Patient1

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-P1.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] 0.1284251 0.1818392 0.2276481 0.2576118 0.2872661 0.3132465 0.3335796
```

```
##   [8] 0.3527293 0.3681385 0.3820825 0.3947036 0.4062525 0.4173842 0.4280308
##  [15] 0.4375966 0.4465732 0.4543799 0.4620049 0.4689290 0.4757182 0.4817589
##  [22] 0.4874297 0.4929791 0.4984840 0.5036335 0.5086756 0.5135455 0.5181183
##  [29] 0.5225876 0.5269454 0.5312656 0.5353798 0.5394489 0.5432898 0.5471219
##  [36] 0.5508520 0.5544628 0.5580128 0.5615106 0.5648759 0.5681696 0.5713843
##  [43] 0.5745205 0.5776289 0.5806993 0.5837051 0.5866697 0.5895515 0.5924070
##  [50] 0.5951867 0.5979022 0.6005987 0.6032598 0.6058986 0.6085061 0.6111015
##  [57] 0.6136590 0.6161896 0.6186408 0.6210702 0.6234894 0.6258829 0.6282246
##  [64] 0.6305573 0.6328475 0.6351255 0.6373934 0.6396218 0.6418282 0.6440231
##  [71] 0.6461943 0.6483202 0.6504356 0.6525346 0.6546134 0.6566890 0.6587281
##  [78] 0.6607632 0.6627869 0.6647940 0.6667813 0.6687531 0.6707123 0.6726591
##  [85] 0.6745922 0.6765038 0.6784058 0.6802965 0.6821739 0.6840322 0.6858838
##  [92] 0.6877256 0.6895625 0.6913841 0.6931918 0.6949893 0.6967818 0.6985605
##  [99] 0.7003318 0.7020908 0.7038486 0.7056030 0.7073536 0.7090855 0.7108106
## [106] 0.7125278 0.7142363 0.7159332 0.7176253 0.7193100 0.7209896 0.7226623
## [113] 0.7243245 0.7259794 0.7276231 0.7292573 0.7308901 0.7325131 0.7341298
## [120] 0.7357434 0.7373450 0.7389380 0.7405306 0.7421171 0.7436984 0.7452743
## [127] 0.7468443 0.7484081 0.7499681 0.7515205 0.7530648 0.7546054 0.7561360
## [134] 0.7576637 0.7591828 0.7606968 0.7622096 0.7637181 0.7652167 0.7667107
## [141] 0.7682001 0.7696846 0.7711627 0.7726376 0.7741053 0.7755707 0.7770307
## [148] 0.7784896 0.7799421 0.7813922 0.7828379 0.7842789 0.7857138 0.7871459
## [155] 0.7885751 0.7899926 0.7914093 0.7928238 0.7942351 0.7956424 0.7970403
## [162] 0.7984342 0.7998233 0.8012091 0.8025885 0.8039604 0.8053312 0.8066948
## [169] 0.8080457 0.8093953 0.8107408 0.8120825 0.8134215 0.8147590 0.8160907
## [176] 0.8174193 0.8187422 0.8200624 0.8213764 0.8226885 0.8239955 0.8252985
## [183] 0.8265976 0.8278930 0.8291873 0.8304766 0.8317647 0.8330474 0.8343268
## [190] 0.8356033 0.8368765 0.8381449 0.8394082 0.8406668 0.8419220 0.8431727
## [197] 0.8444175 0.8456611 0.8469030 0.8481436 0.8493778 0.8506072 0.8518338
## [204] 0.8530583 0.8542780 0.8554950 0.8567085 0.8579187 0.8591236 0.8603230
## [211] 0.8615208 0.8627151 0.8639066 0.8650902 0.8662726 0.8674506 0.8686230
## [218] 0.8697942 0.8709626 0.8721254 0.8732875 0.8744444 0.8755995 0.8767528
## [225] 0.8779018 0.8790467 0.8801904 0.8813300 0.8824672 0.8836011 0.8847317
## [232] 0.8858543 0.8869738 0.8880899 0.8892036 0.8903152 0.8914228 0.8925276
## [239] 0.8936298 0.8947297 0.8958257 0.8969177 0.8980050 0.8990891 0.9001706
## [246] 0.9012497 0.9023274 0.9034014 0.9044665 0.9055287 0.9065863 0.9076431
## [253] 0.9086986 0.9097497 0.9107977 0.9118449 0.9128858 0.9139252 0.9149613
## [260] 0.9159926 0.9170225 0.9180498 0.9190713 0.9200909 0.9211065 0.9221209
## [267] 0.9231316 0.9241400 0.9251473 0.9261482 0.9271489 0.9281477 0.9291423
## [274] 0.9301323 0.9311201 0.9321024 0.9330837 0.9340634 0.9350411 0.9360148
## [281] 0.9369843 0.9379506 0.9389138 0.9398714 0.9408258 0.9417777 0.9427277
## [288] 0.9436767 0.9446210 0.9455622 0.9464963 0.9474277 0.9483578 0.9492835
## [295] 0.9502066 0.9511273 0.9520405 0.9529517 0.9538585 0.9547611 0.9556614
## [302] 0.9565580 0.9574522 0.9583431 0.9592326 0.9601173 0.9609995 0.9618804
## [309] 0.9627590 0.9636339 0.9645072 0.9653751 0.9662365 0.9670962 0.9679525
## [316] 0.9688077 0.9696531 0.9704918 0.9713281 0.9721641 0.9729951 0.9738193
## [323] 0.9746417 0.9754547 0.9762665 0.9770750 0.9778775 0.9786761 0.9794723
## [330] 0.9802584 0.9810366 0.9818025 0.9825608 0.9833120 0.9840564 0.9847937
## [337] 0.9855226 0.9862507 0.9869728 0.9876895 0.9884010 0.9890982 0.9897930
## [344] 0.9904840 0.9911637 0.9918337 0.9924965 0.9931576 0.9938106 0.9944580
## [351] 0.9951030 0.9957330 0.9963593 0.9969815 0.9975988 0.9982117 0.9988151
## [358] 0.9994108 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] tool      buildpart animal    vegetable clothing  furniture building
##  [8] buildpart building  animal    vehicle   tool      kitchen   bodypart
## [15] insect    bodypart  animal    tool      manmade   kitchen   tool
## [22] vehicle   bodypart  buildpart animal    insect    animal    buildpart
## [29] manmade   kitchen   tool      vehicle   building  animal    buildpart
## [36] manmade   bodypart  building  tool      clothing  clothing  furniture
## [43] animal    bodypart  tool      bodypart  insect    vehicle   buildpart
## [50] insect    tool      tool      buildpart bodypart  buildpart insect
## [57] manmade   vegetable buildpart clothing
## 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         1        0        0         1        0         0      1
##     bodypart       0        3        0         0        0         1      0
##     building       0        1        0         1        1         0      0
##     buildpart      0        0        0         1        0         0      1
##     clothing       1        0        0         0        1         0      1
##     furniture      2        0        0         1        0         0      1
##     insect         1        1        1         0        0         0      0
##     kitchen        0        0        0         2        0         1      1
##     manmade        1        1        0         0        0         0      0
##     tool           0        0        0         2        1         0      0
##     vegetable      1        0        3         0        0         0      0
##     vehicle        0        1        0         1        1         0      0
##              Predicted_Values
## Actual_Values kitchen manmade tool vegetable vehicle
##     animal          0       0    0         1       1
##     bodypart        0       0    0         0       1
##     building        1       1    0         0       0
```

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

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

```
## [1] 0.1333333
```

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

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

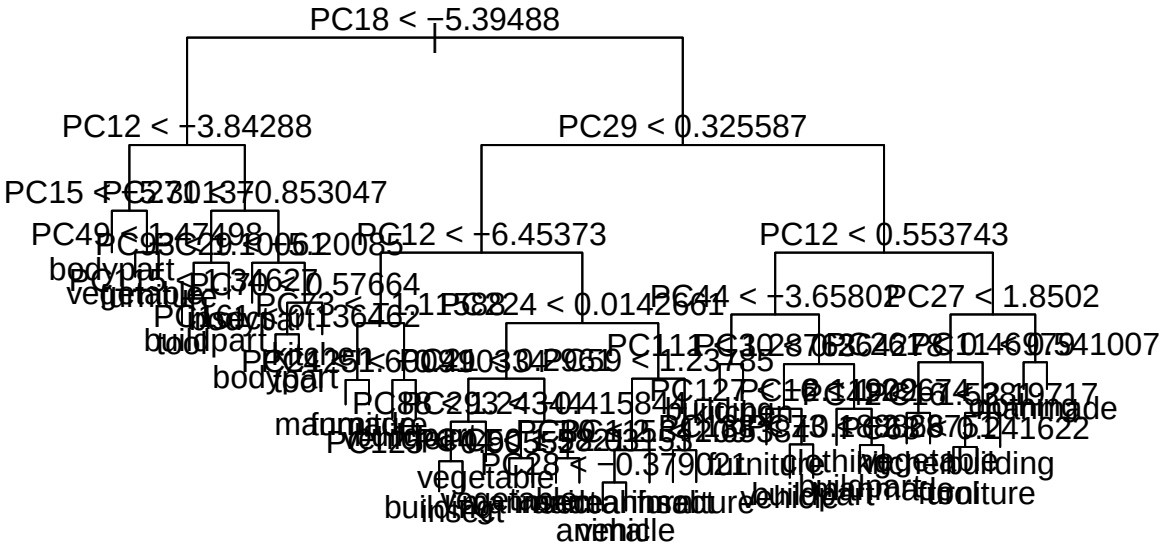
```
## [1] 0.2166667
```

```
# 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] "PC18"  "PC12"  "PC15"  "PC49"  "PC271" "PC93"  "PC115" "PC29"  "PC70"
## [10] "PC161" "PC73"  "PC24"  "PC125" "PC224" "PC21"  "PC88"  "PC293" "PC44"
## [19] "PC50"  "PC59"  "PC30"  "PC28"  "PC111" "PC127" "PC138" "PC16"  "PC27"
## [28] "PC261" "PC42"  "PC68"  "PC11"
## Number of terminal nodes:  41
## Residual mean deviance:  1.689 = 437.5 / 259
## Misclassification error rate: 0.3333 = 100 / 300
```

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




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

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

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

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

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

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

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

```
## [1] 0.2833333
```

```
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 natural
## [10] manmade manmade natural manmade manmade manmade natural manmade natural
## [19] manmade manmade manmade manmade manmade manmade natural natural manmade
## [28] manmade manmade manmade manmade manmade manmade manmade manmade manmade
## [37] natural manmade manmade manmade manmade manmade natural natural manmade
## [46] manmade manmade manmade natural manmade manmade manmade manmade manmade
## [55] manmade 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      14       6
```

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

```
## [1] 0.6833333
```

```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 manmade natural
##       manmade      35       5
##       natural      14       6
```

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

```
## [1] 0.6833333
```

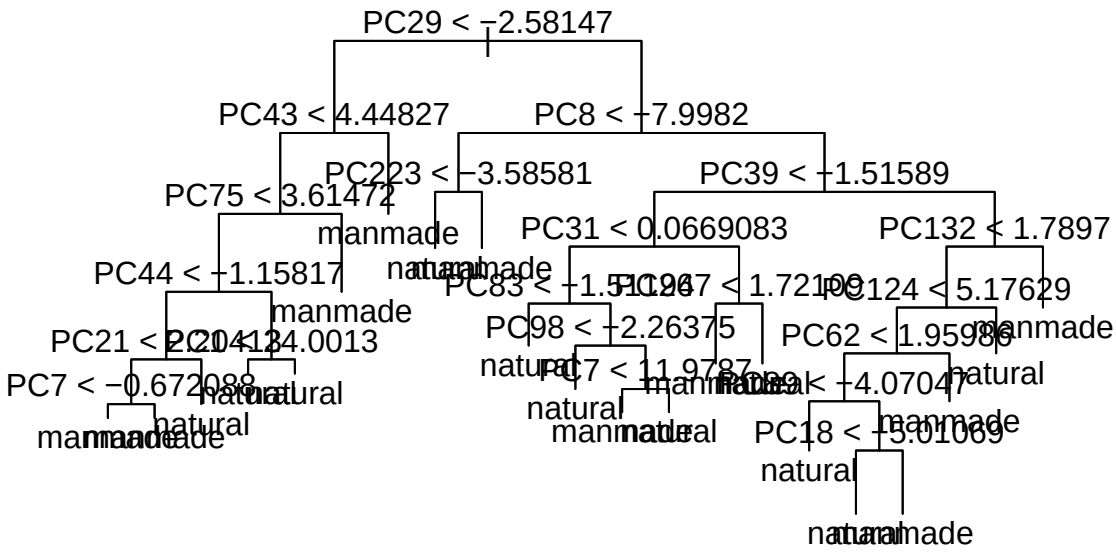
```
# Decision Trees
set.seed(100)
tree.fit <- tree(as.factor(cls) ~ ., data = pcs.train)
```

```
## 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] "PC29"  "PC43"  "PC75"  "PC44"  "PC21"  "PC7"   "PC1"   "PC8"   "PC223"
## [10] "PC39"  "PC31"  "PC83"  "PC98"  "PC267" "PC132" "PC124" "PC62"  "PC89"
## [19] "PC18"
## Number of terminal nodes:  21
## Residual mean deviance:  0.2567 = 71.61 / 279
## Misclassification error rate: 0.05 = 15 / 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] natural natural manmade manmade manmade manmade manmade natural manmade
## [10] manmade manmade manmade manmade manmade manmade manmade manmade natural
## [19] manmade manmade natural manmade manmade manmade manmade manmade manmade
## [28] manmade manmade manmade natural natural manmade manmade manmade natural
## [37] manmade natural manmade manmade manmade natural manmade natural natural
## [46] manmade natural manmade manmade manmade natural natural natural natural
## [55] natural natural natural natural 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      25      15
##       natural      13       7
```

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

```
## [1] 0.5333333
```

```
# 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
```