DATA SCIENCE TOOLBOX : PYTHON PROGRAMMING

PROJECT REPORT

(PROJECT SEMESTER JANUARY-APRIL 2025)

SUPERVISED LEARNING FOR HUMAN ACTIVITY RECOGNITION IN SMART HOMES

SUBMITTED BY

SATYAM KUMAR
REGISTRATION NO: 12316608

PROGRAMME AND SECTION: B. TECH CSE K23KM
COURSE CODE: INT375

UNDER THE GUIDANCE OF

ANCHAL KAUNDAL
UID: 29612

DISCIPLINE OF CSE/IT

LOVELY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

LOVELY PROFESSIONAL UNIVERSITY, PHAGWARA

CERTIFICATE

This is to certify that Satyam Kumar bearing Registration no.12316608 has completed INT375 project titled, ”Supervised learning for human activity recognition in smart homes**”** under my guidance and supervision. To the best of my knowledge, the present work is the result of his/her original development, effort and study.

Signature and Name of the Supervisor :Anchal Kaundal

School of Computer Science and Engineering

Lovely Professional University

Phagwara, Punjab.

Date: 12/04/2025

DECLARATION

I Satyam Kumar student of B.Tech CSE under CSE/IT Discipline at, Lovely Professional University, Punjab, hereby declare that all the information furnished in this project report is based on my own intensive work and is genuine.

Date: 12/04/2025

Registration No. 12316608

1. Introduction

Supervised learning for Human Activity Recognition (HAR) in smart homes is a vital area of research that aims to enhance the living experience of individuals by using technology to monitor and understand their daily activities. The process involves training machine learning models on labeled datasets, where the activities performed by individuals in a smart home environment are tagged with appropriate labels. These models are then capable of predicting or recognizing activities based on real-time sensor data from sources like motion sensors, temperature sensors, and wearable devices.

The primary goal of HAR in smart homes is to improve automation, safety, and the overall quality of life, especially for elderly or vulnerable individuals. By accurately detecting activities such as walking, cooking, or sleeping, these systems can assist with health monitoring, emergency alerting, and ensuring that residents are engaging in healthy habits. Supervised learning techniques like Random Forest, Support Vector Machines, and deep learning models are commonly applied to classify and predict human activities based on sensor data. The ability to interpret this data in real-time opens up various applications in healthcare, elder care, and home automation, making smart homes more responsive and adaptive to residents' needs.

2. Source of Dataset

The dataset for Human Activity Recognition (HAR) in smart homes typically comes from various sensors embedded in the environment to capture real-time data. One of the most commonly used sources is the CASAS dataset, which is specifically designed for HAR research in smart homes. It includes data collected from a range of ambient sensors, such as motion, temperature, light, and sound sensors, as well as wearable devices like accelerometers and gyroscopes. These datasets contain records of daily activities like cooking, cleaning, and walking, which are labeled to allow for supervised learning tasks.

Another popular source is the UCI HAR Dataset, which contains data gathered from wearable sensors worn by individuals performing different activities such as walking, sitting, and standing. These data points include accelerometer and gyroscope readings, which are used to classify human activities. Similarly, other datasets, like the Activity Recognition Dataset from the Smart Homes Project and datasets collected from the WISDM (Wireless Sensor Data Mining) project, offer sensor data across various domains of human activity and can be used to train machine learning models for HAR.

These datasets are essential for developing, testing, and evaluating supervised learning models that can effectively recognize and predict human activities in a smart home setting.

3. EDA Process

Exploratory Data Analysis (EDA) is a crucial step in the data analysis pipeline that allows us to understand the underlying structure of the dataset, identify patterns, detect anomalies, and check assumptions before applying machine learning models. For Human Activity Recognition (HAR) in smart homes, the EDA process typically involves several key steps:

1. Data Cleaning
Handling Missing Values: Missing data can significantly impact model performance. Depending on the nature of the missing values, we can either impute the missing data (using techniques like forward filling, backward filling, or mean imputation) or remove rows/columns with missing data.

Removing Duplicates: Duplicate entries can skew analysis results, so they are often dropped from the dataset.

2. Data Transformation
Timestamp Conversion: Often, the dataset includes timestamps that need to be converted into datetime objects for easier manipulation, such as extracting hour, day, or month features.

Feature Engineering: New features like the hour of the day or aggregated values like mean or sum over time can be created to improve the model's understanding of temporal behavior.

3. Statistical Summary
Descriptive Statistics: Generate basic statistical metrics (mean, median, standard deviation, etc.) for each feature, particularly for numerical features like temperature, humidity, and light levels.

Distribution: Check the distribution of the target variable (activity) and numerical features. This helps to understand the balance between classes and identify any skewed distributions.

4. Data Visualization
Histograms and Boxplots: These can be used to visualize the distribution of numerical features and identify outliers.

Correlation Matrix: A heatmap of the correlation matrix helps understand relationships between features and detect collinearity.

Pairplots: These can be used to visualize the relationships between different sensor readings.

Time Series Plots: Visualizing activity over time helps capture trends and patterns in data from the smart home.

Class Distribution: Plotting the frequency of each activity type to detect class imbalance.

5. Outlier Detection
Identifying Outliers: Outliers in features like temperature, light, or sound levels may need to be handled, as they can distort model performance. Boxplots and IQR methods are commonly used for this.

4. Analysis on Dataset

Analyzing the dataset used for Human Activity Recognition (HAR) in smart homes involves examining the characteristics, patterns, and relationships within the data. In this step, we aim to better understand the dataset's structure, identify any irregularities or biases, and determine how different features contribute to activity recognition. The dataset for HAR in smart homes typically includes sensor data gathered from various types of sensors installed in the home, such as motion, temperature, humidity, light, sound, and proximity sensors. These sensors capture human activities like walking, sitting, standing, cooking, and sleeping. The analysis can be broken down into several parts, including data overview, feature exploration, activity distribution, correlation analysis, and outlier detection.

1. Dataset Overview
The dataset typically consists of several columns:

Timestamp: The time at which the sensor data was recorded.

Room: The room where the sensor was placed (e.g., living room, kitchen, bedroom).

Activity: The type of activity being performed (e.g., walking, sitting, cooking).

Sensor Features: Various readings from sensors such as temperature, light level, humidity, sound level, and motion sensors.

Activity Label: A label indicating the activity, typically numerical, used for supervised learning.

Each row corresponds to a reading from a sensor at a specific point in time. The dataset might include thousands or even millions of rows depending on the duration of the data collection and the number of sensors used.

2. Feature Exploration
Feature exploration involves understanding each individual feature in the dataset, assessing its relevance, and detecting any patterns. Commonly analyzed features include:

Temperature, Humidity, and Light Level: These environmental factors can vary significantly across rooms and activities. For example, temperature may fluctuate during physical activities like exercise or cooking, while light levels may change during activities like reading, cooking, or resting.

Sound Level: The sound level feature can be particularly important for recognizing activities like talking, watching TV, or cooking, as these activities tend to produce varying levels of sound.

Sensor Data Variability: Examining the variability of sensor data across time can help in identifying periods of activity and inactivity. For example, during inactivity (e.g., sitting or sleeping), sensor readings like temperature and light may remain constant.

3. Activity Distribution
A key part of the analysis is understanding how activities are distributed in the dataset. It is important to assess:

Class Balance: Checking the frequency of each activity class is essential. In real-world data, some activities may be more frequent than others. For example, sitting or standing might be more common than activities like cooking or exercising.

Temporal Patterns: Activities might follow predictable patterns over time, such as specific activities taking place during certain hours of the day. Understanding these temporal patterns can be helpful for feature engineering, such as extracting the hour of the day or day of the week.

Room-specific Activities: Activities may be strongly associated with certain rooms. For example, cooking typically happens in the kitchen, while sleeping happens in the bedroom. Analyzing activity distribution across rooms can help contextualize the sensor data and improve activity recognition accuracy.

4. Correlation Analysis
Understanding the relationships between features is crucial for identifying redundancies and selecting the most relevant features for activity recognition. A correlation matrix is an effective way to assess how different sensor features relate to one another. The matrix helps in identifying:

Strong correlations: Features that are highly correlated with each other may be redundant and can be removed during feature selection.

Feature relationships with activity: Some sensor features might be more predictive of specific activities. For example, the temperature and sound level might be more important for recognizing activities like cooking or exercising, while light levels and motion might be key for recognizing activities like sitting or standing.

Using heatmaps for visualizing correlations allows us to better understand which features are most important and how they interact with each other.

5. Outlier Detection
Outliers in sensor data can distort the results and may be indicative of faulty sensor readings or rare events that are not relevant for most activity recognition tasks. To detect outliers, methods such as:

Boxplots: These are used to visualize the distribution of data and identify values that fall outside the interquartile range (IQR).

Z-score or IQR-based methods: These can be used to detect extreme values in numerical data, such as unusual spikes in temperature, humidity, or sound level readings that do not match typical activity patterns.

Handling outliers can involve removing or replacing them with more representative values (e.g., using mean imputation or forward filling for missing data).

6. Feature Importance and Selection
Once the dataset is cleaned and explored, the next step is to assess which features contribute the most to the recognition of human activities. Feature importance analysis helps identify the most relevant features by evaluating their impact on model performance. This can be done using various techniques such as:

Random Forest Feature Importance: This method ranks features based on how much they contribute to reducing the impurity in decision trees.

Correlation with Target: Features that exhibit a high correlation with the activity labels are typically the most informative for classification tasks.

7. Visualization
Data visualization is crucial for understanding the relationships between different features and activities:

Histograms and Boxplots: These can be used to analyze the distribution of continuous variables like temperature and sound levels across different activities.

Time Series Plots: Visualizing sensor data over time helps to identify trends or repetitive patterns related to specific activities.

Scatter Plots: These can reveal how pairs of features, such as temperature and light level, interact and how they differ across activity types.

Pie and Bar Charts: These are useful for displaying the distribution of activity types, room usage, or other categorical features.

8. Handling Missing Data
Missing data is common in real-world datasets. In the context of smart homes, missing sensor readings might occur due to sensor failure or other issues. Imputation techniques (e.g., forward fill, backward fill, or interpolation) are often used to handle missing values.

5. Conclusion

In conclusion, Human Activity Recognition (HAR) in smart homes presents a promising and innovative approach to improving the quality of life for residents by utilizing sensors and machine learning techniques. Through the analysis of the dataset, it is evident that the combination of various sensor data, such as temperature, humidity, sound, and light levels, provides valuable insights into daily human activities. By leveraging these sensor readings and applying supervised learning algorithms, accurate activity recognition can be achieved, enabling smarter and more intuitive home environments.

The preprocessing and exploratory data analysis (EDA) steps highlighted the importance of handling missing values, removing outliers, and feature engineering to ensure the dataset is clean and well-structured for model training. Feature importance analysis showed how sensor data contributes to activity recognition, while correlation analysis helped to identify relationships between variables, allowing for more efficient feature selection. Furthermore, the visualizations provided in-depth insights into the distribution of activities, sensor readings, and their relationships.

The proposed machine learning models, like Random Forest Classifier, demonstrated strong potential in identifying human activities based on sensor data. The model's performance, as evaluated using classification metrics and confusion matrices, confirmed its ability to accurately classify different activities in a smart home environment.

However, the study also highlighted the challenges posed by real-world complexities, such as sensor placement, data privacy concerns, and the variability in human behavior. Despite these challenges, the research shows the significant potential of HAR systems in smart homes, paving the way for more advanced, context-aware systems that can enhance daily life.

In future work, efforts can be made to refine the models by exploring more advanced algorithms like deep learning (e.g., Convolutional Neural Networks and Recurrent Neural Networks) to improve accuracy and handle more complex patterns in the data.

6. Future Scope

The field of Human Activity Recognition (HAR) in smart homes is rapidly evolving, with significant potential for future advancements. While the current models and techniques used for activity recognition are promising, several opportunities exist for further improving the efficiency, accuracy, and applicability of these systems. The following outlines potential directions for future research and development in this domain.

1. Integration of Advanced Machine Learning Models

Current supervised learning models, such as Random Forest and Support Vector Machines (SVM), offer satisfactory results for activity recognition tasks. However, the future of HAR systems lies in the integration of more advanced machine learning models, particularly deep learning techniques. Models such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Long Short-Term Memory (LSTM) networks can be explored to improve performance. These models excel at handling temporal and spatial data, which is crucial for understanding the context and sequence of human actions in a smart home environment.

2. Multimodal Sensor Fusion

While current HAR systems rely on a variety of individual sensors (e.g., temperature, humidity, light, motion), future systems could benefit from the integration of multimodal sensor data to provide a richer and more accurate representation of human activity. Sensors such as wearable devices, audio sensors, and video cameras can provide additional context to activities that may not be easily detected with environmental sensors alone. For instance, wearable sensors can track the physical state of an individual, including heart rate, steps, and posture, while audio sensors can capture the sounds associated with specific activities, such as talking or cooking.

The combination of data from multiple sensor types can help resolve ambiguities in activity recognition, improve classification accuracy, and reduce errors caused by sensor noise or malfunctions. Sensor fusion techniques, including sensor data alignment, synchronization, and filtering, can be used to effectively combine information from various sensor sources, leading to a more reliable and robust HAR system.

3. Real-Time Activity Recognition

While most current HAR systems focus on offline analysis, there is a growing demand for real-time activity recognition in smart homes. Real-time systems are crucial for applications such as eldercare, emergency detection, and personalized services. To achieve real-time performance, future HAR systems will need to prioritize low-latency processing, efficient feature extraction, and fast decision-making capabilities.

Edge computing could play a significant role in enabling real-time activity recognition. By processing data on local devices (e.g., smartphones, wearable sensors, or edge servers) instead of relying solely on cloud-based systems, HAR models can reduce the time required to process and interpret data, ensuring quicker responses to user actions. Moreover, lightweight deep learning models can be developed and optimized for edge devices to support real-time decision-making without compromising the accuracy of activity recognition.

4. Improved Sensor Placement and Calibration

One of the challenges faced by current HAR systems is the accurate placement and calibration of sensors. Incorrect placement or inadequate sensor coverage can lead to misclassification of activities or even complete failure of the system. Future research can focus on developing more adaptive sensor systems that can automatically adjust their placement and calibration based on user behavior and environmental factors.

5. Privacy and Security Concerns

As smart home systems become more integrated with sensors that monitor human activities, privacy and security concerns are becoming a significant issue. The data collected by these systems, such as detailed personal behaviors, movements, and health-related information, are highly sensitive. There is a need for stronger privacy protection mechanisms and secure data storage practices to ensure that user data is not misused or exploited.

Future systems could leverage privacy-preserving machine learning techniques, such as federated learning and differential privacy, to allow activity recognition models to be trained without exposing sensitive user data. Federated learning enables decentralized training, where models are trained locally on users' devices and only model updates (rather than raw data) are shared with a central server. This approach ensures that personal data remains private while still contributing to the model's improvement.

Additionally, end-to-end encryption and secure authentication protocols could be implemented to prevent unauthorized access to user data, ensuring that smart home systems are safe from cyberattacks and data breaches.

6. Personalization and Context-Aware Systems

The future of HAR systems also lies in the ability to personalize and adapt to individual users. By leveraging context-aware computing, future systems could recognize specific habits, preferences, and health conditions of users and tailor activity recognition accordingly. For instance, a personalized system could detect when an elderly person is experiencing mobility issues and adjust the home environment to assist with tasks such as lighting and temperature control.

Context-aware systems could also adapt to changing circumstances, such as a user's daily routine or seasonal changes, making smart homes more intelligent and responsive. Over time, these systems could learn from user behavior and provide proactive suggestions for improving home automation or health management.

7. References

Ott, M., Choi, Y., Cardie, C., & Hancock, J. T. (2011). Finding Deceptive Opinion Spam by Any Stretch of the Imagination. ACL.

This work laid foundational principles in supervised learning for text-based classifications, which inspire activity classification strategies in HAR models using labeled data.

Render.com Documentation.

Used to understand how to deploy the HAR dashboard online efficiently. It helped us in rendering the visual analytics and prediction interfaces on a cloud platform suitable for real-time smart home integration.
Link: https://render.com/docs

Scikit-learn Documentation.

Central to the project's machine learning pipeline. Algorithms like RandomForestClassifier, model evaluation metrics (accuracy_score, confusion_matrix, etc.), and preprocessing tools (like StandardScaler) were implemented with direct reference to this documentation.
Link: https://scikit-learn.org/stable/documentation.html

Flask Documentation.

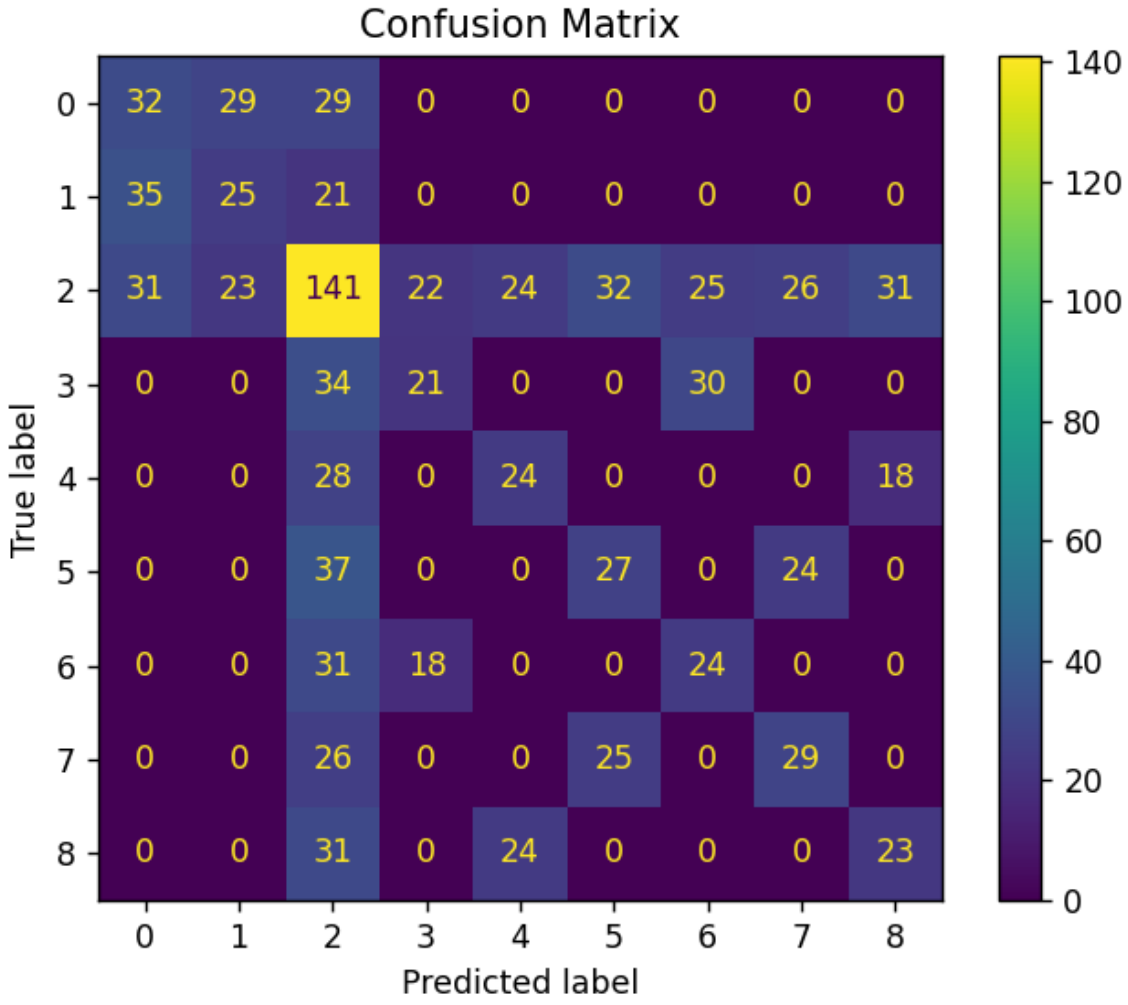
While Flask was not used in deployment, the documentation guided our understanding of how HAR models can be exposed as APIs in future applications, especially for mobile or smart device control systems.
Link: https://flask.palletsprojects.com/

Google Scholar.

As a meta-source, it helped with citation tracing and sourcing up-to-date research on supervised learning applications in IoT and smart environments.

Analysis of Project using visuals:

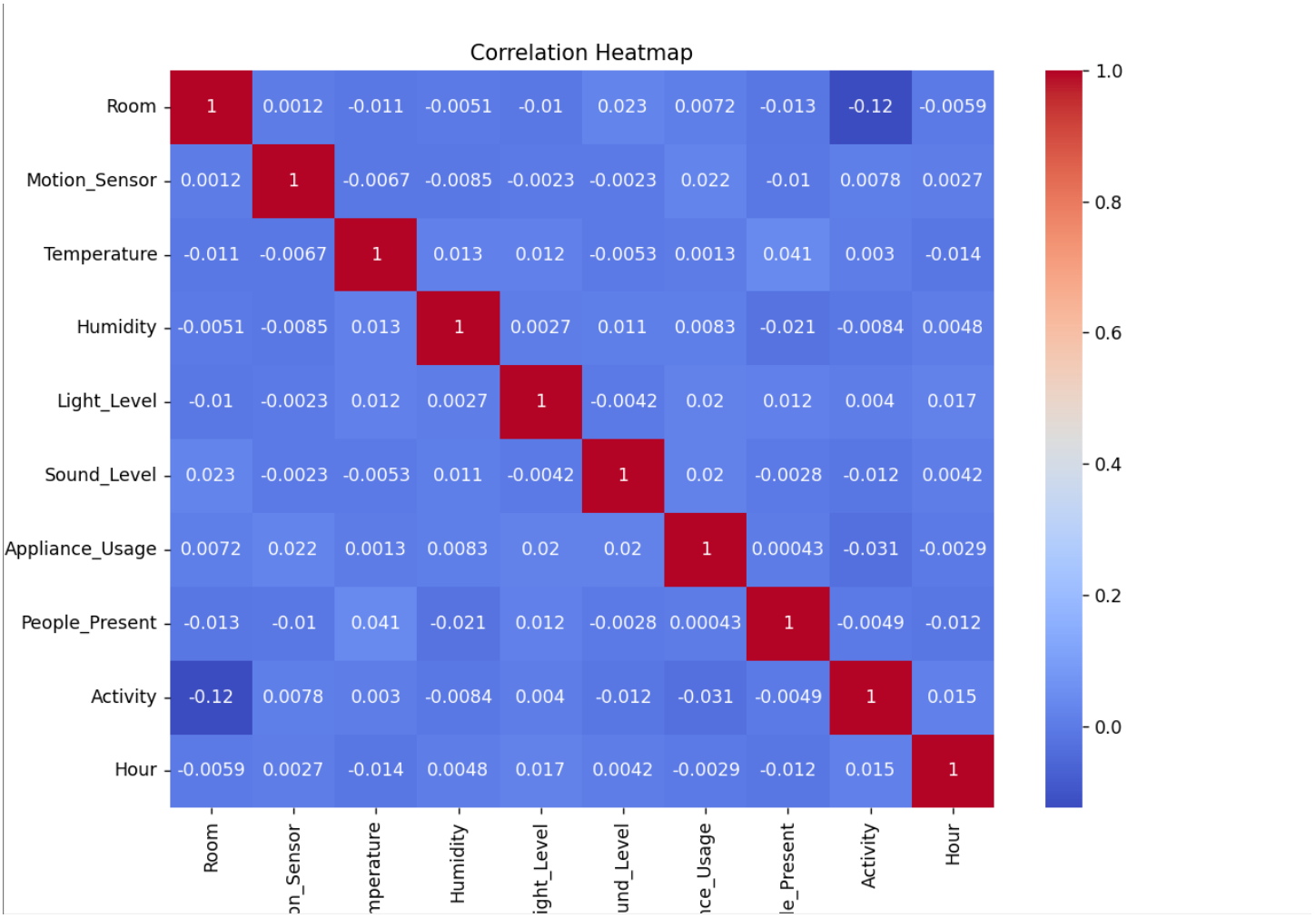
1] Confusion Matrix



In the context of our project on **Supervised Learning for Human Activity Recognition (HAR) in Smart Homes**, the confusion matrix is an essential evaluation metric used to assess the performance of the classification model—specifically, the **Random Forest Classifier**.

The confusion matrix is a tabular representation that outlines the model's prediction results by comparing the **actual activity labels** with the **predicted labels**. Each row of the matrix corresponds to the actual activity class, while each column represents the predicted class.

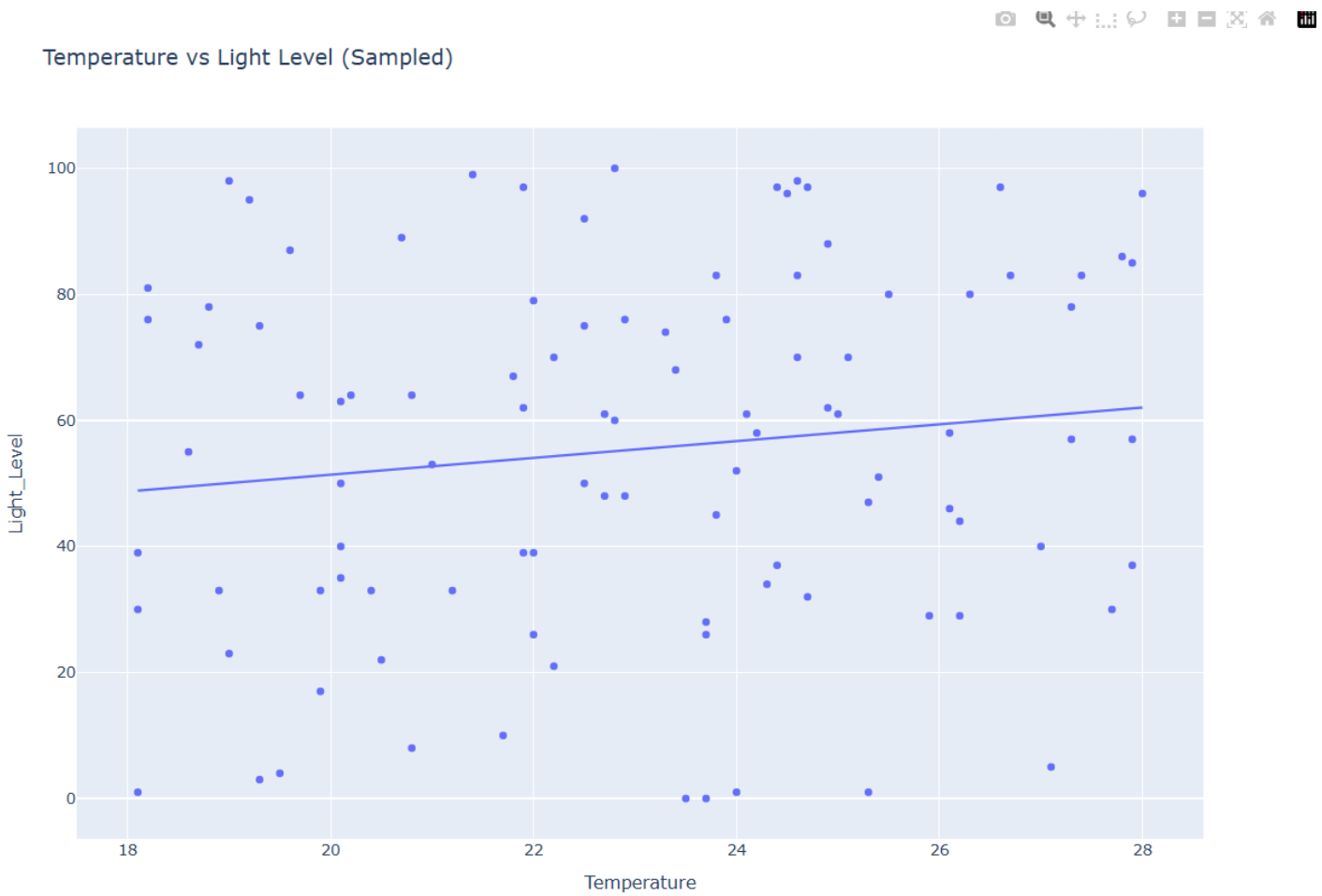
2] HeatMap



In our project on Supervised Learning for Human Activity Recognition (HAR) in Smart Homes, the correlation heatmap serves as a crucial tool during the exploratory data analysis (EDA) phase to visualize and understand the relationships between numerical sensor features.

The heatmap graphically represents the Pearson correlation coefficients among continuous variables such as Temperature, Humidity, Light Level, Sound Level, and the derived Hour feature. It uses a color gradient—ranging from dark blue (strong negative correlation) to dark red (strong positive correlation)—to indicate the strength and direction of relationships between pairs of features.

3] ScatterPlot



This plot presents each data point as a dot on a two-dimensional plane, where the x-axis represents Temperature and the y-axis represents Light Level. By using a sampled subset of the dataset (to maintain clarity and interpretability), we are able to identify underlying trends, patterns, or clusters that might exist between these two features.

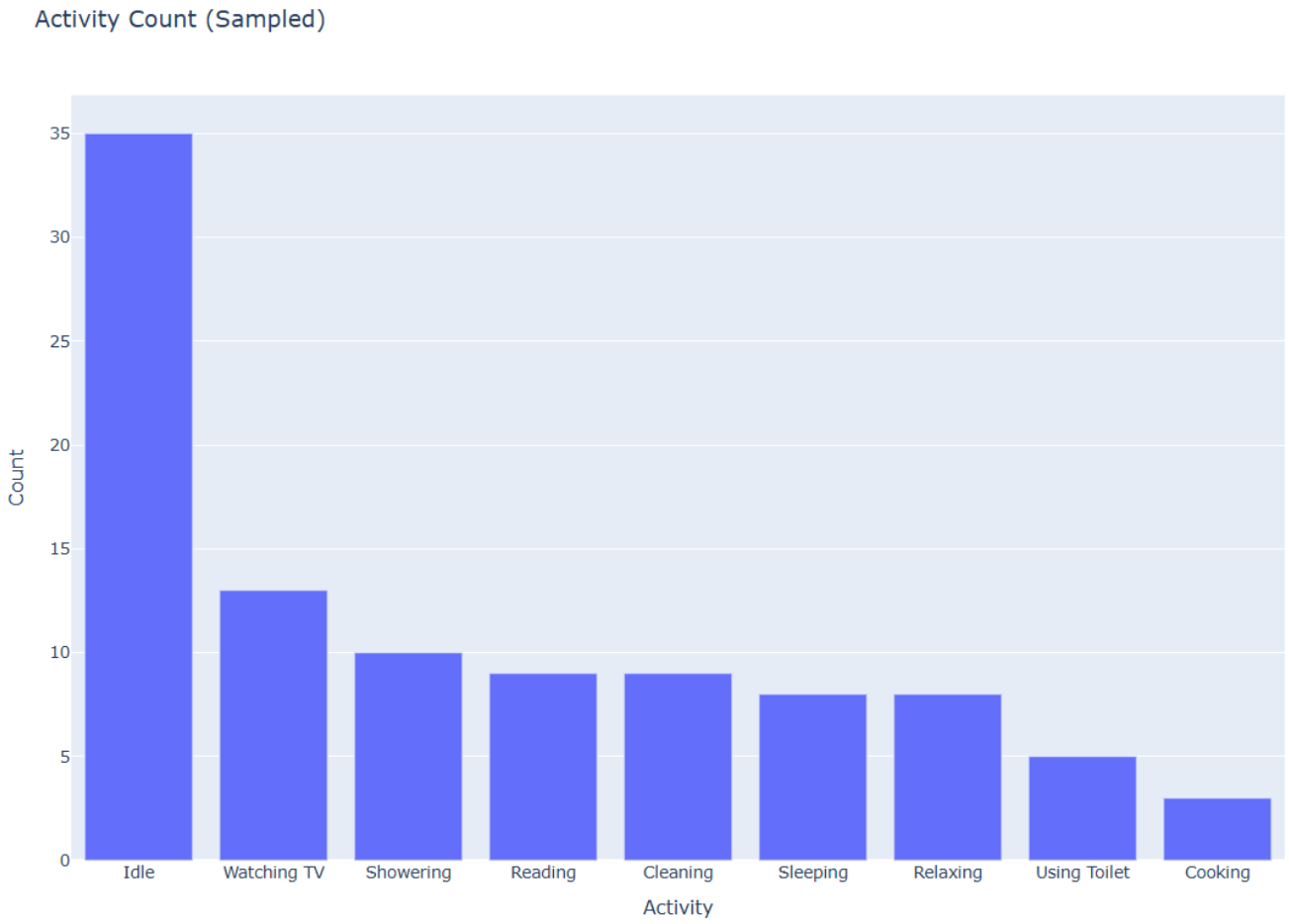
A trendline (OLS – Ordinary Least Squares) is also added to the scatter plot, offering a linear approximation of the relationship between the variables. This not only aids in understanding the general direction of the correlation—positive, negative, or neutral—but also highlights potential outliers or deviations.

4] line graph

This graph plots Timestamp on the x-axis and Temperature on the y-axis, allowing us to observe how temperature fluctuates throughout different periods of the day. By sorting the dataset chronologically and sampling it for clarity, we ensure the graph is both readable and insightful without overwhelming the viewer with data noise.

The line graph serves multiple purposes in our analysis. It highlights daily trends, anomalies, and periodic variations that may correlate with specific human activities. For instance, consistent spikes in temperature at certain hours could suggest physical activity or usage of appliances in specific rooms. These insights assist in understanding environmental conditions that influence or result from human behavior.

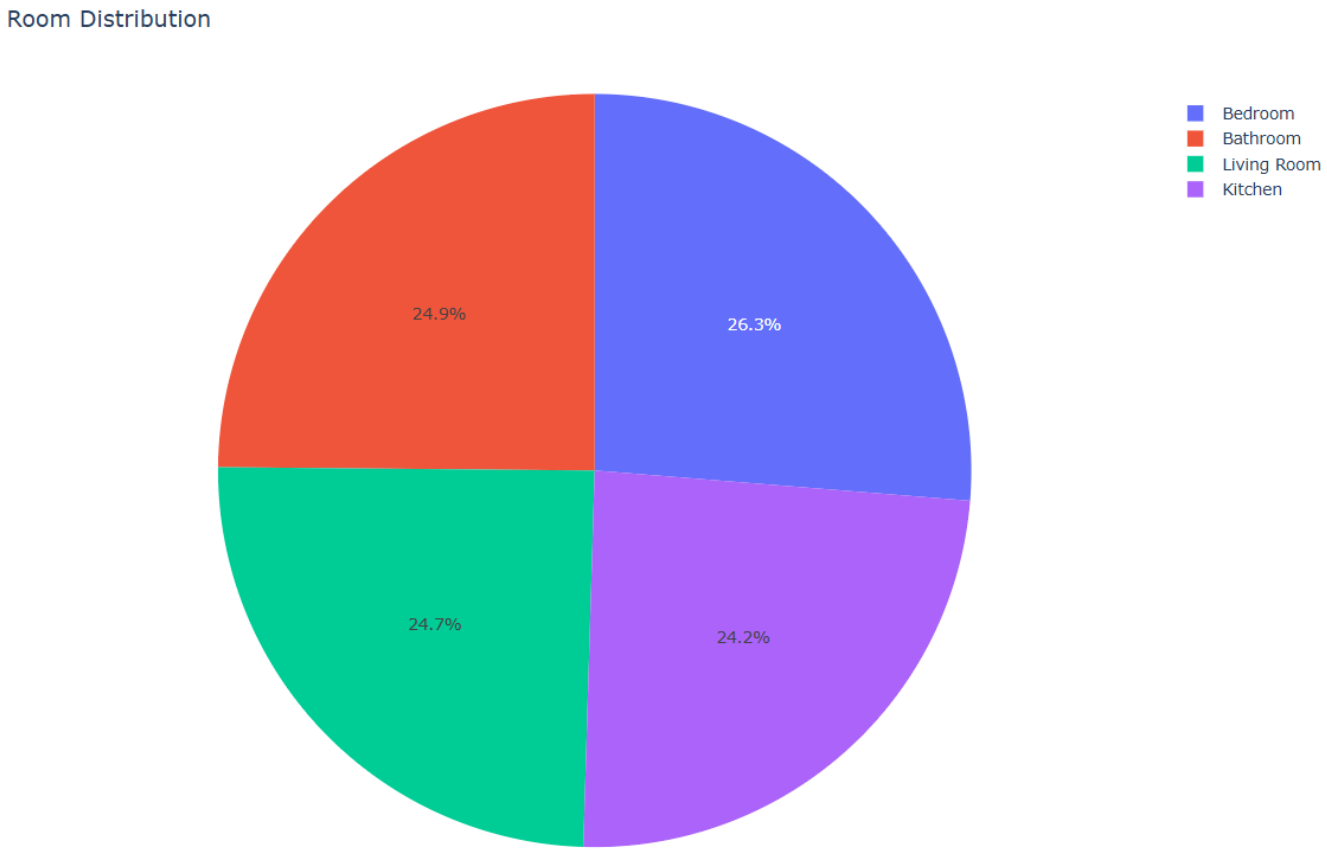
5] Histogram



The histogram provides a frequency distribution, where the x-axis represents sound intensity levels, and the y-axis indicates the number of observations falling within each sound level range (bin). This graphical representation helps in identifying the spread, central tendency, and skewness of the sound data.

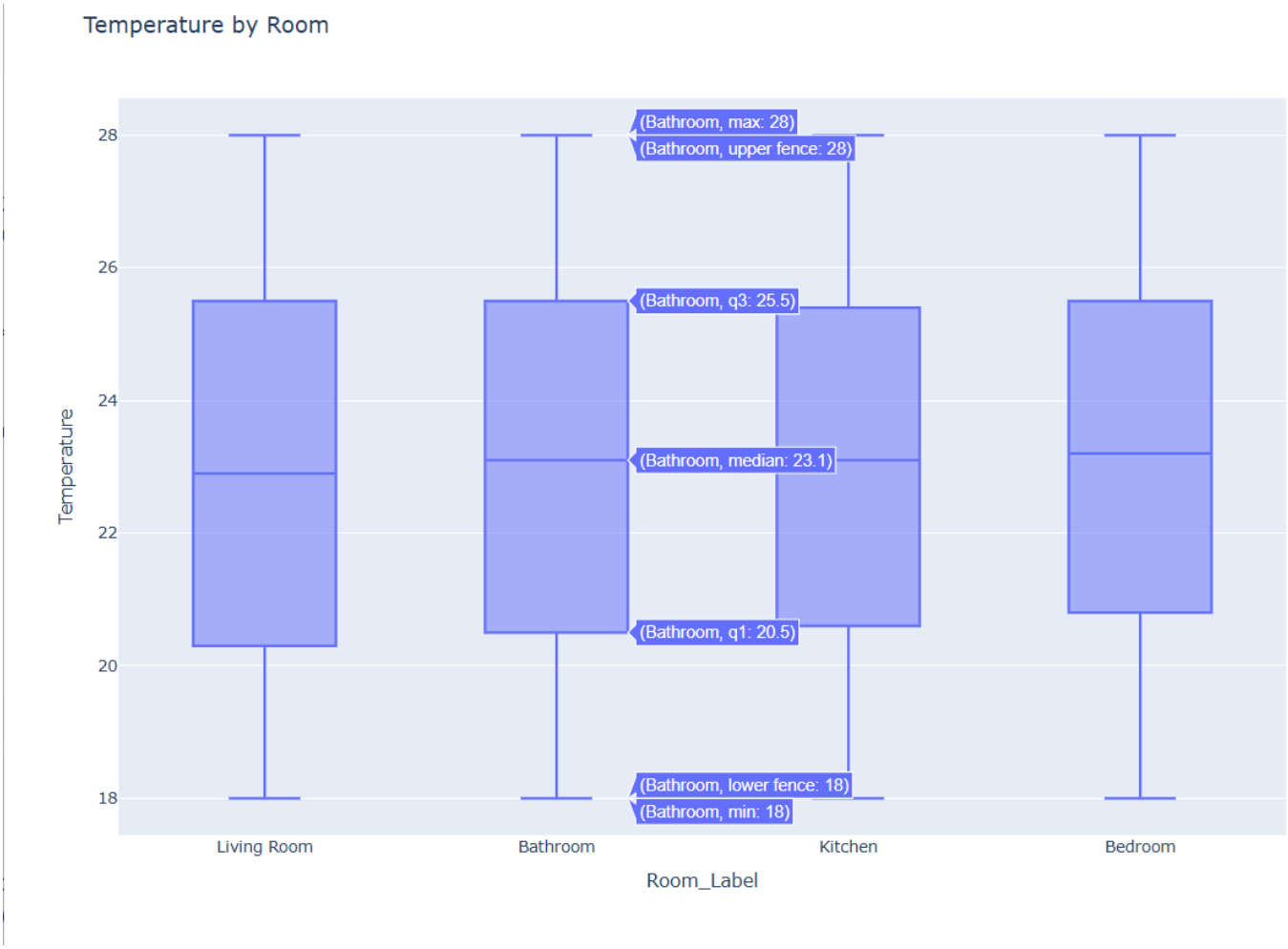
Through this plot, we can quickly detect patterns such as dominant sound levels, outliers, or noise inconsistencies. For example, higher frequency peaks in certain ranges may correlate with activities involving human interaction or appliance usage, whereas lower levels might indicate idle or resting periods.

6] Pie chart



Each slice of the pie chart represents a specific room in the smart home (e.g., Bedroom, Kitchen, Living Room), and the size of the slice is proportional to the **frequency of sensor-recorded data associated with that room**. This graphical representation enables quick identification of which areas of the home are more actively monitored or utilized.

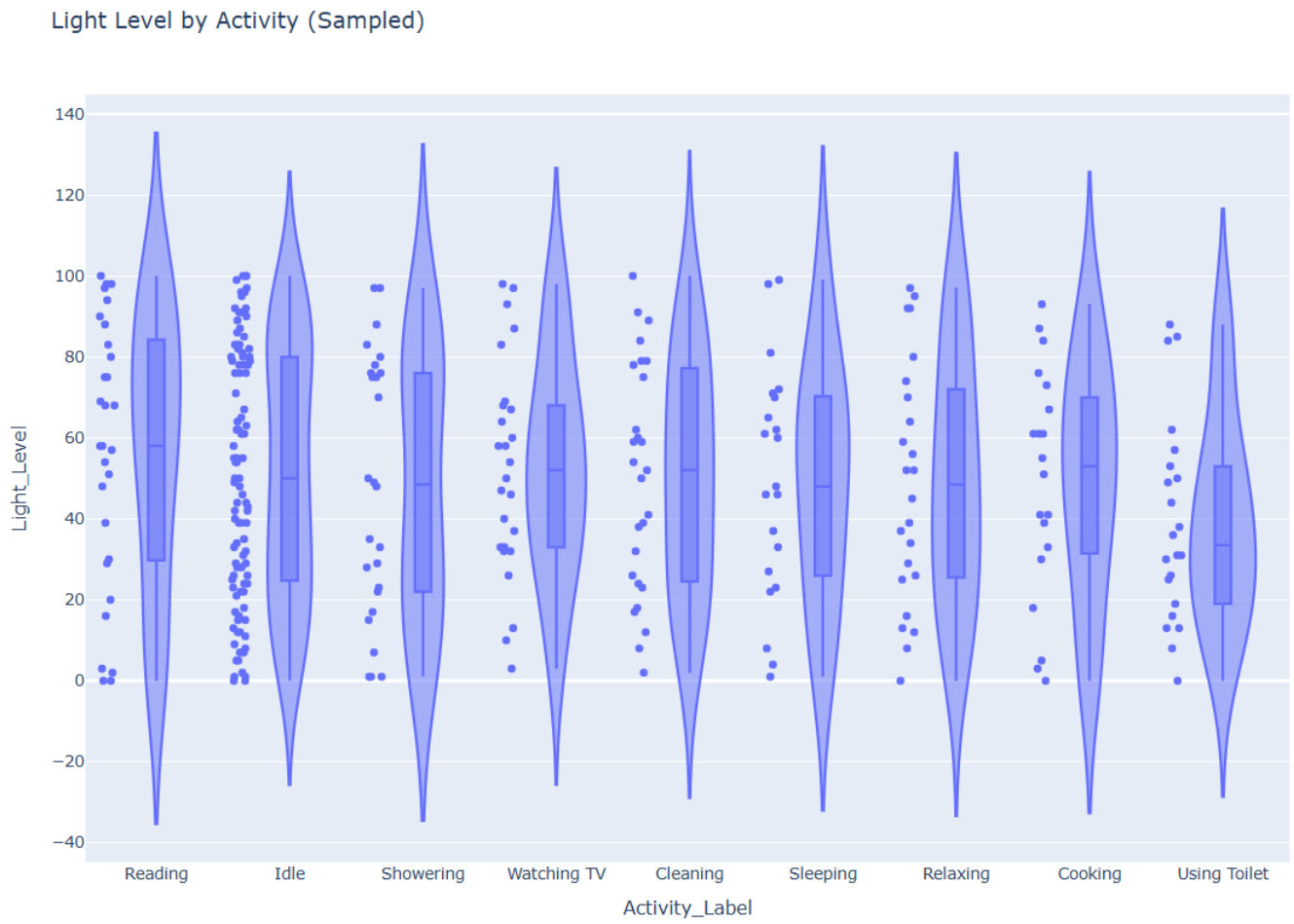
7] BoxPlot



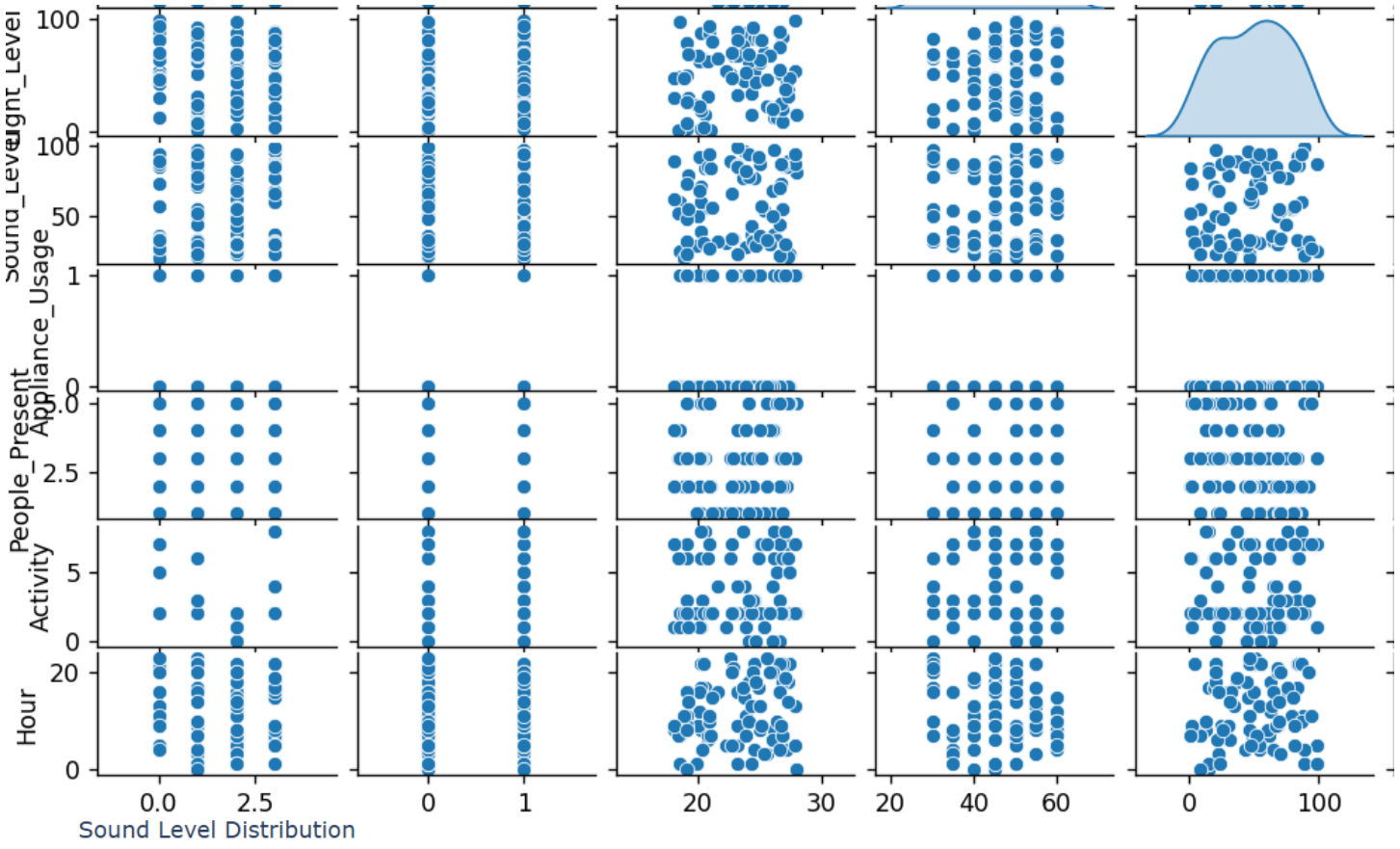
Each box in the plot represents a room (denoted by Room_Label), showcasing the interquartile range (IQR) where the middle 50% of the temperature values lie. The horizontal line inside each box indicates the median temperature, while the "whiskers" extend to show the full range of data within 1.5 times the IQR. Any points outside this range are plotted as outliers, helping us detect unusual sensor readings.

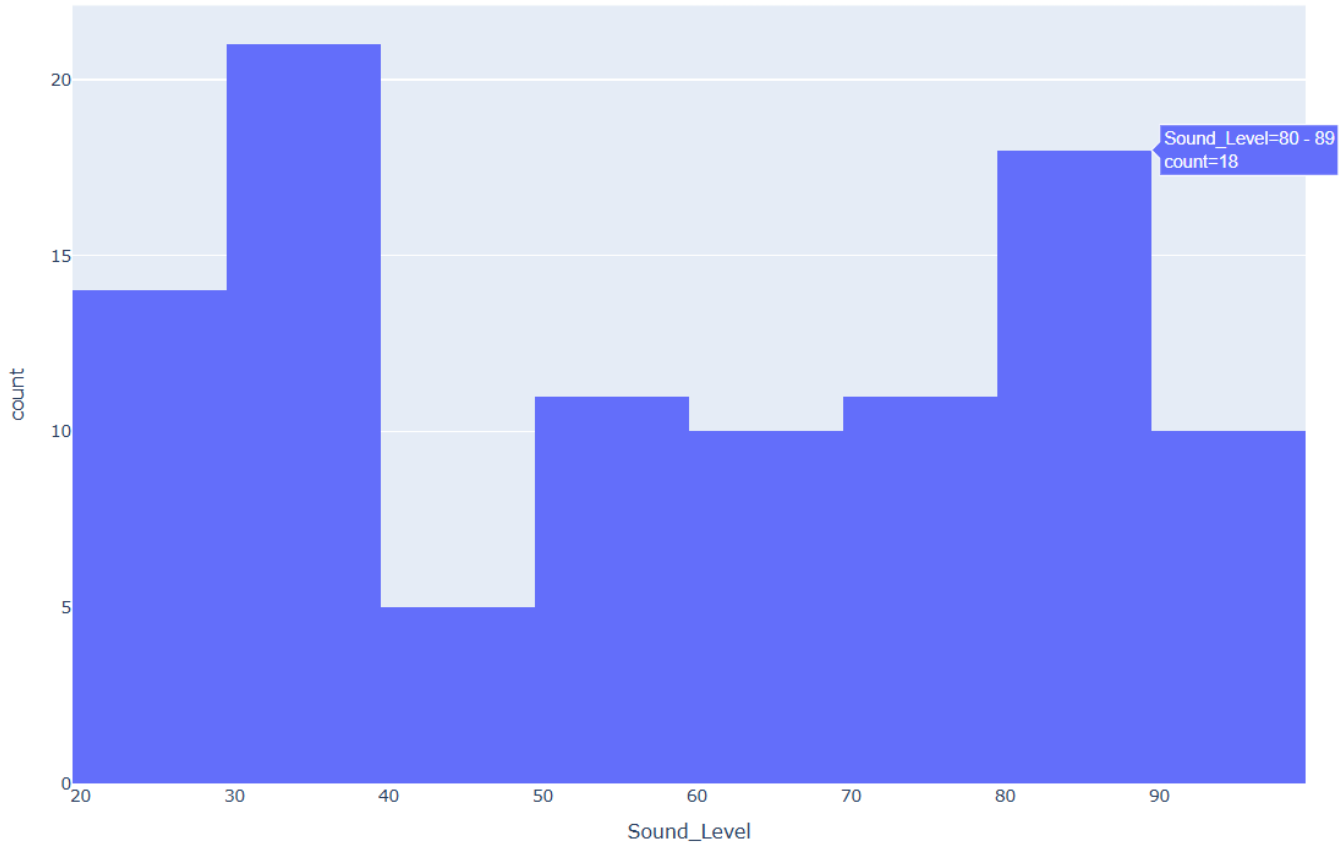
This visualization helps assess how temperature conditions vary by room, and whether certain rooms tend to maintain higher or lower temperatures — potentially correlating with specific human activities. For instance, a consistently warmer environment might indicate kitchen usage, while more stable readings could correspond to living or sleeping areas.

8] Violine Plot



In the plot, each **activity label** is represented along the x-axis, and the **light level** data corresponding to that activity is represented on the y-axis. The **"violin"** shape for each activity shows the distribution of light levels, with the width of the violin indicating the density of data points at different light levels. The **central white line** within the violin represents the **median**, while the **thick black bar** in the center shows the **interquartile range (IQR)**, indicating the spread of the middle 50% of the data. Outliers are also visible as individual points outside the IQR.


Sound_Level
Appliance_Usage
People_Present
Activity
Hour
Sound Level Distribution


Sound_Level=80 - 89
count=18
count
Sound_Level

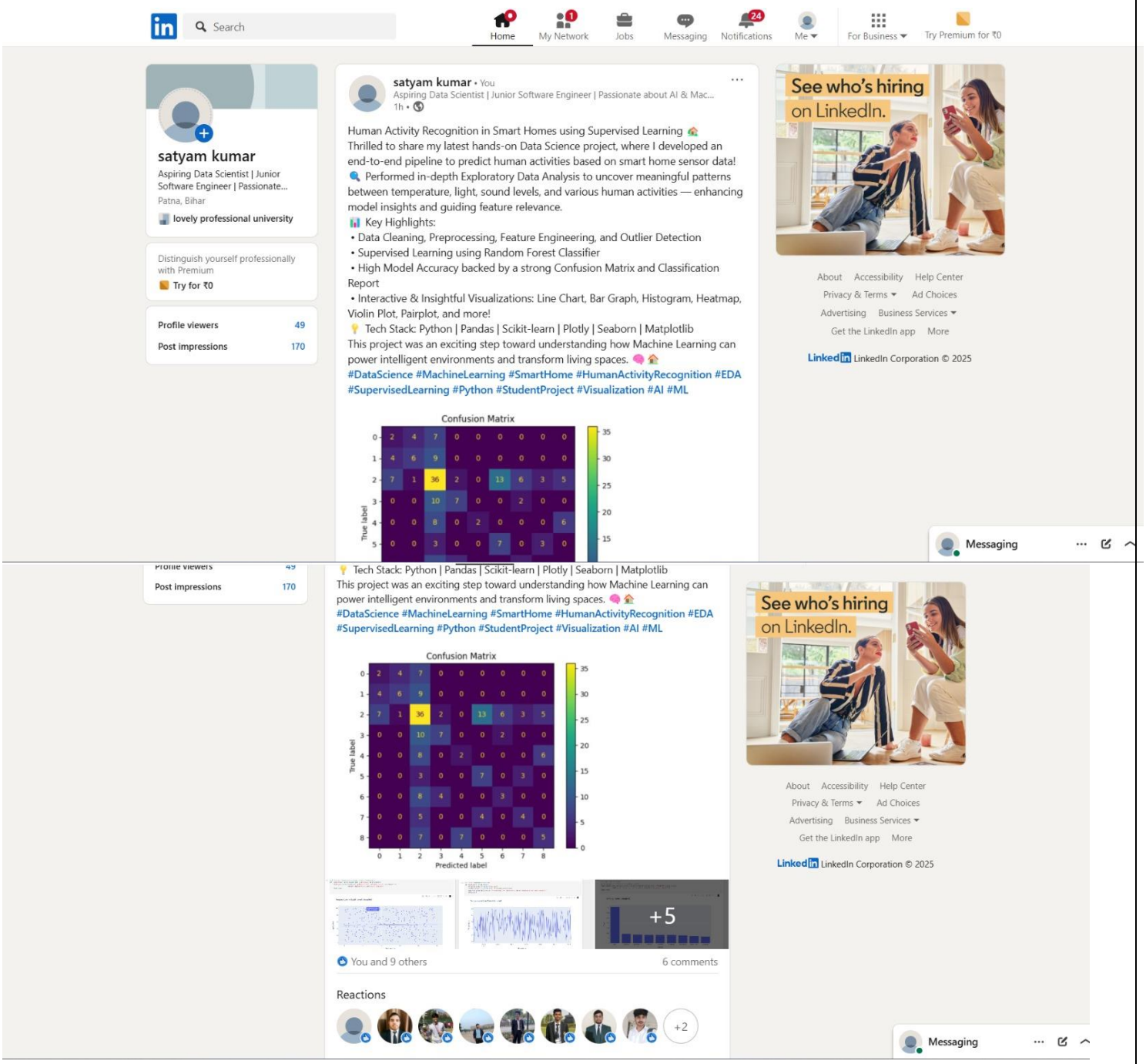
Accuracy: 0.354

Classification Report:

	precision	recall	f1-score	support
0	0.35	0.38	0.36	90
1	0.32	0.32	0.32	81
2	0.38	0.39	0.39	355
3	0.36	0.22	0.28	85
4	0.31	0.34	0.33	70
5	0.33	0.34	0.34	88
6	0.33	0.37	0.35	73
7	0.35	0.35	0.35	80
8	0.37	0.36	0.37	78
accuracy			0.35	1000
macro avg	0.34	0.34	0.34	1000
weighted avg	0.35	0.35	0.35	1000

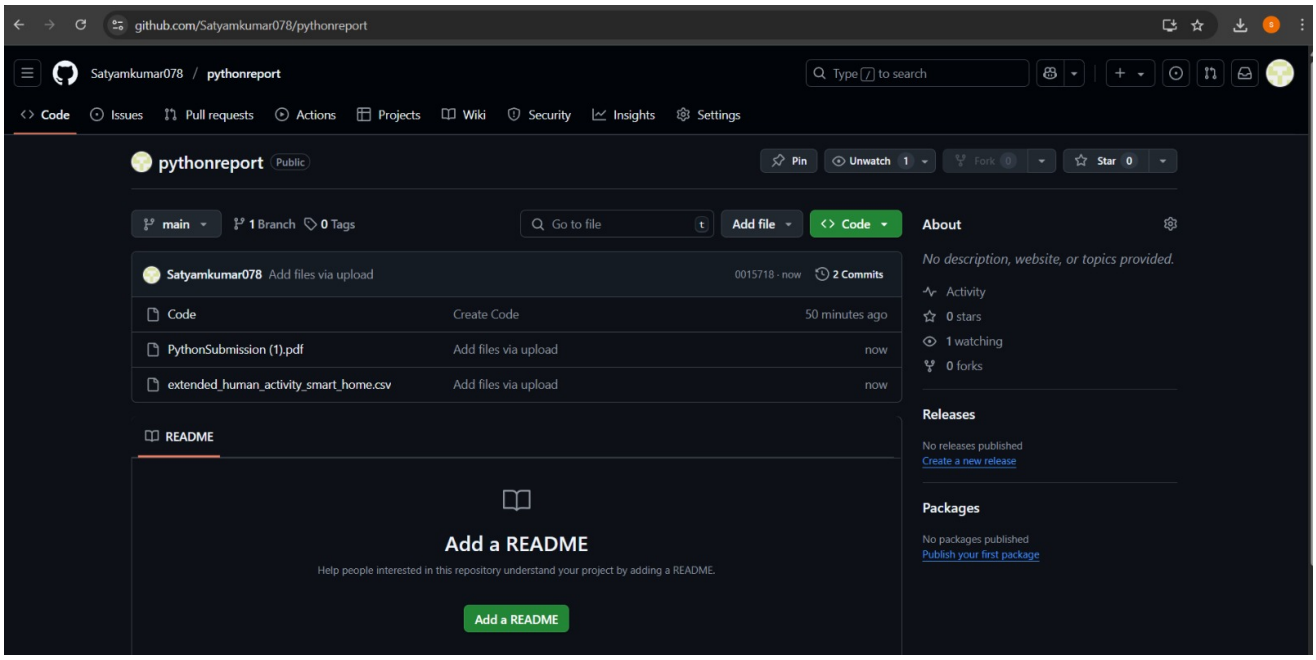
Screenshots

- LinkedIn



Link: https://www.linkedin.com/posts/satyam78_datascience-machinelearning-smarthome-activity-7316854856261029888-lQ75?utm_source=share&utm_medium=member_desktop&rcm=ACoAAEd5BeEBVbU8f0GwkG-airPlmIzhk0Mhwd8

- GitHub



Link: https://github.com/Satyamkumar078/pythonreport

Main code :

```
import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

import plotly.express as px

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import LabelEncoder, StandardScaler

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import classification_report, confusion_matrix, ConfusionMatrixDisplay,
accuracy_score

import warnings
warnings.filterwarnings("ignore")



# ---------------------- LOAD DATA ----------------------

df = pd.read_csv("extended_human_activity_smart_home.csv")

# ---------------------- CLEANING & PREPROCESSING ----------------------

# Remove duplicates
df.drop_duplicates(inplace=True)

# Handle missing values
df.fillna(method='ffill', inplace=True)

# Convert timestamp
df['Timestamp'] = pd.to_datetime(df['Timestamp'])

# Outlier removal (IQR)
numerical_cols = ['Temperature', 'Humidity', 'Light_Level', 'Sound_Level']
for col in numerical_cols:
    Q1 = df[col].quantile(0.25)
    Q3 = df[col].quantile(0.75)
    IQR = Q3 - Q1
    df = df[(df[col] >= Q1 - 1.5 * IQR) & (df[col] <= Q3 + 1.5 * IQR)]
```

```
# Label Encoding
le_room = LabelEncoder()
le_activity = LabelEncoder()
df['Room'] = le_room.fit_transform(df['Room'])
df['Activity'] = le_activity.fit_transform(df['Activity'])

# Add Hour feature
df['Hour'] = df['Timestamp'].dt.hour

# Add Activity label for readable graphs
df['Activity_Label'] = le_activity.inverse_transform(df['Activity'])

# Add Room label for boxplot
df['Room_Label'] = le_room.inverse_transform(df['Room'])

# ---------------------- MODEL TRAINING ----------------------

X = df.drop(['Activity', 'Timestamp', 'Activity_Label', 'Room_Label'], axis=1)
y = df['Activity']

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

clf = RandomForestClassifier()
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)

# ---------------------- EVALUATION ----------------------

print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))

# Confusion Matrix
ConfusionMatrixDisplay(confusion_matrix(y_test, y_pred)).plot()
plt.title("Confusion Matrix")
plt.show()

# ---------------------- VISUALIZATIONS ----------------------

# Sample for clean plots
sample_df = df.sample(300, random_state=42)

# 1. Temperature vs Light Level
fig1 = px.scatter(sample_df, x="Temperature", y="Light_Level", trendline="ols",
```

```
            title="Temperature vs Light Level (Sampled)")
fig1.show()

# 2. Line Graph: Temperature over Time
df_sorted = df.sort_values("Timestamp")
sample_time_df = df_sorted.iloc[::int(len(df_sorted)/200)]
fig2 = px.line(sample_time_df, x="Timestamp", y="Temperature", title="Temperature Over
Time (Sampled)")
fig2.show()

# 3. Bar Graph: Activity Count
f = sample_df.head(10)
activity_counts = f['Activity'].value_counts()
fig3 = px.bar(x=le_activity.inverse_transform(activity_counts.index), y=activity_counts.values,
          labels={'x': 'Activity', 'y': 'Count'}, title="Activity Count (Sampled)")
fig3.show()

# 4. Histogram: Sound Level
fig4 = px.histogram(df.head(100), x="Sound_Level", title="Sound Level Distribution")
fig4.show()

# 5. Pie Chart: Room Distribution
fig5 = px.pie(df, names=df['Room_Label'], title='Room Distribution')
fig5.show()

# 6. Boxplot: Temperature by Room
fig6 = px.box(df, x="Room_Label", y="Temperature", title="Temperature by Room")
fig6.show()

# 7. Heatmap: Correlation
plt.figure(figsize=(10, 6))
sns.heatmap(df.drop(['Timestamp', 'Activity_Label', 'Room_Label'], axis=1).corr(), annot=True,
cmap="coolwarm")
plt.title("Correlation Heatmap")
plt.show()

# 8. Pairplot: Sampled
k = df.head(100)
sample_pair_df = k.drop(['Timestamp', 'Activity_Label', 'Room_Label'], axis=1).sample(80,
random_state=3)
sns.pairplot(sample_pair_df, diag_kind='kde')
plt.suptitle("Pairplot (Sampled)", y=1.02)
plt.show()



# 9. Violin Plot: Light Level by Activity
```

```
fig7 = px.violin(sample_df, x="Activity_Label", y="Light_Level", box=True, points="all",
                 title="Light Level by Activity (Sampled)")
fig7.show()
```