POWER PULSE: HOUSE HOLD ENERGY USAGE FORECAST

1. Project Overview:

This project aims to build an accurate predictive model for household power consumption using time-series and numerical data. The final objective is not only to predict energy usage but also to understand the underlying patterns and feature influences.

Key Goals:

- Build an accurate regression model to forecast household energy consumption.
- Identify and interpret the most influential factors affecting household energy usage.
- Provide data-driven insights into daily and seasonal consumption trends.
- Enable visualization of energy behavior and prediction accuracy.

Tools Used:

- Python (Pandas, Matplotlib, Seaborn)
- Power BI
- Scikit-learn (Linear Regression, Random Forest Regressor, Gradient Boosting Regressor, MLP Regressor)

2. Data Preparation:

Dataset:
https://archive.ics.uci.edu/dataset/235/individual+household+electric+power+consumption

a. Cleaning:

- Loaded dataset and converted numeric columns using numeric function
- Missing values were handled using **median imputation**.

- Dropped "**Global_intensity"** due to **very high correlation** with target variable (Global_active_power).

b. Outlier Handling:

- Used **z-score** method (threshold = 3) to detect and remove extreme outliers.

c. Parsing & Transformation:

- Parsed and merged Date and Time into a datetime object.
- Set as index and extracted features: hour, month, day, is_weekend, etc.

d**. Feature Scaling:**

- Applied "MinMaxScaler" to normalize features for better model convergence.

3. Feature Engineering:

- Created time-based features: hour, month, is_weekend.
- Added "**daily_avg_power"** and "**rolling_avg_power"** (60-minute window).
- Created "**is_peak_hour"** based on whether consumption exceeded the daily average.
- Dropped redundant features post feature extraction.

4. Model Selection and Training:

Model	Random State	Hyperparameters
Linear Regression	30	default
Random Forest Regressor	30	max_depth=10
Gradient Boosting Regressor	30	learning_rate=0.1
MLP Regressor (Neural Net)	30	hidden_layer_sizes=(50,), max_iter=300

Train-Test Split:

- Dataset split into **80% training** and **20% testing** using train_test_split.

- **Random State = 30** was used to ensure **consistent data splits and reproducible results** for training and visualizations.

5. Model Evaluation Metrics:

Metric	Linear Regression	Random Forest Regressor	Gradient Boosting Regressor	MLP Regressor
Train RMSE	0.2170	0.1954	0.1955	0.1998
Test RMSE	0.2168	0.1959	**0.1954**	0.1997
Train MAE	0.0317	0.0237	0.0237	0.0246
Test MAE	0.0316	0.0237	**0.0237**	0.0246
Train R^2	0.7553	0.8391	0.8389	0.8241
Test R^2	0.7564	0.8377	**0.8393**	0.8247

Best Model: **Gradient Boosting Regressor**

- **Lowest RMSE** and **MAE**, indicating better predictive accuracy.
- **Highest R^2** Score, meaning it explains **83.9%** of the variance in energy usage.
- **Less overfitting** compared to Random Forest and MLP (similar train vs test scores).

6. Insights and Recommendations:

Insights

- **Sub_metering_3** is the **most important predictor** of overall household energy usage, followed by Sub_metering_1 and Sub_metering_2.
- **Time-based features** like hour and month also influence energy consumption patterns, indicating temporal trends in usage.
- **Voltage and Global Reactive Power** contribute comparatively less to predicting active power consumption.
- The **correlation analysis** showed a very strong linear relationship between **Global_active_power** and **Global_intensity**, leading to the removal of the latter to avoid multicollinearity.

- Gradient Boosting Regressor was able to capture non-linear relationships in the data and delivered the most **accurate predictions** among all tested models.
- The **Actual vs Predicted plot** revealed that predictions are generally aligned with true values but tend to underpredict at higher values

Recommendations

- **Adopt Gradient Boosting Regressor** as the model for forecasting short-term household energy consumption due to its strong performance.
- Continuously update the model with new data and periodically retrain it to maintain prediction accuracy.
- Use real-time data preprocessing pipelines to handle missing values, outliers, and normalization automatically.
- Consider deploying this model as part of an **energy monitoring dashboard** to assist households in understanding usage patterns and saving energy.
- Incorporate **peak-hour detection logic** into user-facing applications to alert households about high-usage periods.
- Further exploration can include integrating weather data, appliance-level monitoring, or occupancy sensors to improve prediction accuracy.