



**ALANYA
ALAADDIN KEYKUBAT
UNIVERSITY**

ROOM OCCUPANCY PREDICTION

Group Members

İbrahim Ardıç - 180254050

Yahya Murat Turgut - 220254101

Sepehr Latifi Azad - 190254082

Introduction

In the realm of building management, real-time occupancy data holds immense significance in optimizing intelligent HVAC (Heating, Ventilation, and Air Conditioning) systems and advanced lighting setups, thereby substantially enhancing energy conservation efforts and occupant comfort. This project delves into the potential of classical Machine Learning (ML) and Neural Networks by leveraging Internet of Things (IoT) sensors for environmental monitoring, with the aim of accurately estimating room occupancy. The challenge at hand is to effectively utilize IoT data and ML techniques to develop an economical and non-intrusive solution that aligns with objectives of energy efficiency and occupant well-being. The primary objective of this project is to utilize ML for precise room occupancy estimation, thereby reducing energy consumption and fostering sustainability. Specific aims include identifying the most effective model, discerning key sensor features, and extracting insights from the models to understand the role of sensor data in occupancy estimation. This research endeavors to advance sustainable building practices and contribute to energy conservation efforts.

Dataset Information

The dataset utilized in this study is sourced from the UCI Machine Learning repository and is attributed to Adarsh Pal Singh, Vivek Jain, Sachin Chaudhari, Frank Alexander Kraemer, Stefan Werner, and Vishal Garg. This dataset is prominently highlighted in the research paper titled "Machine Learning-Based Occupancy Estimation Using Multivariate Sensor Nodes," which was presented at the 2018 IEEE Globecom Workshops (GC Wkshps).

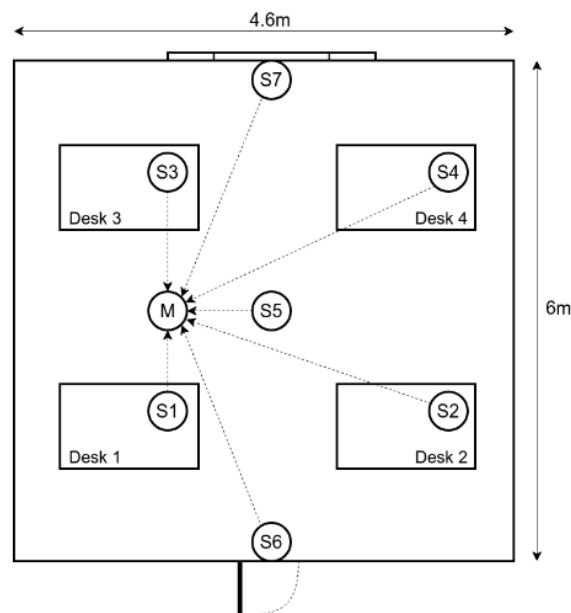


Figure 1: A star network based data acquisition system deployed in a test room.

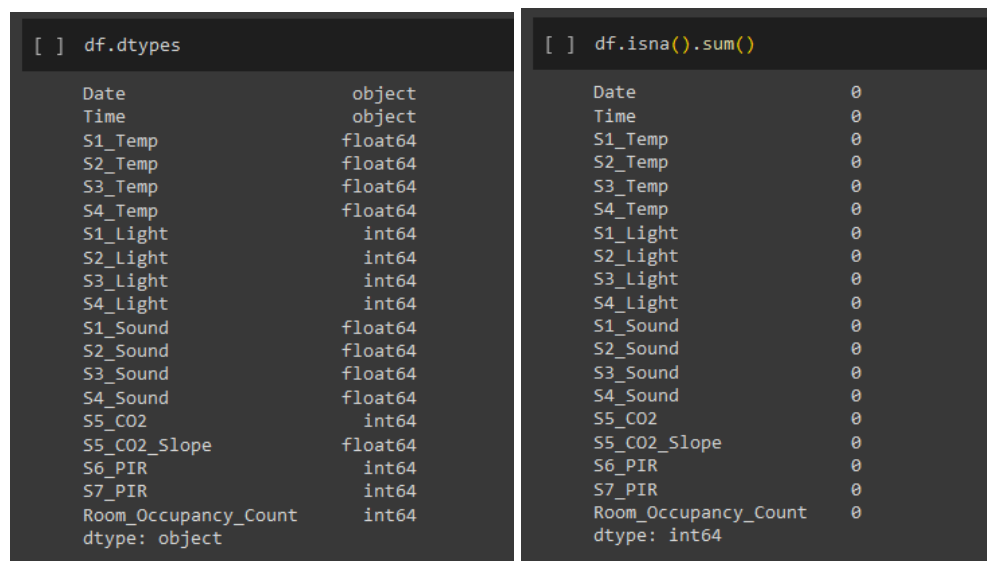
The data collection spanned a period of 7 days, during which the heating, ventilation, and air conditioning (HVAC) systems within the designated building space remained inactive. Occupancy levels varied between 0 and 3 individuals, providing valuable insights into the environmental dynamics under realistic conditions. Measurements were taken at 30-second intervals, facilitated by wireless transceivers. The dataset comprises 10,129 instances, each

containing 19 distinct features, offering a comprehensive view of the room's environment. These features include time series data on temperature, light intensity, CO2 levels, Passive Infrared (PIR) activity, and sound levels. These measurements were systematically collected from sensor nodes strategically positioned within a 6m x 4.6m room.

Data Understanding

Exploratory Data Analysis (EDA) plays a major and crucial role in comprehensively understanding and summarizing the primary characteristics of a dataset, employing both statistical metrics and visualization techniques. By conducting EDA, one can effectively pinpoint missing values, outliers, and discern the distribution patterns of variables, thereby facilitating more informed decision-making in subsequent modeling or analytical tasks.

Our analysis commenced by importing essential libraries and loading the dataset, followed by ascertaining the number of records and columns, along with their respective data types (.dtypes), and to identify whether any null values are present in the dataset. This preliminary exploration lays the groundwork for deeper insights into the dataset's structure and content, aiding in the formulation of robust analytical strategies.



[] df.dtypes	[] df.isna().sum()
Date object	Date 0
Time object	Time 0
S1_Temp float64	S1_Temp 0
S2_Temp float64	S2_Temp 0
S3_Temp float64	S3_Temp 0
S4_Temp float64	S4_Temp 0
S1_Light int64	S1_Light 0
S2_Light int64	S2_Light 0
S3_Light int64	S3_Light 0
S4_Light int64	S4_Light 0
S1_Sound float64	S1_Sound 0
S2_Sound float64	S2_Sound 0
S3_Sound float64	S3_Sound 0
S4_Sound float64	S4_Sound 0
S5_CO2 int64	S5_CO2 0
S5_CO2_Slope float64	S5_CO2_Slope 0
S6_PIR int64	S6_PIR 0
S7_PIR int64	S7_PIR 0
Room_Occupancy_Count int64	Room_Occupancy_Count 0
dtype: object	dtype: int64

Figure 2: Number of records and columns

We utilized the .describe() function to gain insights into the distributional characteristics and statistical information pertaining to each numerical column within our dataset. This method allowed us to obtain summary statistics such as mean, standard deviation, minimum, maximum, and quartile values, enabling a comprehensive understanding of the numerical features' central tendencies and variability.

```
[ ] df.describe()
```

	S1_Temp	S2_Temp	S3_Temp	S4_Temp	S1_Light	S2_Light	S3_Light	S4_Light	S1_Sound	S2_Sound	S3_Sound	S4_Sound	S5_CO2	S5_CO2_Slope	S6_PIR	S7_PIR	Room_Occupancy_Count
count	10129.000000	10129.000000	10129.000000	10129.000000	10129.000000	10129.000000	10129.000000	10129.000000	10129.000000	10129.000000	10129.000000	10129.000000	10129.000000	10129.000000	10129.000000	10129.000000	10129.000000
mean	25.454012	25.546059	25.056621	25.754125	25.445059	26.01629	34.248494	13.220259	0.168178	0.120066	0.158119	0.103840	460.860401	-0.004830	0.090137	0.079574	0.398559
std	0.351361	0.586325	0.427283	0.356434	51.011264	67.30417	58.400744	19.602219	0.316709	0.265503	0.413637	0.120683	199.964940	1.164990	0.286392	0.270645	0.893633
min	24.940000	24.750000	24.440000	24.940000	0.000000	0.000000	0.000000	0.000000	0.060000	0.040000	0.040000	0.050000	345.000000	-6.296154	0.000000	0.000000	0.000000
25%	25.190000	25.190000	24.690000	25.440000	0.000000	0.000000	0.000000	0.000000	0.070000	0.050000	0.060000	0.060000	355.000000	-0.046154	0.000000	0.000000	0.000000
50%	25.380000	25.380000	24.940000	25.750000	0.000000	0.000000	0.000000	0.000000	0.080000	0.050000	0.060000	0.080000	360.000000	0.000000	0.000000	0.000000	0.000000
75%	25.630000	25.630000	25.380000	26.000000	12.000000	14.000000	50.000000	22.000000	0.080000	0.060000	0.070000	0.100000	465.000000	0.000000	0.000000	0.000000	0.000000
max	26.380000	29.800000	26.190000	26.560000	165.000000	258.000000	280.000000	74.000000	3.880000	3.440000	3.670000	3.400000	1270.000000	8.980769	1.000000	1.000000	3.000000

Figure 3: output of “.describe()” function for the dataset

To understand the distribution of records across the dates, the below column chart has been created:

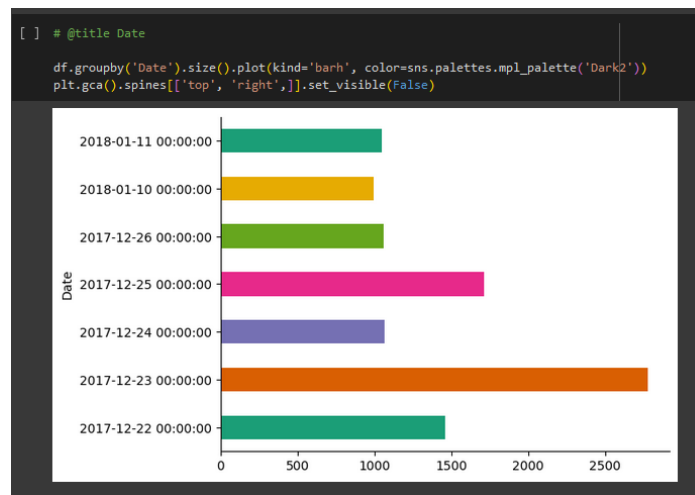


Figure 4: number of records per date

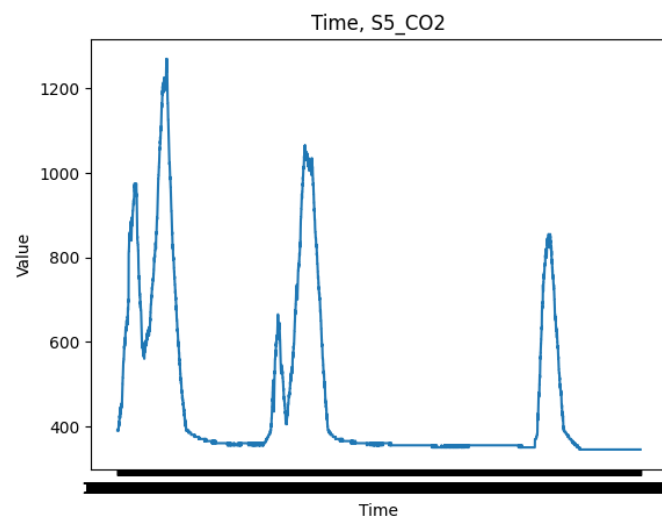


Figure 5: correlation between time and value of CO-2 sensor

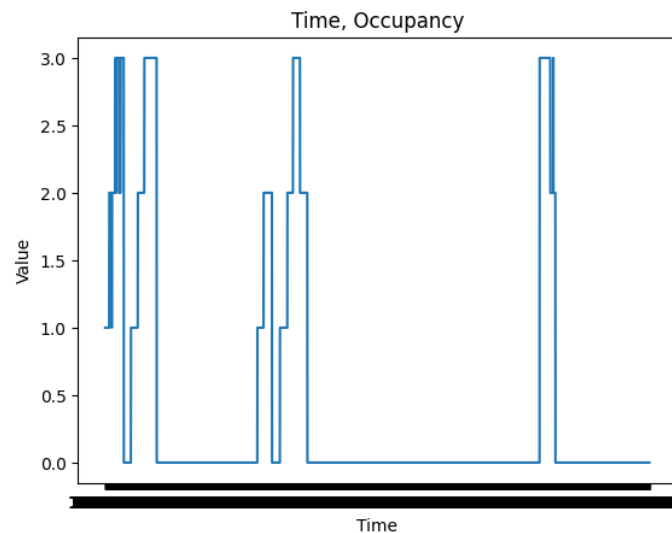


Figure 6: the number of humans in the time period

By comparing the correlation between time and value of CO-2 sensor (figure-4) and the number of humans in the time period (figure-5). It can be observed that as the value of CO-2 increases, the number of people inside the room increases as well, therefore it can be said the the value of CO-2 is proportion to the number of human.

We conducted feature correlation analysis using pandas' ".corr()" function, complemented by visualization through a heatmap in seaborn. The objective was to identify and eliminate correlated variables, consequently improving model performance. Prior to the correlation analysis, we excluded categorical features and the target variable to focus solely on numerical attributes. This approach allowed us to gain insights into the interrelationships between features and ascertain their potential impact on the modeling process.

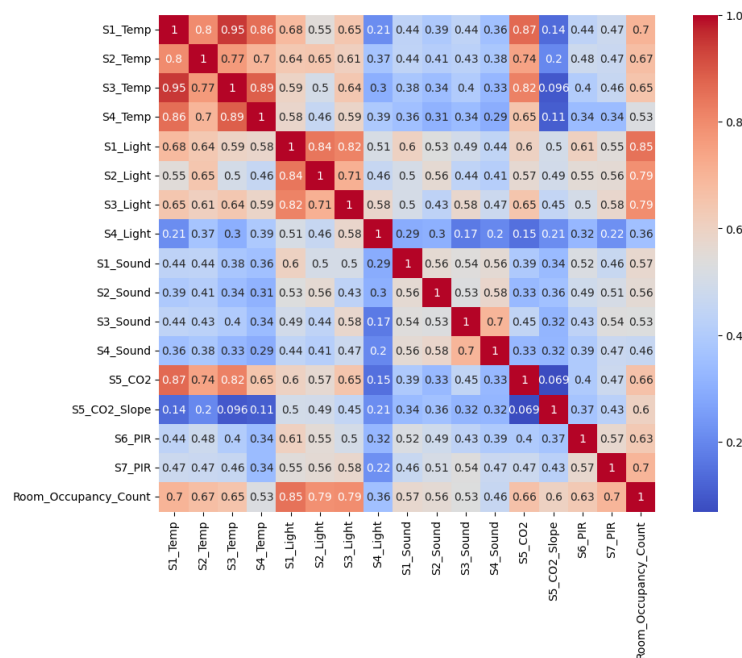


Figure 7: heatmap designed with the feature correlations

Baseline Methods

Random Classifier: A random classifier serves as a baseline model for evaluating the performance of more sophisticated algorithms. It predicts class labels randomly, without considering input features, making it useful for scenarios with little prior knowledge or highly imbalanced datasets. By comparing the performance of other models against the random classifier using evaluation metrics like accuracy, precision, and recall, researchers can assess whether the model learns meaningful patterns from the data. Despite its simplicity and computational efficiency, the random classifier provides valuable insights into model performance and is particularly useful for assessing algorithms in imbalanced datasets. However, it should only be used for experimentation and evaluation purposes, as it doesn't offer meaningful predictions in practical applications.

Dummy Classifier: Dummy classifier, similar to a random classifier, serves as a baseline model for evaluating more complex machine learning algorithms. It generates predictions using simple rules, such as predicting the most frequent class (for stratified strategy) or predicting classes based on class prior probabilities. Dummy classifiers are particularly useful in scenarios with imbalanced datasets or when there is little prior knowledge about the data. By comparing the performance of other models against the dummy classifier using evaluation metrics like accuracy, precision, and recall, researchers can assess the effectiveness of their models in learning meaningful patterns from the data. Despite their simplicity and ease of implementation, dummy classifiers offer valuable insights into model performance and are commonly used in machine learning experimentation and evaluation tasks. However, like random classifiers, they should only be used for assessment purposes and not for making meaningful predictions in real-world applications.

Model Evaluation of Baseline Models

As a result of using the Baseline methods, for Random Classifier, we got the following results:

Accuracy	24.14
Precision	67.36
Recall	24.14
F1 Score	32.02

Table 1: Results of standard metrics using Random Classifier

For the cross-validation scores of Random Classifier, we got the following results:

```
Cross-validation scores (Random Classifier): [0.24749422 0.22050887 0.23688272 0.2507716 0.25848765]  
Mean cross-validation score (Random Classifier): 0.24282901186974692
```

The result of using Dummy Classifier was as follows:

Accuracy	66.78
Precision	66.64
Recall	66.78
F1 Score	66.71

Table 2: Results of standard metrics using Dummy Classifier

As the cross-validation for the dummy classifier, the below results is reached:

Cross-validation scores (Dummy Classifier): [0.67154973 0.6769468 0.66435185 0.67746914 0.66820988]
Mean cross-validation score (Dummy Classifier): 0.6717054789304854

Data Processing and Manipulation

To understand distribution of the number of people during the day, we partitioned the day into five different parts:

1. Morning (6 AM to 11 AM)
2. Noon (11AM to 1PM)
3. Afternoon (1PM to 6PM)
4. Evening (6PM to 11PM)
5. Night (11PM to 6AM)

The below chart, shows the distribution of the number of people by using the above partitioning:

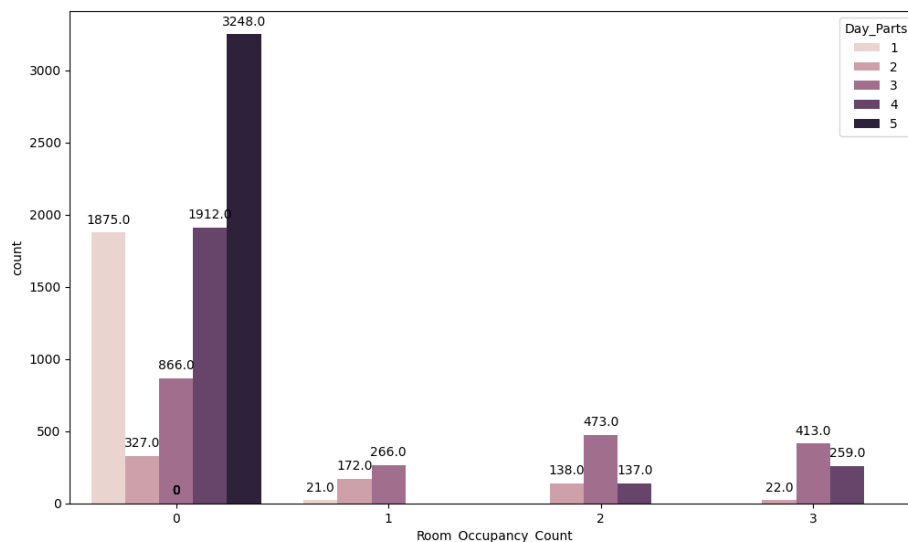


Figure 8: Number of people in destributed in the daytime.

In order to process data easier, faster and to get a higher accuracy, we decided to drop some columns after making a combination of some values rather than using all columns separately. As a result, we calculated the mean value of the S1 and S3 temprature sensors, and created the Avg-temp column instead of using both sensors' values separately. Additionally, for Leveraging the boolean nature of the PIR sensor data, we created a new column 'S6-PIR' to indicate any motion events. This column reflects a logical OR operation between the individual sensor readings. As the result, after the processing and manipulation, we created the below dataset table:

	Date	Time	Avg_Temp(S1-S3)	S2_Temp	S4_Temp	S1_Light	S2_Light	S3_Light	S4_Light	S1_Sound	S2_Sound	S3_Sound	S4_Sound	S5_CO2	S5_CO2_Slope	S6_PIR	Room_Occupancy_Count
0	2017-12-22	1900-01-01 10:49:41	24.750	24.75	25.38	121	34	53	40	0.08	0.19	0.06	0.06	390	0.769231	0	1
1	2017-12-22	1900-01-01 10:50:12	24.750	24.75	25.44	121	33	53	40	0.93	0.05	0.06	0.06	390	0.646154	0	1
2	2017-12-22	1900-01-01 10:50:42	24.750	24.75	25.44	121	34	53	40	0.43	0.11	0.08	0.06	390	0.519231	0	1
3	2017-12-22	1900-01-01 10:51:13	24.780	24.75	25.44	121	34	53	40	0.41	0.10	0.10	0.09	390	0.388462	0	1
4	2017-12-22	1900-01-01 10:51:44	24.780	24.75	25.44	121	34	54	40	0.18	0.06	0.06	0.06	390	0.253846	0	1
...
10124	2018-01-11	1900-01-01 08:58:07	24.875	25.13	25.31	6	7	33	22	0.09	0.04	0.06	0.08	345	0.000000	0	0
10125	2018-01-11	1900-01-01 08:58:37	24.875	25.06	25.25	6	7	34	22	0.07	0.05	0.05	0.08	345	0.000000	0	0
10126	2018-01-11	1900-01-01 08:59:08	24.910	25.06	25.25	6	7	34	22	0.11	0.05	0.06	0.08	345	0.000000	0	0
10127	2018-01-11	1900-01-01 08:59:39	24.910	25.06	25.25	6	7	34	22	0.08	0.08	0.10	0.08	345	0.000000	0	0
10128	2018-01-11	1900-01-01 09:00:09	24.910	25.06	25.25	6	7	34	22	0.08	0.05	0.06	0.08	345	0.000000	0	0

Figure 9: Table of the dataset after processing and data manipulation

References

Adarsh Pal Singh, Vivek Jain, Sachin Chaudhari, Frank Alexander Kraemer, Stefan Werner and Vishal Garg, "Machine Learning-Based Occupancy Estimation Using Multivariate Sensor Nodes," in 2018 IEEE Globecom Workshops (GC Wkshps), 2018.

ROOM OCCUPANCY PREDICTION, Kirthana Shri Chandra Sekar, Anjali Dayaram Kshirsagar, Neeraj Rangwani, Samarth Saxena ([click here!](#))