Energy Consumption Forecasting and Uncertainty Quantification

Besan Musallam

August 5, 2025

1 Introduction

This project was my first hands-on experience with time series forecasting. Unlike previous projects where I worked with static data like images or audio, this task was all about capturing patterns over time. My main goal wasn't just to predict energy consumption but also to estimate how certain the model is about its predictions, which is super important in the energy sector for planning and risk assessment.

2 Objectives

- Prepare and clean a high-frequency time series dataset with missing and noisy data.
- Analyze temporal patterns like seasonality and usage habits.
- Engineer useful features based on time and external data (e.g., holidays).
- Build multiple models, including probabilistic ones for uncertainty estimation.
- Evaluate models on accuracy and how well they capture uncertainty.
- Translate the forecasts into business-relevant insights.

3 Dataset Overview

I used the **Individual Household Electric Power Consumption** dataset from the UCI Repository, which includes over 2 million minute-level readings between 2006 and 2010.

3.1 Dataset Link

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

3.2 Main Features

- **Global Active Power (kW)** – The main target variable.
- **Global Reactive Power (kVAR)** – Power oscillations.
- **Voltage (V)** – Voltage readings per minute.
- **Sub-meterings** – Appliance-specific energy usage.

4 Data Preparation and Exploration

4.1 Cleaning the Data

Here's what I did to get the data in shape:

- Merged Date and Time columns into a single datetime index.
- Removed rows with invalid entries (e.g., '?').
- Dropped the **Global Intensity** column since it was redundant.
- Applied log transformations to highly skewed features.

4.2 Initial Insights

Through EDA, I noticed:

- Huge outliers in **Global Active Power** — values above 10kW were unrealistic.
- Sub-meterings were mostly zeros, indicating they're only active occasionally.
- Voltage readings had normal-like distribution but with minor fluctuations.

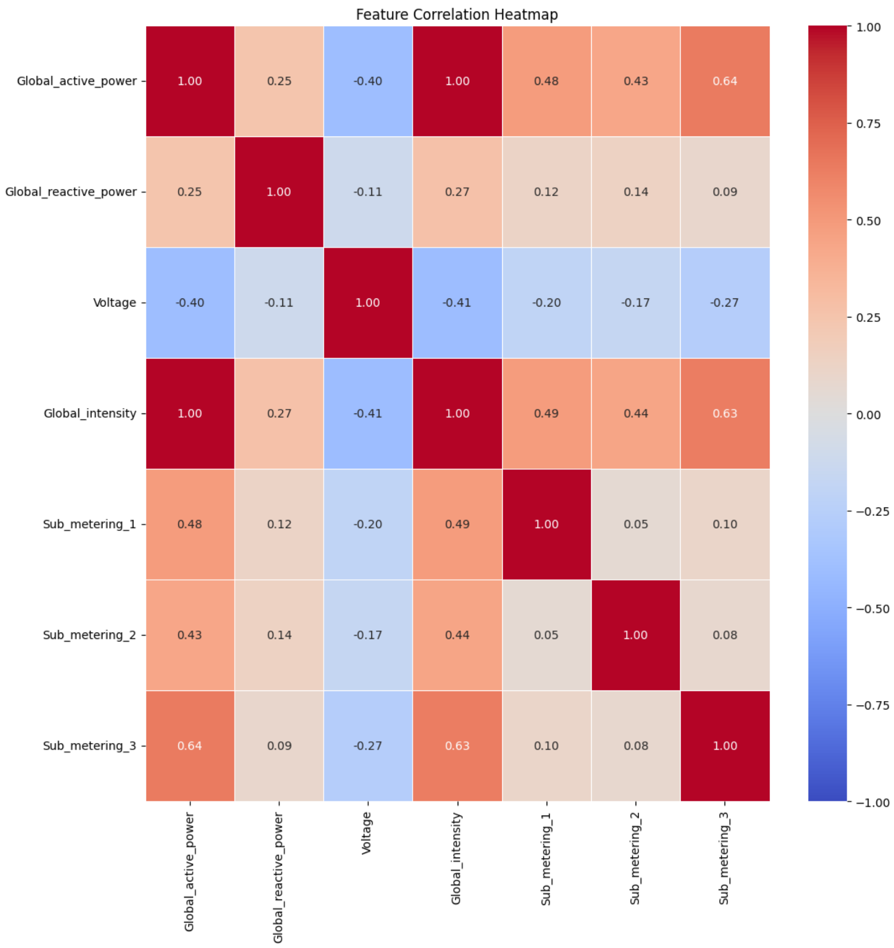

Figure 1: Correlation matrix showing relationships between main features.

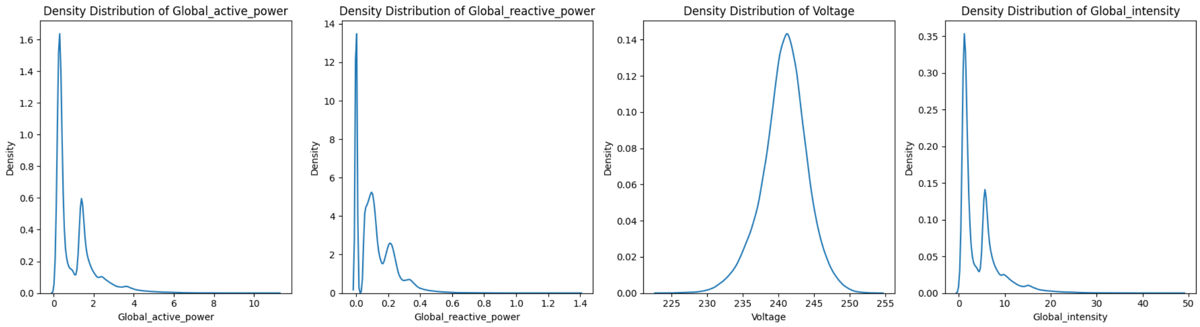

Figure 2: Feature distributions after applying log transformations.

4.3 Seasonality Patterns

Aggregating data to daily averages revealed clear patterns:

- Winter months showed peak consumption (likely heating).
- Sharp drops in August (probably due to vacations).
- Weekdays had a zig-zag pattern compared to weekends.

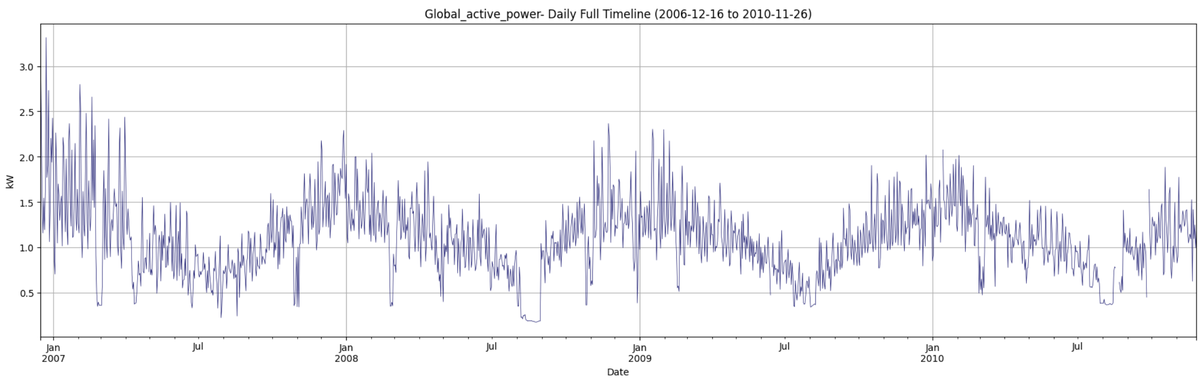

Figure 3: Daily average consumption highlighting seasonal effects and weekday-weekend cycles.

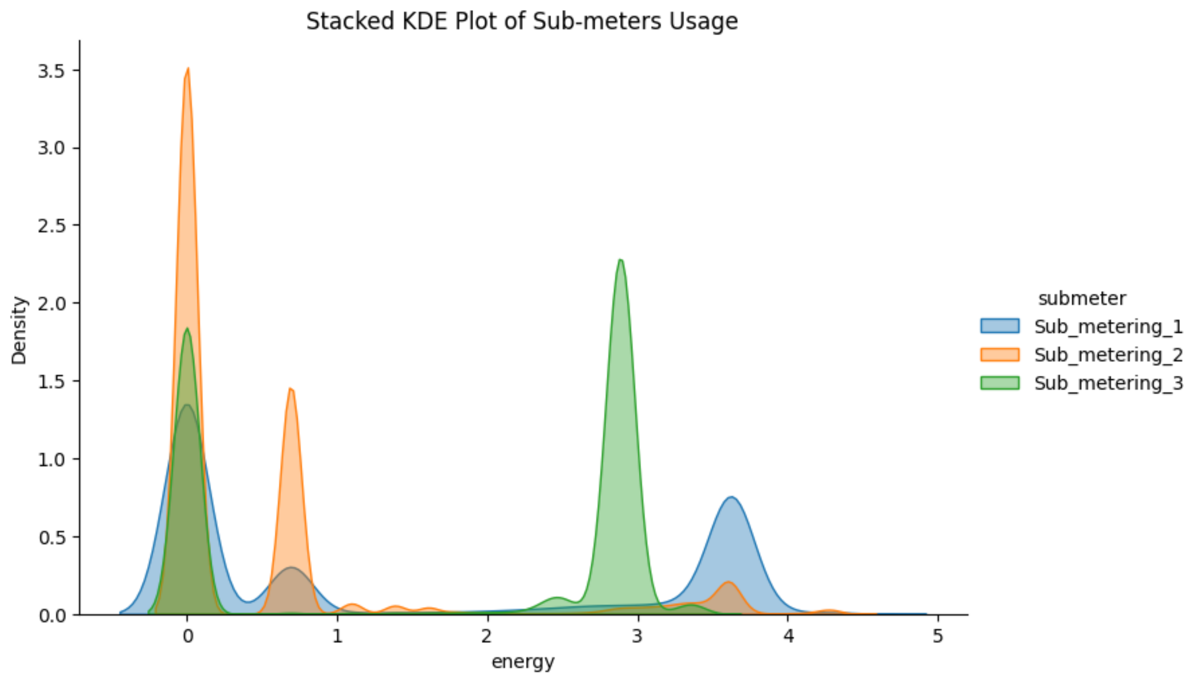

Figure 4: Sub-meter usage showing how infrequent certain appliances are used.

5 Feature Engineering

To make models smarter, I added:

- Time-of-day tags (morning, afternoon, night).
- Day-type (weekday or weekend).
- Month and quarter indicators.
- Lag features to capture past consumption behavior.
- Rolling statistics (mean, std dev) over short windows.
- External data like public holidays.

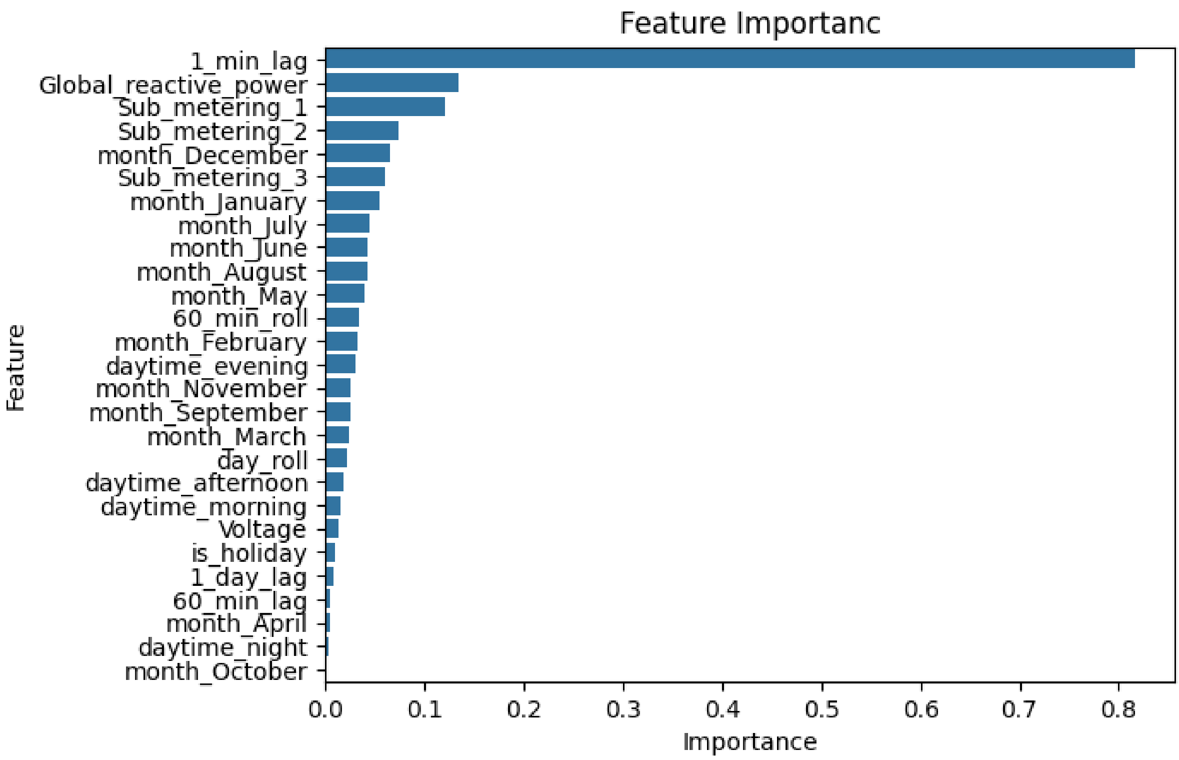

Figure 5: Feature importance after engineering. Time-related features ranked highest.

6 Modeling and Evaluation

6.1 Train-Test Split

I used a sequential split to avoid data leakage:

- Train: 80%
- Validation: 10%
- Test: 10%

6.2 Models Built

- **Linear Regression** — Simple baseline.
- **XGBoost Regressor** — To handle non-linear patterns.
- **Bidirectional LSTM** — For temporal dependencies.

```
def lstm_model(input_shape):
    model = Sequential()
    model.add(Bidirectional(LSTM(64, return_sequences=True), input_shape=in
    model.add(Dropout(0.3))
    model.add(Bidirectional(LSTM(32)))
    model.add(Dropout(0.4))
```

```
model.add(Dense(1))
model.compile(optimizer='adam', loss='mse')
return model
```

6.3 Performance Metrics

Table 1: Performance Comparison

Model	R^2	MAE	MSE
Linear Regression	0.9458	0.0996	0.0403
XGBoost	0.9686	0.0632	0.0233
LSTM	0.9657	0.0761	0.0255
Ensemble	**0.9667**	**0.0721**	**0.0248**

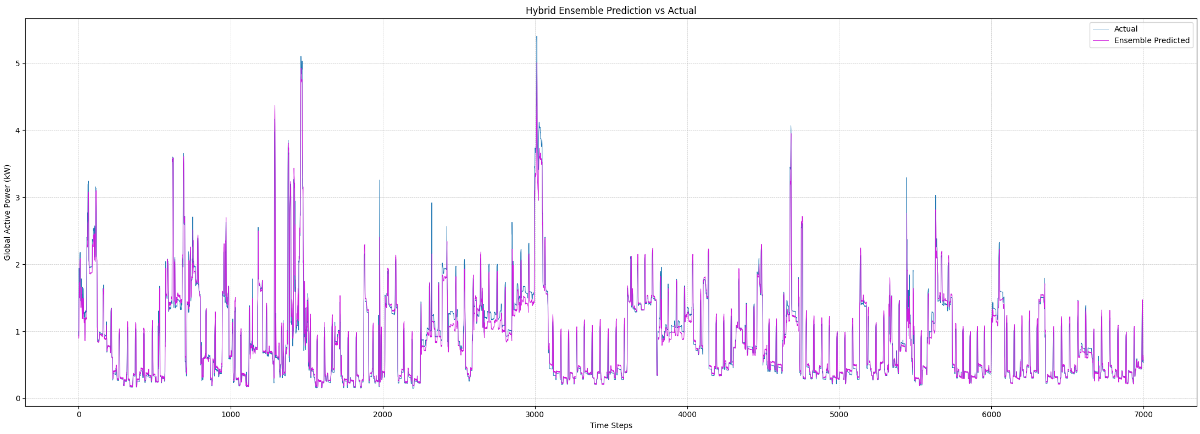

Figure 6: Predicted vs Actual Consumption (Ensemble Model).

7 Uncertainty Quantification

For estimating forecast reliability, I calculated:

- **Prediction Interval Coverage Probability (PICP)** = 80%
- **Mean Prediction Interval Width (MPIW)** = 0.1617 kW

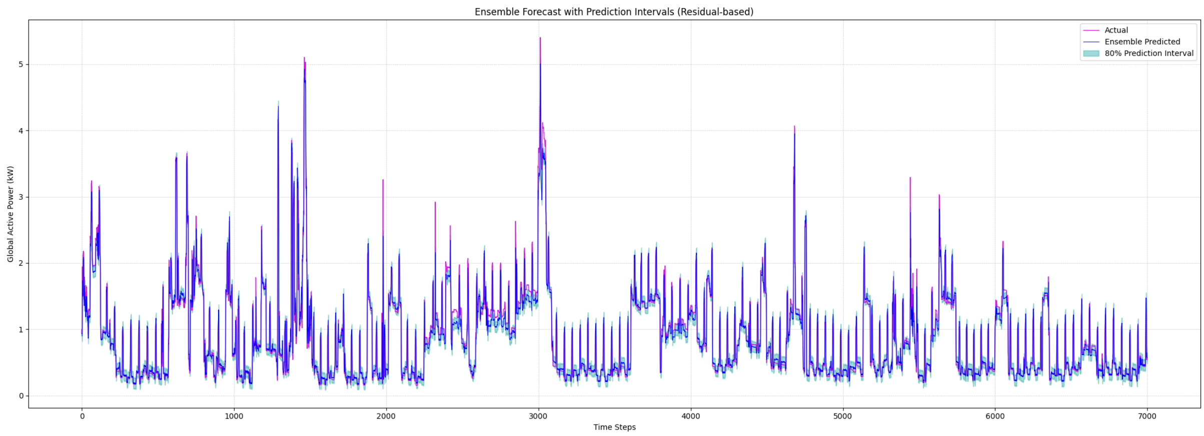

Figure 7: Uncertainty intervals around predictions.

8 Insights and Challenges

- Feature engineering was the most impactful step due to hardware limits.
- The ensemble model provided the best balance of accuracy and stability.
- Lag/rolling features compensated for not using full sliding windows.
- External factors like holidays noticeably impacted consumption.

9 Business Relevance

- **Grid Planning**: Anticipate high-load periods.
- **Risk Management**: Use uncertainty bands for better decisions.
- **Dynamic Pricing**: Adjust tariffs based on forecast confidence.

10 Conclusion

This project was a deep dive into time series forecasting and uncertainty quantification. I combined classical models with deep learning to build a practical forecasting pipeline. Handling large datasets, crafting relevant features, and ensuring uncertainty estimates were reliable gave me a solid foundation in data-driven decision-making for energy consumption scenarios.