patient9

Anuhya B S

2022-06-09

```
#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-P9.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] 0.3675054 0.4327803 0.4790144 0.5014683 0.5228530 0.5410088 0.5566538
##   [8] 0.5700061 0.5829515 0.5948079 0.6048651 0.6130246 0.6209160 0.6280424
##  [15] 0.6344185 0.6403233 0.6457196 0.6505302 0.6551881 0.6595356 0.6637704
##  [22] 0.6679014 0.6719160 0.6758192 0.6795210 0.6830529 0.6865717 0.6899127
##  [29] 0.6932039 0.6963851 0.6994571 0.7024948 0.7054560 0.7083006 0.7110526
##  [36] 0.7136801 0.7163041 0.7188245 0.7213023 0.7236958 0.7260182 0.7283126
##  [43] 0.7305439 0.7327689 0.7349290 0.7370403 0.7391118 0.7411553 0.7431807
##  [50] 0.7451570 0.7470798 0.7489629 0.7508308 0.7526480 0.7544500 0.7562357
##  [57] 0.7579627 0.7596730 0.7613595 0.7630095 0.7646394 0.7662472 0.7678327
##  [64] 0.7694076 0.7709668 0.7724999 0.7740295 0.7755265 0.7769938 0.7784536
##  [71] 0.7798828 0.7813104 0.7827116 0.7840983 0.7854802 0.7868442 0.7881949
##  [78] 0.7895282 0.7908598 0.7921692 0.7934639 0.7947543 0.7960359 0.7972954
##  [85] 0.7985464 0.7997855 0.8010161 0.8022453 0.8034636 0.8046715 0.8058586
##  [92] 0.8070419 0.8082216 0.8093933 0.8105522 0.8117045 0.8128505 0.8139932
##  [99] 0.8151275 0.8162599 0.8173893 0.8185038 0.8196168 0.8207219 0.8218146
## [106] 0.8229052 0.8239880 0.8250634 0.8261285 0.8271913 0.8282487 0.8292966
## [113] 0.8303404 0.8313764 0.8324094 0.8334379 0.8344598 0.8354771 0.8364904
## [120] 0.8374988 0.8384961 0.8394893 0.8404784 0.8414622 0.8424379 0.8434106
## [127] 0.8443814 0.8453484 0.8463113 0.8472724 0.8482262 0.8491788 0.8501275
## [134] 0.8510741 0.8520168 0.8529578 0.8538960 0.8548312 0.8557559 0.8566794
## [141] 0.8575989 0.8585139 0.8594251 0.8603331 0.8612389 0.8621410 0.8630415
## [148] 0.8639370 0.8648303 0.8657229 0.8666094 0.8674924 0.8683731 0.8692507
## [155] 0.8701272 0.8709971 0.8718649 0.8727304 0.8735955 0.8744562 0.8753123
## [162] 0.8761651 0.8770131 0.8778608 0.8787030 0.8795448 0.8803805 0.8812139
## [169] 0.8820449 0.8828732 0.8836977 0.8845199 0.8853399 0.8861584 0.8869757
## [176] 0.8877894 0.8886004 0.8894081 0.8902128 0.8910152 0.8918176 0.8926182
## [183] 0.8934176 0.8942134 0.8950064 0.8957971 0.8965860 0.8973727 0.8981586
## [190] 0.8989424 0.8997233 0.9005012 0.9012774 0.9020488 0.9028170 0.9035843
## [197] 0.9043486 0.9051101 0.9058702 0.9066294 0.9073870 0.9081420 0.9088935
## [204] 0.9096441 0.9103946 0.9111418 0.9118867 0.9126308 0.9133706 0.9141084
## [211] 0.9148441 0.9155766 0.9163080 0.9170362 0.9177637 0.9184875 0.9192081
## [218] 0.9199273 0.9206441 0.9213608 0.9220751 0.9227883 0.9234990 0.9242087
## [225] 0.9249152 0.9256194 0.9263224 0.9270227 0.9277220 0.9284197 0.9291153
## [232] 0.9298090 0.9304999 0.9311902 0.9318790 0.9325655 0.9332507 0.9339343
## [239] 0.9346150 0.9352936 0.9359685 0.9366428 0.9373147 0.9379847 0.9386531
## [246] 0.9393191 0.9399835 0.9406455 0.9413059 0.9419659 0.9426231 0.9432773
## [253] 0.9439283 0.9445788 0.9452263 0.9458716 0.9465152 0.9471578 0.9477977
## [260] 0.9484362 0.9490723 0.9497079 0.9503410 0.9509718 0.9516007 0.9522280
## [267] 0.9528533 0.9534765 0.9540984 0.9547181 0.9553372 0.9559542 0.9565689
## [274] 0.9571824 0.9577940 0.9584028 0.9590092 0.9596136 0.9602173 0.9608196
## [281] 0.9614190 0.9620178 0.9626143 0.9632101 0.9638053 0.9643974 0.9649859
## [288] 0.9655696 0.9661522 0.9667334 0.9673143 0.9678945 0.9684724 0.9690492
## [295] 0.9696219 0.9701928 0.9707618 0.9713294 0.9718929 0.9724551 0.9730161
## [302] 0.9735746 0.9741307 0.9746864 0.9752376 0.9757873 0.9763349 0.9768804
## [309] 0.9774233 0.9779631 0.9785016 0.9790376 0.9795716 0.9801045 0.9806370
## [316] 0.9811659 0.9816919 0.9822156 0.9827358 0.9832530 0.9837692 0.9842828
## [323] 0.9847920 0.9852946 0.9857964 0.9862918 0.9867765 0.9872593 0.9877372
## [330] 0.9882126 0.9886746 0.9891304 0.9895840 0.9900332 0.9904767 0.9909155
## [337] 0.9913511 0.9917858 0.9922156 0.9926414 0.9930597 0.9934737 0.9938859
## [344] 0.9942941 0.9947009 0.9951034 0.9955016 0.9958964 0.9962862 0.9966725
## [351] 0.9970564 0.9974368 0.9978130 0.9981868 0.9985590 0.9989268 0.9992887
## [358] 0.9996486 1.0000000 1.0000000
```

```r
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   clothing  manmade   vegetable vehicle   insect    vehicle
##  [8] furniture vehicle   insect    buildpart buildpart manmade   insect
## [15] bodypart  tool      vegetable manmade   manmade   manmade   vehicle
## [22] tool      vegetable vegetable manmade   manmade   insect    vehicle
## [29] buildpart vegetable manmade   vegetable vegetable manmade   insect
## [36] clothing  vehicle   manmade   clothing  clothing  manmade   insect
## [43] vehicle   building  insect    furniture vehicle   insect    vehicle
## [50] furniture tool      tool      vegetable animal    building  clothing
## [57] tool      vegetable manmade   buildpart
## 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        0        0         1        0         0      0
##     bodypart       0        0        1         0        0         1      2
##     building       0        0        0         0        0         0      1
##     buildpart      0        0        1         0        2         0      0
##     clothing       0        0        0         0        1         0      1
##     furniture      0        1        0         1        0         0      1
##     insect         0        0        0         0        0         0      1
##     kitchen        0        0        0         0        1         2      1
##     manmade        1        0        0         0        0         0      0
##     tool           0        0        0         0        1         0      1
##     vegetable      0        0        0         0        0         0      0
##     vehicle        0        0        0         2        0         0      0
##               Predicted_Values
## Actual_Values kitchen manmade tool vegetable vehicle
##     animal          0       1    0         1       2
##     bodypart        0       0    1         0       0
##     building        0       2    1         0       1
##     buildpart       0       1    0         1       0
##     clothing        0       2    0         0       1
##     furniture       0       1    0         1       0
##     insect          0       1    0         0       3
##     kitchen         0       0    0         1       0
##     manmade         0       1    2         0       1
##     tool            0       1    1         1       0
##     vegetable       0       2    0         2       1
##     vehicle         0       1    0         2       0
```

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

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

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

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

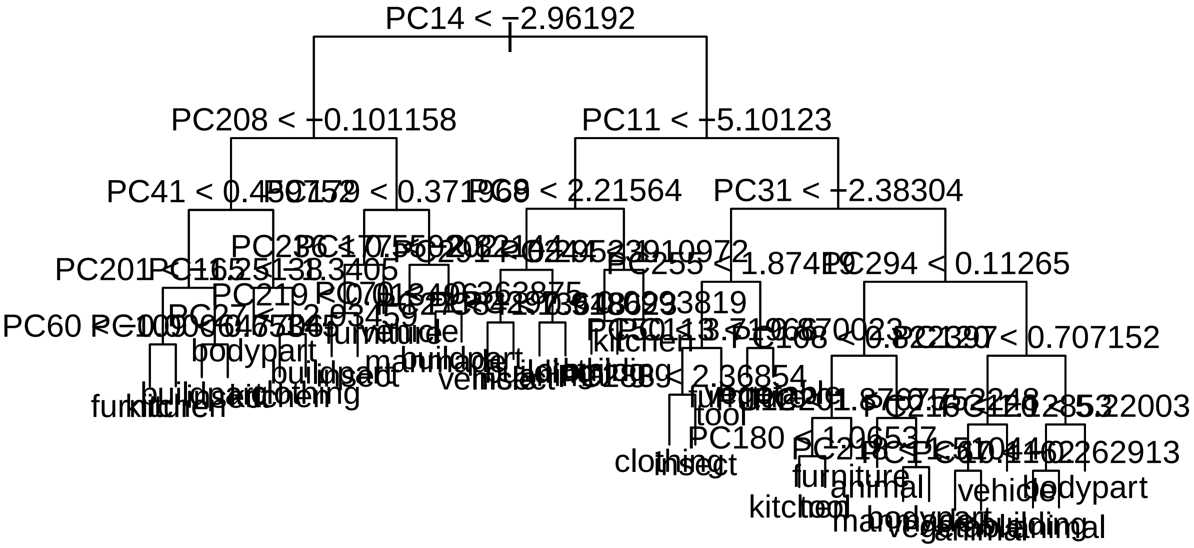
```
## [1] 0.08333333
```

```r
# 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] "PC14"  "PC208" "PC41"  "PC201" "PC60"  "PC109" "PC165" "PC27"  "PC179"
## [10] "PC236" "PC219" "PC177" "PC70"  "PC11"  "PC8"   "PC291" "PC84"  "PC244"
## [19] "PC297" "PC31"  "PC255" "PC50"  "PC283" "PC113" "PC294" "PC108" "PC12"
## [28] "PC180" "PC218" "PC120" "PC216" "PC1"   "PC67"
## Number of terminal nodes:  37
## Residual mean deviance:  2.009 = 528.3 / 263
## Misclassification error rate: 0.3867 = 116 / 300
```

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



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

```
##  [1] furniture buildpart clothing  building  building  building  building
##  [8] vehicle   clothing  building  tool      animal    vehicle   vegetable
## [15] buildpart furniture kitchen   manmade   kitchen   clothing  building
## [22] furniture insect    furniture animal    kitchen   buildpart insect
## [29] kitchen   vehicle   animal    clothing  insect    kitchen   building
## [36] buildpart insect    clothing  bodypart  buildpart kitchen   manmade
## [43] animal    insect    buildpart animal    kitchen   building  manmade
## [50] vehicle   vegetable furniture furniture kitchen   vehicle   kitchen  
## [57] insect    manmade   insect    manmade  
## 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         0        0        0         0        1         0      1
##     bodypart       1        0        1         0        0         1      1
##     building       0        0        0         0        0         0      1
##     buildpart      0        1        1         0        0         0      0
##     clothing       1        0        2         1        0         0      0
##     furniture      1        0        1         1        0         0      0
##     insect         0        0        1         1        1         1      1
##     kitchen        0        0        1         1        1         0      0
##     manmade        1        0        0         0        0         1      0
##     tool           0        0        0         2        0         2      1
##     vegetable      1        0        1         0        1         0      1
##     vehicle        0        0        0         0        1         1      1
##               tree.pred
## pcs.test.labs kitchen manmade tool vegetable vehicle
##     animal          1       1    1         0       0
##     bodypart        0       1    0         0       0
##     building        1       1    0         1       1
##     buildpart       2       0    0         0       1
##     clothing        1       0    0         0       0
##     furniture       2       0    0         0       0
##     insect          0       0    0         0       0
##     kitchen         0       0    0         0       2
##     manmade         1       1    0         1       0
##     tool            0       0    0         0       0
##     vegetable       0       0    0         0       1
##     vehicle         1       1    0         0       0
```

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

```
## [1] 0.03333333
```

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

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

```
## [1] 0.06666667
```

```
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
```

```
# 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
```

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

```
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      33       7
##       natural      17       3
```

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

```
## [1] 0.6
```

```
# KNN

knn.pred <- knn(pcs.train.x, pcs.test.x, pcs.train.labs, k=7)

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

```
##               knn.pred
## pcs.test.labs manmade natural
##       manmade      35       5
##       natural      17       3
```

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

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

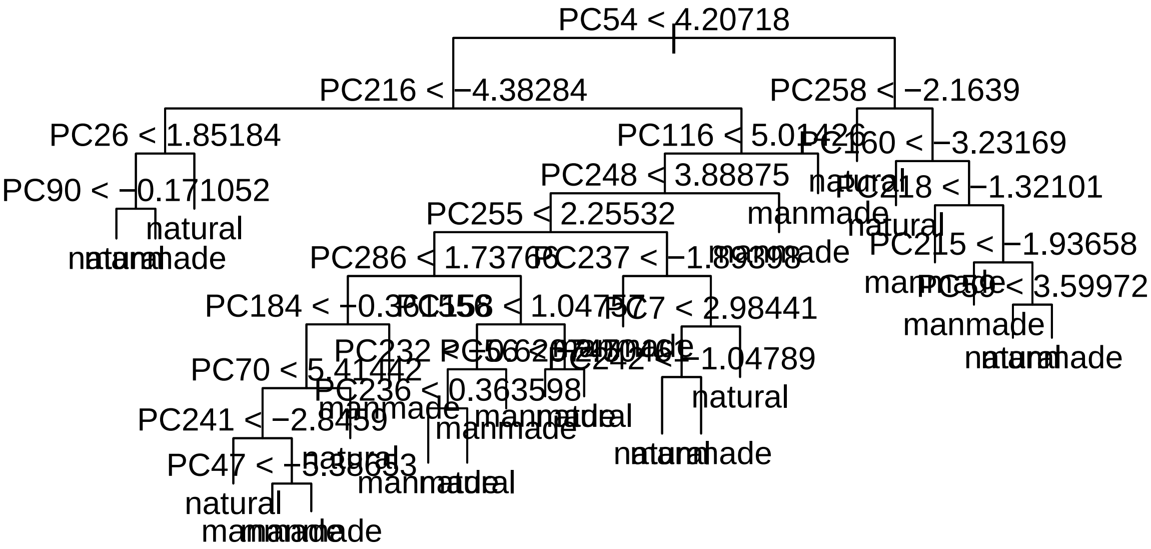
```
# Decision Trees
set.seed(100)
tree.fit <- tree(as.factor(cls) ~ ., data = pcs.train)
```

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

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

```
##
## Classification tree:
## tree(formula = as.factor(cls) ~ ., data = pcs.train)
## Variables actually used in tree construction:
##  [1] "PC54"  "PC216" "PC26"  "PC90"  "PC116" "PC248" "PC255" "PC286" "PC184"
## [10] "PC70"  "PC241" "PC47"  "PC158" "PC232" "PC236" "PC56"  "PC237" "PC7"
## [19] "PC242" "PC258" "PC160" "PC218" "PC215" "PC59"
## Number of terminal nodes:  25
## Residual mean deviance:  0.1521 = 41.83 / 275
## Misclassification error rate: 0.03333 = 10 / 300
```

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



```
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] manmade natural manmade natural manmade manmade manmade manmade manmade
## [10] natural manmade natural manmade natural natural natural natural natural
## [19] manmade manmade manmade manmade manmade manmade manmade natural manmade
## [28] manmade manmade natural manmade manmade natural manmade natural manmade
## [37] natural manmade natural manmade manmade manmade manmade manmade natural
## [46] manmade natural manmade natural manmade natural manmade manmade manmade
## [55] manmade manmade manmade manmade natural manmade
## 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      25      15
##       natural      15       5
```

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

```
## [1] 0.5
```

```
# 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
```