Comparing Classifiers for Predicting Room Occupancy

Vishal Saikrishnan
vsaikris

Due Wed, Dec 2, at 8:00PM (Pittsburgh time)

Contents

Introduction **1**

Exploratory Data Analysis **2**
Variables . . . 2
Relationships between Variables . . . 2
Classification Pairs . . . 4

Modeling **5**
Binary Logistic Regression . . . 5
Linear Discriminant Analysis (LDA) . . . 5
Quadratic Discriminant Analysis (QDA) . . . 6
Classification Trees . . . 6
Final Prediction . . . 7

Discussion **8**

```
set.seed(151)
library("knitr")
library("kableExtra")
library("pander")
library("readr")
library("magrittr")
library("car")
library("MASS")
library("klaR")
library("tree")
library("rpart")
library("rpart.plot")
```

```
occupancy_train <- readr::read_csv("http://stat.cmu.edu/~gordonw/occupancy_train.csv")

occupancy_test <- readr::read_csv("http://stat.cmu.edu/~gordonw/occupancy_test.csv")
```

Introduction

All around the world, we try to prepare for disasters. For example, every building has a fire escape or a route that all people know of to evacuate the building. In the case of an emergency, everything is in chaos. People are all over the place and there's only a limited amount of security and first responders to help. We know how many people are in the building but where they are is completely guesswork. Security cameras are the only solution but usually no one is watching them inside of a burning building (the security officer has probably already evacuated.)

In this paper, we will try and use certain classifiers to predict if a room is occupied or empty. This way, during emergencies, we can identify which rooms/floors have people and then decide the best pathway for them to evacuate. [Data from: Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models. Luis M. Candanedo, Véronique Feldheim. Energy and Buildings. Volume 112, 15 January 2016, Pages 28-39.]

Exploratory Data Analysis

Variables

Here are the available predictor variables:

- **Temperature** – Room Temperatture in Celsius
- **Humidity** – Room relative humidity, in percent
- **CO2** – Room's Carbon dioxide in ppm
- **Hour** – Hour of the day (in military time 0-23)

The response variable is:

- **Occupancy** – Binary, 0 is not occupied, 1 is occupied

```
table(occupancy_train$Occupancy)
```

```
##
##    0    1
## 4497 1203
```

```
prop.table(table(occupancy_train$Occupancy))
```

```
##
##         0         1
## 0.7889474 0.2110526
```

Summary of Response variable *in the training data*:

We see that there are 5700 observations in the training data. 4,497 of these observations have Occupancy 0, which means that the room is *Not occupied*. The corresponding proportion is 0.789. On the other hand, 1,203 of the observations have Occupancy 1, which means that they are *Occupied*. The corresponding proportion is 0.211.

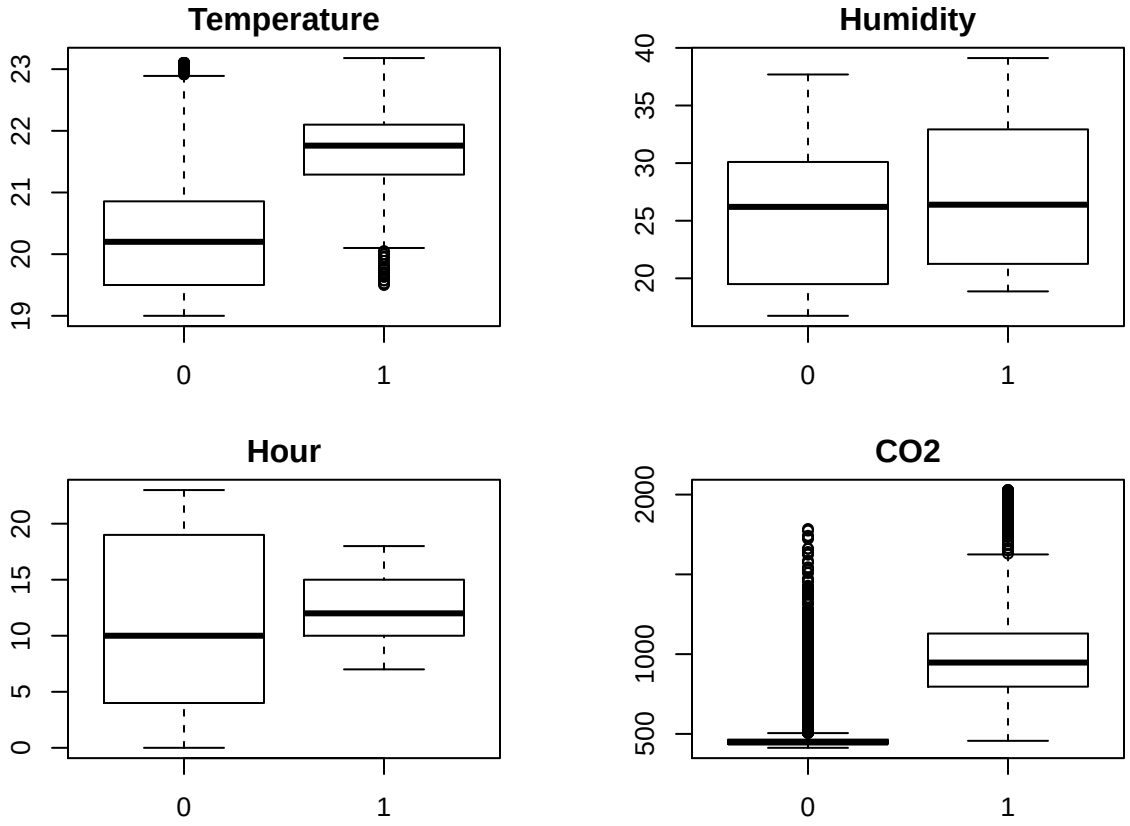
Relationships between Variables

We will now explore the relationships between the response variable **Occupancy** and each predictor variable. Since all of our predictor variables are quantitative, we can use boxplots:

```
par(mfrow = c(2,2),
    mai = c(0.5, 0.5, 0.3, 0.5))

boxplot(Temperature ~ Occupancy,
        main = "Temperature",
        data = occupancy_train)
boxplot(Humidity ~ Occupancy,
```

```
        main = "Humidity",
        data = occupancy_train)
boxplot(Hour ~ Occupancy,
        main = "Hour",
        data = occupancy_train)
boxplot(CO2 ~ Occupancy,
        main = "CO2",
        data = occupancy_train)
```



Looking at the boxplots, we see that there is some evidence of a noticeable relationship between some of the predictor variables and **Occupancy**.

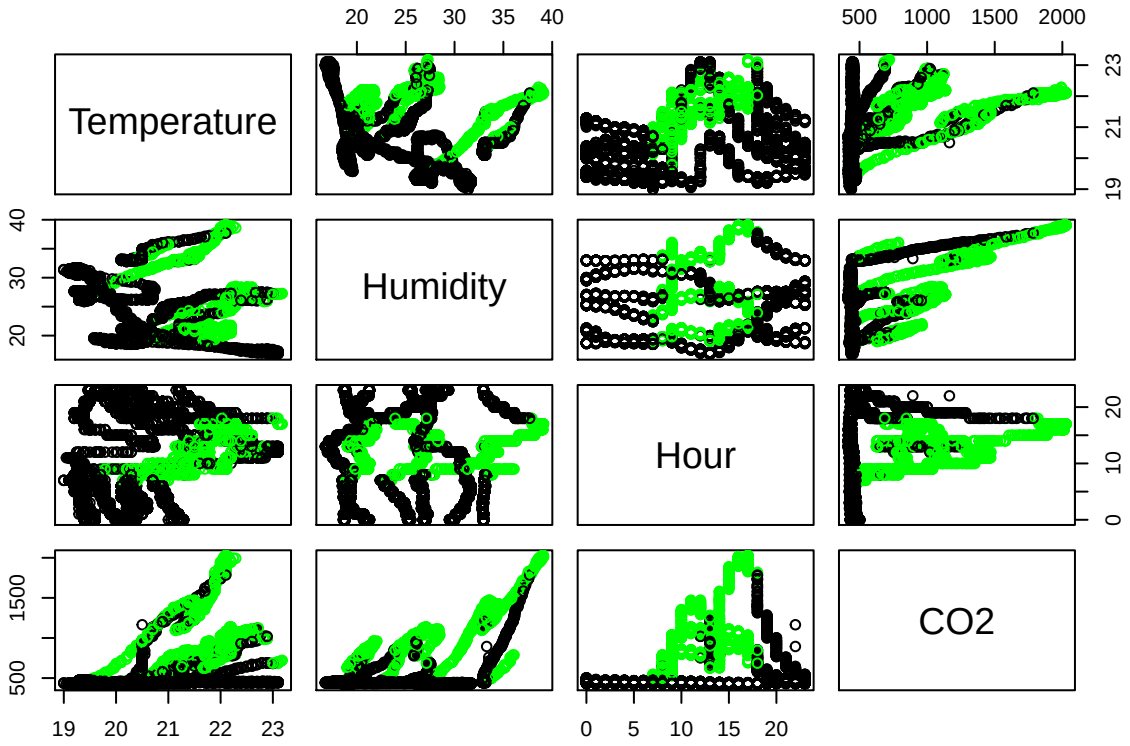
- Specifically, there appears to be a relationship between **CO2** and **Occupancy**. This makes sense because no one will be in a room with low or no Oxygen.
- **Tempearture** and **Occupancy** seem to be related. The higher the temperature, the more likely the room is occupied. This is indicated by the median of 22 C for *Occupied*, while *Not occupied* has a median of 20.2 C. The IQRs do NOT overlap, which provides further evidence that there is a relationship between these two variables. Again, this makes sense since people would rather occupy warmer rooms than colder rooms due to personal comfort.
- For **Hour** and **Occupancy**, the medians seem to be pretty similar, but the IQRs are extremely different. For *Occupied*, the IQR and range are both "squished" and much smaller than for *Not Occupied*. This makes sense because most people work around the same time and shifts only last in between 8 AM and 18 (8 PM).
- Finally, for **Humidity**, there appears to not be a significant relationship with **Occupancy**. The medians, IQR. and ranges are almost the same. However, the IQR is slightly higher for *Occupied* (22-33)

compared to 20-30 for *Not occupied*.

Classification Pairs

Now that we have observed the singular relationships between each predictor variable and the response variable, let's look at bivariate plots. These pair plots compare two predictor variables and plot the responses. This helps us understand which variables would have the easiest boundaries to calculate.

```
pairs(occupancy_train[ , c(1,2,3,4)],
      col=ifelse(occupancy_train$Occupancy == 1, "green", "black"))
```



In the pairs plot, if **Occupancy** is 1, the point is green, and if **Occupancy** is 0, the point is black.

This is a good visual to see which two variables could be used in clearly creating boundaries between *Occupied* and *Not Occupied*. First, we will go over the variables that don't show any reasonable boundaries: **Humidity** & **Temperature** and **Humidity** & **Hour** don't show any clear separation. The main problem is that the *Not Occupied* points are all over the place. Now, we look for variables that DO show some reasonable separation. For the most part, **CO2** seems to be the best variable with regards to creating boundaries. We will keep this in mind, as CO2 may turn out to be an important variable for our classifier models. Also, **Temperature** & **Hour** seem to have a good separation with the *Occupied* points being clumped in the middle.

Modeling

To build our model, we have four options of classifiers: Binary Logistic Regression, Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), and Classification trees. We will build all 4 of these models and then see which one turns out the best. After building our models, we will test it on the occupancy_test data, which was randomly split from the full data. So far, we have only been using the occupancy_train data, and we will use this data to create the models as well.

Binary Logistic Regression

First, lets consider a binary logistic regression. The logistic model gives us probabilities, but if we want to predict **Occupancy**, we need to choose a threshold probability. If the p>threshold, then it is classified as either *Occupied* or *Not Occupied*. So before we fit the model, let's see what R classifies as the default **Occupancy**. This default will be classified as the one where probability is less than the threshold we choose.

```
levels(factor(occupancy_test$Occupancy))
```

```
## [1] "0" "1"
```

In this case, "0" or *Not Occupied* is taken as the default **Occupancy**. So when the p < threshold, the model is predicting the room to be *Not Occupied*.

```
occupancy.log <- glm(factor(Occupancy) ~  Temperature + Humidity + Hour +  CO2,
                     data = occupancy_train,
                     family = binomial(link = "logit"))
occupancy.log.prob <- predict(occupancy.log,
                              as.data.frame(occupancy_test),
                              type  = "response")
occupancy.log.pred <- ifelse(occupancy.log.prob > 0.5, 1, 0)
```

```
table(occupancy.log.pred, occupancy_test$Occupancy)
```

```
##
## occupancy.log.pred    0    1
##                  0 1849   96
##                  1   68  430
```

Conclusion

Using a threshold probability of 0.5, the logistic regression gives an error rate of (96+68)/2443 = 0.0671. The error rate for predicting *Not Occupied* is smaller at 68/(68+1849)=0.0355.

Linear Discriminant Analysis (LDA)

For the LDA, we are able to use all the predictor variables

```
occupancy.lda <- lda(Occupancy ~ Temperature + Humidity + Hour + CO2,
                     data = occupancy_train)
occupancy.lda.pred <- predict(occupancy.lda,
                          as.data.frame(occupancy_test))
table(occupancy.lda.pred$class, occupancy_test$Occupancy)
```

```
##
##        0    1
##   0 1844  111
##   1   73  415
```

Conclusion

The LDA gives an error rate of (111+73)/2443 = 0.0753. The LDA does extremely well at finding the rooms that are *Not Occupied*. The error rate for this is only 73/(1844+73) = 0.0380.

Quadratic Discriminant Analysis (QDA)

Again, for the QDA, we can use all the predictor variables

```
occupancy.qda <-qda(Occupancy ~ Temperature + Humidity + Hour + CO2,
                    data = occupancy_train)
occupancy.qda.pred <- predict(occupancy.qda,
                          as.data.frame(occupancy_test))
table(occupancy.qda.pred$class, occupancy_test$Occupancy)
```

```
##
##        0    1
##   0 1832   81
##   1   85  445
```

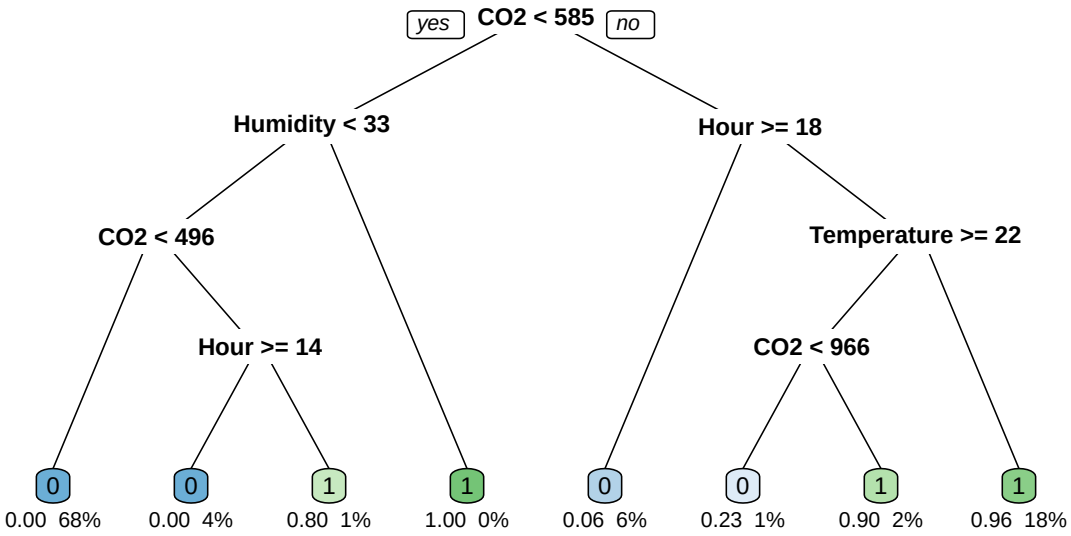
Conclusion

The QDA gives an error rate of (81+85)/2443 = 0.0679, which is better than the LDA. This is expected because the QDA uses quadratic equations for the boundaries, instead of linear equations. Again, the QDA does better at finding the rooms that are *Not Occupied*. The error rate for this is 85/(85+1832) = 0.0443.

Classification Trees

Another model type we could use for our data is Classification Trees. Here is the classification tree for the training data:

```
occupancy.tree <- rpart(Occupancy ~ Temperature + Humidity + Hour + CO2,
                        data = occupancy_train,
                        method = "class")
rpart.plot(occupancy.tree,
           type = 0,
           clip.right.labs = FALSE,
           branch = 0.1,
           under  = TRUE)
```



We can see that the classification tree chose CO2 as the first break. This reinforces what we said earlier in the bivariate modeling about CO2 being important for the classifiers. The next breaks were at Humidity and Hour. The classification tree was able to use all 4 variables. The only problem this could cause is overfitting. Let's look at the prediction table to see if we do have a problem of overfitting.

```
occupancy.tree.pred <- predict(occupancy.tree,
                               as.data.frame(occupancy_test),
                               type = "class")
table(occupancy.tree.pred, occupancy_test$Occupancy)
```

```
##
## occupancy.tree.pred    0    1
##                   0 1883   15
##                   1   34  511
```

Conclusion

The error rate for the classification tree model is (34+15)/2443 = 0.0200. The tree did exceptionally well at finding the rooms *Not Occupied*. The error rate for this was only 34/(34+1883)= 0.0177.

Also, since the Classification Tree wasn't 'perfect' with error rate 0, overfitting is probably not a problem.

Final Prediction

Out of the four classifiers, the Classification Tree was the best classifier. It had an error rate of only 0.0200, which was smaller than the logistic regression model, LDA, and QDA. The logistic regression and QDA performed almost the exact same, both with an error rate of around 0.067. And the LDA performed the

worse, relative to the other 3 classifiers. One thing that was interesting was that all four classifiers were better at predicting if a room was *Not Occupied*.

Based on this information, our prediction is the Classification Tree. Using the classification tree and given some data, we would be able to predict Occupancy of a Room with an error rate of only 0.0200. Still, we could run into the problem of overfitting with the classification tree. If that is the case, the next best option is the logistic regression. We could also modify the classification tree to include less branches and that could solve the problem of overfitting, but it would most likely increase the error rate.

Discussion

Overall, our models performed decently well at predicting occupancy. It would have been nice to find a model with error rate less than 0.01, but the Classification Tree was the best we could do. The next steps would be to gather more data and run the classification tree on this data to see if there is evidence of overfitting. As always, more data will help get better and more accurate results.

In future studies, if we could find another predictor variable that affects Occupancy of a room, it could help increase the accuracy of the classifiers. This variable could be something like cleanliness of the room or something else that psychologically affects a person's desire to enter a room or not. This paper was a good start but there could definetely be more done. This research is important, as well, because more accuracy could translate to more lives saved during emergencies and evacuation situations.