -none- numeric
## forest            14   -none- list
## y                300   factor numeric
## test               0   -none- NULL
## inbag              0   -none- NULL
## terms              3   terms  call
```

```
rf.pred <- predict(rf.fit, newdata = pcs.test, type = "class")
confusion_mat.rf = as.matrix(table(pcs.test.labs, rf.pred))
print(confusion_mat.rf)
```

```
##              rf.pred
## pcs.test.labs manmade natural
##       manmade      40       0
##       natural      10      10
```

```
print(mean(rf.pred== pcs.test$cls))
```

```
## [1] 0.8333333
```