Occupancy_Project_Report

LH

11/8/2019

1. Introduction

Original occupancy data is retrieved from the UCI Machine Learning Repository at https://archive.ics.uci.edu/ml/machine-learning-databases/00357/occupancy_data.zip. There are 3 datasets in the zip file. One is training set and other two are test sets. The project goal is to detect occupancy status of room based on attributes such as date, temperature, humidity, CO2, light, and humidityratio. Occupancy variable has 0 and 1 values that represents not occupied and occupied respectively. In this report, I will use training data set for building model and test model on the first test set, the second test set(test2) will be used as my validaiton data set in the results section. Top two highest accuracy models will be recommanded at the end of the report. Three data sets along with a R script, a Rmd file and a pdf report are in the sampe folder at githut: https://github.com/lmhuvt/occupancy.

2. Methods/Analysis

2.1 Load library

```r
# load library, if not installed, install them
if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org")
if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")
if(!require(data.table)) install.packages("data.table", repos = "http://cran.us.r-project.org")
if(!require(lubridate)) install.packages("lubridate", repos = "http://cran.us.r-project.org")
if(!require(ggplot2)) install.packages("ggplot2", repos = "http://cran.us.r-project.org")
if(!require(randomForest)) install.packages("randomForest", repos = "http://cran.us.r-project.org")
if(!require(rpart)) install.packages("rpart", repos = "http://cran.us.r-project.org")
if(!require(purrr)) install.packages("purrr", repos = "http://cran.us.r-project.org")
library(lubridate)
library(ggplot2)
library(tidyverse)
library(caret)
library(randomForest)
library(rpart)
library(data.table)
library(purrr)
```

2.2 Load data through relative path, all project related files are in github: https://github.com/lmhuvt/occupancy

```
data_training <- read.table("./datatraining.txt",header=TRUE,sep=",")
data_testing <- read.table("./datatest.txt",header=TRUE,sep=",")
data_testing2 <- read.table("./datatest2.txt",header=TRUE,sep=",")
```

2.3 Review data set

2.3.1 Structure of data sets

```
str(data_training) # 8143 obs and 7 varibles
```

```
## 'data.frame':    8143 obs. of  7 variables:
##  $ date         : Factor w/ 8143 levels "2015-02-04 17:51:00",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ Temperature  : num  23.2 23.1 23.1 23.1 23.1 ...
##  $ Humidity     : num  27.3 27.3 27.2 27.2 27.2 ...
##  $ Light        : num  426 430 426 426 426 ...
##  $ CO2          : num  721 714 714 708 704 ...
##  $ HumidityRatio: num  0.00479 0.00478 0.00478 0.00477 0.00476 ...
##  $ Occupancy    : int  1 1 1 1 1 1 1 1 1 1 ...
```

```
str(data_testing) # 2665 obs and 7 varibles
```

```
## 'data.frame':    2665 obs. of  7 variables:
##  $ date         : Factor w/ 2665 levels "2015-02-02 14:19:00",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ Temperature  : num  23.7 23.7 23.7 23.7 23.8 ...
##  $ Humidity     : num  26.3 26.3 26.2 26.1 26.2 ...
##  $ Light        : num  585 578 573 494 489 ...
##  $ CO2          : num  749 760 770 775 779 ...
##  $ HumidityRatio: num  0.00476 0.00477 0.00477 0.00474 0.00477 ...
##  $ Occupancy    : int  1 1 1 1 1 1 1 1 1 1 ...
```

```
str(data_testing2) # 9752 obs and 7 varibles
```

```
## 'data.frame':    9752 obs. of  7 variables:
##  $ date         : Factor w/ 9752 levels "2015-02-11 14:48:00",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ Temperature  : num  21.8 21.8 21.8 21.8 21.8 ...
##  $ Humidity     : num  31.1 31 31.1 31.1 31.1 ...
##  $ Light        : num  437 437 434 439 437 ...
##  $ CO2          : num  1030 1000 1004 1010 1006 ...
##  $ HumidityRatio: num  0.00502 0.00501 0.00502 0.00502 0.00503 ...
##  $ Occupancy    : int  1 1 1 1 1 1 1 1 1 1 ...
```

Overall, from data review, all data set are data frames, there are 6 predictors and 1 outcome. 6 predictors: Date is factor, Temperature, Humidity, Light, CO2, HumidityRatio are numbers. 1 outcome "Occupancy" is a binary value, 1 was occupied and 0 was not occupied.

2.3.2 Occupancy distribution in the data sets

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

```
##
##         0         1
## 0.7876704 0.2123296
```

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

```
##
##        0        1
## 0.635272 0.364728
```

```
prop.table(table(data_testing2$Occupancy))
```

```
##
##         0         1
## 0.7898893 0.2101107
```

Table shows probability of occupancy "0"(unoccupied) is higher than occupancy "1"(occupied)

2.3.3 Check missing(NA) data

```
sum(is.na(data_training)) # no missing data
```

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

```
sum(is.na(data_testing)) # no missing data
```

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

```
sum(is.na(data_testing2)) # no missing data
```

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

2.3.4 Check how many distinct predictors

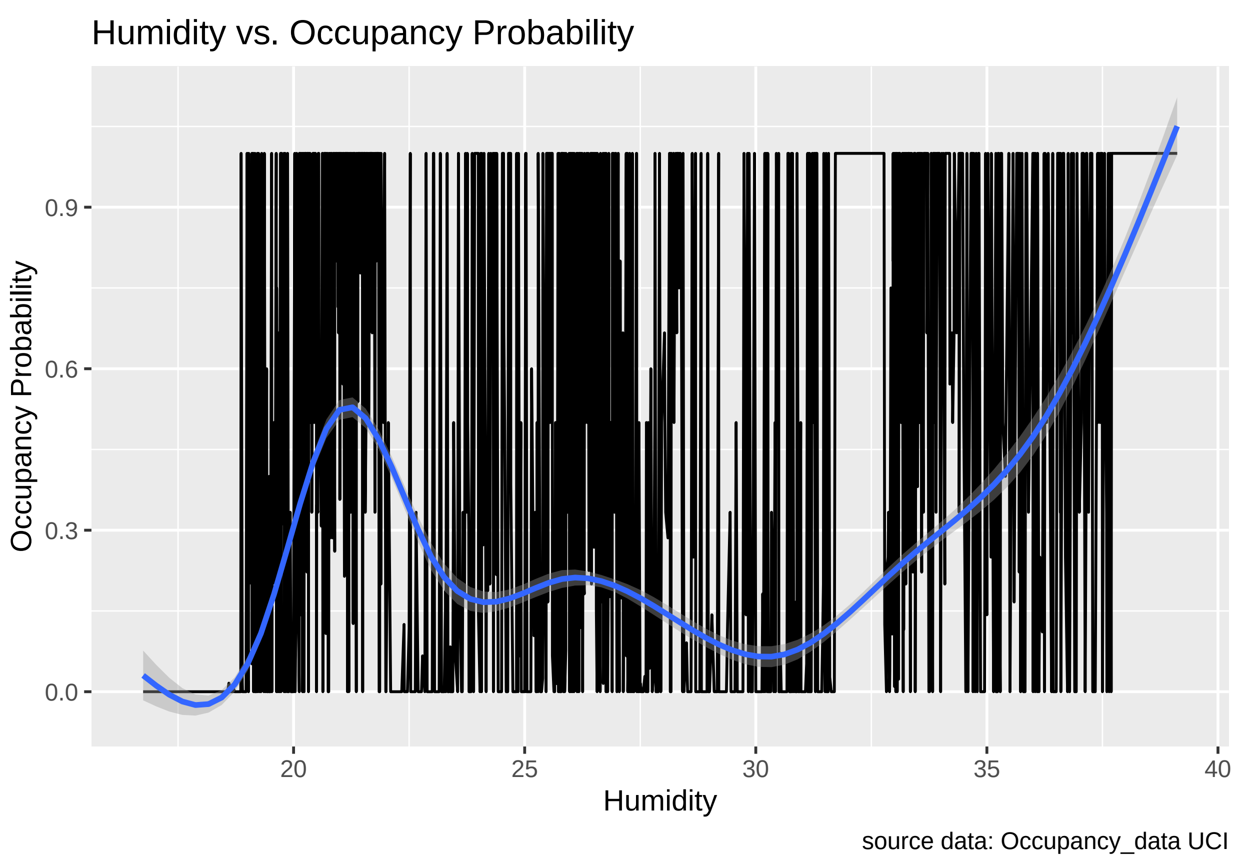
```
n_predictor <- data.frame(n_Humidity = n_distinct(data_training$Humidity),
                          n_HumidityRatio = n_distinct(data_training$HumidityRatio),
                          n_Temperature = n_distinct(data_training$Temperature),
                          n_Light = n_distinct(data_training$Light),
                          n_CO2 = n_distinct(data_training$CO2))
n_predictor
```

```
##   n_Humidity n_HumidityRatio n_Temperature n_Light n_CO2
## 1       1325            3583           265     889  2282
```

2.4 Visulization predictors effect

2.4.1 Humidity effect

```
t1 <- data_training %>%
  group_by(Humidity)%>%
  mutate(prob=mean(Occupancy == "1"))%>%
  select(Humidity,prob)
t1 %>% ggplot(aes(x=Humidity, y=prob))+geom_line()+geom_smooth()+
  labs(x="Humidity", y="Occupancy Probability",
       caption = "source data: Occupancy_data UCI")+
  ggtitle("Humidity vs. Occupancy Probability")
```



It is hard to interpret Humidity plot because there are too many points, added line and soomth fuction to see correlation between Humidity and occupancy probability.

```
# calculate correlation
correlation_Humidity1 <- data_training %>% select(Humidity, Occupancy)%>%
  summarize(c_Humidity1= cor(Humidity, Occupancy, method = "spearman"))%>%
```

```
  pull(c_Humidity1)
correlation_Humidity1
```

```
## [1] 0.1292351
```

```
correlation_Humidity2 <-data_training %>% select(Humidity, Occupancy)%>%
  summarize(c_Humidity2= cor(Humidity, Occupancy, method = "pearson"))%>%
  pull(c_Humidity2)
correlation_Humidity2
```

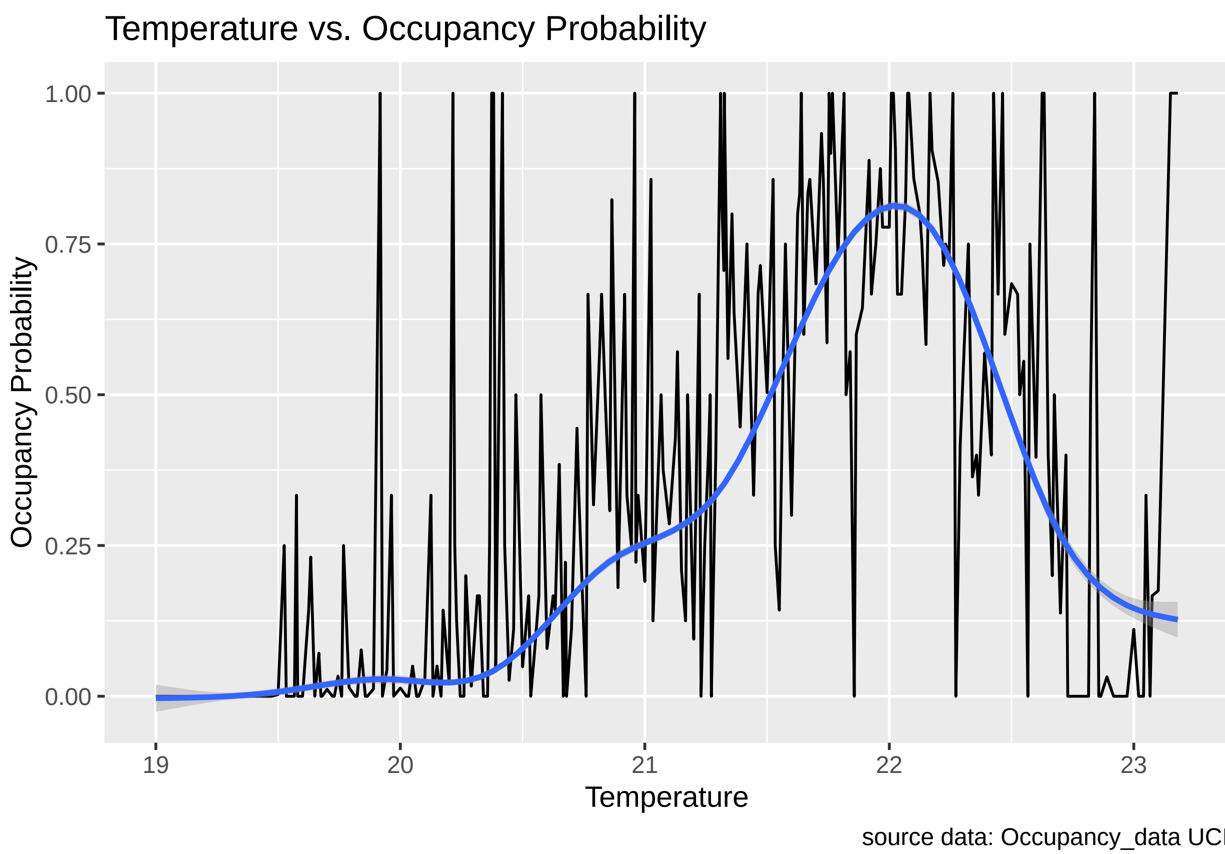
```
## [1] 0.1329642
```

```
correlation_results <- tibble(predictor = "Humidity",
                              spearman = correlation_Humidity1,
                              pearson = correlation_Humidity2)
```

Calculaiton confirmed correlation between Humidity and occupancy probability.

2.4.2 Temperature effect

```
t2 <- data_training %>%
  #mutate(round_temperature = round(Temperature))%>%
  group_by(Temperature)%>%
  mutate(prob=mean(Occupancy == "1"))%>%
  select(Temperature,prob)
t2 %>% ggplot(aes(x=Temperature, y=prob))+geom_line()+geom_smooth()+
  labs(x="Temperature", y="Occupancy Probability",
       caption = "source data: Occupancy_data UCI")+
  ggtitle("Temperature vs. Occupancy Probability")
```



Temperature plot showed correlation between Temperature and occupancy probability.

```
# calculate correlation
correlation_Temperature1 <- data_training %>% select(Temperature, Occupancy)%>%
  summarize(c_Temperature1= cor(Temperature, Occupancy, method = "spearman"))%>%
  pull(c_Temperature1)
correlation_Temperature1
```

```
## [1] 0.5328303
```

```
Correlation_Temperature2 <- data_training %>% select(Temperature, Occupancy)%>%
  summarize(c_Temperature2= cor(Temperature, Occupancy, method = "pearson"))%>%
  pull(c_Temperature2)
Correlation_Temperature2
```

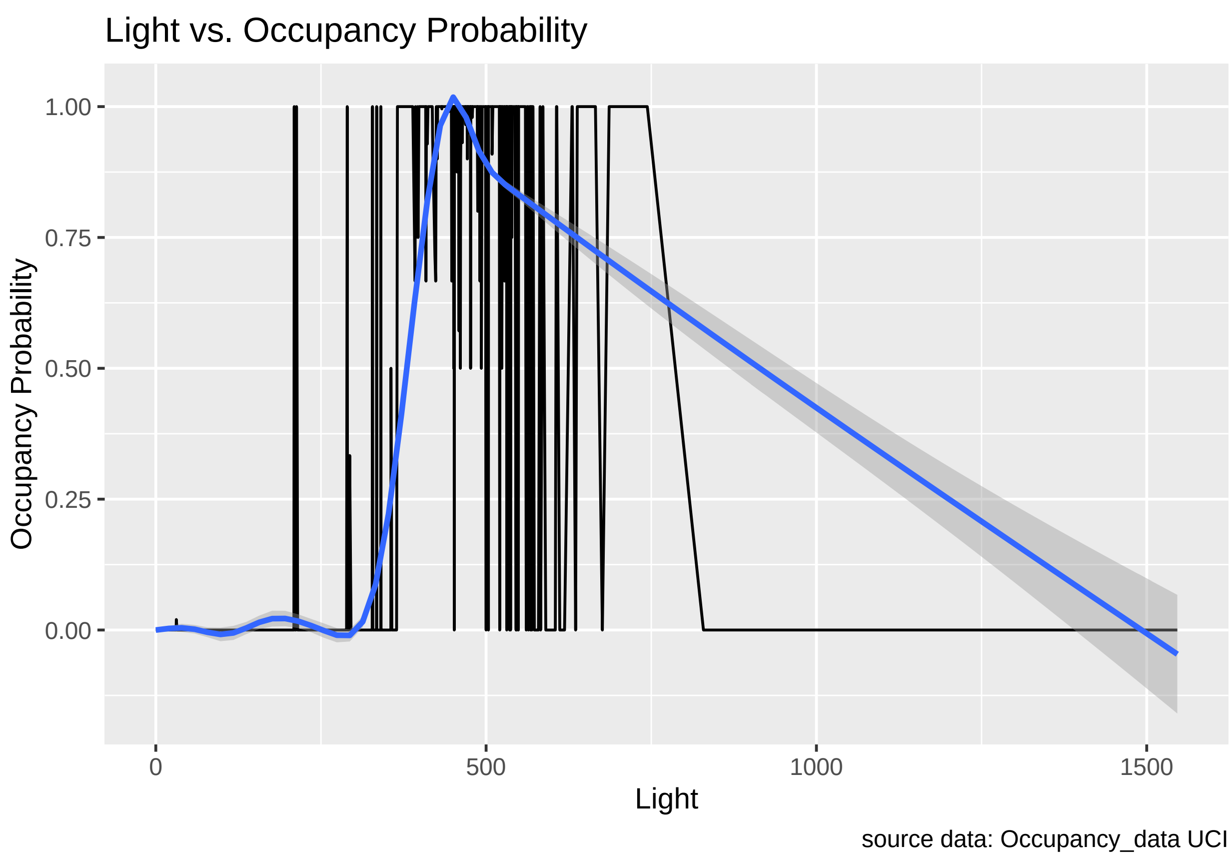
```
## [1] 0.5382197
```

```
correlation_results <- bind_rows(correlation_results,
                                 tibble(predictor ="Temperature",
                                        spearman = correlation_Temperature1,
                                        pearson = Correlation_Temperature2))
```

Calculaiton confirmed correlation between Temperature and occupancy probability.

2.4.3 Light effect

```
t3 <- data_training %>%
  group_by(Light)%>%
  mutate(prob=mean(Occupancy == "1"))%>%
  select(Light,prob)
t3 %>% ggplot(aes(x=Light, y=prob))+geom_line()+geom_smooth()+
  labs(x="Light", y="Occupancy Probability",
       caption = "source data: Occupancy_data UCI")+
  ggtitle("Light vs. Occupancy Probability")
```



It is hard to interpret Light plot because there are too many points, added line and soomth fuction to see correlation between Light and occupancy probability.

```
# calculate correlation
correlation_Light1 <- data_training %>% select(Light, Occupancy)%>%
  summarize(c_Light1= cor(Light, Occupancy, method = "spearman"))%>%
  pull(c_Light1)
correlation_Light1
```

```
## [1] 0.8046454
```

```
correlation_Light2 <- data_training %>% select(Light, Occupancy)%>%
  summarize(c_Light2= cor(Light, Occupancy, method = "pearson"))%>%
  pull(c_Light2)
correlation_Light2
```

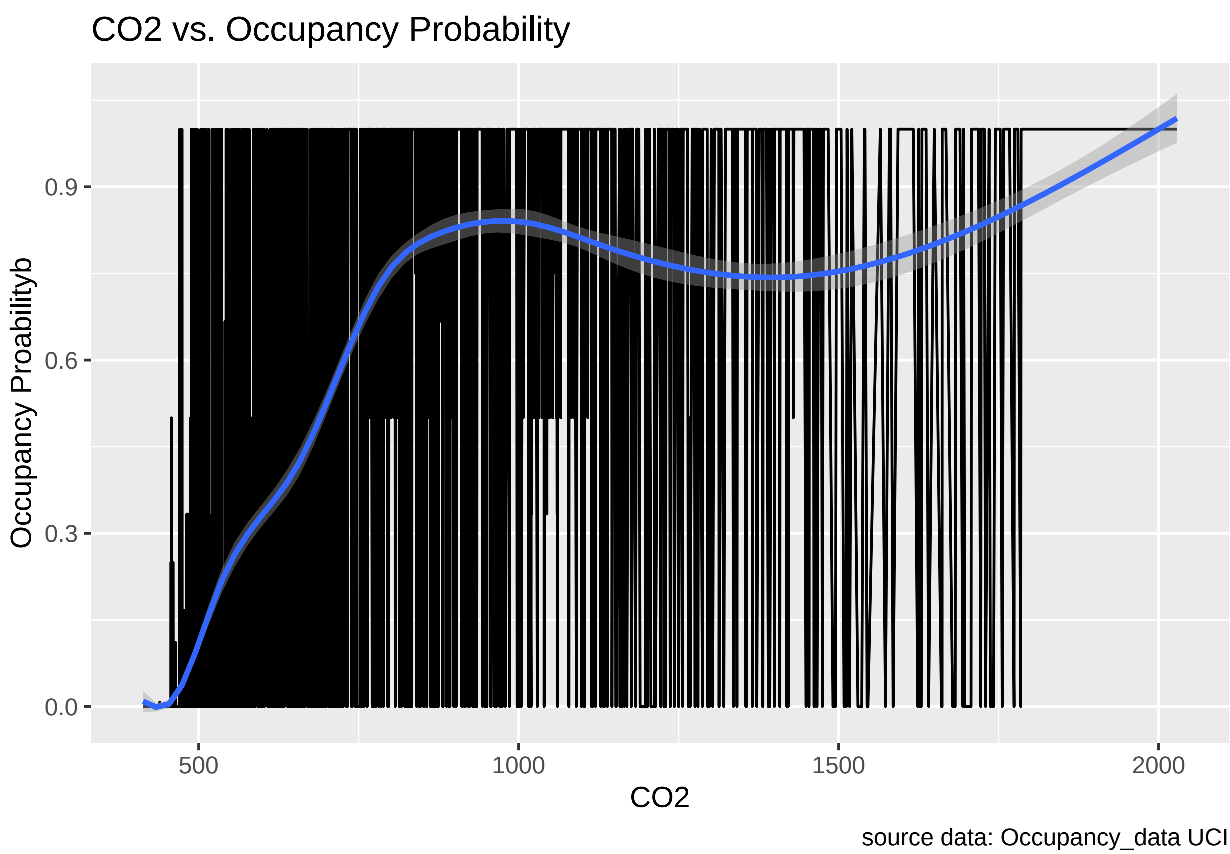
```
## [1] 0.9073521
```

```
correlation_results <- bind_rows(correlation_results,
                                 tibble(predictor ="Light",
                                        spearman = correlation_Light1,
                                        pearson = correlation_Light2))
```

Calculation confrimed strong correlation between Light and Occupancy.

2.4.4 CO2 effect

```
t4 <- data_training %>%
  group_by(CO2)%>%
  mutate(prob=mean(Occupancy == "1"))%>%
  select(CO2,prob)
t4 %>% ggplot(aes(x=CO2, y=prob))+geom_line()+geom_smooth()+
  labs(x="CO2", y="Occupancy Proabilityb",
       caption = "source data: Occupancy_data UCI")+
  ggtitle("CO2 vs. Occupancy Probability")
```



It is hard to interpret CO2 plot because there are too many points, added line and soomth fuction to see correlation between CO2 and occupancy probability.

```
# calculate correlation
correlation_CO2_1 <- data_training %>% select(CO2, Occupancy)%>%
  summarize(c_CO2_1= cor(CO2, Occupancy, method = "spearman"))%>%
  pull(c_CO2_1)
correlation_CO2_1
```

```
## [1] 0.6566513
```

```
correlation_CO2_2 <- data_training %>% select(CO2, Occupancy)%>%
  summarize(c_CO2_2= cor(CO2, Occupancy, method = "pearson"))%>%
  pull(c_CO2_2)
correlation_CO2_2
```

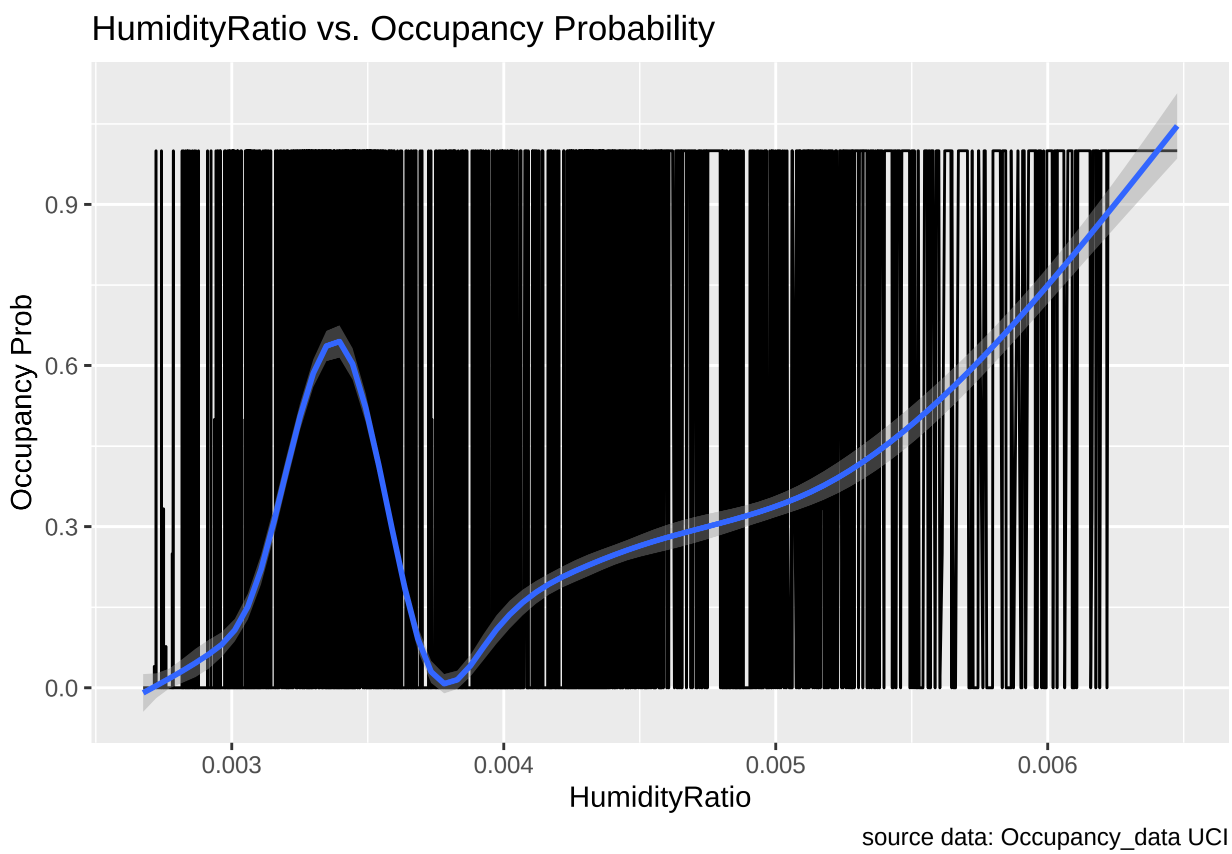
```
## [1] 0.7122352
```

```
correlation_results <- bind_rows(correlation_results,
                                 tibble(predictor ="CO2",
                                        spearman = correlation_CO2_1,
                                        pearson = correlation_CO2_2))
```

Calculation confirmed correlation between CO2 and Occupancy.

2.4.5 HumidityRatio effect

```
t5 <- data_training %>%
  group_by(HumidityRatio)%>%
  mutate(prob=mean(Occupancy == "1"))%>%
  select(HumidityRatio,prob)
t5 %>% ggplot(aes(x=HumidityRatio, y=prob))+geom_line()+geom_smooth()+
  labs(x="HumidityRatio", y="Occupancy Prob",
       caption = "source data: Occupancy_data UCI")+
  ggtitle("HumidityRatio vs. Occupancy Probability")
```



It is hard to interpret HumidityRatio plot because there are too many points, added line and soomth fuction to see correlation between Humidityratio and occupancy probability.

```
# calcualte correlation
correlation_HR1 <- data_training %>% select(HumidityRatio, Occupancy)%>%
  summarize(c_HR1= cor(HumidityRatio, Occupancy, method = "spearman"))%>%
  pull(c_HR1)
correlation_HR1
```

```
## [1] 0.255836
```

```
correlation_HR2 <- data_training %>% select(HumidityRatio, Occupancy)%>%
  summarize(c_HR2= cor(HumidityRatio, Occupancy, method = "pearson"))%>%
  pull(c_HR2)
correlation_HR2
```

```
## [1] 0.3002816
```

```
correlation_results <- bind_rows(correlation_results,
                                 tibble(predictor ="HumidityRatio",
                                        spearman = correlation_HR1,
                                        pearson = correlation_HR2))
```

Calculation confirmed correlation between Humidityratio and Occupancy.

2.4.6 Summary of predictor correlation

```
options(pillar.sigfig = 4) # keep 4 significant figures in table
correlation_results
```

```
## # A tibble: 5 x 3
##   predictor     spearman pearson
##   <chr>            <dbl>   <dbl>
## 1 Humidity        0.1292  0.1330
## 2 Temperature     0.5328  0.5382
## 3 Light           0.8046  0.9074
## 4 CO2             0.6567  0.7122
## 5 HumidityRatio   0.2558  0.3003
```

5 plots shows correlation between numeric predictors and outcome(Occupancy). From strongest to weakest correlation: Light > CO2 > Temperature > HumidityRatio > Humidity

2.4.7 Date effect

Date is not a numeric factor, it needs further data cleaning and processng. In order to look into date effect, I will convert outcome occupancy to factor for data process.

```
data_training$Occupancy <- as.factor(data_training$Occupancy)
data_testing$Occupancy <- as.factor(data_testing$Occupancy)
data_testing2$Occupancy <- as.factor(data_testing2$Occupancy)
# make copies of all data set without changing the original data sets
data_training_m <- copy(data_training)
data_testing_m <- copy(data_testing)
data_testing2_m <- copy(data_testing2)
# take a look of date/time effect, timestamp need to covert to a easy process format
data_training_m$date <- as.POSIXct(data_training_m$date,tz="UTC")
data_testing_m$date <- as.POSIXct(data_testing_m$date,tz="UTC")
data_testing2_m$date <- as.POSIXct(data_testing2_m$date,tz="UTC")
# weekday and weekend are supposed to have different occupancy
# I need to convert date into a format which can be esay to process
# x is POSIXct format timestamp
weekend_weekday <- function(x) {
  val <- weekdays(x)
  if (val == "Saturday" | val == "Sunday") {
    val2 = "Weekend"
  }
  else {
    val2= "Weekday"
  }
  return(val2)
}
# for ploting purpose, 0 repersents weekend, 1 repersents weekday
# function to convert character weekday/weekend into numeric
```

```r
Relevel_weekend <- function(y) {
  if (y == "Weekend") {
    val2 = 0
  }
  else {
    val2= 1
  }
  return(val2)
}
# add weekday/weekend column into copy data set
data_training_m$WeekStatus <-unlist(lapply(data_training_m$date,
                                           weekend_weekday))
data_testing_m$WeekStatus <-unlist(lapply(data_testing_m$date,
                                          weekend_weekday))
data_testing2_m$WeekStatus <-unlist(lapply(data_testing2_m$date,
                                           weekend_weekday))
# add WeekStatus2 column into copy data set, use "1" repersent weekday
# use "0" repersent weekend
data_training_m$WeekStatus2 <- unlist(lapply(data_training_m$WeekStatus,
                                             Relevel_weekend))
data_testing_m$WeekStatus2 <- unlist(lapply(data_testing_m$WeekStatus,
                                            Relevel_weekend))
data_testing2_m$WeekStatus2 <- unlist(lapply(data_testing2_m$WeekStatus,
                                             Relevel_weekend))
str(data_training_m)
```

```
## 'data.frame':    8143 obs. of  9 variables:
##  $ date         : POSIXct, format: "2015-02-04 17:51:00" "2015-02-04 17:51:59" ...
##  $ Temperature  : num  23.2 23.1 23.1 23.1 23.1 ...
##  $ Humidity     : num  27.3 27.3 27.2 27.2 27.2 ...
##  $ Light        : num  426 430 426 426 426 ...
##  $ CO2          : num  721 714 714 708 704 ...
##  $ HumidityRatio: num  0.00479 0.00478 0.00478 0.00477 0.00476 ...
##  $ Occupancy    : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  $ WeekStatus   : chr  "Weekday" "Weekday" "Weekday" "Weekday" ...
##  $ WeekStatus2  : num  1 1 1 1 1 1 1 1 1 1 ...
```

```r
str(data_testing_m)
```

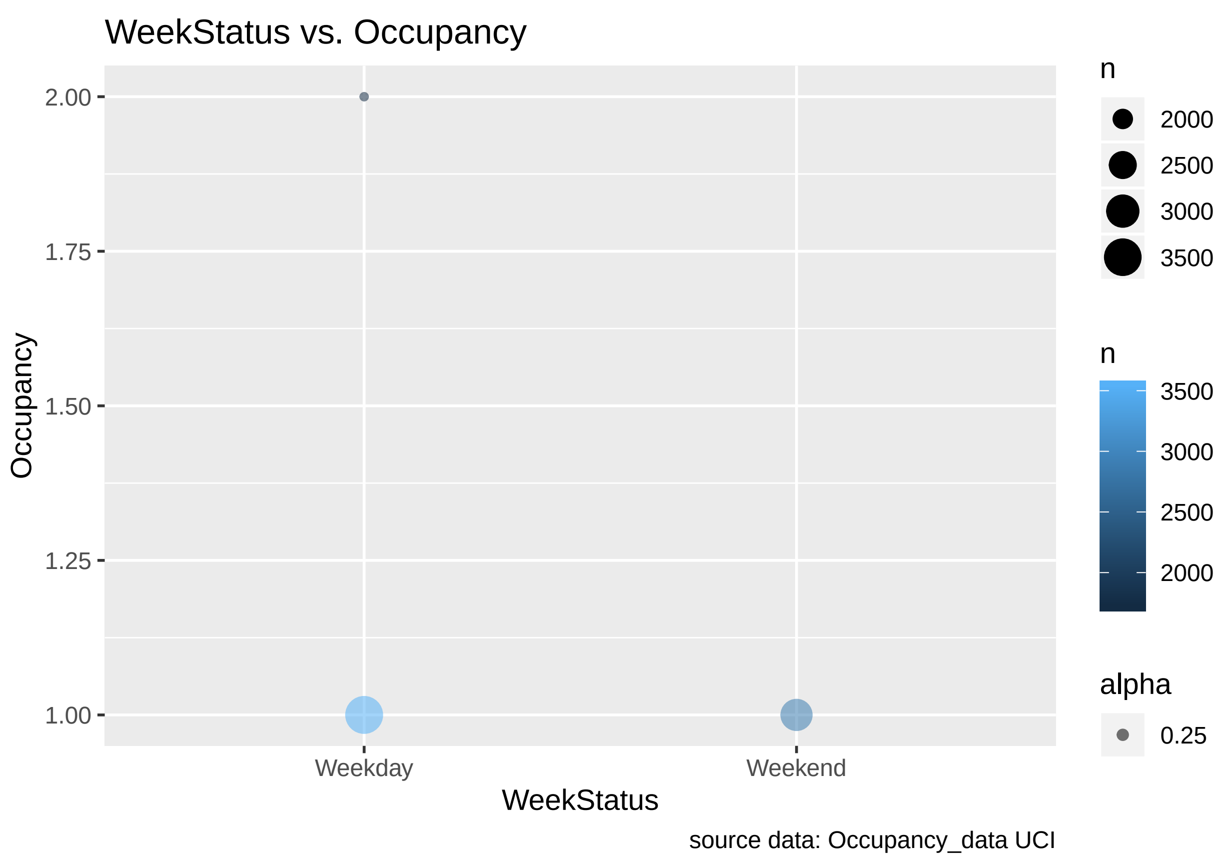
```
## 'data.frame':    2665 obs. of  9 variables:
##  $ date         : POSIXct, format: "2015-02-02 14:19:00" "2015-02-02 14:19:59" ...
##  $ Temperature  : num  23.7 23.7 23.7 23.7 23.8 ...
##  $ Humidity     : num  26.3 26.3 26.2 26.1 26.2 ...
##  $ Light        : num  585 578 573 494 489 ...
##  $ CO2          : num  749 760 770 775 779 ...
##  $ HumidityRatio: num  0.00476 0.00477 0.00477 0.00474 0.00477 ...
##  $ Occupancy    : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  $ WeekStatus   : chr  "Weekday" "Weekday" "Weekday" "Weekday" ...
##  $ WeekStatus2  : num  1 1 1 1 1 1 1 1 1 1 ...
```

```r
str(data_testing2_m)
```

```
## 'data.frame':    9752 obs. of  9 variables:
##  $ date         : POSIXct, format: "2015-02-11 14:48:00" "2015-02-11 14:49:00" ...
##  $ Temperature  : num  21.8 21.8 21.8 21.8 21.8 ...
##  $ Humidity     : num  31.1 31 31.1 31.1 31.1 ...
##  $ Light        : num  437 437 434 439 437 ...
##  $ CO2          : num  1030 1000 1004 1010 1006 ...
##  $ HumidityRatio: num  0.00502 0.00501 0.00502 0.00502 0.00503 ...
##  $ Occupancy    : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  $ WeekStatus   : chr  "Weekday" "Weekday" "Weekday" "Weekday" ...
##  $ WeekStatus2  : num  1 1 1 1 1 1 1 1 1 1 ...
```

Plot correlation between date and occupancy

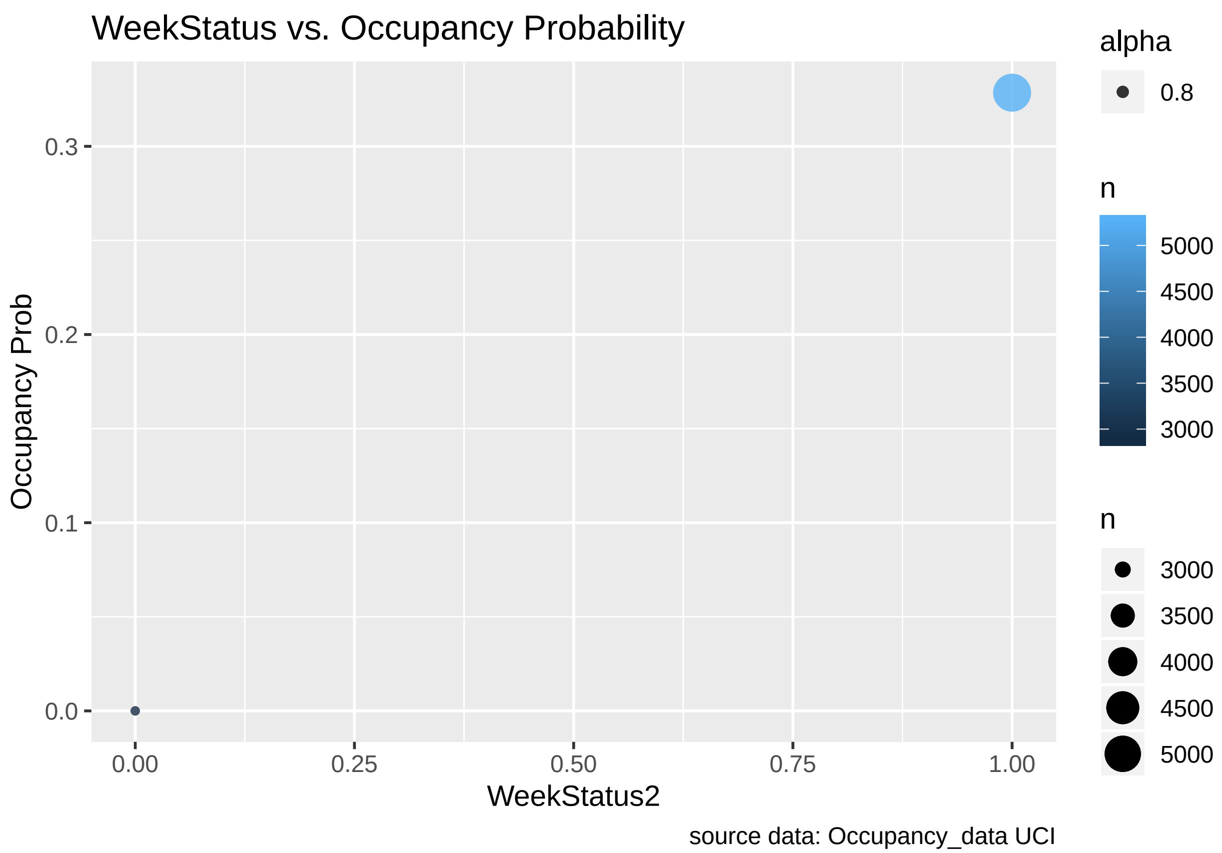
```r
plot_date <- data_training_m %>%
  ggplot(aes(x= WeekStatus, y = as.numeric(Occupancy)))+
  geom_count(aes(alpha=0.25,color= ..n.., size = ..n..))+
  labs(x="WeekStatus", y="Occupancy", caption = "source data: Occupancy_data UCI")+
  ggtitle("WeekStatus vs. Occupancy")
plot_date
```



The plot_date showed correlation between date and Occupancy.

Caculate occupancy probability and plot correlation, the probability of occupancy might be easier to see the correlation.

```
data_date <- data_training_m %>%
  group_by(WeekStatus2)%>%
  mutate(prob=mean(Occupancy == "1"))%>%
  select(WeekStatus2,prob)
data_date %>% ggplot(aes(x=WeekStatus2, y=prob))+
  geom_count(aes(alpha=0.8,color= ..n.., size = ..n..))+
  geom_count(aes(alpha=0.8,color= ..n.., size = ..n..))+
  labs(x="WeekStatus2", y="Occupancy Prob",
       caption = "source data: Occupancy_data UCI")+
  ggtitle("WeekStatus vs. Occupancy Probability")
```



The occupancy probability plot showed correlation between date and occupancy.

Summary of date effect: date and probability plots both showed correlation between date and occupancy, and the probability plot is better to show the correlation.

2.4 Modeling approach

Build model based on training data set and test model in test data set, test2 data will be used as a validation data set in results section. date is factor, and the rest of predictors are numeric.I will remove date out of data sets to simplify model training.

```
data_training_1 <- subset(data_training_m,
                          select = c("Temperature", "Humidity", "Light",
                                     "CO2", "HumidityRatio",
                                     "Occupancy"))
data_testing_1 <- subset(data_testing_m,
                         select = c("Temperature", "Humidity", "Light",
                                    "CO2", "HumidityRatio",
                                    "Occupancy"))
data_testing2_1 <- subset(data_testing2_m,
                          select = c("Temperature", "Humidity", "Light",
                                     "CO2", "HumidityRatio",
                                     "Occupancy"))
# check all data sets
str(data_training_1)
```

```
## 'data.frame':    8143 obs. of  6 variables:
##  $ Temperature  : num  23.2 23.1 23.1 23.1 23.1 ...
##  $ Humidity     : num  27.3 27.3 27.2 27.2 27.2 ...
##  $ Light        : num  426 430 426 426 426 ...
##  $ CO2          : num  721 714 714 708 704 ...
##  $ HumidityRatio: num  0.00479 0.00478 0.00478 0.00477 0.00476 ...
##  $ Occupancy    : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
```

```
str(data_testing_1)
```

```
## 'data.frame':    2665 obs. of  6 variables:
##  $ Temperature  : num  23.7 23.7 23.7 23.7 23.8 ...
##  $ Humidity     : num  26.3 26.3 26.2 26.1 26.2 ...
##  $ Light        : num  585 578 573 494 489 ...
##  $ CO2          : num  749 760 770 775 779 ...
##  $ HumidityRatio: num  0.00476 0.00477 0.00477 0.00474 0.00477 ...
##  $ Occupancy    : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
```

```
str(data_testing2_1)
```

```
## 'data.frame':    9752 obs. of  6 variables:
##  $ Temperature  : num  21.8 21.8 21.8 21.8 21.8 ...
##  $ Humidity     : num  31.1 31 31.1 31.1 31.1 ...
##  $ Light        : num  437 437 434 439 437 ...
##  $ CO2          : num  1030 1000 1004 1010 1006 ...
##  $ HumidityRatio: num  0.00502 0.00501 0.00502 0.00502 0.00503 ...
##  $ Occupancy    : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
```

New data sets show 6 varibles including one outcome and 5 predictors. qda, knn, rpart, rf model will be used on traing and test data set.

2.4.1 qda model

```
# Temperature
set.seed(1, sample.kind = "Rounding")
train_qda_Temperature <- train(Occupancy~ Temperature,
                               method ="qda", data = data_training_1)
set.seed(1, sample.kind = "Rounding")
accuracy_qda_Temperature_train <-confusionMatrix(predict(train_qda_Temperature,
                                                         data_training_1),
                                                 data_training_1$Occupancy
                                                 )$overall["Accuracy"]
set.seed(1, sample.kind = "Rounding")
accuracy_qda_Temperature_test <-confusionMatrix(predict(train_qda_Temperature,
                                                        data_testing_1),
                                                data_testing_1$Occupancy
                                                )$overall["Accuracy"]
options(pillar.sigfig = 7) # accuracy reulsts have 7 significant figures
accuracy_results <- tibble(method = "qda",
                           predictor = "Temperature",
                           Accuracy_Train = accuracy_qda_Temperature_train,
                           Accuracy_Test = accuracy_qda_Temperature_test)
# Humidity
set.seed(1, sample.kind = "Rounding")
train_qda_Humidity <- train(Occupancy~Humidity,
                            method ="qda", data = data_training_1)
set.seed(1, sample.kind = "Rounding")
accuracy_qda_Humidity_train <-confusionMatrix(predict(train_qda_Humidity,
                                                       data_training_1),
                                               data_training_1$Occupancy
                                              )$overall["Accuracy"]
set.seed(1, sample.kind = "Rounding")
accuracy_qda_Humidity_test <-confusionMatrix(predict(train_qda_Humidity,
                                                      data_testing_1),
                                              data_testing_1$Occupancy
                                             )$overall["Accuracy"]
accuracy_results <- bind_rows(accuracy_results,
                              tibble(method = "qda",
                                     predictor = "Humidity",
                                     Accuracy_Train = accuracy_qda_Humidity_train,
                                     Accuracy_Test = accuracy_qda_Humidity_test))
# Light
set.seed(1, sample.kind = "Rounding")
train_qda_Light <- train(Occupancy~Light,
                         method ="qda", data = data_training_1)
set.seed(1, sample.kind = "Rounding")
accuracy_qda_Light_train <-confusionMatrix(predict(train_qda_Light,
                                                    data_training_1),
                                            data_training_1$Occupancy
                                            )$overall["Accuracy"]
set.seed(1, sample.kind = "Rounding")
```

```r
accuracy_qda_Light_test <-confusionMatrix(predict(train_qda_Light,
                                                  data_testing_1),
                                          data_testing_1$Occupancy
                                          )$overall["Accuracy"]
accuracy_results <- bind_rows(accuracy_results,
                              tibble(method = "qda",
                                     predictor = "Light",
                                     Accuracy_Train = accuracy_qda_Light_train,
                                     Accuracy_Test = accuracy_qda_Light_test))
# CO2
set.seed(1, sample.kind = "Rounding")
train_qda_CO2 <- train(Occupancy~CO2,
                       method ="qda", data = data_training_1)
set.seed(1, sample.kind = "Rounding")
accuracy_qda_CO2_train <- confusionMatrix(predict(train_qda_CO2,
                                                  data_training_1),
                                          data_training_1$Occupancy
                                          )$overall["Accuracy"]
set.seed(1, sample.kind = "Rounding")
accuracy_qda_CO2_test <- confusionMatrix(predict(train_qda_CO2,
                                                 data_testing_1),
                                         data_testing_1$Occupancy
                                         )$overall["Accuracy"]
accuracy_results <- bind_rows(accuracy_results,
                              tibble(method = "qda",
                                     predictor = "CO2",
                                     Accuracy_Train = accuracy_qda_CO2_train,
                                     Accuracy_Test =accuracy_qda_CO2_test))
# HumidityRatio
set.seed(1, sample.kind = "Rounding")
train_qda_HumidityRatio <- train(Occupancy~HumidityRatio,
                                 method ="qda", data = data_training_1)
set.seed(1, sample.kind = "Rounding")
accuracy_qda_HumidityRatio_train <-confusionMatrix(predict(train_qda_HumidityRatio,
                                                           data_training_1),
                                                   data_training_1$Occupancy
                                                   )$overall["Accuracy"]
set.seed(1, sample.kind = "Rounding")
accuracy_qda_HumidityRatio_test <-confusionMatrix(predict(train_qda_HumidityRatio,
                                                          data_testing_1),
                                                  data_testing_1$Occupancy
                                                  )$overall["Accuracy"]
accuracy_results <- bind_rows(accuracy_results,
                              tibble(method = "qda",
                                     predictor = "HumidityRatio",
                                     Accuracy_Train = accuracy_qda_HumidityRatio_train,
                                     Accuracy_Test = accuracy_qda_HumidityRatio_test))
# all
set.seed(1, sample.kind = "Rounding")
train_qda <- train(Occupancy~.,
                   method ="qda", data = data_training_1)
varImp(train_qda) # Importance of different predictors
```

```
## ROC curve variable importance
##
##               Importance
## Light             100.00
## CO2                93.26
## Temperature        71.33
## HumidityRatio      22.39
## Humidity            0.00
```

```r
set.seed(1, sample.kind = "Rounding")
accuracy_qda_train <- confusionMatrix(predict(train_qda,data_training_1),
                                      data_training_1$Occupancy
                                      )$overall["Accuracy"]
set.seed(1, sample.kind = "Rounding")
accuracy_qda_test <-confusionMatrix(predict(train_qda,data_testing_1),
                                    data_testing_1$Occupancy
                                    )$overall["Accuracy"]
accuracy_results <- bind_rows(accuracy_results,
                              tibble(method = "qda",
                                     predictor = "All",
                                     Accuracy_Train = accuracy_qda_train,
                                     Accuracy_Test = accuracy_qda_test))
accuracy_results
```

```
## # A tibble: 6 x 4
##   method predictor     Accuracy_Train Accuracy_Test
##   <chr>  <chr>                  <dbl>         <dbl>
## 1 qda    Temperature        0.8419501     0.8487805
## 2 qda    Humidity           0.7896353     0.6352720
## 3 qda    Light              0.9772811     0.9771107
## 4 qda    CO2                0.9022473     0.8727955
## 5 qda    HumidityRatio      0.8155471     0.6915572
## 6 qda    All                0.9888248     0.9774859
```

Summary of qda model: The accuracy__results table shows qda model used all 5 predictors has hihgest accuracy in training set and first test set. The qda model with 5 predictors will be used on validation data set in results section.

```r
rm(accuracy_results) # remove table before next model training and testing
```

2.4.2 CART model

```r
# Temperature
set.seed(1, sample.kind = "Rounding")
train_rpart_Temperature <- train(Occupancy~Temperature, method = "rpart",
                                 data = data_training_1)
set.seed(1, sample.kind = "Rounding")
accuracy_rpart_Temperature_train <- confusionMatrix(predict(train_rpart_Temperature,
                                                            data_training_1),
```

```
                                                    data_training_1$Occupancy
                                                    )$overall["Accuracy"]
set.seed(1, sample.kind = "Rounding")
accuracy_rpart_Temperature_test <- confusionMatrix(predict(train_rpart_Temperature,
                                                           data_testing_1),
                                                   data_testing_1$Occupancy
                                                   )$overall["Accuracy"]
accuracy_results <- tibble(method = "rpart",
                           predictor = "Temperature",
                           Accuracy_Train = accuracy_rpart_Temperature_train,
                           Accuracy_Test =accuracy_rpart_Temperature_test)
# Humidity
set.seed(1, sample.kind = "Rounding")
train_rpart_Humidity <- train(Occupancy~Humidity, method = "rpart",
                                 data = data_training_1)
set.seed(1, sample.kind = "Rounding")
accuracy_rpart_Humidity_train <- confusionMatrix(predict(train_rpart_Humidity,
                                                         data_training_1),
                                                 data_training_1$Occupancy
                                                 )$overall["Accuracy"]
set.seed(1, sample.kind = "Rounding")
accuracy_rpart_Humidity_test <- confusionMatrix(predict(train_rpart_Humidity,
                                                        data_testing_1),
                                                data_testing_1$Occupancy
                                                )$overall["Accuracy"]
accuracy_results <- bind_rows(accuracy_results, tibble(method = "rpart",
                           predictor = "Humidity",
                           Accuracy_Train = accuracy_rpart_Humidity_train,
                           Accuracy_Test =accuracy_rpart_Humidity_test))
# Light
set.seed(1, sample.kind = "Rounding")
train_rpart_Light <- train(Occupancy~Light, method = "rpart",
                              data = data_training_1)
set.seed(1, sample.kind = "Rounding")
accuracy_rpart_Light_train <- confusionMatrix(predict(train_rpart_Light,
                                                      data_training_1),
                                              data_training_1$Occupancy
                                              )$overall["Accuracy"]
set.seed(1, sample.kind = "Rounding")
accuracy_rpart_Light_test <- confusionMatrix(predict(train_rpart_Light,
                                                     data_testing_1),
                                             data_testing_1$Occupancy
                                             )$overall["Accuracy"]
accuracy_results <- bind_rows(accuracy_results,
                              tibble(method = "rpart",
                                     predictor = "Light",
                                     Accuracy_Train = accuracy_rpart_Light_train,
                                     Accuracy_Test =accuracy_rpart_Light_test))
# CO2
set.seed(1, sample.kind = "Rounding")
train_rpart_CO2 <- train(Occupancy~CO2, method = "rpart",
                           data = data_training_1)
set.seed(1, sample.kind = "Rounding")
```

```r
accuracy_rpart_CO2_train <- confusionMatrix(predict(train_rpart_CO2,
                                                    data_training_1),
                                            data_training_1$Occupancy
                                            )$overall["Accuracy"]
set.seed(1, sample.kind = "Rounding")
accuracy_rpart_CO2_test <- confusionMatrix(predict(train_rpart_CO2,
                                                   data_testing_1),
                                           data_testing_1$Occupancy
                                           )$overall["Accuracy"]
accuracy_results <- bind_rows(accuracy_results,
                              tibble(method = "rpart",
                                     predictor = "CO2",
                                     Accuracy_Train = accuracy_rpart_CO2_train,
                                     Accuracy_Test =accuracy_rpart_CO2_test))
#HumidityRatio
set.seed(1, sample.kind = "Rounding")
train_rpart_HumidityRatio <- train(Occupancy~HumidityRatio, method = "rpart",
                         data = data_training_1)
set.seed(1, sample.kind = "Rounding")
accuracy_rpart_HumidityRatio_train <- confusionMatrix(predict(train_rpart_HumidityRatio,
                                                              data_training_1),
                                                      data_training_1$Occupancy
                                                      )$overall["Accuracy"]
set.seed(1, sample.kind = "Rounding")
accuracy_rpart_HumidityRatio_test <- confusionMatrix(predict(train_rpart_HumidityRatio,
                                                             data_testing_1),
                                                     data_testing_1$Occupancy
                                                     )$overall["Accuracy"]
set.seed(1, sample.kind = "Rounding")
accuracy_results <- bind_rows(accuracy_results,
                              tibble(method = "rpart",
                              predictor = "HumidityRatio",
                              Accuracy_Train = accuracy_rpart_HumidityRatio_train,
                              Accuracy_Test =accuracy_rpart_HumidityRatio_test))
# all
set.seed(1, sample.kind = "Rounding")
train_rpart <- train(Occupancy~.,
                     method ="rpart", data = data_training_1)
varImp(train_rpart) # Importance of different predictors
```

```
## rpart variable importance
##
##               Overall
## Light         100.000
## CO2            63.062
## Temperature    30.287
## HumidityRatio   4.391
## Humidity        0.000
```

```r
set.seed(1, sample.kind = "Rounding")
accuracy_rpart_train <- confusionMatrix(predict(train_rpart,
                                                data_training_1),
                                        data_training_1$Occupancy
```

```r
                                        )$overall["Accuracy"]
set.seed(1, sample.kind = "Rounding")
accuracy_rpart_test <- confusionMatrix(predict(train_rpart,
                                               data_testing_1),
                                       data_testing_1$Occupancy
                                       )$overall["Accuracy"]
accuracy_results <- bind_rows(accuracy_results,
                              tibble(method = "rpart",
                                     predictor = "All",
                                     Accuracy_Train = accuracy_rpart_train,
                                     Accuracy_Test =accuracy_rpart_test))
accuracy_results
```

```
## # A tibble: 6 x 4
##   method predictor     Accuracy_Train Accuracy_Test
##   <chr>  <chr>                  <dbl>         <dbl>
## 1 rpart  Temperature        0.8587744     0.6652908
## 2 rpart  Humidity           0.8053543     0.6352720
## 3 rpart  Light              0.9878423     0.9786116
## 4 rpart  CO2                0.9182120     0.8487805
## 5 rpart  HumidityRatio      0.8577920     0.5422139
## 6 rpart  All                0.9930001     0.9557223
```

Summary of rpart model: accuracy__results table shows rpart model with all predictors has accuracy 99% in training but 96% accuracy in test set. Light predictor only rpart model has accuracy 99% in training and 98% in test data set. Using all 5 predictors might be overtaining the rpart model, I will use rpart model with only Light predictor at the validation data set in results section.

```r
rm(accuracy_results) # remove table before next model training and testing
```

2.4.3 knn model

Select best k with all predictors, because all predictors are numeric, knn might the best because I am dealing with the distance.

```r
set.seed(1, sample.kind = "Rounding")
train_knn <- train(Occupancy ~ ., method = "knn",
                   data = data_training_1,
                   tuneGrid = data.frame(k = seq(3, 51, 2)))
train_knn # different k and accuracy
```

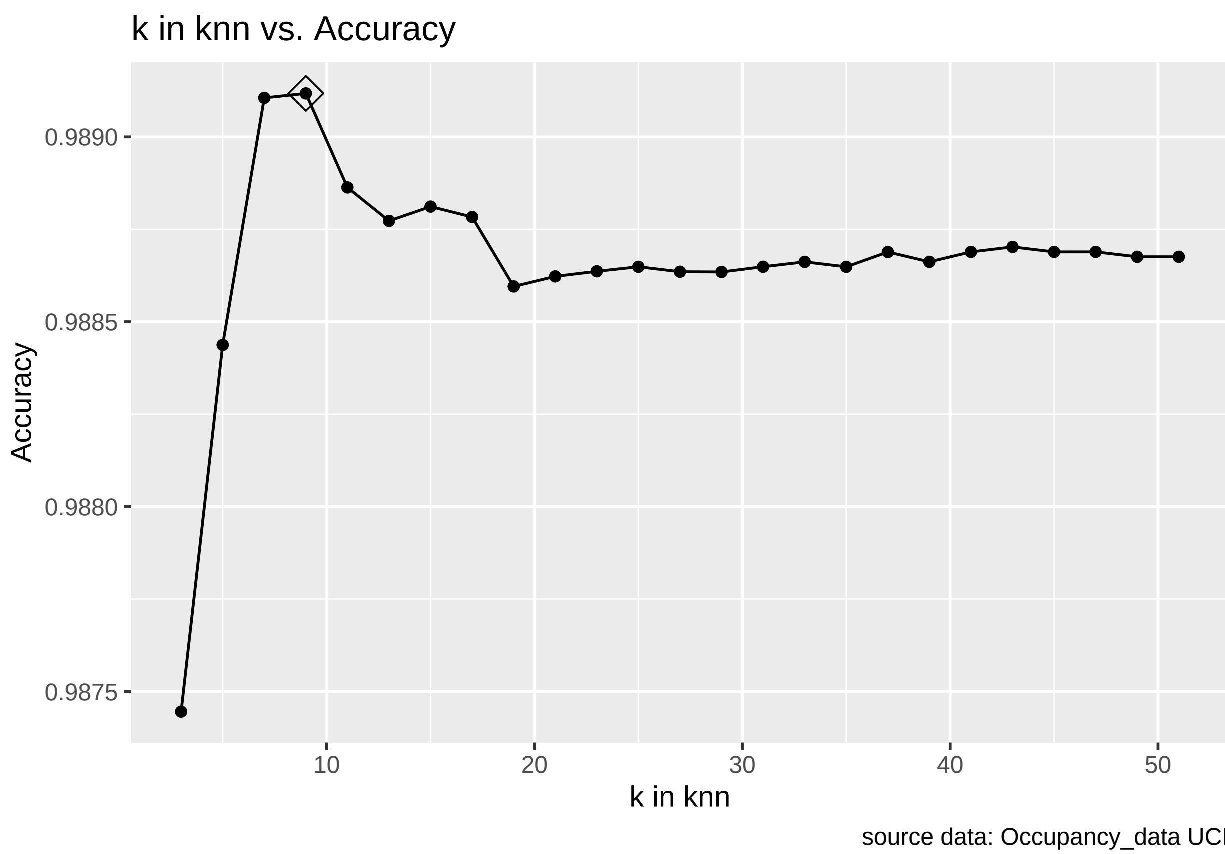
```
## k-Nearest Neighbors
##
## 8143 samples
##    5 predictor
##    2 classes: '0', '1'
##
## No pre-processing
```

```
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 8143, 8143, 8143, 8143, 8143, 8143, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    3  0.9874453  0.9627133
##    5  0.9884372  0.9658100
##    7  0.9891055  0.9678914
##    9  0.9891176  0.9679446
##   11  0.9888632  0.9672370
##   13  0.9887728  0.9670151
##   15  0.9888115  0.9671498
##   17  0.9887833  0.9671025
##   19  0.9885955  0.9665596
##   21  0.9886227  0.9666513
##   23  0.9886365  0.9666942
##   25  0.9886486  0.9667393
##   27  0.9886353  0.9667017
##   29  0.9886347  0.9667090
##   31  0.9886488  0.9667650
##   33  0.9886619  0.9668066
##   35  0.9886486  0.9667676
##   37  0.9886886  0.9668859
##   39  0.9886621  0.9668103
##   41  0.9886889  0.9668840
##   43  0.9887023  0.9669212
##   45  0.9886889  0.9668835
##   47  0.9886889  0.9668815
##   49  0.9886757  0.9668444
##   51  0.9886757  0.9668444
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 9.
```

```r
varImp(train_knn) # Importance of different predictors
```

```
## ROC curve variable importance
##
##               Importance
## Light             100.00
## CO2                93.26
## Temperature        71.33
## HumidityRatio      22.39
## Humidity            0.00
```

```r
# plot to see best k
ggplot(train_knn, highlight = TRUE)+
  labs(x="k in knn", y="Accuracy",
       caption = "source data: Occupancy_data UCI")+
  ggtitle("k in knn vs. Accuracy")
```




```
train_knn$bestTune # the best k
```

```
##   k
## 4 9
```

```
set.seed(1, sample.kind = "Rounding")
accuracy_knn_train <- confusionMatrix(predict(train_knn,
                                              data_training_1, type = "raw"),
                data_training_1$Occupancy)$overall["Accuracy"]
set.seed(1, sample.kind = "Rounding")
accuracy_knn_test <- confusionMatrix(predict(train_knn,
                                             data_testing_1, type = "raw"),
                data_testing_1$Occupancy)$overall["Accuracy"]
accuracy_results <- tibble(method = "knn",
                           predictor = "All",
                           Accuracy_Train = accuracy_knn_train,
                           Accuracy_Test =accuracy_knn_test)
accuracy_results
```

```
## # A tibble: 1 x 4
##   method predictor Accuracy_Train Accuracy_Test
##   <chr>  <chr>              <dbl>         <dbl>
## 1 knn    All            0.9896844     0.9617261
```

```r
# try 10-flod cross validation to see any further accuracy improvement
set.seed(1, sample.kind = "Rounding")
control <- trainControl(method = "cv", number = 10, p = .9)
train_knn_cv <- train(Occupancy ~ ., method = "knn",
                      data = data_training_1,
                      tuneGrid = data.frame(k = seq(3, 51, 2)),
                      trControl = control)
train_knn_cv
```

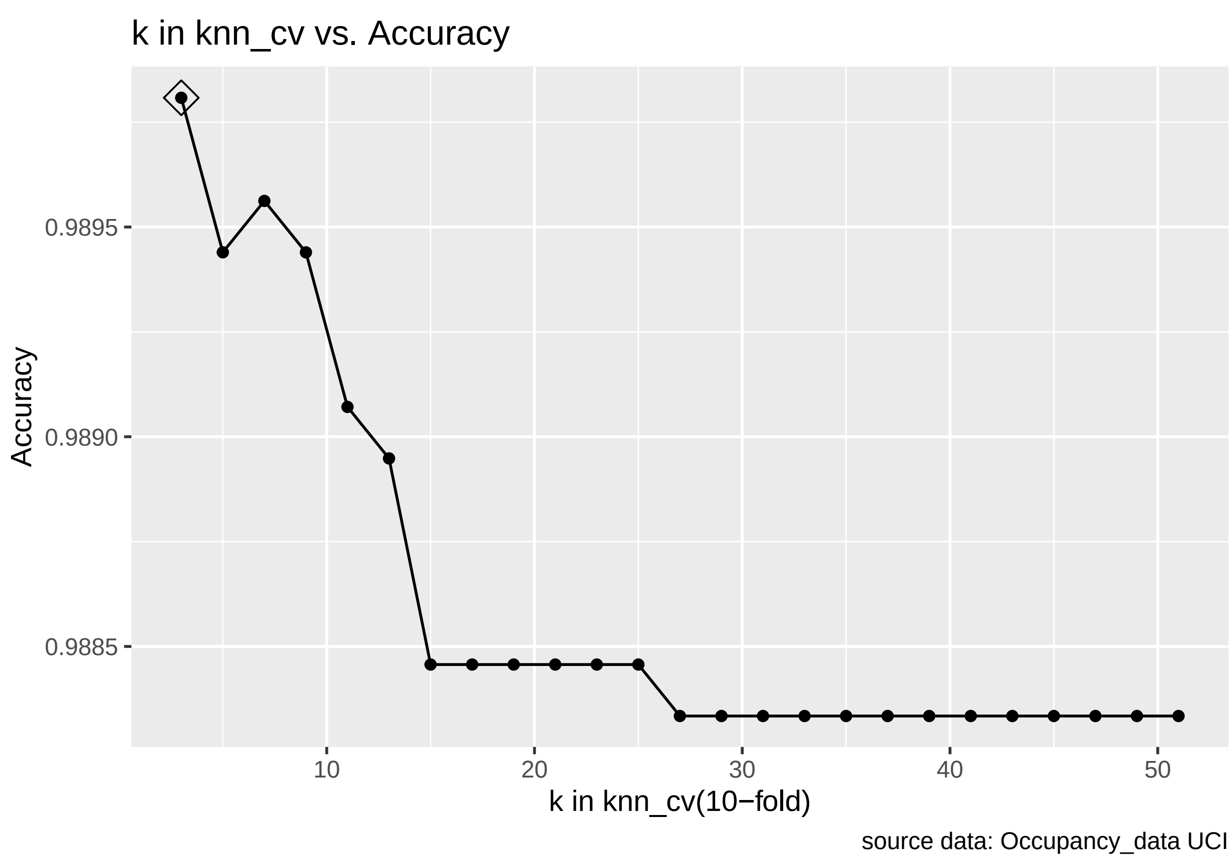
```
## k-Nearest Neighbors
##
## 8143 samples
##    5 predictor
##    2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 7328, 7328, 7329, 7328, 7329, 7330, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    3  0.9898081  0.9697455
##    5  0.9894397  0.9688083
##    7  0.9895623  0.9692094
##    9  0.9894396  0.9688873
##   11  0.9890712  0.9678504
##   13  0.9889483  0.9675367
##   15  0.9884569  0.9661214
##   17  0.9884569  0.9661435
##   19  0.9884569  0.9661435
##   21  0.9884569  0.9661435
##   23  0.9884569  0.9661435
##   25  0.9884569  0.9661435
##   27  0.9883342  0.9657760
##   29  0.9883342  0.9657760
##   31  0.9883342  0.9657760
##   33  0.9883342  0.9657760
##   35  0.9883342  0.9657760
##   37  0.9883342  0.9657760
##   39  0.9883342  0.9657760
##   41  0.9883342  0.9657760
##   43  0.9883342  0.9657760
##   45  0.9883342  0.9657760
##   47  0.9883342  0.9657760
##   49  0.9883342  0.9657760
##   51  0.9883342  0.9657760
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 3.
```

```r
varImp(train_knn_cv) # Importance of different predictors
```

```
## ROC curve variable importance
```

```
##
##                  Importance
## Light                100.00
## CO2                   93.26
## Temperature           71.33
## HumidityRatio         22.39
## Humidity               0.00
```

```
ggplot(train_knn_cv, highlight = TRUE)+
  labs(x="k in knn_cv(10-fold)", y="Accuracy",
       caption = "source data: Occupancy_data UCI")+
  ggtitle("k in knn_cv vs. Accuracy")
```




```
train_knn_cv$bestTune # the best k
```

```
##   k
## 1 3
```

```
set.seed(1, sample.kind = "Rounding")
accuracy_knn_cv_train <- confusionMatrix(predict(train_knn_cv,
                                                 data_training_1, type = "raw"),
                                         data_training_1$Occupancy)$overall["Accuracy"]
set.seed(1, sample.kind = "Rounding")
accuracy_knn_cv_test <- confusionMatrix(predict(train_knn_cv,
```

```r
                                                data_testing_1, type = "raw"),
                                     data_testing_1$Occupancy)$overall["Accuracy"]
accuracy_results <- bind_rows(accuracy_results,
                              tibble(method = "knn_cv",
                              predictor = "All",
                              Accuracy_Train = accuracy_knn_cv_train,
                              Accuracy_Test =accuracy_knn_cv_test))
accuracy_results
```

```
## # A tibble: 2 x 4
##   method predictor Accuracy_Train Accuracy_Test
##   <chr>  <chr>              <dbl>         <dbl>
## 1 knn    All            0.9896844     0.9617261
## 2 knn_cv All            0.9948422     0.9350844
```

Summary of knn model, added 10-fold validation has lower accuracy in test data set. And the 10-fold knn_cv model has k=3 which might be overtraining the model. I will keep knn without 10-fold cross validaiton at validation data set in results section. I also learned Knn model use more computer time than qda and rpart model.

```r
rm(accuracy_results) # remove accuracy table before next model training and testing
```

2.4.4 Radom forest

Random forest is good for classification and regression. It can also be used in unsupervised mode for assessing proximities among data points.I will use all predictors at this model.

```r
set.seed(1, sample.kind = "Rounding")
train_rf <- train(Occupancy~., method = "rf", data = data_training_1)
train_rf
```

```
## Random Forest
##
## 8143 samples
##    5 predictor
##    2 classes: '0', '1'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 8143, 8143, 8143, 8143, 8143, 8143, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##   2     0.9936679  0.9811675
##   3     0.9935344  0.9807610
##   5     0.9927199  0.9783111
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 2.
```

```r
varImp(train_rf) # Importance of different predictors
```

```
## rf variable importance
##
##                   Overall
## Light           100.0000
## CO2              40.1372
## Temperature      11.9446
## HumidityRatio     0.9605
## Humidity          0.0000
```

```r
set.seed(1, sample.kind = "Rounding")
accuracy_rf_train <- confusionMatrix(predict(train_rf, data_training_1),
                                     data_training_1$Occupancy
                                     )$overall["Accuracy"]
set.seed(1, sample.kind = "Rounding")
accuracy_rf_test <- confusionMatrix(predict(train_rf, data_testing_1),
                                    data_testing_1$Occupancy
                                    )$overall["Accuracy"]
accuracy_results <- tibble(method = "rf",
                           predictor = "All",
                           Accuracy_Train = accuracy_rf_train,
                           Accuracy_Test =accuracy_rf_test)
accuracy_results
```

```
## # A tibble: 1 x 4
##   method predictor Accuracy_Train Accuracy_Test
##   <chr>  <chr>              <dbl>         <dbl>
## 1 rf     All                    1     0.9500938
```

Summary rf model: rf model has 100% accuracy in trainind data set, but 97% in testing data set. I will keep it for the validation and check accuracy in the results section for now.I also learned rf model use more computer time than qda and rpart model.

```r
rm(accuracy_results) # remove all accuray talbe before summary
```

2.4.5 Summary of methods/analysis section:

```r
accuracy_results <- tibble(method = c("qda","rpart","knn", "rf"),
                           predictors = c("All","Light", "All","All"),
                           Accuracy_Train = c(accuracy_qda_train,
                                              accuracy_rpart_Light_train,
                                              accuracy_knn_train,
                                              accuracy_rf_train),
                           Accuracy_Test = c(accuracy_qda_test,
                                             accuracy_rpart_Light_test,
                                             accuracy_knn_test,
                                             accuracy_rf_test))
accuracy_results
```

```
## # A tibble: 4 x 4
##   method predictors Accuracy_Train Accuracy_Test
##   <chr>  <chr>               <dbl>         <dbl>
## 1 qda    All             0.9888248     0.9774859
## 2 rpart  Light           0.9878423     0.9786116
## 3 knn    All             0.9896844     0.9617261
## 4 rf     All             1             0.9500938
```

Summary of modeling: Single predictor model has lower accuracy compare to using all 5 predictors in qda model. Light predictor in rpart model has higher accuracy than using all predictors. knn and rf model showed high accuracy in training data set. Overall, these 4 models showed high accuracy, I will use these 4 model in results section for validation.

3. Results

From the data analysis, I learned that qda, rpart, knn, and random forest gave me high accuracy model in train and test set. I am going to apply them on the validaiton set(test2 data set) and pick two final models for recommendation.

```r
# qda with 5 predictors
set.seed(1, sample.kind = "Rounding")
accuracy_qda_test2 <- confusionMatrix(predict(train_qda, data_testing2_1),
                                      data_testing2_1$Occupancy
                                      )$overall["Accuracy"]
final_accuracy_validation <- tibble(
  method = "qda",
  predictors = "Temperature+Humidity+Light+CO2+HumidityRatio",
  Accuracy_validattion =accuracy_qda_test2)

# rpart with only Light predictor
set.seed(1, sample.kind = "Rounding")
accuracy_rpart_Light_test2 <-  confusionMatrix(predict(train_rpart_Light,data_testing2_1),
                                              data_testing2_1$Occupancy
                                              )$overall["Accuracy"]
final_accuracy_validation <- bind_rows(
  final_accuracy_validation,
  tibble(method = "rpart",
         predictors = "Light",
         Accuracy_validattion =accuracy_rpart_Light_test2))

# knn with 5 predictors
set.seed(1, sample.kind = "Rounding")
accuracy_knn_test2 <- confusionMatrix(predict(train_knn, data_testing2_1),
                                              data_testing2_1$Occupancy
                                              )$overall["Accuracy"]
final_accuracy_validation <- bind_rows(
  final_accuracy_validation,
  tibble(method = "knn",
         predictors = "Temperature+Humidity+Light+CO2+HumidityRatio",
         Accuracy_validattion =accuracy_knn_test2))
```

```r
# rf with 5 predictors
set.seed(1, sample.kind = "Rounding")
accuracy_rf_test2 <- confusionMatrix(predict(train_rf, data_testing2_1),
                                     data_testing2_1$Occupancy
                                     )$overall["Accuracy"]
final_accuracy_validation <- bind_rows(
  final_accuracy_validation,
  tibble(method = "rf",
         predictors = "Temperature+Humidity+Light+CO2+HumidityRatio",
         Accuracy_validattion =accuracy_rf_test2))

final_accuracy_validation
```

```
## # A tibble: 4 x 3
##   method predictors                                    Accuracy_validattion
##   <chr>  <chr>                                                        <dbl>
## 1 qda    Temperature+Humidity+Light+CO2+HumidityRatio                 0.9867719
## 2 rpart  Light                                                        0.9931296
## 3 knn    Temperature+Humidity+Light+CO2+HumidityRatio                 0.9656481
## 4 rf     Temperature+Humidity+Light+CO2+HumidityRatio                 0.9746719
```

4. Conclusion

Although I didn't use date in the models, the final validaition showed accuracy from 97% to 99%. Predictor Light has greatest effect on occupancy prediction in rpart model. Based on my results, I would like to recommand two models: qda and rpart models because both them have high accuracy 99% in validation set and use less computer time to run the model comparing to knn and rf models. From the results, some models has very high accuracy in train data set, but the test and validation dataset accuracy is lower, there might be some overtraining in the models. So I think my further analysis will try to avoid overtraining models and imroving models accuracy. I would also like to investigating more date effect on the models. Another approach I think it will be good to try is combine all three data sets togehter and randomly set training, test, and validation data set to test models.