Room Occupancy Estimation

Project Report

Name: Kalla Praveen

Reg.no:12016581

Course code:INT354

Computer Science and Engineering

Lovely Professional University

Phagwara, Punjab

ABSTRACT

The amount of people using a space has a significant impact on how much energy it uses. It is possible to minimize costs by controlling energy use by accurately calculating the number of occupants. Two different energy sources that are very beneficial to individuals and should be used appropriately are temperature and light. For a variety of causes, these energies can occasionally be wasted. Individuals occasionally forget to turn off the lights or the air conditioner, wasting electricity. A room occupancy-based regulating system can be quite helpful in resolving these issues. Based on a variety of factors, machine learning algorithms can be used to accurately forecast the number of people in a room. In addition to light and temperature, other factors such as CO2, PIR activity, and noise can help determine how many people are in a given space. IoT devices can be used in conjunction with machine learning algorithms to autonomously control the air conditioner or lights. In this study, we used machine learning to infer the occupancy of a room using sensor readings. We gathered a number of occupancy estimation-related features. Then, we used the Support Vector Machine (SVM), K Nearest Neighbors (KNN), Logistic Regression, Decision Tree, and Nave Bayes machine learning techniques. Accuracy, precision, recognition, and f1 score were used to assess the classifiers' performance. The proposed approach can also be used for estimating occupancy of other locations such as offices, schools, etc. to reduce cost and energy consumption.

Contents

Chapter 1

Introduction

The waste of energy is a major problem worldwide, not only reducing energy but also increasing costs. Energy can be wasted for various reasons and from various sources. Sometimes energy is wasted intentionally or unintentionally by people. Wasting energy can have serious consequences. When the light is on for a long time, heat energy is generated which increases the temperature of the environment. Air conditioning produces various gases that are very harmful to the environment. So, the energy consumption of a place has to be kept in mind to keep the environment safe and healthy. Small rooms like a living room or dining room use energy differently depending on how many people are there. There are typically more lamps or AC units on when there are lots of people in the room. Hence, if you correctly calculate the population of a room, you can also predict how much energy is used. The energy usage in a room increases with the number of people present. When there are fewer individuals present, there is also a correspondingly lower energy consumption.

The total number of people in a room relies on a variety of variables. Several of these elements, including temperature, light, sound, CO2 levels, and PIR, are included in the current study. There will probably be more people in the room than normal, which will raise the temperature. People's body temperatures might be one reason for this. Moreover, smoking raises a room's temperature, which is only feasible when people are present. Moreover, cooking can raise a room's temperature. Once more, it is based on the number of individuals in a space. The number of individuals in a room also affects how much light is present. When there are more people in the room, the light is turned on more frequently as a result of the activities that are going on. They turn out the light after they leave. So, if there are not many people there, the room will be less bright. The number of individuals in a space has a significant impact on the volume as well. The level should be increased if there are many people present because they will likely be chatting, laughing, watching TV, and/or listening to music at the same time. A room is usually quiet when there are not many people in the room. The total movement also increases with the number of people, as movement is only possible when people are in a room. If the room is empty or there are

no people in the room, there is also no movement. Finally, room occupancy also depends on the CO2 concentration. The CO2 level tends to be higher when more people are present. Since people exhale CO2 when they breathe, the amount of CO2 in the room increases.

Calculating the population of a room can be made easier with the aid of machine learning (ML). The number of persons in a room can be accurately predicted by ML using the features mentioned above. A subset of artificial intelligence called machine learning (ML) finds patterns in the data that is available and predicts target features from a set of predictor features. To identify patterns in data, a model must be trained. The model uses the predictor features and information learned during training to predict the label when a new sample is received. We can avoid wasting energy by using machine learning and IoT devices to autonomously manage the lighting and climate in a space. An IoT device that incorporates ML can forecast the occupancy of a room based on the data at hand. Following that, the appliance operates the AC or lights on its own. The lighting or air conditioning is turned down if there are few people around. When there are a lot of people in a room, the lighting or AC are automatically turned up.

In this study, we employed ML algorithms to gauge a room's occupancy. We used the UCI machine learning repository's room occupancy estimation dataset. There are a total of 18 predictive features in the dataset. A dataset with 25 features was produced after we extracted a number of additional features. Based on this dataset, we made predictions about the number of people in a room depending on factors like light, temperature, sound, CO2 level, PIR, and more. We used the Support Vector Machine (SVM), K Nearest Neighbors (KNN), Logistic Regression, Decision Tree, and Naive Bayes machine learning methods. Accuracy, precision, recall, and f1 score were used to assess the classifiers' performance. The models' respective levels of accuracy are 95.26%, 97.83%, 97.09%, 95.85%, and 95.11%. By accurately estimating room occupancy, the AC or lighting can be set to the appropriate level, lowering expenses and energy usage. The findings show that the suggested method can also be used to calculate room occupancy in other locations, such as workplaces, schools, etc., in order to save money and energy.

Chapter 2

Literature Reviews

[1] The authors of this study presented the outcomes of applying machine learning to data from an IoT LoRa-based indoor environment monitoring system at Aalborg University, Denmark, to recognize the presence of humans and estimate the number of occupants in the office. They used a two-layer feed forward neural network on the data after identifying the issue as binary or multi-class categorization. IoT sensor ambient data and manually recorded door and window statuses make up the data needed to train, validate, and test the network. The results show that the classifier is able to correctly determine the occupancy of office spaces based on the IoT sensor measurements with an accuracy of up to 94.6% and 91.5% for the binary (presence or absence of people) and multiclass (no person, one person or two or more people)problems, respectively.

[2] With the help of various statistical classification models and the free, open-source R programme, the accuracy of estimating the occupancy of an office space using data from light, temperature, humidity, and CO2 sensors was examined in this study. In this study, three data sets were used: one for training, two for testing the models with the office door open and closed. In general, training the Linear Discriminant Analysis (LDA), Classification and Regression Trees (CART), and Random Forest (RF) models yields the best accuracy results (between 95% and 99%). The findings demonstrate that the accuracy forecast can be significantly impacted by the selection of characteristics and the use of an appropriate classification model. The information from the timestamp was incorporated into the models and usually increases the accuracy of detection. Interestingly, when only one predictor (temperature) was used, the LDA model was able to estimate the occupancy with 85% and 83% accuracy in the two test sets.

[3] The sensor utility network approach (SUN) for calculating building occupancy was introduced here by the authors. The SUN estimator generates occupancy estimates by resolving a convex optimization problem with a past horizon based on inputs from a large number of sensor

measurements as well as previous information on building occupancy. Existing online occupancy algorithms rely on erroneous and expensive indirect data, including CO2 readings or people counting sensors. Experiments conducted in an office building environment have tested the recently presented approach. It has been demonstrated that include all accessible data in the estimator greatly increases the estimation's accuracy. In particular, it is found that the average estimation error at the building level is reduced from 70% to 11% with the SUN estimator, compared to the conventional approach that relies solely on the measurement of occupancy density or flow of people.

[4] In this study, three common machine learning algorithms, including k-nearest neighbors (kNN), support vector machine (SVM) and artificial neural network (ANN), were compared with three data sources, including environmental data, Wi-Fi data and fused data, to optimize the performance of occupancy models in different scenarios. Three error measures, mean average error (MAE), mean percentage error (MAPE) and root mean square error (RMSE), were used to compare the performance of the models. The results show that the model based on ANN performs best on fused data, while the SVM model performs better on Wi-Fi data. The results also indicate that the fused dataset does not necessarily improve model accuracy compared to independent data sources, but has better robustness in occupancy prediction. The MAPE value for the models were 44.4%, 36.6%.

[5 In this work, passive infrared sensors (PIR) and booking information were used to train models that assessed occupancy. A system with two artificial neural network models was developed, and it predicts the occupancy status of rooms, or whether they are empty or not, using a binary classification model. A subsequent regression model estimates the number of inhabitants in a room if it is thought to be inhabited. Several neural network regression models were created by the author, and they were trained using manually gathered data from actual assemblies. The models' performance was then compared, and a bidirectional design with long-term memory produced the greatest outcomes. It has mean absolute error of 0.94 and mean square error of 2.27.It predicts the number of people with a margin of error of one person with 85% and 49% accuracy for residents from 1 to 7 and 8 to 14, respectively. With a margin of error of two, the results for the same intervals are 94% and 66%accurate,respectively.

Chapter 3

Background Study

3.1 K Nearest Neighbors

The K-nearest neighbor 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 KNN classifier works by taking a test instance, finding the k closest 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 makes no assumptions about the underlying data and is therefore very versatile. This algorithm first calculates the distance between the new data point and all training data points. It then determines the number of nearest neighbors (usually using Euclidean distance) and assigns the new data point the majority class. One of the main advantages of the k nearest neighbors (KNN) classifier is that it is very easy to implement and can be applied to a wide range of datasets. In addition, the classifier is insensitive to outliers and resistant to overfitting.

3.2 Logistic Regression

Logistic regression is a classification technique used to predict the probability of a categorical dependent variable. The dependent variable in logistic regression is binary in nature and has data values in the form of 0 and 1. In simple words, the dependent variable in logistic regression follows the Bernoulli distribution. A logistic regression classifier predicts the probability of an event occurring by fitting the 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, is more robust and 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. In addition, logistic regression is not affected by multicollinearity and can be used to predict both categorical and numerical data.

3.3 Naive Bayes

A Naive Bayes classifier is a simple probabilistic classifier based on the application of Bayes theorem with strong (naive) independence assumptions. A more descriptive name for the independence assumption would be "unconditional independence". Simply put, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of another feature. For example, a fruit can be considered an apple if it is red, round and has a diameter of about 3 inches. Even though these features depend on each other or on the presence of the other features, all of these features independently contribute to the probability that this fruit is an apple, which is why it is called 'naive'.Naive Bayes classifiers are highly scalable and require a number of parameters that is linear to 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 an 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.

Chapter 4

Methodology

5.1 Dealing with NULL Values

There are 10,129 samples overall inside this data set. But, if some of these samples include null values, there will be an issue. And therefore, we began by searching for such samples. We noticed that not a single dataset's characteristics have null values. As a result, there was no reason to take any samples out of the dataset.

5.2 Feature Extraction

Using the features of the original dataset, we extracted some fresh features. Date, Time, Day of the Week, Hour of the Day, Mean Temperature, Mean Light, Mean Sound, Footsteps Sound, PIR Level, MA CO2, and MV CO2 were listed.

By joining the Date and Time features of the original dataset, Date Time was derived. To distinguish between weekdays and weekends, the term "day of the week" was introduced.

To determine the most people, the hour of day formula was used.

By obtaining the average data from the temperature sensors, mean temperature was calculated.

The average values of the four light sensors were obtained by calculating mean light.

The mean of the sound sensors was used to determine mean sound. This sensor came in 4 different variations.

The average PIR sensor value and the average sound sensor value were combined to create sound.PIR Level was derived to evaluate the degree of crowding by observing how individuals moved about the space.To determine the average CO2 level over the last consecutive days, MA CO2 was determined. However, MV CO2 was calculated from the variation in CO2 level over the preceding days.

5.3 Data Visualization

To represent our generated dataset graphically, we have displayed some of our data. This aids in providing fresh insight into the data. We have displayed the temperature, CO2 levels, and impact of footsteps at various times during the day.

5.4 Oversampling

Very few examples of the minority class exist in unbalanced classification for a model to successfully learn the decision boundary. The minority class's examples can be oversampled as one approach to resolving this issue. Simple replication of samples from the minority class in the training dataset before model fitting can do this. Although it can balance the class distribution, this doesn't give the model any new data. Synthesizing fresh minority-class examples is an improvement over using duplicate minority-class examples. This kind of data augmentation for tabular data can be extremely successful. The Synthetic Minority Oversampling Technique, or SMOTE, is the method that is most frequently employed to create new samples. When using SMOTE, examples that are near together in the feature space are chosen, a line is drawn between them, and a new sample is drawn at a location along the line. To be more precise, a random representative from the minority class is initially picked. Then, for that example, k of the closest neighbors are located (usually, k=5). A synthetic example is produced at a randomly chosen position in feature space between the two instances and a randomly determined neighbor is then picked. As many synthetic examples of the minority class as needed can be produced using this approach.

The strategy works because it generates convincing new synthetic examples from the minority class that are substantially near in feature space to already existing examples from the minority class.

Due to the extreme imbalance in our initial dataset, we employed this method in our work. In order to provide far better forecasts, we have included more instances of minority classes to our dataset.

To be more precise, a random representative from the minority class is initially picked.

A synthetic example is produced at a randomly chosen position in feature space between the two instances and a randomly determined neighbor is then picked. As many synthetic examples of the minority class as needed can be produced using this approach.

5.5 Separating Features and Labels

All of the columns have been divided into two pieces. The label has been applied to the final column, often known as the target column. On the other hand, this research has employed all of the previous columns as prediction features.

5.6 Dataset Normalization

Normalization aims to scale down features to a similar scale. This enhances the model's functionality and training stability. Scaling to a range, clipping, log scaling, and z-score are four popular normalising methods that could be helpful. In our work, we have applied the scaling to a range technique. This procedure has been used for all the features. Using the following straightforward formula to scale to a range, one can change floating-point feature values from their natural range (for instance, 100 to 900) into a standard range, which is typically 0 and 1 (or occasionally -1 to +1):

$$X' = (X- X\ min)/(X\ max -X\ min)$$

We made a good decision to scale to a range since the following criteria were satisfied: With few or no outliers, we were able to determine the approximate upper and lower boundaries on our data. Over that range, our data are rather well dispersed.

5.7 Splitting the Dataset

The full datasets were divided into two halves. We distributed 80% of the data in the train section, and the remaining 20% went into the test portion.

5.8 Models

Five different models were employed. The Naive Bayes Classifier, the Decision Tree, the K-nearest Neighbor, the Support Vector Machine, and the Logistic Regression are some of these.
Gaussian Naive Bayes was utilized for the Normal Distribution in the Naive Bayes Classifier.
Max depth in the Decision Tree was set to 3. It was done to make it clear that the tree's maximum depth will be three levels. We have utilized the standard kernel type of "rbf" in SVM.

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%
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:

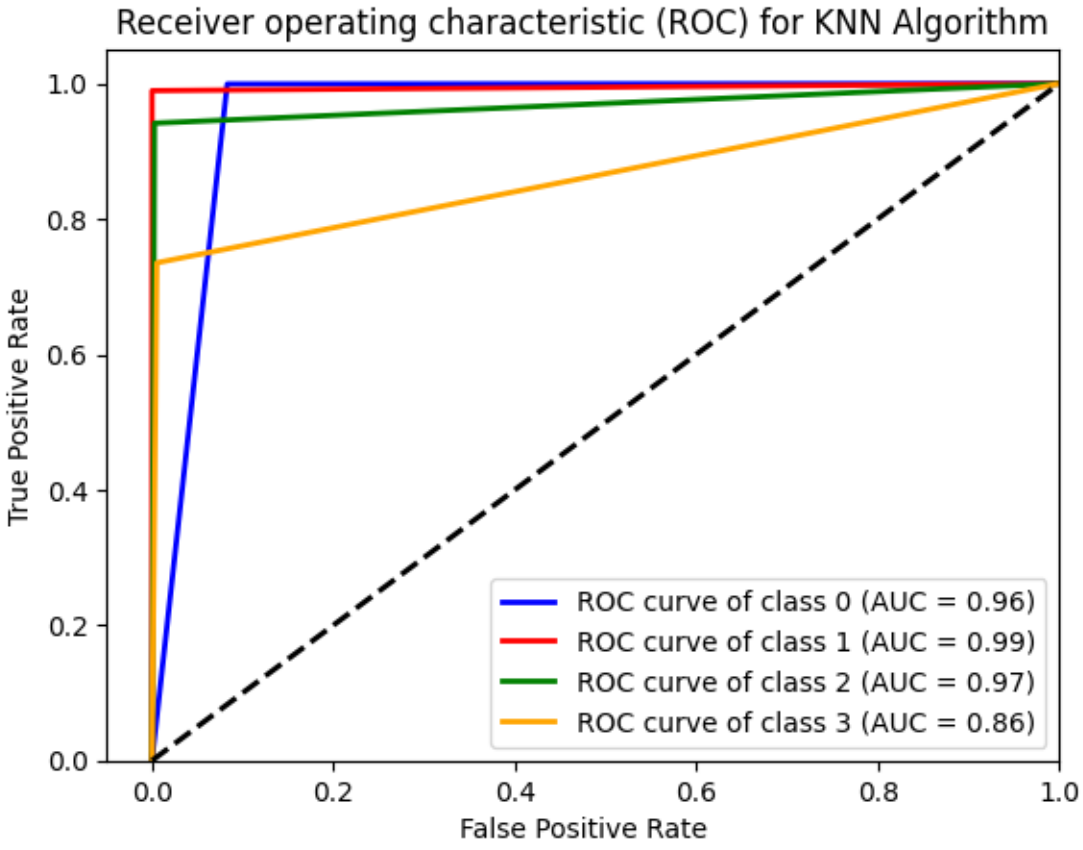

Figure 6.1: Receiver operating characteristic (ROC) curve for KNN Algorithm

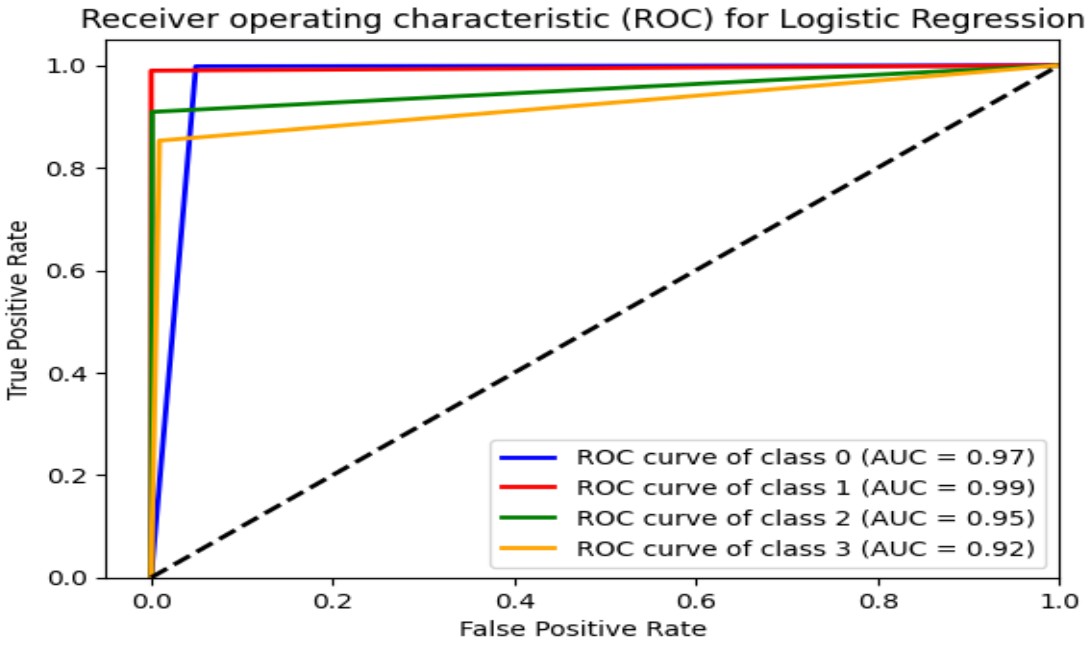

Figure 6.2: Receiver operating characteristic (ROC) for logistic regression

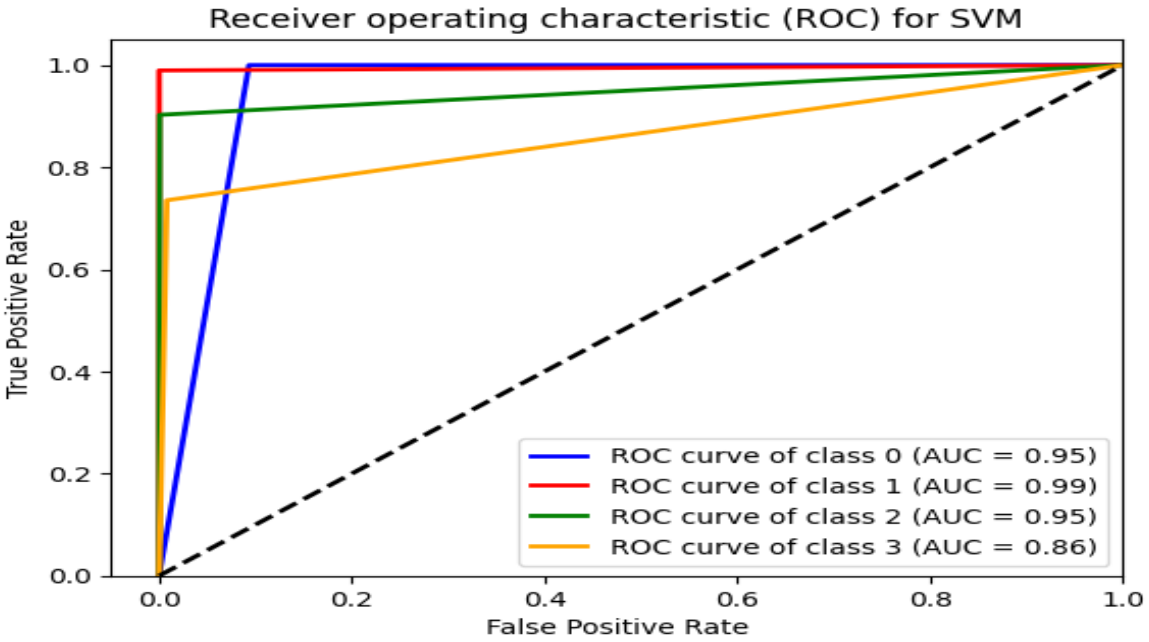

Figure 6.3: Receiver operating characteristic (ROC) curve for SVM

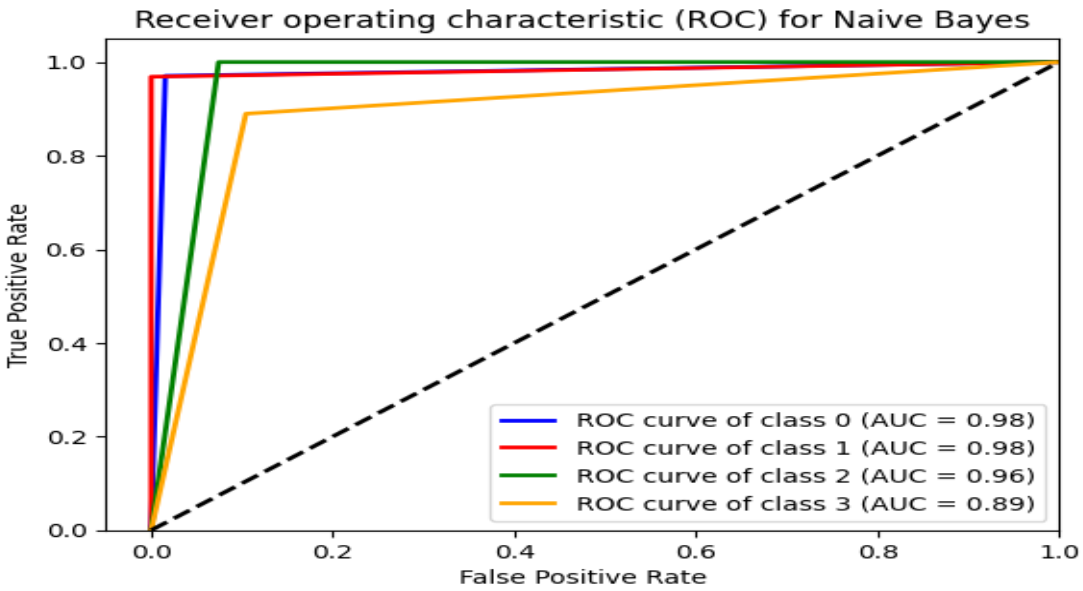

Figure 6.4: Receiver operating characteristic (ROC) curve for Naïve Bayes

As can be seen, the Naive Bayes model has the lowest accuracy (95.11%), while the KNN model has the highest accuracy (97.83%).

KNN is a straightforward algorithm that uses the local minimum of the target function to learn an unknown function with the appropriate precision and accuracy, which explains why it performs so well. The algorithm also determines a parameter's range or distance from an unknown input as well as its surroundings. According to the "information gain" theory, the algorithm determines which method is best for predicting an unknown value.

Regrettably, the accuracy of the Naive Bayes is the lowest. This algorithm's poor performance as an estimator is what led to the outcome. It does this by assuming that each feature is independent. Yet, that wasn't the case in our sample.

Almost all of our models have produced results that are extremely high performing. due to the balanced nature of our dataset. The SVM model is practical and offers generalisation. Moreover, SVM has a lower overfitting risk. The decision tree model follows each path to a conclusion and forces the examination of all potential decision outcomes. It generates a thorough analysis of the outcomes along each branch and pinpoints decision points that require additional research. As a result, it did well in our investigation. Logistic regression was the chosen model in the end. Several classes (multinomial regression) and a natural probabilistic perspective of class predictions can be added with ease.

.

Chapter 7

Future Work and Conclusion

The number of people in a room can be accurately calculated by utilizing various machine learning methods. The cost and use of various forms of useful energy are minimized as a result. Without the usage of machine learning-based estimation, a room's air conditioner and light will constantly be on. The cost of the room environment will be reduced if it can be adjusted in accordance with the number of occupants. As a result, in this study, we developed a machine learning-based system that can estimate the number of people in a room based on environmental variables. By utilizing our methodology, an IoT device can operate the electronic gadgets automatically without the need for a person to be there In this investigation, we have simply employed conventional machine learning classifiers. Deep learning methods can also be used to determine the accurate room occupancy estimation, though. Only up to four persons can be estimated to be occupied using our proposed methodology. By strengthening the model, it may be expanded to include more users. The model can be tested for different locations, such as offices, schools, etc., even though it was only designed for a standard room.

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.

GITHUB LINK

https://github.com/Praveen353/Machine-Learning-Project-12016581-