## Project Scope

The project aims to classify devices into "Efficient" and "Inefficient" categories based on various features like energy consumption, usage patterns, and reported issues. This classification can be used in optimizing device performance, reducing energy consumption, and improving overall user satisfaction.

## Project Methodology

Data Collection: The dataset was loaded and preprocessed to handle missing values and ensure data quality.

Exploratory Data Analysis (EDA): EDA was performed to understand the structure of dataset,distribution of data and relationship between various attributes.It helps to identify the anomalies,trends,patterns and correlation between features.

Feature Selection: Feature selection performed to improve the model's accuracy and efficiency while reducing the total number features.The features with high correlation are given more priority.

Model Building: Decision Tree algorithm was built using the selected features for easy interpretation as it can be visualized and understood

Model Evaluation: The model was evaluated based on the accuracy,precision,recall and confusion matrix to ensure its effectiveness

Hyperparameter Tuning: GridSearchCV was used to optimize hyperparameters for the decision tree to achieve best performance.max_depth,min_sample_split,min_sample_leaf and criterion were optimized to prevent overfitting

# Design & Development

### Data Analysis

Data Preprocessing: This involves handling missing values, categorical variables, and normalizing or standardizing features if necessary.

Feature Selection: Identify the most important features that will be used in the decision tree. You can use techniques like correlation matrices, feature importance scores, or recursive feature elimination.

### Algorithm Selection

Decision Tree Algorithm:They work by recursively splitting the data based on the feature that provides the most information gain (for classification) or reduction in variance (for regression).

### Hyperparameter Tuning

Max Depth (max_depth): Limits the maximum depth of the tree, preventing overfitting by stopping the tree from growing too deep.

Min Samples Split (min_samples_split): The minimum number of samples required to split an internal node, which controls the decision to create further splits in the tree.

Min Samples Leaf (min_samples_leaf): The minimum number of samples required to be at a leaf node, ensuring that leaf nodes have sufficient data.

Criterion: The function used to measure the quality of a split (e.g., Gini impurity or entropy for classification, mean squared error for regression).

**Tuning Process**:

Grid Search: Use grid search to systematically explore a range of hyperparameter values.
Cross-Validation: Implement cross-validation during grid search to evaluate the model's performance across different splits of the data. This helps in selecting hyperparameters that generalize well.

Random Search: If grid search is computationally expensive, random search is an alternative that randomly samples hyperparameter combinations, covering a broader search space with fewer computations.

**Evaluation**

Performance Metrics: Evaluate the model's performance using appropriate metrics (e.g., accuracy, precision, recall.

# Test Scripts/Methods – How do you test your data

**Train-Test Split**

To evaluate the model's performance on unseen data.

Method: Split your dataset into training and testing sets, typically using an 80-20 or 70-30 split. This allows you to train your model on the majority of the data and test it on a smaller portion to see how well it generalizes.

**Cross-Validation**

Objective: To ensure that your model is not overfitting and performs well across different subsets of the data.

Method: Use k-fold cross-validation to split the data into k subsets (folds), train the model on k-1 folds, and validate it on the remaining fold. This process is repeated k times with different folds each time, and the results are averaged.

**Hyperparameter Testing**

Objective: To find the optimal hyperparameters for your decision tree model.

Method: Use GridSearchCV or RandomizedSearchCV to test different combinations of hyperparameters and find the best performing set.

**Model Evaluation**

Objective: To evaluate the model's performance on the test set and ensure it meets the desired criteria.

Method: Use metrics like accuracy, precision, recall, F1-score (for classification tasks), or RMSE, MAE (for regression tasks) to assess model performance.

**Confusion Matrix**

Objective: To understand how well the model is distinguishing between different classes.

Method: Plot a confusion matrix to visualize the performance of your classification model.

**Feature Importance**

Objective: To identify which features are most important for making predictions.

Method: Evaluate feature importance to understand how each feature impacts the decision-making process of your model.

**Overfitting and Underfitting Check**

Objective: To ensure your model is not overfitting or underfitting the data.

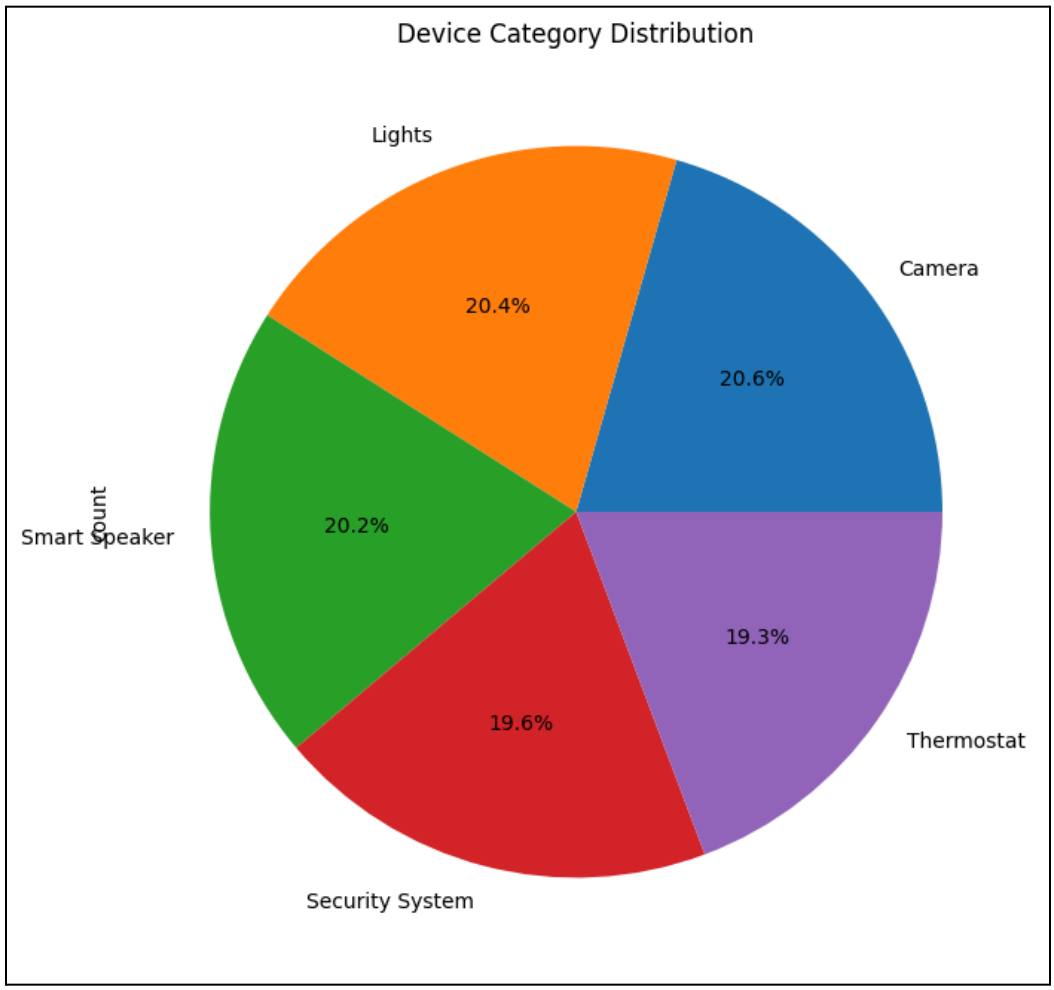
Method: Compare the training and testing scores. A significant difference between the two indicates overfitting, while poor performance on both may indicate underfitting.

**Error Analysis**

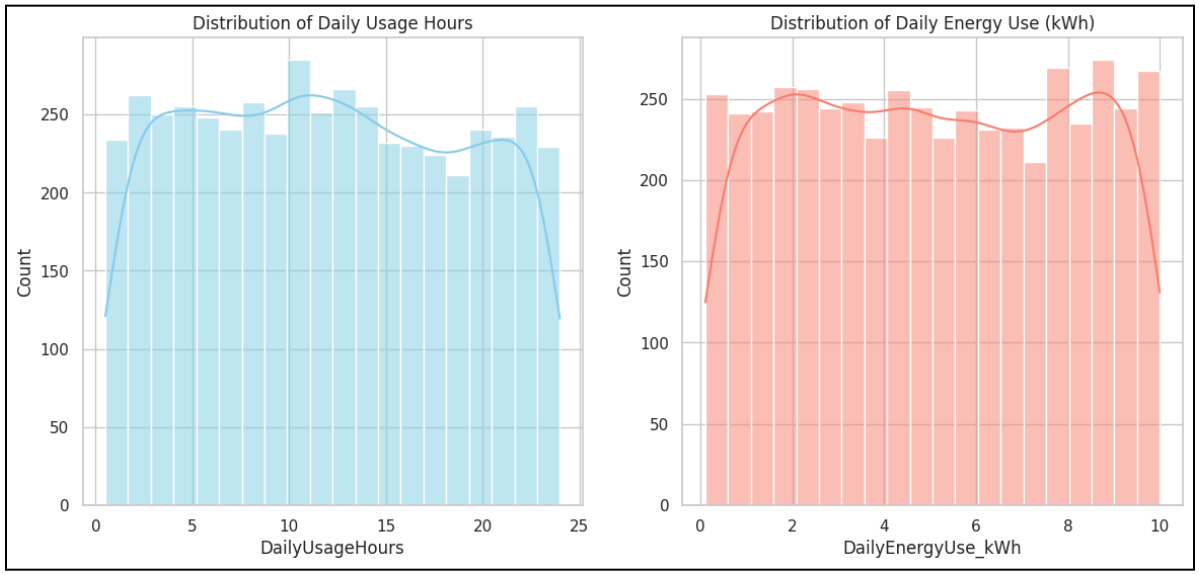
Objective: To understand where the model is making errors.

Method: Analyze misclassified instances or high-error predictions to identify patterns or potential improvements.

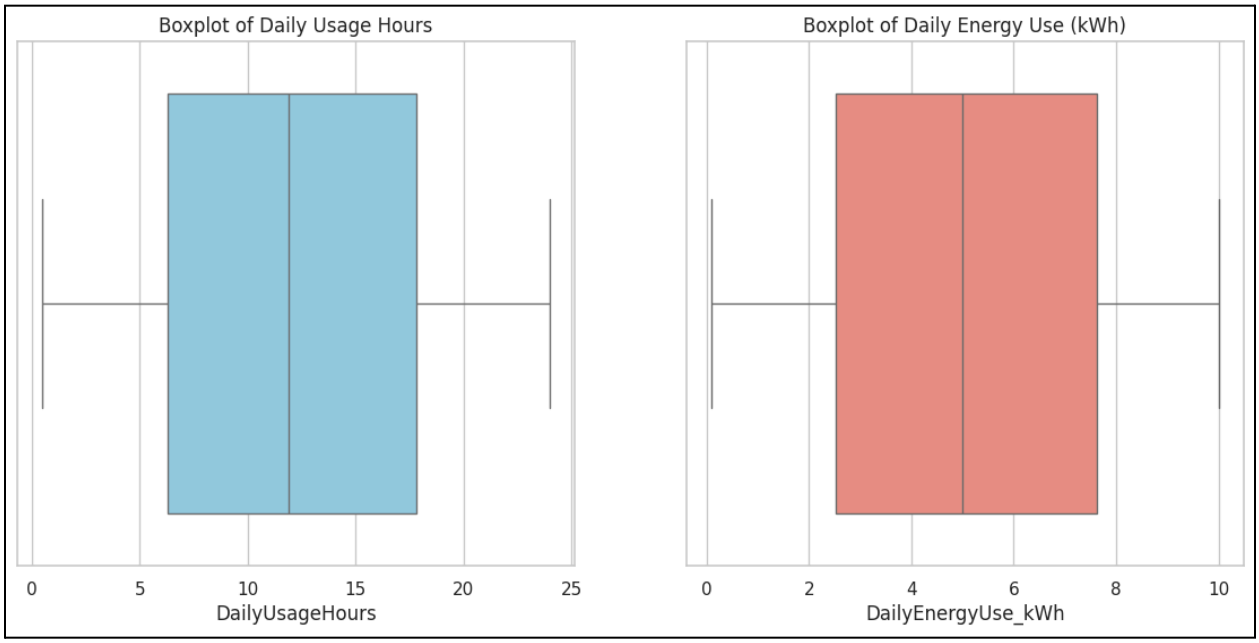
**VISUALISATION**




The devices are evenly distributed in the dataset,each device contributing approximately 20% of the data.

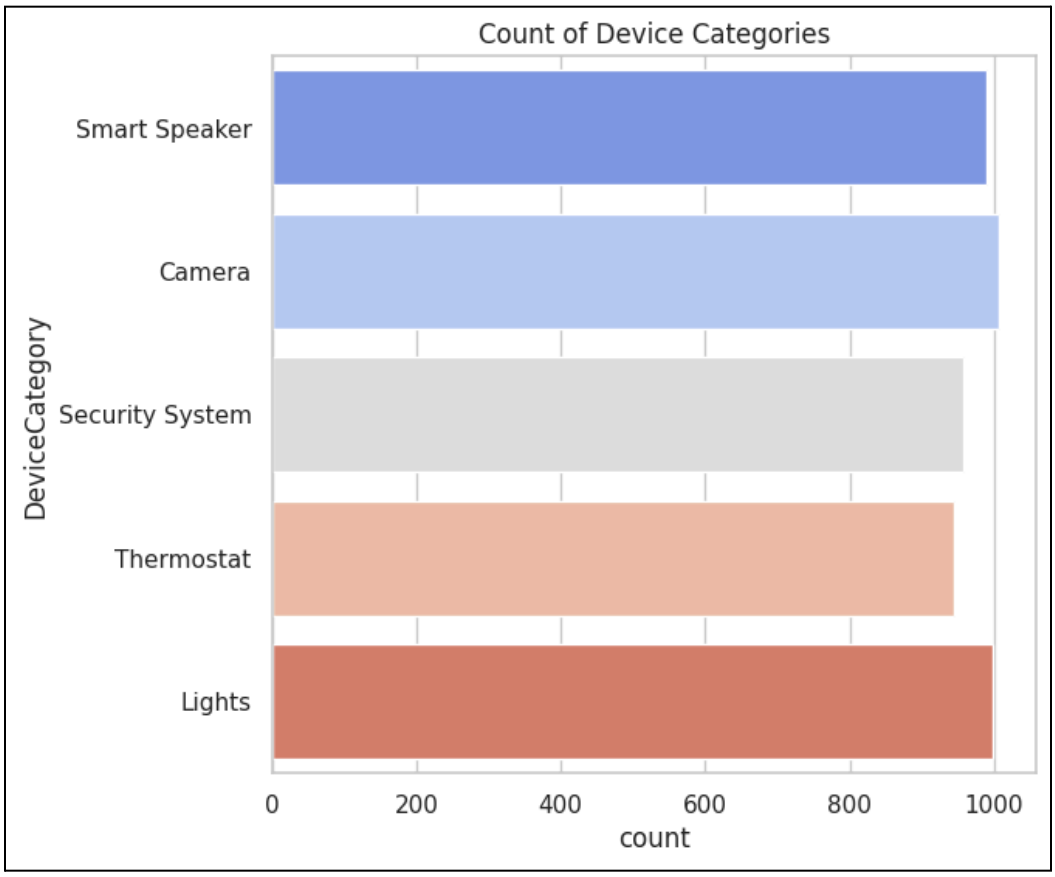



The daily Usage hours are evenly distributed with few outliers same with the Daily Energy units

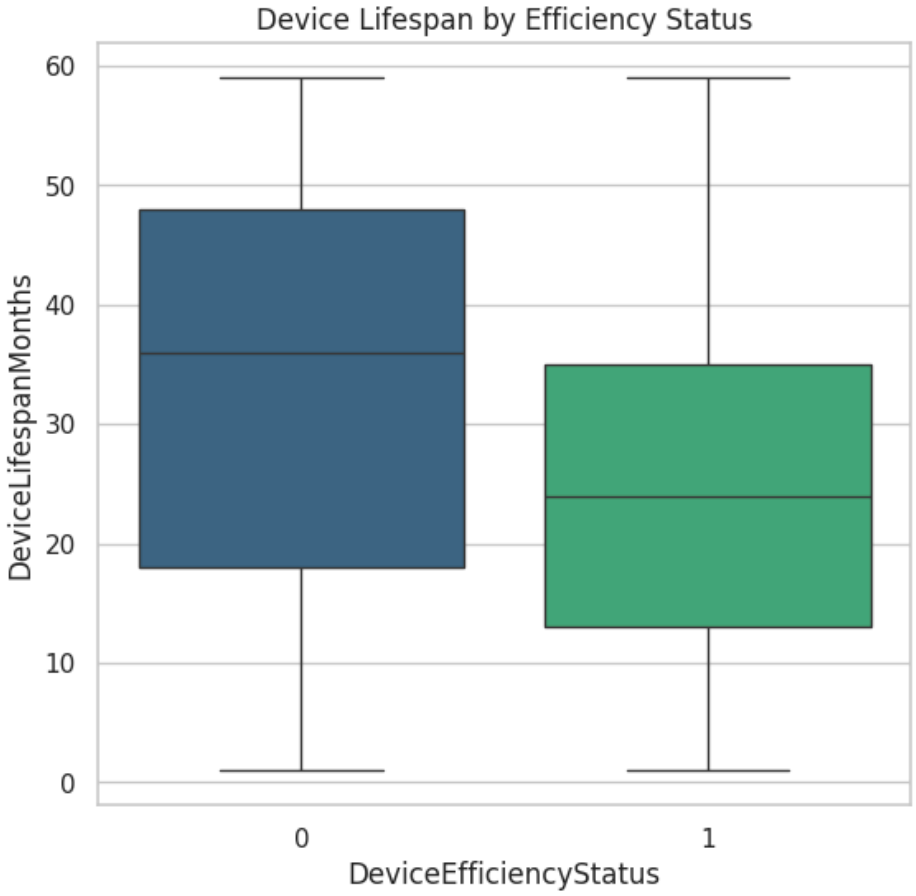



The boxplots specifies:

The median value for the daily usage hours falls in between the range of 10-15 and the units in between 4 and 6




Camera is frequently used followed by lights.smart speaker.Security Systems and thermostats are used less frequently compared to other devices and thermostat being the least




The boxplot suggests that the devices which are inefficient  have a higher median lifespan compared to the efficient devices which have a lesser median lifespan .

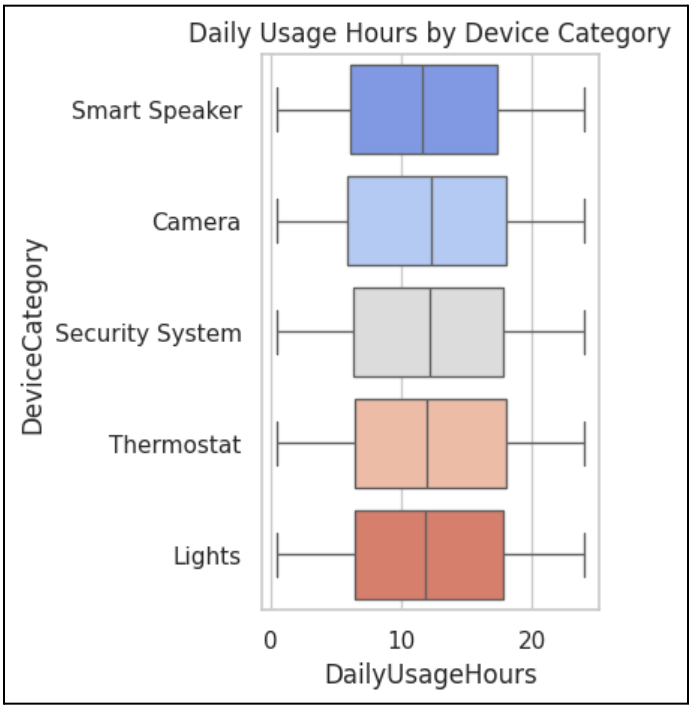
The inefficient devices are having higher variability(some lasts longer and some not) in lifespan compared to efficient devices.

Insights:

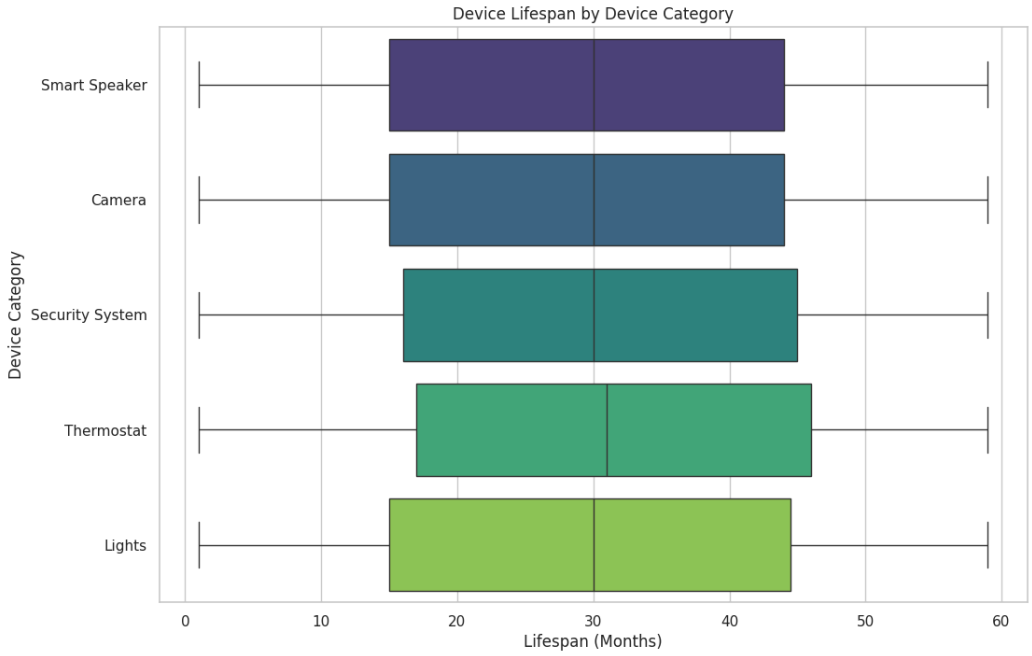
Older and less efficient devices are more durable

Newer, more efficient devices are not built to last as long

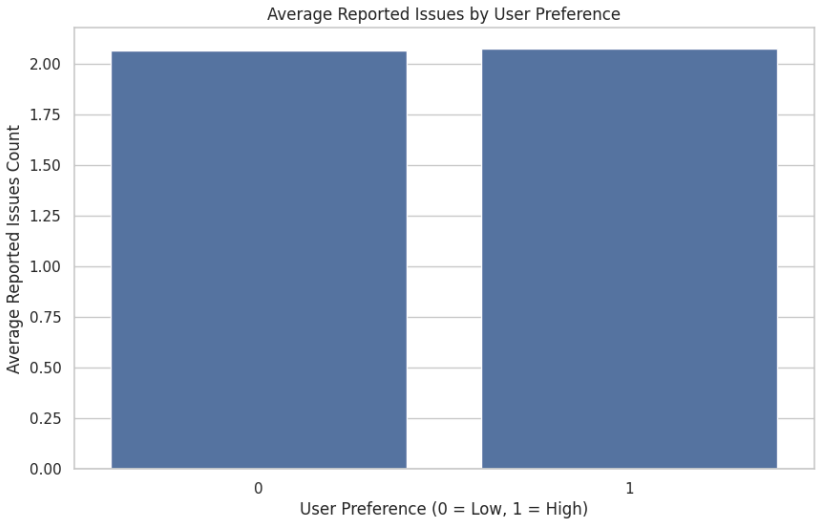
Efficient devices tend to have more predictable lifespan

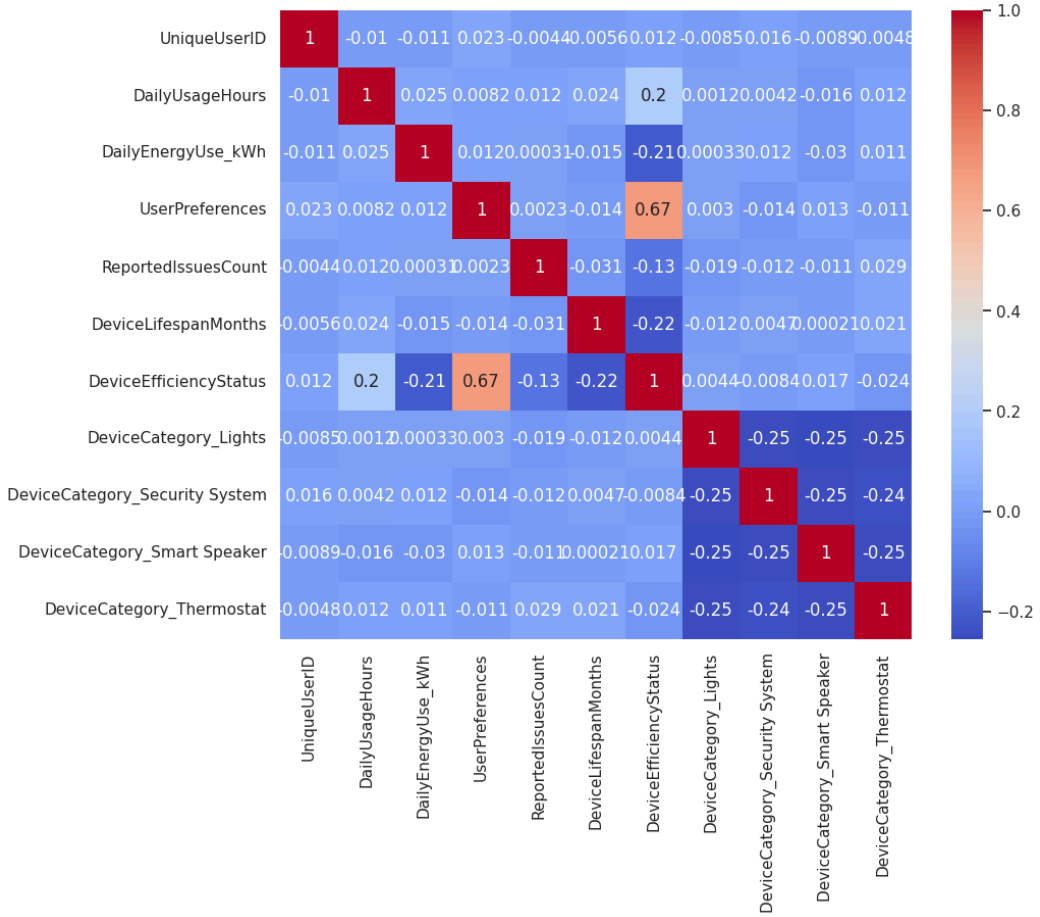



Camera has the highest usage hours along with Security System,Thermostat and Lights

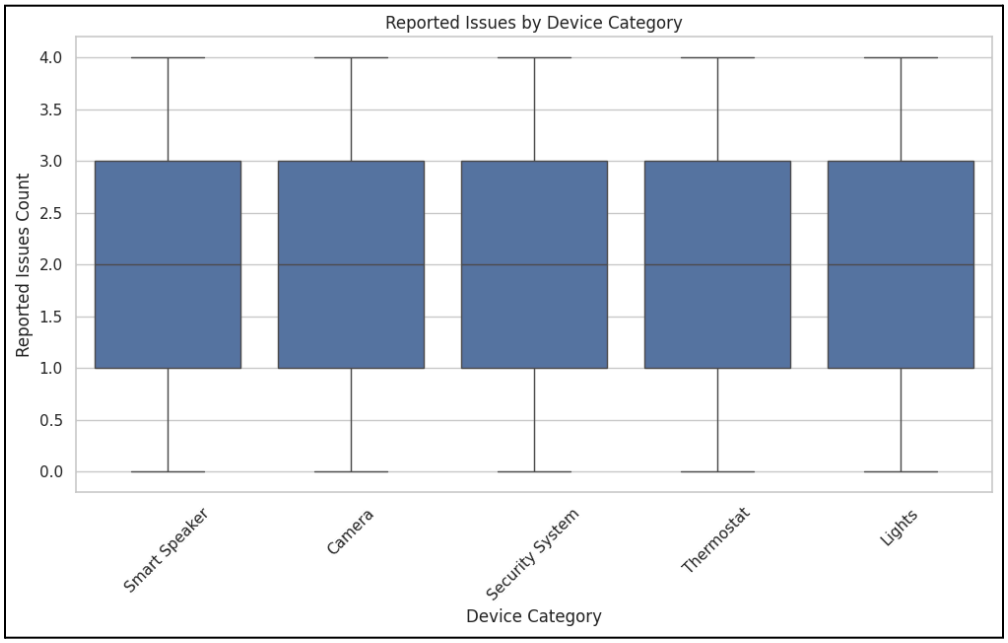


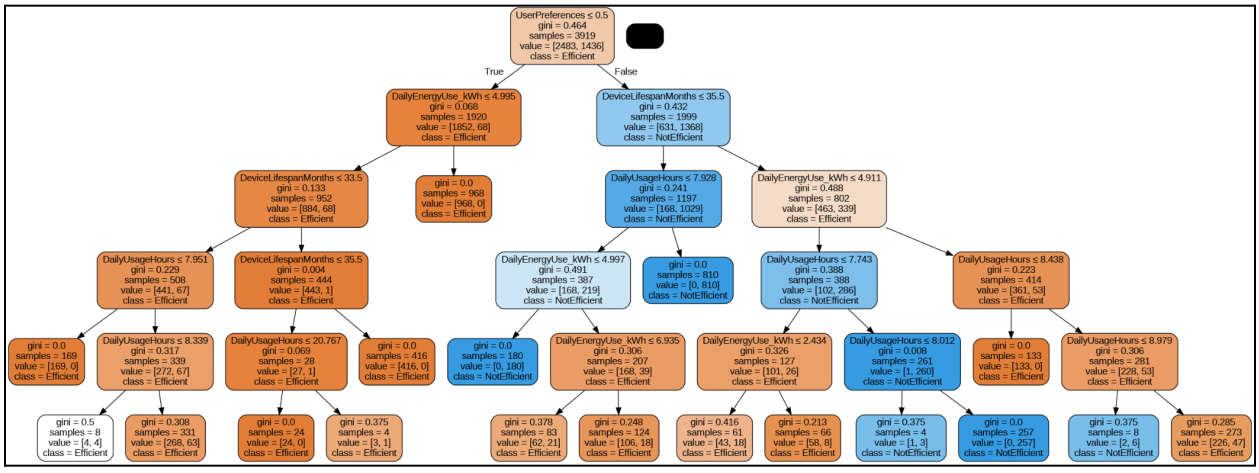

The median of lifespan of thermostat is higher than the other device categories




The highest correlation of target variable , deviceEfficiency Status is with the features Usage hours, Units used,User Preference,Lifespan.


Reported Issues by Device Category
Reported Issues Count
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Smart Speaker
Camera
Security System
Thermostat
Lights
Device Category


UserPreferences ≤ 0.5
gini = 0.464
samples = 3919
value = [2483, 1436]
class = Efficient
True
False
DailyEnergyUse_kWh ≤ 4.995
gini = 0.068
samples = 1920
value = [1852, 68]
class = Efficient
DeviceLifespanMonths ≤ 35.5
gini = 0.432
samples = 1999
value = [631, 1368]
class = NotEfficient
DeviceLifespanMonths ≤ 33.5
gini = 0.133
samples = 952
value = [884, 68]
class = Efficient
gini = 0.0
samples = 968
value = [968, 0]
class = Efficient
DailyUsageHours ≤ 7.928
gini = 0.241
samples = 1197
value = [168, 1029]
class = NotEfficient
DailyEnergyUse_kWh ≤ 4.911
gini = 0.488
samples = 802
value = [463, 339]
class = Efficient
DailyUsageHours ≤ 7.951
gini = 0.229
samples = 508
value = [441, 67]
class = Efficient
DeviceLifespanMonths ≤ 35.5
gini = 0.004
samples = 444
value = [443, 1]
class = Efficient
DailyEnergyUse_kWh ≤ 4.997
gini = 0.491
samples = 387
value = [168, 219]
class = NotEfficient
gini = 0.0
samples = 810
value = [0, 810]
class = NotEfficient
DailyUsageHours ≤ 7.743
gini = 0.388
samples = 388
value = [102, 286]
class = NotEfficient
DailyUsageHours ≤ 8.438
gini = 0.223
samples = 414
value = [361, 53]
class = Efficient
gini = 0.0
samples = 169
value = [169, 0]
class = Efficient
DailyUsageHours ≤ 8.339
gini = 0.317
samples = 339
value = [272, 67]
class = Efficient
DailyUsageHours ≤ 20.767
gini = 0.069
samples = 28
value = [27, 1]
class = Efficient
gini = 0.0
samples = 416
value = [416, 0]
class = Efficient
gini = 0.0
samples = 180
value = [0, 180]
class = NotEfficient
DailyEnergyUse_kWh ≤ 6.935
gini = 0.306
samples = 207
value = [168, 39]
class = Efficient
DailyEnergyUse_kWh ≤ 2.434
gini = 0.326
samples = 127
value = [101, 26]
class = Efficient
DailyUsageHours ≤ 8.012
gini = 0.008
samples = 261
value = [1, 260]
class = NotEfficient
gini = 0.0
samples = 133
value = [133, 0]
class = Efficient
DailyUsageHours ≤ 8.979
gini = 0.306
samples = 281
value = [228, 53]
class = Efficient
gini = 0.5
samples = 8
value = [4, 4]
class = Efficient
gini = 0.308
samples = 331
value = [268, 63]
class = Efficient
gini = 0.0
samples = 24
value = [24, 0]
class = Efficient
gini = 0.375
samples = 4
value = [3, 1]
class = Efficient
gini = 0.378
samples = 83
value = [62, 21]
class = Efficient
gini = 0.248
samples = 124
value = [106, 18]
class = Efficient
gini = 0.416
samples = 61
value = [43, 18]
class = Efficient
gini = 0.213
samples = 66
value = [58, 8]
class = Efficient
gini = 0.375
samples = 4
value = [1, 3]
class = NotEfficient
gini = 0.0
samples = 257
value = [0, 257]
class = NotEfficient
gini = 0.375
samples = 8
value = [2, 6]
class = NotEfficient
gini = 0.285
samples = 273
value = [225, 47]
class = Efficient

**DataSet Characteristics**

**Description**:

This dataset captures smart home device usage metrics, offering insights into user behaviour, device efficiency, and preferences. It includes data on device types, usage patterns, energy consumption, malfunction incidents, and user satisfaction metrics.

**Features**:

- · UniqueUserID: Unique identifier for each user.
- · DeviceCategory: Type of smart home device (e.g., Lights, Thermostat).
- · DailyUsageHours: Average hours per day the device is used.
- · DailyEnergyUse_kWh: Daily energy consumption of the device (kWh).
- · UserPreference: User preference for device usage (0 - Low, 1 - High).
- · ReportedIssuesCount: Number of malfunction incidents reported.
- · DeviceLifespanMonths: Age of the device in months.
- · DeviceEfficiencyStatus (Target Variable): Efficiency status of the smart home device (0 - Inefficient, 1 - Efficient).

- 4899 rows and 8 features
- 7 numerical and 1 categorical variable(device category)
- Target variable is device efficiency
- Boolean variables are user preferences and efficiency status
- Average usage hours 12 , range 0.5-24
- Average Units 5 units,range 0-9.9
- Average issues reported 2, range 0-4
- No null values were detected
- No duplicate values found
- Even distribution of usage hours and units used with few outliers
- The median of usage hours between 10 and 15, units between 4 and 6
- Users prefer camera the most and thermostat the least

**Median Lifespan by Device Category:**

Devices with a higher median lifespan are generally those that last longer overall. This suggests that certain categories of devices are designed or built to be more durable, which could be a key factor in their long-term value.

**Thermostat Lifespan vs. Preference:**

The thermostat has a higher lifespan but is less preferred compared to other devices. This could indicate that while thermostats are built to last longer, they might not be as popular due to factors such as functionality, user experience, or market trends.

**Efficiency and Lifespan:**

The higher lifespan of inefficient devices suggests that these devices are built to last despite their inefficiency. This might indicate that the design or materials used in these devices contribute to their durability, even though they may not be energy-efficient.

**User Preference Ranking:**

The ranking of devices based on user preference is as follows: Camera > Lights > Smart Speaker > Security System > Thermostat. This preference ranking could inform product development and marketing strategies, suggesting that features, ease of use, or other factors influencing user satisfaction are more crucial for cameras and lights compared to other devices.

**High Energy Consumption Outliers:**

There are devices with extremely high usage, indicating a few outliers that consume significantly more energy. This could point to potential inefficiencies or issues with certain devices, or it may reflect specific user behaviors. Addressing these outliers might involve investigating the reasons behind high energy consumption and exploring opportunities for optimizing energy use.

**The Model**

- Decision Tree Algorithm
- Best Parameters are DecisionTreeClassifier(max_depth=5, min_samples_leaf=4, random_state=42)
- The most important feature is 'UserPreferences' with an importance score of 0.45328625495143476
- Cross-validation accuracy scores: [0.95153061 0.96045918 0.93622449 0.96045918 0.94380587]
- Mean cross-validation accuracy: 0.9504958688456225
-
- Accuracy:  0.9489795918367347
- Precision: 0.9935897435897436
- Recall: 0.8659217877094972
- Confusion Matrix:

  [[620   2]

  [ 48 310]]

**********