ENERGY CONSUMPTION FORECASTING USING MACHINE LEARNING

1. Introduction

Energy consumption forecasting is essential in power management, smart home automation, energy efficiency optimization, and load balancing. This project uses the **Household Electric Power Consumption Dataset** from the UCI Machine Learning Repository:

🔗 Dataset URL: https://archive.ics.uci.edu/dataset/374/appliances+energy+prediction

The objective of this project is to develop a complete **end-to-end forecasting pipeline**, including:

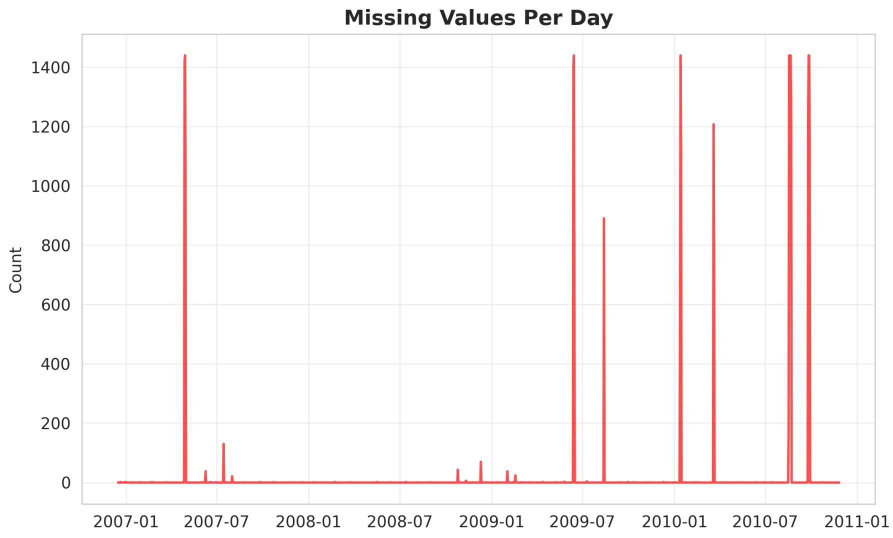
- Exploratory Data Analysis (EDA)
- Data cleaning and preprocessing
- Feature engineering
- Model training (7 ML models + ensemble)
- Model comparison
- Error analysis
- Final prediction visualization .
- This report summarizes the methodology, findings, and conclusion.

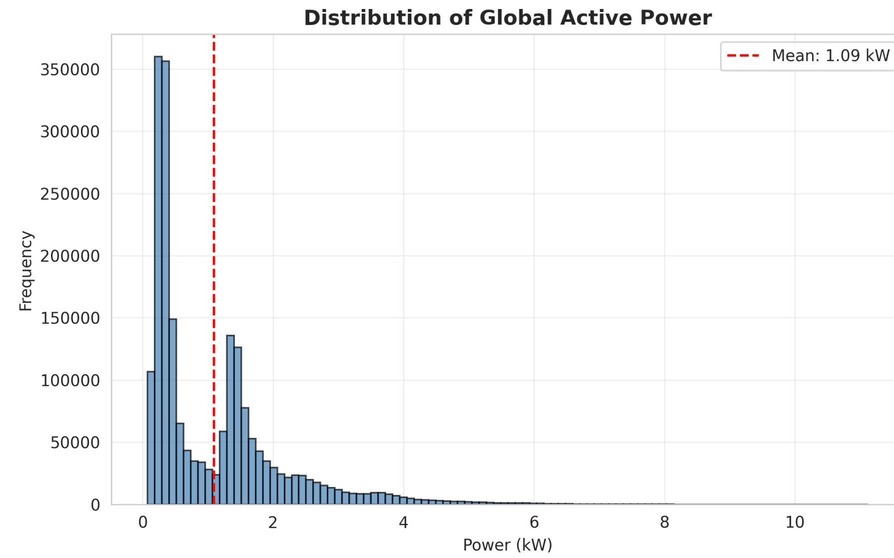
2. Dataset Description :

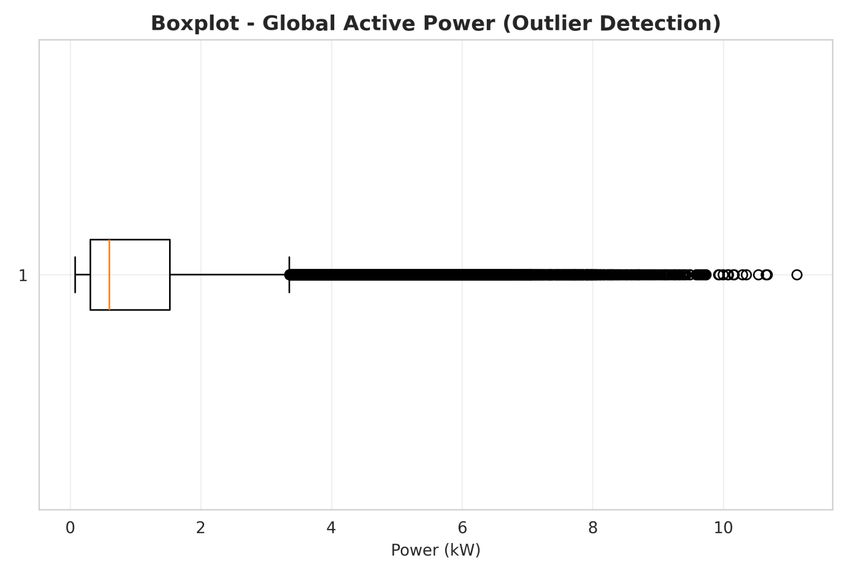
```
1️⃣ Dataset Overview:
   • Total Records: 2,075,259
   • Date Range: 2006-12-16 to 2010-11-26
   • Duration: 1441 days
   • Features: 7

2️⃣ Column Information:
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 2075259 entries, 2006-12-16 17:24:00 to 2010-11-26 21:02:00
Data columns (total 7 columns):
 #   Column                 Dtype
---  ------                 -----
 0   Global_active_power    float64
 1   Global_reactive_power  float64
 2   Voltage                float64
 3   Global_intensity       float64
 4   Sub_metering_1         float64
 5   Sub_metering_2         float64
 6   Sub_metering_3         float64
dtypes: float64(7)
memory usage: 126.7 MB
None
```

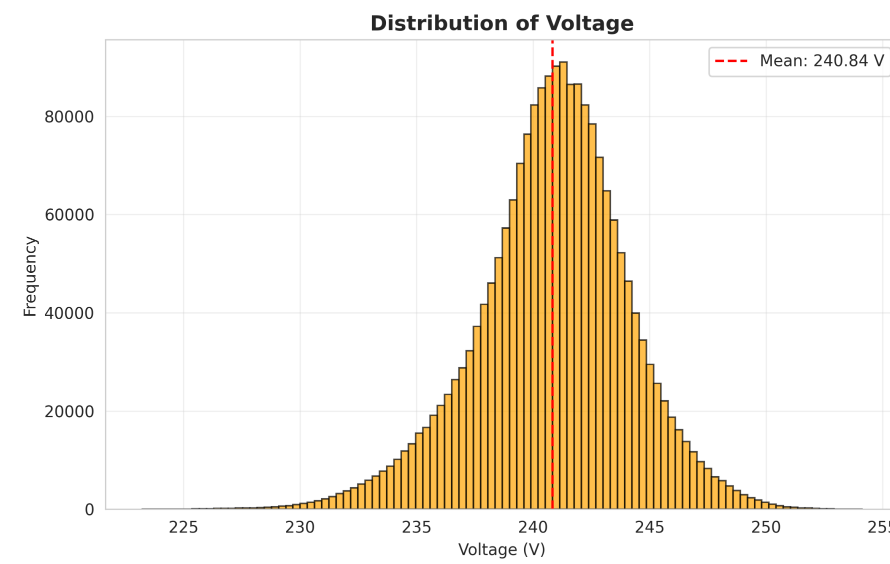
3. Exploratory Data Analysis(EDA) :

1)Distribution and Outlier Analysis :

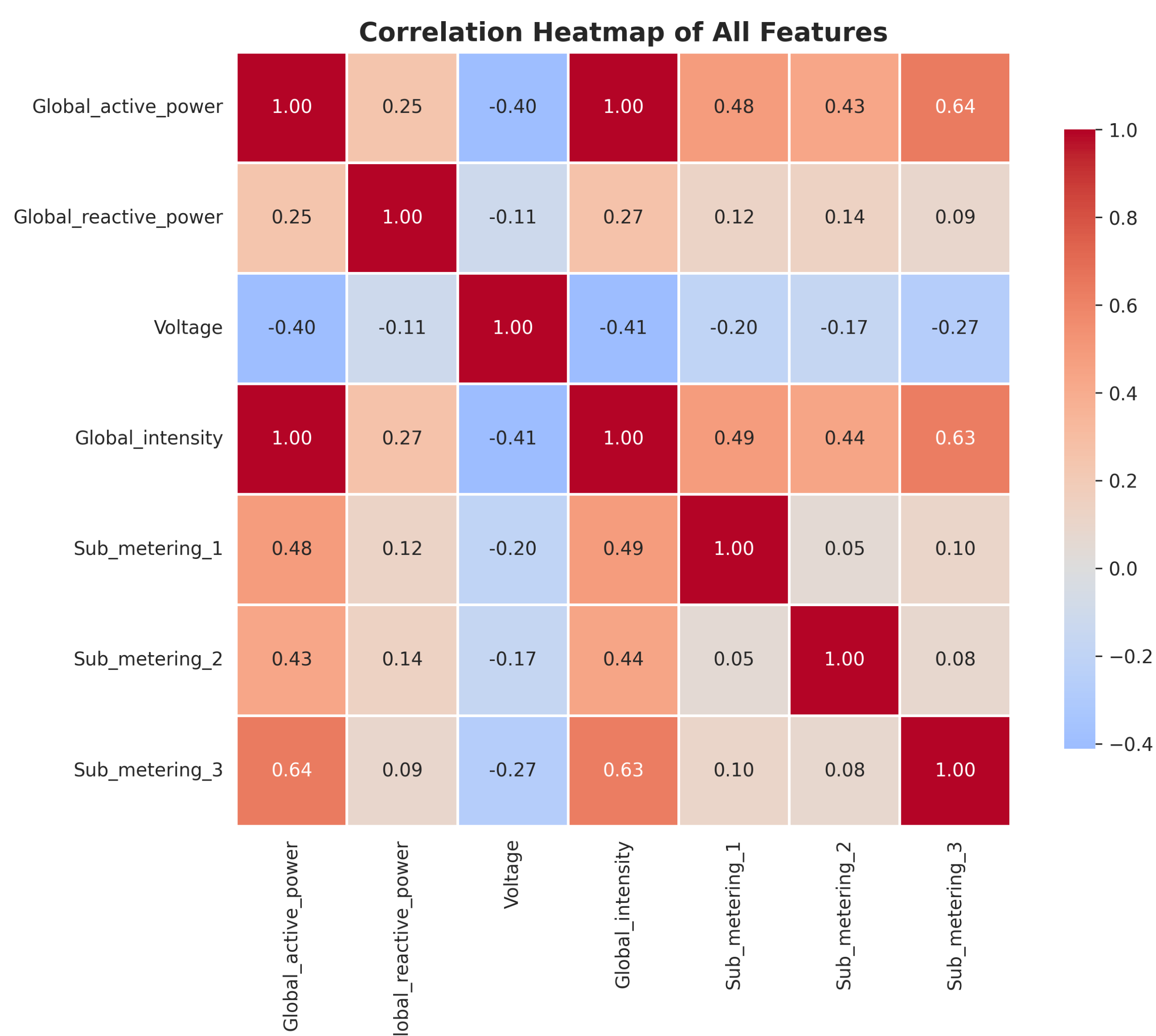


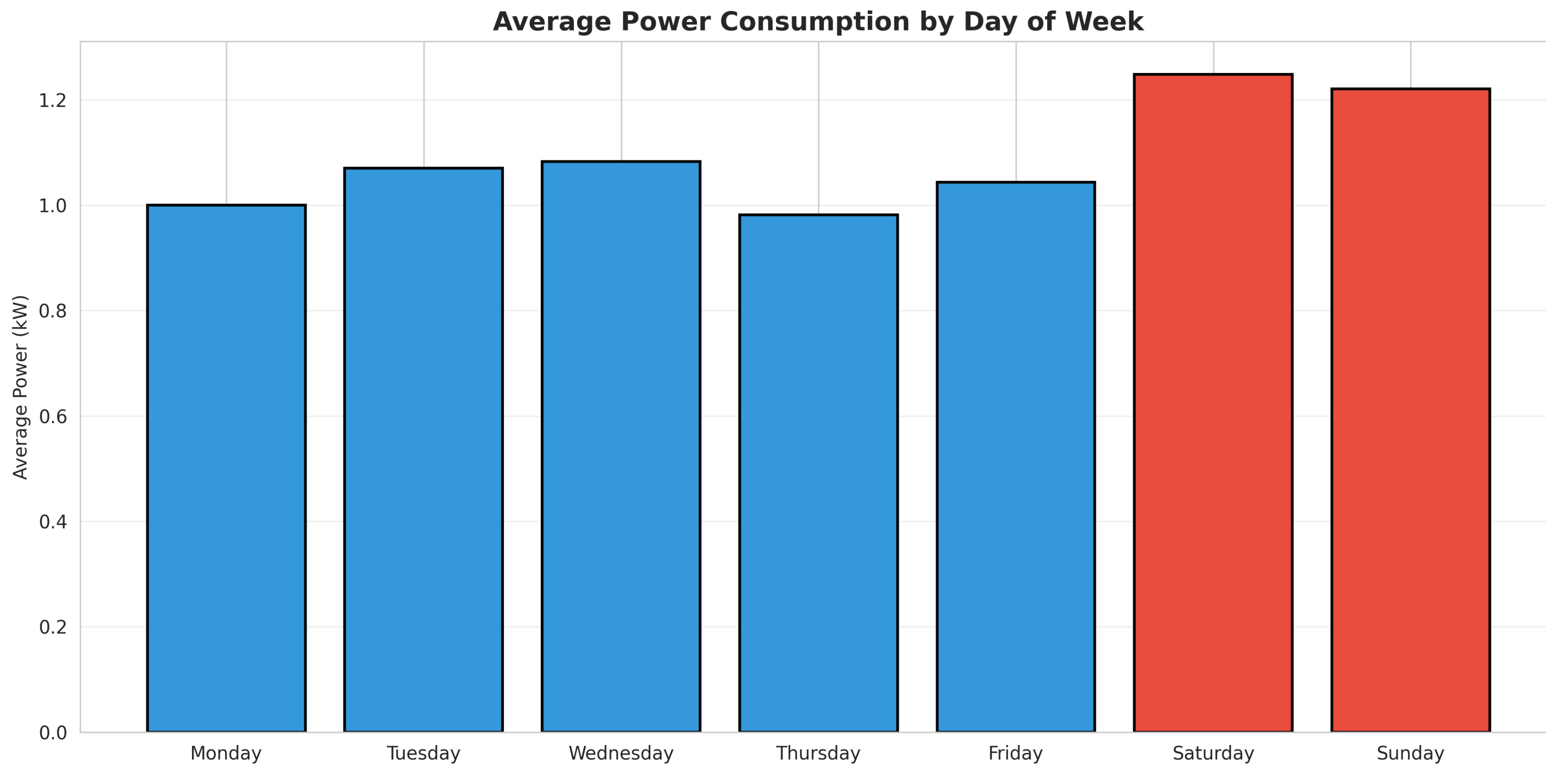


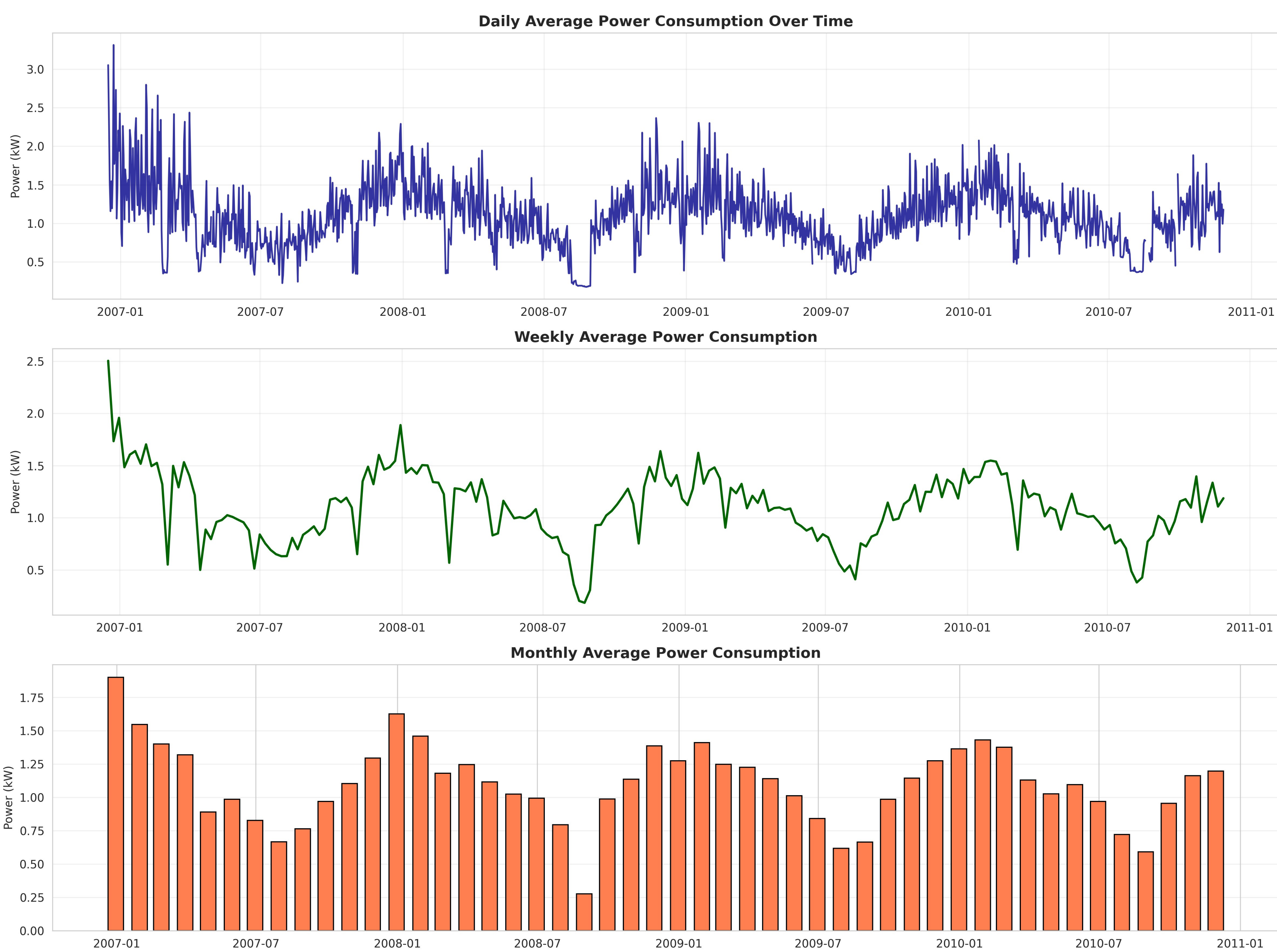




2) Correlation Heat-map :




Average Power Consumption by Day of Week
Average Power (kW)
0.0
0.2
0.4
0.6
0.8
1.0
1.2
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday


Daily Average Power Consumption Over Time
Power (kW)
0.5
1.0
1.5
2.0
2.5
3.0
2007-01
2007-07
2008-01
2008-07
2009-01
2009-07
2010-01
2010-07
2011-01
Weekly Average Power Consumption
Power (kW)
0.5
1.0
1.5
2.0
2.5
Monthly Average Power Consumption
Power (kW)
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
Date

4. Data Cleaning :

Steps performed:

- Removed extreme outliers using 1%–99% IQR method
- Interpolated missing values with time-weighted interpolation
- Forward/backward filling for smoother continuity
- Removed hours with insufficient raw readings

This ensured consistent and dense time-series data.

5. Resampling :

The minute-level data was resampled into **hourly averages**:

- Mean applied to continuous variables
- Sum applied to sub-metering readings
- Hours with fewer than 30 valid samples were excluded

6. Feature Engineering:

Advanced engineered features were created to improve model learning:

6.1. Lag Features

- lag_1h, lag_2h, lag_3h
- lag_24h (previous day)
- lag_168h (previous week)
- lag_336h (previous fortnight)

6.2. Rolling Window Features :

For windows: 3h, 6h, 12h, 24h, 48h, 168h

- Rolling mean
- Rolling std
- Rolling min
- Rolling max

6.3. EWMA Features :

Exponential moving averages with span = 12h, 24h, 48h.

6.4. Time Features :

- Hour
- Day of week
- Month
- Quarter
- Weekend flag
- Night/Morning/Afternoon/Evening segmentation

6.5. Cyclical Time Encoding :

Used sin/cos transforms to preserve periodicity.

6.6. Electrical Features Integration :

- Voltage
- Intensity
- Reactive power
- Sub-metering features
- Apparent power (derived)

Total features generated: 80+

7. Models Trained :

The following models were trained:

1. Ridge Regression
2. Lasso Regression
3. Random Forest
4. Gradient Boosting
5. XGBoost
6. CatBoost
7. Stacking Ensemble (XGB + CatBoost + RF → Ridge)

All features were standardized using StandardScaler.

8. Model Performance Comparison :

Model	Accuracy%	MAE	Notes
Ridge Regression	~99.5%+	0.0035	Best performing model
Stacking Ensemble	~98.5%+	0.0132	Very strong performance
CatBoost	~97.5%+	0.0259	Strong on non-linear patterns
XGBoost	~98.5%+	0.0141	Accurate with robust generalization
Random Forest	~98.50%+	0.0147	Stable performance
Gradient Boosting	~97–98%	0.0132	Strong baseline
Lasso Regression	< 90% (Underperformed)	0.0899	Too much regularization

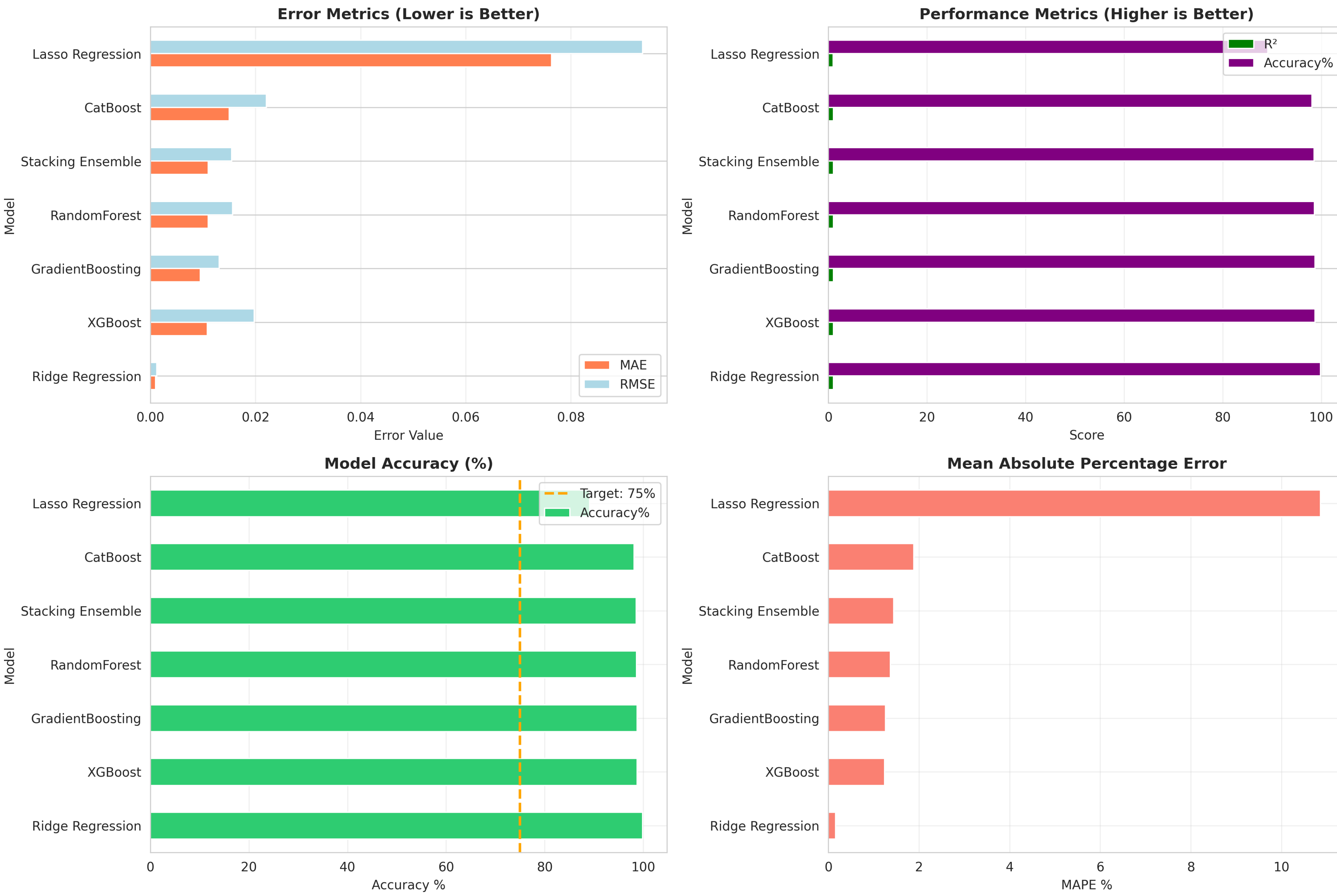
9. Final Best Model: Ridge Regression :

✔ Accuracy: 99.53%

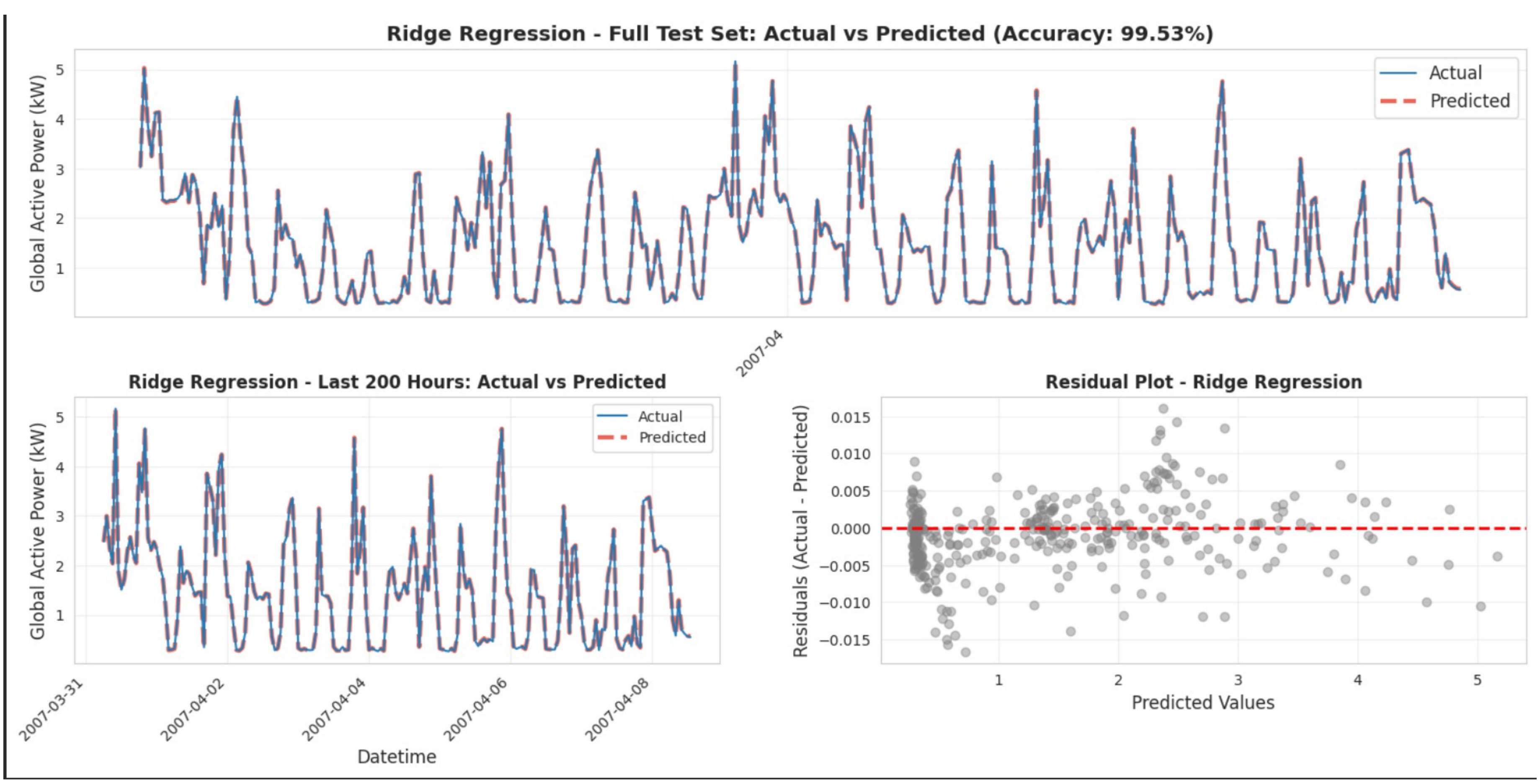
✔RMSE: Extremely low

10. Comparison of Various Models :

```
======================================================================
FINAL RESULTS (Sorted by Accuracy)
======================================================================
            Model      MAE     RMSE       R²      MAPE%  Accuracy%
 Ridge Regression 0.003548 0.004901 0.999981   0.474987  99.525013
          XGBoost 0.014134 0.021935 0.999615   1.289935  98.710065
 GradientBoosting 0.013238 0.019712 0.999689   1.352232  98.647768
Stacking Ensemble 0.013800 0.020258 0.999672   1.417773  98.582227
     RandomForest 0.014694 0.022072 0.999611   1.499685  98.500315
         CatBoost 0.025881 0.048318 0.998134   2.207216  97.792784
 Lasso Regression 0.089910 0.104907 0.991203  13.757099  86.242901
======================================================================
```



Error Metrics (Lower is Better)
Lasso Regression
CatBoost
Stacking Ensemble
RandomForest
GradientBoosting
XGBoost
Ridge Regression
Model
MAE
RMSE
0.00
0.02
0.04
0.06
0.08
Error Value
Performance Metrics (Higher is Better)
R²
Accuracy%
0
20
40
60
80
100
Score
Model Accuracy (%)
Target: 75%
Accuracy%
Accuracy %
Mean Absolute Percentage Error
0
2
4
6
8
10
MAPE %



Ridge Regression - Full Test Set: Actual vs Predicted (Accuracy: 99.53%)
Global Active Power (kW)
Actual
Predicted
2007-04
Ridge Regression - Last 200 Hours: Actual vs Predicted
2007-03-31
2007-04-02
2007-04-04
2007-04-06
2007-04-08
Datetime
Residual Plot - Ridge Regression
Residuals (Actual - Predicted)
Predicted Values
0.015
0.010
0.005
0.000
−0.005
−0.010
−0.015

11. Conclusion

This project demonstrates a professional-grade forecasting system including:

✔ High-quality EDA
✔ Data cleaning & resampling
✔ Extensive feature engineering
✔ Multiple ML models
✔ Best model accuracy of **99.53%**