University of East London

MSc. In Data Science

ADVANCED DECISION MAKING

DS7003

COURSEWORK

BY

DAMILOLA OLUWASENI OMISORE

2272153

ON

OCCUPANCY DETECTION USING MACHINE LEARNING

ABSTRACT

In building automation and energy management, occupancy detection is an essential responsibility. In this paper, I investigate the application of machine learning methods for commercial building occupancy detection. I assess how well several classification models—such as logistic regression, k-nearest neighbours, and naive bayes—perform. To forecast the occupancy state, we employ a collection of sensor values for temperature, humidity, CO2, and light levels. Accuracy, precision, recall, and F1 score criteria are used to assess the models. The logistic regression model, which I find has the best accuracy of 98.9%, suggests that it is an appropriate technique for occupancy detection. The findings demonstrates that effective occupancy detection has the potential to significantly reduce energy use and enhance building management. Future studies may examine the use of more sophisticated machine learning methods for applications like indoor air quality prediction and workplace design optimisation, in addition to occupancy detection.

Table of Contents

ABSTRACT 2

INTRODUCTION 5

METHODOLOGY 6

DATA ACQUISITION 6

DATA CLEANING AND DESCRIPTION 6

RESULT OF ANALYSIS 7

DATA EXPLORATION 7

MODELLING 12

LOGISTIC REGRESSION MODEL 12

K-NEAREST NEIGHBOUR MODEL 17

NAÏVE BAYES MODEL 20

CONCLUSION 22

REFERENCES 22

APPENDIX 23

Table of Figures

Figure 1: 2-D plots of all dependent variables. 7

Figure 2: Missingness map for the data set. 8

Figure 3: Box plots for all independent variables. 9

Figure 4: Violin plots for all the independent variables. 10

Figure 5: Histogram of Humidity, Light and CO_2 showing the normal curve. 11

Figure 6: Correlation plot for the data 12

Figure 7: KNN Model 1 accuracy plot. 17

Table of Boxes

Box 1: Summary of the independent variables 8

Box 2: Logistic regression (LR) model 1 of the dependent variable against all independent variables. 12

Box 3: The calculated odd ratio for Logistic Regression Model 1. 13

Box 4: McFadden's pseudo R-squared method. 14

Box 5: Confusion matrix and statistics for Logistic Regression Model 1. 14

Box 6: Logistic Regression (LR) Model 2. 15

Box 7: The calculated odd ratio for Logistic Regression Model 2. 16

Box 8: Confusion matrix and statistics for KNN Model 1. 18

Box 9: Confusion matrix and statistics for KNN Model 2 (All variables except "humidity ratio". 19

Box 10: Naïve Bayes Model 1 for all variables. 20

Box 11: Confusion matrix and statistics for Naïve Bayes Model 1. 20

Table of Tables

Table 1: Data set attribute table. 6

Table 2: Model Ranking table. 21

Table of Appendix

Appendix 1: Data Reading 23

Appendix 2: Data Exploration 23

Appendix 3: Logistic Regression 25

Appendix 4: K-NEAREST NEIGHBOUR 28

Appendix 5: NAÏVE BAYES 30

INTRODUCTION

In recent years, using sensor data to automate occupancy detection has gained popularity since it provides a number of advantages over more conventional approaches like human counting or motion detection. With the help of sensor-based occupancy detection, resources can be used more effectively, and buildings can be managed better since the data is more precise and dependable and there is real-time feedback on occupancy trends. Motion sensors, sound sensors, and environmental sensors including temperature, humidity, and CO2 sensors are just a few of the several types of sensors that may be utilised for occupancy detection. Depending on the occupancy patterns and required degree of precision, different types of sensors may be more suited for different sorts of areas. For occupancy detection based on sensor data, statistical learning models including decision trees, support vector machines, and artificial neural networks have been extensively employed. These models have the capacity to learn and adjust over time to changing patterns and can analyse various sensor inputs to produce precise forecasts of occupancy levels. Occupancy detection powered by machine learning may deliver more precise and trustworthy data as well as real-time feedback on occupancy trends, enabling more effective resource utilisation and better building management (Dai, Liu and Zhang, 2020).

From the end of 2015 to 2016, a project to accurately detect the occupancy of an office room using various environmental measurements such as light, temperature, CO_2 levels, and humidity, through the use of machine learning models by Candanedo l and Feldheim V was carried out. The project's aim was to create an accurate and automatic method for detecting occupancy in real time with the intention of solving the issue of energy efficiency and occupant comfort in an office building. Additionally, the research aimed to enhance the reliability of current occupancy detection techniques that depend on manual entry or motion sensors and are susceptible to mistakes and inconsistencies (Candanedo and Feldheim, 2016). A lot of manual processes have been used for this same objective which is to detect the occupancy of a room or building such as Infrared (IR) cameras, Passive infrared (PIR) sensors, Ultrasonic sensors, CO_2 sensors, and a lot more. But some of these have disadvantages like the cost implication, lack of sensitivity to movement, not being friendly to all living organisms, requiring calibration before usage, inertia and needing some time to show the correct results (*Occupancy Detection Methods - Resources | SoftServe*, no date). For the research, the authors compared the performance of several models using metrics such as accuracy, precision, recall, and F1-score to determine the best model for predicting occupancy in an office room. The models previously compared are decision tree, random forest, k-Nearest neighbour (KNN), support vector machines (SVM), and artificial neural networks (ANN). They then concluded that the Random Forest model performed most effectively at predicting the desired outcome. The Random Forest model outperformed the Decision Tree, KNN, SVM, and ANN models with an accuracy of 98.29%. The authors also discovered that measures of temperature and CO2 were more crucial in predicting occupancy than measurements of humidity and light. This implies that compared to humidity and light levels, the temperature and CO2 levels in an office environment have a stronger correlation with occupancy (Candanedo and Feldheim, 2016).

The aim of this work is to determine what machine learning model is the best for predicting if a room is occupied or not using commonly used factors such as temperature, light, humidity, humidity ratio, and CO_2 leveraging on models not previously used (specifically the Logistic regression model). These factors are considered because they are commonly utilised resources in previous research and have been proven to have a significant contribution to identifying whether a room or building is occupied or empty. The data was retrieved during the month of February 2015 by using several equipments such as a DTH22 sensor, Light sensor, CO_2 sensor, ZigBee radio, a microcontroller card and a digital camera controlled by a Raspberry Pi. The DTH22 sensor was used to take the reading of

the temperature in degrees Celsius (°C) and the humidity in percentage (%) while the humidity ratio was calculated in kg_w/kg_{da} using the measured temperature and relative humidity (Candanedo and Feldheim, 2016). To properly acquire readings, the measurements were carried out when the door was opened and closed, and the same for the window blinds. The measurements were taken at intervals of 14 s, or three to four times every minute, and then the average was calculated for that minute (Candanedo and Feldheim, 2016). All the data used have already been collected and I will give further description of the variables and where it was retrieved from.

METHODOLOGY

For this study, the research design is going to be experimental research where I will be comparing different machine learning models to identify which will be the best fit for predicting the occupancy of a room or building.

DATA ACQUISITION

The data was retrieved from a web page called "UCI Machine Learning Repository" which is a collection of databases, domain hypotheses, and data generators that the machine learning community uses to analyse machine learning algorithms empirically (*UCI Machine Learning Repository*, no date). The dataset to be used for this work is called the "Occupancy Detection Data Set" which is gotten from https://archive.ics.uci.edu/ml/datasets/Occupancy+Detection+. The data was donated and previously used by Luis Candanedo in the research called "Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models". The data contains attributes as earlier discussed which are temperature, light, humidity, humidity ratio, and CO_2 which are all the independent variables and then Occupancy is the dependent variable. The data set has multivariate and time-series characteristics and has a classification associated task with 20,560 (Twenty thousand five hundred and sixty) instances.

DATA CLEANING AND DESCRIPTION

The data set downloaded from the UCI web page was in a "txt" format. This was then converted to a CSV (Comma-Separate Values) file. The data set was already split into the training set and then two testing sets, but I decided to combine everything into one to randomise and split again. The combining of the data was carried out using Microsoft Excel and this was because the data set was already arranged in the same format, so the only action needed was to copy them all one after another into one CSV file and then sort them according to the date. The attributes and description are listed below.

***Table 1**: Data set attribute table.*

ATTRIBUTE	INFORMATION	TYPE OF VARIABLE
Date	date time year-month-day hour:minute:second	Independent
Temperature	Temperature, in Celsius	Independent
Humidity	Relative Humidity, %	Independent
Light	Light, in Lux	Independent
CO2	CO2, in ppm	Independent

Humidity Ratio	Humidity Ratio, Derived quantity from temperature and relative humidity, in kgwater-vapor/kg-air	Independent
Occupancy	Occupancy, 0 or 1, 0 for not occupied, 1 for occupied status	Dependent

This research is done based on previous research and data as there is a limitation in resources. A series of data exploration and visualisation techniques will be carried out on the data. Techniques like data missingness, duplicate check, and normality test for the exploration and then histogram, box plot, QQ plot, and violin plot for the visualisation, then machine learning models such as logistic regression models, decision tree, k-nearest neighbour, and random forest would be built and compared to identify the best model for predicting occupancy.

RESULT OF ANALYSIS

DATA EXPLORATION

The raw data set contains 20,560 instances which is before any form of analysis or actions. I then performed a simple 2-D plot on Microsoft Excel to check if there were any patterns in the data during collection probably due to season or weather. Made the plot in pairs of twos and then the humidity ratio as one because on their scaling.

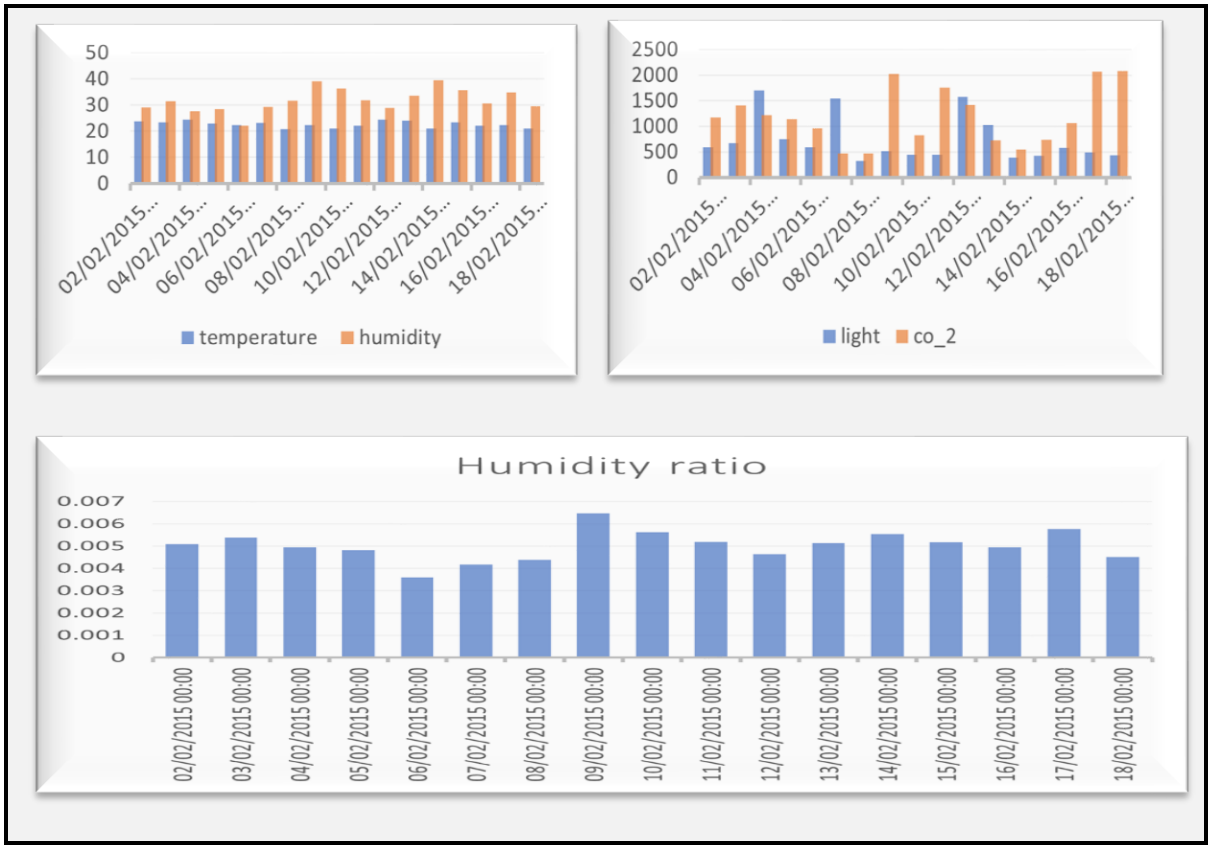

***Figure 1**: 2-D plots of all dependent variables.*

The readings were collected at certain times of each day in February 2015. The 2-D plot in Figure 1 above visualises the readings' pattern for each day. The temperature maintained the same or close to the same trend for each day and this could probably be because the same individuals come in

every day while the humidity and humidity ratio had little fluctuations but still had a relatively close trend. The light on the other hand had an up-down trend which is reasonable as the lights would not be in use during weekends as there are no activities while CO_2 also had an up-down trend which is not questionable.

Proceeding into the next phase, I went on to check if the data set had any missing data using RStudio.

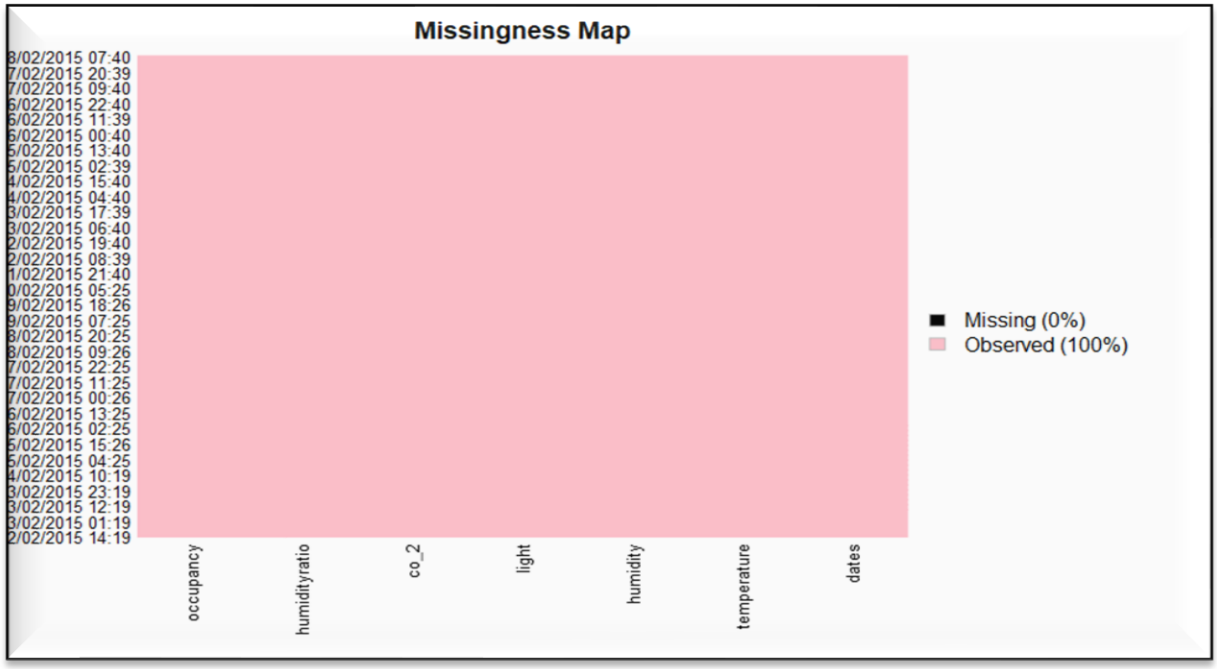

***Figure 2**: Missingness map for the data set.*

After the check for missing data, it was observed that there are no missing data as seen in Figure 2 above. This means that there is no need to either delete missing rows or fill in with one of the techniques for handling missing data. The next step is to check for duplicates in the data and the outcome of this was 54 duplicates which were then taken out of the data set on RStudio as shown in Appendix 3 which then resulted in 20,506 instances for the data set. I then proceeded to remove the irrelevant variable from the data set. The variable removed was the "dates" variable which was not needed for the remaining part of this research as it was not of significance to the work.

***Box 1**: Summary of the independent variables*

```
temperature       humidity         light             co_2        humidityratio
 Min.   :19.00   Min.   :16.75   Min.   :   0.0   Min.   : 412.8   Min.   :0.002674
 1st Qu.:20.20   1st Qu.:24.50   1st Qu.:   0.0   1st Qu.: 460.0   1st Qu.:0.003719
 Median :20.70   Median :27.29   Median :   0.0   Median : 565.7   Median :0.004291
 Mean   :20.91   Mean   :27.65   Mean   : 131.1   Mean   : 690.8   Mean   :0.004228
 3rd Qu.:21.52   3rd Qu.:31.29   3rd Qu.: 304.4   3rd Qu.: 805.0   3rd Qu.:0.004831
 Max.   :24.41   Max.   :39.50   Max.   :1697.2   Max.   :2076.5   Max.   :0.006476
>
```

From the summary, as shown in Box 1, I could easily identify what variable is normally distributed and which is not from the difference between the mean and the median. Temperature, humidity and humidity ratio had very little differences which means the variables had normal distributions. While light and CO_2 had significant differences between their mean and median meaning not normally distributed. This can also be confirmed using visualisation techniques such as box plots, histograms, QQ plots, and a lot more but I made use of just a couple to confirm my claims.

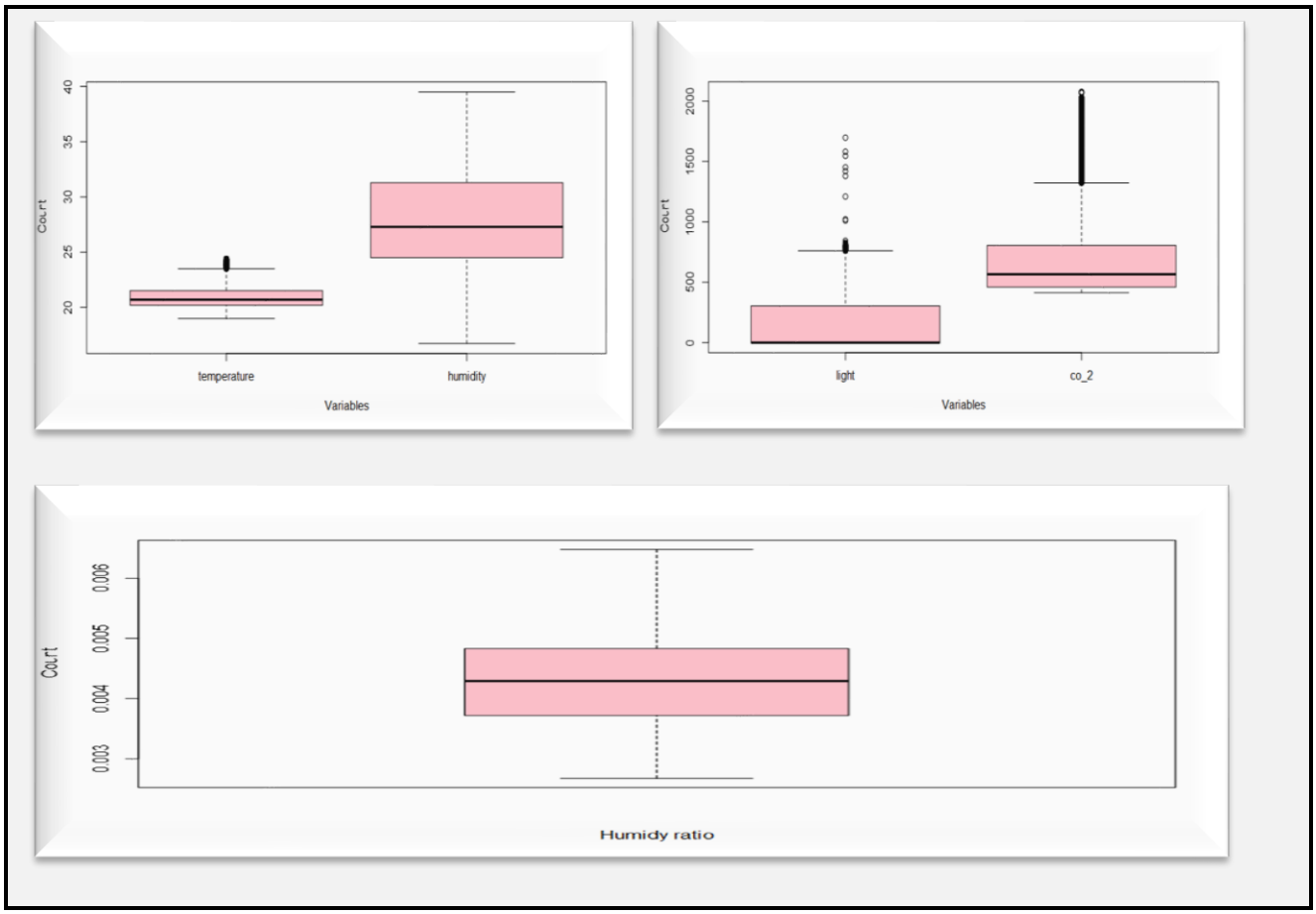

***Figure 3**: Box plots for all independent variables.*

As earlier claimed, temperature, humidity and humidity ratio look normally distributed but temperature has a couple of outliers. To better understand the box plot, I used the violin plot which provides a kernel density plot which offers a more informative and intuitive visualisation than the box plot alone.

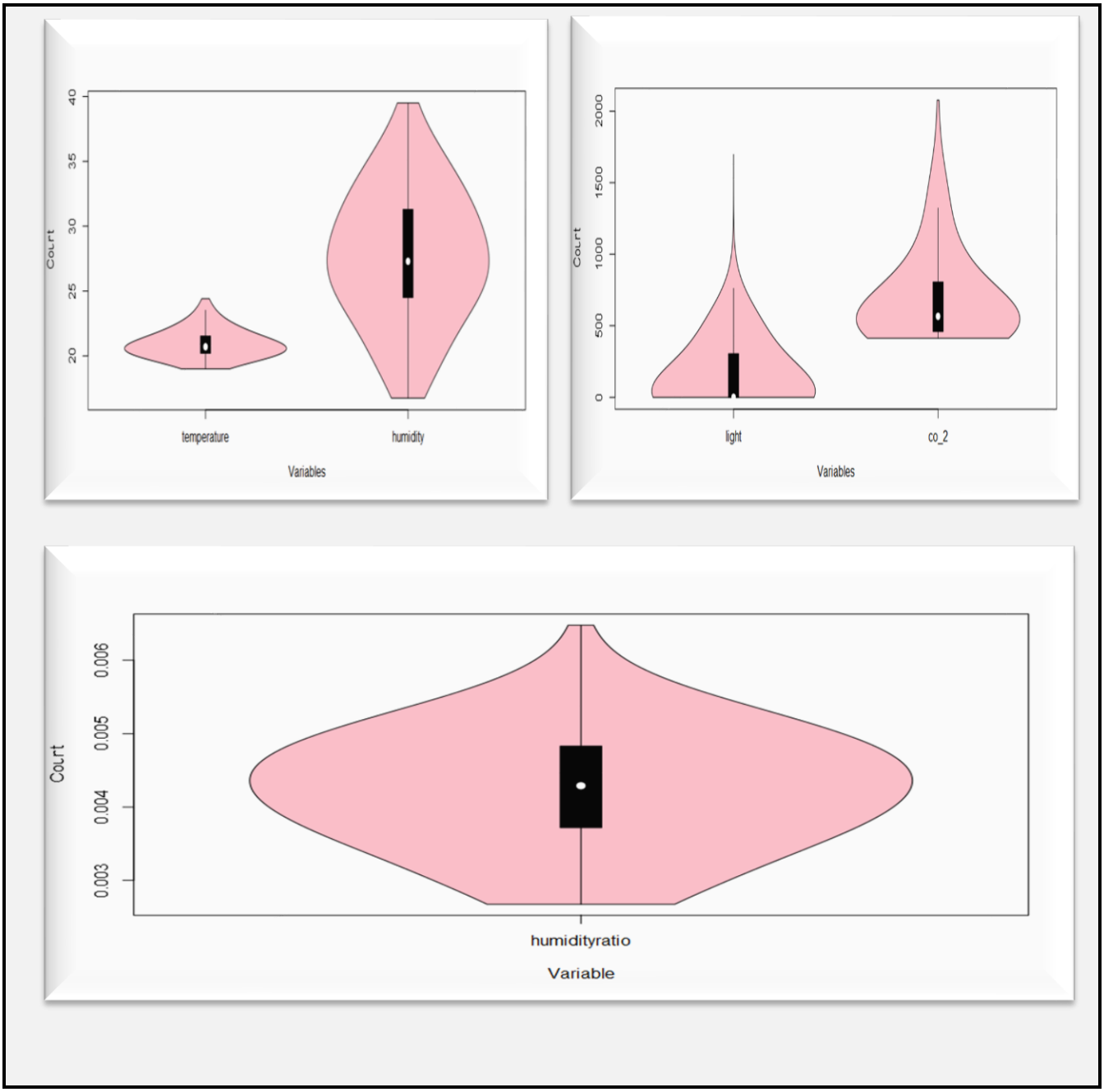

***Figure 4**: Violin plots for all the independent variables.*

From the violin plot in Figure 4, we can clearly notice that the light and $C0_2$ variables appear skewed and not normally distributed. To properly confirm the variables that are not normally distributed, I went forward to use histograms with density and normal curve.

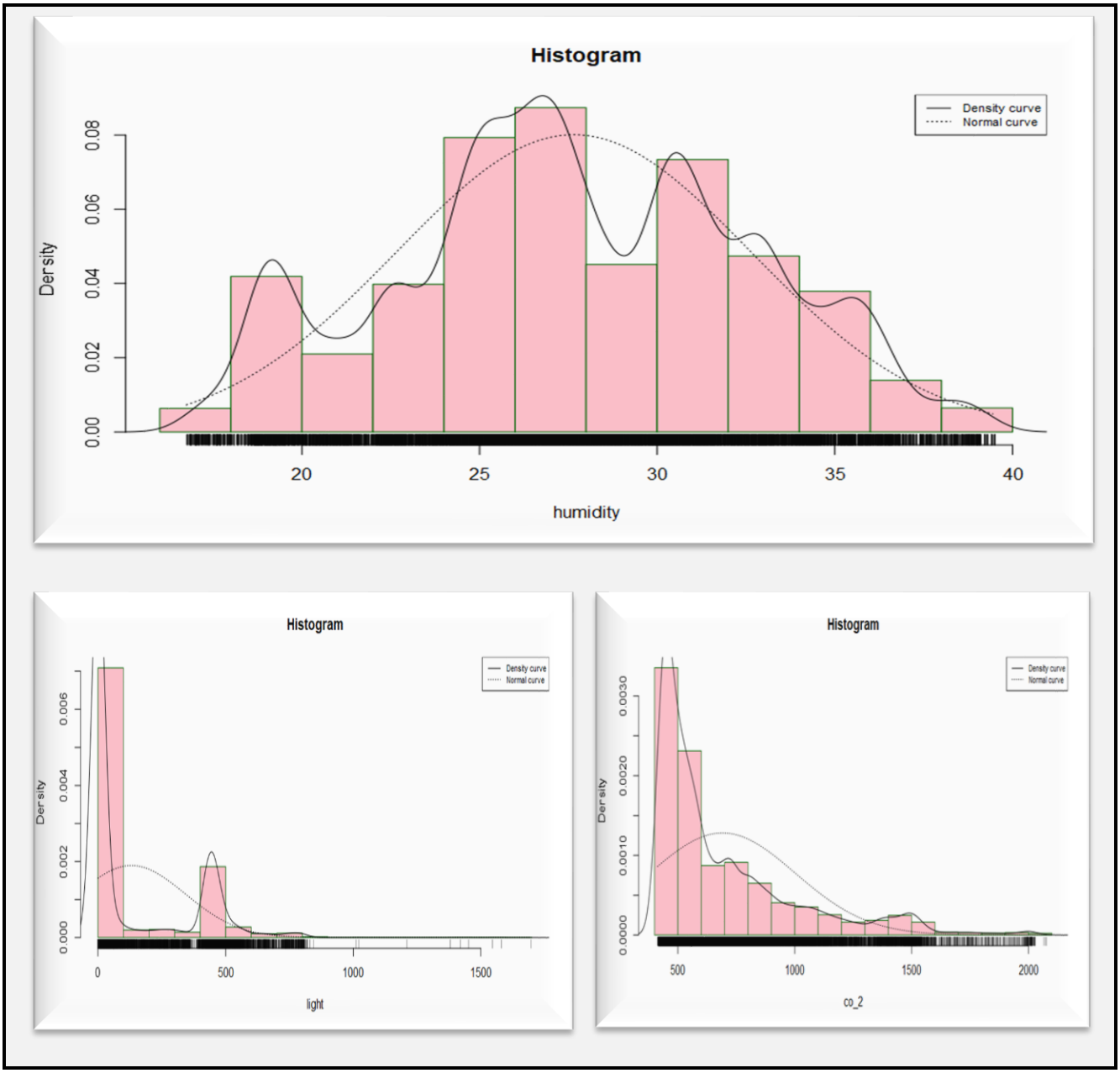

***Figure 5**: Histogram of Humidity, Light and CO_2 showing the normal curve.*

As previously mentioned, the light and $C0_2$ variables are not normally distributed. From Figure 4, we can see that they do not fall into the bell-shaped curve while the normally distributed humidity conforms to the normal curve. There is also a reason why the dependent variable is not being tested for normality, this is because it is a categorical type which consists of just 2 (two) factors and normality is a characteristic of continuous data. There is also no need to make data normally distributed because the algorithms used such as logistic regression, decision tree, and random forest can handle non-normal data and do not assume any specific distribution of the data.

The next phase was to check the correlation between the variables, and this was done by creating a correlation matrix on R.

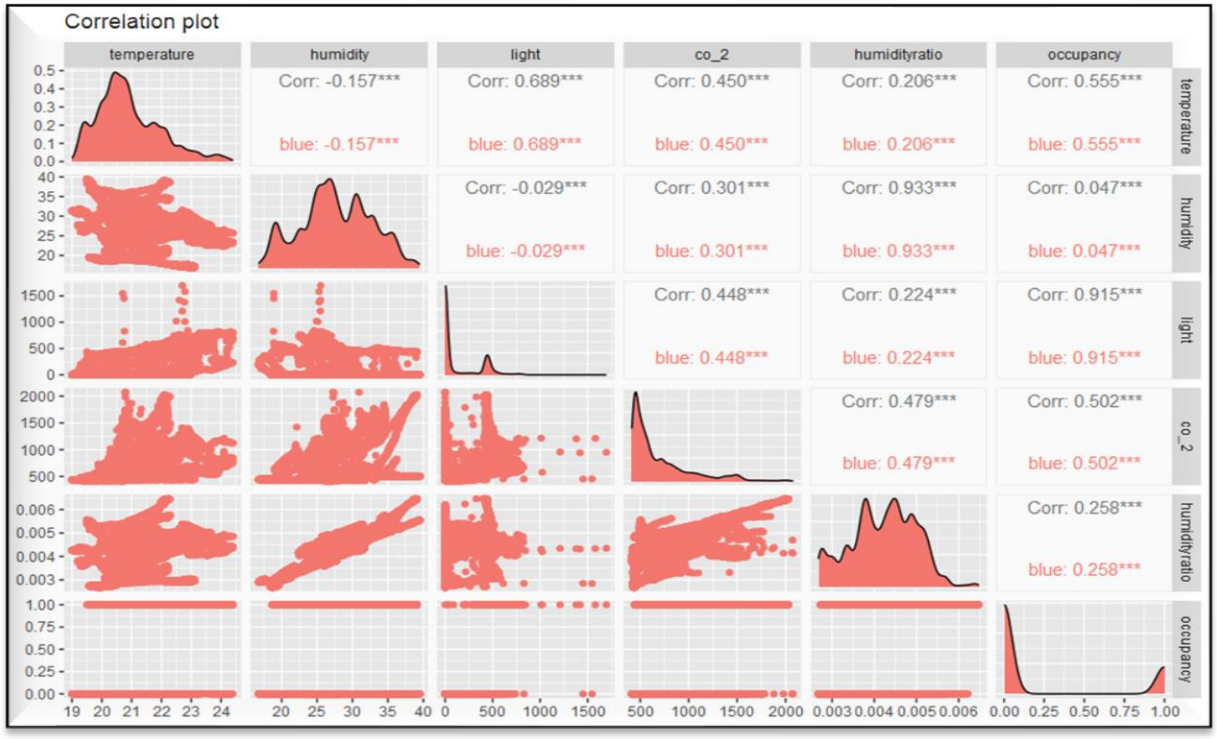

***Figure 6**: Correlation plot for the data.*

After checking the correlation, the independent variable that had a very strong correlation with the dependent variable is Light while humidity had the weakest correlation. It was also observed that there were also strong correlations between independent variables. Humidity and humidity ratio had a very strong correlation with each other. Also, a moderate to strong correlation was between temperature and light but not as strong as “humidity” and “humidity ratio”.

MODELLING

LOGISTIC REGRESSION MODEL

Logistic regression is a statistical method that is first used for this work. The reason is that it is a method used to model the probability of a binary outcome based on one or more predictor variables. In this logistic regression model, the outcome variable is categorical which is the “occupancy” variable, and all other variables are the predictor variables. The data was first split into 2, the training set which will be used to build the model and the test set for the prediction.

***Box 2**: Logistic regression (LR) model 1 of the dependent variable against all independent variables.*

```
Call:

glm(formula = occupancy ~ ., family = "binomial", data =
train_data)


Deviance Residuals:

    Min       1Q   Median       3Q      Max
```

```
-7.2209  -0.0416  -0.0263  -0.0160   4.2034

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)    7.310e+01  1.444e+01   5.064 4.11e-07 ***
temperature   -3.992e+00  6.731e-01  -5.931 3.02e-09 ***
humidity      -1.814e+00  3.999e-01  -4.536 5.74e-06 ***
light          2.371e-02  6.952e-04  34.109  < 2e-16 ***
co_2           3.561e-03  3.258e-04  10.932  < 2e-16 ***
humidityratio  1.192e+04  2.519e+03   4.733 2.22e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 15565.0  on 14291  degrees of freedom
Residual deviance:  1478.6  on 14286  degrees of freedom
AIC: 1490.6

Number of Fisher Scoring iterations: 9
```

The first model built consists of all the variables meaning the outcome is based on all the predictor variables. From the model, all independent variables have a very high significance in Logistic Model 1 which can be seen from the P-value as shown in Box 2. From the coefficients section, it is observed that the “temperature” and “humidity” variables have negative coefficients indicating that as “temperature” and “humidity” increases, the odds of the outcome reduce. While “light”, “CO_2”, and “humidity ratio” all have positive coefficients indicating that as “light”, “CO_2”and “humidity ratio” increases, the odds of the outcome increase. The model's residual deviation, 1478.6, is substantially lower than the null deviance, 15565.0, demonstrating the model's suitability for the data. The model's AIC, which assesses how well it fits in comparison to other models, is 1490.6 in total. This model fits the data quite well since lower AIC values suggest a good model fit.

***Box 3**: The calculated odd ratio for Logistic Regression Model 1.*

```
> exp(cbind(OR = coef(log_model1), confint(log_model1)))
Waiting for profiling to be done...
                          OR        2.5 %       97.5 %
(Intercept)    5.591684e+31 4.218124e+19 1.502116e+44
temperature    1.846351e-02 4.840753e-03 6.756825e-02
humidity       1.630422e-01 7.391949e-02 3.540381e-01
light          1.023995e+00 1.022662e+00 1.025459e+00
co_2           1.003568e+00 1.002923e+00 1.004207e+00
humidityratio           Inf          Inf          Inf
```

This result informs us that the chances of the outcome variable rise by a factor of 1.023995e+00 for every unit increase in "light" and also by a factor of 1.003568e+00 for every unit increase in "CO2", and that this impact is statistically significant at 95% confidence level (because the confidence interval does not contain 1). However, because the confidence interval includes 1, the chances of the outcome variable for "temperature" and "humidity" are 1.846351e-02 and 1.630422e-01 respectively, which are not statistically significant. I then evaluate the goodness of fit using McFadden's pseudo R-squared method which provides a measure of how well the model fits the data relative to a null model.

***Box 4**: McFadden's pseudo R-squared method.*

```
> pR2(log_model1)
fitting null model for pseudo-r2
          llh       llhNull            G2      McFadden          r2ML          r2CU
 -739.3096444 -7782.4911327 14086.3629766     0.9050035     0.6267892     0.9447129
```

McFadden's pseudo-R-squared does not depend on the magnitude of the dependent variable, unlike other R-squared measures like the Cox-Snell R-squared and the Nagelkerke R-squared. This makes it more reliable and suitable for use with outcomes that are categorical, continuous, or binary. After the McFadden's pseudo R-squared method, the value of R-squared is approximately 0.91 which means the model explains about 91% of the variation in the data compared to a null model indicating that the data is a good fit for the model. So, the next thing is to predict using the test data on the model and create a confusion matrix to test the accuracy.

***Box 5**: Confusion matrix and statistics for Logistic Regression Model 1.*

```
Confusion Matrix and Statistics
          Reference
Prediction    0    1
         0 4752    4
         1   62 1396

               Accuracy : 0.9894
```

```
                 95% CI : (0.9865, 0.9918)
    No Information Rate : 0.7747
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.97

 Mcnemar's Test P-Value : 2.28e-12

            Sensitivity : 0.9971
            Specificity : 0.9871
         Pos Pred Value : 0.9575
         Neg Pred Value : 0.9992
             Prevalence : 0.2253
         Detection Rate : 0.2247
   Detection Prevalence : 0.2346
      Balanced Accuracy : 0.9921

       'Positive' Class : 1
```

The model's accuracy is about 98.9% (0.9894), sensitivity of 99.7% (0.9971), precision of 95.7% (0.9575), and a F1 score of 97.7% (0.977) which is a very good result. But because there is internal correlation between predictor variables, the model can be assumed to be biased, leading to incorrect conclusions as there exists multicollinearity between the predictor variables. To confirm this, I then use the variance inflation factor (VIF) test on the model and the conclusion was that there was collinearity between "Humidity" and "Humidity ratio". I will then proceed to create a new model and take out the "humidity ratio" variable.

***Box 6**: Logistic Regression (LR) Model 2.*

```
Call:
glm(formula = occupancy ~ temperature + humidity + light + co_2,
    family = "binomial", data = train_data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-7.2062  -0.0437  -0.0228  -0.0144   4.1790
```

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  5.7533433  1.8636622   3.087  0.00202 **
temperature -0.8782829  0.0893947  -9.825  < 2e-16 ***
humidity     0.0824314  0.0183619   4.489 7.15e-06 ***
light        0.0242515  0.0006991  34.689  < 2e-16 ***
co_2         0.0036795  0.0003146  11.697  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 15565.0  on 14291  degrees of freedom
Residual deviance:  1502.5  on 14287  degrees of freedom
AIC: 1512.5

Number of Fisher Scoring iterations: 25
```

The second logistic model consists of all predictor variables except "humidity ratio" because of the multicollinearity from Logistic Model 1. As seen in Box 6 above, the P-value for all the predictor variables are less than 0.05 which concludes that they are all significant predictors in the model. From the Logistic Model 1, both "temperature" and "humidity" had negative coefficients while the others had positive coefficients. But for the LR Model 2, just the "temperature" variable has a negative coefficient while "humidity", "light" and "CO_2" have positive coefficients. The model is shown to be appropriate for the data by the residual deviation, which is 1502.5 for the model and 15565.0 for the null deviance. The overall AIC of the model, which measures how well it fits in relation to other models, is 1512.6. Although AIC values are greater than LR Model 1, it still indicates a strong model fit, and this model does a decent job of fitting the data.

***Box 7**: The calculated odd ratio for Logistic Regression Model 2.*

```
> exp(cbind(OR = coef(log_model2), confint(log_model2)))
Waiting for profiling to be done...
                    OR     2.5 %       97.5 %
(Intercept) 315.2428538 8.4166570 1.257491e+04
temperature   0.4154957 0.3476078 4.935384e-01
humidity      1.0859242 1.0477828 1.126047e+00
```

```
light          1.0245480 1.0232059 1.026020e+00
co_2           1.0036863 1.0030645 1.004304e+00
```

The outcome variable's chances increase by a factor of 1.0859242 for every unit increase in "humidity", by a factor of 1.0245480 for every unit increase in “light” and by a factor of 1.0036863 for every unit increase in "CO_2”, and this impact is statistically significant at a 95% confidence level (because the confidence interval does not contain 1). The odds of the outcome variable for the "temperature" variable is 0.4154957, which is not statistically significant. The accuracy, precision, recall and F1 score all result to the same as LR Model 1 meaning it had no change to the fitness, but one major advantage of this model is there are no internal correlations between the variable. To conclude with my logistic regression modelling, I created a third model by oversampling the data which made the data evenly distributed having same amount of instances for both outcomes and it resulted in a model with lesser accuracy and fitness as compared to LR Model 1 and 2.

K-NEAREST NEIGHBOUR MODEL

K-Nearest Neighbours (KNN) is a well-known technique for classification jobs because it is straightforward, simple to comprehend, and occasionally useful. Being a non-parametric approach, it makes no assumptions on the fundamental distribution of the data. For the first model built with KNN, I used all the predictor variables after scaling all continuous variables, used cross-validation and grid search to determine the best value of K.

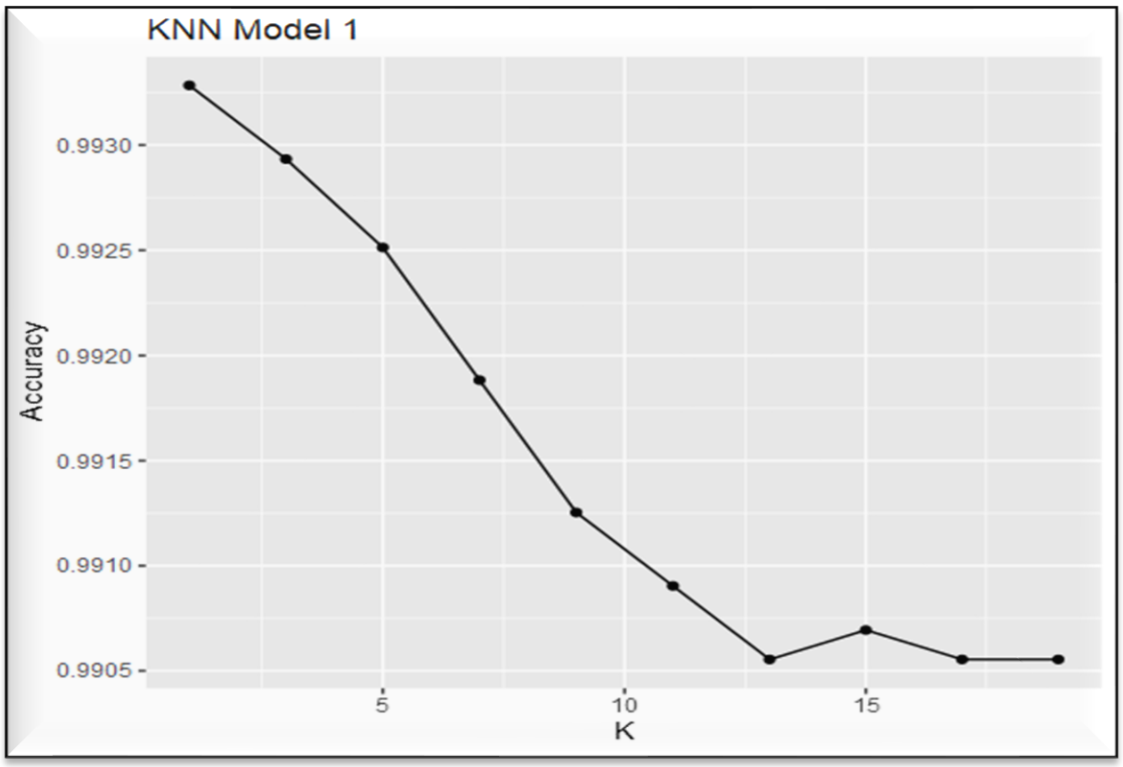

***Figure 7**: KNN Model 1 accuracy plot.*

The plot demonstrates how the value of K (the number of nearest neighbours to take into account) affects the accuracy of the KNN model. From the accuracy plot, it was easily concluded that the perfect value for K is 1. This is because any value above 9 had no significant increase in the accuracy of the model. To properly analyse the fitness of the model, I created a confusion matrix to help calculate the precision, recall and F1 score.

***Box 8**: Confusion matrix and statistics for KNN Model 1.*

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 4789   23
         1   25 1377

               Accuracy : 0.9923
                 95% CI : (0.9898, 0.9943)
    No Information Rate : 0.7747
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9779

 Mcnemar's Test P-Value : 0.8852

            Sensitivity : 0.9836
            Specificity : 0.9948
         Pos Pred Value : 0.9822
         Neg Pred Value : 0.9952
             Prevalence : 0.2253
         Detection Rate : 0.2216
   Detection Prevalence : 0.2256
      Balanced Accuracy : 0.9892

       'Positive' Class : 1
```

The accuracy of the model after the prediction is 99.2% (0.9923) which indicates that the model is performing well and is relatively good. The recall or sensitivity is 98.3% (0.9836), precision of 96.1% (0.9611) and the F1 score is 98.3% (0.983) which indicates that the model is making fewer false

positives and false negatives, respectively. Due to the fact that there is an internal correlation between “humidity” and “humidity ratio”, I will create a second model to test if there is any change in the accuracy.

***Box 9**: Confusion matrix and statistics for KNN Model 2 (All variables except “humidity ratio”.*

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 4755    6
         1   59 1394

               Accuracy : 0.9895
                 95% CI : (0.9867, 0.9919)
    No Information Rate : 0.7747
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9704

 Mcnemar's Test P-Value : 1.12e-10

            Sensitivity : 0.9957
            Specificity : 0.9877
         Pos Pred Value : 0.9594
         Neg Pred Value : 0.9987
             Prevalence : 0.2253
         Detection Rate : 0.2243
   Detection Prevalence : 0.2338
      Balanced Accuracy : 0.9917

       'Positive' Class : 1
```

The KNN Model 1 gave a more accurate prediction than the second KNN model with a 0.01% difference and the best value of K is 2. According to the confusion matrix, there were 1394 true positives (predicted occupied when actually occupied) and 4755 true negatives (predicted unoccupied when actually unoccupied) out of 6214 samples. There were also 59 false positives (predicted occupied when actually unoccupied) and 6 false negatives (predicted unoccupied when

actually occupied). The KNN Model 2 had an accuracy of 98.9% (0.9895), a recall of 99.5% (0.9957), a precision of 95.9 (0.9594) and an F1 score of 97.7 (0.977). Comparing the two KNN models, Model 2 is to be the best as it is of best fit having more accuracy, precision and F1 score.

NAÏVE BAYES MODEL

This technique is used because the Naive Bayes model can withstand certain irrelevant variables. Naive Bayes can still achieve decent accuracy even if certain features don't contribute much to the classification. Knowing irrelevant variables does not affect accuracy, I proceeded to building with all predictor variables.

***Box 10**: Naïve Bayes Model 1 for all variables.*

```
Naive Bayes


14292 samples
    5 predictor
    2 classes: '0', '1'


No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 12863, 12863, 12863, 12862, 12863, 12862,
...
Resampling results across tuning parameters:


  usekernel  Accuracy   Kappa
  FALSE      0.9669041  0.9120313
   TRUE      0.9789389  0.9416536
```

From the Naïve Bayes model, the accuracy of predicting FALSE (Not occupied) is 96.6% (0.9669) and for predicting TRUE (Occupied) is 97.8% (0.978). This is just the hypothetical result, so I will do the prediction with the test data and create the confusion matrix to get the actual overall accuracy.

***Box 11**: Confusion matrix and statistics for Naïve Bayes Model 1.*

```
Confusion Matrix and Statistics


          Reference
Prediction    0    1
         0 4740   60
         1   74 1340
```

```
               Accuracy : 0.9784
                 95% CI : (0.9745, 0.9819)
    No Information Rate : 0.7747
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9384

 Mcnemar's Test P-Value : 0.2614

            Sensitivity : 0.9571
            Specificity : 0.9846
         Pos Pred Value : 0.9477
         Neg Pred Value : 0.9875
             Prevalence : 0.2253
         Detection Rate : 0.2156
   Detection Prevalence : 0.2276
      Balanced Accuracy : 0.9709

       'Positive' Class : 1
```

Although the model is a good model, it has the lowest accuracy compared to other models created. The Naïve Bayes Model 1 has an accuracy of 97.8% which is still a strong one also a sensitivity to predicting "occupied" is 95.7% and to predicting "unoccupied" is 98.4%. This is most likely because the data is trained mostly with more samples on "unoccupied".

To further compare all models, I explore the accuracy, precision, recall and F1 score of all models.

***Table 2**: Model Ranking table.*

MODEL	ACCURACY	PRECISION	RECALL	F1 SCORE	RANK
LR Model 1	0.9894	0.9575	0.9971	0.977	4
LR Model 2	0.9894	0.9575	0.9971	0.977	2
KNN Model 1	0.9923	0.9611	0.9836	0.983	1
KNN Model 2	0.9895	0.9957	0.9594	0.977	3
Naïve Bayes Model 1	0.9784	0.9477	0.9571	0.952	5

After comparing all models, the Logistic Regression Model 2 consisting of all predictor variables except for "humidity ratio" had a good performance but the k-nearest neighbour is the best model with accuracy of 99.2%. This is because the accuracy is high and more importantly, it predicts the "Occupied" outcome the best with a correctness of 99.7%. These findings led us to the decision that

the k-nearest neighbour model was the most appropriate one to use for the "Occupancy detection" dataset. Although the accuracy of the logistic regression model was comparable, its lower precision indicates that it would be less correct for classes with “Occupied”. The k-nearest neighbour model also has the benefit of being easier to read, which may be significant in some applications.

CONCLUSION

In this work, we investigated the application of machine learning methods for office building occupancy detection. We assessed the effectiveness of a number of classification methods, including logistic regression, k-nearest neighbours and naïve bayes. The k-nearest neighbour model, which we discovered to has an accuracy of 99.2%, suggests that it is an appropriate technique for occupancy detection. Based on the result of this analysis, it is observed that the K-Nearest Neighbour model performed better than the Random Forest model used in the previous project to accurately occupancy detection of an office room by Candanedo l and Feldheim V. The Random Forest model had an accuracy of 98.29% while the K-Nearest Neighbour had an accuracy of 99.23%. The findings of this study have significant ramifications for energy management and building automation. Building managers may optimise the usage of lighting and HVAC (heating, ventilation, and air conditioning) systems by precisely identifying occupancy, which results in considerable energy savings. Although the methodologies and algorithms I employed in our study were restricted to a single commercial structure, they can be used in various buildings and environments. Future studies might look at the application of more sophisticated machine learning methods, such as deep learning and neural networks, for occupancy detection. Research can also look at additional uses of occupancy data, such as optimising workspace design and forecasting indoor air quality. In conclusion, machine learning techniques provide hope for commercial building occupancy detection. With reliable occupancy detection, the study shows the potential for large energy savings and enhanced facility management.

REFERENCES

Candanedo, L.M. and Feldheim, V. (2016) ‘Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models’, *Energy and Buildings*, 112, pp. 28–39. Available at: https://doi.org/10.1016/j.enbuild.2015.11.071.

Dai, X., Liu, J. and Zhang, X. (2020) ‘A review of studies applying machine learning models to predict occupancy and window-opening behaviours in smart buildings’, *Energy and Buildings*. Elsevier Ltd. Available at: https://doi.org/10.1016/j.enbuild.2020.110159.

Occupancy Detection Methods - Resources | SoftServe (no date). Available at: https://www.softserveinc.com/en-us/resources/occupancy-detection-methods (Accessed: 17 April 2023).

UCI Machine Learning Repository (no date). Available at: https://archive.ics.uci.edu/ml/index.php (Accessed: 13 April 2023).

APPENDIX

***Appendix 1**: Data Reading*

```
#-----1. Data Reading----------------------------------------


# set working directory
setwd(dirname(file.choose()))
getwd()


# read in data from csv file
Occ_data <- read.csv(file.choose())
head(Occ_data)
str(Occ_data)
```

***Appendix 2**: Data Exploration*

```
#-----2. Data Exploration-----------------------------------------
attach(Occ_data)


# check for missing data
apply(Occ_data, MARGIN = 2, FUN = function(x) sum(is.na(x)))
library(Amelia)
missmap(Occ_data, y.labels =dates, col = c("black", "pink"), legend = TRUE)


# remove duplicates
Occ_data <- unique(Occ_data)


# select relevant variables
Occ_data2 <- Occ_data[2:7]


# Checking if data is normally distributed
detach(Occ_data)
attach(Occ_data2)
# summary of variables
summary(Occ_data2[1:5])
```

```
# boxplot for variables
boxplot(Occ_data2[1:2], xlab="Variables", ylab="Count", col = "pink")
boxplot(Occ_data2[3:4], xlab="Variables", ylab="Count", col = "pink")
boxplot(Occ_data2[5], xlab="Humidy ratio", ylab="Count", col = "pink")

# create a violin plot of Sepal.Length by Species
library(vioplot)

vioplot(Occ_data2[1:2], col = "pink", xlab = "Variables", ylab = "Count")
vioplot(Occ_data2[3:4], col = "pink", xlab = "Variables", ylab = "Count")
vioplot(Occ_data2[5], col = "pink", xlab = "Variable", ylab = "Count")

#histogram to confirm normalization
#histogram for humidity
hist(humidity, col = "pink", border = "dark green", freq = F,
     xlab = "humidity", main = "Histogram")
rug (humidity)
lines (density(sort(humidity)))
xfit <- seq(from = min(humidity), to = max(humidity), by = 0.1)
yfit = dnorm(xfit, mean(humidity), sd(humidity))
lines(xfit, yfit, lty = "dotted")
rm(xfit, yfit)
legend("topright", legend = c("Density curve", "Normal curve"),
       lty = c("solid", "dotted"), cex = 0.7)

#histogram for light
hist(light, col = "pink", border = "dark green", freq = F,
     xlab = "light", main = "Histogram")
rug (light)
lines (density(sort(light)))
xfit <- seq(from = min(light), to = max(light), by = 0.1)
yfit = dnorm(xfit, mean(light), sd(light))
lines(xfit, yfit, lty = "dotted")
rm(xfit, yfit)
```

```
legend("topright", legend = c("Density curve", "Normal curve"),
       lty = c("solid", "dotted"), cex = 0.7)

#histogram for co_2
hist(co_2, col = "pink", border = "dark green", freq = F,
     xlab = "co_2", main = "Histogram")
rug (co_2)
lines (density(sort(co_2)))
xfit <- seq(from = min(co_2), to = max(co_2), by = 0.1)
yfit = dnorm(xfit, mean(co_2), sd(co_2))
lines(xfit, yfit, lty = "dotted")
rm(xfit, yfit)
legend("topright", legend = c("Density curve", "Normal curve"),
       lty = c("solid", "dotted"), cex = 0.7)
# correlation of all the variables
# correlation matrix
corr_test <- cor(Occ_data2, method = "pearson")
corr_test <- round(corr_test, digits = 2)

# correlation plot
install.packages("rlang")
update.packages("rlang")
library(ggplot2)
library(GGally)
GGally::ggpairs(Occ_data2)
ggpairs(Occ_data2, columns = 1:6,
                ggplot2::aes(colour="blue"),
                upper = list(continuous = "cor"),
                title = "Correlation plot")
```

***Appendix 3**: Logistic Regression*

```
#-----3. BINOMIAL LOGISTIC REGRESSION-----------------------------------------
-

# make Occupancy a factor

Occ_data2$occupancy <- factor(Occ_data2$occupancy)

str(Occ_data2)


# confirm data is balanced (For imbanced, take not of Accuracy & Sensitivity)

table(Occ_data2$occupancy)


# data partitioning

set.seed(246)

part <- sample(2, nrow(Occ_data), replace = T, prob = c(0.7, 0.3))


train_data <- Occ_data2[part==1,]

test_data <- Occ_data2[part==2,]


# model training

log_model1 = glm(occupancy ~., data = train_data, family = "binomial")

summary(log_model1)


# Calculate Odds Ratio - Exp(b) with 95% confidence intervals (2 tail)

exp(cbind(OR = coef(log_model1), confint(log_model1)))


#Evaluate the goodness of fit

library(pscl)

pR2(log_model1)


# reiduals check

plot(log_model1$residuals)


# model test

library(caret)

log_pred1 <- predict(log_model1, newdata = test_data, type = "response")

log_pred1 <- factor(ifelse(log_pred1 > 0.5, 1, 0), levels = c(1, 0))

log_pred1
```

```
conf_matrix1 <- confusionMatrix(log_pred1, test_data$occupancy, positive = "1")
conf_matrix1


log_recall1 <- conf_matrix1$byClass["Recall"]
log_f1_1 <- conf_matrix1$byClass["F1"]



# test for multicollinearity
library(car)
vif(log_model1)
sqrt(vif(log_model1)) > 2



# LOGISTIC MODEL 2
log_model2 = glm(occupancy ~ temperature + humidity + light + co_2, data =
train_data, family = "binomial")
summary(log_model2)


# Calculate Odds Ratio - Exp(b) with 95% confidence intervals (2 tail)
exp(cbind(OR = coef(log_model2), confint(log_model2)))


#Evaluate the goodness of fit
pR2(log_model2)



plot(log_model2$residuals)


log_pred2 <- predict(log_model2, newdata = test_data, type = "response")
log_pred2 <- factor(ifelse(log_pred2 > 0.5, 1, 0), levels = c(1, 0))
log_pred2
conf_matrix2 <- confusionMatrix(log_pred2, test_data$occupancy, positive = "1")
conf_matrix2


recall2 <- conf_matri2$byClass["Recall"]
f1_2 <- conf_matrix2$byClass["F1"]
```

```
vif(log_model2)
sqrt(vif(log_model2)) > 2

# LOGISTIC MODEL 3 with OVERSAMPLING
table(train_data$occupancy)
over_train_data = upSample(train_data, train_data$occupancy)
table(over_train_data$occupancy)

log_model3 = glm(occupancy ~ temperature + humidity + light + co_2, data =
over_train_data, family = "binomial")
summary(log_model3)

vif(log_model3)
sqrt(vif(log_model3)) > 2

plot(log_model3$residuals)

log_pred3 <- predict(log_model3, newdata = test_data, type = "response")
log_pred3 <- factor(ifelse(log_pred3 > 0.5, 1, 0), levels = c(1, 0))
log_pred3
conf_matrix3 <- confusionMatrix(log_pred3, test_data$occupancy, positive = "1")
conf_matrix3

recall3 <- conf_matrix3$byClass["Recall"]
f1_3 <- conf_matrix3$byClass["F1"]
```

***Appendix 4**: K-NEAREST NEIGHBOUR*

```
#-----6. K NEAREST NEIGHBOUR-----------------------------------------
install.packages("class")
library(class)

k_values <- data.frame(k = seq(1, 20, by = 2))
```

```
attach(Occ_data2)
Occ_data3 <- scale(cbind(temperature, humidity, light, co_2, humidityratio))
Occ_data3 <- merge(Occ_data3, Occ_data2[6])
rm(Occ_data3)

train_data2 <- train_data
test_data2 <- test_data

train_data2$temperature <- scale(train_data2$temperature)
test_data2$temperature <- scale(test_data2$temperature)
train_data2$humidity <- scale(train_data2$humidity)
test_data2$humidity <- scale(test_data2$humidity)
train_data2$light <- scale(train_data2$light)
test_data2$light <- scale(test_data2$light)
train_data2$co_2 <- scale(train_data2$co_2)
test_data2$co_2 <- scale(test_data2$co_2)
train_data2$humidityratio <- scale(train_data2$humidityratio)
test_data2$humidityratio <- scale(test_data2$humidityratio)

# Create a KNN model using cross-validation and grid search
knn_model1 <- train(occupancy ~ ., data = train_data2, method = "knn",
                    trControl = trainControl(method = "cv", number = 10),
                    tuneGrid = k_values)
knn_model1$results

# Plot the accuracy as a function of K
ggplot(data = knn_model1$results, aes(x = k, y = Accuracy)) +
  geom_line() +
  geom_point() +
  labs(x = "K", y = "Accuracy") +
  ggtitle("KNN Model 1")

# Predict the test set using the trained model
knn_pred1 <- predict(knn_model1, test_data2)
```

```
knn_pred1


conf_matrix4 <- confusionMatrix(knn_pred1, test_data2$occupancy, positive =
"1")

conf_matrix4


knn_f1_1 <- conf_matrix4$byClass["F1"]




# Create a KNN model using cross-validation and grid search

knn_model2 <- train(occupancy ~ temperature + humidity + light + co_2, data =
train_data2, method = "knn",

                    trControl = trainControl(method = "cv", number = 10),

                    tuneGrid = k_values)

knn_model2$results


# Plot the accuracy as a function of K

ggplot(data = knn_model2$results, aes(x = k, y = Accuracy)) +

  geom_line() +

  geom_point() +

  labs(x = "K", y = "Accuracy") +

  ggtitle("KNN Model 2")


# Predict the test set using the trained model

knn_pred2 <- predict(knn_model1, test_data2)

knn_pred2


conf_matrix5 <- confusionMatrix(knn_pred2, test_data2$occupancy, positive =
"1")

conf_matrix5


knn_f1_2 <- conf_matrix5$byClass["F1"]
```

***Appendix 5**: NAÏVE BAYES*

```
#-----7. NAIVE BAYES----------------------------------------

install.packages("naivebayes")

install.packages("e1071")

library(naivebayes)

library(caret)

library(e1071)


# Train the Naive Bayes model

nb_model1 <- train(occupancy ~ ., data = train_data, method = "naive_bayes",
trControl = trainControl(method = "cv", number = 10), na.action = na.pass,
tuneLength = 10)

nb_model1


# Predict the test set using the trained model

nb_pred1 <- predict(nb_model1, test_data)

nb_pred1


# Confusion matrix

conf_matrix7 <- confusionMatrix(nb_pred1, test_data$occupancy, positive = "1")

conf_matrix7


nb_f1_1 <- conf_matrix7$byClass["F1"]
```