# Predicting Room Occupancy to Signal Fire Emergencies

*Kenneth Huang*
*khuang2*

*Due Mon, Nov 22, at 8:00PM*

## Contents



```
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")
```

## Introduction

We were trained as children the proper reactions to fire emergencies. First, the smoke detector would go off and an alarm begins to sound. When we hear the alarm, we evacuate the building as orderly as possible. However, in a real situation, the pressure of needing to leave a building as soon as possible creates chaos, and fully evacuating a building may take longer than expected.

Now, what if there was a way to give people more time to evacuate? By the time a smoke detector goes off, the fire in a building may already be spreading. With machine learning and data analysis of past room occupations, we can help machines predict exactly when a building should be evacuated.

In this paper, we will predict room occupancy by using and evaluating various machine learning classification techniques. We will train our models to predict room occupancy based on a room's temperature, humidity, CO2 level, and hour of the day.

(Data pulled from *Energy and Buildings*, written by Luis M. Candanedo and Véronique Feldheim.)

# Exploratory Data Analysis

## Dataset Overview

We will work with two datasets in this paper: occupancy_train and occupancy_test. occupany_train will be used to train our machine learning classification models, and occupancy_test will be used to test their accuracy.

Below is a summary of each file:

- *occupancy_train*: 5700 observations, 5 variables
- *occupancy_test*: 2443 observations, 5 variables

Our datasets have 5 variables each: 4 exploratory variables and 1 response variable.

Exploratory Variables:

- *Temperature*: room temperature in degrees C
- *Humidity*: room relative humidity, in percent
- *CO2*: room's carbon dioxide in ppm
- *Hour*: hour of the day, from 0 to 23

Response Variable:

- *Occupancy*: binary, 0 for not occupied, 1 for occupied status

## Summary of Occupancy in the Dataset

For the purposes of conducting exploratory data analysis, we will analyze data from occupany_train to get a better idea of what we will run our machine learning classification models on.

We will begin exploring our training dataset by providing a summary of our response variable, occupancy. This will give us a picture of what proportion of rooms are occupied in our dataset.

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

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

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

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

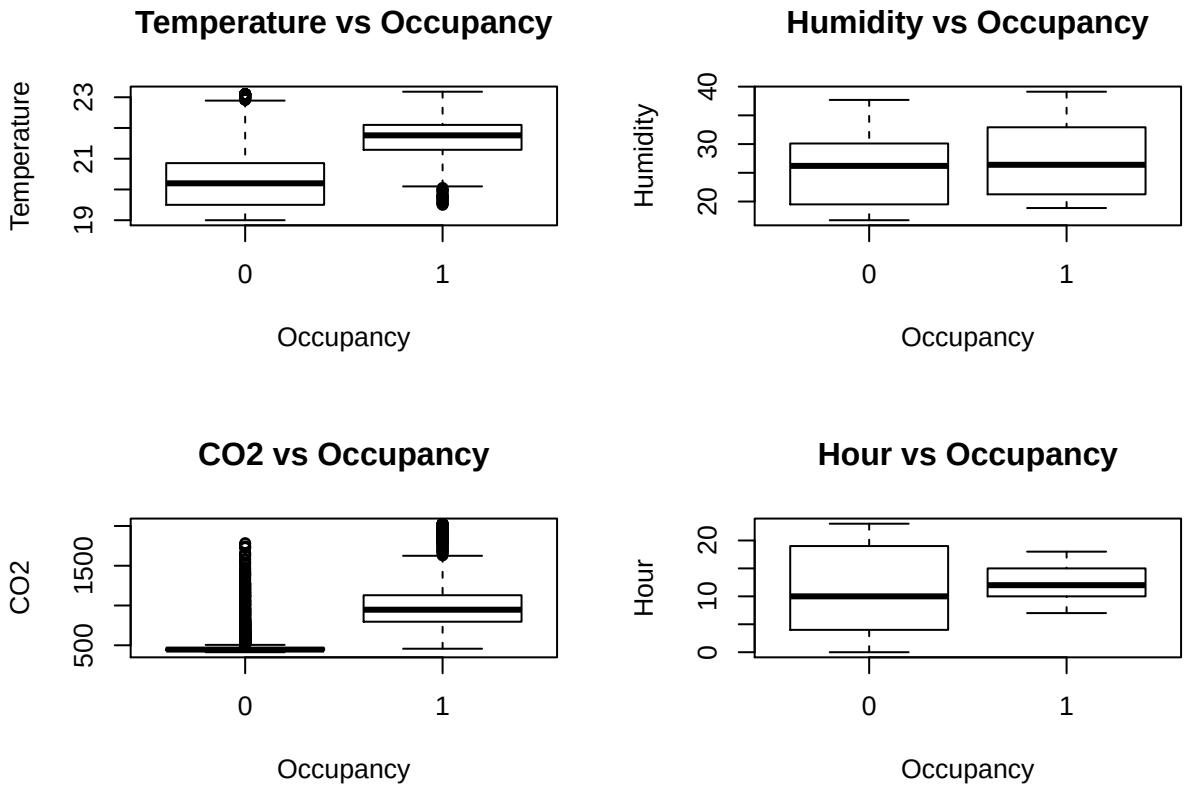
We notice that from our dataset, 4497 of observations (78.89%) contain data that corresponds to rooms that are not occupied, while 1203 of observations (21.11%) contain data that corresponds to rooms that are occupied.

With a better idea of the proportion of rooms occupied in our dataset, we can move onto exploring our explanatory variables.

### Explanatory variables vs. Response Variables

Let's create boxplots to visualize how each explanatory variable relates to room occupancy.

```
par(mfrow = c(2,2))
boxplot(data = occupancy_train, Temperature ~ Occupancy, main = "Temperature vs Occupancy")
boxplot(data = occupancy_train, Humidity ~ Occupancy, main = "Humidity vs Occupancy")
boxplot(data = occupancy_train, CO2 ~ Occupancy, main = "CO2 vs Occupancy")
boxplot(data = occupancy_train, Hour ~ Occupancy, main = "Hour vs Occupancy")
```




Temperature vs Occupancy:

Looking just at the boxplots, there appears to be a difference in Temperature between rooms that are occupied vs. those that are not. We notice that occupied rooms have higher temperatures on average, indicating that heat may be on.

Humidity vs Occupancy:

We see that the median humidity of rooms occupied vs not occupied are about the same, which may indicate that humidity does not have a significant impact on whether a room is occupied.

CO2 vs Occupancy:

This boxplot may be the most interesting of the four, where the CO2 levels of rooms unoccupied are on average far less than the CO2 levels of rooms occupied. This is explained by the fact that humans breath out CO2, and when there are no people in the room, low amounts of CO2 will be produced. However, it is fair to note that for rooms unoccupied, there spread of outliers is astonishingly high.

Hour vs Occupancy:

We notice that the spread of hours for rooms occupied are far less than the spread of hours for unoccupied rooms. This can be explained by people only being in buildings during a certain time in the day. However, we also note that there is not much of a difference in the median hours of rooms occupied vs. those not occupied.

# Modeling

Now that we have a general sense of each variable and their relationships with occupancy, we can move forward with predicting room occupancy using various classification techniques.

In this section, we wil explore the effectiveness of four classification techniques: LDA, QDA, binary classification tree, and binary logistic regression.

Note that for our LDA and QDA models, we will only use our continuous explanatory variables.

## Using Linear Discriminant Analysis (LDA)

The LDA is constructed as shown below:

```
occupancy_lda <- lda(data = occupancy_train,
                     Occupancy ~ Temperature + Humidity + CO2)
```

To assess the performance of our LDA, we will run:

```
occupancy_lda_pred <- predict(occupancy_lda,
                              as.data.frame(occupancy_test))
table(occupancy_lda_pred$class, occupancy_test$Occupancy)
```

```
##
##        0    1
##   0 1842  116
##   1   75  410
```

Next, we will calculate the error rate below:

Looking at the table, we see that we have an overall error rate of $(75+116)/(1842+116+75+410) = 7.82\%$. In particular, our error rate for determining if a room is unoccupied is $75/(1842+75) = 3.91\%$, while the the rate for determining if a room is occupied is $116/(116+410) = 22.05\%$. These numbers suggest that LDA struggles with determining if a room is occupied, and we should explore other techniques to see if this error rate can be diminished.

## Using Quadratic Discriminant Analysis (QDA)

The QDA is constructed and assessed as show below:

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

```
##
##        0    1
##   0 1834   98
##   1   83  428
```

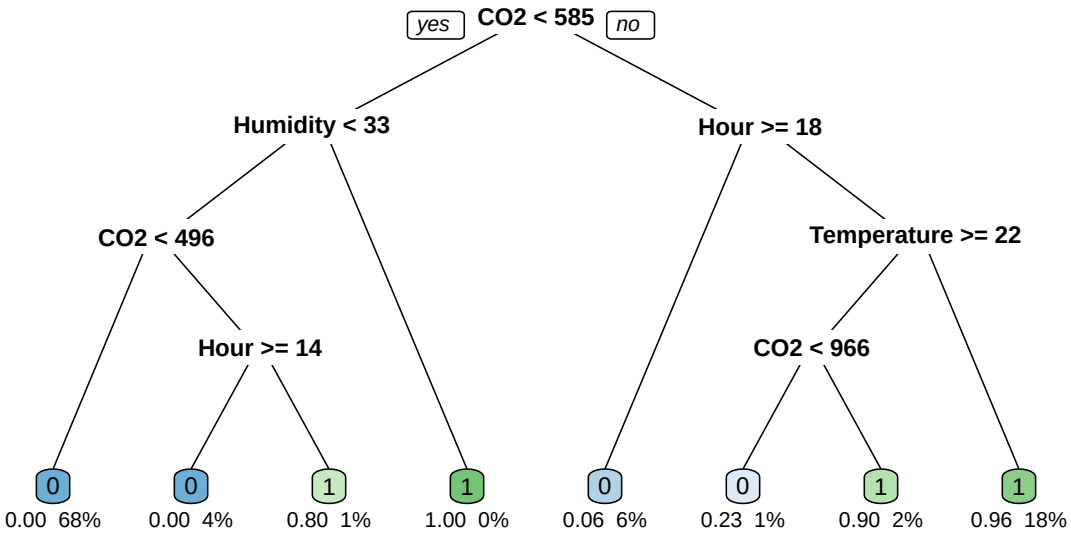
Using QDA, we receive an overall error rate of $(83 + 98)/(1834 + 98 + 83 + 428) = 7.41\%$, slightly lower than the overall error rate when we used LDA. In addition, our error rate of predicting an unoccupied room is $83/(1834 + 83) = 4.32\%$ and our error rate of predicting an occupied room is $98/(98 + 428) = 18.63\%$. In comparison to the LDA, our QDA does a slightly worse job at predicting an unoccupied room, while it does a greater job at predicting an occupied room.

## Using a binary classification tree

With a binary classification tree, we can now predict room occupancy with one additional variable, Hour. Below is our classification tree, fitted on our training data occupancy_train:

```
occupancy_tree <- rpart(data = occupancy_train,
                        Occupancy ~ Temperature + Humidity + Hour + CO2,
                        method = "class")

rpart.plot(occupancy_tree,
           type = 0,
           clip.right.labs = FALSE,
           branch = 0.1,
           under = TRUE)
```




We see that the most important indicator of room occupancy is the CO2 variable, which is not surprising when we refer back to the boxplot made for CO2 vs Occupancy. The next most important indicators of room occupancy are the Humidity (when CO2 $< 585$) and the Hour (when CO2 $>= 585$) variables. We also notice that Temperature is the least important variable, which agrees with our observations made when we looked at the boxplot for Temperature vs Occupancy.

Let's take a look at the accuracy of this classification tree on some testing data:

```
occupancy_tree_predict <- predict(occupancy_tree,
                                  as.data.frame(occupancy_test),
                                  type = "class")
table(occupancy_tree_predict, occupancy_test$Occupancy)
```

```
##
## occupancy_tree_predict    0    1
##                      0 1883   15
##                      1   34  511
```

```
## [1] 0.02005731
```

```
## [1] 0.01773605
```

```
## [1] 0.02851711
```

Using our binary classification tree, we receive an overall error rate of $(34+15)/(1883+15+34+511) = 2.01\%$. Looking at the error rate of predicting unoccupied rooms, we receive an error rate of $34/(1883+34) = 1.78\%$, while the rate for predicting occupied rooms is $15/(15+511) = 2.85\%$. Compared to the LDA and QDA models, our classification tree has both a lower overall error rate and a lower error rate for predicting unoccupied and occupied rooms.

## Using a binary logistic regression model

Lastly, we will explore how a binary logistic regression model compares to our LDA, QDA, and classification tree. We will train our binary logistic regression model as shown below:

```
occupancy_logit <- glm(data = occupancy_train,
                       factor(Occupancy) ~ Temperature + Humidity + Hour + CO2,
                       family = binomial(link = "logit"))
```

And we apply our model to the testing data as follows:

```
occupancy_logit_pred <- predict(occupancy_logit,
                                as.data.frame(occupancy_test),
                                type = "response")
```

Our current results display probabilities rather than classifications for room occupancy. To combat this, we will set a threshold of 0.5, where probabilities of $> 0.5$ will indicate a classification of occupancy, while a probability of $<= 0.5$ will indicate the other option of occupancy.

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

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

```
occupancy_logit_pred <- ifelse(occupancy_logit_pred > 0.5, "1", "0")
table(occupancy_logit_pred, occupancy_test$Occupancy)
```

```
##
## occupancy_logit_pred    0    1
##                    0 1849   96
##                    1   68  430
```

Using a binary logistic regression model, we receive an overall error rate of $(68+96)/(1849+96+68+430) = 6.71\%$. In particular our error rates for predicting unoccupied and occupied rooms are $68/(1849+68) = 3.54\%$ and $96/(96+430) = 18.25\%$, respectively.

### Final Recommendation

We recommend that for predicting occupancy of a room, a binary classification tree should be used because of its high prediction accuracy in comparison to the other methods.

A summary of the error rates for each model is listed below:

Overall Error Rate:

- LDA: 0.07818256 (HIGHEST)
- QDA: 0.07408923
- Classification Tree: 0.02005731 (LOWEST)
- Logistic Regression: 0.06713058

Error Rate of Predicting an Unoccupied Room:

- LDA: 0.03912363
- QDA: 0.04329682 (HIGHEST)
- Classification Tree: 0.01773605 (LOWEST)
- Logistic Regression: 0.03547209

Error Rate of Predicting an Occupied Room:

- LDA: 0.2205323 (HIGHEST)
- QDA: 0.1863118
- Classification Tree: 0.02851711 (LOWEST)
- Logistic Regression: 0.1825095

From these points, we see that our classification tree consistently outperforms the other models, where it displays the lowest overall and individual error rates.

## Discussion

In this paper, we discussed and determined the best classification technique to use for predicting whether a building/room is occupied. We concluded that a binary classification tree would be the best technique to use because of its considerably low error rate relative to other techniques.

Using the binary classification tree, we also see that CO2 levels is the most significant predictor of occupancy. Despite using a classification tree, we need to be aware of some limitations of this technique. Most importantly, we need to tune our tree's maximum depth in order to make sure that our model does not begin to overfit the data.

Overall, this paper helped motivate further discussion on the uses of predicting room occupancy. As we mentioned in the introduction, these results can be used to determine exactly when a building should be evacuated. By using this research to determine evacuation rules, we can decrease the incident rate of building fires. We hope that future data analysts can build onto this research by considering more classification techniques, and by fine-tuning our binary classification tree.