Project Report

INTRODUCTION

1.1 Project Overview

Power consumption analysis for households is an ML project that involves using machine learning algorithms to analyse and predict the energy consumption patterns of residential buildings. The goal of this project is to help homeowners and utility companies better manage their energy usage, reduce waste, and lower costs.

The project involves collecting data on energy consumption and related factors such as weather, time of day, and occupancy. This data is then used to train machine learning models to make accurate predictions of future energy consumption based on these factors. The models can be used to identify patterns in energy usage and make recommendations for ways to reduce energy waste and improve efficiency.

Overall, power consumption analysis for households is an important application of machine learning that has the potential to make a significant impact on energy usage and sustainability.

1.2 Purpose

The purpose is to provide an overview and description of the project on power consumption analysis for households using machine learning. It explains the goal of the project, which is to analyze and predict energy consumption patterns in residential buildings. This also highlights the potential benefits of the project, including helping homeowners and utility companies manage energy usage, reduce waste, and lower costs. Additionally, it mentions the data collection process and how machine learning models can be used to make accurate predictions and provide recommendations for improving energy efficiency. This emphasizes the importance of this application of machine learning in promoting sustainable energy usage.

2. IDEATION & PROPOSED SOLUTION

2.1 Problem Statement Definition

- The problem is the lack of a reliable energy consumption analysis system for energy-efficient appliances.

- Current methods for evaluating the energy efficiency of appliances are often inaccurate or inefficient, leading to unreliable information for consumers and limited opportunities for manufacturers to improve their products.

- This hinders the adoption of energy-efficient appliances, increases energy consumption, and contributes to environmental degradation.

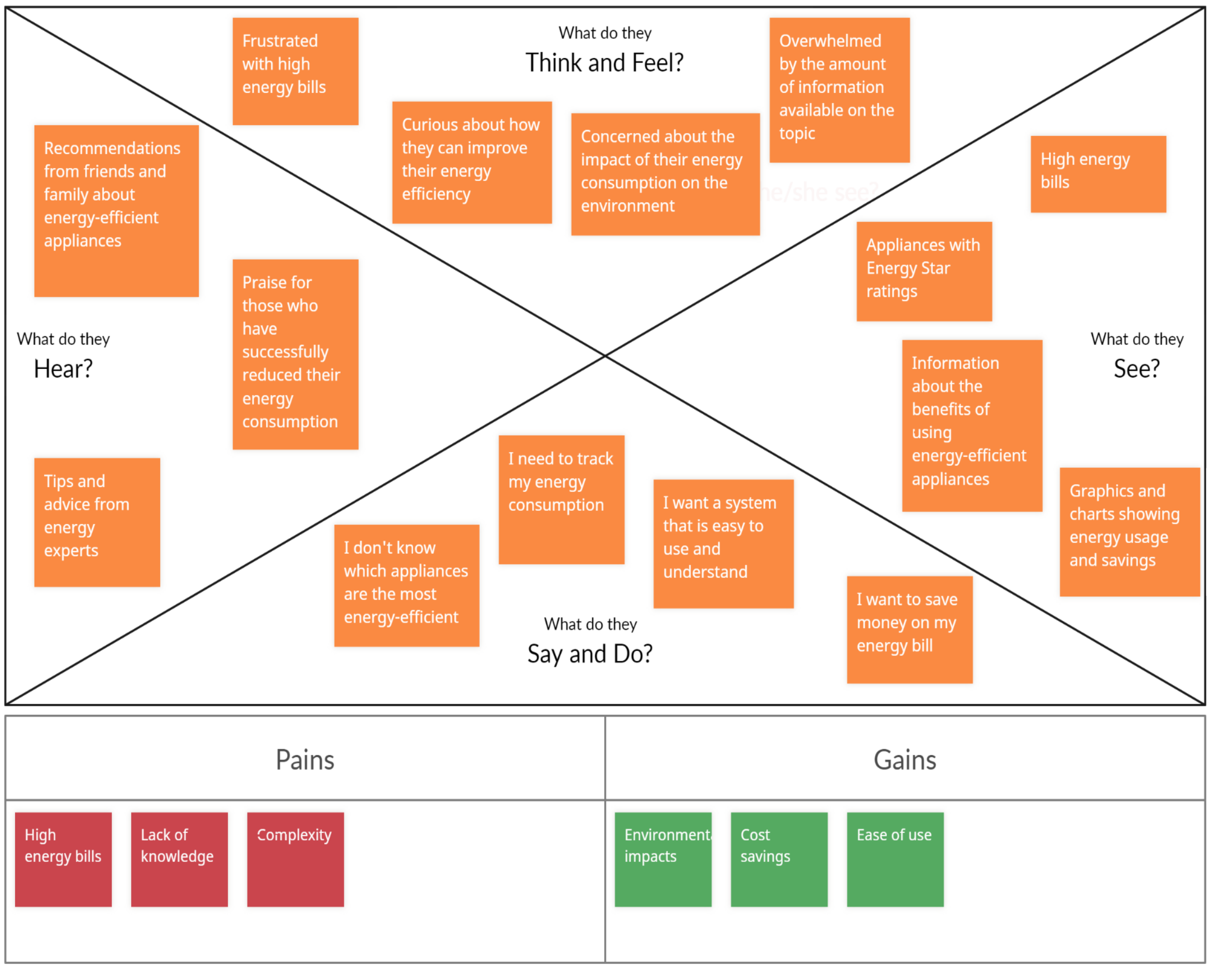
- there is a need for a robust and accurate system that can effectively measure and analyze energy consumption, enabling consumers to make informed decisions and manufacturers to develop more energy-efficient appliances.

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	a homeowner who is environmentally conscious and wants to reduce my energy consumption and electricity bills while maintaining a comfortable living environment.	monitor and manage my household energy usage effectively with energy-efficient appliances.	I face the challenge of not having an accurate and dependable system to track and analyze my energy consumption.	the current systems available are not designed to work with energy-efficient appliances and lack the necessary features and functionalities to provide reliable energy consumption analysis	frustrated and unable to make informed decisions about my energy consumption, leading to a sense of helplessness and wasted efforts in trying to reduce my carbon footprint.
PS-2	a small business owner who operates a manufacturing plant with multiple energy-intensive machines and equipment.	reduce my energy consumption and associated costs while maintaining optimal production levels	I face the challenge of not having a reliable energy consumption analysis system that accurately measures the energy usage of each machine and provides meaningful insights to optimize energy usage.	the current systems available do not account for the unique energy requirements of each machine, and lack the necessary features and functionalities to provide accurate and real-time energy consumption analysis.	overwhelmed and unable to make informed decisions about my energy usage, leading to higher electricity bills and reduced profits, as well as a sense of frustration and helplessness in addressing my environmental impact.

2.2 Empathy Map Canvas

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes.

In the context of the project "A Reliable Energy Consumption Analysis System for Energy-Efficient Appliances," an empathy map would be used to gain a deeper understanding of the individuals who are interested in reducing their energy consumption and using energy-efficient appliances.

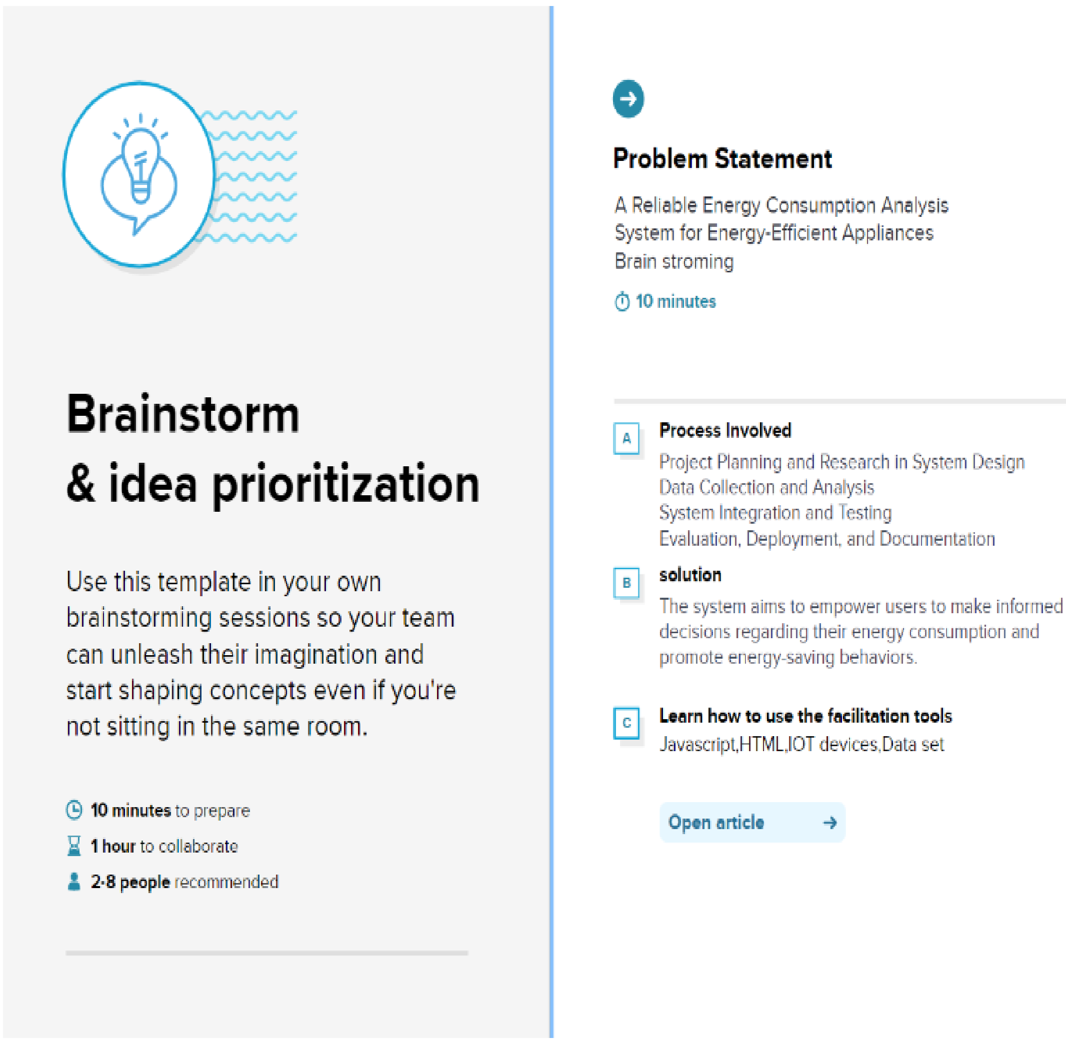


2.3 Ideation & Brainstorming

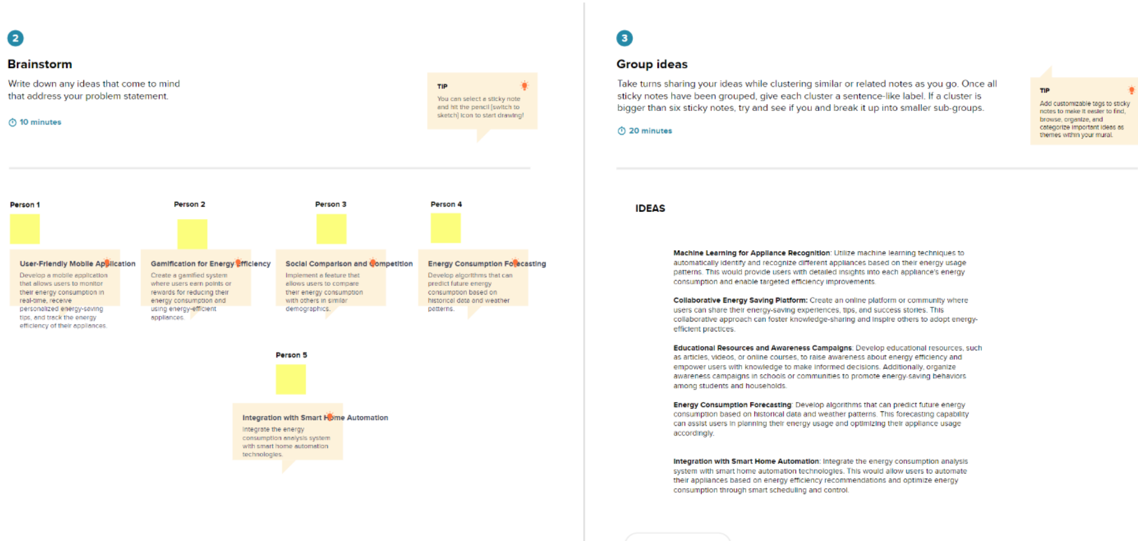
Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving.

Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

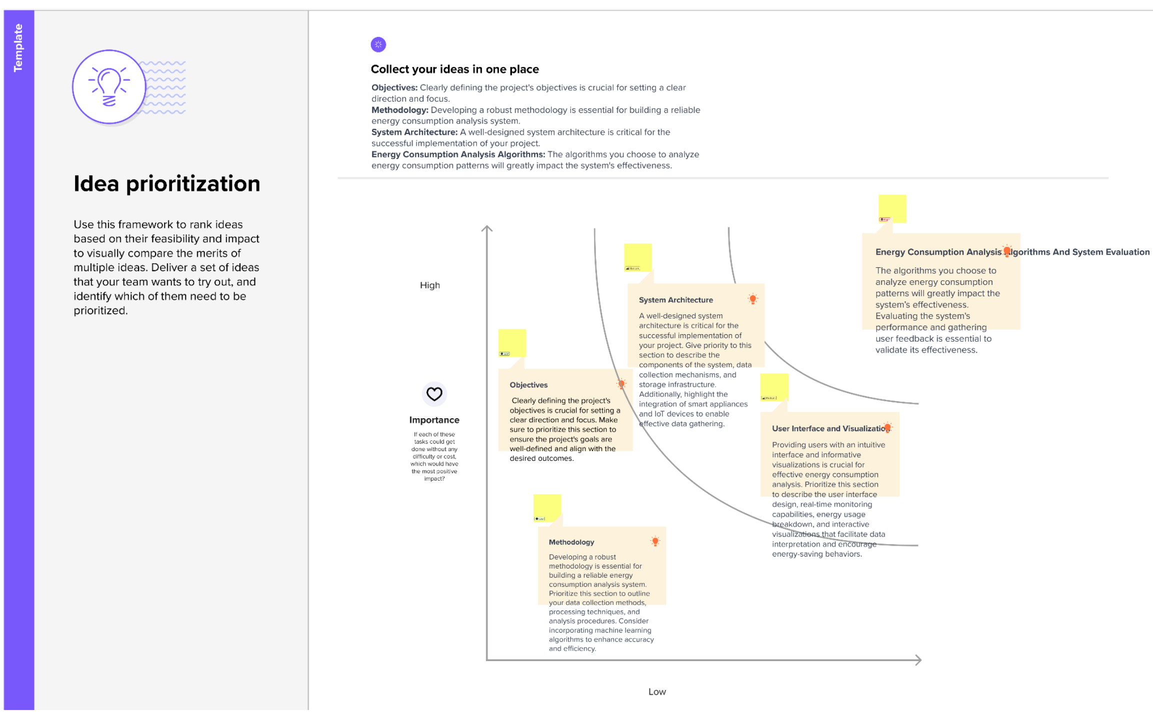
Step-1: Team Gathering, Collaboration and Select the Problem Statement



Step-2: Brainstorm, Idea Listing and Grouping



Step-3: Idea Prioritization



2.4 Proposed Solution

S.No.	Parameter	Description
1	Problem Statement (Problem to be solved)	The objective is to accurately measure and analyze the energy consumption of various appliances, identify energy-efficient appliances, and provide recommendations for optimizing energy usage. The system aims to help households and businesses to reduce their energy consumption, save money on energy bills, and contribute to a sustainable future.
2	Idea / Solution description	The proposed solution is to develop a reliable energy consumption analysis system that uses data science techniques such as statistical analysis, machine learning, and data visualization to analyze the collected data. The system will collect data from sensors installed on appliances, store the data in a secure and reliable database, and provide insights into energy-efficient appliances. the aim is to help the people manage the energy and maintain them at lower cost
3	Novelty / Uniqueness	The uniqueness of this project lies in its ability to provide customized recommendations for optimizing energy usage, based on the specific usage patterns and preferences of each household or business. By collecting data from sensors installed on appliances and using machine learning algorithms, the system can identify which appliances are consuming the most energy and provide tailored recommendations for reducing energy consumption. Additionally, the system can provide real-time feedback on energy usage, allowing households and businesses to adjust their usage patterns and save even more on energy bills.
4	Social Impact / Customer Satisfaction	The proposed system can help customers feel empowered and in control of their energy usage. By providing real-time feedback on energy consumption and identifying areas for improvement, the system can help customers take action to reduce their energy consumption and save money on energy bills. This can lead to increased customer satisfaction and loyalty as customers see tangible benefits from using the system. Overall, the proposed system has the potential to make a significant social impact by promoting energy efficiency, reducing energy poverty, and contributing to a sustainable future

5	Business Model (Revenue Model)	Subscription-based service model where customers pay a monthly or annual fee to access the system's features. Partnership model with energy companies or appliance manufacturers who sponsor the system and offer discounts or incentives to customers. Sale of data insights and analytics to third-party organizations, such as energy companies or government agencies, which can inform policy decisions or new energy-saving initiatives. The revenue model could be flexible and evolve over time as the system gains traction and new opportunities emerge.
6	Scalability of the Solution	Cloud-based architecture enables easy scalability to handle increased user traffic and data storage needs. Data-driven approach allows the system to adapt to new data sources, increasing the accuracy of energy consumption predictions over time. Modular design enables new features to be added to the system without significant changes to the existing system, reducing downtime and ensuring continuous operation. Open-source technology is highly customizable and can be easily modified to meet specific needs, increasing scalability potential.

3. **REQUIREMENT ANALYSIS**

3.1 Functional requirement

Data Collection: The system should be able to collect data on energy consumption from residential buildings, including factors such as weather, time of day, and occupancy.

Data Preprocessing: The system should preprocess the collected data by cleaning, transforming, and normalizing it for further analysis.

Machine Learning Model Training: The system should train machine learning models using the preprocessed data to predict future energy consumption patterns.

Prediction and Analysis: The system should be able to use the trained models to make accurate predictions of energy consumption based on input factors.

Pattern Identification: The system should identify patterns in energy usage through data analysis to provide insights and recommendations for energy management.

Energy Waste Reduction Recommendations: The system should generate recommendations for homeowners and utility companies on ways to reduce energy waste and improve efficiency based on the identified patterns.

User Interface: The system should provide a user-friendly interface for homeowners and utility companies to interact with, view energy consumption data, and access recommendations.

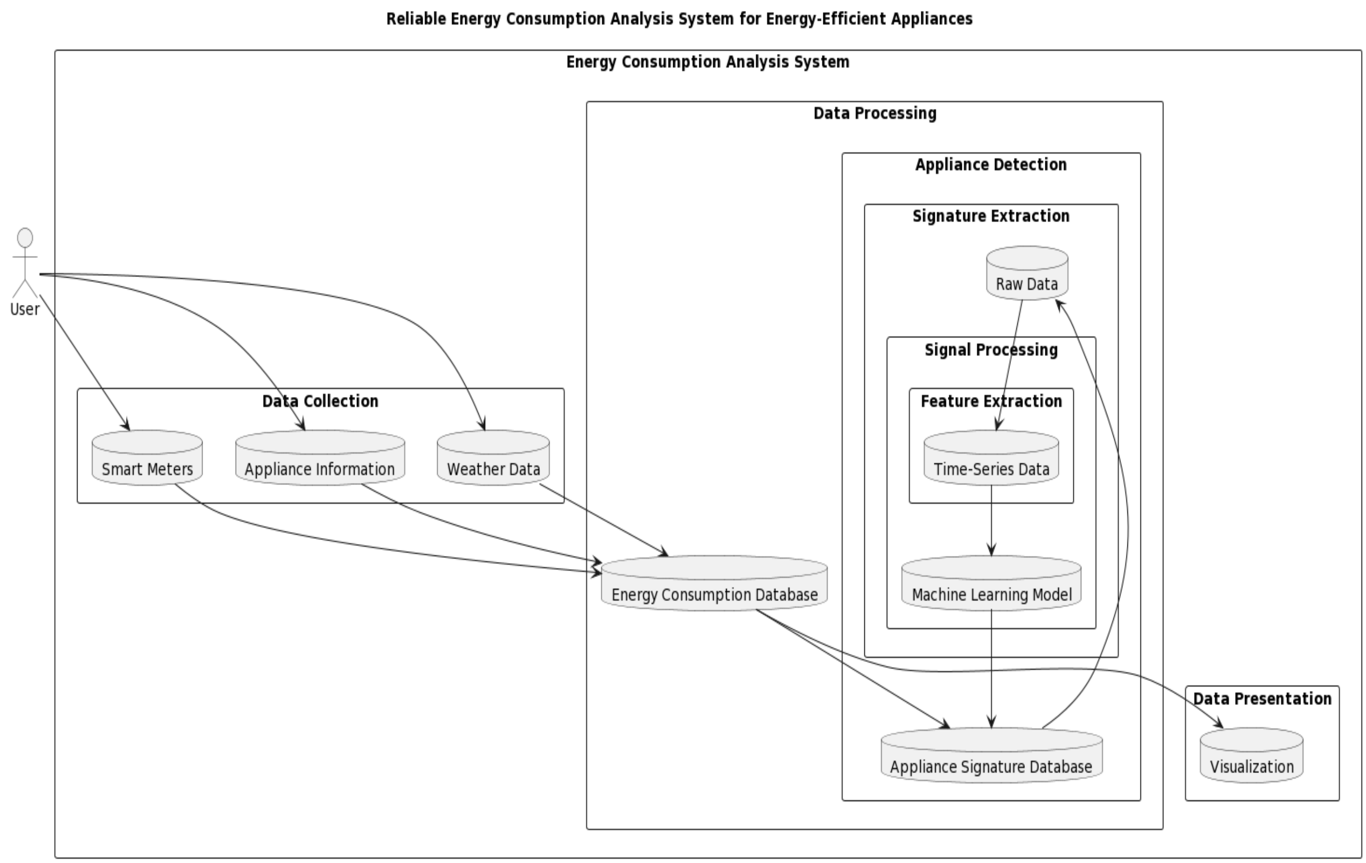
3.2 Non-Functional requirements

- Accuracy: The machine learning models should achieve a high level of accuracy in predicting energy consumption patterns to provide reliable insights and recommendations.
- Scalability: The system should be able to handle a large volume of data from multiple residential buildings and scale accordingly as the number of users and data sources increase.
- Performance: The system should respond quickly and efficiently to user queries, providing real-time predictions and recommendations.
- Security: The system should ensure the security and privacy of the collected data, adhering to relevant data protection regulations.
- Reliability: The system should be highly reliable, minimizing downtime and ensuring data integrity throughout the process.
- Usability: The user interface should be intuitive, easy to navigate, and provide a pleasant user experience for homeowners and utility company personnel.
- Maintainability: The system should be designed and developed in a modular and maintainable manner, allowing for easy updates, bug fixes, and future enhancements.
- Compatibility: The system should be compatible with different types of energy consumption monitoring devices and systems commonly used in residential buildings.
- Documentation: The system should have comprehensive documentation, including user manuals, technical specifications, and guidelines for system maintenance and troubleshooting.

4. PROJECT DESIGN

4.1 Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

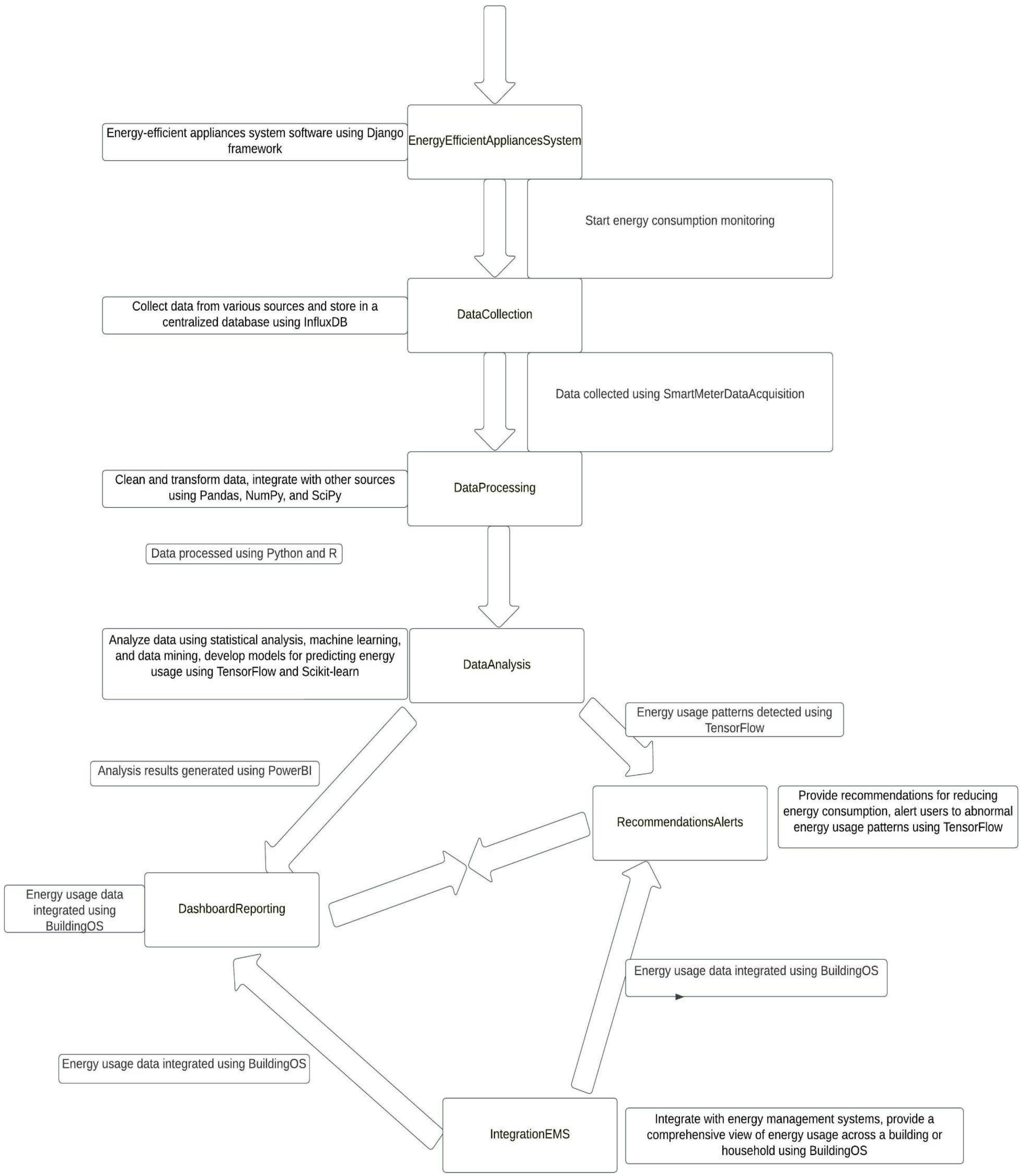


4.2 Solution & Technical Architecture

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions.

Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.


EnergyEfficientAppliancesSystem
Energy-efficient appliances system software using Django framework
Start energy consumption monitoring
DataCollection
Collect data from various sources and store in a centralized database using InfluxDB
Data collected using SmartMeterDataAcquisition
DataProcessing
Clean and transform data, integrate with other sources using Pandas, NumPy, and SciPy
Data processed using Python and R
DataAnalysis
Analyze data using statistical analysis, machine learning, and data mining, develop models for predicting energy usage using TensorFlow and Scikit-learn
Energy usage patterns detected using TensorFlow
Analysis results generated using PowerBI
RecommendationsAlerts
Provide recommendations for reducing energy consumption, alert users to abnormal energy usage patterns using TensorFlow
DashboardReporting
Energy usage data integrated using BuildingOS
Energy usage data integrated using BuildingOS
Energy usage data integrated using BuildingOS
IntegrationEMS
Integrate with energy management systems, provide a comprehensive view of energy usage across a building or household using BuildingOS

4.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Team Member
Administrator	Energy Usage Tracking	US-001	As an administrator, I want to view energy consumption data	The system should display energy consumption data in a user-friendly interface	High	Kanish
Customer	Energy Efficiency Analysis	US-002	As a customer, I want to receive recommendations for energy-efficient appliances	The system should suggest energy-efficient appliances based on usage patterns	Medium	Kesavardhan
Technician	Data Visualization	US-003	As a technician, I want to analyze energy consumption trends	The system should provide graphical representations and statistical analysis	High	Mugilan
Customer	Energy Usage Comparison	US-004	As a customer, I want to compare energy consumption of different appliances	The system should allow users to compare energy usage of various appliances	Low	Lincy
Administrator	User Management	US-005	As an administrator, I want to add and remove users	The system should provide functionality to manage user accounts	Medium	Hari Raja

5. CODING & SOLUTIONING

5.1 **Machine Learning (Linear Regression):**

The predict_energy_consumption function uses a linear regression model from scikit-learn (LinearRegression) to predict energy consumption based on given features: power rating, usage pattern, and energy efficiency rating.

The generate_suggestions function provides suggestions based on the user inputs, such as recommending energy-efficient appliances or optimizing usage patterns.

The visualize_energy_consumption function generates plots to visualize energy consumption analysis and trends.

5.2 **Data Handling:**

The code uses the Pandas library to read and manipulate data. It loads data from CSV files (appliance.csv and trend.csv) for training the model and plotting energy consumption trends.

5.3 **CSV files**

We used data.csv file to store the input trend and analyze the data.

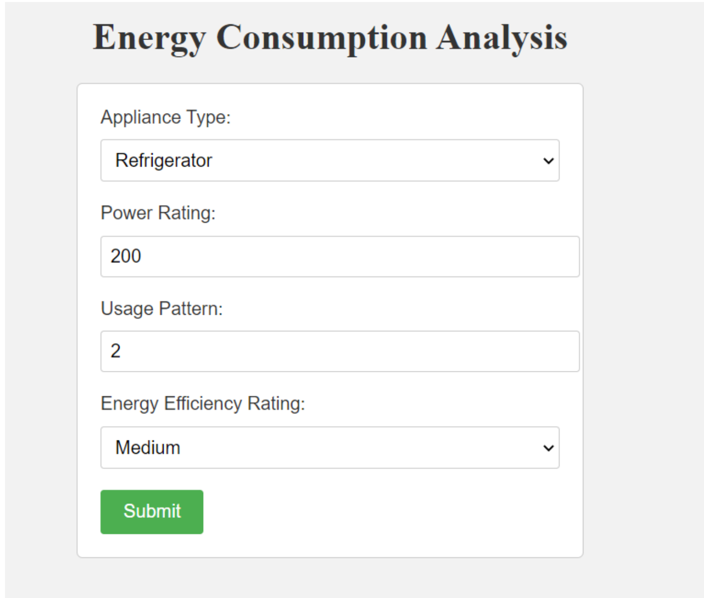
6. RESULTS

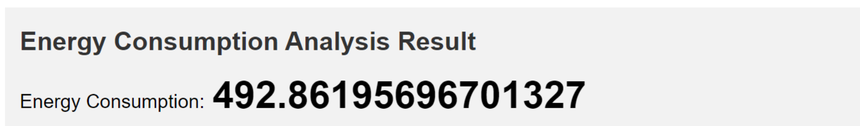
6.1 Performance Metrics

The model metrics in the above project evaluate the performance of two regression models: Linear Regression and Random Forest Regression. The metrics provide insights into how well the models are able to predict the target variable (Efficiency) based on the features (Appliance).

The following metrics are calculated for each model:

- **Energy Efficiency Rating**
- **Energy Consumption Value**





7. ADVANTAGES & DISADVANTAGES

Advantages:

- Energy Efficiency: By analyzing energy consumption patterns and providing recommendations for reducing waste, the system can help homeowners and utility companies improve energy efficiency, leading to cost savings and reduced environmental impact.
- Cost Reduction: By identifying areas of high energy consumption and suggesting strategies to optimize energy usage, the system can help homeowners and utility companies lower their energy bills and operating costs.
- Sustainability: The system promotes sustainable energy usage by encouraging efficient energy management practices and reducing overall energy waste.
- Data-Driven Insights: Through machine learning algorithms and data analysis, the system can provide valuable insights into energy consumption patterns and trends, enabling informed decision-making for

energy management.

- User-Friendly Interface: With a user-friendly interface, the system makes it easy for homeowners and utility company personnel to access and interpret energy consumption data, recommendations, and insights.
- Scalability: The system can handle a large volume of data from multiple residential buildings, accommodating growth in the number of users and data sources.

Disadvantages:

- Data Dependency: The accuracy and effectiveness of the system heavily rely on the availability and quality of data related to energy consumption, weather conditions, occupancy, etc. Inaccurate or insufficient data may impact the system's ability to provide reliable predictions and recommendations.
- Initial Implementation and Integration: Implementing the system and integrating it with existing energy monitoring systems or devices in residential buildings may require time, effort, and potentially additional costs.
- User Adoption and Engagement: Encouraging homeowners and utility company personnel to actively engage with the system and follow its recommendations may present a challenge. Motivating behavior change and ensuring user compliance can be difficult.
- Privacy Concerns: The system collects and analyzes data on energy consumption, which may include personal information. Ensuring data privacy and complying with relevant regulations and privacy policies is essential to maintain user trust.
- Model Accuracy Limitations: Although machine learning models can provide accurate predictions, there may be inherent limitations or uncertainties in predicting complex energy consumption patterns. Unforeseen variables or unusual events may impact the accuracy of predictions.

8. CONCLUSION

The power consumption analysis system using machine learning algorithms offers significant benefits in managing energy usage, reducing waste, and lowering costs for residential buildings.

By collecting and analyzing data on energy consumption, weather conditions, occupancy, and other relevant factors, the system can accurately predict energy consumption patterns and provide actionable insights and recommendations. This promotes energy efficiency, sustainability, and cost savings for homeowners and utility companies. The user-friendly interface facilitates easy access to data and recommendations, enhancing user engagement.

9. FUTURE SCOPE

Integration of Real-Time Data: Incorporating real-time data, such as energy market prices or renewable energy availability, can enhance the accuracy of predictions and enable users to make more informed decisions.

Advanced Analytics Techniques: Exploring advanced analytics techniques, such as anomaly detection or clustering algorithms, can help identify abnormal energy usage patterns and provide targeted recommendations for energy optimization.

Smart Home Integration: Integrating the system with smart home devices and automation systems can enable real-time control and optimization of energy consumption based on predictive models and user preferences.

Demand Response Programs: Collaborating with utility companies to support demand response programs can help homeowners actively manage their energy usage during peak demand periods, further reducing costs and enhancing grid stability.

10. APPENDIX

10 (a) Source Code

10 (b) GitHub & Project Video Demo Link

10 (a) **Source Code:**

Folder structure:

app.py
averall.py
energy_analysis.py
- data
    - -data.csv
    - -appliance.csv

- static
  (images)
- templates
  -index.html
  -eca.html
  -login.html
  -analysis.html

Homepage (eca.html)

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Energy Consumption Analysis</title>
   <script>
      function mylink1(){
         window.location.href="analysis.html";
      }
      function mylink2(){
         window.location.href="login.html";
      }
   </script>
   <style>
      table{
  border: 1px;
  border-radius: 10px;
  margin-left: auto;
  margin-right: auto;
}
body {
  background-image: url('ecaimg.webp');
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-size: cover;
}
tr {
 border: 1px;
```

```
 border-radius: 10px;
 height: 200px;
}
td {
  border: 1px;
  border-radius: 10px;
  width: 33.3%;
}
input[type="submit"] {
        width: 90%;
        height: 180px;
        background-color: #1E5128;
        color: white;
        padding: 14px 20px;
        margin: 8px 0;
        border: 1px solid;
        border-color: #1E5128;
        border-radius: 10px;
        cursor: pointer;
        font-weight: bold; font-family: 'Gill Sans'; font-size:30px;
      }
      input[type="submit"]:hover {
        background-color: #4E9F3D;
      }
    </style>
</head>
<body >
   <h1 align="center" style="color:#191A19; font-size: 50px;">Energy Consumption
Analysis</h1>
   <div>
    <table align="left">
        <tr>
            <td><input type="submit" value="Overall Data Analysis" name="Overall"
onclick="mylink1()"></td>
            <td align="right"><input type="submit" value="Individual Data Analysis"
name="Individual" onclick="mylink2()"></td>
        </tr>
    </table>
   </div>
</body>
</html>
```

Login.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Energy-Login</title>
  <script>
    function validateForm(){
      var x = document.forms["loginForm"]["uname"].value;
      var p = document.forms["loginForm"]["pwd"].value;
      if (
     x == "" &&
     p == ""
    ) {
     alert("Enter Username and Password");
     return false;
    } else if (x == "") {
     alert("Username not entered");
     return false;
    } else if (p == "") {
     alert("Password not entered");
     return false;
    }


  </script>
  <style>
    body {
 background-image: url('log.jpeg');
 background-repeat: no-repeat;
 background-attachment: fixed;
 background-size: cover;
}
   input[type="text"],
   select {
    width: 100%;
    padding: 12px 20px;
    margin: 8px 0;
```

```css
  display: inline-block;
  border: 1px solid #ccc;
  border-radius: 4px;
  box-sizing: border-box;
}
input[type="password"],
select {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  display: inline-block;
  border: 1px solid #ccc;
  border-radius: 4px;
  box-sizing: border-box;
}
input[type="submit"] {
  width: 50%;
  background-color: #05BFDB;
  color: white;
  padding: 14px 20px;
  margin: 8px 0;
  border: none;
  border-radius: 4px;
  cursor: pointer;
}
input[type="submit"]:hover {
  background-color: #00FFCA;
}
input[type="reset"] {
  width: 50%;
  background-color: #05BFDB;
  color: white;
  padding: 14px 20px;
  margin: 8px 0;
  border: none;
  border-radius: 4px;
  cursor: pointer;
}
input[type="reset"]:hover {
  background-color: #00FFCA;
}
div {
```

```
        border-radius: 5px;

        padding: 20px;
        height: 100%;
      }
      table{
  border: 1px;
  border-radius: 10px;

  width: 40%;
}
td {
  border: 1px;
  border-radius: 10px;
  width: 50%;
}
    </style>
</head>
<body style="background-color: #FFA3FD;">
  <div>   <h2 align="right"><a href="eca.html" style="color: rgb(234, 223,
223);">EXIT</a></h2>
  </div>
    <form name="loginForm" action="index.html" method="get">
        <div>
            <table align="left">
                <tr>
                    <th align="center" style="color: #31E1F7; font-size: 50px;"
colspan="2">Individual User Analysis</th>
                </tr>
                <tr>
                <th align="center" style="color: #19A7CE;" colspan="2"><h2>Login
Page</h2></th>
                </tr>
                <tr>
                    <td align="center" style="font-family: 'Gill Sans'; color: #97DEFF;font-weight:
bolder;">USERNAME:</td>
                    <td><input type="text" name="uname"></td>
                </tr>
                <tr>
                    <td align="center" style="font-family: 'Gill Sans'; color: #97DEFF;font-weight:
bolder;">PASSWORD:</td>
                    <td><input type="password" name="pwd"></td>
```

```
            </tr>
            <tr>
                <td style="font-weight: bolder;" align="center" colspan="2"><input
type="submit" value="LOGIN" onclick=" return validateForm()"></td>
            </tr>
            <tr>
                <td align="center" colspan="2"><input type="reset" value="RESET"></td>
            </tr>
        </table>
    </div>
  </form>
</body>
</html>
```

<u>Form.html</u>

```
<html>
<head>
  <title>Energy Consumption Analysis</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f2f2f2;
      margin: 0;
      padding: 20px;
    }

    h1 {
      color: #333333;
    }

    form {
      max-width: 400px;
      margin: 0 auto;
      background-color: #ffffff;
      padding: 20px;
      border: 1px solid #cccccc;
      border-radius: 5px;
    }
```

```
        label {
            display: block;
            margin-bottom: 10px;
            color: #333333;
        }

        select,
        input[type="number"] {
            width: 100%;
            padding: 8px;
            font-size: 16px;
            border-radius: 3px;
            border: 1px solid #cccccc;
        }

        input[type="submit"] {
            background-color: #4CAF50;
            color: white;
            border: none;
            padding: 10px 20px;
            font-size: 16px;
            cursor: pointer;
            border-radius: 3px;
        }

        input[type="submit"]:hover {
            background-color: #45a049;
        }
    </style>
</head>
<body>
    <div>   <h3 align="right"><a href="eca.html" style="color:
#000000;">LOGOUT</a></h3> </div>
    <h1 align="center" style="font-family: 'Times New Roman', Times, serif;">Energy
Consumption Analysis</h1>
    <form action="/predict" method="POST">
        <label for="appliance_type">Appliance Type:</label>
        <select name="appliance_type" required>
            <option value="Refrigerator">Refrigerator</option>
            <option value="Air Conditioner">Air Conditioner</option>
            <option value="Washing Machine">Washing Machine</option>
            <!-- Add more appliance types here -->
```

```
        </select><br><br>
        <label for="power_rating">Power Rating:</label>
        <input type="number" name="power_rating" required><br><br>
        <label for="usage_pattern">Usage Pattern:</label>
        <input type="number" name="usage_pattern" required><br><br>
        <label for="energy_efficiency_rating">Energy Efficiency Rating:</label>
        <select name="energy_efficiency_rating" required>
            <option value="1">Low</option>
            <option value="2">Medium</option>
            <option value="3">High</option>
        </select><br><br>
        <input type="submit" value="Submit">
    </form>
</body>
</html>
```

App.py

```
from flask import Flask, render_template, request
import pandas as pd
import numpy as np
import pdfkit
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

app = Flask(__name__)

# Home route
@app.route('/')
def index():
    return render_template('index.html')

@app.route('/predict', methods=['POST'])
def predict():
    # Extract user input from the form
    power_rating = float(request.form['power_rating'])
    usage_pattern = float(request.form['usage_pattern'])
    energy_efficiency_rating = int(request.form['energy_efficiency_rating'])

    # Generate energy consumption prediction and plot
```

```
    energy_consumption = predict_energy_consumption(power_rating, usage_pattern,
energy_efficiency_rating)
    plot_path = visualize_energy_consumption(power_rating, usage_pattern,
energy_efficiency_rating)

    # Generate suggestions using OpenAI based on the plot
    suggestions = generate_suggestions(power_rating, usage_pattern,
energy_efficiency_rating)

    # Render the result page with energy consumption, plot, and suggestions
    return render_template('result.html', energy_consumption=energy_consumption,
plot_path=plot_path, suggestions=suggestions)


# Function to predict energy consumption
def predict_energy_consumption(power_rating, usage_pattern, energy_efficiency_rating):
    data = pd.read_csv('data/appliance.csv')
    X = data[['power_rating', 'usage_pattern', 'energy_efficiency_rating']]
    y = data['energy_consumption']

    model = LinearRegression()
    model.fit(X, y)

    input_data = pd.DataFrame({'power_rating': [power_rating], 'usage_pattern':
[usage_pattern], 'energy_efficiency_rating': [energy_efficiency_rating]})
    prediction = model.predict(input_data)

    return prediction[0]

# Future trends route
@app.route('/future-trends')
def future_trends():
    data = pd.read_csv('data/appliance.csv')
    x = data['power_rating']
    y = data['energy_consumption']

    # Perform linear regression
    model = LinearRegression()
    model.fit(x.values.reshape(-1, 1), y)

    # Predict future trends
    future_x = np.linspace(min(x), max(x), num=100)
```

```
    future_y = model.predict(future_x.reshape(-1, 1))

    # Plot the future trends
    plt.figure()
    plt.plot(x, y, label='Actual')
    plt.plot(future_x, future_y, color='green', linestyle='--', label='Future Trend')
    plt.xlabel('Power Rating')
    plt.ylabel('Energy Consumption')
    plt.title('Future Energy Consumption Trends')
    plt.legend()

    # Save the plot
    plot_path = 'static/future_trends.png'
    plt.savefig(plot_path)
    plt.close()

    return render_template('future_trends.html', plot_path=plot_path)
def generate_suggestions(power_rating, usage_pattern, energy_efficiency_rating):
    suggestions = []

    # Check power rating
    if power_rating > 200:
        suggestions.append("Consider using appliances with lower power ratings.")
    else:
        suggestions.append("Ensure your appliances are energy-efficient.")

    # Check usage pattern
    if usage_pattern < 1.0:
        suggestions.append("Optimize your usage pattern to avoid unnecessary energy
consumption.")
    elif usage_pattern > 2.0:
        suggestions.append("Adjust your usage pattern to reduce energy consumption
during peak hours.")

    # Check energy efficiency rating
    if energy_efficiency_rating == 1:
        suggestions.append("Upgrade to appliances with higher energy efficiency ratings.")
    elif energy_efficiency_rating == 2:
        suggestions.append("Ensure regular maintenance of your appliances to maximize
energy efficiency.")

    return suggestions
```

```
# Function to visualize energy consumption
def visualize_energy_consumption(power_rating, usage_pattern,
energy_efficiency_rating):
    data = pd.read_csv('data/appliance.csv')
    x = data['power_rating']
    y = data['energy_consumption']

    fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(10, 5))

    # Plot the data points and linear regression
    ax1.scatter(x, y, label='Actual')
    ax1.scatter(power_rating, predict_energy_consumption(power_rating, usage_pattern,
energy_efficiency_rating),
                color='red', label='User Input')
    model = LinearRegression()
    model.fit(x.values.reshape(-1, 1), y)
    ax1.plot(x, model.predict(x.values.reshape(-1, 1)), color='orange', label='Linear
Regression')
    ax1.set_xlabel('Power Rating')
    ax1.set_ylabel('Energy Consumption')
    ax1.set_title('Energy Consumption Analysis')
    ax1.legend()

    # Plot the energy consumption trend
    trend_data = pd.read_csv('data/trend.csv')
    trend_x = trend_data['day']
    trend_y = trend_data['energy_consumption']
    ax2.plot(trend_x, trend_y, marker='o')
    ax2.set_xlabel('Day')
    ax2.set_ylabel('Energy Consumption')
    ax2.set_title('Energy Consumption Trend')
    ax2.grid(True)
    plt.xticks(rotation=45)
    plt.tight_layout()

    # Save the plot
    plot_path = 'static/plot.png'
    plt.savefig(plot_path)
    plt.close()
```

```python
    return plot_path

if __name__ == '__main__':
    app.run(debug=True)
```

Overall.py

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.ensemble import RandomForestRegressor
from sklearn.preprocessing import LabelEncoder

# Load the data
data = pd.read_csv("data/data.csv")

# Define valid columns
valid_columns = ["Efficiency", "Appliance"]

# Check if valid columns exist in the dataset
missing_columns = [column for column in valid_columns if column not in data.columns]
if missing_columns:
    raise ValueError(f"Columns {missing_columns} not found in the dataset.")

# Select only valid columns
data = data[valid_columns]

# Drop rows with missing values if any
data = data.dropna()

# Encode categorical variables using LabelEncoder
label_encoder = LabelEncoder()
categorical_columns = ["Efficiency", "Appliance"]
for column in categorical_columns:
    data[column] = label_encoder.fit_transform(data[column])

# Bivariate Analysis: Scatter plot of Efficiency vs. Appliance
plt.scatter(data["Efficiency"], data["Appliance"])
plt.xlabel("Efficiency")
plt.ylabel("Appliance")
plt.title("Bivariate Analysis: Efficiency vs. Appliance")
plt.savefig("templates/bivariate.png")  # Save the plot as an image
```

```
# Univariate Analysis: Histogram of Appliance
plt.hist(data["Appliance"])
plt.xlabel("Appliance")
plt.ylabel("Frequency")
plt.title("Univariate Analysis: Appliance")
plt.savefig("templates/univariate.png")  # Save the plot as an image

# Multivariate Analysis: Correlation Matrix Heatmap
correlation_matrix = data.corr()
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap="RdYlGn")
plt.title("Multivariate Analysis: Correlation Matrix")
plt.savefig("templates/multivariate.png")  # Save the plot as an image

# Random Forest Analysis
X = data.drop(["Appliance"], axis=1)  # Features
y = data["Appliance"]  # Target variable

# Create and fit the random forest model
rf = RandomForestRegressor(n_estimators=100, random_state=42)
rf.fit(X, y)

# Feature Importance
importance = rf.feature_importances_
feature_names = X.columns

# Plotting feature importances
plt.barh(range(len(importance)), importance, align="center")
plt.yticks(range(len(feature_names)), feature_names)
plt.xlabel("Feature Importance")
plt.ylabel("Features")
plt.title("Random Forest: Feature Importance")
plt.savefig("templates/feature_importance.png")  # Save the plot as an image

# Generate HTML file (same code as before)
html_content = f'''
<html>
<head>
   <title>Data Analysis Plots</title>
</head>
<body>
```

```
    <h1>Bivariate Analysis: Efficiency vs. Appliance</h1>
    <img src="templates/bivariate.png" alt="Bivariate Analysis">

    <h1>Univariate Analysis: Appliance</h1>
    <img src="templates/univariate.png" alt="Univariate Analysis">

    <h1>Multivariate Analysis: Correlation Matrix</h1>
    <img src="templates/multivariate.png" alt="Multivariate Analysis">

    <h1>Random Forest: Feature Importance</h1>
    <img src="templates/feature_importance.png" alt="Feature Importance">
</body>
</html>
'''

# Save HTML file (same code as before)
with open("templates/analysis.html", "w") as file:
    file.write(html_content)
```

Future.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Future Energy Consumption Trends</title>
</head>
<body>
    <h1>Future Energy Consumption Trends</h1>
    <img src="{{ plot_path }}" alt="Future Trends Plot">
</body>
</html>
```

Overall.py

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.ensemble import RandomForestRegressor
```

```
from sklearn.preprocessing import LabelEncoder

# Load the data
data = pd.read_csv("data/data.csv")

# Define valid columns
valid_columns = ["Efficiency", "Appliance"]

# Check if valid columns exist in the dataset
missing_columns = [column for column in valid_columns if column not in data.columns]
if missing_columns:
    raise ValueError(f"Columns {missing_columns} not found in the dataset.")

# Select only valid columns
data = data[valid_columns]

# Drop rows with missing values if any
data = data.dropna()

# Encode categorical variables using LabelEncoder
label_encoder = LabelEncoder()
categorical_columns = ["Efficiency", "Appliance"]
for column in categorical_columns:
    data[column] = label_encoder.fit_transform(data[column])

# Bivariate Analysis: Scatter plot of Efficiency vs. Appliance
plt.scatter(data["Efficiency"], data["Appliance"])
plt.xlabel("Efficiency")
plt.ylabel("Appliance")
plt.title("Bivariate Analysis: Efficiency vs. Appliance")
plt.savefig("templates/bivariate.png")  # Save the plot as an image

# Univariate Analysis: Histogram of Appliance
plt.hist(data["Appliance"])
plt.xlabel("Appliance")
plt.ylabel("Frequency")
plt.title("Univariate Analysis: Appliance")
plt.savefig("templates/univariate.png")  # Save the plot as an image

# Multivariate Analysis: Correlation Matrix Heatmap
correlation_matrix = data.corr()
plt.figure(figsize=(8, 6))
```

```
sns.heatmap(correlation_matrix, annot=True, cmap="RdYlGn")
plt.title("Multivariate Analysis: Correlation Matrix")
plt.savefig("templates/multivariate.png")  # Save the plot as an image

# Random Forest Analysis
X = data.drop(["Appliance"], axis=1)  # Features
y = data["Appliance"]  # Target variable

# Create and fit the random forest model
rf = RandomForestRegressor(n_estimators=100, random_state=42)
rf.fit(X, y)

# Feature Importance
importance = rf.feature_importances_
feature_names = X.columns

# Plotting feature importances
plt.barh(range(len(importance)), importance, align="center")
plt.yticks(range(len(feature_names)), feature_names)
plt.xlabel("Feature Importance")
plt.ylabel("Features")
plt.title("Random Forest: Feature Importance")
plt.savefig("templates/feature_importance.png")  # Save the plot as an image

# Generate HTML file (same code as before)
html_content = f'''
<html>
<head>
    <title>Data Analysis Plots</title>
</head>
<body>
    <h1>Bivariate Analysis: Efficiency vs. Appliance</h1>
    <img src="templates/bivariate.png" alt="Bivariate Analysis">

    <h1>Univariate Analysis: Appliance</h1>
    <img src="templates/univariate.png" alt="Univariate Analysis">

    <h1>Multivariate Analysis: Correlation Matrix</h1>
    <img src="templates/multivariate.png" alt="Multivariate Analysis">

    <h1>Random Forest: Feature Importance</h1>
    <img src="templates/feature_importance.png" alt="Feature Importance">
```

```
</body>
</html>
'''

# Save HTML file (same code as before)
with open("templates/analysis.html", "w") as file:
    file.write(html_content)
```

analysis.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Data Analysis Plots</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f2f2f2;
            margin: 0;
            padding: 20px;
        }

        h1 {
            font-size: 24px;
            color: #333333;
            margin-top: 30px;
        }

        img {
            display: block;
            margin: 20px auto;
            max-width: 800px;
            border: 1px solid #cccccc;
            box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
        }

        .container {
            margin-bottom: 40px;
            padding-bottom: 20px;
            border-bottom: 1px solid #cccccc;
        }
```

```
        .text-area {
          width: 100%;
          height: 100px;
          resize: none;
          border: 1px solid #cccccc;
          padding: 10px;
        }

        .button {
          background-color: #4CAF50;
          border: none;
          color: white;
          padding: 10px 20px;
          text-align: center;
          text-decoration: none;
          display: inline-block;
          font-size: 16px;
          margin-top: 10px;
          cursor: pointer;
        }

        .button:hover {
          background-color: #45a049;
        }
    </style>
</head>
<body>
    <div>   <h3 align="right"><a href="eca.html" style="color: #000000;">EXIT</a></h3>
</div>
    <div class="container">
        <h1>Bivariate Analysis: Efficiency vs. Appliance</h1>
        <img id="bivariate-img" src="bivariate.png" alt="Bivariate Analysis">
        <textarea id="bivariate-text" class="text-area" readonly>The scatter plot above
illustrates the relationship between "Efficiency" and "Appliance." Each data point
represents a specific combination of efficiency and appliance.
            The horizontal axis represents the efficiency values, while the vertical axis
represents the appliance values.</textarea>
        <button id="bivariate-button" class="button" onclick="toggleText('bivariate')">Toggle
Text</button>
    </div>
```

```
    <div class="container">
        <h1>Univariate Analysis: Appliance</h1>
        <img id="univariate-img" src="univariate.png" alt="Univariate Analysis">
        <textarea id="univariate-text" class="text-area" readonly>The histogram above
represents the distribution of the "Appliance" variable. The x-axis displays the different
categories or types of appliances, while the y-axis represents the frequency or count of
each category.</textarea>
        <button id="univariate-button" class="button"
onclick="toggleText('univariate')">Toggle Text</button>
    </div>

    <div class="container">
        <h1>Multivariate Analysis: Correlation Matrix</h1>
        <img id="multivariate-img" src="multivariate.png" alt="Multivariate Analysis">
        <textarea id="multivariate-text" class="text-area" readonly>The correlation matrix
heatmap provides visual representation of the correlations between different variables in
the dataset. The heatmap uses a color scale to indicate the strength and direction of the
correlations. Positive correlations are shown in shades of green, negative correlations in
shades of red, and no or weak correlations in shades of yellow. The numerical values in
each cell of the heatmap provide additional information about the strength of the
correlations.</textarea>
        <button id="multivariate-button" class="button"
onclick="toggleText('multivariate')">Toggle Text</button>
    </div>

    <div class="container">
        <h1>Random Forest: Feature Importance</h1>
        <img id="feature-importance-img" src="feature_importance.png" alt="Feature
Importance">
        <textarea id="feature-importance-text" class="text-area" readonly>The code
provided generates a horizontal bar chart to visualize the feature importances in a
random forest model. The feature importances represent the relative importance of each
feature in predicting the target variable.

The y-axis of the chart represents the different features, while the x-axis represents the
corresponding feature importance values. Each bar in the chart represents a feature, and
its length corresponds to its importance value.</textarea>
        <button id="feature-importance-button" class="button"
onclick="toggleText('feature-importance')">Toggle Text</button>
    </div>

    <script>
```

```
        function toggleText(id) {
            var textArea = document.getElementById(id + '-text');
            var button = document.getElementById(id + '-button');

            if (textArea.style.display === 'none') {
                textArea.style.display = 'block';
                button.innerHTML = 'Hide Text';
            } else {
                textArea.style.display = 'none';
                button.innerHTML = 'Show Text';
            }
        }
    </script>
</body>
</html>
```

Data.csv

```
Name,Phone,City,Weather,Appliance,Efficiency,Use Rate
Jane Smith,9876543210,Los Angeles,Cloudy,Air Conditioner,3,8
David Johnson,5551234567,Chicago,Rainy,Washing Machine,2,5
Sarah Williams,7775551234,Houston,Sunny,Oven,4,9
Michael Brown,2223334444,Phoenix,Hot,Television,5,7
Emily Davis,4445556666,San Francisco,Foggy,Light Bulb,1,3
Daniel Taylor,8889990000,Boston,Windy,Computer,3,6
Olivia Anderson,1112223333,Dallas,Sunny,Microwave,2,4
John Smith,555-1234,New York,Sunny,Refrigerator,4,10
Emily Johnson,555-5678,Los Angeles,Cloudy,Air Conditioner,3,8
Michael Williams,555-9012,Chicago,Rainy,Washing Machine,2,5
Emma Brown,555-3456,Houston,Sunny,Oven,4,9
Daniel Davis,555-7890,Phoenix,Hot,Television,5,7
Olivia Taylor,555-2345,San Francisco,Foggy,Light Bulb,1,3
Sophia Anderson,555-6789,Boston,Windy,Computer,3,6
Aiden Martin,555-0123,Dallas,Sunny,Microwave,2,4
John Smith,555-1234,New York,Sunny,Refrigerator,4,10
Emily Johnson,555-5678,Los Angeles,Cloudy,Air Conditioner,3,8
Michael Williams,555-9012,Chicago,Rainy,Washing Machine,2,5
Emma Brown,555-3456,Houston,Sunny,Oven,4,9
Daniel Davis,555-7890,Phoenix,Hot,Television,5,7
Olivia Taylor,555-2345,San Francisco,Foggy,Light Bulb,1,3
Sophia Anderson,555-6789,Boston,Windy,Computer,3,6
```

Aiden Martin,555-0123,Dallas,Sunny,Microwave,2,4
Sophie Wilson,555-7890,New York,Cloudy,Refrigerator,4,10
Benjamin Thompson,555-2345,Los Angeles,Sunny,Air Conditioner,3,8
John Smith,555-1234,New York,Sunny,Refrigerator,4,10
Emily Johnson,555-5678,Los Angeles,Cloudy,Air Conditioner,3,8
Michael Williams,555-9012,Chicago,Rainy,Washing Machine,2,5
Emma Brown,555-3456,Houston,Sunny,Oven,4,9
Daniel Davis,555-7890,Phoenix,Hot,Television,5,7
Olivia Taylor,555-2345,San Francisco,Foggy,Light Bulb,1,3
Sophia Anderson,555-6789,Boston,Windy,Computer,3,6
Aiden Martin,555-0123,Dallas,Sunny,Microwave,2,4
Sophie Wilson,555-7890,New York,Cloudy,Refrigerator,4,10
Benjamin Thompson,555-2345,Los Angeles,Sunny,Air Conditioner,3,8
Alice Davis,555-5678,Chicago,Rainy,Washing Machine,2,5
James Johnson,555-9012,Houston,Sunny,Oven,4,9
Liam Wilson,555-3456,Phoenix,Hot,Television,5,7
Ava Taylor,555-6789,Austin,Windy,Blender,1,4
Alexander Anderson,555-0123,San Francisco,Sunny,Microwave,4,10
Ella Martin,555-7890,New York,Cloudy,Refrigerator,3,8
William Thompson,555-2345,Los Angeles,Sunny,Air Conditioner,2,5
Sophia Davis,555-5678,Chicago,Rainy,Washing Machine,4,9
Michael Johnson,555-9012,Houston,Sunny,Oven,5,7
Charlotte Wilson,555-3456,Phoenix,Hot,Television,1,3
Noah Taylor,555-6789,Boston,Windy,Computer,4,6
Mia Anderson,555-0123,Dallas,Sunny,Microwave,2,4
Jennifer Adams,555-1234,New York,Sunny,Refrigerator,4,10
David Anderson,555-5678,Los Angeles,Cloudy,Air Conditioner,3,8
Sarah Bennett,555-9012,Chicago,Rainy,Washing Machine,2,5
Matthew Campbell,555-3456,Houston,Snowy,Oven,4,9
Jessica Carter,555-7890,Phoenix,Hot,Television,5,7
Ryan Clark,555-2345,San Francisco,Foggy,Light Bulb,1,3
Emily Collins,555-6789,Boston,Windy,Computer,3,6
Nathan Cooper,555-0123,Dallas,Sunny,Microwave,2,4
Victoria Davis,555-8901,Miami,Partly Cloudy,Dishwasher,4,8
Oliver Evans,555-4567,Seattle,Rainy,Coffee Maker,3,5
Sophia Foster,555-7890,Denver,Snowy,Toaster,2,3
Daniel Garcia,555-2345,San Diego,Sunny,Fan,4,7
Amelia Gonzalez,555-6789,Portland,Foggy,Heater,5,9
Henry Gray,555-0123,Austin,Windy,Blender,1,4
Isabella Green,555-7890,San Francisco,Sunny,Microwave,4,10
Ethan Hall,555-3456,New York,Cloudy,Refrigerator,3,8
Charlotte Harris,555-6789,Los Angeles,Sunny,Air Conditioner,2,5

Michael Hill,555-9012,Chicago,Rainy,Washing Machine,4,9
Sophia Hughes,555-2345,Houston,Sunny,Oven,5,7
Emily Jackson,555-6789,Phoenix,Hot,Television,1,3
Ava James,555-0123,Boston,Windy,Computer,4,6
Logan Johnson,555-7890,Dallas,Sunny,Microwave,2,4
Grace Jones,555-3456,Miami,Partly Cloudy,Dishwasher,4,8
Daniel King,555-6789,Seattle,Rainy,Coffee Maker,3,5
Sophia Lee,555-0123,Denver,Snowy,Toaster,2,3
Jacob Lewis,555-7890,San Diego,Sunny,Fan,4,7
Olivia Lopez,555-2345,Portland,Foggy,Heater,5,9
Noah Martinez,555-6789,Austin,Windy,Blender,1,4
Liam Miller,555-0123,San Francisco,Sunny,Microwave,4,10
Sophia Adams,555-1234,New York,Sunny,Refrigerator,4,10
Ethan Anderson,555-5678,Los Angeles,Cloudy,Air Conditioner,3,8
Ava Bennett,555-9012,Chicago,Rainy,Washing Machine,2,5
Noah Campbell,555-3456,Houston,Snowy,Oven,4,9
Olivia Carter,555-7890,Phoenix,Hot,Television,5,7
Liam Clark,555-2345,San Francisco,Foggy,Light Bulb,1,3
Isabella Collins,555-6789,Boston,Windy,Computer,3,6
Mia Cooper,555-0123,Dallas,Sunny,Microwave,2,4
Charlotte Davis,555-8901,Miami,Partly Cloudy,Dishwasher,4,8
Lucas Evans,555-4567,Seattle,Rainy,Coffee Maker,3,5
Amelia Foster,555-7890,Denver,Snowy,Toaster,2,3
Oliver Garcia,555-2345,San Diego,Sunny,Fan,4,7
Sophia Gonzalez,555-6789,Portland,Foggy,Heater,5,9
Aiden Gray,555-0123,Austin,Windy,Blender,1,4
Emma Green,555-7890,San Francisco,Sunny,Microwave,4,10
Jackson Hall,555-3456,New York,Cloudy,Refrigerator,3,8
Ava Harris,555-6789,Los Angeles,Sunny,Air Conditioner,2,5
Carter Hill,555-9012,Chicago,Rainy,Washing Machine,4,9
Harper Hughes,555-2345,Houston,Sunny,Oven,5,7
Ella Jackson,555-6789,Phoenix,Hot,Television,1,3
Leo James,555-0123,Boston,Windy,Computer,4,6
Mila Johnson,555-7890,Dallas,Sunny,Microwave,2,4
Henry Jones,555-3456,Miami,Partly Cloudy,Dishwasher,4,8
Alexander King,555-6789,Seattle,Rainy,Coffee Maker,3,5
Luna Lee,555-0123,Denver,Snowy,Toaster,2,3
Jack Lewis,555-7890,San Diego,Sunny,Fan,4,7
Sophia Lopez,555-2345,Portland,Foggy,Heater,5,9
Lucy Martinez,555-6789,Austin,Windy,Blender,1,4
Logan Miller,555-0123,San Francisco,Sunny,Microwave,4,10

trend.csv

day,energy_consumption
2023-05-01,100
2023-05-02,120
2023-05-03,110
2023-05-04,130
2023-05-05,115
2023-05-06,105
2023-05-07,125
2023-05-08,135
2023-05-09,140
2023-05-10,150
2023-05-11,145
2023-05-12,155
2023-05-13,165
2023-05-14,160
2023-05-15,170
2023-05-16,160
2023-05-17,180
2023-05-18,175
2023-05-19,195

Code github link:

Demo video link: https://www.youtube.com/watch?v=FkLLLJbjNMQ