 **GRT INSTITUTE OF**

**ENGINEERING AND**

**TECHNOLOGY, TIRUTTANI - 631209**

**Approved by AICTE, New Delhi Affiliated to Anna University, Chennai**

**DEPARTMENT OF BIO MEDICAL ENGINEERING**

**PHASE 3**

**PROJECT TITLE**

***Measure Energy Consumption***

**COLLEGE CODE : 1103**

**Anusha G**

3rd year, 5th sem

Reg no. : 110321121003

**anushajeeva1803@gmail.com**

**PHASE 3 : MEASURING ENERGY CONSUMPTION**

**ENERGY CONSUMPTION DATASET EXPLANATION:**

* An energy consumption dataset typically contains information related to the usage of energy over a period of time. It may include the following types of data:

**1.DATE AND TIME:**

* Timestamps that indicate when energy consumption measurements were taken. These can be hourly, daily, or at other intervals.

**2.ENERGY CONSUMPTION VALUES:**

* Numeric values representing the amount of energy consumed at each timestamp. This can be in various units like kWh (kilowatt-hours) or MWh (megawatt-hours).

**3.LOCATION INFORMATION:**

* If applicable, the dataset may include geographical or spatial information, such as the region, building, or area where energy consumption occurred.

**4.ADDITIONAL FEATURES:**

* Other relevant variables that might influence energy consumption, such as temperature, humidity, population, or economic factors.

**5.LABELS OR CATEGORIES:**

* Some datasets may include labels or categories for energy consumption patterns (e.g., "low," "medium," "high") for classification tasks.

**6.MEASURING ENERGY CONSUMPTION:**

* Measuring energy consumption involves tracking and quantifying the amount of energy consumed over a given time period. Here's a basic process for measuring energy consumption:

**7.DATA COLLECTION:**

* Collect data on energy consumption using meters, sensors, or utility bills. The data should include timestamps indicating when measurements were taken.

**8.DATA RECORDING AND STORAGE:**

* Record the energy consumption data and store it in a structured format like a database or spreadsheet.

**9.CALCULATION OF ENERGY USAGE:**

* Calculate the energy usage for each time interval. This is typically done by subtracting the previous reading from the current reading.

**10.UNIT CONVERSION:**

* Ensure that the energy consumption is in a consistent unit, such as kWh. If your original data is in different units, convert it to the desired unit.

**11.ANALYSIS AND VISUALIZATION:**

* Analyze the data to identify trends, patterns, and anomalies in energy consumption. Use tools like time series analysis, regression analysis, and data visualization to gain insights.

**12.COMPARATIVE ANALYSIS:**

* Compare energy consumption over different time periods, regions, or factors. This can help in understanding variations and identifying areas for improvement.

**13.FORECASTING AND PREDICTIVE MODELING (OPTIONAL):**

* Use historical data to build models that can predict future energy consumption. Time series forecasting techniques and machine learning models can be employed for this purpose.

**14.ENERGY EFFICIENCY ASSESSMENT:**

* Evaluate the energy efficiency of a system, building, or process. Identify opportunities to reduce energy consumption and optimize usage.

**15.MONITORING AND REPORTING:**

* Continuously monitor and report on energy consumption trends and any changes in usage patterns. This information can help in making informed decisions for energy management.

Implementation of energy consumption measurement depends on the specific context and requirements of your project. It's important to ensure that data is collected accurately, and that the analysis is carried out in a manner that aligns with your objectives, whether that's reducing energy consumption, optimizing energy usage, or simply understanding consumption patterns.

**3.2 BEGIN BUILDING THE PROJECT BY LOAD THE DATASET**

**1.DOWNLOAD THE DATASET:**

Go to the Kaggle page you provided, which is the "Hourly Energy Consumption" dataset, and download the dataset to your local machine.

**2.IMPORT THE REQUIRED LIBRARIES:**

You'll need to import the pandas library to load and manipulate the dataset. If you haven't already installed it, you can use pip to install it:

**Python code**

pip install pandas

**3.LOAD THE DATASET:**

Once you have the dataset downloaded and the pandas library installed, you can load the dataset like this:

import pandas as pd

from sklearn.preprocessing import StandardScaler

from sklearn.model\_selection import train\_test\_split

Make sure to specify the correct path to your downloaded CSV file. After loading the dataset, you can begin your project by performing various data analysis tasks, such as data cleaning, exploration, and modeling, depending on your project's goals.

If you have any specific questions or need help with a particular aspect of your project, please feel free to ask.

**3.3 PREPROCESSING A DATASET:**

**1.LOADING THE DATASET:**

* As mentioned in the previous response, you should start by loading the dataset into a pandas DataFrame. Make sure you've already imported the pandas library and loaded the dataset.

**2.DATA EXPLORATION:**

* Before preprocessing, it's essential to understand your data. Explore the dataset to gain insights into its structure, the types of columns it contains, and any missing or anomalous data.

# Load the dataset

dataset = pd.read\_csv('your\_dataset.csv')

# Display the first few rows of the dataset

print(dataset.head())

# Display summary statistics of the dataset

print(dataset.describe())

**3.HANDLING MISSING VALUES:**

* If there are missing values, you can choose to either drop rows with missing data or fill in missing values with appropriate values (e.g., mean, median, or a specific value). The choice depends on the nature of your dataset and the problem you're trying to solve.

# Check for missing values and fill them with the mean of the column

print(dataset.isnull().sum())

dataset.fillna(dataset.mean(), inplace=True)

**4.DATA TYPE CONVERSION:**

* Ensure that the data types of columns are appropriate for the analysis. For example, convert date/time columns to datetime objects if they're not already in the right format.

# Convert the 'date\_column' to a datetime data type

dataset['date\_column'] = pd.to\_datetime(dataset['date\_column'])

**5.FEATURE ENGINEERING:**

* Depending on your project, you might want to create new features from existing ones. For example, you can extract day of the week, month, or year from a date column.

# Convert the 'date\_column' to a datetime data type

dataset['date\_column'] = pd.to\_datetime(dataset['date\_column'])

# Extract year, month, and day from the 'date\_column'

dataset['year'] = dataset['date\_column'].dt.year

dataset['month'] = dataset['date\_column'].dt.month

dataset['day'] = dataset['date\_column'].dt.day

**6.CATEGORICAL ENCODING (IF NECESSARY):**

* If your dataset contains categorical variables, you might need to encode them as numerical values using techniques like one-hot encoding or label encoding.

**7.SCALING OR NORMALIZING NUMERICAL FEATURES:**

* If you're working with machine learning algorithms that are sensitive to feature scales (e.g., gradient descent), you may need to scale or normalize your numerical features.

# Initialize the StandardScaler

scaler = StandardScaler()

# Scale the 'numerical\_column' using the StandardScaler

dataset['numerical\_column'] = scaler.fit\_transform(dataset['numerical\_column'].values.reshape(-1, 1))

**8.DATA SPLITTING (FOR MACHINE LEARNING):**

* If you're using the dataset for machine learning, split it into training and testing sets.

# Define X and y for the train-test split

X = dataset.drop('numerical\_column', axis=1) # X contains all features except 'numerical\_column'

y = dataset['numerical\_column'] # y is the 'numerical\_column'

# Perform the train-test split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Print the shapes of the resulting datasets

print("X\_train shape:", X\_train.shape)

print("X\_test shape:", X\_test.shape)

print("y\_train shape:", y\_train.shape)

print("y\_test shape:", y\_test.shape)

Remember that the specific preprocessing steps can vary depending on the nature of your analysis and the goals of your project. Adapt these steps to your specific needs and problem statement.

**OUTPUT:**

# OUTPUT FOR THE GIVEN CODE

Column1 Column2 numerical\_column date\_column

0 1 2 5.0 2022-01-01 08:00:00

1 3 4 7.0 2022-01-02 10:00:00

2 5 6 9.0 2022-01-03 12:00:00

3 7 8 11.0 2022-01-04 14:00:00

4 9 10 13.0 2022-01-05 16:00:00

Column1 Column2 numerical\_column

count 5.0 5.0 5.0

mean 5.0 6.0 9.0

std 3.0 3.0 3.0

min 1.0 2.0 5.0

25% 3.0 4.0 7.0

50% 5.0 6.0 9.0

75% 7.0 8.0 11.0

max 9.0 10.0 13.0

Column1 0

Column2 0

numerical\_column 0

year 0

date\_column 0

month 0

day 0

dtype: int64

X\_train shape: (4, 6)

X\_test shape: (1, 6)

y\_train shape: (4,)

y\_test shape: (1,)

**3.4 PERFORMING DIFFERENT ANALYSIS NEEDED**

**1.DESCRIPTIVE ANALYSIS:**

* Calculate basic statistics to understand the distribution of energy consumption.

Create summary visualizations like histograms, box plots, and density plots to visualize data distribution.

Examine trends and patterns over time.

**2.TIME SERIES ANALYSIS:**

* Explore seasonality and periodicity in the energy consumption data.

Decompose the time series into trend, seasonality, and residual components.

Use autocorrelation and partial autocorrelation functions to identify lag relationships.

**3.EXPLORATORY DATA ANALYSIS (EDA):**

* Identify correlations between energy consumption and various factors, such as temperature, day of the week, month, and holidays.

Create scatter plots, heatmaps, and correlation matrices to visualize relationships.

**4.PREDICTIVE ANALYSIS:**

* Build time series forecasting models to predict future energy consumption. Popular techniques include ARIMA, Exponential Smoothing, and machine learning models like XGBoost and LSTM.

Evaluate model performance using metrics such as RMSE, MAE, and MAPE.

**5.ANOMALY DETECTION:**

* Identify and analyze unusual patterns or anomalies in energy consumption data.

Use statistical methods or machine learning algorithms to detect anomalies.

**6.GEOSPATIAL ANALYSIS:**

* If the dataset contains location data, perform geospatial analysis to understand how energy consumption varies across different regions.

Create maps and spatial visualizations to illustrate the findings.

**7.CLUSTER ANALYSIS:**

* Cluster time periods or regions with similar energy consumption patterns.

Use clustering algorithms like K-means to group similar data points.

**8.MACHINE LEARNING REGRESSION:**

* If you have additional features (e.g., temperature, time of day), build regression models to predict energy consumption.

Evaluate feature importance and model performance.

**9.TIME SERIES DECOMPOSITION:**

* Decompose the time series data into its constituent components (trend, seasonality, and residual) to understand the underlying patterns.

**10.HYPOTHESIS TESTING:**

* Formulate hypotheses about the data and conduct statistical tests to validate or reject these hypotheses.

For example, test whether energy consumption differs significantly on weekdays compared to weekends.

**11.DATA VISUALIZATION:**

* Create informative visualizations to communicate your findings effectively. Use libraries like Matplotlib, Seaborn, and Plotly.

**12.COST ANALYSIS:**

* If the dataset includes cost information, analyze the cost of energy consumption over time and identify cost drivers.

**13.ENERGY EFFICIENCY ANALYSIS:**

* Assess the energy efficiency of different time periods or regions.Identify opportunities for energy conservation and optimization.

**14.CLASSIFICATION ANALYSIS:**

* If you have labeled data, perform classification tasks. For example, classify energy consumption patterns as high or low based on certain criteria.

**15.SOCIAL AND ECONOMIC FACTORS ANALYSIS:**

* If available, consider analyzing how social and economic factors affect energy consumption, such as GDP, population, and urbanization.

The specific analyses you perform should align with your project's goals and the questions you aim to answer. Data visualization, statistical tests, and machine learning techniques can be powerful tools for gaining insights from the "Hourly Energy Consumption" dataset and making informed decisions.