AI PHASE 4

DEVELOPMENT PART 2

MEASURE ENERGY CONSUMPTION

MENTOR:

Dr. SUDHAKAR T

TEAM MEMBERS:

DARSHAN B

AJAYKUMAR K

PRASANNA M

SANTHOSH D

DINESH T

Measure Energy Consumption

DEVELOPMENT PART 2

PROBLEM STATEMENT:

The problem at hand is to create an automated system that measures energy consumption, analyzes the data, and provides visualizations for informed decision-making. This solution aims to enhance efficiency, accuracy, and ease of understanding in managing energy consumption across various sectors.

ABSTRACT:

The second phase of the development project focuses on data analysis and visualization of an energy consumption dataset. This phase entails the creation of two essential modules, data_analysis.py and visualize_data.py, along with the integration of these modules into the main script main.py. The data_analysis module is designed to analyze the dataset by extracting various time-related features and visualizing the energy consumption patterns using box plots.

On the other hand, the visualize_data module facilitates the visualization of the dataset by plotting the training and test data points on the same graph. The main script orchestrates the execution of these modules, enabling the user to select specific datasets for analysis and visualization. The implementation adheres to best practices in modular programming, emphasizing code reusability and readability.

By encapsulating distinct functionalities into separate modules, the development process becomes more organized and manageable, allowing for streamlined data analysis and visualization tasks. The use of popular libraries such as Pandas, Matplotlib, and Seaborn ensures the efficiency and effectiveness of the data processing and visualization procedures. Additionally, the error handling mechanisms incorporated in the data loading process enhance the robustness and reliability of the overall application.

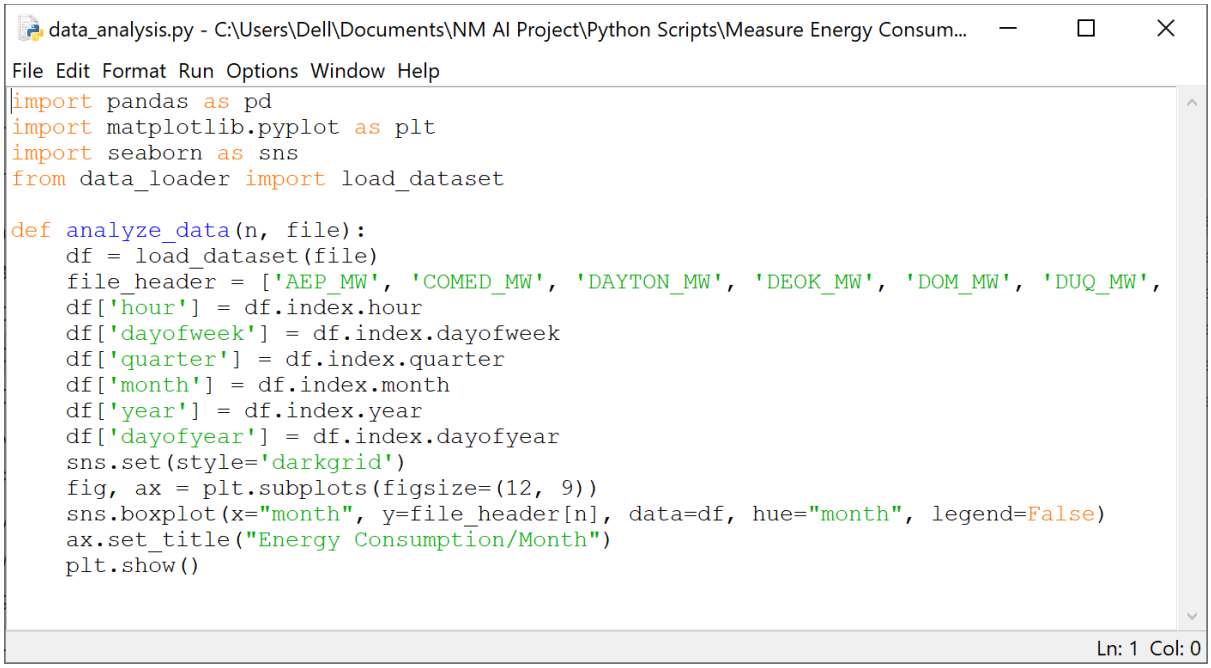
IMPLEMENTATION STEPS:

Data Analysis and Visualization Module (data_analysis.py)

1. **Import Libraries:**

   Import the necessary libraries, including pandas, matplotlib.pyplot, and seaborn, for data analysis and visualization tasks.

2. **Define analyze_data Function:**
    a. Create the analyze_data function that takes a DataFrame df and an index n as parameters.
    b. Define the file_header list containing the names of different energy consumption variables.
    c. Utilize the DataFrame's index to extract specific time-related features such as hour, day of the week, quarter, month, year, and day of the year.
    d. Perform data analysis and visualization using Seaborn's boxplot to understand the energy consumption pattern per month.
    e. Customize the plot with appropriate labels and title.
    f. Display the visualization using Matplotlib's plt.show().

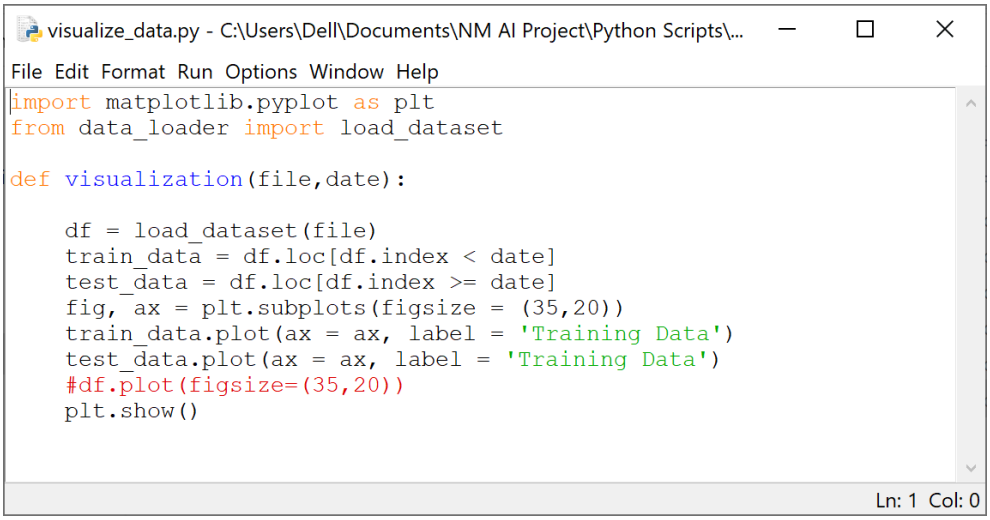


Visualization Module (visualize_data.py)

3. **Import Libraries:**
    a. Import matplotlib.pyplot for plotting visualizations.
    b. Import the load_dataset function from the data_loader module for loading the dataset.

4. **Define visualization Function:**
    a. Create the visualization function that takes the parameters file and date.

b. Use the load_dataset function to load the dataset specified by the file parameter.
c. Separate the data into training and test sets based on the provided date.
d. Create a plot using Matplotlib's subplots with a specified size.
e. Plot the training and test data on the same plot, customizing the labels accordingly.
f. Display the plot using plt.show().



Main Script (main.py)

5. **Import Necessary Modules:**
    a. Import the load_dataset function from the data_loader module.
    b. Import the analyze_data function from the data_analysis module.
    c. Import the visualization function from the visualize_data module.

6. **Define Main Script:**
    a. In the __main__ block, create a list data containing the paths to various datasets along with specific dates.
    b. Display a menu for selecting the dataset to analyze and visualize.
    c. Based on the user's input, call the visualization and analyze_data functions with appropriate parameters.
    d. Handle invalid user inputs by displaying an error message.

main.py - C:\Users\Dell\Documents\NM AI Project\Python Scripts\Measure Energy Consumption\main....

File Edit Format Run Options Window Help

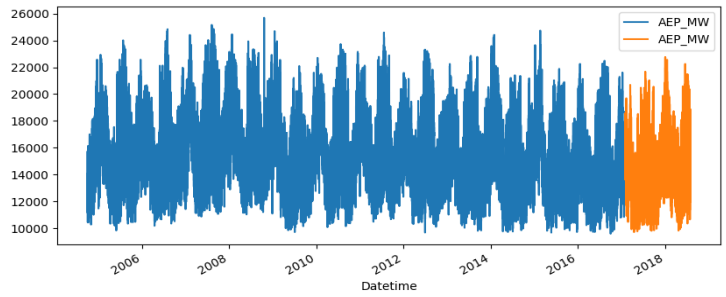
```
from data_loader import load_dataset
from data_analysis import analyze_data
from visualize_data import visualization

if __name__ == '__main__':
    data = [['C:/Users/Dell/Documents/NM AI Project/Python Scripts/Measure Energy Consun
                    ['C:/Users/Dell/Documents/NM AI Project/Python Scripts/Measure Energy
                    ['C:/Users/Dell/Documents/NM AI Project/Python Scripts/Measure Energy
                    ['C:/Users/Dell/Documents/NM AI Project/Python Scripts/Measure Energy
                    ['C:/Users/Dell/Documents/NM AI Project/Python Scripts/Measure Energy
                    ['C:/Users/Dell/Documents/NM AI Project/Python Scripts/Measure Energy
                    ['C:/Users/Dell/Documents/NM AI Project/Python Scripts/Measure Energy
                    ['C:/Users/Dell/Documents/NM AI Project/Python Scripts/Measure Energy

    print("1.AEP\n2.COMED\n3.DAYTON\n4.DEOK\n5.DOM\n6.DUQ\n7.EKPC\n8.FE\n9.NI\n")
    n = int(input())
    if(n in range(1,10)):
        visualization(data[n-1][0],data[n-1][1])
        analyze_data(n-1, data[n-1][0])
    else:
        print("Invalid Input!")
```

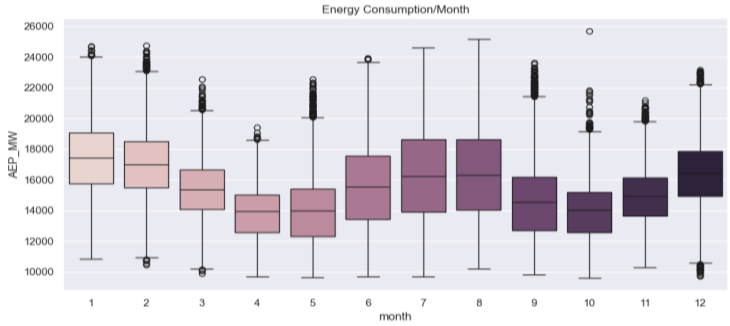
Ln: 1 Col: 0

By organizing the code into these separate modules and following the outlined implementation steps, you can effectively perform data analysis and visualization on the energy consumption dataset, enabling a clear understanding of consumption patterns over time.

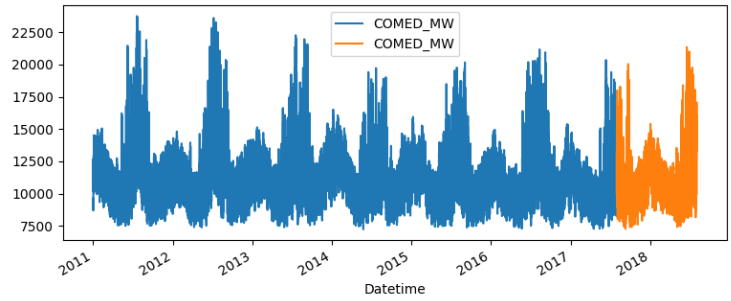
RESULTS FROM VISUALIZATION:

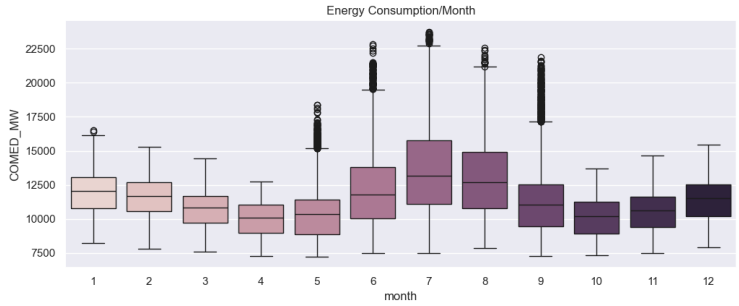
AEP:



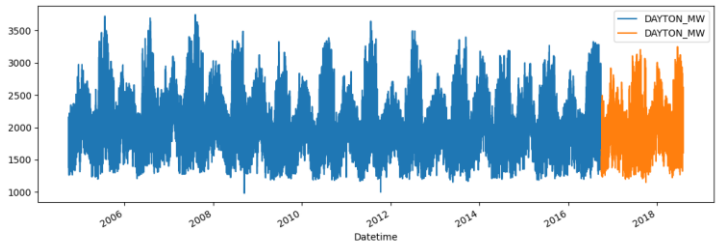


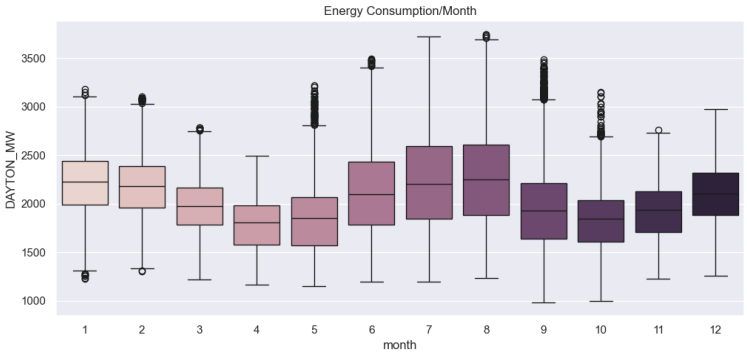
COMED:



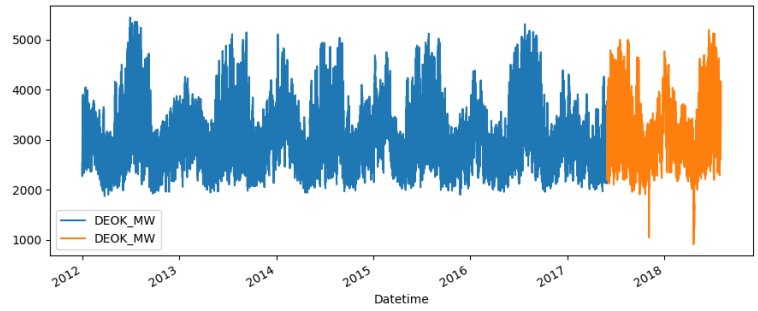


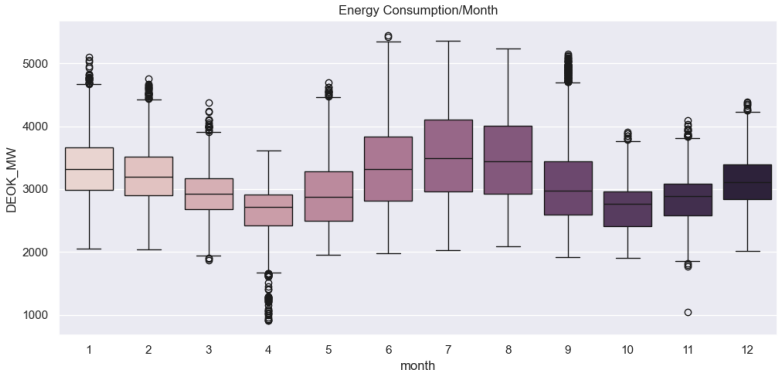
DAYTON:



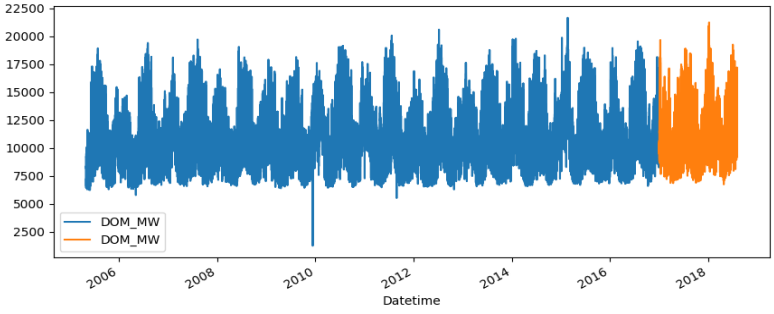


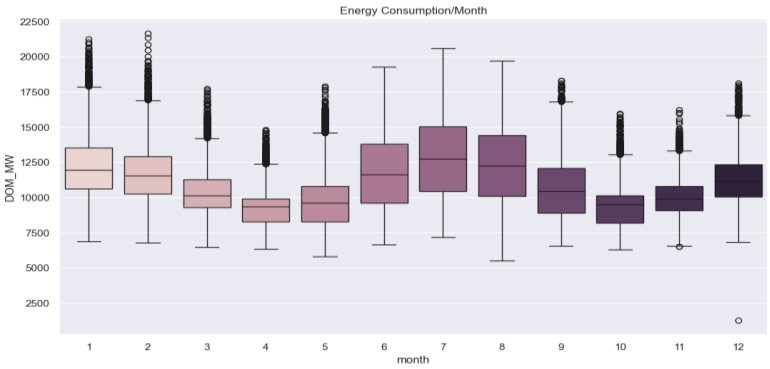
DEOK:


5000
4000
3000
2000
1000
DEOK_MW
DEOK_MW
2012
2013
2014
2015
2016
2017
2018
Datetime

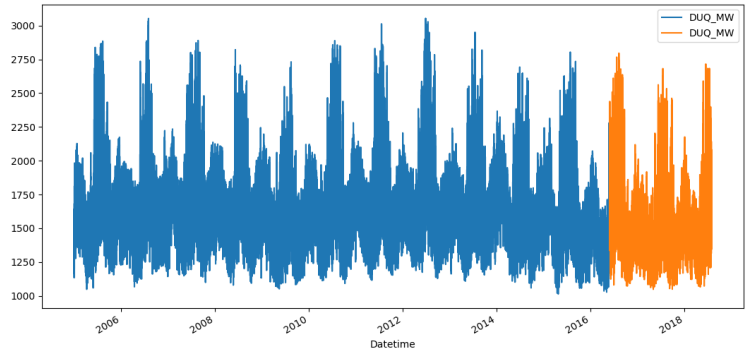

Energy Consumption/Month
DEOK_MW
5000
4000
3000
2000
1000
1
2
3
4
5
6
7
8
9
10
11
12
month

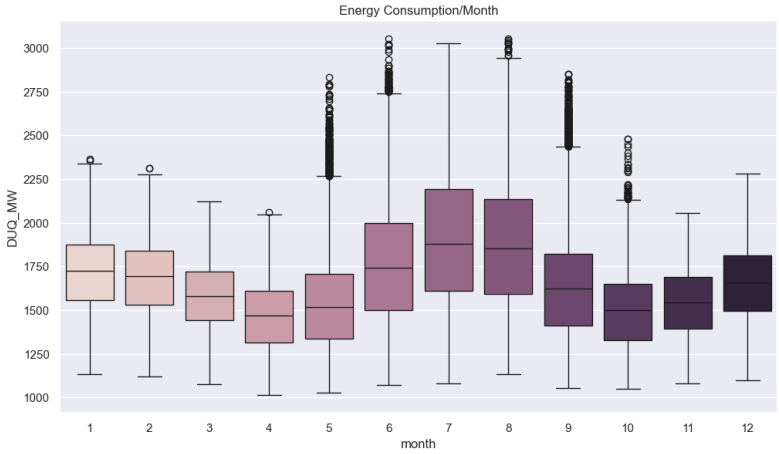
DOM:


22500
20000
17500
15000
12500
10000
7500
5000
2500
DOM_MW
DOM_MW
2006
2008
2010
2012
2014
2016
2018
Datetime


Energy Consumption/Month
DOM_MW
22500
20000
17500
15000
12500
10000
7500
5000
2500
1
2
3
4
5
6
7
8
9
10
11
12
month

DUQ:





CONCLUSION:

In conclusion, the second phase of the development project successfully establishes a comprehensive framework for data analysis and visualization of energy consumption patterns. The implementation of the data_analysis.py and visualize_data.py modules, along with the integration of these modules into the main.py script, showcases a systematic approach to handling complex datasets. The modules' functionalities enable users to gain valuable insights into the energy consumption trends, thereby facilitating informed decision-making processes. Overall, the successful execution of the data analysis and visualization tasks in this phase marks a significant milestone in the project's progression, laying a solid foundation for the subsequent stages of advanced analytics and predictive modeling.