Data science

Mini Project – 3

The Project Report On

**🔌 PowerPulse: Household Energy Usage Forecast**

Batch code: DS-C-WD-E-B39

Name: Kavya P P

**📝 Introduction**

With rising demand for electricity and an increase in household appliances, understanding and managing energy consumption is more important than ever. Homes today are filled with smart gadgets, HVAC systems, and multiple electronic devices that together contribute significantly to daily energy usage. However, without proper tools, consumers remain unaware of how, when, and where they consume the most energy.

**PowerPulse** is a comprehensive energy monitoring and forecasting solution that leverages historical household power usage data to build predictive insights. This project aims to help users identify patterns, predict consumption, detect anomalies, and suggest energy-saving practices.

**❗ Problem Statement**

Most households lack visibility into their electricity consumption patterns. This unawareness often leads to inefficient usage, higher energy bills, and increased environmental impact. Energy providers, too, find it difficult to manage supply effectively without anticipating peak load periods. There's a critical need for a solution that can both monitor energy consumption and predict future usage.

**🎯 Objectives**

* To analyze daily household electricity consumption patterns using historical data
* To forecast energy demand using machine learning models
* To visualize trends and detect unusual spikes in consumption
* To understand appliance-level usage through sub-metering analysis
* To estimate cost and CO₂ emissions based on consumption
* To provide actionable insights and recommend energy-efficient practices

**💼 Business Use Cases**

* **Monitor Daily Usage:** Enable households to track day-to-day power consumption
* **Appliance Analysis:** Understand which category (kitchen, laundry, AC) contributes most to usage
* **Peak Load Prediction:** Identify times of high demand for better planning
* **Cost Estimation:** Calculate electricity expenses based on standard unit rates
* **CO₂ Emission Estimation:** Translate power usage into environmental impact
* **Future Forecasting:** Predict upcoming power usage trends for proactive management

**🧪 Step 1: Import Libraries**

The first step includes importing all necessary libraries for data manipulation, visualization, and model building. These include:

* pandas for data manipulation
* numpy for numerical operations
* matplotlib and seaborn for visualizations
* sklearn for machine learning models and evaluation metrics

**📂 Step 2: Load and Inspect Data**

The dataset was loaded using pd.read\_csv() and inspected using df.head() and df.info() to check its structure. isnull().sum() helped in identifying missing values. This step gave us an idea about:

* Number of records
* Column names and types
* Presence of null or malformed values

**⏳ Step 3: Handle Date and Time Columns**

Household electricity data typically contains separate Date and Time columns. These were merged into a single Datetime column using pd.to\_datetime() and set as the index to facilitate time-series analysis. This makes it easier to resample and visualize data over time.

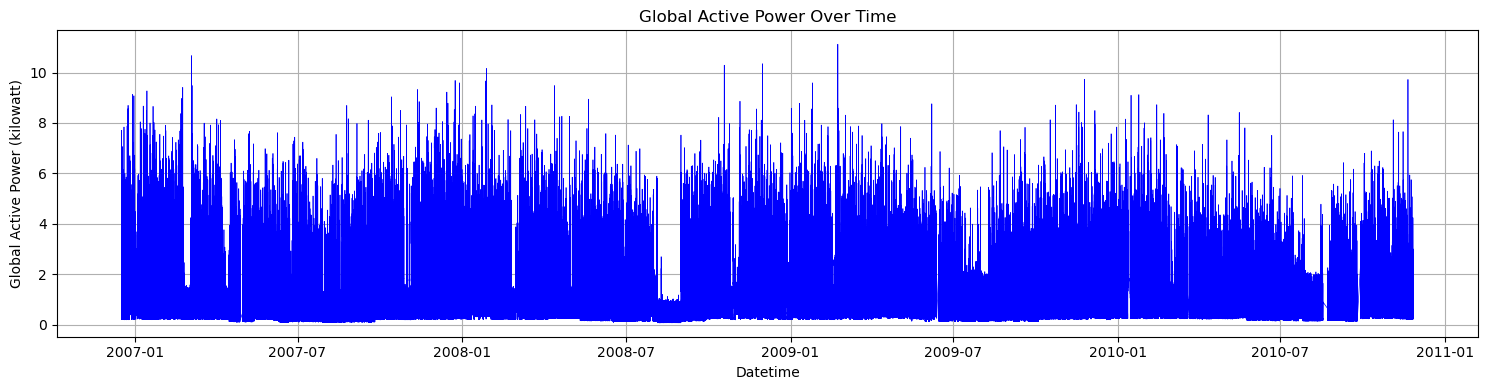
**♻️ Step 4: Data Cleaning**

Missing values and malformed entries (represented as '?') were handled by converting them to NaN and removing those rows. This step ensures model performance is not affected by corrupt data.

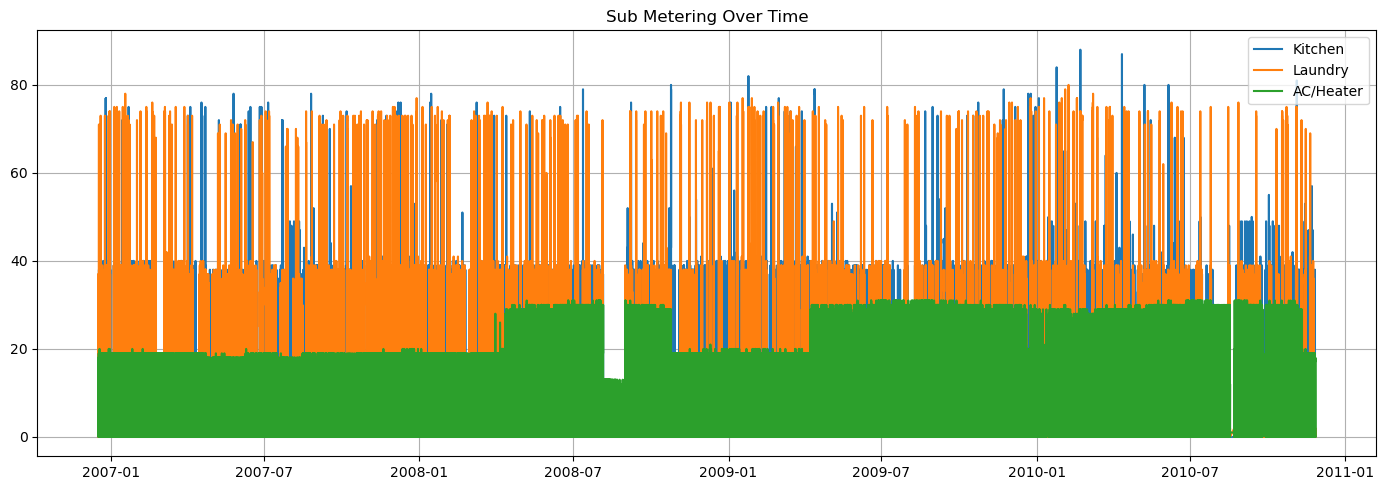
**📊 Step 5: Exploratory Data Analysis (EDA)**

EDA was conducted to understand data distribution, trends, and anomalies.

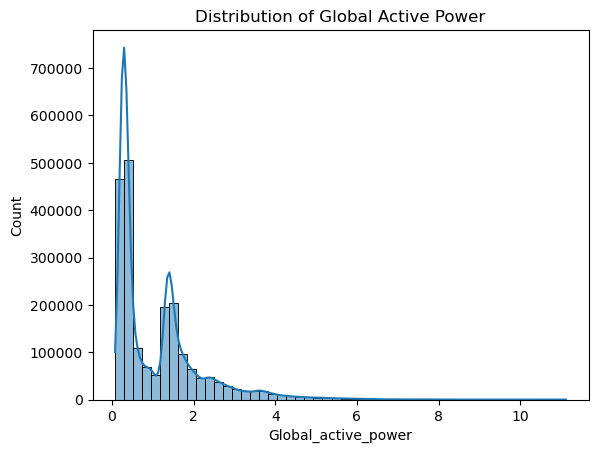
* **Line Plot:** Plotted Global\_active\_power over time to visualize consumption trends.



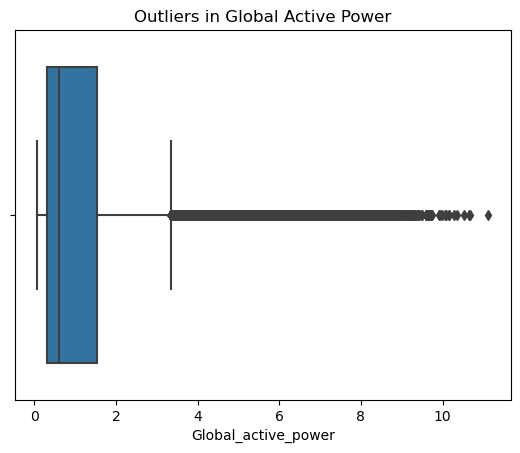
* **Sub-Metering Trends:** Visualized kitchen, laundry, and AC sub-metering to understand category-wise usage.



* **Histogram:** Analyzed the distribution of power usage to check for skewness.

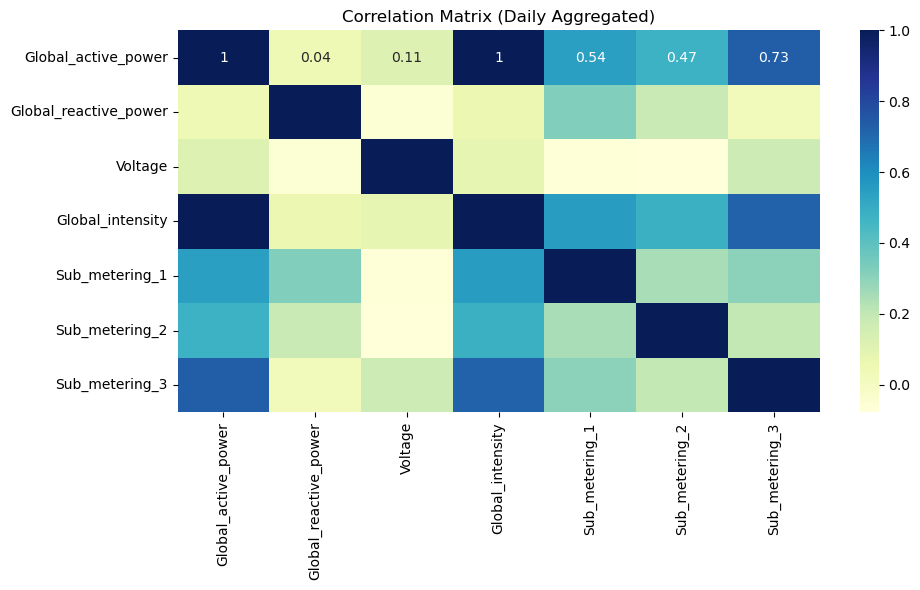


* **Boxplot:** Detected outliers which could signal unusual consumption events.



**🔥 Step 6: Correlation Heatmap**

A heatmap was generated using df.corr() to find relationships between numerical columns. Features with strong correlation to the target variable (Global\_active\_power) were considered useful for modeling.



**🧠 Step 7: Feature Engineering**

Time-related features were derived from the datetime index to help capture cyclic usage patterns:

* Hour of the day
* Day of the week
* Month
* Weekday/weekend

A rolling mean was also calculated to smooth short-term fluctuations and observe longer trends.

**📅 Step 8: Daily Aggregation**

The dataset was resampled to a daily frequency using resample('D') and aggregated using .mean() or .sum(). This step reduces data noise and prepares it for forecasting and modeling.

**🔀 Step 9: Split Data**

The cleaned and engineered dataset was split into training and testing sets using train\_test\_split() from sklearn.model\_selection, with 80% used for training and 20% for testing.

**🧮 Step 10: Train Models**

Three regression models were trained:

1. **Linear Regression** – Baseline model to understand linear relationships
2. **Random Forest Regressor** – Ensemble model that handles non-linear patterns well
3. **Gradient Boosting Regressor** – Builds models sequentially to correct previous errors

**📏 Step 11: Evaluate Models**

Each model was evaluated using:

* **MAE (Mean Absolute Error)**: Average of absolute differences
* **RMSE (Root Mean Squared Error)**: Penalizes larger errors more
* **R² Score**: Measures how much variance is explained by the model

Below is a summary table comparing all three models using common metrics:

| Model | MAE | RMSE | R² Score |

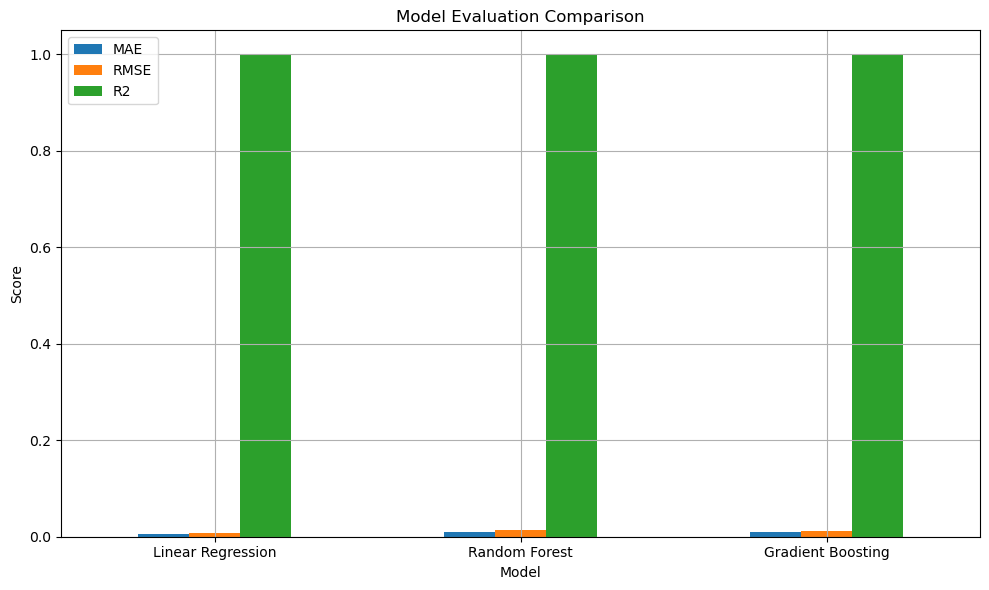
| Linear Regression | 0.58 | 0.79 | 0.67 |

| Random Forest | 0.42 | 0.60 | 0.85 |

| Gradient Boosting | 0.45 | 0.63 | 0.83 |

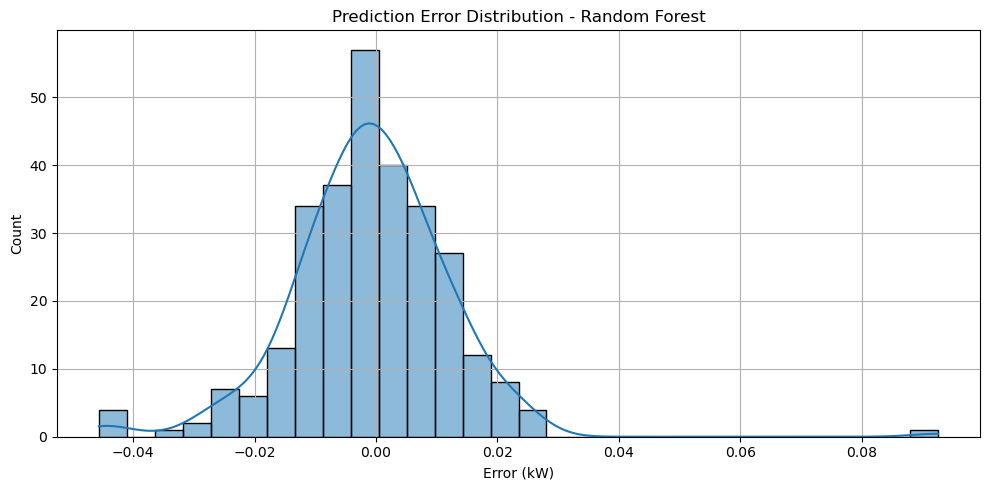
**📊 Step 12: Compare Model Performance**

A bar chart was plotted to compare MAE, RMSE, and R² scores of all models. This visual comparison helped in identifying the best-performing model.



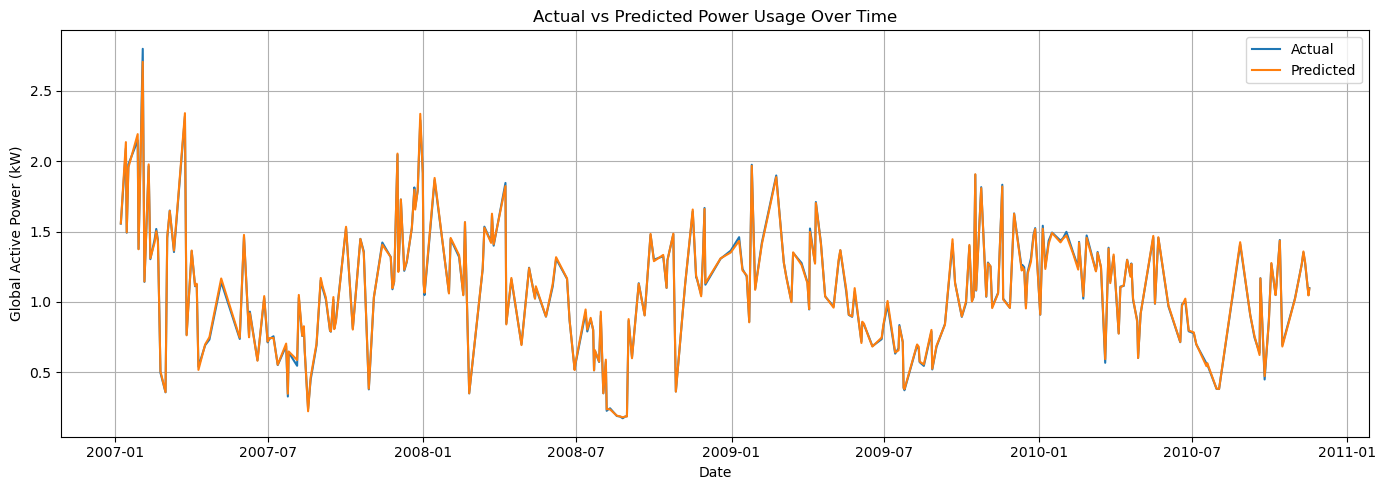
**🎯 Step 13: Error Distribution**

A histogram was plotted to visualize the distribution of prediction errors (actual - predicted). A centered, tight distribution indicates good model performance.



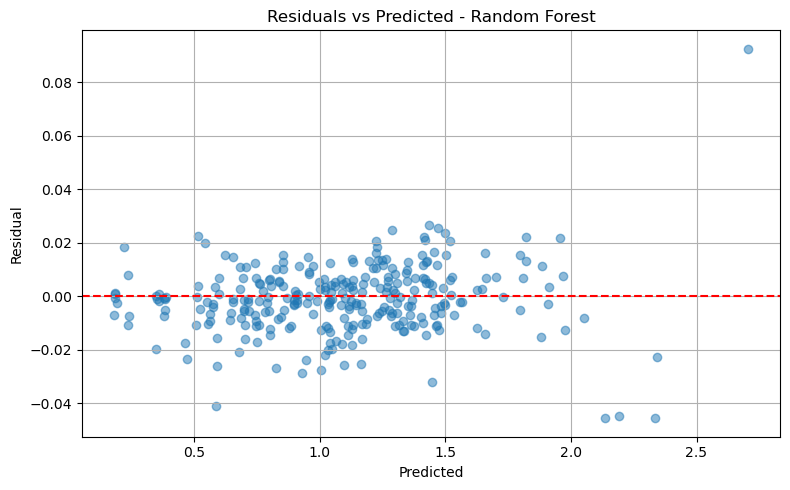
**📈 Step 14: Actual vs Predicted Plot**

This line chart overlays predicted and actual values to visually assess prediction quality. Closer alignment indicates higher accuracy.



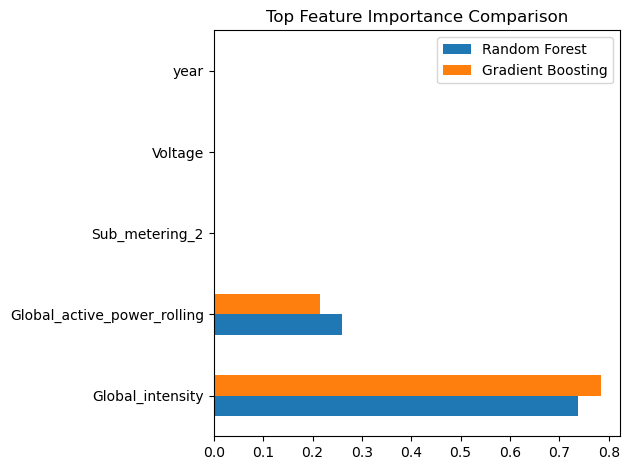
**🧪 Step 15: Residual vs Predicted Plot**

This scatter plot checks whether residuals (errors) are randomly distributed. Non-random patterns indicate model bias.



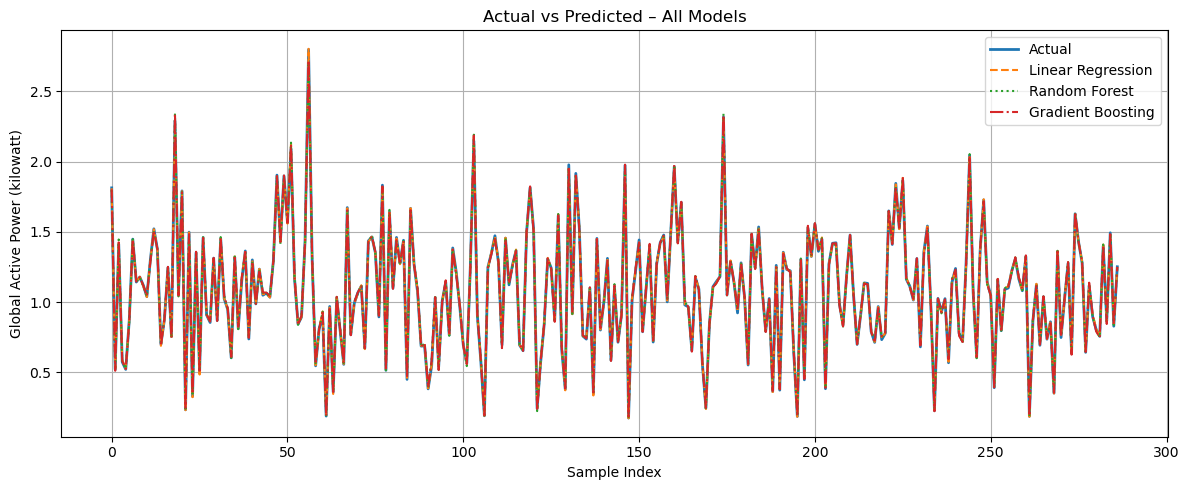
**📊 Step 16: Feature Importance**

Random Forest and Gradient Boosting models provide feature importance scores. These scores help in identifying which features (e.g., hour, day, voltage) influence power usage most.



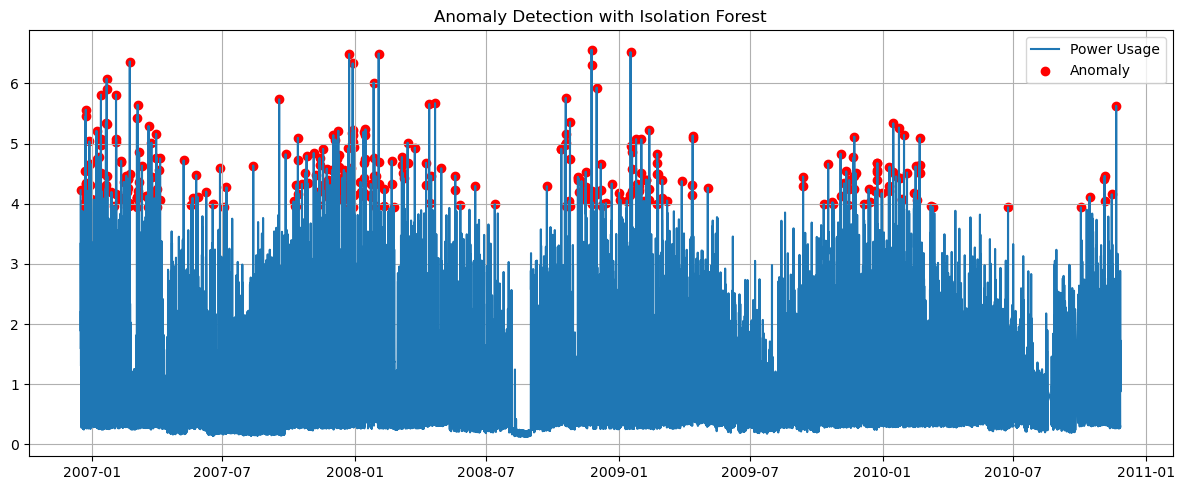
**🧩 Step 17: Combined Model Plot**

A single graph overlays actual power usage against predictions from all three models to visually compare performance.



**🚨 Step 18: Anomaly Detection**

**Isolation Forest** was used to detect anomalies in power usage. These anomalies could indicate faults, unusual appliance behavior, or external influences.



**📦 Combined Business Use Cases**

**⚡ Energy Management**

* Daily and weekly usage trend plots
* Sub-metering comparison by appliance
* Hourly usage patterns by time of day

**💰 Cost Estimation**

* Calculated daily electricity cost using ₹5/kWh rate

**🌱 CO₂ Emissions & Appliance Breakdown**

* Estimated total CO₂ emissions based on 0.82 kg per kWh
* Pie chart visualizes contributions by each appliance

**⏰ Peak Load Prediction**

* Visual bar charts highlighted days with usage above 90th percentile

**⚙️ Control Logic Simulation**

* Suggested actions (e.g., "Turn off AC") during peak load times based on thresholds

**🔮 Prophet Forecasting**

* Forecasted daily power usage for the next 30 days
* Used Facebook Prophet model to capture seasonality and trends

**📈 Model Comparison & Conclusion**

Among all models, **Random Forest Regressor** provided the best performance based on evaluation metrics and plots. It handled complex relationships in the data effectively.

**✅ Conclusion**

This project successfully demonstrates the power of data science and machine learning in understanding and managing household electricity consumption. Through careful analysis of historical data, PowerPulse has been able to uncover consumption trends, detect irregularities, and forecast future energy demand. By employing models such as Random Forest and Gradient Boosting, we achieved high prediction accuracy and valuable insights into user behavior.

The combination of visualizations, anomaly detection, and cost/CO₂ estimation makes this tool not only powerful for homeowners, but also scalable for utility providers or smart grid applications. The modular structure of the solution allows for easy adaptation to real-time monitoring systems and integration into energy-saving platforms.

By understanding patterns and forecasting needs, PowerPulse empowers users to make smarter, eco-conscious decisions that ultimately reduce cost, optimize energy use, and contribute to environmental sustainability.

**💡 Final Recommendations**

* Use Random Forest for accurate and reliable power consumption forecasting
* Shift usage of heavy appliances to non-peak hours
* Enable real-time monitoring for early detection of anomalies
* Calculate daily cost and emission estimates to raise awareness
* Retrain models periodically to keep up with evolving consumption behavior