Using Sensor Data and ML to Estimate Room Occupancy

Final Results and Project Report

Jeffrey Fitzpatrick

500728133

Supervisor: Ceni Babaoglu

December 2, 2024

Ryerson University

Table of Contents

Abstract **3**
Problem 3
Research Questions 3
Data 3
Techniques and Tools 4
Contribution of Work Compared to Past Research **4**
Introduction 4
Previous Studies on Occupancy Detection 4
Previous Studies on Occupancy Estimation 5
Recent Studies on Occupation Detection and Estimation 6
Additional Studies 8
Conclusion 8
GitHub Repository **10**
Applied Methodology and Study Design **11**
Initial Analysis **12**
Univariate Analysis 12
Data Dictionary 12
Dependent (target) Variable 13
Summary Statistics 13
Frequency of Categorical Variables 14
Bivariate Analysis 15
Pairwise Visualizations 15
Time Series Analysis 16
Correlation Analysis 18
Exploratory Analysis **19**
Dimensionality Reduction **21**
Experimental Design **23**

Multinomial Logistic Regression 23
All Features 23
PCA Features 25
Random Forest 25
All Features 25
PCA Features 27
Support Vector Machine 28
All Features 28
PCA Features 29
Findings 30
Summary of Model Results 30
Summary of Sensor Data 30
Limitations and Future Work 31
References 32

Abstract

Problem

To provide comfort for occupants, commercial buildings rely on heating, ventilation, and air conditioning (HVAC) systems and lighting systems. Large amounts of energy are wasted, especially during non-working hours (Masoso and Grobler, 2010).

To optimize these systems, a research paper (Singh et al., 2018) described an experiment for accurately estimating the number of occupants in a room using “non-intrusive” environment sensors and machine learning (ML) models. Multiple sensor nodes were placed throughout a 6m by 4.6m test room in a wireless sensor network (WSN). Low-cost sensors were deployed at each of the four desks to measure temperature, light, and sound. A carbon dioxide (CO_2) sensor was deployed in the middle of the room to provide the most accurate reading. Two motion detection sensors were deployed on the ceiling, above the door and large window. Because of privacy concerns, video-based systems are not considered appropriate for detecting occupancy.

Some previous studies focused on using occupancy detection (i.e., determining whether a room is occupied or not) to save energy. On the other hand, the goal of ML occupancy estimation research is to design adaptive systems that can detect the exact number of occupants, resulting in additional energy savings and improved comfort for occupants.

Research Questions

For my project, I chose the theme of Classification for building predictive models.

To reduce energy consumption in buildings, I investigated these research questions:

- Which of the implemented supervised learning techniques perform the best in predicting occupancy?
- Which types of sensor data (temperature, light, sound, motion, CO_2) show the most promising results?
- Based on the research, what alternative types of sensor data could be used for ML-based occupancy estimation?

Data

For my project, I chose the Room Occupancy Estimation dataset, available from the UC Irvine Machine Learning Repository. The Room Occupancy Estimation dataset can be downloaded from https://archive.ics.uci.edu/dataset/864/room+occupancy+estimation.

The dataset contains over 10000 data points and 16 features. Each feature represents data (temperature, light, sound, motion, or CO2) from a particular sensor. Measurements were recorded over several days in 30 second intervals. The actual occupancy was established by having participants register and record the exact time each time they entered or left the room.

Techniques and Tools

To solve the stated problem, I implemented supervised learning algorithms including Multinomial Logistic Regression, Random Forest, and Support Vector Machine (SVM). In addition, I investigated how an unsupervised learning algorithm named Principal Component Analysis (PCA) can be used for dimensionality reduction.

For this project, I used R for the initial data analysis and Python for exploratory analysis, dimensionality reduction, and implementation of the machine learning algorithms. All the supervised and unsupervised learning algorithms were implemented using the scikit-learn open-source library. To evaluate the models, I used multiple performance metrics including accuracy, confusion matrices, and F1-score.

Contribution of Work Compared to Past Research

Introduction

As part of efforts to reduce energy wasted in buildings and combat climate change, ML-based occupancy estimation research has increased in recent years (Li et al., 2024). Before selecting my dataset and conducting this review, I did not know anything about this topic. I know that office buildings use a lot of energy for heating, cooling, and lighting. Much of this energy is wasted, especially in this post-COVID-19 era of hybrid work and half-empty offices.

In this review, I have summarized papers related to this research. I first summarized research papers in occupancy detection and estimation that were published before 2018. I then summarized more recent papers, starting with the introductory paper (Singh, et al., 2018) for my chosen dataset. I finished by summarizing a couple of papers that take a broader perspective than an individual study.

Previous Studies on Occupancy Detection

Some previous studies focused on using occupancy detection to save energy. The goal of occupancy detection is to determine whether a room is occupied or not. Unlike occupancy estimation, occupancy detection does not try to determine the actual number of occupants in a room at any one time.

In one study (Hailemariam et al., 2011), the authors deployed multiple low-cost sensors within an office cubicle to measure light, sound, CO_2, power use, and motion. After collecting data for a week, the authors used the Decision Tree classification method to predict whether the cubicle was occupied. In their study, features related to motion performed the best in predicting the presence of a worker. Light features, on the other hand, performed the worst. Combining features had mixed results. Notably, none of the feature combinations outperformed the features derived from using motion sensors alone. The study did not explore classification methods other than Decision Tree.

In (Candanedo & Feldheim, 2016), the authors used data from light, temperature, humidity, and CO_2 sensors to predict whether an office room was occupied. The classification models were tested under two data sets, depending on whether the office door was closed. The authors noted that including timestamps in the models led to better results in most cases. Accuracies of 97% or higher were achieved using Linear Discriminant Analysis (LDA) with two predictors (e.g., temperature and light) and Classification and Regression Trees (CART) with light as the top node.

This table summarizes these occupancy detection studies:

Study	Sensor Data	Classification Methods	Data	Evaluation Metrics	Best Results
[3]	Light, sound, CO_2, power use, motion	Decision Tree	7 days	Accuracy	Motion features (98.4%)
[4]	Light, temperature, humidity, CO_2	CART Random Forest GBM LDA	2 days	Accuracy	LDA with two predictors (97%) CART (97% or higher)

Previous Studies on Occupancy Estimation

The goal of occupancy estimation research is to design systems that can determine the actual number of occupants in a room at any one time. As previously mentioned, demand-driven HVAC and lighting systems can result in additional energy savings and improved comfort for occupants.

In (Dong et al., 2010), three sensor networks were deployed in an open-plan office building:

- Gas detection sensor network to measure pollutants such as CO_2
- Wireless ambient sensor network to measure lighting, temperature, humidity, motion, and sound
- Independent CO_2 sensor network

During feature selection, the authors determined that the CO_2 features had the largest information gain. As a result, these features were used as inputs to the classification models: Hidden Markov Model (HMM), Artificial Neural Network (ANN), and Support Vector Machine (SVM). The authors concluded that HMM performed the best overall with an accuracy of 75%.

In (Yang et al., 2012), the authors used data from temperature, humidity, CO_2, light, sound, and motion sensors to estimate the number of occupants in two shared lab spaces. After collecting data from 20 days, the authors used radial basis function (RBF) neural network for classification. They reported an average detection rate of 87.62% for self-estimation (model is trained and tested using the same lab) and 64.83% for cross-estimation (model is trained and tested using different labs). To account for rounding errors produced by the model, a tolerance

of 1 was used. For example, if the model predicted 2.8 occupants, an error was not reported if the actual number of occupants was 2 or 3.

This table summarizes these occupancy estimation studies:

Study	Sensor Data	Classification Methods	Data	Evaluation Metrics	Best Results
[5]	CO_2, light, temperature, humidity, motion, sound,	HMM ANN SVM	14 days	Accuracy	HMM with CO_2 features as input (75%)
[6]	Temperature, humidity, CO_2, light, sound, motion	RBF neural network	20 days	Accuracy (tolerance = 1)	Self-estimation (87.62%)

Recent Studies on Occupation Detection and Estimation

Research by Singh et al. (2018) is the basis for this project. In that study, the authors deployed multiple light, temperature, sound, and CO_2 sensors in a test room as described in the Abstract. To estimate the number of occupants in the room, four classification models were used: Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), Support Vector Machine (SVM), and Random Forest. For SVM, the results were evaluated with both a linear and radial basis function (RBF) kernel. With all features included, SVM with RBF kernel performed the best, with an accuracy of 98% and an F1-score of 95%. After employing Principal Component Analysis (PCA), the authors concluded that an accuracy of 92% and F1-score of 72% was achievable with only four components.

In (Wang et al., 2021), the authors proposed a cost-effective, non-intrusive occupancy detection system that they said could be easily installed in residential buildings. Installed in a living lab, the system used temperature and motion sensors to detect human activities (such as opening the front door or running water) over 54 days. Data on the human activities was then used to train and test four classification models: Random Forest, Decision Tree, K-Nearest Neighbor, and Support Vector Machine (SVM). The authors concluded that Random Forest performed the best overall, with an accuracy and F1-score of 98% or higher.

In (Kim et al., 2023), the authors used Internet of Things (IoT) sensors to estimate the number of occupants in a living lab over 55 days. Overnight and weekend data were excluded as no occupants were in the lab at that time. Random Forest and Artificial Neural Network (ANN) classification models were used, with data from CO2 concentration, differential pressure (air flow), and the ventilation system state used as inputs. For predicting occupancy, the authors concluded that the Random Forest model had the lowest root mean square error (RMSE) when the ventilation system state data was added to CO_2 concentration as input values. Conversely, including differential pressure data tended to decrease accuracy and increase RMSE in the models.

In (Mao et al., 2023), the authors adapted a predictive framework for room occupancy using the same dataset that is the focus of this project. Several classification methods were used: Logistic Regression, Linear Discriminant Analysis (LDA), Support Vector Machine (SVM), Multi-layer Perceptron (MLP), LightGMB, XGBoost, and Random Forest. The models were evaluated using balanced accuracy, F1-score, and Area Under ROC Curve (AUC). Among the methods, Random Forest performed the best in all three metrics. In the Random Forest model, the light values from sensor 1 and sensor 2 had the largest impact in predicting room occupancy. The authors claimed their results improved on the performance of the original paper, with a balanced accuracy and AUC above 99% and F1-score above 98%.

In (Banihashemi et al., 2024), the authors deployed IoT sensors in two rooms of an office building over multiple months. The sensors collected CO_2, temperature, relative humidity, indoor air quality, sound pressure level, and light data. The authors then trained Random Forest, XGBoost, and dense feedforward neural network (DFNN) on the dataset to determine the best features. The best features were then used to reduce dimensionality for more complex, sequential models. For occupancy detection, the authors concluded that using six days of sound pressure level, CO_2, and light data could achieve an accuracy above 95% and an F1-score above 93%.

This table summarizes these occupancy detection and estimation studies:

Study	Sensor Data	Classification Methods	Data	Evaluation Metrics	Best Results
[2]	Temperature, light, sound, motion, CO_2	LDA QDA SVM Random Forest	7 days	Accuracy F1-score Confusion Matrix	SVM (RBF) with all features (Accuracy = 98%; F1-score = 95%)
[7]	Temperature, motion	Random Forest Decision Tree K-Nearest Neighbor SVM	54 days	Accuracy F1-score	Random Forest (98% or higher for both)
[8]	CO_2 concentration, differential pressure, ventilation system state	Random Forest ANN	55 days	Accuracy RMSE	Random Forest with ventilation system state added to CO_2 concentration (RMSE = 1.462)
[9]	Temperature, light, sound, motion, CO_2	Logistic Regression LDA SVM MLP LightGBM XGBoost Random Forest	7 days	Balanced Accuracy F1-score AUC	Random Forest Accuracy (>99%) F1-score (>98%) AUC (>99%)

Study	Sensor Data	Classification Methods	Data	Evaluation Metrics	Best Results
[10]	CO2, temperature, humidity, air quality, sound, light	Random Forest XGBoost DFNN	80 days, 20 days	Accuracy, F1-score	Office A: DFNN (Accuracy = 97%, F1-score = 95%) Office B: XGBoost (Accuracy = 95%, F1-score = 94%) Sound pressure, CO2, light (6 days)

Additional Studies

The research papers cited in this review present specific studies that use sensor data and machine learning techniques to detect or estimate room occupancy in a test environment. Other studies take a broader perspective.

In (Li et al., 2024), the authors reviewed the development of data collection methods and predictive algorithms. To enhance data collection, the authors advocated using new Internet of Things (IoT) technology such as Bluetooth signals, Wi-Fi, camera images, and GPS data. To increase prediction accuracy, they advocated for further research into hybrid machine learning models. The authors noted that interest in occupancy prediction research has increased since 2012, except during the COVID-19 pandemic.

In (Khan et al., 2024), the authors reviewed the advantages and limitations of data collection methods for occupancy detection, estimation, and prediction. Newer technologies include UWB radar, Bluetooth low energy (BLE), Wi-Fi, cameras, and electric meters. For example, UWB radar technology has high precision and can be used to detect the movements of people; limitations include their high cost and potential for privacy issues. The authors also discussed how combining data from different types of sensors into a unified system can increase prediction accuracy.

Conclusion

As can be seen from this review, studies varied the types of sensors, classification methods, evaluation metrics, and the amount of data collected. There were also differences in the number of sensors deployed and the type, size, and number of test rooms that were used. Two types of experiments were performed: occupancy detection (determining whether a room was occupied) and occupancy estimation (determining the exact number of occupants at any one time).

The focus of this project is occupancy detection using the Room Occupancy Estimation dataset, available from the UC Irvine Machine Learning Repository. At least two research papers have been published using this dataset. The introductory paper (Singh, et al., 2018) is the basis for

this project. A more recent paper (Mao et al., 2023) adapted a predictive framework originally used in water quality forecasting. In this latter study, the authors claimed to have achieved better results than the original paper.

In the Abstract, I listed three research questions that I want to investigate:

Which of the implemented supervised learning techniques perform the best in predicting occupancy?

In most of the recent studies I've reviewed, Random Forest performed the best in predicting occupancy. Research by Wang et al. (2021), Kim et al. (2023), and Mao et al. (2023) all concluded that Random Forest performed the best based on accuracy or F1-score. Research by Banihashemi et al. (2024) found that Random Forest and DFNN models had the best results for office A, whereas XGBoost had the best results for office B. Notably, the introductory paper by Singh, et al. (2018) found that SVM (RBF) outperformed other learning techniques including Random Forest.

For my project, I used Random Forest and SVM as two of the classification methods and compared the results.

Which types of sensor data (temperature, light, sound, motion, CO_2) show the most promising results?

In (Singh, et al., 2018), the authors found that the CO_2 slope feature showed the most promising results. This feature was derived by the authors from the actual CO_2 values, which are subject to time delay, using linear regression. When both CO_2 features were combined, the performance of the algorithms improved significantly. The temperature, sound, and motion features performed well in terms of accuracy, but not in terms of F1-score. Light features performed the best overall but were rejected by the authors since the results relied on occupants turning on desk lights when they arrived and turning them off again when they left. Indeed, lights may be controlled by the environmental system itself based on whether the room is occupied.

Using the same dataset, Mao et al. (2023) found that the light values from sensors 1 and 2 had the largest impact in predicting room occupancy in the Random Forest model. Unlike the earlier paper, they found that the sound feature (sensors 1 and 3) and CO_2 (sensor 5) had a larger impact than the CO_2 slope feature. They did find a linear relationship between the CO_2 slope and the room occupancy count. As the value of the CO_2 slope increased, the room occupancy count increased a corresponding amount.

As part of my analysis, I compared my results with the Singh et al. (2018) and Mao et al. (2023) studies.

In other studies, motion (Hailemariam et al., 2011), CO_2 (Dong et al., 2010), and CO_2 in combination with other features (Kim et al., 2023; Banihashemi et al., 2024) achieved the best results.

Based on the research, what alternative types of sensor data could be used for ML-based occupancy estimation?

In the studies I cited, CO_2 was the most common sensor data that was used (8), followed by temperature (7), light (7), sound (6), and motion (6). The next most common sensor type was relative humidity (4). Power use, differential pressure (air flow), ventilation system state (activated or not), and air quality were used in a single study.

With the advent of Internet of Things (IoT) technologies, alternative types of sensor data have been proposed. These include Bluetooth signals, Wi-Fi, camera images, GPS data, UWB radar, and electric meters. In their research, Khan et al. (2024), reviewed the advantages and limitations of data collection methods for occupancy detection and estimation.

This table summarizes some of the advantages and limitations of alternative sensor types:

Sensor Data	Advantages	Limitations
UWB radar	High precision Can detect movements	High cost Limited range Potential privacy issues
Bluetooth Low Energy	Non-intrusive Low cost Easy to implement	Limited range Signal interruption False readings if occupants carry multiple devices
Wi-Fi	Non-intrusive Low cost Easy to implement	Limited range Signal interruption False results
Cameras	High accuracy Can count occupants	Expensive Privacy issues Difficult to install
Electric Meter	Non-intrusive Cost-effective	False results due to weather Requires knowledge for data analysis

Source: Khan et al, 2024

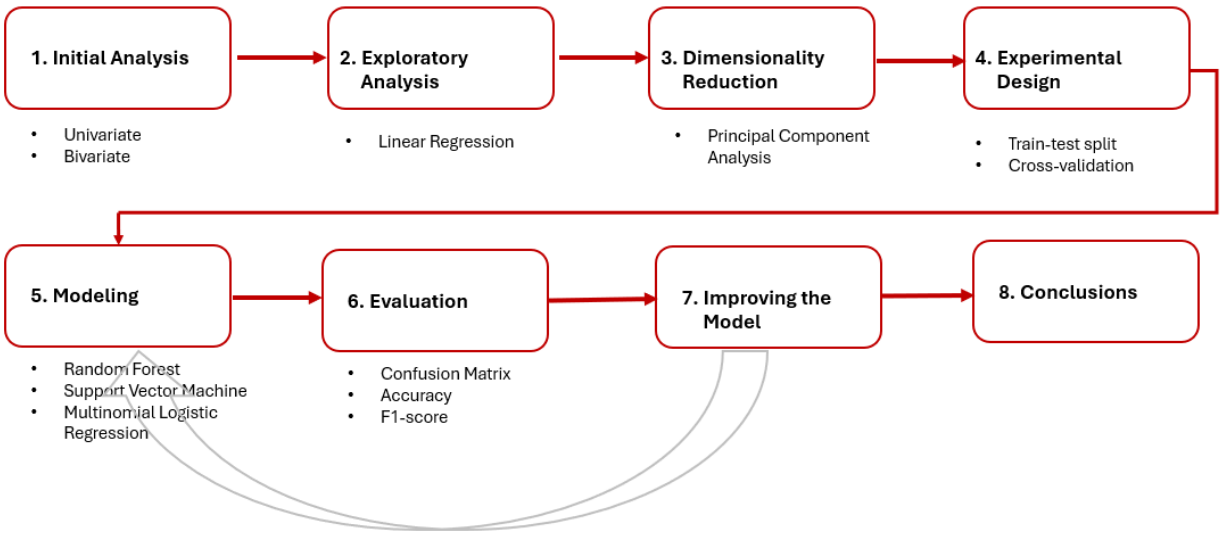
GitHub Repository

For this project, I'm using the following repository in GitHub:

https://github.com/jeffreyfitzpatrick/Big-Data-Analytics-Capstone-Project

Applied Methodology and Study Design

This flowchart shows the steps of the data analysis process I followed for this project:



1. Initial Analysis
   In the univariate analysis, I included a detailed data dictionary, summary statistics for the numeric attributes, and bar plots showing the frequency of the categorical variables.

   In the bivariate analysis, I included a scatter plot matrix of the continuous variables, time series analysis, and correlation analysis.

2. Exploratory Analysis

   As part of exploratory analysis, I used linear regression as a baseline model.

3. Dimensionality Reduction

   I investigated how an unsupervised learning algorithm named Principal Component Analysis (PCA) can be used for dimensionality reduction.

4. Experimental Design

   I split the data into training and test sets and investigated k-fold cross validation to evaluate the stability of the models.

5. Modeling

   I implemented three classification algorithms: Multinomial Logistic Regression, Random Forest, and Support Vector Machine (SVM).

6. Evaluation

   I evaluated the models using accuracy, confusion matrices, and F1-score.

7. Improving the Model

   Improving the model is an iterative process.

8. Conclusions

   In this final report, I compared my results with the two studies that used the same dataset.

Initial Analysis

The data set contains 10129 data points and 16 features. Each feature represents data (temperature, light, sound, motion, or CO_2) from a particular sensor. Measurements were recorded over several days in 30 second intervals.

There are no missing values in the dataset.

Univariate Analysis

Data Dictionary

This table describes the variables in the Room Occupancy Estimation data set:

Field Name	Data Type	Description
Date	Date	Date of observation in YYYY/MM/DD
Time	Date	Time of observation in HH:MM:SS
S1_Temp	Continuous	Temperature reading from sensor 1 in degrees Celsius
S2_Temp	Continuous	Temperature reading from sensor 2 in degrees Celsius
S3_Temp	Continuous	Temperature reading from sensor 3 in degrees Celsius
S4_Temp	Continuous	Temperature reading from sensor 4 in degrees Celsius
S1_Light	Integer	Light reading from sensor 1 in lux
S2_Light	Integer	Light reading from sensor 2 in lux
S3_Light	Integer	Light reading from sensor 3 in lux
S4_Light	Integer	Light reading from sensor 4 in lux
S1_Sound	Continuous	Sound reading from sensor 1 in volts (amplifier output read by ADC)

Field Name	Data Type	Description
S2_Sound	Continuous	Sound reading from sensor 2 in volts (amplifier output read by ADC)
S3_Sound	Continuous	Sound reading from sensor 3 in volts (amplifier output read by ADC)
S4_Sound	Continuous	Sound reading from sensor 4 in volts (amplifier output read by ADC)
S5_CO2	Integer	CO_2 reading from sensor 5 in ppm
S5_CO2_Slope	Continuous	Derived slope of $C0_2$ values taken in a sliding window **Note:** The slope was estimated using linear regression.
S6_PIR	Binary	Binary value conveying motion detection from passive infrared (PIR) sensor 6: **0:** No motion events detected in 30 second frame **1:** At least one motion event detected in 30 second frame
S7_PIR	Binary	Binary value conveying motion detection from passive infrared (PIR) sensor 7: **0:** No motion events detected in 30 second frame **1:** At least one motion event detected in 30 second frame
Room_Occupancy_Count	Integer	Number of occupants in the room at one time (ground truth)

Dependent (target) Variable

The dependent (target) variable is Room_Occupancy_Count.

Summary Statistics

This table shows the summary statistics of the numeric attributes in the data set:

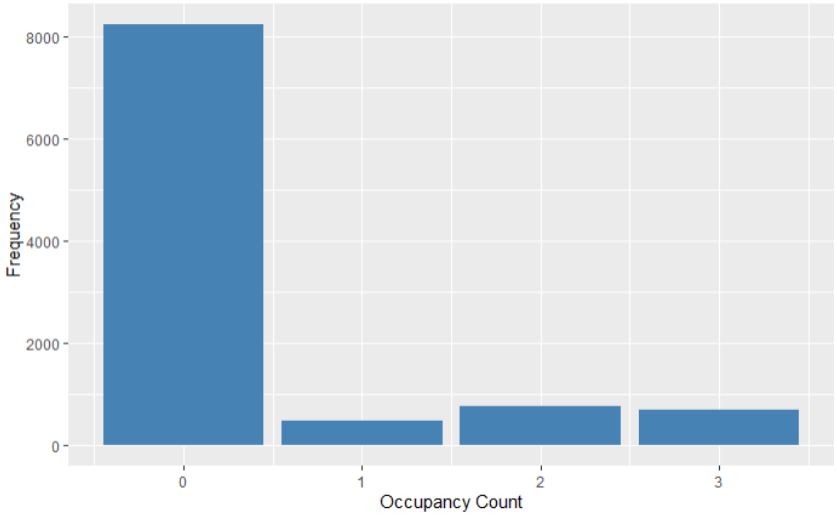
Variable	Minimum	Q1	Median	Mean	Q3	Maximum
S1_Temp	24.94	25.19	25.38	25.45	25.63	26.38
S2_Temp	24.75	25.19	25.38	25.55	25.63	29.00

Variable	Minimum	Q1	Median	Mean	Q3	Maximum
S3_Temp	24.44	24.69	24.94	25.06	25.38	26.19
S4_Temp	24.94	25.44	25.75	25.75	26.00	26.56
S1_Light	0.00	0.00	0.00	25.45	12.00	165.00
S2_Light	0.00	0.00	0.00	26.02	14.00	258.00
S3_Light	0.00	0.00	0.00	34.25	50.00	280.00
S4_Light	0.00	0.00	0.00	13.22	22.00	74.00
S1_Sound	0.06	0.07	0.08	0.1682	0.08	3.88
S2_Sound	0.04	0.05	0.05	0.1201	0.06	3.44
S3_Sound	0.04	0.06	0.06	0.1581	0.07	3.67
S4_Sound	0.05	0.06	0.08	0.1038	0.1	3.4
S5_CO2	345	355	360	460.9	465	1270
S5_CO2_Slope	-6.29615	-0.04615	0	-0.00483	0	8.98077

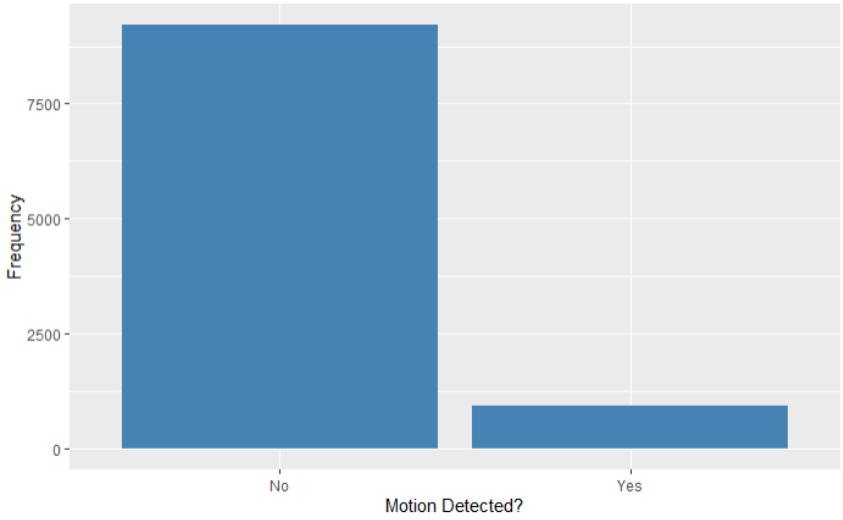
Frequency of Categorical Variables

In about 80% of the data points, the room was unoccupied.

This chart shows the frequency of the Room_Occupancy_Count (target) variable:



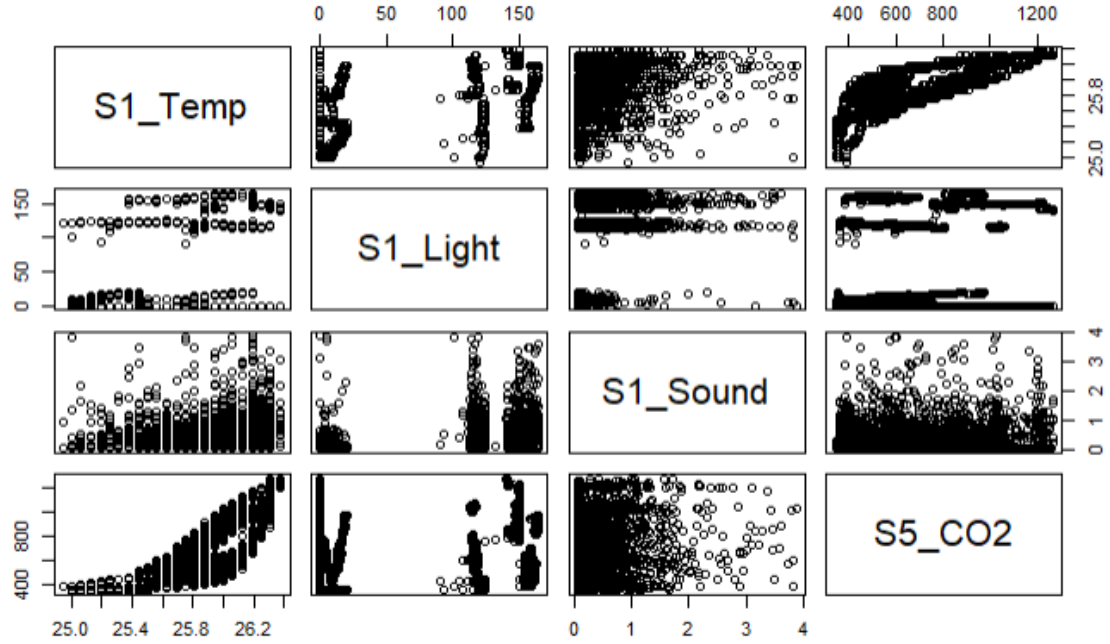
This bar chart shows the frequency of the S6_PIR variable used to detect motion:



Bivariate Analysis

Pairwise Visualizations

This chart shows the scatter plot matrix of the temperature (S1), light (S1), sound (S1), and CO_2 (S5) values:

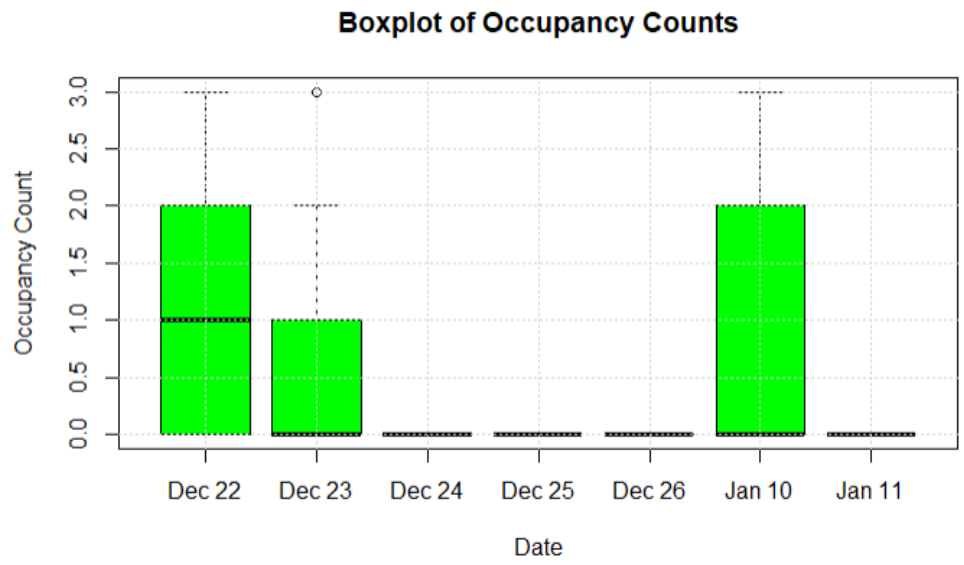


Time Series Analysis

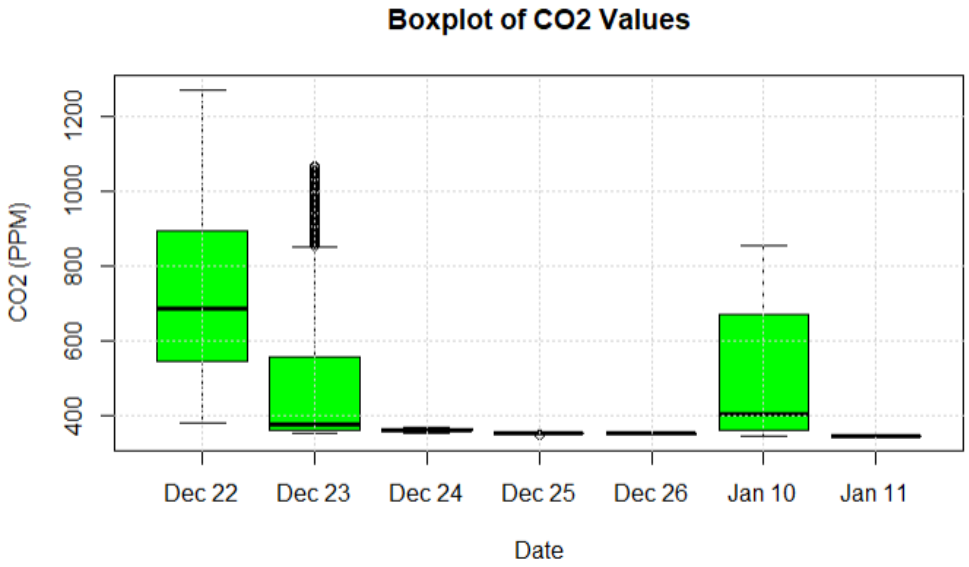
Based on the time series analysis, I observed the following:

- There were occupants only on December 22, December 23, and January 10.
- Significant CO_2 values were only detected on days with occupants.
- The highest temperatures occurred on the days with occupants.
- The highest light values occurred on days with occupants. Notably, light was detected on Christmas, indicating that lights were turned on during part of that day.
- The highest sound values occurred on days with occupants.

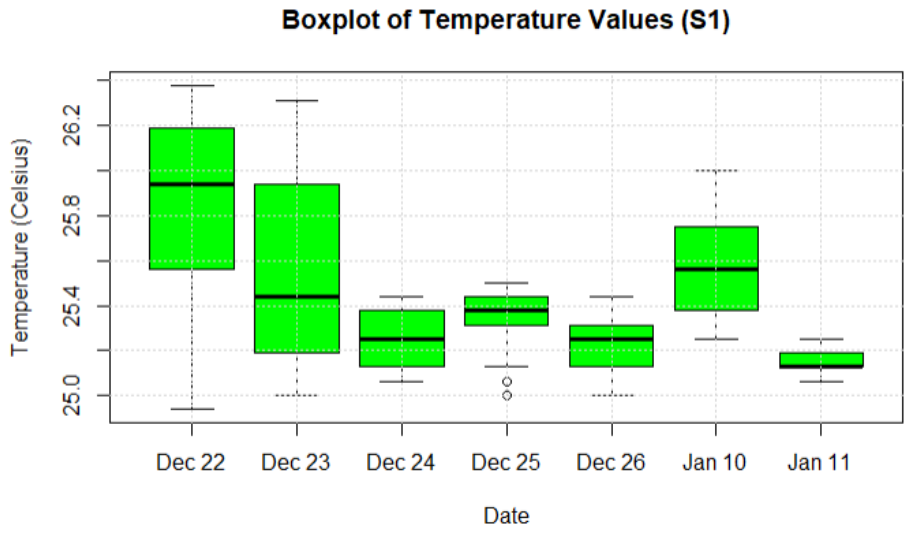
This boxplot shows the distribution of room occupancy for each day:



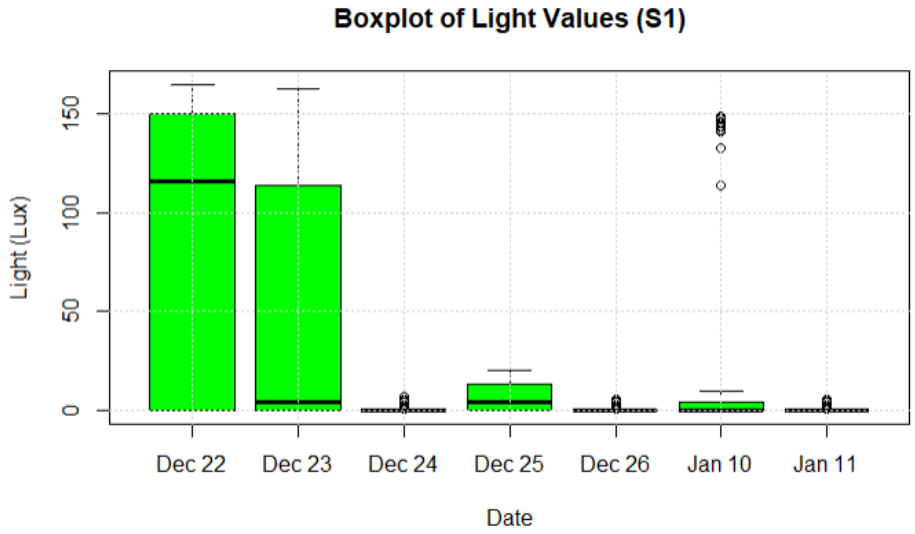
This boxplot shows the distribution of CO_2 values for each day:



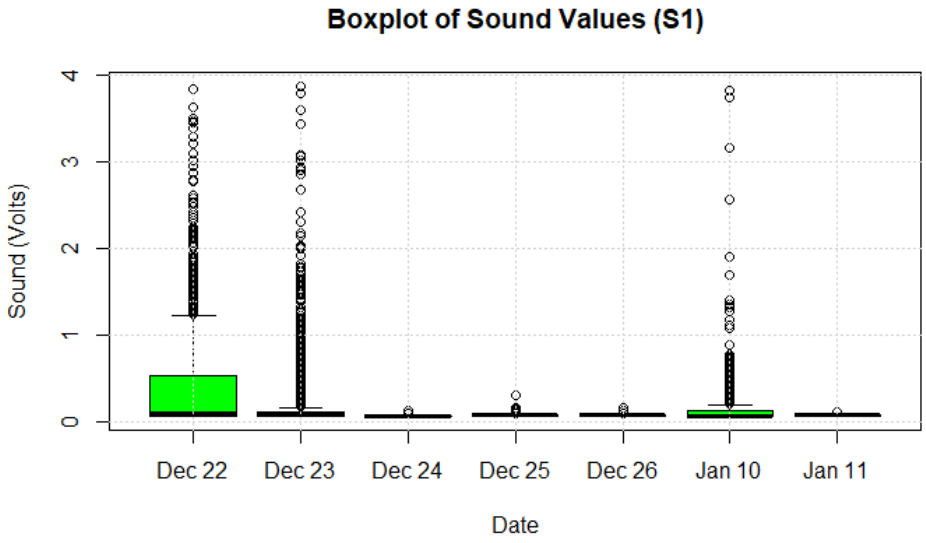
This boxplot shows the distribution of temperature values (sensor 1) for each day:



This boxplot shows the distribution of light values (sensor 1) for each day:



This boxplot shows the distribution of sound values (sensor 1) for each day:



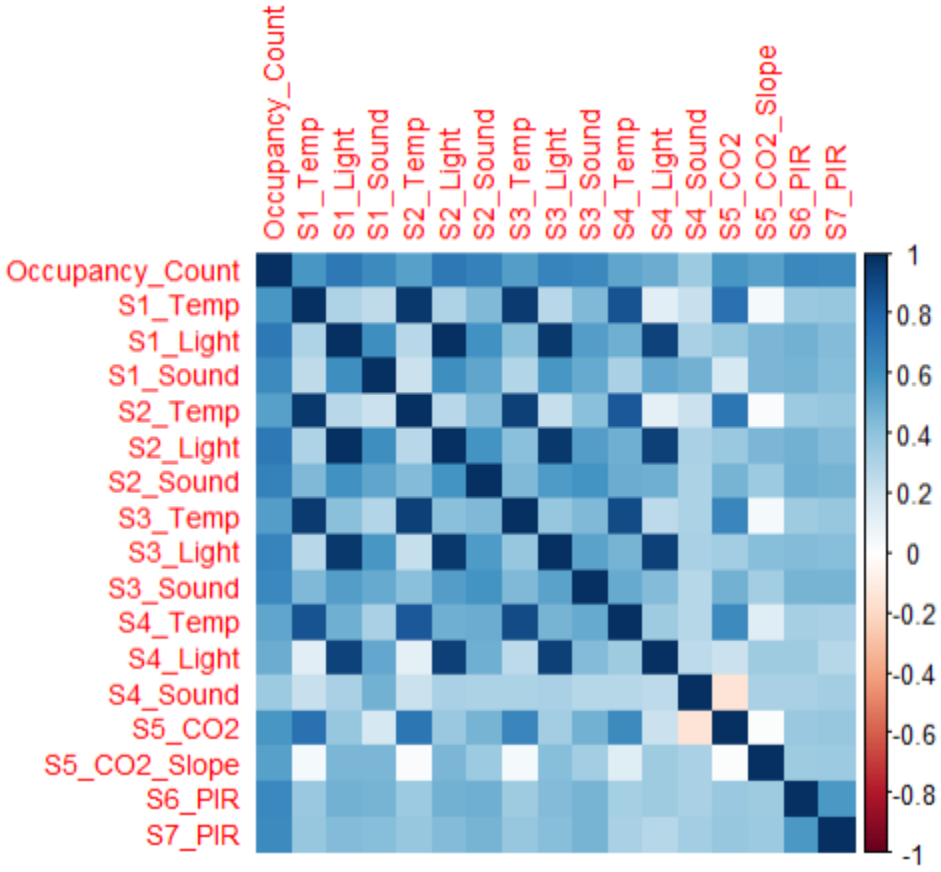
Correlation Analysis

In this analysis, Spearman correlation was used as the attributes do not have a normal distribution.

Based on the correlation analysis, I observed the following:

- There is a moderate correlation between room occupancy and temperature variables.
- There is a strong correlation of temperature values between sensors.
- There is a moderate correlation between room occupancy and light variables.
- There is a strong correlation of light values between sensors, indicating that the lights were generally turned on at the same time or turned off.
- There is a moderate correlation between room occupancy and sound variables, except for sensor 4.
- There is a relatively weak correlation of sound values between sensors, indicating that the sensors could distinguish sounds of occupants in different parts of the room.
- There is a moderate correlation between room occupancy and CO_2 variables.
- There is a moderate correlation between room occupancy and motion variables.
- There is no correlation between CO_2 and the derived CO_2 slope values.

This chart shows a plot of the correlation matrix:



Exploratory Analysis

For exploratory analysis, I fitted various linear regression models to predict room occupancy based on the sensor data.

In the following table, the R-squared value represents the proportion of variance in the dependent variable that is explained by the independent variables. The higher the R-squared value, the better the fit of the model. The intercept term represents the predicted value of the dependent variable when all independent variables are zero. P-values associated with each coefficient indicate the significance of the corresponding variable. P-values below 0.05 suggest that the corresponding variable is statistically significant in predicting the target variable (Room_Occupancy_Count).

Model to predict room occupancy	R-squared	Intercept	Coefficients (P-value)
Temperature data (S1_Temp, S2_Temp, S3_Temp, S4_Temp)	0.548	-37.4029	S1_Temp: 1.6245 (0.000) S2_Temp: 0.4826 (0.000) S3_Temp: 0.1581 (0.001) S4_Temp: -0.7703 (0.000)
Light data (S1_Light, S2_Light, S3_Light, S4_Light)	0.792	0.0410	S1_Light: 0.0076 (0.000) S2_Light: 0.0032 (0.000) S3_Light: 0.0058 (0.000) S4_Light: -0.0091 (0.000)
Sound data (S1_Sound, S2_Sound, S3_Sound, S4_Sound)	0.444	0.0900	S1_Sound: 0.8700 (0.000) S2_Sound: 0.9411 (0.000) S3_Sound: 0.5396 (0.000) S4_Sound: -0.3559 (0.000)
CO_2 data (S5_CO2, S5_CO2_Slope)	0.746	-0.8794	S5_CO2: 0.0028 (0.000) S5_CO2_Slope: 0.4281 (0.000)
CO_2 slope data (S5_CO2_Slope)	0.361	0.4008	S5_CO2_Slope: 0.4611 (0.000)
Motion data (S6_PIR, S7_PIR)	0.401	0.2181	SS_PIR: 6.559e+11 **(0.265)** S6_PIR: -6.559e+11 **(0.265)**
All features	0.894	-8.0051	S1_Temp: 0.1335 (0.003) S2_Temp: 0.1166 (0.000) S3_Temp: 0.7861 (0.000) S4_Temp: -0.6959 (0.000) S1_Light: 0.0054 (0.000) S2_Light: 0.0009 (0.000) S3_Light: 0.0026 (0.000) S4_Light: -0.0040 (0.000) S1_Sound: 0.1009 (0.000) S2_Sound: 0.1957 (0.000) S3_Sound: -0.0763 (0.000) S4_Sound: -0.3485 (0.000) S5_CO2: 3.42e-05 **(0.364)** S5_CO2_Slope: 0.1896 (0.000) S6_PIR: 0.1800 (0.000) S7_PIR: 0.4139 (0.000)

Regarding p-values and coefficient values, see Limitations and Future Work.

Using all sensor data provided the best model fit, with an R-squared value of 89.4%. The results suggest that the S5_CO2 value can be dropped since it's p-value is greater than 0.05.

Based on the remaining models, light data had the highest R-squared value at 79.2% followed by C02 data at 74.6%. This is consistent with the results of the introductory paper (Singh et al., 2018), which found that light features performed the best overall and that the CO_2 features had promising results.

For the linear regression model containing only S5_CO2_Slope, I was able to duplicate the results of the Mao et al. (2023) study, which found:

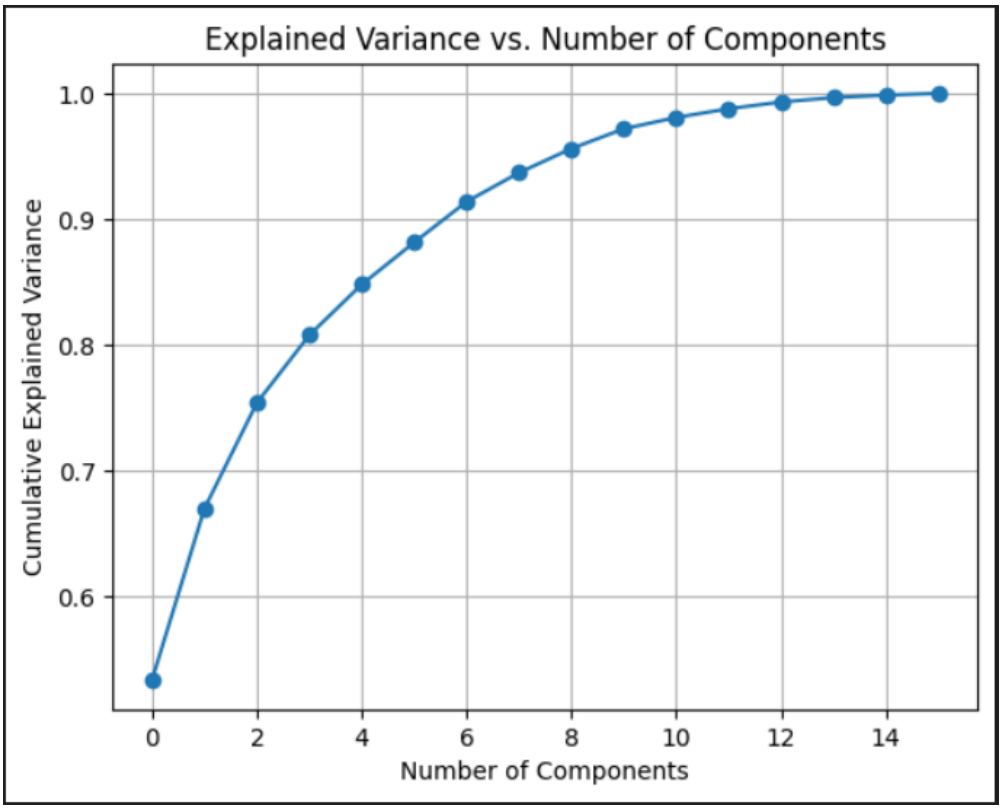
- An intercept value of approximately 0.4008, indicating the expected value of Room_Occupancy_Count when S5_CO2_Slope is zero.
- Positive slope of approximately 0.4611, indicating that for each unit increase in S5_CO2_Slope we can anticipate an increase of approximately 0.4611 in Room_Occupancy_Count.

For the linear regression model containing only motion data, p-values of 0.265 indicate that the motion data by itself is not statistically significant in predicting the target variable (Room_Occupancy_Count).

Dimensionality Reduction

For dimensionality reduction, I used Principal Component Analysis (PCA). Since PCA is sensitive to the scale of the features, I first standardized the dataset so that all the numerical features have a mean of 0 and a variance of 1. This step ensures that features that have a larger variance because of scale do not dominate the principal components.

To determine the optimal number of components, I created a cumulative explained variance graph:

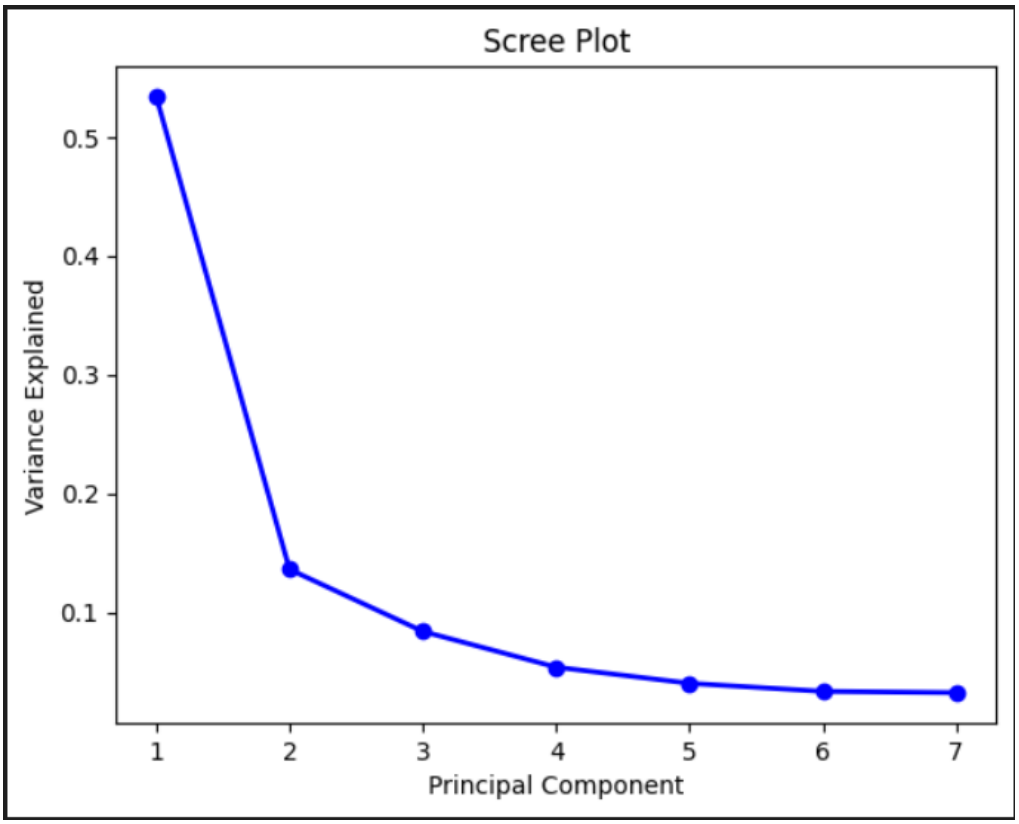


To capture at least 90% of the variance, I chose to select 7 components. This reduced the number of dimensions from 16 in the original dataset to 7 in the PCA dataset.

After applying PCA with 7 components, I calculated the percentage of total variance explained by each principal component:

- The first principal component explained **53.42%** of the total variation in the dataset.
- The second principal component explained **13.61%** of the total variation in the dataset.
- The third principal component explained **8.39%** of the total variation in the dataset.
- The fourth principal component explained **5.38%** of the total variation in the dataset.
- The fifth principal component explained **4.02%** of the total variation in the dataset.
- The sixth principal component explained **3.34%** of the total variation in the dataset.
- The seventh principal component explained **3.23%** of the total variation in the dataset.

This can be visualized using a scree plot:



Finally, I analyzed the feature contributions (loadings) for the first two principal components.

For the first component, S1_Light, S3_Light, S1_Temp, S2_Temp, and S3_Temp had the largest contributions.

For the second component, S5_CO2_Slope, S4_Sound, S2_Sound, S1_Sound, and S3_Sound had the largest contributions.

Experimental Design

To prepare for logistic regression and Support Vector Machine, I scaled the numeric features so that the mean would be approximately 0 and the standard deviation approximately 1. All features except for motion data were scaled. Random Forest does not require the scaling of numeric features.

For all models, I split the data into training and testing sets, with 80% of the data points devoted to training and 20% of the datapoints devoted to testing. To deal with the unbalanced dataset, I undersampled from the majority class in the training set. After rebalancing, the distribution of the target variable (Room_Occupancy_Count) was as follows:

- Class 0: 31.6%
- Class 1: 29.9%
- Class 2: 19.2%
- Class 3: 19.2%

To test the stability of the model, I applied 10-fold cross-validation on the training set. I then trained the model on the full training set and evaluated the model on the test set.

Multinomial Logistic Regression

All Features

For Multinomial Logistic Regression, the average cross-validation accuracy on the training set was 97.9%.

To implement Multinomial Logistic Regression, I chose a logistic regression classifier (newton-cg) that supports multiclass classification.

This table summarizes the evaluation metrics on the test set using all features:

Class	Accuracy	Precision	Recall	F1-score	Support
0		100.0%	99.2%	99.6%	1619
1		100.0%	100.0%	100.0%	103
2		97.0%	97.0%	97.0%	164
3		88.2%	96.4%	92.2%	140
Average	98.9%	96.3%	98.1%	97.2%	

Accuracy: The average accuracy was 98.9%.

Precision:

- Out of the total predictions for class 0, 100.0% belong to class 0 in the actual dataset.
- Out of the total predictions for class 1, 100.0% belong to class 1 in the actual dataset.
- Out of the total predictions for class 2, 97.0% belong to class 2 in the actual dataset.
- Out of the total predictions for class 3, 88.2% belong to class 3 in the actual dataset.

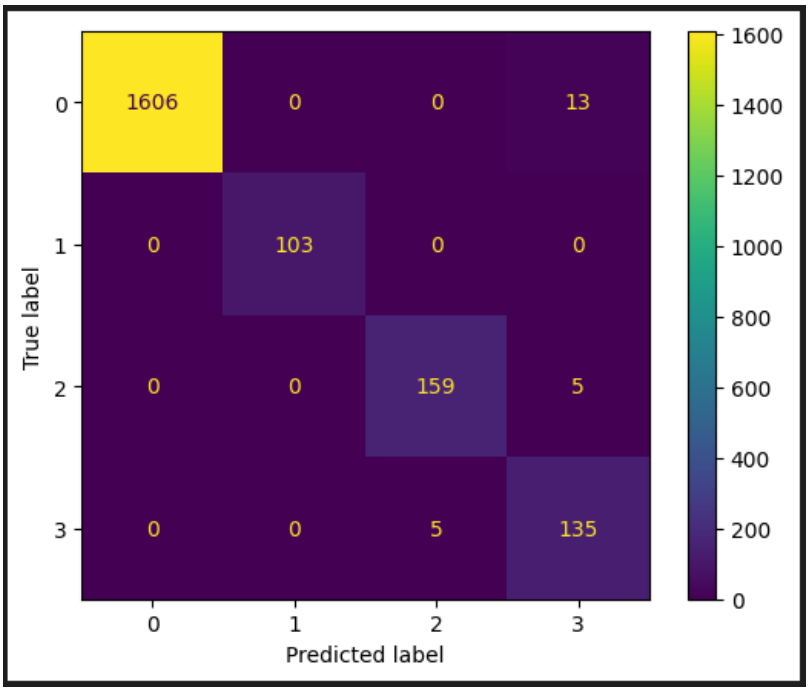
Recall:

- Out of the total entries in the dataset for class 0, 99.2% were classified in class 0 by the model.
- Out of the total entries in the dataset for class 1, 100.0% were classified in class 1 by the model.
- Out of the total entries in the dataset for class 2, 97.0% were classified in class 2 by the model.
- Out of the total entries in the dataset for class 3, 96.4% were classified in class 3 by the model.

F1-score: The average F1-score was 97.2%. Since this value is close to 1, the model did an excellent job of predicting the number of occupants in a room. The model did a better job of predicting 0-2 occupants then it did at predicting 3 occupants.

Support: In the actual test dataset, there are 1619 entries in class 0, 103 entries in class 1, 164 entries in class 2, and 140 entries in class 3.

From the confusion matrix, I observed that most of the wrong predictions were in predicting class 3:



PCA Features

I reran the model with PCA using seven components.

This table summarizes the evaluation metrics on the test set using PCA:

Class	Accuracy	Precision	Recall	F1-score	Support
0		99.2%	99.4%	99.3%	1619
1		89.7%	93.2%	91.4%	103
2		71.5%	68.9%	70.2%	164
3		64.0%	63.6%	63.8%	140
Average	94.1%	81.1%	81.3%	81.2%	

Accuracy: The average accuracy was 94.1% with PCA compared to 98.9% without PCA.

F1-score: The average F1-score was 81.2.% with PCA compared to 97.2% without PCA. A significant drop in performance can be observed in both precision and recall for class 2 and class 3.

Random Forest

All Features

For Random Forest, the average cross-validation accuracy on the training set was 99.0%.

This table summarizes the evaluation metrics on the test set using all features:

Class	Accuracy	Precision	Recall	F1-score	Support
0		100.0%	99.8%	99.9%	1619
1		99.0%	100.0%	99.5%	103
2		99.4%	99.4%	99.4%	164
3		97.9%	99.3%	98.6%	140
Average	99.8%	99.1%	99.6%	99.3%	

Accuracy: The average accuracy was 99.8%.

Precision:

- Out of the total predictions for class 0, 100.0% belong to class 0 in the actual dataset.

- Out of the total predictions for class 1, 99.0% belong to class 1 in the actual dataset.
- Out of the total predictions for class 2, 99.4% belong to class 2 in the actual dataset.
- Out of the total predictions for class 3, 97.9% belong to class 3 in the actual dataset.

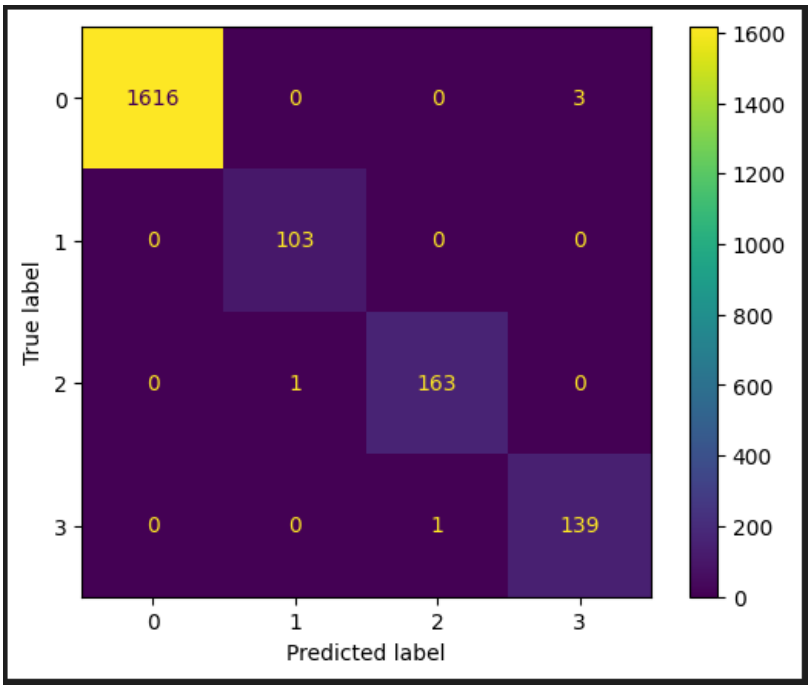
Recall:

- Out of the total entries in the dataset for class 0, 99.8% were classified in class 0 by the model.
- Out of the total entries in the dataset for class 1, 100.0% were classified in class 1 by the model.
- Out of the total entries in the dataset for class 2, 99.4% were classified in class 2 by the model.
- Out of the total entries in the dataset for class 3, 99.3% were classified in class 3 by the model.

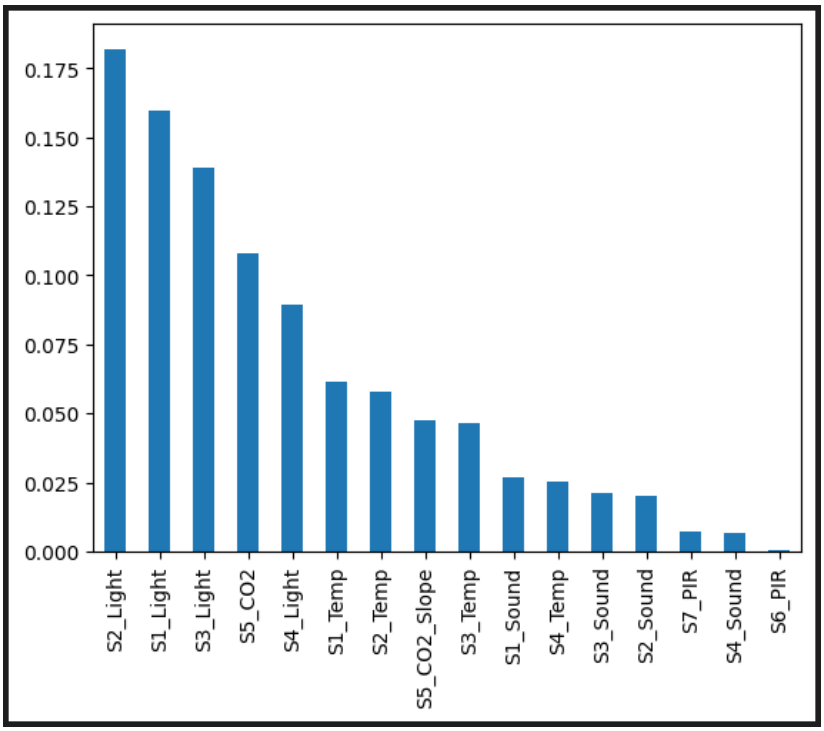
F1-score: The average F1-score was 99.3%. Since this value is close to 1, the model did an excellent job of predicting the number of occupants in a room.

Support: In the actual test dataset, there are 1619 entries in class 0, 103 entries in class 1, 164 entries in class 2, and 140 entries in class 3.

From the confusion matrix, I observed that there were only a few wrong predictions:



For Random Forest, I plotted the importance of each feature. As shown in the following graph, the most important features were light. The least important features were sound and motion.



PCA Features

I reran the model with PCA using seven components.

This table summarizes the evaluation metrics on the test set using PCA:

Class	Accuracy	Precision	Recall	F1-score	Support
0		99.8%	99.8%	99.8%	1619
1		97.1%	97.1%	97.1%	103
2		97.5%	94.5%	96.0%	164
3		92.5%	96.4.%	94.4%	140
Average	99.0%	96.7%	96.9%	96.8%	

Accuracy: The average accuracy was 99.0% with PCA compared to 99.8% without PCA.

F1-score: The average F1-score was 96.8.% with PCA compared to 99.3% without PCA. The results suggest that good performance could be achieved with fewer than 7 components.

Support Vector Machine

All Features

For SVM, the average cross-validation accuracy on the training set was 97.8%.

To implement SVM, I chose a non-linear kernel (rbf).

This table summarizes the evaluation metrics on the test set using all features:

Class	Accuracy	Precision	Recall	F1-score	Support
0		100.0%	99.4%	99.7%	1619
1		100.0%	100.0%	100.0%	103
2		97.0%	97.0%	97.0%	164
3		90.6%	96.4%	93.4%	140
Average	99.1%	96.9%	98.2%	97.5%	

Accuracy: The average accuracy was 99.1%.

Precision:

- Out of the total predictions for class 0, 100.0% belong to class 0 in the actual dataset.
- Out of the total predictions for class 1, 100.0% belong to class 1 in the actual dataset.
- Out of the total predictions for class 2, 97.0% belong to class 2 in the actual dataset.
- Out of the total predictions for class 3, 90.6% belong to class 3 in the actual dataset.

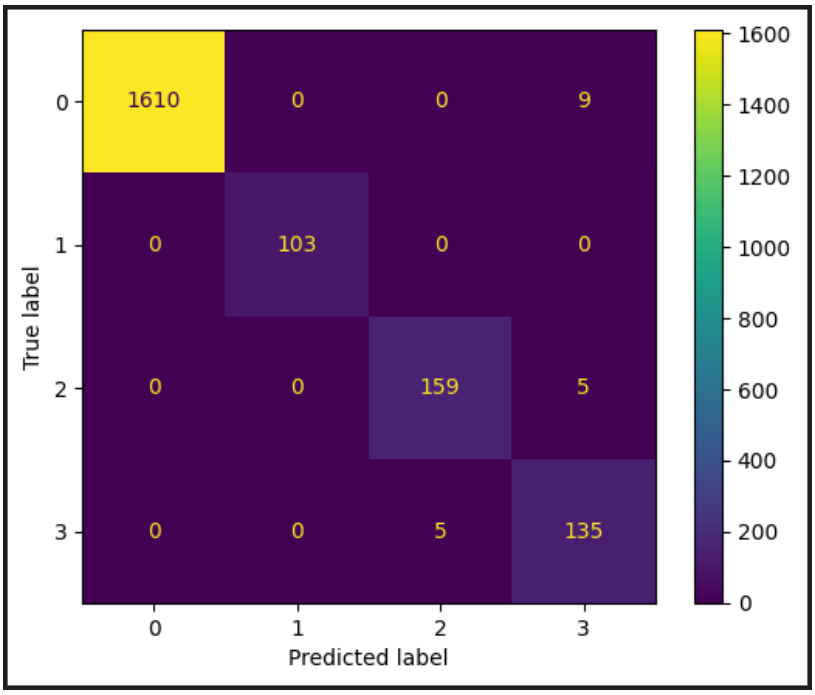
Recall:

- Out of the total entries in the dataset for class 0, 99.4% were classified in class 0 by the model.
- Out of the total entries in the dataset for class 1, 100.0% were classified in class 1 by the model.
- Out of the total entries in the dataset for class 2, 97.0% were classified in class 2 by the model.
- Out of the total entries in the dataset for class 3, 96.4% were classified in class 3 by the model.

F1-score: The average F1-score was 97.5%. Since this value is close to 1, the model did an excellent job of predicting the number of occupants in a room. The model did a better job of predicting 0-2 occupants then it did at predicting 3 occupants.

Support: In the actual test dataset, there are 1619 entries in class 0, 103 entries in class 1, 164 entries in class 2, and 140 entries in class 3.

From the confusion matrix, I observed that most of the wrong predictions were in predicting class 3:



PCA Features

I reran the model with PCA using seven components.

This table summarizes the evaluation metrics on the test set using PCA:

Class	Accuracy	Precision	Recall	F1-score	Support
0		99.9%	99.8%	99.9%	1619
1		100%	100%	100%	103
2		95.8%	96.3%	96.0%	164
3		95.0%	95.7%	95.4%	140
Average	99.3%	97.7%	98.0%	97.8%	

Accuracy: The average accuracy was 99.3% with PCA, which exceeds the model without PCA (99.1%).

F1-score: The average F1-score was 97.8.% with PCA, which exceeds the model without PCA (97.5%). The results suggest that good performance could be achieved with fewer than 7 components.

Findings

Summary of Model Results

This table summarizes the performance metrics of the models:

	All Features			PCA	
Model	Accuracy Training Set	Accuracy Test Set	F1-Score Test Set	Accuracy Test Set	F1-Score Test Set
Multinomial Logistic Regression	97.9%	98.9%	97.2%	94.1%	81.2%
Random Forest	99.0%	99.8%	99.3%	99.0%	96.8%
Support Vector Machine	97.8%	99.1%	97.5%	99.3%	97.8%

When using all features, Random Forest performed the best in terms of both accuracy and F1-score. These results are consistent with Mao et al. (2023), which also found that Random Forest performed the best in all three of their metrics.

When using PCA, Support Vector Machine performed the best in terms of both accuracy and F1-score. The results suggest that good results could be achieved with fewer than 7 components. Indeed, Singh et al. (2018) concluded that an accuracy of 92% and F1-score of 72% was achievable with only four components.

Good results were also achieved when using PCA with Random Forest. However, using PCA with Multinomial Logistic Regression is not recommended based on the reduction in F1-score.

Summary of Sensor Data

Using linear regression, light data had the highest R-squared value at 79.2% followed by $C0_2$ data at 74.6%. Less promising features were temperature data (54.8%), sound data (44.4%), and motion data (40.1%). Using all sensor data provided the best model fit, with an R-squared value of 89.4%.

For the first component of PCA, light data from sensor 1 had the largest contribution, followed by light data from sensor 3, and temperature data from the first three sensors.

In the Random Forest model, the most important features were light data from sensor 2, sensor 1, and sensor 3. Similarly, Mao et. al (2023) found that light values from sensor 1 and sensor 2 had the largest impact in predicting room occupancy. The fourth most important feature was CO_2 data. The least important features were motion and sound data.

Based on these results, it could be concluded that light data performed the best overall in predicting room occupancy. However, in Singh, et al. (2018), the authors rejected using light data since the results relied on occupants turning on desk lights when they arrived and turning them off again when they left. Indeed, lights may be best controlled by the environmental system itself based on whether the room is occupied.

Although CO_2 data showed promising results, the best approach is to use a combination of different types of sensor data, including $C0_2$.

With the advent of Internet of Things (IoT) technologies, alternative types of sensor data such as Bluetooth signals, Wi-Fi, camera images, GPS data, UWB radar, and electric meters could also be considered. While these technologies have their advantages, limitations such as high cost, limited range, false results, and privacy issues may prevent them from becoming widely adopted (Khan et al., 2024).

Limitations and Future Work

Linear regression and logistic regression models assume a linear relationship between the independent variables and the dependent variable. In addition, linear regression and logistic regression models assume that the independent variables are not correlated with each other. However, in the Room Occupancy Estimation dataset, the sensor data are moderately to highly correlated with each other. This is known as multicollinearity. As a result, it can be difficult to isolate the impact of an individual feature on the dependent variable. For example, if the temperature value of a sensor increases, the temperature value of nearby sensor will also increase.

Because of multicollinearity, the reported p-values and coefficients in some of the linear regression models may not be reliable. To fix multicollinearity, PCA can be used to create uncorrelated features.

The Logistic Regression Model was rerun using uncorrelated features created by PCA. However, a significant drop in performance was noted in both precision and recall for class 2 and class 3.

Linear regression models are also sensitive to outliers. Data points that are far away from the cluster of points can have a major impact on the calculated error. In the Room Occupancy Estimation dataset, outliers were detected in all the numerical sensor data (temperature, light, sound, and CO_2).

This paper concludes that Random Forest performed the best when using all features and SVM performed the best when using PCA. However, the difference in model results may not be statistically significant. Hypothesis testing of the model results, for example using the Friedman test, could be considered for future work.

In addition, future work could involve feature selection (for example, removal of light features), using PCA with fewer components, and the evaluation of more classification algorithms.

References

1. Masoso, O.T., & Grobler, L.J. (2010). The dark side of occupants' behaviour on building energy use. *Energy and Buildings, 42*(2), 173-177. https://doi.org/10.1016/j.enbuild.2009.08.009

2. Singh, A.P., Jain, V., Chaudhari, S., Kraemer, F.A., Werner, S., & Garg, V. (2018). Machine learning-based occupancy estimation using multivariate sensor nodes. *IEEE Globecom Workshops (GC Wkshps)*, 1-6. https://doi.org/10.1109/GLOCOMW.2018.8644432

3. Hailemariam, E., Goldstein, R., Attar, R., & Khan, A. (2011). Real-time occupancy detection using decision trees with multiple sensor types. *Proc. Symp. Simulation Architecture and Urban Des. (SimAUD)*, 141-148. https://damassets.autodesk.net/content/dam/autodesk/www/autodesk-reasearch/Publications/pdf/realtime-occupancy-detection-using.pdf

4. Candanedo, L.M., & Feldheim, V. (2016). Accurate occupancy detection of an office room from light, temperature, humidity and CO_2 measurements using statistical learning models. *Energy and Buildings, 112*, 28-39. https://doi.org/10.1016/j.enbuild.2015.11.071

5. Dong, B., Andrews, B., Lam, K.P., Hoynck, M., Zhang, R., Chiou, Y., & Benitez, D. (2010). An information technology enabled sustainability test-bed (itest) for occupancy detection through an environmental sensing network. *Energy and Buildings, 42*(7), 1038-1046.

6. Yang, Z., Li, N., Becerik-Gerber, B., & Orosz, M. (2012). A multi-sensor based occupancy estimation model for supporting demand driven HVAC operations. *Proc. Symp Simulation Architecture and Urban Des. (SimAUD)*, 49-56.

7. Wang, C., Jiang, J., Roth, T., Nguyen, C., Liu, Y., & Lee, H. (2021). Integrated sensor data processing for occupancy detection in residential buildings. *Energy and Buildings, 237*. https://doi.org/10.1016/j.enbuild.2021.110810

8. Kim, J., Bang, J., Choi, A., Moon, H.J., & Sung, M. (2023). Estimation of occupancy using IoT sensors and a carbon dioxide-based machine learning model with ventilation system and differential pressure data. *Sensors, 23*(2). https://doi.org/10.3390/s23020585

9. Mao, S., Yuan, Y., Li, Y., Wang, Z., Yao, Y., & Kang, Y. (2023). Room occupancy prediction: Exploring the power of machine learning and temporal insights. *American Journal of Applied Mathematics and Statistics*. https://doi.org/10.48550/arXiv.2312.14426

10. Banihashemi, F., Weber, M., Deghim, F., Zong, C., & Lang, W. (2024). Occupancy modeling on non-intrusive indoor environmental data through machine learning. *Building and Environment, 254*. https://doi.org/10.1016/j.buildenv.2024.111382

11. Li, T., Liu, X., Li, G., Wang, X., Ma, J., Xu, C., & Mao, Q. (2024). A systematic review and comprehensive analysis of building occupancy prediction. *Renewable and Sustainable Energy Reviews, 193*. https://doi.org/10.1016/j.rser.2024.114284

12. Khan, I., Zedadra, O., Guerrieri, A., & Spezzano, G. (2024). Occupancy prediction in IoT-enabled smart buildings: technologies, methods, and future directions. *Sensors, 24*(11). doi.org/10.3390/s24113276