Project Development Phase
Functional features- Data Analytics

Date	20 May 2023
Team ID	NM2023TMID17415
Project Name	Project on A Reliable Energy Consumption Analysis System for Energy-Efficient Appliances

Web Application Framework: The code is written using the Flask framework, which provides a simple and efficient way to build web applications in Python.

User Input Processing: The code utilizes the request object from Flask to extract user input from an HTML form. The user inputs are then used for further calculations and analysis.

Calculation of Overall Efficiency: The calculate_overall_efficiency function takes user inputs such as power rating, usage pattern, energy efficiency rating, and standby power to calculate the overall efficiency of an energy system. The function uses a weighted formula to combine these inputs into a single efficiency score.

Univariate Analysis: The perform_univariate_analysis function generates a histogram plot to perform univariate analysis on the overall efficiency scores. It demonstrates the distribution and frequency of efficiency scores.

Bivariate Analysis: The perform_bivariate_analysis function creates a scatter plot to analyze the relationship between power rating and overall efficiency. It visualizes how these two variables are related to each other.

Multivariate Analysis: The perform_multivariate_analysis function generates a 3D scatter plot to analyze the relationship between power rating, usage pattern, and

overall efficiency. It allows for the visualization of the interaction between multiple variables.

Model Training and Prediction: The code does not explicitly include a model training and prediction step, but it is possible to incorporate machine learning models, such as the Linear Regression model from scikit-learn, to train and predict energy efficiency based on the provided inputs.

Visualization: The code uses the matplotlib library to generate various plots, such as histograms, scatter plots, and 3D plots, for visualizing the analysis results. These plots provide a visual representation of the data and help in understanding the relationships and patterns.

HTML Template Rendering: The Flask render_template function is used to render HTML templates for the home page and the result page. These templates define the structure and content of the web pages that are displayed to the user.

Routing and Request Handling: The code defines different routes using the @app.route decorator. It handles HTTP requests to different routes, such as the home route ("/") and the predict route ("/predict"), and executes the corresponding functions.

User Interface: The code uses HTML templates to provide a user interface where users can input values and view the analysis results. It allows users to interact with the application and obtain insights about energy efficiency based on their inputs.

Deployment: The code includes a section that allows the Flask application to run in debug mode using the app.run() method. This enables the application to be deployed and accessed through a web browser.