**A close-up of a logo

Description automatically generated**

**EDS 6340 Introduction to Data Science**

**Group\_2 Project Report**

**Room Occupancy Estimation**

|  |  |
| --- | --- |
| **Group Members** | **PS\_ID** |
| KARTHIKEYAN RAVICHANDRAN | 2260006 |
| TEJAS MURALI | 1612032 |
| SUNIL KUMAR KOMMINENI | 2304770 |
| POOYAN GHODRATI | 2342802 |
| LEKHA CHITTAJALLU | 2311347 |
| VINEETH YADAMA | 2295464 |

**ABSTRACT**

Room Occupancy Estimation plays a major role in optimizing energy usage, enhancing security, and improving overall efficiency in various environments. This dataset was obtained from the UCI Machine Learning Repository. This study represents a comprehensive data analysis of the dataset collected for room occupancy estimation. Data set for estimating the precise number of occupants in a room using multiple non-intrusive environmental sensors like temperature, light, sound, CO2, and PIR. Exploratory data analysis techniques are applied to gain a deep understanding of the dataset characteristics and correlation of the features. Later various machine learning techniques are used on the dataset with many variations like feature selection to finally decide which machine learning technique works best on this dataset.

**INTRODUCTION**

The effective estimation of room occupancy is a pivotal element in the optimization of resource utilization, energy efficiency, and overall operational management in diverse environments. In the era of smart buildings and IoT-driven technologies, accurate predictions of when and how spaces are occupied hold significant implications for enhancing security, optimizing energy consumption, and improving overall user experience. This study delves into a comprehensive dataset specifically curated for room occupancy estimation, employing advanced data analysis techniques to extract meaningful insights and contribute to the refinement of occupancy prediction models.

**DATA DESCRIPTION**

The experimental testbed for occupancy estimation was deployed in a 6m x 4.6m room. The setup consisted of 7 sensor nodes and one edge node in a star configuration with the sensor nodes transmitting data to the edge every 30s using wireless transceivers. Five different types of non-intrusive sensors were used in this experiment: temperature, light, sound, CO2, and digital passive infrared (PIR). The CO2, sound and PIR sensors needed manual calibration. For the CO2 sensor, zero-point calibration was manually done. The sound sensor is essentially a microphone with a variable-gain analog amplifier attached to it. Therefore, the output of this sensor is analog which is read by the microcontroller ADC in volts. The potentiometer tied to the gain of the amplifier was adjusted to ensure the highest sensitivity. The PIR sensor has two trim pots: one to tweak the sensitivity and the other to tweak the time for which the output stays high after detecting motion. Both were adjusted to the highest values. Sensor nodes S1-S4 consisted of temperature, light, and sound sensors, S5 had a CO2 sensor and S6 and S7 had one PIR sensor each that were deployed on the ceiling ledges at an angle that maximized the sensor's field of view for motion detection. The data was collected for 4 days in a controlled manner with the occupancy in the room varying between 0 and 3 people.

A screenshot of a computer

Description automatically generated

**PRE-PROCESSING**

Data preprocessing is a crucial step in the data analysis pipeline, especially when working with datasets for room occupancy estimation.

**Handling Missing/duplicate Values:**

Identify and handle missing values in your dataset. Options include dropping rows with missing values, filling missing values with the mean or median, or using advanced imputation techniques. Duplicate values are replaced with the mean or mode of the features.

|  |  |
| --- | --- |
| A screenshot of a computer code  Description automatically generated | A screenshot of a computer code  Description automatically generated |

**Outliers:**

Identify and handle outliers. Use statistical methods or visualization techniques to detect outliers and decide whether to remove or transform them.

A screenshot of a graph

Description automatically generated

After the data preprocessing, some of the visualizations are:

A pie chart with different colored circles

Description automatically generated

A screenshot of a graph

Description automatically generated

**MODELLING TECHNIQUES**

**1. Linear Regression:**

Linear regression is a statistical method used for modelling the relationship between a dependent variable (target) and one or more independent variables (features). The relationship is assumed to be linear, meaning that a change in the value of the independent variable(s) is associated with a linear change in the dependent variable.

**2. K Nearest Neighbours (KNN):**

KNN is a simple and intuitive machine-learning algorithm used for both classification and regression tasks. It works by finding the 'k' training data points closest to a new data point and making predictions based on the majority class (for classification) or average (for regression) of those neighbours.

**3. Random Forest:**

Random Forest is an ensemble learning method that builds a multitude of decision trees during training and merges them to get a more accurate and stable prediction.

**4. SVM with linear and non-linear kernels:**

SVM with a linear kernel is designed for linearly separable data. It finds the hyperplane that best separates the classes by maximizing the margin, the distance between the hyperplane and the nearest data points of each class. SVM with non-linear kernels allows for handling non-linearly separable data by mapping the input features into a higher-dimensional space.

**5. Gradient boosting:**

Gradient Boosting is an ensemble learning method that builds a predictive model in a stage-wise fashion by combining the predictions of multiple weak learners, often decision trees, to create a strong predictive model.

A graph of different sizes of blue squares

Description automatically generated with medium confidence

**VARIABLE SELECTION**

Variable selection is a crucial step in the process of building a predictive model. It involves choosing a subset of relevant features (variables) from the original set of features to improve model performance, interpretability, and computational efficiency. Overfitting can occur if variable selection is based solely on the training set performance.

**LASSO (L1 Regularization):**

Introduces a penalty term in the model training process, leading to sparse coefficients and feature selection. Lasso's ability to shrink certain coefficients to zero makes it particularly useful for feature selection in high-dimensional datasets, where irrelevant or redundant features can be effectively excluded from the model. After L1 regularization,

A graph of different sizes of blue squares

Description automatically generated with medium confidence

**HYPERPARAMETER TUNING**

Employ grid search or random search to explore a range of hyperparameter combinations. Grid search exhaustively evaluates predefined hyperparameter values, while random search randomly samples hyperparameter values within defined ranges. After tuning the hyperparameters of various models,

A graph of different colored squares

Description automatically generated

**Variance-bias tradeoff:**

The bias-variance tradeoff highlights the inverse relationship between bias and variance – as one decreases, the other increases. Adjusting the complexity of a model influences the bias and variance. Increasing complexity often reduces bias but raises variance, while reducing complexity can increase bias but decrease variance.

**Cross Validation:**

Cross-validation is a resampling technique used to assess a model's performance by partitioning the dataset into subsets, training the model on some of them, and evaluating the remaining subsets. Cross-validation helps detect issues like overfitting or underfitting by providing a more robust evaluation of a model's generalization performance, reducing the impact of data variability on model assessment.

**Wrapper Methods:**

Wrapper methods assess the performance of a model by using subsets of features and are guided by the predictive performance of the model.

**Forward selection:** Add features one by one, selecting the one that improves model performance the most.

**Backward elimination:** Start with all features and remove them one by one, eliminating the one that has the least impact on performance.

**Recursive Feature Elimination (RFE):** Iteratively removes the least important features based on model performance.