# Occupancy Detection

## MT7038

Anton Stråhle & Max Sjödin

Fall 2020

# Data

The occupancy status of a room was observed for a few days. Snapshots of the features below were taken every minute.

- Features
  - Temperature
  - CO2
  - Humidity
  - HumidityRatio
  - Light
- Response
  - Occupancy
    - Occupied
    - Unoccupied

# Data

The occupancy status of a room was observed for a few days. Snapshots of the features below were taken every minute.

- Features
    - Temperature
    - CO2
    - Humidity
    - HumidityRatio
    - ~~Light~~
- Response
    - Occupancy
        - Occupied
        - Unoccupied

Light is excluded as the best classifier would otherwise become *Are the lights on?*

# Brief Exploration

- Unbalanced data set
  - Many more unoccupied data points than occupied

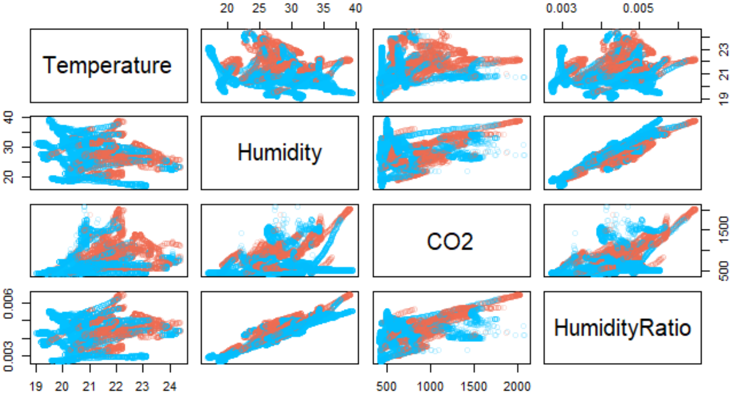


Figure: Pairplots of Features

- Non-linearity?

# Brief Exploration

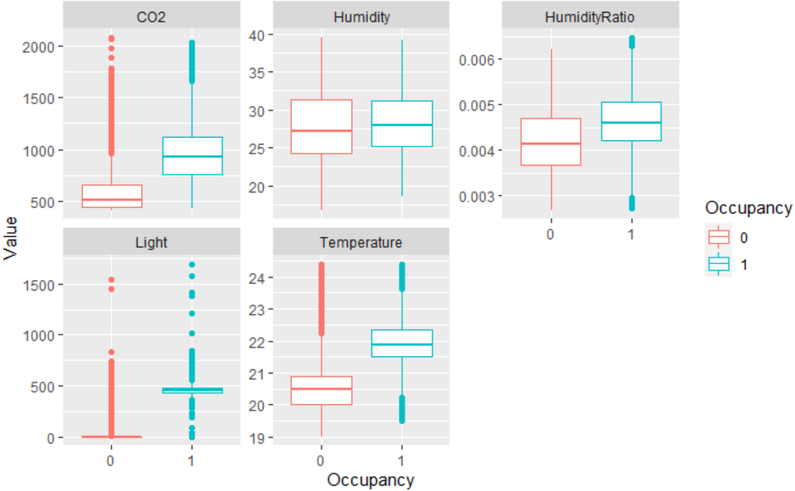


Figure: Boxplots of Features: Standardized and unstandardize

# Methodology

- SVM
  - Linear, Radial & Polynomial
- KNN
  - Regular & Weighted
- Decision Trees
  - Single Tree, Bagging, Boosting

# Methodology

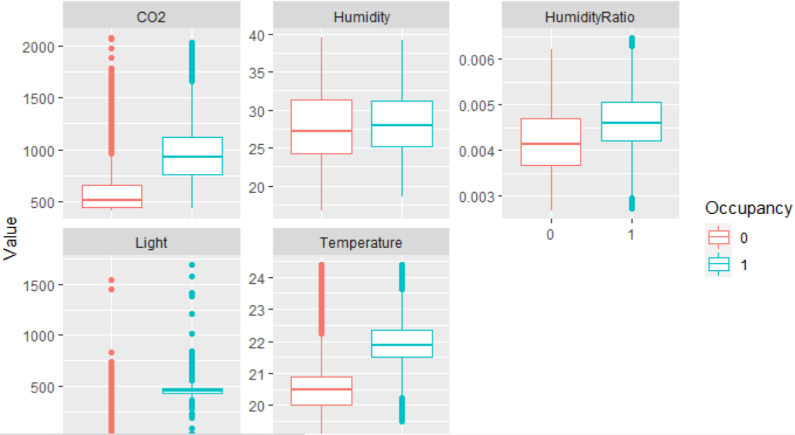
## SVM

Why? ▷ Good for non-linear classification problems.

How? ▷ Using the package **e1071** and the function **svm**

▷ Linear, polynomial and radial kernels

▷ Coarse-to-fine parameter search



# Methodology

## KNN

Why?
- ▷ Good for non-linear classification problems
- ▷ Good with large training data sets
- ▷ Good if data is not noisy

How?
- ▷ Regular KNN using the package **class** and the function **knn**
- ▷ Weighted KNN using the package **kknn** and the function **kknn**
  - ▷ Epanechnikov kernel

# Methodology

## KNN - Regular

After a coarse search for a good value of $k$ we noted that the best classifier was a 1-NN which further indicates that the data is not very noisy at all. The 1-NN achieved a testing accuracy of 93%.

A possible improvement is to use a weighted KNN where we put more emphasis on training points closer to the point which we want to predict than those further away.

# Methodology

## KNN - Weighted

In order to weight our data points we use the kernel distance from the point we want to predict to the $k$ nearest neighbours. The choice of kernel is of course important but in our case all the available kernels in the function **kknn** generated approximatley the same results. As such we resorted to the Epanechnikov kernel as it is one we've encountered before.

When search for a good value of $k$ in this case we found that the best validation accuracy was obtained for $k = 25$ which seems a bit more stable than using a 1-NN. This was also reflected in the testing accuracy which turned out to be 97.5%.

# Methodology

## Decision Trees

Why? ▷ Good with non-linear data

▷ Good with binary classification

How? ▷Bagging using the package **ipred** using function **bagging**

▷ Boosting using the package **adabag** using function **boosting**

# Methodology

## Decision Trees - Single

# Methodology

## Decision Trees - Bagging

# Methodology

## Decision Trees - Boosting

# Discussion