Title

XiangCheng Xu xiangchx

Due Monday, August 7, at 11:59PM

Contents


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

Introduction

With the booming of technology, people are able to determine the occupancy of a room by using different ways. One of them is to build statistic model to make prediction through several predictor. In this paper, it will mainly focus on the prediction of the occupancy of a room by building statistic model with four different predictors, which are the room temperature, the room relative humidity, carbon dioxide level of the room, and the hour of the day.

Exploratory Data Analysis

Overview/background of dataset and variables

In this paper, there will be two major data set, "occupancy_train" and "occupancy_test". The first one will be used to build and train final model, and the second one will be taken as the verification and evaluation data for the created model. The table will show the first several lines of the data, as follows:

```
head(occupancy_test)
```

```
## # A tibble: 6 x 5
##   Temperature Humidity  Hour   CO2 Occupancy
##         <dbl>    <dbl> <dbl> <dbl>     <dbl>
```

```
## 1        23.2     27.3    17  714          1
## 2        23.1     27.2    17  701          1
## 3        23.1     27.1    18  691          1
## 4        23       27.1    18  680.         1
## 5        22.9     27.3    18  685          0
## 6        22.9     27.5    18  688          0
```

```
head(occupancy_train)
```

```
## # A tibble: 6 x 5
##   Temperature Humidity  Hour   CO2 Occupancy
##         <dbl>    <dbl> <dbl> <dbl>     <dbl>
## 1        21.4     25.7    22  486          0
## 2        20.8     19.6    19  547          0
## 3        19.3     31.2     9  431.         0
## 4        19.2     31.2     7  431          0
## 5        20.3     32.9     3  452          0
## 6        20.4     18.6     5  433          0
```

In order to make the classification more accurate and precise, predictors chosen are: + Temperature: room temperature in degrees of Celsius + Humidity: room relative humidity, in percent + CO2: room's carbon dioxide in ppm + Hour: hour of the day, from 0 to 23 and the response label that will be predicted is: + Occupancy: binary, 0 for not occupied, 1 for occupied status

Relationship between response and quantitative explanatory

Before building a classification model with the given data. It is necessary to have some analysis towards both the response labels and given predictors/classifiers.

Probability of response label

It is not hard for us to notice that there are totally 5700 observations in the training data set, with 1203 of them marked as "1" that stands for "Occupied" and 4497 of them marked as "0" that stands for "Unoccupied". By using the R studio, we successfully computed the probability for each type of factor in the response label that is contained in this paper. About 78.89% of data in the response label is marked as "Unoccupied", and about 21.11% of data in the response labels are marked as "Occupied". The detailed information of the training data set is provided in the following space.

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

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

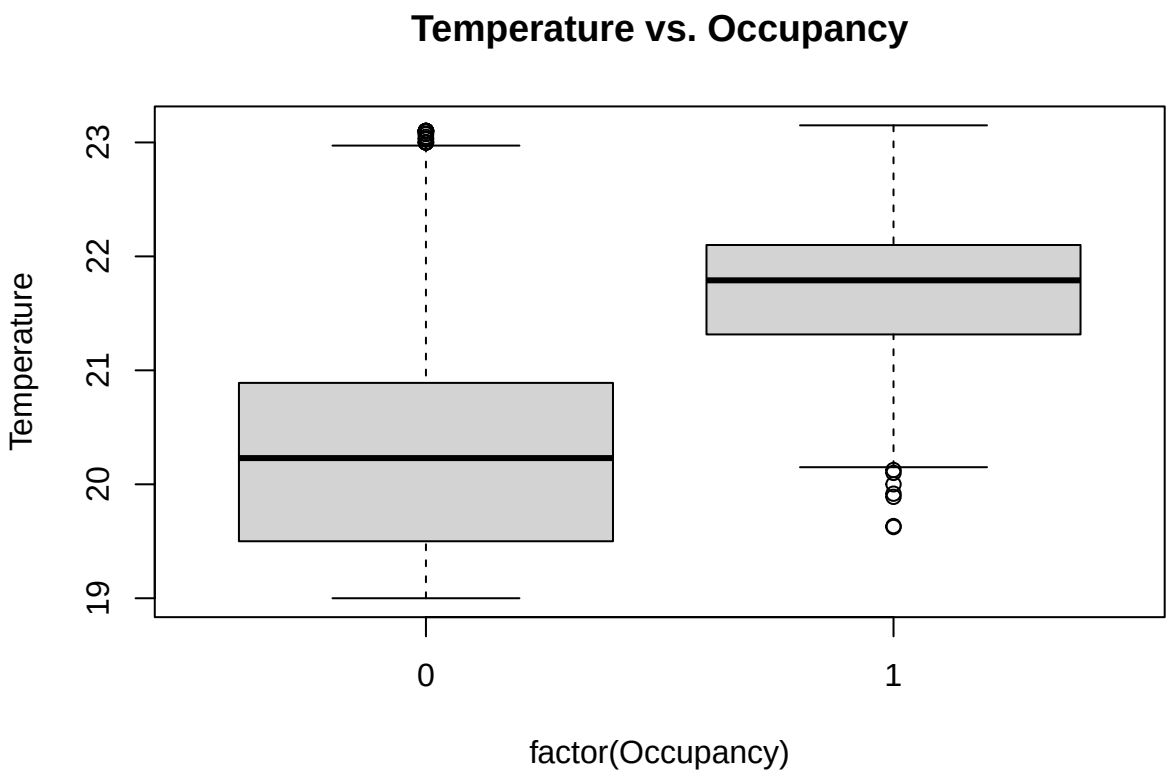
```
prop.table(table(occupancy_train$Occupancy))
```

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

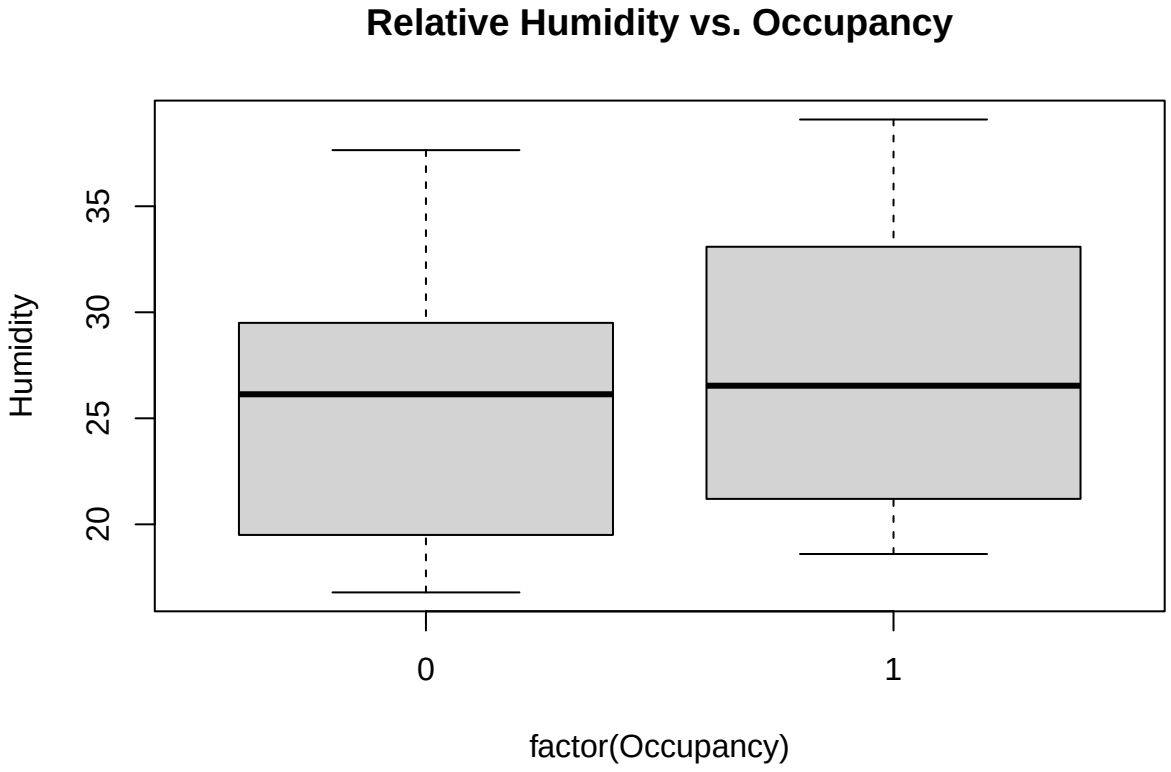
EDA on relationships between the response label and predictors/classifier

Since the response label in the training data set is categorical, and predictors/classifiers in the data set is quantitative, it tends to a good idea to visualize relationships between the response label and predictors/classifiers with the use of boxplots, which is shown as follows:

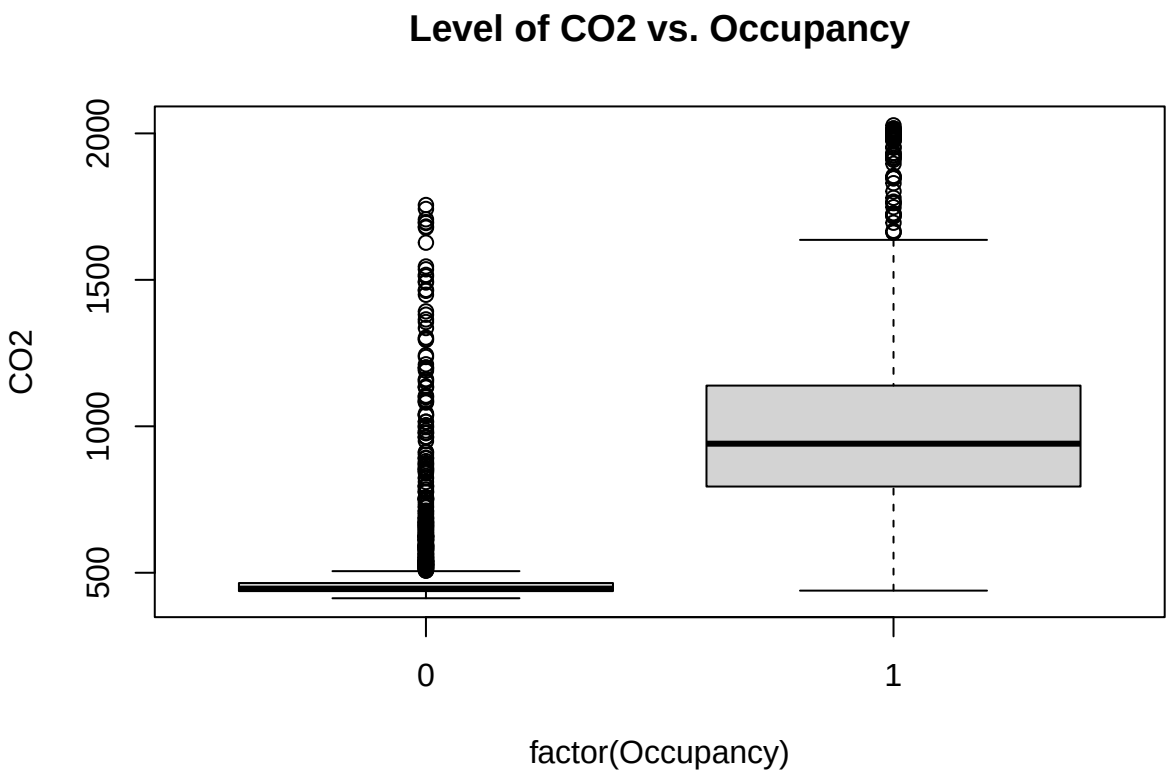
```
boxplot(Temperature ~ factor(Occupancy), data = occupancy_test, main = "Temperature vs. Occupancy")
```



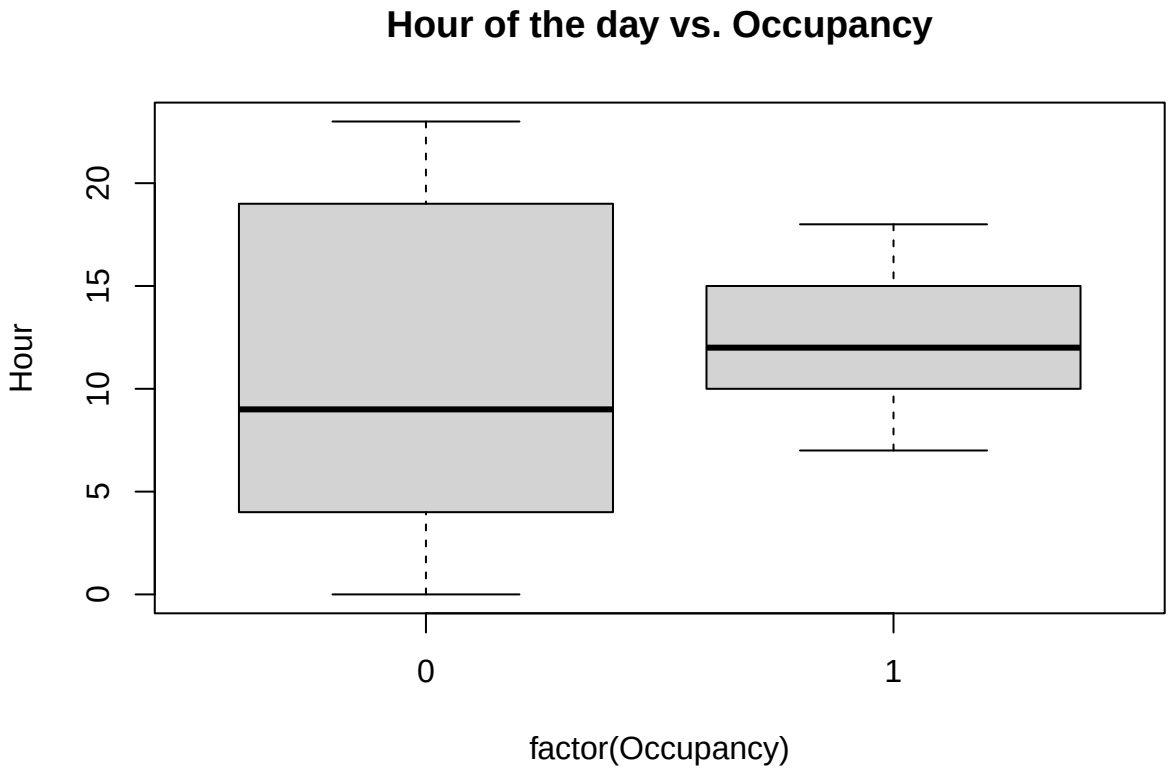

```
boxplot(Humidity ~ factor(Occupancy), data = occupancy_test, main = "Relative Humidity vs. Occupancy")
```




```
boxplot(CO2 ~ factor(Occupancy), data = occupancy_test, main = "Level of CO2 vs. Occupancy")
```




```
boxplot(Hour ~ factor(Occupancy), data = occupancy_test, main ="Hour of the day vs. Occupancy")
```



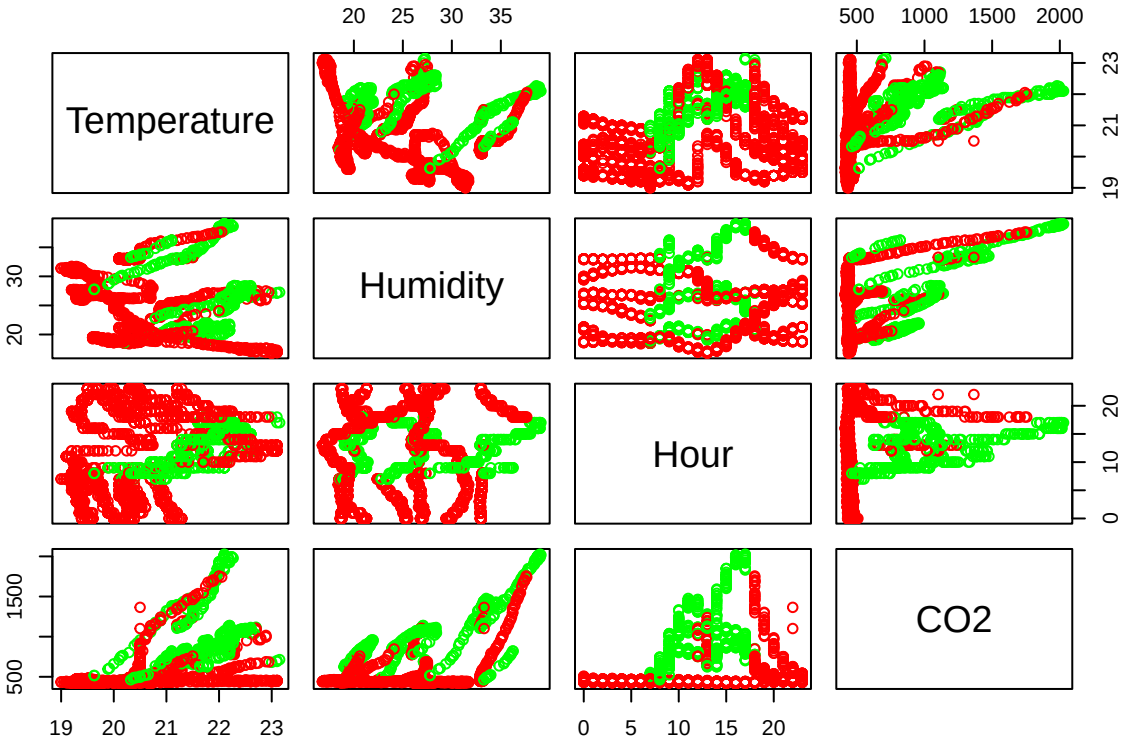
From boxplots above, we can tell some general relationships between the response label and each predictor/classifier that might be useful when building the model. The occupancy of a room can be positively related with the hour of a day, the level of carbon dioxide in a room, the relative humidity in a room(which

might not be as significant as the others), and the temperature in a room

Visual EDA between predictors/classifiers

Before building models for the response label, we can use the pair plot to help us determine whether there are any pairs of plot to help us to determine the classification of the response label. The pair plot is shown as follows:

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



From the pair plot generated by R, we can visualize relationships between the response label and each pair of predictors/classifiers. Most of them do not show any reasonable seperation, for example: Humidity and Hour, Humidity and temperature, Temperature and Humidity, and so on. However, there is an exception which is the plot of Hour and CO2. It seems that comparing with the level of CO2 in a room, the hour of the day might be a better predictor, as the higher of the hour of the day, the less likely a room is occupied.

Even though we might see some simple, potential trends among the classification of the response label and predictors/classifiers, it is still necessary to create a model that is more complicated for further determination.

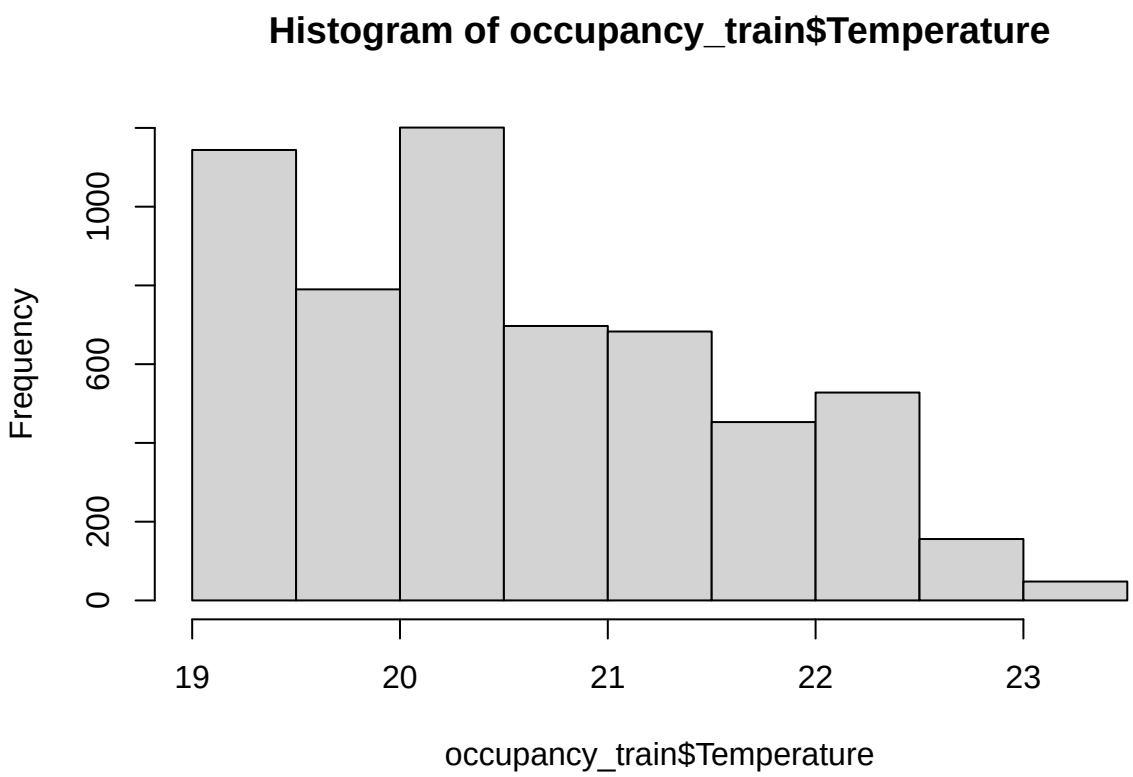
Modeling

After doing EDA for the response label, relationships between the response label and predictors, and relationships among pairs of predictors/classifiers and the response label, it is the time to build classification model for the occupancy of a room. In this paper, 4 different classification models will be tested, which are linear discriminant analysis (lda), quadratic discriminant analysis (qda), classification trees, and binary logistic regression.

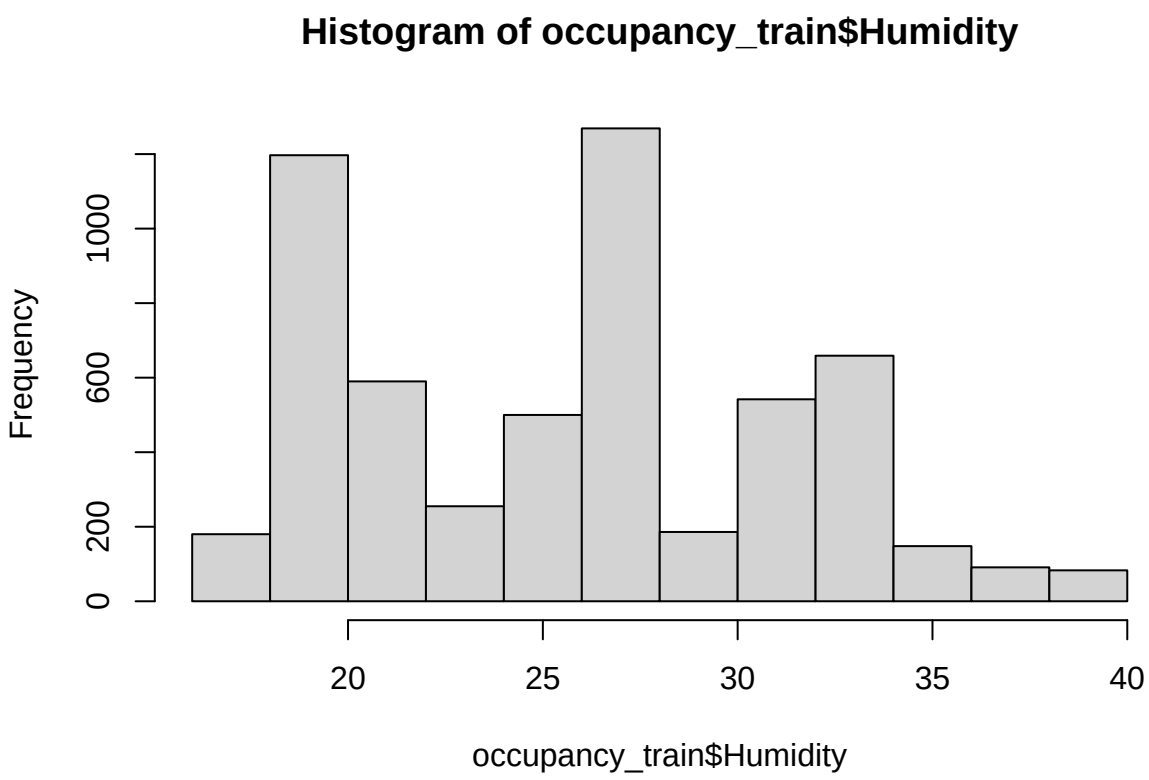
Discriminant Analysis

If we want to apply LDA and QDA models to classify the response variable, the predictors/classifiers used should be continuous,and predictors/classifiers need to be normally distributed. Therefore, we can build histograms for predictors/classifiers to see whether the data validate assumptions.

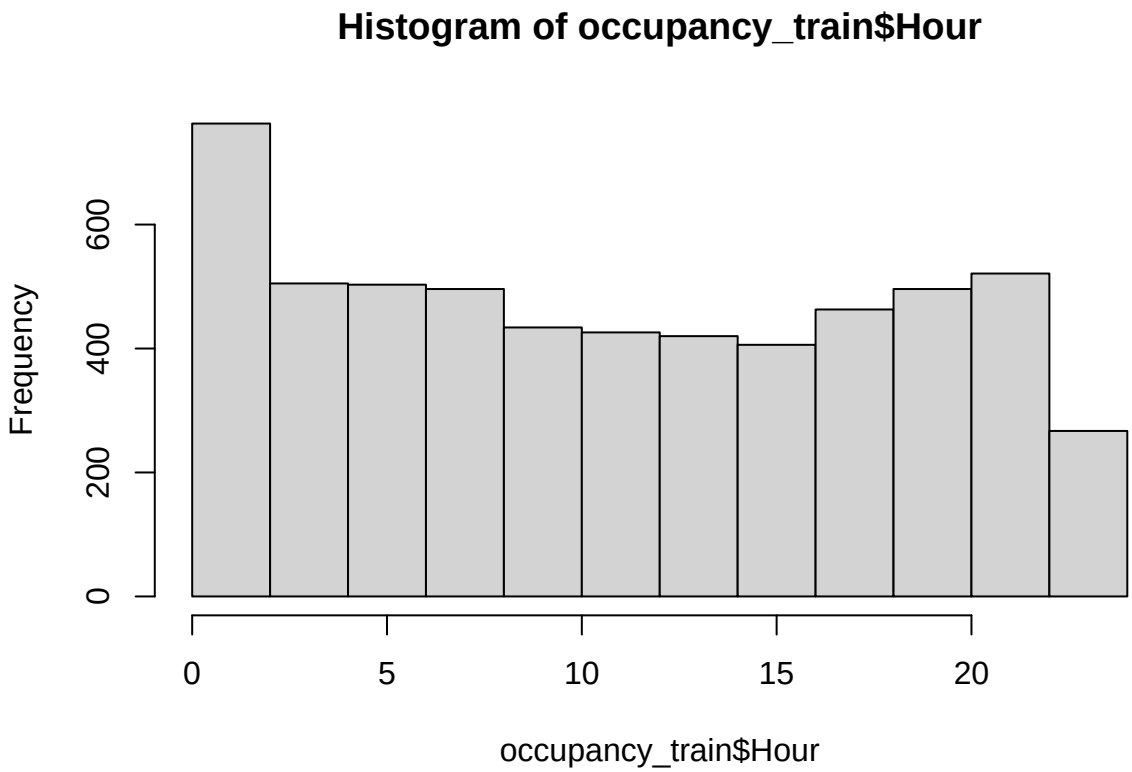
```
hist(occupancy_train$Temperature)
```



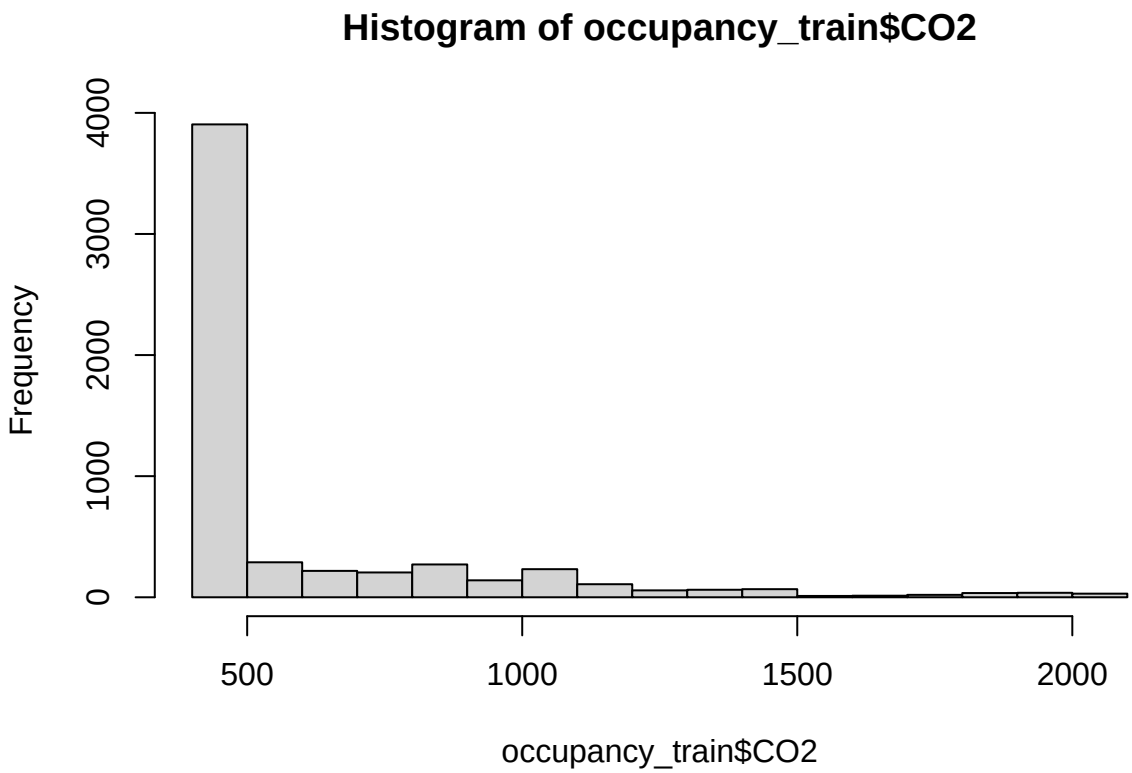

```
hist(occupancy_train$Humidity)
```



```
hist(occupancy_train$Hour)
```




```
hist(occupancy_train$CO2)
```



Since all predictors in the training data set are quantitative variables, each of the predictor/classifier have the

data type of "double"(see the beginning part of the *Exploratory Data Analysis*), and their histograms all look continuous, it is safe to say that predictors/classifiers in this paper validate the assumption of continuous variables. Nevertheless, histograms for predictors/classifiers are not normally distributed in this case, thereby might leading to the possibility of having high error rate when testing both of discriminant models. In order to see whether the data needs to be justified or not, we can firstly to see the performance of models using the original data.

Linear Discriminant Analysis

The LDA model and its predicted-value is built as follows:

```
occupancy.lda <- lda(factor(Occupancy) ~ Temperature + Humidity + Hour + CO2,
                             data = occupancy_train)
occupancy.pred.lda <- predict(occupancy.lda, occupancy_test)
```

Then, we are able to determine the performance of LDA model by creating the confusion matrix with the test data and the predicted-value as follows:

```
table(occupancy.pred.lda$class, occupancy_test$Occupancy)
```

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

and the error rate of the model is

```
(111+73)/(1844+111+73+415)
```

```
## [1] 0.07531723
```

From the result computed by R, we can see that the overall error rate for the LDA model using the original, unconverted data is about 0.07532, which is quite low. To be more specific, the error rate for unoccupied is 111/(1844+111), which is about 0.0568, and the error rate of occupied is 73/(73+415), which is about 0.15. As a result, the LDA model using original data tends to be acceptable.

Quadratic Discriminant Analysis

The QDA model and its predicted-value is built as follows:

```
occupancy.qda <- qda(factor(Occupancy) ~ Temperature + Humidity + Hour + CO2,
                             data = occupancy_train)
occupancy.pred.qda <- predict(occupancy.qda, occupancy_test)
```

Then, it is the time to determine the performance of QDA model by creating the confusion matrix with the test data and the predicted-value as follows:

```
table(occupancy.pred.qda$class, occupancy_test$Occupancy)
```

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

and the error rate for the model is:

```
(81+85)/(1832+81+85+445)
```

```
## [1] 0.06794924
```

From the result computed by R, we can see that the overall error rate for the QDA model is about 0.067, which also seems to be quite small. In particular, the error rate for unoccupied is 81/(1832+81), which is

about 0.04434, and the error rate for occupied is 85/(85+445), which is about 0.1604. As a result, the QDA model using the original data seems to be acceptable.

Logistic regression

The next model that is considered in this paper is the logistic regression model. In this paper, the response label will be binary, as it is classified in to "Occupied" or "Not occupied". Therefore, we are going to build a binary logistic regression model to help us to do the classification of the response label.

At the same time, since the response label is classified as 0 and 1, shown as follows. Therefore, we might separate the response label by determine whether the value of occupancy is smaller than 1.

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

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

The detail code for building the model and the predicted-value is shown as follows:

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

occupancy.logit.prob <-predict(occupancy.logit, occupancy_test, type = "response")

occupancy.pred.logit <- ifelse(occupancy.logit.prob > 0,"Occupied", "Not Occupied")
```

The confusion matrix for the binary logistic model is shown as follows

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

```
##
## occupancy.pred.logit    0    1
##             Occupied 1917  526
```

and the error rate for the model is

```
(49+168)/(1868+168+49+358)
```

```
## [1] 0.08882521
```

As a result, the overall error rate for the model is about 0.0888, which is a low value. To be more specific, the error rate for the "Not Occupied" is 168/(168+1868), which is 0.0825, and the error rate of the "Occupied" is 49/(49+358), which is 0.1204. Comparing with LDA and QDA, the binary logistic regression tends to perform better in the error rate of the "Occupied", but poorer in the "Occupied" and the overall error rate.

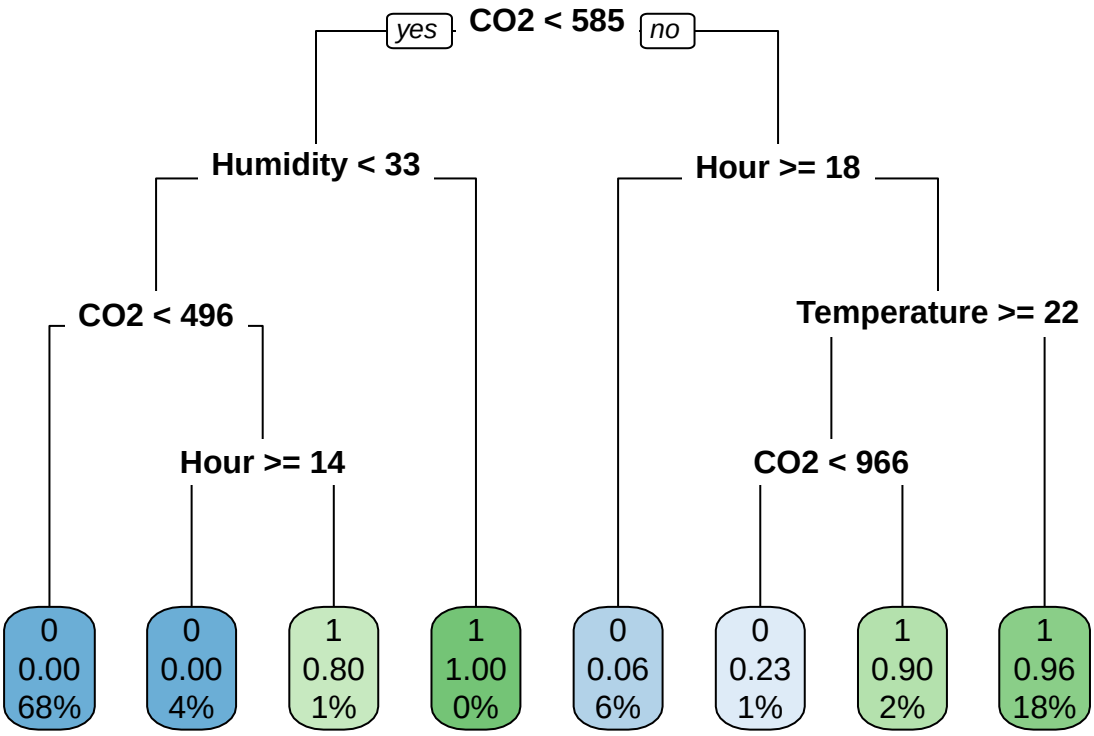
Classification Tree

Last but not the least, is the use of the classification tree. We can fit a classification tree, the predicted-value and the classification tree plot by using the following code:

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

occupancy.pred.tree <- predict(occupancy.tree, occupancy_test, type = "class")
```

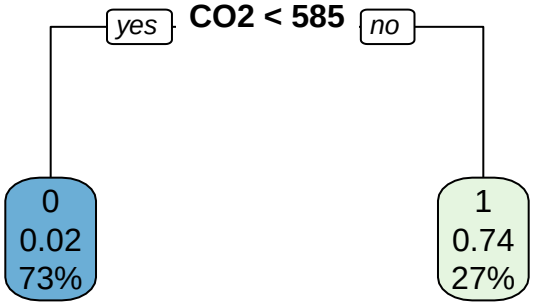
```
rpart.plot(occupancy.tree,
           type = 0)
```



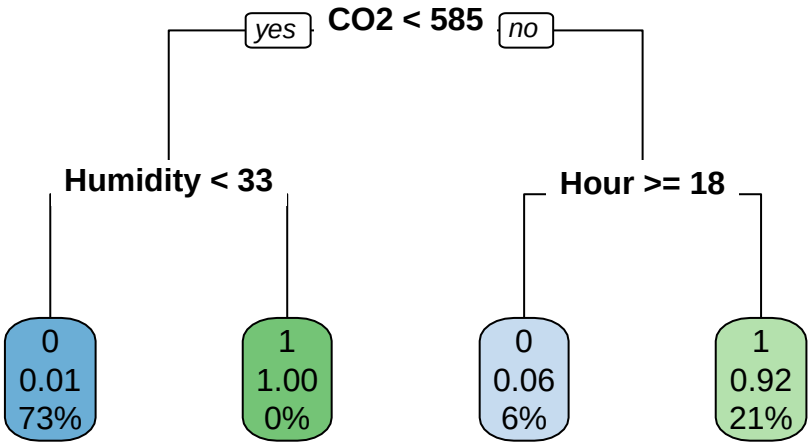
However, when applying the classification tree, it is necessary for us to consider the problem of over fitting. In this situation, it can be understood as the classification tree might perform well in training data, but not testing data. As a matter of fact, it is necessary to control the number of branches of the tree. The following codes build a "smaller" tree with maximum depth from 1 to 3.

```
occupancy.tree.tiny <- rpart(factor(Occupancy) ~ Temperature + Humidity + CO2 + Hour,
                        data = occupancy_train,
                        method = "class",
                        control = rpart.control(maxdepth=1))

occupancy.pred.tree.tiny <- predict(occupancy.tree.tiny, occupancy_test, type = "class")
rpart.plot(occupancy.tree.tiny,
           type = 0)
```

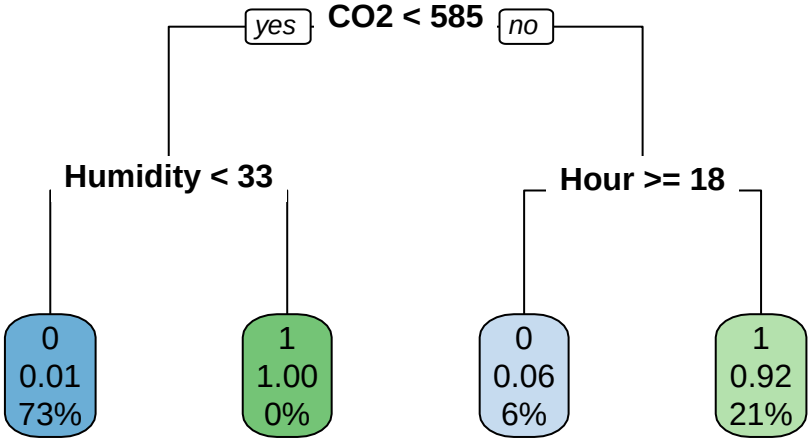



```
occupancy.tree.smaller <- rpart(factor(Occupancy) ~ Temperature + Humidity + CO2 + Hour,
                        data = occupancy_train,
                        method = "class",
                        control = rpart.control(maxdepth=2))
occupancy.pred.tree.smaller <- predict(occupancy.tree.smaller, occupancy_test, type = "class")
rpart.plot(occupancy.tree.smaller,
           type = 0)
```



slightly smaller

```
occupancy.tree.slightsmaller <- rpart(factor(Occupancy) ~ Temperature + Humidity + CO2 + Hour,
                         data = occupancy_train,
                         method = "class",
                         control = rpart.control(maxdepth=3))
occupancy.pred.tree.slightsmaller <- predict(occupancy.tree.slightsmaller, occupancy_test, type = "clas
rpart.plot(occupancy.tree.slightsmaller,
           type = 0)
```



From plots that are generated by the R, we can clearly see that graphs for tree with maximum depth of 2 and 3 are exactly same. So we only need to apply one of them in the further testing since the R think a tree with maximum depth of 2 is same as a tree with maximum depth of 3.

In order to see that whether the original tree is apporiate or not, we can do the in sample test for each tree to see their error rate as follows:

```
pred.insample.smaller.tree <- predict(occupancy.tree.smaller, occupancy_train, type = "class")
table(pred.insample.smaller.tree, occupancy_train$Occupancy)
```

```
##
## pred.insample.smaller.tree    0    1
##                          0 4404   74
##                          1   93 1129
```

```
pred.insample.tiny.tree <- predict(occupancy.tree.tiny, occupancy_train, type = "class")
table(pred.insample.tiny.tree, occupancy_train$Occupancy)
```

```
##
## pred.insample.tiny.tree    0    1
##                        0 4090   71
##                        1  407 1132
```

and the error rate for the in sample test are:

```
(93+74)/(4404+74+93+1129)
```

```
## [1] 0.02929825
```

```
(407+71)/(4090+71+407+1132)
```

```
## [1] 0.08385965
```

Testing for the testing data are

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

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

```
table(occupancy.pred.tree.slightsmaller, occupancy_test$Occupancy)
```

```
##
## occupancy.pred.tree.slightsmaller    0    1
##                                   0 1869   30
##                                   1   48  496
```

```
table(occupancy.pred.tree.smaller, occupancy_test$Occupancy)
```

```
##
## occupancy.pred.tree.smaller    0    1
##                            0 1869   30
##                            1   48  496
```

```
table(occupancy.pred.tree.tiny, occupancy_test$Occupancy)
```

```
##
## occupancy.pred.tree.tiny    0    1
##                         0 1739   35
##                         1  178  491
```

the error rate for the testing data are:

```
(15+34)/(1883+15+34+511)
```

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

```
(48+30)/2443
```

```
## [1] 0.03192796
```

```
(48+30)/2443
```

```
## [1] 0.03192796
```

```
(178+35)/2443
```

```
## [1] 0.08718788
```

From the result of both the in sample data test and the testing data that are computed by R, we can see that the overall error rate for the original classification tree is 0.02, which is the lowest error rate in the whole paper. Therefore it is safe to say that the over fitting problem might not seems to exist in this situation.

Final Recommendation

From the final predicted-value computed by R, we can tell that the classification tree model with maximum depth of 4 seems to have the lowest error rate, which is about 0.02, and the highest error rate in among those models is 0.08882521 from logistic regression model. As a result, we recommend the use of classification tree with maximum depth of 4. If it is too complicated, the classification tree with maximum depth of 2 or 3 can be used, as it has the second lowest error rate of 0.032.

Discussion

Overall, all of the four models did good in predicting the occupancy of the room, and it is surprising that the classification tree has the best performance among all other models.

However, there are still some problems in this paper that can be done better in the future. For example is the multicollinearity problem that happens between predictors/classifiers. If that problem do exist, it can affect that the accuracy and the precision of the model. So the error rate for models in this paper can be improved in the future.

Besides, the distribution of predictors/classifiers are not normally distributed. Though both the LDA and QDA did quite good job in making prediction, using transformed data that is normally distributed can help to make better prediction in those two model.

There might still be some potential problems that are not discussed in this section. But I still believe that one day it is possible to build a more advance model to eliminate those problems.