**GRT INSTITUTE OF ENGINEERING AND TECHNOLOGY**

**PROJECT NAME: MEASURE ENERGY CONSUMPTION**

**NAME: B. RUBINI**

**GMAIL ID:** [**rubini2134@gmail.com**](mailto:rubini2134@gmail.com)

**PHASE :3 DEVELOPMENT PART1**

**ABSTRACT**:

To measure the total energy consumption of a house in Python, you'll need data sources such as utility bills or energy monitoring devices. Here's a basic example of how you can calculate and track energy consumption using Python:

**Formula:** E=p\*(t/1000)

P=power

T=Time

E=Energy ( kwh)

Measuring energy consumption in a house using Python typically involves working with energy data sources, such as smart meters, and performing data analysis. Here’s a simplified example of how you can get started:

1. \*\***Data Collection**\*\*: First, you need to collect data from your energy source, such as a smart meter or other sensors. This data can be in the form of time-series energy usage readings.
2. \*\***Data Analysis\***\*: Use Python libraries like Pandas for data analysis. You can load your energy consumption data into a Pandas DataFrame and perform various calculations and analyses.
3. \*\***Visualization**\*\*: Matplotlib or Seaborn can help you create visualizations of your energy consumption data, such as line charts or histograms to better understand patterns.
4. \*\***Calculations**\*\*: You can calculate various metrics like daily, weekly, or monthly energy consumption, peak usage, and average consumption

**Python program**:

# Create a list to store monthly energy consumption data in kWh

monthly\_consumption = [500, 600, 700, 550, 750, 800, 700, 600, 650, 700, 550, 600]

# Calculate the total annual energy consumption

total\_consumption = sum(monthly\_consumption)

# Calculate the average monthly consumption

average\_monthly\_consumption = total\_consumption / len(monthly\_consumption)

# Calculate the maximum monthly consumption

max\_monthly\_consumption = max(monthly\_consumption)

# Calculate the minimum monthly consumption

min\_monthly\_consumption = min(monthly\_consumption)

# Print the results

print(f"Total Annual Energy Consumption: {total\_consumption} kWh")

print(f"Average Monthly Energy Consumption: {average\_monthly\_consumption} kWh")

print(f"Maximum Monthly Energy Consumption: {max\_monthly\_consumption} kWh")

print(f"Minimum Monthly Energy Consumption: {min\_monthly\_consumption} kWh")

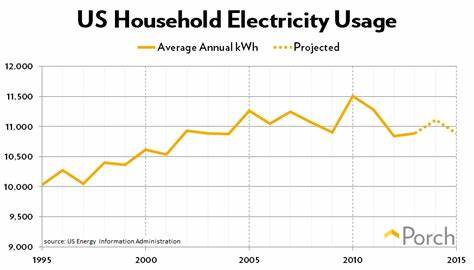
**OUTPUT:**

Total annual energy consumption:7700 kwh

Average monthly energy consumption: 641.666666666 kwh

Maximum monthly energy consumption: 800 kwh

Minimum monthly energy consumption: 500 kwh



In this example, we assume you have a list of monthly energy consumption data in kWh. You can adapt the `monthly\_consumption` list to include your actual data. The script calculates the total annual consumption, average monthly consumption, and identifies the maximum and minimum monthly consumption.

Keep in mind that this is a simplified example, and in a real-world scenario, you might want to incorporate data from sensors or utility providers for more accurate measurements.

**SAMPLE ENERGY USAGE DATA:**

To measure monthly total energy consumption for a house in Python, you’ll need data on energy usage, typically in kilowatt-hours (kWh), and a way to process that data. Here’s a basic example of how to do this:

# Sample energy usage data for each month (you would replace this with your actual data)

Energy\_data = {

“January”: 350,

“February”: 400,

“March”: 375,

“April”: 320,

“May”: 310,

“June”: 330,

“July”: 360,

“August”: 380,

“September”: 390,

“October”: 370,

“November”: 350,

“December”: 400,

}

# Calculate the total energy consumption for the year

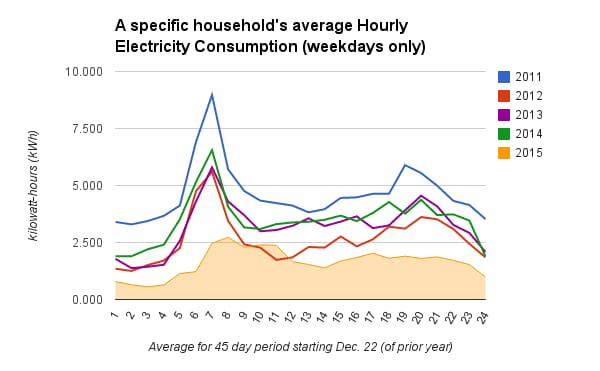
Total\_energy = sum(energy\_data.values())

# Print the result

Print(f”Total energy consumption for the year: {total\_energy} kwh”)

**Output:**

Total energy consumption for the year:4335 kwh

**Model graph:**

This code uses a dictionary `energy\_data` to store the energy consumption data for each month. It calculates the sum of all the values in the dictionary to obtain the total energy consumption for the year. You can replace the sample data with your actual energy consumption data.

Make sure to have your actual energy data in a similar format, and this code will give you the total energy consumption for the year.

**CONCLUSION:**

In conclusion, investigating cutting-edge methods to anticipate future energy consumption trends has enormous promise for resolving the issues presented by our dynamic energy environment. We can obtain deeper insights into consumption trends by combining advanced data analytics, machine learning, and IoT technologies. This will allow for more effective resource allocation, a smaller environmental impact, and improved energy sustainability. These initiatives are a critical first step towards a more intelligent, resilient, and ecologically friendly energy future.