

Phase 3: Submission Document

Project Title: *Measure Energy Consumption*

Phase 3: Development Part 1

Topic: Measure Energy Consumption by loading and preprocessing the dataset



Measure Energy Consumption

Introduction:

Measuring energy consumption is a critical process in our modern world, where the demand for energy continues to rise, and the need for sustainability and efficient resource utilization becomes increasingly pressing. Energy consumption measurement plays a pivotal role in understanding, managing, and optimizing our energy usage, whether it's in homes, industries, or the broader infrastructure.

This introduction sets the stage for exploring the various aspects of measuring energy consumption, ranging from the tools and technologies used to monitor usage to the broader implications for sustainability and resource management.

Data Acquisition:

- ☐ Obtain the dataset from a reliable source, which could be a government agency, an energy provider, or any other relevant source.
- ☐ Ensure that you have the necessary permissions and rights to use the data for your analysis.

Data Exploration:

- ☐ Load the dataset into your preferred data analysis environment, such as Python (using libraries like pandas) or R

- ☐ Check for any missing values, outliers, or inconsistencies in the data

Data Cleansing:

Handle missing data: Decide on a strategy to deal with missing values, which may include imputation or removal of rows or columns. Address outliers: Identify and handle outliers that can skew the analysis. You might choose to remove outliers or transform them. Standardize data types: Ensure that data types are appropriate for analysis (e.g., numeric data types for numerical features, datetime data types for timestamps).

Data Transformation:

- ☐ Convert categorical data: If the dataset contains categorical variables (e.g., location, building type), you may need to encode them using techniques like one-hot encoding or label encoding.
- ☐ Feature scaling: Scale numerical features if necessary, using techniques like Min-Max scaling or standardization (z-score scaling).
- ☐ Time-series data: If the dataset involves time series data, sort the data by timestamps and potentially resample or aggregate the data to the desired time intervals.

Program:

```
import time

# Simulated energy consumption data for demonstration purposes
# Replace this with actual data from your energy meters or sensors
current_energy_consumption = 100 # Initial energy consumption in
watts

def read_energy_data():
    # Replace this with code to read data from your energy meters or
    sensors
    # This could involve using hardware-specific libraries or APIs

    # For demonstration purposes, we simulate a change in energy
    consumption
    global current_energy_consumption
    current_energy_consumption += 10
    return current_energy_consumption

def main():
    while True:
        try:
            # Read energy consumption data
            energy_data = read_energy_data()

            # Process and log the data
            # You can save the data to a file, database, or display it in
            real-time
            print(f"Current Energy Consumption: {energy_data} watts")

            # You can add your energy analysis or control logic here
```

```
# Sleep for a specified interval (e.g., every 5 seconds)
time.sleep(5)
```

```
except KeyboardInterrupt:
    print("Measurement program stopped.")
    break
```

```
if __name__ == "__main__":
    main()
```

Output:

Current Energy Consumption: 110 watts
Current Energy Consumption: 120 watts
Current Energy Consumption: 130 watts
...