**AI\_PHASE-1 – MEASURE ENERGY CONSUMPTION**

**ABSTRACT:**

Measuring energy consumption, it involves tracking and analyzing the amount of energy used by a particular device, system, or building. This information can help identify energy-saving opportunities and make more informed decisions about energy usage. It’s an important step in promoting energy efficiency and sustainability.

**MODULE:**

When it comes to measuring energy consumption, there are a few key modules that are typically involved:

1️.Sensors: These devices are responsible for collecting data on energy usage. They can include electricity meters, smart plugs, or other monitoring devices.

2️.Data Acquisition: This module focuses on gathering and recording the energy consumption data from the sensors. It ensures that the data is accurately captured and stored for further analysis.

3️.Data Processing: This module involves analyzing and processing the collected data. It often uses programming languages like Python and data analysis tools like pandas to calculate energy consumption metrics and identify patterns or trends.

4️.Visualization: This module helps to present the energy consumption data in a visual format. It can include charts, graphs, or dashboards that make it easier to understand and interpret the data.

These modules work together to provide insights into energy usage and help identify opportunities for energy efficiency improvements.

**MODEL DEVELOPMENT:**

To develop a model for measuring energy consumption, you can follow these steps:

1. Collect data: Gather energy consumption data from sensors or smart meters. This data should include variables like time, date, and energy usage.

2. Pre-process the data: Clean the data by handling missing values, outliers, and formatting issues. You may need to convert the data into a suitable format for analysis.

3. Feature engineering: Extract relevant features from the data that can help in predicting energy consumption. This can include factors like weather conditions, occupancy, or time of day.

4. Split the data: Divide the dataset into training and testing sets. The training set is used to train the model, while the testing set is used to evaluate its performance.

5. Choose and train a model: Select a suitable machine learning algorithm, such as linear regression, decision trees, or neural networks. Train the model using the training data.

6. Evaluate the model: Assess the model's performance using evaluation metrics like mean squared error or R-squared. This will help you understand how well the model predicts energy consumption.

7. Fine-tune and validate: Adjust the model's parameters and hyperparameters to improve its performance. Validate the model using cross-validation techniques to ensure its generalizability.

8. Deploy and monitor: Once you're satisfied with the model's performance, deploy it in a production environment. Continuously monitor its predictions and update the model as needed**.**

**IMPLEMENTATION USING PYTHON :**

#python

Import pandas as pd

Import matplotlib.pyplot as plt

# Read energy consumption data from a CSV file

Data = pd.read\_csv(‘energy\_data.csv’)

To represent energy consumption using a bar diagram, you can use Python’s matplotlib library.

# Sample data

Categories = [‘Category 1’, ‘Category 2’, ‘Category 3’]

Consumption = [10, 20, 15] # Energy consumption values

# Create the bar plot

Plt.bar(categories, consumption)

# Add labels and title

Plt.xlabel(‘Categories’)

Plt.ylabel(‘Energy Consumption’)

Plt.title(‘Energy Consumption by Category’)

# Show the plot

Plt.show()

```

This code creates a bar plot with categories on the x-axis and energy consumption values on the y-axis.

# Plot the energy consumption over time

Plt.plot(data[‘Date’], data[‘Energy Consumption’])

Plt.xlabel(‘Date’)

Plt.ylabel(‘Energy Consumption’)

Plt.title(‘Energy Consumption Over Time’)

Plt.show()

Import matplotlib.pyplot as plt # plotting

Import numpy as np # linear algebra

Import os # accessing directory structure

Import pandas as pd # data processing, CSV file I/O (e.g. pd.read\_csv)

Import seaborn as sns

Plt.style.use(‘ggplot’) # Make it pretty

In [2]:

# Data is saved in parquet format so schema is preserved.

Df = pd.read\_parquet(‘../input/est\_hourly.paruqet’)