Patient5

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-P5.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] 0.2783881 0.3843781 0.4413599 0.4853602 0.5257814 0.5620984 0.5880783
```

```
##   [8] 0.6124448 0.6343115 0.6497491 0.6624918 0.6731251 0.6831103 0.6920812
##  [15] 0.7005098 0.7077374 0.7147431 0.7209366 0.7265718 0.7319013 0.7371756
##  [22] 0.7418888 0.7463349 0.7505778 0.7544464 0.7582190 0.7617867 0.7650833
##  [29] 0.7683080 0.7714570 0.7744618 0.7773108 0.7800957 0.7828020 0.7854524
##  [36] 0.7879426 0.7903696 0.7927771 0.7950932 0.7972761 0.7994128 0.8014850
##  [43] 0.8035331 0.8054904 0.8074197 0.8092795 0.8110901 0.8128883 0.8146652
##  [50] 0.8164356 0.8181479 0.8198407 0.8214718 0.8230712 0.8246437 0.8262078
##  [57] 0.8277566 0.8292743 0.8307538 0.8321945 0.8336038 0.8350077 0.8363701
##  [64] 0.8377160 0.8390417 0.8403407 0.8416090 0.8428569 0.8440905 0.8453058
##  [71] 0.8464923 0.8476647 0.8488199 0.8499655 0.8510936 0.8522138 0.8533105
##  [78] 0.8543917 0.8554607 0.8565224 0.8575560 0.8585721 0.8595826 0.8605887
##  [85] 0.8615882 0.8625664 0.8635403 0.8645016 0.8654543 0.8664006 0.8673403
##  [92] 0.8682669 0.8691807 0.8700893 0.8709873 0.8718799 0.8727530 0.8736199
##  [99] 0.8744808 0.8753354 0.8761820 0.8770143 0.8778445 0.8786708 0.8794865
## [106] 0.8802907 0.8810905 0.8818806 0.8826635 0.8834418 0.8842175 0.8849918
## [113] 0.8857590 0.8865209 0.8872777 0.8880269 0.8887677 0.8895049 0.8902416
## [120] 0.8909695 0.8916950 0.8924191 0.8931354 0.8938465 0.8945513 0.8952540
## [127] 0.8959545 0.8966516 0.8973423 0.8980264 0.8987090 0.8993875 0.9000599
## [134] 0.9007289 0.9013943 0.9020591 0.9027182 0.9033762 0.9040298 0.9046799
## [141] 0.9053286 0.9059711 0.9066114 0.9072479 0.9078823 0.9085122 0.9091395
## [148] 0.9097651 0.9103890 0.9110098 0.9116250 0.9122375 0.9128478 0.9134551
## [155] 0.9140592 0.9146560 0.9152511 0.9158441 0.9164353 0.9170228 0.9176085
## [162] 0.9181911 0.9187733 0.9193538 0.9199314 0.9205034 0.9210751 0.9216461
## [169] 0.9222149 0.9227831 0.9233500 0.9239122 0.9244739 0.9250338 0.9255881
## [176] 0.9261419 0.9266937 0.9272437 0.9277931 0.9283394 0.9288823 0.9294249
## [183] 0.9299644 0.9305033 0.9310397 0.9315756 0.9321054 0.9326341 0.9331605
## [190] 0.9336848 0.9342065 0.9347274 0.9352470 0.9357643 0.9362810 0.9367936
## [197] 0.9373062 0.9378172 0.9383258 0.9388327 0.9393383 0.9398408 0.9403404
## [204] 0.9408389 0.9413362 0.9418323 0.9423271 0.9428215 0.9433150 0.9438047
## [211] 0.9442930 0.9447795 0.9452656 0.9457496 0.9462315 0.9467117 0.9471903
## [218] 0.9476671 0.9481421 0.9486158 0.9490890 0.9495605 0.9500298 0.9504975
## [225] 0.9509645 0.9514293 0.9518940 0.9523573 0.9528183 0.9532780 0.9537361
## [232] 0.9541925 0.9546484 0.9551020 0.9555541 0.9560058 0.9564566 0.9569047
## [239] 0.9573514 0.9577956 0.9582397 0.9586829 0.9591244 0.9595636 0.9600015
## [246] 0.9604377 0.9608733 0.9613070 0.9617396 0.9621707 0.9626015 0.9630308
## [253] 0.9634573 0.9638830 0.9643064 0.9647283 0.9651490 0.9655679 0.9659864
## [260] 0.9664024 0.9668171 0.9672301 0.9676425 0.9680540 0.9684630 0.9688707
## [267] 0.9692776 0.9696836 0.9700887 0.9704922 0.9708949 0.9712969 0.9716958
## [274] 0.9720937 0.9724908 0.9728857 0.9732798 0.9736729 0.9740643 0.9744549
## [281] 0.9748453 0.9752331 0.9756198 0.9760050 0.9763901 0.9767733 0.9771541
## [288] 0.9775337 0.9779116 0.9782883 0.9786634 0.9790374 0.9794109 0.9797824
## [295] 0.9801537 0.9805238 0.9808927 0.9812598 0.9816260 0.9819905 0.9823533
## [302] 0.9827150 0.9830761 0.9834365 0.9837958 0.9841510 0.9845054 0.9848590
## [309] 0.9852109 0.9855622 0.9859106 0.9862586 0.9866059 0.9869492 0.9872924
## [316] 0.9876328 0.9879726 0.9883100 0.9886464 0.9889818 0.9893141 0.9896459
## [323] 0.9899760 0.9903051 0.9906310 0.9909532 0.9912725 0.9915908 0.9919067
## [330] 0.9922181 0.9925278 0.9928360 0.9931417 0.9934454 0.9937446 0.9940397
## [337] 0.9943300 0.9946188 0.9949056 0.9951860 0.9954643 0.9957391 0.9960077
## [344] 0.9962746 0.9965397 0.9968032 0.9970646 0.9973225 0.9975794 0.9978340
## [351] 0.9980882 0.9983388 0.9985837 0.9988278 0.9990684 0.9993069 0.9995424
## [358] 0.9997733 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] buildpart furniture buildpart building  building  vehicle   buildpart
##  [8] tool      clothing  building  vegetable vehicle   animal    animal
## [15] furniture tool      animal    building  buildpart animal    clothing
## [22] furniture tool      buildpart buildpart buildpart vegetable insect
## [29] vegetable buildpart animal    manmade   tool      kitchen   animal
## [36] insect    vegetable buildpart insect    insect    building  buildpart
## [43] vehicle   buildpart building  animal    insect    insect    buildpart
## [50] buildpart insect    insect    vehicle   furniture building  manmade
## [57] buildpart clothing  clothing  tool
## 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        1         0      2
##     bodypart       1        0        0         2        0         0      1
##     building       2        0        1         2        0         0      0
##     buildpart      0        0        2         1        0         0      1
##     clothing       1        0        2         1        0         0      1
##     furniture      2        0        0         0        0         1      0
##     insect         0        0        0         1        2         0      0
##     kitchen        0        0        0         1        0         1      0
##     manmade        0        0        1         0        0         2      1
##     tool           0        0        1         1        1         0      2
##     vegetable      0        0        0         4        0         0      0
##     vehicle        1        0        0         0        0         0      0
##              Predicted_Values
## Actual_Values kitchen manmade tool vegetable vehicle
##     animal          0       0    0         1       0
##     bodypart        0       0    1         0       0
##     building        0       0    0         0       0
```

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

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

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

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

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

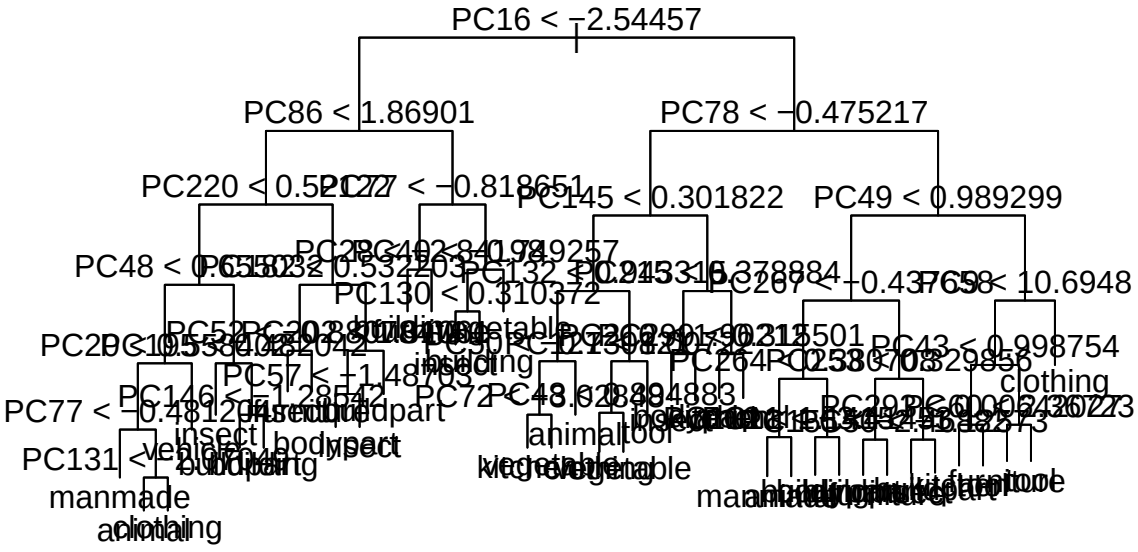
```
## [1] 0.1166667
```

```
# 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] "PC16"  "PC86"  "PC220" "PC48"  "PC20"  "PC77"  "PC131" "PC195" "PC146"
## [10] "PC182" "PC52"  "PC202" "PC57"  "PC28"  "PC40"  "PC130" "PC78"  "PC145"
## [19] "PC132" "PC50"  "PC72"  "PC127" "PC215" "PC210" "PC299" "PC49"  "PC267"
## [28] "PC264" "PC112" "PC11"  "PC253" "PC155" "PC9"   "PC43"  "PC293" "PC60"
## Number of terminal nodes:  40
## Residual mean deviance:  2.003 = 520.7 / 260
## Misclassification error rate: 0.41 = 123 / 300
```

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




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

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

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

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

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

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

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

```
## [1] 0.15
```

```
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 natural manmade manmade manmade manmade manmade manmade manmade
## [19] manmade manmade manmade manmade manmade manmade manmade manmade natural
## [28] manmade manmade manmade manmade manmade manmade manmade manmade manmade
## [37] natural manmade manmade natural manmade manmade manmade manmade manmade
## [46] natural manmade manmade manmade manmade natural natural 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      37       3
##       natural      15       5
```

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

```
## [1] 0.7
```

```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      19       1
```

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

```
## [1] 0.4666667
```

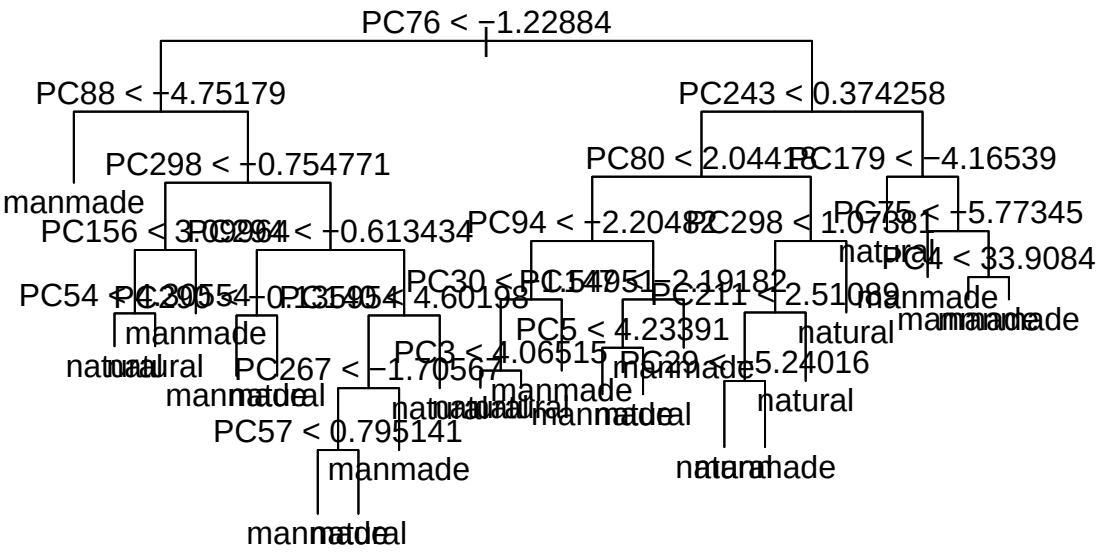
```
# Decision Trees
set.seed(100)
tree.fit <- tree(as.factor(cls) ~ ., data = pcs.train)
```

```
## 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] "PC76"  "PC88"  "PC298" "PC156" "PC54"  "PC294" "PC290" "PC140" "PC267"
## [10] "PC57"  "PC243" "PC80"  "PC94"  "PC30"  "PC3"   "PC147" "PC5"   "PC211"
## [19] "PC29"  "PC179" "PC75"  "PC4"
## Number of terminal nodes:  24
## Residual mean deviance:  0.2388 = 65.91 / 276
## Misclassification error rate: 0.05667 = 17 / 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 manmade natural manmade manmade natural manmade
## [10] manmade manmade manmade manmade manmade manmade natural manmade natural
## [19] manmade manmade manmade manmade manmade manmade natural manmade natural
## [28] natural manmade manmade manmade natural natural manmade natural manmade
## [37] manmade manmade manmade manmade natural manmade manmade manmade manmade
## [46] natural natural manmade manmade manmade natural manmade manmade manmade
## [55] manmade natural natural manmade manmade 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      30      10
##       natural      13       7
```

```
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
```