Occupancy Detection Project Report

Raj Krishna, PMP, PMI-ACP, CSM

2019-12-30

Contents

Introduction

The objective of this project is to assess various prediction models by their accuracy in detecting occupancy of an office room based on **Light, Temperature, CO_2, Date(converted to Weekday), Humidity and HumidityRatio**. The dataset for the project was obtained from the UC Irvine Machine Learning Repository. This is a binary classification scenario where the model attempts to predict whether a room is occupied or not.

The occupancy data[1], at the source, is divided into 3 separate datasets, one training dataset **datatraining** with **8,143** records, and two test datasets, **datatest** and **datatest2** with **2,665** and **9,752** records respectively. Apart from the 6 features mentioned above, the file also contains **Occupancy** (binary value, 0 - Not Occupied, 1 - Occupied) and an **index** which would be dropped during preprocessing of the data.

As part of preprocessing of data the three files - datatraining, datatest, and datatest2, would be combined to create one dataset with **20,560** records. After running basics statistics on the dataset, an automatic feature selection alorithm would be run to verify the best set of features.

The dataset would then be split into train and test datasets, and a total for 15 models namely **"glm", "lda", "naive_bayes", "svmLinear", "knn", "gamLoess", "multinom", "qda", "mda", "rpart", "rf", "C5.0", "fda", "pda", "gbm"** would be trained using the train dataset. The fits obtained would be resampled and the best fit identified by accuracy would then be used to predict on the test dataset. The final accuracy would be obtained for this to conclude the project.

Datasets

The occupancy detection data dataset created as part of extraction of data from the source location would contain **20,560** records which would be split into a train datasets with **80%** data containing **16,448** rcords and test dataset with **20%** of the data containing the remaining **4,112** records.

Overview

Following code would be used to download the data and combine the three datasets into a single data set.

Load necessary libraries

```
#############################################
# Installing or loading necessary libraries #
#############################################

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(kableExtra))
  install.packages("kableExtra", repos = "http://cran.us.r-project.org")
if(!require(mlbench))
  install.packages("mlbench", repos = "http://cran.us.r-project.org")
if(!require(data.table))
  install.packages("data.table", repos = "http://cran.us.r-project.org")
if(!require(wordcloud))
  install.packages("wordcloud", repos = "http://cran.us.r-project.org")
if(!require(DataExplorer))
  install.packages("DataExplorer", repos = "http://cran.us.r-project.org")
```

[1] Source: https://archive.ics.uci.edu/ml/machine-learning-databases/00357/

```
if(!require(psych))
  install.packages("psych", repos = "http://cran.us.r-project.org")
if(!require(mda))
  install.packages("mda", repos = "http://cran.us.r-project.org")
if(!require(rpart))
  install.packages("rpart", repos = "http://cran.us.r-project.org")
if(!require(C50))
  install.packages("C50", repos = "http://cran.us.r-project.org")
if(!require(fda))
  install.packages("earth", repos = "http://cran.us.r-project.org")
if(!require(gbm))
  install.packages("gbm", repos = "http://cran.us.r-project.org")
```

Download Occupancy data

Following code would download the three occupancy data datasets and combine the to form a single dataset of 20,560 records.

```
##########################################################################################
# Occupancy data:                                                                        #
# https://archive.ics.uci.edu/ml/machine-learning-databases/00357/occupancy_data.zip     #
# This zip contains data in 3 files,  namely datatraining.txt, datatest.txt, and         #
# datatest2.txt.                                                                         #
# The below code downloads the zip files, reads the 3 files and combines it to deliver   #
# a single dataset of 20,560 observation and 7 variables.                                #
# An extraneous columns, apart from the 7 named observations, exists in the file, which  #
# I am dropping as it's not pertinent to our project. While I am retaining date, for the #
# purposes of this project, I would not be using it in its current format.               #
##########################################################################################

#############################################
# Data setup                                #
#############################################

# Downloading, reading and combining files to create the dataset
dl <- tempfile()
download.file(
  "https://archive.ics.uci.edu/ml/machine-learning-databases/00357/occupancy_data.zip",
  dl)

# List of file names in the zip file
files <- c("datatraining.txt", "datatest.txt", "datatest2.txt")

# Column names for the data in the file
colnames <- c("Date","Temperature","Humidity","Light","CO2","HumidityRatio","Occupancy")

# Function definition to read data and drop extraneous first column which is an index
read_and_combine_files <- function(file, dl, colnames) {
  data <- fread(text = gsub(",", "\t", readLines(unzip(dl, file))),
              drop = 1, col.names = colnames)
  return(data)
}

# Combining the data from the files into a dataframe
```

```
occupancy_detection_data <- bind_rows(lapply(files, read_and_combine_files, dl, colnames))
```

Remove temporary objects

Let's remove variables that are no longer necessary.

```
# Removing variables that are no longer necessary
remove(dl, files, colnames)
```

The occupancy_detection_data dataset contains the following 8 variables:

1. **Date**: Date(in timestamp format) when the readings were taken
2. **Weekday**: Day of the week extracted from the Date
3. **Temprature**: Temprature of the room
4. **Humidity**: Humidity of the room
5. **Light**: Light in the room
6. **CO_2**: Measure of Carbon diOxide in the room
7. **HumidityRatio**: Ratio of the mass of water vapor in the humid air - to the mass of dry air
8. **Occupancy**: Occupancy status (binary value, 0 - Not Occupied, 1 - Occupied)

Methods and Analysis

Exploratory Data Analysis

Let explore and pre-process the data before delving deep into the project.

Data pre-processing

Let's take a peek at the data:

```
# List the structure of the occupancy detectinon data
glimpse(occupancy_detection_data)
```

```
## Observations: 20,560
## Variables: 7
## $ Date          <chr> "2015-02-04 17:51:00", "2015-02-04 17:51:59", "2015-0...
## $ Temperature   <dbl> 23.180, 23.150, 23.150, 23.150, 23.100, 23.100, 23.10...
## $ Humidity      <dbl> 27.27200, 27.26750, 27.24500, 27.20000, 27.20000, 27....
## $ Light         <dbl> 426.0, 429.5, 426.0, 426.0, 426.0, 419.0, 419.0, 419....
## $ CO2           <dbl> 721.2500, 714.0000, 713.5000, 708.2500, 704.5000, 701...
## $ HumidityRatio <dbl> 0.004792988, 0.004783441, 0.004779464, 0.004771509, 0...
## $ Occupancy     <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0,...
```

As we can see the Date field is in Character format, this would not do for the purposes of our investigation, we would need to convert this to a date time format. Also, the Occupancy data is integer, this would also create problems with while training models. We'll conver this into factor and also encode the values to ones that are valid variable names in R. Further, we will rearrange the dataset so that Date is in the begining and Occupancy is at the end. This will facilitate easier reference in some of our processing.

```
# Converting occupancy to factor
occupancy_detection_data$Occupancy <- as.factor(occupancy_detection_data$Occupancy)

# Renaming the levels for occupancy as '0' & '1' are not valid variable names in R and
# would cause some of the models to fail
levels(occupancy_detection_data$Occupancy) <- c("Not_Occupied", "Occupied")

# Formating the Date data
```

```r
occupancy_detection_data$Date <- as.POSIXct(occupancy_detection_data$Date, tz = "UTC")

# Extracting day of the week from date
occupancy_detection_data$Weekday <- wday(occupancy_detection_data$Date)

# Rearranging the dataframe
occupancy_detection_data <- occupancy_detection_data %>%
  select(Date, Weekday, everything())

# Check the conversions and changes have worked
glimpse(occupancy_detection_data)
```

```
## Observations: 20,560
## Variables: 8
## $ Date          <dttm> 2015-02-04 17:51:00, 2015-02-04 17:51:59, 2015-02-04...
## $ Weekday       <int> 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,...
## $ Temperature   <dbl> 23.180, 23.150, 23.150, 23.150, 23.100, 23.100, 23.10...
## $ Humidity      <dbl> 27.27200, 27.26750, 27.24500, 27.20000, 27.20000, 27....
## $ Light         <dbl> 426.0, 429.5, 426.0, 426.0, 426.0, 419.0, 419.0, 419....
## $ CO2           <dbl> 721.2500, 714.0000, 713.5000, 708.2500, 704.5000, 701...
## $ HumidityRatio <dbl> 0.004792988, 0.004783441, 0.004779464, 0.004771509, 0...
## $ Occupancy     <fct> Occupied, Occupied, Occupied, Occupied, Occupied, Occ...
```

Data Exploration

Now that we have the data in the form we wanted, let's explore the details of the data that we have.

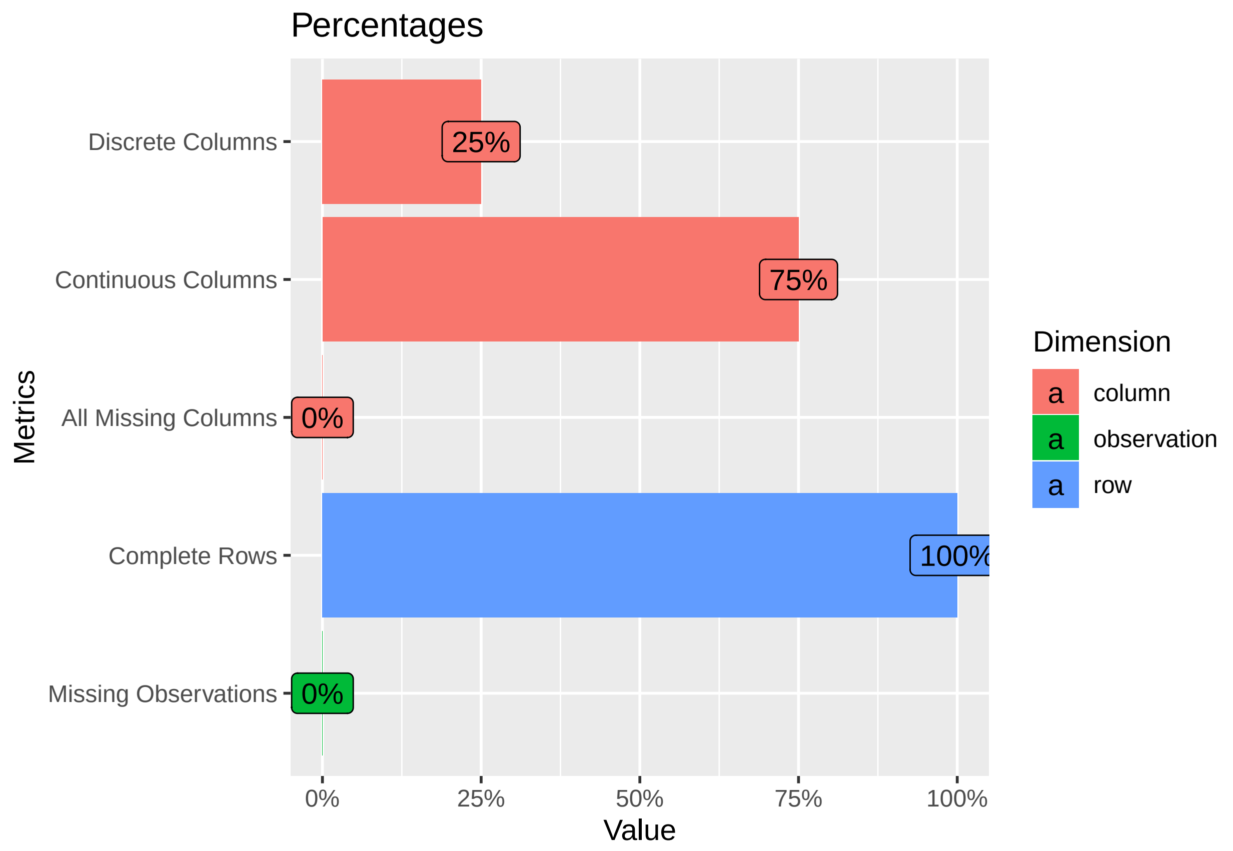
```r
# Basic statistics of the dataset

introduce(occupancy_detection_data) %>%
  mutate(memory_usage = paste0(round(memory_usage/ 2 ^ 20, 1), " Mb")) %>%
  rename(Rows = rows, Columns = columns,
         "Discrete Columns" = discrete_columns,
         "Continous Columns" = continuous_columns,
         "All missing columns" = all_missing_columns,
         "Total Missing Values" = total_missing_values,
         "Complete Rows" = complete_rows,
         "Total Observations" = total_observations,
         "Memory Usage" = memory_usage) %>%
  gather() %>% rename(Name = key, Value = value) %>%
  knitr::kable("latex", caption = "Basic statistics of the dataset",
               escape = FALSE, linesep = "", booktabs = TRUE,
               align = c('l', 'r')) %>%
  kable_styling(latex_options = c("striped", "HOLD_position"),
                full_width = F , position = "center") %>%
  row_spec(0, bold = TRUE, color = "white" , background ="red") %>%
  footnote(general = "Values are in count except for \"Memory Usage\"",
           general_title = "Note:", footnote_as_chunk = TRUE)
```

Table 1: Basic statistics of the dataset

Name	Value
Rows	20560
Columns	8
Discrete Columns	2
Continous Columns	6
All missing columns	0
Total Missing Values	0
Complete Rows	20560
Total Observations	164480
Memory Usage	1.1 Mb

Note: Values are in count except for "Memory Usage"

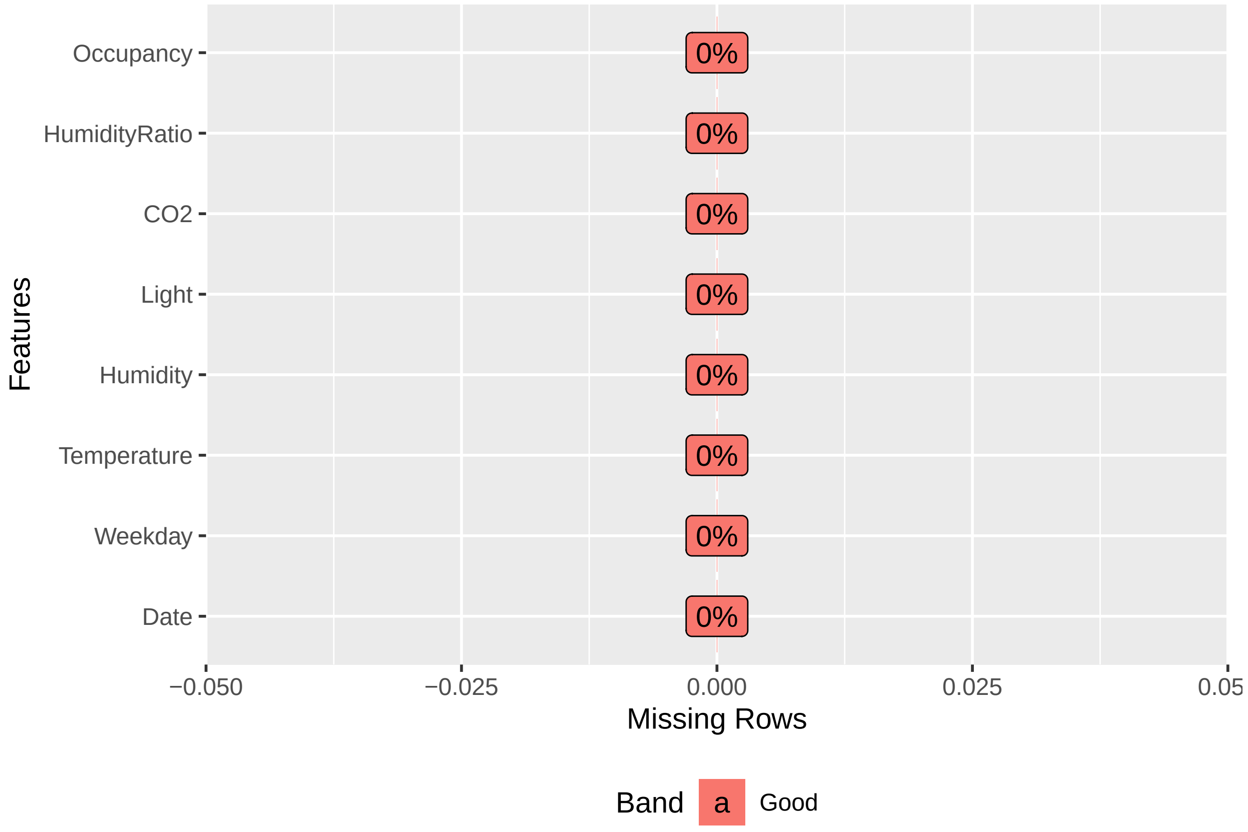
```
# Percentages
plot_intro(occupancy_detection_data, title = "Percentages")
```



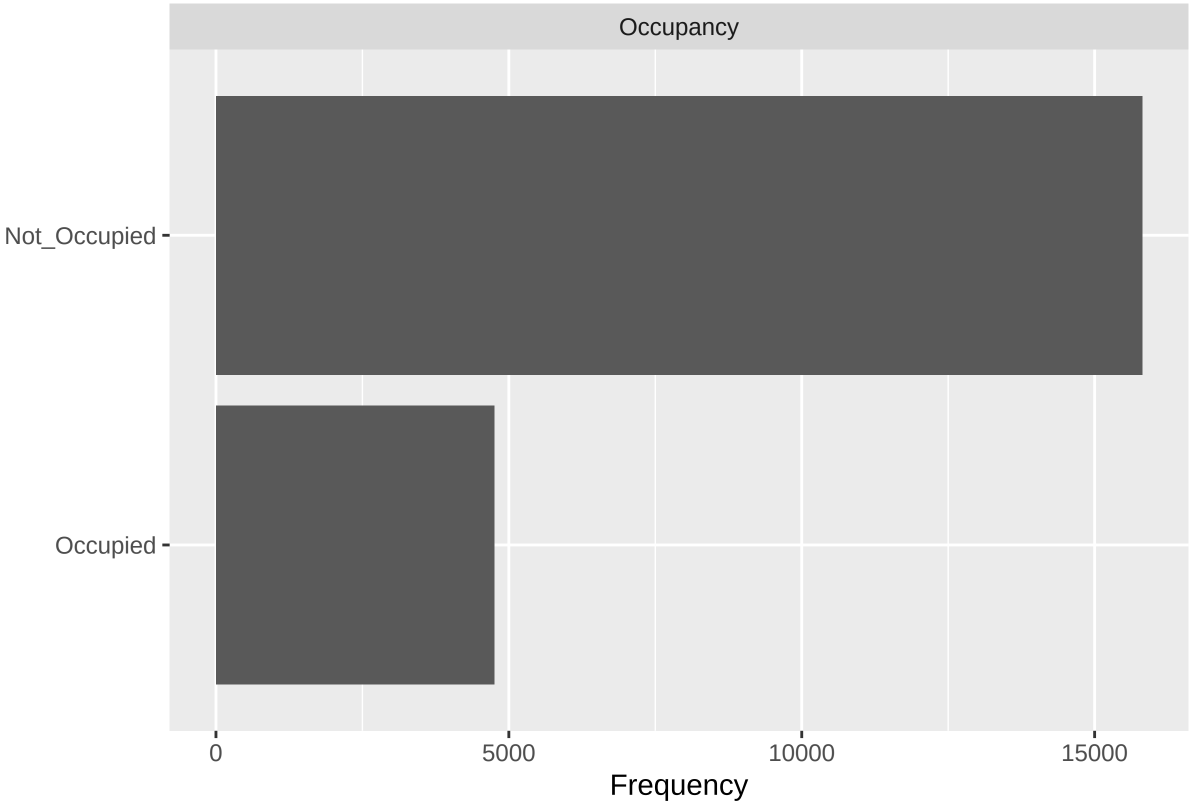
```
# Summary of the data
summary(occupancy_detection_data)
```

```
##       Date                        Weekday       Temperature       Humidity
##  Min.   :2015-02-02 14:19:00   Min.   :1.000   Min.   :19.00   Min.   :16.75
##  1st Qu.:2015-02-06 11:05:45   1st Qu.:2.000   1st Qu.:20.20   1st Qu.:24.50
##  Median :2015-02-10 00:45:30   Median :4.000   Median :20.70   Median :27.29
##  Mean   :2015-02-10 13:42:06   Mean   :3.916   Mean   :20.91   Mean   :27.66
##  3rd Qu.:2015-02-14 19:39:14   3rd Qu.:6.000   3rd Qu.:21.52   3rd Qu.:31.29
##  Max.   :2015-02-18 09:19:00   Max.   :7.000   Max.   :24.41   Max.   :39.50
##      Light             CO2          HumidityRatio            Occupancy
##  Min.   :   0.0   Min.   : 412.8   Min.   :0.002674   Not_Occupied:15810
##  1st Qu.:   0.0   1st Qu.: 460.0   1st Qu.:0.003719   Occupied    : 4750
##  Median :   0.0   Median : 565.4   Median :0.004292
##  Mean   : 130.8   Mean   : 690.6   Mean   :0.004228
##  3rd Qu.: 301.0   3rd Qu.: 804.7   3rd Qu.:0.004832
##  Max.   :1697.2   Max.   :2076.5   Max.   :0.006476
```

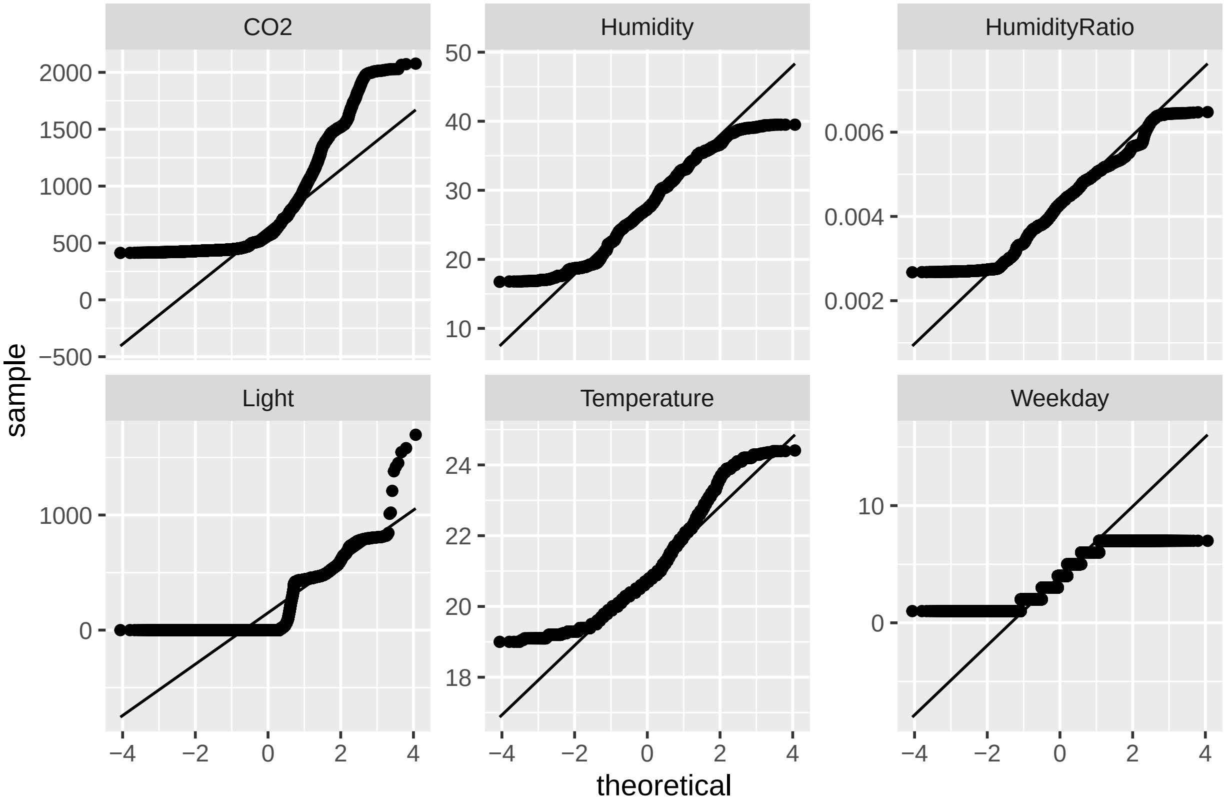
```
# Missing data profile
plot_missing(occupancy_detection_data)
```



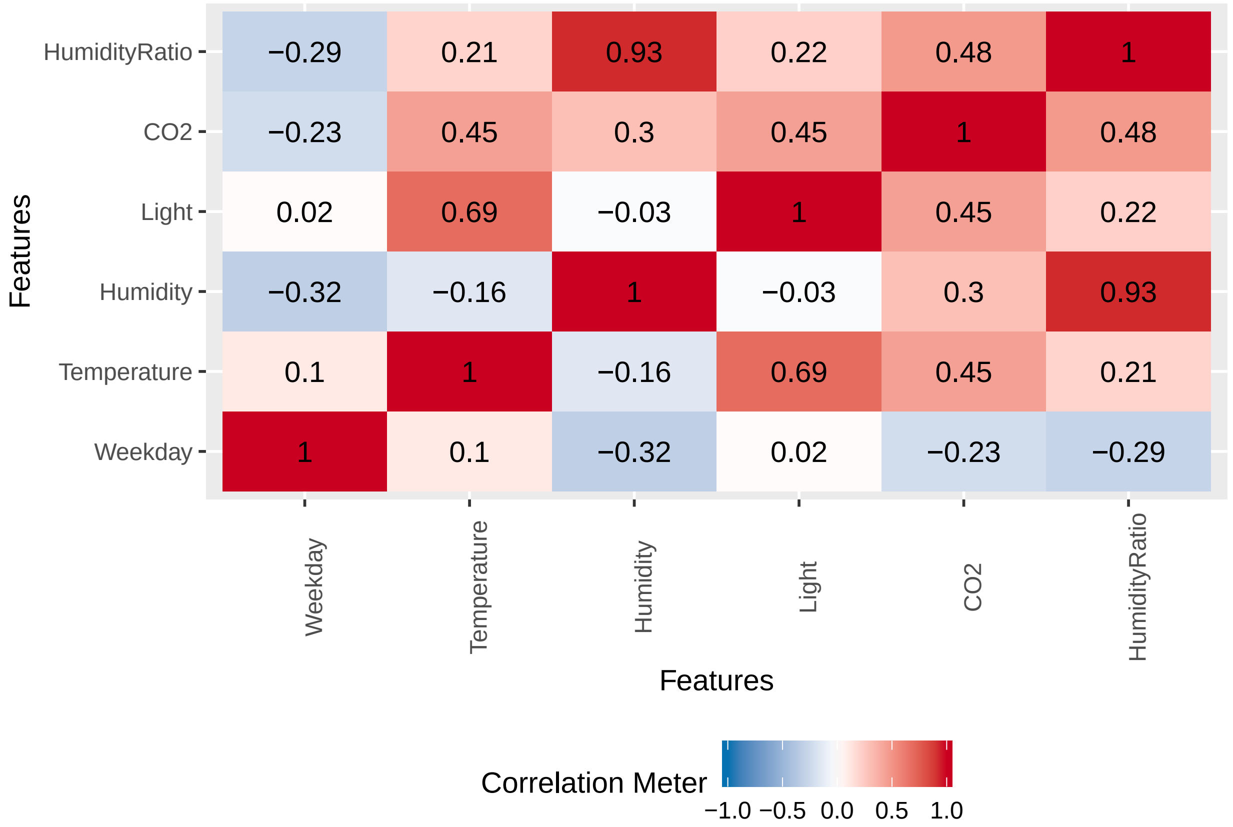
```
# Bar Chart by frequency: Occupation
plot_bar(occupancy_detection_data)
```



```
## QQ plot
occupancy_detection_data %>%
  plot_qq()
```



```
# Correlation Analysis
occupancy_detection_data %>% select(-Date,-Occupancy) %>%
plot_correlation(type = "all")
```



We can see from the correlation matrix that Humidity and HumidityRatio are highly correlated. Light and Temprature also have significant correlation.

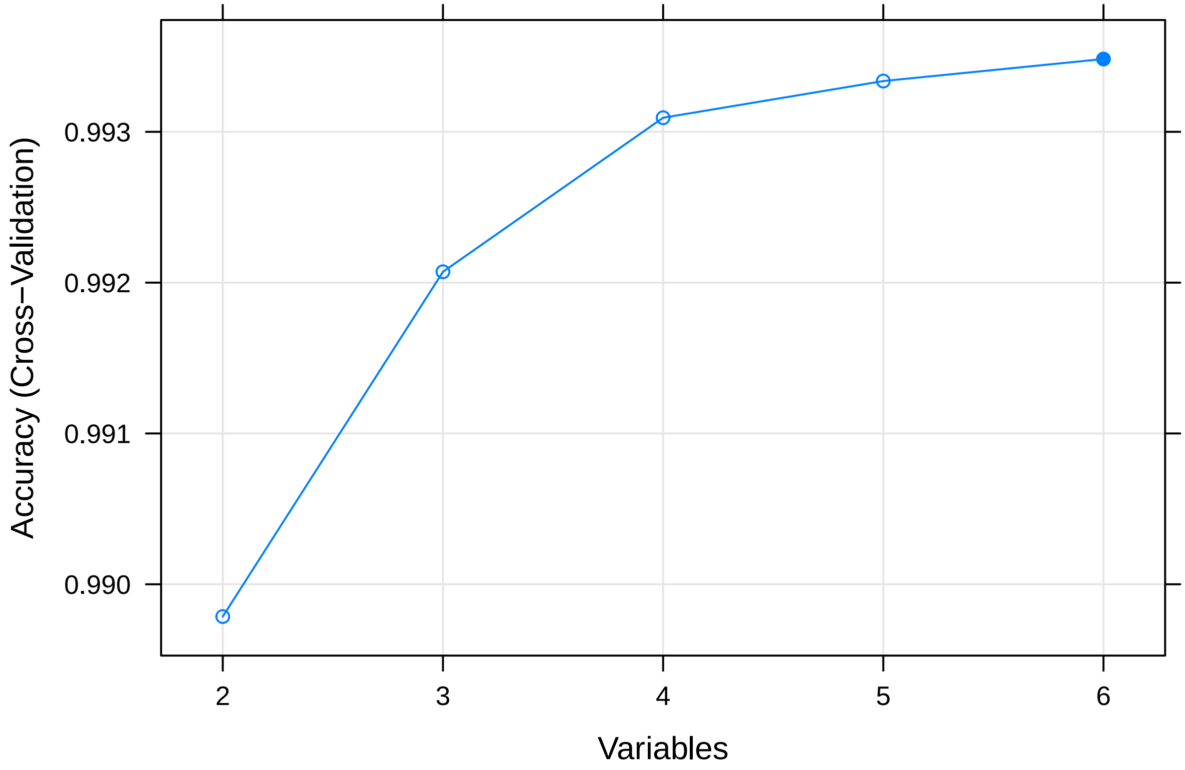
Automatic Feature Selection

We will now use Recursive Feature Elimination for feature selection. The following code would idnetify the optimum number of features that would maximize the accuracy

```
# Define control for RFE, we are using Random Forest function for selection
rfe_control <- rfeControl(functions = rfFuncs, method = "cv", number = 10)

# RFE algorithm
rfe_results <- rfe(occupancy_detection_data[, 2:7], occupancy_detection_data[[8]],
                   sizes = c(2:7), rfeControl = rfe_control)

# Summarize the results
print(rfe_results)
```

```
##
## Recursive feature selection
##
## Outer resampling method: Cross-Validated (10 fold)
##
## Resampling performance over subset size:
##
##  Variables Accuracy  Kappa AccuracySD  KappaSD Selected
##          2   0.9898 0.9715   0.002693 0.007452
##          3   0.9921 0.9778   0.002864 0.007943
##          4   0.9931 0.9806   0.002358 0.006605
##          5   0.9933 0.9813   0.002937 0.008206
##          6   0.9935 0.9817   0.002646 0.007386        *
##
## The top 5 variables (out of 6):
##    Light, Temperature, CO2, Weekday, Humidity
```

```
# Plotting the results
plot(rfe_results, type = c("g", "o"))
```



The RFE has selected Light, Temperature, CO_2, Weekday, Humidity as the top features. This would make sense since we have already seen from correlation analysis that Humidity and HumidityRatio are highly correlated. Using all 6 features does bring a small improvement in the accuracy as depicted in the plot above. While RFE recommends using all 6 features, we'll be dropping HumidityRatio from further processing.

Assessing models

We'll now proceed with splitting the occupancy_detection_data dataset into 80:20 train and test datasets.

```
# Splitting the data set for training
set.seed(1, sample.kind = "Rounding")
test_index <- createDataPartition(occupancy_detection_data$Occupancy,
                                  times = 1, p = 0.2, list = FALSE)
occupancy_train <- occupancy_detection_data[-test_index,]
occupancy_test <- occupancy_detection_data[test_index,]

# remove text index
rm(test_index)
```

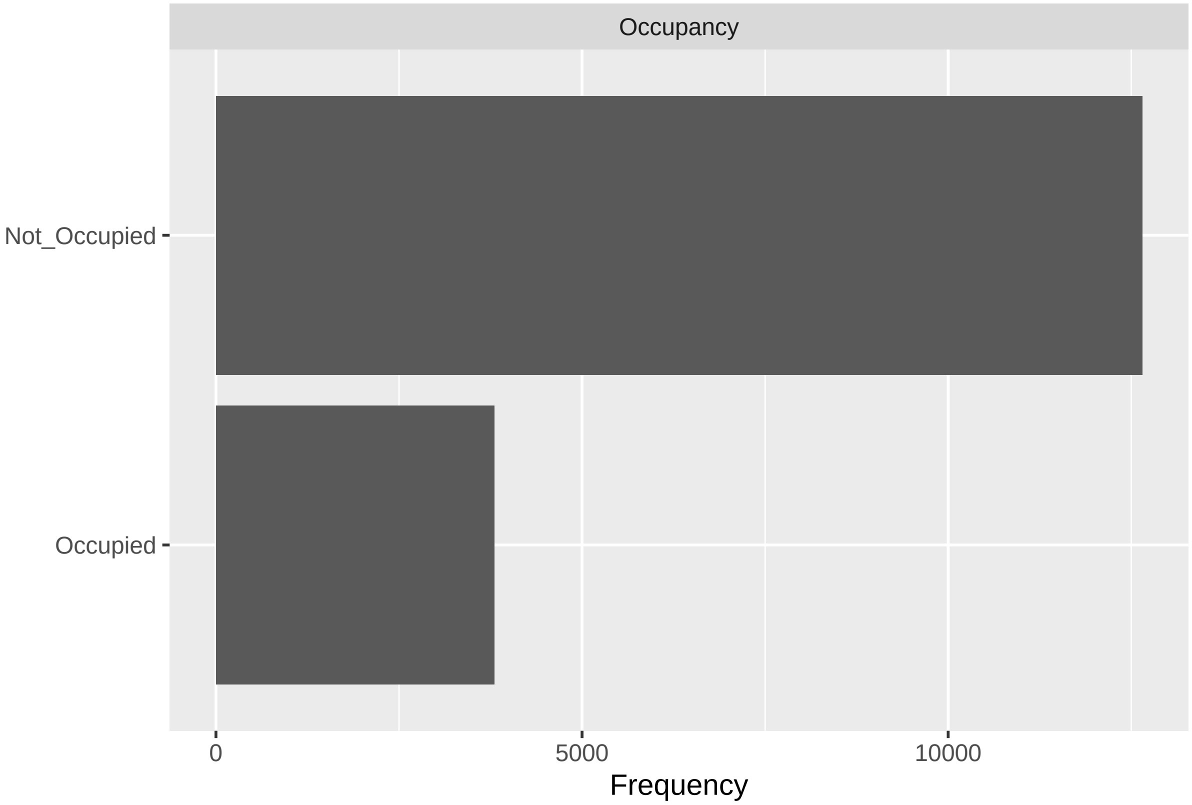
```r
# Verify the split
glimpse(occupancy_train)
```

```
## Observations: 16,448
## Variables: 8
## $ Date          <dttm> 2015-02-04 17:51:00, 2015-02-04 17:51:59, 2015-02-04...
## $ Weekday       <int> 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,...
## $ Temperature   <dbl> 23.180, 23.150, 23.150, 23.150, 23.100, 23.100, 23.10...
## $ Humidity      <dbl> 27.27200, 27.26750, 27.24500, 27.20000, 27.20000, 27....
## $ Light         <dbl> 426.0, 429.5, 426.0, 426.0, 419.0, 419.0, 419.0, 419....
## $ CO2           <dbl> 721.2500, 714.0000, 713.5000, 708.2500, 701.0000, 701...
## $ HumidityRatio <dbl> 0.004792988, 0.004783441, 0.004779464, 0.004771509, 0...
## $ Occupancy     <fct> Occupied, Occupied, Occupied, Occupied, Occupied, Occ...
```

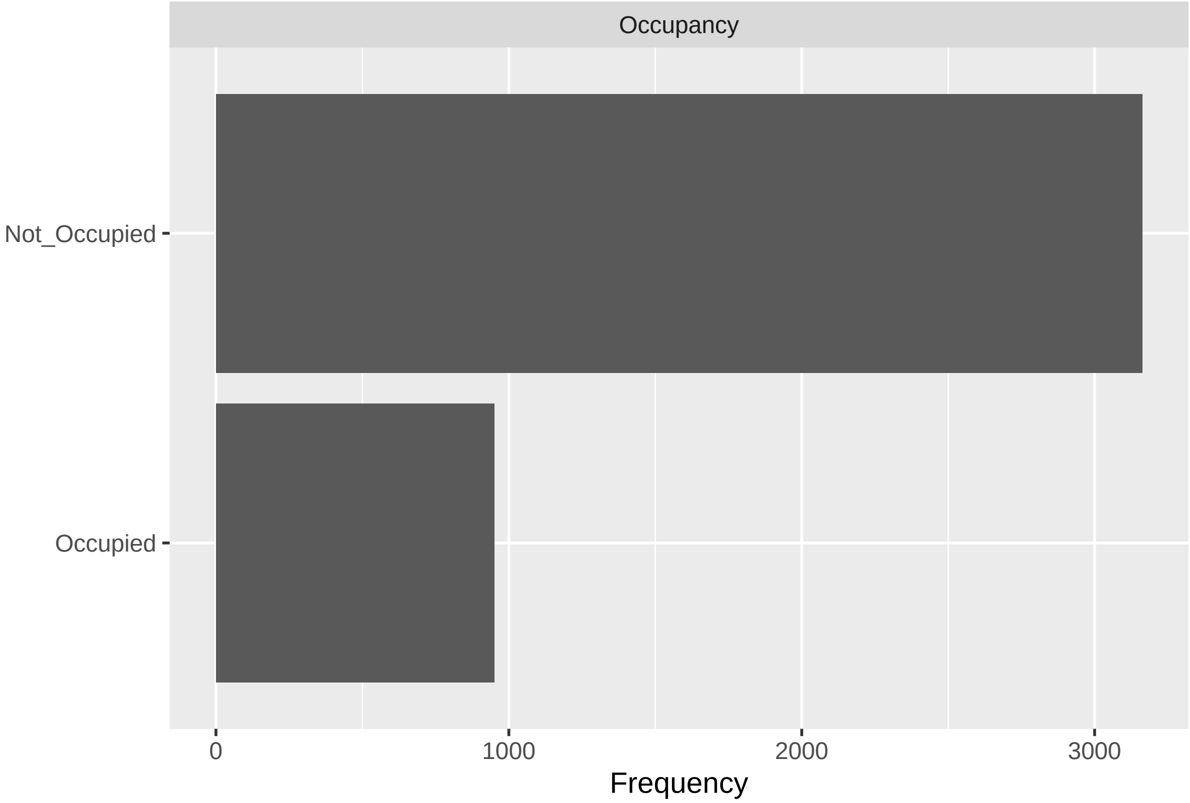
```r
glimpse(occupancy_test)
```

```
## Observations: 4,112
## Variables: 8
## $ Date          <dttm> 2015-02-04 17:55:00, 2015-02-04 18:04:59, 2015-02-04...
## $ Weekday       <int> 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,...
## $ Temperature   <dbl> 23.100, 23.000, 22.890, 22.890, 22.700, 22.700, 22.60...
## $ Humidity      <dbl> 27.20000, 27.12500, 27.50000, 27.50000, 27.46333, 27....
## $ Light         <dbl> 426, 419, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ CO2           <dbl> 704.5000, 686.0000, 688.0000, 689.5000, 668.6667, 670...
## $ HumidityRatio <dbl> 0.004756993, 0.004714942, 0.004748645, 0.004748645, 0...
## $ Occupancy     <fct> Occupied, Occupied, Not_Occupied, Not_Occupied, Not_O...
```

```r
# Plot the split data
plot_bar(occupancy_train)
```



```
plot_bar(occupancy_test)
```



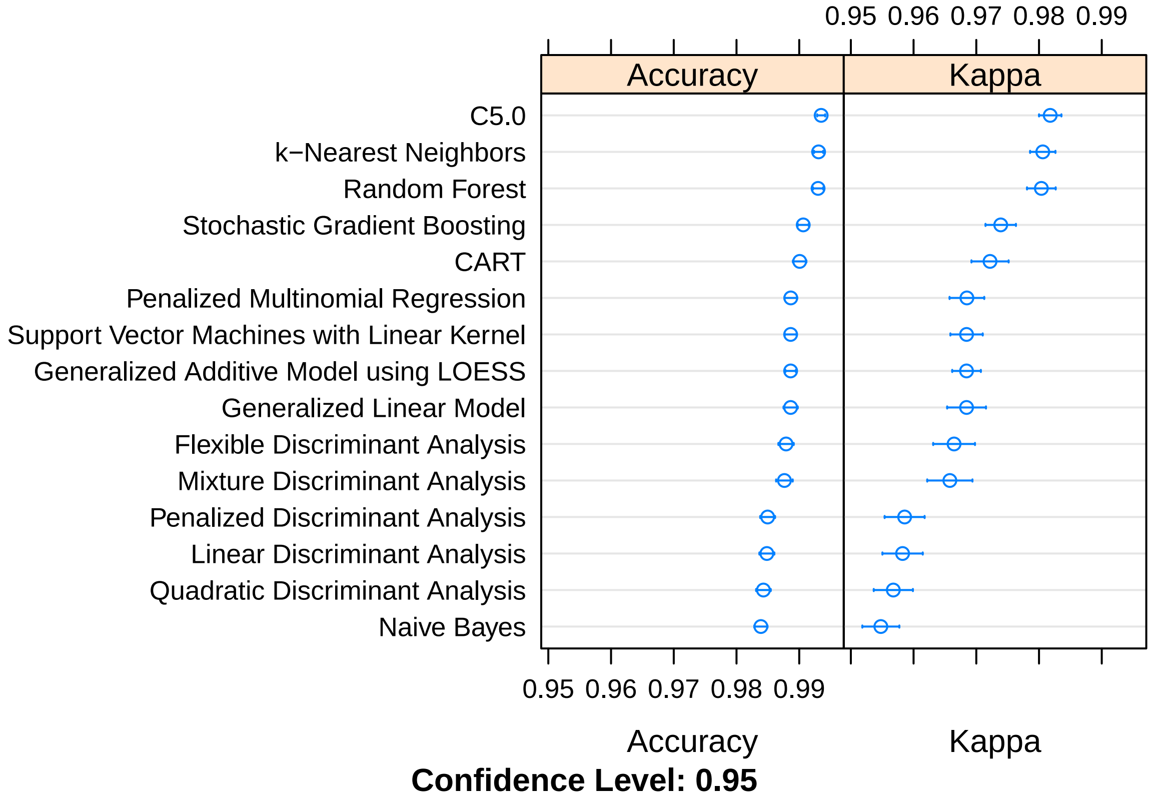
We will now assess the following models for our project as listed below:

- **Generalized Linear Model(glm)**
- **Linear Discriminant Analysis(lda)**
- **Naive Bayes(naive_bayes)**
- **Support Vector Machines with Linear Kernel(svmLinear)**
- **k-Nearest Neighbors(knn)**
- **Generalized Additive Model using LOESS(gamLoess)**
- **Penalized Multinomial Regression(multinom)**
- **Quadratic Discriminant Analysis(qda)**
- **Mixture Discriminant Analysis(mda)**
- **CART(rpart)**
- **Random Forest(rf)**
- **C5.0(C5.0)**
- **Flexible Discriminant Analysis(fda)**
- **Penalized Discriminant Analysis(pda)**
- **Stochastic Gradient Boosting(gbm)**

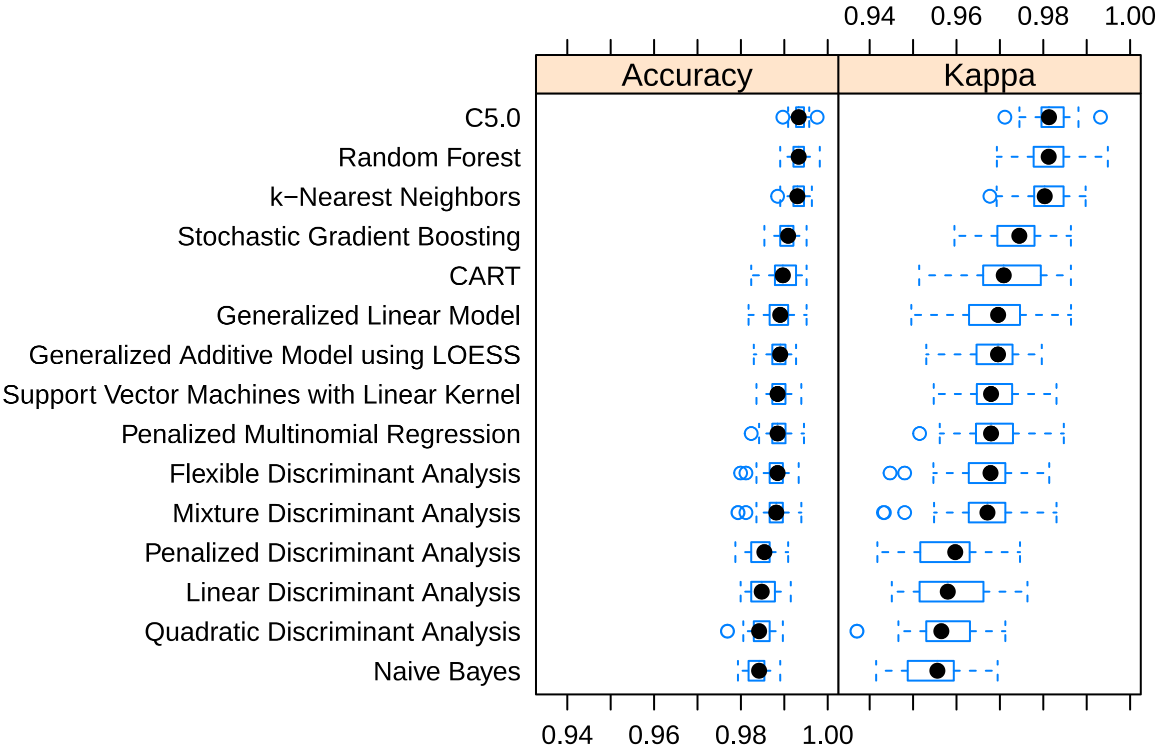
We will be using 10 fold cross validation with 3 repeats for this. We'll resample the results and plot the the same to identify the best model based on the accuracy.

```
# Create list of candidate models
models <- c("glm", "lda", "naive_bayes", "svmLinear",
            "knn", "gamLoess", "multinom", "qda", "mda",
            "rpart", "rf", "C5.0", "fda", "pda", "gbm")

# Run algorithms using 10-fold cross validation
control <-  trainControl(method = "repeatedcv", number = 10, repeats = 3,
```

```r
                         savePredictions = "final", classProbs = TRUE)
preProcess = c("center", "scale")

# Training models while suppressing the in-function messages
invisible(capture.output(fits <- lapply(models, function(model){
  print(paste0("Now training ", model, " model ..."))
  train(Occupancy ~ .-Date-HumidityRatio, method = model,
        data = occupancy_train, trControl = control,
        preProc = preProcess)
})))

# Resampling the results
results <- resamples(list("Generalized Linear Model" = fits[[1]],
                          "Linear Discriminant Analysis" = fits[[2]],
                          "Naive Bayes" = fits[[3]],
                          "Support Vector Machines with Linear Kernel" = fits[[4]],
                          "k-Nearest Neighbors" = fits[[5]],
                          "Generalized Additive Model using LOESS" = fits[[6]],
                          "Penalized Multinomial Regression" = fits[[7]],
                          "Quadratic Discriminant Analysis" = fits[[8]],
                          "Mixture Discriminant Analysis" = fits[[9]],
                          "CART" = fits[[10]],
                          "Random Forest" = fits[[11]],
                          "C5.0" = fits[[12]],
                          "Flexible Discriminant Analysis" = fits[[13]],
                          "Penalized Discriminant Analysis" = fits[[14]],
                          "Stochastic Gradient Boosting" = fits[[15]]),
                     decreasing = TRUE)

# Check the model accuracy
dotplot(results)
```



```r
# Accuracy with box whisker plots
bwplot(results)
```



```
accuracies <- c()

# Identifying maximum accuracies for each fit
for(ind in 1:length(fits)) {
  accuracies[ind] <- max(fits[[ind]]$results["Accuracy"])
}

# Identifying the index with the maximum accuracy
best_model_index <- which.max(accuracies)
best_model_name <- fits[[best_model_index]]$method

# Name of the best model
best_model_name
```

```
## [1] "C5.0"
```

We can see from the plots that $C5.0$ is the top model for this classification problem based on the maximum accuracy of 0.9934946. Let's now predict the occupancy in the test set to see how $C5.0$ works in generalization by making prediction and checking the accuracy.

Results

We will now run prediction on the test dataset with $C5.0$ and see the results.

```
# Assign best model
best_model <- fits[[best_model_index]]
```

```r
# Make predictions
occupancy_preds <- predict(best_model, occupancy_test)

# Final accuracy from the best model in our list of models
final_accuracy <- confusionMatrix(as.factor(occupancy_preds),
                                  occupancy_test$Occupancy)$overall["Accuracy"]
final_accuracy
```

```
##  Accuracy
## 0.9931907
```

And so, we get an accuracy of 0.9931907 with C5.0.

Conclusions

As part of our assessment we ran Automatic Feature Selection though RFE(Recursive Feature Elimination). We then analyzed 15 different models on the occupancy_train data set with 5 of the 6 features of our dataset. For this we used 10 fold cross validation with 3 repeats. We identified C5.0 as the best model with an accuracy of 0.9934946. We then predicted with C5.0 on our test data set to obtain a final accuracy of 0.9931907.

Limitations

1. Although this was a classification scenario, regression models were also used as candidate models for assessment.
2. Only Weekday derived from Date was used was used for assessment, whereas the time of the day could also have had an impact on the occupancy.
3. A single model was used as the final model, whereas a combination or ensemble would definitely have improved the accurace of the predictions.

Future work

1. Exploring the impact of time of day on the accuracy of the predictions.
2. Exploring Ensemble techniques to see if there would be an improvement in the accurace of predictions.

References

1. Luis M. Candanedo, Véronique Feldheim. Energy and Buildings. Volume 112, 15 January 2016, Pages 28-39
2. Feature Selection with the Caret R Package
3. Introduction to Data Science, Rafael A. Irizarry (2019)
4. Create Awesome LaTeX Table with knitr::kable and kableExtra, Hao Zhu (2019)
5. An Example R Markdown (2017)