Patient2

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-P2.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] 0.2256587 0.3302634 0.3828161 0.4225564 0.4582193 0.4893703 0.5174784
```

```
##   [8] 0.5331294 0.5456276 0.5567261 0.5658891 0.5749577 0.5835982 0.5907722
##  [15] 0.5974944 0.6040386 0.6101400 0.6159957 0.6216657 0.6271564 0.6320288
##  [22] 0.6367942 0.6411057 0.6452899 0.6493287 0.6531179 0.6568051 0.6602454
##  [29] 0.6636152 0.6667650 0.6698704 0.6729370 0.6758938 0.6787623 0.6815864
##  [36] 0.6843343 0.6870187 0.6896485 0.6921915 0.6946628 0.6971086 0.6994939
##  [43] 0.7018098 0.7040707 0.7062688 0.7084444 0.7105888 0.7127175 0.7147741
##  [50] 0.7168209 0.7187780 0.7207125 0.7226251 0.7245314 0.7264160 0.7282663
##  [57] 0.7300852 0.7318955 0.7336780 0.7354336 0.7371745 0.7388809 0.7405730
##  [64] 0.7422410 0.7438820 0.7454964 0.7471020 0.7486832 0.7502592 0.7518296
##  [71] 0.7533834 0.7549171 0.7564367 0.7579430 0.7594175 0.7608871 0.7623511
##  [78] 0.7638078 0.7652445 0.7666697 0.7680901 0.7695055 0.7709053 0.7722871
##  [85] 0.7736627 0.7750312 0.7763755 0.7777184 0.7790478 0.7803739 0.7816943
##  [92] 0.7830033 0.7843103 0.7856125 0.7868957 0.7881752 0.7894495 0.7907152
##  [99] 0.7919789 0.7932298 0.7944680 0.7956994 0.7969239 0.7981441 0.7993589
## [106] 0.8005667 0.8017678 0.8029642 0.8041540 0.8053386 0.8065189 0.8076901
## [113] 0.8088584 0.8100215 0.8111801 0.8123322 0.8134788 0.8146208 0.8157575
## [120] 0.8168911 0.8180230 0.8191459 0.8202638 0.8213725 0.8224784 0.8235833
## [127] 0.8246804 0.8257699 0.8268566 0.8279399 0.8290226 0.8300944 0.8311650
## [134] 0.8322320 0.8332977 0.8343582 0.8354101 0.8364616 0.8375073 0.8385493
## [141] 0.8395830 0.8406156 0.8416439 0.8426686 0.8436865 0.8447032 0.8457145
## [148] 0.8467239 0.8477295 0.8487336 0.8497322 0.8507271 0.8517199 0.8527087
## [155] 0.8536952 0.8546783 0.8556603 0.8566400 0.8576167 0.8585906 0.8595619
## [162] 0.8605278 0.8614903 0.8624506 0.8634089 0.8643638 0.8653149 0.8662630
## [169] 0.8672088 0.8681513 0.8690925 0.8700271 0.8709615 0.8718919 0.8728158
## [176] 0.8737368 0.8746569 0.8755732 0.8764885 0.8774006 0.8783096 0.8792137
## [183] 0.8801152 0.8810155 0.8819116 0.8828067 0.8836994 0.8845895 0.8854757
## [190] 0.8863599 0.8872402 0.8881190 0.8889969 0.8898714 0.8907418 0.8916107
## [197] 0.8924750 0.8933360 0.8941962 0.8950525 0.8959049 0.8967551 0.8976036
## [204] 0.8984503 0.8992952 0.9001386 0.9009783 0.9018127 0.9026456 0.9034764
## [211] 0.9043055 0.9051322 0.9059576 0.9067778 0.9075970 0.9084132 0.9092289
## [218] 0.9100432 0.9108562 0.9116667 0.9124741 0.9132774 0.9140795 0.9148788
## [225] 0.9156725 0.9164660 0.9172569 0.9180450 0.9188308 0.9196160 0.9203985
## [232] 0.9211789 0.9219569 0.9227323 0.9235057 0.9242782 0.9250485 0.9258159
## [239] 0.9265802 0.9273414 0.9280995 0.9288558 0.9296106 0.9303629 0.9311136
## [246] 0.9318635 0.9326109 0.9333559 0.9340995 0.9348407 0.9355786 0.9363154
## [253] 0.9370500 0.9377835 0.9385126 0.9392404 0.9399667 0.9406902 0.9414133
## [260] 0.9421333 0.9428521 0.9435686 0.9442828 0.9449945 0.9457044 0.9464137
## [267] 0.9471212 0.9478253 0.9485272 0.9492269 0.9499234 0.9506182 0.9513122
## [274] 0.9520029 0.9526905 0.9533763 0.9540604 0.9547421 0.9554221 0.9560989
## [281] 0.9567746 0.9574478 0.9581191 0.9587879 0.9594546 0.9601208 0.9607840
## [288] 0.9614446 0.9621016 0.9627584 0.9634105 0.9640605 0.9647078 0.9653539
## [295] 0.9659963 0.9666380 0.9672775 0.9679136 0.9685474 0.9691804 0.9698132
## [302] 0.9704412 0.9710666 0.9716886 0.9723100 0.9729284 0.9735459 0.9741599
## [309] 0.9747712 0.9753792 0.9759859 0.9765903 0.9771932 0.9777953 0.9783945
## [316] 0.9789891 0.9795778 0.9801634 0.9807467 0.9813265 0.9819049 0.9824775
## [323] 0.9830476 0.9836126 0.9841763 0.9847345 0.9852906 0.9858406 0.9863778
## [330] 0.9869134 0.9874392 0.9879519 0.9884584 0.9889594 0.9894540 0.9899389
## [337] 0.9904177 0.9908951 0.9913702 0.9918421 0.9923094 0.9927738 0.9932313
## [344] 0.9936858 0.9941333 0.9945773 0.9950180 0.9954583 0.9958930 0.9963227
## [351] 0.9967454 0.9971662 0.9975843 0.9979992 0.9984080 0.9988122 0.9992154
## [358] 0.9996145 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] vegetable vegetable clothing  furniture building  tool      kitchen
##  [8] kitchen   buildpart building  vegetable kitchen   furniture vehicle
## [15] clothing  insect    manmade   furniture kitchen   kitchen   insect
## [22] tool      vegetable buildpart clothing  tool      animal    furniture
## [29] manmade   clothing  clothing  vegetable manmade   manmade   kitchen
## [36] clothing  vehicle   manmade   clothing  manmade   manmade   manmade
## [43] tool      vehicle   vegetable manmade   tool      building  vegetable
## [50] clothing  manmade   buildpart kitchen   vegetable clothing  buildpart
## [57] insect    vegetable bodypart  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         0        1         1      0
##     bodypart       0        0        1         0        0         0      1
##     building       0        0        0         0        0         1      1
##     buildpart      0        0        0         1        2         1      0
##     clothing       0        0        2         0        2         0      0
##     furniture      0        0        0         0        1         0      0
##     insect         1        0        0         1        0         0      1
##     kitchen        0        0        0         0        1         0      0
##     manmade        0        0        0         0        0         1      0
##     tool           0        1        0         2        0         0      0
##     vegetable      0        0        0         0        2         0      0
##     vehicle        0        0        0         1        0         0      0
##              Predicted_Values
## Actual_Values kitchen manmade tool vegetable vehicle
##     animal          0       0    1         2       0
##     bodypart        0       2    0         0       1
##     building        0       1    0         1       1
```

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

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

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

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

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

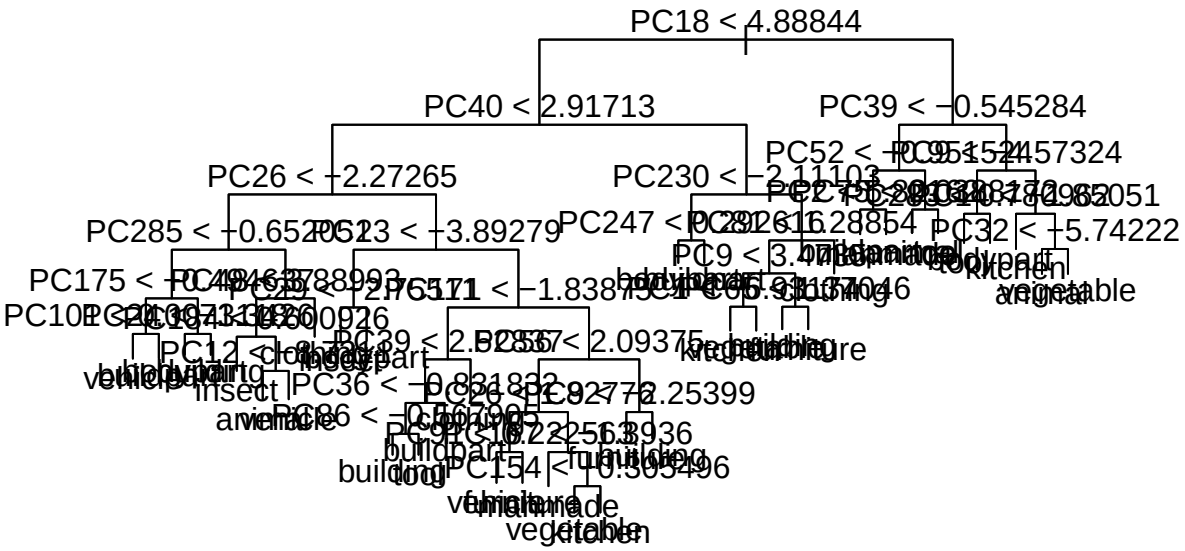
```
## [1] 0.1166667
```

```
# 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"  "PC40"  "PC26"  "PC285" "PC175" "PC101" "PC24"  "PC48"  "PC184"
## [10] "PC12"  "PC23"  "PC29"  "PC171" "PC39"  "PC36"  "PC86"  "PC56"  "PC91"
## [19] "PC167" "PC154" "PC9"   "PC230" "PC247" "PC81"  "PC1"   "PC65"  "PC52"
## [28] "PC2"   "PC75"  "PC283" "PC14"  "PC32"
## Number of terminal nodes:  37
## Residual mean deviance:  1.978 = 520.3 / 263
## Misclassification error rate: 0.3833 = 115 / 300
```

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




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

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

```
## [50] buildpart furniture vegetable vehicle   building  vegetable manmade
## [57] insect    insect    bodypart  clothing
## 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        1         0        0         1      1
##     bodypart       0        1        0         1        0         0      1
##     building       0        1        1         1        1         0      1
##     buildpart      0        0        0         0        0         0      1
##     clothing       0        1        2         0        1         0      0
##     furniture      0        1        0         0        0         1      1
##     insect         1        1        0         0        1         0      1
##     kitchen        0        0        0         1        1         1      1
##     manmade        0        0        1         0        0         1      1
##     tool           1        1        0         1        0         0      0
##     vegetable      0        0        1         0        1         1      1
##     vehicle        0        0        0         1        1         1      1
##              tree.pred
## pcs.test.labs kitchen manmade tool vegetable vehicle
##     animal          1       0    0         1       0
##     bodypart        1       0    0         1       0
##     building        0       0    0         0       0
##     buildpart       1       1    0         2       0
##     clothing        0       0    0         1       0
##     furniture       0       0    0         1       1
##     insect          0       0    0         1       0
##     kitchen         0       1    0         0       0
##     manmade         0       0    0         1       1
##     tool            0       0    0         2       0
##     vegetable       0       0    1         0       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         4        0        0         0        0         0      0
##     bodypart       1        2        1         0        0         1      0
##     building       0        0        1         2        0         1      0
##     buildpart      0        1        2         1        0         0      0
##     clothing       0        0        1         0        1         0      2
##     furniture      0        0        0         2        0         2      0
##     insect         3        1        0         0        0         0      1
##     kitchen        0        0        0         0        3         1      0
##     manmade        0        0        0         0        0         1      0
##     tool           1        0        0         1        0         0      0
##     vegetable      0        0        2         0        0         1      0
##     vehicle        0        0        0         1        0         1      1
##              rf.pred
## pcs.test.labs kitchen manmade tool vegetable vehicle
##     animal          0       0    0         1       0
##     bodypart        0       0    0         0       0
##     building        1       0    0         0       0
##     buildpart       0       0    1         0       0
##     clothing        0       1    0         0       0
##     furniture       1       0    0         0       0
##     insect          0       0    0         0       0
##     kitchen         0       0    1         0       0
##     manmade         1       0    1         1       1
##     tool            0       0    2         0       1
##     vegetable       0       0    0         2       0
##     vehicle         1       0    0         0       1
```

```
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 natural manmade manmade manmade manmade manmade manmade manmade
## [10] manmade manmade manmade manmade manmade manmade manmade manmade manmade
## [19] manmade manmade manmade manmade natural natural manmade manmade natural
## [28] manmade manmade manmade manmade natural manmade manmade manmade manmade
## [37] natural natural manmade manmade manmade manmade manmade manmade 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      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      27      13
##       natural      12       8
```

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

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

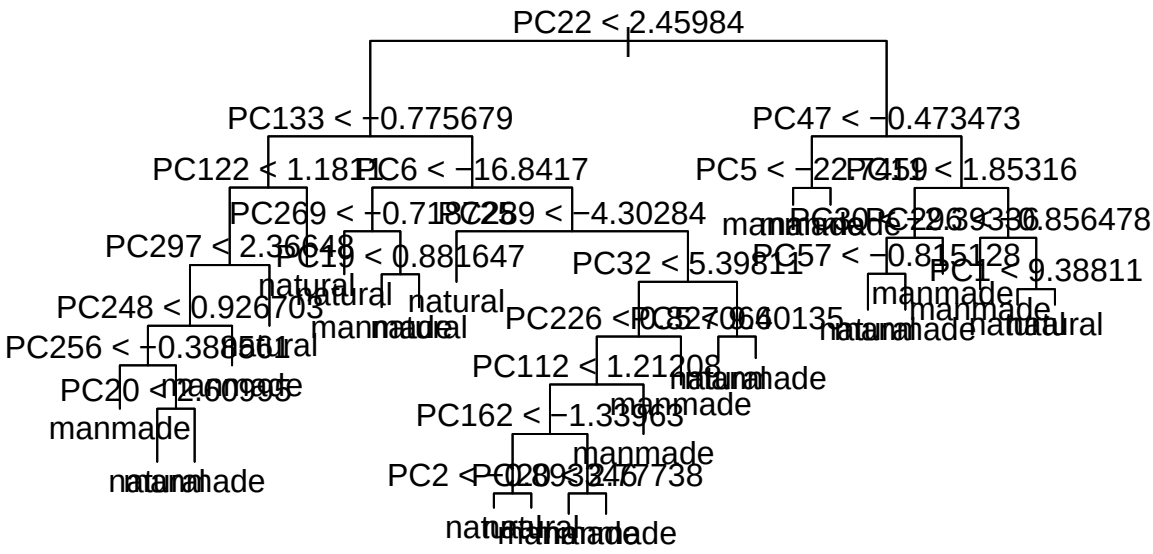
```r
# Decision Trees
set.seed(100)
tree.fit <- tree(as.factor(cls) ~ ., data = pcs.train)
```

```
## 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] "PC22"  "PC133" "PC122" "PC297" "PC248" "PC256" "PC20"  "PC6"   "PC269"
## [10] "PC19"  "PC289" "PC32"  "PC226" "PC112" "PC162" "PC2"   "PC5"   "PC47"
## [19] "PC11"  "PC30"  "PC57"  "PC296" "PC1"
## Number of terminal nodes:  26
## Residual mean deviance:  0.2282 = 62.54 / 274
## Misclassification error rate: 0.04667 = 14 / 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] manmade natural natural manmade natural manmade natural manmade manmade
## [10] natural manmade manmade manmade manmade manmade manmade natural manmade
## [19] manmade natural natural natural natural manmade manmade natural manmade
## [28] natural manmade natural manmade natural manmade natural manmade manmade
## [37] natural manmade manmade manmade manmade natural manmade manmade natural
## [46] manmade natural natural manmade manmade manmade manmade natural manmade
## [55] manmade manmade manmade 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      27      13
##       natural      10      10
```

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

```
## [1] 0.6166667
```

```
# 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
```