Household Electric Consumption Prediction using Machine Learning

Project Hand-out, Faculty Development Program

# HOUSEHOLD ELECTRIC CONSUMPTION PREDICTION USING MACHINE LEARNING

## Project Flow:

* - User inputs data (or data collected over time).
* - Data is analyzed and fed into a trained model.
* - Model predicts electricity consumption for given intervals.
* - Results are visualized on the UI.

## Milestone 1: Define Problem / Problem Understanding

* **Activity 1:** Specify the Business ProblemPredicting electric power consumption means using past records of electricity usage to forecast future usage. This is done through statistical and machine learning methods. The goal is to understand patterns and make accurate forecasts of how much electricity a household will consume in the future.
* **Activity 2:** Business Requirements By predicting future consumption patterns, utility companies can accurately estimate how much electricity will be needed at different times of the day or year. This helps avoid overproduction or underproduction, optimizing resource usage and reducing costs. Assist households in understanding consumption trends. Providing users with detailed insights into their electricity usage allows them to identify high-consumption periods or appliances. This empowers them to make smarter energy-saving decisions, ultimately lowering their electricity bills. Prevent overload and assist in grid load balancing. Predicting peak load times in advance helps grid operators distribute electricity efficiently across regions. It prevents grid failures, reduces the chances of blackouts, and ensures a smooth power supply even during high-demand periods
* **Activity 3:** Literature Survey Various studies in recent years have demonstrated that **machine learning (ML)** and **artificial intelligence (AI)** techniques outperform traditional statistical models such as **ARIMA** when it comes to forecasting electric power consumption, especially in the presence of non-linear and complex patterns.

Research papers have explored models like **Random Forest, XGBoost, Support Vector Regression (SVR), and LSTM (Long Short-Term Memory networks)** for predicting household power usage. These models have shown higher accuracy, better generalization, and robustness to noise.

For example, LSTM, a type of recurrent neural network, is highly effective for time-series data due to its ability to retain long-term dependencies. Studies found it particularly useful in capturing daily and weekly cycles in electricity usage.

Additionally, hybrid models that combine statistical approaches (like ARIMA) with ML models (e.g., ARIMA + XGBoost) have also been tested, achieving improved performance through ensemble learning.

Literature also suggests the importance of feature engineering (e.g., lag features, rolling averages, seasonal decomposition) in improving the predictive capability of ML models for electricity consumption data.

* **Activity 4:** Social or Business Impact Reducing electricity waste. By accurately predicting consumption patterns, both consumers and providers can identify inefficient usage and reduce unnecessary power consumption. This leads to more sustainable energy use and cost savings. Empowering smart homes and IoT devices.Machine learning models integrated with smart home systems can automate decisions such as when to run appliances, charge batteries, or switch off unused devices—leading to a more intelligent, responsive, and energy-efficient living environment. Enabling real-time usage analytics.Real-time predictions and consumption insights help users track their electricity usage on dashboards or apps. This awareness enables better energy budgeting, preventive maintenance, and smarter electricity planning at both individual and community levels.

## Milestone 2: Data Collection & Preparation

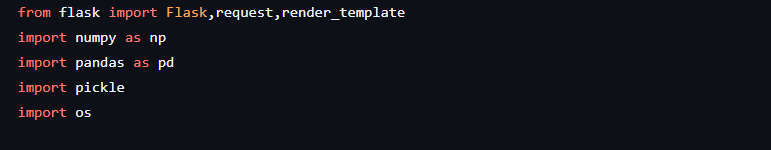
* **Activity 1: Collect the Dataset**

This activity focused on sourcing and importing the dataset required for the project. The dataset used is the *‘Individual household electric power consumption Data Set’* obtained from the **UCI Machine Learning Repository**, which provides a comprehensive record of household electricity usage measured over a period of nearly four years.

**Dataset Source: ‘Individual household electric power consumption Data Set’**

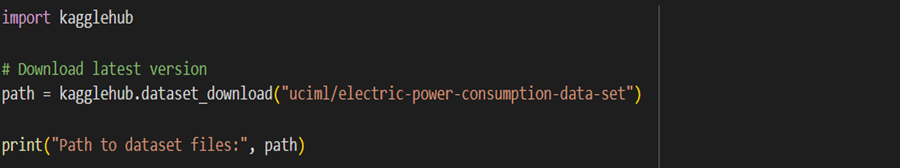
Steps Taken in Notebook:

**1.1: Imported Libraries:**Python libraries like pandas, numpy, and matplotlib were imported to handle data numerical operations, visualizations.



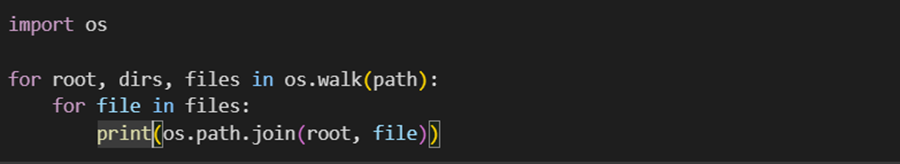
**1.2: Dataset Downloaded via kagglehub:**

The dataset was programmatically accessed using the kagglehub utility to ensure reliable and up-to-date retrieval.



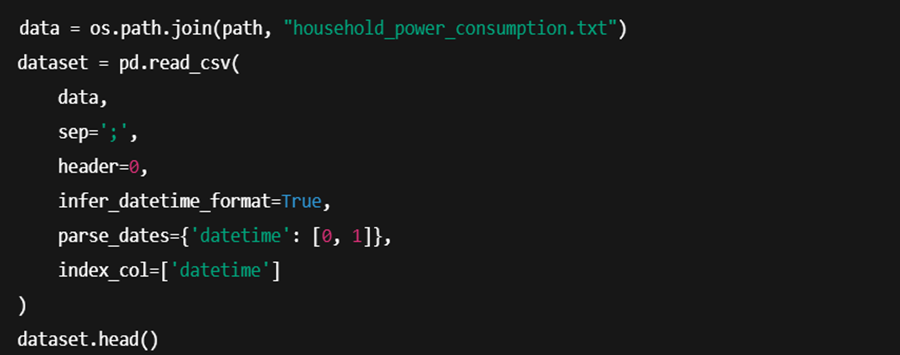
**1.3:Explored Files in Dataset Directory:**

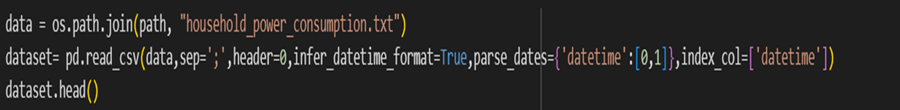
The contents of the downloaded dataset directory were listed and verified to confirm the correct file (household\_power\_consumption.txt) was present.

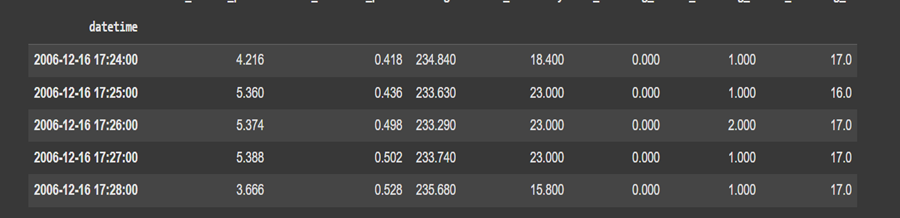


**1.4: Loaded the Dataset into a Pandas DataFrame:**

The dataset was loaded into a DataFrame using pandas.read\_csv() with necessary parameters to handle separators, missing values, and time-based columns.







**1.5: Checked the Shape of the Dataset:**

The number of rows and columns was verified to confirm successful loading and completeness of data.



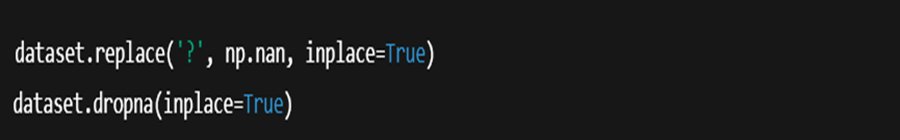
* **Activity 2: Data Preparation**

Before training machine learning models, the raw data was preprocessed to ensure quality, consistency, and relevance of features. This step is essential to improve model accuracy and interpretability.

**2.1:** **Handle missing values (rows with '?').**

Missing or undefined values in the dataset (represented as '?') were replaced with NaN and subsequently removed. This step helps eliminate potential sources of noise and ensures the dataset is clean for analysis.

Missing values can distort analysis and must be removed or imputed.



**2.2:Convert Date and Time into datetime object.**

Converting Date and Time to a datetime format enables time-based operations.

The Date and Time columns were merged and converted into a single datetime object. This enables the dataset to be used in time-series analysis and feature extraction.



**2.3: Convert categorical to numerical values where applicable**.

Machine learning models require numerical input for effective processing.

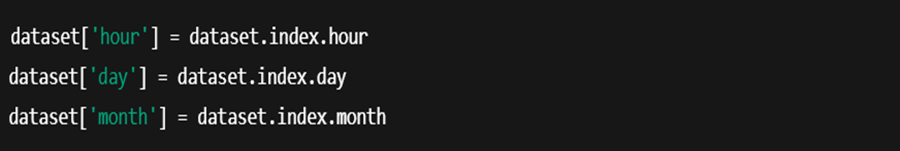
Though the dataset is mostly numeric, any non-numeric values were checked and converted to numeric form using typecasting. This is important because machine learning algorithms require numerical inputs.



**2.4: Create time-based features (hour, day, month).**

Time-based features help identify seasonal, daily, or hourly trends.

These new features such as hour, day, and month were derived from the datetime column. These features help capture periodic consumption patterns and improve the predictive power of the model.



## Milestone 3: Exploratory Data Analysis

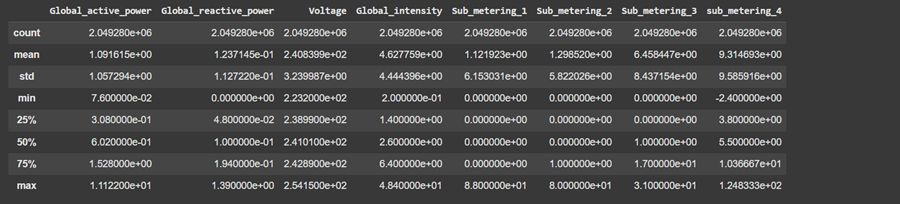
* **Activity 1:Descriptive Statistics**

Understanding the statistical summary of the dataset helps in identifying the central values, variability, and data distribution. It also reveals potential anomalies or unusual values that need further analysis.

**1.1:Summary statistics of Global Active Power**

Descriptive statistics such as mean, median, standard deviation, and quartiles help summarize the general pattern and distribution of power usage in kilowatts.





**1.2:Summary statistics of Sub-metering 1, 2, and 3**

Analyzing sub-metering values shows how energy is consumed across different household appliances or zones, aiding in consumption pattern recognition.

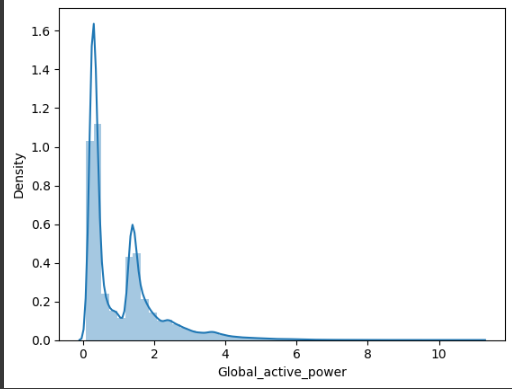
* **Activity 2**: **Visual Analysis**

Visualization techniques help to intuitively understand trends, variability, and anomalies in the data, which are not always obvious in raw tables or statistics.

**2.1:Line plots for time vs power usage.**

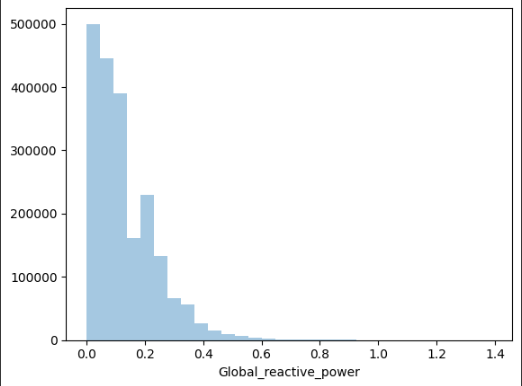
Line plots visually represent electricity consumption over time, highlighting fluctuations, trends, and possible seasonality.

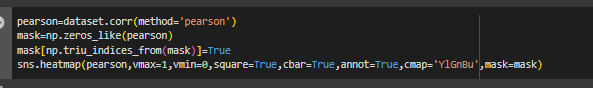


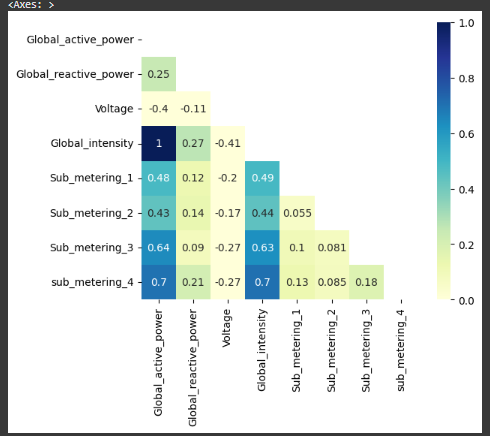


**2.2:Histograms for distribution of consumption.**

Histograms provide a visual distribution of consumption values, showing how frequently each consumption range occurs.





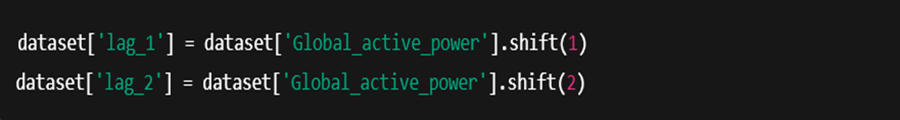


* **Activity 3:Feature Engineering**

Feature engineering transforms raw data into meaningful inputs that enhance model accuracy and interpretability.

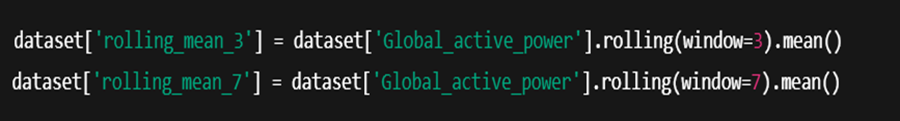
**3.1:Creation of lag features**.

Lag features use previous time steps as predictors to capture temporal dependencies in energy consumption.



**3.2:Rolling average features.**

Rolling averages smooth out short-term variations and highlight long-term trends, improving model learning for sequential data.



## Milestone 4: Model Building

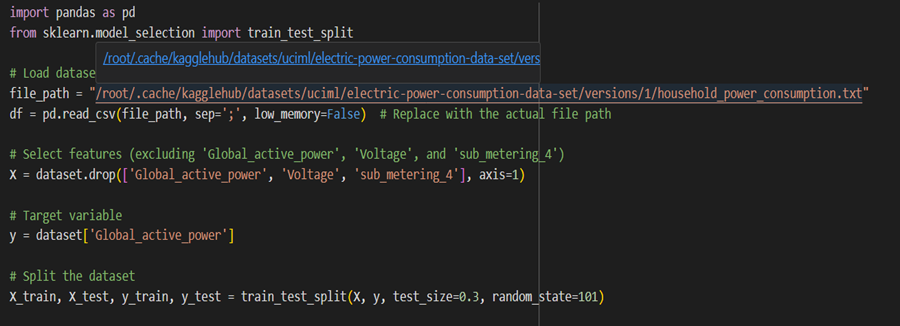
* **Activity 1:Training with Multiple Algorithms**

Model building involves training multiple regression models on the processed dataset to predict future electricity consumption and compare their performance.

**1.1:Split the data into training and testing sets**

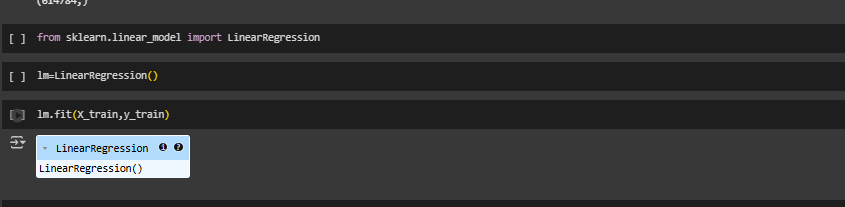
Dividing the dataset ensures model evaluation is performed on unseen data, preventing overfitting and bias.





**1.2:Train a Linear Regressor**

A Linear regressor is trained upon the cleaned dataset.

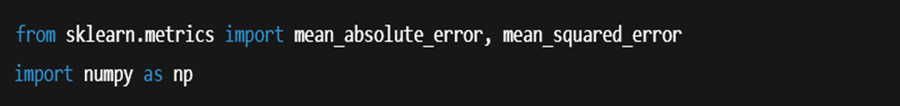


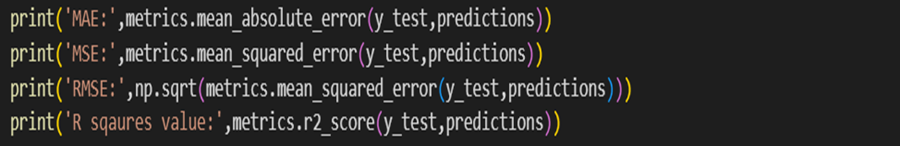
* **Activity 2:Evaluation**

Evaluation helps compare models and measure how well predictions match actual values using performance metrics.

**2.1:Import evaluation metrics**

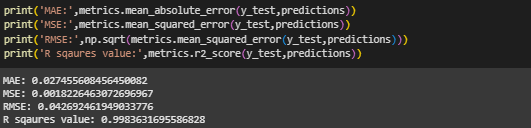
Metrics such as MAE, MSE, and RMSE are used to quantify prediction errors.





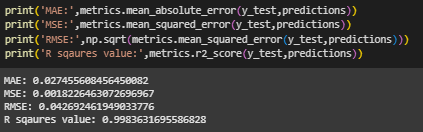
**2.2:Evaluate the Linear Regressor**

Each model is evaluated on test data using the above metrics to determine predictive performance and generalization ability.



**2.5: Print and see the result**

A comparison of various metric values helps identify the performance.

.

## Milestone 5: Performance Testing

Once the models are trained, it's crucial to evaluate their generalization capability and fine-tune their performance. This milestone focuses on comparing the trained model using cross-validation and then optimizing them through hyperparameter tuning to achieve the best predictive accuracy.

* **Activity 1:Compare performance**

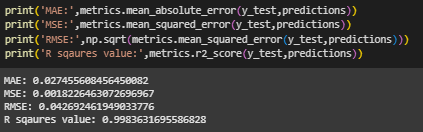
After training model, it’s important to evaluate how it performs on unseen data.We compare models using test set results and apply cross-validation to ensure the model generalize well across different data splits.

**1.1:Print evaluation metrics on test data**

The trained model was first evaluated using the test set (20% of the data). Key evaluation metrics used were:

* **MAE (Mean Absolute Error):** Average magnitude of prediction errors.
* **MSE (Mean Squared Error):** Penalizes larger errors by squaring them.
* **RMSE (Root Mean Squared Error):** Most commonly used; interpretable in the same units as the target variable.

This step gives a quick overview of how well the model performs on unseen data.



## Milestone 6: Model Deployment

After evaluating and fine-tuning the machine learning models, the final step in this project involves deploying the best-performing model so that it can be used in a real-world scenario. Deployment bridges the gap between model development and user interaction by allowing predictions to be made dynamically based on user input. This milestone focuses on saving the trained model and building a web-based user interface using Flask — a lightweight Python web framework.

* **Activity 1: Save the Best Model**Once the model was finalized after hyperparameter tuning, it was saved for later use without the need to retrain it each time.

Objective: To preserve the trained model in a reusable format for real-time prediction.

Implementation: The model was saved using the joblib library as a .pkl (pickle) file.

* **Activity 2:Flask Web App**

A web-based application was developed using Flask to allow users to interact with the model through a user-friendly interface.

**Frontend Functionality:**

* The UI was designed using HTML/CSS and allows the user to enter key input features such as:
  + Date and time (which are converted into time-based features)
  + Sub-metering or environmental parameters (if required)

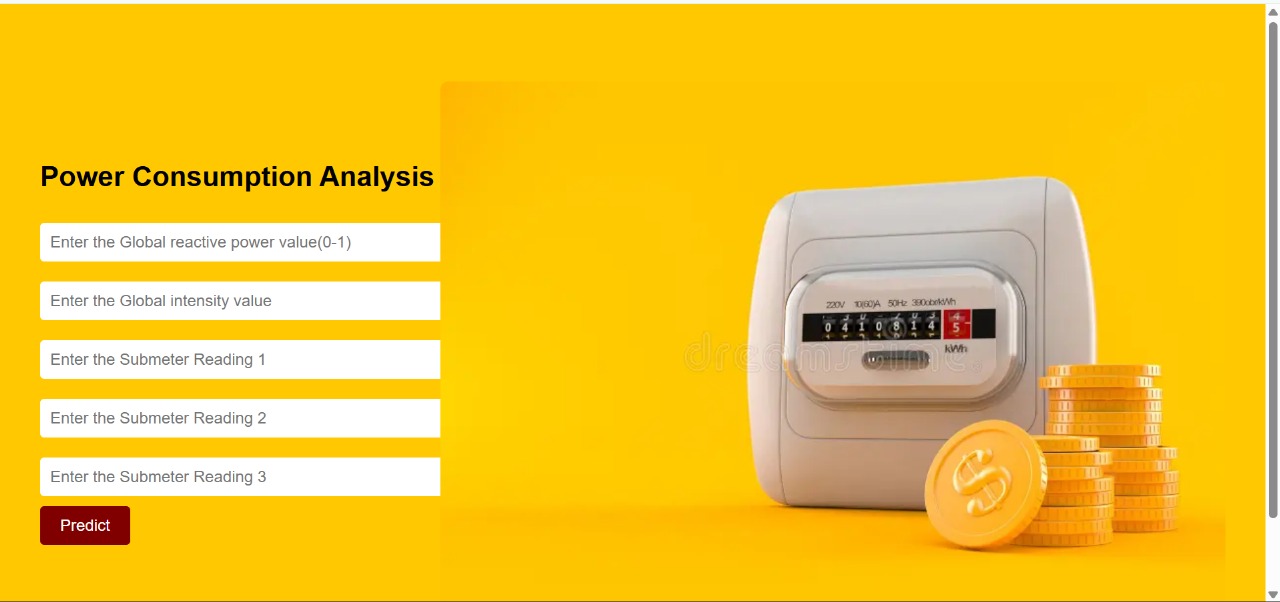
**Backend Functionality:**

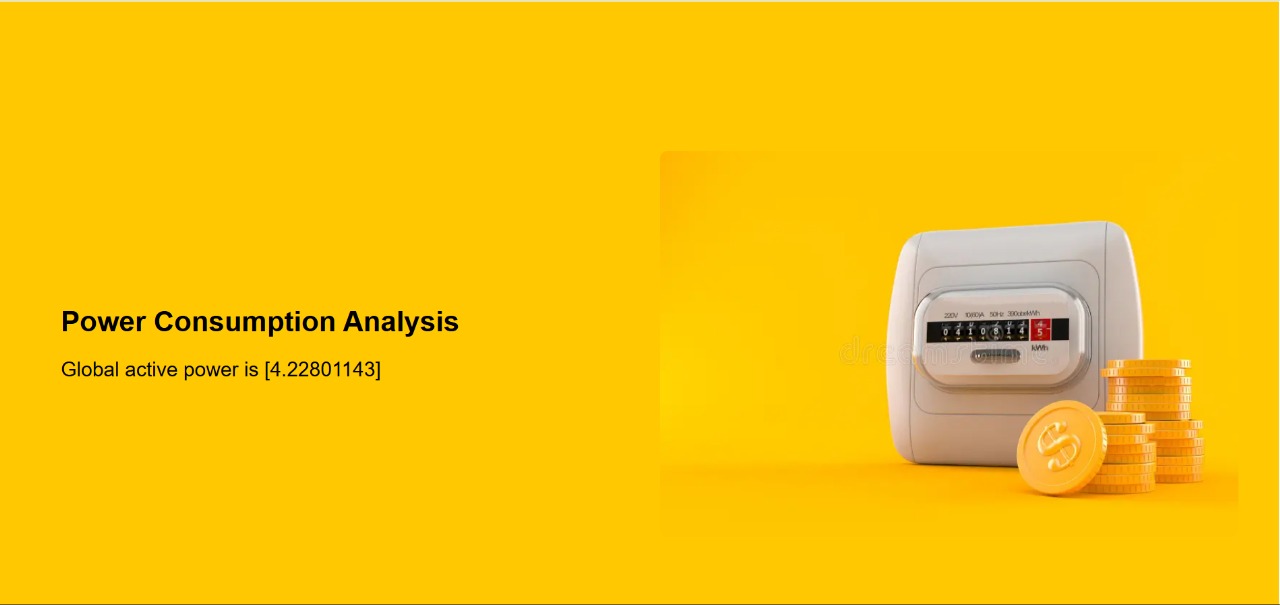
* Flask handles routing and backend logic.
* The trained model (model.pkl) is loaded into the Flask app using joblib.load().
* When the user submits input through the form, the data is processed and passed to the model, which returns a prediction.

**Prediction Display:**

* The predicted electricity consumption (in kilowatts) is displayed on the webpage in a readable format.
* This simulates a real-world application where a household user or utility provider could get instant feedback on expected power usage.

**Screenshots Of Project**





Conclusion:

## This project is a good demonstration of an end-to-end, practical, and full implementation of a machine learning pipeline for household electric power consumption forecasting. From defining the problem and business requirement analysis, the project methodically went through phases that included data gathering, preprocessing, exploratory data analysis (EDA), feature engineering, model development, performance assessment, hyperparameter optimization, and deployment.

## Through the use of the 'Individual household electric power consumption' data from the UCI Machine Learning Repository, we were able to derive useful patterns from actual electricity consumption data. Through the application of statistical methods and machine learning models — such as Decision Tree, Random Forest, and XGBoost Regressors — it was possible to create reliable predictive models. Cross-validation and hyperparameter tuning improved the reliability and generalizability of the models.

## With visual observation, summary statistics, and feature engineering methods such as lag and rolling average features, we gained a better understanding of temporal patterns in energy consumption behavior. The effective application of metrics like MAE, MSE, and RMSE enabled us to comparatively evaluate models objectively and choose the best performing model for rollout.

## The model was implemented in a real-world setting using Flask to create a basic web interface where users can enter date/time attributes and get instant consumption predictions. This transforms the project from a theoretical exercise to an actionable solution with real-world applications.

## At a higher level, this project has substantial social and business relevance:

## • It provides utility providers with prediction tools for demand, resulting in improved resource planning and grid management.

## • It enables homes to learn about and optimize electricity consumption, assisting energy-saving choices.

## • It helps advance sustainability aims by encouraging effective energy use, eliminating waste, and facilitating smart home automation by predictive analysis.

## In summary, this project well illustrates how data-intensive technologies such as machine learning can be applied to real-world problems in the energy industry. The experience gained — in dealing with data, model building, testing, and deployment — is applicable to many real-world arenas, providing a solid foundation for subsequent work in predictive modeling and smart energy management.

## Links for Reference:

🔹UCI Dataset: <https://archive.ics.uci.edu/ml/datasets/individual+household+electric+power+consumption>

This dataset provides real-world electric power consumption measurements from a single household, forming the foundation for model training and analysis in this project.

* Scikit-learn Documentation: <https://scikit-learn.org>

Scikit-learn is a widely used Python library offering efficient tools for data mining, machine learning model building, evaluation, and hyperparameter tuning.

* Flask tutorial: <https://flask.palletsprojects.com>

Flask is a lightweight Python web framework used in this project to deploy the trained machine learning model and build a simple web interface for predictions.