Room Occupancy Estimation

Monjure Mowla	**180104027**
Kazi Fuad Bin Akhter	**180104038**
Nawrin Tabassum	**180104045**

Project Report

Course ID: CSE 4214

Course Name: Pattern Recognition Lab

Semester: Fall 2021

Department of Computer Science and Engineering

Ahsanullah University of Science and Technology

Dhaka, Bangladesh

08 September, 2022

Room Occupancy Estimation

Submitted by

Monjure Mowla	**180104027**
Kazi Fuad Bin Akhter	**180104038**
Nawrin Tabassum	**180104045**

Submitted To

Faisal Muhammad Shah

Sajib Kumar Saha Joy,

Department of Computer Science and Engineering

Ahsanullah University of Science and Technology

Department of Computer Science and Engineering

Ahsanullah University of Science and Technology

Dhaka, Bangladesh

08 September, 2022

ABSTRACT

The number of occupants in a room is a crucial factor in determining the room's energy consumption. By correctly estimating the number of occupants, energy use can be controlled, resulting in cost minimization. Temperature and light are two different sources of energy that are very useful for human beings and should be utilized properly. Sometimes, these energies are wasted for varying causes. People might forget to turn off the air conditioner or light, leading to waste of this energy. To overcome these problems, a room occupancy-based controlling system can play a vital role. Machine learning algorithms can be useful to correctly predict the number of occupants in a room from various features. Apart from light and temperature, CO2 level, PIR level, and sound can also contribute to finding the number of people present in the room. Machine learning techniques can be used along with IoT devices to automatically control the light or air conditioner. In this study, we applied machine learning in sensor readings to estimate the number of occupants in a room. We extracted several features that are related to occupancy estimation. Then we applied five machine learning algorithms: Support Vector Machine (SVM), K Nearest Neighbors (KNN), Logistic Regression, Decision Tree, and Naïve Bayes. The performance of the classifiers was evaluated using accuracy, precision, recall, and f1-score. The proposed approach can also be used for occupancy estimation in other places like offices, schools, etc. to reduce cost and energy consumption.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The waste of energy is a major problem worldwide which not only reduces energy but also increases cost. Energy can be wasted for various reasons and from various sources. Sometimes the energy is wasted intentionally or unintentionally by humans. The waste of energy may lead to serious consequences. If the light is turned on for a long time, heat energy is produced which increases the temperature of the environment. The air conditioner produces different gases which are very harmful to the environment. So, the energy consumption of a place must be taken care of to keep the environment safe and healthy.

The energy consumption of a small place like a living or dining room depends mostly on the number of people available in that room. When a large number of people are present in the room, the number of lights or AC that has been kept on is also high. So, by correctly estimating the number of people in a room, the amount of energy consumption can also be detected. If the number of people in a room is very high, the energy consumption will also be very high. If the number of people is less, the consumption of energy will be low consequently.

The total number of occupants in a room depends on various factors. The current study includes a few such related features such as temperature, light, sound, CO2 level, and PIR level. When there are many people in a room, the temperature is supposed to be higher than usual. A possible reason for it can be the body temperature of humans. Moreover, smoking increases the temperature of a room which is only possible when there are people inside a room. Cooking may also increase the temperature of a room. Again, it also depends on the availability of people in a room. The amount of light in a room also depends on the number of people. When there are more people, the room will be more lightly as people will be involved in different activities that require the lights to be turned on. When people go outside, they turn the lights off. So, the room will be less lightly when there are not many people in the room. The amount of sound is also highly related to the number of occupants in a room. When there are many people in the room, the amount of sound is supposed to be

higher as people are usually involved in various activities inside the room such as, talking, laughing, watching TV, and listening to music. A room is usually quiet when there are not many people in the room. The overall motion also increases with the number of people since motion is only possible when there are people available in a room. When the room is empty or there are no people in the room, there will be no motion. Finally, the room occupancy also depends on the CO2 level. The CO2 level tends to be higher with the presence of more people. As people exhale CO2 while breathing, it increases the amount of CO2 in the room.

Machine learning (ML) can be useful in estimating the number of occupants in a room. From the features described above, ML can correctly predict the number of persons in a room. ML is a subset of artificial intelligence which extract patterns from available data and predicts target features from a list of predictor features. A model is trained to find patterns from data. When a new sample arrives, the model predicts its label from the predictor features using the knowledge gained from the training. Machine learning can be used with IoT devices to automatically control the light and air conditioner of a room which can save us from energy waste. If ML is integrated with an IoT device, it will predict the occupancy of a room from available data. Then the device will control the AC or light automatically. If there are not many people in the room, the light or AC level will be low. When the number of people is very high in a room, the light or AC level will be made higher automatically.

In this study, we have utilized ML algorithms to estimate the occupancy of a room. We used the room occupancy estimation dataset from the UCI machine learning repository. There is a total of 18 predictor features in the dataset. We extracted several other features, resulting in a dataset of 25 features. From this dataset, we predicted the number of occupants in a room from various features such as light, temperature, sound, CO2 level, PIR level, etc. We applied five machine learning algorithms: Support Vector Machine (SVM), K Nearest Neighbors (KNN), Logistic Regression, Decision Tree, and Naïve Bayes. The performance of the classifiers was evaluated using accuracy, precision, recall, and f1-score. The accuracy for the models is 95.26%, 97.83%, 97.09%, 95.85%, and 95.11% respectively. The purpose of this study is to reduce the cost and energy consumption by correctly estimating the room occupancy so that the AC or light can be adjusted accordingly. The results suggest that the proposed approach can also be used for occupancy estimation in other places like offices, schools, etc. to reduce cost and energy consumption.

Chapter 2

Literature Reviews

[1] In this paper, the authors presented results on the application of machine learning to the detection of human presence and estimation of the number of occupants in offices using data from an IoT LoRa-based indoor environment monitoring system at Aalborg University, Denmark. They identified the problem as either binary or multi-class classification and apply a two layer feed forward neural network to the data. The data used for training, validation and testing of the network comprises of environmental data from the IoT sensors and manual recordings of the door and window states. Results show that the classifier is able to correctly determine occupancy of offices from the IoT sensor measurements with accuracy up to 94.6% and 91.5% for the binary (presence or absence of persons) and multiclass (no person, one person or two or more persons) problems, respectively.

[2] In this paper , the accuracy of the prediction of occupancy in an office room using data from light, temperature, humidity and CO2 sensors has been evaluated with different statistical classification models using the open source program R. Three data sets were used in this work, one for training, and two for testing the models considering the office door opened and closed during occupancy. Typically the best accuracies (ranging from 95% to 99%) are obtained from training Linear Discriminant Analysis (LDA), Classification and Regression Trees (CART) and Random Forest (RF) models. The results show that a proper selection of features together with an appropriate classification model can have an important impact on the accuracy prediction. Information from the time stamp has been included in the models, and usually it increases the accuracy of the detection. Interestingly, using only one predictor (temperature) the LDA model was able to estimate the occupancy with accuracies of 85% and 83% in the two testing sets.

[3] Here , the authors introduced the sensor-utility-network (SUN) method for occupancy estimation in buildings. Based on inputs from a variety of sensor measurements, along with historical data regarding building utilization, the SUN estimator produces occupancy estimates through the solution of a recedinghorizon convex optimization problem. State-of-the-art on-line occupancy algorithms rely on indirect measurements, such as CO2 levels, or people counting sensors which are subject to significant errors and cost. The newly proposed method was evaluated via experiments in an office building environment. Estimation accuracy is shown to improve significantly when all available data is incorporated in the estimator. In particular, it is found that the average estimation error at the building level is reduced from 70% to 11% using the SUN estimator, when compared to the conventional approach that relies solely on occupancy level or flow measurements.

[4] This study compared three popular machine learning algorithms, including k-nearest neighbors (kNN), support vector machine (SVM), and artificial neural network (ANN), combined with three data sources, including environmental data, Wi-Fi data, and fused data, to optimize the occupancy models' performance in various scenarios. Three error measurement metrics, the mean average error (MAE), mean average percentage error (MAPE), and root mean squared error (RMSE), have been employed to compare the models' accuracies. Examined with an on-site experiment, the results suggest that the ANN-based model with fused data has the best performance, while the SVM model is more suitable with Wi-Fi data. The results also indicate that, comparing with independent data sources, the fused data set does not necessarily improve model accuracy but shows a better robustness for occupancy prediction.The MAPE value for models were 44.4%, 36.6% and 37.3%.

[5] In this study, occupancy count has been estimated by training models on passive infrared (PIR) sensor and booking data.A system was built with two artificial neural network models, where a binary classification model predicts the occupancy state of rooms, i.e. if it is empty or not. If a room is considered occupied, a subsequent regression model predicts the number of occupants. The author developed different neural network regression models and trained them on manually collected ground truth data from real meetings.The performance of the models were then compared and the best results were obtained with a bidirectional long-short term memory architecture. It reaches a mean squared error of 2.27 and mean absolute error of 0.94. It predicts the number of people with an error margin of one individual at 85% respectively 49% accuracy for occupants ranging from 1 to 7 and 8 to 14. At an error margin of two individuals the results for the same intervals are 94% respectively 66% accuracy.

[6] In this paper, several machine learning techniques (Ridge Regression, Kernel Ridge Regression, Multilayer Perceptron, Radial Basis Function Networks) were surveyed in order to construct models to forecast daily occupancy rates for a hotel, given historical records of bookings and occupation. Several approaches related to dataset construction and model validation are discussed. The results obtained in terms of the Mean Absolute Percentage Error (MAPE) are promising, and supportthe use of machine learning models as a tool to help solve the problem of occupancy rates and demand forecasting.The MAPE values for Ridge Regression, Kernel Ridge Regression, Multilayer Perceptron and RBF Network were 8.6962%, 8.69622%, 12.89174% and 26.32086%

Chapter 3

Background Study

3.1 Support Vector Machine

A support vector machine (SVM) is a type of supervised machine learning classification algorithm. The goal of an SVM is to find a hyperplane (a line in 2 dimensions, or a plane in 3 dimensions) that maximally separates a dataset into two classes. The goal is to find the line (or plane) that is as far away from all of the data points of one class as possible, while still being close to all of the data points of the other class.

Once this hyperplane have been found, it can be used to classify new data points as belonging to either one class or the other. Data points that are on the correct side of the hyperplane are classified correctly, while data points that are on the wrong side are classified incorrectly.

The thickness of the margin is a measure of how well the SVM is doing at separating the two classes. A wider margin indicates a better separation, while a narrower margin indicates a worse separation. The SVM algorithm is implemented in a number of ways, with the most common being the use of quadratic programming. This is a method of solving optimization problems that involve variables that are subject to linear constraints.The SVM algorithm can be used for both linear and non-linear classification. When using SVMs for linear classification, the data must be linearly separable. This means that there must be a straight line (or hyperplane in higher dimensions) that can be used to perfectly separate the two classes.

The SVM algorithm has a number of advantages over other classification algorithms. First, it is very effective in high dimensional spaces. Second, it is relatively insensitive to overfitting. Third, it can use different kernel functions to non-linearly transform the data, making it very versatile.Fourth, SVMs have been shown to perform well on a variety of real-world datasets. Finally, the resulting classifier is very easy to interpret, as it is simply a decision boundary between the two classes.

3.2 K Nearest Neighbors

The k nearest neighbors classifier is a supervised learning algorithm that can be used for both classification and regression tasks. The algorithm is based on the principle that similar instances tend to belong to the same class. The k nearest neighbors classifier works by taking a test instance, finding the k nearest training instances, and then predicting the class of the test instance based on the class labels of the training instances.KNN is a non-parametric algorithm. It does not make any assumptions about the underlying data and therefore is very versatile.This algorithm first calculates the distance between the new data point and all training data points. The number of nearest neighbors is then determined (usually using Euclidean distance) and the majority class label is assigned to the new data point

One of the main advantages of the k nearest neighbors (KNN) classifier is that it is very simple to implement and can be applied to a variety of data sets. Additionally, the classifier is not sensitive to outliers and is resistant to overfitting.

3.3 Logistic Regression

Logistic regression is a classification technique that is used to predict the probability of a categorical dependent variable. The dependent variable in logistic regression is binary in nature having data values in the form of 0s and 1s. In simple words, the dependent variable in logistic regression follows Bernoulli Distribution. A logistic regression classifier predicts the probability of occurrence of an event by fitting data to a logit function. The dependent variable in logistic regression is binary. Logistic regression can be used to predict both categorical and numerical data. It uses Maximum Likelihood Estimation (MLE) to estimate the parameters of the model.It is more robust.It can also be used for feature selection.

The main advantage of logistic regression is that it is a relatively simple and easy to interpret model. Additionally, logistic regression is not affected by multicollinearity and can be used to predict both categorical and numerical data.

3.4 Naive Bayes

A Naive Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions. A more descriptive term for the independence assumption would be "unconditioned independence".

In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. For example, a fruit may be

considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as 'naive'.

Naive Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. Maximum-likelihood training can be done by evaluating a closed-form expression, which takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers.

Naive Bayes classifiers are often used in text classification problems, such as spam detection and sentiment analysis. They are also commonly used in problems where the amount of data is limited.

3.5 Decision Tree

Decision trees are a type of machine learning algorithm that are used to predict the output of a target variable, based on a set of input variables. Decision trees are a non-parametric method, which means that they do not make any assumptions about the distribution of the data.

The algorithm works by recursively partitioning the data into different subsets, based on the values of the input variables. Each subset is then used to predict the output of the target variable. The predictions from each subset are then combined to form the final prediction.

Decision trees are a powerful tool for classification, as they are able to handle both linear and non-linear data. They are also able to handle data with missing values. However, decision trees can be overfit if the data is not split properly.

To avoid overfitting, it is important to tune the parameters of the decision tree algorithm. This can be done by using cross-validation. Cross-validation is a technique that is used to split the data into multiple subsets, and then use each subset to train and test the model. This helps to prevent overfitting, as the model is not trained on all of the data.

Once the decision tree model has been trained, it can be used to predict the output of the target variable for new data.

Chapter 4

Dataset

We have used the room occupancy estimation dataset from the UCI machine learning repository [7]. The dataset consists of multivariate time-series data. There are a total of 10129 samples and 16 features in the dataset. 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 sensors were used in this experiment: temperature, light, sound, CO2, and passive infrared (PIR). Four temperatures, four lights, four sounds, and two PIR sensors were employed in the room for data collection. The data was collected for a period of 4 days in a controlled manner with the occupancy in the room varying between 0 and 3 people. The ground truth of the occupancy count in the room was noted manually. The attribute information of the dataset is given below –

1. Date: YYYY/MM/DD
2. Time: HH:MM:SS
3. Temperature: In degree Celsius
4. Light: In Lux
5. Sound: In Volts (amplifier output read by ADC)
6. CO2: In PPM
7. CO2 Slope: Slope of CO2 values taken in a sliding window
8. PIR: Binary value conveying motion detection
9. Room Occupancy Count: Ground Truth

The dataset comprises four classes labeled as 0, 1, 2, and 3. The dataset is heavily imbalanced as shown in Table 4.1. In order to solve this problem, we oversampled the data. Our final oversampled dataset consisted of 26360 samples.

Table 4.1: Class Distribution

Class label	No. of samples
0	8228
1	450
2	748
3	694

Chapter 5

Methodology

5.1 Dealing with NULL Values

The data set consists of 10,129 samples. But there will be a problem if some of these samples have null values. So, at first we checked whether there were such samples. We observed that each of the features of the dataset is free of null values. Thus, we didn't need to remove any sample from the dataset.

5.2 Feature Extraction

We extracted some new features from the original dataset's features. These were: Date_Time, Day of week, Hour of day, Mean_Temperature, Mean_light, Mean_sound, Footsteps_Sound, PIR_Level, MA_CO2, MV_CO2.

Date_Time was extracted by concatenating the Date and Time features of the original dataset.

Day of week was created to differentiate between weekdays and weekends.

Hour of day was derived to find out the crowdest phases of the day.

Mean_Temperature was determined from getting the average values of the temperature sensors.

Mean_light was calculated from getting the average values of the 4 light sensors.

Mean_sound was found out from the mean of the sound sensors. There were also 4 types of this sensor.

Footsteps_Sound was used as the product of the average PIR sensor value and the average sound sensor value.

PIR_Level was extracted to estimate the crowding of the room by getting a sense of the movement of people in the room.

MA_CO2 was calculated to find the average level of CO2 for previous consecutive days. On the other hand, MV_CO2 was derived to the variance of CO2 level for the previous days.

5.3 Data Visualization

We have visualized some of our data to graphically represent our derived dataset. This helps in understanding the data from a new perspective. We have shown the temperature, CO2 levels and the effect of footsteps at different time of a day.

5.4 Oversampling

The problem with imbalanced classification is that there are too few examples of the minority class for a model to effectively learn the decision boundary. One way to solve this problem is to oversample the examples in the minority class. This can be achieved by simply duplicating examples from the minority class in the training dataset prior to fitting a model. This can balance the class distribution but does not provide any additional information to the model. An improvement on duplicating examples from the minority class is to synthesize new examples from the minority class. This is a type of data augmentation for tabular data and can be very effective. The most widely used approach to synthesizing new examples is called the Synthetic Minority Oversampling TEchnique, or SMOTE for short. SMOTE works by selecting examples that are close in the feature space, drawing a line between the examples in the feature space and drawing a new sample at a point along that line. Specifically, a random example from the minority class is first chosen. Then, k of the nearest neighbors for that example are found (typically k=5). A randomly selected neighbor is chosen and a synthetic example is created at a randomly selected point between the two examples in feature space. . This procedure can be used to create as many synthetic examples for the minority class as are required.

The approach is effective because new synthetic examples from the minority class are created that are plausible, that is, are relatively close in feature space to existing examples from the minority class.

In our paper, we have used this strategy because our initial dataset was very imbalanced. As such, we have multiple instances of minority classes added to our dataset to give far better predictions.

5.5 Separating Features and Labels

We have splitted all the columns into 2 parts. The last column, known as the target column, has been used as the label. On the other hand, all the preceding columns have been used as features for prediction in this paper.

5.6 Dataset Normalization

The goal of normalization is to transform features to be on a similar scale. This improves the performance and training stability of the model. Four common normalization techniques may be useful: scaling to a range, clipping, log scaling, z-score. In our work, we have used the method of scaling to a range. We have applied this process for all the features. This method means converting floating-point feature values from their natural range (for example, 100 to 900) into a standard range—usually 0 and 1 (or sometimes -1 to +1) using the following simple formula to scale to a range:

$$X^{`} = (X - Xmin)/(Xmax - Xmin)$$

Scaling to a range was a good choice for us because the following conditions were met: We knew the approximate upper and lower bounds on our data with few or no outliers. Our data is approximately uniformly distributed across that range.

5.7 Splitting the Dataset

We have split the entire datasets into two segments. In the train portion, we have distributed 80% of data whereas the rest of the 20% data have been put into the test portion.

5.8 Models

We have used 5 different models. These are: Naive Bayes Classifier, Decision Tree, K-nearest Neighbor, Support Vector Machine, and Logistic Regression.

In the Naive Bayes Classifier, Gaussian Naive Bayes was used for Normal Distribution.

In the Decision Tree, max_depth was used as 3. It was done to specify that the maximum depth of the tree will be 3 levels.

In SVM, we have used the default kernel type ‘rbf’.

In KNN, we have taken 50 as the value of 'n_neighbors'.

In Logistic Regression, we have specified 2 parameters. For multiclass problems, 'lbfgs' is one of the algorithms to handle multinomial loss. Hence, we used it as the 'solver' in the parameter. Another parameter we have used is 'multi_class'. We have taken 'auto' as the value of 'multi_class' because 'auto' selects 'ovr' if the data is binary, or if solver='liblinear', and otherwise selects 'multinomial'.

Chapter 6

Experiments and Results

We have taken 4 metrics for evaluating the performance of our 5 models. These are: accuracy, precision, recall and F1-score. The table presents our experiment results.

Table 6.1: Performance Metrics of the Models

Model	Accuracy	Precision	Recall	F1 score
Naive Bayes	95.11%	85.32%	87.08%	85.92%
Decision Tree	95.85%	89.32%	96.09%	91.93%
KNN	97.83%	93.40%	97.02%	95.00%
SVM	95.26%	89.62%	93.58%	90.80%
Logistic Regression	97.09%	92.82%	94.58%	93.47%

We have also plotted the ROC curves for all of these models. These are given below:

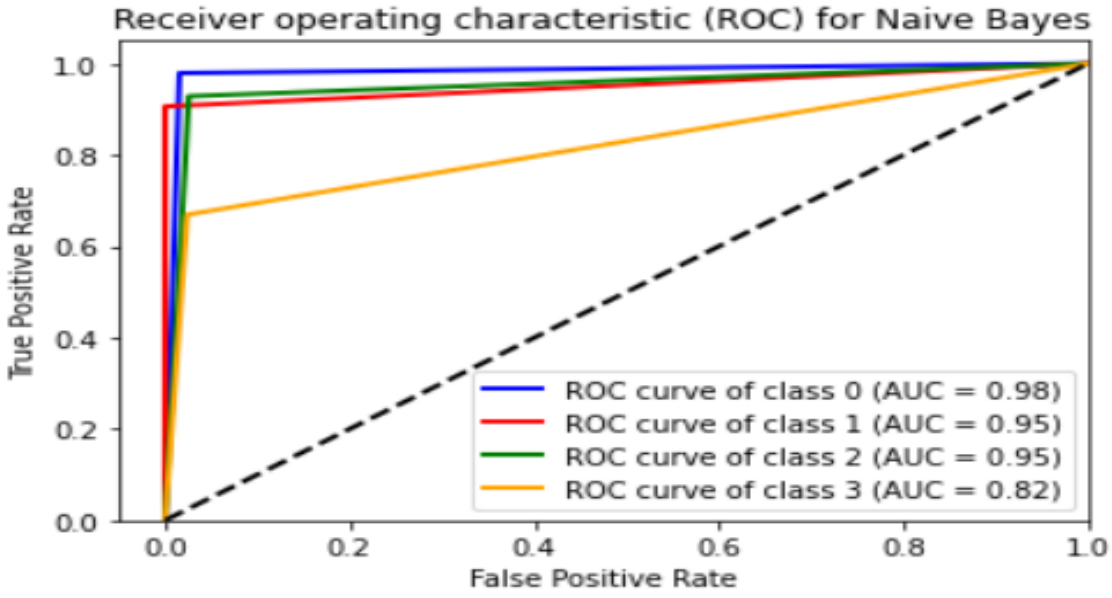

Figure 6.1: Receiver operating characteristic (ROC) curve for Naive Bayes

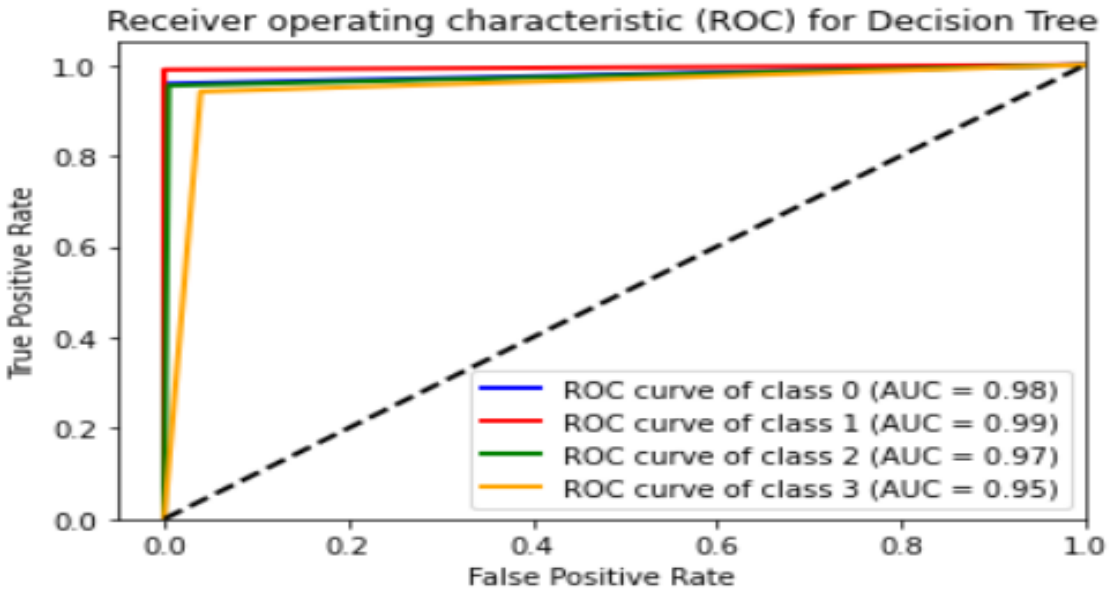

Figure 6.2: Receiver operating characteristic (ROC) curve for Decision Tree

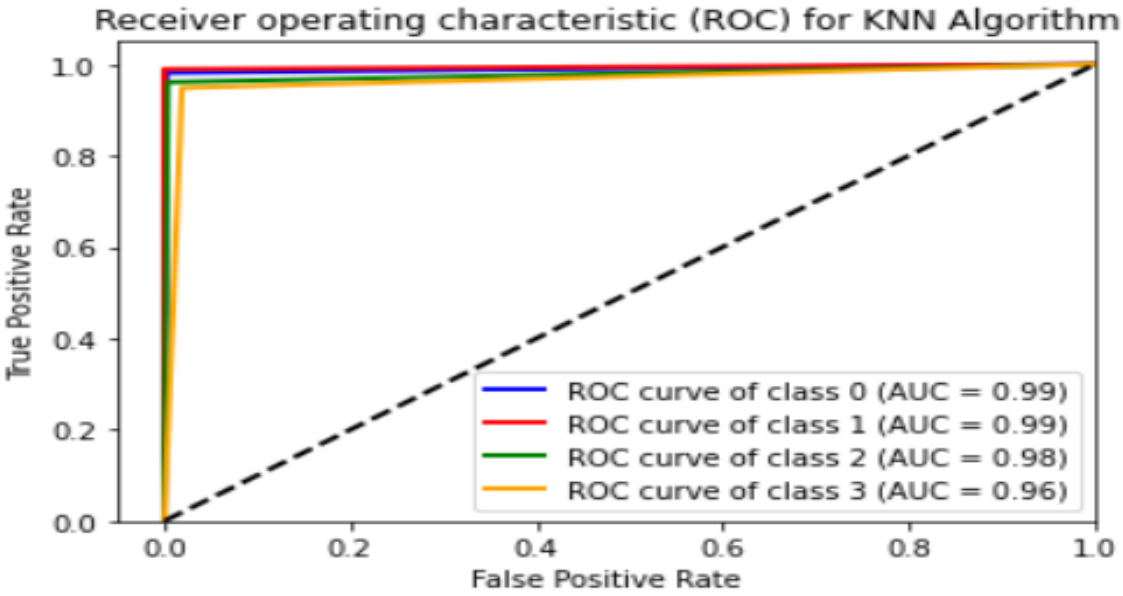

Figure 6.3: Receiver operating characteristic (ROC) curve for KNN

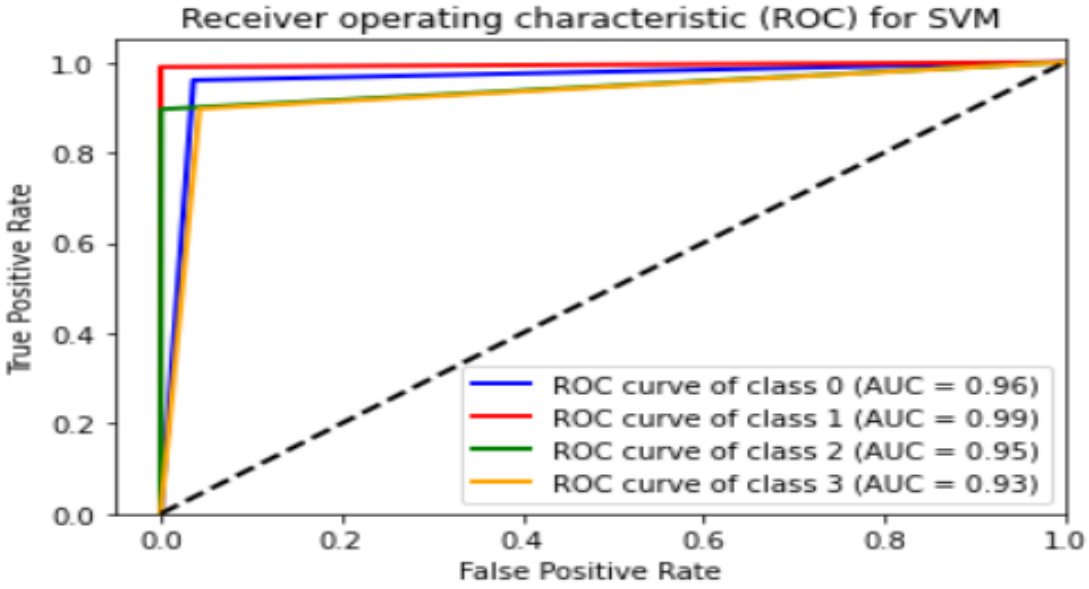

Figure 6.4: Receiver operating characteristic (ROC) curve for SVM

We can see that the KNN model has the highest accuracy (97.83%) whereas the Naive Bayes has the lowest accuracy (95.11%).

The reason why KNN worked so well is because KNN is a simple algorithm, based on the local minimum of the target function which is used to learn an unknown function of desired precision and accuracy. The algorithm also finds the neighborhood of an unknown input, its range or distance from it, and other parameters. It's based on the principle of "information gain" — the algorithm finds out which is most suitable to predict an unknown value.

Unfortunately, the Naive Bayes has the lowest accuracy. The cause behind this result is that this algorithm is a lousy estimator. Because it assumes that all the features are independent. But in our dataset, that wasn't the case.

Almost all of our models have shown a very high performing result. Because our dataset was well balanced. The SVM model has generalization in practice. Also, the risk of overfitting is less in SVM. The decision tree model forces the consideration of all possible outcomes of a decision and traces each path to a conclusion. It creates a comprehensive analysis of the consequences along each branch and identifies decision nodes that need further analysis. As such, it has performed well in our study. The final model was Logistic Regression. It can easily extend to multiple classes(multinomial regression) and a natural probabilistic view of class predictions. As a consequence, this model too has a high performance in this experiment.

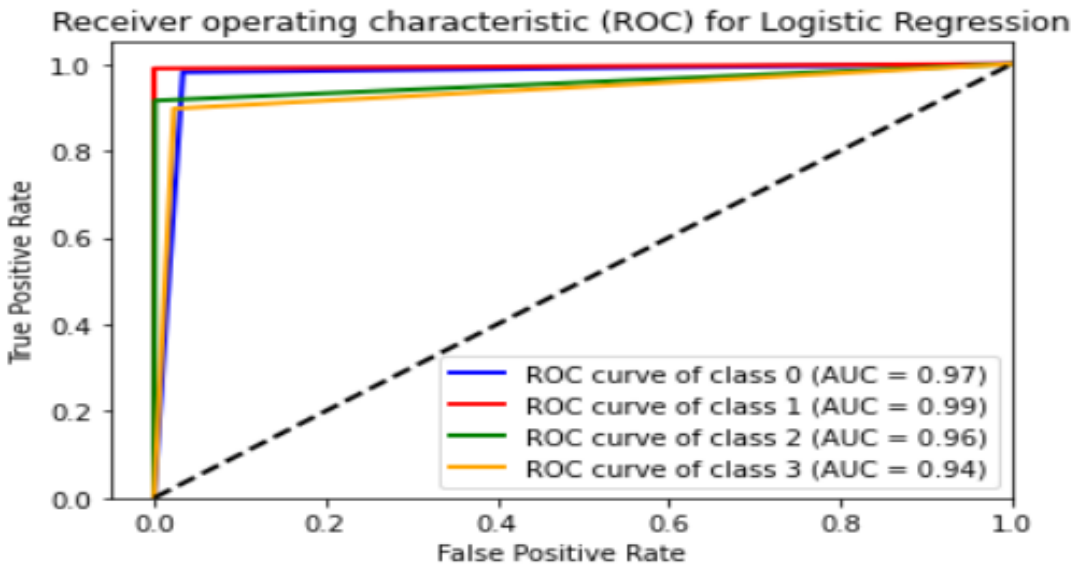

Figure 6.5: Receiver operating characteristic (ROC) curve for Logistic Regression

Chapter 7

Future Work and Conclusion

By using different machine learning algorithms, the number of occupants in a room can be correctly estimated. This results in the minimization of cost and consumption of different kinds of useful energy. Without the use of machine learning based estimation, the AC or light of a room will always be kept switched on. If it can be controlled according to the number of occupants in a room, it will result in a cost-minimized room ecosystem. So, in this study, we have built a machine learning based system where the number of occupants in a room can be estimated for environmental features. By using our model, the electronic devices can be controlled automatically by an Iot device without any human intervention. We have only used the traditional machine learning classifiers in this study. However, deep learning techniques can also be applied for finding the correct room occupancy estimation. Our proposed model is only applicable for occupancy estimation of up to four people. It can be extended for more people by making the model more robust. Though we have developed the model only for a typical room, it can be tested for other places like offices, schools, etc.

References

[1] Indoor Occupancy Detection and Estimation using Machine Learning and Measurements from an IoT LoRa-based Monitoring System Ramoni Adeogun, Ignacio Rodriguez, Mohammad Razzaghpour, Gilberto Berardinelli Per Hartmann Christensen and Preben Elgaard Mogensen†. Wireless Communication Networks Section, Department of Electronic Systems, Aalborg University, Denmark.

[2] Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models.Luis M. Candanedo, Véronique Feldheim.

[3] A Sensor-Utility-Network Method for Estimation of Occupancy Distribution in Buildings.Sean Meyn , Amit Surana, Yiqing Lin, Stella M. Oggianu, Satish Narayanan and Thomas A. Frewen.

[4] Occupancy prediction through machine learning and data fusion of environmental sensing and Wi-Fi sensing in buildings.Wei Wang, Jiayu Chen and Tianzhen Hong.

[5] Determining Room Occupancy with Machine Learning Techniques. Daniel Myhrman

[6] A Machine Learning Model for Occupancy Rates and Demand Forecasting in the Hospitality Industry. William Caicedo-Torres and Fabi´an Payares Department of Computer Science, Universidad Tecnol´ogica de Bol´ıvar, Parque Industrial y Tecnol´ogico Carlos V´elez Pombo,Km 1 V´ıa Turbaco, Cartagena, Colombia.

[7] “Room Occupancy Estimation Data Set.” https://archive.ics.uci.edu/ml/datasets/Room+Occupancy+Estimation. Accessed:2022-09-05.

Generated using Undegraduate Thesis LaTeX Template, Version 1.4. Department of Computer Science and Engineering, Ahsanullah University of Science and Technology, Dhaka, Bangladesh.

This project report was generated on Thursday 8th September, 2022 at 1:45am.