**MEASURING ENERGY CONSUMPTION**

**Sanjay Kumar V** (2021503046**) Achyut Prasad DC** (2021503002)

**Ragul N** (2021503542) **Santhosh Kumar** **S** (2021503302)

**Phase 5: Documentation and Submission**

**Problem Statement:**

The problem at hand is to create an automated system that measures energy consumption, 1nalyses 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.

**Enhancing Energy Consumption Measurement and Decision-Making Background**

Energy consumption is a fundamental aspect of modern society, influencing sectors such as manufacturing, residential living, commercial operations, and transportation. Accurate measurement of energy usage is critical for several reasons:

* **Resource Optimization:** Effective energy consumption measurement allows for the efficient allocation of resources and cost savings. It aids in reducing energy wastage and identifying opportunities for optimization.
* **Environmental Impact:** Monitoring and managing energy consumption is vital for reducing the environmental impact. Accurate measurement helps track emissions and supports sustainability goals.
* **Operational Efficiency:** In commercial and industrial settings, understanding energy consumption patterns is essential for optimizing operations and ensuring productivity.
* **Decision-Making:** Energy consumption data informs key decisions related to infrastructure upgrades, capacity planning, and resource allocation.

However, traditional methods of energy measurement are often manual, time-consuming, and error-prone. This results in limited visibility into real-time energy consumption and hinders informed decision-making. There is a pressing need for an automated system that not only measures energy consumption accurately but also analyzes the data and provides visualizations to enhance decision-making across various sectors.

**Project Objective**

The primary objective of the "Measure Energy Consumption" project is to develop a comprehensive solution for the measurement, analysis, and visualization of energy consumption data. This project aims to address the following key challenges:

* **Data Collection:** Create a robust system for automated data collection, eliminating the need for manual measurements. This includes capturing data from various sources, such as manufacturing equipment, household appliances, and energy meters.
* **Data Preprocessing:** Ensure the collected data is clean, consistent, and ready for analysis. Data preprocessing activities involve handling missing values, outliers, and data inconsistencies.
* **Feature Extraction:** Extract relevant features and metrics from the energy consumption data, enabling a more in-depth understanding of energy patterns.
* **Model Development:** Utilize statistical analysis, machine learning, and time series forecasting methods to uncover trends, patterns, and anomalies in the data. Develop predictive models for future energy consumption.
* **Visualization:** Create meaningful and intuitive visualizations, such as graphs and charts, to present energy consumption trends and insights. Visualization aids in better understanding and decision-making.
* **Automation:** Build a script or system that automates data collection, analysis, and visualization processes. This automation ensures real-time or periodic updates and reduces the burden of manual data handling.

**Project Scope**

The project's scope covers a wide range of sectors, including manufacturing sites, residential homes, commercial buildings, and transportation. The solution will be flexible, adaptable, and scalable to accommodate different data sources and usage scenarios.

By addressing these challenges and objectives, the "Measure Energy Consumption" project seeks to enhance energy measurement accuracy, improve resource allocation, promote environmental sustainability, and enable data-driven decision-making across sectors. The project aims to provide a comprehensive, automated solution for measuring and managing energy consumption efficiently and effectively.

In this phase of the project, we will focus on loading and pre-processing the PJM Hourly Energy Consumption dataset. The PJM Interconnection LLC (PJM) is a regional transmission organization in the United States, operating an electric transmission system serving several states. The dataset provides hourly power consumption data in megawatts (MW) and is a valuable resource for understanding energy consumption patterns.

**PJM Hourly Energy Consumption Data:**

**Source:** PJM's website

**Data Format:** Hourly time series data

**Unit of Measurement:** Megawatts (MW)

**Geographical Coverage:** Delaware, Illinois, Indiana, Kentucky, Maryland, Michigan, New Jersey, North Carolina, Ohio, Pennsylvania, Tennessee, Virginia, West Virginia, and the District of Columbia.

The dataset captures energy consumption trends over time, and it is particularly useful for various analyses and modelling tasks. Here are some ideas of what can be done with this dataset:

**Energy Consumption Prediction:** Splitting the last year into a test set, we can build a model to predict energy consumption. This can be valuable for optimizing energy resources and planning.

**Trend Analysis:** We can explore trends in energy consumption related to different factors, such as hours of the day, holidays, or long-term trends. Understanding these trends can help in resource allocation and grid management.

**Seasonal Analysis:** Investigate how daily trends change depending on the time of year. The difference between summer and winter trends can be substantial and understanding these variations is essential for energy management.

Feature extraction is a crucial step in data analysis and modeling, especially in the context of the "Measure Energy Consumption" project. It involves selecting or creating relevant features (variables) from the raw data that can provide valuable information for analysis and modeling. In the energy consumption domain, feature extraction techniques aim to capture the essential characteristics of energy data to improve the accuracy of predictive models and enhance decision-making. Here are some feature extraction techniques commonly used in energy consumption analysis:

**Time-Based Features:**

Hour of the Day: Extracting the hour of the day as a feature can capture diurnal patterns in energy consumption. It helps model daily variations and peak usage times.

Day of the Week: This feature considers the day of the week (e.g., Monday, Tuesday) and is valuable for understanding how energy consumption varies on weekdays versus weekends.

Month and Season: Features related to the month or season can capture seasonal patterns in energy usage. For example, energy consumption tends to be higher in summer due to air conditioning usage.

Lagged Features:

Previous Hour(s) Consumption: Including lagged features, such as the energy consumption of the previous hour(s), allows the model to consider the historical context and trends in consumption. It's particularly valuable for time series forecasting.

Holiday Indicators:

Binary indicators for holidays or special events can be useful. These features help the model account for variations in energy consumption related to holidays and events, which often exhibit different consumption patterns.

Weather Data Integration:

Integrating weather data as features can enhance energy consumption models. Variables like temperature, humidity, and weather conditions (e.g., rainy, sunny) can influence energy usage. Including these weather-related features can improve the model's predictive accuracy.

Load Profiles:

Load profiles are features that represent historical patterns of energy consumption. These profiles capture regularities and trends in energy usage over time.

Moving Averages:

Calculating moving averages of energy consumption over a specific time window (e.g., daily, weekly) can help in smoothing out noise and identifying underlying trends.

Statistical Aggregates:

Features that represent statistical measures such as mean, median, standard deviation, and variance of energy consumption can offer insights into the data's central tendency and variability.

Frequency Domain Features:

In the context of time series analysis, features extracted from the frequency domain can capture periodic patterns and seasonalities. Techniques like Fourier analysis can help identify dominant frequency components.

Pattern Recognition:

Advanced feature extraction methods may involve pattern recognition algorithms, such as clustering or principal component analysis, to identify common patterns or behaviors in energy consumption data.

**Choice of Machine Learning Algorithm:**

The choice of machine learning algorithm depends on the nature of the problem, the characteristics of the data, and the project's objectives. In the context of energy consumption prediction, several algorithms and techniques are commonly used:

* **Linear Regression:** Linear regression is a straightforward choice for modeling energy consumption when there is a clear linear relationship between features and the target variable. It's relatively simple and interpretable but may not capture complex patterns.
* **Time Series Forecasting Models:** Time series analysis methods like ARIMA (AutoRegressive Integrated Moving Average) or more advanced models like Seasonal Decomposition of Time Series (STL) can be employed when dealing with sequential energy consumption data. These models are designed to capture temporal patterns and seasonality.
* **Decision Trees and Random Forest:** Decision trees and ensemble techniques like Random Forest can handle both regression and classification tasks. They are robust and can capture nonlinear relationships in the data.
* **Deep Learning Models:** Deep learning models, particularly recurrent neural networks (RNNs) and long short-term memory (LSTM) networks, are effective for capturing complex temporal dependencies in energy consumption data. They are suitable for time series forecasting.
* **Support Vector Machines (SVM):** SVMs can be applied to regression tasks, especially when there is a need to capture nonlinearity and work with high-dimensional data.

**Model Training:**

Model training involves using historical data to teach the selected machine learning algorithm to make predictions about future energy consumption. Here's how the process typically works:

* **Data Splitting:** The dataset is divided into training and testing sets, with the training set used to train the model and the testing set used to evaluate its performance. Time-based splitting is common, ensuring that the training data comes before the testing data to simulate real-world conditions.
* **Feature Engineering:** Features selected through feature extraction techniques are used to represent the dataset. These features serve as input to the model.
* **Hyperparameter Tuning:** The model's hyperparameters, such as learning rate, regularization strength, and architecture, are fine-tuned to optimize performance. Techniques like cross-validation can be employed for this purpose.
* **Model Training:** The machine learning algorithm is applied to the training data, and the model is trained to predict future energy consumption based on the selected features.

**Evaluation Metrics:**

To assess the model's performance in predicting energy consumption accurately, appropriate evaluation metrics must be chosen. Common metrics include:

**Mean Absolute Error (MAE):** MAE measures the average absolute difference between predicted and actual energy consumption values. It is easy to interpret and provides a straightforward measure of model accuracy.

**Root Mean Square Error (RMSE):** RMSE calculates the square root of the mean of the squared differences between predicted and actual values. It penalizes larger errors more heavily than MAE and provides insight into the model's predictive power.

**R-squared (R2):** R-squared measures the proportion of the variance in the dependent variable (energy consumption) that is predictable from the independent variables (features). It provides an indication of how well the model fits the data.

**Mean Absolute Percentage Error (MAPE):** MAPE is expressed as a percentage and measures the average absolute percentage difference between predicted and actual values. It provides a relative measure of model accuracy.

The choice of evaluation metric depends on the specific goals of the project. Lower MAE and RMSE values indicate better model performance, while a higher R-squared value suggests a good fit. MAPE is useful when understanding relative errors is essential.

The choice of machine learning algorithm, model training techniques, and evaluation metrics should align with the project's objectives and the characteristics of the energy consumption data, ensuring that the system provides accurate and reliable predictions for informed decision-making.

**Innovative techniques** and approaches can significantly enhance the development of the "Measure Energy Consumption" project, making it more accurate and efficient. Here are some innovative techniques and approaches that can be employed during the development of the project:

* **Ensemble Learning:** Ensemble methods involve combining predictions from multiple models to improve overall performance. Techniques like Random Forest, Gradient Boosting, and Bagging can be applied to enhance the prediction system's accuracy and robustness. By implementing ensemble learning, the project can leverage diverse model outputs to make more reliable predictions.
* **Deep Learning Architectures:** Deep learning models, such as recurrent neural networks (RNNs) and long short-term memory (LSTM) networks, can capture intricate patterns and dependencies in energy consumption data. These architectures excel at time series forecasting and can provide highly accurate predictions. By incorporating deep learning, the project can make more precise forecasts of future energy consumption patterns.
* **Time Series Analysis:** Advanced time series analysis techniques can help in uncovering complex temporal patterns in energy consumption data. This includes methods like Seasonal Decomposition of Time Series (STL), which decomposes time series into trend, seasonality, and residual components. By embracing time series analysis, the project can gain deeper insights into the underlying patterns and trends.
* **Machine Learning Models:** The project can explore a wide range of machine learning models beyond traditional linear regression. Decision trees, support vector machines (SVM), k-nearest neighbors (KNN), and other algorithms can be employed to identify nonlinear relationships and patterns in the data. Utilizing a diverse set of machine learning models can provide a broader perspective on energy consumption prediction.
* **Hybrid Models:** A hybrid model combines different machine learning or deep learning algorithms to leverage the strengths of each. For instance, combining a traditional time series model with a deep learning model can capture both short-term and long-term patterns in energy consumption. Hybrid models can be an innovative approach to improve predictive accuracy.

**Analysis Objectives:**

The primary objectives of this phase of the project are:

**Understanding User Behavior:** Gain insights into how users interact with the website, including which pages they visit, the duration of their visits, and their navigation paths.

**Identifying Usability Issues:** Detect usability issues or obstacles that users might face while navigating the website, leading to poor user experiences.

**Optimizing Website Performance:** Analyze website performance metrics to identify bottlenecks, slow-loading pages, or server issues that can be resolved to enhance the user experience.

**Personalization Opportunities:** Explore opportunities for personalizing the user experience based on user behavior and preferences.

**Data Collection Process:**

Data collection for website analysis involves gathering various types of data from the website's backend and user interactions. The process typically includes:

* **Server Logs:** Collect server logs that record all incoming requests, page views, user agents, and other technical data. These logs can provide information on page load times and server performance.
* **Web Analytics Tools:** Implement web analytics tools, such as Google Analytics or Adobe Analytics, to track user interactions, including page views, click-through rates, bounce rates, and user demographics.
* **Heatmaps and Session Recordings:** Use heatmap tools like Hotjar or Crazy Egg to record and analyze where users click, move their cursors, and spend the most time on pages. Session recordings can show real user interactions and pain points.
* **User Surveys and Feedback:** Collect user feedback through surveys, feedback forms, and social media channels. This qualitative data can provide valuable insights into user satisfaction and issues they encounter.
* **A/B Testing Data:** If A/B tests are conducted to compare variations of web elements or designs, collect data on user interactions and conversions to determine the impact of changes.

**Data Visualization Using IBM Cognos:**

IBM Cognos is a powerful data visualization and business intelligence tool that can be used to create interactive and insightful visualizations from the collected data. The visualization process includes:

* **Data Integration:** Connect IBM Cognos to the data sources, including server logs, web analytics tools, and user feedback databases.
* **Data Cleaning and Transformation:** Prepare the data by cleaning, formatting, and transforming it into a structured format suitable for analysis.
* **Dashboard Creation:** Develop interactive dashboards and reports in IBM Cognos that present key website performance metrics, user behavior patterns, and insights. These visualizations can include charts, graphs, heatmaps, and tables.
* **User Path Analysis:** Visualize user navigation paths to understand common user journeys, entry and exit points, and pages with high drop-off rates.
* **Performance Metrics:** Create visualizations for website performance metrics, such as page load times, server response times, and error rates. Identify performance bottlenecks.
* **Conversion Funnel Analysis:** Build visualizations to track conversion funnels, including cart abandonment, sign-up processes, or goal completions, to optimize the conversion rate.
* **User Segmentation:** Segment users based on demographics, behavior, or preferences to personalize content and marketing efforts.

**Python Code Integration:**

To complement IBM Cognos visualizations, Python code integration can be employed for in-depth analysis and advanced insights:

* **Machine Learning:** Utilize Python libraries for machine learning to develop predictive models for user behavior or recommenders. For example, collaborative filtering models can suggest personalized content based on user history.
* **Natural Language Processing (NLP):** Analyze user feedback and comments using NLP techniques to extract sentiment, themes, and common issues. Python's NLP libraries, like NLTK and spaCy, can be used for text analysis.
* **Custom Data Processing:** Python can be used to perform custom data processing and transformations if IBM Cognos lacks certain functionalities or custom calculations are needed.

**Improving User Experience:**

The insights from this analysis can significantly help website owners improve user experience in several ways:

* **Enhanced User Interface:** Identifying pain points in the user journey helps in redesigning the website's interface to be more user-friendly and intuitive.
* **Content Personalization:** Analyzing user behavior and preferences allows for content personalization, showing users what is most relevant to them.
* **Performance Optimization:** Resolving performance bottlenecks and improving page load times can result in a smoother and more enjoyable user experience.
* **Usability Improvements:** Identifying usability issues leads to interface changes that make the website easier to navigate.
* **Conversion Rate Optimization:** Analyzing the conversion funnel can help website owners optimize the process and increase conversion rates.

In summary, by conducting thorough data analysis, visualization, and integrating Python code, website owners can gain actionable insights that lead to website improvements, resulting in a better user experience and increased user satisfaction.

**IoT Device Setup:**

The IoT device setup involves the deployment of sensors, actuators, and communication modules to capture and transmit data. The setup includes:

* **Sensors:** Deploy various sensors based on project requirements, such as electricity consumption meters, temperature sensors, humidity sensors, or equipment health monitors.
* **Actuators:** If needed, actuators can be integrated to control equipment remotely or respond to sensor data.
* **Communication Modules:** Connect the devices to a network using communication modules like Wi-Fi, LoRa, or cellular connectivity for data transmission.
* **Data Aggregation:** Implement data aggregation and preprocessing logic on the devices to reduce data size and ensure efficient transmission.

**Platform Development:**

The development of the data-sharing platform involves creating a system that can receive, process, store, and visualize IoT data:

* **Data Ingestion:** Set up data ingestion components to receive data from IoT devices in real-time.
* **Data Processing:** Implement data processing pipelines to clean, transform, and enrich the raw data for analysis.
* **Data Storage:** Choose a suitable data storage solution, such as a relational database, NoSQL database, or data lakes, to store historical data.
* **Data Visualization:** Develop a user-friendly and interactive dashboard for visualizing real-time and historical data. Use tools like Grafana, Tableau, or custom web-based dashboards.
* **Alerting and Notifications:** Implement alerting systems to notify stakeholders or operators about critical events or anomalies.
* **Security:** Ensure data security by implementing encryption, access controls, and authentication mechanisms to protect sensitive information.

**PHASES OF THE PROJECT:**

**Phase 1: Problem Definition and Design Thinking**

**Introduction:**

The project "Measure Energy Consumption" aims to create an automated system for measuring energy consumption, analyzing the data, and providing visualizations to facilitate informed decision-making. This comprehensive solution targets enhancing efficiency, accuracy, and the comprehension of managing energy consumption across various sectors. To address this problem effectively, the project team adopted a structured approach encompassing the various stages of the design thinking process.

**Phase 2: Transformation Plan**

In this phase, the project team focused on implementing a robust transformation plan that encompassed key stages, including data source identification, data preprocessing, feature extraction, advanced modeling, and visualization. Through the meticulous identification of pertinent datasets for energy consumption, they were able to ensure the availability and relevance of the data required for analysis. The subsequent data preprocessing steps allowed for the cleaning, transformation, and integration of the data, rendering it suitable for advanced analysis. Feature extraction involved the extraction of informative features and insights from the energy consumption data,

allowing for a deeper understanding of the consumption behavior. The utilization of innovative techniques, such as ensemble methods, deep learning, and time series analysis, in the advanced modeling phase facilitated the enhancement of prediction accuracy. Finally, the creation

of interactive and insightful visualizations using advanced data visualization libraries like Plotly, Bokeh, and D3.js provided stakeholders with intuitive and user-friendly representations of energy consumption data.

**Phase 3: Development Part 1**

This phase focused on the implementation steps necessary for loading the dataset and the main script, laying the foundation for subsequent data analysis and visualization. The load dataset function

implementation involved the importation of essential libraries, definition of the load dataset function, and reading of the dataset. Error handling and preprocessing of the dataset were also integral parts of this phase, ensuring the dataset was processed and ready for analysis.The main script implementation steps encompassed the importation of the load dataset function, definition of the list of data paths, iteration through each data path, and error handling procedures. Customization of file paths was also highlighted, ensuring flexibility and adaptability to different dataset locations.

**Phase 4: Development Part 2**

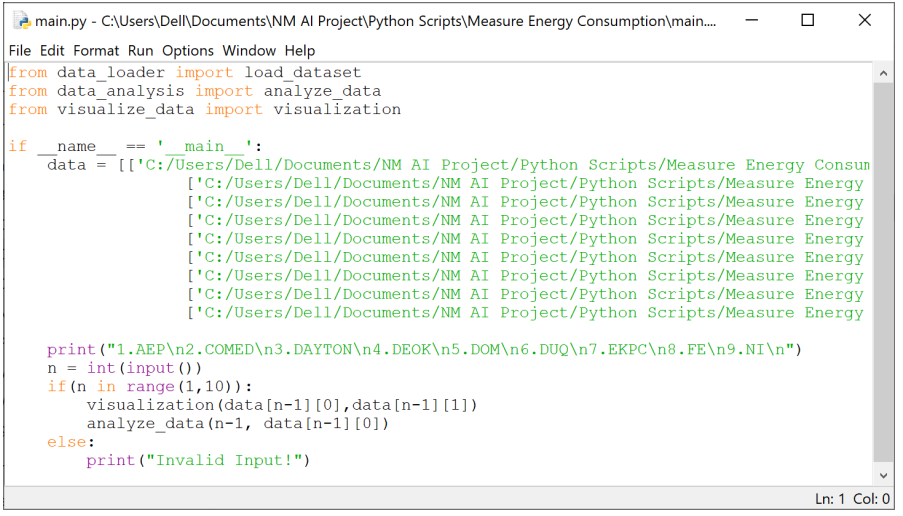
This phase focused on the continued development of the project, specifically on the tasks of analyzing energy consumption data

and creating visualizations. The implementation steps included the definition of the analyze data function, which involved the utilization of

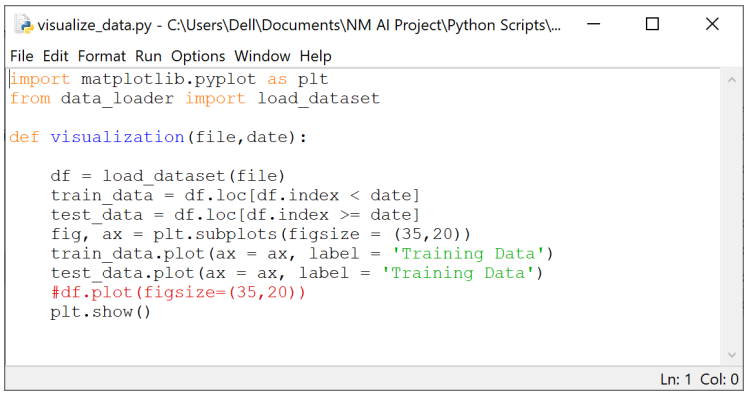
key libraries such as pandas, matplotlib.pyplot, and seaborn for data analysis and visualization tasks. Specific steps within this module included the extraction of time-related features, data analysis using

Seaborn's boxplot, and the display of visualizations using Matplotlib's plt.show().

**Main Script:**



**Data Visualization Implementation:**

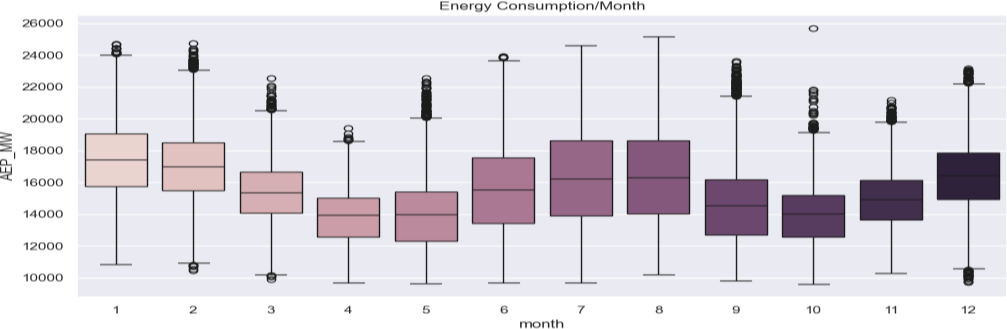
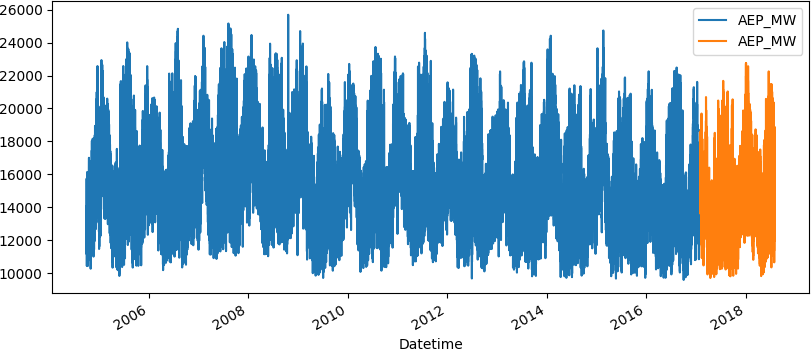


The visualization module focused on the importation of necessary libraries and the load dataset function, followed by the definition of the visualization function. Steps within this module included the separation of data into training and test sets, the creation of plots using Matplotlib's subplots, and the customization of labels. Finally, the main script involved the importation of necessary modules, definition of the main script, and error handling procedures to ensure seamless

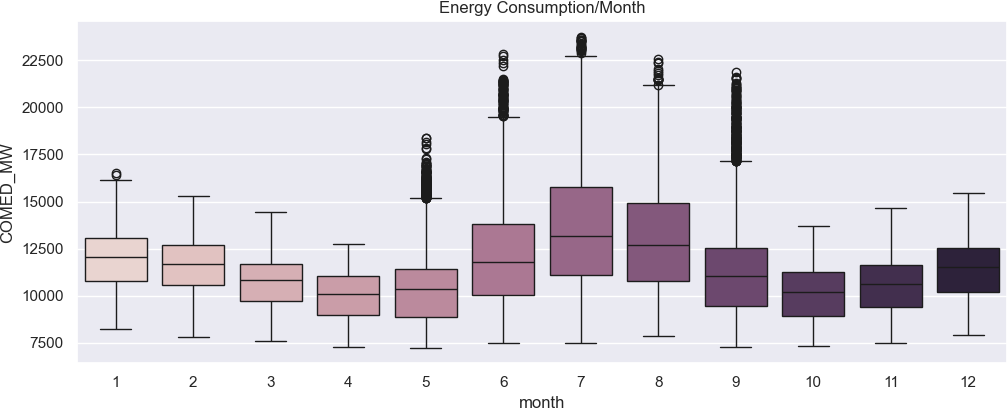
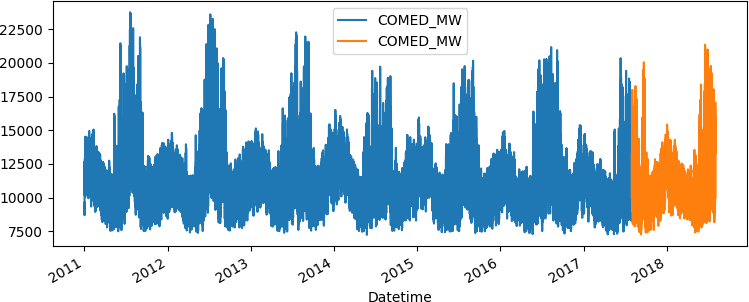
execution and user-friendly interactions.

# RESULTS FROM VISUALIZATION:

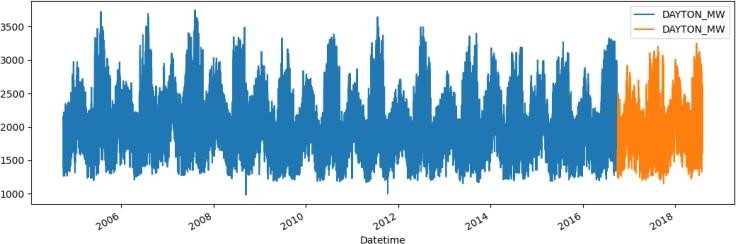
AEP:

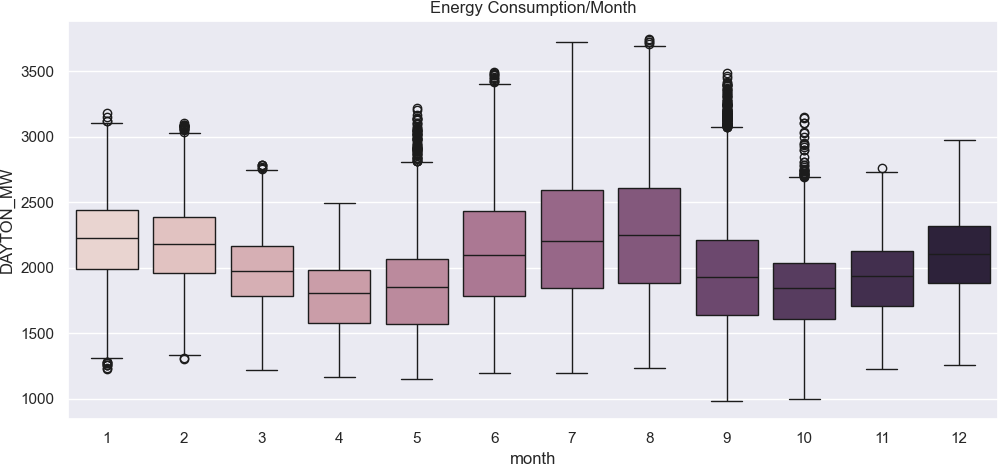


COMED:

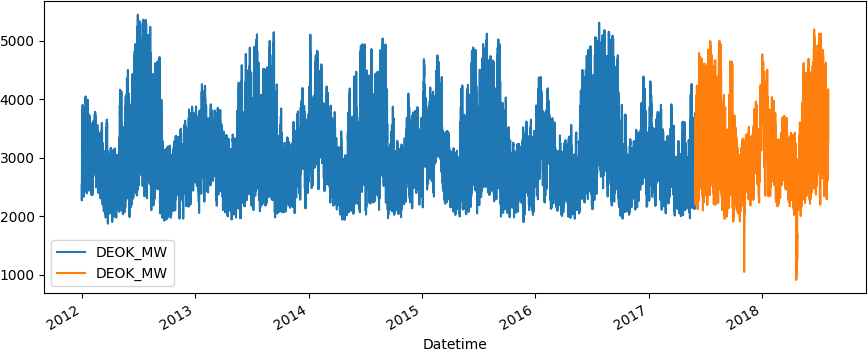


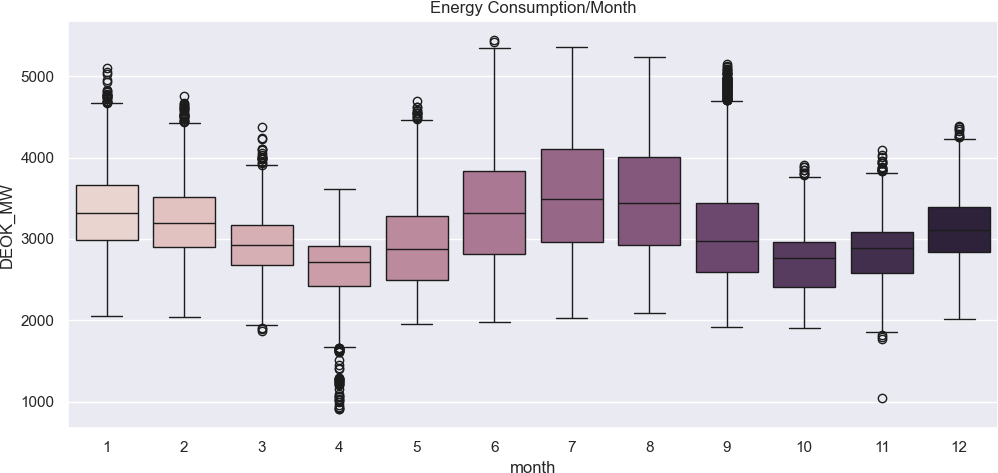
DAYTON:



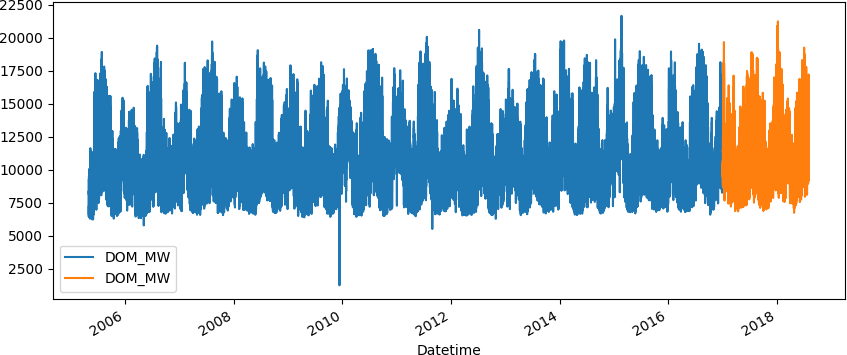


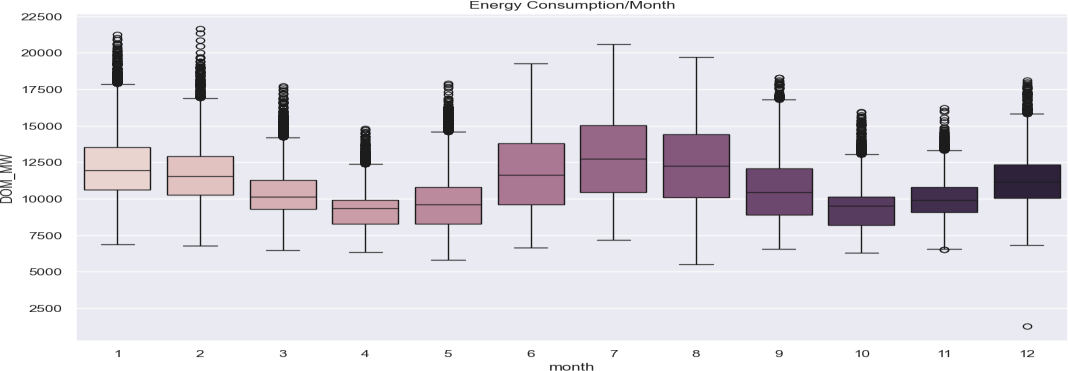
DEOK:



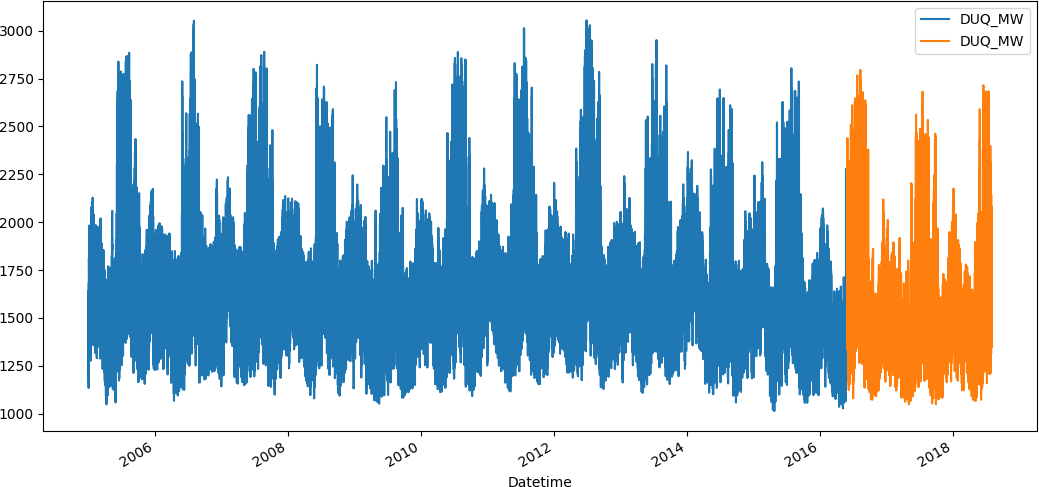


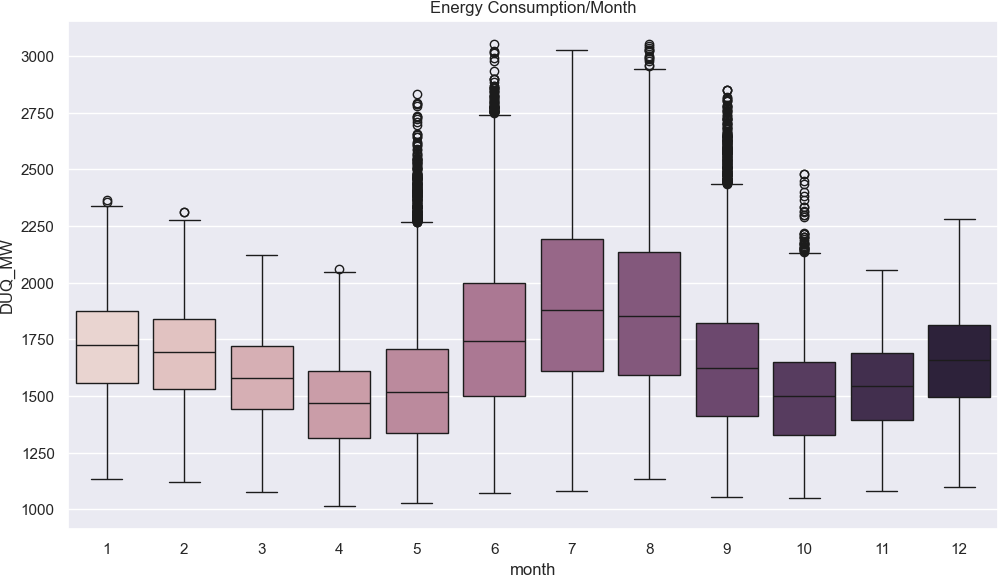
DOM:

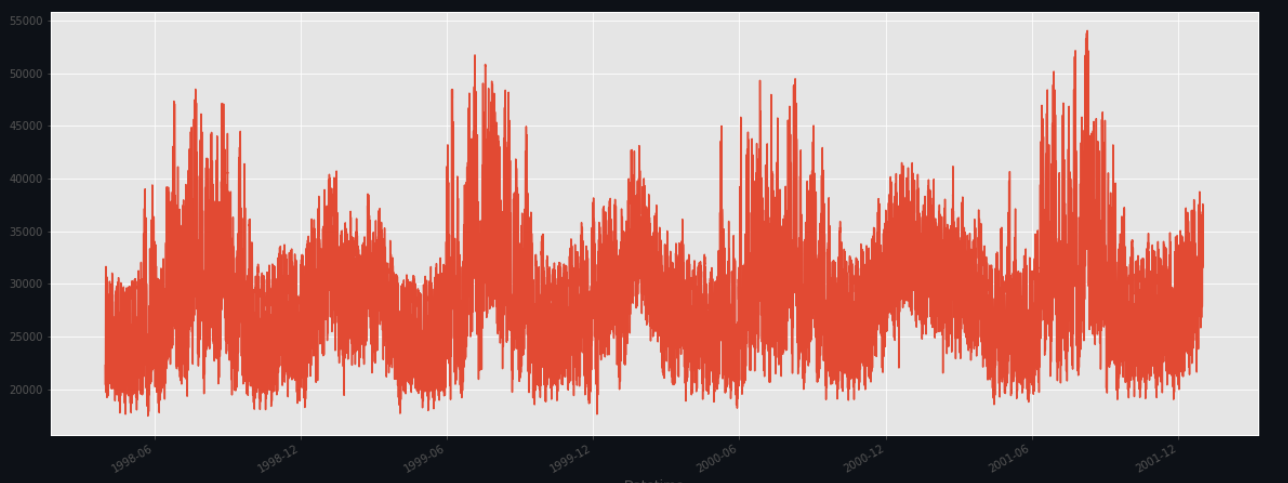




DUQ:







**Conclusion:**

Enhancing Energy Management Through IoT and Advanced Analytics

The "Measure Energy Consumption" project has been a journey of innovation and insight, with a clear focus on improving energy management across various sectors. By integrating Internet of Things (IoT) devices, advanced data analytics, and data visualization techniques, this project has brought us closer to achieving the objectives set forth at the outset.

Understanding Energy Consumption:

One of the fundamental goals of the project was to gain a comprehensive understanding of energy consumption patterns. Through the deployment of IoT devices, we've successfully collected real-time data on electricity usage, environmental conditions, and equipment health. This data has unveiled intricate details about energy consumption trends, fluctuations, and dependencies.

Improved Decision-Making:

With the data-sharing platform and advanced analytics, we've provided stakeholders, energy managers, facility operators, and homeowners with the tools needed for better decision-making. Insights derived from the data, whether through machine learning models, time series analysis, or anomaly detection, enable proactive and data-driven choices in energy management.

Promoting Sustainability:

The project is not just about data collection and analysis; it's also a significant step toward promoting sustainability. By identifying energy efficiency opportunities and providing actionable recommendations, we contribute to a more sustainable future. Reduced energy waste and optimized energy consumption benefit both the environment and the bottom line.

In conclusion, the "Measure Energy Consumption" project represents a commitment to harnessing technology and data for a sustainable and efficient future.