Classification

Muhammad Shariq Azeem and Aarushi Pandey

10/19/2022

Classification:

Data:

For this assignment, I selected a data set that contains information about the room environment, such as, room temperature, humidity, CO2 levels, etc. I need to use that information to decide if the room is occupied or not.

Source for the data: https://archive.ics.uci.edu/ml/datasets/Occupancy+Detection+#

Note: The data sets provided through the link were not meeting the "at least 10K rows" requirement, so I used Excel to combine two data sets into one.

Cleaning the data:

- Got rid of the date column because I don't need that for my model.
- Converted 'Occupancy' attribute to a factor.

```
df <- read.csv("C:/Users/shari/Downloads/data.csv", header=T)
df <- df[,c(2,3,4,5,6,7)]
df$Occupancy <- factor(df$Occupancy)
```

Step A: Divide data into train and test

```
set.seed(1234)
i <- sample(1:nrow(df), 0.80*nrow(df), replace=FALSE)
train <- df[i,]
test <- df[-i,]

knn.train <- train[,c(1,2,3,4,5)]
knn.test <- test[,c(1,2,3,4,5)]
knn.trainLabels <- train$Occupancy
knn.testLabels <- test$Occupancy
```

Step B: Statistical and Graphical Exploration on the training data

str():

str() tells us what type of data is stored in the table. In our case, training data consists of 5 number attributes and 1 attribute with two factors, 0 and 1.

```
str(train)
```

```
## 'data.frame':    9933 obs. of  6 variables:
##  $ Temperature  : num  20.9 21.4 22.6 21.8 21 ...
##  $ Humidity     : num  24.7 27.8 24.9 28.1 25.4 ...
##  $ Light        : num  0 0 732 0 14 0 0 454 433 0 ...
##  $ CO2          : num  572 566 588 594 522 ...
##  $ HumidityRatio: num  0.00377 0.00438 0.00423 0.00453 0.0039 ...
##  $ Occupancy    : Factor w/ 2 levels "0","1": 1 1 2 1 1 1 1 2 2 1 ...
```

summary():

summary() gives us general statistics about the data. In our case, it tells us the minimum value, median/mean value, and maximum value of our quantitative attributes, and the counts of each factor of our qualitative attribute.

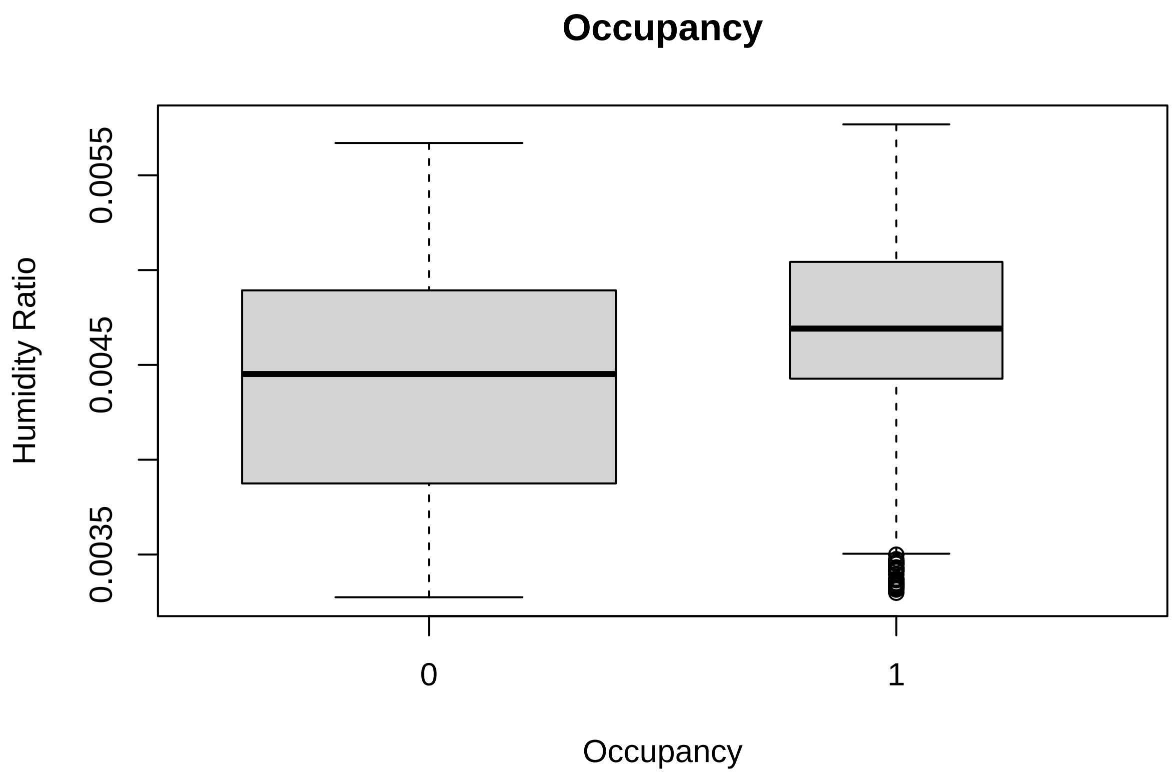
```
summary(train)
```

```
##   Temperature       Humidity         Light              CO2
##  Min.   :19.50   Min.   :21.86   Min.   :   0.0   Min.   : 427.5
##  1st Qu.:20.39   1st Qu.:25.39   1st Qu.:   0.0   1st Qu.: 528.0
##  Median :20.79   Median :28.63   Median :   0.0   Median : 632.7
##  Mean   :21.09   Mean   :28.92   Mean   : 138.1   Mean   : 745.6
##  3rd Qu.:21.67   3rd Qu.:31.86   3rd Qu.: 399.0   3rd Qu.: 857.0
##  Max.   :24.41   Max.   :39.50   Max.   :1581.0   Max.   :2076.5
##  HumidityRatio      Occupancy
##  Min.   :0.003275   0:7512
##  1st Qu.:0.003984   1:2421
##  Median :0.004512
##  Mean   :0.004468
##  3rd Qu.:0.004940
##  Max.   :0.005769
```

Box Plot:

The Box Plot below shows us that the Humidity Ratio increases, as the room goes from being empty to being occupied.

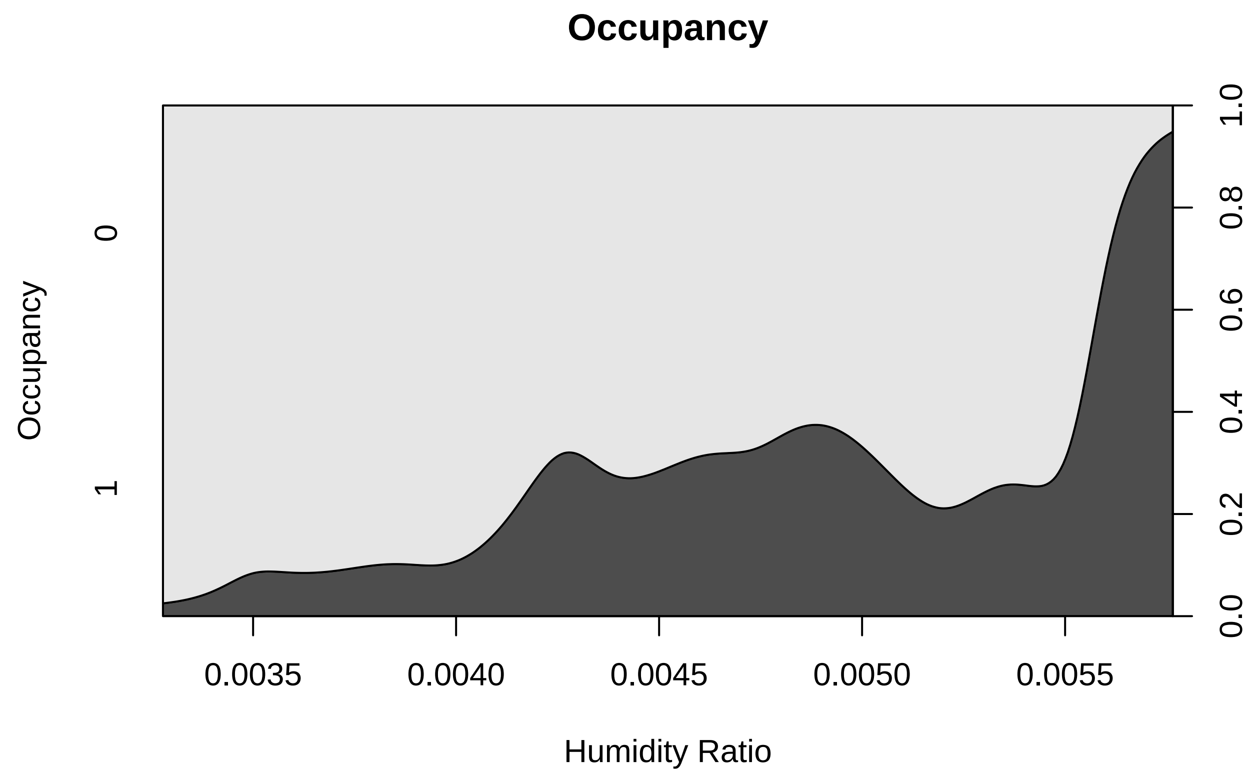
```
plot(HumidityRatio~Occupancy, data=train, main="Occupancy",
     xlab = "Occupancy", ylab = "Humidity Ratio", varwidth = TRUE)
```



CD Plot:

The conditional density (CD) plot below tells us the same thing as the box plot above, but it is just visualized differently. As the humidity ratio increases, there is more chance of the room being occupied.

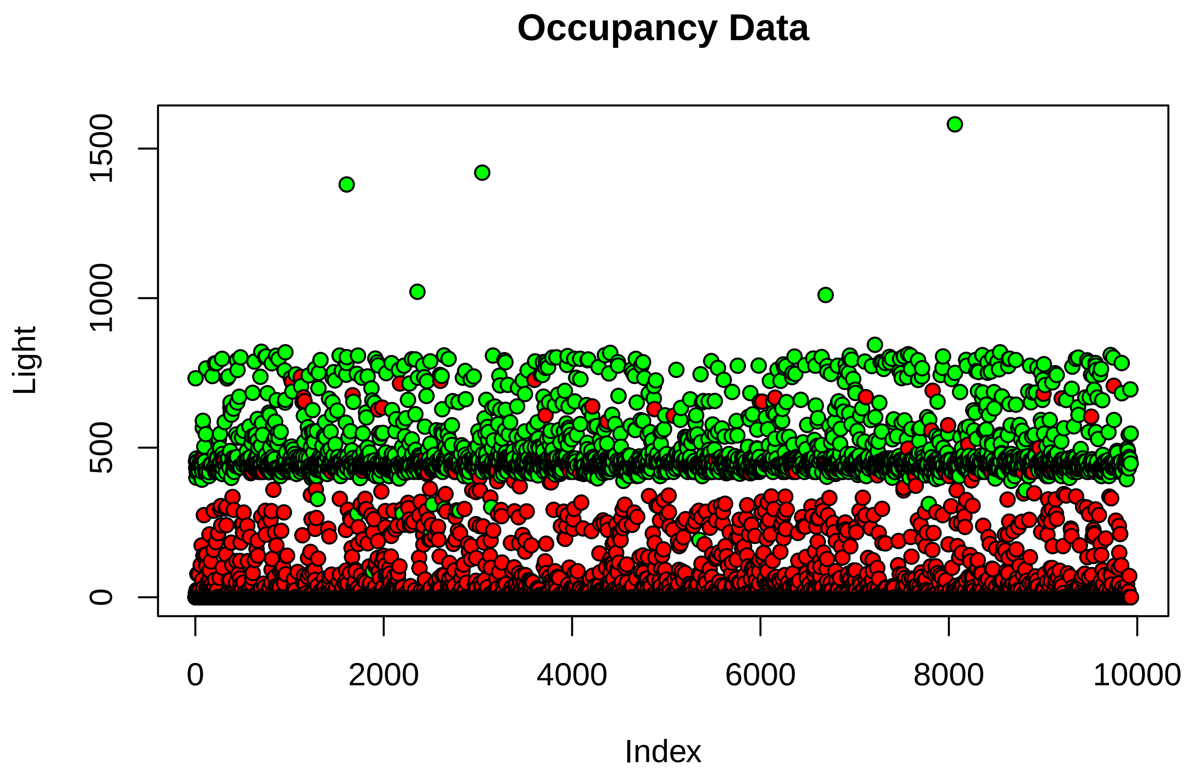
```
cdplot(train$Occupancy~train$HumidityRatio, main="Occupancy",
       ylab = "Occupancy", xlab = "Humidity Ratio")
```



Plot:

The plot below shows us that Light is a great predictor for predicting if a room is occupied or not because we can see a pretty good separation between empty rooms and occupied rooms, based on their Light value.

```
plot(train$Light, pch=21, bg=c("red","green")
     [unclass(train$Occupancy)], main="Occupancy Data", ylab = "Light")
```



Step C: Perform SVM Classification

Try a linear kernal with cost = 0.1

```
library(e1071)
library(MASS)

svm1 <- svm(Occupancy~., data=train, kernel="linear", cost=0.1, scale=TRUE)
pred1 <- predict(svm1, newdata=test)

acc1 <- mean(pred1==test$Occupancy)
table(pred1, test$Occupancy)
```

```
##
## pred1    0    1
##     0 1870    1
##     1   14  599
```

```
print(paste("Accuracy of linear kernal with cost = 0.1:", acc1))
```

```
## [1] "Accuracy of linear kernal with cost = 0.1: 0.993961352657005"
```

Try a linear kernal with cost = 1

```
svm2 <- svm(Occupancy~., data=train, kernel="linear", cost=1, scale=TRUE)
pred2 <- predict(svm2, newdata=test)

acc2 <- mean(pred2==test$Occupancy)
table(pred2, test$Occupancy)
```

```
##
## pred2    0    1
##     0 1871    1
##     1   13  599
```

```
print(paste("Accuracy of linear kernal with cost = 1:", acc2))
```

```
## [1] "Accuracy of linear kernal with cost = 1: 0.994363929146538"
```

Try a polynomial kernel with cost = 0.01

```
svm3 <- svm(Occupancy~., data=train, kernel="polynomial", cost=0.01, scale=TRUE)
pred3 <- predict(svm3, newdata=test)

acc3 <- mean(pred3==test$Occupancy)
table(pred3, test$Occupancy)
```

```
##
## pred3    0    1
##     0 1873   37
##     1   11  563
```

```
print(paste("Accuracy of polynomial kernal with cost = 0.01:", acc3))
```

```
## [1] "Accuracy of polynomial kernal with cost = 0.01: 0.980676328502415"
```

Try a polynomial kernel with cost = 0.1

```
svm4 <- svm(Occupancy~., data=train, kernel="polynomial", cost=0.1, scale=TRUE)
pred4 <- predict(svm4, newdata=test)

acc4 <- mean(pred4==test$Occupancy)
table(pred4, test$Occupancy)
```

```
##
## pred4    0    1
##     0 1872    2
##     1   12  598
```

```r
print(paste("Accuracy of polynomial kernal with cost = 0.1:", acc4))
```

```
## [1] "Accuracy of polynomial kernal with cost = 0.1: 0.994363929146538"
```

Try a radial kernel with cost = 10 and gamma = 1

```r
svm5 <- svm(Occupancy~., data=train, kernel="radial", cost=10, gamma=1, scale=TRUE)
pred5 <- predict(svm5, newdata=test)

acc5 <- mean(pred5==test$Occupancy)
table(pred5, test$Occupancy)
```

```
##
## pred5    0    1
##     0 1872    6
##     1   12  594
```

```r
print(paste("Accuracy of radial kernal with cost = 10 and gamma = 1", acc5))
```

```
## [1] "Accuracy of radial kernal with cost = 10 and gamma = 1 0.992753623188406"
```

Try a radial kernel with cost = 10 and gamma = .1

```r
svm6 <- svm(Occupancy~., data=train, kernel="radial", cost=10, gamma=.1, scale=TRUE)
pred6 <- predict(svm6, newdata=test)

acc6 <- mean(pred6==test$Occupancy)
table(pred6, test$Occupancy)
```

```
##
## pred6    0    1
##     0 1871    1
##     1   13  599
```

```r
print(paste("Accuracy of radial kernel with cost = 10 and gamma = .1:", acc6))
```

```
## [1] "Accuracy of radial kernel with cost = 10 and gamma = .1: 0.994363929146538"
```

Analyze the results

First, with a linear kernel, when I set the cost hyperparameter to 0.1, I got an accuracy of 0.9939, but when I increased the cost by 10 times and set it to 1, I got an accuracy of 0.9943. That is an increase of 0.04%. Even though it was a small increment, the reason the accuracy increased is that by increasing the cost hyperparameter, I allowed for more slack variables, and that kept the training data from overfitting.

Second, for the polynomial kernel, when I set the cost hyperparameter to 0.01, I got an accuracy of 0.9807, but when I increased the cost by 10 times and set it to 0.1, I got an accuracy of 0.9943. That is an increase of 1.36%. The reason the accuracy increased is probably because of a large cost value, similar to the linear kernel.

Lastly, for the radial kernel, I kept the cost hyperparameter same for both models. When I set the gamma hyperparameter to 1, I got an accuracy of 0.9927, but when I reduced the gamma by 10 times and set it to 0.1, I got an accuracy of 0.9943. That is an increase of 0.16%. The reason the accuracy increased is that by decreasing the gamma hyperparameter, I kept the training data from overfitting which resulted in better performance on the test data.